МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГУ»)

Факультет прикладной математики, информатики и механики
Кафедра прикладной механики и информатики

Отчет по лабораторной работе №3

на тему:

«Приближенное решение краевой задачи для системы обыкновенных дифференциальных уравнений методом стрельбы»

Выполнил: студент 3 к. 1 гр. ПМИ в.о.

Бедарев Анатолий Андреевич

Проверил: к.ф-м.н, доц.

Гудович Николай Николаевич

Воронеж – 2018

Содержание

Постановка задачи

1. **Вар.2**. Написать программу приближенного решения краевой задачи с выводом графиков приближенных решений на экран. Использовать метод приближенного решения:

- исправленный метод Эйлера (старт);

- явный метод Адамса 2 порядка точности.

Решить задачу методом стрельбы.

Исходные данные:

- отрезок [a, b];

- функции y'(x) и z'(x):

$$y' = \frac{y^2}{z - x}, \ (1)$$

$$z' = y + 1 \ (2)$$

- краевые условия:

$$y(0) = 2, \ (3)$$

$$z\left(\frac{1}{2}\right) = \frac{1 + 4e}{2} \ (4)$$

- точное решение:

$$y = 2e^{x}, \ (5)$$

$$z = x + 2e^{2x} \ (6)$$

Указания к выполнению лабораторной работы

Метод стрельбы (краевая задача) — численный метод, заключающийся в сведении краевой задачи к некоторой задаче Коши для той же системы дифференциальных уравнений. Суть: первое решение при последовательном изменении аргумента и повторении вычислений становится точнее.

Описание метода. Рассматривается задача для системы двух уравнений первого порядка с краевыми условиями общего вида:

система

$$u'(x) = f(x, u, v)$$

$$v'(x) = g(x, u, v)$$

граничные условия

$$a \leqslant x \leqslant b$$

$$\varphi[u(a), v(a)] = 0$$

$$\psi[u(b), v(b)] = 0$$

Ход выполнения работы

Выполнение лабораторной работы проводилось с использованием среды технического моделирования MatLab R2018a.
Для выполнения поставленных задач было разработано несколько программных модулей (см. Приложение).

На первом этапе в среде MatLab реализован алгоритм приближения исправленным методом Эйлера в виде отдельных модулей:

- *EilerCorrectedMethod.m* – модуль, реализующий приближение исправленным методом Эйлера;
- *F1.m* – модуль, реализующий вычисление функции y'(x) по ф.(1);
- *F2.m* – модуль, реализующий вычисление функции z'(x) по ф.(2);
- *F4.m* – модуль, реализующий вычисление точного решения по ф.(5) и ф.(6).

На втором этапе разработан алгоритм приближения явным методом Адамса 2 порядка точности, реализованный в модуле *Adams2ExplicitMethod.m*

На третьем этапе разработан алгоритм, реализующий метод стрельбы для решения краевой задачи (сведение краевой задачи к некоторой задаче Коши для той же системы дифференциальных уравнений), выполненный в модуле *ShootingMethod.m*

На четвертом этапе в среде MatLab разработаны дополнительные модули *plotAdams2ExplicitMethod.m* и *plotFrontierTask.m* для формирования графиков.

На пятом этапе выполнен численный эксперимент. Результаты представлены на рис. 1 – 4.

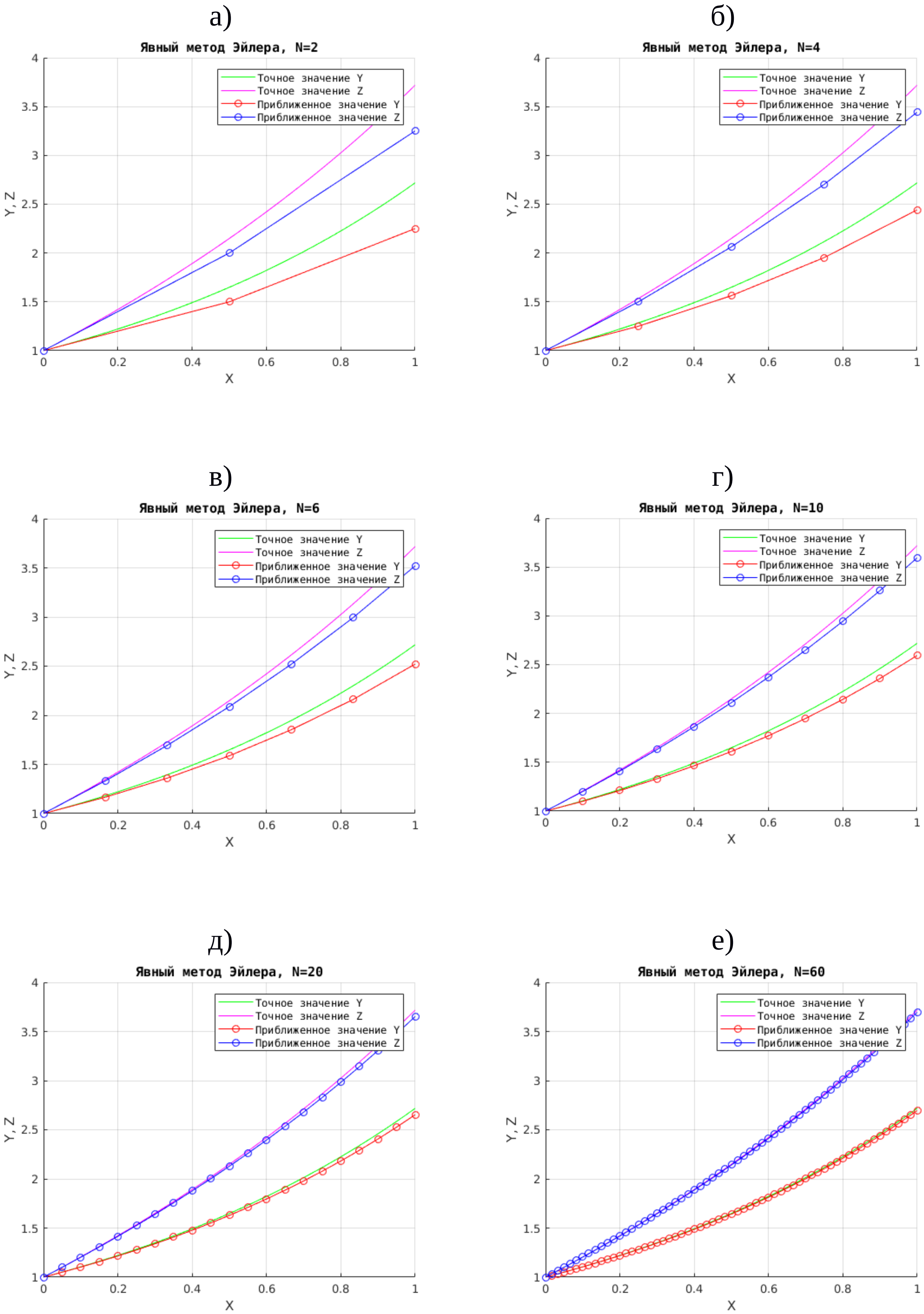

Рисунок 1 – Сравнение точности приближения при различных значениях числа частичных отрезков разбиения в явном методе Эйлера (метод 1-го порядка)

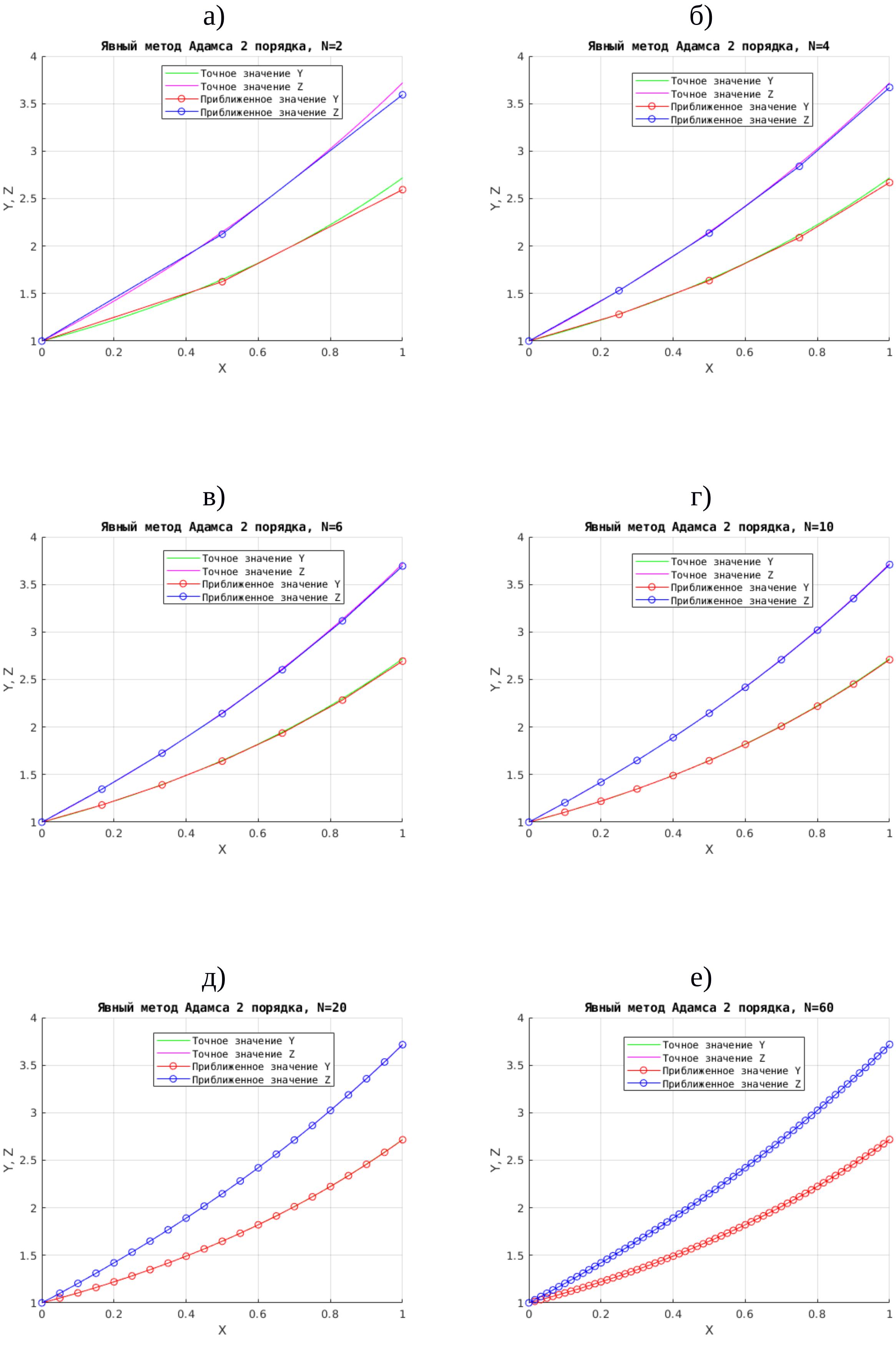

Рисунок 2 – Сравнение точности приближения при различных значениях числа частичных отрезков разбиения в явном методе Адамса 2-го порядка

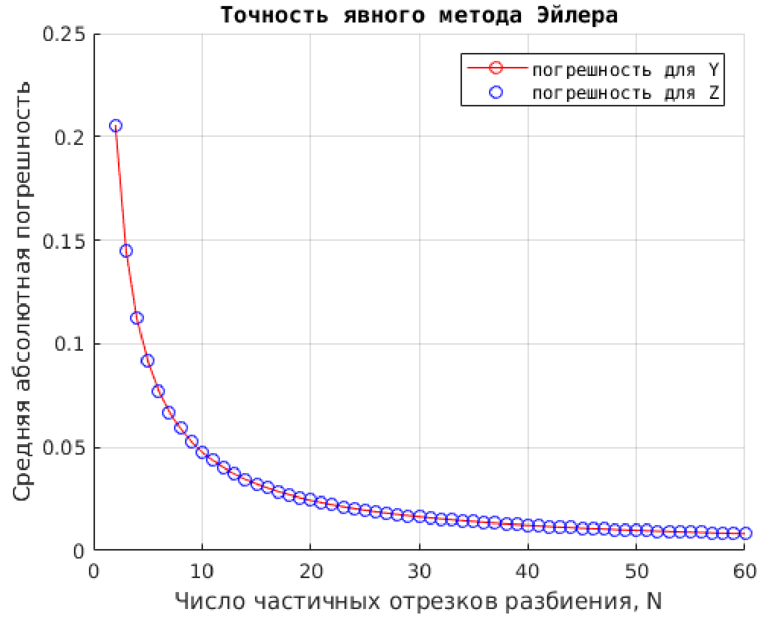

Рисунок 3 – Зависимость средней ошибки приближения
от числа частичных отрезков разбиения, явный метод Эйлера

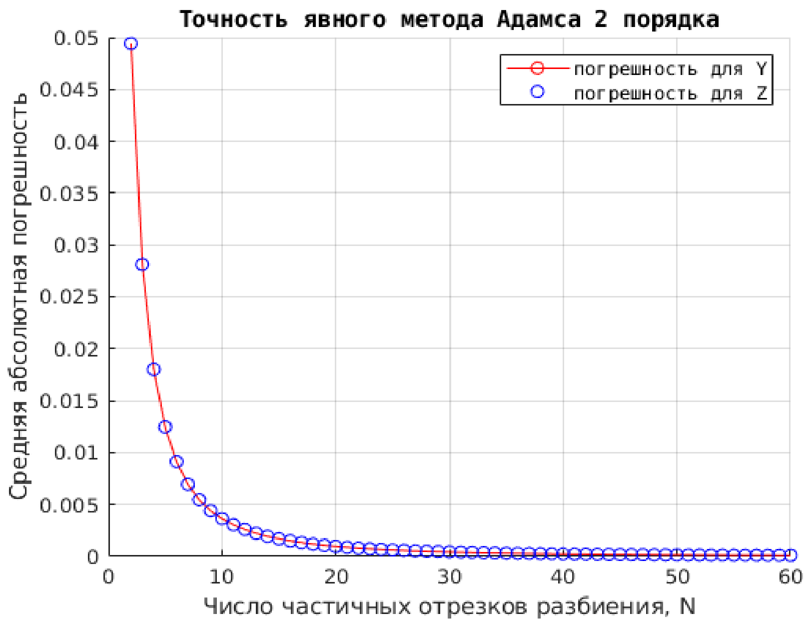

Рисунок 4 – Зависимость средней ошибки приближения
от числа частичных отрезков разбиения, явный метод Адамса 2 порядка

Как видно из рис.4, точность явного метода Адамса даже при достаточно малых значениях числа частичных отрезков разбиения на порядок выше точности явного метода Эйлера и обладает большей скоростью сходимости.

На рис. 5 показаны результаты применения метода стрельбы для решения краевой задачи:

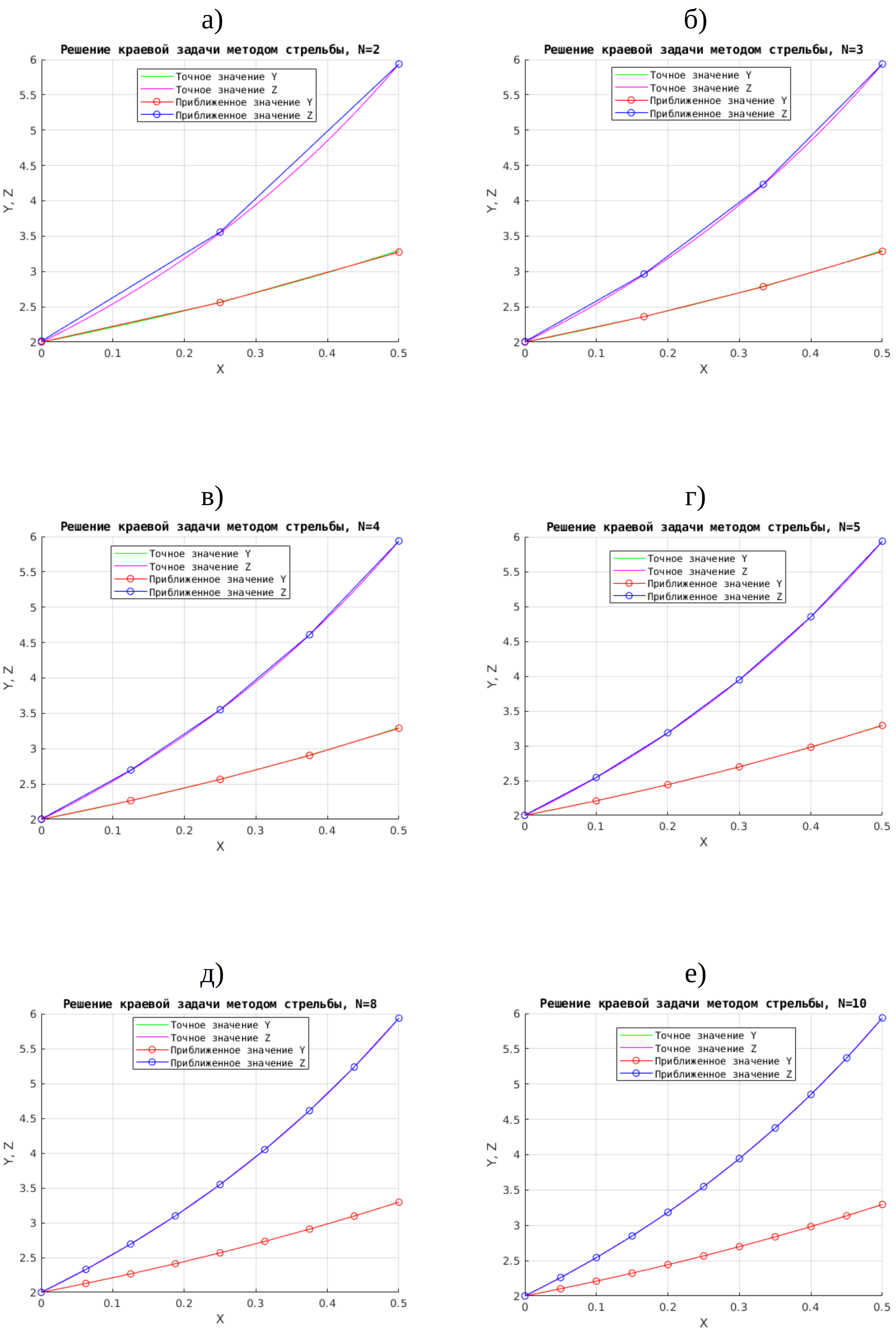

Рисунок 5 – Решение краевой задачи методом стрельбы

Как видно из рис. 5, найденное решение удовлетворяет левому и правому краевым условиям, при этом тем точнее приближаясь к точному решению, чем больше количество частичных отрезков разбиения N. Отметим, что уже при N=5 точное и приближенное решения практически совпадают, что говорит о высокой точности используемых численных методов приближения.

Выводы по работе

1. В ходе лабораторной работе рассмотрен способ приближенного решения краевой задачи для системы обыкновенных дифференциальных уравнений методом стрельбы с использование метода Эйлера первого порядка (старт) и метода Адамса второго порядка (марш), разработаны функциональные модули для системы MatLab, с помощью которых можно проводить такое приближенное решение и строить графики.

2. В результате численного эксперимента установлено, что приближенное решение краевой задачи обладает сходимостью, которая тем выше, чем больше число частичных отрезков разбиения. Для заданных условий N=5 можно считать достаточным.

3. Использование метода стрельбы (метода половинного деления) обеспечивает быструю сходимость к решению, но осложняет решение необходимостью ручного выбора начальных точек. От этого выбора зависит, будет ли возможно приближение, и в ряде случаев приходится проводить дополнительные исследования поведения функции с целью их поиска.

4. Установлено, что метод Адамса второго порядка по эффективности превосходит метод Эйлера того же порядка точности. Это обусловлено тем, что, в отличие от одношаговых методов Рунге-Кутта, он использует для вычисления очередного значения искомого решения не одно, а несколько значений, которые уже вычислены в предыдущих точках.

Список литературы

1. *Кетков, Ю.Л. и др.* Matlab 7: программирование, численные методы / Ю.Л. Кетков, А.Ю. Кетков, М.М. Шульц, - СпБ.: БХВ-Петербург, 2005 г. – 752 с.

Приложение (листинг)

Модуль MatLab **F1.m**

```
function F1=F1(X, Y, Z)

F1 = Y.^2./(Z-X);

end
```

Модуль MatLab **F2.m**

```
function F2=F2(X, Y, Z) % тут X и Z лишь в качестве заглушки

F2 = Y+1;

end
```

Модуль MatLab **F4.m**

```
function [Y, Z]=F4(X)

% Точное решение

Y = exp(X);

Z = X + exp(X);

end
```

Модуль MatLab **EilerCorrectedMethod.m**

```
function [X, Y, Z] = EilerCorrectedMethod(a, b, N, stPy, stPz)

% Функция для приближения исправленным методом Эйлера

% a - начало отрезка

% b - конец отрезка

% N - количество отрезков разбиения

% stPy, stPz - начальные условия, то есть значения Y и Z в точке X=a

h = abs(b-a)/N; % Расстояние между узлами

X = a:h:b; % Формируем вектор X узловых точек

Y = X * 0; % Формируем пока еще пустой вектор Y

Z = Y; % Формируем пока пустой вектор Z
```

```
% Подставляем начальные условия:

Y(1) = stPy;

Z(1) = stPz;

% Начинаем обходить последовательно все точки и вычисляем значения на

% текущем шаге исходя из данных предыдущего шага:

for i = 2 : N+1

Ya = Y(i-1) + h * F1(X(i-1), Y(i-1), Z(i-1)); % явное

Za = Z(i-1) + h * F2(X(i-1), Y(i-1), Z(i-1)); % явное

Y(i) = Y(i-1) + h/2 * (F1(X(i-1), Y(i-1), Z(i-1)) + F1(X(i), Ya, Za));

Z(i) = Z(i-1) + h/2 * (F2(X(i-1), Y(i-1), Z(i-1)) + F2(X(i), Ya, Za));

end

end
```

Модуль MatLab **Adams2ExplicitMethod.m**

```
function [X, Y, Z] = Adams2ExplicitMethod(a, b, N, stPy, stPz)

% Функция для приближения явным методом Адамса 2 порядка (марш) со стартом

% исправленным методом Эйлера

% a - начало отрезка

% b - конец отрезка

% N - количество отрезков разбиения

% stPy, stPz - начальные условия, то есть значения Y и Z в точке X=a

h = abs(b-a)/N; % Расстояние между узлами

% Для старта используем исправленный метод Эйлера:

[X, Y, Z] = EilerCorrectedMethod(a, b, N, stPy, stPz);

% Но нам нужны только первые две точки оттуда, все остальное перезапишем

% вычислениями марша явного метода Адамса:

% Начинаем обходить последовательно все точки и вычисляем значения на

% текущем шаге исходя из данных двух предыдущих шагов:

for i = 2 : N

Y(i+1) = Y(i) + h * ( 3/2*F1(X(i), Y(i), Z(i)) - 1/2*F1(X(i-1), Y(i-1), Z(i-1)) );

Z(i+1) = Z(i) + h * ( 3/2*F2(X(i), Y(i), Z(i)) - 1/2*F2(X(i-1), Y(i-1), Z(i-1)) );

end
```

```
end
```

Модуль MatLab **deltaAdams.m**

```
function d = deltaAdams(Y1, Y2)
% Функция вычисления погрешности между значениями функции и значениями
% приближения
n = length(Y1); % Количество узловых точек
D = Y1 * 0; % Матрица разности
for i = 1 : n
D(i) = Y1(i) - Y2(i);
end
d = 0;
for i = 1 : n
d = d + abs(D(i));
end
d = d / n;
end
```

Модуль MatLab **plotDeltaAdams.m**

```
function plotDeltaAdams(a, b, N_start, N_end, stPy, stPz)
% Функция печати графика зависимости погрешности приближения от числа
% узловых точек
% a - начало отрезка
% b - конец отрезка
% N_start - начальное количество отрезков разбиения
% N_end - конечное количество отрезков разбиения
% stPy, stPz - начальные условия, то есть значения Y и Z в точке X=a
% 1 Создаем новое окно для графика и подписываем оси
figure;
xlabel('Число частичных отрезков разбиения, N');
ylabel('Средняя абсолютная погрешность');
```

```
grid on; hold on;
% 2 Выполняем приближение
j = 1;
deltaY = N_start : N_end;
deltaY = deltaY * 0;
deltaZ = deltaY;
for N = N_start : N_end
[X, Y, Z] = Adams2ExplicitMethod(a, b, N, stPy, stPz);
[Yt, Zt] = F4(X);
deltaY(j) = deltaAdams(Yt, Y);
deltaZ(j) = deltaAdams(Zt, Z);
j = j + 1;
end
titleValue = 'Точность явного метода Адамса 2 порядка';
X = N_start : N_end;
% 3 Печатаем графики погрешностей:
plot(X, deltaY, 'r-o');
plot(X, deltaZ, 'bo');
% 4 Подписываем легенду
title(titleValue, 'FontName', 'Courier');
h1 = legend('погрешность для Y', 'погрешность для Z');
set(h1, 'FontName', 'Courier');

end
```

Модуль MatLab **ShootingMethod.m**

```
function [stPz, k, enPz_rs] = ShootingMethod(a, b, N, stPy, enPz)
% Метод стрельбы для решения краевой задачи (сведение краевой задачи к
% некоторой задаче Коши для той же системы дифференциальных уравнений)
% a - начало отрезка
% b - конец отрезка
% N - количество отрезков разбиения
```

```
% stPy, enPz - краевые условия, то есть значения Y в точке X=a и Z в точке
% X=b
stPz = 0; % начальное приближение к stPz
stPz_l = 0.1; % начальное приближение к stPz слева
stPz_r = 80; % начальное приближение к stPz справа
enPz_rs = 0.5;
k = 0; % счетчик шагов
while true % пока шаг превышет погрешность, считаем дальше
stPz = (stPz_l+stPz_r)/2;
[Xl, Yl, Zl] = Adams2ExplicitMethod(a, b, N, stPy, stPz_l);
[Xr, Yr, Zr] = Adams2ExplicitMethod(a, b, N, stPy, stPz_r);
[Xo, Yo, Zo] = Adams2ExplicitMethod(a, b, N, stPy, stPz);
k = k + 1;
if Zl(N+1) > enPz
stPz_l = stPz_l + 0.1;
continue;
end
if enPz > Zr(N+1)
stPz_l = stPz_l - 0.1;
continue;
end
if Zl(N+1) < enPz & enPz < Zo(N+1)
stPz_r = stPz;
enPz_rs = Zo(N+1);
continue;
end
if Zo(N+1) < enPz & enPz < Zr(N+1)
stPz_l = stPz;
enPz_rs = Zo(N+1);
continue;
end
break;
```

```
end

end
```

Модуль MatLab **plotAdams2ExplicitMethod.m**

```
function plotAdams2ExplicitMethod(a, b, N, stPy, stPz)

% Основная функция, вычисляет приближение к функциям по

% методу Адамса второго порядка и строит сравнительные графики для исходной функции и

% приближающей функции, с нанесением узловых точек

% a - начало отрезка

% b - конец отрезка

% N - количество отрезков разбиения

% stPy, stPz - начальные условия, то есть значения Y и Z в точке X=a

% 1 Создаем новое окно для графика и подписываем оси

figure;

xlabel('X');

ylabel('Y, Z');

hold on;

grid on;

% 2 Находим приближающие значения метдом Эйлера:

X = zeros(1);

Y = X;

Z = X;

[X, Y, Z] = Adams2ExplicitMethod(a, b, N, stPy, stPz);

titleValue = strcat('Явный метод Адамса 2 порядка, N=', int2str(N));

% 3 Считаем и печатаем график точных функций:

X1 = a:0.001:b;

[Yt, Zt] = F4(X1);

plot(X1, Yt, 'green');

plot(X1, Zt, 'm');

% 4 Печатаем графики приближений:

plot (X, Y, 'r-o');
```

```
plot (X, Z, 'b-o');

% 5 Подписываем легенду

title(titleValue, 'FontName', 'Courier');

h1 = legend('Точное значение Y', 'Точное значение Z', 'Приближенное значение Y', 'Приближенное значение Z');

set(h1, 'FontName', 'Courier');

end
```

Модуль MatLab **plotFrontierTask.m**

```
function plotFrontierTask(a, b, N, stPy, enPz)

% Основная функция

% a - начало отрезка

% b - конец отрезка

% N - количество отрезков разбиения

% stPy, stPz - начальные условия, то есть значения Y и Z в точке X=a,

% type - вариант метода Эйлера (0 - если явный, 1 - неявный)

% 1 Создаем новое окно для графика и подписываем оси

figure;

xlabel('X');

ylabel('Y, Z');

hold on;

grid on;

% 2 Находим приближающие значения метдом Эйлера:

X = zeros(1);

Y = X;

Z = X;

[stPz_o, k, enPz_s] = ShootingMethod2(a, b, N, stPy, enPz);

[X, Y, Z] = Adams2ExplicitMethod(a, b, N, stPy, stPz_o);

titleValue = strcat('Решение краевой задачи методом стрельбы, N=', int2str(N));

% 3 Считаем и печатаем график точных функций:

X1 = a:0.001:b;

[Yt, Zt] = F4(X1);

plot(X1, Yt, 'green');

plot(X1, Zt, 'm');
```

```
% 4 Печатаем графики приближений:

plot (X, Y, 'r-o');

plot (X, Z, 'b-o');

% 5 Подписываем легенду

title(titleValue, 'FontName', 'Courier');

h1 = legend('Точное значение Y', 'Точное значение Z', 'Приближенное значение Y', 'Приближенное значение Z');

set(h1, 'FontName', 'Courier');

end
```
