

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ИМЕНИ М.В. ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ
И КИБЕРНЕТИКИ

КАФЕДРА ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

Отчет

О ПРОДЕЛАННОЙ РАБОТЕ В РАМКАХ ПРАКТИКУМА
НА ЭВМ В 6 СЕМЕСТРЕ

Автор:

П.К. АХРАМЕЕВ

Руководители:

Б.А. БУДАК

Ю.Н. КИСЕЛЕВ

С.Н. АВВАКУМОВ

А.В. НИЧИПОРЧУК

Москва — 2013

Содержание

1	Постановка задачи	3
2	Теоретическое описание метода	3
3	Реализация программы	6
4	Примеры	8
4.1	Краевая задача двух тел	8
4.2	Предельные циклы в системе Эквейлера	9
4.3	функционал типа “энергия” для трехкратного интегратора	11
5	Пользовательский интерфейс программы	11
6	Исходный код программы	13
6.1	Решение внутренней задачи Коши	14
6.2	Функция создания индикатора выполнения процесса	15
6.3	Решение внешней задачи Коши	16
6.4	Внешняя задача Коши	16
6.5	Функция сохранения графиков	17

1 Постановка задачи

рамках практикума на ЭВМ VI семестра кафедры оптимального управления факультета ВМК МГУ необходимо было реализовать программу для решения краевых задач методом продолжения по параметру. Строгая математическая постановка задания будет изложена далее.

2 Теоретическое описание метода

ассматриваем краевую задачу

$$\begin{cases} x \dot{=} f(t, x), a \leq t \leq b, \\ R(x(a), x(b)) = 0, \\ x \in \mathbb{E}^n. \end{cases} \quad (1)$$

Здесь $f(t, x) : \mathbb{E}^1 \times \mathbb{E}^n \rightarrow \mathbb{E}^n$, $R(x, y) : \mathbb{E}^n \times \mathbb{E}^n \rightarrow \mathbb{E}^n$ являются гладкими векторными функциями. Предполагаем, что решение краевой задачи (1) существует. Выберем точку $t_* \in [a, b]$ и рассмотрим задачу Коши

$$\begin{cases} x \dot{=} f(t, x), a \leq t \leq b, \\ x|_{t=t_*} = p \in \mathbb{E}^n. \end{cases} \quad (2)$$

Пусть

$$x(t, p), a \leq t \leq b, \quad (3)$$

— решение задачи Коши (2). Предполагается продолжимость решения (3) на весь отрезок $[a; b]$ для любого p . Начальное значение параметра $p \in \mathbb{E}^n$ ищется из условий выполнения векторного граничного условия в задаче (1), т.е. искомое p является решением уравнения

$$\Phi(p) \equiv R(x(a, p), x(b, p)) = 0. \quad (4)$$

Итак, краевая задача (1) сведена к конечному векторному уравнению (4). Далее к уравнению (4) применяется метод продолжения по параметру. Для этого рассмотрим вспомогательное уравнение

$$\Phi(p(\mu)) = (1 - \mu)\Phi(p_0), \mu \in [0, 1], \quad (5)$$

где $p_0 \in \mathbb{E}^n$ — произвольное фиксированное начальное приближение.

Всюду далее считаем выполненными следующие два предложения.

Предположение 1. (о гладкой ветви) Уравнение (5) при любом $\mu \in [0, 1]$ имеет решение

$$p = p(\mu), 0 \leq \mu \leq 1, \quad (6)$$

функция (6) является гладкой функцией параметра μ и удовлетворяет начальному условию

$$p(\mu)|_{\mu=0} = p_0. \quad (7)$$

Предположение 2. (о невырожденности) Матрица

$$\Phi'(p) = (\partial \Phi_i(p) / \partial p_j)_{i,j=1}^n, \quad (8)$$

невырождена вдоль $p\mu$

Справедливость **Предположений 1, 2** зависит от уравнения (4) и от выбора точки p_0 . Конечно, прямая проверка **Предположений 1, 2** в сложных нелинейных задачах невозможна. Успешное завершение вычислительного процесса может служить косвенным подтверждением выполнения этих предположений.

Продифференцируем тождество (5) по μ :

$$\Phi'(p(\mu)) \frac{dp(\mu)}{d\mu} = -\Phi(p_0). \quad (9)$$

Из **Предположения 1** и формулы (9) следует, что функция $p(\mu)$ является решением векторной задачи Коши

$$\begin{cases} \frac{dp}{d\mu} = -[\Phi]^{-1} \Phi(p_0), 0 \leq \mu \leq 1, \\ p(0) = p_0. \end{cases} \quad (10)$$

Численное решение задачи Коши (10) позволяет найти функцию $p(\mu)$, $0 \leq \mu \leq 1$; вектор

$$p(\mu)|_{\mu=1} \quad (11)$$

должен быть точным решением (4) в идеальной ситуации точного нахождения $p(\mu)$. В реальных вычислениях вектор (11) дает новое приближение для решения; точность решения зависит от использованного численного метода и его параметров. Один шаг итерационной процедуры ассоциируется с решением задачи Коши (10). Итерационный процесс p^0, p^1, p^2, \dots поиска решения уравнения (4) представим схемой

$$p^0 \xrightarrow[p_0=p^0]{(10)} p^1 = p(\mu)|_{\mu=1} \xrightarrow[p_0=p^1]{(10)} p^2 = p(\mu)|_{\mu=1} \xrightarrow[p_0=p^2]{(10)} p^3 \dots \quad (12)$$

В нашей задаче матрица $\Phi'(p)$ имеет вид

$$\Phi'(p) = R'_x \frac{\partial x(a, p)}{\partial p} + R'_y \frac{\partial x(b, p)}{\partial p}.$$

Здесь $(n \times n)$ – матрицы $R'_x(x, y), R'_y(x, y)$ вычисляются вдоль решения (3), т.е. при $x = x(a, p), y = y(b, p)$. Введем обозначение

$$X(t, p) = \frac{\partial x(t, p)}{\partial p}$$

для $(n \times n)$ – матрицы производных решения (3) по начальному условию. Матрица $X(t, p)$ определяется дифференциальным уравнением в вариациях

$$\begin{cases} X \dot{=} AX, a \leq t \leq b, \\ X|_{t=t_0} = I, \end{cases}$$

где $A = A(t, p) = f'_x(t, x)|_{x=x(t, p)}$ есть $(n \times n)$ – матрица, I — единичная матрица. Для одновременного вычисления векторной функции $x(t, p)$ и матричной функции $X(t, p)$ может быть записана следующая векторно-матричная задача Коши

$$\begin{cases} x \dot{=} f(t, x), x|_{t=t_*} = p, \\ X \dot{=} f'_x(t, x)X, X|_{t=t_*} = I, a \leq t \leq b. \end{cases} \quad (13)$$

Задачу Коши (10) будем называть внешней задачей, задачу Коши (13) — внутренней задачей. Таким образом, предлагается итерационный процесс (12) для решения рассматриваемой краевой задачи (1) на основе внешней задачи (10) и внутренней задачи (13). На одном шаге итерационного процесса выпняется решение внешней задачи (10), в ходе решения которой происходит многократное обращение к решению внутренней задачи Коши (13) при различных значениях параметра p . При формировании матриц f'_x, R'_x, R'_y привлекаются возможности среды по выполнению аналитических вычислений.

3 Реализация программы

программе реализованы:

1. Описанный выше алгоритм
2. Пользовательский интерфейс
 - (а) Основной экран программы
 - Поля ввода данных
 - Поле ввода размерности системы дифференциальных уравнений
 - Поля ввода параметров временной сетки: правый и левый концы; шаг сетки
 - Поле ввода параметра t_* для описанного выше алгоритма
 - Поля выбора численных методов решения внешней и внутренней задачи из списка
 - i. Рунге-Кутта-Фельберга 2-3
 - ii. Рунге-Кутта-Фельберга 4-5
 - iii. Адамса
 - iv. Эйлера
 - Поля ввода погрешности расчетов внутренней и внешней задачи
 - Поле ввода количества итераций (вычисления внешней задачи Коши)
 - После вычисления становится доступно поле ввода функционала и кнопка для его вычисления
 - По кнопке "Показать/скрыть краевую задачу" можно открыть/скрыть таблицу для ввода краевой задачи
 - i. Правая часть системы дифференциальных уравнений $f(t, x)$ в формате $x = (x_1, x_2, \dots, x_n)$.
Примечание: формат ввода: x1, x2, ..., xn
 - ii. Краевые условия в виде функции $R(x, y)$ от переменных $x = (x_1, x_2, \dots, x_n)$ и $y = (y_1, y_2, \dots, y_n)$.
Примечание: формат ввода x1, x2, ..., xn, y1, y2, ..., yn

- iii. Вектор p_0 — вектор начального приближения в итерационном процессе (12)
 - iv. Параметр отображения траектории на графике
 - Поля вывода результатов
 - Координатная плоскость для вывода графиков
 - Таблица результатов со значениями решения на временной сетке
 - Поля настройки отображения данных на графике (применяются по нажатию на кнопку "Обновить")
 - Списки выбора стиля и цвета линии
Примечание: списки значений легко редактируются текстовым редактором (внимательно изучите содержимое папки /Settings в каталоге программы)
 - Поля для изменения масштаба графиков
 - Поля для вывода конкретной фазовой траектории на графике
3. Механизм загрузки и сохранения примеров, результатов
 4. Редактор таблиц с возможностью экспорта их в MS Excel (требуется MS Excel for Windows®)
 5. Библиотека примеров из книги [1]
 6. Пункты меню "Помощь" и "О программе"

4 Примеры

Сравнение результатов работы программы с результатами, приведенными в книге [1].

4.1 Краевая задача двух тел

$$\begin{cases} x_1 \dot{=} x_3, \\ x_2 \dot{=} x_4, \\ x_3 \dot{=} -x_1(x_1^2 + x_2^2)^{-3/2}, \\ x_4 \dot{=} -x_2(x_1^2 + x_2^2)^{-3/2}, \\ x_1(0) = 2, \\ x_2(0) = 0, \\ x_1(7) = 1 : 0738644361, \\ x_2(7) = 1 : 0995343576, \\ t_* = 0, \\ p_0^1 = [2, 0, -0.5, 0.5], \\ p_0^2 = [2, 0, 0.5, -0, 5]. \end{cases}$$

Библиотека примеров: `/Examples/edged2figures.mat`.

Результаты эксперимента

- Для вектора p_0^1 : $p_0 = [2, 0, 0, 0.5]$.
- Для вектора p_0^2 : $p_0 = [2, 0, 0.4511, 0.2994]$

Замечу, что результаты эксперимента близки к результатам, указанным в [1]:

- $ans1 = [2., 0., 0.0000004834, 0.5000000745]$,
- $ans2 = [2., 0., 0.4510782034, -0.2994186665]$.

На рисунке (1) приведены фазовые траектории $x_2(x_1)$ для $p_0^1 = [2, 0, -0.5, 0.5]$ и $p_0^2 = [2, 0, 0.5, -0, 5]$.

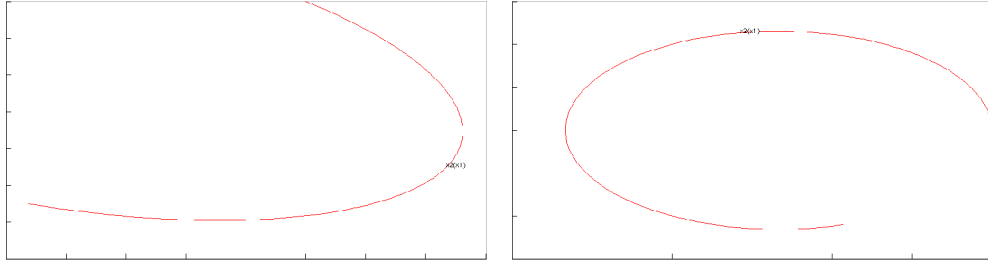


Рис. 1: Фазовые траектории $x_2(x_1)$ в обоих случаях

4.2 Пределные циклы в системе Эквейлера

$$\left\{ \begin{array}{l} x_1 \doteq x_2 x_3, \\ x_2 \doteq x_3 (-x_1 + \sin(x_2)), \\ x_3 \doteq 0, \\ x_4 \doteq 0, \\ x_1(0) = x_4(0), \\ x_2(0) = 0, \\ x_1(1) = x_4(1), \\ x_2(1) = 0, \\ t_* = 0, \\ p_0^1 = [2, 0, 2\pi, 2], \\ p_0^2 = [6.5, 0, 2\pi, 6.5], \\ p_0^3 = [9, 0, 2\pi, 9], \end{array} \right.$$

Библиотека примеров: `/Examples/limitLoopsInEkveiler.mat`.

Результаты эксперимента

- Для вектора p_0^1 : $p_0 = [3.96554684478714, -2.62629602035752e - 15, 6.46614183079246, 3.96554684478714]$.
- Для вектора p_0^2 : $p_0 = [7.10786344954066, -2.27592750257336e - 18, 6.33878906325561, 7.10786344954066]$
- Для вектора p_0^3 : $p_0 = [10.2456945992873, -8.69444146067045e - 14, 6.31011276249781, 10.2456945992873]$

Результаты эксперимента, указанные в [1]:

- $ans1 = [3.9655467678, 0, 6.4661401325, 3.9655467678]$,
- $ans2 = [7.1078664573, 0, 6.3387892836, 7.1078664573]$,
- $ans3 = [10.2456910360, 0, 6.3101121791, 10.2456910360]$.

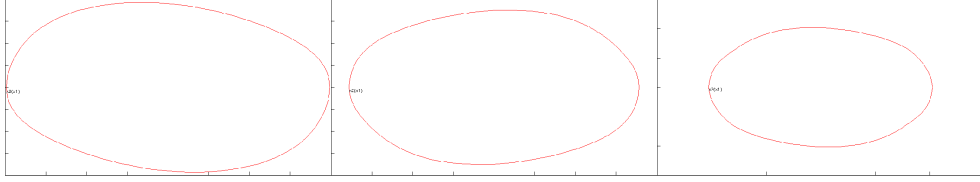


Рис. 2: Фазовые траектории $x_2(x_1)$ для p_0^1, p_0^2 и p_0^3

На рисунке (2) приведены фазовые траектории $x_2(x_1)$ для $p_0^1 = [2, 0, 2\pi, 2]$, $p_0^2 = [6.5, 0, 2\pi, 6.5]$ и $p_0^3 = [9, 0, 2\pi, 9]$.

4.3 функционал типа “энергия” для трехкратного интегратора

$$\left\{ \begin{array}{l} x_1 \dot{=} x_2, \\ x_2 \dot{=} x_3, \\ x_3 \dot{=} \frac{1}{2}(\sqrt{\nu + (x_6 + 1)^2} - \sqrt{\nu + (x_6 - 1)^2}), \\ x_4 \dot{=} 0, \\ x_5 \dot{=} -x_4, \\ x_6 \dot{=} -x_5, \\ x_1(0) = 1, \\ x_2(0) = 0, \\ x_3(0) = 0, \\ x_1(3.275) = 0, \\ x_2(3.275) = 0, \\ x_3(3.275) = 0, \\ t_* = 3.275, \\ p_0 = [0, 0, 0, -3, 5, -3], \\ \nu = 10^{-10} \end{array} \right.$$

Библиотека примеров: `/Examples/energy3integrate.mat`.

Результат эксперимента

$p_0 = [1.56258154391785e - 33, 1.05187183714427e - 17,$
 $-2.80659384611396e - 18, -2.98505141670188, 4.88802217076114, -2.90839776271499].$

Результат, указанный в [1]:

$ans = [0, 0, 0, -2.9850435834, 4.8880088678, -2.9083874537].$

5 Пользовательский интерфейс программы

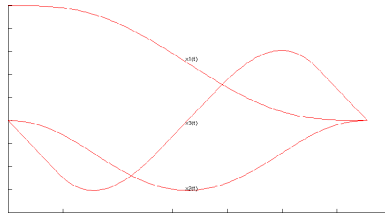


Рис. 3: Траектории $x_1(t)$, $x_2(t)$ и $x_3(t)$

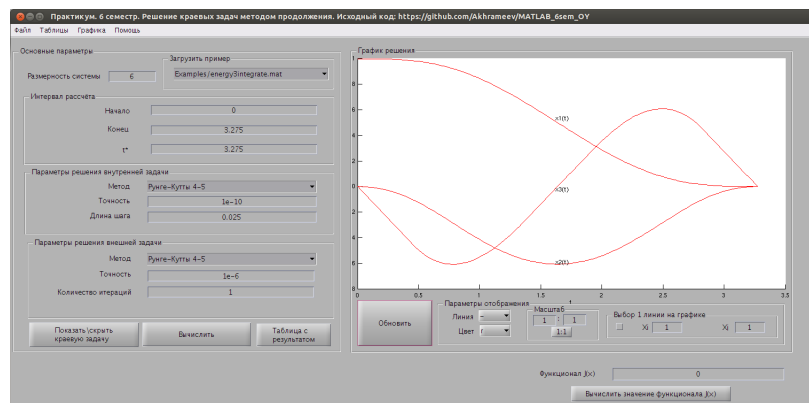


Рис. 4: Главный экран программы после выполнения вычислений

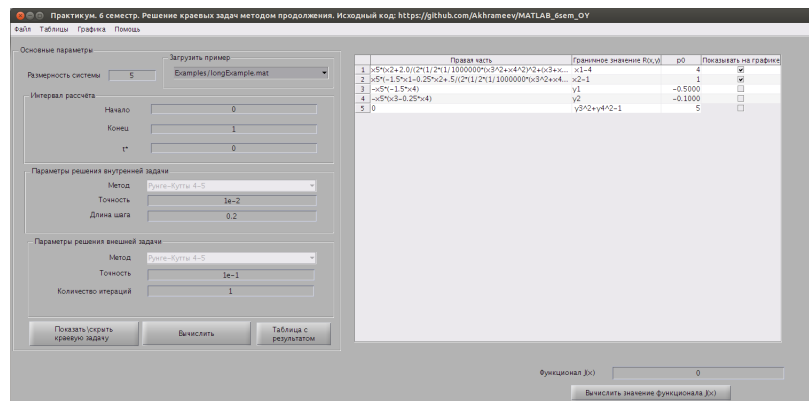


Рис. 5: Главный экран программы: таблица с краевой задачей

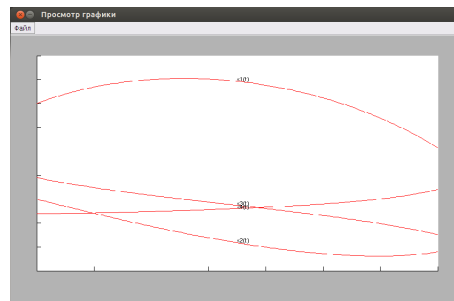


Рис. 6: Окно просмотра и сохранения изображений

1	2	3	4
1	0	0.4611	-0.2394
2	2.0222	-0.0150	0.4987
3	2.0439	-0.0299	0.4266
4	2.0649	-0.0449	0.4148
5	2.0854	-0.0598	0.4032
6	2.1052	-0.0748	0.3918
7	2.1245	-0.0897	0.3807
8	2.1433	-0.1046	0.3697
9	2.1615	-0.1194	0.3590
10	2.1792	-0.1342	0.3484
11	2.1964	-0.1490	0.3380
12	2.2130	-0.1638	0.3278
13	2.2291	-0.1785	0.3178
14	2.2448	-0.1932	0.3079
15	2.2599	-0.2079	0.2981
16	2.2746	-0.2224	0.2885
17	2.2888	-0.2370	0.2791
18	2.3025	-0.2515	0.2698
19	2.3158	-0.2660	0.2606
20	2.3286	-0.2804	0.2515
21	2.3409	-0.2947	0.2425
22	2.3528	-0.3090	0.2336
23	2.3643	-0.3232	0.2249
24	2.3753	-0.3374	0.2162
25	2.3859	-0.3515	0.2077
26	2.3961	-0.3655	0.1992
27	2.4058	-0.3795	0.1909
28	2.4152	-0.3934	0.1826
29	2.4241	-0.4073	0.1744
30	2.4326	-0.4211	0.1663

Рис. 7: Редактор таблиц с открытыми результатами вычислений

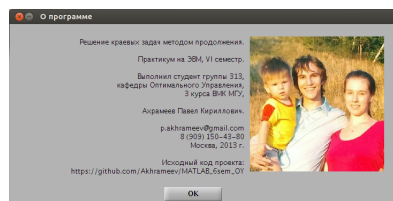


Рис. 8: О программе

6 Исходный код программы

Исходный код программы доступен:

https://github.com/Akhrameev/MATLAB_6sem_OY.

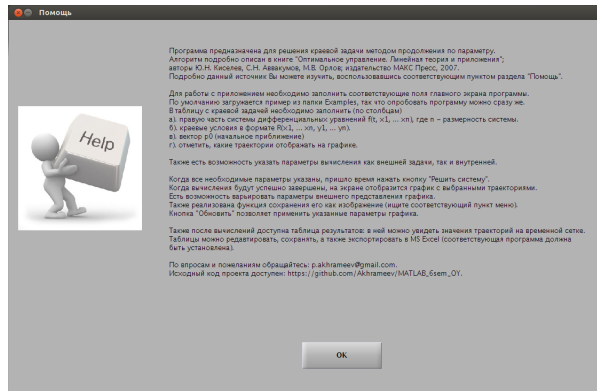


Рис. 9: Помощь

6.1 Решение внутренней задачи Коши

```
function [ya, yb, Xa, Xb] = solveInternal (p)
global solvingInternalEpsilon;
global solvingInternalMethod solvingTimeBegin;
global solvingTimeEnd solvingStep;
global systemForSolvingDimension solvingTimeStarred;
n = systemForSolvingDimension;
l = solvingTimeBegin;
r = solvingTimeEnd;
s = solvingStep;
t_s = solvingTimeStarred;
opts = odeset('RelTol', solvingInternalEpsilon);
nMult = n * (n + 1);
q = zeros(1, nMult);
q(1 : n) = p;
I = eye(n);
for i = 1 : n
    q(i * n + 1 : n * (i + 1)) = I(i, :);
end
if t_s == l
    [~, X] = solvingInternalMethod(@Internal, [l:s:r], q, opts);
elseif t_s == r
    [~, X1] = solvingInternalMethod(@Internal, [r:-s:l], q, opts);
```

```

X = ones (length (X1),nMult);
X (1 : end, :) = X1 (end:-1:1,:);
else
    [~,X1] = solvingInternalMethod (@Internal,[t_s:-s:1],q,opts);
    [~,X2] = solvingInternalMethod (@Internal,[t_s: s:r],q,opts);
    X = ones(length(X1) + length(X2) - 1, nMult);
    X(1 : length(X1),      :) = X1(end:-1:1, :);
    X(length(X1) + 1 : end, :) = X2(2:end,      :);
end
Xa = zeros (n,n);
Xb = zeros (n,n);
ya = X(1, 1:n);
yb = X(end, 1:n);
ya = ya';
yb = yb';
for i = 1 : n
    Xa(:,i) = X(1, i * n + 1 : (i + 1) * n);
    Xb(:,i) = X(end, i * n + 1 : (i + 1) * n);
end
end

```

6.2 Функция создания индикатора выполнения процесса

```

function createWaitbar (handles)
global jProgressbar progressbarOK jInternalProgressbar;
global solvingIterationCount solvingIterationCurrent;
if (progressbarOK)

else
    try
        externalProgressbar = handles.externalProgressbar;
        internalProgressbar = handles.internalProgressbar;
        jProgressbarPosition = get (externalProgressbar, 'Position');
        jInternalProgressbarPosition = get (internalProgressbar, ...
            'Position');
        jProgressbar = jcontrol(handles.figure1, ...
            javax.swing.JProgressBar(),...

```

```

        'Position', jProgressbarPosition, 'Visible', 'off');
jProgressbar.setMaximum (solvingIterationCount);
jProgressbar.setMinimum (0);
jProgressbar.setValue (solvingIterationCurrent);
jProgressbar.setBorderPainted (true);
jProgressbar.setStringPainted (true);
jInternalProgressbar = jcontrol (handles.figure1, ...
    javax.swing.JProgressBar(), ...
    'Position', jInternalProgressbarPosition, 'Visible', 'off');
jInternalProgressbar.setIndeterminate (true);
progressbarOK = 1;
catch
    jProgressbar = 0;
    jInternalProgressbar = 0;
    progressbarOK = 0;
end;
end
end

```

6.3 Решение внешней задачи Коши

```

function [ dp ] = solveExternal( count, p )
global q0 ;
updateWaitbar ();
[xa, xb, Xa, Xb] = solveInternal (q0);
F = R(xa, xb);
[xa, xb, Xa, Xb] = solveInternal (p);
dp = -(Rx (xa, xb) * Xa + Ry (xa, xb) * Xb) \ F;
end

```

6.4 Внешняя задача Коши

```

function [ p ] = external (p0, iterationCount)
global solvingExternalEpsilon solvingExternalMethod;
global q0 solvingIterationCurrent;
global pause;
q0 = p0;
opts = odeset('RelTol', solvingExternalEpsilon);
[xa, xb, ~, ~] = solveInternal (p0);
while (solvingIterationCurrent < iterationCount)

```



```

    if (pause)
        break;
    end
    [~, ps] = solvingExternalMethod (@solveExternal, [0 1], p0, opts);
    if norm(R (xa, xb)) >= solvingExternalEpsilon
        q0 = ps (end, :);
    end
    [xa, xb, ~, ~] = solveInternal (q0);
    solvingIterationCurrent = solvingIterationCurrent + 1;
end
p = q0;
end

```

6.5 Функция сохранения графиков

```

function SaveMenuItem_Callback(hObject, eventdata, handles)
% hObject      handle to SaveMenuItem (see GCBO)
% eventdata    reserved — to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
tableList = getAllFiles ('Graphics/');
listSize = size(tableList);
listSize = listSize(2);
defaultName = strcat ('Graphics/graphic', num2str(listSize + 1), '.png');
filename = uinputfile ('Graphics/', 'Save_current_graphic', defaultName);
if (filename == 0)
    return;
end
filename = strcat ('Graphics/', filename);
F = getframe(handles.axes1);
image(F.cdata);
axis ('image');
axis ('off');
try
    imwrite(F.cdata, filename);
catch
    filename = strcat (filename, '.png');
    imwrite (F.cdata, filename);
end

```

Список литературы

- [1] Орлов М. В. Киселев Ю. Н., Аввакумов С. Н. *Оптимальное управление. Линейная теория и приложения*. М.:МАКС Пресс, 2007.