

Федеральное агентство по образованию
Государственное образовательное учреждение высшего
профессионального образования
Самарский государственный аэрокосмический университет имени
академика С.П. Королева

Составители: Никоноров А.В., Фурсов В.А.

УДК

Компьютерное моделирование процессов и систем с использованием пакета Matlab

Методические указания к лабораторной работе № 5

Компьютерное моделирование процессов и систем с использованием пакета Matlab / Сост. Никоноров А.В., Фурсов В.А. - Самара: Изд-во Самарского государственного аэрокосмического университета, 2009. ___с.

В настоящих методических указаниях приведен материал, необходимый для выполнения лабораторных работ по дисциплине «Информатика» учебных планов машиностроительных факультетов.

Целью лабораторных работ является изучение системы Matlab и знакомство с методами и алгоритмами математического моделирования динамических объектов.

Методические указания предназначены для студентов дневного и вечернего отделений всех специальностей машиностроительных факультетов.

Печатается по решению редакционно-издательского совета Самарского государственного аэрокосмического университета

Рецензент: Попов С.Б.

СОДЕРЖАНИЕ

1. Цель лабораторной работы	5
2. Необходимое оборудование.....	5
3. Теоретическая часть	6
3.1 Математические расчеты в системе Matlab	6
3.2 Математическая модель равноускоренного движения	14
4. Контрольные вопросы	18
5. Задания для лабораторной работы	18
6. Список литературы	19

1. Цель лабораторной работы

- Изучение основ работ с пользовательским интерфейсом системы Matlab;
- приобретение практических знаний и навыков программирования в системе Matlab;
- получение навыков разработки и реализации программ на основе математической модели объекта;
- разработка компьютерной модели равноускоренного движения тела.

2. Необходимое оборудование

Вычислительный класс Intel совместимых компьютеров под управлением Windows или Linux, предустановленный пакет Matlab.

3. Теоретическая часть

3.1 Математические расчеты в системе Matlab

В процессе работы как инженера, ученого, так и простого студента, ил приходится прибегать к математическим расчетам. Для сложных расчетов можно применять ряд математических пакетов, одним из которых и является MatLAB.

MATLAB (MATrix LABoratory) – интерактивный матрично-ориентированный пакет, предназначенный для выполнения научных и инженерных расчетов.

Окна системы MATLAB

По умолчанию после запуска пакета MATLAB на экране появляется комбинированное окно, включающее четыре наиболее важные панели (рис. 1):

- **Command Window** (Окно команд) – самая используемая панель. В ней набирают команды пользователя, подлежащие немедленному исполнению. Здесь же выдаются результаты выполненных команд.
- **Command History** (История команд) хранит все команды, набираемые пользователем, однако в отличие от содержимого Command Window (Окно команд) сюда не попадают сообщения системы и результаты вычислений.
- **Workspace** (Рабочее пространство) отображает текущий набор переменных, заведенных пользователем в командном окне.
- **Current Directory** (Текущий каталог) является аналогом известной программы Проводник, но имеет для MATLAB свое особое предназначение. Дело в том, что, кроме работы с математическими выражениями из командного окна, пользователь также может работать с файлами.

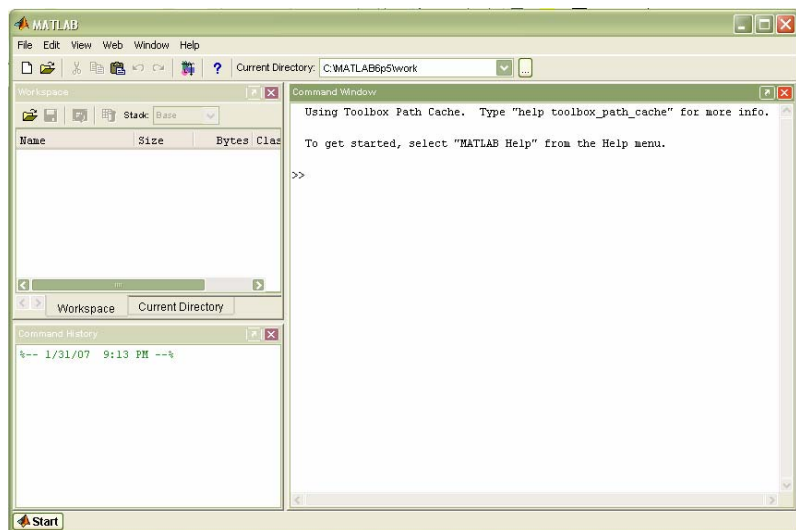


Рисунок 1. Общий вид главного окна пакета MATLAB

Основные типы данных

Система MatLAB работает со следующими базовыми типами данных.

Число – вещественное числовое значение.

Массив – это данные (объекты) одной природы сгруппированные по одному и тому же характерному признаку.

Матрица – массив представленный в виде прямоугольной таблицы, каждый элемент имеет номер (индекс), определяющий однозначно его положение в матрице, в индексировании идет сначала номер строки, а потом номер столбца, где расположен элемент.

Многомерный массив – пространственная матрица имеющая три и более размерностей, каждый элемент также имеет индекс, однозначно определяющий его положение. Грубо говоря многомерный массив – это матрица матриц.

Вектор – одномерная матрицы. Без особых указаний со стороны пользователя это матрица-столбец.

Структура – это набор разнотипных полей. Поле может содержать как массив так и число так и строку. Одно поле содержит данные только одного типа.

Строка – набор (массив) символов символьных таблиц компьютера.

Числа, переменные, функции

Числа в MATLAB могут быть положительными и отрицательными, целыми и дробными, действительными и комплексными. Они могут представляться с фиксированной и плавающей точкой, с мантиссой и порядком.

Особенности представления чисел в MATLAB:

- мнимая единица кодируется с помощью двух символов: i или j ;
- целая часть числа от дробной отделяется точкой;
- отделение порядка числа от мантиссы осуществляется символом e .

Форматы чисел:

- `format short` – короткое представление (5 знаков числа);
- `format short e` – короткое представление в экспоненциальной форме (5 знаков мантиссы, 3 знака порядка);
- `format long` – длинное представление числа (15 знаков);
- `format long e` – длинное представление в экспоненциальной форме (15 знаков мантиссы, 3 знака порядка).

Переменные – это символы, используемые для обозначения некоторых хранимых данных. Переменная имеет имя, называемое *идентификатором*. Имя переменной начинается с буквы и может состоять из букв и цифр и некоторых (допустимых) символов.

Константы – это численное значение уникального имени, имеющего математический смысл. Наиболее часто в MATLAB используются следующие константы:

- `pi` – число π ;
- `inf` – машинная бесконечность;
- `ans` – имя переменной, хранящей результат вычисления;
- `NaN` – нечисловой характер данных, возникает, например, как результат операции $1/0$.

Элементарные функции:

- `abs(x)` – абсолютное значение x ;
- `exp(x)` – экспоненциальная функция e^x ;
- `log(x)`, `log10(x)`, `log2(x)` – логарифмы чисел с основанием e , 10, 2;
- `sqrt(x)` – корень квадратный из x ;
- `sin(x)`, `cos(x)`, `tan(x)`, `cot(x)`, `sec(x)`, `csc(x)` – тригонометрические функции $\sin x$, $\cos x$, $\operatorname{tg} x$, $\operatorname{ctg} x$, $\operatorname{sec} x$, $\operatorname{cosec} x$;

- $\text{asin}(x)$, $\text{acos}(x)$, $\text{atan}(x)$, $\text{acot}(x)$, $\text{asec}(x)$, $\text{acsc}(x)$ – обратные тригонометрические функции $\arcsin x$, $\arccos x$, $\arctg x$, $\text{arctg } x$, $\text{arcsec } x$, $\text{arccosec } x$;
- $\text{sinh}(x)$, $\text{cosh}(x)$, $\text{tanh}(x)$, $\text{coth}(x)$, $\text{sech}(x)$, $\text{csch}(x)$ – гиперболические функции $\text{sh } x$, $\text{ch } x$, $\text{th } x$, $\text{cth } x$, $\text{sch } x$, $\text{csch } x$;
- $\text{asinh}(x)$, $\text{acosh}(x)$, $\text{atanh}(x)$, $\text{acoth}(x)$, $\text{asech}(x)$, $\text{acsch}(x)$ – обратные гиперболические функции $\text{arsh } x$, $\text{arch } x$, $\text{arth } x$, $\text{arth } x$, $\text{arsch } x$, $\text{arsch } x$.

Функцию пользователя можно создать следующим образом:

1. Вызов окна редактора m-файлов путем нажатия кнопки **New M-File** (Создать m-файл).
 2. Ввод строки
`function Z=expxp(x)`
 Ключевое слово **function** объявляет новую функцию, имя которой **expxp**, а ее параметр – x . Символ Z определяет значение функции при аргументе x .
 3. Задание новой функции (функции пользователя). Пусть
`Z=exp(x)/x`
 4. Сохранение функции пользователя на диске. Для этого достаточно щелкнуть мышью по кнопке **Save** (Сохранить).
 5. Закрытие окна редактора m-файлов.
- Функция пользователя $Z=\text{exp}(x)/x$ создана.

Для вычисления функции при данном аргументе x достаточно набрать имя функции и значение аргумента в круглых скобках: $z=\text{expxp}(1)$. На экране получим значение функции $z = 2.7183$.

Отметим несколько важных особенностей системы программирования Matlab. В языке присутствует символ строчного комментария – %:

% Это комментарий

Использование точки с запятой не является обязательным. Символ ; используется для подавления вывода результата выполнения команды на экран. Так строка

`x = 1;`

выполнит присвоение переменной x значения 1, а строка

`x = 1`

выполнит присвоение и выведет на экран результат – $x = 1$.

Матрицы и векторы

Матрицы в Matlab это прямоугольные массивы чисел. Число это так же матрица, размерности 1×1 , скаляр, матрица, имеющая один столбец или одну строку – вектор. В системе Matlab все числовые данные можно рассматривать как матрицы. В отличие от других языков, оперирующих отдельными числами, Matlab оперирует целыми матрицами.

Ввод матрицы осуществляется использованием метода так называемого объединения элементов. Это объединение записывается так:
`>> A=[1 1 1 2];`

Здесь мы сформировали вектор (или матрицу столбец, причем для системы не важна ее пространственная ориентация, т.е. что размерность $1 \times n$, что $n \times 1$, все едино). Элементы при таком формировании разделяются пробелами или запятыми. Для формирования матрицы, производят построчную запись, отделяя при этом строки друг от друга точкой с запятой:

`>> A=[1 2 8; 3 7 0; 7 9 5];`

При этом можно производить и составления матрицы путем объединения матриц или векторов между собой (предполагается, что вектора $A1$ $A2$ были определены заранее):

`>> Q=[A1;A2];`

При этом вектор $A1$ будет формировать первую строку матрицы, а вектор $A2$ вторую.

При обращении к элементу матрицы используют запись вида $A(1, 8)$, при этом произошло обращение к элементу первой строки восьмого столбца. Если необходимо произвести обращение не к одному а к нескольким элементам, а точнее выделить из общей матрицы подматрицу то используют запись вида

`B = A(1:5, 6:8);`

Матрице B присвоена подматрица A содержащая строки с 1 по 5, и столбцы с 6 по 8. Если необходимо выделить весь столбец (строку), то вместо указания конкретных номеров можно просто поставить пустое двоеточие ($A(:, 6:8)$).

Если требуется сформировать вектор, элементу которого расположены в арифметической прогрессии то проще всего это сделать так:

`x=xn:h:xk`

xn – начальное значение;

xk – конечное значение;

h – шаг изменения.

Таким образом, хорошо задавать, например время изменения сигнала или создавать массив значений координаты.

Матрицы можно складывать, умножать и возводить в степень по правилам линейной алгебры. Дополнительно введены поэлементные операции с матрицами такие как: `.*` – поэлементное сложение, `.*` – поэлементное умножение. Поясним различие между обычными и поэлементными операциями на примере:

```
A = [1, 2; 3, 4];
B = [5, 6; 7, 8];
% умножение матриц по правилам линейной алгебры
C = A * B;
% поэлементное умножение матриц
D = A .* B;
```

В результате получим: C = [19, 22; 43, 50], D = [5, 12; 21, 32];

Визуализация вычислений

Система MATLAB имеет богатые возможности графического представления информации. Она позволяет строить двумерные и трехмерные графики функций, заданных в аналитическом виде, в виде векторов и матриц, дает возможность построения множества функций на одном графике: позволяет представлять графики разными цветами, типами точек и линий и в различных системах координат.

Основными функциями двумерной графики являются:

```
plot(x, y)
plot(x, y, s)
plot(x1, y1, s1, x2, y2, s2, ..., xn, yn, sn)
```

где:

- ◆ x – аргумент функции, задаваемой в виде вектора;
- ◆ y – функция, представленная в аналитическом виде или в виде вектора или матрицы;
- ◆ s – вектор стилей графика; константа, определяющая цвет линий графика, тип точек и тип линий;
- ◆ x1, x2, ..., xn – аргументы n функций, изображаемых на одном графике;
- ◆ y1, y2, ..., yn – функции, изображаемые на одном графике.

В таблице 1 приведены стили графиков системы MATLAB.

Таблица 1. Стили графиков

Тип точки		Цвет линии		Тип линии	
.	Точка	Y	Желтый	-	Сплошная
O	Окружность	M	Фиолетовый	:	Двойной пунктир

x	Крест	C	Голубой	-.	Штрих-пунктир
+	Плюс	R	Красный	--	Штриховая
*	Восьмиконечная снежинка	G	Зеленый		
S	Квадрат	B	Синий		
D	Ромб	W	Белый		
V, ^, <, >	Треугольник вверх, вниз, влево, вправо	K	Черный		
P	Пятиконечная звезда				
H	Шестиконечная звезда				

```
Рассмотрим пример построения графика функции y=sin x·e-x.
В окне Command Window задается программа:
>> x=-5:0.5:5; % задание промежутка [-5;5] с шагом 0,1
>> y=sin(x).*exp(-x); % задание функции y
>> plot(x,y,['R','*','-'.']) % выводение графика красного цвета (R), точки графика в виде снежинок (*), линии штрихпунктирные (-.)
>> grid on % задание сетки
```

График функции приведен на рис. 2.

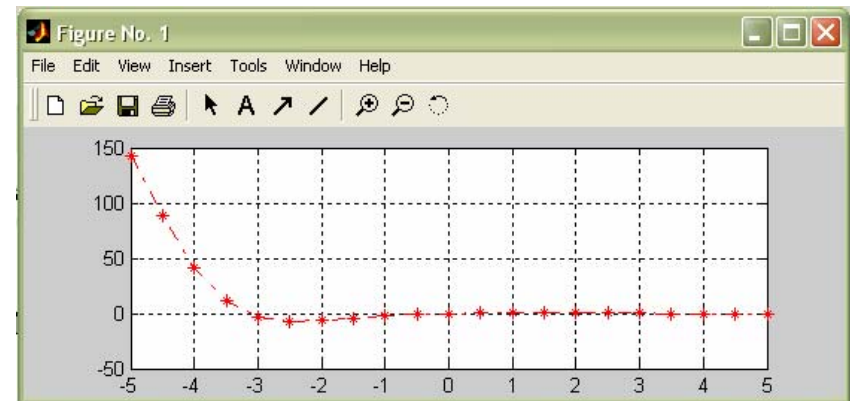


Рисунок 2. График функции $y = \sin x \cdot e^{-x}$.

В дополнение к управлению выводом координатной сетки, можно настроить режим затирания предыдущего графика при выводе нового. По-умолчанию каждая следующая команда plot затирает вывод предыдущей. При помощи команды **hold on** затирание отключается, команда **hold off** вновь включает режим затирания.

3.2 Математическая модель равноускоренного движения

В настоящей работе будет рассмотрено решение простейших задач математического моделирования с использованием системы Matlab.

Задача равноускоренного движения тела под действием силы тяжести.

Рассмотрим тело, начинающее движение со скоростью V_0 в момент времени $t_0 = 0$ под углом к горизонту α и движущееся под действием ускорения свободного падения. В начальный момент времени, тело имеет координаты $x(t_0) = x_0$, $y(t_0) = y_0$. Необходимо построить траекторию движения тела.

Эта простая математическая модель может, например, в упрощенном виде служить описанием движения ракеты или баллистического снаряда.

Описанное движение происходит в двух координатах, по оси y оно является равноускоренным, а по оси x – равномерным:

$$x(t) = x_0 + \cos(\alpha)v_0 t,$$

$$y(t) = y_0 + \sin(\alpha)v_0 t - \frac{1}{2}gt^2.$$

Здесь $g = 9.8 \frac{\text{M}}{\text{c}^2}$ – ускорение свободного падения.

Для заданных уравнений может быть решен целый набор задач. Приведем несколько примеров: построение траектории движения тела, построение графиков функций $x(t)$ и $y(t)$, нахождение времени полета тела.

Последняя задача одна из наиболее интересных. Рассмотрим ее решение. Полет тела начинается в момент времени $t = t_0$, завершается в момент времени $t = t_1$. Полет тела завершается, очевидно, при условии $y(t_1) = 0$, значит t_1 может быть найдено как решение уравнения:

$$y_0 + \sin(\alpha)v_0 t_1 - \frac{1}{2}gt_1^2 = 0.$$

Решение уравнения:

$$t_1 = \frac{v_0 \sin(\alpha) \mp \sqrt{v_0^2 \sin^2(\alpha) + 2gy_0}}{g}.$$

Очевидно, $v_0 \sin(\alpha) \leq \sqrt{v_0^2 \sin^2(\alpha) + 2gy_0}$, поэтому, окончательно,

$$t_1 = \frac{v_0 \sin(\alpha) + \sqrt{v_0^2 \sin^2(\alpha) + 2gy_0}}{g}.$$

В частном случае, при $y(t_0) = 0$ получаем:

$$t_1 = \frac{2v_0 \sin(\alpha)}{g}.$$

Проведенные расчеты позволяют нам составить программу на языке Matlab, для построения графика траектории полета тела под действием ускорения свободного падения.

Дополнительно, усложним задачу требованием, вывести на экран точки, представляющей собой цель, которую должен поразить наш воображаемый снаряд. Более миролюбивая трактовка задачи – обеспечение посадки ракеты в заданной точке.

Приведем пример функции на языке Matlab, которая решает эту задачу.

```
function plot_rocket(a_grad, v0, x_t, y_t)
% Функция рассчитывает траекторию, выводит ее график
и точку цели
% Параметры функции:
% a_grad угол альфа в градусах,
% v0 - начальная скорость
% x_t, y_t - координаты цели

% константа g - ускорение свободного падения
G = 9.8;
% выполним пересчет угла в радианы
a = pi * a_grad / 180;
% начальное положение тела
x0= 0;
y0= 0;

% время полета
t1 = (v0*sin(a) + (v0^2*sin(a)^2 +
2*G*y0)^0.5) / G
% вывод координаты падения тела
x1 = x0 + v0 * cos(a) * t1
% задание интервала времени в виде вектора -
прогрессии, с шагом 0.05 сек.
t = [0:0.05:t1, t1];
```

```
% вычисление координат траектории x и y
x = x0 + v0 * cos(a) * t;
y = y0 + v0 * sin(a) * t - 0.5 * G * t.^2;

% режим затирания при выводе включен
hold off
% вывод графика траектории
plot(x, y);

% режим затирания при выводе выключен
hold on;
% вывод графика точки цели в виде красной окружности
plot(x_t, y_t, 'or')
```

Сохраним эту функцию в рабочем каталоге в файле plot_rocket.m. В рабочем окне Matlab вызываем эту функцию с параметрами. Например:

```
>> rocket_plot(50,31.25, 89, 10);
```

В результате будет выведена траектория с начальной скоростью 50м/с, начальным углом к горизонту 32.25 градуса, цель с координатами (89, 10) (рис. 3).

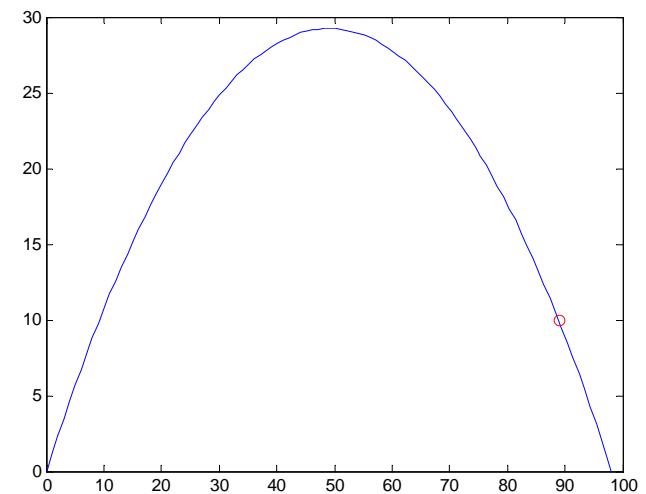


Рисунок 3. Траектория равноускоренного движения

Траектория на графике пересекает точку цели, такого совпадения можно добиться перебором различных значений угла и скорости. Также возможно аналитическое решение этой задачи, рассмотрим его в качестве дальнейшего примера.

Положим, что начальные координаты тела нулевые $(x_0, y_0) = (0, 0)$ и пусть (x_1, y_1) – координаты цели, тогда они удовлетворяют уравнениям:

$$x_1 = \cos(\alpha)v_0 t_1,$$

$$y_1 = \sin(\alpha)v_0 t_1 - \frac{1}{2}gt_1^2.$$

Отсюда следует:

$$\alpha = \arctg\left(\frac{\frac{1}{2}gt_1^2 + y_1}{x_1}\right),$$

$$v_0 = \frac{x_1}{\cos(\alpha)t_1}.$$

Эти соотношения определяют различные значения начальной скорости и угла в зависимости от требуемого времени достижения цели.

Приведем фрагмент функции на языке Matlab выполняющей расчет угла и скорости по заданному значению времени достижения цели t1:

$$a = \text{atan}((y_t + 0.5 * G * t1^2) / x_t);$$

$$v0 = (x_t / (\cos(a) * t1)).$$

4. Контрольные вопросы

1. Для чего предназначена система MATLAB?
2. Какие символы может содержать имя переменной?
3. Назовите наиболее используемые в MATLAB константы?
4. Какие элементарные функции вы знаете? Как они обозначаются в системе MATLAB?
5. Как создать функцию пользователя?
6. Перечислите основные функции двумерной графики.

5. Задания для лабораторной работы

Пусть тело, начинающее движение со скоростью V_0 в момент времени $t_0 = 0$ под углом к горизонту α и движется под действием ускорения свободного падения. В начальный момент времени, тело имеет координаты $x(t_0) = x_0$, $y(t_0) = y_0$. Необходимо написать функцию, получающую в качестве параметров указанные величины и выполняющую указанные действия.

1. Построить график траектории движения тела. В качестве параметров задается начальное положение тела (x_0, y_0) , угол к горизонту и начальная скорость, известно, что конечное положение тела имеет $y_1 = 0$.
2. Построить график траектории движения тела и целевую точку. В качестве параметров задаются координаты цели (x_1, y_1) , и время достижения цели, началом движения считать точку $(0, 0)$.
3. Построить графики нескольких траекторий движения тела, соответствующих нескольким значениям времени достижения цели, и целевую точку. В качестве параметров задаются координаты цели (x_1, y_1) , началом движения считать точку $(0, 0)$.
4. Построить графики нескольких траекторий движения тела, соответствующих нескольким значениям начального угла, определить угол, обеспечивающий максимальную дальность полета при неизменной начальной скорости. В качестве параметров задается угол к горизонту и начальная скорость, известно, что конечное положение тела имеет $y_1 = 0$, началом движения считать точку $(0, 0)$.
5. Построить графики зависимостей $x(t)$, $y(t)$ движения тела, соответствующих нескольким значениям начального угла. В качестве параметров задается угол к горизонту и начальная скорость, известно, что конечное положение тела имеет $y_1 = 0$, началом движения считать точку $(0, 0)$.

6. Построить графики зависимостей $x(t)$, $y(t)$ движения тела, соответствующих нескольким начальной скорости. В качестве параметров задается угол к горизонту и начальная скорость, известно, что конечное положение тела имеет $y_1 = 0$, началом движения считать точку $(0,0)$.
7. Построить графики изменения угла с течением времени $\alpha(t)$, соответствующие нескольким начальной скорости. В качестве параметров задается начальная скорость, известно, что конечное положение тела имеет $y_1 = 0$, началом движения считать точку $(0,0)$.

Указание: воспользуйтесь тем, что $\alpha(t) = \arctg \frac{y(t)}{x(t)}$.

8. Построить график экономичной траектории достижения телом целевой точки и целевую точку. В качестве параметров задаются координаты цели (x_1, y_1) , началом движения считать точку $(0,0)$. Траектория считается экономичной, если цель достигается с минимальной начальной скоростью.
9. Построить графики траекторий встречного движения двух тел. Два тела движутся навстречу друг другу, причем начальная точка траектории первого тела является конечной точкой траектории второго тела и наоборот, старт тел происходит одновременно. Необходимо построить траектории так, чтобы тела не столкнулись.
10. Построить графики траекторий встречного движения двух тел. Два тела движутся навстречу друг другу, причем начальная точка траектории первого тела является конечной точкой траектории второго тела и наоборот, старт тел происходит одновременно. Необходимо построить траектории так, чтобы тела встретились на некоторой точке траектории, точку стыковки выделить.

6. Список литературы

1. Потемкин В.Г. Вычисления в среде MATLAB. М., Диалог МИФИ, 2004.
2. Ануфриев И., Смирнов А., Смирнова Е. MATLAB 7 в подлиннике. С.-П., БХВ-Петербург, 2005.
3. Кетков Ю., Кетков А., Шульц М. MATLAB 7 программирование, численные методы. БХВ-Петербург, 2005.

4. Чен К., Джиблин П., Ирвинг А. MATLAB в математических исследованиях. М, Мир, 2001.
5. Поршнева С.В. Компьютерное моделирование физических процессов в пакете MATLAB. М., Горячая линия – Телеком, 2003.
6. Мартынов Н.Н. MATLAB 7: Элементарное введение. КУДИЦ-Образ, 2005г.
7. Тихоненко А.В. Компьютерный практикум по общей физике. Часть I. Кинематика.
<http://www.exponenta.ru/educat/systemat/tikhonenko/theme01/theme01.asp#1>.