

Брянский государственный аграрный университет

А.В. Кубышкина

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Учебно-методическое пособие

Брянск
2016

УДК 681.3.06

Рекомендовано

*учебно-методическим советом экономического факультета 25 мая 2016 г.,
протокол №8*

Рецензенты:

*Подольникова Е.М., зав. кафедрой менеджмента и маркетинга; канд.
эконом. наук.*

*Н.Д. Ульянова, зав. кафедрой информационных систем и технологий,
канд. эконом. наук.*

Кубышкина А.В.

*Математическое моделирование. Учебно-методическое пособие. – Брянск:
БГАУ, 2016*

В учебно-методическом пособии рассматриваются примеры реализации математических моделей экономических систем и процессов: приводятся математические постановки задач моделирования, излагаются теоретические сведения о методах решения задач, освещаются практические аспекты программной реализации моделей с использованием инструментария среды Matlab. Учебно-методическое пособие содержит варианты индивидуальных заданий для лабораторных работ.

Учебное издание предназначено для студентов, обучающихся по направлениям подготовки 09.04.03 – «Прикладная информатика», а также может быть использовано студентами других направлений подготовки и специальностей, применяющих методы вычислительной математики и инструментальные средства ППП Matlab для решения прикладных задач экономики.

ББК ****

© Брянский государственный аграрный университет, 2016

© Кубышкина А.В., 2016

ВВЕДЕНИЕ

В настоящее время знание методов экономико-математического моделирования и приобретение навыков практической технологии компьютерного моделирования экономических процессов играют всевозрастающую роль в исследовании финансовых систем, понимании причинно-следственных связей в прогнозировании, планировании, принятии решений в менеджменте. Владение профессиональными компетенциями в данной области требует освоения методов и практических приемов моделирования экономических систем и процессов на основе методов нелинейного анализа; приобретения опыта построения динамических моделей и исследования влияния обратных связей и нелинейности на характер экономических систем; изучения алгоритмов имитационного моделирования экономической динамики и возможностей их компьютерной реализации в среде современных пакетов прикладных программ.

Материал практикума тесно связан с основными разделами математических дисциплин (линейная алгебра, математический анализ, дифференциальные уравнения, исследование операций, методы оптимизации), а также дисциплин цикла «Программирование». Практикум предназначен для организации лабораторных занятий и самостоятельной работы студентов.

Особенностью методического пособия является его прикладная направленность. В процессе изучения курса студенты знакомятся с методами математического моделирования, средствами компьютерной реализации моделей экономических систем в среде Matlab, типовыми этапами общей схемы построения и проведения вычислительного эксперимента: формулировка содержательной, концептуальной, математической постановок задач моделирования, построение вычислительной схемы реализации модели, алгоритмизация и программирование, проверка адекватности модели, интерпретация и анализ результатов моделирования. Практикум включает следующие разделы: основные приемы работы с ППП Matlab, имитационное моделирование оптимальной ставки налоговых отчислений, оценка доходности облигаций и банковских операций, экономические приложения задач математического программирования (оценка до-

ходности портфеля ценных бумаг – задача Марковица), детерминированные линейные динамические модели в экономике, детерминированные нелинейные динамические модели: модели соперничества в экономике, моделирование систем массового обслуживания.

1 ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ С ПАКЕТОМ MATLAB

Matlab – интерактивный матрично-ориентированный пакет, предназначенный для выполнения научных и инженерных расчетов. Пакет содержит обширную библиотеку программ по численным методам, использует двух- и трехмерную графику, а также форматы языков высокого уровня. Техника записи циклов, использование логических отношений, вызов подпрограмм и редактирование находят непосредственное применение в среде пакета Matlab.

Найти дополнительную информацию о командах, опциях и примерах можно, воспользовавшись справкой в диалоговом режиме, литературой и прилагаемой к пакету документацией.

1.1 Работа в командном окне

1.1.1 Вход в систему Matlab. Доступ к справочной информации

При работе в операционных системах Windows 9x, Windows 2000, Windows XP вход в пакет Matlab осуществляется простым кликом левой кнопки мыши по соответствующей иконке. В версиях пакета 5.0 и выше имеется собственный встроенный редактор текста. Командное окно пакета находится в одном окне, а редактор текста – в другом.

Существуют следующие способы получить информацию о функциях пакета Matlab в процессе работы.

1. Команда **help**

Команда **help** <имя_группы> выведет на экран список функций, размещенных в этой группе с краткими пояснениями. Например, команда

```
>> help elmat
```

выведет список функций, предназначенных для создания и работы с матрицами специального вида. Ввод команды с именем определенной функции выдаст на экран описание этой функции.

2. Команда **lookfor**

Эта команда позволяет выполнить поиск m-функции по ключевому слову, при этом анализируется первая строка комментария и она же выводится на экран, если в ней встретилось ключевое слово. Например, в системе Matlab нет m-функции с именем `inverse` и поэтому на команду

```
>> help inverse
```

ответом будет

`inverse.m not found.` Однако команда `lookfor inverse` покажет все совпадения, найденные в справочных файлах.

3. Меню **Help**.

4. Обращение к Web-серверу фирмы The Mathworks.

1.1.2 Арифметические операции. Встроенные скалярные функции.

Операторы присваивания. Форматы вывода данных

Matlab можно использовать как обыкновенный калькулятор, который выполняет основные арифметические операции: + (сложение), - (вычитание), * (умножение), / (деление), ^ (возведение в степень) и содержит набор констант (`pi`, `e`, `i`).

Пример.

```
>> (2+3*pi)/2
```

```
ans =
```

```
5.7124
```

Для более сложных вычислений в пакете предусмотрены дополнительные операции. Кратко перечислим основные функции, имеющиеся в пакете. Описание других можно найти, используя справку в диалоговом режиме.

Функция	Описание	Функция	Описание
<code>abs()</code>	абсолютное значение числа	<code>cos()</code>	$\cos(x)$
<code>sin()</code>	$\sin(x)$	<code>tan()</code>	$\text{tg}(x)$
<code>exp()</code>	e^x	<code>log()</code>	$\ln(x)$ – натуральный логарифм
<code>acos()</code>	$\arccos(x)$	<code>sqrt()</code>	\sqrt{x} – извлечение кв. корня

atan()	arctg(x)	round()	Округление числа до ближайшего целого
ceil()	Наибольшее целое для данного действительного числа	floor()	Наименьшее целое число для данного действительного числа
rem()	Остаток от деления двух чисел	sign()	Знак числа (1, если знак + и -1, если знак -)

Matlab является интерпретирующим языком непосредственных вычислений. Операторы пакета обычно имеют форму:

>> имя_переменной=выражение

или просто

>> выражение

Выражение формируется из операторов, функций и имен переменных. Если имя переменной и знак равенства в левой части отсутствуют, автоматически генерируется переменная **ans**, которой присваивается результат вычислений. Matlab различает строчные и прописные буквы в именах команд, функций и переменных!

Например, для присвоения выражениям различных имен используется знак равенства.

```
>> s=3-floor(exp(3.3))
s =
-24
```

Точка с запятой позволяет задать несколько выражений и вывод значений результата после нажатия клавиши <Enter> не производится:

```
>> d=sin(5.6);           Задание d
>> a=cos(6.5);         Задание a
>> a+d                 Вывод на экран a+d
ans =
0.3453
```

Поскольку все вычисления в Matlab выполняются с двоичной точностью, формат вывода может изменяться с помощью следующих команд:

Название формата	Описание формата представления числа
<code>format short</code>	С фиксированной точкой и 4 знаками после точки (по умолчанию)
<code>format long</code>	С фиксированной точкой и 14 знаками после точки
<code>format short e</code>	Научная нотация с 4 десятичными знаками
<code>format long e</code>	Научная нотация с 15 десятичными знаками
<code>format short g</code>	С фиксированной точкой и 3 знаками после точки
<code>format long g</code>	Научная нотация с 13 десятичными знаками

Например,

```
>> format long
>> 3*cos(1.2)
ans =
    1.08707326343002
```

1.1.3 Создание матриц. Операции с матрицами и массивами.

Векторные и матричные функции

Matlab работает с одним видом объектов – числовыми прямоугольными матрицами, элементами которых могут быть в общем случае комплексные числа. Все переменные представляют собой матрицы, матрицы 1×1 интерпретируются как скаляры, матрицы с одной строкой или одним столбцом – как вектора. В системе Matlab матрицы могут быть созданы разными способами:

1. введены явно с помощью списка элементов;
2. сгенерированы встроенными операторами или функциями;
3. созданы в m-файлах;
4. загружены из внешнего файла данных.

Например, в результате выполнения оператора

```
>> A=[1 2 3;3 2 1;4 5 6]
```

или

```
>> A=[1 2 3  
3 2 1  
4 5 6]
```

Matlab создает матрицу 3×3 и присваивает ее значение переменной A.

```
A =  
1 2 3  
3 2 1  
4 5 6
```

В пакет Matlab встроен ряд функций, позволяющих создавать матрицы специального вида. Приведем некоторые из них:

1. функция **rand(n)** создает матрицу размера $n \times n$ (функция **rand(m,n)** – матрицу размера $m \times n$), каждый элемент которой – случайное число с равномерным законом распределения в диапазоне $[0,1]$;

2. функция **magic(n)** создает матрицу размера $n \times n$, которая является магическим квадратом;

3. функция **zeros(m,n)** создает нулевую матрицу размера $m \times n$;

4. функция **ones(m,n)** создает матрицу размера $m \times n$, каждый элемент которой равен единице;

5. функция **diag(x)** создает матрицу, у которой на главной диагонали стоят элементы вектора x ;

6. функция **diag(A)** создает матрицу, у которой на главной диагонали стоят диагональные элементы матрицы A .

Ссылки на отдельные элементы матриц и векторов осуществляются с помощью индексов в круглых скобках.

Приведем несколько примеров задания матриц, выделения элементов и подматриц.

```
>> Z=zeros(4,5); % Создание нулевой матрицы размера 4x5.  
>> O=ones(3,2); % Создание единичной матрицы размера 3x2.  
>> X=0:0.5:2 % Вывод на экран вектора X размера 1x5.
```

```

X =
    0    0.5000    1.0000    1.5000    2.0000
>> Y=sin(X); % Создание вектора Y размера 1x5, каждый элемент
которого является sin от X.
>> A(2,3)
ans =
    1 % Выделение элемента (2,3) из матрицы A.
>> A(8) % Альтернативное выделение элемента(2,3)
ans = % как 8-го по счету, если записать матрицу в
    1 % один вектор-столбец.
>> A(1:2,1:3) % Выделение подматриц A.
ans =
    1    2    3
    3    2    1
>> A([1 2],[1 3])
ans =
    1    3
    3    1
>> A(2,2)=round(tan(8)) % Замена элемента матрицы A.
ans =
    1    2    3
    3 -7    1
    4    5    6
>> A(:,2) % Выделение 2-го столбца.
ans =
    2
   -7
    5
>> A(2,:) % Выделение 2-й строки.
ans =
    3 -7    1

```

В пакете Matlab доступны следующие матричные операции: + (сложение), - (вычитание), * (умножение), ^ (возведение в степень, применима

только для квадратных матриц!), ' (транспонирование), / (правое деление), \ (левое деление). Если размерность матриц не соответствует выполняемой операции, то система генерирует сообщение об ошибке. Приведенные операции могут стать поэлементными, если перед ними поставить точку. Приведем примеры использования матричных операций.

```
>> D=[1 2;4 8];
>> C=D'
C =
     1     4
     2     8
>> 5*(D*C)^2
ans =
    2125    8500
    8500    3400
>> C^2 % Произведение матриц C*C.
ans =
     9    36
    18    72
>> C.^2 % Квадрат каждого элемента матрицы C.
ans =
     1    16
     4    64
>> sin(D./2)
ans =
    0.4794    0.8415
    0.9093    0.7568
```

Часто возникает необходимость использования векторных или матричных функций.

Аргументами векторных функций являются векторы (строки или столбцы). Если в качестве аргумента указана матрица размера $m \times n$, то данная функция действует постолбцово, т.е. результатом действия является вектор-столбец, каждый элемент которого – результат действия этой функции на соответствующий столбец. Например, максимальный элемент прямоугольной

матрицы находится с помощью команды `max(max(A))`. Наиболее употребительными матричными функциями являются:

- `inv(X)` – обратная матрица,
- `det(X)` – определитель матрицы,
- `size(X)` – размерность матрицы,
- `norm(X)` – норма вектора или матрицы,
- `rank(X)` – ранг матрицы,
- `cond(X)` – число обусловленности.

1.2 Циклы. Условные операторы и операторы отношения

Операторы управления Matlab и операторы отношения при использовании работают также, как и в большинстве языков программирования.

Оператор цикла `for` в общем виде записывается как:

```
for <переменная_цикла>=<выражение_цикла>  
<Выполняемые_операторы>  
end
```

Оператор цикла `while` записывается следующим образом:

```
while <условие>  
<Выполняемые_операторы>  
end
```

Условный оператор `if` имеет следующий синтаксис:

```
if <условие>  
<Выполняемые_операторы1>  
else  
<Выполняемые_операторы2>  
end
```

В пакете Matlab возможно также множественное ветвление.

В пакете Matlab используются следующие операторы отношений и логические операторы:

<code>==</code>	равно	<code>~</code>	не
<code>~=</code>	не равно	<code>&</code>	и

<	меньше			ИЛИ
>	больше			
<=	меньше или равно			
>=	больше или равно			

Приведем несколько примеров использования циклов и условий.

Пример. Найдем значение 5!.

```
>> a=1;
>> for i=1:5
a=a*i;
end
>> a
a =
    120
```

Пример. Для различных значений x , полученных делением на 3 чисел от 0 до 10, выведем значения квадратов и кубов этих чисел.

```
>> n=10;
>> k=0;
>> while k<=n
x=k/3;
disp([x x^2 x^3])
k=k+1;
end
```

Пример. В зависимости от s присвоим переменной a разные значения.

```
>> s=10;
>> if (s<0)|(s>10)
a=s^2+5;
else
a=0;
end
>> a
a =
    0
```

1.3 М – файлы

Matlab может последовательно выполнять последовательность операторов, записанных в файл на диске. Имена таких файлов имеют вид <имя>.m и такие файлы называются m-файлами. Большая часть работы в Matlab заключается в создании, редактировании и выполнении таких m-файлов. Существуют два типа m-файлов: файлы-программы (сценарии) и файлы-функции.

1.3.1 Файлы-программы

Файлы-программы состоят из последовательности обычных операторов пакета Matlab. Если m-файл с таким сценарием имеет имя, например, start.m, то команда start, введенная в командной строке, вызовет выполнение последовательности операторов этого файла. Переменные в программе являются глобальными и могут изменить значения переменных с теми же именами в командном окне. В таких файлах легко исправить ошибки, внутри файла можно ссылаться на другие m-файлы и рекурсивно на этот же файл.

Приведем пример программы, которая подсчитывает число отрицательных элементов массива, сохраненной под именем script.m:

```
S=input('Введите массив S'); % команда input для запроса массива у пользователя
N=length(S); % считываем размерность массива S
count=0; % инициализация счетчика
for i=1:N
    if S(i)>0
        count=count+1; % подсчет числа отрицательных элементов
    end
end
disp(count); % команда disp для вывода на экран
```

Вызов в командной строке

```
>> script
```

приведет к выполнению операторов программы:

```
>> script
```

```
Введите массив S    [3 -7 5 -8 0 -42]
```

1.3.2 Файлы-функции

Файлы-функции дают возможность расширять Matlab, поскольку определенные пользователем функции имеют тот же статус, что и другие функции Matlab. Переменные по умолчанию являются локальными и не влияют на имена и значения переменных в текущей рабочей области Matlab.

Общий вид задания:

```
function [x1,x2,...]=name(y1,y2,...)
```

<тело функции>

Первая строка функции содержит объявление выходных аргументов, имени функции и входные аргументов, без этой строки файл является программой, но не функцией.

Совпадение имени функции и имени файла является обязательным условием в Matlab. Символ % указывает на то, что вся строка после него является комментарием и игнорируется при выполнении программы.

Приведем пример функции funpr.m, которая производит подсчет количества элементов массива, кратных заданному числу и вычисляет сумму таких элементов:

```
function [n,S]=funpr(P,d)
N=length(P); % определение размерности массива P
n=0; % обнуление счетчика
S=0; % обнуление суммы
for i=1:N
    if rem(P(i),d)==0 % поиск элементов массива, кратных d
        n=n+1; % подсчет количества элементов
        S=S+P(i); % подсчет суммы
    end
end
```

Вызов в командной строке (после задания массива P):

```
>> [n,S]=funpr([2 -9 8 5 0 -6],2)
n = 4
S = 4
```

присвоит переменным n и S значения выходных параметров.

Для вычисления значения функции, имя которой передается в функцию через строковую переменную, часто используется функция `feval`, аргументами которой являются имя функции и координата точки, в которой вычисляется значение данной функции.

Приведем пример использования функции `feval`. Зададим свою функцию `funpr2` изменений аргумента:

```
function z=funpr2(x)
z=x.^3-3*x.^2+2*x+3;
```

И пусть, например, требуется обратиться к этой функции в другом файле-функции, вычисляющей значение функции в середине указанного отрезка:

```
function z=aver(funpr2,a,b)
c=(a+b)/2;
z=feval(funpr2,c)
```

Результат обращения к функции `aver` будет иметь вид:

```
>> aver('funpr2',-2,2)
z = 3
ans = 3
```

1.3.3 Текстовые строки. Сообщения об ошибках

Текстовые строки вводятся в Matlab в виде текста в одинарных кавычках: `m='Mistake'`. Вывод текстовой строки осуществляется при помощи оператора `disp`. Оператор `disp(m)` выведет на экран сообщение:

```
Mistake
```

Сообщение об ошибках лучше выводить при помощи функции `error`, так как после обращения к данной функции выполнение `m`-файла будет прекращено. Для выхода из цикла (и из программы) можно воспользоваться командой `break`. В `m`-файле запрос на ввод данных можно организовать при помощи оператора `input`. При вводе оператора `m=input('введите размерность массива')` на экран выводится запрос и выполнение работы файла приостанавливается до ввода требуемых данных с клавиатуры. После нажатия клавиши <Enter> данные присваиваются переменной `m`.

1.4 Графики

Пакет Matlab позволяет строить двух- и трехмерные графики кривых и поверхностей. Дополнительные возможности и описание графиков можно найти, используя справку в диалоговом режиме.

1.4.1 Построение двумерных графиков

Команду `plot` используют для построения графика двумерной функции. Следующий пример иллюстрирует, как построить графики функций

$$y = \sin(x) + x^2 \text{ и } y = 10\sin^2(x) \text{ на отрезке } \left[-\frac{3\pi}{2}, \frac{3\pi}{2}\right]:$$

```
>> x=-3*pi/2:0.01:3*pi/2;  
>> y1=sin(x)+x.^2;  
>> y2=10*(sin(x)).^2;  
>> plot(x,y1,x,y2)  
>> grid on
```

В первой строке программы задается область с шагом 0.01. В следующих двух строках задаются две функции. Первые три строки заканчиваются точкой с запятой – это предотвращает вывод векторов x , y , z на экран. Команда `plot` строит графики сплошной линией (рисунок 1.1).

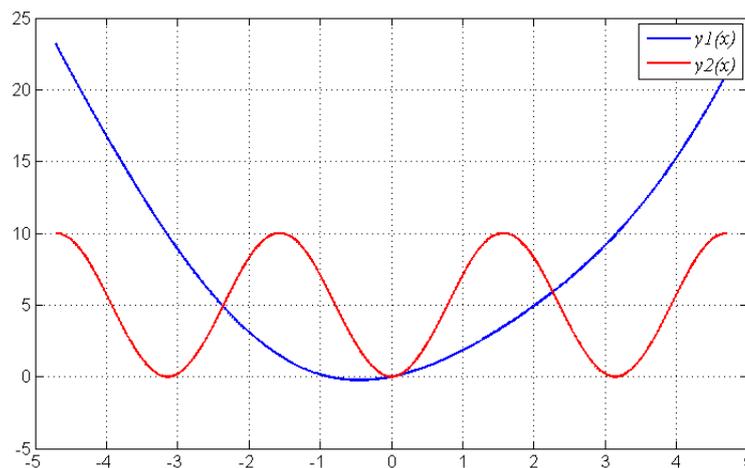


Рис. 1.1. Графики двух функций, построенные в одной системе координат.

Команда `plot` позволяет легко задать стиль (тип маркера и тип линий) и цвет линий, например,

```
>> plot(x, f, 'k-', x, g, 'k:')
```

осуществляет построение первого графика сплошной черной линией, а второго – черной пунктирной.

В таблице приведены возможные значения аргументов с указанием результата.

Цвет		Тип маркера		Тип линии	
y	желтый	.	точка	-	сплошная
m	розовый	o	кружок	:	пунктирная
c	голубой	x	крестик	-.	штрихпунктирная
r	красный	+	знак «плюс»	--	штриховая
g	зеленый	*	звездочка		
b	синий	s	квадрат		
w	белый	d	ромб		
k	черный	v, ^, <, >, v	треугольники		
		p, h	звезды		

Удобство оформления графиков зависит от дополнительных элементов оформления: координатной сетки, подписей к осям, заголовка и легенды. Сетка наносится командой **grid on**, функции **xlabel**, **ylabel** служат для размещения подписей к осям, **title** – для заголовка, **legend** – сопровождение легендой.

1.4.2 Графики функций двух переменных

Для отображения функции двух переменных следует:

1. сгенерировать матрицы с координатами узлов сетки на прямоугольной области определения функции;
2. вычислить функцию в узлах сетки и записать полученные значения в матрицу;
3. использовать одну из встроенных функций пакета.

Сетка генерируется функцией **meshgrid**, вызываемой с двумя входными аргументами:

```
>> [X,Y]=meshgrid(-2:0.1:2, -2:0.1:2);
>> Z=sin(X.^2).*cos(Y.^2).(1-X.^2).(1-Y.^2);
```

Функция **surf** строит каркасную поверхность графика функции и заливает каждую клетку определенным цветом, зависящим от значения функции в точках, соответствующим углам клетки (рисунок 1.2):

```
>> surf(X,Y,Z)
```

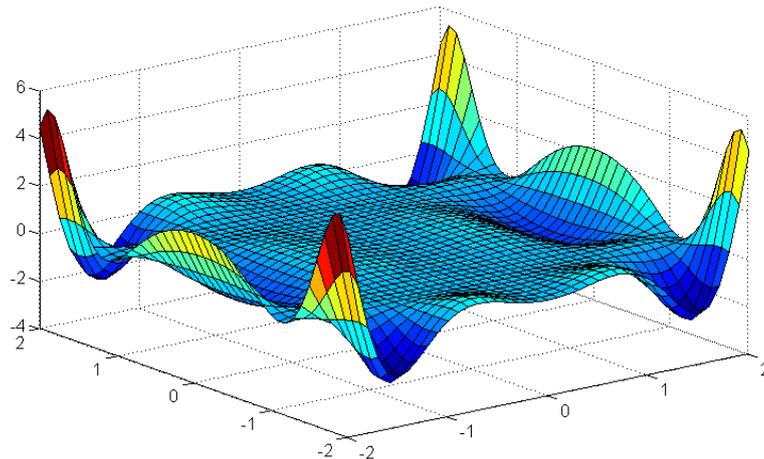


Рис. 1.2. Пример визуализации графика функции двух переменных с помощью команды `surf`.

Команда `shading flat` позволяет убрать каркасные линии, а команда `shading interp` позволяет получить поверхность плавно-залитую цветом. Команда `colorbar` выводит рядом с графиком цветовую шкалу. Линии уровня можно построить с помощью команд `meshc`, `surfс`, которые имеют те же входные аргументы, что и `mesh`, `surf`.

1.4.3 Анимированные графики

При изучении движения точки на плоскости или в пространстве полезно не только построить траекторию точки, но и следить за движением точки по этой траектории. Для построения анимированных графиков применяются функции `comet` и `comet3`.

```
>> t=[0:0.001:10];  
>> x=sin(t)./(t+1);  
>> y=cos(t)./(t+1);  
>> comet(x,y)
```

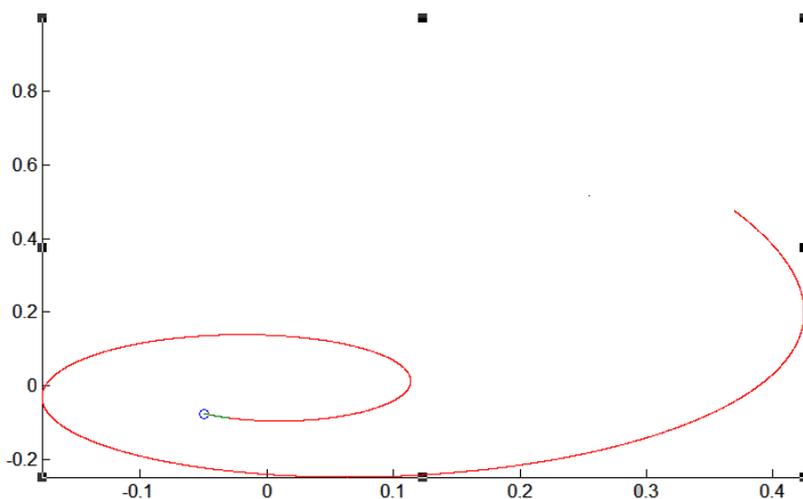


Рис.1.3. Визуализация анимированного графика (последний момент времени).

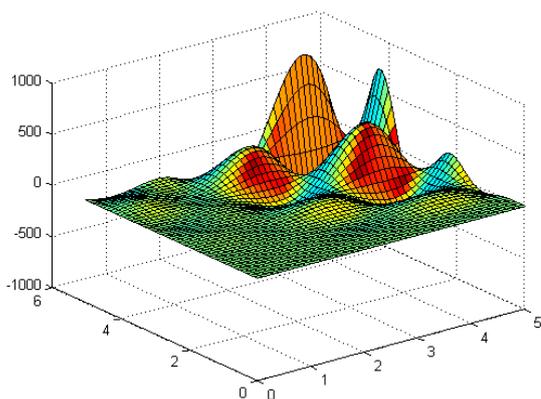
1.4.4 Работа с несколькими графиками

Обычно графики выводятся в специальное графическое окно с заголовком **Figure 1**. При следующем построении графика новый график выводится в то же самое окно. Matlab представляет возможность вывода: каждого графика в свое окно, нескольких графиков в одно окно, в пределах одного окна нескольких графиков, каждого в своих осях.

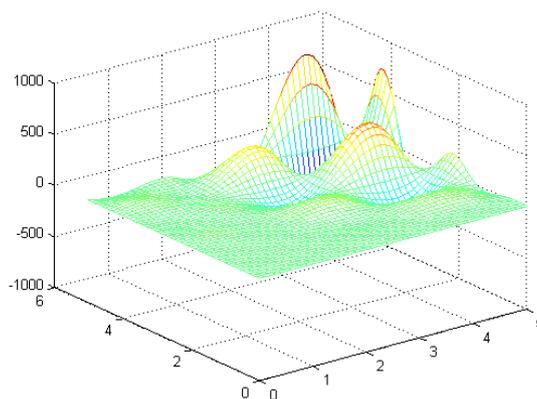
Следующая последовательность команд построит графики функций в отдельных окнах **Figure1** и **Figure2**, причем последнее окно будет текущим и все последующие изменения будут соответственно отражаться именно на этом окне. Чтобы сделать текущим другое окно следует два раза щелкнуть по нему мышкой и вернуться к продолжению команд:

```
>> Z=3*sin(2*X).*cos(3*Y).(1-X.^2).*Y.^2;
>> figure
>> surf1(X,Y,Z)
>> figure
>> mesh(X,Y,Z)
```

Результат на экране (рисунок 1.4):



a



б

Рис. 1.4. Графический вывод функции двух переменных в различные графические окна (*a* – с помощью команды `surf`, *б* – с помощью команды `mesh`).

Matlab позволяет разместить в графическом окне несколько осей и вывести на них различные графики. Для это можно воспользоваться функцией `subplot`, которая располагает оси в виде матрицы с тремя параметрами: число подграфиков по вертикали, горизонтали, номер подграфика. Следующий пример иллюстрирует использование этой функции.

```
>> [X,Y]=meshgrid(-1:0.05:1,0:0.05:1);
>> Z=4*sin(2*pi*X).*cos(1.5*Y*pi).*(1-X.^2).*Y.*(1-Y);
>> subplot(2,2,1)
>> mesh(X,Y,Z)
>> title('mesh')
>> subplot(2,2,2)
>> surf(X,Y,Z)
>> title('surf')
>> subplot(2,2,3)
>> meshc(X,Y,Z)
>> title('meshc')
>> subplot(2,2,4)
>> surf1(X,Y,Z)
>> shading interp
>> title('surf1')
>> colormap (cool)
```

Результат показан на экране (рисунок 1.5.):

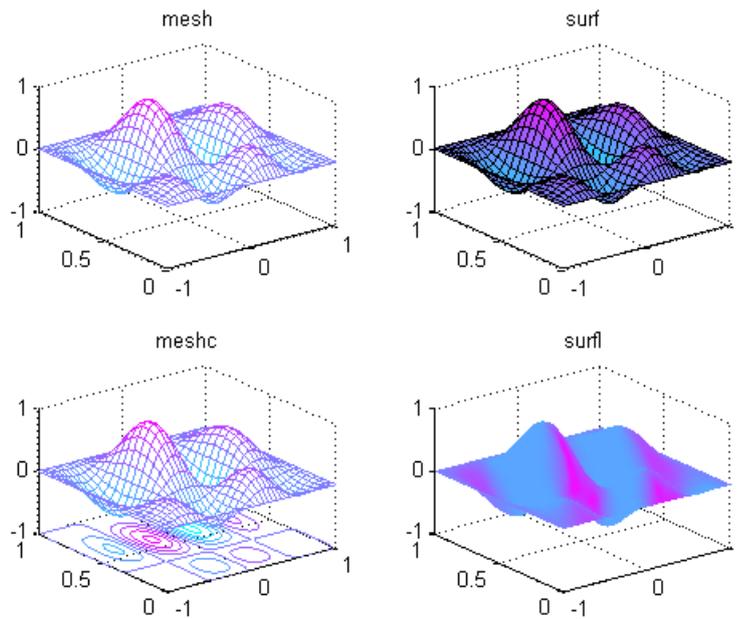


Рис. 1.5. Вывод функции двух переменных на одно графическое окно (с помощью различных функций).

1.5 Диаграммы и гистограммы

Наглядным способом представления векторных и матричных данных являются диаграммы и гистограммы. Значение элемента вектора пропорционально высоте столбика диаграммы или площади сектора диаграммы. Гистограммы используются для получения информации о распределении данных по заданным интервалам.

Отображение вектора в виде столбчатой диаграммы осуществляется функцией **bar**.

```
>> data=[2.2 2.7 3.2 3.7 2.5 3.3 3.1 1.5 1.2 1.2];
>> bar(data)
```

Результат на экране (рисунок 1.6).

Разметку горизонтальной оси можно задать вектором с возрастающими значениями, что учитывается в первом аргументе функции **bar**. Выбор ширины столбцов осуществляется заданием третьего дополнительного аргумента. Например,

```
>> t=-1:0.1:1;
>> x=sin(t).*exp(t);
```

```
>> bar(t,x,1.0)
```

Результат на экране (рисунок 1.7).

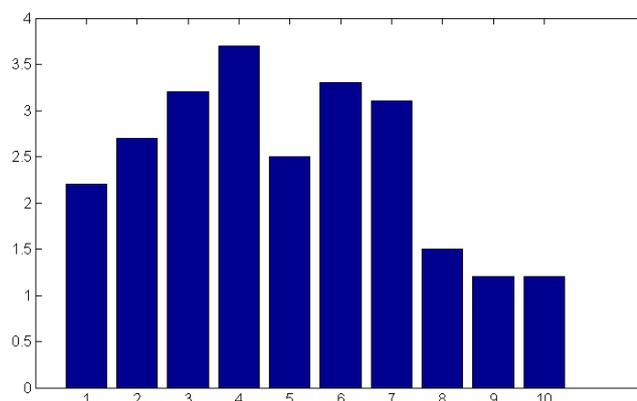


Рис. 1.6. Визуализация столбчатой диаграммы.

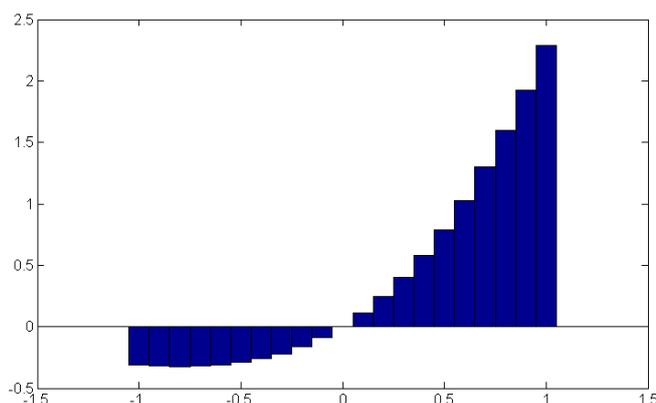


Рис. 1.7. Визуализация столбчатой диаграммы с учетом вариации параметров.

Для учета вклада каждого из элементов вектора в общую сумму его элементов оказываются полезными круговые диаграммы:

```
>> data=[1.2 1.7 2.2 2.4 2.5 1.3 1.1 0.2];  
>> pie(data)
```

Результат на экране (рисунок 1.8).

Визуализация векторных данных может быть осуществлена при помощи **pie3** и **bar3**, которые строят трехмерные круговые и столбчатые диаграммы.

Для получения наглядного представления о распределении данных служит функция **hist**.

```
>> data=randn(100000,1);
```

```
>> hist(data,50)
```

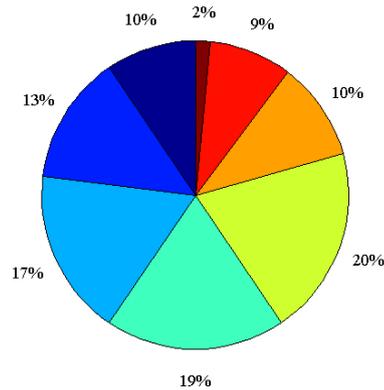


Рис. 1.8. Визуализация круговой диаграммы.

Данная последовательность команд заполняет вектор `data` числами, распределенными по нормальному закону, разбивает интервал, которому они принадлежат, на пятьдесят равных частей и строят гистограмму попадания в каждый из интервалов (рисунок 1.9).

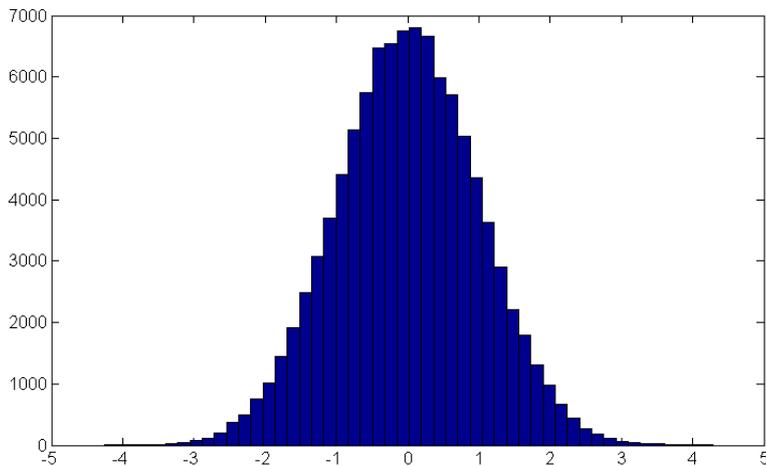


Рис. 1.9. Визуализация гистограммы.

Предположим в матрице `DATA`, состоящей из четырех строк и трех столбцов, содержатся результаты измерений трех величин за четыре момента времени. Для построения столбчатой диаграммы используется команда `bar`:

```
>> DATA=[1.3 1.5 1.05;2.7 2.5 3.1;2 2.9 2.3;5.2 5.7 5.1];  
>> bar(DATA)
```

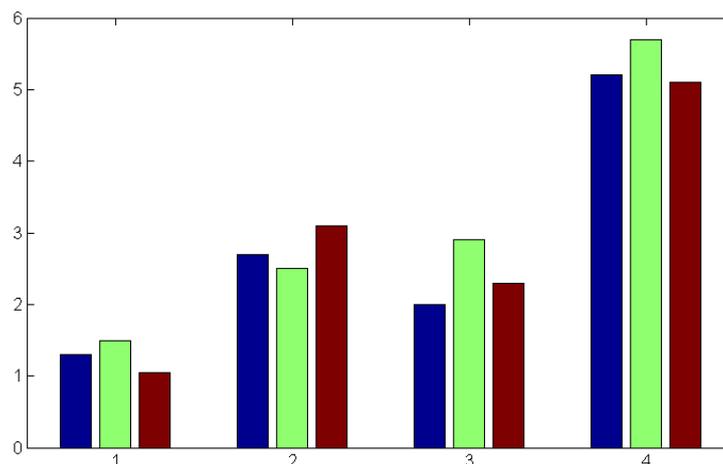


Рис. 1.10. Визуализация столбчатой диаграммы (при наличии нескольких измерений).

1.6 Индивидуальные задания

№ варианта	Задание 1
1	Вычислить минимальный элемент произвольно введенного массива C и его номер среди элементов $C_i < 0$.
2	Для матрицы $A(N, M)$ заменить элементы выше главной диагонали и на ней: отрицательные на ноль, положительные – на 1.
3	Вычислить минимальный элемент произвольно введенного массива C и его порядковый номер среди элементов $C_i > 0$.
4	Для матрицы $A(N, M)$ заменить элементы ниже главной диагонали и на ней: отрицательные на 2, положительные – на -2.
5	Вычислить среднее геометрическое элементов произвольно введенного массива Y таких, что $Y_i > 0$.
6	Для матрицы $A(N, M)$ найти максимальный и минимальный элемент из диагональных и поменять их местами.
7	Вычислить максимальный элемент произвольно введенного массива C и его порядковый номер среди элементов $C_i > 0$.
8	Для матрицы $A(N, M)$ поменять первый столбец с последним, второй с предпоследним и т. д.

9	Вычислить среднее арифметическое элементов произвольно введенного массива S таких, что $S_i > 0$.
10	Для матрицы $A(N, M)$ найти максимальный и минимальный элемент из элементов, расположенных под главной диагональю и поменять их местами.
11	Вычислить сумму и количество элементов произвольно введенного массива M таких, что $M_i > 0$.
12	Для матрицы $A(N, M)$ вычислить сумму и количество положительных элементов, кратных 3.
13	Вычислить среднее арифметическое элементов произвольно введенного массива S таких, что $0 < S_i < 10$.
14	Для матрицы $A(N, M)$ вычислить сумму и количество положительных элементов, кратных 3.
15	Найти максимальный и минимальный элемент произвольно введенного массива и поменять их местами.
16	В каждой строке матрицы $A(N, M)$ вычислить среднее геометрическое элементов этой строки таких, что $0 < a_{i,j} < 10$.
17	Вычислить минимальный элемент произвольно введенного массива C и его порядковый номер среди элементов $C_i > 0$.
18	Для матрицы $A(N, M)$ поменять все элементы главной диагонали на противоположные по знаку.
19	Вычислить среднее арифметическое элементов произвольно введенного массива L таких, что $L_i > 0$.
20	Для матрицы $A(N, M)$ найти максимальный и минимальный элемент из диагональных и поменять их местами.

Задание 2. Построить графики функций одной переменной на отрезке $[0.01, 2\pi]$ $f(x) = a \frac{\sin bx}{x}$, $g(x) = c \cdot e^{-dx} \cdot \cos fx$. Выведите графики различными способами: а) в отдельные графические окна, б) в одно окно на одни оси, в) в

одно окно на отдельные оси. Дайте заголовки, разместите подписи, легенду, используйте различные цвета, стили линий и типы маркеров, нанесите сетку.

№ варианта	Параметры	№ варианта	Параметры
1, 11	$a = 1, b = 1, c = 1,$ $d = 1, f = 1$	2, 12	$a = 0.5, b = 2, c = 1,$ $d = 0.5, f = 2$
3, 13	$a = 0.5, b = 1.5, c = 0.5,$ $d = 1, f = 1$	4, 14	$a = 1.8, b = 1, c = 1.2,$ $d = 1, f = 1$
5, 15	$a = 1.9, b = 1, c = 0.8,$ $d = 1, f = 1$	6, 16	$a = 1, b = 2, c = 1,$ $d = 1, f = 2$
7, 17	$a = 0.5, b = 1.3, c = 2,$ $d = 1, f = 1.6$	8, 18	$a = 3, b = 2, c = 1.2,$ $d = 0.9, f = 2$
9, 19	$a = 1.6, b = 1.5, c = 5,$ $d = 1, f = 1.8$	10, 20	$a = 1, b = 3, c = 0.9,$ $d = 4, f = 1.4$

Задание 3. Визуализируйте график функции двух переменных на прямоугольной области определения: $z(x, y) = (\sin ax^2 + \cos by^2)^c$, $x \in [-1, 1], y \in [-1, 1]$. Выведите графики двумя различными способами (по выбору) в одно графическое окно. Параметры a, b, c взять из таблицы к заданию 2.

Задание 4. Фирма, занимающаяся реализацией сезонного товара, имеет следующие средние доходности $S = (50, 45, 51, 55, 79, 90, 110, 100, 92, 76, 79, 88)$. Каждый элемент вектора S соответствует прибыли, полученной фирмой за соответствующий календарный месяц. Построить столбчатую и круговую диаграммы, интерпретирующие модельное распределение для данного примера (диаграммы вывести в одно графическое окно).

2 ОПТИМАЛЬНАЯ СТАВКА НАЛОГА. РАСЧЕТЫ ПО ОБСЛУЖИВАНИЮ КРЕДИТОВ

2.1 Имитационная модель определения оптимальной ставки налога

2.1.1 Концептуальная постановка задачи

В работе методами имитационного моделирования найдём оптимальную ставку налогообложения прибыли. Используем простейший метод поиска на сетке факторов. Планирование экспериментов заключается в изменении факторов с постоянным шагом и построении экспериментальных графиков зависимости ставки от факторов.

Сформулируем концептуальную постановку задачи. Государство стремится увеличить налоги, чтобы наполнить бюджет для выполнения своих социально-экономических и оборонных функций. Бизнес жалуется, что налоговое бремя велико и налоговые ставки надо уменьшить. Экономисты утверждают, что большие налоги сдерживают развитие экономики, а значит и будущее наполнение бюджета. Таким образом, ставится задача определения оптимальной величины налоговой ставки. Примем рабочую гипотезу: поступления в бюджет за определенный период времени будут наибольшими не при максимальной, а при оптимальной для бюджета ставке налога, т.е. с ростом налоговой ставки поступления в бюджет будут увеличиваться, а затем уменьшаться. Цель работы состоит в исследовании зависимости поступлений в бюджет от величины налоговой ставки.

Несмотря на массу налогов и терминов, источником развития бизнеса и источником налогового наполнения бюджета в конечной инстанции является прибыль, т.е. превышение доходов над расходами. Будем считать, что законодатель объявляет ставку налога. Бюджет получает налоговые отчисления от прибыли предприятий. Предприятия по налоговой ставке на прибыль отчисляют средства в бюджет.

Фирмы обладают собственным капиталом, получают прибыль, отчисляют по налоговой ставке средства в бюджет. Постналоговая прибыль как нераспре-

деленная прибыль полностью включается в собственный капитал фирмы. Дивиденды не выплачиваются, никаких других отчислений от прибыли не производится. Вся прибыль распределяется только на два потока: в бюджет, а остаток в собственный капитал фирмы.

2.1.2 Математическая постановка задачи.

Описание параметров и переменных модели

Сумма налоговых поступлений в бюджет B за моделируемый период времени представлена формулой:

$$B^i = \sum_{t=t_0}^{tN} B_t, \quad B_t = P_t \cdot tx, \quad (2.1)$$

где B_i – сумма, поступивших в бюджет средств от начала моделирования к концу года $t=i$, руб., B_t – сумма, поступивших в бюджет средств за год t , руб., P_t – доналоговая прибыль, полученная предприятием за год t , руб/год, tx – ставка налога на прибыль (параметр модели), t – время, год (для запасов этот момент – конец года, для потоков это интервал времени определенного года), t_0 – начальный год моделирования, tN – последний год моделирования.

Прибыль предприятия за t год равна:

$$P_t = C_{t-1} \cdot R, \quad (2.2)$$

где R – рентабельность капитала предприятия (параметр модели), C_{t-1} – капитал сальдо в предшествующий год, руб.

Капитализируемый предприятием за период моделирования остаток прибыли:

$$S_t = P_t \cdot (1 - tx). \quad (2.3)$$

Капитал сальдо в текущий период определяется как:

$$C_t = C_{t-1} + S_t. \quad (2.4)$$

Таким образом, при требуется инициализировать параметры модели: налоговая ставка, рентабельность, начальный капитал фирмы, временной интервал моделирования и далее провести имитационное моделирование процесса развития предприятия и накопление налоговых средств в бюджете во времени

следующим образом. Для каждого такта времени (один год) следует определить: прибыль P_t по формуле (2.2), капитализируемый предприятием остаток S_t по формуле (2.3), капитал сальдо C_t согласно (2.4), текущую B_t и итоговую сумму поступлений в бюджет к году i : B^i по соотношению (2.1).

2.1.3 Реализация модели определения налоговых поступлений в бюджет

Последовательность вычислений характеристик модели определения налоговых поступлений в бюджет в постановке (2.1)-(2.4) представлена следующим листингом.

```
% инициализация параметров модели
tx=0.7;
R=1.2;
C0=100;
n=10;
P=zeros(n,1);
S=zeros(n,1);
C=zeros(n,1);
B=zeros(n,1);
Sum=zeros(n,1);
% вычисление характеристик
P(1)=C0*R;
S(1)=P(1)*(1-tx);
C(1)=C0+S(1);
B(1)=P(1)*tx;
Sum(1)=B(1);
SSS=0;
for i=2:n
    P(i)=C(i-1)*R; % прибыль предприятия за год
    S(i)=P(i)*(1-tx); % капитализируемый остаток прибыли
    C(i)=C(i-1)+S(i); % капитал сальдо в текущий период
    B(i)=P(i)*tx; % сумма налоговых поступлений
    Sum(i)=Sum(i-1)+B(i); % итоговая сумма поступлений за период
end
SSS=sum(B);
```

```
t=1:10;
```

```
plot(t,B,'ro',t,B)
```

Здесь в качестве параметров модели установлены:

$R=120\%$ – рентабельность капитала предприятия,

$tx=70\%$ – ставка налога на прибыль,

$C0=100$ усл. ед. – капитал сальдо на начальный момент времени.

Величина SSS соответствует суммарному значению налоговых выплат в бюджет за рассматриваемый период. Динамика изменения налоговых поступлений в бюджет показана на рисунке 2.1.

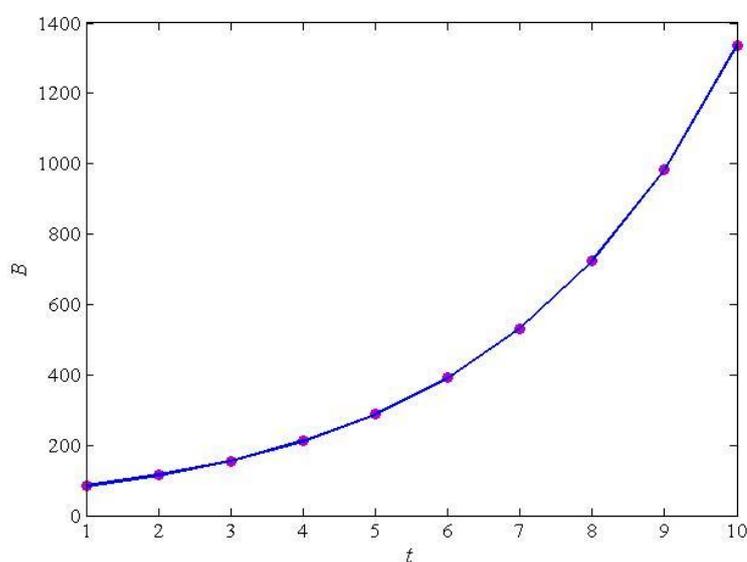


Рис. 2.1. Результат моделирования суммарного значения налоговых выплат с течением времени.

2.2 Использование функций ППП Matlab

для выполнения расчетов по обслуживанию кредитов

ППП Matlab содержит ряд инструментариев, позволяющих проводить финансовые и экономические расчеты, одним из которых является Financial Toolbox. Данный пакет позволяет рассчитывать потоки платежей и анализировать их, в том числе оценивать инвестиционные проекты. Рассмотрим функции пакета по обслуживанию долгов и кредитов.

Пусть требуется принять решение о погашении долга 150 тыс. руб. в течение трех лет. На кредитном рынке процентная ставка составляет 20 % годово-

вых. Рассмотрим схему погашения равными ежегодными платежами. При равных выплатах 20 % от суммы составляют проценты по сумме оставшегося долга, а остальная часть идет в счет погашения суммы займа. Для иллюстрации частичного погашения кредита срочными платежами (одинаковыми суммами за год) выберем сумму платежа 70 тыс. руб. Хотя такой суммы недостаточно для погашения ссуды, используем такой пример для демонстрации функций пакета:

Год	Остаток по займу, т.р.	Платеж по сумме займа, т.р.	Проценты по остатку, т.р.	Полный платеж за год, т.р.
1	150	40	30	70
2	110	48	22	70
3	62	57.6	12.4	70
Итоги	4.4	145.6	64.4	210

Вся структура погашения кредита может быть вычислена с помощью функции `amortize`, вызов которой имеет следующий синтаксис:

`[за_кредит, за_проценты, долг, платеж]=amortize(ставка, число периодов, сумма, остаток, момент)`

Первые три аргументы обязательны:

ставка – процентная ставка по кредиту в один период,

число периодов – полное число периодов расчета по займу,

сумма – сумма полученных заемщиком денежных средств.

Два следующих параметра необязательны: *момент* (по умолчанию равен 0) – флаг, равный 1, если выплаты производятся в начале периода, 0 – в конце, *остаток* (по умолчанию равен 0) – та сумма, которая останется непоплаченной.

Первые три выходных параметров являются одномерными массивами размерности входного параметра *число периодов* и содержат:

за_кредит – платежи по сумме кредита за каждый период,

за_проценты – выплаты по начисленным процентам за остаток кредита в каждый период,

долг – оставшийся долг после текущего платежа,

платеж – сумма общей разовой выплаты.

```
[SC, SI, B, P]=amortize(0.2, 3, 150, -4.4)
```

Результат на экране:

```
SC =   40.0000   48.0000   57.6000
SI =   30.0000   22.0000   12.4000
B =  110.0000   62.0000   4.4000
P =    70.0000
```

Величину процентной ставки платежа за один период можно вычислить с использованием функции **annurate**:

ставка=annurate(число периодов, платеж, сумма, остаток, момент)

Пример вызова: `annurate(3, 70, 150)`

Результат на экране:

```
ans =    0.1891
```

Расчет суммы одноразового платежа дает функция

платеж=pauper(ставка, число периодов, сумма, остаток, момент)

Пример вызова (платеж для полного погашения кредита без остатка):

```
pauper(0.2, 3, 150)
```

Результат на экране:

```
ans =    71.2088
```

2.3 Индивидуальные задания

1. Провести модельные расчеты по определению налоговых поступлений в бюджет и имитационный эксперимент по определению оптимальной налоговой ставки.

1.1 Провести однофакторный имитационный эксперимент: построить графики временных зависимостей всех переменных модели определения налоговых поступлений в бюджет в интервале времени от 1 года до 10 лет при значении параметров согласно варианту:

Вариант	Налоговая ставка $tx, \%$	Рентабельность $R, \%$	Начальный капитал фирмы $C_0, \text{млн. руб.}$
1	40	110	150
2	50	115	200
3	60	117	250
4	70	120	300
5	45	103	350
6	50	105	150
7	50	110	200
8	60	115	250
9	70	120	300
10	70	125	350

1.2 Провести двухфакторный имитационный эксперимент: исследовать зависимость бюджетно-оптимальной ставки от эффективности работы фирмы. С этой целью вычислить и провести анализ зависимости налоговых поступлений в бюджет (SSS) за фиксированный период времени (10 лет) от величины налоговой ставки на прибыль предприятий (значение tx варьировать от 10% до 70%). Вывести несколько графических зависимостей при значениях рентабельности R в интервале от 40 до 120 с шагом 20.

Сделать вывод относительно оптимальной ставки налогообложения при фиксированном значении рентабельности.

1.3 Оформить отчет по работе.

2. Используя встроенные функции пакета Matlab выполнить расчет временной динамики погашения кредита при следующих начальных данных:

- сумма денежных средств – 10000 усл.ед.,
- число периодов – 10,
- процентная ставка по кредиту – 17 %,
- нулевой остаток.

3 ОЦЕНКА ДОХОДНОСТИ БАНКОВСКИХ ОПЕРАЦИЙ. ОЦЕНКА ДОХОДНОСТИ ОБЛИГАЦИЙ

Важнейшей задачей в процессе принятия решений в управлении корпоративным капиталом является оценка доходности банковских операций и ценных бумаг, в частности, облигаций.

3.1 Математическая модель оценки доходности облигации: концептуальная и математическая постановка задачи

Сформулируем концептуальную постановку задачи моделирования доходности облигации. Предположим, что куплена облигация с определенной номинальной стоимостью через определенный период после ее выпуска, определенным сроком погашения и с определенной купонной ставкой. Необходимо найти значение доходности такой облигации при условии, что ее владелец погашает ее в указанный срок.

Выделим характерные параметры модели:

A_n – текущая стоимость облигации (усл. ден. ед.);

P – номинальная стоимость (усл. ден. ед.);

N – срок, прошедший после выпуска облигации (в годах);

T – срок погашения (в годах);

$n = T - N$ – срок, оставшийся до погашения облигации (в годах);

k – купонная процентная ставка (в долях от ед.);

$I = P \cdot k$ – величина купонных платежей (произведение номинальной стоимости на купонную процентную ставку) (усл. ден. ед.);

Переменная модели: i – процентная ставка.

Для анализа доходности облигации воспользуемся математической моделью оценки доходности облигации, согласно которой текущая стоимость определяется выражением:

$$D_k = \frac{I}{(1+i)^k}, \quad (3.1)$$

где D_k – текущее (приведённое) значение суммы купонных выплат I , относящихся к будущему моменту времени $t_k = k$, причём i – это сложные проценты.

Временная диаграмма, иллюстрирующая расчет текущей стоимости облигации, приведена на рисунке 1.

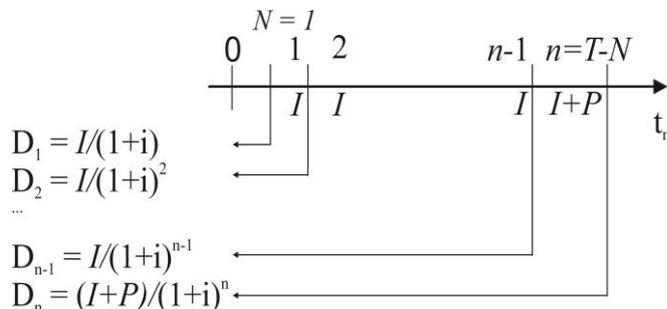


Рис. 3.1. Временная диаграмма для расчёта текущей стоимости облигации.

Текущая стоимость облигации выражается в виде:

$$An = D_1 + D_2 + \dots + D_{n-1} + D_n = \frac{I}{(1+i)} + \frac{I}{(1+i)^2} + \dots + \frac{I}{(1+i)^{n-1}} + \frac{I+P}{(1+i)^n}. \quad (3.2)$$

Представим An в виде двух сумм:

$$An = \left(\frac{I}{(1+i)} + \frac{I}{(1+i)^2} + \dots + \frac{I}{(1+i)^{n-1}} + \frac{I}{(1+i)^n} \right) + \frac{P}{(1+i)^n}.$$

Вынесем I за скобки:

$$An = I \cdot \left(\frac{1}{(1+i)} + \frac{1}{(1+i)^2} + \dots + \frac{1}{(1+i)^{n-1}} + \frac{1}{(1+i)^n} \right) + \frac{P}{(1+i)^n}.$$

По формуле суммы геометрической прогрессии

$$S_n = a_1 + a_2 + a_3 + \dots + a_n, \text{ сумма ряда равна } S_n = \frac{a_n \cdot q - a_1}{q - 1}, \text{ где } q = \frac{a_{n+1}}{a_n} -$$

знаменатель геометрической прогрессии, вычислим знаменатель ряда:

$$q = \frac{1}{(1+i)^2} \div \frac{1}{(1+i)} = \frac{1}{(1+i)}.$$

Найдём сумму ряда для An и преобразуем к виду:

$$S_n = \frac{\frac{1}{(1+i)^n} \cdot q - \frac{1}{(1+i)}}{q-1} = \frac{\frac{1}{(1+i)^{n+1}} - \frac{1}{(1+i)}}{\frac{1}{(1+i)} - 1} = \frac{(1-(1+i)^n) \cdot (1+i)^{-n}}{-i} =$$

$$= \frac{(1+i)^{-n} - 1}{-i} = \frac{1 - (1+i)^{-n}}{i}$$

Запишем итоговую формулу:

$$An = I \cdot \frac{1 - (1+i)^{-n}}{i} + P \cdot (1+i)^{-n}, \quad (3.3)$$

где $I \cdot \frac{1 - (1+i)^{-n}}{i}$ – это текущее значение последовательных процентных вы-

плат, $P \cdot (1+i)^{-n}$ – текущее значение стоимости облигации при погашении.

Приведём формулу текущей стоимости облигации к виду уравнения, пригодного для решения. Для начала найдём общий знаменатель:

$$An = \frac{I - I \cdot (1+i)^{-n} + i \cdot P \cdot (1+i)^{-n}}{i},$$

левую и правую часть умножим на i и всё перенесём в правую часть:

$$I - I \cdot (1+i)^{-n} + i \cdot P \cdot (1+i)^{-n} - i \cdot An = 0.$$

Добавим и вычтем $P \cdot (1+i)^{-n}$ и An :

$$I - I \cdot (1+i)^{-n} + i \cdot P \cdot (1+i)^{-n} + P \cdot (1+i)^{-n} - P \cdot (1+i)^{-n} - i \cdot An + An - An = 0$$

Введем замену переменной

$$x = 1+i, \quad (3.4)$$

получим:

$$I - I \cdot (1+i)^{-n} + P \cdot (1+i)^{-n} \cdot (1+i) - P \cdot (1+i)^{-n} - An \cdot (1+i) + An = 0.$$

Преобразуем уравнение для приемлемого вида:

$$-I \cdot (1+i)^{-n} + P \cdot (1+i)^{-n} \cdot (1+i) - P \cdot (1+i)^{-n} - An \cdot (1+i) + (An + I) = 0.$$

Умножим левую и правую часть уравнения на (-1)

$$(I + P) \cdot (1+i)^{-n} - P \cdot (1+i)^{-n+1} + An \cdot (1+i) - (An + I) = 0$$

и введём замену переменной $x = 1 + i$.

Таким образом, математическая постановка задачи оценки доходности облигаций формулируется в виде нелинейного алгебраического уравнения с неизвестной x :

$$(I + P) \cdot x^{-n} - P \cdot x^{-n+1} + An \cdot x - (An + I) = 0. \quad (3.5)$$

Как известно, первый этап решения уравнений вида

$$f(x) = 0 \quad (3.6)$$

представляет собой процесс локализации корней уравнения.

Уравнение (3.5) в общем случае не решается известными аналитическими методами и требует привлечения численных процедур. Для решения нелинейного уравнения в ППП Matlab можно воспользоваться встроенными функциями.

Введем в рассмотрение дополнительные характеристики доходности облигации. Цена облигации в конце периода N_j ($N_j \leq n$), ден.ед.:

$$Ak = I \cdot \frac{1 - (1+i)^{-n+N_j}}{i} + P \cdot (1+i)^{-n+N_j}, j = 0, 1, 2, \dots$$

Текущая доходность облигации, %:

$$dt = \frac{I}{Ak} \cdot 100\% .$$

Доходность облигации за счёт изменения цены в конце периода, %:

$$dch = \frac{An - Ak}{An} \cdot 100\% .$$

Норма прибыли облигации за период, %:

$$Np = \frac{I + (An - Ak)}{An} \cdot 100\% .$$

Цена облигации в конце срока погашения и с установленной процентной ставкой, руб.:

$$Ak_{k,l} = I \cdot \frac{1 - (1+i_l)^{-T_k}}{i_l} + P \cdot (1+i_l)^{-T_k}, k = 0, 1, 2, \dots; l = 0, 1, 2, \dots \quad (3.7)$$

3.2 Математическая модель оценки доходности банковских операций: концептуальная и математическая постановка задачи

Определим экономический объект, подлежащий моделированию – доходность банковской операции. Пусть банк предоставляет фирме кредит в течение n лет платежами по S_k денежных единиц за период k ($k=0, \dots, n-1$). Фирма возвращает долг платежами C_p в конце периода p ($p=n, \dots, N$) с остатком долга D на конец периода N . Оценить доходность такой операции для банка.

Для оценки операции с позиции банка будем рассматривать все поступления в банк положительными величинами, а все банковские выплаты – отрицательными. Банковская операция представляет собой текущее значение всех платежей, которые представляют собой ренту периодических разнонаправленных выплат с остатком на конец периода. Временная схема платежей приведена на рисунке 3.2.

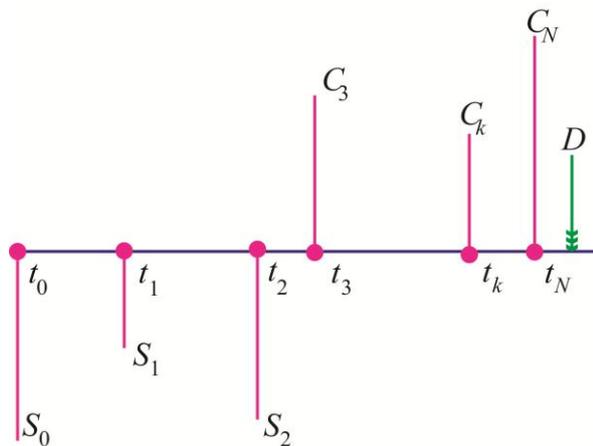


Рис. 3.2. Временная диаграмма для модели банковских операций.

Для получения текущего значения всех разнонаправленных выплат, суммы банковского кредита S_k , погашения долга фирмой C_p и остатка платежей в конце периода D дисконтируем в момент времени t_0 и суммируем, учитывая направление платежей. Текущее значение платежей зависит от выплат, которые делает банк, и от платежей, приносящих средства банку. Отсчет временных периодов начинается с момента времени t_0 . Сумма этих платежей при эффективной процентной ставке i должна равняться нулю (эффективной ставкой называется годовая ставка сложных процентов, дающая то же соотношение между

выданной суммой и суммой, которая получена при любой схеме выплат, она измеряет тот относительный доход, который может быть получен в целом за год). Таким образом, на величину текущей стоимости такой банковской операции влияют величины разовых выплат S_k и C_k , момент выплаты (начало и конец периода), наличие и величина остатка платежей в конце периода D .

Выплаты и платежи представляют собой ренту с различной величиной платежей, поэтому справедливо равенство:

$$-\sum_{k=0}^{n-1} \frac{S_k}{(1+i)^k} + \sum_{p=n}^N \frac{C_p}{(1+i)^p} + \frac{D}{(1+i)^N} = 0, \quad (3.8)$$

где первое выражение – сумма банковских выплат фирме, второе – сумма платежей банку, третье – остаток платежей в конце периода. Все платежи рассчитаны с банковской ставкой i , которая станет эффективной в случае равенства разнонаправленных потоков платежей.

Таким образом, математическая постановка задачи определения эффективной банковской процентной ставки формулируется в виде нелинейного уравнения вида (3.8) с неизвестной i . Введя замену $x=1+i$, перейдем к уравнению вида:

$$-\sum_{k=0}^{n-1} S_k \cdot x^{-k} + \sum_{p=n}^N C_p \cdot x^{-p} + D \cdot x^{-N} = 0. \quad (3.9)$$

3.3 Примеры решения задач в Matlab

Функция **fzero** позволяет приближенно вычислить корень уравнения на некотором интервале или ближайший к заданному начальному приближению. В простейшем варианте **fzero** вызывается с двумя входными и одним выходным аргументом $\mathbf{x}=\mathbf{fzero}(\mathbf{func_name}, \mathbf{x0})$, где **func_name** – имя файл-функции, вычисляющей левую часть уравнения, **x0** – начальное приближение к корню, **x** – найденное приближенное значение корня.

Пример 1. Решить на отрезке $[-5,5]$ уравнение $\sin x - x^2 \cos x = 0$.

Выполним локализацию корня, для этого создадим файл-функцию:

```
function y=fun(x)
```

```
y=sin(x)-(x.^2).*cos(x);
```

Оформим вызов в окне команд для вывода графика функции в графическое окно:

```
>> fplot('fun', [-5 5])
```

```
>> grid on
```

Результат показан на рисунке 3.3.

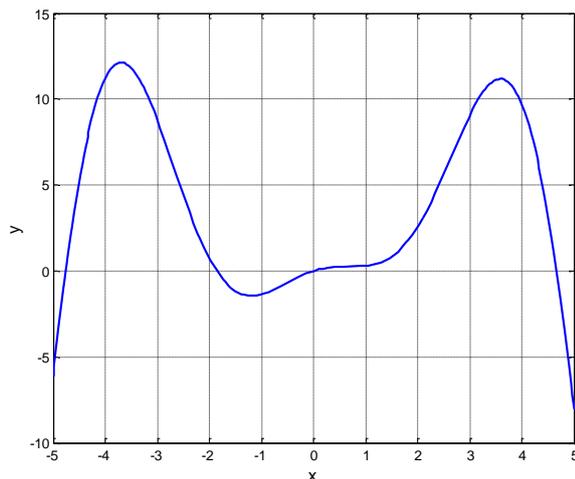


Рис. 3.3. Графическая локализация нулей функции $f(x)$ для примера 1.

Найдем значение корня вблизи $x = -5$.

Для того чтобы увидеть больше значащих цифр, установим формат **long**. Алгоритм функции **fzero** находит корень уравнения с точностью **eps**.

Важной особенностью функции **fzero** является то, что она вычисляет только те корни, в которых функция меняет свой знак:

```
>> x1=fzero('fun', -5)
```

```
x1 =
```

```
-4.7566
```

```
>> format long
```

```
>> x1
```

```
x1 =
```

```
-4.756559405702904
```

```
>> eps
```

```
ans =  
2.220446049250313e-016
```

Optimization Toolbox позволяет решать нелинейные уравнения с помощью функции **fsolve**. В общем случае вызов этой функции со всеми принятыми по умолчанию настройками имеет вид **x=fsolve('func_name',x0, optimset('fsolve'))**.

Пример 2. Решить уравнение из примера 1 с использованием функции **fsolve**.

Оформим вызов соответствующей функции:

```
x1=fsolve('fun', -5,optimset('fsolve'))
```

```
Optimization terminated: first-order optimality is less than  
options.TolFun.
```

```
x1 =  
-4.756559405702904
```

Пример 3. Локализуем корни уравнения (3.5). Инициализируем управляющие параметры модели. Пусть купонная процентная ставка – $k=0.15$; номинальная стоимость – $P=1000$; срок, оставшийся до погашения облигации (в годах) – $n=14$; текущая стоимость облигации – $An=1368.33$; величина купонных платежей (произведение номинальной стоимости на купонную процентную ставку) – $I = P \cdot k = 150$.

Для решения уравнения (3.5) построим график функции $f(x)$ и локализуем нули функции $f(x) = (I + P) \cdot x^{-n} - P \cdot x^{-n+1} + An \cdot x - (An + I)$. Для построения графика функции воспользуемся символьным построением графика с помощью функции **ezplot**:

```
hold on;  
syms x;  
k=0.15;  
P=1000;  
n=14;  
An=1368.33;  
I=k*P;
```

```

f=(I+P)*x.^(-n)-P*x.^-(n-1)+An*x-(An+I);
ezplot(f,[-10 10]),grid
legend('f=(I+P)*x^{-n}-P*x^{-(n-1)}+An*x-(An+I)')
title('Локализация корней')
C=[1368.33 -1518.33 0 0 0 0 0 0 0 0 0 0 0 0 -1000 1150];
X=roots(C);
%Локализация корней
plot(X(1),0,'ro',X(14),0,'ro',X(15),0,'ro')

```

Результат построения графика функции и локализации трёх корней уравнения приведён на рисунке 3.4.

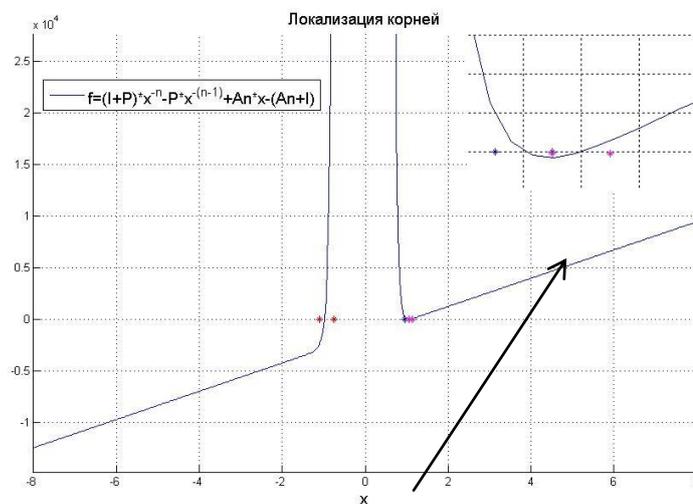


Рис. 3.4. Локализация корней нелинейного уравнения (3.5).

Локализуем первый корень в интервале $-1.1 < x_1 < -0.75$, второй корень в интервале $0.95 < x_2 < 1.05$, третий корень в интервале $1.05 < x_3 < 1.15$. Экономический смысл имеет только корень x_3 , так как только в этом случае процентная ставка оказывается положительной.

Пример 4. Локализуем корни уравнения (3.9). Создадим файл-функцию, соответствующую левой части уравнения **bank.m**.

```

function z=bank(x)
S=[1 1 1 1];
C=[1 1 1 1 2];
N=9;
D=0;
n=length(S);

```

```

Sum1=0;
for i=1:n
    z1=Sum1-S(i)*x.^(-i+1);
    Sum1=z1;
end
Sum2=0;
for p=1:(N-n)
    z2=Sum2+C(p)*x.^(-p-n+1);
    Sum2=z2;
end
z=z1+z2+D*x.^(-N);

```

Параметры моделирования примем равными: продолжительность выплат фирме $n=4$ года (начиная с нулевого момента времени), продолжительность банковской операции 8 лет (начиная с нулевого момента времени), следовательно $N=9$, значения банковских выплат, сделанных в течение 4 лет в начале каждого года $S=[1 \ 1 \ 1 \ 1]$ ден. ед., значения платежей в конце 4, 5, 6, 7, 8 годов составят $C=[1 \ 1 \ 1 \ 1 \ 2]$ ден. ед., величина остатка платежей $D=0$ ден. ед. Оформим ее вызов для построения графика функции:

```

x=0.8:0.001:2;
y=bank(x);
plot(x,y,'LineWidth',1.5); grid on

```

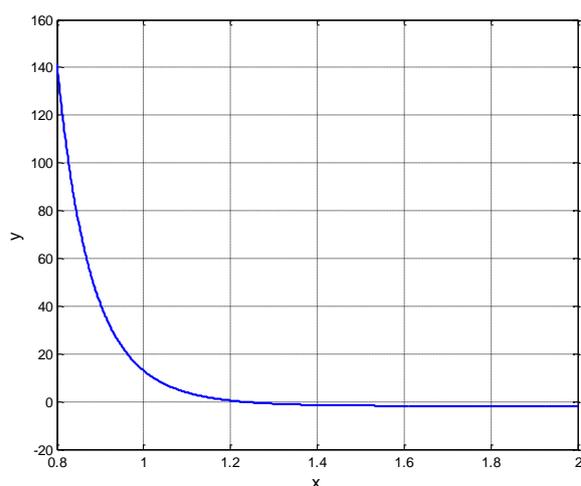


Рис. 3.4. Локализация корней нелинейного уравнения (3.9).

Для определения решения можно воспользоваться одним из указанных вариантов использования встроенного инструментария пакета:

```
xsol1=fsolve('bank',1.1, optimset('fsolve'))
xsol2=fzero('bank',1.1)
```

3.4 Индивидуальные задания

1. Реализовать модель оценки доходности облигаций, используя встроенные функции пакета (провести локализацию корня, найти значение рыночной процентной ставки) для набора параметров, указанных в условии.

2. Провести моделирование дополнительных характеристик для исследования изменения доходности облигации с изменением периода времени с момента ее выпуска, выбирая следующие значения: $N_0 = 3, N_1 = 5, N_2 = 7, N_3 = 9, N_4 = 11, N_5 = 13$. Проведите анализ, ответив на вопросы: как изменяются норма прибыли, текущая доходность, доходность облигации за счёт изменения цены с ростом периода? Покажите, что облигации обесцениваются с ростом периода (чем выше доход, приносимый облигацией, тем ниже её рыночная стоимость An).

3. Изменяя срок погашения облигации T ([1 5 7 10 15] лет) и рыночную процентную ставку i ([5% 10% 15% 20% 25%]) исследуйте зависимости цены облигации в конце периода A_k от срока погашения при различных значениях процентной ставки и зависимость цены облигации в конце периода A_k от значений процентной ставки при различных значениях срока погашения. Сделайте вывод относительно цены облигации при меньшем, большем и равном значении рыночной процентной ставки по сравнению с купонными процентами. Сделайте вывод относительно соотношения риска процентных ставок и прибыльности облигаций.

Вариант	Параметры моделирования			
	k	P	n	An
1	0.15	1000	12	1500
2	0.2	1000	12	1500

3	0.1	1000	12	1500
4	0.15	2000	10	2600
5	0.2	2000	10	2600
6	0.1	2000	10	2300
7	0.15	3000	11	3800
8	0.2	3000	11	3500
9	0.1	3000	11	3500
10	0.15	1000	14	1600

4. Провести исследование банковской операции согласно модели (9), определив эффективную процентную ставку для задания согласно варианту. Решение нелинейного уравнения провести встроенными функциями пакета.

Вариант	Параметры моделирования			
	S	C	n	N
1	[1 1 1 1]	[1 1 1 1 2]	4	9
2	[1 1 1 1]	[1 1 1 1 2 2]	4	10
3	[1 1 1 1]	[1 1 1 1 2 2 3]	4	11
4	[1 1 1 1]	[1 1 1 1 2 2 3 3]	4	12
5	[1 1 1]	[1 1 1 1 2 2]	3	9
6	[1 1 1]	[1 1 1 1 2 2 3]	3	10
7	[1 1 1]	[1 1 1 1 2 2 3 3]	3	11
8	[1 1 1 1 1]	[1 1 1 3]	5	9
9	[1 1 1 1 1]	[1 1 1 1 5]	5	10
10	[1 1 1 1 1 1]	[1 1 2]	6	9

$D=3$ ден. ед. для всех вариантов заданий. Проведите вычислительный эксперимент, сделайте вывод.

4 ЭКОНОМИЧЕСКИЕ ПРИЛОЖЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Для реализации моделей, формулируемых в постановке линейных и нелинейных задач оптимизации, в ППП Matlab используют специальные функции, которые входят в состав Optimization Toolbox. Интерфейс этих функций достаточно гибкий, все они допускают обращение с переменным числом входных и выходных аргументов в зависимости от данных задачи и искомых величин.

Классическими задачами математического программирования являются задачи линейного программирования: задача об использовании сырья, задача о назначениях, транспортная задача, задача о рационе и др.

4.1 Решение задачи линейного программирования в ППП Matlab

Задача линейного программирования состоит в нахождении вектора x , который минимизирует целую линейную функцию:

$$f^T x,$$

где f – вектор коэффициентов, причем x удовлетворяет заданным линейным ограничениям:

неравенствам:

$$Ax \leq b$$

и равенствам:

$$A_{eq}x = b_{eq}.$$

Также могут быть сформулированы двусторонние покомпонентные ограничения в векторной форме:

$$lb \leq x \leq ub.$$

В задачах оптимизации могут быть заданы не все типы ограничений, например, ограничения – равенства могут отсутствовать.

Для решения задач сформулированного вида предназначена специализированная функция **linprog** (**f**, **A**, **b**, **Aeq**, **beq**, **lb**, **ub**). Первым аргументом является вектор f , далее задаются матрица A и вектор b , A_{eq} , b_{eq} , lb , ub .

В задачах квадратичного программирования целевая функция имеет вид:

$$\frac{1}{2} x^T H x + f^T x,$$

а ограничения имеют вид аналогичный предыдущей задаче:

$$Ax \leq b, \quad Aeqx = beq, \quad lb \leq x \leq ub.$$

Для решения задач квадратичного программирования служит функция **quadprog**, интерфейс которой не отличается, в основном, от интерфейса функции **linprog**, за исключением того, что первыми двумя входными аргументами являются массив H и вектор f , соответствующие целевой функции.

Одним из важнейших приложений задач квадратичного программирования в экономике является задача Марковица об определении состава инвестиционного портфеля рискованных ценных бумаг.

4.2 Математическая модель определения доходности портфеля ценных бумаг

Модель оценки доходности ценных бумаг основана на том, что показатели доходности различных ценных бумаг взаимосвязаны: с ростом доходности одних бумаг наблюдается одновременный рост по другим бумагам, доходность остается без изменения – по третьим, а по четвертым доходность, наоборот, снижается. Такая зависимость не является детерминированной, т.е. однозначно определенной, а является стохастической, мерой которой может служить корреляция. Модель Марковица имеет следующие основные допущения:

- в качестве доходности ценной бумаги принимается математическое ожидание доходности;
- в качестве риска ценной бумаги принимается среднее квадратическое отклонение доходности;

- принимается, что данные прошлых периодов, используемые при расчете доходности и риска, в полной мере отражают будущие значения доходности;
- степень и характер взаимосвязи между ценными бумагами выражается коэффициентом линейной корреляции.

По модели Марковица доходность портфеля ценных бумаг – это средневзвешенная доходность бумаг, его составляющих, и она определяется формулой:

$$R_p = \sum_{i=1}^N W_i \cdot \hat{r}_i,$$

где N – количество ценных бумаг в портфеле, W_i – процентная доля данной бумаги в портфеле, r_i – доходность данной бумаги.

Средняя доходность отдельной ценной бумаги r_i рассчитывается по правилу:

$$r_i = \frac{1}{T} \cdot \sum_{t=1}^T r_{it}, \quad (4.1)$$

где T – количество прошлых наблюдений доходности данной ценной бумаги, r_{it} – значение доходности i -ой бумаги в период t .

Риск портфеля ценных бумаг определяется среднеквадратическим отклонением доходности портфеля:

$$\sigma_p = \sqrt{\sum_{a=1}^N \sum_{b=1}^N (W_a \cdot \hat{\sigma}_a \cdot W_b \cdot \hat{\sigma}_b \cdot \rho_{ab})}, \quad (4.2)$$

где W_a, W_b – процентные доли данных бумаг в портфеле, $\hat{\sigma}_a, \hat{\sigma}_b$ – риск данных бумаг (среднеквадратическое отклонение), ρ_{ab} – коэффициент линейной корреляции.

Риск отдельной ценной бумаги (в виде оценки среднеквадратического отклонения) можно вычислить по соотношению:

$$\hat{\sigma}_i = \sqrt{\frac{1}{T-1} \cdot \sum_{t=1}^T (r_{it} - \hat{r}_i)^2}. \quad (4.3)$$

Статистическая оценка коэффициента корреляции между показателями доходности двух ценных бумагами:

$$\hat{\rho}_{ab} = \frac{1}{(T-1) \cdot \hat{\sigma}_a \cdot \hat{\sigma}_b} \cdot \sum_{t=1}^T [(r_{at} - \hat{r}_a) \cdot (r_{bt} - \hat{r}_b)], \quad (4.4)$$

где r_{at} , r_{bt} – доходность ценных бумаг a и b в период t .

Матрица ковариации вычисляется следующим образом:

$$V_{ab} = \hat{\sigma}_a \cdot \hat{\sigma}_b \cdot \hat{\rho}_{ab}. \quad (4.5)$$

Итак, с использованием модели Марковица задача расчета характеристик портфеля приобретает вид задачи квадратичного программирования с целевой функцией:

$$\sum_{i=1}^N W_i \cdot \hat{r}_i \rightarrow \max, \quad (4.6)$$

определяющей доходность портфеля ценных бумаг при ограничениях

$$\begin{cases} \sqrt{\sum_{a=1}^N \sum_{b=1}^N (W_a \cdot \sigma_a \cdot W_b \cdot \sigma_b \cdot \rho_{ab})} \leq \sigma_{req}; \\ W_i \geq 0; \\ \sum W_i = 1. \end{cases} \quad (4.7)$$

Задача Марковица в постановке (4.6)-(4.7) называется прямой. Однако на практике получила распространение обратная задача Марковица: найти состав портфеля W , при котором риск портфеля σ_p^2 принимал бы наименьшее значение при заданной доходности R_p . Целевая функция задается как

$$\sigma_p^2 = W^T \cdot V \cdot W \rightarrow \min, \quad (4.8)$$

при ограничениях

$$\begin{cases} \sum_{i=1}^N W_i \cdot r_i^T = R_p, \\ \sum_{i=1}^N W_i \geq 0, \\ \sum e^T W_i = 1, \end{cases} \quad (4.9)$$

где R_p – желаемая доходность портфеля, V – матрица ковариации, r – вектор доходности ценных бумаг.

Очевидно, что задача (4.8)-(4.9) может быть решена как задача квадратичного программирования, вектор неизвестной которой $x=W$ требуется определить при заданном наборе параметров R_p (подлежит инициализации из эмпирических данных как свободный параметр), V (согласно соотношению (4.5)), r (по формуле (4.1)), вектор f^T – нуль-вектор.

4.3 Примеры решения задач в Matlab

Пример 1. Решить задачу линейного программирования о составлении рациона питания. Имеются три типа продукта П1, П2, П3 разной цены, каждый из которых содержит определенное количество питательных ингредиентов И1, И2, И3, И4 (см. данные в таблице). Известно, что в день требуется: И1 – не менее 250 ед., И2 – не менее 60 ед., И3 – не менее 100 ед., И4 – не менее 220 ед. Требуется минимизировать затраты на приобретение продуктов.

Ингредиенты	Продукт П1	Продукт П2	Продукт П3
И1	4	6	15
И2	2	2	0
И3	5	3	4
И4	7	3	12
Цена	44	35	100

Обозначим количества планируемых к приобретению продуктов типов П1, П2, П3: x_1, x_2, x_3 единиц соответственно. Тогда минимизируемая целевая функция выразится как суммарные затраты на приобретение продуктов:

$$f^T x = 44x_1 + 35x_2 + 100x_3 \rightarrow \min .$$

Запишем систему ограничений на количество ингредиентов. Очевидно также, что количество приобретаемых продуктов не может быть отрицательным.

$$\begin{cases} 4x_1 + 6x_2 + 15x_3 \geq 250 \\ 2x_1 + 2x_2 \geq 60 \\ 5x_1 + 3x_2 + 4x_3 \geq 100 \\ 7x_1 + 3x_2 + 12x_3 \geq 220 \end{cases}, \quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$$

В обозначениях входных переменных функции Matlab:

$$A = \begin{bmatrix} -4 & -6 & -15 \\ -2 & -2 & 0 \\ -5 & -3 & -4 \\ -7 & -3 & -12 \end{bmatrix}, \quad b = \begin{bmatrix} -250 \\ -60 \\ -100 \\ -220 \end{bmatrix}, \quad lb = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

При вызове функции **linprog** вместо неиспользуемых аргументов задаются пустые массивы.

Создадим скрипт-файл для решения задачи:

```
A=[-4 -6 -15; -2 -2 0; -5 -3 -4; -7 -3 -12];
b=[-250; -60; -100; -220];
f=[44; 35; 100];
lb=[0; 0; 0];
[x, p]=linprog(f,A,b,[],[],lb,[])
```

Результаты вызова на экране:

```
Optimization terminated.
```

```
x =
```

```
13.2143
```

```
16.7857
```

```
6.4286
```

```
p = 1.8118e+003
```

Таким образом, минимальная стоимость рациона составит 1811.8 ед.

Пример 2. Проиллюстрируем постановку задачи квадратичного программирования на примере задачи Марковица об определении состава инвестиционного портфеля рискованных ценных бумаг.

Инвестор предполагает вложить свободные денежные средства в рыночные активы (ценные бумаги, акции) с целью получения дохода в будущем периоде. Для уменьшения рисков он выбрал для вложения четыре различных акции, которые обозначим A1, A2, A3, A4. Перед инвестором стоит задача опре-

делить, какую часть своих средств вложить в каждый актив так, чтобы получить желаемую доходность портфеля с наименьшим риском. Портфель определяется вектором долей от суммы инвестиций для покупки акций:

$$W = x = (x_1, x_2, x_3, x_4)^T, x_1 + x_2 + x_3 + x_4 = 1, x_i \geq 0 \quad (i = 1, 2, 3, 4)$$

Будем считать, что инвестор описанным выше образом провел оценку ожидаемой доходности для каждой ценной бумаги, т.е. построил вектор доходностей r и вычислил матрицу ковариации V . Величина желаемой доходности портфеля R_p должна быть не меньше минимальной и не больше максимальной доходности выбранных для инвестирования ценных бумаг. Первые два ограничения в (4) объединим в одно, построив матрицу и векторы, связанные с ограничениями:

$$Aeq = \begin{pmatrix} r^T \\ e^T \end{pmatrix}, \quad beq = \begin{pmatrix} R_p \\ 1 \end{pmatrix}, \quad lb = [0 \quad 0 \quad 0 \quad 0]^T,$$

где $e^T = (1, 1, 1, 1)$.

Создадим скрипт-файл для решения задачи:

```
V=[100 27 -52 66; 27 150 45 -66; -52 45 250 60; 66 -66 60 280];
Aeq=[12 15 16 18; 1 1 1 1];
beq=[15; 1];
lb1=[0; 0; 0; 0];
x1=quadprog(V, [], [], [], Aeq, beq, lb1)
```

Результаты вызова на экране:

```
x1 =
    0.2694
    0.3479
    0.1700
    0.2127
```

Таким образом, ответ о распределении денежных средств будет следующим: следует приобрести 26.9 % акций A1, 34.8 % акций A2, 17% акций A3 и 21.3 акций A4.

5.4 Индивидуальные задания

1. Построить математическую модель задачи линейного программирования согласно предложенному варианту (см. далее). Решить задачу средствами ППП Matlab. Сделать вывод.

2. Принимая в качестве исходных данных еженедельные котировки акций четырех предприятий в течение определённого периода, решить задачу Марковица об определении состава инвестиционного портфеля рискованных ценных бумаг. Значение доходности портфеля установите равным n % (где n – номер варианта). Решить задачу средствами ППП Matlab. Сделать вывод. Для оценки параметров моделирования в таблице приведены данные о доходности акций в рассматриваемый период:

	НОМЕР ПЕРИОДА							
Ценные бумаги:	1	2	3	4	5	6	7	8
Акции 1	-2.1%	0.50%	3.57%	-8.28%	28.91%	-7.22%	-9.6%	10.05%
Акции 2	-16.3%	0.01%	60.31%	9.2%	28.99%	3.22%	5.37%	-9.5%
Акции 3	12.00%	3.00%	16.67%	15.21%	-6.45%	3.33%	-27.17%	-6.53%
Акции 4	1.32%	0.50%	1.28%	0.10%	0.80%	0.17%	15.78%	0.39%
	НОМЕР ПЕРИОДА							
Ценные бумаги:	9	10	11	12	13	14	15	
Акции 1	1.00%	-3.58%	-9.70%	40.10%	9.5%	-7.65%	-0.77%	
Акции 2	0.59%	4.53%	4.40%	-8.47%	-10.03%	3.00%	9.30%	
Акции 3	21.31%	9.11%	0.51%	0.64%	10.99%	1.28%	-37.02%	
Акции 4	0.37%	4.23%	0.90%	-0.35%	0.22%	-2.78%	-9.2%	

Найдите зависимость риска от доходности портфеля при вариации значения доходности в интервале от 1 до 10 %.

3. Оформить отчет по лабораторной работе.

ВАРИАНТЫ К ЗАДАНИЮ 1

Вариант 1

1. Для производства двух видов изделий A и B используется три типа технологического оборудования.

Для производства единицы изделия A оборудование первого типа используется в течение 2 ч., оборудование второго типа – 1 ч., оборудование третьего типа – 3 ч. Для производства единицы изделия B оборудование первого типа используется в течение 4 ч., оборудование второго типа – 1 ч., оборудование третьего типа – 1 ч.

На изготовление всех изделий предприятие может использовать оборудование первого типа не более 64 ч., оборудование второго типа – не более 20 ч., оборудование третьего типа – не более 50 ч.

Прибыль от реализации единицы готового изделия A – составляет 5 денежных единицы, а изделие B – 3 денежных единицы.

Составить план производства изделий A и B , обеспечивающий максимальную прибыль от их реализации.

Сводная таблица условий задачи имеет вид:

Тип оборудования	Вид изделия		Ресурс изделия
	A	B	
I	2	4	64
II	1	1	20
III	3	1	50
Прибыль	5	3	

Вариант 2

Нефтеперерабатывающий завод получает четыре полуфабриката:

- 400 тыс. л. алкилата;
- 250 тыс. л. крекинг-бензина;
- 350 тыс. л. бензина прямой перегонки;
- 250 тыс. л. изопентона;

В результате смешивания этих четырёх компонентов в разных пропорциях образуются три сорта авиационного бензина:

- Бензин А – 2 : 3 : 5 : 2 ;
- Бензин В – 3 : 1 : 2 : 1 ;
- Бензин С – 2 : 2 : 1 : 3 ;

Стоимость 1 тыс. л. указанных сортов бензина:

- Бензин А – 245 руб.
- Бензин В – 200 руб.
- Бензин С – 310 руб.

Необходимо определить план смешения компонентов, при котором будет достигнута максимальная стоимость всей продукции. При следующих условиях: бензина каждого сорта должно быть произведено не менее 300 тыс. л., неиспользованного крекинг-бензина должно остаться не более 50 тыс. л.

Сводная таблица условий задачи имеет вид:

Компоненты, используемые для производства трёх видов бензина	Сорта производимого бензина			Объем ресурсов (тыс. л.)
	<i>A</i>	<i>B</i>	<i>C</i>	
Алкилат	$\frac{2}{12}$	$\frac{3}{7}$	$\frac{2}{8}$	400
Крекинг-бензин	$\frac{3}{12}$	$\frac{1}{7}$	$\frac{2}{8}$	250
Бензин прямой перегонки	$\frac{5}{12}$	$\frac{2}{7}$	$\frac{1}{8}$	300
Изопентат	$\frac{2}{12}$	$\frac{1}{7}$	$\frac{3}{8}$	250
Цена бензина (рублей за 1 тыс.л.)	245	200	310	

Вариант 3

Один из цехов машиностроительного предприятия выпускает изделия двух видов: корпуса и задвижки. Для производства этих изделий требуются три вида сырья: алюминий, сталь и пластмасса.

На выпуск одного корпуса расходуется 20 кг алюминия, 20 кг стали и 5 кг пластмассы. На выпуск одной задвижки расходуется 5 кг алюминия, 10 кг стали и 20 кг пластмассы. Запасы ресурсов ограничены: за рабочую смену цех может израсходовать не более 200 кг алюминия, 500 кг стали и 500 кг пластмассы.

Выпуск одного корпуса приносит предприятию прибыль в размере 250 ден. ед., одной задвижки – 350 ден. ед.

Требуется составить оптимальный план работы цеха, т.е. найти, сколько корпусов и задвижек требуется выпускать, чтобы получить максимальную прибыль (при соблюдении ограничений на ресурсы).

Вариант 4

Косметическая фирма производит два вида зубной пасты. При этом используются два вида продукта: A – масло кедровое и B – масло мяты, запасы которых ограничены.

Максимально возможный суточный запас продукта A составляет 6 кг, продукта B – 8 кг. При производстве 1 кг пасты вида 1 используется 0.1 кг продукта A и 0.2 кг продукта B , а при производстве 1 кг пасты вида 2 – 0.2 кг продукта A и 0.2 кг продукта B . Доход от реализации 1 кг пасты вида 1 составляет 300 руб., а пасты вида 2 – 500 руб., причем суточный спрос на пасту вида 2 не превышает 2 кг.

При каких значениях суточного выпуска пасты видов 1 и 2 доход от продажи пасты будет максимальным?

Вариант 5

Рацион для питания животных на ферме состоит из двух видов кормов. Один килограмм корма вида I стоит 80 ден. ед и содержит: 1 ед. жиров, 3 ед. белков, 1 ед. углеводов, 2 ед. нитратов. Один килограмм корма вида II стоит 10 ден. ед. и содержит: 3 ед. жиров, 1 ед. белков, 8 ед. углеводов, 4 ед. нитратов. Составить наиболее дешевый рацион питания, обеспечивающий содержание

жиров – не менее 6 ед., белков – не менее 9 ед., углеводов – не менее 8 ед, нитратов не более 16 ед.

Вариант 6

Предприятие химической промышленности выпускает соляную и серную кислоту. Выпуск одной тонны соляной кислоты – 43 ден.ед., выпуск одной тонны серной кислоты – 54 ден. ед.

Для выполнения государственного заказа необходимо выпустить не менее 300 т соляной и не менее 130 т серной кислоты. Кроме того, необходимо учитывать, что выпуск кислот связан с образованием опасных отходов. При выпуске одной тонны соляной кислоты образуется 0.5 т опасных отходов, при выпуске одной тонны серной кислоты – 1.2 т опасных отходов. Общее количество опасных отходов не должно превышать 600 т, так как превышение этого ограничения приведет к выплате предприятием крупного штрафа.

Требуется определить, сколько соляной и серной кислоты должно выпустить предприятие, чтобы получить максимальную прибыль.

Вариант 7

Каждый из трех видов удобрений содержит азот и фосфат в заданных соотношениях. Для теплицы требуется как минимум 25 единиц азота и 20 единицы фосфата. Стоимости удобрения I, II и III вида соответственно равны 100 руб., 80 руб. и 60 руб. за 1 кг. С прошлого сезона остался 1 кг удобрений III вида. Какое количество удобрений нужно приобрести для теплицы, что минимизировать затраты на их приобретение?

Удобрения	Азот	Фосфат
I	10	4
II	8	3
III	2	3

Вариант 8

Для выращивания крупного рогатого скота на ферме используют два вида концентрированных кормов, содержащих питательные вещества – углеводы и белки в заданных пропорциях. В дневном рационе животных должно содержаться не менее 100 ед. углеводов и 30 ед. белков. Какое количество концентрированных кормов надо приобрести, чтобы затраты на данную статью расходов фермы были минимальными?

Питательные вещества	Количество питательных веществ ед. /кг корма	
	концентрированный корм I вида	концентрированный корм II вида
Углеводы	5	8
Белки	1	2
Цена за 1кг, ден.ед.	20	30

Вариант 9

Решите задачу о составлении рациона питания. Имеются три типа продукта I, II, III. Стоимость за единицу продукта указана в таблице. Каждый продукт содержит определенное количество питательных веществ A , B , C , M (см. таблицу). Известно, что в день требуется: A – не менее 200 ед., B – не менее 75 ед., C – не менее 80 ед., M – не менее 180 ед. Требуется минимизировать затраты на приобретение продуктов.

	I	II	III
A	2	3	7
B	1	1	0
C	5	3	4
M	4	1	6
Стоимость, усл. ед.	25	30	50

Вариант 10

Для производства двух видов сока A и B концерн использует три вида сырья. Для производства единицы сока A и единицы изделия B затрачивается сырье в определенных объемах (см таблицу). Запасы сырья ограничены следующими значениями: сырье вида I – 50 ед, вида II – 40 ед, вида III – 50 ед. Прибыль от реализации единицы сока вида A – составляет 8 ден. ед., а сока вида B – 5 ден. ед. Составить план производства сока вида A и вида B , обеспечивающий максимальную прибыль от их реализации.

Сырье	Вид продукции		Ресурс сырья
	A	B	
I	2	4	50
II	1	3	40
III	5	2	30
Прибыль	8	5	

5 ДИНАМИЧЕСКИЕ (ЛИНЕЙНЫЕ И НЕЛИНЕЙНЫЕ) МОДЕЛИ

5.1 Задача о насыщении рынка автомобилей

Сформулируем *содержательную и концептуальную постановку задачи*. Объектом исследования будет являться рынок автомобилей в стране. Пусть к началу исследований в стране было N легковых автомобилей, в течение ближайших лет предполагается производить по P автомобилей в год. Средний срок службы автомобилей K лет. Требуется определить максимальное количество автомобилей в стране при таком производстве и рассмотреть динамику роста количества автомобилей.

Будем рассматривать процесс пополнения рынка (выпуск автомобилей) и процесс уменьшения количества автомобилей на рынке (ограничение срока службы автомобиля) в течение года. Для упрощения модели другие процессы, влияющие на динамику рынка, рассматривать не будем. Состояние рынка характеризуется количеством автомобилей, действующих в определённый момент времени t . Пусть количество автомобилей на рынке является функцией времени и равно $y(t)$. Предположим, что пополнение автомобильного рынка происходит только за счет выпуска P автомобилей в год. Численность автомобилей на рынке будет уменьшаться за счет выхода автомобилей из строя. Процесс выхода автомобиля из строя будет характеризоваться интенсивностью $j(t)$, равной $1/K$.

Математическая постановка задачи описывает динамику изменения численности автомобиля с течением времени:

$$y'(t) = P - j(t) \cdot y(t) \quad (5.1)$$

с начальным условием $y(t_0) = y_0$ – количество автомобилей в начальный момент времени.

5.2 Модели соперничества в экономике: модель финансовой пирамиды

Сформулируем *содержательную и концептуальную постановку задачи*. Объект исследования – финансовая пирамида, которая строится в городе, где

проживает N жителей. Происходит выпуск акций номиналом p_0 усл. ден. ед. и процентной ставкой ps единичного периода начисления по начальному вкладу. Курс покупки акций – k усл. ден. ед., а продажи p – усл. ден. ед.

Среди жителей города царит некий ажиотаж, который можно выразить через коэффициент ажиотажа a . Он определяется такими факторами, как уровень инфляции, затраты на рекламную кампанию, наличие на рынке ценных бумаг других компаний и др. В простейшем случае коэффициент ажиотажа $a = \text{const}$.

Акции продаются в среднем через T дней после покупки. Известно, что в первый день с начала эмиссии D горожан стали держателями акций.

В данной модели финансовая пирамида будет рассматриваться как процесс купли-продажи, где все жители города делятся на покупателей и продавцов. Основные величины, определяющие поведение модели: количество акционеров и инвесторов в зависимости от времени, доход организатора, объёмы купли-продажи ценных бумаг.

Определим количество потенциальных покупателей акций среди всех жителей города как y_k , а количество держателей акций (продавцов) – y_p . Между состояниями покупателей и продавцов постоянно происходит движение элементов системы (рисунок).

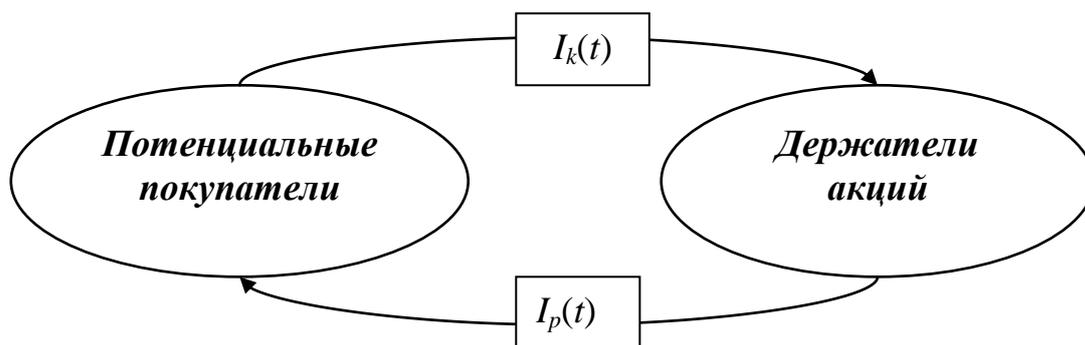


Рис. 5.1. Схема процесса купли-продажи ценных бумаг.

Каждый день некоторая часть жителей продает или покупает акции компании-устроителя пирамиды. Весь процесс характеризуется интенсивностями

купли $I_k(t)$ и продажи $I_p(t)$ акций компании. Интенсивность покупки акций $I_k(t)$ – функция времени, она пропорциональна числу держателей акций и зависит от коэффициента ажиотажа a и определяется как:

$$I_k(t) = a \cdot y_k(t). \quad (5.2)$$

Процесс продажи акций будет происходить, если с момента t покупки прошло T дней, т.е. $t > T$. Следовательно, интенсивность продажи $I_p(t)$ определяется так:

$$I_p(t) = \begin{cases} 0, & t \leq T \\ I_p(t), & t > T \end{cases}. \quad (5.3)$$

В каждый момент времени t происходят процессы купли и продажи акций горожанами. Обозначим количество купленных акций через $K(t)$, оно определяется следующим образом:

$$K(t) = I_k(t) \cdot y_k(t) = a \cdot y_k(t) \cdot y_p(t), \quad (5.4)$$

а количество проданных акций как $P(t)$. По условию задачи акция не продается в течение T дней после покупки, т.е. количество продаж равно нулю, если $t > T$. Таким образом,

$$P(t) = \begin{cases} 0, & t \leq T \\ I_p(t) \cdot y_p(t), & t > T. \end{cases} \quad (5.5)$$

Но количество акций проданных в момент времени t , равно количеству акций, купленных T дней назад, т.е. $P(t) = K(t-T)$. Следовательно,

$$P(t) = \begin{cases} 0, & t \leq T \\ K(t-T) = I_k(t-T) \cdot y_k(t-T), & t > T. \end{cases} \quad (5.6)$$

Отсюда следует, что

$$I_p(t) \cdot y_p(t) = K(t-T) = I_k(t-T) \cdot y_k(t-T). \quad (5.7)$$

Из (5.7) определим значение интенсивности продажи акций:

$$I_p(t) = \frac{K(t-T)}{y_p(t)} = \frac{I_k(t-T) \cdot y_k(t-T)}{y_p(t)}. \quad (5.8)$$

Подставим в (5.8) значение интенсивности покупки акций получим

$$I_p(t) = \frac{K(t-T)}{y_p(t)} = \frac{a \cdot y_p(t-T) \cdot y_k(t-T)}{y_p(t)}. \quad (5.9)$$

Подставив значение интенсивности (9) в (4), получим

$$P(t) = \begin{cases} 0, & t \leq T \\ I_k(t-T) \cdot y_k(t-T) = a \cdot I_k(t-T) \cdot y_k(t-T) = K(t-T), & t > T \end{cases} \quad (5.10)$$

Количество продавцов акций за промежуток времени Δt увеличивается на количество жителей, купивших акции $K(t)$ за этот период, так как каждый купивший акции автоматически становится их продавцом,

$$K(t) \cdot \Delta t = y_k(t) \cdot I_k(t) \cdot \Delta t = y_k(t) \cdot a \cdot y_p(t) \cdot \Delta t. \quad (5.11)$$

И уменьшится на число жителей города, продавших свои акции, т.е. объём продаж акций с учётом (5.9) за промежуток времени Δt составляет

$$P(t) \cdot \Delta t = \begin{cases} 0 \cdot \Delta t, & t \leq T \\ K(t-T) \cdot \Delta t = y_k(t-T) \cdot I_k(t-T) \cdot \Delta t, & t > T \end{cases} \quad (5.12)$$

Вычитая из (5.11) (5.12), получим уравнение баланса численности продавцов акций за промежуток времени Δt

$$\begin{aligned} \Delta y_p &= y_p(t + \Delta t) - y_p(t) = K(t) \cdot \Delta t - P(t) \cdot \Delta t = \\ &= y_k(t) \cdot I_k(t) \cdot \Delta t - y_k(t-T) \cdot a \cdot y_p(t-T) \cdot \Delta t \end{aligned} \quad (5.13)$$

Разделим обе части уравнения (5.13) на Δt

$$\frac{\Delta y_p}{\Delta t} = \frac{y_p(t + \Delta t) - y_p(t)}{\Delta t} = K(t) - P(t) = y_k(t) \cdot I_k(t) - y_k(t-T) \cdot a \cdot y_p(t-T).$$

Переходя к пределу при $\Delta t \rightarrow 0$, получим в левой части уравнения производную $y_p'(t)$

$$y_p'(t) = P(t) - K(t) = I_k(t) \cdot y_k(t) - a \cdot y_k(t-T) \cdot y_p(t-T). \quad (5.14)$$

Аналогично запишем уравнение изменения численности покупателей акций в промежутке времени Δt

$$y_k'(t) = K(t) - P(t) = a \cdot y_p(t-T) \cdot y_k(t-T) - I_k(t) \cdot y_k(t). \quad (5.15)$$

Таким образом, получили систему дифференциальных уравнений с запаздывающим аргументом:

$$\begin{cases} y'_p(t) = P(t) - K(t) = I_k(t) \cdot y_k(t) - a \cdot y_k(t-T) \cdot y_p(t-T) \\ y'_k(t) = K(t) - P(t) = a \cdot y_p(t-T) \cdot y_k(t-T) - I_k(t) \cdot y_k(t) \end{cases} \quad (5.16)$$

Общий вид последней системы:

$$\begin{cases} y'_p(t) = f(t, y_k, y_p) \\ y'_k(t) = g(t, y_k, y_p) \end{cases} \quad (5.17)$$

которую требуется решать относительно $y'_p(t)$ и $y'_k(t)$ – функций покупок и продаж акций в городе.

Для замыкания математической постановки задачи необходимо систему дифференциальных уравнений дополнить начальными условиями:

$$\begin{cases} y_p(t_0) = y_{p0} \\ y_k(t_0) = y_{k0} \end{cases}$$

Значение y_{p0} определим как число горожан, ставших обладателями акций в первый день исследуемого периода: $y_{p0} = D$. Для y_{k0} значение совпадает с количеством жителей города N за исключением тех, кто купил акции в первый день: $y_{k0} = N - D$.

Так же для рассматриваемой модели можно определить и другие показатели. Для их формализации определим ряд параметров:

1) Ежедневный прирост курса акций за счёт инфляции d_i в i – день. Пусть, каждый день прирост покупки за счёт инфляции увеличивается, например, на 0.5 %, тогда:

$$d_{i+1} = d_i \cdot (1 + 0.005).$$

2) Стоимость акций в $i+1$ – день при покупке (с учётом роста инфляции):

$$k_{i+1} = k_0 + d_i$$

3) Процентная ставка единого периода начисления по начальному вкладу (простые проценты), например, $ps=0.06$.

4) Стоимость акций в $i+1$ – день при продаже (с учётом роста инфляции):

$$p_{i+1} = k_i(1 + ps).$$

5) Количество денег в кассе в первый день развития пирамиды, например, $profit_0=70000$ рублей, ежедневный расход на строительство пирамиды составляет 500000 рублей. Процент ежедневного дохода учредителей пирамиды составляет, например, 3 % от ежедневного кассового сбора.

Расчет доходности пирамиды состоит в определении следующих показателей:

1) Количество денег в кассе компании-учредителя пирамиды в $i+1$ – день:

$$profit_{i+1} = \begin{cases} profit_i + k_i \cdot K_i - p_i \cdot P(t_i - T) - profit_i \cdot 0.03 - 500000, t_i > T \\ profit_i + k_i \cdot K_i - profit_i \cdot 0.03 - 500000, t_i \leq T \end{cases}.$$

2) Абсолютный доход организатора пирамиды в $i+1$ – день составит:

$$income_{i+1} = income_i + profit_i \cdot 0.03.$$

3) Предельный доход организатора пирамиды в $i+1$ – день составит:

$$predin_{i+1} = \begin{cases} income_i - income_{i+1}, & t_i \geq 2 \\ 0, & t_i < 2 \end{cases}.$$

4) Доход от одной акции для акционера в $i+1$ – день при условии продажи акции через T дней составит:

$$turn_{i+1} = \begin{cases} p_i - k(t_i - T), t_i > T \\ 0, & t_i \leq T \end{cases}.$$

5) Доход инвестора относительно затрат на покупку акции при условии продажи акции через T дней составит:

$$tempt_{i+1} = \begin{cases} tempt_0, t_i \leq T \\ \frac{p_i - k(t_i - T)}{k(t_i - T)}, t_i > T \end{cases}.$$

5.3 Приближенное решения задач Коши для обыкновенных дифференциальных уравнений в ППП Matlab

Рассмотрим обыкновенное дифференциальное уравнение первого порядка: $y' = f(x, y)$, $x \in [x_0, b]$ с начальным условием: $y(x_0) = y_0$, где $f(x, y)$ – некоторая заданная, в общем случае, нелинейная функция двух переменных.

Решение задачи Коши для дифференциальных уравнений методом Рунге-Кутты IV порядка может быть реализовано в пакете Matlab в виде функции **ode45**.

Пример 1. Найти решение задачи Коши дифференциального уравнения:

$$\frac{dy}{dx} = x^2, \quad y(0) = 1.3. \quad (5.18)$$

Найдем решение задачи, используя встроенные функции пакета Matlab. Создадим функцию `ff.m`, содержащую значение $f(x,y)$:

```
function z=ff(x,y)
z=x.^2;
```

Построим решение в Matlab, используя функцию `ode45`. Входными данными для функции `ode45` являются: имя файла, содержащего определение функции, стоящей в правой части уравнения, вектор, определяющий интервал интегрирования и начальное условие:

```
>> [X,Y]=ode45('ff',[1:0.01:5],1.3);
>> plot(X,Y)
>> grid on
```

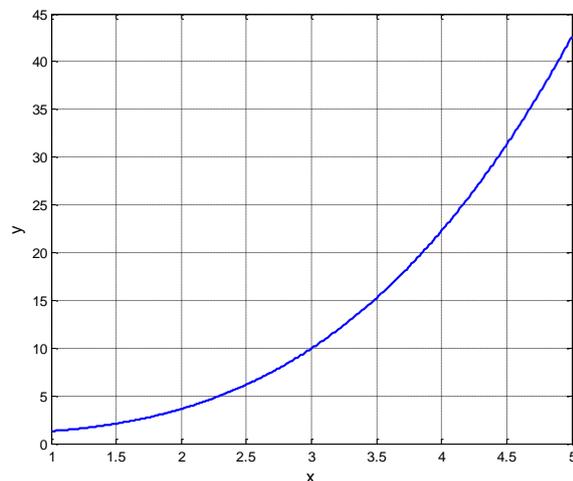


Рис. 5.1. Визуализация решения уравнения (5.18).

Пример 2. Рассмотрим решение дифференциального уравнения второго порядка на примере моделирования колебательного процесса в системе под

воздействием внешней силы при наличии сопротивления колебаниям. Процесс описывается уравнением второго порядка:

$$y'' + 2y' + 10y = \sin t \text{ с начальными условиями: } y(0) = 1, y'(0) = 0. \quad (5.19)$$

Для решения задачи средствами пакета, выполним понижение порядка уравнения, с этой целью введем замену переменной и уравнение второго порядка представим в виде системы: $y_1 = y, y_2 = y'$.

$$\begin{cases} y_1' = y_2 \\ y_2' = -2y_2 - 10y_1 + \sin t \end{cases} \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Построим решение в Matlab, используя функцию **ode**. Создадим вектор-функцию **oss** правых частей системы дифференциальных уравнений:

```
function z=oss(t,y)
z=[y(2); -2*y(2)-10*y(1)+sin(t)];
```

и выполним вызов функции **ode113**, входными параметрами которой являются: ссылка на функцию **oss**, отрезок для построения решения и вектор начальных условий:

```
Y0=[1;0]; %формирование вектора начальных значений
[T,Y]=ode113('oss',[0 15],Y0); % вызов солвера
plot(T,Y(:,1),'r.-') % вывод графика решения
hold on
plot(T,Y(:,2),'b.:') %вывод графика производной
```

Результат представлен на рисунке 5.2.

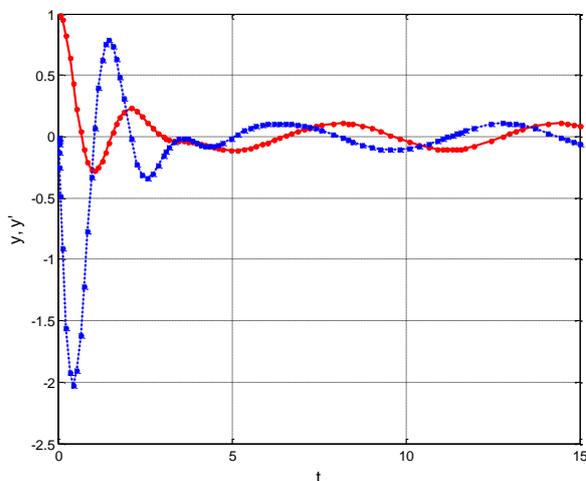


Рис. 5.2. Визуализация решения уравнения (5.19).

Пример 3. Опишем реализацию двух вариантов модели Лотки-Вольтерра «хищник-жертва». Классическая модель соперничества описывается задачей Коши для системы обыкновенных дифференциальных уравнений:

$$\begin{cases} y_1' = P \cdot y_1 - p \cdot y_1 \cdot y_2 \\ y_2' = -R \cdot y_2 + r \cdot y_1 \cdot y_2 \end{cases}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix},$$

где y_1 – число жертв, y_2 – число хищников, P – параметр, отвечающий за увеличение числа жертв в отсутствие хищников, R – параметр, отвечающий за уменьшение числа хищников в отсутствие жертв. Вероятность поедания хищником жертвы пропорциональна числу $y_1 \cdot y_2$, слагаемое $-p \cdot y_1 \cdot y_2$ – соответствует вымиранию жертв, $r \cdot y_1 \cdot y_2$ – появлению хищников.

Реализацию этой модели проведем с помощью встроенной функции пакета **ode23s**, вызов которой аналогичен функции `ode45` (функция `ode23s` предназначена для решения жестких задач, которой и является модель Лотки-Вольтерра).

Модификация этой модели предполагает, что увеличение числа хищников происходит не мгновенно, а по истечении некоторого времени τ после встречи хищника и жертвы. Модель трансформируется в систему дифференциальных уравнений с запаздыванием по аргументу:

$$\begin{cases} y_1'(t) = P \cdot y_1(t) - p \cdot y_1(t) \cdot y_2(t) \\ y_2'(t) = -R \cdot y_2(t) + r \cdot y_1(t - \tau) \cdot y_2(t - \tau) \end{cases}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix}.$$

Для реализации модели с запаздыванием воспользуемся встроенной функцией пакета **dde23**, входными аргументами которой являются: функция правой части системы, моменты запаздывания, функция предыстории, границы отрезка рассмотрения решения. Результатом вызова является – структура **sol**, которая состоит из:

sol.x – вектор-строка со значениями независимой переменной t , для которых найдено приближенное решение,

sol.y – массив со значениями y , каждая строка содержит его компоненты,

sol.yp – массив со значениями компонент производных от y .

Листинг инициализации вектор-функций правых частей уравнений системы, функция предыстории и вызовы солвером, приведен далее.

```
function F=oscil1(t,y)
% вектор-функция правых частей СОДУ
P=0.8;
R=1;
p=0.001;
r=p;
F=[P*y(1)-p*y(1)*y(2); -R*y(2)+r*y(1)*y(2)];

function z=oscil(t,y,Z)
% вектор-функция правых частей СОДУ с запаздыванием
P=0.8;
R=1;
p=0.001;
r=p;
z=[P*y(1)-p*y(1)*y(2); -R*y(2)+r*Z(1)*Z(2)];

function H=OsHist(t)
% вектор предыстории - будем считать предысторию постоянной
H=[1000; 1100];

% вызов солверов и построение графиков
sol=dde23('oscil',0.1,'OsHist',[0 50]);
[T,Y]=ode23s('oscil1',[0 50],[1000 1100]);
% фазовый портрет решения задачи с запаздыванием
plot(sol.y(1,:),sol.y(2),'bd-')
hold on
% фазовый портрет решения задачи для классической модели
plot(Y(:,1),Y(:,2),'r*-')
```

Результат на экране:

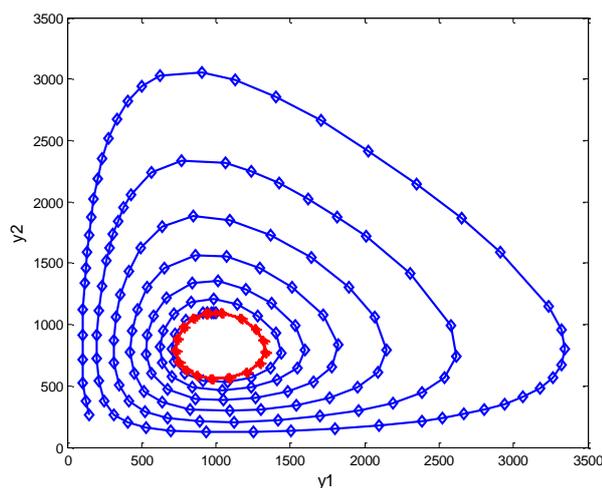


Рис. 5.3. Визуализация результата реализации модели хищник-жертва.

Модель «хищник-жертва» используется во многих приложениях для описания процессов соперничества, в экономике такими примера моделей являются: модель слияния и поглощения компаний, модель предприятия с учетом зависимости ставки заработной платы от численности трудящихся (объем производства – ставка заработной платы), модель дуопольного рынка с учетом инвестиционных и амортизационных расходов (объемы производств конкурирующих фирм), модель финансовой пирамиды, макроэкономическая модель Гудвина и др.

5.4 Индивидуальные задания

1. Реализовать модель о насыщении рынка автомобилей, математическая постановка которой представлена в виде (5.1) для следующих значений параметров:

<i>Параметр, ед.</i>	<i>Обозначение</i>	<i>Значение</i>
Период наблюдения динамики рынка, лет	$\Delta t, [t_0 t]$	[1 100]
Ежегодный выпуск автомобилей, ед.	P	10^6
Число автомобилей к началу исследований, ед.	y_0	$2 \cdot 10^6$
Срок службы автомобиля, лет	K	10

Для проведения анализа результатов моделирования ответить на вопросы. В течение какого периода времени наблюдается рост количества автомобилей? В какой период времени рост замедляется? В какой период времени наблюдается стабилизация рынка (оценить время релаксации)? Какое максимальное значение автомобилей достигается на рынке в условиях этой задачи?

2. Провести компьютерную реализацию модели развития финансовой пирамиды, описанной в разделе 5.2, при параметрах:

<i>Параметр, ед.</i>	<i>Обозначение</i>	<i>Значение</i>
Период наблюдения, день	$\Delta t, [t_0 t]$	[1 365]
Коэффициент ажиотажа, 1/день	a	10^{-7}
Число горожан, купивших акции в первый день, чел.	D	7
Число жителей, чел.	N	10^6
Временной лаг для покупки акций, день	T	50

Примечание

% модификация файла-функции правых частей СОДУ – продажи начинаются спустя T дней после покупки

```
if t<=T
```

```
z=[a*y(1)*y(2); -a*y(1)*y(2)];
```

```
else
```

```
z=[a*y(1)*y(2)-a*z(1)*z(2); a*z(1)*z(2)-a*y(1)*y(2)];
```

```
end
```

Для проведения анализа результатов моделирования визуализировать следующие зависимости и ответить на следующие вопросы.

1) Временная зависимость динамики изменения числа покупателей и держателей акций. Какой характер построенных зависимостей? Какой период времени требуется для перехода системы в стационарное состояние? Какими значениями числа покупателей и продавцов оно характеризуется (каково число

обманутых вкладчиков)? В какой день развития пирамиды число покупателей примерно равно числу продавцов?

2) Ежедневное количество покупок $K(t)$ и продаж $P(t)$ акций. Наблюдается ли процесс стабилизации? Как на характере поведения кривых отображается наличие временного лага?

3) Рассчитать показатели доходности пирамиды на основе данных моделирования. Проследите зарождение, кульминацию и гибель финансовой пирамиды для вкладчика и учредителя.

6 МАРКОВСКИЕ СЛУЧАЙНЫЕ ПРОЦЕССЫ

6.1 Понятие о марковских процессах

Пусть система может находиться в различных состояниях $E_0, E_1, \dots, E_n, \dots$, например, прибор может находиться в состояниях E_0 – исправен, E_1 – «барахлит» и требует мелкого ремонта, E_2 – неисправен и требует капитального ремонта. Обозначим вероятность нахождения системы в дискретный момент времени i в состоянии E_n через $P_n(i)$.

Тогда стохастический вектор $P^T = (P_0(i), P_1(i), \dots, P_n(i), \dots)$, $0 \leq P_n(i) \leq 1$, $n, i, > 0$, $\sum_n P_n = 1$ будет характеризовать состояние системы в дискретные моменты времени i .

Марковский процесс определяется как процесс, будущее состояние которого зависит только от состояния в текущий момент времени и не зависит от предыстории прихода системы в это состояние. Каждой паре состояний n, n' можно поставить в соответствие условную вероятность $P_{nn'}$ того, что система, находясь в состоянии E_n перейдет в состояние $E_{n'}$ в следующий момент времени. Для марковского процесса эта вероятность должна зависеть только от состояний n, n' . Для вероятности $P_{n'}(i+1)$ можно написать формулу, учитывая все возможные варианты перехода в это состояние:

$$P_{n'}(i+1) = \sum_n P_n(i) P_{nn'}, \quad n' = 1, 2, \dots \quad (6.1)$$

В векторной форме эта система может быть записана в следующем виде:

$$P^T(i+1) = P^T(i) \cdot J, \quad (6.2)$$

где квадратная матрица J образована из элементов $P_{nn'}$ и называется матрицей перехода. Марковский процесс полностью определяется матрицей перехода J и начальным состоянием системы $P(0)$:

$$\text{Т.о. } P_{i+1} = J^T \cdot P_i \quad (6.3)$$

Рассмотрим следующий пример для трех описанных выше состояний прибора.

Пример 1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.6 & 0.3 \\ 0.2 & 0.1 & 0.7 \end{pmatrix}, \quad P_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Каждая строка в этой матрице в сумме равна единице и отражает вероятности перехода из состояния строки в состояние столбца. Так из первой строки видно, что исправный прибор через некоторое время, принимаемое за единицу, останется в исправном с вероятностью 0.7, «забарахлит» с вероятностью 0.2 и перейдет в неисправное состояние с вероятностью 0.1.

Следует отметить, что подобным образом моделируется не сам вероятностный процесс, а его детерминированные вероятностные характеристики, отражающие усреднение множества отдельных процессов. Моделирование же отдельного процесса достигается средствами имитационного моделирования.

6.2 Уравнения Колмогорова. Примеры решения задач в Matlab

Для исследования изменения вероятностных характеристик в непрерывном времени используются уравнения Колмогорова как предельный случай дискретного марковского процесса при переходе к непрерывному времени. Пусть заданы интенсивности g_{mn} перехода системы из состояния n в состояние n' . При достаточно малых интервалах времени dt вероятность возникновения события перехода из одного состояния в другое для простейшего потока событий пропорциональна длине интервала:

$$P_1(dt) = \lambda dt,$$

где λ – интенсивность потока. Тогда по (2) имеем

$$\frac{P(t+dt) - P(t)}{dt} = (J^T - E)P + \frac{o(dt)}{dt},$$

где E – единичная матрица.

Переходя к пределу, получаем дифференциальное уравнение Колмогорова для условных вероятностей J :

$$\frac{dP(t)}{dt} = (J^T - E)P, \quad P(0) = P_0. \quad (6.4)$$

Процедуру составления уравнений Колмогорова для интенсивностей потока событий рассмотрим на следующем примере модели развития популяции.

Пример 2. Пусть имеем следующие 4 состояния особи некоторой популяции: E_1 с вероятностью $p_1(0)$ – рождение особи; E_2 с вероятностью $p_2(0)$ – особь в здоровом состоянии; E_3 с вероятностью $p_3(0)$ – особь больна и E_4 с вероятностью $p_4(0)$ – смерть особи. Процесс развития популяции можно представить в виде графа состояний и переходов (рисунок 6.1):

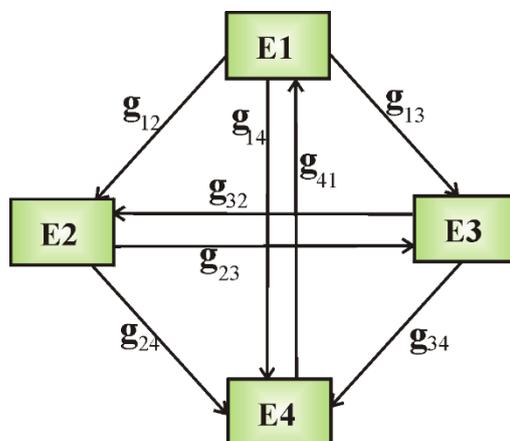


Рис. 6.1. Граф состояний популяции.

Уравнения Колмогорова, соответствующие этому графу:

$$\frac{dP_1}{dt} = -(g_{12} + g_{13} + g_{14})P_1 + g_{41}P_4,$$

$$\frac{dP_2}{dt} = g_{12}P_1 + g_{32}P_3 - (g_{23} + g_{24})P_2,$$

$$\frac{dP_3}{dt} = g_{13}P_1 + g_{23}P_2 - (g_{32} + g_{34})P_3,$$

$$\frac{dP_4}{dt} = g_{14}P_1 + g_{24}P_2 + g_{34}P_3 - g_{41}P_4.$$

Учитывая, что система может находиться только в этих состояниях с вероятностью, равной 1, окончательно, отбрасывая одно из дифференциальных уравнений, получим систему уравнений Колмогорова с начальными условиями,

отражающими вероятность нахождения системы в одном из состояний в начальный момент времени:

$$\begin{cases} \frac{dP_1}{dt} = -(g_{12} + g_{13} + g_{14})P_1 + g_{41}P_4, \\ \frac{dP_2}{dt} = g_{12}P_1 + g_{32}P_3 - (g_{23} + g_{24})P_2, \\ \frac{dP_3}{dt} = g_{13}P_1 + g_{23}P_2 - (g_{323} + g_{34})P_3, \\ P_1 + P_2 + P_3 + P_4 = 1, \end{cases}$$

$$P_1(0) = P_1^0, P_2(0) = P_2^0, P_3(0) = P_3^0.$$

Исключив из этой системы P_4 , окончательно получим:

$$\begin{cases} \frac{dP_1}{dt} = -(g_{12} + g_{13} + g_{14})P_1 + g_{41}(1 - P_1 - P_2 - P_3), \\ \frac{dP_2}{dt} = g_{12}P_1 + g_{32}P_3 - (g_{23} + g_{24})P_2, \\ \frac{dP_3}{dt} = g_{13}P_1 + g_{23}P_2 - (g_{323} + g_{34})P_3, \end{cases} \quad (6.5)$$

$$P_1(0) = P_1^0, P_2(0) = P_2^0, P_3(0) = P_3^0.$$

Программа в ППП Matlab, интегрирующая уравнения Колмогорова в форме (6.5) для рассматриваемого примера с матрицей интенсивностей, приведена ниже (рисунок 6.2).

```
function F=oscil(t,Y)
F=[-12*Y(1)-2*Y(2)-2*Y(3)+2;8*Y(1)-3*Y(2)+5*Y(3);Y(1)+Y(2)-7*Y(3)];

function solveode
Y0=[0.3; 0.7; 0];
[T,Y]=ode113('oscil',[0 1],Y0);
plot(T,Y(:,1),'r.-')
grid on;
hold on
plot(T,Y(:,2),'k.-')
plot(T,Y(:,3),'b.-')
legend('p1','p2','p3',3)
```

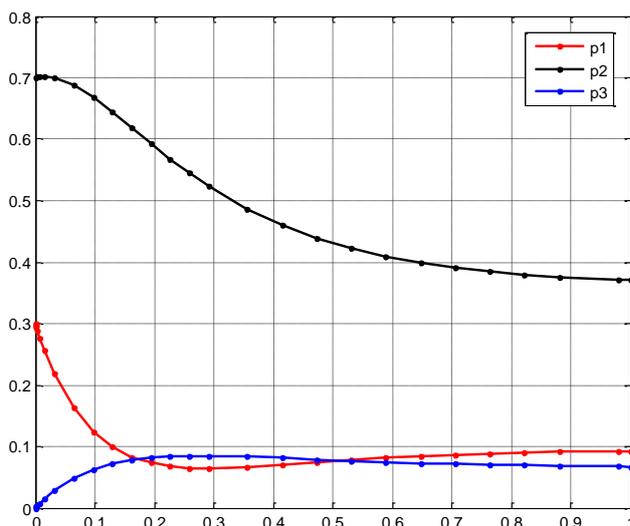


Рис. 6.2. Реализация примера 2.

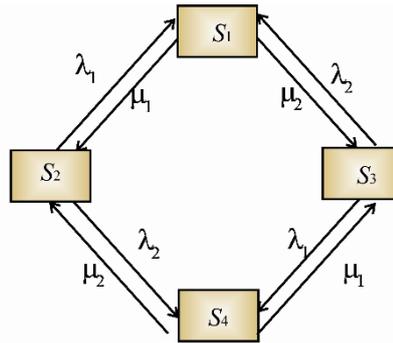
Интегрирование уравнений Колмогорова с интенсивностями перехода.

6.3 Индивидуальные задания

Выполнить задание согласно предложенному варианту. Оформить отчет по лабораторной работе.

Вариант	Задания
1	<p>1. Пусть известна матрица перехода для данного процесса и его начальное состояние:</p> $J = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.2 & 0.2 & 0.6 \end{pmatrix}, P_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$ <p>Рассмотреть поведение системы за 30 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?</p> <p>2. Работа системы, состоящей из двух независимо работающих</p>

приборов, задается графом состояний:



где состояние S_1 – оба прибора исправны и работают, S_2 – первый прибор вышел из строя, а второй – исправно работает, S_3 – второй прибор вышел из строя, а первый – исправно работает, S_4 – оба прибора вышли из строя. Интенсивности переходов задаются величинами: $\lambda_1 = 5, \mu_1 = 2, \lambda_2 = 2, \mu_2 = 1$.

Провести имитационное моделирование системы при начальных данных $S = [0 \ 1 \ 0 \ 0]$.

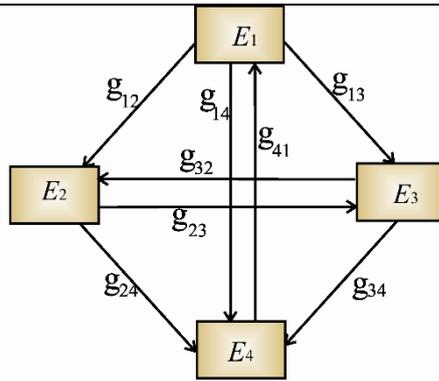
2

1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.5 & 0.1 & 0.4 \\ 0.2 & 0.7 & 0.1 \\ 0.15 & 0.15 & 0.7 \end{pmatrix} \quad P_0 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Рассмотреть поведение системы за 35 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Пусть имеем следующие 4 состояния особи некоторой популяции: E_1 с вероятностью $p_1(0)$ – рождение особи; E_2 с вероятностью $p_2(0)$ – особь в здоровом состоянии; E_3 с вероятностью $p_3(0)$ – особь больна и E_4 с вероятностью $p_4(0)$ – смерть особи. Процесс развития популяции можно представить в виде графа состояний и переходов:



Провести моделирование работы этой системы при параметрах: $g_{12} = 5; g_{13} = 2; g_{24} = 2; g_{34} = 3; g_{32} = g_{23} = 2.5; g_{14} = g_{41} = 3.5$ и при начальных данных $E = [0.7 \ 0.2 \ 0.1 \ 0]$.

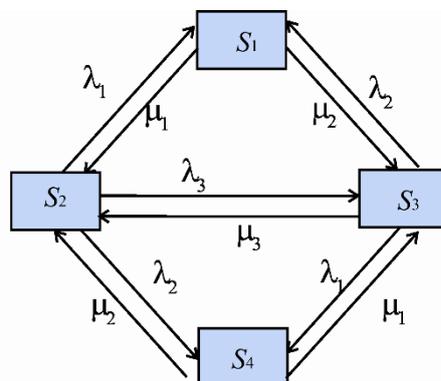
3

1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.25 & 0.25 & 0.5 \\ 0.2 & 0.6 & 0.2 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} \quad P_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Рассмотреть поведение системы за 70 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Работа системы обслуживания клиентов, состоящей из двух работающих касс, задается графом состояний:



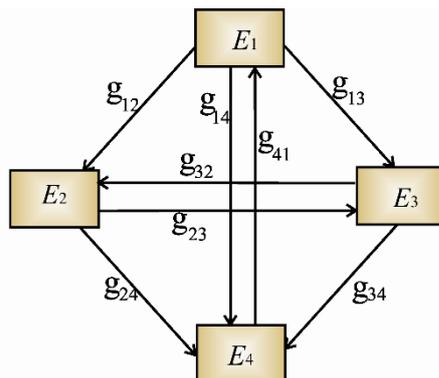
где состояние S_1 – обе кассы работают, S_2 – первая касса не обслуживает, а вторая – обслуживает, S_3 – обслуживает только первая касса, S_4 – обе кассы не обслуживают. Интенсивности пере-

	<p>ходов задаются величинами:</p> $\lambda_1 = 5, \mu_1 = 1, \lambda_2 = 2.5, \mu_2 = 2, \lambda_3 = 7, \mu_3 = 3.$ <p>Провести имитационное моделирование системы при начальных данных $S = [0 \ 0.5 \ 0.5 \ 0]$.</p>
4	<p>1. Пусть известна матрица перехода для данного процесса и его начальное состояние:</p> $J = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0.3 & 0.6 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix} \quad P_0 = \begin{pmatrix} 0 \\ 0.5 \\ 0.5 \end{pmatrix}.$ <p>Рассмотреть поведение системы за 100 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?</p> <p>2. Работа системы, состоящей из двух независимо работающих приборов, задается графом состояний:</p> <p>где состояние S_1 – оба прибора исправны и работают, S_2 – первый прибор вышел из строя, а второй – исправно работает, S_3 – второй прибор вышел из строя, а первый – исправно работает, S_4 – оба прибора вышли из строя. Интенсивности переходов задаются величинами: $\lambda_1 = 3, \mu_1 = 1, \lambda_2 = 5, \mu_2 = 1$.</p> <p>Провести имитационное моделирование системы при начальных данных $S = [0 \ 0.5 \ 0.5 \ 0]$.</p>
5	<p>1. Пусть известна матрица перехода для данного процесса и его начальное состояние:</p>

$$J = \begin{pmatrix} 0.3 & 0.3 & 0.4 \\ 0.1 & 0.8 & 0.1 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} \quad P_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Рассмотреть поведение системы за 55 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Пусть имеем следующие 4 состояния особи некоторой популяции: E_1 с вероятностью $p_1(0)$ – рождение особи; E_2 с вероятностью $p_2(0)$ – особь в здоровом состоянии; E_3 с вероятностью $p_3(0)$ – особь больна и E_4 с вероятностью $p_4(0)$ – смерть особи. Процесс развития популяции можно представить в виде графа состояний и переходов:



Провести моделирование работы этой системы при параметрах: $g_{12} = 10; g_{13} = 8; g_{24} = 9; g_{34} = 5; g_{32} = g_{23} = 2; g_{14} = g_{41} = 3$ и при начальных данных $E = [0.25 \ 0.25 \ 0.25 \ 0.25]$.

6

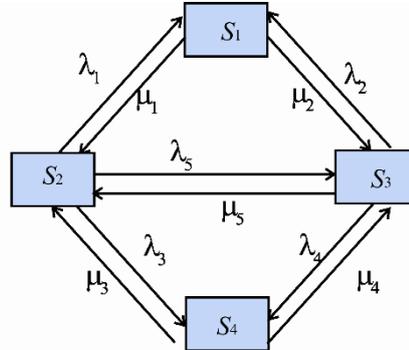
1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.15 & 0.15 & 0.7 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.4 & 0.4 \end{pmatrix} \quad P_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Рассмотреть поведение системы за 100 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состоя-

ние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Работа системы технологического производства, состоящей из двух работающих станков, задается графом состояний:



где состояние S_1 – оба станка работают, S_2 – работает только второй станок, S_3 – работает только первый станок, S_4 – оба станка не работают. Интенсивности переходов задаются величинами: $\lambda_1 = 2, \mu_1 = 1, \lambda_2 = 3, \mu_2 = 2, \lambda_3 = 3, \mu_3 = 3, \lambda_4 = 1, \mu_4 = 3, \lambda_5 = 5, \mu_5 = 7$. Провести имитационное моделирование системы при начальных данных $S = [1 \ 0 \ 0 \ 0]$.

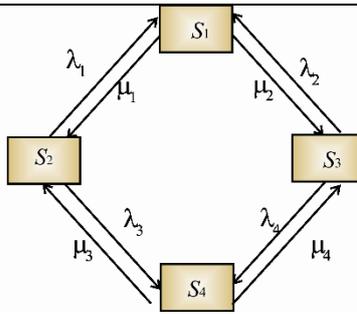
7

1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.3 & 0.45 & 0.25 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.2 & 0.6 \end{pmatrix} \quad P_0 = \begin{pmatrix} 0 \\ 0.5 \\ 0.5 \end{pmatrix}.$$

Рассмотреть поведение системы за 500 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Работа системы, состоящей из двух независимо работающих приборов, задается графом состояний:



где состояние S_1 – оба прибора исправны и работают, S_2 – первый прибор вышел из строя, а второй – исправно работает, S_3 – второй прибор вышел из строя, а первый – исправно работает, S_4 – оба прибора вышли из строя. Интенсивности переходов задаются величинами:

$$\lambda_1 = 2, \mu_1 = 5, \lambda_2 = 3, \mu_2 = 1, \lambda_3 = 3, \mu_3 = 5, \lambda_4 = 2, \mu_4 = 2.$$

Провести имитационное моделирование системы при начальных данных $S = [1 \ 0 \ 0 \ 0]$.

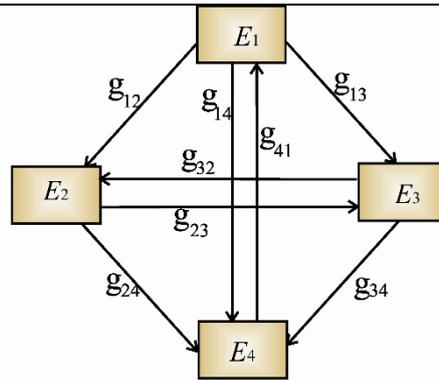
8

1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.25 & 0.25 & 0.5 \\ 0.2 & 0.8 & 0 \\ 0.6 & 0 & 0.4 \end{pmatrix} \quad P_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Рассмотреть поведение системы за 50 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Пусть имеем следующие 4 состояния особи некоторой популяции: E_1 с вероятностью $p_1(0)$ – рождение особи; E_2 с вероятностью $p_2(0)$ – особь в здоровом состоянии; E_3 с вероятностью $p_3(0)$ – особь больна и E_4 с вероятностью $p_4(0)$ – смерть особи. Процесс развития популяции можно представить в виде графа состояний и переходов:



Провести моделирование работы этой системы при параметрах: $g_{12} = 1; g_{13} = 5; g_{24} = 3; g_{34} = 7; g_{32} = 1; g_{23} = 2; g_{14} = 5; g_{41} = 8$ и при начальных данных $E = [1 \ 0 \ 0 \ 0]$.

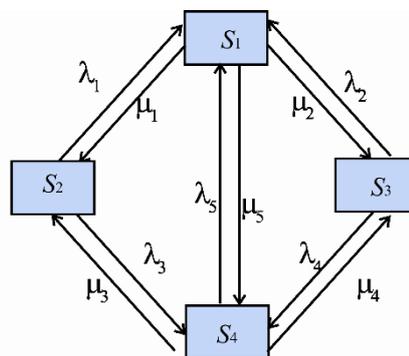
9

1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.15 & 0.15 & 0.7 \\ 0.3 & 0.6 & 0.1 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} \quad P_0 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Рассмотреть поведение системы за 200 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Работа системы технологического производства, состоящей из двух работающих станков, задается графом состояний:



где состояние S_1 – оба станка работают, S_2 – работает только второй станок, S_3 – работает только первый станок, S_4 – оба станка не работают. Интенсивности переходов задаются величинами: $\lambda_1 = 2, \mu_1 = 1, \lambda_2 = 4, \mu_2 = 2, \lambda_3 = 3, \mu_3 = 3, \lambda_4 = 1, \mu_4 = 1, \lambda_5 = 5, \mu_5 = 7$

. Провести имитационное моделирование системы при начальных данных $S = [1 \ 0 \ 0 \ 0]$.

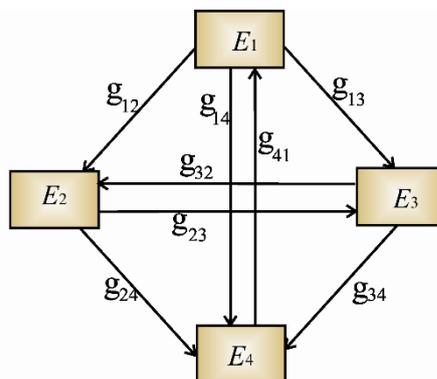
10

1. Пусть известна матрица перехода для данного процесса и его начальное состояние:

$$J = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.1 & 0.9 & 0 \\ 0.25 & 0.35 & 0.4 \end{pmatrix} \quad P_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Рассмотреть поведение системы за 100 интервалов времени, воспользовавшись соотношением (6.3). Через сколько шагов состояние процесса стабилизируется? Что означает полученное распределение вероятностей?

2. Пусть имеем следующие 4 состояния особи некоторой популяции: E_1 с вероятностью $p_1(0)$ – рождение особи; E_2 с вероятностью $p_2(0)$ – особь в здоровом состоянии; E_3 с вероятностью $p_3(0)$ – особь больна и E_4 с вероятностью $p_4(0)$ – смерть особи. Процесс развития популяции можно представить в виде графа состояний и переходов:



Провести моделирование работы этой системы при параметрах: $g_{12} = 5; g_{13} = 7; g_{24} = 2; g_{34} = 1; g_{32} = 4; g_{23} = 5; g_{14} = 1; g_{41} = 3$ и при начальных данных $E = [1 \ 0 \ 0 \ 0]$.

7 СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ

Реальные моделируемые процессы часто носят вероятностный характер. Такой характер имеет, например, время заявок на переговоры, поступающих в телефонную сеть. Такого рода вероятностные процессы описываются моделями систем массового обслуживания (СМО), в простейшем случае СМО, имеющими структуру, показанную на рисунке 7.1.



Рис. 7.1. Структурная схема СМО.

В этой схеме входной поток заявок представляет собой события (точки на временной оси). Заявки поступают на блок очереди (если она есть) и в соответствии с дисциплиной очереди (например, по времени поступления заявки) поступают на прибор обслуживания, где происходит его обработка, например ремонт. В состав СМО входят блок очереди и обслуживающий прибор. Примерами СМО могут служить бензозаправки, парикмахерские, переговорные пункты, компьютерные сети и множество других систем обслуживания. Во всех этих системах обращение в СМО носит случайный характер, такой же характер имеет и время обслуживания каждой заявки. Однако для оценки работы СМО необходимо иметь детерминированные характеристики. Ими могут быть вероятность обслуживания, среднее время обслуживания, средняя длина очереди и др., На основе таких оценок можно делать выводы о качестве СМО и давать рекомендации по улучшению их работы.

СМО принято классифицировать по ряду признаков. Прежде всего они разделяются на марковские и немарковские. Такое разделение аналогично раз-

делению на линейные и нелинейные модели физических процессов. Для марковских процессов возможен их детальный анализ.

Данные системы подразделяются на СМО с потерями и без потерь. В первом случае заявка может покинуть систему, если она занята, во втором ожидает своей очереди.

В одноканальных СМО имеется один прибор обслуживания, многоканальные имеют несколько таких приборов.

В многофазных СМО заявка после обслуживания одним прибором поступает для дальнейшего обслуживания на прибор другого типа. Наконец, в сложных СМО могут присутствовать СМО различных типов, образуя сеть СМО.

7.1 Одноканальная СМО с отказами

Применим рассмотренный аппарат для анализа простейшей СМО. В одноканальной СМО с отказами имеется один прибор для обслуживания. Если он занят, заявка покидает систему, Примером такой СМО может служить справочная с одной телефонной линией. Обозначим интенсивность потока заявок (число заявок в единицу времени) через λ , интенсивность потока обслуживания через μ . Среднее время между заявками и среднее время обслуживания составят соответственно величины $\frac{1}{\lambda}$, $\frac{1}{\mu}$. В графе состояний P_0 соответствует свободно-му каналу – отсутствию заявки, P_1 – занятому (рисунок 7.1).

Уравнения Колмогорова имеют следующий вид:

$$\begin{cases} \frac{dp_0}{dt} = \mu p_1 - \lambda p_0, \\ \frac{dp_1}{dt} = -\mu p_1 + \lambda p_0. \end{cases} \quad (7.1)$$

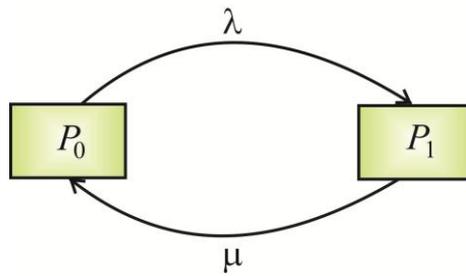


Рис. 7.1. Граф состояний одноканальной СМО с отказами.

Используя равенство $p_0 + p_1 = 1$, получим одно дифференциальное уравнение

$$\frac{dp_0}{dt} = \mu - (\lambda + \mu)p_0.$$

Решая его с начальным условием $p_0(0) = p_0^0$, получим выражение для вероятности свободного канала

$$p_0(t) = p_0^0 \exp(-(\lambda + \mu)t) + \frac{\mu}{\lambda + \mu} (1 - \exp(-(\lambda + \mu)t)).$$

При $t \rightarrow \infty$ получаем установившуюся вероятность, не зависящую от того, в каком состоянии была система в начальный момент времени:

$$p_0(\infty) = \frac{\mu}{\lambda + \mu}.$$

Этот же результат можно получить, не решая дифференциальных уравнений. Приравнявая к нулю производные в системе (2), получаем систему алгебраических линейных уравнений для установившегося состояния:

$$\mu - (\lambda + \mu)p_0 = 0.$$

7.2 Многоканальная СМО с отказами

В многоканальной СМО имеется n приборов для обслуживания заявок, что отражено в графе состояний такой системы (рисунок 7.2). Система уравнений Колмогорова для многоканальной СМО с отказами будет следующей:

$$\begin{cases} \frac{dp_0}{dt} = \mu p_1 - \lambda p_0, \\ \frac{dp_i}{dt} = \lambda p_{i-1} - (\lambda + i\mu)p_i + (i+1)\mu p_{i+1}, \quad i = 1, 2, \dots, n-1, \\ \frac{dp_n}{dt} = \lambda p_{n-1} - n\mu p_n. \end{cases}$$

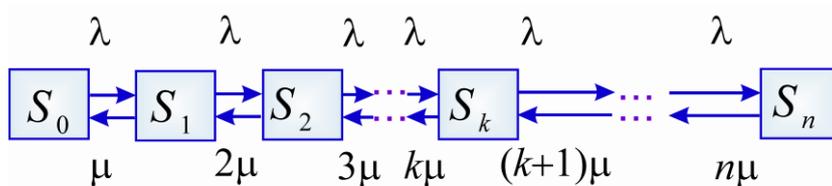


Рис. 7.2. Граф состояний для многоканальной СМО с отказами.

В установившемся режиме работы решение алгебраической системы уравнений для установившегося режима дает следующие результаты:

$$p_i = p_0 \frac{\rho^i}{i!}, \quad \rho = \frac{\lambda}{\mu}, \quad p_0 = \frac{1}{\sum_{i=0}^n \frac{\rho^i}{i!}}.$$

Определим характеристики такой СМО:

1) вероятность простоя (отсутствие заявки)

$$p_0 = \frac{1}{\sum_{i=0}^n \frac{\rho^i}{i!}}.$$

2) вероятность отказа (в системе находится s заявок, все каналы заняты):

$$P_{отк} = p_n = \frac{\rho^n}{n!} p_0.$$

3) среднее число заявок, поступивших за время T : $\lambda \cdot T$;

4) среднее время обслуживания одной заявки $\frac{1}{\mu}$;

5) среднее время обслуживания одним каналом заявок, поступивших за

время T : $\frac{\lambda}{\mu} T = \rho T$;

б) среднее число занятых каналов

$$\bar{k} = \left(1 - \frac{\rho^n}{n!} p_0 \right) \rho.$$

7.3 Многоканальная СМО с очередью

В случае многоканальной СМО с очередью организуется очередь, рассчитанная на n заявок. Если число заявок превышает число каналов, следующая заявка ожидает своей очереди на обслуживание. Если число заявок в системе превышает $n+r$, заявка покидает систему, граф состояний для такой СМО показан на рисунке 7.3.

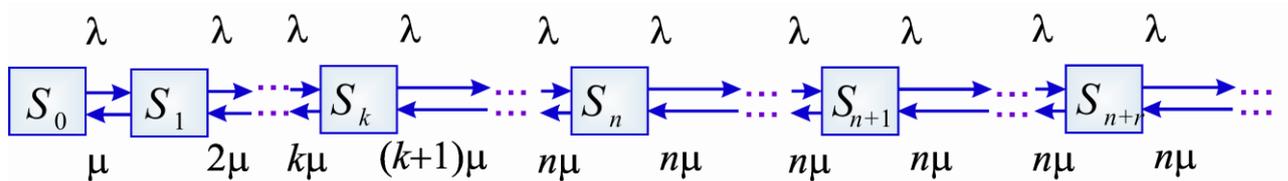


Рис. 7.3. Граф состояний для многоканальной системы с очередью.

Уравнения Колмогорова для такой системы имеют следующий вид:

$$\begin{cases} \frac{dp_0}{dt} = \mu p_1 - \lambda p_0, \\ \frac{dp_i}{dt} = \lambda p_{i-1} - (\lambda + i\mu) p_i + (i+1)\mu \lambda p_{i+1}, \quad i = 1, 2, \dots, n-1, \\ \frac{dp_i}{dt} = \lambda p_{i-1} - (\lambda + n\mu) p_i + n\mu p_{i+1}, \quad i = n+1, n+2, \dots, n+r-1, \\ \frac{dp_{n+r}}{dt} = \lambda p_{n+r-1} - n\mu p_{n+r}. \end{cases}$$

Решение алгебраической системы в установившемся состоянии дает следующие соотношения:

$$p_0 = \left(\sum_{i=0}^{n-1} \frac{\rho^i}{i!} + \sum_{i=n}^{n+r} \frac{n^n}{n!} \left(\frac{\rho}{n} \right)^i \right)^{-1},$$

$$p_1 = \rho p_0, p_2 = \frac{\rho^2}{2!} p_0, \dots, p_k = \frac{\rho^k}{k!} p_0, \dots, p_n = \frac{\rho^n}{n!} p_0,$$

$$P_{n+1} = \frac{\rho^{n+1}}{n! \cdot n} P_0, \dots, P_{n+r} = \frac{\rho^{n+r}}{n! \cdot n} P_0, \dots$$

Для случая неограниченных мест в очереди последнее выражение записывается проще:

$$P_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \dots + \frac{\rho^{n-1}}{(n-1)!} + \frac{\rho^n n}{n!(n-p)} \right)^{-1}.$$

В этих выражениях $\rho/n < 1$ должно быть меньше 1, иначе приборы не успевают обработать поток заявок, и очередь неограниченно растет.

Можно выделить следующие, дополнительные по сравнению со СМО без очереди, характеристики системы:

1) среднее число заявок в очереди:

$$\bar{v} = \frac{\rho^{n+1}}{(n-1)!(n-\rho)^2} P_0;$$

2) общее число заявок в системе:

$$\bar{n} = \bar{v} + \rho;$$

3) среднее время ожидания заявки:

$$\bar{t} = \frac{\bar{v}}{\lambda};$$

4) среднее время пребывания в системе:

$$\bar{T} = \frac{\bar{n}}{\lambda}.$$

7.4 Индивидуальные задания

1. Провести моделирование работы одноканальной СМО с отказами (найти зависимости вероятностей простоя и отказа). Принять условия, соответствующие варианту. К какому значению будет стремиться вероятность отказа при $\lambda \gg \mu$? Провести и оформить анализ результатов.

Вариант	Параметры
1	$\lambda = 5, \mu = 7, p_0 = 1, t = 0,1$

2	$\lambda = 10, \mu = 7, p_0 = 1, t = \overline{0,2}$
3	$\lambda = 4, \mu = 6, p_0 = 1, t = \overline{0,1}$
4	$\lambda = 2, \mu = 3, p_0 = 1, t = \overline{0,1}$
5	$\lambda = 3, \mu = 6, p_0 = 1, t = \overline{0,1}$
6	$\lambda = 4, \mu = 3, p_0 = 1, t = \overline{0,1}$
7	$\lambda = 10, \mu = 6, p_0 = 1, t = \overline{0,1}$
8	$\lambda = 4, \mu = 7, p_0 = 1, t = \overline{0,1}$
9	$\lambda = 2, \mu = 3, p_0 = 1, t = \overline{0,1}$
10	$\lambda = 1, \mu = 3, p_0 = 1, t = \overline{0,1}$

2. Провести моделирование СМО (найти зависимости вероятностей простоя и отказа): автомобили подходят к автозаправочной станции, среднее количество – N в час, заправка каждого автомобиля занимает в среднем $T_{\text{обсл}}$. Число колонок на АЗС – n . Рассмотреть процесс в течение времени $t \in [0, T]$. Считать, что при всех занятых колонках автомобиль покинет заправку.

Найти характеристики многоканальной СМО с отказами в динамике, а также в установившемся режиме. Провести и оформить анализ результатов.

Вариант	Параметры
1	$N = 120, T_{\text{обсл}} = 3 \text{ мин}, n = 4, T = 10.$
2	$N = 60, T_{\text{обсл}} = 5 \text{ мин}, n = 5, T = 10.$
3	$N = 60, T_{\text{обсл}} = 5 \text{ мин}, n = 3, T = 10.$
4	$N = 120, T_{\text{обсл}} = 3 \text{ мин}, n = 4, T = 10.$
5	$N = 60, T_{\text{обсл}} = 8 \text{ мин}, n = 4, T = 10.$
6	$N = 120, T_{\text{обсл}} = 4 \text{ мин}, n = 4, T = 10.$
7	$N = 150, T_{\text{обсл}} = 7 \text{ мин}, n = 4, T = 10.$
8	$N = 120, T_{\text{обсл}} = 6 \text{ мин}, n = 4, T = 10.$

9	$N = 40, T_{обсл} = 5 \text{ мин}, n = 10, T = 10.$
10	$N = 90, T_{обсл} = 5 \text{ мин}, n = 4, T = 10.$

3. Провести моделирование СМО (найти зависимости вероятностей простоя и отказа): клиенты обращаются к электронной справке железнодорожного вокзала, средняя интенсивность заявок λ (у.е.), интенсивность обслуживания – μ (у.е.). Число электронных терминалов справки – n . Рассмотреть процесс в течение времени $t \in [0, T]$.

Решить динамическую задачу, считая, что в очереди клиентов есть только r мест. Рассчитайте также основные характеристики многоканальной СМО с очередью в установившемся режиме.

Провести и оформить анализ результатов.

Вариант	Параметры
1	$\lambda = 2, \mu = 3, n = 8, T = 10, r = 3$
2	$\lambda = 2, \mu = 5, n = 8, T = 10, r = 5$
3	$\lambda = 5, \mu = 6, n = 4, T = 10, r = 3$
4	$\lambda = 5, \mu = 3, n = 8, T = 10, r = 4$
5	$\lambda = 2, \mu = 2, n = 8, T = 10, r = 4$
6	$\lambda = 3, \mu = 4, n = 8, T = 10, r = 3$
7	$\lambda = 2, \mu = 3, n = 5, T = 10, r = 4$
8	$\lambda = 3, \mu = 5, n = 8, T = 10, r = 3$
9	$\lambda = 2, \mu = 3, n = 8, T = 10, r = 2$
10	$\lambda = 4, \mu = 3, n = 5, T = 10, r = 4$

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. Matlab 7. – СПб.: БХВ – Петербург, 2005. – 1104 с.
2. Вентцель Е.С. Исследование операций: задачи, принципы, методология. – М.: Дрофа, 2006. – 206 с.
3. Иглин С.П. Математические расчеты на базе Matlab. – СПб.: БХВ-Петербург, 2005.– 640 с.
4. Мажукин В.И., Королева О.Н. Математическое моделирование в экономике: Часть III. Экономические приложения: учеб. пособие – М.: Флинта: МПСИ, 2005. – 176 с.
5. Мэтьюз Д.Г., Куртис Ф.Д. Численные методы. Использование Matlab / пер. с англ. – М.: Вильямс, 2001. – 720 с.
6. Охорзин В.А. Компьютерное моделирование в системе Mathcad. – М.: Финансы и статистика, 2006. – 144 с.
7. Цисарь И.Ф., Нейман В.Г. Компьютерное моделирование экономики. – М.: Диалог МИФИ, 2008. – 384 с.

СОДЕРЖАНИЕ

Введение	3
1 Основные приемы работы с пакетом Matlab	5
1.1 Работа в командном окне	5
1.2 Циклы. Условные операторы и операторы отношения	12
1.3 М-файлы	13
1.4 Графики	17
1.5 Диаграммы и гистограммы	22
1.6 Индивидуальные задания	25
2 Оптимальная ставка налога. Расчеты по обслуживанию кредитов	28
2.1 Имитационная модель определения оптимальной ставки налога	28
2.2 Использование функций ППП Matlab для выполнения расчетов по обслуживанию кредитов	31
2.3 Индивидуальные задания	33
3 Оценка доходности банковских операций. Оценка доходности облигаций	35
3.1 Математическая модель оценки доходности облигации: концептуальная и математическая постановка задачи	35
3.2 Математическая модель оценки доходности банковских операций: концептуальная и математическая постановка задачи	39
3.3 Примеры решения задач в Matlab	40
3.4 Индивидуальные задания	45
4 Экономические приложения задач математического программирования	47
4.1 Решение задачи линейного программирования в ППП Matlab	47
4.2 Математическая модель определения доходности портфеля ценных бумаг	48
4.3 Примеры решения задач в Matlab	51
4.4 Индивидуальные задания	54
5 Динамические (линейные и нелинейные модели)	61

5.1	Задача о насыщении рынка автомобилей	61
5.2	Модели соперничества в экономике: модель финансовой пирамиды	61
5.3	Приближенное решения задач Коши для обыкновенных дифференциальных уравнений в ППП Matlab	66
5.4	Индивидуальные задания	71
6	Марковские случайные процессы	74
6.1	Понятие о марковских процессах	74
6.2	Уравнения Колмогорова. Примеры решения задач в Matlab	75
6.3	Индивидуальные задания	78
7	Системы массового обслуживания	87
7.1	Одноканальная СМО с отказами	88
7.2	Многоканальная СМО с отказами	89
7.3	Многоканальная СМО с очередью	91
7.4	Индивидуальные задания	92
	Библиографический список	95

*Александра Васильевна Кубышкина,
доцент кафедры информационных систем и технологий
кандидат экономических наук, доцент*

Математическое моделирование: учебно-методическое пособие

Изд-во Брянский ГАУ. Подписано к печати 30.05.2016. Формат 60×84/16.

Усл. печ. л. 6. Тираж 100. Заказ ***.

Отпечатано в издательстве ФГБОУ ВО Брянский ГАУ.