А. Ф. Дащенко, В. Х. Кириллов, Л. В. Коломиец, В. Ф. Оробей

MATLAB

В ИНЖЕНЕРНЫХ И НАУЧНЫХ РАСЧЕТАХ

Одесса «Астропринт» 2003

ББК
Д
УДК 539.3:681.3

Монография посвящена иллюстрации возможностей одной из самых эффективных систем компьютерной математики MATLAB в решении ряда научных и инженерных проблем. Рассмотрены примеры решения задач математического анализа. Классические численные методы дополнены примерами более сложных инженерных и научных задач математической физики. Подробно изложены алгоритмы краевых задач статики, динамики и устойчивости упругих систем в методе граничных элементов. Рассмотрены пакеты расширений системы MATLAB символьных вычислений, оптимизации и моделирования. Применение компьютерных технологий решения задач посредством системы MATLAB иллюстрируется большим количеством упражнений и примеров.

Предназначена для студентов и аспирантов высших технических учебных заведений, будет полезна специалистам и научным работникам широкого профиля.

Рецензенты: заведующий кафедрой сопротивления материалов и строительной механики Одесского государственного морского университета, доктор технических наук, профессор **В.А. Гришин**;
заведующий кафедрой высшей математики Одесского национального политехнического университета, доктор технических наук, профессор **А. В. Усов**

Рекомендовано к изданию ученым советом института машиностроения Одесского национального политехнического университета от 17 декабря 2002 г.

Д $\frac{-016}{549-2003}$ Без объявл.

ISBN 966-549-

© А.Ф. Дащенко, В. Х. Кириллов,
Л. В. Коломиец, В. Ф. Оробей, 2003

ВВЕДЕНИЕ

Данная книга посвящена иллюстрации возможностей одной из самых эффективных систем компьютерного программного обеспечения – пакета универсальных интегрированных программ MATLAB. Любознательному читателю предлагается ознакомиться в первом приближении с основами языка программирования и комплексной визуализации результатов решения ряда научных и инженерных задач. Рассматриваются такие проблемы как табулирование функций, решение нелинейных уравнений, поиск оптимальных решений, решение задач Коши, численное интегрирование и другие задачи, традиционно включаемые в курс численных методов. Алгоритм этих задач хорошо известен и разработчики системы MATLAB (фирма Math Works, Inc., U.S.A.) учли опыт численного решения и программирования задач вычислительной математики за все время существования вычислительной техники. Поэтому в системе MATLAB по каждой проблеме имеется несколько программ (иногда их более 10), предназначенных для ее решения в зависимости от особенностей данной задачи. Кроме чисто научных задач, средствами MATLAB могут быть успешно решены и довольно сложные инженерные проблемы, такие, как поиск спектра частот собственных колебаний и критических сил потери устойчивости стержневых, пластинчатых и оболочечных систем, решение краевых задач для упругих систем и задач сейсмостойкости сооружений и др. Численные результаты таких задач должны сопровождаться соответствующими эпюрами и формами, т. е. визуализацией расчетов. MATLAB обеспечивает решение и этой проблемы, что очень привлекательно для учебного процесса, где студенты тратят достаточно много времени на графическую часть курсовых и дипломных проектов. Кроме MATLAB существуют и другие, довольно мощные среды программирования и визуализации, такие как Visual Digital Fortran, Delphi, Visual C++ и т. п. Однако, по мнению авторов, в системе MATLAB получаются наиболее простые и в то же время эффективные программы. Читателю предлагается самому убедить-

ся в этом, сравнив приведенные в данной книге программы с программами других сред. Вероятно, нельзя создать в других средах программирования программы более простые, чем в системе MATLAB.

В качестве инженерных задач в пособии рассматриваются задачи строительной механики – науки о расчетах сооружений на статическую, динамическую нагрузки и устойчивость. Для решения задач строительной механики разработано множество методов – методы сил и перемещений, метод конечных элементов, метод конечных разностей, метод R-функций, метод граничных элементов и др. Здесь для иллюстрации возможностей MATLAB выбран один из наиболее эффективных методов – метод граничных элементов (МГЭ), позволяющий существенно упростить алгоритм решаемых задач [2]. Объединение возможностей MATLAB и МГЭ позволяет на качественно более высоком уровне представлять решения инженерных задач, что существенно повышает научное и практическое содержание учебного процесса. В этой связи, кроме основ языка программирования MATLAB, добавлены краткие сведения о МГЭ.

Слово MATLAB состоит из начальных букв слов MATrix LABoratory – матричная лаборатория. Название системы полностью отражает ее суть. Это действительно матричная лаборатория, где начальным кирпичиком является не простая переменная или константа, а матрица и ее частные случаи - вектор-строка, вектор-столбец.

Систему MATLAB разработал Молер (C.B. Moler) в 70-х г. г. XX века, которая использовалась на больших ЭВМ. В начале 80-х г. г. Джон Литл (John Little) из фирмы Math Works, Inc. Модернизировал эту систему для персональных компьютеров типа IBM PC, VAX и Macintosh. Далее к расширению системы были привлечены крупнейшие ученые и научные школы в математике, программировании и естествознании. Это позволило MATLAB стать признанным лидером в решении различных проблем науки и техники среди других подобных систем. Этому способствовало создание языка программирования, который вобрал в себя преимущества традиционных языков

(Fortran, Pascal, Basic, C++) и достаточно мощных средств визуализации и моделирования. Более подробно о преимуществах и возможностях системы MATLAB можно узнать в специализированных изданиях [1].

Данная работа содержит 4 главы. В первой главе представлены основные элементы языка программирования MATLAB и средств визуализации результатов расчетов. Вторая глава посвящена решению в системе MATLAB задач вычислительной математики и варианты заданий. Третья глава содержит подробные алгоритмы и программы решения задач статики, динамики и устойчивости упругих систем методом граничных элементов. Приведены так-же варианты заданий на самостоятельную работу. Четвертая глава посвящена пакетам расширений Symbolic Math., Optimization Toolbox u Simulink, позволяющих выполнять символьные вычисления, решать задачи оптимизации и моделировать динамические системы и устройства. Иллюстрация возможностей этих пакетов выполнена на задачах теоретической механики.

ГЛАВА 1. Основные элементы языка программирования и визуализации расчетов в системе MATLAB

1.1. Алфавит языка программирования

В MATLAB, как и в других системах, используются все буквы латинского алфавита от A до Z и арабские цифры от 0 до 9. Как и в C++, большие и малые буквы это разные переменные и константы. Кроме букв латинского алфавита используются все специальные символы клавиатуры компьютера.

1.2. Арифметические и логические операторы

Число *арифметических операторов* в MATLAB значительно расширено и включает в себя матричные и арифметические операции. В таблице 1.1 приводится список арифметических операторов.

Таблица 1.1

Функция	Обозначение (синтаксис)
Сложение	**+ (M1+M2)**
Вычитание	**- (M1 - M2)**
Матричное умножение	*** (M1*M2)**
Поэлементное умножение массивов	**.* (M1.*M2)**
Возведение матрицы в степень	Ù **(M1**Ù ***x*)**
Поэлементное возведение массива в степень	.Ù **(M1**.Ù ***x*)**
Деление матриц слева направо	**/ (M1 / M2)**
Поэлементное деление массивов слева направо	**./ (M1 ./ M2)**
Деление матриц справа налево	**\ (M1 \ M2)**
Поэлементное деление массивов справа налево	.\ (**M1** .\ **M2**)

В математических выражениях операторы имеют определенный приоритет исполнения. В MATLAB приоритет логических операторов выше, чем арифметических, приоритет возведения в степень выше приоритетов умно-

жения и деления, приоритет умножения и деления выше сложения и вычитания. Для повышения приоритета операций нужно использовать круглые скобки. Степень вложения скобок не ограничивается.

Операторы отношения служат для сравнения двух величин, векторов или матриц, все операторы отношения имеют две сравниваемые величины и записываются, как показано в таблице 1.2.

Таблица 1.2

Функция	Оператор (синтаксис)
Равно	**= = (*x* = = *y*)**
Не равно	**~ = (*x* ~ = *y*)**
Меньше	**< (*x* < *y*)**
Больше	**> (*x* > *y*)**
Меньше или равно	**< = (*x* < = *y*)**
Больше или равно	**> = (*x* > = *y*)**

Данные операторы выполняют поэлементное сравнение векторов или матриц одинакового размера и логическое выражение принимает значение 1 (True), если элементы идентичны, и значение 0 (False) в противном случае.

Логические операторы служат для реализации поэлементных логических операций над элементами одинаковых по размеру массивов согласно таблице 1.3.

Таблица 1.3

Функция	Оператор (синтаксис)
Логическое И	**&; *and* (*and* (*a, b*))**
Логическое ИЛИ	**ê; *or* (*or* (*a, b*))**
Логическое НЕ	**~ ; *not* (*not* (*a, b*))**
Исключающее ИЛИ	***xor* (*xor* (*a, b*))**
Верно, если все элементы вектора равны нулю	***any* (*any* (a))**
Верно, если все элементы вектора не равны нулю	***all* (*all* (*a*))**

1.3. Элементарные функции

Набор элементарных функций представим их описанием. В тригонометрических функциях углы измеряются в *радианах*.

Таблица 1.4

Функция	Синтаксис
$\lvert x \rvert$ – модуль	*abs*(*x*)
e^x – экспонента	*exp*(*x*)
$\ln x$ – натуральный логарифм	*log*(*x*)
$\log_2 x$ – логарифм по основанию 2	*log*2(*x*)
$\lg x$ – десятичный логарифм	*log*10(*x*)
2^x – 2 в степени *x*	*pow*(*x*)
$\sqrt{x}$ – квадратный корень	*sqrt*(*x*)
arccos x – арккосинус	*acos*(*x*)
arcctg x – арккотангенс	*acot*(*x*)
arccosec x – арккосеканс	*acsc*(*x*)
arcces x – арксеканс	*asec*(*x*)
arcsin x – арксинус	*asin*(*x*)
arctg x – арктангенс	*atan*(*x*)
cos x – косинус	*cos*(*x*)
ctg x – котангенс	*cot*(*x*)
sec x – секанс	*sec*(*x*)
cosec x – косеканс	*csc*(*x*)
sin x – синус	*sin*(*x*)
tg x – тангенс	*tan*(*x*)
arcch x – арккосинус гиперболический	*acosh*(*x*)
arcctg x – арккотангенс гиперболический	*acoth*(*x*)
arccosech x – арккосеканс гиперболический	*acsch*(*x*)
arcsech x – арксеканс гиперболический	*asech*(*x*)

Функция	Синтаксис
arcsh x – арккосинус гиперболический	*asinh*(*x*)
arctgh x – арктангенс гиперболический	*atanh*(*x*)
ch x – косинус гиперболический	*cosh*(*x*)
ctgh x – котангенс гиперболический	*coth*(*x*)
cosech x – косеканс гиперболический	*csch*(*x*)
sech x – секанс гиперболический	*sech*(*x*)
sh x – синус гиперболический	*sinh*(*x*)
tgh x – тангенс гиперболический	*tanh*(*x*)

Следует помнить, что все элементарные функции должны записываться в программах *малыми буквами*. Существуют также специальные математические функции, на которых мы не будем останавливаться. Их описание см. в [1].

1.4. Понятие о файлах-сценариях и файлах-функциях

При загрузке системы MATLAB на мониторе появляется основное окно системы, в котором можно выделить *окно команд* (Command Window). Система готова к проведению вычислений и созданию программ в командном режиме. Для этого можно на языке MATLAB записывать программы. Операторы заканчиваются символом **;** – точка с запятой. Одновременно точка с запятой *блокирует* вывод численного значения результата этого оператора в окне команд. В одной строке можно записать несколько операторов, а сами строки автоматически нумеруются при нажатии клавиши **Enter**. Если программа полностью записана и выходные величины не имеют символа **;**, то после нажатия клавиши **Enter** она выполняется. Ниже программы появляется ее результат. В таком режиме выполнять решения задач нецелесообразно, т.к. исправить возможные ошибки после нажатия клавиши **Enter** уже нельзя. Поэтому записывать программы, их редактировать и отлаживать необходимо в так называемых М-файлах. М-файл создается при выполнении команды

New меню **File**. Для ускорения этой команды выведена специальная пиктограмма в виде белой странички с загнутым уголком на панели инструментов. Щелкнув по пиктограмме стрелкой мышки, получаем окно М-файла, на котором можно записывать, редактировать и отлаживать любые программы решения научных и инженерных задач. Данный М-файл по умолчанию имеет название Untitled (Безымянный). Чтобы дать ему имя, необходимо в меню этого окна **File** выполнить команду **Save as** и в другом окне указать папку и имя этого файла. После указания имени и сохранения М-файла он готов для выполнения записанной программы. Для этого необходимо щелкнуть мышкой по пиктограмме **Выполнить**. Она выполнена в виде страницы со стрелкой, направленной вниз ↓ . Результат выполнения программы или сообщения об ошибках появится в окне команд. Описанный процесс называется созданием *М-файла сценария сессии*. Файл-сценарий, именуемый также *Script*-файлом, имеет весьма простую структуру:

% Основной комментарий, если необходимо.
% Дополнительный комментарий, если необходимо.
Тело программы с любыми выражениями.

Важными являются следующие свойства файлов-сценариев:

1. Они не имеют входных и выходных аргументов.
2. Работают с данными из рабочей области.
3. В процессе выполнения не компилируются.
4. Представляют собой *последовательность операций*, аналогичную той, что используется в сессии.

Кроме М-файла сценария, в MATLAB существует *М-файл функция*. Отличие М-файла функции от сценария состоит в том, что он является аналогом подпрограммы типа function в языке Pascal.

Структура М-файла функции с одним выходным параметром имеет вид:

function var = f _ name (Список параметров)

% Основной комментарий, если необходимо.

% Дополнительный комментарий, если необходимо.

Тело программы с любыми выражениями.

var = выражение

М-файл функция обладает такими свойствами:

1. Он начинается с ключевого слова function, после которого указывается имя переменной var – выходного параметра, имя самой функции f _ name и список ее входных параметров, отделенных запятой.

Внимание: Имя М-файла функции *должно совпадать* с самой f _ name (именем самой функции). MATLAB автоматически присваивает данное имя при выполнении команды **Save as**.

2. Результат выполнения М-файла функции присваивается имени функции, которое может использоваться в математических выражениях подобно функциям $sin(x)$, $log(x)$ и т. п.

3. Все переменные, используемые в файле-функции, являются локальными, т.е. действуют только в пределах тела функции.

4. Файл-функция является самостоятельным программным модулем, который связан с другими модулями и головной программой через входные и выходные параметры.

5. При обнаружении файла-функции он компилируется и затем исполняется.

Ниже в главах 2, 3 показаны примеры создания и использования в практических расчетах М-файлов сценариев и функций.

1.5. Основы программирования

1.5.1. *Оператор присваивания*

Программирование, т. е. создание определенного набора команд, в системе MATLAB является средством ее расширения и использовании в реше-

нии специфических проблем. Отдельные вопросы программирования изложены выше, здесь рассмотрим правила, дополняющие синтаксис языка MATLAB.

Программы оперируют с *переменными* и *константами*. Переменные – это имеющие имена объекты, способные хранить разные по значению данные. В зависимости от этих данных переменные могут быть *числовыми* или *символьными*, *векторными* или *матричными*.

Для задания переменным определенных значений используется *оператор присваивания*, вводимый знаком равенства =

Имя _ переменной = Выражение ;

Типы переменных заранее *не декларируются*. Они определяются выражением, значение которого присваивается переменной. Имя переменной может содержать сколько угодно символов, но идентифицируется только 31 начальный символ. Имя любой переменной должно быть *уникальным*. Имя должно начинаться *с буквы*, может содержать буквы, цифры и символ подчеркивания _ . Недопустимо включать в имена *пробелы* и *специальные знаки*.

1.5.2. *Перенос строки*

Если математическое выражение выходит за размер экрана монитора, то целесообразно перенести его часть на следующую строку. Для этого используется символ многоточие **…** – *три и более точки*. В командном режиме число возможных символов в одной строке – 4096, в М-файле – не ограничено, но с такими длинными строками работать неудобно. Поэтому применение в файлах-сценариях символа переноса строки улучшает наглядность программ.

1.5.3. *Ввод и вывод данных*

В языке MATLAB нет явных операторов ввода вывода данных. Эта проблема решается для ввода данных *оператором присваивания* и использованием *системных констант*. Вывод данных осуществляется еще проще.

Для этого необходимо после математического выражения *не ставить символ* **;** – точку с запятой. К системным константам относятся:

Pi = 3,1415 …	–	число “ПИ”;
i или ***j***	–	мнимые единицы;
NaN	–	неопределенность в виде $\frac{o}{o}$;
Inf	–	бесконечность типа a/o ;
ans		результат последней операции и др.

1.5.4. *Форматы чисел*

При вычислениях в MATLAB используется режим двойной точности. Однако, при выводе результатов, по умолчанию выдаются числа с 4 цифрами после десятичной точки в действительной форме. Чтобы изменить данную форму вывода, необходимо в программе перед выводимой величиной использовать команду **format name**, где **name** – имя формата. Для числовых данных **name** может быть следующим сообщением:

short – короткое представление в фиксированном формате (5 знаков);

short e – короткое представление в экспоненциальной форме (5 знаков мантиссы и 3 знака порядка);

long – длинное представление в фиксированном формате (15 знаков);

long e – длинное представление в экспоненциальной форме (15 знаков мантиссы и 3 знака порядка).

В качестве примера рассмотрим вывод вектора, содержащий 2 числа:

format name

x = [5/3 1.2783 e – 7]

В различных форматах вывод вектора x будет иметь следующий вид:

shor t	1.6667	0.0000
shor t e	1.6667E+000	1.2783E – 007
long	1.66666666666667	0.00000012783000
long e	1.66666666666667E+000	1.27830000000000E – 007

Задание формата сказывается только на форме вывода чисел. Вычисления же происходят в режиме двойной точности, а ввод чисел осуществляется в любом удобном виде.

1.5.5. *Формирование векторов и матриц*

Описанные правила вычислений распространяются и на более сложные вычисления, которые при использовании обычных языков программирования (типа Pascal, Fortran, C++ и др.) требуют составления специальных программ. MATLAB специально предназначен для проведения сложных вычислений с векторами и матрицами. При этом по умолчанию предполагается, что каждая переменная – это вектор или матрица. Например, если задано $x = 1$, то это значит, что x – это вектор с одним элементом, равным 1. Если надо задать вектор из трех элементов, то их значения надо перечислить в квадратных скобках, разделяя пробелами.

```
>>V     =   [1  2  3]
   V    =
                1   2   3
```

В данном случае задан вектор-строка. Если разделить элементы точкой с запятой, то получим вектор-столбец.

```
>>V     =   [1; 2; 3]
   V    =
                1
                2
                3
```

Задание матрицы требует указания несколько строк. Для разграничения строк используется символ **;** (точка с запятой).

```
>>T     =   [1 2 3; 4 5 6; 7 8 9]
   T    =
                1   2   3
                4   5   6
                7   8   9
```

Для указания отдельного элемента вектора или матрицы используются выражения вида V(*i*) или T(*i, j*). Например:

```
>>T (3,2)
    ans      =
                 8
```

Если элементу T(*i, j*) нужно присвоить новое значение *x*, то используют оператор присваивания

T (3,2) = *x*;

Выражение T(*i*) с одним индексом дает доступ к элементам матрицы, развернутым *в один столбец*. Такая матрица образуется из исходной, если подряд выписать ее столбцы. Например:

```
>>T (3)
    ans      =
                 7
>>T (8)
    ans      =
                 6
```

Наряду с операциями над отдельными элементами матриц и векторов MATLAB позволяет производить арифметические операции сразу над всеми элементами. Для этого перед знаком операции *ставится точка*.

Имеются также ряд особых функций для задания векторов и матриц. Отметим функции *ones* и *zeros*. Эти функции служат для создания одномерных и многомерных массивов. Функция *ones* создает массив с единичными элементами

```
>> a = ones (3, 2)
    a        =
                 1   1
                 1   1
                 1   1
```

Функция *zeros* создает массив с нулевыми элементами

```
>> b = zeros (2, 3)
    b        =
                 0   0   0
                 0   0   0
```

В краевых задачах поиска частот собственных колебаний и критических сил потери устойчивости упругих систем функция *zeros* находит широкое применение, что существенно упрощает тексты соответствующих программ (см. главу 3).

1.5.6. *Типы данных*

Структура типов данных MATLAB представлена схемой по рис. 1.1.

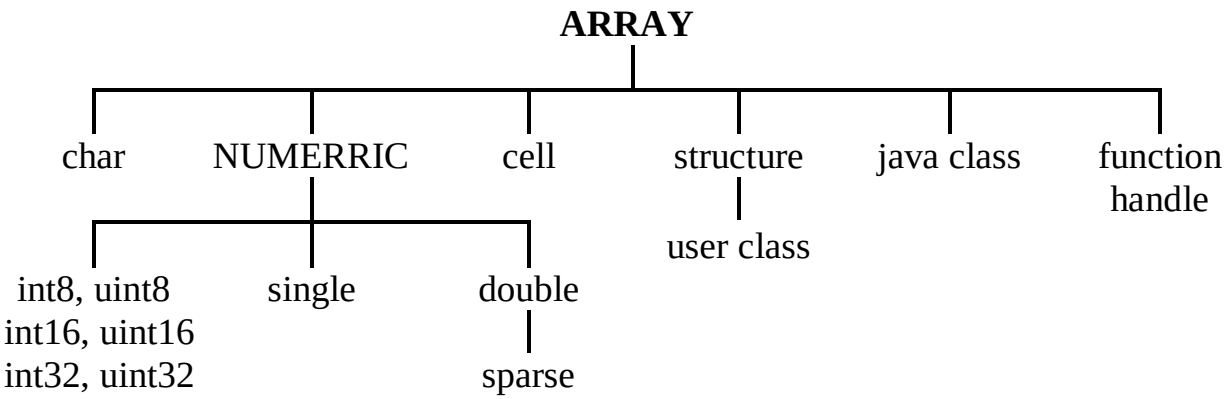


Рис. 1.1

Типы данных *array* и *numerric* являются *виртуальными* (“кажущимися”), т. к. к ним нельзя отнести какие-либо переменные. Они служат для определения и комплектования некоторых типов данных. Таким образом, в MATLAB определены типы данных, представляющие собой многомерные массивы:

1. *single* – числовые массивы с числами одинарной точности;

2. *double* – числовые массивы с числами удвоенной точности;

3. *char* – строчные массивы с элементами – символами;

4. *sparse* – разреженные матрицы с элементами – числами двойной точности;

5. *cell* – массивы ячеек;

6. *structure* – массивы структур с полями;

7. *function – handle* – дескрипторы функций;

8. *int8, …, uint32* – массивы 8-, 16-, 32-разрядных целых чисел со знаками и без знаков.

1.5.7. *Оператор двоеточие* :

Весьма часто необходимо выполнить формирование упорядоченных числовых последовательностей. Такие последовательности нужны для создания векторов или значений аргументов *x* при построении графиков. В MATLAB для этого используется оператор двоеточие **:**, который представляется следующим образом:

***x* = Начальное _ значение : Шаг : Конечное _ значение ;**

Эта конструкция создает возрастающую последовательность чисел, которая начинается с начального значения, изменяется на заданный шаг и завершается конечным значением. Если шаг не задан, то он принимает значение 1. Если конечное значение указано меньшим, чем начальное значение, – то выдается сообщение об ошибке. Примеры:

```
>> x      =   0 : 5
   x      =
                0   1   2   3   4   5
>> cos(x)
   ans    =
                1.0000   0.5403 – 0.4161 – 0.9900 – 0.6536   0.2837
>> x      =   1 :  –  0.2  :  0
   x      =
                1.0000   0.8000   0.6000   0.4000   0.2000   0
```

и т.д.

1.5.8. *Оператор разветвления if*

Условный оператор **if** в MATLAB записывается в общем виде так:

if* Логическое условие Оператор 1 *elseif* Логическое условие Оператор 2 *else* Оператор 3 *end ;

Эта конструкция имеет несколько частных вариантов:

if* Логическое условие Оператор 1 *end ;

if* Логическое условие Оператор 1 *else* Оператор 2 *end ;

Логическое условие записывается в виде:

Выражение 1 Оператор отношения Выражение 2

В качестве *операторов отношения* используются операторы: **= =, <, >, < =, > =, ~ =**. Если логическое условие принимает значение 1(*true* – истина), то выполняются соответствующие операторы. Если логическое условие принимает значение 0(*false* – ложь), то операторы, следующие за логическим условием, не выполняются. Оператор *end* указывает на конец условного оператора *if*. В понятие *Оператор 1* входят один или несколько операторов. В последнем случае они разделяются символами **,** (запятой) или **;** (точкой с запятой).

Как и в других алгоритмических языках, оператор *if* позволяет осуществить разветвление процесса вычислений в зависимости от какого-либо условия. Примеры применения этого оператора представлены в главе 3.

1.5.9. *Операторы циклов*

В MATLAB существует 3 типа операторов цикла. С оператором **:** (двоеточие) мы познакомились в п.1.5.7. Следующий оператор *for* … *end* используется для организации цикла с фиксированным числом повторений. Он имеет вид:

***for var* = Выражение Операторы *end* ;**

Здесь *var* – счетчик цикла – любая переменная, обычно это *i, j, k, l, m* и т. д. *Выражение* записывается в виде ***s : d : e***, где *s* – начальное значение счетчика цикла *var*, *d* – шаг изменения и *e* – конечное значение *var*. Возможна и запись в виде ***s : e***, тогда *d* = 1. Список операторов завершается ключевым словом *end*. Оператор *continue* передает управление в следующую итерацию цикла, пропуская операции, которые записаны за ним. Оператор *break* используется для досрочного прерывания цикла. Возможны вложенные циклы >> ***for i* = 1 : 3 *for j* = 1 : 3 *a* (*i, j*) = *i* * *j* ; *end* ; *end* ;**

В результате выполнения этого цикла формируется матрица *a*

```
>> a
   a      =
              1  2  3
              2  4  6
              3  6  9
```

Циклы типа *while … end* выполняются до тех пор, пока выполняется заданное условие. Оператор записывается в виде:

while **Логическое условие Операторы** ***end*** **;**

1.5.10. *Сообщения об ошибках и исправление ошибок*

Система MATLAB контролирует правильность написания программ и, при наличии ошибок, выдает соответствующее сообщение в окне команд. При этом указывается номер строки, где допущена ошибка, и характер ошибки. После уяснения сути ошибки ее необходимо исправить в тексте программы, запомнить М-файл командой **Save** и снова выполнить программу. Перед этим желательно очистить окно команд от сообщения об ошибках (чтобы не загромождать полученную картинку) с помощью команды **Clear Command Windows** (Очистить окно команд) в меню *Edit*.

1.5.11. *Вычисление определителя квадратной матрицы*

Для вычисления определителя квадратной матрицы используется функция *det*(*a*). Если матрица *a* содержит только целые числа, то результат – тоже целое число. Определитель вычисляется на основе треугольного разложения методом исключения Гаусса. Пример:

```
>> a      =  [2  3  6 ;  1  8  4;  3  6  7]
   a      =
              2  3  6
              1  8  4
              3  6  7
>> det(a)
   ans    =
              – 29
```

Данная функция широко используется в задачах поиска спектров частот собственных колебаний и критических сил потери устойчивости упругих систем (см. главу 3).

1.5.12. *Обращение матриц*

Обращение матриц – одна из наиболее распространенных операций задач строительной механики и других наук. *Обратной* называют матрицу, получаемую в результате деления единичной матрицы E на исходную матрицу x, т. е. $x^{-1} = E/x$. Эту процедуру выполняет функция $inv(x)$, которая вычисляет элементы обратной матрицы для исходной квадратной матрицы x. Выдается предупреждающее сообщение, если матрица x плохо масштабирована или близка к вырожденной. На практике вычисление обратной матрицы не так уж необходимо. Чаще обращение применяют для решения систем линейных алгебраических уравнений вида $ax = b$. Один из путей решения этой системы – $x = inv(a) * b$, хотя лучше использовать метод исключения Гаусса без формирования обратной матрицы, например $x = a/b$ или $x = b/a$.

1.6. Основы графической визуализации вычислений

Во многих областях науки и техники численное решение задач недостаточно для анализа результатов. Необходима еще графическая интерпретация в виде эпюр параметров напряженно-деформированного состояния элементов упругих систем, формы колебаний и потери устойчивости, поведение решений на заданном интервале и т. п. MATLAB позволяет решать эти задачи достаточно простыми процедурами. Вначале необходимо задать интервал изменения аргумента x от начального значения x_0 до конечного $x_к$ с шагом Δx, что осуществляется оператором двоеточие **:** x_0 **:** Δx **:** $x_к$. Далее используется команда построения графика какой-либо функции $y = f(x)$, которая носит имя *plot*.

Plot строит не истинный график функции $f(x)$, а лишь заданное числом элементов вектора x число точек. Эти точки затем соединяются отрезками

прямых, т. е. выполняется кусочно - линейная интерполяция данных графика. Если число точек достаточно велико, то полученная кривая воспринимается как вполне истинный график функции $y = f(x)$, при 10 – 20 точках получается ломаная кривая.

1.6.1. *Построение графиков отрезками прямых*

Для построения графика функции $y = f(x)$ необходимо задать совокупность точек x и y. Для аргумента x это выполняется оператором *двоеточие*, для y – *надлежащим программированием* выражения для функции, т. е. необходимо применить знаки арифметических операций над массивами:

: . * ; . / ; . ^{L}.

Для отображения таких функций используется декартовая прямоугольная система координат. Команда построения графика функции $y = f(x)$ *plot* имеет ряд параметров, которые рассмотрим ниже.

plot(x, y) – строит график функции $y = f(x)$, координаты точек (x, y) которой берутся из векторов *одинакового размера* x, y.

plot(x, y, s) – аналогична команде *plot*(x, y), но тип линии графика можно задавать с помощью строковой константы s, значения которой представлены в таблице 1.5.

Таблица 1.5

Цвет линии	Тип точки	Тип линии
s = y – желтый m – фиолетовый c – голубой r – красный g – зеленый b – синий w – белый k – черный	s = · – точка **O** – окружность ***x*** – крест **+** – плюс * – звездочка ***s*** – квадрат ***d*** – ромб Ú – треугольник (вниз) Ù – треугольник (вверх) < – треугольник (влево) > – треугольник (вправо) ***p*** – пятиугольник ***h*** – шестиугольник	s = - – сплошная **:** – двойной пунктир - **.** – штрихпунктир - - – штриховая

Таким образом, с помощью строковой константы *s* можно менять цвет линии, представлять узловые точки различными отметками и менять тип линии графика. Рассмотрим пример построения графиков трех функций $y_1 = sin\ x$; $y_2 = sin^2\ x$; $y_3 = sin^3\ x$ с различным стилем:

>> *x* = - 2 * *pi* : O . 01 * *pi* : 2 * *pi* ; *y*1 = *sin* (*x*) ; *y*2 = *sin* (*x*) . Ù 2 ; *y*3 = *sin* (*x*)) . Ù 3 ; >> *plot*(*x*, *y*1, ¢ - *m* ¢, *x*, *y*2, ¢ - . + *r* ¢, *x*, *y*3, ¢ - - ok ¢)

Здесь график функции *y*1 строится *сплошной фиолетовой линией*, график *y*2 строится *штрихпунктирной линией с точками в виде знака* “+” *красного цвета*, график *y*3 строится *штриховой линией с кружками черного цвета*.

1.6.2. *Графики дискретных отсчетов функции*

В сопротивлении материалов и строительной механике принято графики параметров напряженно-деформированного состояния закрашивать (штриховать). Для построения подобных графиков используется команда *stem*(…).

stem(*x, y*) – строит график с закрашенными частями между нулевой линией и кривой $y = f(x)$.

stem(… , ′ LINESPEC ′) – аналогична команде *stem*(*x, y*), но имеет спецификацию линий ′ LINESPEC ′ , подобную строковой константе *s* команды *plot*(*x, y, s*).

Stem(… , ′ *filled* ′) – строит график функции $y = f(x)$ с закрашенными маркерами.

1.6.3. *Создание массивов данных для трехмерной графики*

Трехмерные поверхности описываются функцией двух переменных $z = f(x, y)$. Построение трехмерных графиков требует определение для *x* и *y* двухмерных массивов – матриц. Для создания таких массивов служит функция *meshgrid*, которая записывается следующим образом:

[X, Y] = *meshgrid*(*x, y*) ;

В основном она используется совместно с функциями построения графиков трехмерных поверхностей. Функция преобразует область заданную векторами *x* и *y*, в массивы X и Y, которые могут быть использованы для вычисления функции двух переменных и построения трехмерных графиков. Строки выходного массива X являются копиями вектора *x*, а столбцы Y – копиями вектора *y*.

Пример:

```
>> [X Y]    =   mesh grid (4 : 7 , 9  :  13)
     X      =
                 4   5   6   7
                 4   5   6   7
                 4   5   6   7
                 4   5   6   7
                 4   5   6   7
     Y      =
                 9   9   9   9
                 10  10  10  10
                 11  11  11  11
                 12  12  12  12
                 13  13  13  13
```

Приведем еще пример применения функции *meshgrid*:

```
>> [X Y] = meshgrid (- 1 : 0.1  :  1,  - 1  : 0.1  :  1);
```

Такой вызов функции создает опорную плоскость для построения трехмерной поверхности при изменении *x* и *y* от – 1 до 1 с шагом 0.1.

1.6.4. *Построение графиков поверхностей*

Для построения графиков функции $z = f(x, y)$ используется команды *plot*3 (…), которая является аналогом команды *plot* (…). Она строит аксонометрическое изображение трехмерной поверхности и имеет следующие формы:

*plot*3 (*x*, *y*, *z*) – строит массив точек, представленных векторами *x*, *y*, *z* и соединяет их отрезками прямых.

*plot*3 (X, Y, Z), где X, Y, Z – три матрицы одинакового размера, строит точки с координатами X(*i*, :), Y(*i*, :) и Z(*i*, :) и соединяет их отрезками прямых. Пример построения графика трехмерной поверхности Z = $x^2 + y^2$:

```
>> [X Y]          =   meshgrid ([ - 3 : 0.15 : 3])
>>  Z             =   X. ^ 2 + Y. ^ 2 ;
>> plot 3(X, Y, Z)
```

*plot*3 (X, Y, Z, S) – обеспечивает построение графика поверхности, но со спецификацией стиля линий и точек, соответствующей спецификации команды *plot*.

1.6.5. *Включение и выключение масштабной сетки*

При построении графиков наряду с разметкой осей часто необходимо иметь масштабную сетку. Команды *grid* позволяют управлять этим процессом. Если после команды построения графиков добавить команду включения или выключения масштабной сетки, то можно получить график с требуемым видом:

grid on – добавляет сетку к текущему графику;

grid off – отключает сетку;

grid – последовательно производит включение и отключение сетки.

1.6.6. *Представление нескольких графиков в одном окне*

Иногда необходимо в одном окне поместить несколько графиков без наложения их друг на друга. Особенно удобно такое представление при построении эпюр напряженно-деформированного состояния элементов стержневых и пластинчатых систем. Для этого служит команда *subplot*, которую необходимо записать перед командой *plot*.

subplot (*m*, *n*, *p*) – разбивает графическое окно на *m* ´ *n* подокон, при этом *m* – число подокон по горизонтали, *n* – число окон по вертикали, *p* – номер подокна, в которое будет выводиться текущий график.

1.6.7. *Ввод текста на график с помощью мыши*

Для маркировки графиков можно ввести любой текст с помощью мыши командой *gtext*. Команда помещается после команды *plot*.

gtext (*′string* ⌀) – выводит на график перемещаемый мышкой маркер в виде крестика. Установив маркер в нужное место и щелкнув кнопкой мыши, получим текст на графике.

1.6.8. *Управление свойствами осей графиков*

Если не задавать масштаб графика, то он строится командой *plot* автоматически. Не всегда этот масштаб удовлетворяет пользователя. Команда *axis* позволяет установить любой масштаб.

axis ([*x min x max y min y max*]) – устанавливает нужный диапазон координат графика по осям *x* и *y*.

В заключение параграфа отметим, что более полно о возможностях графической визуализации MATLAB можно узнать в специализированных изданиях [1] и др.

ГЛАВА 2. MATLAB в задачах вычислительной математики

Система MATLAB, как следует из первой главы книги, обладает большими возможностями программирования и комплексной визуализации результатов инженерных расчетов и научных исследований. В этой связи покажем применение богатых возможностей MATLAB в решении задач вычислительной математики. Развитие многих наук привело исследователей к необходимости численного решения различных проблем, т. е. применению численных методов. Численные методы создаются и исследуются высококвалифицированными специалистами – математиками и, как в системе MATLAB, предлагаются для применения в виде готового инструмента. Поэтому задачей данной главы является изложение основных идей численных методов и ознакомление с последовательностью применения численных процессов в современных компьютерных системах. В таком виде книга окажет методическую помощь учащимся высших учебных заведений, необходимую при выполнении курсовых и дипломных работ, а также в научно-исследовательской работе.

Программы, реализующие какой-либо численный метод, необходимо записывать в М-файл. Если не дать имени М-файлу, то он запишется при выполнении программы в рабочую папку под именем Untitled (Безымянный). Такой ситуации следует избегать для исключения появления множества файлов с неопределенным именем. Рассмотрим решение различных проблем вычислительной математики, имеющих важное значение при изучении различных наук.

2.1. Табулирование функций

Данная задача широко используется в экологии, теплофизике и других дисциплинах. Обычно функции, описывающие какой-либо процесс, весьма громоздки и создание таблиц их значений требует большого объема вычислений.

Рассмотрим два случая табулирования функции:

1. С постоянным шагом изменения аргументов.

2. С произвольным набором значений аргумента.

Алгоритм реализуется путем организации какого-либо цикла.

Пример 1. Вычислить

$$y_i = R\sqrt[3]{\left|\ln\left(1-\lambda^{x_i}\right)^2 - x_i^3\right|},$$

при $R = 4.28 \cdot 10^{-2}$; $\lambda = 2.87$;

x_i изменяется с шагом $\Delta x = 2$; $x_n = 2$; $x_к = 10$.

Введем обозначение $\lambda \to la = 2.87$.

<u>Протокол программы:</u>

R = 4.28e-02; la = 2.87;

% Задается начальное значение x, шаг dx и конечное значение x

x = 2.0 : 2.0 : 10.0;

$$y = R_*\left(abs\left(\log\left(\left(1 - la.\hat{}\,x\right).\hat{}\,2\right) - x.\hat{}\,3\right)\right).\hat{}\left(1/3\right); \quad [x; y]$$

% Для вывода значения y в конце строки символ **;** *не ставится*!

В окне команд появляются после нажатия кнопки *выполнить* значения функции y, которые затем можно скопировать в какой-либо файл.

Результаты вычислений:

ans =

2.0000	4.0000	6.0000	8.0000	10.0000
0.0682	0.1634	0.2517	0.3386	0.4250

Пример 2. Вычислить и вывести на экран значения функции

$$y_i = \frac{1 + \sin^2\left(b^2 + x_i^2\right)}{\sqrt[3]{a^2 + x_i^2}};$$

при $x_1 = 12.8$; $x_2 = 23.4$; $x_3 = 27.2$; $x_4 = 17.8$; $x_5 = 16.3$; $x_6 = 14.9$; $a = 1.35$; $b = 0.98$.

Данную задачу можно программировать не изменяя обозначения переменных. Цикл организуется для одномерного массива.

<u>Протокол программы:</u>

a = 1.35; b = 0.98; x(1) = 12.8; x(2) = 23.4; x(3) = 27.2; x(4) = 17.8; x(5) = 16.3; x(6) = 14.9;

$$for \quad m = 1:6 \quad y = \left(1 + \sin\left(b\text{^}2 + x(m).\text{^}2\right).\text{^}2\right)/\left(a\text{^}2 + x(m.).\text{^}2\right).\text{^}\left(1/3\right), end;$$

% В конце строки вычисления функции *y* символ **;** *не ставится*.

y =

0.3609

y =

0.2327

y =

0.1473

y =

0.1800

y =

0.1771

y =

0.1658

Данные вычислений можно вывести в виде таблицы, если использовать запись [*x*; *y*] без точки с запятой или [x y].

Варианты заданий

Составить программу вычисления значений функции y_i для значений аргумента x_i. Данные взять из таблицы 2.1.

Таблица 2.1

№ п/п	Функция $y_i = f(x_i)$	Задача А					Задача В				
		a	**в**	x_H	x_K	Δx	x_1	x_2	x_3	x_4	x_5
1	$y = \dfrac{1 + \sin^2\left(b^3 + x^3\right)}{\sqrt[3]{b^3 + x^3}}$	**-**	**2.5**	**1.28**	**3.28**	**0.4**	**1.1**	**2.4**	**3.6**	**1.7**	**3.9**
2	$y = \dfrac{\sqrt[3]{ax + b}}{\lg^2 x}$	**1.35**	**0.98**	**1.14**	**4.24**	**0.62**	**0.35**	**1.28**	**3.51**	**5.21**	**4.16**
3	$y = \dfrac{1 + \lg^2 \dfrac{x}{a}}{b - e^{\frac{x}{a}}}$	**2.0**	**0.95**	**1.25**	**2.75**	**0.3**	**2.2**	**3.78**	**4.51**	**6.58**	**1.2**
4	$y = \sqrt[4]{\left\|x^2 - 2{,}5\right\|} + \sqrt[3]{\lg x^2}$	**-**	**-**	**1.25**	**3.25**	**0.4**	**1.84**	**2.71**	**3.81**	**4.56**	**5.62**
5	$\begin{cases} y = \dfrac{\lg^2\left(a^2 + x\right)}{(a + x)^2}, & x > 5 \\ \dfrac{(a + bx)^{2,5}}{1{,}8 + \cos^3(ax)}, & \geq 5 \end{cases}$	**- 2.5**	**3.4**	**3.5**	**6.5**	**0.6**	**2.89**	**3.54**	**5.21**	**6.28**	**3.48**
6	$y = 1{,}2^x - x^{1,2}, \quad \|x\| \geq 1$ $\arccos x, \quad \|x\| < 1$	**-**	**-**	**0.2**	**2.2**	**0.4**	**0.1**	**0.9**	**1.2**	**1.5**	**2.3**

№ п/п	Функция $y_i = f(x_i)$	Задача А					Задача В				
		а	**в**	x_H	x_K	Δx	x_1	x_2	x_3	x_4	x_5
7	$y=\dfrac{a^x-b^x}{\lg\dfrac{a}{b}}\sqrt[3]{ab}$	**0.4**	**0.8**	**3.2**	**6.2**	**0.6**	**4.48**	**3.56**	**2.78**	**5.28**	**3.21**
8	$y=\dfrac{b^3+\sin^2 ax}{\arccos(xbx)+e^{-x/2}}$	**1.2**	**0.48**	**0.7**	**2.2**	**0.3**	**0.25**	**0.36**	**0.56**	**0.94**	**1.28**
9	$y=\dfrac{\lg(x^2-1)}{\log_5(ax^2-b)}$	**1.1**	**0.09**	**1.2**	**2.2**	**0.2**	**1.21**	**1.76**	**2.53**	**3.48**	**4.52**
10	$y=\dfrac{\arccos(x^2-b^2)}{\arcsin(x^2-a^2)}$	**0.05**	**0.06**	**0.2**	**0.95**	**0.15**	**0.15**	**0.26**	**0.37**	**0.48**	**0.56**
11	$y=\arcsin(x^a)+\arccos(x^b)$	**2.0**	**3.0**	**0.11**	**0.36**	**0.05**	**0.08**	**0.26**	**0.35**	**0.41**	**0.53**
12	$y=a^{x^2-1}-\lg(x^2-1)+\sqrt[3]{x^2-1}$	**1.6**	**-**	**1.2**	**3.7**	**0.5**	**1.28**	**1.36**	**2.47**	**3.68**	**4.56**
13	$y=\dfrac{a\sqrt{x}-b\log_5 x}{\lg\lvert x-1\rvert}$	**4.1**	**2.7**	**1.2**	**5.2**	**0.8**	**1.9**	**2.15**	**2.34**	**2.73**	**3.16**
14	$y=\sqrt{\dfrac{\lvert a-bx\rvert}{\lg^3 x}}$	**7.2**	**4.2**	**1.81**	**5.31**	**0.7**	**2.4**	**2.8**	**3.9**	**4.7**	**3.16**
15	$y=(\arcsin^2 x+\arccos^4 x)^3$	**-**	**-**	**0.26**	**0.66**	**0.08**	**0.1**	**0.35**	**0.4**	**0.55**	**0.6**
16	$y=\dfrac{\ln_a\lvert b^2-x^2\rvert}{\sqrt[5]{\lvert x^2-a^2\rvert}}$	**2.0**	**1.1**	**0.08**	**1.08**	**0.2**	**0.1**	**0.3**	**0.4**	**0.45**	**0.65**
17	$y=\dfrac{a+tg^2 bx}{b+ctg^2 ax}$	**0.1**	**0.5**	**0.15**	**1.37**	**0.25**	**0.2**	**0.3**	**0.44**	**0.6**	**0.56**
18	$y=\dfrac{(a+bx)^{2,5}}{1+\lg(a+bx)}$	**2.5**	**4.6**	**1.1**	**3.6**	**0.5**	**1.2**	**1.28**	**1.36**	**1.46**	**2.35**
19	$y=\dfrac{\lg^2(a+x)}{(a+x)^2}$	**2.0**	**-**	**1.2**	**4.2**	**0.6**	**1.16**	**1.32**	**1.47**	**1.65**	**1.93**
20	$y=\dfrac{\sqrt[3]{(x-a)^2}+\sqrt[5]{\lvert x+b\rvert}}{\sqrt[9]{x^2-(a+b)^2}}$	**0.8**	**0.4**	**1.23**	**7.23**	**1.2**	**1.88**	**2.26**	**3.84**	**4.55**	**-6.21**
21	$y=(\sin^3 x+\cos^3 x)\ln x$	**-**	**-**	**0.11**	**0.36**	**0.05**	**0.2**	**0.3**	**0.38**	**0.43**	**0.57**
22	$y=a^{x^2-1}-\lg(x^2-1)+\sqrt[3]{x^2-1}$	**2.25**	**-**	**1.2**	**2.7**	**0.3**	**1.31**	**1.39**	**1.44**	**1.56**	**1.92**
23	$y=\dfrac{a\sqrt[3]{x}-b\log_5 x}{\lg^3(x-1)}$	**4.1**	**2.7**	**1.5**	**3.5**	**0.4**	**1.9**	**2.15**	**2.34**	**2.74**	**3.16**
24	$y=\sqrt[5]{\dfrac{a+bx}{\lg^3 x}}$	**7.2**	**1.3**	**1.56**	**4.71**	**0.63**	**2.4**	**2.8**	**3.9**	**4.7**	**3.16**
25	$y=\sqrt[7]{\arcsin^4 x+\arccos^6 x}$	**-**	**-**	**0.22**	**0.92**	**0.14**	**0.1**	**0.35**	**0.4**	**0.55**	**0.6**

№ п/п	Функция $y_i = f(x_i)$	Задача А					Задача В				
		а	**в**	x_H	x_K	Δx	x_1	x_2	x_3	x_4	x_5
26	$y = \dfrac{\log_a\left(b^2 - x^2\right)}{\sqrt[3]{\left\|x^2 - a^2\right\|}}$	**2.0**	**4.1**	**0.77**	**1.77**	**0.2**	**1.24**	**1.38**	**2.38**	**3.21**	**0.68**
27	$y = \dfrac{\sqrt[3]{a} + tg^{4.5}bx}{\sqrt[5]{b} + ctg^{2.7}ax}$	**0.1**	**0.5**	**0.33**	**1.23**	**0.18**	**0.5**	**0.36**	**0.40**	**0.62**	**0.78**
28	$y = \dfrac{\sin(a + bx)^{3.5}}{1 + \cos[\lg(a + bx)]}$	**2.5**	**4.6**	**1.15**	**3.05**	**0.38**	**1.2**	**1.36**	**1.57**	**1.93**	**2.25**
29	$y = \dfrac{tg\left[\lg^3(a + x)\right]}{\sqrt[7]{(a + x)^2}}$	**2.0**	**-**	**1.08**	**1.88**	**0.16**	**1.16**	**1.35**	**1.48**	**1.52**	**1.96**
30	$y = \dfrac{\sqrt[3]{x - a} + \sqrt[5]{x + b}}{\sqrt[7]{x} - \sqrt[9]{x^2 - (a + b)^2}}$	**0.8**	**0.4**	**1.42**	**3.62**	**0.44**	**1.6**	**1.81**	**2.24**	**2.65**	**3.38**

2.2. Решение систем линейных алгебраических уравнений методом исключения Гаусса

К решению систем линейных уравнений сводятся многочисленные практические задачи, например различные краевые задачи для обыкновенных и в частных производных дифференциальных уравнений. Можно с полным основанием утверждать, что данная проблема является одной из самых распространенных и важных задач вычислительной математики.

Пусть задана система n линейных алгебраических уравнений с n неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = в_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = в_2 \\ ---------- \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = в_n \end{cases}. \qquad (2.1)$$

Система уравнений (2.1) в матричной форме представляется следующим образом:

$$\mathbf{AX} = \mathbf{B}, \qquad (2.2)$$

где **A** – квадратная матрица коэффициентов, размером $n \times n$ строк и столбцов;

X – вектор-столбец неизвестных;

B – вектор-столбец правых частей.

Систему уравнений (2.2) можно решить различными методами. Один из наиболее простых и эффективных методов является метод исключения Гаусса и его модификации. Алгоритм метода основан на приведении матрицы **A** к треугольному виду (прямой ход) и последовательном вычислении неизвестных (обратный ход). Эти процедуры можно выполнять над невыраженными матрицами, в противном случае метод Гаусса неприменим.

Недостатком метода является накапливание погрешностей в процессе округления, поэтому метод Гаусса без выбора главных элементов используется обычно для решения сравнительно небольших ($n \leq 100$) систем уравнений с плотно заполненной матрицей и не близким к нулю определителем. Если матрица **A** сильно разрежена, а ее определитель не близок к нулю, то метод Гаусса пригоден для решения больших систем уравнений. В MATLAB имеется обширный арсенал методов решения систем уравнений (2.2) методом исключения Гаусса. Для этого применяются следующие операторы

| / | – правое деление;

| \ | – левое деление;

| ∧ - 1 | – возведение в степень –1;

| inv(A) | – обращение матрицы **A**.

Выражения

```
X = B/A
X = B* A^ – 1
X = B* inv(A)
X = A\B
```

дают решения ряда систем линейных уравнений **AX = B**, где **A** – матрица размером $m \times n$, **B** – матрица размером $n \times к$. Более сложные случаи решения систем уравнений (2.2) с плохо обусловленной матрицей **A** освещены в работе [1].

Пример 3.

Решить систему 4-х линейных уравнений:

$$\begin{cases} 1.1161x_1 + 0.1397x_2 + 0.1254x_3 + 0.1490x_4 = 1.5471; \\ 0.1582x_1 + 0.1768x_2 + 1.1675x_3 + 0.1871x_4 = 1.6471; \\ 0.1968x_1 + 1.2168x_2 + 0.2071x_3 + 0.2271x_4 = 1.7471; \\ 0.2368x_1 + 0.2568x_2 + 0.2471x_3 + 1.2671x_4 = 1.8471. \end{cases}$$

<u>Протокол программы (в М-файле)</u>

```
a   =  [1.1161   0.1397   0.1254   0.1490 ;
        0.1582   0.1768   1.1675   0.1871 ;
        0.1968   1.2168   0.2071   0.2271 ;
        0.2368   0.2568   0.2471   1.2671] ;
b   =  [1.5471 ; 1.6471 ; 1.7471 ; 1.8471] ;
```

X4 = *a* \ *b*

Эта программа выдает решение заданной системы с помощью четвертого оператора в виде матрицы – столбца

```
X4   =
        1.0406
        0.9351
        0.9870
        0.8813
```

<u>Внимание</u>. В М-файле матрица ***a*** *набирается по строкам,* а *элементы* матрицы правых частей ***b*** *отделяются символом* **;** , т. е. *тоже набираются по строкам*. Решение другими операторами системы уравнений (2.2) требует набора матрицы ***a*** *по столбцам*, а *элементы* правых частей ***b*** *отделяются только пробелом*!

```
a   =  [1.1161   0.1582   0.1968   0.2368 ;
        0.1397   0.1768   1.2168   0.2568 ;
        0.1254   1.1675   0.2071   0.2471 ;
        0.1490   0.1871   0.2271   1.2671] ;
b   =  [1.5471   1.6471   1.7471   1.8471] ;
```

X1 = *b*/*a*

X2 = *b** *a* ^ – 1

X3 = *b** *inv*(*a*)

Результаты решения

```
X1    =
          1.0406   0.9351   0.9870   0.8813
X2    =
          1.0406   0.9351   0.9870   0.8813
X3    =
          1.0406   0.9351   0.9870   0.8813
```

Варианты заданий. Решить систему линейных алгебраических уравнений с помощью 4-х операторов. Данные взять из таблицы 2.2.

Таблица 2.2

1	$a_{ij}=\begin{bmatrix}2 & -4 & 3 & 1\\ -1 & 5 & -7 & -3\\ 10 & -2 & 4 & 4\\ -1 & 1 & -1 & -1\end{bmatrix};\quad b_i=\begin{bmatrix}7\\ -24\\ 34\\ -6\end{bmatrix}$	**2**	$a_{ij}=\begin{bmatrix}5 & -3 & 4 & -2\\ 10 & 3 & -4 & 2\\ 7 & -5 & 8 & -10\\ 4 & 5 & -8 & 10\end{bmatrix};\quad b_i=\begin{bmatrix}4\\ 11\\ 0\\ 11\end{bmatrix}$
3	$a_{ij}=\begin{bmatrix}2 & 2 & -1 & 1\\ 4 & 3 & -1 & 2\\ 8 & 5 & -3 & 4\\ 3 & 3 & -2 & 2\end{bmatrix};\quad b_i=\begin{bmatrix}4\\ 6\\ 12\\ 6\end{bmatrix}$	**4**	$a_{ij}=\begin{bmatrix}1 & 1 & -6 & 4\\ 3 & -1 & -6 & -4\\ 2 & 3 & 9 & 2\\ 3 & 2 & 3 & 8\end{bmatrix};\quad b_i=\begin{bmatrix}6\\ 2\\ 6\\ -7\end{bmatrix}$
5	$a_{ij}=\begin{bmatrix}1 & 5 & 3 & 4\\ 7 & 14 & 20 & 27\\ 5 & 10 & 16 & 19\\ 3 & 5 & 6 & 13\end{bmatrix};\quad b_i=\begin{bmatrix}0\\ 0\\ -2\\ 5\end{bmatrix}$	**6**	$a_{ij}=\begin{bmatrix}2 & 4 & 3 & 4\\ 4 & -2 & 5 & 6\\ 6 & -3 & 7 & 8\\ 8 & -4 & 9 & 10\end{bmatrix};\quad b_i=\begin{bmatrix}5\\ 7\\ 9\\ 11\end{bmatrix}$
7	$a_{ij}=\begin{bmatrix}1 & 1 & 1 & 1\\ 2 & 3 & 1 & 2\\ -1 & 2 & 4 & 1\\ 2 & -1 & 1 & -1\end{bmatrix};\quad b_i=\begin{bmatrix}1\\ 1\\ 18\\ -12\end{bmatrix}$	**8**	$a_{ij}=\begin{bmatrix}5 & -3 & 2 & 4\\ 4 & -2 & 3 & 7\\ 8 & -6 & -1 & -5\\ 7 & -3 & 7 & 17\end{bmatrix};\quad b_i=\begin{bmatrix}7\\ 10\\ 4\\ 17\end{bmatrix}$
9	$a_{ij}=\begin{bmatrix}1 & 2 & 15 & 11\\ -2 & 4 & 3 & 61\\ -3 & -8 & 11 & 12\\ 15 & 7 & 8 & -4\end{bmatrix};\quad b_i=\begin{bmatrix}2\\ 4\\ 5\\ 9\end{bmatrix}$	**10**	$a_{ij}=\begin{bmatrix}5 & 3 & -4 & -15\\ 6 & 1 & 10 & 2\\ 1 & 5 & 3 & 8\\ -4 & 5 & -1 & 9\end{bmatrix};\quad b_i=\begin{bmatrix}6\\ 9\\ 12\\ -9\end{bmatrix}$
11	$a_{ij}=\begin{bmatrix}7 & -8 & 9 & 11\\ -5 & 6 & 5 & 3\\ -4 & -7 & 3 & 6\\ 2 & 1 & 5 & 9\end{bmatrix};\quad b_i=\begin{bmatrix}3\\ -4\\ -6\\ 11\end{bmatrix}$	**12**	$a_{ij}=\begin{bmatrix}1 & 2 & 5 & -11\\ 2 & -6 & -3 & 8\\ 3 & 4 & 5 & 1\\ 7 & 6 & 2 & 20\end{bmatrix};\quad b_i=\begin{bmatrix}7\\ 11\\ 15\\ -9\end{bmatrix}$

13	$a_{ij}=\begin{bmatrix}5 & 3 & 3 & -4\\ -6 & -4 & 2 & 17\\ 12 & 26 & 1 & 5\\ 5 & 8 & 9 & 7\end{bmatrix};\quad b_i=\begin{bmatrix}7\\ -24\\ 34\\ -6\end{bmatrix}$	**14**	$a_{ij}=\begin{bmatrix}4 & 7 & 6 & -19\\ 5 & 9 & 12 & 3\\ -4 & -3 & 1 & -8\\ 2 & 5 & 9 & 10\end{bmatrix};\quad b_i=\begin{bmatrix}10\\ 12\\ -14\\ 22\end{bmatrix}$
15	$a_{ij}=\begin{bmatrix}12 & 8 & 24 & -5\\ 16 & 25 & 9 & 9\\ 2 & 7 & 3 & 4\\ 21 & -9 & -7 & 6\end{bmatrix};\quad b_i=\begin{bmatrix}5\\ -17\\ 22\\ 3\end{bmatrix}$	**16**	$a_{ij}=\begin{bmatrix}-3 & -7 & 16 & 8\\ -4 & 6 & 12 & 13\\ 2 & 3 & 7 & 5\\ 18 & -14 & 1 & 2\end{bmatrix};\quad b_i=\begin{bmatrix}6\\ -12\\ 31\\ 41\end{bmatrix}$
17	$a_{ij}=\begin{bmatrix}2 & 5 & 9 & 16\\ 11 & 16 & 3 & 1\\ 25 & 14 & -38 & 7\\ -8 & 9 & -16 & -42\end{bmatrix};\quad b_i=\begin{bmatrix}1\\ 1\\ -3\\ 6\end{bmatrix}$	**18**	$a_{ij}=\begin{bmatrix}2 & -1 & 3 & 2\\ 3 & 3 & 3 & 2\\ 3 & -1 & -1 & -2\\ 3 & -1 & 3 & -1\end{bmatrix};\quad b_i=\begin{bmatrix}4\\ 6\\ 6\\ 6\end{bmatrix}$
19	$a_{ij}=\begin{bmatrix}2 & 2 & -1 & 1\\ 4 & 3 & -1 & 2\\ 8 & 5 & -3 & 4\\ 3 & 3 & -2 & 2\end{bmatrix};\quad b_i=\begin{bmatrix}4\\ 6\\ 12\\ 6\end{bmatrix}$	**20**	$a_{ij}=\begin{bmatrix}2 & 3 & 11 & 5\\ 1 & 1 & 5 & 2\\ 2 & 1 & 3 & 2\\ 1 & 1 & 3 & 4\end{bmatrix};\quad b_i=\begin{bmatrix}2\\ 1\\ -3\\ -3\end{bmatrix}$
21	$a_{ij}=\begin{bmatrix}2 & 5 & 4 & 1\\ 1 & 3 & 2 & 1\\ 2 & 10 & 9 & 9\\ 3 & 8 & 2 & 2\end{bmatrix};\quad b_i=\begin{bmatrix}20\\ 11\\ 40\\ 37\end{bmatrix}$	**22**	$a_{ij}=\begin{bmatrix}3 & 4 & 1 & 2\\ 3 & 5 & 3 & 5\\ 6 & 8 & 1 & 5\\ 3 & 5 & 3 & 7\end{bmatrix};\quad b_i=\begin{bmatrix}-3\\ -6\\ -8\\ -8\end{bmatrix}$
23	$a_{ij}=\begin{bmatrix}3.2 & 5.4 & 4.2 & 2.2\\ 2.1 & 3.2 & 3.1 & 1.1\\ 1.2 & 0.4 & -0.8 & -0.8\\ 4.7 & 10.4 & 9.7 & 9.7\end{bmatrix};\quad b_i=\begin{bmatrix}2.6\\ 4.8\\ 3.6\\ -8.4\end{bmatrix}$	**24**	$a_{ij}=\begin{bmatrix}3 & -2 & -5 & 1\\ 2 & -3 & 1 & 5\\ 1 & 2 & 0 & -4\\ 1 & -1 & -4 & 9\end{bmatrix};\quad b_i=\begin{bmatrix}3\\ -3\\ -3\\ 22\end{bmatrix}$
25	$a_{ij}=\begin{bmatrix}1 & 1 & -6 & -4\\ 3 & -1 & -6 & -4\\ 2 & 3 & 9 & 2\\ 3 & 2 & 3 & 8\end{bmatrix};\quad b_i=\begin{bmatrix}6\\ 2\\ 6\\ -7\end{bmatrix}$	**26**	$a_{ij}=\begin{bmatrix}2 & -1 & 1 & 2\\ 6 & -4 & 2 & 4\\ 6 & -3 & 4 & 8\\ 4 & -2 & 1 & 1\end{bmatrix};\quad b_i=\begin{bmatrix}2\\ 3\\ 9\\ 1\end{bmatrix}$
27	$a_{ij}=\begin{bmatrix}12 & 14 & -15 & 24\\ 16 & 18 & -22 & 29\\ 18 & 20 & -21 & 32\\ 10 & 12 & -16 & 20\end{bmatrix};\quad b_i=\begin{bmatrix}5\\ 8\\ 9\\ 4\end{bmatrix}$	**28**	$a_{ij}=\begin{bmatrix}24 & 14 & 30 & 40\\ 36 & 25 & 45 & 61\\ 48 & 28 & 60 & 82\\ 60 & 35 & 75 & 99\end{bmatrix};\quad b_i=\begin{bmatrix}28\\ 43\\ 58\\ 69\end{bmatrix}$
29	$a_{ij}=\begin{bmatrix}5 & -3 & 2 & 4\\ 4 & -2 & 3 & 7\\ 8 & -6 & -1 & -5\\ 7 & -3 & 7 & 17\end{bmatrix};\quad b_i=\begin{bmatrix}3\\ 1\\ 9\\ 15\end{bmatrix}$	**30**	$a_{ij}=\begin{bmatrix}2 & -1 & 3 & 4\\ 4 & -8 & 5 & 6\\ 6 & -5 & 7 & 8\\ 12 & -4 & -9 & 10\end{bmatrix};\quad b_i=\begin{bmatrix}5\\ 7\\ 9\\ 13\end{bmatrix}$

2.3. Аппроксимация функций

Одним из распространенных и практически важных случаев связи между аргументом и функцией является задание этой связи в виде некоторой таблицы $\{x_i\ ;\ y_i\}$, например, экспериментальные данные. На практике часто приходится использовать табличные данные для приближенного вычисления y при любом значении аргумента x (из некоторой области). Этой цели служит задача о приближении (аппроксимации) функций: данную функцию $f(x)$ требуется приближенно заменить некоторой функцией $g(x)$ так, чтобы отклонение $g(x)$ от $f(x)$ в заданной области было наименьшим. Функция $g(x)$ при этом называется аппроксимирующей. Если приближение строится на заданном дискретном множестве точек $\{x_i\}$, то аппроксимация называется точечной. К ней относятся интерполирование, среднеквадратичное приближение и др. При построении приближения на непрерывном множестве точек (например, на отрезке $[a,\ b]$) аппроксимация называется непрерывной или интегральной. MATLAB имеет мощные средства точечной и непрерывной аппроксимации с визуализацией результата. Рассмотрим наиболее важную точечную аппроксимацию (обработка экспериментальных данных).

Пример 4. Используя линейную и полиномиальную аппроксимации, получить эмпирические формулы для функции $y=f(x)$, заданной в табличном виде:

x_i	0.75	1.50	2.25	3.00	3.75
y_i	2.50	1.20	1.12	2.25	4.28

Оценить погрешность эмпирических формул.

<u>Протокол программы</u>. В окне команд набираются значения x_i и y_i. Далее выполняется команда построения графика только узловых точек.

```
>> x = [0.75, 1.50, 2.25, 3.00, 3.75] ;
>> y = [2.50, 1.20, 1.12, 2.25, 4.28] ;
>> plot (x, y, ' 0 ') ;
```

Появляется окно с символами ′ 0 ′ на месте узловых точек (рис. 2.1).

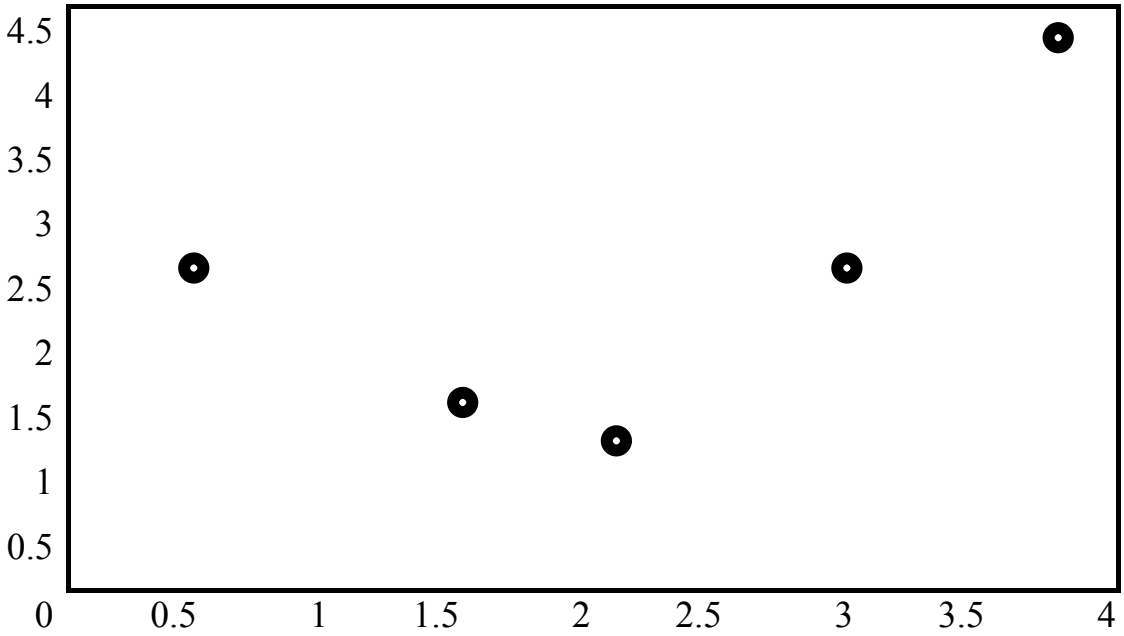


Рис. 2.1

Внимание. Следует помнить, что при полиномиальной аппроксимации максимальная степень полиноме на 1 меньше числа экспериментальных точек!

На панели инструментов окна графика узловых точек в меню **Tools** исполняем команду **Basic Fitting**. Появляется окно **Основной Монтаж**. В этом окне птичкой отмечаются необходимые данные аппроксимации. В частности, можно задать следующие операции:

- *показать уравнение* аппроксимирующей функции $y = g(x)$;
- *выбрать метод подбора*: сплайн интерполяции
  эрмитовская интерполяция
  линейный
  квадратные
  кубический
  …….

В нашей задаче выбираем линейную и полиномиальную аппроксимации. В окне графика появляются соответствующие графики разноцветом и формулы аппроксимирующих функций (рис. 2.2).

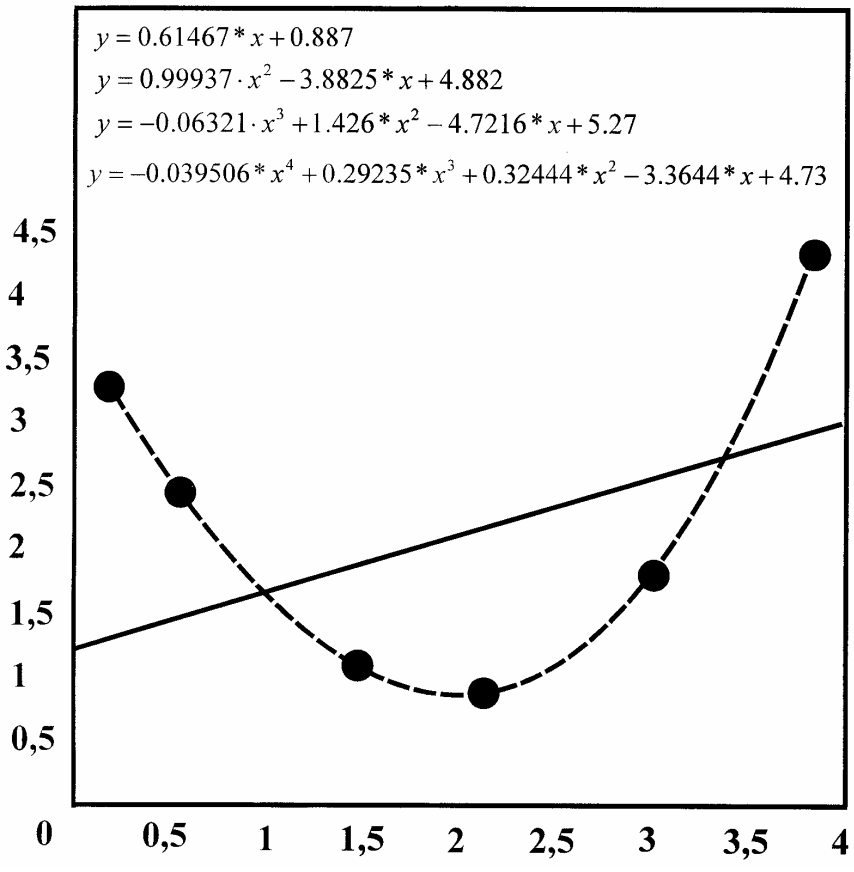


Рис. 2.2

Чтобы узнать погрешность аппроксимации, надо *отметить птичкой* параметр ***График остатка*** в окне *Основной Монтаж,* и *Показать норму остатков*. График погрешностей с нормами можно вынести в *отдельное окно*, или вместе с графиком и аппроксимирующих функций – суб-график. Норма погрешностей указывает на статистическую оценку среднеквадратической погрешности. Чем она меньше, тем точнее полученная аппроксимирующая функция $y = g(x)$. В нашем примере:

Linear : norm of residuals (норма погрешности) = **2.1061**

Quadratic : norm of residuals = **0.10736**

Cubic : norm of residuals = **0.035857**

4 ***th degree : norm of residuals*** = **9.6305*e*-015.**

График погрешностей можно выводить в виде диаграмм (зоны), линий (линии) или отдельных точек (фрагменты). Сам график погрешностей представляет собой зависимость разности $g(x) - f(x)$ в условных точках, соединенных прямыми линиями.

Кроме линейной и полиномиальной аппроксимации можно выбрать *сплайн- аппроксимацию* – когда на каждом интервале приближения используется *кубический полином* с новыми коэффициентами. В этом случае нельзя получить выражение для аппроксимирующей функции, т. е. такая аппроксимация является неполной. Аналогичными свойствами обладает и *Эрмитовая аппроксимация*. Она имеет только графическую интерпретацию.

Варианты заданий. Получить эмпирические формулы и оценить их погрешность для функции $y = f(x)$, заданной таблично. Данные взять из таблицы 2.3.

Таблица 2.3

1.	x_i	-3	-2	-1	0	1	2	3
	y_i	-0.71	-0.01	0.51	0.82	0.88	0.51	0.49
2.	x_i	-6.6	-5.38	-3.25	-1.76	2.21	3.6	4.5
	y_i	2.89	1.41	0.29	-0.41	-0.69	-0.7	1.2
3.	x_i	0	1	2	3	4	5	6
	y_i	-0.31	0.9	2.11	3.3	4.51	5.73	6.93
4.	x_i	-2	-1	0	1	2	3	4
	y_i	7.1	3.9	1.1	0.8	3.1	4.5	5.3
5.	x_i	-2	-1	-0.5	0	1.5	2	3.5
	y_i	5.9	2.8	2.1	3.2	6.1	7.6	4.3
6.	x_i	-3	-2	-1	0	1	2	3
	y_i	3.1	0.9	0.9	2.8	7.1	6.5	4.1
7.	x_i	0	1	2	3	4	5	6
	y_i	10.0	7.5	5.5	4.0	3.0	2.0	2.24
8.	x_i	-2	-1	0	1.5	2.3	2.6	2.9
	y_i	4.2	5.6	6.8	7.2	9.4	10.5	11.8
9.	x_i	10.0	12.0	13.0	15.0	18.0	20.0	21.0
	y_i	0.66	0.89	1.24	1.36	1.56	1.76	1.92
10.	x_i	22.0	24.0	27.0	30.0	31.0	35.0	40.0
	y_i	1.24	1.37	1.46	1.26	1.66	1.84	1.99
11.	x_i	-7.0	-6.0	-5.0	-4.0	-3.0	-2.0	-1.0
	y_i	22.6	24.7	25.6	24.6	23.5	21.8	19.3

12.	x_i	-25.0	-23.0	-21.0	-18.0	-17.2	-15.4	-14.0
	y_i	0.76	0.74	0.61	0.58	0.84	0.92	1.22
13.	x_i	-4.0	-3.0	-2.0	-1.0	0.0	1.0	2.0
	y_i	1.71	1.56	1.24	1.36	1.78	2.21	4.31
14.	x_i	-22.0	-20.0	-18.0	-16.0	-14.0	-12.0	-10.0
	y_i	-2.26	-1.84	-1.92	-1.76	-1.56	-1.64	-1.34
15.	x_i	23.0	24.0	25.0	26.0	27.0	28.0	29.0
	y_i	1.26	1.37	1.44	1.56	1.15	1.28	1.06
16.	x_i	30.0	33.0	35.0	37.0	39.0	41.0	43.0
	y_i	-2.6	-3.7	-2.5	-4.3	-2.3	-5.6	-1.9
17.	x_i	44.0	45.0	46.0	47.0	48.0	49.0	50.0
	y_i	2.24	3.46	5.36	1.89	1.76	1.54	2.12
18.	x_i	52.0	54.0	56.0	58.0	60.0	62.0	64.0
	y_i	-1.28	-1.33	-1.44	-1.67	-1.77	-2.81	-2.16
19.	x_i	2.2	2.6	3.0	3.4	3.8	4.2	4.6
	y_i	1.88	1.65	1.61	1.73	1.56	1.24	1.99
20.	x_i	5.1	5.3	5.5	5.7	5.9	6.1	6.3
	y_i	-2.8	-3.6	-5.7	-3.4	-1.9	-1.7	-1.5
21.	x_i	7.15	7.35	7.55	7.75	7.95	8.15	8.35
	y_i	-2.2	-3.6	-1.7	-2.8	-1.6	-4.5	-2.2
22.	x_i	9.1	9.2	9.3	9.4	9.5	9.6	9.7
	y_i	1.48	1.16	2.08	1.96	1.81	2.31	5.61
23.	x_i	-10.2	-10.1	-10.0	-9.9	-9.8	-9.7	-9.6
	y_i	-6.5	-7.8	-10.2	-5.4	-4.6	-9.5	-10.3
24.	x_i	11.0	14.0	17.0	20.0	23.0	26.0	29.0
	y_i	1.2	1.6	1.9	1.1	1.16	1.24	1.36
25.	x_i	-50.0	-48.0	-46.0	-44.0	-42.0	-40.0	-38.0
	y_i	1.23	1.32	1.57	1.19	1.16	1.10	2.28
26.	x_i	-36.0	-34.0	-32.0	-30.0	-28.0	-26.0	-24.0
	y_i	1.1	1.3	2.1	1.9	1.7	1.5	1.8
27.	x_i	21.0	23.0	24.0	28.0	31.0	32.0	36.0
	y_i	1.24	1.37	1.56	1.64	1.84	1.26	1.14
28.	x_i	10.0	13.0	17.0	22.0	28.0	35.0	43.0
	y_i	1.21	1.36	1.51	1.84	1.06	1.21	1.36
29.	x_i	-1.0	0.0	3.0	5.0	8.0	12.0	15.0
	y_i	-2.1	-3.6	1.2	-4.3	1.8	2.6	-0.2
30.	x_i	-8.0	-7.0	-5.0	-3.0	-1.0	2.0	5.0
	y_i	1.36	1.88	2.45	-2.1	-10.2	-4.4	1.16

2.4. Численное решение обыкновенных дифференциальных уравнений

Многие задачи физики, химии, экологии, механики и других разделов науки и техники при их математическом моделировании сводятся к дифференциальным уравнениям. Поэтому решение дифференциальных уравнений является одной из важнейших математических задач. В вычислительной математике изучаются численные методы решения дифференциальных уравнений, которые особенно эффективны в сочетании с использованием персональных компьютеров.

Среди множества численных методов решения дифференциальных уравнений наиболее простые – это явные одношаговые методы. К ним относятся различные модификации метода Рунге-Кутта.

Постановка задачи:

Требуется найти функцию $y = y(x)$, удовлетворяющую уравнению

$$\frac{dy}{dx} = F(x, y) \tag{2.3}$$

и принимающую при $x = x_0$ заданное значение y_0:

$$y(x_0) = y_0 . \tag{2.4}$$

При этом решение необходимо получить в интервале $x_0 \le x \le x_к$. Из теории дифференциальных уравнений известно, что решение $y(x)$ задачи Коши (2.3), (2.4) существует, единственно и является гладкой функцией, если правая часть $F(x, y)$ удовлетворяет некоторым условиям гладкости. Численное решение задачи Коши методом Рунге-Кутта 4-го порядка заключается в следующем. На заданном интервале $[x_0, x_к]$ выбираются узловые точки. Значение решения в нулевой точке известно $y(x_0) = y_0$. В следующей точке $y(x_1)$ определяется по формуле

$$y_1 = y_0 + (k_0 + 2k_1 + 2k_2 + k_3)/6, \tag{2.5}$$

здесь

$$\begin{gathered} k_0 = hF(x_0, y_0); \quad k_1 = hF(x_0 + h/2, \ y_0 + k_0/2); \\ k_2 = hF(x_0 + h/2, \ y_0 + k_1/2); \quad k_3 = hF(x_0 + h, \ y_0 + k_2); \qquad \text{h - шаг сетки,} \end{gathered} \tag{2.6}$$

т. е. данный вариант метода Рунге-Кутта требует на каждом шаге четырехкратного вычисления правой части уравнения (2.3). Этот алгоритм реализован в программе **ode45.** Кроме этой программы MATLAB располагает обширным набором аналогичных программ, позволяющих успешно решать обыкновенные дифференциальные уравнения.

Пример 5. Решить задачу Коши

$$\frac{dy}{dx} = 2\left(x^2 + y^2\right) \quad \text{на } [0,1] \quad \text{при } y(o) = 1. \tag{2.7}$$

Точное решение имеет вид

$$y(x) = 1.5e^{2x} - x^2 - x - 0.5.$$

Выполним решение данной задачи с помощью программы **ode45**. Вначале в М-файл записываем правую часть уравнения (2.7), сам М-файл оформляется как файл – функция, даем ему имя **F**:

***function dydx* = F(*x, y*)**

***dydx* = *zeros*(1,1);**

***dydx*(1) = 2*(x^2+y(1));**

Для численного решения задачи Коши в окне команд набираются следующие операторы.

Протокол программы.

>>[X Y] = ode45 (@ F , [0 1] , [1]) ;

% Дескриптор @ обеспечивает связь с файлом – функцией правой части

% [0 1] – интервал на котором необходимо получить решение

% [1] – начальное значение решения

>> plot (X,Y) ;

>> % Построение графика численного решения задачи Коши (2.7)

>> hold on; gtext (′ *y*(*x*) ′)

% Команда позволяет с помощью мышки нанести на график надпись *y*(*x*)

>> [X Y]

>> % Последняя команда выводит таблицу численного решения задачи.

Результаты решения. График решения задачи Коши (2.7) показан на рис. 2.3. Численное решение представлено в таблице 2.4, где приведены только отдельные узловые точки. В программе **ode45** по умолчанию интервал разбивается на 40 точек с шагом $h = 1/40 = 0.025$.

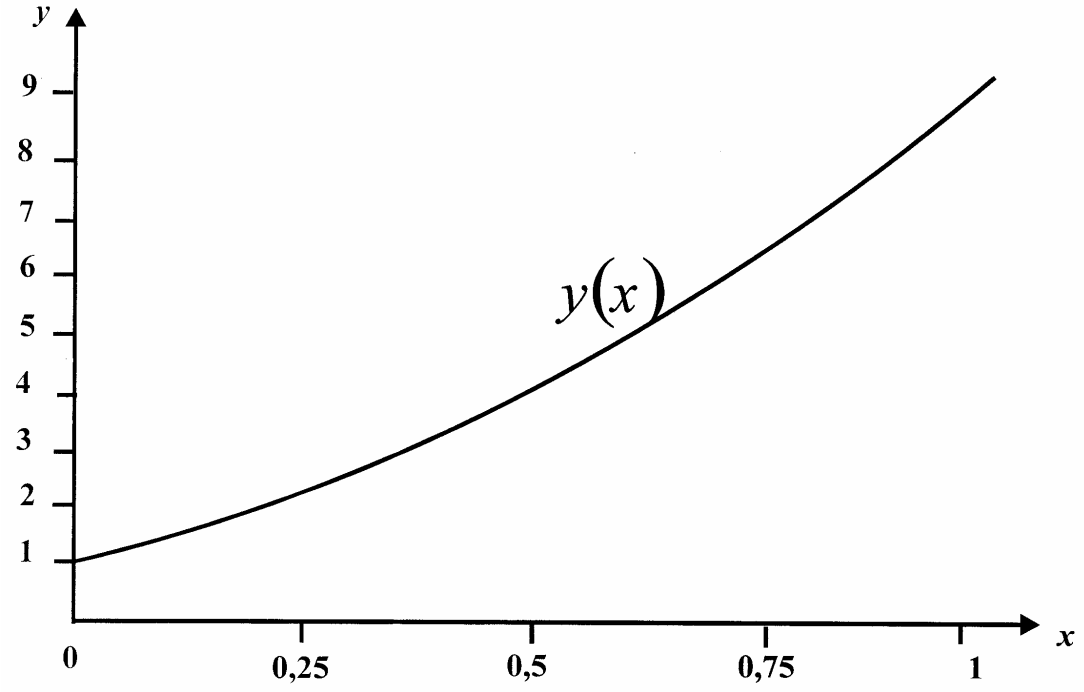


Рис. 2.3

Таблица 2.4

x_i	Метод Рунге-Кутта	Точное решение
0.0	1.0	1.0
0.1	1.2221	1.2221
0.2	1.4977	1.4977
0.3	1.8432	1.8432
0.4	2.2783	2.2783
0.5	2.8274	2.8274
0.6	3.5202	3.5202
0.7	4.3928	4.3928
0.8	5.4895	5.4895
0.9	6.8645	6.8645
1.0	8.5836	8.5836

Как следует из таблицы 2.4 численное решение программой **ode45** является точным.

Варианты заданий. Построить график и вывести в виде таблицы решение задачи Коши на интервале [0; 1] методом Рунге-Кутта 4-го порядка. Данные взять из таблицы 2.5.

Таблица 2.5

№ п/п	f(x,y)	y_0
1.	$x^3 \sin y + 1$	0.0
2.	$x^2 \sin y - 1$	0.1
3.	$e^{+x} + 3y$	2.0
4.	$\sqrt{y^2 + x^3}$	0.3
5.	$\sqrt{y^3 + x^2}$	0.4
6.	$1/(1 + y^2) + x^2$	0.0
7.	$1/(1 + y^2) + xy$	0.1
8.	$\cos y + xy$	0.2
9.	$x^2 \cos y + 0.1$	0.3
10.	$x^3 \cos y + 0.1$	0.4
11.	$\cos(xy) - 0.5$	0.5
12.	$e^{-y} + e^x - 2$	0.0
13.	$e^{-y} - e^x - 0.1$	0.5
14.	$e^{-xy} + 1$	0.4
15.	$\sqrt{y^2 + x^4}$	0.3
16.	$(xy)^2 + \sqrt{x}$	0.2
17.	$\cos(x + y) + x^2$	0.1
18.	$\sin(x + y) - x^2$	0.0
19.	$\sin(xy) + 1$	0.1
20.	$\sin y + xy$	0.2
21.	$(x^2 + y^2)x$	0.3
22.	$e^x(1 + xy)$	0.4
23.	$e^y(1 + xy)$	0.5
24.	$\ln(1 + x^2) + y$	0.6
25.	$1 + ye^{x^2}$	0.7
26.	$\sin x^2 + y^2$	0.0
27.	$\cos x^2 + xy$	0.1
28.	$\sin[y/(1 + x^2)] + 1$	0.2
29.	$\ln(1 + y^2) + 1$	0.3
30.	$1 + xe^{y^2}$	0.4

2.5. Приближенное вычисление определенных интегралов

К вычислениям определенных интегралов сводятся многие практические задачи физики, химии, экологии, механики и других естественных наук. На практике формулой Ньютона-Лейбница не всегда удается воспользоваться. В этом случае используются методы численного интегрирования. Они основаны на следующих соображениях: с геометрической точки зрения определенный интеграл $\int_a^b f(x)dx$ представляет собой площадь криволинейной трапеции. Идея методов численного интегрирования сводится к разбиению интервала [a; b] на множество меньших интервалов и нахождению искомой площади как совокупности элементарных площадей, полученных на каждом частичном промежутке разбиения. В зависимости от использованной аппроксимации получаются различные формулы численного интегрирования, имеющие различную точность. Рассмотрим методы трапеций и Симпсона (парабол).

Метод трапеций.

Здесь используется линейная аппроксимация, т. е. график функции $y = f(x)$ представляется в виде ломаной, соединяющей точки y_i. Формула трапеций при постоянном шаге $h = \frac{b-a}{n}$, где n – число участков, имеет вид

$$\int_a^b f(x)dx = h\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right). \qquad (2.8)$$

В MATLAB данную формулу реализует программа ***trapz* (*x,y*).**

Метод Симпсона

Если подынтегральную функцию заменить параболой, то формула Симпсона с постоянным шагом интегрирования предстанет в виде

$$\int_a^b f(x)dx = \frac{h}{3}\left[y_0 + 4(y_1 + y_3 + \ldots + y_{n-1}) + 2(y_2 + y_4 + \ldots + y_{n-2}) + y_n\right]. \qquad (2.9)$$

В MATLAB формула Симпсона реализуется программой ***quad***. Подынтегральная функция может задаваться с помощью дескриптора @, тогда она программируется в файле – функции, или с помощью апострофов, тогда она запи-

сывается в самой программе ***quad***. Точность вычисления интегралов по умолчанию принята равной $1 \cdot 10^{-6}$.

Пример 6. Вычислить и вывести на печать по методам трапеций и Симпсона значения интеграла

$$\int_0^1 \frac{dx}{1+x^2}$$

Протокол программы метода трапеций

>> *x* = 0 : 0.0001 : 1.0 ;

>> *y* = 1./ (1+*x*.^2) ;

>> *z* = *trapz*(*x*, *y*)

Результат вычислений

z =

0.7854

Протокол программы метода Симпсона

>> *quad* (' (1./(1+*x*.^2)) ', 0, 1)

Результат вычислений

ans =

0.7854

Точное значение интеграла равно 0.785398163.

Как видно из примера 6 полученные результаты являются практически точными, а сами протоколы весьма просты.

Варианты заданий. Вычислить и вывести на печать значения определенного интеграла методами трапеций и Симпсона. Данные взять из таблицы 2.6.

Таблица 2.6

№ п/п	Подынтегральная функция f(x)	Интервал интегрирования [a; b]	Точность вычислений интеграла
1	$\ln x / x\sqrt{1+\ln x}$	[1; 3.5]	0.001
2	$tg^2 x + ctg^2 x$	[π/6; [π/3]	0.002
3	$1/x\ln x$	[1.5; 3.]	0.0001
4	$\ln^2 x / x$	[1.0; 4,0]	0.003
5	$\sqrt{e^x - 1}$	[0; ln2]	0.0015
6	$xe^x \sin x$	[1.0; 4.0]	0.002
7	$x\dfrac{e^x - e^{-x}}{2}$	[0.0; 2.0]	0.001
8	$1/\sqrt{9+x^3}$	[2.0; 5.0]	0.001
9	$\sin(1/x)x^4$	[1.0; 2.5]	0.0005
10	$x^3 arctgx$	[0.0; $\sqrt{3}$]	0.003
11	$\arcsin\sqrt{x/(1+x)}$	[0.0; 3,0]	0.001
12	$x^2(1+\ln x)$	[1.5; 3.0]	0.0025
13	$1/\sqrt{1+3x+2x^2}$	[0,0; 5.0]	0.001
14	$\sqrt{x^2 - 0.14}/x$	[2.3; 6.0]	0.002
15	$2^{3x}\ln\lvert\cos x\rvert$	[0.0; π/2]	0.001
16	$(e^{3x}+1)/(e^x+1)$	[0.0; 2.0]	0.0015
17	$xarctgx/\sqrt{1+x^2}$	[0.0; 2.0]	0.002
18	$\sin^5 x/\ln(1+x^3)$	[0.0; π/4]	0.001
19	$x^2\sqrt{4-x^2}$	[0.0; 1.8]	0.0006
20	$e^x \cos^2 x$	[0.0; π]	0.0016
21	$(1.5x^2+x)/(x^3+2)$	[0.0; 1.2]	0.002
22	$(\sin - x^3)/(x^2+2.7)$	[2.0; 4.4]	0.0014
23	$(3.5tgx+x)/(x^3+3.7)$	[0.0; 1.2]	0.002
24	$(1.2-3.2x^2)/(1+\sin^2 x)$	[2.0; 4.4]	0.0011
25	$(3.17x+4,2)/\cos^2 x$	[1.0; 2.2]	0.0023
26	$(x^2+3.2)/(x^5+4.7)$	[0,0; 1.8]	0.0015
27	$(1.5x^2+x)/(x^3+2)$	[0.0; 1.2]	0.001
28	$e^x \ln\lvert\sin x^2\rvert$	[1.0; 3.0]	0.002
29	tgx^2/e^{-x^2}	[0.0; 1.0]	0.0013
30	$(1.6x-2,7)/(1.5x^3+3.9)$	[1.0; 2.2]	0.0025

2.6. Численное решение нелинейных уравнений

Задача нахождения корней нелинейных уравнений встречается в различных областях научно-технических исследований. Проблема формулируется следующим образом. Пусть задана непрерывная функция $f(x)$ и требуется найти корень уравнения

$$f(x) = 0.$$

Будем предполагать, что имеется интервал изменения x $[a; b]$, на котором необходимо исследовать функцию $f(x)$ и найти значение x_0, при котором $f(x_0)$ равно или весьма мало отличается от нуля.

Данная задача в системе MATLAB может быть решена следующим образом. Вначале необходимо построить график функции $f(x)$ на заданном интервале и убедиться в существовании корня или нескольких корней. Затем применить программы поиска корней. Если существует *один* корень и график $f(x)$ *пересекает* ось *ox*, то можно применить программу ***fzero***. Если $f(x)$ имеет больше одного корня и может касаться и пересекать ось *ox,* то следует применить более мощную программу ***fsolve*** из пакета ***Optimization Toolbox***, которая решает задачу методом наименьших квадратов. Программа ***fzero*** использует известные численные методы: деление отрезка пополам, секущей и обратной квадратичной интерполяции.

Пример 7. Найти корень нелинейного уравнения $10^x + 2x - 100 = 0$ на интервале [1.0; 2.0].

Протокол программы

```
>> % Строим график заданной функции
>> x = 1.0 : 0.001 : 2.0;  y = 10.0.^x + 2.0*x – 100.0;
>> plot (x, y) ;  grid on
```

Появляется окно с графиком функции $10^x + 2x - 100$ (см. рис. 2.4), из которого следует, что корень функции на заданном интервале существует. Для точного определения корня применяем ***fzero*** и ***fsolve***.

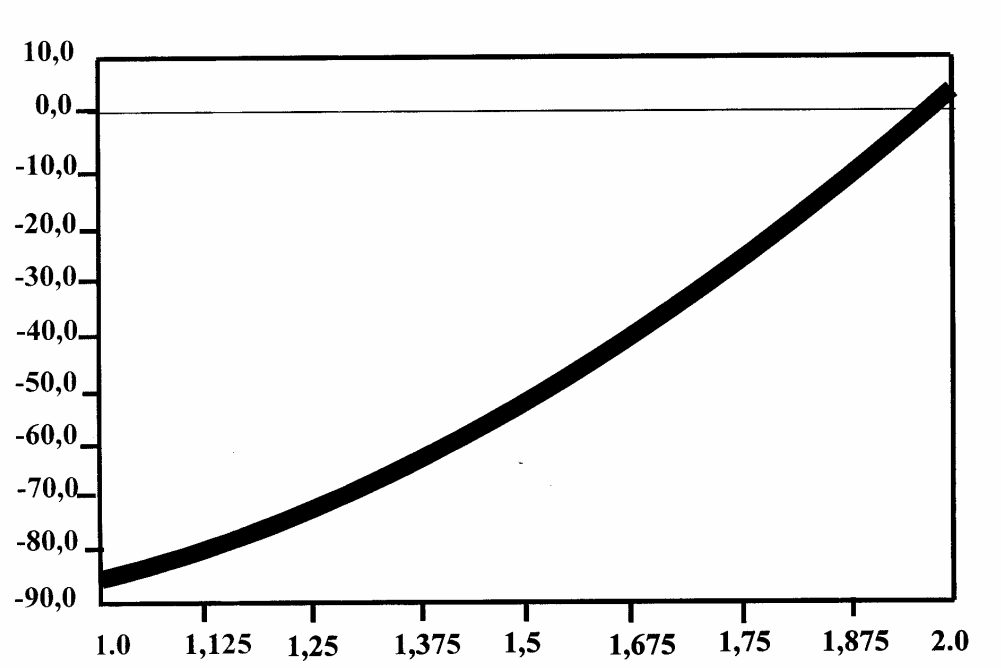


Рис. 2.4

>> X1 = *fzero* (′ (10.0.^*x* + 2.0x* – 100.0) ′, [1.0 2.0])**

Результат решения

X1 =

1.9824

>> X2 = *fsolve* (′ (10.0.^*x* + 2.0x* – 100.0) ′, 1.0 : 2.0)**

Результат решения

*X*2 =

1.9824 1.9824

Варианты заданий. Построить график и найти корень нелинейного уравнения. Данные взять из таблицы 2.7.

Таблица 2.7

№ п/п	Уравнение $f(x) = 0$	Отрезок [a; b]
1	$x\,arctg - 1 = 0$	[1.0; $\sqrt{3}$]
2	$e^{x-2} - \ln(x+2) = 0$	[2.0; 3.0]
3	$x^3 - 9x^2 + 5x - 6 = 0$	[8.0; 9.0]
4	$e^x - \frac{1}{x} - 1 = 0$	[0.5; 1.0]

5	$arctg2x - \frac{1}{1+x} = 0$	[0.0; 1.0]
6	$e^x - \ln x - 20 = 0$	[3.0; 3.2]
7	$\sqrt{x} - tgx(1-x) = 0$	[0.0; 1.0]
8	$\sin\sqrt{x} - \cos\sqrt{x} + 2\sqrt{x} = 0$	[0.0; 0.2]
9	$x^4 + 2x^3 - x - 1 = 0$	[0.8; 1.0]
10	$x^3 - e^{4x} - 5.5 = 0$	[2.6; 3.0]
11	$x^6 - 3x^2 + x - 1 = 0$	[1.0; 1.5]
12	$\sqrt[3]{5-x} - x = 0$	[1.0; 2.0]
13	$x^2 - \ln x = 0$	[0.0; 1.0]
14	$x^2 - \cos x = 0$	[0.0; 1.0]
15	$\ln x - arctgx = 0$	[3.0; 4.0]
16	$x^2 arctgx - 1 = 0$	[1.0; 1.2]
17	$x^2 + \ln x - 4 = 0$	[1.0; 2.0]
18	$x - arctg\sqrt[3]{x} = 0$	[0.0; 1.0]
19	$x^2 - e^x - 2 = 0$	[-0.2; -0.1]
20	$x^2 - \ln(x+1) = 0$	[0.1; 0.9]
21	$x^5 - x - 2 = 0$	[1.0; 1.4]
22	$x - 2 - \sqrt[4]{x} = 0$	[3.0; 4.0]
23	$x - tgx = 0$	[0.0; 1.5]
24	$x + \ln x - 0.5 = 0$	[0.0; 1.0]
25	$\ln x + \sqrt{x} = 0$	[0.1; 1.0]
26	$\sqrt{x} - \cos\sqrt{x} = 0$	[0.4; 0.6]
27	$x^2 - \ln(1+x^2) - 9.75 = 0$	[3.0; 4.0]
28	$x + \sqrt[3]{x} - 6.09 = 0$	[4.0; 5.0]
29	$x^3 - \sqrt{x} - 9.5 = 0$	[2.0; 3.0]
30	$\arccos 2x - x^2 - 0.35 = 0$	[0.0; 0.48]

2.7. Численное решение оптимизационных задач

Под оптимизацией понимают процесс выбора наилучшего варианта из всех возможных. С точки зрения инженерных расчетов методы оптимизации позволяют выбрать наилучший вариант конструкции, наилучшее распределение ресурсов, минимальный урон природной среде и т. п. В процессе решения задачи оптимизации необходимо найти оптимальные значения некоторых параметров, их называют проектными параметрами. Выбор оптимального реше-

ния проводится с помощью некоторой функции, называемой целевой функцией. Целевую функцию можно записать в виде

$$u = f(x_1, x_2, ..., x_n), \tag{2.10}$$

где x_1, x_2, … , x_n – проектные параметры.

Можно выделить 2 типа задач оптимизации – *безусловные* и *условные*. Безусловная задача оптимизации состоит в отыскании максимума или минимума функции (2.10) от n действительных переменных и определении соответствующих значений аргументов на некотором множестве G n-мерного пространства. Обычно рассматриваются задачи минимизации; к ним легко сводятся и задачи на поиск максимума путем замены знака целевой функции на противоположный. Условные задачи оптимизации – это такие, при формулировке которых задаются некоторые условия (ограничения) на множестве G. Здесь рассмотрим только безусловные задачи оптимизации.

Поиск минимума функции одной переменной.

Для решения этой задачи используются методы золотого сечения или параболической интерполяции (в зависимости от формы задания функции), реализованные в программе ***fminbnd***.

Пример 8. Найти и вывести на печать минимальное значение функции

$f(x) = 24 - 2x/3 + x^2/30$ на [5; 20].

Строим график этой функции, чтобы убедиться в наличии минимума на заданном интервале.

Протокол программы

>> *x* = 5.0 : 0.001 : 20.0 ; *y* = 24 – 2* *x*/3 + *x*.^2/30 ;

>> *plot*(*x, y*) ; *grid on*

Появляется окно с графиком этой функции (рис. 2.5), где отмечаем наличие минимума.

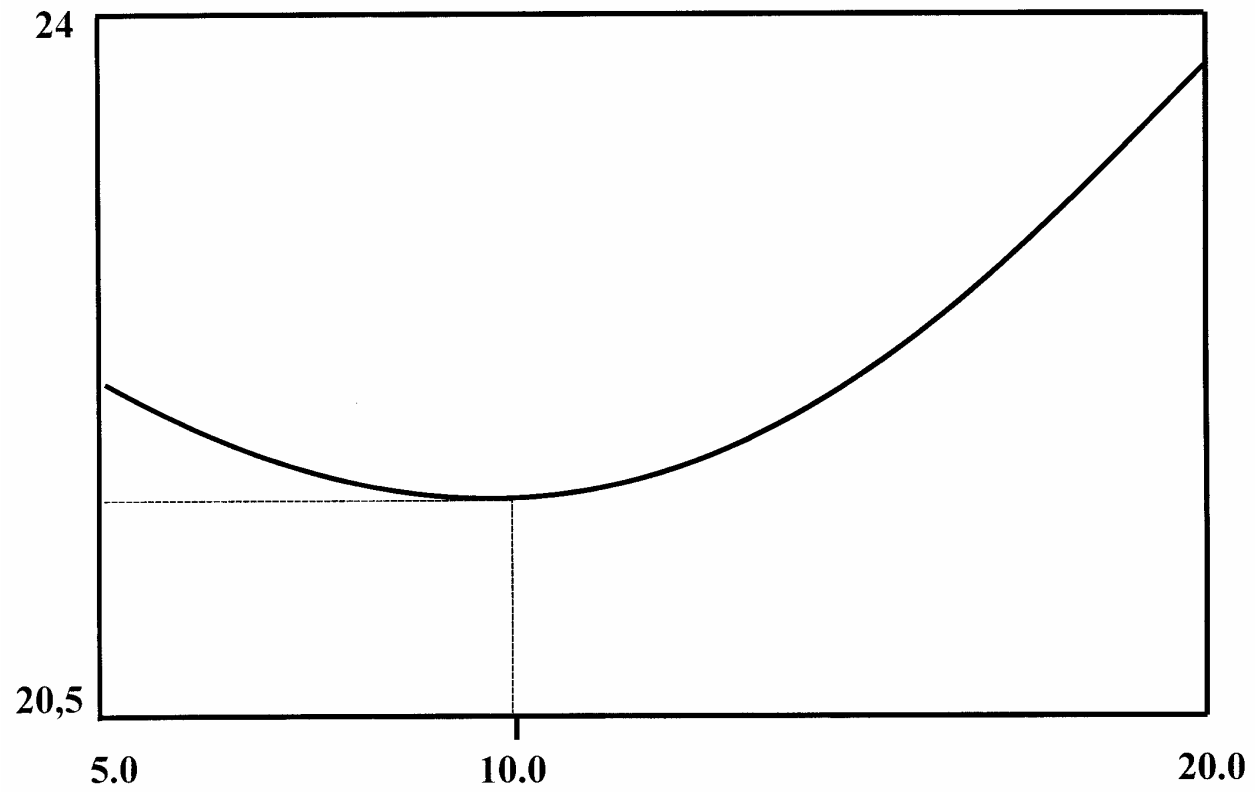


Рис. 2.5

Далее, для точного определения координаты и значения минимума привлекаем программу ***fminbnd***.

>> [*x, y*] = *fminbnd* (′ (24.0 – 2* *x*/3 + *x*.^2/30) ′, 5.0, 20.0)

Результат поиска

x =

10.0000

y =

20.6667

Варианты заданий. Найти и вывести на печать координату и минимальное значение функции $f(x)$ на $[a;\ b]$. Данные взять из таблицы 2.8.

Таблица 2.8

№ п/п	Функция $f(x)$	Отрезок [a; b]
1	$f(x) = x / \ln x$	[1.2; 4]
2	$f(x) = x - 2\sin x$	[0; π/2]
3	$f(x) = (x^2 - 1)/(x^2 + 1)$	[-2; 2]
4	$f(x) = 3x^4 - 4x^3 + 12x^2 + 1$	[-2; 2]
5	$f(x) = x - 2\ln x$	[1; 3]
6	$f(x) = e^x \cos x$	[π; 3π/2]
7	$f(x) = (1 - x + x^2)/(1 + x - x^2)$	[0; 1]
8	$f(x) = -\sqrt{2x - x^2}$	[0; 2]
9	$f(x) = (x-2)^5 (2x+1)^4$	[-0.5; 1.5]
10	$f(x) = x^x$	[0,1; 1.0]
11	$f(x) = e^{-1/x^2}$	[-0.5; 0,5]
12	$f(x) = x\sqrt{1 - x^2}$	[-1.0; 0]
13	$f(x) = ((e^x + e^{-x})/2)^3 + 1$	[-0.5; 0.5]
14	$f(x) = -x/(x^3 + 2)$	[0.5; 1.5]
15	$f(x) = x^2 / \sqrt[3]{x^3 - 4}$	[1.6; 2.2]
16	$f(x) = \sqrt{\lvert x^2 - 3 \rvert} / x$	[1; 2]
17	$f(x) = x^2 \sqrt{\lvert x^2 - 1 \rvert}$	[1.1; 1.6]
18	$f(x) = 1/(\sin x + \cos x)$	[0; π/3]
19	$f(x) = x/2 + arctgx$	[0.5; 1.2]
20	$f(x) = xe^{-x^2/2}$	[-1.5; -0.5]
21	$f(x) = e^{-1x^2} / x$	[-2.0; -1.0]
22	$f(x) = -\lvert x \rvert^3 e^{-x^2/2}$	[-2.0; -1.0]
23	$f(x) = x^2 \ln x$	[0.1; 1.0]
24	$f(x) = x \ln^2 \lvert x \rvert$	[-0,05; -0.2]
25	$f(x) = xarctgx$	[-0.5; 0.5]
26	$f(x) = \sin x + \cos x$	[π; 3π/2]
27	$f(x) = -\ln x / x^2$	[1.0; 2.0]
28	$f(x) = x^2 \ln^2 x$	[0.1; 0.5]
29	$f(x) = \sin x / x$	[π; 2π]
30	$f(x) = -x^{1/x}$	[2.0; 3.0]

Поиск минимума функций нескольких переменных

Данная задача значительно сложнее первой. Рассмотрим ее решение на примере функции двух переменных. Алгоритм может быть распространен на функции большего числа переменных. Для минимизации функций нескольких переменных MATLAB использует симплекс – метод Нелдера-Мида. Данный метод является одним из лучших методов поиска минимума функций многих переменных, где не вычисляются производные или градиент функции. Он сводится к построению симплекса в n-мерном пространстве, заданного $n + 1$ вершиной. В двухмерном пространстве симплекс является треугольником, а в трехмерном – пирамидой. На каждом шаге итераций выбирается новая точка решения внутри или вблизи симплекса. Она сравнивается с одной из вершин симплекса. Ближайшая к этой точке вершина симплекса заменяется этой точкой. Таким образом, симплекс перестраивается и позволяет найти новое, более точное положение точки решения. Алгоритм поиска повторяется, пока размеры симплекса по всем переменным не станет меньше заданной погрешности решения. Программу, реализующую симплекс-методы Нелдера-Мида, удобно использовать в следующей записи

[*x*, min *f*] = *f* min search (…),

где x – вектор координат локального минимума;

min f – значение целевой функции в точке минимума.

Саму целевую функцию удобно представить с помощью дескриптора @ в М-файле.

Пример 9. Найти и вывести на печать координаты и значение минимума функции двух переменных $f(x, y) = (x^2 + y^2 - 3)^2 + (x^2 + y^2 - 2x - 3)^2 + 1$, если начальная точка поиска имеет координаты M_0 (1; 1). Анализ функции показывает, что $min\, f = 1$ $x = 0$, $y = \sqrt{3} = 1.73205$.

Строим трехмерный график этой функции, чтобы убедиться в наличии минимума. Возьмем интервал $x \in [-1; 1]$; $y \in [1; 3]$.

Протокол программы

>> [X,Y] = *mesh grid* ([-1 : 1, 1 : 3]) ;

>> Z = (X.^2 + Y.^2 – 3).^2 + (X.^2 + Y.^2 – 2*X – 3).^2 + 1 ;

>> *plot* 3 (X,Y,Z)

После построения трехмерного графика выполняем поиск минимума. В М-файле программируем целевую функцию

function f = F *xy*(*x*)

f = (*x*(1) ^ 2 + *x*(2) ^ 2 –3)^2 – 3)^2 + (*x*(1) ^ 2 + *x*(2) ^ 2 –2* *x*(1) – 3)^2 + 1;

Решаем поставленную задачу в окне команд

>> [*xmin, mi f*] = *fminsearch* (@ F*xy*, [1; 1])

Результаты поиска

xmin =

- 0.0000 1.7320

minf =

1.0000

Как видно, результаты решения задачи точные.

Варианты заданий. Найти и вывести на печать координаты и минимальное значение функции двух переменных. Поиск начать с точки M_0 (x_0, y_0). Данные взять из таблицы 2.9.

Таблица 2.9

№ п/п	Функция $f(x, y)$	Координаты начальной точки M_0 (x_0, y_0).
1	2	3
1	$(2x^2 - y - 3)^2 + x^2 + 2x + 2$	(1; 1)
2	$(xy + 2)^2 + y^2 + 2y + 4$	(2; 2)
3	$(x^2y^2 - y + 2)^2 + x^2 + 1$	(2; 2)
4	$(3x^2 + 2y^2 - 1)^2 + (xy - 3)^2$	(2; 2)
5	$(2x^2 - 7y^2 - 2)^2 + (x^2 + y^2 - 20)^2 + 3$	(2; 2)
6	$(x^2 + y^2 - 2x - 3)^2 + (x^2 + y^2 - 2y - 3)^2$	(2; 2)
7	$(x^2 - 6x + y^2 + 8)^2 + x^2y^2 + 1$	(2; 2)
8	$(x^2 - y = 2)^2 + (x - y + 3)^2$	(2; 2)
9	$\ln(1 + x^2 + y^2)^2 + (x - y - 1)^2$	(2; 2)

Продолжение таблицы 2.9

1	2	3
10	$\left(x^2+y^2-1\right)^2+\left(x^2-6x+y^2+8\right)^2$	(2; 2)
11	x^3+y^3-3xy	(0.5; 0.5)
12	$x^2+xy+y^2-3x-6y$	(0.5; 3.5)
13	$-xy^2\left(1-x-y\right)$	(0; 0)
14	$3x^2-x^3+3y^2+4y$	(0.1; -1.0)
15	$xy+50/x+20/y$	(4; 1)
16	$x^2+y^2-2\ln x-18\ln y$	(0.5; 2.5)
17	$x^3+3xy^2-15x-12y$	(1.5; 0.5)
18	$2x^3-xy^2+5x^2+y^2$	(0.5; 0.5)
19	$\left(2x^2+y^2\right)e^{-\left(x^2+y^2\right)}$	(0.3; 0.3)
20	$-2+\sqrt[3]{x^2+y^2}$	(0.25; 0.25)
21	$\left(x-1\right)^2+\left(y-2\right)^2-1$	(0.5; 1.5)
22	$\sin\left(x^2+y^2-0.5\right)$	(0.5; 0.5)
23	$x^2-xy+y^2+2x-2y+1$	(-1.5; 0.5)
24	$xy\left(x+y-4\right)$	(1.0; 1.0)
25	$x^3y^2\left(x+y-5\right)$	(2.0; 1.5)
26	$x^2+xy+y^2+1/x+1/y$	(0.2; 0.3)
27	$-\left(\sin x+\sin y+\sin\left(x+y\right)\right)$	(π/4; π/4)
28	$-\sin x\sin y\sin\left(x+y\right)$	(π/4; π/4)
29	$x^3y^3-9xy+1$	(2.5; 2.5)
30	$x^4+y^4-2x^2+4xy-2y^2+1$	(1.0; -1.0)

В заключение отметим, что представленные примеры использования возможностей MATLAB в решении различных научно-технических проблем являются лишь небольшой иллюстрацией значительно большего потенциала системы, а материал главы может служить некоторым введением при самостоятельном изучении и применении интегрированного пакета математического моделирования. Следующая глава ориентирована на решение в среде MATLAB прикладных задач сугубо инженерной науки – строительной механики, где нашли свое применение отдельные вопросы численных методов.

ГЛАВА 3. Применение MATLAB для решения задач строительной механики стержневых систем

Изложенные выше основные данные о программировании и использовании численных методов в интегрированном пакете математического моделирования MATLAB показывают его большие возможности в решении задач прикладных наук. Одной из наиболее значимых прикладных наук в механико-инженерном образовании является *строительная механика*, где рассматриваются вопросы расчета машиностроительных, судостроительных, авиационных, строительных и т. п. конструкций. Объединение в одном пакете программирования и визуализации результатов расчетов позволяет существенно обогатить учебный процесс, повысить его методический уровень и наглядность теории, значительно уменьшить время на выполнение индивидуальных заданий и обеспечить глубокое овладение современными компьютерными технологиями. По этим причинам внедрение MATLAB в учебный процесс является актуальной задачей высшей школы. Важным условием высокого научного уровня учебного процесса является также широкое использование современных достижений соответствующих наук. В частности, в строительной механике накоплен большой арсенал методов расчета стержневых конструкций на статическую, динамическую нагрузки и устойчивость. Одним из наиболее эффективных методов строительной механики является метод граничных элементов (МГЭ), разработанный рядом зарубежных и отечественных ученых. В данной книге применен численно-аналитический вариант МГЭ, который с исчерпывающей полнотой изложен в работе [2]. Как ниже будет показано, реализация МГЭ в пакете MATLAB исключает какое-либо программирование численных процессов. Сами программы представляют собой лишь процесс введения исходных матриц и вызов численных процедур, т. е. реализуются в полном объеме идеи структурного программирования, что значительно уменьшает количество ошибок и повышает достоверность результатов. Дальнейшая графическая визуализация результатов расчетов ло-

гически завершает постановку задачи и ее решение. Таким образом, объединение современных достижений науки и компьютерных технологий будет способствовать подготовке высококвалифицированных специалистов и магистров. Кратко рассмотрим суть МГЭ в задачах деформирования стержневых систем.

3.1 Основные положения метода граничных элементов

Значительное число задач механики упругого стержня сводится к решению линейного неоднородного дифференциального уравнения с постоянными коэффициентами

$$a_0 y^{(n)} + a_1 y^{(n-1)} + \ldots + a_n y = q(x), \tag{3.1}$$

удовлетворяющего начальным условиям

$$y(o) = y_0; \quad y'(o) = y'_0; \ \ldots \ y^{(n-1)}_{(o)} = y^{(n-1)}_0 \ . \tag{3.2}$$

Как известно, такая задача определения частного решения уравнения (3.1), удовлетворяющего условием (3.2), называется задачей Коши. Для изгиба, поперечных колебаний, кручения тонкостенных стержней, продольно-поперечного изгиба и т. п. видов деформирования, решение задачи Коши можно записать в матричной форме.

$$\begin{bmatrix} \mathbf{EIn(x)} \\ \mathbf{EI\,j\,(x)} \\ \mathbf{M(x)} \\ \mathbf{Q(x)} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} & \mathbf{-A_{13}} & \mathbf{-A_{14}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} & \mathbf{-A_{23}} & \mathbf{-A_{24}} \\ \mathbf{-A_{31}} & \mathbf{-A_{32}} & \mathbf{A_{33}} & \mathbf{A_{34}} \\ \mathbf{-A_{41}} & \mathbf{-A_{42}} & \mathbf{A_{43}} & \mathbf{A_{44}} \end{bmatrix} \begin{bmatrix} \mathbf{EIn(o)} \\ \mathbf{EI\,j\,(o)} \\ \mathbf{M(o)} \\ \mathbf{Q(o)} \end{bmatrix} + \int_0^x \begin{bmatrix} \mathbf{A_{14}(x\text{-}x)} \\ \mathbf{A_{24}(x\text{-}x)} \\ \mathbf{-A_{34}(x\text{-}x)} \\ \mathbf{-A_{44}(x\text{-}x)} \end{bmatrix} \mathbf{q}(x)\mathbf{d}x \ , \tag{3.3}$$

или компактно

$$\mathbf{Y}(x) = \mathbf{A}(x)\,\mathbf{X}(o) + \mathbf{B}(x), \tag{3.4}$$

где **Y**(x) – вектор параметров напряженно-деформированного состояния стержня в текущей точке;

A(x) – квадратная матрица фундаментальных ортонормированных функций уравнения (3.1);

X(o) – вектор начальных параметров;

B – вектор (матрица-столбец) внешней нагрузки.

Если несколько стержней соединены в единую конструкцию, то для системы стержней можно составить матричное уравнение типа (3.4). Матрица **A** преобразуется к квазидиагональному виду, а векторы **Y, X** и **B** будут содержать параметры состояния всех стержней по следующей структуре

$$\mathbf{A}=\begin{bmatrix} \mathbf{A}_1 & & & & \\ & \mathbf{A}_2 & & & \\ & & \mathbf{O} & & \\ & & & \mathbf{A}_{n-1} & \\ & & & & \mathbf{A}_n \end{bmatrix};\ \mathbf{Y}=\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_{n-1} \\ \mathbf{Y}_n \end{bmatrix};\ \mathbf{X}=\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_{n-1} \\ \mathbf{X}_n \end{bmatrix};\ \mathbf{B}=\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_{n-1} \\ \mathbf{B}_n \end{bmatrix} \qquad (3.5)$$

Если координате x каждого стержня дать граничное значение $\mathbf{l}_i$, то для системы матриц (3.5) можно выполнить достаточное простое преобразование по схеме

$$\mathbf{Y}(\mathbf{l})=\mathbf{A}(\mathbf{l})\mathbf{X}(o)+\mathbf{B}(\mathbf{l})\rightarrow\mathbf{A}(\mathbf{l})\mathbf{X}(o)-\mathbf{Y}(\mathbf{l})=-\mathbf{B}(\mathbf{l})\ ®\ \mathbf{A}_*(\mathbf{l})\mathbf{X}_*(o,\mathbf{l})=-\mathbf{B}(\mathbf{l}), \qquad (3.6)$$

где конечные граничные параметры матрицы **Y** переносятся на место нулевых параметров вектора **X**. При этом, эти векторы дополняются уравнениями равновесия и совместности перемещений узловых точек и граничными условиями. В конце схемы преобразований (3.6) получается система линейных алгебраических уравнений относительно начальных и конечных параметров всех стержней конструкции. После вычисления начальных параметров стержней их напряженно-деформированное состояние определяется по матричному уравнению (3.3). Таким образом, решение прямых задач строительной механики стержневых систем в МГЭ сводится к решению одной системы линейных алгебраических уравнений и вычислению напряженно-деформированного состояния в промежуточных точках по соотношениям метода начальных параметров. Такая схема решения обеспечивает получение весьма точных и достоверных результатов, которые можно представить средствами MATLAB в виде обычных эпюр, форм свободных колебаний, потерь устойчивости и т. п.

Главной операцией в схеме (3.6) является перенос параметров из **Y** в **X**. Процесс переноса конечных параметров вектора **Y** в вектор **X** основан на следующих положениях. Векторы **Y, X** любой стержневой (и не стержневой) конструкции при граничном значении координаты $x = \mathbf{l}_i$ будут содержать 3 группы параметров.

Первая группа – это нулевые граничные параметры, что определяется заданными условиями опирания (краевыми условиями).

Вторая группа – это зависимые параметры, связанные между собой обычными уравнениями равновесия и совместности перемещений узлов конструкции.

Третья группа граничных параметров никак не связаны между собой. Эти параметры условно могут быть названы независимыми. Перенос параметров из вектора **Y** в вектор **X** должен компенсироваться ненулевыми элементами матрицы **A**, иначе нарушается исходное уравнение схемы (3.6). Очевидно, что независимые параметры вектора **Y** должны быть перенесены на место нулевых параметров вектора **X**, а зависимые параметры переносятся в соответствии с уравнениями их связи. Перед операцией переноса параметров необходимо освободить поля матрицы **A** от элементов, связанных с нулевыми параметрами вектора **X**, т. е. обнулить столбцы матрицы **A**, номера которых равны номерам нулевых строк матрицы **X**. Далее в матрицу **A** вводятся ненулевые компенсирующие элементы и преобразования по схеме (3.6) завершены, поскольку в матрице **B** изменяются только знаки элементов. Правило для определения величины и положения компенсирующих элементов при переносе параметров включает 3 основных случая.

1-й случай. При переносе независимого параметра вектора **Y** в вектор **X** компенсирующий элемент матрицы **A** равен коэффициенту при переносимом параметре со своим знаком по схеме

	i	*j*	*k*	*l*
i		**l**	$-\mathbf{l}^2/2$	$-\mathbf{l}^3/6$
j		**1**	$-\mathbf{l}$	$-\mathbf{l}^2/2$
k	***-a***		**1**	**l**
l				**1**

$\mathbf{X}_{i1}=\mathbf{0};\mathbf{Y}_{k1}$
$\mathbf{X}_{j1}$
$\mathbf{X}_{k1}$
$\mathbf{X}_{l1}$

-

i	$\mathbf{Y}_{i1}=0$
j	$\mathbf{Y}_{j1}=0$
k	$a\mathbf{Y}_{k1}$
l	$\mathbf{Y}_{l1}=0$

, (3.7)

т. е. компенсирующий элемент матрицы **A** равен (– *a*) и должен появиться на месте (*k, i*), где *k* – номер строки матрицы **Y**, где находился параметр, *i* – номер строки матрицы **X**, куда переносится параметр. Другими словами, первый индекс положения компенсирующего элемента указывает на старый адрес переносимого параметра, а второй индекс – новый адрес в матрице **X**.

<u>2-й случай</u>. Перенос зависимых параметров представляет собой повторение операцией 1-го случая с той лишь разницей, что в матрице **X** не появляются новые параметры, а в матрице $\mathbf{A}_*$ соответствующие строки могут содержать несколько компенсирующих элементов

	i	*j*	*k*	*l*
i	***-a***	***b***		
j				
k				
l	**-1**			

$\mathbf{X}_{i1}=\mathbf{0};\mathbf{Y}_{l1}$
$\mathbf{X}_{j1}$
$\mathbf{X}_{k1}$
$\mathbf{X}_{l1}$

-

i	$\mathbf{Y}_{i1}=a\mathbf{Y}_{l1}-b\mathbf{X}_{j1}$
j	$\mathbf{Y}_{j1}=0$
k	$\mathbf{Y}_{k1}=0$
l	$\mathbf{Y}_{l1}$

. (3.8)

<u>3-й случай</u>. В сложных конструкциях в одном узле могут находиться несколько начальных точек стержней. В этом случае возникают уравнения связи между начальными параметрами и требуется переносить параметры в пределах вектора **X**. Компенсирующие элементы в этом случае формируются из системы фундаментальных функций, столбцы которой получают сдвиг в соответствии со схемой

$$
\begin{array}{c|cccc|c}
 & i & j & k & l & \\
\hline
i & 1 & \mathbf{l} & -\mathbf{l}^2/2 & -\mathbf{l}^3/6 & \mathbf{X}_{i1} \\
j & & 1 & -\mathbf{l} & -\mathbf{l}^2/2 & \mathbf{X}_{j1} \\
k & & & 1 & \mathbf{l} & \mathbf{X}_{k1}=a\,\mathbf{X}_{i1} \\
l & & & & 1 & \mathbf{X}_{l1} \\
\hline
\end{array}
;
$$

$$
\begin{array}{c|cccc|c}
 & i & j & k & l & \\
\hline
i & 1-a\mathbf{l}^2/2 & \mathbf{l} & & -\mathbf{l}^3/6 & \mathbf{X}_{i1} \\
j & -al & 1 & & -\mathbf{l}^2/2 & \mathbf{X}_{j1} \\
k & a & & & \mathbf{l} & \\
l & & & & 1 & \mathbf{X}_{l1} \\
\hline
\end{array}
. \qquad (3.9)
$$

Видно, что элементы матрицы **A** сдвигаются на место столбца, номер которого равен номеру строки нового положения параметра. Компенсирующие элементы равны произведению коэффициента при переносимом параметре на элементы матрицы **A**. При этом возрастает число компенсирующих элементов по сравнению со случаями 1 и 2. Поэтому необходимо, по возможности, избегать случаев, когда в узле сходятся только начальные точки стержней.

Более подробно алгоритм формирования системы линейных алгебраических уравнений типа (3.6) для стержневых систем рассмотрен ниже, при решении конкретных задач.

3.2. Расчет неразрезной балки

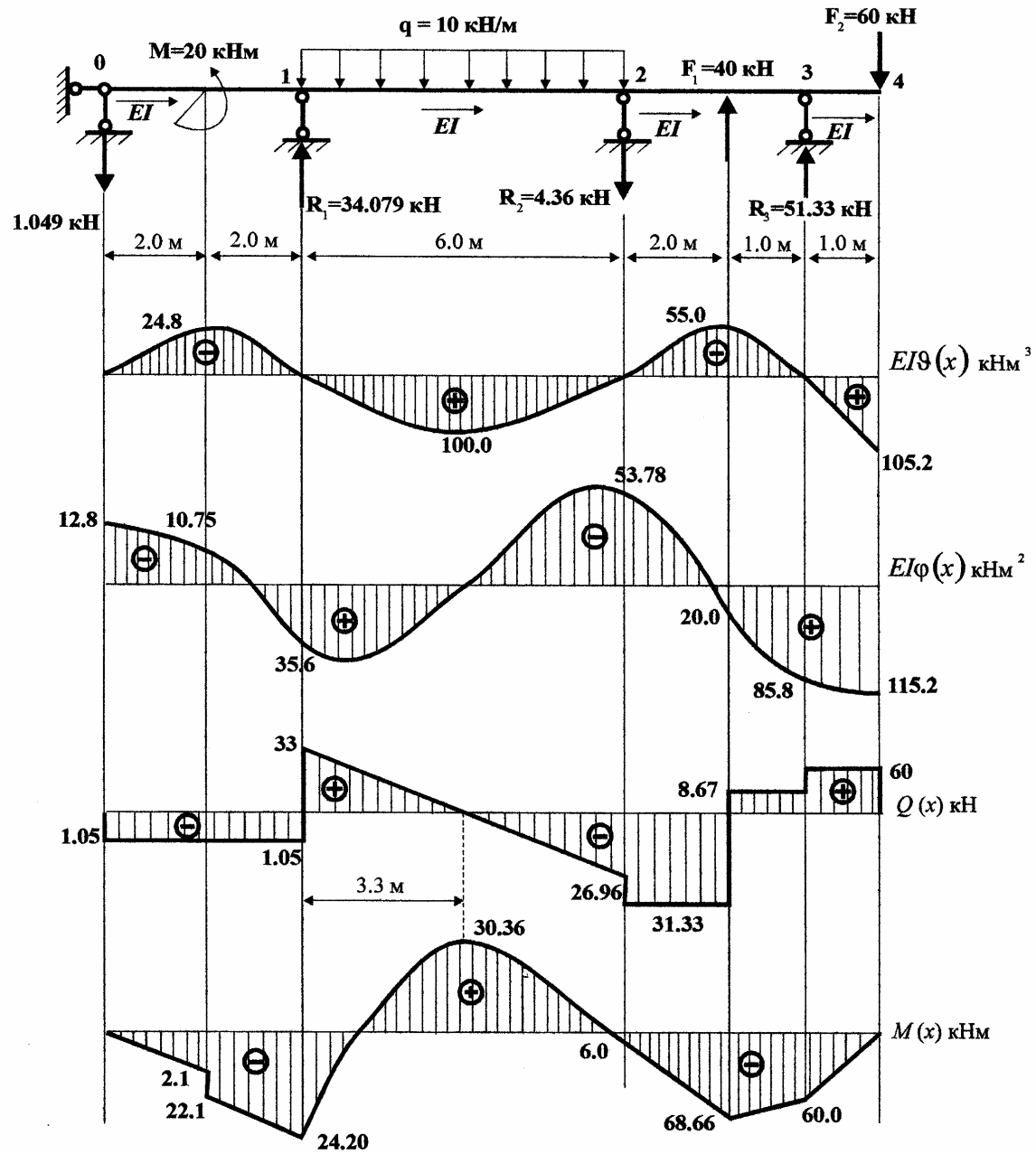
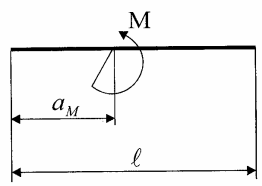
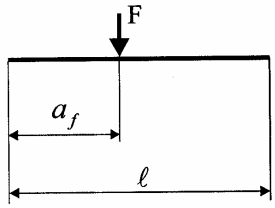
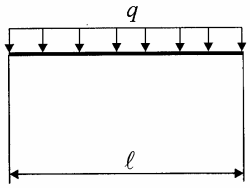


Рис. 3.1







b(1,1) =	**-M* (*l* – am)^2/2;**	*b*(1,1) =	**-F* (*l* –*af*)^3/6;**	*b*(1,1) =	**-q**l*^4/24;**
b(2,1) =	**-M* (*l* – am);**	*b*(2,1) =	**-F* (*l* –*af*)^2/2;**	*b*(2,1) =	**-q**l*^3/6;**
b(3,1) =	**M;**	*b*(3,1) =	**F* (*l* –*af*);**	*b*(3,1) =	**q**l*^2/2;**
b(4,1) =	**O;**	*b*(4,1) =	**F**	*b*(4,1) =	**q**l*.**

(3.10)

Рис. 3.2

3.2.1. *Определение граничных параметров при статической нагрузке*

Исходные данные расчета балки:

l_1 = 4.0; l_2 = 6.0; l_3 = 3.0; l_4 = 1.0; m = 20.0; q = 10.0; f_1 = - 40.0; f_2 = 60.0; am = 2.0; af_1 = 2.0; af_2 = 1.0.

Система линейных уравнений для граничных параметров балки

$$
\begin{array}{c|cccccccccccccccc|c}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & \mathbf{X(16,1)} \\
\hline
1 & & & & & & & & & & & & & & & & & EI\theta^{0-1}(o)=0;Q^{0-1}(\ell) \\
2 & & \ell_1 & & -\ell_1^3/6 & & & & & & & & & & & & & EI\varphi^{0-1}(o) \\
3 & & 1 & & -\ell_1^2/2 & & -1 & & & & & & & & & & & M^{0-1}(o)=0;Q^{1-2}(\ell) \\
4 & & & & \ell_1 & & & -1 & & & & & & & & & & Q^{0-1}(o) \\
5 & -1 & & & 1 & & & & & & & & & & & & & EI\theta^{1-2}(o)=0;Q^{2-3}(\ell) \\
6 & & & & & & \ell_2 & -\ell_2^2/2 & -\ell_2^3/6 & & & & & & & & & EI\varphi^{1-2}(o) \\
7 & & & & & & 1 & -\ell_2 & -\ell_2^2/2 & & -1 & & & & & & & M^{1-2}(o) \\
8 & & & & & & & 1 & \ell_2 & & & -1 & & & & & & Q^{1-2}(o) \\
9 & & & -1 & & & & & 1 & & & & & & & & & EI\theta^{2-3}(o)=0;EI\theta^{3-4}(\ell) \\
10 & & & & & & & & & \ell_3 & -1 & & & & & & & EI\varphi^{2-3}(o) \\
11 & & & & & -1 & & & & -\ell_3^3/2 & -\ell_3 & 1 & -\ell_3^3/6 & & & & & M^{2-3}(o) \\
12 & & & & & & & & & -\ell_3^3/2 & -\ell_3^3/2 & \ell_3 & 1 & & & & & Q^{2-3}(o) \\
13 & & & & & & & & & & & & & & & & -\ell_4^3/6 & EI\theta^{3-4}(o)=0;EI\varphi^{3-4}(\ell) \\
14 & & & & & & & & & & -1 & & & -1 & & & -\ell_4^2/2 & EI\varphi^{3-4}(o) \\
15 & & & & & & & & & & & -1 & & & 1 & & -\ell_4 & M^{3-4}(o) \\
16 & & & & & & & & & & & & & & \ell_4 & 1 & -\ell_4^3/2 & Q^{3-4}(o) \\
\hline
\mathbf{Y(16,1)} & EI\theta^{0-1}(\ell)=0 & EI\varphi^{0-1}(\ell)=EI\varphi^{1-2}(o) & M^{0-1}(\ell)=M^{1-2}(o) & Q^{0-1}(\ell) & EI\theta^{1-2}(\ell)=0 & EI\varphi^{1-2}(\ell)=EI\varphi^{2-3}(\ell) & M^{1-2}(\ell)=M^{2-3}(o) & Q^{1-2}(\ell) & EI\theta^{2-3}(\ell)=0 & EI\varphi^{2-3}(\ell)=EI\varphi^{3-4}(o) & M^{2-3}(\ell)=M^{3-4}(o) & Q^{2-3}(\ell) & EI\theta^{3-4}(\ell) & EI\varphi^{3-4}(\ell) & M^{3-4}(\ell)=0 & Q^{3-4}(\ell)=0 & \\
\hline
\mathbf{b(16,1)} & -M(\ell_1-a_M)^2/2 & -M(\ell_1-a_M) & M & 0 & -q\ell_2^4/24 & -q\ell_2^3/6 & q\ell_2^2/2 & q\ell_2 & -f_1(\ell_3-af_1)^3/6 & -f_1(\ell_3-af_1)^2/2 & f_1(\ell_3-af_1) & f_1 & -f_2(\ell_4-af_2)^3/6 & -f_2(\ell_4-af_2)^2/2 & f_2(\ell_4-af_2) & f_2 & \\
\end{array}
\qquad (3.11)
$$

Программа расчета граничных параметров балки

Протокол с комментариями

```
% Описание матрицы a(16,16); b(16,1) и X(16,1);
a = zeros(16,16); ); b = zeros (16,1); X = zeros (16,1);
% Исходные данные расчета балки
l1 = 4.0; l2 = 6.0; l3 = 3.0; l4 = 1.0; m = 20.0; q = 10.0;
 f1 = - 40.0; f2 = 60.0;  am = 2.0; af1 = 2.0; af2 = 1.0;
% Ввод матриц a(16,16) и b(16,1) с помощью операторов присваивания;
a(1,2) = l1; a(1,4) = -l1^3/6; b(1,1) = -m*(l1-am)^2/2;
a(2,2) = 1.0;  a(2,4) = -l1^2/2; a(2,6) = -1.0;  b(2,1) = -m*(l1-am);
a(3,4) = l1;  a(3,7) = -1.0;  b(3,1) = m; a(4,1) = -1.0;  a(4,4) = 1.0;
a(5,6) = l2;  a(5,7) = -l2^2/2;  a(5,8) = -l2^3/6;  b(5,1) = -q*l2 ^4/24;
a(6,6) = 1.0;  a(6,7) = -l2;  a(6,8) = -l2^2/2;  a(6,10) = -1.0;
b(6,1) = -q*l2 ^3/6; a(7,7) = 1.0;  a(7,8) = l2;  a(7,11) = -1.0;
b(7,1) = q*l2^2/2;  a(8,3) = -1.0;  a(8,8) = 1.0;  b(8,1) = q*l2;
a(9,10) = l3;  a(9,11) = -l3^2/2;  a(9,12) = -l3^3/6;
b(9,1) = -f1*(l3-af1)^3/6;  a(10,10) = 1.0;  a(10,11) = -l3;
a(10,12) = -l1^2/2;  a(10,14) = -1.0;  b(10,1) = -f1*(l3-af1)^2/2;
a(11,11) = 1.0;  a(11,12) = l3;  a(11,15) = -1.0;  b(11,1) = f1*(l3-af1);
a(12,5) = -1.0;  a(12,12) = 1.0;  ;  b(12,1) = f1;  a(13,9) = -1.0;
a(13,14) = l4;  a(13,15) = -l4^2/2;  a(13,16) = -l4^3/6;
b(13,1) = -f2*(l4-af2)^3/6;  a(14,13) = -1.0;  a(14,14) = 1.0;
a(14,15) = -l4;   a(14,16) = -l4^2/2;  b(14,1) = -f2*(l4-af2)^2/2;
a(15,15) = -1.0;   a(15,16) = l4;  b(15,1) = f2*(l4-af2);  a(16,16) = 1.0;
b(16,1) = f2;
% Решение системы уравнений ax = b и вывод результатов
X = a\ b
```

Результаты вычисления граничных параметров представлены в таблице 3.1.

Таблица 3.1

1	$Q_{(1)}^{0-1} = X(1,1) = -1.0494$ кН	9	$EIn_{(1)}^{3-4} = X(9,1) = 105.2263$ кНм^3
2	$EIj_{(o)}^{0-1} = X(2,1) = -12.7984$ кНм^2	10	$EIj_{(o)}^{2-3} = X(10,1) = -53.7860$ кНм^2
3	$Q_{(1)}^{1-2} = X(3,1) = -26.9684$ кН	11	$M_{(o)}^{2-3} = X(11,1) = -6.0082$ кНм
4	$Q_{(o)}^{0-1} = X(4,1) = -1.0494$ кН	12	$Q_{(o)}^{2-3} = X(12,1) = -31.3306$ кН
5	$Q_{(1)}^{2-3} = X(5,1) = 8.6694$ кН	13	$EIj_{(1)}^{3-4} = X(13,1) = 115.2263$ кНм^2
6	$EIj_{(o)}^{1-2} = X(6,1) = 35.5967$ кНм^2	14	$EIj_{(o)}^{3-4} = X(14,1) = 85.2263$ кНм^2
7	$M_{(o)}^{1-2} = X(7,1) = -24.1975$ кНм	15	$M_{(o)}^{3-4} = X(15,1) = -60.0000$ кНм
8	$Q_{(o)}^{1-2} = X(8,1) = 33.0316$ кН	16	$Q_{(o)}^{3-4} = X(16,1) = 60.0000$ кН

Как видно из этого примера, не требуется переставлять строки матриц **A** и **B** для исключения нулевых ведущих элементов, что является дополнительным преимуществом MATLAB по сравнению с непосредственным программированием метода исключения Гаусса в таких средах, как Delphi, Visual Fortran, Visual C++ и т.п.

3.2.2. *Построение эпюр прогибов, углов поворота, поперечных сил и изгибающих моментов*

Эпюры параметров изгиба можно строить для каждого стержня в отдельности, используя соотношения метода начальных параметров. При этом удобно выполнять построение эпюр в одном окне, для чего можно привлекать процедуру subplot. Для соблюдения требуемого масштаба можно использовать процедуру axis. Масштабная сетка эпюр появится, если задействовать процедуру grid on. Выполнение этих требований приводит к следующим операторам построения эпюр

```
subplot (2, 2, 1), plot (x, EIν); axis ( [0  2  -150  150] ); grid on
subplot (2, 2, 2), plot (x, EIfi); axis ( [0  2  -150  150] ); grid on
subplot (2, 2, 3), plot (x, Q); axis ( [0  2  -100  100] ); grid on
subplot (2, 2, 4), plot (x, M); axis ( [0  2  -100  100] ); grid on.
```

Протокол построения эпюр удобно поместить в отдельный М-файл и по мере необходимости корректировать масштаб изображения графиков. Вы-

ражения для параметров изгиба стержня по методу начальных параметров имеют вид

$$
\begin{aligned}
&EIn(x) = EIn(o) + EIj(o)\cdot x - M(o)\cdot\frac{x^2}{2} - Q(o)\cdot\frac{x^3}{6} + B_{11}(x);\\
&EIj(x) = EIj(o) - M(o)\cdot x - Q(o)\cdot\frac{x^2}{2} + B_{21}(x);\\
&Q(x) = Q(o) - B_{41}(x);\\
&M(x) = M(o) + Q(o)\cdot x - B_{31}(x)\ ,
\end{aligned}
\tag{3.12}
$$

где элементы от внешней нагрузки запишутся следующим образом:

$$
\begin{aligned}
&B_{11}(x) = \underset{x>a_M}{\mathrm{I}}\frac{M(x-a_M)^2}{2} + \underset{x>a_F}{\mathrm{I}}\frac{F(x-a_F)^3}{6} + \underset{x>a_H}{\mathrm{I}}\frac{q(x-a_H)^4}{24} - \underset{x>a_K}{\mathrm{I}}\frac{q(x-a_K)^4}{24};\\
&B_{21}(x) = \underset{x>a_M}{\mathrm{I}}M\cdot(x-a_M) + \underset{x>a_F}{\mathrm{I}}\frac{F(x-a_F)^2}{2} + \underset{x>a_H}{\mathrm{I}}\frac{q(x-a_H)^3}{6} - \underset{x>a_K}{\mathrm{I}}\frac{q(x-a_K)^3}{6};\\
&B_{41}(x) = \underset{x>a_F}{\mathrm{I}}F + \underset{x>a_H}{\mathrm{I}}q\cdot(x-a_H) - \underset{x>a_K}{\mathrm{I}}q\cdot(x-a_K);\\
&B_{31}(x) = \underset{x>a_M}{\mathrm{I}}M + \underset{x>a_F}{\mathrm{I}}F(x-a_F) + \underset{x>a_H}{\mathrm{I}}\frac{q(x-a_H)^2}{2} - \underset{x>a_K}{\mathrm{I}}\frac{q(x-a_K)^2}{2};\quad \underset{x>a_m}{\mathrm{I}}\begin{cases}1, x>a_M;\\ 0, x\le a_M\end{cases}
\end{aligned}
\tag{3.13}
$$

Положительные направления внешней нагрузки и их координаты показаны на рис. 3.3

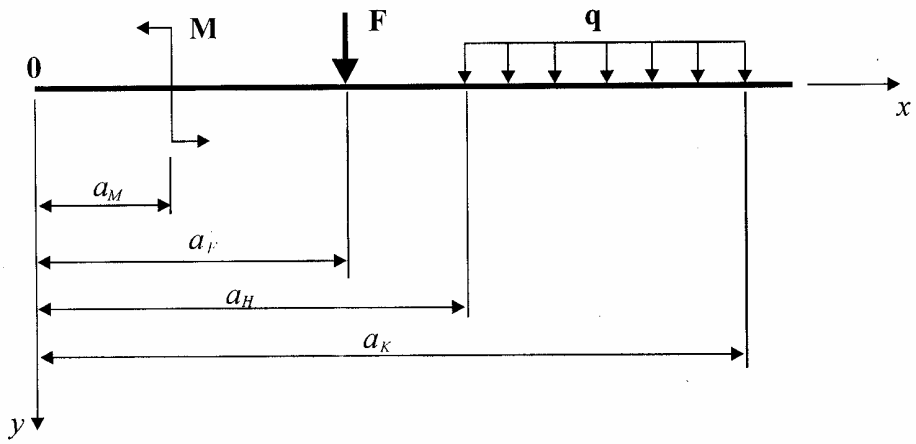


Рис.3.3

Для балки по рис. 3.1 протоколы построения эпюр примут вид:

<u>Стержень 0 – 1</u>

x = 0 : 0.001 : 2.0;

EI*n* = - (X(2,1).* – X(4,1).*x.^3/6);

EI*fi* = - (X(2,1) – X(4,1).*x.^2/2);

Q = X(4,1); M = X(4,1).*x;

subplot (2, 2, 1), plot (x, EIν); axis ([0 2 -150 150]); grid on

subplot (2, 2, 2), plot (x, EI*fi*); axis ([0 2 -150 150]); grid on

subplot (2, 2, 3), plot (x, Q); axis ([0 2 -100 100]); grid on

subplot (2, 2, 4), plot (x, M); axis ([0 2 -100 100]); grid on.

После построения графиков (нажатия кнопки ВЫПОЛНИТЬ) необходимо продолжить формирование команд для завершения эпюр на интервале (2; 4) в М-файле

x = 2.0 : 0.001 : 4.0; m = 20.0;

EI*n* = - (X(2,1).* – X(4,1).*x.^3/6 + m.*(x – 2).^2/2);

EI*fi* = - (X(2,1) – X(4,1).*x.^2/2 + m.*(x – 2));

Q = X(4,1); M = X(4,1).*x – m;

subplot (2, 2, 1), plot (x, EIν); axis ([2 4 -150 150]); grid on

subplot (2, 2, 2), plot (x, EI*fi*); axis ([2 4 -150 150]); grid on

subplot (2, 2, 3), plot (x, Q); axis ([2 4 -100 100]); grid on

subplot (2, 2, 4), plot (x, M); axis ([2 4 -100 100]); grid on.

<u>Стержень 1 – 2</u>

x = 0 : 0.001 : 6.0; q = 10.0;

EI*n* = - (X(6,1).*x – X(7,1).*x.^2/2 – X(8,1).*x.^3/6 + q.*x.^4/24);

EI*fi* = - (X(6,1) – X(7,1).*x – X(8.1).*x.^2/2 + q.*x.^3/6);

Q = X(8,1) – q.*x; M = X(7,1) + X(8,1).*x – q.*x.^2/2;

subplot (2, 2, 1), plot (x, EIν); axis ([0 6 -150 150]); grid on

subplot (2, 2, 2), plot (x, EI*fi*); axis ([0 6 -150 150]); grid on

subplot (2, 2, 3), plot (x, Q); axis ([0 6 -100 100]); grid on

subplot (2, 2, 4), plot (x, M); axis ([0 6 -100 100]); grid on.

<u>Стержень 2 – 3</u>

x = 0 : 0.001 : 2.0;

EI*n* = - (X(10,1).*x – X(11,1).*x.^2/2 – X(12,1).*x.^3/6);

EI*fi* = - (X(10,1) – X(11,1).*x – X(12,1).*x.^2/2);

Q = X(12,1); M = X(11,1) + X(12,1).*x;

subplot (2, 2, 1), plot (x, EIν); axis ([0 2 -150 150]); grid on

subplot (2, 2, 2), plot (x, EI*fi*); axis ([0 2 -150 150]); grid on

subplot (2, 2, 3), plot (x, Q); axis ([0 2 -100 100]); grid on

subplot (2, 2, 4), plot (x, M); axis ([0 2 -100 100]); grid on.

Далее вторая часть эпюр строится по операторам

x =2. 0 : 0.001 : 3.0;

EI*n* = - (X(10,1).*x – X(11,1).*x.^2/2 – X(12,1).*x.^3/6 – *f*1.*(x –2).^3/6);

EI*fi* = - (X(10,1) – X(11,1).*x – X(12,1).*x.^2/2 – *f*1.*(x –2).^2/2);

Q = X(12,1) + *f*1; M = X(11,1) + X(12,1).*x + *f*1.*(x –2);

subplot (2, 2, 1), plot (x, EIν); axis ([2 3 -150 150]); grid on

subplot (2, 2, 2), plot (x, EI*fi*); axis ([2 3 -150 150]); grid on

subplot (2, 2, 3), plot (x, Q); axis ([2 3 -100 100]); grid on

subplot (2, 2, 4), plot (x, M); axis ([2 3 -100 100]); grid on

<u>Стержень 3 – 4</u>

x = 0 : 0.001 : 1;

EI*n* = - (X(14,1).*x – X(15,1).*x.^2/2 – X(16,1).*x.^3/6);

EI*fi* = - (X(14,1) – X(15,1).*x– X(16,1).*x.^2/2);

Q = X(16,1); M = X(15,1) + X(16,1).*x;

subplot (2, 2, 1), plot (x, EIν); axis ([0 1 -150 150]); grid on

subplot (2, 2, 2), plot (x, EI*fi*); axis ([0 1 -150 150]); grid on

subplot (2, 2, 3), plot (x, Q); axis ([0 1 -100 100]); grid on

subplot (2, 2, 4), plot (x, M); axis ([0 1 -100 100]); grid on.

Следует помнить, что в окно команд необходимо вводить значения неизвестных начальных параметров балки, т. е. вектор X. Это легко выполняется с помощью отдельного М-файла, на котором записана процедура решения матричного уравнения неразрезной балки. После вектора X нужно ввести также значения внешней нагрузки и их координаты. Эпюры EI*n*(x), EI*j*(x), Q(x) и M(x) представлены на рис. 3.1.

3.2.3. *Определение частот собственных колебаний*

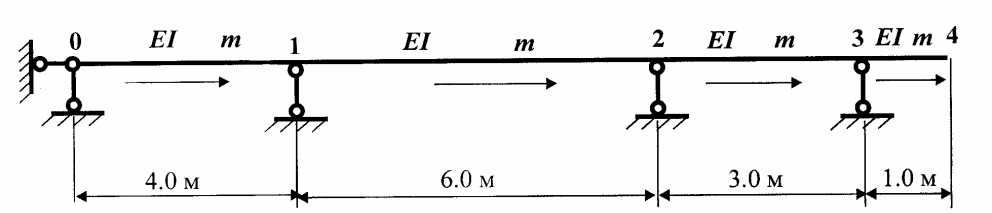


Рис. 3.4

Исходные данные: $\mathbf{l}_1^{0-1} = 4$ м, $\mathbf{l}_2^{1-2} = 6$ м, $\mathbf{l}_3^{2-3} = 3$ м, $\mathbf{l}_4^{3-4} = 1$ м,

Матрица фундаментальных функций поперечных колебаний

$$\mathbf{A_i} = \begin{bmatrix} A_{11} & A_{12} & -A_{13} & -A_{14} \\ l^4 A_{14} & A_{11} & -A_{12} & -A_{13} \\ -l^4 A_{13} & -l^4 A_{14} & A_{11} & A_{12} \\ -l^4 A_{12} & -l^4 A_{13} & l^4 A_{14} & A_{11} \end{bmatrix}; \quad l = \sqrt[4]{\frac{w^2 m}{EI}};\ A_{11}\frac{ch\,lx + \cos lx}{2};$$

$$A_{12} = \frac{sh\,lx + \sin lx}{2l};\ A_{13} = \frac{ch\,lx - \cos lx}{2l^2}; \qquad (3.14)$$

$$A_{14} = \frac{sh\,lx - \sin lx}{2l^3}.$$ В расчетах принимаем

$$m = EI = 1.$$

Частоты определяются как корни трансцендентного уравнения

$$|\mathbf{A}_*(\omega)| = 0,$$

где матрица $\mathbf{A}_*(\omega)$ берется из задачи статики с заменой матриц фундаментальных функций изгиба на матрицы функций поперечных колебаний

$\mathbf{A}_*(\omega)=$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1		A_{12}		$-A_{14}$													
2		A_{11}		$-A_{13}$		-1											
3		$-l^4A_{14}$		A_{12}			-1										0-1
4	-1	$-l^4A_{13}$		A_{11}													
5						A_{12}	$-A_{13}$	$-A_{14}$									
6						A_{11}	$-A_{12}$	$-A_{13}$		-1							
7						$-l^4A_{14}$	A_{11}	A_{12}			-1						1-2
8			-1			$-l^4A_{13}$	$-l^4A_{14}$	A_{11}									
9										A_{12}	$-A_{13}$	$-A_{14}$					
10										A_{11}	$-A_{12}$	$-A_{13}$		-1			
11										$-l^4A_{14}$	A_{11}	A_{12}			-1		2-3
12					-1					$-l^4A_{13}$	l^4A_{14}	A_{11}					
13									-1					A_{12}	$-A_{13}$	$-A_{14}$	
14													-1	A_{11}	$-A_{12}$	$-A_{13}$	
15														$-l^4A_{14}$	A_{11}	A_{12}	3-4
16														$-l^4A_{13}$	l^4A_{14}	A_{11}	

(3.15)

Наиболее просто корни могут быть найдены методом перебора в сочетании с процедурой вычисления определителя матрица $\mathbf{A}_*(\omega)$. Организуется цикл вычисления определителя $|\mathbf{A}_*(\omega)|$, значения которого вместе с частотой ω выводятся на печать в формате ***long e***. При просмотре таблицы их значений определяются точки, где определитель *изменяет знак*. Эти точки и будут являться частотами собственных колебаний. Иногда бывают случаи, когда график функции $d = |\mathbf{A}_*(\omega)|$ касается оси ω. Для контроля этих точек полезно построить график $y = d(\omega)$, просмотр которого позволяет определить интервалы, где находятся корни и существуют ли точки касания графика оси ω. Далее корни могут быть уточнены повторными прогонами программы (без построения графика $y = d(\omega)$. Рекомендуется начинать поиск корней с начального значения частоты $w_0 = 0.01\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}$, с шагом $\Delta w_0 = 0.01\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}$. Число вычислений n = 300 – 500 позволяет надежно определить первую частоту, старшие частоты определяются при расширении интервалов вычисления определителя $|\mathbf{A}_*(\omega)|$. При решении данной задачи принимается, что l=EI=m=1.

<u>Обозначения переменных, принятых в программе</u>

a – матрица $\mathbf{A}_*(\omega)$; n – порядок матрицы $\mathbf{A}_*(\omega)$; n_1 – число циклов вычисления определителя $|\mathbf{A}_*(\omega)|$; *am* – частота собственных колебаний; *dam* – шаг изменения частоты ω; *m* – счетчик цикла вычисления определителя; *l*1, *l*2, *l*3, *l*4 – длины стержней неразрезной балки; *la* – коэффициент λ фундаментальных функций; X – вектор значений частоты ω; Y – вектор значений определителя $|\mathbf{A}_*(\omega)|$.

<u>Текст программы</u>

```
l1 = 4.0;  l2 = 6.0;  l3 = 3.0;  l4 = 1.0;  n = 16;  n1 = 300;  am = 0,01;
dam = 0.01;  X = zeros (n1,1);  Y = zeros (n1,1);
for  m = 1 : n1
la = sqrt (am);  a = zeros (n,n);
a(1,2) = (sinh(la*l1) + sin (la*l1))/(2*la);
```

a(1,4) = – (*si h*(*la***l*1) - *sin* (*la***l*1))/(2**la*^3);

a(2,2)=(*cosh*(*la***l*1)+*cos*(*la***l*1))/2; *a*(2,4) = – (*cos h*(*la***l*1) - *cos* (*la***l*1))/(2**la*^2);

a(2,6) = – 1; *a*(3,2) = *la*^ 4**a*(1,4); *a*(3,4) = *a*(1,2); *a*(3,7) = – 1; *a*(4,1) = – 1;

a(4,2) = *la*^ 4**a*(2,4); *a*(4,4) = *a*(2,2);

a(5,6) = (*sinh*(*la***l*2) + *sin* (*la***l*2))/(2**la*);

a(5,7)= – (*cosh*(*la***l*2) –*cos*(*la***l*2))/(2**la*^2);

a(5,8) = – (*sinh*(*la***l*2) – *sin* (*la***l*2))/(2**la*^ 3);

a(6,6)=(*cosh*(*la***l*2)+*cos*(*la***l*2))/2; *a*(6,7) = – *a*(5,6); *a*(6,8) = *a*(5,7);

a(6,10) = – 1; *a*(7,6) = *la*^ 4**a*(5,8); *a*(7,7) = *a*(6,6); *a*(7,8) = *a*(5,6); *a*(7,11) = – 1;

a(8,3) = – 1; *a*(8,6) = *la*^ 4**a*(5,7); *a*(8,7) = – *a*(7,6); *a*(8,8) = *a*(6,6);

a(9,10) = (*sinh*(*la***l*3) + *sin* (*la***l*3))/(2**la*);

a(9,11)= – (*cosh*(*la***l*3) –*cos*(*la***l*3))/(2**la*^2);

a(9,12) = – (*sinh*(*la***l*3) – *sin* (*la***l*3))/(2**la*^ 3);

a(10,10)=(*cosh*(*la***l*3)+*cos*(*la***l*3))/2; *a*(10,11) = – *a*(9,10); *a*(10,12) = *a*(9,11);

a(10,14) = – 1; *a*(11,10) = *la*^ 4**a*(9,12); *a*(11,11) = *a*(10,10); *a*(11,12) = *a*(9,10);

a(11,15) = – 1; *a*(12,5) = – 1; *a*(12,10) = *la*^ 4**a*(9,11); *a*(12,11) = – *a*(11,10);

a(12,12) = *a*(10,10); *a*(13,9) = – 1; *a*(13,14) = (*sin h*(*la***l*4) + *sin* (*la***l*4))/(2**la*);

a(13,15)= – (*cosh*(*la***l*4) –*cos*(*la***l*4))/(2**la*^2);

a(13,16) = – (*sinh*(*la***l*4) – *sin* (*la***l*4))/(2**la*^ 3); *a*(14,13) = – 1;

a(14,14)=(*cosh*(*la***l*4)+*cos*(*la***l*4))/2; *a*(15,14) = *la*^ 4**a*(13,16);

a(15,15) = *a*(14,14); *a*(15,16)= *a*(13,14); *a*(16,14) = *la*^ 4**a*(13,15);

a(16,15) = – *a*(15,14); *a*(16,16) = *a*(14,16);

d = *det*(*a*); *X*(*m*,1) = *am*; *Y*(*m*,1) = *d*; *am* = *am* + *dam*; *end*;

% Построение графика *y* = *d*(*am*)

plot (X,Y); *grid on*

% Вывод на печать таблицы значений *am* и *d*

format long e

[X Y]

График определителя d = | A*(ω) | для балки принимает вид (рис. 3.5)

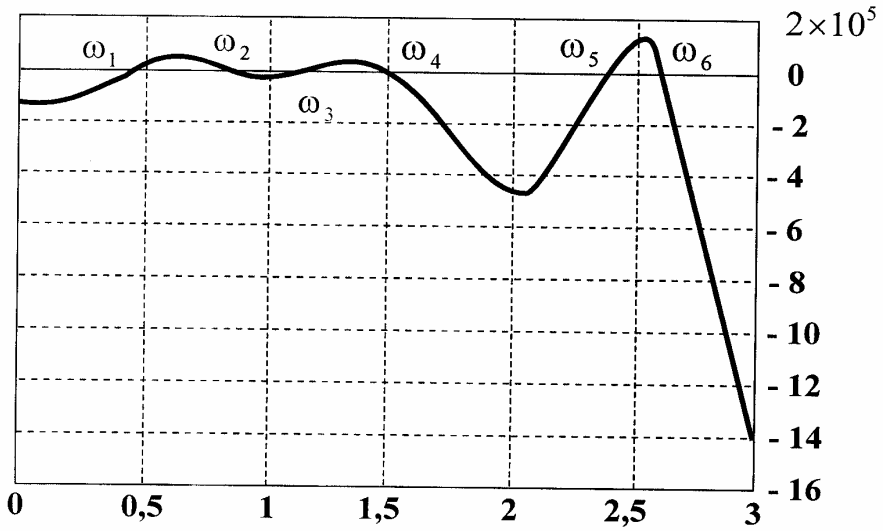


Рис. 3.5

Как следует из рис. 3.5, график определителя d = | A*(ω) | не имеет точек разрыва 2-го рода (это большое преимущество метода граничных элементов), не касается оси *w* и имеет точки пересечения с горизонталью. Эти точки отмечены символами w_1, w_2 и т. д. Уточнение частот приводит к следующим значениям

$$w_1 = 0.4055125 l^{-2}\sqrt{\frac{EI}{m}};\quad w_2 = 0.7818475 l^{-2}\sqrt{\frac{EI}{m}};\quad w_3 = 1.1001125 l^{-2}\sqrt{\frac{EI}{m}};$$

$$w_4 = 1.4852935 l^{-2}\sqrt{\frac{EI}{m}};\quad w_5 = 2.4234325 l^{-2}\sqrt{\frac{EI}{m}}\quad \frac{1}{сек}$$

и т. д.

3.2.4. *Построение форм собственных колебаний*

Для построения форм собственных колебаний необходимо определить граничные параметры стержней балки. Эта задача может быть решена следующим образом:

1. Используется *статический* или *кинематический* способ возбуждения собственных колебаний. Для этого *приравнивается единице* статический или кинематический параметр матрицы **Y**(**l**), равный нулю. Далее этот параметр переносится в матрицу нагрузки **B**. Из системы уравнений (3.11) следует, что наиболее удобным параметром является $Q^{3\text{-}4}$(**l**) = 1, тогда в (16,1) = 1.

$$
\begin{array}{c|cccccccccccccccc|c|c}
 & & & & & & & \mathbf{A_*(\omega)} & & & & & & & & & & \mathbf{X_*} & \mathbf{B} \\
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & & \\
\hline
1 & & A_{12} & & -A_{14} & & & & & & & & & & & & & Q^{0-1}(\ell) & \\
2 & & A_{11} & & -A_{13} & & -1 & & & & & & & & & & & EI\varphi^{0-1}(o) & \\
3 & & -\lambda^4 A_{14} & & A_{12} & & & -1 & & & & & & & & & & Q^{1-2}(\ell) & \\
4 & -1 & -\lambda^4 A_{13} & & A_{11} & & & & & & & & & & & & & Q^{0-1}(o) & \\
\hline
5 & & & & & & A_{12} & -A_{13} & -A_{14} & & & & & & & & & Q^{2-3}(\ell) & \\
6 & & & & & & A_{11} & -A_{12} & -A_{13} & & -1 & & & & & & & EI\varphi^{1-2}(o) & \\
7 & & & & & & -\lambda^4 A_{14} & A_{11} & A_{12} & & & -1 & & & & & & M^{1-2}(o) & \\
8 & & & -1 & & & -\lambda^4 A_{13} & \lambda^4 A_{14} & A_{11} & & & & & & & & & Q^{1-2}(o) & \\
\hline
9 & & & & & & & & & & A_{12} & -A_{13} & -A_{14} & & & & & EI\vartheta^{3-4}(\ell) & = \\
10 & & & & & & & & & & A_{11} & -A_{12} & -A_{13} & & -1 & & & EI\varphi^{2-3}(o) & \\
11 & & & & & & & & & & -\lambda^4 A_{14} & A_{11} & A_{12} & & & -1 & & M^{2-3}(o) & \\
12 & & & & & -1 & & & & & -\lambda^4 A_{13} & \lambda^4 A_{14} & A_{11} & & & & & Q^{2-3}(o) & \\
\hline
13 & & & & & & & & & -1 & & & & & A_{12} & -A_{13} & -A_{14} & EI\varphi^{3-4}(\ell) & \\
14 & & & & & & & & & & & & & -1 & A_{11} & -A_{12} & -A_{13} & EI\varphi^{3-4}(o) & \\
15 & & & & & & & & & & & & & & -\lambda^4 A_{14} & A_{11} & A_{12} & M^{3-4}(o) & \\
16 & & & & & & & & & & & & & & -\lambda^4 A_{13} & \lambda^4 A_{14} & A_{11} & Q^{3-4}(o) & Q^{3-4}(\ell) = 1
\end{array}
\tag{3.16}
$$

Решением линейной системы уравнений (при собственной частоте w_I) определяются граничные параметры балки, которые нужно нормировать относительно какого-либо перемещения. Для данной балки в качестве нормирующей величины можно взять перемещение консольной части

$$EIn^{3\text{-}4}(\mathbf{l}) = X(9,1).$$

2. По нормированным начальным параметрам строятся формы собственных колебаний балки средствами MATLAB.

Уравнение для граничных параметров балки отличается от уравнения (3.11) только фундаментальными функциями и вектором правой части (см. (3.16)).

Программа вычисления граничных параметров набирается в отдельном М-файле и принимает вид

```
a = zeros (16,16);  b = zeros (16,1); X = zeros (16,1); la = sqrt (0,4055125);
l1 = 4.0;  l2 = 6.0;  l3 = 3.0;  l4 = 1.0;
a(1,2) = (sinh(la*l1) + sin (la*l1))/(2*la);
a(1,4) = – (sinh(la*l1) – sin (la*l1))/(2*la^3);
a(2,2)=(cosh(la*l1)+cos(la*l1))/2; a(2,4) = – (cosh(la*l1) - cos (la*l1))/(2*la^2);
a(2,6) = – 1;  a(3,2) = la^ 4*a(1,4); a(3,4) = a(1,2); a(3,7) = – 1; a(4,1) = – 1;
a(4,2) = la^ 4*a(2,4); a(4,4) = a(2,2); a(5,6) = (sinh(la*l2) + sin (la*l2))/(2*la);
a(5,7)= – (cosh(la*l2) – cos(la*l2))/(2*la^2);
a(5,8) = – (sinh(la*l2) – sin (la*l2))/(2*la^ 3); a(6,6)=(cosh(la*l2)+cos(la*l2))/2;
a(6,7) = –a(5,6); a(6,8) =a(5,7); a(6,10) = –1; a(7,6) =la^ 4*a(5,8); a(7,7) =a(6,6);
a(7,8) = a(5,6); a(7,11) = – 1; a(8,3) = – 1; a(8,6) = la^ 4*a(5,7); a(8,7) = – a(7,6);
a(8,8) = a(6,6); a(9,10) = (sinh(la*l3) + sin (la*l3))/(2*la);
a(9,11)= – (cosh(la*l3) –cos(la*l3))/(2*la^2);
a(9,12) =–(sinh(la*l3)–sin(la*l3))/(2*la^3); a(10,10)=(cosh(la*l3)+cos(la*l3))/2;
a(10,11) = – a(9,10); a(10,12) = a(9,11); a(10,14) = – 1;  a(11,10) = la^ 4*a(9,12);
a(11,11) =a(10,10); a(11,12)= a(9,10); a(11,15)= –1; a(12,5)= –1; a(12,11) = – a(11,10);
a(12,10)=la^ 4*a(9,11); a(12,12)=a(10,10); a(13,14)=(sinh(la*l4) + sin (la*l4))/(2*la);
```

$a(13,9) = -1$; $a(13,15) = -(\cos h(la*l4) - \cos(la*l4))/(2*la\wedge 2)$;

$a(13,16) = -(\sinh(la*l4) - \sin(la*l4))/(2*la\wedge 3)$; $a(14,13) = -1$;

$a(14,14)=(\cosh(la*l4)+\cos(la*l4))/2$; $a(14,15) = -a(13,14)$; $a(14,16) = a(13,15)$;

$a(15,14) = la\wedge 4*a(13,16)$; $a(15,15) = a(14,14)$; $a(15,16) = a(13,14)$; $a(16,14) = la\wedge 4*a(13,15)$;

$a(16,15) = -a(15,14)$; $a(16,16) = a(14,14)$; $b(16,1) = 1$; X = $a\backslash b$;

X = X \ X(9,1)

Результаты вычислений относительных значений граничных параметров балки при собственных частотах сведены в таблицу 3.2.

Таблица 3.2

№ п/п	Граничные параметры балки	Относительные значения граничных параметров при частотах				
		w_1=0,4055125	w_2=0,7818475	w_3=1,1001125	w_4=1,4852935	w_5=2,4234325
1	$Q^{0-1}(\mathbf{l}) = X(1,1) =$	0.1677	- 3.5779	0.3375	- 0.4569	- 0.9724
2	$EIj^{0-1}(o) = X(2,1) =$	- 1.4419	3.5754	- 0.2308	0.3784	- 0.4260
3	$Q^{1-2}(o) = X(3,1) =$	- 1.6643	- 1.3379	- 0.0981	1.8113	0.9831
4	$Q^{0-1}(o) = X(4,1) =$	- 0.6979	2.6728	- 0.2409	0.5453	- 1.0318
5	$Q^{2-3}(\mathbf{l}) = X(5,1) =$	0.6626	0.6263	0.5250	0.2441	- 1.6235
6	$EIj^{1-2}(o) = X(6,1) =$	2.2068	- 1.8784	- 0.0844	0.4282	- 0.4491
7	$M^{1-2}(o) = X(7,1) =$	- 1.1280	- 2.3814	0.4101	- 0.8967	0.0745
8	$Q^{1-2}(o) = X(8,1) =$	1.5705	0.6324	- 0.5215	1.7308	- 1.2045
9	$EIu^{3-4}(\mathbf{l}) = X(9,1) =$	1.0000	1.0000	1.0000	1.0000	1.0000
10	$EIj^{2-3}(o) = X(10,1) =$	- 1.7388	- 1.1200	- 0.4732	0.2324	0.5331
11	$M^{2-3}(o) = X(11,1) =$	- 1.5543	- 0.5464	0.4042	1.2018	- 0.1983
12	$Q^{2-3}(o) = X(12,1) =$	0.3185	- 0.4130	- 0.9820	- 1.1665	1.5478
13	$EIj^{3-4}(\mathbf{l}) = X(13,1) =$	1.0055	1.0203	1.0400	1.0723	1.1873
14	$EIj^{3-4}(o) = X(14,1) =$	0.9850	0.9443	0.8905	0.8024	0.4932
15	$M^{3-4}(o) = X(15,1) =$	- 0.0547	- 0.2021	- 0.3971	- 0.7145	- 1.8141
16	$Q^{3-4}(o) = X(16,1) =$	0.0819	0.3020	0.5910	1.0566	2.6174

Выражение для прогиба (формы колебания) стержня при собственных колебаниях в соответствии с методом начальных параметров имеет вид

$$EIJ(x) = -\left(EIJ(o)\frac{ch\,lx + \cos lx}{2} + EIj(o)\frac{sh\,lx + \sin lx}{2l} - \right.$$
$$\left. - M(o)\frac{ch\,lx - \cos lx}{2l^2} - Q(o)\frac{sh\,lx - \sin lx}{2l^3} \right). \qquad (3.17)$$

Протокол построения форм колебаний балки принимает вид

```
x1 = 0 : 0.001 : 4.0;  x2 = 0 : 0.001 : 6.0;  x3 = 0 : 0.001 : 3.0;
x4 = 0 : 0.001 : 1.0;  la = sqrt (0.4055125);
EIv1 = – (X(2,1)*(sinh(la*x1)+sin(la*x1))/(2*la) – …
X(4,1)*(sinh(la*x1)–sin(la*x1))/(2*la^3));
EIv2 = – (X(6,1)*(sinh(la*x2)+sin(la*x2))/(2*la) – …
X(7,1)*(cosh(la*x2)–cos(la*x2))/(2*la^2) – …
X(8,1)*(sinh(la*x2)–sin(la*x2))/(2*la^3));
EIv3 = – (X(10,1)*(sinh(la*x3)+sin(la*x3))/(2*la) – …
X(11,1)*(cosh(la*x3)–cos(la*x3))/(2*la^2) – …
X(12,1)*(sinh(la*x3)–sin(la*x3))/(2*la^3));
EIv4 = – (X(14,1)*(sinh(la*x4)+sin(la*x4))/(2*la) – …
X(15,1)*(cosh(la*x4)–cos(la*x4))/(2*la^2) – …
X(16,1)*(sinh(la*x4)–sin(la*x4))/(2*la^3));
subplot (2,2,1), plot (x1, EIv1); axis ([0 4 – 0.5 0.5]); grid on
subplot (2,2,2), plot (x2, EIv2); axis ([0 6 – 0.6 0.6]); grid on
subplot (2,2,3), plot (x3, EIv3); axis ([0 3 – 0.5 0.5]); grid on
subplot (2,2,4), plot (x4, EIv4); axis ([0 1 – 1    1   ]); grid on
```

Перед выполнением данного протокола из отдельного М-файла в окно команд помещается вектор X – вектор значений граничных параметров балки при w_i . Выполняется это прогонкой программы, решающей систему уравнений (3.16). Далее отрабатывается протокол построения форм колебаний при собственной частоте w_i . Первые 5 форм собственных колебаний балки при

условии, что $EIJ_{(1)}^{3-4}=1$, представлены на рис 3.6. Можно рекомендовать также объединить протоколы вычисления вектора X и построения форм колебаний, но только после четкого усвоения навыка построения графиков и выбора их масштабов.

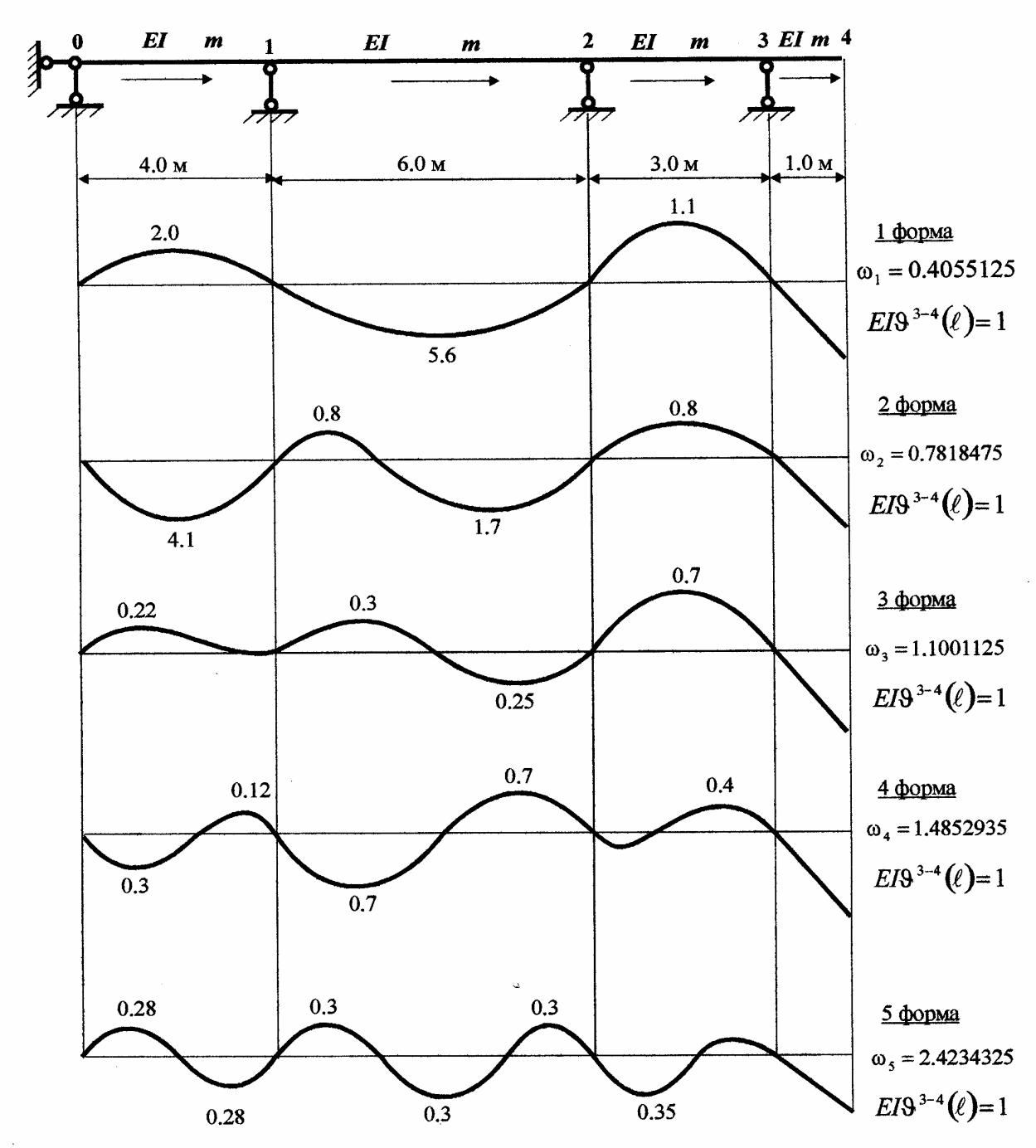


Рис. 3.6

Добавим, что аналогично строятся формы колебаний для упругих систем, у которых узлы имеют только угловые перемещения. В этом случае гра-

ничные параметры могут нормироваться относительно угла поворота какого-либо граничного сечения.

3.2.5. *Расчет на вынужденные колебания*

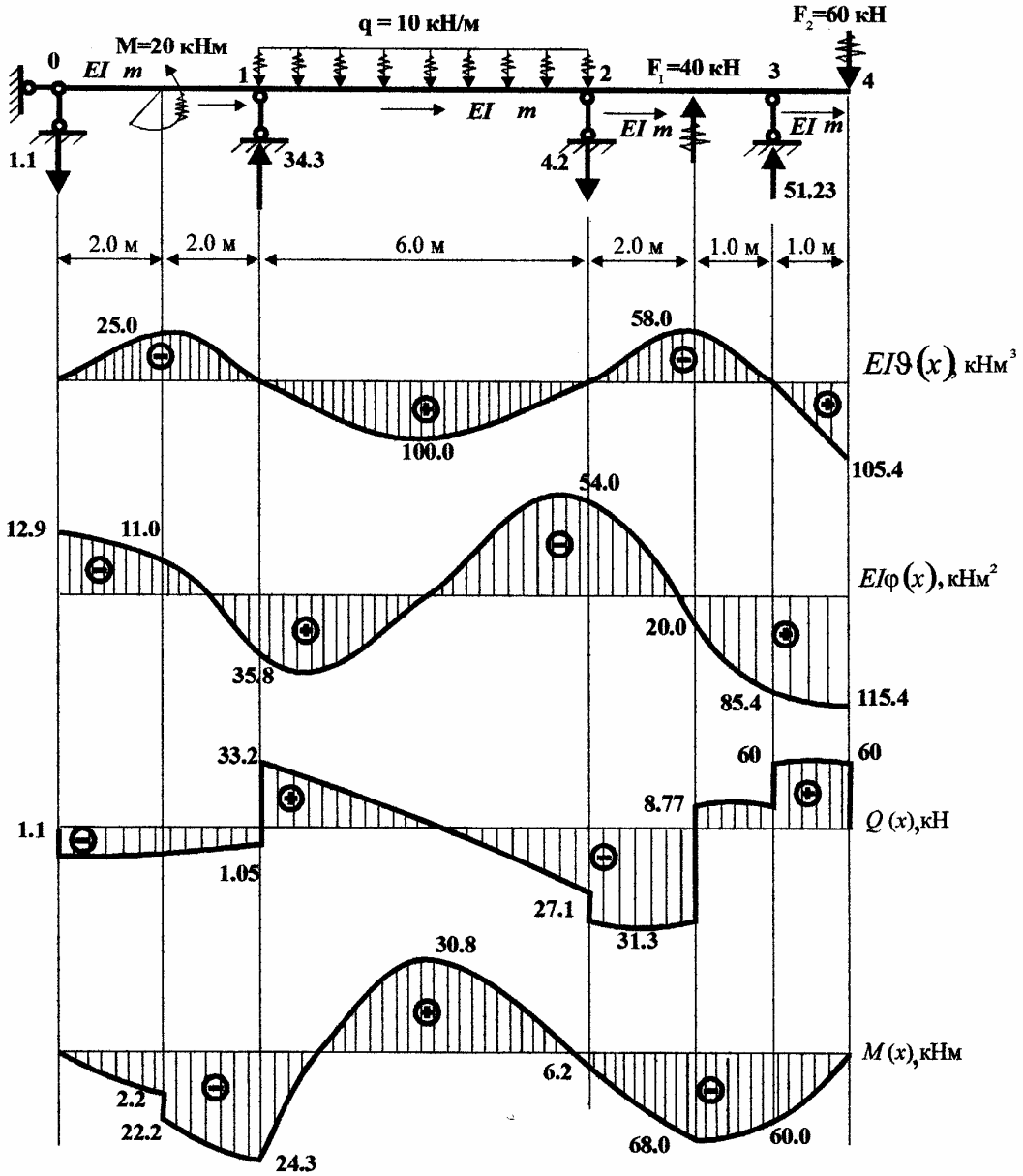


Рис. 3.7

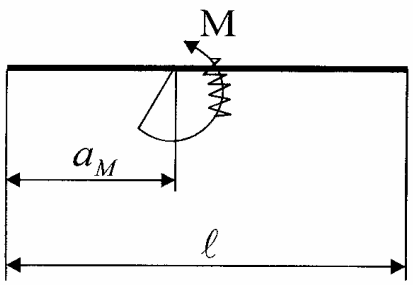


Рис. 3.8

$$b(1,1) = -M\,\frac{ch\,l\,(1 - a_M) - \cos l\,(1 - a_M)}{2\,l^2};$$

$$b(2,1) = -M\,\frac{sh\,l\,(1 - a_M) + \sin l\,(1 - a_M)}{2\,l};$$

$$b(3,1) = M\,\frac{ch\,l\,(1 - a_M) + \cos l\,(1 - a_M)}{2}; \qquad (3.18)$$

$$b(4,1) = M\,\frac{l\,(sh\,l\,(1 - a_M) - \sin l\,(1 - a_M))}{2}.$$

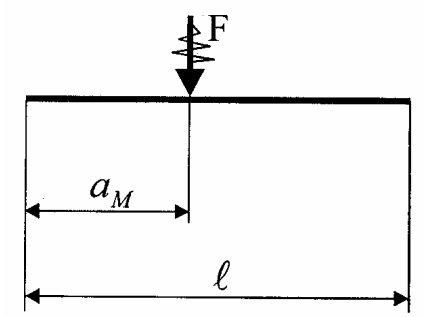


Рис. 3.9

$l = \sqrt{w}$

$$b(1,1) = -F\frac{sh\,l(\mathbf{l}-a_F) - \sin l(\mathbf{l}-a_F)}{2l^3};$$
$$b(2,1) = -F\frac{ch\,l(\mathbf{l}-a_F) - \cos l(\mathbf{l}-a_F)}{2l^2};$$
$$b(3,1) = F\frac{sh\,l(\mathbf{l}-a_F) + \sin l(\mathbf{l}-a_F)}{2l};$$
$$b(4,1) = F\frac{ch\,l(\mathbf{l}-a_F) + \cos l(\mathbf{l}-a_F)}{2}. \qquad (3.19)$$

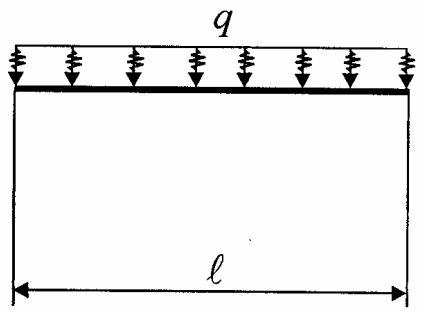


Рис. 3.10

$$b(1,1) = -q\frac{ch\,l\mathbf{l} + \cos l\mathbf{l} - 2}{2l^4};$$
$$b(2,1) = -q\frac{sh\,l\mathbf{l} - \sin l\mathbf{l}}{2l^3};$$
$$b(3,1) = q\frac{ch\,l\mathbf{l} - \cos l\mathbf{l}}{2l^2}; \qquad (3.20)$$
$$b(4,1) = q\frac{sh\,l\mathbf{l} + \sin l\mathbf{l}}{2l}.$$

Расчет балки на вынужденные колебания сводится к решению матричного уравнения $\mathbf{A}_*\mathbf{X}_* = -\mathbf{B}$, где матрица **A** повторяет матрицу частотного уравнения (см. п.3.2.3), а матрица **B** представлена в этом разделе. Программа также записывается в отдельном М-файле, куда переносится матрица $\mathbf{A}_*$ задачи определения частот собственных колебаний и дополняется элементами вектора **B**. Сама программа имеет вид.

```
a = zeros (16,16);  b = zeros (16,1); X = zeros (16,1); la = sqrt (0.1*0,315365);
l1 = 4.0;  l2 = 6.0;  l3 = 3.0;  l4 = 1.0; m = 20.0; q = 10.0; f1 = - 40.0;
f2 = 60.0; am = 2.0; af1 = 2.0; af2 = 1.0;
a(1,2) = (sinh(la*l1) + sin (la*l1))/(2*la);
a(1,4) = – (sinh(la*l1) – sin (la*l1))/(2*la^3);
a(2,2) = (cosh(la*l1)+cos(la*l1))/2;
a(2,4) = – (cosh(la*l1) – cos (la*l1))/(2*la^2); a(2,6) = – 1;
a(3,2) = la^ 4*a(1,4); a(3,4) = a(1,2); a(3,7) = – 1; a(4,1) = – 1;
a(4,2) = la^ 4*a(2,4); a(4,4) = a(2,2);
b(1,1) = – m* (cosh(la*(l1–am))– cos (la*(l1–am))) / (2*la^2);
```

$b(2,1) = -m*(sinh(la*(l1-am)) + sin(la*(l1-am)))/(2*la);$

$b(3,1) = m*(cosh(la*(l1-am)) + cos(la*(l1-am)))/2;$

$b(4,1) = m*la*(sinh(la*(l1-am)) - sin(la*(l1-am)))/2;$

$a(5,6) = (sinh(la*l2) + sin(la*l2))/(2*la);$

$a(5,7)= -(cosh(la*l2) - cos(la*l2))/(2*la^2);$

$a(5,8) = -(sinh(la*l2) - sin(la*l2))/(2*la^3);$

$a(6,6)=(cosh(la*l2)+cos(la*l2))/2;$ $a(6,7) = -a(5,6);$ $a(6,8) =a(5,7);$

$a(6,10) = -1;$ $a(7,6) =la^4*a(5,8);$ $a(7,7) =a(6,6);$ $a(7,8) = a(5,6);$

$a(7,11) = -1;$ $a(8,3) = -1;$ $a(8,6) = la^4*a(5,7);$ $a(8,7) = -a(7,6);$

$a(8,8) = a(6,6);$

$b(5,1) = -q*(cosh(la*l2) + cos(la*l2) - 2)/(2*la^4);$

$b(6,1) = -q*(sinh(la*l2) - sin(la*l2))/(2*la^3);$

$b(7,1) = q*(cosh(la*l2) - cos(la*l2))/(2*la^2);$

$b(8,1) = q*(sinh(la*l2) + sin(la*l2))/(2*la);$

$a(9,10) = (sinh(la*l3) + sin(la*l3))/(2*la);$

$a(9,11)= -(cosh(la*l3) -cos(la*l3))/(2*la^2);$

$a(9,12) =-(sinh(la*l3)-sin(la*l3))/(2*la^3);$

$a(10,10)=(cosh(la*l3)+cos(la*l3))/2;$ $a(10,11) = -a(9,10);$ $a(10,12) = a(9,11);$

$a(10,14) = -1;$ $a(11,10) = la^4*a(9,12);$ $a(11,11) =a(10,10);$ $a(11,12)= a(9,10);$

$a(11,15)= -1;$ $a(12,5)= -1;$ $a(12,11) = -a(11,10);$ $a(12,10)=la^4*a(9,11);$

$a(12,12)=a(10,10);$

$b(9,1) = -f1*(sinh(la*(l3-a\ f1))- sin(la*(l3-a\ f1)))/(2*la^3);$

$b(10,1) = -f1*(cosh(la*(l3-a\ f1)) - cos(la*(l3-a\ f1)))/(2*la^2);$

$b(11,1) = f1*(sinh(la*(l3-a\ f1)) + sin(la*(l3-a\ f1)))/(2*la);$

$b(12,1) = f1*(cosh(la*(l3-a\ f1)) + cos(la*(l3-a\ f1)))/2;$ $a(13,9) = -1;$

$a(13,14)=(sinh(la*l4) + sin(la*l4))/(2*la);$

$a(13,15)= -(cosh(la*l4) -cos(la*l4))/(2*la^2);$

$a(13,16) = -(sinh(la*l4) - sin(la*l4))/(2*la^3);$ $a(14,13) = -1;$

$a(14,14)=(cosh(la*l4)+cos(la*l4))/2$; $a(14,15) = - a(13,14)$; $a(14,16) = a(13,15)$;

$a(15,14) = la\wedge 4*a(13,16)$; $a(15,15) = a(14,14)$; $a(15,16) = a(13,14)$;

$a(16,14) = la\wedge 4*a(13,15)$; $a(16,15) = – a(15,14)$; $a(16,16) = a(14,14)$;

$b(13,1) = – f2* (sinh(la*(l4–a\ f2))– sin\ (la*(l4–a\ f2))) / (2*la\wedge 3)$;

$b(14,1) = – f2* (cosh(la*\ l4–a\ f2)) – cos\ (la*(l4–a\ f2))) / (2*la\wedge 2)$;

$b(15,1) = f2* (sinh(la*(l4–a\ f2)) + cos\ (la*(l4–a\ f2))) / (2*la)$;

$b(16,1) = f2* (cosh(la*(l4–a\ f2)) + cos\ (la*(l4–a\ f2))) / 2$;

X = $a\backslash b$

Для ухода от резонанса частоту вынужденных колебаний принимаем

$$q = 0.1 \cdot 0.315365 \mathbf{l}^{-2} \sqrt{\frac{EI}{m}} \ \frac{1}{сек}.$$

Тогда

$$l = \sqrt{0.1 \cdot 0.315365} \ \frac{1}{м} \ .$$

<u>Результаты расчета динамических граничных параметров балки</u>

Таблица 3.3

X =

1	X(1,1) = $Q^{0-1}(\mathbf{l})$ = – 1.0486 кН
2	X(2,1) = $EIj^{0-1}(o)$ = – 12.9428 кНм^2
3	X(3,1) = $Q^{1-2}(\mathbf{l})$ = – 27.1657 кН
4	X(4,1) = $Q^{0-1}(o)$ = – 1.1133 кН
5	X(5,1) = $Q^{2-3}(\mathbf{l})$ = 8.7703 кН
6	X(6,1) = $EIj^{1-2}(o)$ = 35.8382 кНм^2
7	X(7,1) = $M^{1-2}(o)$ = – 24.3387 кНм
8	X(8,1) = $Q^{1-2}(o)$ = 33.2107 кН
9	X(9,1) = $EIu^{3-4}(\mathbf{l})$ = 105.4015 кНм^3
10	X(10,1) = $EIj^{2-3}(o)$ = – 54.0152 кНм^2
11	X(11,1) = $M^{2-3}(o)$ = – 6.1711 кНм
12	X(12,1) = $Q^{2-3}(o)$ = – 31.3403 кН
13	X(13,1) = $EIj^{3-4}(\mathbf{l})$ = 115.4049 кНм^2
14	X(14,1) = $EIj^{3-4}(o)$ = 85.3921 кНм^2
15	X(15,1) = $M^{3-4}(o)$ = – 60.0338 кНм
16	X(16,1) = $Q^{3-4}(o)$ = 60.0499 кН

3.2.6. *Построение эпюр напряженно-деформированного состояния балки при вынужденных колебаниях*

Эпюры напряженно-деформированного состояния можно построить для каждого стержня в отдельности, используя соотношения метода начальных параметров.

$$
\begin{aligned}
&EIJ(x) = -\left(EIJ(o)\frac{ch\,lx + \cos lx}{2} + EIj(o)\frac{sh\,lx + \sin lx}{2l} - \right.\\
&\left. - M(o)\frac{ch\,lx - \cos lx}{2l^2} - Q(o)\frac{sh\,lx - \sin lx}{2l^3} + b(1{,}1) \right);\\
&EIj(x) = -\left(EIJ(o)\cdot l^4 \cdot \frac{sh\,lx - \sin lx}{2l^3} + EIj(o)\frac{ch\,lx + \cos lx}{2} - \right.\\
&\left. - M(o)\frac{sh\,lx + \sin lx}{2l} - Q(o)\frac{ch\,lx - \cos lx}{2l^2} + b(2{,}1) \right);\\
&M(x) = -EIJ(o)\cdot l^4 \cdot \frac{ch\,lx - \cos lx}{2l^2} - EIj(o)\cdot l^4 \cdot \frac{sh\,lx - \sin lx}{2l^3} +\\
&+ M(o)\frac{ch\,lx + \cos lx}{2} + Q(o)\frac{sh\,lx + \sin lx}{2l} - b(3{,}1);\\
&Q(x) = -EIJ(o)\cdot l^4 \cdot \frac{sh\,lx + \sin lx}{2l} - EIj(o)\cdot l^4 \cdot \frac{ch\,lx - \cos lx}{2l^2} +\\
&+ M(o)\cdot l^4 \cdot \frac{sh\,lx - \sin lx}{2l^3} + Q(o)\frac{ch\,lx + \cos lx}{2} - b(4{,}1);
\end{aligned}
\tag{3.21}
$$

где выражения от внешней нагрузки имеют вид

$$
\begin{aligned}
&b(1{,}1) = \mathop{\mathrm{I}}_{x>a_M} M\frac{ch\,l(x-a_M) - \cos l(x-a_M)}{2l^2} + \mathop{\mathrm{I}}_{x>a_F} F\frac{sh\,l(x-a_F) - \sin l(x-a_F)}{2l^3} +\\
&\mathop{\mathrm{I}}_{x>a_H} q\frac{ch\,l(x-a_H) + \cos l(x-a_H) - 2}{2l^4} - \mathop{\mathrm{I}}_{x>a_K} q\frac{ch\,l(x-a_K) + \cos l(x-a_K) - 2}{2l^4};\\
&b(2{,}1) = \mathop{\mathrm{I}}_{x>a_M} M\frac{sh\,l(x-a_M) + \sin l(x-a_M)}{2l} + \mathop{\mathrm{I}}_{x>a_F} F\frac{ch\,l(x-a_F) - \cos l(x-a_F)}{2l^2} +\\
&\mathop{\mathrm{I}}_{x>a_H} q\frac{sh\,l(x-a_H) - \sin l(x-a_H)}{2l^3} - \mathop{\mathrm{I}}_{x>a_K} q\frac{sh\,l(x-a_K) - \sin l(x-a_K)}{2l^3};\\
&b(3{,}1) = \mathop{\mathrm{I}}_{x>a_M} M\frac{ch\,l(x-a_M) + \cos l(x-a_M)}{2l} + \mathop{\mathrm{I}}_{x>a_F} F\frac{sh\,l(x-a_F) + \sin l(x-a_F)}{2l} +\\
&\mathop{\mathrm{I}}_{x>a_H} q\frac{ch\,l(x-a_H) - \cos l(x-a_H)}{2l^2} - \mathop{\mathrm{I}}_{x>a_K} q\frac{ch\,l(x-a_K) - \cos l(x-a_K)}{2l^2}
\end{aligned}
\tag{3.22}
$$

$$b(4,1)=\underset{x>a_M}{\mathrm{I}}\,M\frac{l\left(sh\,l\left(x-a_M\right)-\sin\,l\left(x-a_M\right)\right)}{2l}+\underset{x>a_F}{\mathrm{I}}\,F\frac{ch\,l\left(x-a_F\right)+\cos\,l\left(x-a_F\right)}{2l}+$$
$$\underset{x>a_H}{\mathrm{I}}\,q\frac{sh\,l\left(x-a_H\right)+\sin\,l\left(x-a_H\right)}{2l}-\underset{x>a_K}{\mathrm{I}}\,q\frac{sh\,l\left(x-a_K\right)+\sin\,l\left(x-a_K\right)}{2l};\qquad(3.22)$$

$\underset{x>a_M}{\mathrm{I}}=\begin{cases}1, x>a_M\\ 0, x\le a_M\end{cases}$ – единичная функция Хевисайда.

M, F, q – сосредоточенные момент, сила и распределенная нагрузка.

a_M, a_F, a_H, a_K – координаты внешней нагрузки.

Протоколы построения эпюр $EIJ(x)$, $EIj(x)$, $M(x)$ и $Q(x)$ примут вид

<u>стержень 0-1</u>

```
x = 0 : 0.0001 : 2.0;  la = sqrt (0.1*0.315365);
EIv = – (X(2,1)*(sinh(la*x)+sin(la*x))/(2*la) – X(4,1)* …
(sinh(la*x)–sin(la*x))/(2*la^3));
EIfi = – (X(2,1)*(cosh(la*x)+cos(la*x)) / 2 – X(4,1)* …
(cosh(la*x)–cos(la*x))/(2*la^2));
Q = – X(2,1)*la^4*(cosh(la*x)–cos(la*x)) / (2*la^2) + …
X(4,1)* (cosh(la*x)+cos(la*x)) / 2;
M = – X(2,1)*la^4*(sinh(la*x)–sin(la*x))/(2*la^3) + …
X(4,1)*(sinh(la*x)+sin(la*x))/(2*la);
subplot (2,2,1), plot (x, EIv); axis ([0 2 – 30  30]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([0 2 – 30  30]); grid on
subplot (2,2,3), plot (x, Q); axis ([0 2 – 10  10]); grid on
subplot (2,2,4), plot (x, M); axis ([0 2 – 3    3 ]); grid on
```

Перед выполнением этого протокола необходимо поместить в окно команд вектор граничных параметров X (см. п. 3.2.5), значения нагрузки и их координаты.

```
x = 2.0 : 0.0001 : 4.0; m = 20.0; am = 2.0; la = sqrt (0.1*0.315365);
EIv = – (X(2,1)*(sinh(la*x)+sin(la*x))/(2*la) – X(4,1)* …
(sinh(la*x)–sin(la*x))/(2*la^3)+m*(cosh(la*(x-am)) – …
cos (la*(x-am))) / (2*la^2));
```

```
EIfi = – (X(2,1)*(cosh(la*x)+cos(la*x)) / 2 – X(4,1)*(cosh(la*x)– …
cos(la*x))/(2*la^2))+ m*(sinh(la*(x-am))+sin(la*(x-am))) / (2*la));
Q = – X(2,1)*la^4*(cosh(la*x)–cos(la*x)) / (2*la^2) + …
X(4,1)* (cosh(la*x)+cos(la*x)) / 2– m*(sinh(la*(x-am))– …
sin(la*(x-am)))* la / 2;
M = – X(2,1)*la^4*(sinh(la*x)–sin(la*x))/(2*la^3) + X(4,1)* …
(sinh(la*x)+sin(la*x))/(2*la))– m*(cosh(la*(x-am)+ …
cos(la*(x-am))) / 2;
subplot (2,2,1), plot (x, EIv); axis ([2  4 – 30  30]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([2  4 – 40  40]); grid on
subplot (2,2,3), plot (x, Q); axis ([2  4 – 2    2]); grid on
subplot (2,2,4), plot (x, M); axis ([2  4 – 30  30]); grid on
```

<u>стержень 1-2</u>

```
x = 0 : 0.0001 : 6.0; q = 10.0; la = sqrt (0.1*0.315365);
EIv = – (X(6,1)*(sinh(la*x)+sin(la*x))/(2*la) – X(7,1)* …
(cosh(la*x)–cos(la*x))/(2*la^2)) – X(8,1)*(sinh(la*x)–sin(la*x) …
) / (2*la^3)) + q* (cos(la*x) + cos(la*x) – 2) / (2*la^4));
EIfi = – (X(6,1)*(cosh(la*x)+cos(la*x)) / 2 – X(7,1)* (sinh(la*x) + …
sin(la*x)) / (2* la) – X(8,1)* (cosh(la*x) – cos(la*x)) / (2*la^2) + …
q*(sinh(la*x)–sin(la*x))/(2*la^3));
Q = – X(6,1)*la^4*(cosh(la*x)–cos(la*x)) / (2*la^2) + X(7,1)*  …
la^4 *(sinh(la*x)–sin(la*x))/(2*la^3) + X(8,1)* (cosh(la*x) + cos(la*x) …
) / 2 – q*(sinh(la*x)+sin(la*x))/(2*la);
M = – X(6,1)*la^4*(sinh(la*x)–sin(la*x))/(2*la^3) + X(7,1)*  …
(cosh(la*x)+cos(la*x)) / 2 + X(8,1)*(sinh(la*x)+sin(la*x))/(2*la) – …
q*(cosh(la*x)–cos(la*x)) / (2*la^2);
subplot (2,2,1), plot (x, EIv); axis ([0 6 – 110  110]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([0 6 –   60     60]); grid on
```

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0 6 – 40 40]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0 6 – 40 40]); *grid on*

<u>стержень 2-3</u>

x = 0 : 0.0001 : 2.0; *la* = *sqrt* (0.1*0.315365);

EIv = – (X(10,1)*(sinh(*la***x*)+sin(*la***x*))/(2**la*) – X(11,1)* (…

cosh(*la***x*)–cos(*la***x*))/(2**la*^2)–X(12,1)*(sinh(*la***x*)–sin(*la***x*)) …

/(2**la*^3));

EI*fi* = – (X(10,1)*(cosh(*la***x*)+cos(*la***x*)) / 2 – X(11,1)*(sinh(*la***x*) + …

sin(*la***x*))/(2**la*) – X(12,1)*(cosh(*la***x*)–cos(*la***x*))/(2**la*^2));

Q = – X(10,1)*(cosh(*la***x*)–cos(*la***x*))**la*^4 / (2**la*^2) + X(11,1)* …

la^4*(sinh(*la***x*)–sin(*la***x*))/(2**la*^3) + X(12,1)*(cosh(*la***x*) + …

cos(*la***x*)) / 2;

M = – X(10,1)*(sinh(*la***x*)–sin(*la***x*))**la*^4/(2**la*^3) + X(11,1)* …

(cosh(*la***x*)+cos(*la***x*)) / 2 + X(12,1)*(sinh(*la***x*) +(sinh(*la***x*))/(2**la*);

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0 2 – 60 60]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0 2 – 90 90]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0 2 – 40 40]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0 2 – 70 70]); *grid on*

Протоколы эпюр второй части стержня имеют вид:

x = 2.0 : 0.0001 : 3.0; *la* = *sqrt* (0.1*0.315365); *f*1 = – 40.0; *af*1 = 2.0;

EIv = – (X(10,1)*(sinh(*la***x*)+sin(*la***x*))/(2**la*) – X(11,1)* (…

cosh(*la***x*)–cos(*la***x*)) / (2**la*^3) – X(12,1)*(sinh(*la***x*)–sin(*la***x*)) …

/(2**la*^3)+ *f*1*(sinh(*la**(*x*-*a f*1))–sin(*la**(*x*-*a f*1))) / (2**la*^3));

EI*fi* = – (X(10,1)*(cosh(*la***x*)+cos(*la***x*)) / 2 – X(11,1)*(sinh(*la***x*)+ …

sin(*la***x*))/(2**la*)) – X(12,1)*(cosh(*la***x*) - cosh(*la***x*)) / (2**la*^2) + …

*f*1*(cosh(*la**(*x*-*a f*1))–cos(*la**(*x*-*a f*1))) / (2**la*^2));

Q = – X(10,1*(cosh(*la***x*)–cos(*la***x*))**la*^4 / (2**la*^2) + X(11,1)* …

la^4*(sinh(*la***x*)–sin(*la***x*))/(2**la*^3) + X(12,1)*(cosh(*la***x*) + …

cos(*la***x*)) / 2 – *f*1*(cosh(*la**(*x*-*a f*1))+cos(*la**(*x*-*a f*1))) / 2;

```
M = – X(10,1)*(sinh(la*x)–sin(la*x))*la^4/(2*la^3) + X(11,1)* …
 (cos(la*x)+ cos(la*x)) / 2 + X(12,1)*(sinh(la*x)+sin(la*x))/(2*la) …
- f1*(sinh(la*(x-a f1))+sin(la*(x-a f1))) / (2*la);
subplot (2,2,1), plot (x, EIv); axis ([2  3 – 60  60]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([2  3 – 90  90]); grid on
subplot (2,2,3), plot (x, Q); axis ([2  3 – 10    10]); grid on
subplot (2,2,4), plot (x, M); axis ([2  3 – 70    70]); grid on
```

<u>стержень 3-4</u>

```
x = 0 : 0.0001 : 1.0;  la = sqrt (0.1*0.315365);
EIv = – (X(14,1)*(sinh(la*x)+sin(la*x))/(2*la) – X(15,1)*(cosh(la*x) …
–cos(la*x))/(2*la^2) – X(16,1)*(sinh(la*x)–sin(la*x))/(2*la^3));
EIfi = – (X(14,1)*(cosh(la*x)+cos(la*x)) / 2 – X(15,1)*(sinh(la*x)+ …
sinh(la*x))/(2*la) – X(16,1)*(cosh(la*x)–cos(la*x))/(2*la^2));
Q = – X(14,1)*la^4*(cosh(la*x)–cos(la*x)) / (2*la^2) + X(16,1)* la^4* …
(sinh(la*x)–sin(la*x))/(2*la^3)+X(16,1)*(cosh(la*x)+cos(la*x)) / 2;
M=–X(14,1)*la^4*(sinh(la*x)–sin(la*x))/(2*la^3)+X(15,1)*(cosh(la*x)+…
cos(la*x))/2+X(16,1)*(sinh(la*x)+sin(la*x))/(2*la);
subplot (2,2,1), plot (x, EIv); axis ([0 1 – 110  110]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([0 1  – 120  120]); grid on
subplot (2,2,3), plot (x, Q); axis ([0 1  – 70  70]); grid on
subplot (2,2,4), plot (x, M); axis ([0 1  – 70  70]); grid on
```

Анализ показывает, что несмотря на большой объем протоколов построения эпюр параметров балки, сам процесс довольно простой. В отдельном М-файле необходимо только дополнять недостающие элементы параметров и менять индексы вектора X. Эпюры EIJ(x), EIj (x), Q(x) и M(x) представлены на рис. 3.7.

3.2.7. *Определение критических сил потери устойчивости*

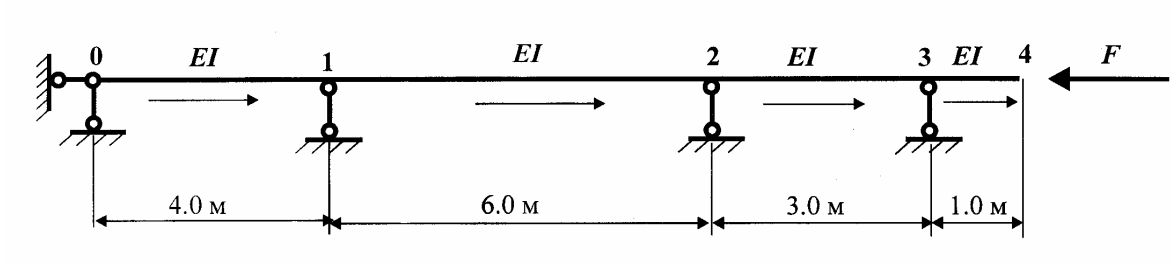


Рис. 3.11

Исходные данные: $\mathbf{l}_1^{0-1} = 4.0$ м.; $\mathbf{l}_2^{1-2} = 6.0$ м.; $\mathbf{l}_3^{2-3} = 3.0$ м.; $\mathbf{l}_4^{3-4} = 1.0$ м.; $EI = 1$.

Матрица фундаментальных функций продольно-поперечного изгиба (задача устойчивости плоских стержневых систем)

$$\mathbf{A_i} = \begin{bmatrix} 1 & A_{12} & -A_{13} & -A_{14} & \\ & A_{22} & -A_{12} & -A_{13} & \\ & -A_{32} & A_{22} & A_{12} & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

$$n = \sqrt{\frac{F}{EI}}; \quad A_{12} = \frac{\sin nx}{n}; \quad A_{13} = \frac{1-\cos nx}{n^2};$$
$$A_{14} = \frac{nx - \sin nx}{n^3}; \quad A_{22} = \cos nx; \qquad (3.23)$$
$$A_{32} = -n \sin nx.$$

Критические силы определяются как корни трансцендентного уравнения $|\mathbf{A}_*(F)| = 0$, где матрицу $\mathbf{A}_*$ можно взять из задач статики или динамики данной балки с заменой фундаментальных функций. Для неразрезной балки можно не учитывать нормальные силы N, потому матрица $\mathbf{A}_*(F)$ примет размер 16×16 строк и столбцов.

Корни (критические силы) можно определить методом последовательного перебора в сочетании с прямым ходом метода исключения Гаусса. Организуется цикл вычисления определителя $|\mathbf{A}_*(F)|$, в конце которого величины F и d выводятся в окно команд. При просмотре таблицы значений F и d определяются точки, где *изменяется знак* определителя $d = |\mathbf{A}_*(F)|$. Эти точки и есть критические силы потери устойчивости. Рекомендуется начинать вычисления с начального значения сжимающей силы $F_0 = 0.001$ с шагом $\Delta F = 0.001$. Число вычислений определителя $d = |\mathbf{A}_*(F)|$ $п_1 = 300 - 500$ позволяет надежно и достаточно точно определить первую и старшие критиче-

ские силы. При решении задачи принимаем, что EI = 1. Матрица устойчивости неразрезной балки принимает вид

$$A_*(F)=\begin{array}{c|cccc|cccc|cccc|cccc|c}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & \\
\hline
1 & & A_{12} & & -A_{14} & & & & & & & & & & & & & \\
2 & & A_{22} & & -A_{13} & & -1 & & & & & & & & & & & \\
3 & & -A_{32} & & A_{12} & & & -1 & & & & & & & & & & 0\text{-}1 \\
4 & -1 & & & 1 & & & & & & & & & & & & & \\
\hline
5 & & & & & & A_{12} & -A_{13} & -A_{14} & & & & & & & & & \\
6 & & & & & & A_{22} & -A_{12} & -A_{13} & & -1 & & & & & & & \\
7 & & & & & & -A_{32} & A_{22} & A_{12} & & & -1 & & & & & & 1\text{-}2 \\
8 & & & -1 & & & & & 1 & & & & & & & & & \\
\hline
9 & & & & & & & & & & A_{12} & -A_{13} & -A_{14} & & & & & \\
10 & & & & & & & & & & A_{22} & -A_{12} & -A_{13} & & -1 & & & \\
11 & & & & & & & & & & -A_{32} & A_{22} & A_{12} & & & -1 & & \\
12 & & & & & -1 & & & & & & & 1 & & & & & 2\text{-}3 \\
\hline
13 & & & & & & & & & -1 & & & & & A_{12} & -A_{13} & -A_{14} & \\
14 & & & & & & & & & & & & & -1 & A_{22} & -A_{12} & -A_{13} & \\
15 & & & & & & & & & & & & & & -A_{32} & A_{22} & A_{12} & 3\text{-}4 \\
16 & & & & & & & & & & & & & & & & 1 &
\end{array} \qquad (3.24)$$

Обозначения переменных, принятых в программе

a **–** матрица $\mathbf{A}_*(F)$; п – порядок матрицы $\mathbf{A}_*(F)$; $п_1$ – число циклов вычисления определителя $d = |\mathbf{A}_*(F)|$; d – величина определителя; f – сжимающая сила F; df – шаг изменения сжимающей силы F; m – счетчик цикла вычисления определителя; **l**1 – **l**4 – длины стержней балки; $п_2$ – коэффициент п фундаментальных функций; X, Y – векторы значений сжимающих сил F и определителя d.

<u>Текст программы</u>

n = 16; *n*1 = 300; *f* = 0.01; *df* = 0.01; X = *zeros* (*n*1,1); Y = *zeros* (*n*1,1);

*l*1 = 4.0; *l*2 = 6.0; *l*3 = 3.0; *l*4 = 1.0;

for *m* = 1 : *n*1 *n*2 = *sqrt* (*f*); *a* = *zeros* (*n*,*n*);

a(1,2) = *sin* (*n*2**l*1) / *n*2; *a*(1,4) = – (*n*2**l*1–*sin* (*n*2**l*1)) / (*n*2^3);

a(2,2)=*cos* (*n*2**l*1); *a*(2,4) = – (1–*cos* (*n*2**l*1)) / (*n*2^2); *a*(2,6) = – 1;

a(3,2) = *n*2*sin(*n*2**l*1); *a*(3,7) = – 1; *a*(4,1) = – 1; *a*(4,4) = 1;

a(5,6) = *sin* (*n*2**l*2) / *n*2; *a*(5,7)= – (1–*cos* (*n*2**l*2)) / (*n*2^2);

a(5,8) = – (*n*2**l*2–*sin* (*n*2**l*2)) / (*n*2^ 3); *a*(6,6)=*cos* (*n*2**l*2); *a*(6,7) = – *a*(5,6);

a(6,8) = *a*(5,7); *a*(6,10) = – 1; *a*(7,6) = *n*2*sin(*n*2**l*2); *a*(7,7) = *a*(6,6);

a(7,8) = *a*(5,6); *a*(7,11) = – 1; *a*(8,3) = – 1; *a*(8,6) = 1; *a*(9,10) = (*sin* (*n*2**l*3) / *n*2;

a(9,11)= – (1–*cos* (*n*2**l*3)) / (*n*2^2); *a*(9,12) = – (*n*2**l*3–*sin* (*n*2**l*3)) / (*n*2^ 3);

a(10,10)=*cos* (*n*2**l*3); *a*(10,11) = – *a*(9,10); *a*(10,12) = *a*(9,11); *a*(10,14) = – 1;

a(11,10) = *n*2**sin*(*n*2**l*3); *a*(11,11) = *a*(10,10); *a*(11,12) = *a*(9,10); *a*(11,15) = – 1;

a(12,5) = – 1; *a*(12,12) = 1; *a*(13,9) = – 1; *a*(13,14) = *sin* (*n*2**l*4) / *n*2;

a(13,15)= – (1–*cos* (*la**l*4)) / (*n*2^2); *a*(13,16) = – (*n*2**l*4–*sin* (*n*2**l*4)) / (*n*2^ 3);

a(14,13) = – 1; *a*(14,14)=*cos* (*n*2**l*4); *a*(14,15) = - *a*(13,14); *a*(14,16) = *a*(13,15);

a(15,14) = *n*2**sin*(*n*2**l*4); *a*(15,15) = *a*(14,14); *a*(15,16) = *a*(13,14);

a(16,16) = 1; *d* = *det*(*a*); *X*(*m*,1) = *f*; *Y*(*m*,1) = *d*; *f* = *f* + *df*; *end*;

% Построение графика определителя матрицы устойчивости

plot (X,Y); *grid on*

% Вывод таблицы значений *f* и *d* в формате *long e*

format long e

[X Y]

График зависимости d = | **A***(F) | позволяет лишь грубо определить интервалы, где находятся корни (критические силы). Как и в динамике, этот график *не имеет* точек разрыва 2-го рода (в методах сил и перемещений аналогичный график *имеет* разрывы 2-го рода) и служит вспомогательным средством при поиске критических сил. Уточнить критические силы можно при повторных прогонах программы (без построения графиков) с новыми интервалами для F.

<u>Результаты поиска критических сил потери устойчивости</u>

$F_1 = 0.443685\dfrac{EI}{\mathbf{l}^2}$; $F_2 = 0.652565\dfrac{EI}{\mathbf{l}^2}$; $F_3 = 1.051035\dfrac{EI}{\mathbf{l}^2}$;

$F_4 = 1.6187795\dfrac{EI}{\mathbf{l}^2}$; $F_5 = 2.430485\dfrac{EI}{\mathbf{l}^2}$ кН.

3.2.8. *Построение форм потери устойчивости*

Аналогично задаче динамики (п.3.2.4) примем, что b(16,1) = $Q^{3\text{-}4}(\mathbf{l}) = 1$. Уравнение для граничных параметров балки примет вид (3.16), где нужно заменить фундаментальные функции поперечных колебаний на фундаментальные функции продольно-поперечного изгиба (3.23). Тогда уравнение для определения граничных параметров балки при потере устойчивости предстанет следующим образом (см. (3.25)).

Программа вычисления граничных параметров имеет вид

a = *zeros* (16,16); *b* = *zeros* (16,1); X = *zeros* (16,1); *n*2 =*sqrt*(0.443685);

*l*1 = 4.0; *l*2 = 6.0; *l*3 = 3.0; *l*4 = 1.0;

a(1,2) =*sin*(*n*2**l*1)/*n*2; *a*(1,4) = – (*n*2**l*1–*sin* (*n*2**l*1)) / (*n*2^3); *a*(2,2)=*cos* (*n*2**l*1);

a(2,4) = – (1–*cos* (*n*2**l*1)) / (*n*2^2); *a*(2,6) = – 1; *a*(3,2) = *n*2*sin(*n*2**l*1);

a(3,4) = *a*(1,2); *a*(3,7) = – 1; *a*(4,1) = – 1; *a*(4,4) = 1;

a(5,6) = *sin* (*n*2**l*2) / *n*2; *a*(5,7)= – (1–*cos* (*n*2**l*2)) / (*n*2^2);

a(5,8) = – (*n*2**l*2–*sin* (*n*2**l*2)) / (*n*2^ 3); *a*(6,6)=*cos* (*n*2**l*2); *a*(6,7) = – *a*(5,6);

a(6,8) = *a*(5,7); *a*(6,10) = – 1; *a*(7,6) = *n*2*sin(*n*2**l*2); *a*(7,7) = *a*(6,6);

a(7,8) = *a*(5,6); *a*(7,11) = – 1; *a*(8,3) = – 1; *a*(8,8) = 1;

a(9,10) = *sin* (*n*2**l*3) / *n*2; *a*(9,11)= – (1–*cos* (*n*2**l*3)) / (*n*2^2);

a(9,12) = – (*n*2**l*3–*sin* (*n*2**l*3)) / (*n*2^ 3); *a*(10,10)=*cos* (*n*2**l*3);

a(10,11) = – *a*(9,10); *a*(10,12) = *a*(9,11); *a*(10,14) = – 1;

a(11,10) = *n*2**sin*(*n*2**l*3); *a*(11,11) = *a*(10,10); *a*(11,12) = *a*(9,10);

a(11,15) = – 1; *a*(12,5) = – 1; *a*(12,12) = 1; *a*(13,9) = – 1;

a(13,14) = *sin* (*n*2**l*4) / *n*2; *a*(13,15)= – (1–*cos* (*la***l*4)) / (*n*2^2);

a(13,16) = – (*n*2**l*4–*sin* (*n*2**l*4)) / (*n*2^ 3); *a*(14,13) = – 1;

a(14,14)=*cos* (*n*2**l*4); *a*(14,15) = - *a*(13,14); *a*(14,16) = *a*(13,15);

a(15,14) = *n*2**sin*(*n*2**l*4); *a*(15,15) = *a*(14,14); *a*(15,16) = *a*(13,14);

a(16,16) = 1; *b* (16,1) = 1; X = *a*\ *b*; X = X / X(9,1)

Результаты вычислений относительных значений граничных параметров балки при потере устойчивости сведены в таблицу 3.4.

$$
\underset{\boldsymbol{a}(\mathbf{F})}{\begin{array}{c|cccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
\hline
1 & & A_{12} & & -A_{14} & & & & & & & & & & & & \\
2 & & A_{22} & & -A_{13} & & -1 & & & & & & & & & & \\
3 & & -A_{32} & & A_{12} & & & -1 & & & & & & & & & \\
4 & -1 & & & 1 & & & & & & & & & & & & \\
5 & & & & & & A_{12} & -A_{13} & -A_{14} & & & & & & & & \\
6 & & & & & & A_{22} & -A_{12} & -A_{13} & & -1 & & & & & & \\
7 & & & & & & A_{32} & A_{22} & A_{12} & & & -1 & & & & & \\
8 & & & -1 & & & & & 1 & & & & & & & & \\
9 & & & & & & & & & & A_{12} & -A_{13} & -A_{14} & & & & \\
10 & & & & & & & & & & A_{22} & -A_{12} & -A_{13} & & -1 & & \\
11 & & & & & & & & & & -A_{32} & A_{22} & A_{12} & & & -1 & \\
12 & & & & & -1 & & & & & & & 1 & & & & \\
13 & & & & & & & & & -1 & & & & & A_{12} & -A_{13} & -A_{14} \\
14 & & & & & & & & & & & & & -1 & A_{22} & -A_{12} & -A_{13} \\
15 & & & & & & & & & & & & & & -A_{32} & A_{22} & A_{12} \\
16 & & & & & & & & & & & & & & & & 1
\end{array}}
\quad
\begin{array}{c|c}
\mathbf{X}(16,1) & \mathbf{B}(16,1) \\
\hline
Q^{0-1}(\ell) & \\
EI\varphi^{0-1}(o) & \\
Q^{1-2}(\ell) & \\
Q^{0-1}(o) & \\
Q^{2-3}(\ell) & \\
EI\varphi^{1-2}(o) & \\
M^{1-2}(o) & \\
Q^{1-2}(o) & \\
EI\vartheta^{3-4}(\ell) & \\
EI\varphi^{2-3}(o) & \\
M^{2-3}(o) & \\
Q^{2-3}(o) & \\
EI\varphi^{3-4}(\ell) & \\
EI\varphi^{3-4}(o) & \\
M^{3-4}(o) & \\
Q^{3-4}(o) & Q^{3-4}(\ell)=1
\end{array}
\tag{3.25}
$$

Таблица 3.4

№ п/п	Относительные граничные параметры балки	Относительные формы потери устойчивости при критических силах				
		F_1=0,443685	F_2=0,6525565	F_3=1,051035	F_4=1,618795	F_5=2,430485
1	$X(1,1)=Q^{0-1}(\mathbf{l})=$	-0.05447	-0.0103	0.2820	-0.5033	0.0329
2	$X(2,1)=EIj^{0-1}(o)=$	-0.5915	0.5841	-1.6119	2.0126	-1.8039
3	$X(3,1)=Q^{1-2}(\mathbf{l})=$	-0.0026	0.0659	-0.0127	0.5231	-0.0775
4	$X(4,1)=Q^{0-1}(o)=$	-0.0547	-0.0103	0.2820	-0.5033	0.0329
5	$X(5,1)=Q^{2-3}(\mathbf{l})=$	0.0697	-0.3356	-0.7009	-0.9147	-0.6990
6	$X(6,1)=EIj^{1-2}(o)=$	0.7580	-0.5503	0.5032	0.9372	-1.8019
7	$X(7,1)=M^{1-2}(o)=$	-0.2186	-0.0411	1.1278	-2.0131	0.1316
8	$X(8,1)=Q^{1-2}(o)=$	-0.0026	0.0659	-0.0127	0.5231	-0.0775
9	$X(9,1)=EIu^{3-4}(\mathbf{l})=$	1.0000	1.0000	1.0000	1.0000	1.0000
10	$X(10,1)=EIj^{2-3}(o)=$	-0.7352	-0.2117	0.6437	1.4961	1.8552
11	$X(11,1)=M^{2-3}(o)=$	-0.2345	0.3542	1.0518	1.1254	-0.3335
12	$X(12,1)=Q^{2-3}(o)=$	-0.0697	-0.3356	-0.7009	-0.9147	-0.6990
13	$X(13,1)=EIj^{3-4}(\mathbf{l})=$	1.0780	1.1177	1.1993	1.3312	1.5591
14	$X(14,1)=EIj^{3-4}(o)=$	0.8475	0.7724	0.6224	0.3914	0.0184
15	$X(15,1)=M^{3-4}(o)=$	-0.4437	-0.6526	-1.0510	-1.6189	-2.4305
16	$X(16,1)=Q^{3-4}(o)=$	0.0000	0.0000	0.0000	0.0000	0.0000

Выражение для прогиба (формы потери устойчивости) согласно методу начальных параметров имеет вид

$$EIJ(x)=-\left(EIJ(o)\cdot 1+EIj(o)\cdot\frac{\sin nx}{n}-M(o)\frac{1-\cos nx}{n^2}-\right.$$
$$\left.-Q(o)\frac{nx-\sin nx}{n^3}\right);\qquad n=\sqrt{\frac{F}{EI}}\ . \tag{3.26}$$

Протокол построения форм потери устойчивости запишется в виде

x1 = 0 : 0.001 : 4.0; x2 = 0 : 0.001 : 6.0; x3 = 0 : 0.001 : 3.0;

x4 = 0 : 0.001 : 1.0; *n*2 = *sqrt* (0.443685);

EIv1 = – (X(2,1)*(sin(*n*2*x1) / *n*2 – X(4,1)*(*n*2*x1–sin(*n*2*x1)) …

/ *n*2^3);

EIv2 = – (X(6,1)*sin(*n*2*x2) / *n*2 – X(7,1)*(1–cos(*n*2*x2)) / *n*2^2) – …

X(8,1)*(*n*2*x2–sin(*n*2*x2)) / *n*2^3);

EIv3 = – (X(10,1)*sin(*n*2*x3) / *n*2 – X(11,1)*(1–cos(*n*2*x3)) / *n*2^2) – …

X(12,1)*(*n*2*x3–sin(*n*2*x3)) / *n*2^3);

EIv4 = – (X(14,1)*sin(*n*2*x*4) / *n*2 – X(15,1)*(1–cos(*n*2*x*4)) / *n*2^2 – …

X(16,1)*(*n*2*x*4–sin(*n*2*x*4)) / *n*2^3);

subplot (2,2,1), *plot* (*x*1, EIv1); *axis* ([0 4 – 2 2]); *grid on*

subplot (2,2,2), *plot* (*x*2, EIv2); *axis* ([0 6 – 2 2]); *grid on*

subplot (2,2,3), *plot* (*x*3, EIv3); *axis* ([0 3 – 2 2]); *grid on*

subplot (2,2,4), *plot* (*x*4, EIv4); *axis* ([0 1 – 2 2]); *grid on*

Перед выполнением данного протокола из отдельного М-файла в окно команд помещается вектор X – вектор относительных значений граничных параметров балки. Выполняется это прогонкой программы, решающей систему уравнений (3.25). Далее нужно задействовать протокол построения форм потери устойчивости балки при разных критических силах F_i. Первые 5 форм потери устойчивости балки при условии, что $EIJ^{3-4}(\mathbf{l})=1$, представлены на рис 3.12.

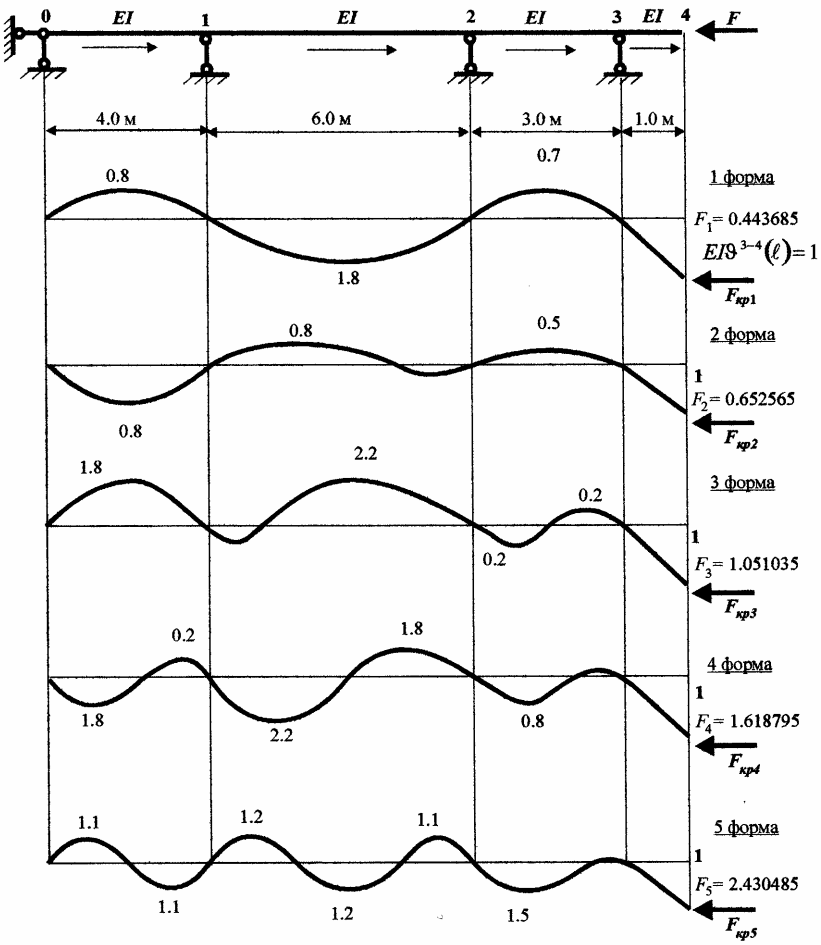


Рис. 3.12

3.2.9. *Варианты заданий*

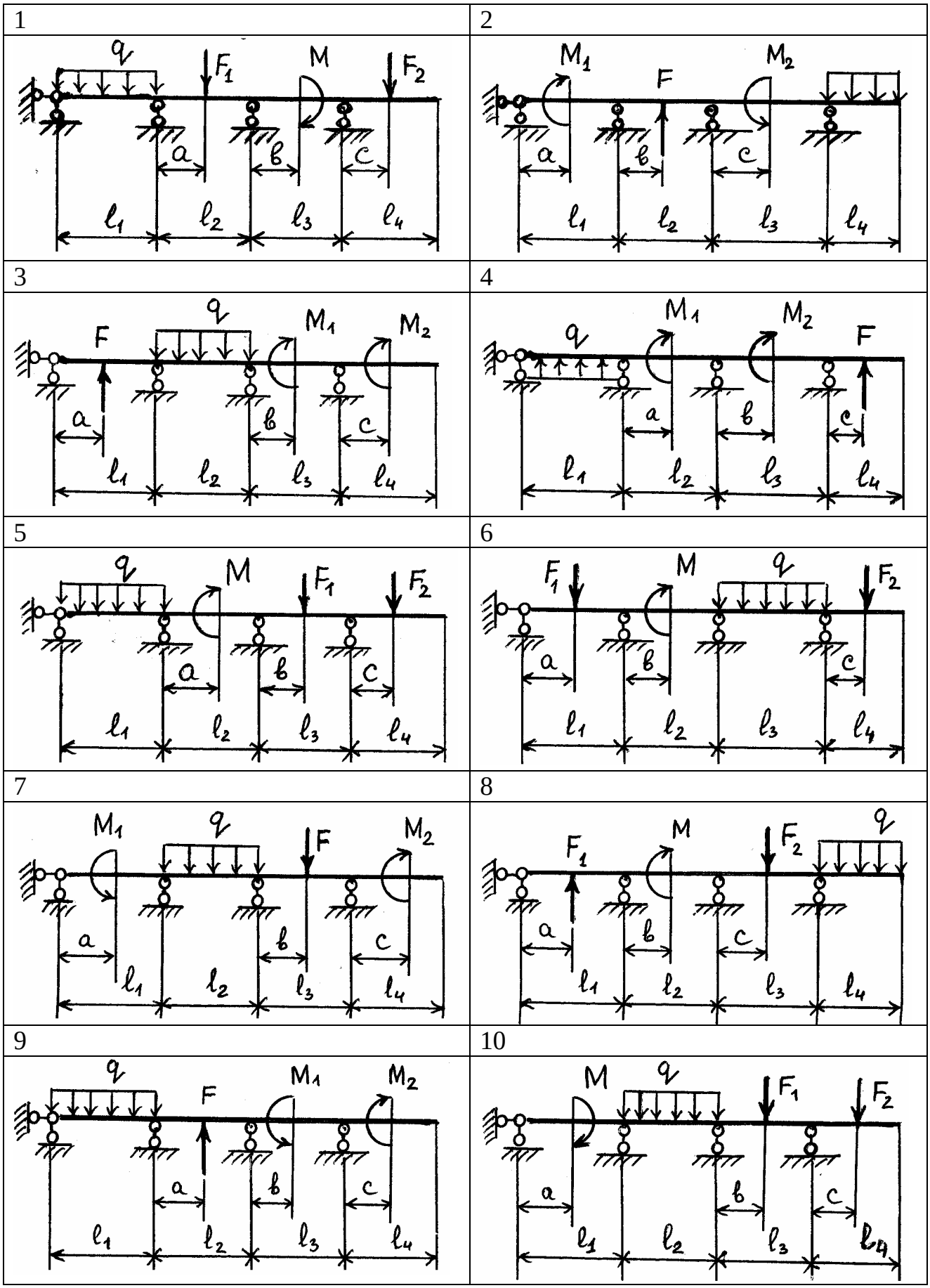
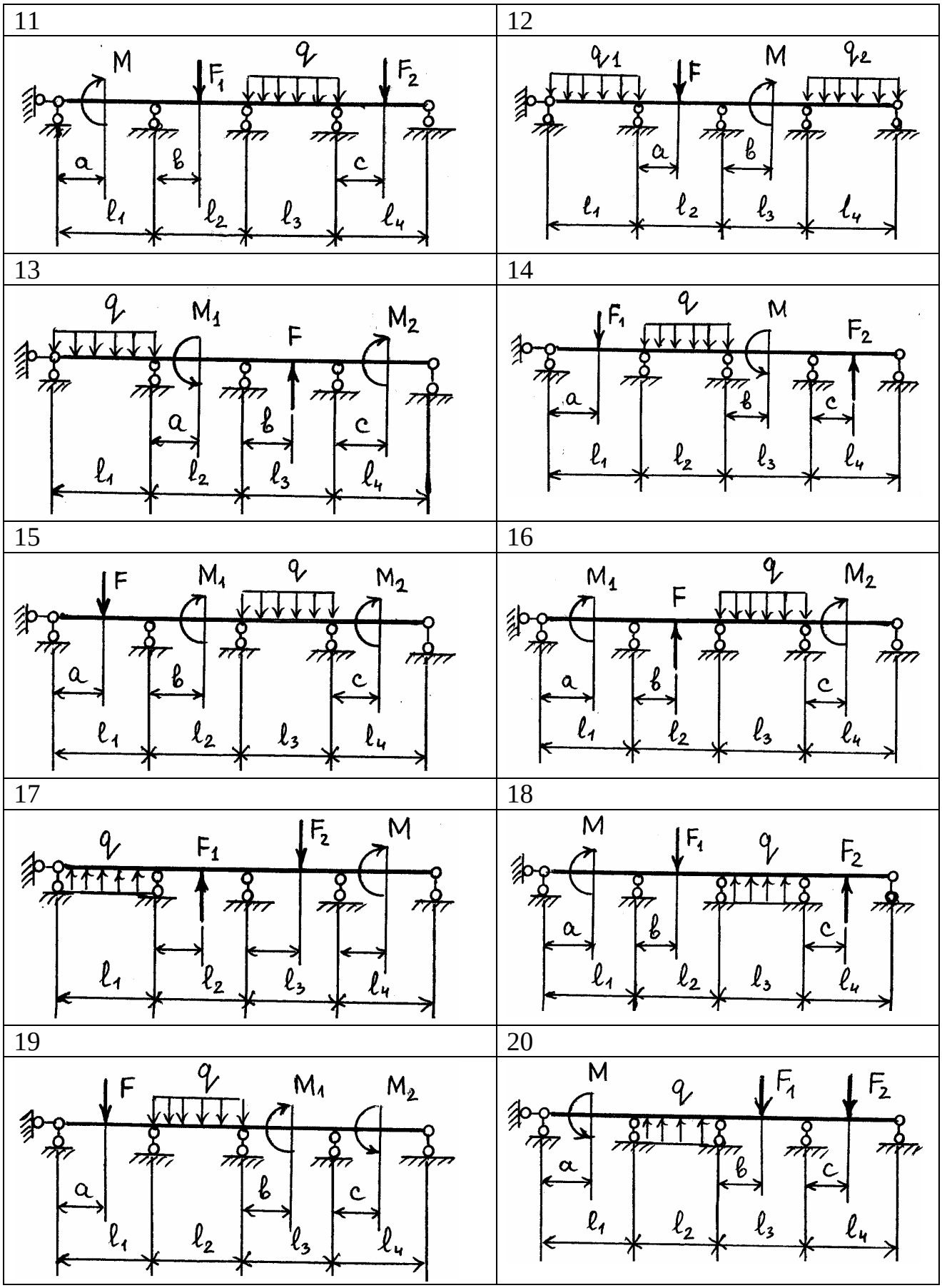
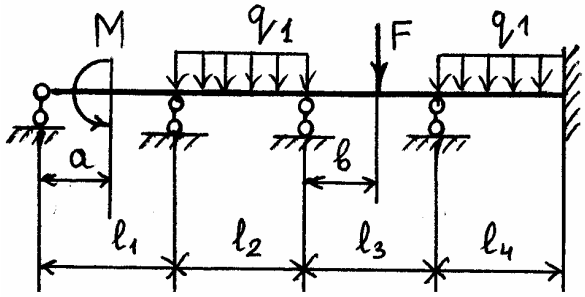
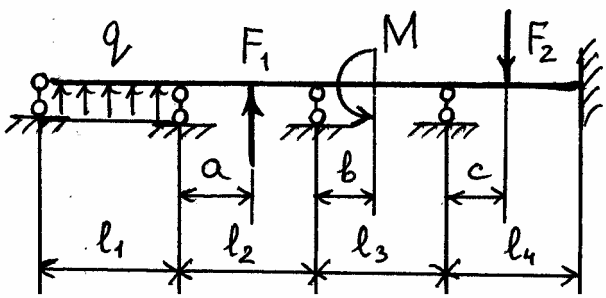
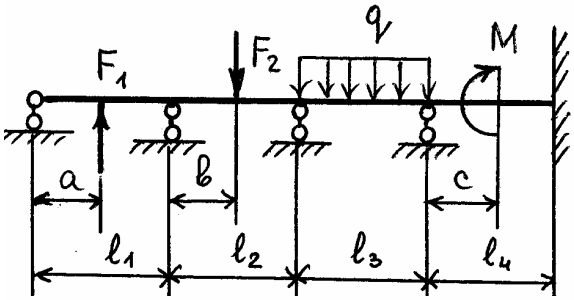
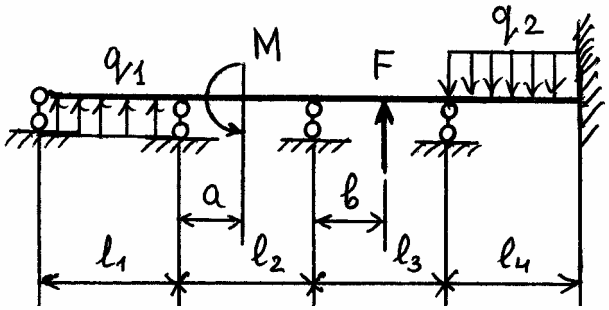
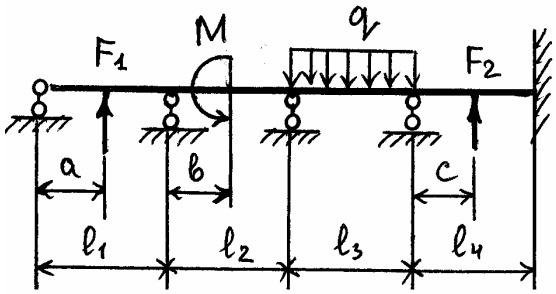
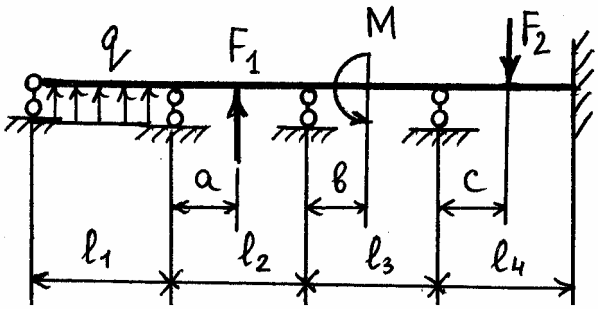
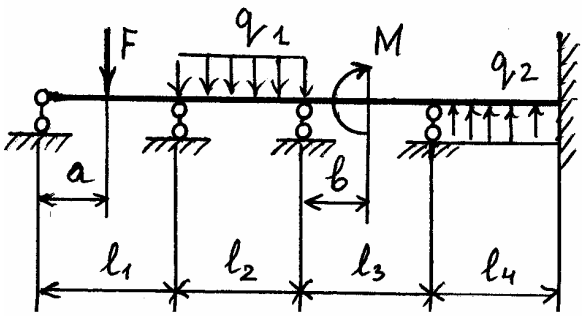
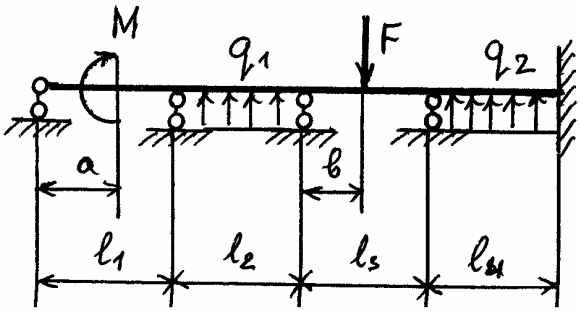
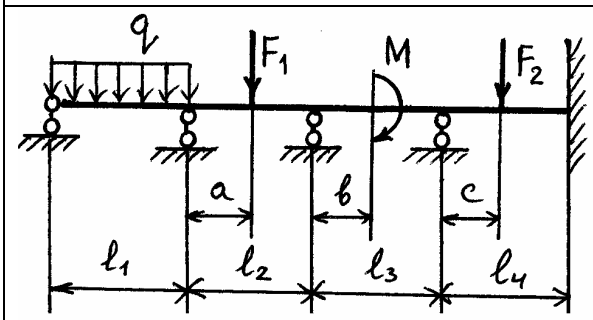
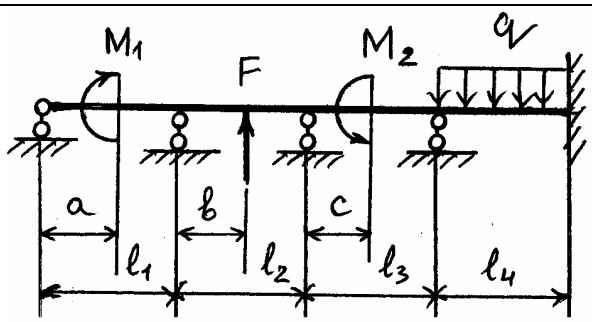


Рис. 3.13



Продолжение рис. 3.13

21	22
	
23	24
	
25	26
	
27	28
	
29	30
	

Окончание рис. 3.13

Таблица 3.5

Исходные данные к расчету неразрезной балки

№ варианта заданий	l_1	l_2	l_3	l_4	a	b	c	F	F_1	F_2	M	M_1	M_2	q	q_1	q_2
	м							кН			кНм			кН/м		
1	2	3	3	1	1	2	1	10	20	30	6	8	10	10	12	8
2	2	4	4	2	2	1	2	40	50	60	10	8	6	8	15	6
3	4	3	5	1,5	3	3	1,5	5	15	20	20	30	40	6	10	12
4	5	6	6	3	2	2	3	25	35	45	25	36	38	4	8	6
5	6	8	7	2,5	1	1	1	10	8	6	12	6	18	2	4	3
6	7	5	8	1	3	2,5	1,5	10	15	25	15	20	25	12	2	7
7	6	4	7	1,5	1	1,5	2,5	30	40	50	40	35	16	20	8	10
8	5	3	6	2	2	2	1,5	60	70	80	10	15	20	15	12	15
9	4	4	5	1	1,5	1	2	30	15	18	25	30	35	12	2	5
10	3	6	4	1,5	3	1,5	1	28	32	24	40	45	50	8	4	10
11	2,5	5	5	1	2	1,5	1,5	12	14	16	10	18	28	6	8	15
12	4	8	3	2,8	1	1,5	2	5	15	10	40	12	10	10	12	2
13	3	10	4	4	1	2	2	3	22	30	10	14	16	18	12	14
14	5	4	5	2	1	1,5	1,6	40	24	26	20	30	35	10	30	15
15	2	5	6	1,5	1,7	1,5	1,2	15	10	8	15	18	20	5	20	10
16	2	6	3	1	0,8	1,4	1,0	10	20	30	6	8	10	10	12	8
17	3	7	4	5	2	1	2	40	50	60	10	8	6	8	15	6
18	4	5	5	1,5	3	3	1,5	5	15	20	20	30	40	6	10	12
19	5	3	6	3	2	2	1,5	25	35	45	25	36	38	4	8	6
20	6	5	5	2,5	1	1	1	10	8	6	12	6	18	2	4	3
21	7	4	7	1	3	1,5	1,5	10	25	35	15	20	25	12	2	7
22	6	8	6	1,5	1	2	2,5	30	40	50	40	35	16	20	8	10
23	5	6	5	2	2	2	1,2	60	70	80	10	15	20	15	12	15
24	4	7	4	1	1,5	1	2	30	15	18	25	30	35	12	2	5
25	3	5	3	1,5	3	1,5	1	28	32	24	40	45	50	8	4	10
26	2	4	4	1	1,3	1,4	1,5	12	14	16	10	18	28	6	8	15
27	4	6	3	2	1	1,5	1,7	5	15	10	40	12	10	10	12	2
28	3	5	4	4	1	2	2	3	22	30	10	14	16	18	12	14
29	5	4	5	2	1	1,5	1,5	40	24	26	20	30	35	10	30	15
30	2	3	6	4	1,5	2	1	15	10	8	15	18	20	5	20	10

3.3. Расчет плоской рамы

3.3.1. *Определение граничных параметров при статической нагрузке*

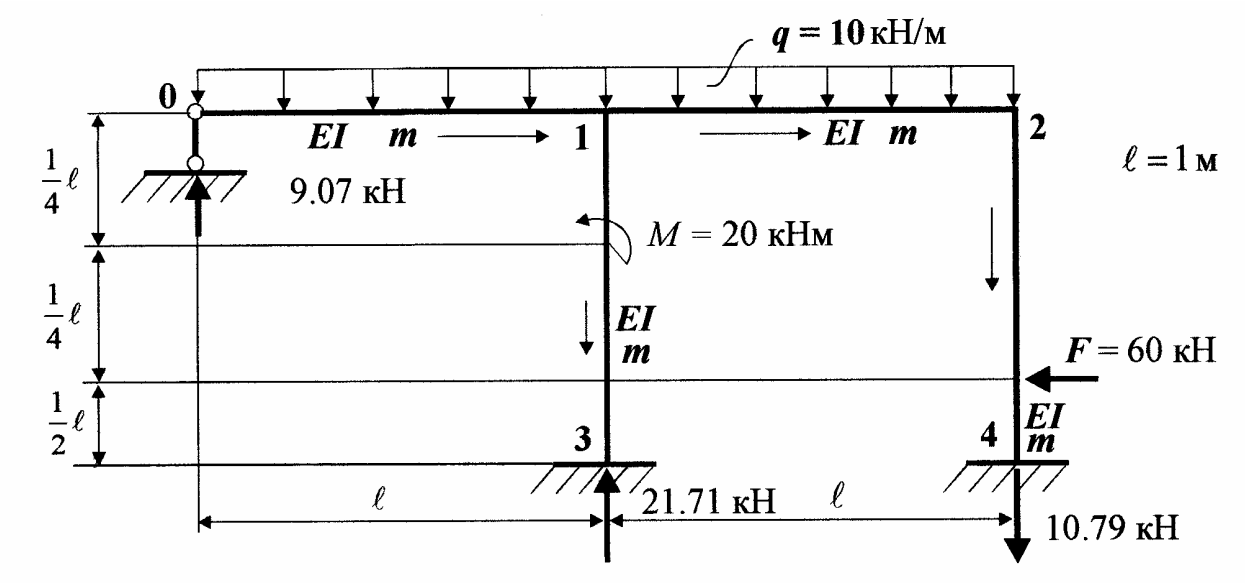
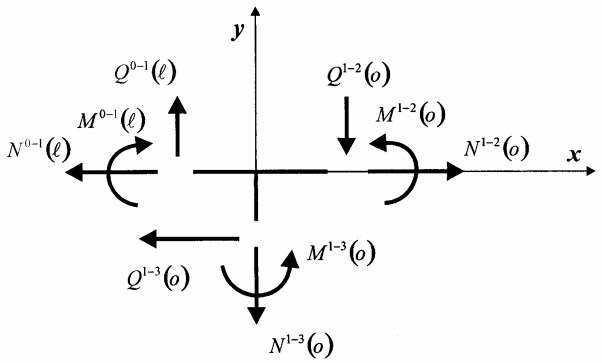


Рис. 3.14, *a*

Для построения эпюр напряженно-деформированного состояния рамы формируем систему линейных алгебраических уравнений краевой задачи по МГЭ. Для этого:

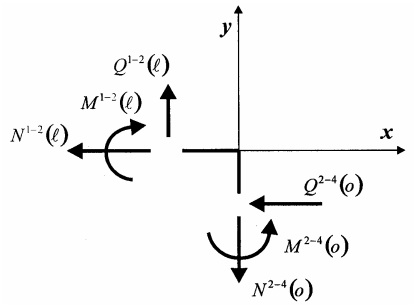
1. Разбиваем раму на 4 стержня. Нумеруем узлы и стрелками обозначаем начало и конец каждого стержня, т. е. формируем орграф расчета рамы (рис. 3.14, а).

2. Составляем уравнения равновесия и совместности перемещений узлов рамы. Уравнения равновесия узлов 1 и 2 составляем для недеформированного состояния, а уравнения совместности перемещений в соответствии с деформированным состоянием по рис. 3.14, d.



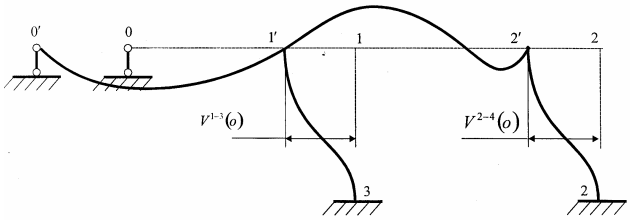
$$\sum y = 0 \to Q^{0-1}(\mathbf{l}) = Q^{1-2}(o) + N^{1-3}(o);$$
$$\sum x = 0 \to N^{0-1}(\mathbf{l}) = N^{1-2}(o) - Q^{1-3}(o);$$
$$\sum m_{(1)} = 0 \to M^{0-1}(\mathbf{l}) = M^{1-2}(o) + M^{1-3}(o);$$

Рис. 3.14, *b*



$$\sum y = 0 \to Q^{0-1}(\mathbf{l}) = N^{2-4}(o);$$
$$\sum x = 0 \to N^{1-2}(\mathbf{l}) = -Q^{2-4}(o);$$
$$\sum m_{(1)} = 0 \to M^{1-2}(\mathbf{l}) = M^{2-4}(o).$$

Рис. 3.14, *c*



$$j^{0-1}(\mathbf{l}) = j^{1-3}(o) = j^{1-2}(o);$$
$$j^{1-2}(\mathbf{l}) = j^{2-4}(o);$$
$$J^{1-3}(\mathbf{l}) = J^{2-4}(o).$$

Рис. 3.14, *d*

При расчете рамы полагаем, что стержни нерастяжимы и несжимаемы. Составленные уравнения равновесия и совместности перемещений узлов рамы помещаем в матрицу конечных параметров $\mathrm{Y}(\mathbf{l})$. Матрицы $\mathbf{X}_*$ и $\mathrm{Y}(\mathbf{l})$ при учете граничных условий примут вид

$\mathbf{X}_*=$

1	$EIJ^{0-1}(o)=0;\ M^{2-4}(\mathbf{l})$
2	$EIj^{0-1}(o)$
3	$M^{0-1}(o)=0;\ Q^{2-4}(\mathbf{l})$
4	$Q^{0-1}(o)$
5	$N^{0-1}(o)=0;\ N^{2-4}(\mathbf{l})$
6	$EIJ^{1-2}(o)=0;\ M^{1-3}(\mathbf{l})$
7	$EIj^{1-2}(o)=EIj^{1-3}(o);\ Q^{1-3}(\mathbf{l})$
8	$M^{1-2}(o)$
9	$Q^{1-2}(o)$
10	$N^{1-2}(o)$
11	$EIJ^{2-4}(o)=EIJ^{1-3}(o);\ N^{1-3}(\mathbf{l})$
12	$EIj^{2-4}(o)$
13	$M^{2-4}(o)$
14	$Q^{2-4}(o)$
15	$N^{2-4}(o)$
16	$EIJ^{1-3}(o)$
17	$EIj^{1-3}(o)$
18	$M^{1-3}(o)$
19	$Q^{1-3}(o)$
20	$N^{1-3}(o)$

; **Y=**

1	$EIJ^{0-1}(\mathbf{l})=0$
2	$EIj^{0-1}(\mathbf{l})=EIj^{1-3}(o)$
3	$M^{0-1}(\mathbf{l})=M^{1-2}(o)+M^{1-3}(o)$
4	$Q^{0-1}(\mathbf{l})=Q^{1-2}(o)+N^{1-3}(o)$
5	$N^{0-1}(\mathbf{l})=N^{1-2}(o)-Q^{1-3}(o)$
6	$EIJ^{1-2}(\mathbf{l})=0$
7	$EIj^{1-2}(\mathbf{l})=EIj^{2-4}(o)$
8	$M^{1-2}(\mathbf{l})=M^{2-4}(o)$
9	$Q^{1-2}(\mathbf{l})=N^{2-4}(o)$
10	$N^{1-2}(\mathbf{l})=-Q^{2-4}(o)$
11	$EIJ^{2-4}(\mathbf{l})=0$
12	$EIj^{2-4}(\mathbf{l})=0$
13	$M^{2-4}(\mathbf{l})$
14	$Q^{2-4}(\mathbf{l})$
15	$N^{2-4}(\mathbf{l})$
16	$EIJ^{1-3}(\mathbf{l})=0$
17	$EIj^{1-3}(\mathbf{l})=0$
18	$M^{1-3}(\mathbf{l})$
19	$Q^{1-3}(\mathbf{l})$
20	$N^{1-3}(\mathbf{l})$

(3.27)

Из анализа матрицы $\mathbf{X}_*$ следует, что в матрице $\mathbf{A}_*$ нужно обнулить 1, 3, 5, 6, 7 и 11 столбцы. На место нулевых строк матрицы $\mathbf{X}_*$ переносим независимые параметры матрицы $\mathbf{Y}$. Зависимые параметры матрицы $\mathbf{Y}$ переносим в матрицу $\mathbf{X}_*$ в соответствии с уравнениями их связи. В матрице $\mathbf{A}_*$ появятся компенсирующие элементы. Разрешающее уравнение задачи статики при $\mathbf{l} = 1\text{м}$ примет вид

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20			
1		1		-1/6																	$M^{2-4}(\ell)$		$-\frac{ql^4}{24} = -\frac{5}{12}$
2		1		-1/2													-1				$EI\varphi^{0-1}(o)$		$-\frac{ql^3}{6} = -\frac{5}{3}$
3				1				-1										-1			$Q^{2-4}(\ell)$		$\frac{ql^2}{2} = 5.0$
4				1					-1											-1	$Q^{0-1}(o)$		$ql = 10.0$
5										-1									1		$N^{2-4}(\ell)$		
6								-1/2	-1/6								1				$M^{1-3}(\ell)$		$-\frac{ql^4}{24} = -\frac{5}{12}$
7								-1	-1/2			-1					1				$Q^{1-3}(\ell)$		$-\frac{ql^3}{6} = -\frac{5}{3}$
8								1	1				-1								$M^{1-2}(o)$		$\frac{ql^2}{2} = 5.0$
9									1						-1						$Q^{1-2}(o)$		$ql = 10.0$
10										1											$N^{1-2}(o)$		
11												1	-1/2	-1/6		1					$N^{1-3}(\ell)$		$-F\frac{(\ell-l/2)^2}{6} = -\frac{5}{4}$
12												1	-1	-1/2							$EI\varphi^{2-4}(o)$		$-F\frac{(\ell-l/2)^2}{2} = -\frac{15}{2}$
13			-1										1	1							$M^{2-4}(o)$		$F\left(1-\frac{l}{2}\right) = 30.0$
14				-1										1							$Q^{2-4}(o)$		$F = 60.0$
15					-1										1						$N^{2-4}(o)$		
16																1	1	-1/2	-1/6		$EI\vartheta^{1-3}(o)$		$-M\frac{(\ell-l/4)^2}{2} = -\frac{45}{8}$
17																	1	-1	-1/2		$EI\varphi^{1-3}(o)$		$-M\left(1-\frac{l}{4}\right) = -15.0$
18						-1												1	1		$M^{1-3}(o)$		$M = 20.0$
19							-1												1		$Q^{1-3}(o)$		
20											-1									1	$N^{1-3}(o)$		

(3.28)

Программа определения граничных параметров рамы запишется следующим образом (в отдельном файле)

a = *zeros* (20,20); *b* = *zeros* (20,1); X = *zeros* (20,1);

a(1,2) = 1; *a*(1,4) = – 1/6; *a*(2,2) = 1; *a*(2,4) = – 1/6; *a*(2,17) = – 1;

a(3,4) = 1; *a*(3,8) = – 1; *a*(3,18) = – 1; *a*(4,4) = 1; *a*(4,9) = – 1; *a*(4,20) = – 1;

a(5,10) = – 1; *a*(5,19) = 1; *b*(1,1) = – 5/12; *b*(2,1) = – 5/3; *b*(3,1) = 5; *b*(4,1) = 10;

a(6,8) = –1/2; *a*(6,9) = –1/6; *a*(6,17) = 1; *a*(7,8) = –1; *a*(7,9) = – 1/2; *a*(7,12) = – 1;

a(7,17) = 1; *a*(8,8) = 1; *a*(8,9) = 1; *a*(8,13) = – 1; *a*(9,9) = 1; ; *a*(9,15) = – 1;

a(10,10) = 1; *a*(10,14) = 1; *b*(6,1) = – 5/12; *b*(7,1) = – 5/3; *b*(8,1) = 5; *b*(9,1) = 10;

a(11,12)=1; *a*(11,13)= –1/2; *a*(11,14)= –1/6; *a*(11,16)=1; *a*(12,12)=1; *a*(12,13)=1;

a(12,14)= –1/2; *a*(13,1) = –1; *a*(13,13)=1; *a*(13,14)=1; *a*(14,3) = –1; ; *a*(14,14)=1;

a(15,5)= –1; *a*(15,15)= 1; *b*(11,1) =–5/4; *b*(12,1) =–15/2; *b*(13,1)=30; *b*(14,1)= 60;

a(16,16)=1; *a*(16,17)=1; *a*(16,18)=–1/2; *a*(16,19)=–1/6; *a*(17,17)=1; *a*(17,18)= –1;

a(17,19)=–1/2; *a*(18,6)= –1; *a*(18,18)=1; *a*(18,19)=1; *a*(19,7)= –1; *a*(19,19)=1;

a(20,11)= –1; *a*(20,20)=1; ; *b*(16,1) =–45/8; *b*(17,1)= –15; ; *b*(18,1)= 20;

X = *a* \ *b*

Значения граничных параметров рамы сведены в таблицу 3.6.

Таблица 3.6

№ п/п	Граничные параметры
1	$M^{2-4}(\mathbf{l})$= X(1,1) = – 22.8591 кНм
2	$EIj^{0-1}(o)$ = X(2,1) = 1.0959 кНм^2
3	$Q^{2-4}(\mathbf{l})$ = X(3,1) = – 59.0041 кН
4	$Q^{0-1}(o)$ = X(4,1) = 9.0752 кН
5	$N^{2-4}(\mathbf{l})$ = X(5,1) = 10.7927 кН
6	$M^{1-3}(\mathbf{l})$= X(6,1) = – 7.2730 кНм
7	$Q^{1-3}(\mathbf{l})$ = X(7,1) = – 0.9959 кН
8	$M^{1-2}(o)$= X(8,1) = – 9.6477 кНм
9	$Q^{1-2}(o)$ = X(9,1) = 20.7927 кН
10	$N^{1-2}(o)$ = X(10,1) = – 0.9959 кН
11	$N^{1-3}(\mathbf{l})$ = X(11,1) = – 21.7175 кН
12	$EIj^{2-4}(o)$= X(12,1) = – 0.8570 кНм^2
13	$M^{2-4}(o)$= X(13,1) = 6.1450 кНм
14	$Q^{2-4}(o)$ = X(14,1) = 0.9959 кН
15	$N^{2-4}(o)$ = X(15,1) = 10.7927 кН
16	$EIJ^{1-3}(o)$= X(16,1) = 2.8455 кНм^3
17	$EIj^{1-3}(o)$= X(17,1) = – 1.7751 кНм^2
18	$M^{1-3}(o)$= X(18,1) = 13.7229 кНм
19	$Q^{1-3}(o)$ = X(19,1) = – 0.9959 кН
20	$N^{1-3}(o)$ = X(20,1) = – 21.7175 кН

3.3.2. *Построение эпюр напряженно-деформированного состояния элементов рамы*

Для построения эпюр программируем выражения (3.12), (3.13)

<u>стержень 0-1</u>

```
x = 0 : 0.001 : 1.0;  q = 10.0;
EIv = – (X(2,1)*x – X(4,1)*x.^ 3 / 6 + q*x.^4 / 24);
EIfi = – (X(2,1) – X(4,1)*x.^ 2 / 2 + q*x.^3 / 6);
Q = X(4,1) – q*x; M = X(4,1)*x – q*x.^ 2 / 2;
subplot (2,2,1), plot (x, EIv); axis ([0 1 – 0.4   0.4]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([0 1 – 2        2]); grid on
subplot (2,2,3), plot (x, Q); axis ([0 1 –  10    10]); grid on
subplot (2,2,4), plot (x, M); axis ([0 1 –  5      5 ]); grid on
```

<u>стержень 1-2</u>

```
x = 0 : 0.001 : 1.0;  q = 10.0;
EIv = – (X(17,1)*x – X(8,1)*x.^ 2 / 2– X(9,1)*x.^ 3 / 6  + q*x.^4 / 24);
EIfi = – (X(17,1) – X(8,1)*x– X(9,1)*x.^ 2 / 2 + q*x.^3 / 6);
Q = X(9,1) – q*x; M = X(8,1) + X(9,1)*x – q*x.^ 2 / 2;
subplot (2,2,1), plot (x, EIv); axis ([0 1 – 0.4   0.4]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([0 1 – 2        2]); grid on
subplot (2,2,3), plot (x, Q); axis ([0 1 –  22    22]); grid on
subplot (2,2,4), plot (x, M); axis ([0 1 –  10    10]); grid on
```

<u>стержень 1-3</u>

а). интервал (0; $\frac{1}{4}\mathbf{l}$) м.

```
x = 0 : 0.001 : 0.75;
EIv = – (X(16,1) + X(17,1)*x– X(18,1)*x.^ 2 / 2 – X(19,1)*x.^ 3 / 6);
EIfi = – (X(17,1) – X(18,1)*x– X(19,1)*x.^ 2 / 2);
Q = X(19,1); M = X(18,1) + X(19,1)*x;

subplot (2,2,1), plot (x, EIv); axis ([0  0.25 – 3   3]); grid on
```

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0 0.25 – 6 6]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0 0.25 – 2 2]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0 0.25 – 14 14]); *grid on*

б). интервал ($\frac{1}{4}$**l** ; **l**) м.

x = 0.25 : 0.001 : 1.0; *m* = 20.0;

EIv=–(X(16,1)+X(17,1)**x*–X(18,1)**x*.^ 2/2–X(19,1)**x*.^ 3/6+*m**(*x*–0.25).^ 2/2);

EI*fi* = – (X(17,1) – X(18,1)**x*– X(19,1)**x*.^ 2 / 2+*m**(*x*–0.25));

Q = X(19,1); M = X(18,1) + X(19,1)**x* – *m*;

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0.25 1.0 – 3 3]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0.25 1.0 – 6 6]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0.25 1.0 – 2 2]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0.25 1.0 – 15 15]); *grid on*

<u>стержень 2-4</u>

а). интервал (0; $\frac{1}{2}$**l**) м.

x = 0 : 0.001 : 0.5;

EIv = – (X(16,1) + X(12,1)**x*– X(13,1)**x*.^ 2 / 2 – X(14,1)**x*.^ 3 / 6);

EI*fi* = – (X(12,1) – X(13,1)**x*– X(14,1)**x*.^ 2 / 2);

Q = X(14,1); M = X(13,1) + X(14,1)**x*;

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0 0.5 – 3 3]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0 0.5 – 6 6]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0 0.5 – 2 2]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0 0.5 – 15 15]); *grid on*

б). интервал (0.5**l** ; **l**) м.

x = 0.5 : 0.001 : 1.0; *f* = 60.0;

EIv=–(X(16,1)+X(12,1)**x*–X(13,1)**x*.^ 2/2–X(14,1)**x*.^ 3/6+ *f* *(*x*–0.5).^ 3/6);

EI*fi* = – (X(12,1) – X(13,1)**x*– X(14,1)**x*.^ 2 / 2+ *f* *(*x*–0.5).^ 2/2);

Q = X(14,1) – *f*; M = X(13,1) + X(14,1)**x* – *f* *(*x*–0.5);

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0.5 1.0 – 3 3]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0.5 1.0 – 6 6]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0.5 1.0 – 60 60]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0.5 1.0 – 25 25]); *grid on*

Эпюры прогибов EI$J(x)$, углов поворота EI$j(x)$, поперечных сил Q(x) и изгибающих моментов M(x) представлены на рис. 3.15

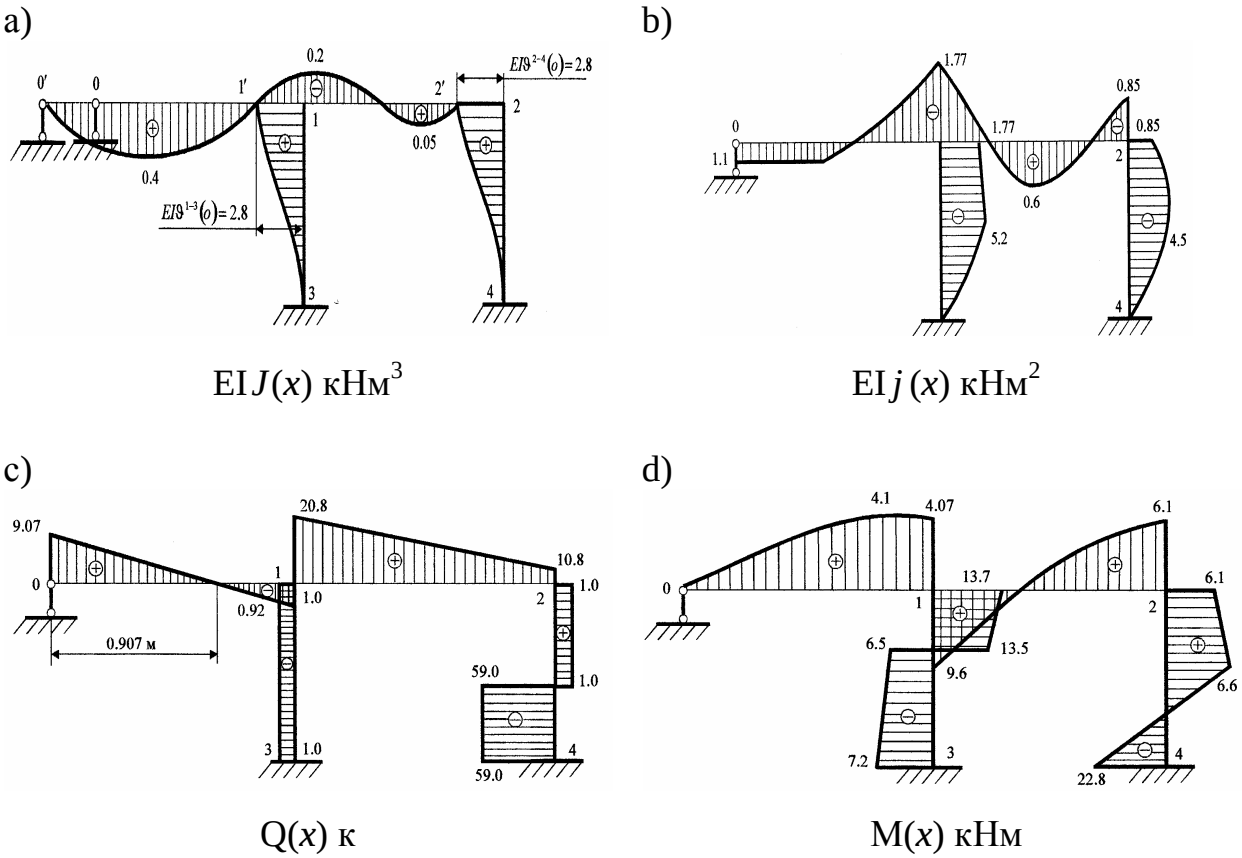


Рис. 3.15

Отметим, что эпюры прогибов EI$J(x)$ позволяют уточнить принятую деформированную схему рамы. Если сравнивать рис. 3.15, а и рис. 3.14, d, то видно, что деформированная схема практически совпадает с действительным состоянием рамы. Последнее обстоятельство в немалой степени обеспечивает достоверность ее расчета.

3.3.3. *Определение частот собственных колебаний*

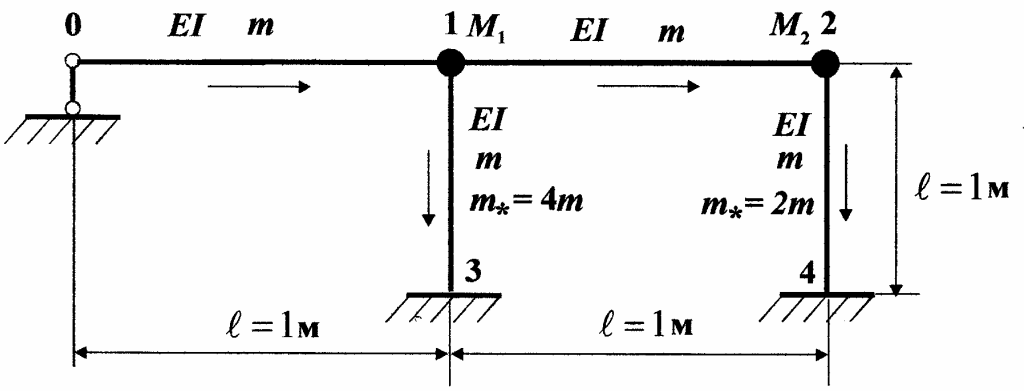


Рис. 3.16

Узлы рамы по рис. 3.16 при свободных колебаниях будут иметь линейные и угловые перемещения, т. е. данная конструкция относится к классу свободных систем. При ее движении возникают силы инерции линейно подвижных стержней 0-1 и 1-2. Учет этих сил инерции можно выполнить по формуле [2]

$$m_* = m + MJ^2(a) / \int_0^1 J^2(x)dx, \qquad (3.29)$$

где $J(x)$ – прогиб несвободного стержня, связанного с линейно подвижным стержнем; М – сосредоточенная масса линейно подвижного стержня; a – координата сосредоточенной массы.

В узле 1 прикладываем сосредоточенную массу стержня 0-1 и половину массы стержня 1-2, в итоге получится $M_1 = 1{,}5m\mathbf{l}$. В узле 2 прикладываем половину массы 1-2, т. е. $M_2 = 0{,}5m\mathbf{l}$ (рис. 3.16).

Для стержней 1-3 и 2-4 принимаем, что прогиб приближенно описывается функцией $J(x) = \cos(px/2\mathbf{l})$, тогда по формуле (3.29) будем иметь

$$m_*^{3-1} = m + 3m = 4m; \quad m_*^{2-4} = m + m = 2m.$$

В дальнейшем, при динамическом анализе данной рамы, необходимо использовать увеличенные массы стержней 1-3 и 2-4, чтобы учесть возникающие силы инерции.

Частоты собственных колебаний рамы определяются из частотного уравнения краевой задачи $|\ \mathbf{A}_*(w)\ | = 0$, где матрица $\mathbf{A}_*(w)$ берется из уравне-

ния (3.28), где достаточно заменить фундаментальные функции статического изгиба на фундаментальные функции поперечных колебаний (3.14).

$$
A_* = \begin{array}{c|cccccccccccccccccccc|l}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & \\
\hline
1 & & A_{12} & & -A_{14} & & & & & & & & & & & & & & & & & \\
2 & & A_{11} & & -A_{13} & & & & & & & & & & & & & -1 & & & & \\
3 & & -\lambda^4 A_{14} & & A_{12} & & & & -1 & & & & & & & & & & -1 & & & 0\text{-}1 \\
4 & & -\lambda^4 A_{13} & & A_{11} & & & & & -1 & & & & & & & & & & & -1 & \\
5 & & & & & & & & & & -1 & & & & & & & & & 1 & & \\
\hline
6 & & & & & & & & -A_{13} & -A_{14} & & & & & & & & A_{12} & & & \\
7 & & & & & & & & -A_{12} & -A_{13} & & & -1 & & & & & A_{11} & & & \\
8 & & & & & & & & A_{11} & A_{12} & & & & -1 & & & & -\lambda^4 A_{14} & & & 1\text{-}2 \\
9 & & & & & & & & \lambda^4 A_{14} & A_{11} & & & & & & -1 & & -\lambda^4 A_{13} & & & \\
10 & & & & & & & & & & 1 & & & & 1 & & & & & & \\
\hline
11 & & & & & & & & & & & & A_{12} & -A_{13} & -A_{14} & & A_{11} & & & & \\
12 & & & & & & & & & & & & A_{11} & -A_{12} & -A_{13} & & \lambda^4 A_{14} & & & & \\
13 & -1 & & & & & & & & & & & -\lambda^4 A_{14} & A_{11} & A_{12} & & -\lambda^4 A_{13} & & & & 2\text{-}4 \\
14 & & & -1 & & & & & & & & & -\lambda^4 A_{13} & \lambda^4 A_{14} & A_{11} & & -\lambda^4 A_{12} & & & & \\
15 & & & & & -1 & & & & & & & & & & 1 & & & & & \\
\hline
16 & & & & & & & & & & & & & & & & A_{11} & A_{12} & -A_{13} & -A_{14} & \\
17 & & & & & & & & & & & & & & & & \lambda^4 A_{14} & A_{11} & -A_{12} & -A_{13} & \\
18 & & & & & & -1 & & & & & & & & & & -\lambda^4 A_{13} & -\lambda^4 A_{14} & A_{11} & A_{12} & & 3\text{-}1 \\
19 & & & & & & & -1 & & & & & & & & & -\lambda^4 A_{12} & -\lambda^4 A_{13} & \lambda^4 A_{14} & A_{11} & & \\
20 & & & & & & & & & & & -1 & & & & & & & & & 1 &
\end{array}
\qquad (3.30)
$$

При поиске частот принимается $EI = m = \mathbf{l} = 1$, так что аргументы фундаментальных функций поперечных колебаний запишутся так:

стержни 0-1; 1-2

$$l^{0-1} = l^{1-2} = \sqrt{w}\,;$$

стержень 1-3 (3.31)

$$l^{1-3} = \sqrt[4]{4w^2}\,;$$

стержень 2-4

$$l^{2-4} = \sqrt[4]{2w^2}\,,$$

где w – частота собственных колебаний. Программа поиска частот принимает вид (обозначения переменных такие же как у неразрезной балки)

```
n = 20; n1 = 300;  am = 0.01; dam = 0.01; X = zeros (n1,1);  Y = zeros (n1,1);
for  m = 1 : n1
a = zeros (n,n);  la1 = sqrt (am);  la2 = sqrt (am); la3 = (2*am^2)^0.25;
la4 = (4*am^2)^0.25;
a(1,2) = (sinh (la1)+ sin(la1))/(2*la1); a(1,4) =–(sinh (la1)– sin(la1))/(2*la1^3);
a(2,2)=(cosh (la1)+cos(la1))/2; a(2,4)=–(cosh(la1)–cos(la1))/(2*la1^2); a(2,17)=–1;
a(3,2) = la1^4*a(2,4);  a(3,4) = a(1,2); a(3,8) = – 1; a(3,18) = – 1;
a(4,2) = la1^4*a(2,4);  a(4,4) = a(2,2); a(4,9) = – 1; a(4,20) = – 1; a(5,10) = – 1;
a(5,19) = 1; a(6,8) = – (cosh(la2) – cos(la2))/(2*la2^2);
a(6,9)=–(sinh(la2)–sin(la2))/(2*la2^3); a(6,17)=(sinh(la2)+sin(la2))/(2*la2);
a(7,8) =– a(6,17); a(7,9) = a(6,8); a(7,12) = – 1; a(7,17)=(cosh(la2)+cos(la2))/2;
a(8,8) = a(7,17); a(8,9) = a(6,17); a(8,13) = – 1; a(8,17) = la2^4*a(6,9);
a(9,8) =– a(8,17); a(9,9) = a(7,17); a(9,15) = – 1; a(9,17) = la2^4*a(6,8);
a(10,10) = 1; a(10,14) = 1; a(11,12)=(sinh(la3)+sin(la3))/(2*la3);
a(11,13)=–(cosh(la3)–cos(la3))/(2*la3^2); a(11,14)=–(sinh(la3)–sin(la3))/(2*la3^3);
a(11,16)=(cosh(la3)+cos(la3))/2; a(12,12) = a(11,16); a(12,13) =– a(11,12);
a(12,14)=a(11,13); a(12,16)=– la3^4*a(11,14); a(13,1)=–1; a(13,12)=– a(12,16);
a(13,13)=a(12,12); a(13,14)=a(11,12); a(13,16)= la3^4*a(11,13); a(13,1)=–1;
a(14,12)=a(13,16); a(14,13)=–a(13,12); a(14,14)=a(12,12); a(14,15)=–la3^4*a(12,13);
a(15,5) = – 1; a(15,15) = 1; a(16,16)=(cosh(la4)+cos(la4))/2;
a(16,17)=(sinh(la4)+sin(la4))/(2*la4); a(16,18)=–(cosh(la4)–cos(la4))/(2*la4^2);
a(16,19)=–(sinh(la4)–sin(la4))/(2*la4^3); a(17,16)=– la4^4*a(16,19);
a(17,17)=a(16,16); a(17,18)=–a(16,17); a(17,19)=a(16,18); a(18,6) = – 1;
a(18,16)= la4^4*a(16,18); ); a(18,17)=–a(17,16);
a(18,18)=a(16,16); a(18,19)=a(16,17); a(19,7) = – 1; a(19,16)=– la4^4*a(16,17);
a(19,17)=a(18,16); a(19,18)=a(17,16); a(19,19)=a(16,16); a(20,11) = – 1;
a(20,20) = 1; d = det(a); X(m,1) = am; Y(m,1) = d; am = am + dam; end;
plot (X,Y); grid on
format long e
[X Y]
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20				
1		A_{12}		$-A_{14}$																	$M^{2-4}(\ell)$			
2		A_{11}		$-A_{13}$													-1				$EI\varphi^{0-1}(o)$			
3		$-\lambda^4 A_{14}$		A_{12}				-1										-1			$Q^{2-4}(\ell)$			
4		$-\lambda^4 A_{13}$		A_{11}					-1											-1	$Q^{0-1}(o)$			
5										-1									1		$N^{2-4}(\ell)$			
6								$-A_{13}$	$-A_{14}$								A_{12}				$M^{1-3}(\ell)$			
7								$-A_{12}$	$-A_{13}$			-1					A_{11}				$Q^{1-3}(\ell)$			
8								A_{11}	A_{12}				-1				$-\lambda^4 A_{14}$				$M^{1-2}(o)$			
9								$\lambda^4 A_{14}$	A_{11}						-1		$-\lambda^4 A_{13}$				$Q^{1-2}(o)$			
10										1				1							$N^{1-2}(o)$			
11												A_{12}	$-A_{13}$	$-A_{14}$		A_{11}					$N^{1-3}(\ell)$	$=$	$EI\vartheta^{2-4}(\ell) = -1$
12												A_{11}	$-A_{12}$	$-A_{13}$		$\lambda^4 A_{14}$					$EI\varphi^{2-4}(o)$		
13	-1											$-\lambda^4 A_{14}$	A_{11}	A_{12}		$-\lambda^4 A_{13}$					$M^{2-4}(o)$		
14			-1									$-\lambda^4 A_{13}$	$\lambda^4 A_{14}$	A_{11}		$-\lambda^4 A_{12}$					$Q^{2-4}(o)$		
15					-1										1						$N^{2-4}(o)$		
16																A_{11}	A_{12}	$-A_{13}$	$-A_{14}$		$EI\vartheta^{1-3}(o)$		$EI\vartheta^{1-3}(\ell) = -1$
17																$\lambda^4 A_{14}$	A_{11}	$-A_{12}$	$-A_{13}$		$EI\varphi^{1-3}(o)$		
18						-1										$-\lambda^4 A_{13}$	$-\lambda^4 A_{14}$	A_{11}	A_{12}		$M^{1-3}(o)$		
19							-1									$-\lambda^4 A_{12}$	$-\lambda^4 A_{13}$	$\lambda^4 A_{14}$	A_{11}		$Q^{1-3}(o)$		
20											-1									1	$N^{1-3}(o)$		

(3.32)

Результаты поиска частот собственных колебаний рамы

$$w_1 = 2.946045\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}; \quad w_2 = 9.431095\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}; \quad w_3 = 13.959385\mathbf{l}^{-2}\sqrt{\frac{EI}{m}};$$

$$w_4 = 15.259475\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}; \quad w_5 = 19.489495\mathbf{l}^{-2}\sqrt{\frac{EI}{m}} \quad \frac{1}{\text{сек}}.$$

3.3.4. *Построение форм собственных колебаний*

Для построения форм собственных колебаний необходимо определить граничные параметры каждого стержня рамы при собственных колебаниях. Воспользуемся кинематическим способом возбуждения колебаний рамы и сообщим опорам рамы единичное перемещение, т. е. $b(11,1) = \mathrm{EI}J^{2\text{-}4}(\mathbf{l}) = b(16,1) = \mathrm{EI}J^{1\text{-}3}(\mathbf{l}) = -1$.

Разрешающее уравнение МГЭ рамы примет вид (3.32), где использован кинематический способ возбуждения собственных колебаний, т. е. задается одинаковое единичное смещение неподвижных опор рамы.

Программа определения относительных граничных параметров (они нормируются относительно начального перемещения $\mathrm{EI}J^{1\text{-}3}(o) = \mathrm{EI}J^{2\text{-}4}(o)$) в отдельном М-файле имеет следующий текст

```
a = zeros (20,20);  b = zeros (20,1);  X = zeros (20,1); x = 2.946045;
la1 = sqrt (x); la2 = sqrt (x); la3 = (2*x^2)^0.25; la4 = (4*x^2)^0.25;
a(1,2) = (sinh (la1)+ sin(la1))/(2*la1); a(1,4) =–(sinh (la1)– sin(la1))/(2*la1^3);
a(2,2)=(cosh (la1)+cos(la1))/2; a(2,4)=–(cosh(la1)–cos(la1))/(2*la1^2); a(2,17)=–1;
a(3,8)=–1; a(3,2)=la1^4*a(1,4); a(3,4)=a(1,2); a(3,18)=–1; a(4,2)=la1^4*a(2,4);
a(4,9)=–1; a(4,4)=a(2,2); a(4,20)=–1; a(5,10)=–1; ; a(5,19)=1;
a(6,17)=(sinh(la2)+sin(la2))/(2*la2); a(6,8) = – (cosh(la2) – cos(la2))/(2*la2^2);
a(6,9)=–(sinh(la2)–sin(la2))/(2*la2^3); a(7,17)=(cosh(la2)+cos(la2))/2;
a(7,8) =– a(6,17); a(7,9) = a(6,8); a(7,12) = – 1; a(8,13) = – 1;
a(8,17) = la2^4*a(6,9); a(8,8) = a(7,17); a(8,9) = a(6,17); a(9,15) = – 1;
a(9,17) = la2^4*a(7,9); a(9,8) =– a(8,17); a(9,9) = a(7,17); a(10,14) = 1;
a(10,10) = 1; a(11,12)=(sinh(la3)+sin(la3))/(2*la3);
```

a(11,13)=–(*cosh*(*la*3)–*cos*(*la*3))/(2**la*3^2); *a*(11,14)=–(*sinh*(*la*3)–*sin*(*la*3))/(2**la*3^3);

a(11,16)=(*cosh*(*la*3)+*cos*(*la*3))/2; *a*(12,12) = *a*(11,16); *a*(12,13) =– *a*(11,12);

a(12,14)=*a*(11,13); *a*(12,16)=– *la*3^4**a*(11,14); *a*(13,1)=–1;

a(13,12)=– *a*(12,16); *a*(13,13)=*a*(11,16); *a*(13,14)=*a*(11,12);

a(13,16)= *la*3^4**a*(11,13); *a*(14,12)=*a*(13,16); *a*(14,13)=*a*(12,16);

a(14,14)=*a*(11,16); *a*(14,3)=–1; *a*(14,16)=*la*3^4**a*(12,13);

a(15,5) = – 1; *a*(15,15) = 1; *a*(16,16)=(*cosh*(*la*4)+*cos*(*la*4))/2;

a(16,17)=(*sinh*(*la*4)+*sin*(*la*4))/(2**la*4); *a*(16,18)=–(*cosh*(*la*4)–*cos*(*la*4))/(2**la*4^2);

a(16,19)=–(*sinh*(*la*4)–*sin*(*la*4))/(2**la*4^3); *a*(17,16)=– *la*4^4**a*(16,19);

a(17,17)=*a*(16,16); *a*(17,18)=–*a*(16,17); *a*(17,19)=*a*(16,18);

a(18,6) = – 1; *a*(18,16)= *la*4^4**a*(16,18); *a*(18,17)=–*a*(17,16); *a*(18,18)=*a*(16,16);

a(18,19)=*a*(16,17); *a*(19,7) = – 1; *a*(19,16)= *la*4^4**a*(17,18); *a*(19,17)=*a*(18,16);

a(19,18)=*a*(17,16); *a*(19,19)=*a*(16,16); *a*(20,11) = – 1; *a*(20,20) = 1;

b(11,1)= –1; *b*(16,1)= –1; X = *a*\ *b*; X = X / X(16,1)

Значения относительных граничных параметров сведены в таблицу 3.7.

Таблица 3.7

№ п/п	Граничные параметры рамы	Относительные значения граничных параметров при собственных частотах					
		w_1=2,946045		w_2= 9,431095	w_3= 13,959385	w_4= 15.259475	w_5= 19.489495
		абсолютные	относительные				
1	$M^{2-4}(\mathbf{l})$= X(1,1)=	$0.4361\cdot10^7$	-5.3246	-43.0898	120.7308	23.4396	86.2844
2	$EIj^{0-1}(o)$=X(2,1)=	$-0.0120\cdot10^7$	0.1463	8.0671	43.7059	-0.5996	-6.7905
3	$Q^{2-4}(\mathbf{l})$=X(3,1)=	$0.8623\cdot10^7$	-10.5288	-155.6993	531.9886	110.7758	449.5673
4	$Q^{0-1}(o)$=X(4,1)=	$-0.0763\cdot10^7$	0.9316	77.0857	578.3407	-8.6533	-126.3574
5	$N^{2-4}(\mathbf{l})$=X(5,1)=	$-0.4059\cdot10^7$	4.9558	77.7808	81.0184	-69.5101	447.6958
6	$M^{1-3}(\mathbf{l})$=X(6,1)=	$0.5355\cdot10^7$	-6.5385	50.4911	-62.0186	42.4069	11.4058
7	$Q^{1-3}(\mathbf{l})$=X(7,1)=	$1.2201\cdot10^7$	-14.9016	220.3370	-320.1960	234.3273	72.8153
8	$M^{1-2}(o)$=X(8,1)=	$0.1871\cdot10^7$	-2.2842	-6.7060	-65.7629	-26.2838	83.1398
9	$Q^{1-2}(o)$=X(9,1)=	$-0.4241\cdot10^7$	5.1781	-60.1557	98.3542	103.1128	-443.2657
10	$N^{1-2}(o)$=X(10,1)=	$-0.2047\cdot10^7$	2.4991	-120.4025	420.3761	8.3444	241.4304
11	$N^{1-3}(\mathbf{l})$=X(11,1)=	$0.3733\cdot10^7$	-4.5580	-10.8130	-923.8486	-90.5222	604.9033
12	$EIj^{2-4}(o)$=X(12,1)=	$0.0475\cdot10^7$	-0.5802	6.9230	-12.0678	-4.9911	2.9949
13	$M^{2-4}(o)$=X(13,1)=	$-0.2330\cdot10^7$	2.8449	3.4289	63.5624	0.6446	90.1745
14	$Q^{2-4}(o)$=X(14,1)=	$0.2047\cdot10^7$	-2.4991	120.4025	-420.3761	-8.3444	-241.4304
15	$N^{2-4}(o)$=X(15,1)=	$-0.4059\cdot10^7$	4.9558	77.7808	81.0184	-69.5101	447.6958
16	$EIJ^{1-3}(o)$=X(16,1)=	$-0.0819\cdot10^7$	1.0000	1.0000	1.0000	1.0000	1.0000
17	$EIj^{1-3}(o)$=X(17,1)=	$0.0227\cdot10^7$	0.2776	-8.6753	-11.3875	0.0157	-4.4001
18	$M^{1-3}(o)$=X(18,1)=	$-0.2514\cdot10^7$	3.0696	10.2235	-112.4317	29.4391	-27.1176
19	$Q^{1-3}(o)$=X(19,1)=	$-0.2047\cdot10^7$	2.4991	-120.4025	420.3761	8.3444	241.4304
20	$N^{1-3}(o)$=X(20,1)=	$0.3733\cdot10^7$	-4.5580	-10.8130	-923.8486	-90.5222	604.9033

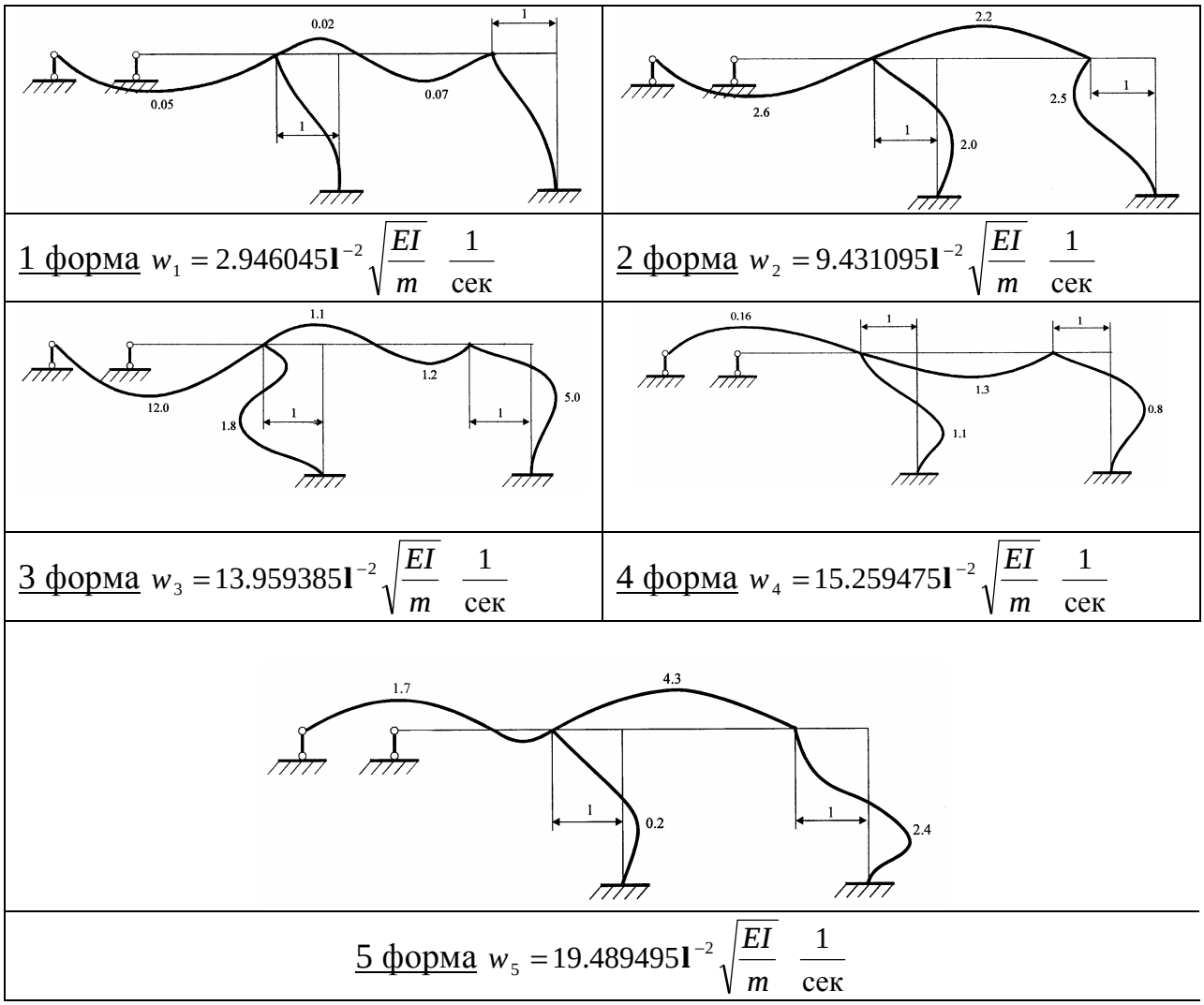
Из таблицы следует, что единичные перемещения опор приводят к весьма значительным усилиям и перемещениям в элементах рамы.

Формы колебаний стержней рамы, построенные по соотношениям метода начальных параметров (3.17) представлены на рис. 3.17. Протокол их построения запишется следующим образом.

Протокол построения форм колебаний принимает вид

```
x = 0 : 0.001 : 1.0;  y = 2.946045; la1 = sqrt (y); la2 = sqrt (y);
la3 = (2*y^2)^0.25; la4 = (4*y^2)^0.25;
EIv1 = – (X(2,1)*(sinh(la1*x)+sin(la1*x))/(2* la1) – X(4,1)*(sinh(la1*x) – …
sin(la1*x))/(2* la1^3));
EIv2 = – (X(17,1)*(sinh(la2*x)+sin(la2*x))/(2* la2) – X(8,1)*(cosh(la2*x) – …
cos(la2*x))/(2* la2^2) – X(9,1)*(sinh(la2*x)–sin(la2*x))/(2* la2^3));
EIv3 = – (X(16,1)*(cosh(la3*x)+cos(la2*x))/2+ X(12,1)*(sinh(la3*x)+ …
sin(la3*x))/(2* la3) – X(13,1)* (cosh(la3*x)–cos(la*x))/(2* la3^2) – …
X(14,1)*(sinh(la3*x)–sin(la3*x))/(2* la3^3));
Eiv4 = – (X(16,1)*(cosh(la4*x)+cos(la4*x))/2 + X(17,1)*(sinh(la4*x) + …
sin(la4*x))/(2*la4) – X(18,1)*(cosh(la4*x)–cos(la4*x))/(2* la4^2) – …
X(19,1)*(sinh(la4*x)–sin(la4*x))/(2* la4^3));
subplot (2,2,1), plot (x, EIv1); axis ([0 1 – 0.1  0.1]); grid on
subplot (2,2,2), plot (x, EIv2); axis ([0 1 – 0.1  0.1]); grid on
subplot (2,2,3), plot (x, EIv3); axis ([0 1  –   1     1]); grid on
subplot (2,2,4), plot (x, EIv4); axis ([0 1  –   1     1]); grid on
```

Как и при расчете неразрезной балки, перед выполнением данного протокола в окно команд помещается вектор относительных граничных параметров X, также необходимо корректировать масштабы форм колебаний.



1 форма $w_1 = 2.946045\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}\ \frac{1}{\text{сек}}$

2 форма $w_2 = 9.431095\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}\ \frac{1}{\text{сек}}$

3 форма $w_3 = 13.959385\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}\ \frac{1}{\text{сек}}$

4 форма $w_4 = 15.259475\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}\ \frac{1}{\text{сек}}$

5 форма $w_5 = 19.489495\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}\ \frac{1}{\text{сек}}$

Рис. 3.17

Отметим, что данный расчет спектра частот и форм собственных колебаний кинематическим способом является *основой расчета* сооружений *на сейсмические воздействия*. Можно заключить, что представленная методика по МГЭ является наиболее простой и эффективной по сравнению с имеющимися [3].

3.3.5. *Расчет на вынужденные колебания*

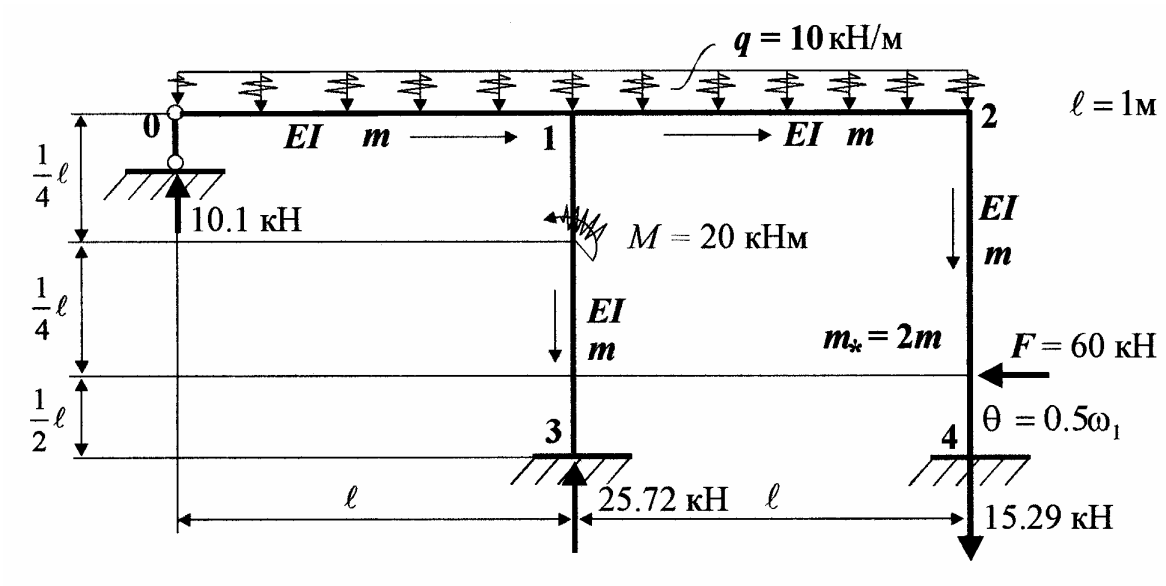


Рис.3.18

Для определения напряженно-деформированного состояния рамы при вынужденных колебаниях с частотой $q = 0.5w_1$ принимаем ориентированный граф по рис. 3.18 с тем, чтобы можно было использовать уже запрограммированную матрицу коэффициентов $\mathbf{A}_*$, добавив к ней элементы матрицы нагрузки **B** по формулам (3.18 – 3.20). Коэффициенты фундаментальных функций с учетом сил инерции линейно подвижных стержней примут вид (3.31), где вместо w нужно подставить частоту вынужденных колебаний. Согласно алгоритму МГЭ разбиваем раму на 4 стержня, нумеруем узлы и стрелками обозначаем начало и конец каждого стержня. Расчет рамы на вынужденные колебания сводится к определению динамических граничных параметров из матричного уравнения $\mathbf{A}_* \mathbf{X}_* = -\mathbf{B}$. Программа решения данного уравнения записывается в отдельном М-файле. После решения матричного уравнения граничные параметры используются для построения эпюр напряженно-деформированного состояния стержней рамы. Для ухода от резонанса частоту вынужденных колебаний принимаем равной

$$q = 0.5w_1 = 0.5 \cdot 2.946045\mathbf{l}^{-2}\sqrt{\frac{EI}{m}}\ \frac{1}{\text{сек}}.$$

Разрешающее уравнение краевой задачи рамы для орграфа по рис. 3.18 принимает вид

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20			
1		A_{12}		$-A_{14}$																	$M^{2-4}(\ell)$	=	$-q(ch\lambda\ell+\cos\lambda\ell-2)/2\lambda^4$
2		A_{11}		$-A_{13}$													-1				$EI\varphi^{0-1}(o)$		$-q(sh\lambda\ell-\sin\lambda\ell)/2\lambda^3$
3		$-\lambda^4 A_{14}$		A_{12}				-1										-1			$Q^{2-4}(\ell)$		$q(ch\lambda\ell-\cos\lambda\ell)/2\lambda^2$
4		$-\lambda^4 A_{13}$		A_{11}					-1											-1	$Q^{0-1}(o)$		$q(sh\lambda\ell+\sin\lambda\ell)/2\lambda$
5										-1									1		$N^{2-4}(\ell)$		
6								$-A_{13}$	$-A_{14}$								A_{12}				$M^{1-3}(\ell)$		$-q(ch\lambda\ell+\cos\lambda\ell-2)/2\lambda^4$
7								$-A_{12}$	$-A_{13}$			-1					A_{11}				$Q^{1-3}(\ell)$		$-q(sh\lambda\ell-\sin\lambda\ell)/2\lambda^3$
8								A_{11}	A_{12}				-1				$-\lambda^4 A_{14}$				$M^{1-2}(o)$		$q(ch\lambda\ell-\cos\lambda\ell)/2\lambda^2$
9								$\lambda^4 A_{14}$	A_{11}						-1		$-\lambda^4 A_{13}$				$Q^{1-2}(o)$		$q(sh\lambda\ell+\sin\lambda\ell)/2\lambda$
10										1				1							$N^{1-2}(o)$		
11												A_{12}	$-A_{13}$	$-A_{14}$		A_{11}					$N^{1-3}(\ell)$		$-F\left(sh\frac{\lambda\ell}{2}-\sin\frac{\lambda\ell}{2}\right)/2\lambda^3$
12												A_{11}	$-A_{12}$	$-A_{13}$		$\lambda^4 A_{14}$					$EI\varphi^{2-4}(o)$		$-F\left(ch\frac{\lambda\ell}{2}-\cos\frac{\lambda\ell}{2}\right)/2\lambda^2$
13	-1											$-\lambda^4 A_{14}$	A_{11}	A_{12}		$-\lambda^4 A_{13}$					$M^{2-4}(o)$		$F\left(sh\frac{\lambda\ell}{2}+\sin\frac{\lambda\ell}{2}\right)/2\lambda$
14			-1									$-\lambda^4 A_{13}$	$\lambda^4 A_{14}$	A_{11}		$-\lambda^4 A_{12}$					$Q^{2-4}(o)$		$F\left(ch\frac{\lambda\ell}{2}+\cos\frac{\lambda\ell}{2}\right)/2$
15					-1										1						$N^{2-4}(o)$		
16																A_{11}	A_{12}	$-A_{13}$	$-A_{14}$		$EI\vartheta^{1-3}(o)$		$-M\left(ch\frac{\lambda\ell 3}{4}-\cos\frac{\lambda\ell 3}{4}\right)/2\lambda^2$
17																$\lambda^4 A_{14}$	A_{11}	$-A_{12}$	$-A_{13}$		$EI\varphi^{1-3}(o)$		$-M\left(sh\frac{\lambda\ell 3}{4}+\sin\frac{\lambda\ell 3}{4}\right)/2\lambda$
18						-1										$-\lambda^4 A_{13}$	$-\lambda^4 A_{14}$	A_{11}	A_{12}		$M^{1-3}(o)$		$M\left(ch\frac{\lambda\ell 3}{4}+\cos\frac{\lambda\ell 3}{4}\right)/2$
19							-1									$-\lambda^4 A_{12}$	$-\lambda^4 A_{13}$	$\lambda^4 A_{14}$	A_{11}		$Q^{1-3}(o)$		$M\lambda\left(sh\frac{\lambda\ell 3}{4}-\sin\frac{\lambda\ell 3}{4}\right)/2$
20											-1									1	$N^{1-3}(o)$		

(3.33)

Программа вычисления граничных параметров рамы имеет следующий текст

```
q = 10.0;  m = 20.0;  f = 60.0;
a = zeros (20,20);  b = zeros (20,1);  X = zeros (20,1); t =0.5* 2.946045;
la1 = sqrt (t); la2 = sqrt (t); la3 = (2*t^2)^0.25; la4 = (4*t^2)^0.25;
a(1,2) = (sinh (la1)+ sin(la1))/(2*la1); a(1,4) =–(sinh (la1)– sin(la1))/(2*la1^3);
a(2,2)=(cosh (la1)+cos(la1))/2; a(2,4)=–(cosh(la1)–cos(la1))/(2*la1^2);
a(2,17)=–1; a(3,8)=–1; a(3,2)=la1^4*a(1,4); a(3,4)=a(1,2); a(3,18)=–1;
a(4,2)=la1^4*a(2,4); a(4,4)=a(2,2); a(4,9)=–1; a(4,20)=–1; a(5,10)=–1; a(5,19)=1;
b(1,1)= – q*(cosh (la1)+cos(la1) –2)/(2*la1^4);
b(2,1)= – q*(sinh (la1)– sin(la1))/(2*la1^3);
b(3,1)= q*(cosh (la1) –cos(la1))/(2*la1^2);
b(4,1)= q*(sinh (la1)+ sin(la1))/(2*la1);
a(6,17)=(sinh(la2)+sin(la2))/(2*la2); a(6,8) = – (cosh(la2) – cos(la2))/(2*la2^2);
a(6,9)=–(sinh(la2)–sin(la2))/(2*la2^3); a(7,17)=(cosh(la2)+cos(la2))/2;
a(7,8) =– a(6,17); a(7,9) = a(6,8); a(7,12) = – 1; a(8,13) = – 1;
a(8,17) = la2^4*a(6,9); a(8,8) = a(7,17); a(8,9) = a(6,17); a(9,15) = – 1;
a(9,17) = la2^4*a(6,8); a(9,8) =– a(8,17); a(9,9) = a(7,17); a(10,14) = 1;
a(10,10) = 1; b(6,1)= – q*(cosh (la2)+cos(la2) –2)/(2*la2^4);
b(7,1)= – q*(sinh (la2)– sin(la2))/(2*la2^3);
b(8,1)= q*(cosh (la2) –cos(la2))/(2*la2^2);
b(9,1)= q*(sinh (la2)+ sin(la2))/(2*la2);
a(11,12)=(sinh(la3)+sin(la3))/(2*la3); a(11,13)=–(cosh(la3)–cos(la3))/(2*la3^2);
a(11,14)=–(sinh(la3)–sin(la3))/(2*la3^3); a(11,16)=(cosh(la3)+cos(la3))/2;
a(12,12)=a(11,16); a(12,13)=–a(11,12); a(12,14)=a(11,13); a(12,16)=–la3^4*a(11,14);
a(13,1)=–1; a(13,12)=– a(12,16); a(13,13)=a(11,16); a(13,14)=a(11,12);
a(13,16)= la3^4*a(11,13); a(14,12)=a(13,16); a(14,13)=a(12,16);
a(14,14)=a(11,16); a(14,3)=–1; a(14,16)=la3^4*a(12,13);
a(15,5) =–1; a(15,15)=1; b(11,1)=–f*(sinh (la3/2)– sin(la3/2))/(2*la3^3);
```

```
b(12,1)= – f*(cosh (la3/2) –cos(la3/2) –2)/(2*la3^2);
b(13,1)= f*(sinh (la3/2)+ sin(la3/2))/(2*la3);
b(14,1)= f*(cosh (la3/2) + cos(la3/2))/2;
 a(16,16)=(cosh(la4)+cos(la4))/2; a(16,17)=(sinh(la4)+sin(la4))/(2*la4);
a(16,18)=–(cosh(la4)–cos(la4))/(2*la4^2); a(16,19)=–(sinh(la4)–sin(la4))/(2*la4^3);
a(17,16)=– la4^4*a(16,19); a(17,17)=a(16,16); a(17,18)=–a(16,17); a(17,19)=a(16,18);
a(18,6)=–1; a(18,16)=la4^4*a(16,18); a(18,17)=–a(16,17); a(18,18)=a(16,16);
a(18,19)=a(16,17); a(19,7) = – 1; a(19,16)=–la4^4*a(17,18); a(19,17)=a(18,16);
a(19,18)=a(17,16); a(19,19)=a(16,16); a(20,11) = – 1; a(20,20) = 1;
b(16,1)= – m*(cosh (3*la4/4) –cos(3*la4/4))/(2*la4^2);
b(17,1)=–m*(sinh (3*la4/4) + sin(3*la4/4))/(2*la4);
b(18,1)= m*(cosh (3*la4/4) + cos(3*la4/4))/2;
b(19,1)= m*(la4sinh (3*la4/4) – sin(3*la4/4))/2;
 X = a\ b
```

Результаты расчета динамических граничных параметров рамы сведены в таблицу 3.8.

Таблица 3.8

1	$M^{2-4}(\mathbf{l})$= X(1,1) = – 27.6953 кНм
2	$EIj^{0-1}(o)$ = X(2,1) = 1.2487 кНм^2
3	$Q^{2-4}(\mathbf{l})$ = X(3,1) = – 68.9032 кН
4	$Q^{0-1}(o)$ = X(4,1) = 10.1048 кН
5	$N^{2-4}(\mathbf{l})$ = X(5,1) = 15.2920 кН
6	$M^{1-3}(\mathbf{l})$= X(6,1) = – 12.7248 кНм
7	$Q^{1-3}(\mathbf{l})$ = X(7,1) = – 12.8813 кН
8	$M^{1-2}(o)$= X(8,1) = – 11.6889 кНм
9	$Q^{1-2}(o)$ = X(9,1) = 25.1927 кН
10	$N^{1-2}(o)$ = X(10,1) = 0.5814 кН
11	$N^{1-3}(\mathbf{l})$ = X(11,1) = – 25.7170 кН
12	$EIj^{2-4}(o)$= X(12,1) = – 1.3367 кНм^2
13	$M^{2-4}(o)$= X(13,1) = 8.6147 кНм
14	$Q^{2-4}(o)$ = X(14,1) = – 0.5814 кН
15	$N^{2-4}(o)$ = X(15,1) = 15.2920 кН
16	$EIJ^{1-3}(o)$= X(16,1) = 3.7219 кНм^3
17	$EIj^{1-3}(o)$= X(17,1) = – 2.0502 кНм^2
18	$M^{1-3}(o)$= X(18,1) = 16.4936 кНм
19	$Q^{1-3}(o)$ = X(19,1) = 0.5814 кН
20	$N^{1-3}(o)$ = X(20,1) = – 25.7170 кН

3.3.6. *Построение эпюр напряженно-деформированного состояния при вынужденных колебаниях*

Эпюры прогибов EIJ(x), углов поворота EIj (x), поперечных сил Q(x), изгибающих моментов M(x) строятся по соотношениям метода начальных параметров (3.21), (3.22). Протоколы построения эпюр EIJ(x), EIj (x), Q(x) и M(x) примут вид

<u>стержень 0-1</u>

```
x = 0 : 0.001 : 1.0; t = 0.5*2.946045; la1 = sqrt (t); q = 10.0;
EIv = – (X(2,1)*(sinh(la1*x)+sin(la1*x))/(2*la1) – X(4,1)*(sinh …
(la1*x)–sin(la1*x))/(2*la1^3) +q*(cosh(la1*x)+cos(la1*x) –2) / (2*la1^4);
EIfi = – (X(2,1)*(cosh(la1*x)+cos(la1*x)) / 2 – X(4,1)* (cosh(la1*x) –…
cos(la1*x))/(2*la1^2)+q*(sinh(la1*x)–sin(la1*x))/(2*la1^3));
Q = – X(2,1)*la1^4*(cosh(la1*x)–cos(la1*x)) / (2* la1^2) + …
X(4,1)*(cosh(la1*x)+cos(la1*x)) / 2– q*(sinh(la1*x)+sin(la1*x))/(2*la1);
M = – X(2,1)*la1^4*(sinh(la1*x)–sin(la1*x))/(2* la1^3) + …
X(4,1)*(sinh(la1*x)+sin(la1*x))/(2*la1)–q*(cosh(la1*x)–cos(la1*x))/(2*la1^2);
subplot (2,2,1), plot (x, EIv); axis ([0 1 – 0.6  0.6]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([0 1 – 2   2]); grid on
subplot (2,2,3), plot (x, Q); axis ([0 1  – 12  12]); grid on
subplot (2,2,4), plot (x, M); axis ([0 1  – 6    6 ]); grid on
```

<u>стержень 1-2</u>

```
x = 0 : 0.001 : 1.0; t = 0.5*2.946045; la2 = sqrt (t); q = 10.0;
EIv = – (X(17,1)*(sinh(la2*x)+sin(la2*x))/(2*la2) – X(8,1)*(cosh (…
la2*x)–cos(la2*x))/(2*la2^3)–X(9,1)*(sinh(la2*x)–sin(la2*x)) / …
(2*la2^3)+q*(cosh(la2*x)+cos(la2*x) –2) / (2*la2^4));
EIfi = – (X(17,1)*(cosh(la2*x)+cos(la2*x)) / 2 – X(8,1)*(sinh(la2*x)+ …
sin(la2*x))/(2*la2) –X(9,1)*(cosh(la2*x) –cos(la2*x))/(2*la2^2)+ …
q*(sinh(la2*x)–sin(la2*x))/(2*la2^3));
```

Q =–X(17,1)*la2^4*(cosh(la2*x)–cos(la2*x))/(2*la2^2)+X(8.1)*la2^4* …

(sinh(la2*x)–sin(la2*x))/(2*la2^3)–X(9,1)*(cosh(la2*x)+cos(la2*x)) / 2 …

– q*(sinh(la2*x)+sin(la2*x))/(2*la2);

M=–X(17,1)*la2^4*(sinh(la2*x)–sin(la2*x))/(2*la2^3)+X(8,1)*(cosh(la2*x)+…

cos(la2*x)) / 2+X(9,1)*(sinh(la2*x)+sin(la2*x))/(2*la2)–q*(cosh(la1*x)– …

cos(la2*x))/(2*la2^2);

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0 1 – 0.4 0.4]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0 1 – 2 2]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0 1 – 28 28]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0 1 – 12 12]); *grid on*

<u>стержень 2-4</u>

а) При $0 \le x \le \mathbf{l}/2$ м;

x = 0 : 0.001 : 0.5; *t* = 0.5*2.946045; *la*3 = (2*t^2)^0.25;

EIv = – (X(16,1)*(cosh(la3*x)+cos(la3*x))/2+X(12,1)*(sinh(la3*x)+ …

sin(la3*x))/(2*la3) – X(13,1)*(cosh(la3*x)–cos(la3*x))/(2*la3^2) – …

X(14,1)*(sinh(la3*x)–sin(la3*x))/(2*la3^3));

EI*fi* = – (X(16,1)*la3^4*(sinh(la3*x) –sin(la3*x))/(2*la3^3)+X(12,1)*(…

cosh(la3*x)+cos(la3*x))/2–X(13,1)*(sinh(la3*x)+sin(la3*x))/(2*la3)– …

X(14,1)*(cosh(la3*x)–cos(la3*x))/(2*la3^3));

Q = – X(16,1)*la3^4*(sinh(la3*x)+sin(la3*x))/(2*la3)–X(12,1)*(…

cosh(la3*x)–cos(la3*x))*la3^4/(2*la3^2) + X(13,1)*la3^4*(…

sinh(la3*x) –sin(la3*x))/(2*la3^3)+X(14,1)*(cosh(la3*x)+cos(la3*x))/2;

M = – X(16,1)*la3^4*(cosh(la3*x)–cos(la3*x))/(2*la3^2)–X(12,1)*la3^4* …

(sinh(la3*x)–sin(la3*x))/(2* la3^3)+X(13,1)*(cosh(la3*x)+cos(la3*x))/2+…

X(14,1)*(sinh(la3*x)+sin(la3*x))/(2*la3);

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0 0.5 – 4 4]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0 0.5 – 6 6]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0 0.5 – 10 10]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0 0.5 – 10 10]); *grid on*

б) При $0.5\mathbf{l} \le x \le \mathbf{l}$ м;

x = 0.5 : 0.001 : 1.0; *t* = 0.5*2.946045; *la*3 = (2**t*^2)^0.25; *f* = 60.0;

EIv = – (X(16,1)*(cosh(*la*3**x*)+cos(*la*3**x*))/2+X(12,1)*(sinh(*la*3**x*)+ …

sin(*la*3**x*))/(2**la*3) – X(13,1)*(cosh(*la*3**x*)–cos(*la*3**x*))/(2**la*3^2) – …

X(14,1)*(sinh(*la*3**x*)–sin(*la*3**x*))/(2**la*3^3)+*f**(sinh(*la*3*(*x*–0.5)) – …

sinh(*la*3*(*x*–0.5)))/(2**la*3^3));

EI*fi* = – (X(16,1)**la*3^4*(sinh(*la*3**x*) –sin(*la*3**x*))/(2**la*3^3)+ …

X(12,1)*(cosh(*la*3**x*)+cos(*la*3**x*))/2– X(13,1)*(sinh(*la*3**x*)+sin(*la*3**x*))/ …

(2**la*3) – X(14,1)*(cosh(*la*3**x*)–cos(*la*3**x*))/(2**la*3^2)+ …

f *(cosh(*la*3*(*x*–0.5))–cos(*la*3*(*x*–0.5)))/(2**la*3^2));

Q = – X(16,1)**la*3^4*(sinh(*la*3**x*)+sin(*la*3**x*))/(2**la*3)–X(12,1)*(…

cosh(*la*3**x*)–cos(*la*3**x*))**la*3^4/(2**la*3^2) + X(13,1)**la*3^4*(sinh(…

*la*3**x*)–sin(*la*3**x*))/(2**la*3^3)+X(14,1)*(cosh(*la*3**x*)+cos(*la*3**x*))/2 – …

f *(cosh(*la*3*(*x*–0.5))+cos(*la*3*(*x*–0.5)))/2;

M = – X(16,1)**la*3^4*(cosh(*la*3**x*)–cos(*la*3**x*))/(2**la*3^2)–X(12,1)**la*3^4* (…

sinh(*la*3**x*)–sin(*la*3**x*))/(2**la*3^3)+X(13,1)*(cosh(*la*3**x*)+cos(*la*3**x*))/2+…

X(14,1)*(sinh(*la*3**x*)+sin(*la*3**x*))/(2**la*3) – *f* *(sinh(*la*3*(*x*–0.5))+ …

sinh(*la*3*(*x*–0.5)))/(2**la*3);

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0.5 1 – 4 4]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0.5 1 – 6 6]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0.5 1 – 70 70]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0.5 1 – 30 30]); *grid on*

<u>стержень 1-3</u>

а) При $0 \le x \le \mathbf{l}/4$ м;

x = 0 : 0.001 : 0.25; *t* = 0.5*2.946045; *la*4 = (4**t*^2)^0.25;

EIv = – (X(16,1)*(cosh(*la*4**x*)+cos(*la*4**x*))/2+X(17,1)*(sinh(*la*4**x*)+ …

sin(*la*4**x*))/(2**la*4) – X(18,1)*(cosh(*la*4**x*)–cos(*la*4**x*))/(2**la*4^2) – …

```
X(19,1)*(sinh(la4*x)–sin(la4*x))/(2*la4^3));
EIfi = – (X(16,1)*la4^4*(sinh(la4*x)–sin(la4*x))/(2*la4^3)+X(17,1)*( …
cosh(la4*x)+cos(la4*x)/2–X(18,1)*(sinh(la4*x)+sin(la4*x))/(2*la4)– …
X(19,1)*(cosh(la4*x)–cos(la4*x))/(2*la4^2));
Q = – X(16,1)*la4^4*(sinh(la4*x)+sin(la4*x))/(2*la4)–X(17,1)*la4^4*( …
cosh(la4*x)–cos(la4*x))/(2*la4^2)+X(18,1)*la4^4*(sin(la4*x) – …
sinh(la4*x))/(2*la4^3)+X(19,1)*(cosh(la4*x)+cos(la4*x))/2;
M = – X(16,1)*la4^4*(cosh(la4*x)–cos(la4*x))/(2*la4^2)–X(17,1)*la4^4*( …
sinh(la4*x)–sin(la4*x))/(2*la4^3)+X(18,1)*(cosh(la4*x)+cos(la4*x))/2+…
X(19,1)*(sinh(la4*x)+sin(la4*x))/(2*la4);
subplot (2,2,1), plot (x, EIv); axis ([0  0.25 – 4  4]); grid on
subplot (2,2,2), plot (x, EIfi); axis ([0  0.25 – 7  7]); grid on
subplot (2,2,3), plot (x, Q); axis ([0  0.25 – 7   7]); grid on
subplot (2,2,4), plot (x, M); axis ([0  0.25 – 18  18]); grid on
```

б) При $\mathbf{l}/4 \le x \le \mathbf{l}$ м;

```
x = 0.25 : 0.001 : 1.0; m = 20.0; t = 0.5*2.946045; la4 = (4*t^2)^0.25;
EIv =–(X(16,1)*(cosh(la4*x)+cos(la4*x))/2+X(17,1)*(sinh(la4*x)+sin(la4*x)) …
/(2*la4) – X(18,1)*(cosh(la4*x)–cos(la4*x))/(2*la4^2)–X(19,1)*(sinh(la4*x) …
–sin(la4*x))/(2*la4^3)+m*(cosh(la4*(x–0.25))–cosh(la4*(x–0.25)))/(2*la4^2));
EIfi = – (X(16,1)*la4^4*(sinh(la4*x)–sin(la4*x))/(2*la4^3)+X(17,1)*( …
cosh(la4*x)+cos(la4*x))/2–X(18,1)*(sinh(la4*x)+sin(la4*x))/(2*la4) – …
X(19,1)*(cosh(la4*x)–cos(la4*x))/(2*la4^2)+m*(sinh(la4*(x–0.25))+ …
sin(la4*(x–0.25)))/(2*la4));
Q = – X(16,1)*la4^4*(sinh(la4*x)+sin(la4*x))/(2*la4)–X(17,1)* la4^4*( …
cosh(la4*x)–cos(la4*x))/(2*la4^2)+X(18,1)*la4^4*(sinh(la4*x)– …
sin(la4*x))/(2*la4^3)+X(19,1)*(cosh(la4*x)+cos(la4*x))/2 – …
m*la4*(sinh(la4*(x–0.25))–sin(la4*(x–0.25)))/2;
```

M = – X(16,1)*la4^4*(cosh(la4*x)–cos(la4*x))/(2*la4^2)–X(17,1)*la4^4*(…

sinh(la4*x)–sin(la4*x))/(2*la4^3)+X(18,1)*(cosh(la4*x)+cos(la4*x))/2+…

X(19,1)*(sinh(la4*x)+sin(la4*x))/(2*la4) –m *(cosh(la4*(x–0.25))+ …

cos(la4*(x–0.25)))/2;

subplot (2,2,1), *plot* (*x*, EIv); *axis* ([0.25 1 – 3 3]); *grid on*

subplot (2,2,2), *plot* (*x*, EI*fi*); *axis* ([0.25 1 – 7 7]); *grid on*

subplot (2,2,3), *plot* (*x*, Q); *axis* ([0.25 1 – 13 13]); *grid on*

subplot (2,2,4), *plot* (*x*, M); *axis* ([0.25 1 – 13 13]); *grid on*

Эпюры EIJ(*x*), EIj (*x*), Q(*x*), M(*x*) представлены на рис. 3.19

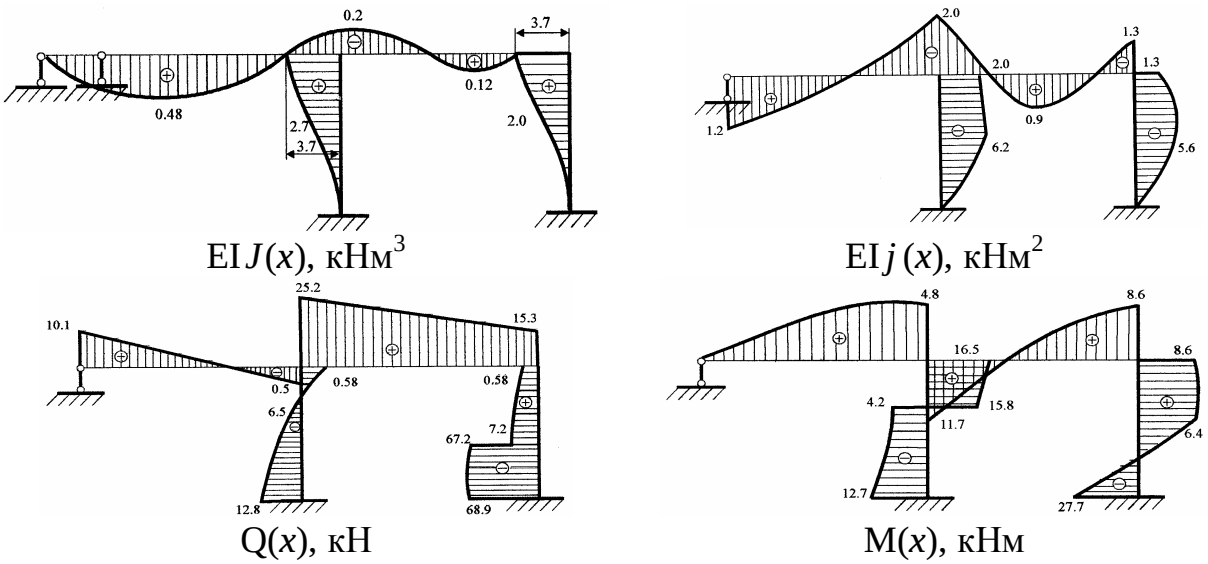


Рис. 3.19

Если сравнить эпюры задачи статики и задачи динамики (рис. 3.15 и рис. 3.19), то видно, что приближение к первой частоте $(q = 0.5w_1)$ заметно влияет на напряженно-деформированное состояние стержней рамы. Особенно значительное различие наблюдается у стержней, к которым добавлены сосредоточенные массы для учета сил инерции линейно подвижных стержней. Если же частота вынужденных колебаний далеко отстоит от первой частоты собственных колебаний и к стержням не добавляются сосредоточенные массы, то статический расчет практически не отличается от динамического расчета (см. расчет неразрезной балки).

3.3.7. *Определение критических сил рамы*

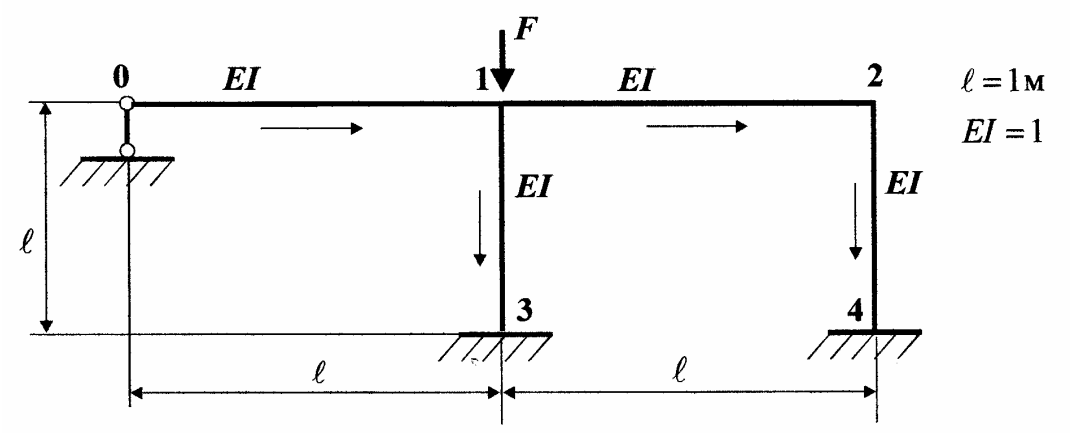


Рис. 3.20

Как и для неразрезной балки, определение критических сил потери устойчивости рамы сводится к поиску корней определителя матрицы устойчивости $|\mathbf{A}_*(F)| = 0$, где фундаментальные функции сжатых стержней определяются набором (3.23). Для рамы по рис. 3.20 при потере устойчивости стержень 1-3 испытывает продольно-поперечный изгиб, а другие стержни – статический изгиб. Принимая ориентированный граф расчета, аналогичный задачам статики и динамики (см. рис. 3.14, 3.16), можно существенно облегчить формирование матрицы устойчивости $\mathbf{A}_*(F)$. Набор компенсирующих элементов остается неизменным, меняются лишь фундаментальные функции стержня 1-3 в сравнении с матрицей задачи статики (3.28). Далее организуется цикл вычисления определителя матрицы устойчивости $|\mathbf{A}_*(F)|$, значения которого и сжимающую силу F выводят в окно команд. При просмотре этих массивов данных легко определить точки, где *изменяет знак* определитель $d = |\mathbf{A}_*(F)|$. В этих точках и вычисляются критические силы потери устойчивости данной рамы. При поиске критических сил рекомендуется начинать вычисление определителя матрицы устойчивости с начального значения F_0=0.01 EI с шагом ΔF=0.01 EI. Число вычислений определителя n_1=200 – 500 позволяет надежно и достаточно точно определить *первую* и последующие критические силы. При решении задачи устойчивости рамы примем, что EI=1, $\mathbf{l} = 1$ м.

Матрица устойчивости рамы примет вид (см. матрицу $\mathbf{A}_*$ задачи статики (3.28))

$\mathbf{A}_*=$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		1		-1/6																
2		1		-1/2													-1			
3				1				-1										-1		
4				1					-1											-1
5										-1									1	
6								-1/2	-1/6								1			
7								-1	-1/2			-1					1			
8								1	1				-1							
9									1						-1					
10										1				1						
11												1	-1/2	-1/6		1				
12												1	-1	-1/2						
13	-1												1	1						
14			-1											1						
15					-1										1					
16																1	A_{12}	$-A_{13}$	$-A_{14}$	
17																	A_{22}	$-A_{12}$	$-A_{13}$	
18						-1											$-A_{32}$	A_{22}	A_{12}	
19							-1												1	
20											-1									1

(3.34)

Обозначения переменных в программе соответствуют задаче устойчивости неразрезной балки. Программа запишется следующим образом.

```
n = 20; n1 = 300; f = 0.01; df = 0.1; X = zeros(n1,1); Y= zeros(n1,1);
for m = 1 : n1   n2 = sqrt(f); a = zeros(n,n);
a(1,2)=1; a(1,4)=-1/6; a(2,2)=1; a(2,4)=-1/2; a(2,17)=-1;
a(3,4)=1; a(3,8)=-1; a(3,18)=-1; a(4,4)=1; a(4,9)=-1; a(4,20)=-1;
a(5,10)=-1; a(5,19)=1; a(6,8)=-1/2; a(6,9)=-1/6; a(6,17)=1; a(7,8)=-1;
a(7,9)=-1/2; a(7,12)=-1; a(7,17)=1; a(8,8)=1; a(8,9)=1; a(8,13)=-1;
a(9,9)=1; a(9,15)=-1; a(10,10)=1; a(10,14)=1; a(11,12)=1;
a(11,13)=-1/2; a(11,14)=-1/6; a(11,16)=1; a(12,12)=1; a(12,13)=-1;
a(12,14)=-1/2; a(13,1)=-1; a(13,13)=1; a(13,14)=1; a(14,3)=-1; a(14;14)=1;
a(15,5)=-1; a(15;15)=1; a(16,16)=1; a(16,17)=sin(n2)/n2;
```

```
a(16,18)=-(1-cos(n2))/(n2^2); a(16,19)=-(n2-sin(n2))/(n2^3);
a(17,17)=cos(n2); a(17,18)=-a(16,17); a(17,19)=a(16,18);
a(18,6)=-1; a(18,17)=n2*sin(n2); a(18,18)=a(17,17); a(18,19)=a(16,17);
a(19,7)=-1; a(19,19)=1; a(20,11)=-1; a(20,20)=1; d=det(a);
X(m,1)=f; Y(m,1)=d; f=f+df; end;
plot(X,Y); grid on
format long e
[X Y]
```

Результаты поиска критических сил потери устойчивости рамы

$F_1 = 15.098365\dfrac{EI}{\mathbf{l}^2}$; $F_2 = 30.980105\dfrac{EI}{\mathbf{l}^2}$; $F_3 = 73.913165\dfrac{EI}{\mathbf{l}^2}$;

$F_4 = 132.291925\dfrac{EI}{\mathbf{l}^2}$; $F_5 = 212.607925\dfrac{EI}{\mathbf{l}^2}$ $кН$.

3.3.8. *Построение форм потери устойчивости рамы*

Для построения форм потери устойчивости рамы необходимо определить граничные параметры рамы при критических силах Fi. Для этого формируется система линейных уравнений краевой задачи по МГЭ, где в правой части в отличие от задачи определения форм собственных колебаний рамы необходимо положить только EI$J^{1\text{-}3}(\mathbf{l})$=-1, т.е. b(16,1)=-1. Матричное уравнение данной задачи можно заимствовать из задачи динамики, где необходимо заменить фундаментальные функции поперечных колебаний на фундаментальные функции статического и продольно-поперечного изгибов при граничных значениях координаты x всех стержней (см. (3.35)). Уравнения форм потери устойчивости стержней рамы примут вид (3.12) и (3.26). Программа определения граничных параметров рамы запишется следующим образом:

```
a=zeros(20,20); b= zeros(20,1); X=zeros(20,1); f=15.098365;
n2=sqrt(f); a(1,2)=1; a(1,4)=-1/6; a(2,2)=1; a(2,4)=-1/2; a(2,17)=-1;
a(3,4)=1; a(3,8)=-1; a(3,18)=-1; a(4,4)=1; a(4,9)=-1; a(4,20)=-1;
a(5,10)=-1; a(5,19)=1; a(6,8)=-1/2; a(6,9)=-1/6; a(6,17)=1; a(7,8)=-1;
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20			
1		1		-1/6																	$M^{2-4}(\ell)$		
2		1		-1/2													-1				$EI\varphi^{0-1}(o)$		
3				1				-1										-1			$Q^{2-4}(\ell)$		
4				1					-1											-1	$Q^{0-1}(o)$		
5										-1									1		$N^{2-4}(\ell)$		
6								-1/2	-1/6								1				$M^{1-3}(\ell)$		
7								-1	-1/2			-1					1				$Q^{1-3}(\ell)$		
8								1	1				-1								$M^{1-2}(o)$		
9									1						-1						$Q^{1-2}(o)$		
10										1				1							$N^{1-2}(o)$	=	
11												1	-1/2	-1/6		1					$N^{1-3}(\ell)$		
12												1	-1	-1/2							$EI\varphi^{2-4}(o)$		
13	-1												1	1							$M^{2-4}(o)$		
14			-1											1							$Q^{2-4}(o)$		
15					-1										1						$N^{2-4}(o)$		
16																1	A_{12}	$-A_{13}$	$-A_{14}$		$EI\vartheta^{1-3}(o)$		$EI\vartheta^{1-3}(\ell) = -1$
17																	A_{22}	$-A_{12}$	$-A_{13}$		$EI\varphi^{1-3}(o)$		
18						-1											$-A_{32}$	A_{22}	A_{12}		$M^{1-3}(o)$		
19							-1												1		$Q^{1-3}(o)$		
20											-1									1	$N^{1-3}(o)$		

(3.35)

a(7,9)=-1/2; a(7,12)=-1; a(7,17)=1; a(8,8)=1; a(8,9)=1; a(8,13)=-1; a(9,9)=1; a(9,15)=-1; a(10,10)=1; a(10,14)=1; a(11,12)=1; a(11,13)=-1/2; a(11,14)=-1/6; a(11,16)=1; a(12,12)=1; a(12,13)=-1; a(12,14)=-1/2; a(13,1)=-1; a(13,13)=1; a(13,14)=1;

a(14,3)=-1; a(14,14)=1; a(15,5)=-1; a(15,15)=1; a(16,16)=1;

a(16,17)=sin(n2)/n2; a(16,18)=-(1-cos(n2))/(n2^2);

a(16,19)=-(n2-sin(n2))/(n2^3); a(17,17)=cos(n2); a(17,18)=-a(16,17);

a(17,19)=a(16,18); a(18,6)=-1; a(18,17)=n2*sin(n2); a(18,18)=(17,17);

a(18,19)=a(16,17); a(19,7)=-1; a(19,19)=1; a(20,11)=-1; a(20;20)=1;

b(16,1)=-1; X=a\b; X=X/X(16,1)

Значения относительных граничных параметров сведены в таблицу 3.9.

Таблица 3.9

№ п/п	Граничные параметры рамы	Относительные значения граничных параметров при критических силах				
		F_1=15,098365	F_2=30,980105	F_3=73,913165	F_4=132,291925	F_5=212,607925
1	$M^{2-4}(\mathbf{l}) = X(1,1) =$	-4.6771	24.7402	-6.2350	7.6095	-8.7618
2	$EIj^{0-1}(o) = X(2,1) =$	0.1771	-29.2402	1.7350	-12.1095	4.2618
3	$Q^{1-2}(\mathbf{l}) = X(3,1) =$	-8.0312	80.2205	-12.7049	28.8284	-20.2855
4	$Q^{0-1}(o) = X(4,1) =$	1.0625	-175.4409	10.4098	-72.6569	25.5709
5	$N^{2-4}(\mathbf{l}) = X(5,1) =$	6.0937	-258.6614	20.1147	-104.4853	42.8564
6	$M^{1-3}(\mathbf{l}) = X(6,1) =$	-3.2651	-489.8187	-37.1538	-317.0371	-135.4205
7	$Q^{1-3}(\mathbf{l}) = X(7,1) =$	8.0312	-80.2205	12.7049	-28.8284	20.2855
8	$M^{1-2}(o) = X(8,1) =$	-2.7396	203,1811	-13.6448	83.2663	-31.3327
9	$Q^{1-2}(o) = X(9,1) =$	6.0937	-258.6614	20.1147	-104.4853	42.8564
10	$N^{1-2}(o) = X(10,1) =$	8,0312	-80.2205	12.7049	-28.8284	20.2855
11	$N^{1-3}(\mathbf{l}) = X(11,1) =$	-5.0312	83.2205	-9.7049	31.8284	-17.2855
12	$EIj^{2-4}(o) = X(12,1) =$	-0.6615	-15.3701	0.1175	-6.8047	1.3809
13	$M^{2-4}(o) = X(13,1) =$	3.3542	-55.4803	6.4699	-21.2190	11.5236
14	$Q^{2-4}(o) = X(14,1) =$	-8.0312	80.2205	-12.7049	28.8284	-20.2855
15	$N^{2-4}(o) = X(15,1) =$	6.0937	-258.6614	20.1147	-104.4853	42.8564
16	$EIJ^{1-3}(o) = X(16,1) =$	1.0000	1.0000	1.0000	1.0000	1.0000
17	$EIj^{1-3}(o) = X(17,1) =$	-0.3542	58.4803	-3.4699	24.2190	-8.5236
18	$M^{1-3}(o) = X(18,1) =$	3.8021	-378.6220	24.0546	-155.9232	56.9037
19	$Q^{1-3}(o) = X(19,1) =$	8.0312	-80.2205	12.7049	-28.8284	20.2855
20	$N^{1-3}(o) = X(20,1) =$	-5.0312	83.2205	-9.7049	31.8284	-17.2855

Формы потери устойчивости стержней рамы строятся по соотношениям метода начальных параметров. Для стержня 1-3 это выражение (3.26) продольно-поперечного изгиба, для остальных стержней – выражение (3.12) статического изгиба. Протокол построения форм потери устойчивости примет вид

```
x=0 : 0.001 : 1.0; f=15.098365;  n2=sqrt(f);
EIV1=-(X(2,1)*x-X(4,1)*x.^3/6);
EIV2=-(X(17,1)*x-X(8,1)*x.^2/2-X(9,1)*x.^3/6);
EIV3=-(X(16,1)+X(12,1)*x-X(13,1)*x.^2/2-X(14,1)*x.*3/6);
EIV4=-(X(16,1)+X(17,1)*sin(n2*x)/n2-X(18,1)*(1-cos(n2*x))/n2^2- …
X(19,1)*(n2*x-sin(n2*x))/n2^3);
subplot (2,2,1), plot (x, EIV1); axis ([0  1 – 0.1  0.1]); grid on
subplot (2,2,2), plot (x, EIV2); axis ([0  1 – 0.1  0.1]); grid on
subplot (2,2,3), plot (x, EIV3); axis ([0  1 – 1  1]); grid on
subplot (2,2,4), plot (x, EIV4); axis ([0  1 – 1  1]); grid on
```

Перед выполнением данного протокола в окно команд необходимо поместить вектор относительных граничных параметров X при соответствующей критической силе Fi и не забывать корректировать масштабы форм потери устойчивости. Сами формы потери устойчивости представлены на рис. 3.21.

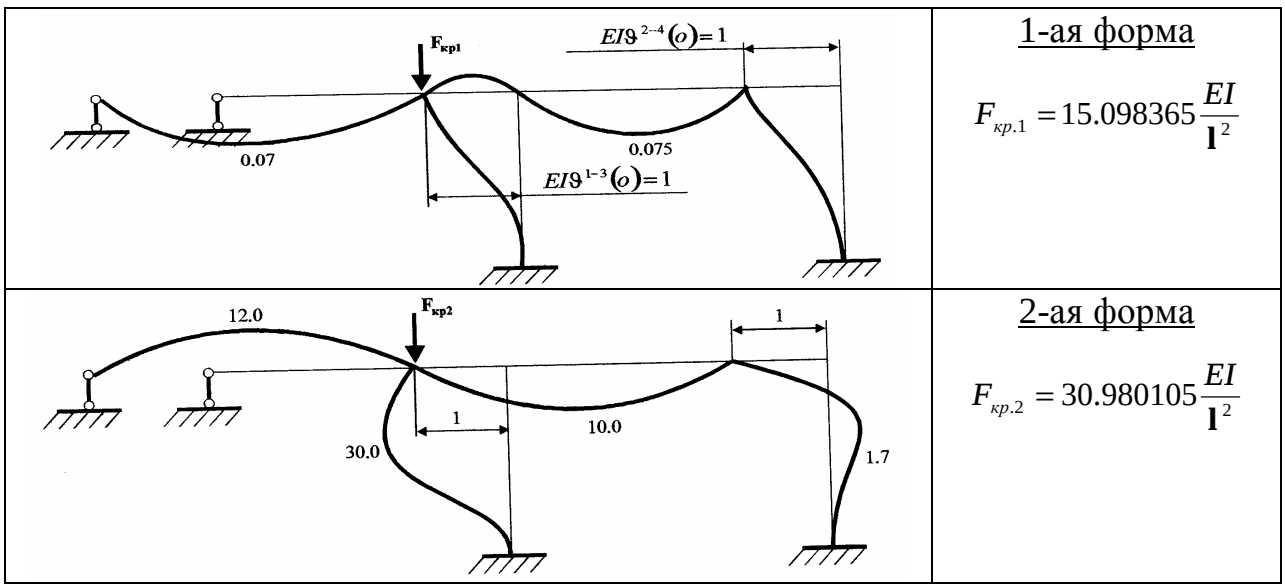
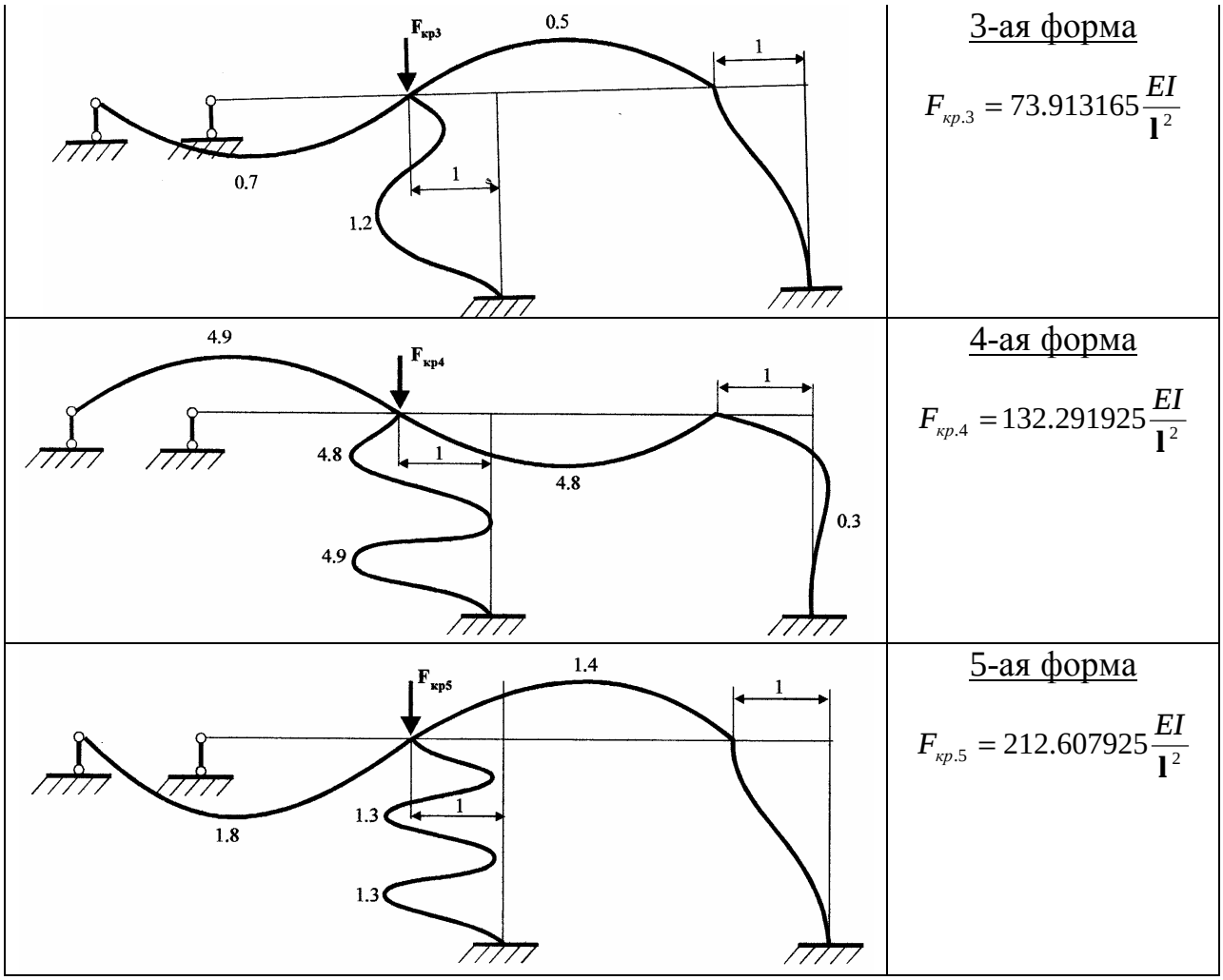


Рис. 3.21



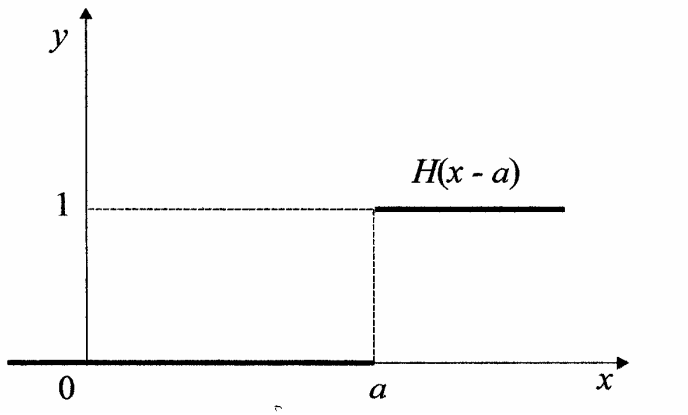
Окончание рис. 3.21

В заключение отметим еще одну замечательную особенность матрицы $\mathbf{A}_*$ задач динамики и устойчивости. Эта матрица описывает связь между собой всех начальных и конечных параметров элементов упругой системы. Поэтому при поиске и построении форм собственных колебаний и потери устойчивости можно назначать единичное значение произвольному элементу матрицы нагрузки **B**. Последующая нормировка компонентов вектора $\mathbf{X}_*$ относительно кинематических или статических параметров упругой системы приводит благодаря структуре матрицы $\mathbf{A}_*$ к одинаковым соотношениям между начальными и конечными граничными параметрами. Другими словами, значения относительных параметров таблиц 3.9, 3.7 и т. д. остаются неизменными независимо от положения единичного элемента матрицы **B**. Небольшие ограничения касаются тех элементов матрицы **B**, которые приводят

к делению на ноль или $\frac{0}{0}$. Например, для уравнения (3.35) это b(19.1)=-1. b(20,1)=-1 и некоторые другие.

3.3.9. *Построение эпюр напряженно-деформированного состояния с разрывами 1-го рода*

Примеры расчета неразрезной балки и рамы, приведенные в п. 3.1, 3.2, показывают, что необходимо учитывать разрывы 1-го рода при построении эпюр напряженно-деформированного состояния. Данная задача требует применения единичной функции Хевисайда H(x-a) со сдвигом в точку a (рис. 3.22). В этом случае сокращаются тексты программ, а соответствующие эпюры можно получить сразу для всей длины стержня.



$$H(x-a)=\begin{cases}0, & x \le a\\ 1, & x > a\end{cases};$$

Рис. 3.22

В MATLAB нет встроенной единичной функции Хевисайда, поэтому ее необходимо программировать с применением операторов условного перехода <u>if … end</u> и логического оператора <u>and</u>. В качестве примера представим программу, позволяющую построить эпюры прогиба EIJ(x), угла поворота EIj (x), поперечных сил Q(x) и изгибающих моментов M(x) неразрезной балки по рис. 3.1, где выигрыш применения единичных функций наиболее очевиден. Для построения эпюр достаточно взять 500 – 600 точек. С учетом компонентов вектора X формулы для параметров напряженно-деформированного состояния балки примут вид:

<u>прогиб EIJ(x), $0 \le x \le 14.0$ м.</u>

$$
\begin{aligned}
EIJ(x) = -\Bigg\{ & X(2,1)\cdot x - \frac{X(4,1)\cdot x^3}{6} + \frac{20\cdot(x-2)^2\cdot H(x-2)}{2} + \\
& + [X(1,1) - X(8,1)]\frac{(x-4)^3\cdot H(x-4)}{6} + \frac{10\cdot(x-4)^4\cdot H(x-4)}{24} - \\
& - \frac{10\cdot(x-10)^4\cdot H(x-10)}{24} + [X(3,1) - X(12,1)]\frac{(x-10)^3\cdot H(x-10)}{6} - \\
& - \frac{40\cdot(x-12)^3\cdot H(x-12)}{6} + [X(5,1) - X(16,1)]\frac{(x-13)^3\cdot H(x-13)}{6}\Bigg\}.
\end{aligned}
\tag{3.36}
$$

Как видно из этого выражения необходимо применить 5 разных единичных функций. В формуле (3.36) разности в квадратных скобках представляют собой реакции опор 1, 2, 3.

$$R_1 = Q^{0-1}(\mathbf{l}) - Q^{1-2}(o) = X(1,1) - X(8,1);$$
$$R_2 = Q^{1-2}(\mathbf{l}) - Q^{2-3}(o) = X(3,1) - X(12,1);$$
$$R_3 = Q^{2-3}(\mathbf{l}) - Q^{3-4}(o) = X(5,1) - X(16,1).$$

<u>Угол поворота EI $j(x)$, $0 \le x \le 14.0$ м.</u>

$$
\begin{aligned}
EIj(x) = -\Bigg\{ & X(2,1)\cdot x - \frac{X(4,1)\cdot x^2}{6} + 20\cdot(x-2)\cdot H(x-2) + \\
& + [X(1,1) - X(8,1)]\frac{(x-4)^2\cdot H(x-4)}{2} + \frac{10\cdot(x-4)^3\cdot H(x-4)}{6} - \\
& - \frac{10\cdot(x-10)^3\cdot H(x-10)}{6} + [X(3,1) - X(12,1)]\frac{(x-10)^2\cdot H(x-10)}{2} - \\
& - \frac{40\cdot(x-12)^2\cdot H(x-12)}{2} + [X(5,1) - X(16,1)]\frac{(x-13)^2\cdot H(x-13)}{2}\Bigg\}.
\end{aligned}
\tag{3.37}
$$

<u>Поперечная сила Q(x), $0 \le x \le 14.0$ м.</u>

$$
\begin{aligned}
Q(x) = {} & X(4,1) - [X(1,1) - X(8,1)]\cdot H(x-4) - 10\cdot(x-4)\cdot H(x-4) + \\
& + 10\cdot(x-10)\cdot H(x-10) - [X(3,1) - X(12,1)]\cdot H(x-10) + 40\cdot H(x-12) - \\
& - [X(5,1) - X(16,1)]\cdot H(x-13).
\end{aligned}
\tag{3.38}
$$

<u>Изгибающий момент M(x), $0 \le x \le 14.0$ м.</u>

$$
\begin{aligned}
M(x) = {} & X(4,1)\cdot x - 20\cdot H(x-2) - [X(1,1) - X(8,1)]\cdot(x-4)\cdot H(x-4) - \\
& - \frac{10(x-4)^2\cdot H(x-4)}{2} + \frac{10(x-10)^2\cdot H(x-10)}{2} - [X(3,1) - X(12,1)]\cdot(x-10)\cdot H(x-10) + \\
& + 40(x-12)\cdot H(x-12) - [X(5,1) - X(16,1)]\cdot(x-13)\cdot H(x-13).
\end{aligned}
\tag{3.39}
$$

Программа, реализующая формулы (3.36) – (3.39), организует цикл, в котором применен блок формирования единичных функций. Далее следуют операторы построения эпюр и вывод таблицы значений параметров напряженно-деформированного состояния всей балки. Здесь необходимо подставлять численные значения элементов вектора граничных параметров **X**.

Текст программы

```
n=700; EIv=zeros(n,1); EIfi=zeros(n,1); Q=zeros(n,1);
M=zeros(n,1); x=0.0; dx=14.0/n; X=zeros(n,1);
for  m = 1 : n
if and(x > = 0.0, x < = 2.0)  h1=0; h2=0; h3=0; h4=0; h5=0; end;
if and(x > 2.0, x < = 4.0)  h1=1; h2=0; h3=0; h4=0; h5=0; end;
if and(x > 4.0, x < = 10.0)  h1=1; h2=1; h3=0; h4=0; h5=0; end;
if and(x > 10.0, x < = 12.0)  h1=1; h2=1; h3=1; h4=0; h5=0; end;
if and(x > 12.0, x < = 13.0)  h1=1; h2=1; h3=1; h4=1; h5=0; end;
if and(x > 13.0)  h1=1; h2=1; h3=1; h4=1; h5=1; end;
EIv(m,1)= – (–12.79835*x – (–1.04938)*x.^3/6 + 10*(x – 2).^2*h1 + …
(–34.08093)*(x – 4).^3*h2/6 + 10*(x – 4).^4*h2/24 – …
10*(x – 10).^4*h3/24 + 4.36214*(x – 10).^3*h3/6 – …
40*(x – 12).^3*h4/6 + (–51.33059)*(x – 13).^3*h5/6);
EIfi(m,1)= – (–12.79835 – (–1.04938)*x.^2/2 + 20*(x – 2)*h1 + …
(–34.08093)*(x – 4).^2*h2/2 + 10*(x – 4).^3*h2/6 – …
10*(x – 10).^3*h3/6 + 4.36214*(x – 10).^2*h3/2 – …
20*(x – 12).^2*h4 + (–51.33059)*(x – 13).^2*h5/2);
Q(m,1)= (–1.04938) – (–34.08093)*h2 – 10*(x – 4)*h2 + …
10*(x – 10)*h3 – 4.36214*h3 + 40*h4 – …
(– 51.33059)*h5;
M(m,1)= (–1.04938)*x – 20*h1-(34.08093)*(x – 4)*h2 – …
– 5*(x – 4).^2*h2 + 5*(x – 10).^2*h3 – (4.36214)*(x – 10)*h3 + …
40*(x – 12)*h4 – (– 51.33059)*(x – 13)*h5; X(m,1)=x;
x = x + dx; end;
```

```
plot(X, EIv); grid on
% plot(X, Q); grid on
% plot(X, EIfi); grid on
% plot(X, M); grid on
[X  EIv  EIfi   Q   M]
```

На экране компьютера появится окно с первой эпюрой, совпадающей с рис. 3.1, затем, последовательно снимая символ комментария %, строятся другие эпюры, а в окно команд выводится таблица численных значений кинематических и статических параметров балки. Добавим, что совершенно аналогично можно строить эпюры с разрывами 1-го рода и для любых стержней рам. Если вместо оператора plot(X, EIv) использовать оператор stem(X, EIv,′filled′), то получится закрашенная эпюра, мало отличающаяся от обычного представления такого рода графиков. Можно сократить число используемых единичных функций до одной. Однако в этом случае единичную функцию необходимо создать в отдельном М-файле, например, такого содержания

```
function f = H(t)
if  t < = 0.0  f = 0.0;  else f = 1.0; end;
```

Этот файл должен быть записан в рабочей папке MATLAB (обычно это папка work) под именем H. Тогда в головной программе построения эпюр можно использовать одну единичную функцию, но от разных аргументов, т.е.

H(x – 2), H(x – 4), H(x – 10) и т.д. Например, выражение для прогиба (3.36) примет вид

```
EIv(m,1) = – (– 12.79835*x – (– 1.04938)*x.^3/6 + 10*(x – 2).^2*H(x – 2) + …
(– 34.08093)*(x – 4).^3*H(x – 4)/6 + 10*(x – 4).^4*H(x – 4)/24 – …
10*(x – 10).^4*H(x – 10)/24 + 4.36214*(x – 10).^3*H(x – 10)/6 – …
40*(x – 12).^3*H(x – 12)/6 – 51.33059*(x – 13).^3*H(x – 13)/6).
```

При выполнении программы на экране компьютера появляется эпюра по рис. 3.1. Аналогично программируются другие параметры.

3.3.10. *Варианты заданий*

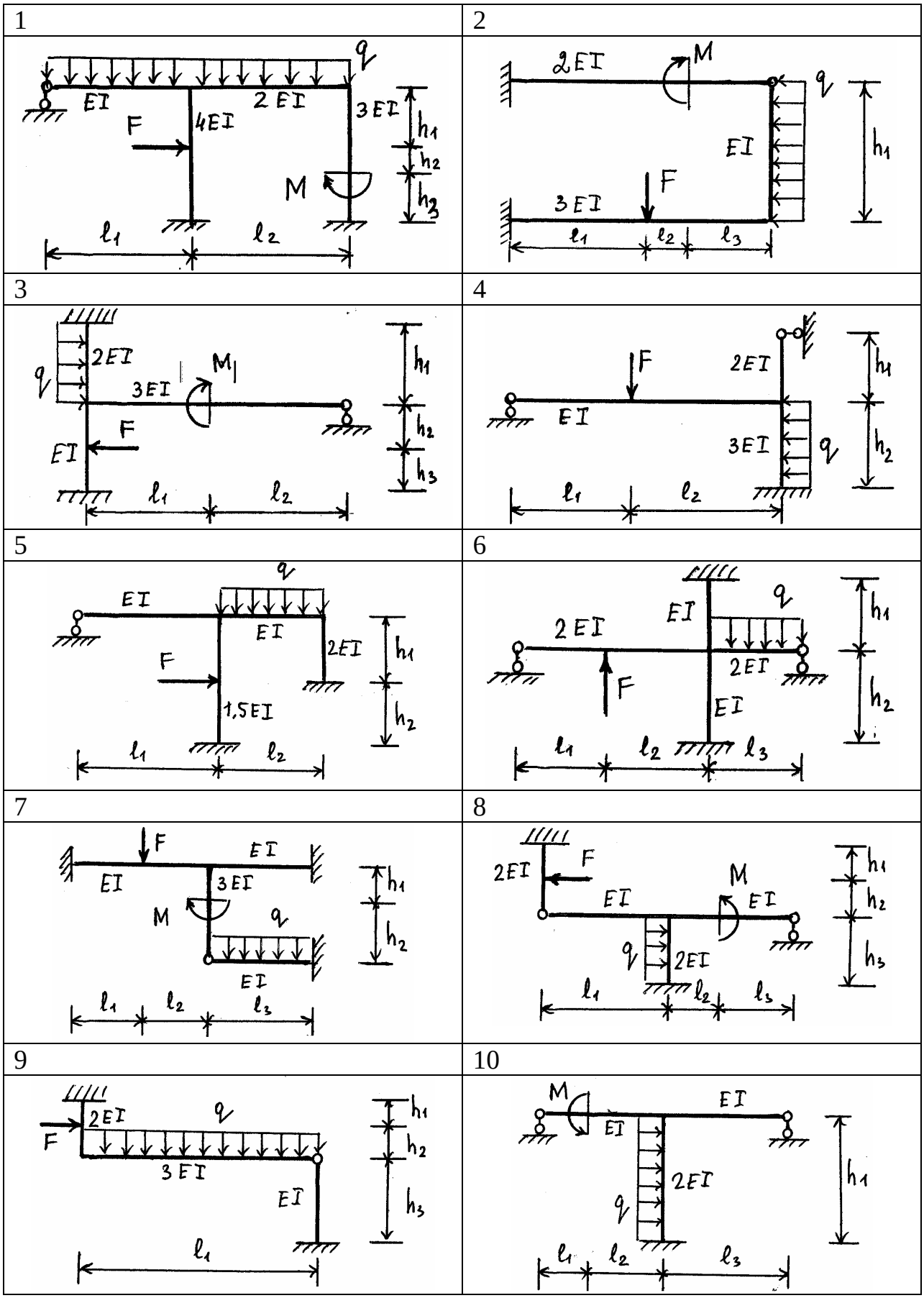
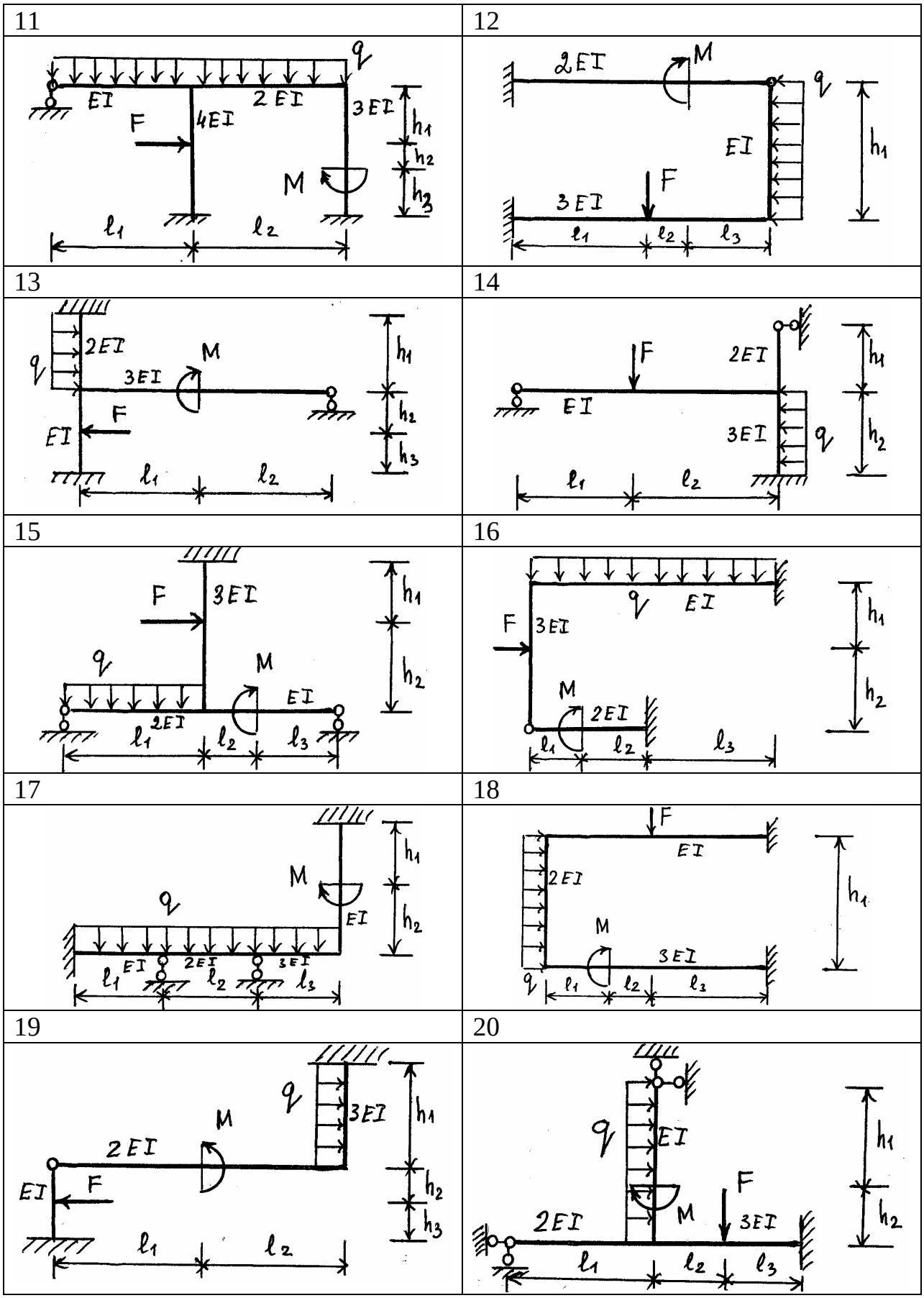
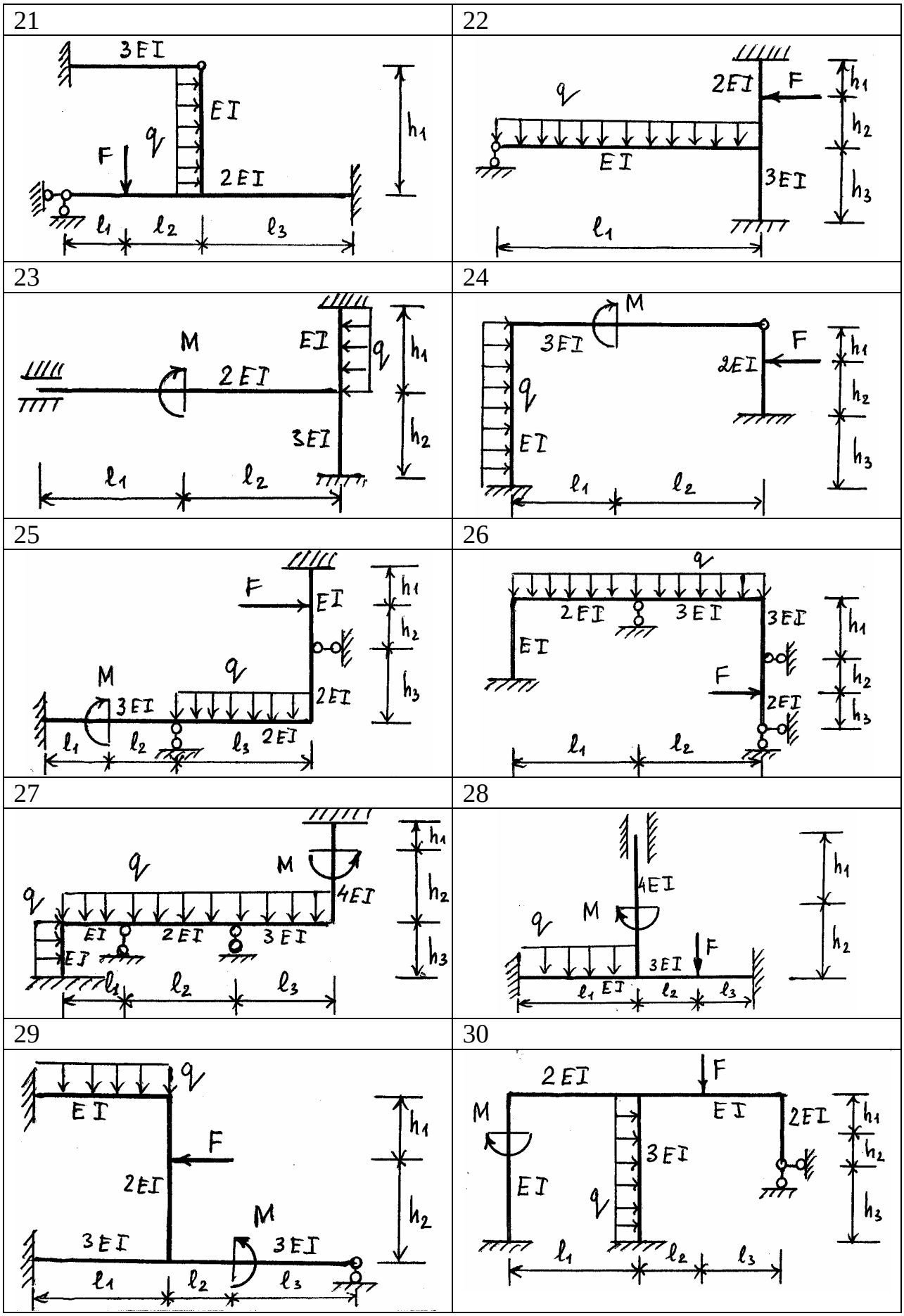


Рис. 3.23



Продолжение рис. 3.23



Окончание рис. 3.23

Таблица 3.10

Исходные данные к расчету плоских рам

№ варианта задания	l_1	l_2	l_3	h_1	h_2	h_3	F	M	q
	м						кН	кНм	кН/м
1	1	3	4	2	3	4	10	16	10
2	2	3	5	3	1	2	40	12	8
3	3	4	2	2	1	2	5	20	6
4	5	4	1	2	1	2	25	25	4
5	5	1	3	3	4	4	10	15	2
6	2	6	4	2	2	1	10	15	12
7	1	3	3	2	1	2	30	40	20
8	1	3	1	1	2	1.5	60	15	15
9	3	4	6	1.5	2	3	30	25	12
10	4	2	3	4	2	1	28	40	8
11	1	1	2	1	2	2	12	15	6
12	4	5	4	1	2	3	5	40	10
13	2	3	4	2	1	1	3	15	18
14	2	1	1	2	4	3	40	20	10
15	5	2	4	4	3	2	15	15	5
16	1	3	2	4	5	1	20	6	12
17	5	4	2	3	2	2	40	10	10
18	1	2	1	4	3	1	25	20	8
19	3	1	4	2.5	4	5	25	25	14
20	5	4	1	2	4	5	15	16	12
21	1	2	3	5	4	3	10	15	16
22	4	2	4	3	2	1	30	40	22
23	5	4	2	4	4	2	60	18	17
24	1	3	5	2	4	5	30	25	14
25	4	2	2	1	2	4	28	40	10
26	5	3	2	2	2	1	20	18	6
27	3	2	1	2	3	4	25	40	10
28	5	4	4	2	3	4	20	15	18
29	2	3	1	3	4	5	40	20	10
30	1	2	5	4	5	1	15	15	7

Глава IV. Применение пакетов расширений MATLAB в инженерно-научных задачах

4.1. Пакет расширений Symbolic Math

Система MATLAB является самой крупной системой компьютерной математики, ориентированной на матричные и численные вычисления. Однако MATLAB имеет также и средства аналитических вычислений. Пакет Symbolic Math Toolbox добавил системе MATLAB качественно новые возможности, связанные с выполнением символьных вычислений и преобразований, которые были доступны только в системе принципиально иного класса, относящихся к компьютерной алгебре. Теперь MATLAB, с учетом новых средств, становится в полной мере *универсальной* системой. Последняя реализация системы символьной математики Maple 6 в своем ядре и в расширениях имеет около 3000 функций. Система MATLAB с пакетом Symbolic, включающим в себя чуть больше сотни символьных команд и функций, намного уступает Maple по количеству таких команд и функций. Однако в данный пакет включены лишь наиболее важные и широко распространенные функции. Кроме того, есть специальная команда, которая дает доступ к ядру Maple, что заметно расширяет круг используемых функций.

Помимо типовых аналитических вычислений (таких как символьное дифференцирование и интегрирование, упрощение математических выражений, подстановка и т. д.) пакет Symbolic позволяет реализовать арифметические операции с произвольной точностью.

С помощью команды

» help symbolic

можно получить перечень входящих в пакет команд и функций. Для получения справки по любой команде или функции можно использовать команду

» help sym / name.m,

где name — это имя соответствующей команды или функции, а name.m — имя m-файла, задающего данную команду или функцию.

С демонстрационными примерами пакета Symbolic можно ознакомиться с помощью директории Symbolic Toolbox.

Создание символьных переменных

Поскольку переменные системы MATLAB по умолчанию не определены и традиционно задаются как векторные, матричные, числовые и т. д., то есть не имеющие отношения к символьной математике, для реализации символьных вычислений нужно прежде всего позаботиться о создании специальных *символьных переменных*. В простейшем случае их можно определить как строковые переменные, заключив имена в апострофы. Например,

```
» sin(x)^2 + cos(x)^2
??? Undefined function or variable 'x'.
» sin('x')^2 + cos('x')^2
ans =
        1
```

В первом случае система MATLAB «возмутилась» нашей небрежностью и сообщила, что функция или переменная x не определена и ни о каких вычислениях синуса и косинуса речи быть не может. Вместе с тем она подсказала, как надо поступить — заключить имя переменной в апострофы, ибо таким образом система получает информацию о необходимости включить символьный режим вычислений. Поэтому во второй раз получен вполне осмысленный результат — сумма квадратов синуса и косинуса переменной 'x' выдана равной 1.

Функция создания символьных переменных *sym*

Для работы с командами ядра Meple в MATLAB определён новый тип переменной ***sym*** – символьный объект. Фактически это строковые переменные. Для проведения аналитических (символьных) операций нужно, чтобы соответствующие переменные были предварительно объявлены.

• S = sym(A) — возвращает символьный объект S класса *'sym'* для входного параметра A. Если A — строка, то будет получена символьная строка или символьная переменная, а если A — это число (скаляр) или матрица, то будут получены их символьные представления.

• x = sym ('x') — возвращает символьную переменную с именем 'x' и записывает результат в x.

Функция создания группы символьных объектов *syms*

Для создания группы символьных объектов служит функция syms:

• syms argl arg2 ... — создает группу символьных объектов.

Функция создания списка символьных переменных *findsym*

В математических выражениях могут использоваться как обычные, так и символьные переменные. Функция *findsym* позволяет выделить символьные переменные в составе выражения S:

• findsym(S) — возвращает в алфавитном порядке список всех символьных переменных выражения S. При отсутствии таковых возвращается пустая строка.

Примеры:

```
» a =2; b = 4 ;
» findsym (a*x^2 + b*y + z)
ans =
x, y, z
» findsym(a + b + x + y + z, 2)
ans =
x, y, z
```

Функция вывода символьных выражений *pretty*

MATLAB в отличие от современных систем MathCAD, Maple или Mathematica, пока не способна выводить выражения и результаты их преобразований в естественной математической форме с использованием общепринятых спецзнаков для отображения интегралов, сумм, произведений и т. д. Тем не менее некоторые ограниченные текстовым форматом возможности близкого к математическому виду вывода обеспечивает функция *pretty*:

- pretty(S) — дает вывод выражения S в формате, приближенном к математическому;

```
« x = sym('x');
« pretty ( x^2 )
                 x 2
```

Арифметика произвольной точности

Функция ***digits*** служит для установки числа цифр в числах арифметики произвольной точности. Она используется в следующем виде:

digits — возвращает число значащих цифр в числах арифметики произвольной точности (по умолчанию 32);

Примеры:

```
» digits
Digits = 32
» vpa pi
ans =
    3.14159265358979323846264338327 95
» digits(6)
» pi
ans =
     3.1416
```

Для проведения вычислений в арифметике произвольной точности служит функция ***vpa***:

- R = vpa(S) — возвращает результат вычислений каждого элемента символьного массива S, используя арифметику произвольной точности с текущим числом цифр D, установленным функцией *digits*. Результат R имеет тип *sym*.
- vpa(S,D) — возвращает результат вычислений каждого элемента массива S, используя арифметику произвольной точности с количеством знаков чисел D.

Примеры:

```
vpa(exp(1),50)
ans =
2.7182818284590450907955982984276488423347473144531
```

Символьные операции с выражениями

1. Функция упрощения выражений – ***simplify***

Функция simplify(S) поэлементно упрощает символьные выражения массива S. Если упрощение невозможно, то возвращается исходное выражение.

Примеры:

```
» syms a b x;
» V = [sin(x)^2 + cos(x)^2   log(a*b)]
» simplify(V)
ans =
    [   1,   log(a*b)]
» simplify((a^2 - 2*a*b + b^2) / (a - b))
ans =
    a – b
```

Дополнительные возможности упрощения обеспечивает функция ***simple***.

2. Функция расширения (раскрытия скобок) выражений - ***expand***

Функция expand(S) расширяет выражения, входящие в массив S. Рациональные выражения она раскладывает на простые дроби, полиномы — на полиномиальные разложения и т. д. Функция работает со многими алгебраическими и тригонометрическими функциями.

Примеры:

```
» syms a b x;
» S=[(x + 2)*(x + 3)*(x + 4)     sin(2*x)];
» expand(S)
  ans =
[ x^3 + 9*x^2 + 26*x + 24 ,  2*sin(x)*cos(x)]
» expand(sin(a + b))
  ans =
sin(a)*cos(b) + cos(a)*sin(b)
» expand((a + b)^:3)
  ans =
      a^3 + 3*a^2*Ь + 3*a*Ь^2 + Ь^3
```

3. Разложение выражения на простые множители – ***factor***

Функция factor(S) поэлементно разлагает выражения вектора S на простые множители, а целые числа — на произведение простых чисел. Следующие примеры иллюстрируют применение функции:

```
» x=sym('x');
» factor(x^7-l)
ans =
    ( x – 1 )*( x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 )
» factor(x^2 – x - 1)
```

```
ans =
    x^2 - x - 1
» factor(sym('123456789'))
ans =
   (3 )^2*(3803)*(3607)
```

4. Комплектование по степеням – ***collect***

Функция co11ect(S,v) обеспечивает комплектование выражений в составе вектора или матрицы S по степеням переменной v.

5. Упрощение выражений – ***simple***

Функция simple(S) выполняет различные упрощения для элементов массива S и выводит как промежуточные результаты, так и самый короткий конечный результат. В другой форме — [R, HOW] = simple(S) – промежуточные результаты не выводятся.

6. Приведение к рациональной форме – ***numden***

Функция [N,D] = numden(A) преобразует каждый элемент массива A в рациональную форму в виде отношения двух неприводимых полиномов с целочисленными коэффициентами. При этом N и D — числители и знаменатели каждого преобразованного элемента массива.

Примеры:

```
» [n,d] = numden(sym(8/10))
n =
4
d =
5
» syms x  y
» [ n, d ] = numden (x*y + y / x)
n =
y*(x^2 + 1)
d =
x
```

7. Обеспечение подстановок – ***subs***

Одной из самых эффектных и часто используемых операций символьной математики является операция подстановки. Она реализуется функцией subs, имеющей ряд форм записи:

- subs(S) — заменяет в символьном выражении S все переменные их символьными значениями, которые берутся из вычисляемой функции или рабочей области системы MATLAB.
- subs(S,NEW) — заменяет все свободные символьные переменные в S из списка NEW.
- subs (S, OLD, NEW) — заменяет OLD на NEW в символьном выражении S. При одинаковых размеров массивов OLD и NEW замена идет поэлементно. Если S OLD — скаляры, а NEW — числовой массив или массив ячеек, то скаляры расширяются до массива результатов.

Пример:

```
» syms a  b  x  y;
» subs(x - y, y, l)
ans =
     x - l
» subs(sin(x) + cos(y), [x,y], [a,b])
 ans =
     sin(a)+cos(b)
```

8. Обращение функции – ***finverse***

Часто возникает необходимость в задании функции, обратной по отношению к заданной функции **f** . Для этого в Symbolic имеется функция обращения inverse, которая задается в двух формах:

- g = finverse(f) — возвращает функцию, обратную к f. Считается, что f — функция одной переменной, например 'x'. Тогда g(f(x)) = x.
- g = finverse(f,v) — возвращает функцию, обратную к f, относительно заданной переменной v, так что g(f(v)) = v. Эта форма используется, если f — функция нескольких переменных.

Примеры:

```
» syms x
» finverse(sinh(x))
ans =
    asnnh(x)
» finverse(expCx))
```

```
ans =
    log(x)
```

9. Суперпозиция функций – ***compose***

- compose(f, g) — возвращает f(g(y)), где f = f(x) и g = g(y). Независимые переменные x и y находятся с помощью функции findsym.

Символьные операции математического анализа

1. Функция вычисления производных – ***diff***

Для вычисления в символьном виде производных от выражения S служит функция diff, записываемая в формате `diff(S,'v')` или diff(S, sym('v')) Она возвращает символьное значение первой производной от символьного выражения или массива символьных выражений S по переменной v. Эта функция возвращает $S'(v) = \frac{dS}{dv}$

- diff(S, n) — возвращает *n-ю (n* — целое число) производную символьного выражения или массива символьных выражений S по переменной v.
- diff(S, V, n) и diff(S, n, V) – возвращает *n-ю* производную S по переменной v.

Примеры:

```
» x =sym( 'x' ); y=sym( 'y' );
» diff(x^y)
ans =
x^y*y / x
» slmplify(ans)
ans =
x^(y-1)*y
» diff(s1n(y*x), x, 3)
ans =
    - cos(y*x)*y^3
» diff([x^3  sin(x)  exp(x)], x)
ans =
    [ 3*x^2,   cos(x),  exp(x)]
```

2. Функция интегрирования – ***int***

Функция ***int*** вычисляет неопределенные и определенные интегралы

- int(S) — возвращает символьное значение неопределенного интеграла от сим-

вольного выражения или массива символьных выражений S по переменной, которая автоматически определяется функцией findsym. Если S — скаляр или матрица, то вычисляется интеграл по переменной 'x'.

- int(S, v) — возвращает неопределенный интеграл от S по переменной v.
- int(S, a, b) — возвращает определенный интеграл от S с пределами интегрирования от a до b, причем пределы интегрирования могут быть как символьными, так и числовыми.
- int(S, v, a, b) — возвращает определенный интеграл от S по переменной v с пределами от a до b.

Примеры:

```
» int(sin(x)^3, x)
ans =
    - l/3*sin(x)^2*cos(x)-2/3*cos(x)
» int(log(2*x), x)
ans =
    log(2*x)*x - x
» int((x^2-2)/(x*3-l), x, l, 2)
ans =
    -inf
» int((x^2-2)/(x*3-l), x, 2, 5)
ans =
    - 2/3*1og(2) + 2/3*1og(31) + 2/3*3^(l/2)*atan(11/3*3^(l/2)) -...
      2/3*log(7) - 2/3*3^(1/2)*atan(5/3*3^(l/2))
» int([x^3   sin(x)   exp(x)], x)
ans =
     [ l/4*x^4,  -cos(x),    exp(x)]
» int(log(sin(x)),x,0,pi/2)
ans =
    -pi/2*log(2)
```

С помощью функции int можно вычислять имеющие аналитическое решение сложные интегралы, например с бесконечными пределами (или одним из пределов), а также кратные интегралы.

```
» int(log(1+exp(-x),x,0,inf)
ans =
    pi^2/12
» syms x a b
» int(int(int(x^2 + y^2)*z, x, 0, a), y, 0, a), z, 0, a)
ans =
    1/3*a^6
```

3. Функция вычисления пределов – ***limit***

Для вычисления пределов аналитически заданной функции F(x) служит функция ***limit***

▪ limit(F, x, a) – возвращает предел символьного выражения F в точке x = a.

Пример:
```
»limit(sin(x)/x, x, 0)
ans =
    1
```

4. Функция разложения в ряд Тейлора – ***taylor***

▪ taylor (f) – возвращает шесть членов ряда Маклорена функции f ;

▪ taylor (f, n, x, a) – возвращает *n* членов ряда Тейлора в точке x = a ;

▪ taylor (f, n) – возвращает *(n - 1)* членов ряда Маклорена функции f ;

▪ taylor (f, a) – возвращает шесть членов ряда Тейлора функции f в точке a.

Примеры:

```
»x = sym('x')
»taylor(sin(x))
ans =
    x – 1/6*x^3 + 1/120*x^5
»taylor(int(sin(x))
ans =
     - 1 +1/2*x^2 – 1/24*x^4
```

5. Функция вычисления суммы рядов – ***symsum***

Для аналитического вычисления суммы ряда служит команда symsum:

- symsum(S) — возвращает символьное значение суммы бесконечного ряда по переменной, найденной автоматически с помощью функции findsym;
- symsum(S, v) — возвращает сумму бесконечного ряда по переменной v;
- symsum(S, a, b) и symsum(S, v, a, b) — возвращают конечную сумму ряда в пределах номеров слагаемых от a до b.

Примеры:
```
» x = sym('x');
» symsum(x^2)
ans =
    1/3*x^3-1/2*x^2+1/6*x
» symsum([x, x^2, x^3], 1, 5)
ans =
     [ 15,  55,  225]
```

6. Функция решение алгебраических уравнений – ***solve***

Для решения систем алгебраических уравнений и одиночных уравнений служит функция solve:

- solve(expr1, expr2,... exprN, var1, var2,... varN) — возвращает значения переменных var1, при которых соблюдаются равенства, заданные выражениями exprI. Если в выражениях не используются знаки равенства, то полагается exprI = 0;
- solve(expr1, expr2, ... exprN) — аналогична предшествующей функции, но переменные, по которым ищется решение, определяются функцией findsym.

Примеры решения уравнений:

```
» syms x y;
» so1ve(x^3 -1, x)
ans =
[            1]
[ -1/2+1/2*i*3^(1/2)]
[ -1/2-1/2*i*3^(1/2)]
» syms a b c
» solve(a*x^2+b*x+c)
ans =
[ 1/2/a*(-b+(b^2-4*a*c)^(1/2))]
[ 1/2/a*(-b-(b^2-4*a*c)^(1/2))]
»S = solve(‘x+y=3’, ‘x*y^2=4’, x, y)
S =
    x : [ 3x1 sym ]
    y : [ 3x1 sym ]
» S.x
ans =
[4]
[1]
[1]
» S.y
ans =
[-1]
[2]
[2]
» solve(‘sin(x)=0.5’, x)
ans =
0.52359877559829887307710723054658
```

7. Решение дифференциальных уравнений – ***dsolve***

Для решения дифференциальных уравнений в форме Коши MATLAB имеет следующую функцию:

• dsolve('eqn1', 'eqn2', ...) — возвращает аналитическое решение системы дифференциальных уравнений с начальными условиями. Они задаются равенствами eqnl (вначале задаются уравнения, затем начальные условия).

По умолчанию независимой переменной считается переменная 't', обычно обозначающая время. Можно использовать и другую переменную, добавив ее в конец списка параметров функции dsolve. Символ D обозначает производную по независимой переменной, то есть d / dt, при этом D2 означает d^2 / dt^2 и т. д. Имя независимой переменной не должно начинаться с буквы D.

Начальные условия задаются в виде равенств 'y(a)=b' или 'Dy(a)=b', где y — независимая переменная, a и b — константы. Если число начальных условий меньше, чем число дифференциальных уравнений, то в решении будут присутствовать произвольные постоянные C1, C2 и т. д.

Примеры применения функции dsolve:

```
» dsolve('D2x = -2*x')
 ans =
     Cl*cos(2^(1/2)*t) + C2*sin(2^(l/2)*t)
 » dsolve('D2y = -2*x + y', 'y(0) = 1 , 'x')
  ans =
      (2*x*exp(x) + (- C2 + 1)*exp(x)^2 + C2 / exp(x)
```

Графические возможности Simbolic

1. Графопостроитель – ***funtool***

Команда funtool создает интерактивный графический калькулятор, позволяющий быстро построить две функции одной переменной - *f(x)* и *g(x)*. Например, одна может задавать собственно функцию, а другая — ее производную. Функции обозначаются как ' f = ' и ' g = ' и после знака равенства можно набрать функции с помощью клавиш калькулятора в его нижней части. С помощью полей 'x = ' и ' a = ' можно задать диапазон изменения переменной x и значение масштабирующего параметра a.

При запуске команды funtool появляются окна для двух функций и окно калькулятора (рис. 4.1). По умолчанию заданы функции $f(x) = x$ и $g(x) = 1$, предел изменения x от -2π до 2π и $a = 1/2$.

Верхний ряд кнопок вычислителя относится только к функции *f(x)* и задает следующие операторы:

- df/dx — символьное дифференцирование *f(x)*;
- int f — символьное интегрирование *f(x)* при наличии замкнутой формы;

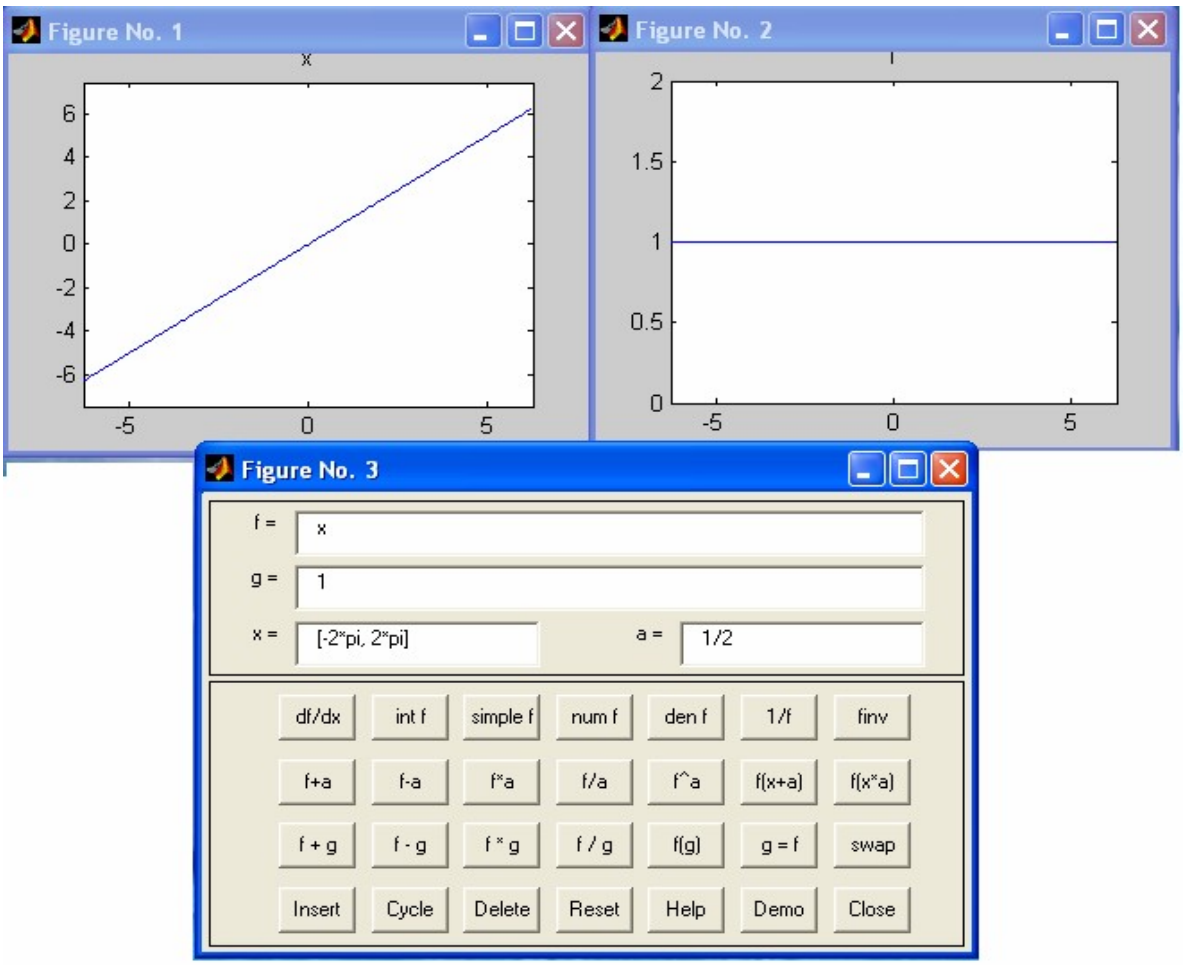


Рис. 4.1. Внешний вид графопостроителя funtool

- simple f — упрощение выражения, если таковое возможно;
- num f — выделение числителя рационального выражения;
- den f — выделение знаменателя рационального выражения;
- 1/f — замена *f(x)* на 1 / *f(x);*
- finv — замена *f(x)* инверсной функцией.

Второй ряд клавиш выполняет операции масштабирования и сдвига *f(x)* с применением параметра 'a'.

Третий ряд клавиш предназначен для осуществления бинарных операций над функциями *f(x)* и *g(x)*.

Четвертый ряд клавиш служит для работы с памятью калькулятор и иных операций:

- Insert — помещает текущую функцию в список функций.
- Cycle — выполняет текущую функцию из списка.
- Delete — удаляет выделенную функцию из списка.
- Reset — устанавливает f, g, x, a и fxl i st в исходное состояние.
- Help — выводит описание калькулятора.
- Demo — запускает демонстрационный пример.
- Close — завершает работу с калькулятором.

Благодаря описанным средствам вычислитель позволяет задать интересующую вас функцию, выполнить ее преобразования (например, дифференцирование и интегрирование) и, наконец, построить график функции и результатов ее преобразования (рис.4.2)

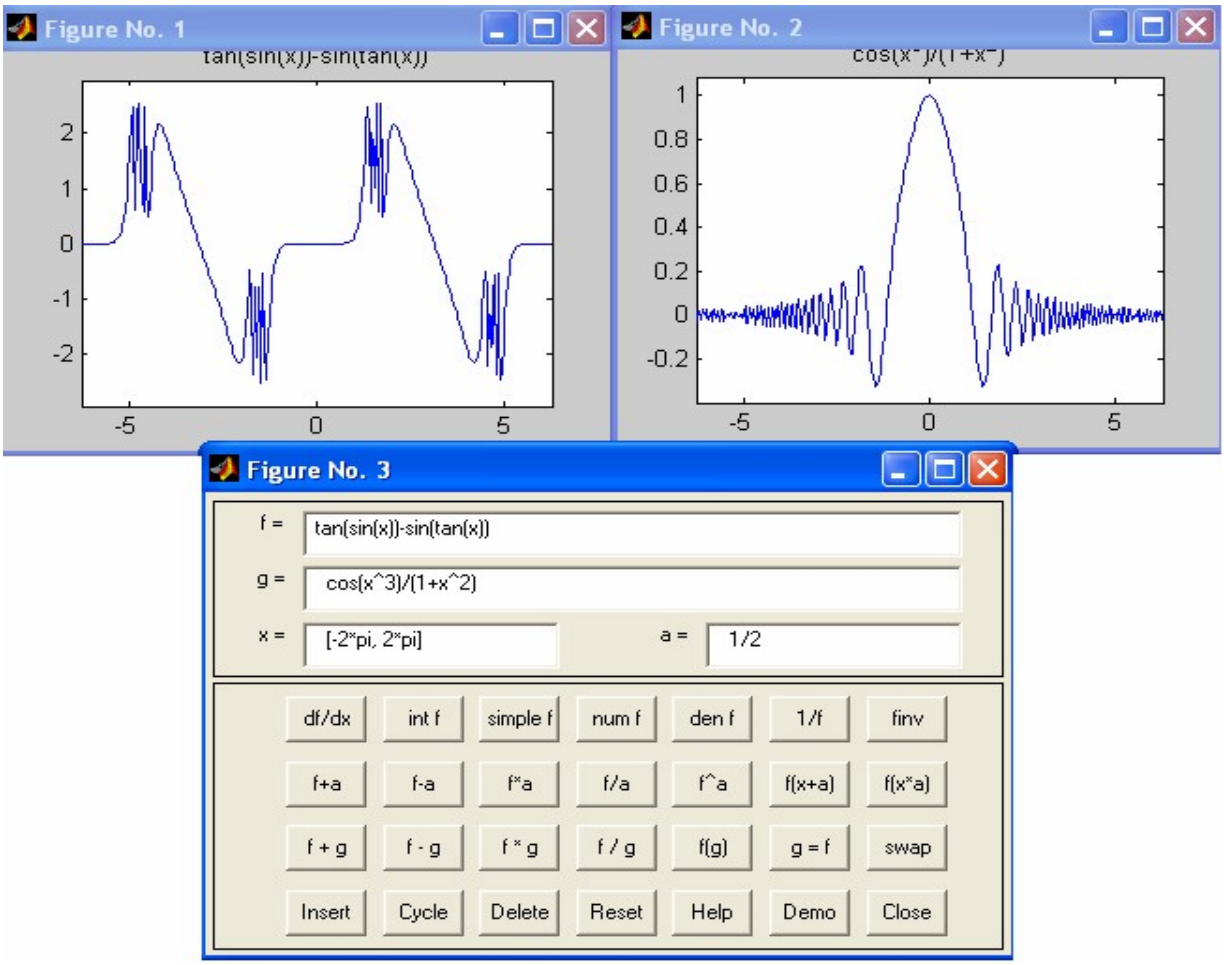


Рис. 4.2. Построения графиков некоторых функций

Таким образом, графопостроитель funtool является весьма удобным средством визуализации графиков самых различных функций.

2. Графики поверхностей – ***ezsurf*** и ***ezsurfc***

Команда ezsurf служит для построения графиков поверхностей, задаваемых функциями двух переменных *f(x, y)*:

- ezsurf(f) — построение поверхности *f(x,y) с* параметрами *x и y*, меняющимися по умолчанию от -2π до 2π;
- ezsurf(f,domain) — построение поверхности*f(x,y)* с пределами изменения *x* и *y*, заданными параметром domain;
- ezsurf(x,y,z) — построение поверхности, заданной параметрически зависимостями *x(s, t), y(s, t), z(s, t)* при *s* и *t*, меняющихся в интервале от -2π до 2π;
- ezsurf(x,y,z,[smin, smax, tmin, tmax]) — построение поверхности, заданной параметрически зависимостями *x(s, t), y(s, t), z(s, t)* при *s* и *t* меняющихся в заданном интервале.

Следующий пример показывает действие этой команды:

```
» syms x y
» ezsurf(rea1(asec(x+i*y)))
```

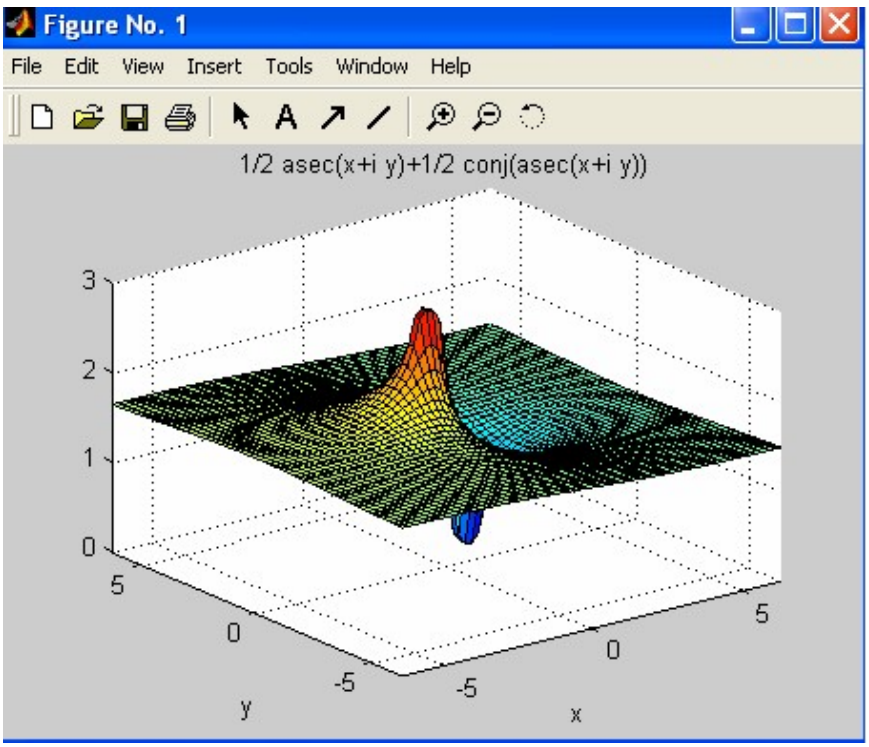


Рис. 4.3. Пример построения графика поверхности командой ezsurf

Аналогичная по синтаксису записи группа команд ezsurfc строит еще и контурный график поверхности на плоскости, лежащей под поверхностью.

Применение средств символьных вычислений в теоретической механике

1. Решение задач статики. При решении некоторых прикладных задач оптимизации желательно получать аналитические решения, устанавливающие функциональную связь между заданными параметрами и величинами, подлежащими определению. В этом случае целевая функция оптимизации имеет явное выражение и её максимум (минимум) определяется обычными методами математического анализа. Рассмотрим следующую задачу о равновесии конструкции:

З а д а н и е С 1. Определение наибольшего значения реакции опоры для составной конструкции

На плоскую конструкцию, состоящую из двух невесомых элементов, которые в точке **C** соединены шарниром, наложены следующие связи:

1) Шарнирная опора на катках в точке **A** ;
2) Неподвижный цилиндрический шарнир в точке **D** ;
3) Горизонтальный невесомый стержень в точке **К.**

На конструкцию действуют:

1. Пара сил с моментом M = 5 кН м (в точке E).
2. Две сосредоточенные силы F_1 = 4 кН (в точке D) и F_2 = 6 кН (в точке A), образующие в точке приложения с контуром элемента соответственно углы α и 30^0 .
3. Перпендикулярно к действующему участку (участок **AB**) равномерно распределённая нагрузка интенсивности q = 10 кН / м.

Длина каждого участка конструкции a = 0,4 м. Предполагая, что конструкция находится в равновесии, найти при каком значении угла наклона α силы

F_1 реакция шарнира D ($R_D = \sqrt{X_D^2 + Y_D^2}$) имеет наибольшее значение, определить для этого случая реакции всех связей и усилия в шарнире **C** .

Решение. Для определения искомых реакций можно поступить двояко, либо мысленно расчленить систему тел (составную конструкцию) на отдельные твёрдые тела и рассмотреть равновесие каждого из тел в отдельности, либо, применяя принцип затвердевания, вначале рассмотреть равновесие всей конструкции, а затем равновесие её отдельных элементов.

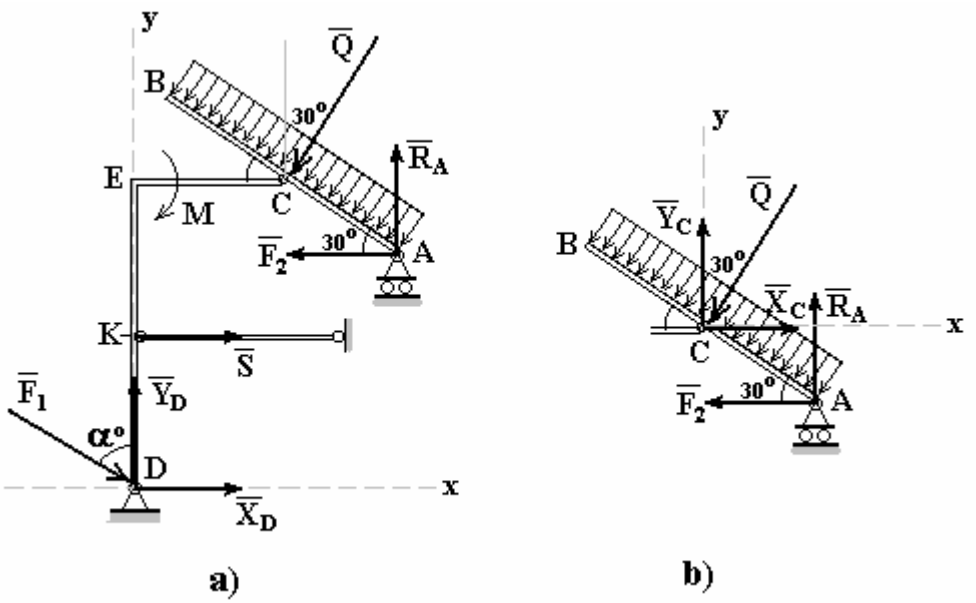


Рис. 4.4. Схема составной конструкции

I. Прежде рассмотрим равновесие всей конструкции.

1. Объект равновесия – составная конструкция.

2. Связи – шарнирная опора на катках в точке **A**, неподвижный цилиндрический шарнир в точке **D** и невесомый стержень в точке **К** . На конструкцию действуют сосредоточенные силы $\mathbf{F_1}$ и $\mathbf{F_2}$, равномерно распределённая нагрузка интенсивности **q** и пара сил с моментом M.

3. Действие связей заменим их реакциями $\mathbf{R_A}$, $\mathbf{X_D}$, $\mathbf{У_D}$, **S** . Равномерно распределённую нагрузку заменим равнодействующей

$$Q = q\ 2\ a = 8\ kH.$$

Другие активные силы - $\mathbf{F_1}$, $\mathbf{F_2}$ и пара сил с моментом M.

4. Составим уравнения равновесия (система координат Dxy изображена на ри-

сунке 4.5а)

$$\sum_{i=1}^{n} F_{ix} = X_D - F_2 + F_1 \sin\alpha - Q\cos 60^0 + S = 0$$

$$\sum_{i=1}^{n} F_{iy} = Y_D - Q \sin 60^0 + R_A - F_1 \cos \alpha = 0 \qquad (а)$$

$$\sum_{i=1}^{n} m_{iC} = - F_2 \sin 30^0 a + F_1 \cos\alpha \; a + F_1 \sin\alpha \; 2a - M - Y_D a +$$

$$+ R_A a \cos 30^0 + S a + X_D 2 a = 0.$$

Данная система уравнений содержит четыре неизвестных $\mathbf{X_D}$, $\mathbf{У_D}$, $\mathbf{S}$, $\mathbf{R_A}$.

II. Рассмотрим теперь равновесие правой части конструкции (Рис. 4.5b).

1. Объект равновесия – правая часть конструкции.

2. Связи – шарнирная опора на катках в точке **A** и соединительный шарнир **C**. На объект равновесия действует сосредоточенная сила $\mathbf{F_2}$, равнодействующая распределённой нагрузки **Q** и реакции связей $\mathbf{X_C}$, $\mathbf{У_C}$, $\mathbf{R_A}$.

3. Составим уравнения равновесия (в системе координат Cxy)

$$\sum_{i=1}^{n} F_{ix} = X_C - F_2 - Q \sin 30^0 = 0$$

$$\sum_{i=1}^{n} F_{iy} = R_A + Y_C - Q \cos 30^0 = 0 \qquad (б)$$

$$\sum_{i=1}^{n} m_{iC} = - F_2 \sin 30^0 a + R_A \cos 30^0 a = 0$$

Решаем систему шести уравнений (а) – (б) с шестью неизвестными, система (б) является независимой и содержит три неизвестных. Находим, из третьего уравнения

$$R_A = F_2 \operatorname{tg} 30^0 = 6 \operatorname{tg} 30^0 = 3{,}464 \text{ кН} ,$$

из первого – $X_C = F_2 + Q \sin 30^0 = 10$ кН,

из второго уравнения $У_C = Q \cos 30^0 – R_A = 3{,}464$ кН.

Решаем систему уравнений (а), для этого воспользуемся символьными вычислениями системы MATLAB

```
syms XD YD S al                                  %символьные переменные
Q1=solve('XD-6+4*sin(al)-8*0.5+S=0','YD-8*0.866+3.464-4*cos(al)=0',...
'-6*0.5*0.4+4*0.4*cos(al)+4*2*0.4*sin(al)-5-YD*0.4+3.464*0.4*0.866+S*0.4+XD*2*0.4=0'...
,XD,YD,S);                                       %решение уравнений равновесия
digits(6)                                        %количество значащих цифр
S=vpa(Q1.S);                                     %величина реакции  S
XD=vpa(Q1.XD);                                   %величина реакции  XD
YD=vpa(Q1.YD);                                   %величина реакции  YD
RD=sqrt(XD^2+YD^2)                               %величина полной реакции шарнира D
ezplot(RD,-pi/2,pi)                              %график функции RD=RD(al)
```

```
solve(diff(RD,al),al)                    %нахождение экстремума
S =
   1.03583
XD =
     5.96418 – 4.*sin(al)
YD =
      3.46400 + 4.*cos(al)
RD =
     ((5.96418 – 4.*sin(al))^2 + (3.46400 + 4.*cos(al))^2)^(1/2)
ans =
      [   2.09698 ]
      [ - 1.04461 ]
```

Таким образом, наибольшее значение полная реакция R_D шарнира D имеет при $\alpha = -1{,}04461$, а наименьшее – при $\alpha = 2{,}09698$.

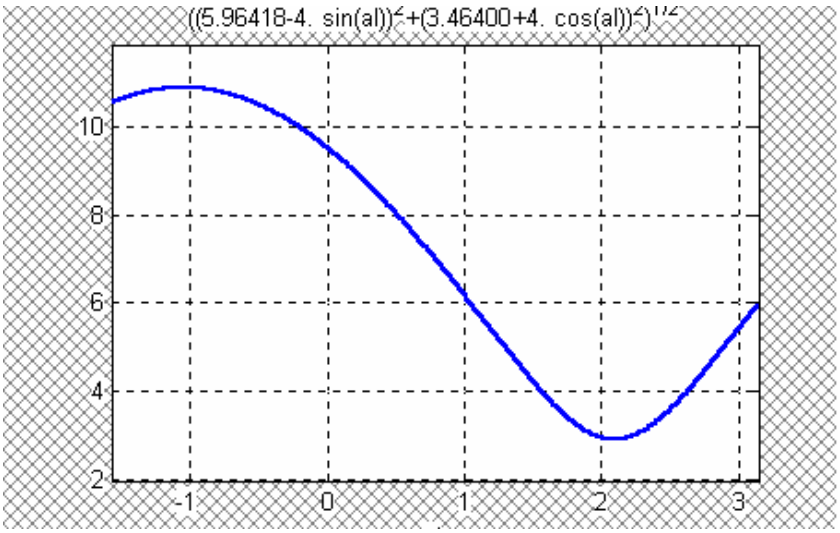


Рис. 4.5. График зависимости $R_D = \sqrt{X_D^2 + Y_D^2} = R_D(\alpha)$.

Реакции связей R_A, X_C, Y_C, S не зависят от угла величины α

$R_A = 3{,}464$ кН , $X_C = 10$ кН , $Y_C = 3{,}464$ кН , $S = 4{,}03583$ кН,

а наибольшее и наименьшее значения полной реакции шарнира D соответственно равны :

$R_{Dmin} =$ 2,89717 кН при $\alpha =$ 2,09698,
$R_{Dmax} =$ 10,989715 кН при $\alpha = -1{,}04461$.

Варианты задания. Конструкция состоит из двух частей. Установить, при каком значении угла α для силы F_1 реакция опоры A ($R_A = \sqrt{X_A^2 + Y_A^2}$) имеет наибольшее значение, определить для этого случая реакции всех опор, а также соединения C.

Необходимые для расчёта данные приведены в таблице 4.1

Таблица 4.1

Номер варианта	P_1	P_2	M, кН м	q, кН/м	Номер варианта	P_1	P_2	M, кН м	q, кН/м
	кН					кН			
1	5,0	-	24,0	0,8	**16**	7,0	10,0	14,0	3,8
2	6,0	10,0	22,0	1,0	**17**	9,0	12,0	26,0	4,0
3	7,0	9,0	20,0	1,2	**18**	11,0	10,0	18,0	3,5
4	8,0	-	18,0	1,4	**19**	13,0	9,0	30,0	3,0
5	9,0	-	16,0	1,6	**20**	15,0	8,0	25,0	2,5
6	10,0	8,0	25,0	1,8	**21**	10,0	7,0	20,0	2,0
7	11,0	7,0	20,0	2,0	**22**	5,0	6,0	15,0	1,5
8	12,0	6,0	15,0		**23**	8,0	5,0	10,0	1,4
9	13,0	-	10,0		**24**	11,0	4,0	5,0	1,3
10	14,0	-	12,0		**25**	14,0	6,0	7,0	1,2
11	15,0	5,0	14,0		**26**	12,0	8,0	9,0	1,1
12	12,0	4,0	16,0		**27**	10,0	7,0	11,0	1,0
13	9,0	6,0	18,0		**28**	8,0	9,0	13,0	1,2
14	6,0	-	20,0		**29**	6,0	10,0	15,0	1,4
15	5,0	8,0	22,0		**30**	10,0	12,0	17,0	1,6

Схемы конструкций для каждого из вариантов приведены на рисунках 4.6 – 4.8

2. Решение задач кинематики точки.

З а д а н и е К1. Определение скорости и ускорения точки по заданным уравнениям её движения

По заданным уравнениям движения точки M $x = x(t)$ $y = y(t)$ установить вид её траектории и для момента времени $t = t_1$ (с) найти положение точки на траектории, её скорость, полное, касательное и нормальное ускорение, а также радиус кривизны траектории.

Необходимые для решения данные приведены в таблице 4.2.

Пример выполнения задания К1. Движение точки на плоскости Oxy определяется уравнениями $\begin{cases} x = 4\cos t^2 \\ y = \sin t^2 \end{cases}$ x,y - в метрах, t - сек.

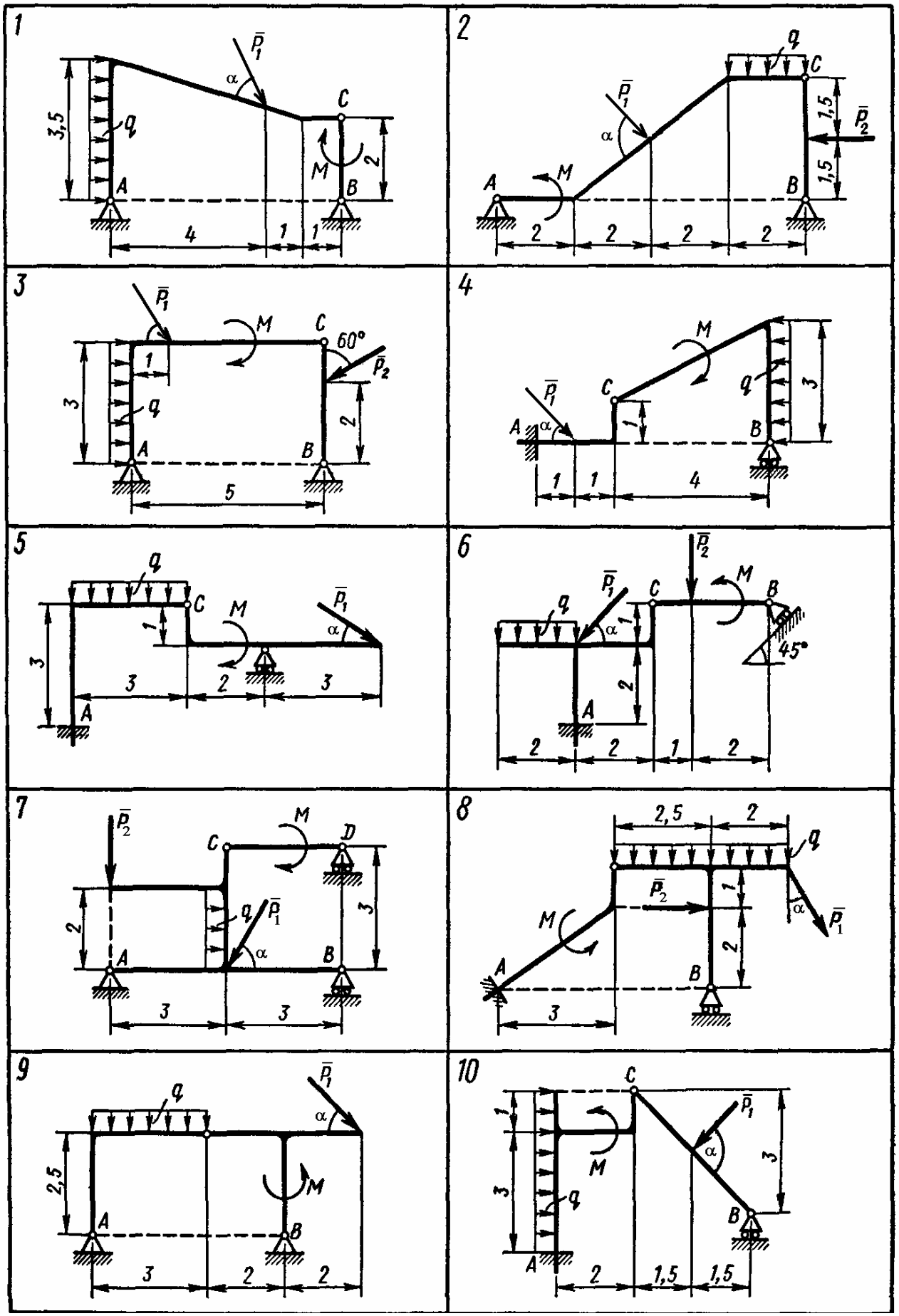


Рис. 4.6

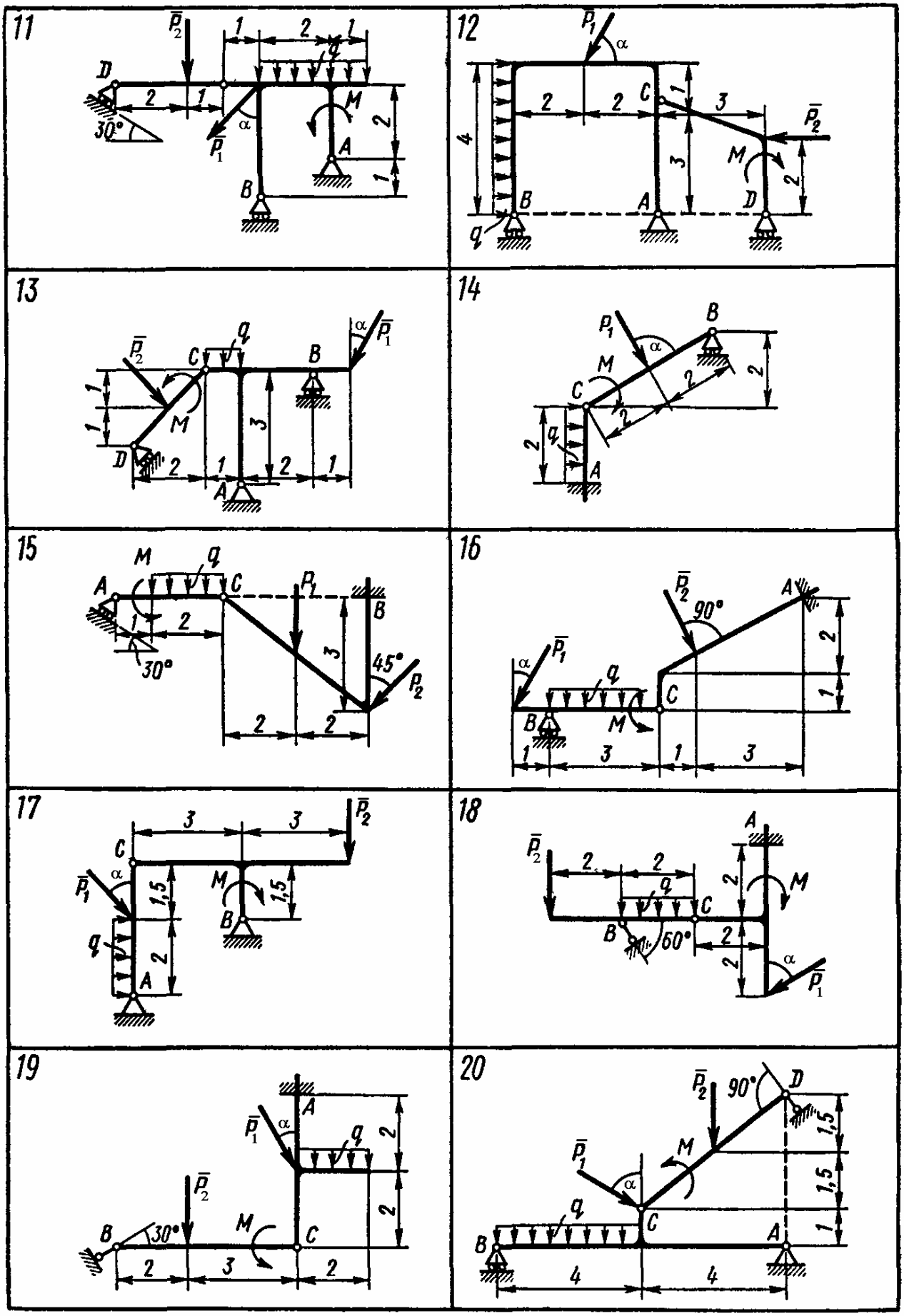


Рис. 4.7

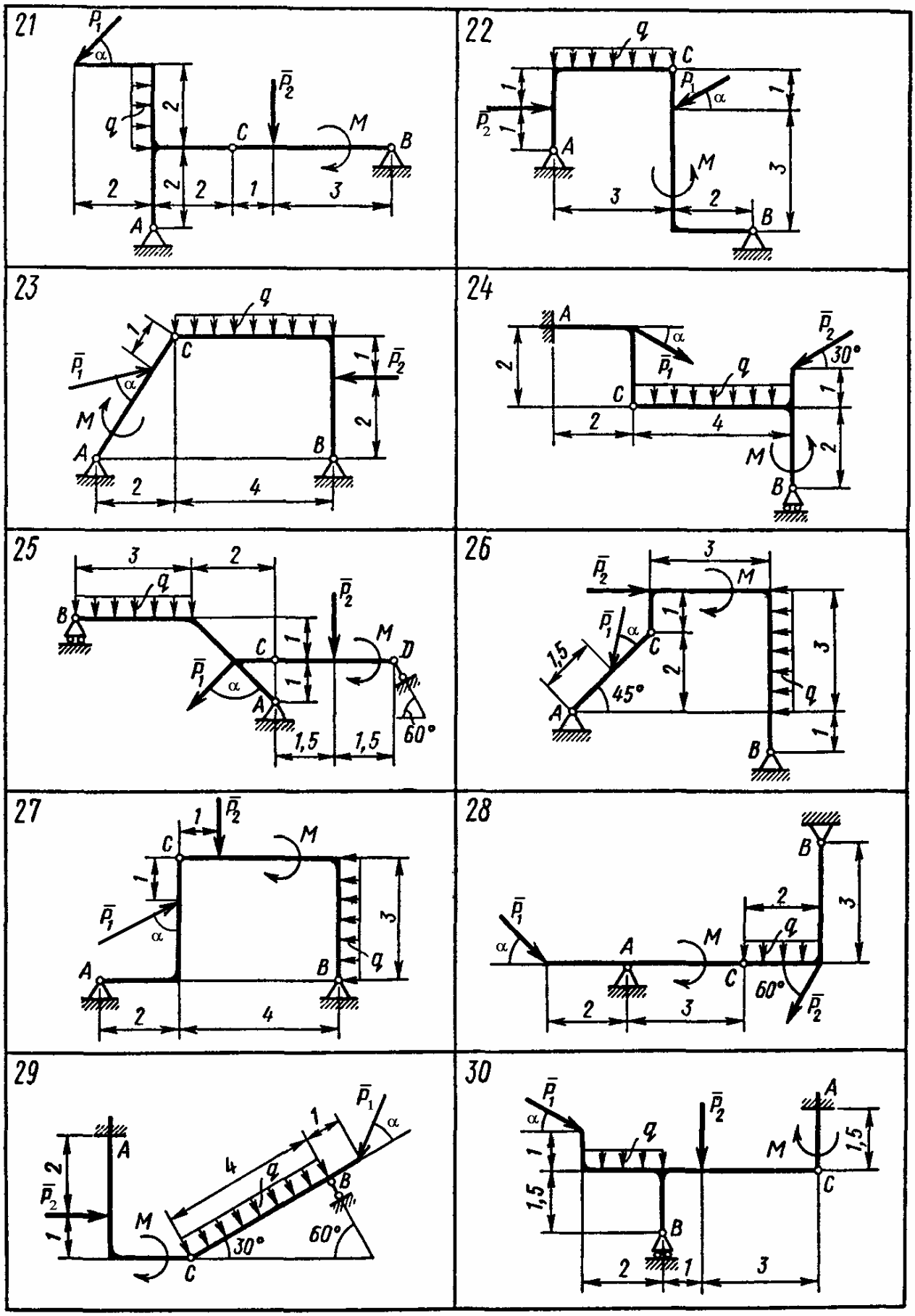


Рис. 4.8

Задан момент времени $t_1 = 0$.

Необходимо:

1. Определить уравнение траектории точки.
2. Построить траекторию и указать на траектории положение точки в заданный момент времени t_1 .
3. Для заданного момента времени найти скорость и ускорение точки, её тангенциальное и нормальное ускорения, значение радиуса кривизны траектории.
4. Векторы скорости и ускорения точки показать на траектории.

<u>Решение.</u>

1. Движение точки задано координатным способом.

Для определения уравнения траектории точки, соответствующей заданным уравнениям движения, исключаем время t . Из уравнений движения имеем

$$\cos^2 t^2 = x^2 / 16 \ , \quad \sin^2 t^2 = y^2 \ .$$

Применяя основное тригонометрическое тождество, получим

$$\frac{x^2}{16} + y^2 = 1 \ .$$

Таким образом, траекторией точки является эллипс.

2. Определяем скорость точки

$V_x = \dot{x} = -8 t \sin t^2$, $V_y = \dot{y} = 2 t \cos t^2$,

$$V = \sqrt{V_x^2 + V_y^2} = 2 t \sqrt{16 \sin t^2 + \cos t^2} \ ,$$

при $t_1 = 0$ с : $V_x = 0$ и $V_y = 0$

$$V = \sqrt{V_x^2 + V_y^2} = 0 \text{ м / с.}$$

При малых значениях t ($0 < t << 1$) скорость определяется выражениями:

$$V_x = -8 t^3 \ , V_y = 2 t \ , V = \sqrt{V_x^2 + V_y^2} = 2 t.$$

3. Определяем ускорение точки

$a_x = \dot{V}_x = -8 \sin t^2 - 16 t^2 \cos t^2$, $a_y = 2 \cos t^2 - 4 t \sin t^2$

при $t_1 = 0$ с $a_x = 0$ м / с^2 , $a_y = 2$ м / с^2 ,

$$a = \sqrt{a_x^2 + a_y^2} = 2 \text{ м / с}^2$$

При малых значениях t ($0 < t << 1$) ускорение представляется в виде:

$$a_x = -8 t^2 - 16 t^2 = -24 t^2 , \quad a_y = 2 - 4 t^3 , \quad a = \sqrt{a_x^2 + a_y^2} = 2 \text{ м / с}^2 .$$

4. По формулам $a_\tau = \dfrac{dV}{dt} = \dfrac{v_x a_x + v_y a_y}{v}$, $a_n = \sqrt{a^2 - a_\tau^2}$

находим тангенциальное и нормальное ускорения точки в момент $t_1 \to 0$ с ,

$$a_\tau = \frac{dV}{dt} = \frac{v_x a_x + v_y a_y}{v} = \frac{8t^3 \times 24t^2 + 2t \times (2 - 4t^3)}{2t} = 2 \text{ м / с}^2,$$

$$a_n = \sqrt{a^2 - a_\tau^2} = \frac{|a_x V_y - a_y V_x|}{V} = \frac{|-24t^2 \cdot 2t + 16t^3|}{2t} = 16\,t^2 \text{ м / с}^2$$

при $t = 0$ $a_n = 0$ м / с^2 .

5. По формуле $\rho = \dfrac{V^2}{a_n}$ находим радиус кривизны траектории $\rho = \dfrac{4t^2}{16t^2} = 0{,}25$ м .

Изобразим траекторию движения точки. В момент времени $t_1 = 0$ с точка М находится в положении, определяемом координатами $x_1 = 4$ м, $y_1 = 0$ м.

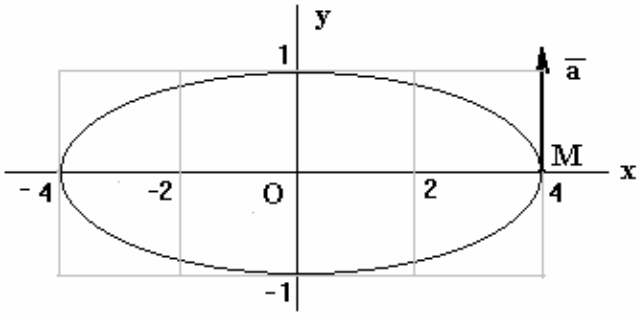


Рис. 4.9

Ответ: при $t_1 = 0$ с : V = 0 м / с; a = 2 м / с^2; a_τ= **2** м / с^2; a_n= 0 м / с^2; ρ = 0,25 м.

Решение задачи кинематика точки в общем виде можно получить путём применения символьных вычислений системы MATLAB. При движении точки на плоскости (x,y) её кинематические характеристики определяются следующей m-функцией:

```
function[Vx,Vy,V,ax,ay,a,at,an,ro]=kinema(x,y);
%кинематика точки
%входные параметры - x(t), y(t)- уравнения движения точки
Vx=diff(x,'t');
Vy=diff(y,'t');
V=sqrt(Vx^2+Vy^2);
ax=diff(Vx,'t');
ay=diff(Vy,'t');
a=sqrt(ax^2+ay^2);
at=(Vx*ax+Vy*ay)/V;
an=sqrt(a^2-at^2);
ro=V^2/an;
```

В качестве примера рассмотрим уравнения движения $x = 4\,t$, $y = 16\,t^2 - 1$. Необходимо найти кинематические характеристики в любой момент времени, а также в момент времени $t_1 = 0{,}5$ с (x, y – м; t – с).

Задача решается с помощью следующего m – сценария:

```
>> syms x y t Vx Vy V ax ay a at an ro
>> x=4*t;y=16*t^2-1;
>> [Vx,Vy,V,ax,ay,a,at,an,ro]=cinema2d(x,y)
Vx =
     4
Vy =
    32*t
V =
    4*(1+64*t^2)^(1/2)
```

```
ax =
    0
ay =
   32
a =
   32
at =
   256*t/(1+64*t^2)^(1/2)
an =
   32*(1-64*t^2/(1+64*t^2))^(1/2)
 ro =
    1/32*(16+1024*t^2)/(1-64*t^2/(1+64*t^2))^(1/2)
```

Вид траектории определяется командой - ezplot(x^2-1,-2,2)

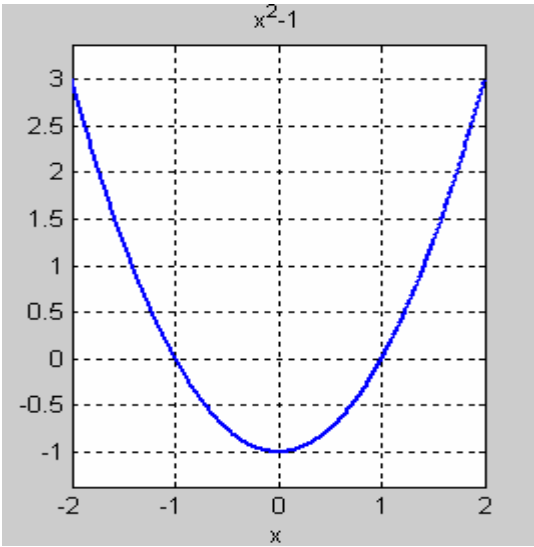


Рис. 4.10. Траектория движения точки

В момент времени t_1 = 0,5 с кинематические характеристики определяются командой подстановки - subs

```
>> K=[Vx,Vy,V,ax,ay,a,at,an,ro];
>> t=0.5;
>> subs(K)
ans =
  Columns 1 through 6
    4.0000   16.0000   16.4924         0   32.0000   32.0000
  Columns 7 through 9
   31.0446    7.7611   35.0464
```

Аналогичным образом решается задача кинематики точки при её движении в пространстве: в этом случае m – функция задаётся в виде

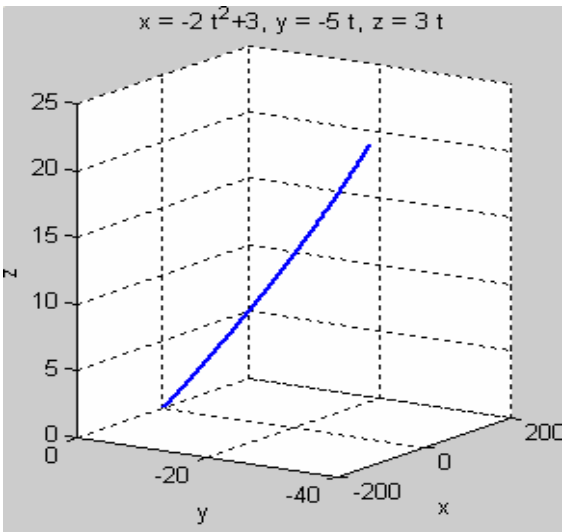
```
function[Vx,Vy,Vz,V,ax,ay,az,a,at,an,ro]=cinema3d(x,y,z);
%кинематика точки
%входные параметры - x(t), y(t), z(t) - уравнения движения точки
Vx=diff(x,'t');
Vy=diff(y,'t');
Vz=diff(z,'t');
V=sqrt(Vx^2+Vy^2+Vz^2);
ax=diff(Vx,'t');
```

```
ay=diff(Vy,'t');
az=diff(Vz,'t');
a=sqrt(ax^2+ay^2);
at=(Vx*ax+Vy*ay+Vz*az)/V;
an=sqrt(a^2-at^2);
ro=V^2/an;
```

Рассмотрим пример

```
>> syms x y z t
>> x=-2*t^2+3;y=-5*t;z=3*t;
>> [Vx,Vy,Vz,V,ax,ay,az,a,at,an,ro]=cinema3d(x,y,z)
Vx =
     -4*t
Vy =
     -5
Vz =
     3
V =
     (16*t^2+34)^(1/2)
ax =
    -4
ay =
     0
az =
     0
a =
     4
at =
    16*t/(16*t^2+34)^(1/2)
an =
    4*(1-16*t^2/(16*t^2+34))^(1/2)
ro =
   1/4*(16*t^2+34)/(1-16*t^2/(16*t^2+34))^(1/2)
```

Траектории движения – ezplot3(x,y,z,[0,8])



В момент времени $t_1 = 0,5$ с

```
>> K=[Vx,Vy,Vz,V,ax,ay,az,a,at,an,ro];
>> t=0.5;
>> subs(K)
```

```
ans =
Columns 1 through 6
-2.0000   -5.0000    3.0000    6.1644   -4.0000         0
Columns 7 through 11
0    4.0000    1.2978    3.7836   10.0433
```

Таблица 4.2

Номер варианта	Уравнения движения		t_1, с
	x = x (t), см	y = y (t), см	
1	$- t^2 + 3$	$- 5 t$	1/2
2	$4 \cos^2(\pi t / 3) + 2$	$4 \sin^2(\pi t / 3) - 1$	1
3	$- \cos(\pi t^2 / 3) + 3$	$\sin(\pi t^2 / 3) - 1$	1
4	$4 t + 4$	$- 4 / (t + 1)$	2
5	$2 \sin(\pi t / 3) – 1$	$- 3 \cos(\pi t / 3) + 4$	1
6	$3 t^2 + 2$	$- 4 t$	1/2
7	$7 \sin(\pi t^2 / 6) + 3$	$2 - 7 \cos(\pi t^2 / 6)$	1
8	$- 3 / (t + 2)$	$3 t + 6$	2
9	$3 t^2 – t + 1$	$5 t^2 – 5 t / 3 - 2$	1
10	$- 4 \cos(\pi t / 3)$	$- 2 \sin(\pi t / 3) – 3$	1
11	$- 4 t^2 + 1$	$- 3 t$	1/2
12	$5 \sin^2(\pi t / 6)$	$- 5 \cos^2(\pi t / 6) – 3$	1
13	$5 \cos(\pi t^2 / 3)$	$- 5 \sin(\pi t^2 / 3)$	1
14	$- 2 t – 2$	$- 2 / (t + 1)$	2
15	$4 \cos(\pi t / 3)$	$- 3 \sin(\pi t / 3)$	1
16	$3 t$	$4 t^2 + 1$	1/2
17	$7 \sin^2(\pi t / 6) – 5$	$- 7 \cos^2(\pi t / 6)$	1
18	$1 + 3 \cos(\pi t^2 / 3)$	$3 \sin(\pi t^2 / 3) + 3$	1
19	$- 5 t^2 – 4$	$3 t$	1
20	$2 – 3 t – 6 t^2$	$3 – 3 t / 2 – 3 t^2$	0
21	$6 \sin(\pi t^2 / 6) – 2$	$6 \cos(\pi t^2 / 6) + 3$	1
22	$7 t^2 – 3$	$5 t$	1/4
23	$3 – 3 t^2 + t$	$4 – 5 t^2 + 5 t / 3$	1
24	$- 4 \cos(\pi t / 3) – 1$	$- 4 \sin(\pi t / 3)$	1
25	$- 6 t$	$- 2 t^2 – 4$	1
26	$8 \cos^2(\pi t / 6) + 2$	$- 8 \sin^2(\pi t / 6) - 7$	1
27	$- 3 – 9 \sin(\pi t^2 / 6)$	$- 9 \cos(\pi t^2 / 6) + 5$	1
28	$- 4 t^2 + 1$	$- 3 t$	1
29	$5 t^2 + 5 t / 3 – 3$	$3 t^2 + t + 3$	1
30	$2 \cos(\pi t^2 / 3) – 2$	$- 2 \sin(\pi t^2 / 3) + 3$	1

4.2. Пакет оптимизации Optimization Toolbox

Назначение и возможности пакета

Пакет оптимизации (Optimization Toolbox) — это библиотека функций, расширяющих возможности системы MATLAB по численным вычислениям и предназначенная для решения задач оптимизации и систем нелинейных уравнений. Поддерживает основные методы оптимизации функций ряда переменных [6]:

• Безусловная оптимизация нелинейных функций.

• Метод наименьших квадратов.

• Решение нелинейных уравнений.

• Линейное программирование.

• Квадратичное программирование.

• Условная минимизация нелинейных функций.

• Методы минимакса.

• Многокритериальная оптимизация.

Рассматриваемый пакет дает возможности решать задачи минимизации функций, нахождения решений уравнений, задачи аппроксимации («подгонки» кривых под экспериментальные данные).

Различные типы таких задач вместе с применяемыми для их решения функциями пакета Optimization Toolbox приведены в таблице 4.3.

Таблица 4.3.

Типы задач, решаемых средствами пакета Optimization Toolbox

Тип задачи	Математическая запись	Функция MATLAB
Задачи минимизации		
Скалярная (одномерная) минимизация	$\min\limits_{x} \mathbf{f(a)}, \quad a_1 < a < a_2$	**fminunc,**

Тип задачи	Математическая запись	Функция MATLAB
	Задачи минимизации	
Безусловная минимизация (без ограничений)	$\min_x \mathbf{f(x)}$	**fminunc,** **fminsearch**
Линейное программирование	$\min_x \mathbf{f}^T\mathbf{x}$ при условиях $\mathbf{A\,x < b}$, $\mathbf{Aeq\,x=beq}$, $\mathbf{x}_L < \mathbf{x} < \mathbf{x}_U$	**linprog**
Квадратичное программирование	$\min_x \frac{1}{2}x^T Hx + f^T x$ при $\mathbf{A\,x}$ £ $\mathbf{b}$, $\mathbf{Aeqx = beq}$, $\mathbf{x}_L < \mathbf{x} < \mathbf{x}_U$	**quadprog**
Минимизация при наличии ограничений	$\min_x \mathbf{f(x)}$ при условиях $\mathbf{c(x)}$ £ $\mathbf{0}$, $\mathbf{ceq(x) = 0}$, $\mathbf{A\,x}$ £ $\mathbf{b}$, $\mathbf{Aeq\,x = beq}$, $\mathbf{x}_L$ £ $\mathbf{x}$ £ $\mathbf{x}_U$	**fmincon**
Достижение цели	$\min_{x,y} \gamma$ при условиях $\mathbf{F(x) - w}g$ £ **goal,** $\mathbf{c(x)}$ £ $\mathbf{0}$, $\mathbf{ceq(x) = 0}$, $\mathbf{Ax}$ £ $\mathbf{b}$, $\mathbf{Aeq\,x = beq}$, $\mathbf{x}_L$ £ $\mathbf{x}$ £ $\mathbf{x}_U$	**fgoalattain**
Минимакс	$\min_x \max_{\{F_i\}} \{\mathbf{F_i(x)}\}$ при условиях $\mathbf{c(x) < 0}$, $\mathbf{ceq(x)=0}$, $\mathbf{Ax}$ £ $\mathbf{b}$, $\mathbf{Aeq\,x = beq}$, $\mathbf{x}_L$ £ $\mathbf{x}$ £ $\mathbf{x}_U$	**fminmax**
Полубесконечная минимизация	$\min_x \mathbf{f(x)}$ при условиях $\mathbf{K(x, w) < 0}$ для всех $\mathbf{w}$, $\mathbf{c(x)}$ £ $\mathbf{0}$, $\mathbf{ceq(x) = 0}$, $\mathbf{Ax}$ £ $\mathbf{b}$, $\mathbf{Aeq\,x = beq}$, $\mathbf{x}_L$ £ $\mathbf{x}$ £ $\mathbf{x}_U$	**fseminf**

Окончание

Тип задачи	Математическая запись	Функция MATLAB
Нахождение решений уравнений		
Линейные уравнения	**C x = d**, *n* уравнений, *n* переменных	\ (оператор левого деления, **slash**)
Нелинейное уравнение одной переменной	**f (a) = 0**	**fzero**
Нелинейные уравнения многих переменных	**F(x) = 0,** *n* уравнений, *n* неизвестных	**fsolve**
Задачи аппроксимации («подгонки» кривых)		
Линейный метод Наименьших квадратов (МНК)	$\min_x \|\|\mathbf{Cx - d}\|\|_2^2$, *m* уравнений, *n* переменных	\ (оператор левого деления, **backslash**)
Неотрицательный линейный МНК	$\min_x \|\|\mathbf{Cx - d}\|\|_2^2$ при условии $x \geq 0$	**lsqnonneg**
Линейный МНК при наличии ограничений	$\min_x \|\|\mathbf{Cx - d}\|\|_2^2$ при условиях **Ax £ b, Aeq x = beq**, $\mathbf{x}_L$ £ **x** £ $\mathbf{x}_U$	**lsqlin**
Нелинейный МНК	$\min_x \|\| F(x) \|\|_2^2 = \frac{1}{2}\sum_i f_i^2(x)$ при условии $\mathbf{x}_L$ £ **x** £ $\mathbf{x}_U$	**lsqnonlin**
Нелинейная «подгонка» кривой	$\min_x \frac{1}{2} \|\| F(x, xdata) - ydata \|\|_2^2$ при условии $\mathbf{x}_L$ £ **x** £ $\mathbf{x}_U$	**lsqcurvefit**

Принятые обозначения:

• *a* — скалярный аргумент; x, y — в общем случае векторные аргументы;

• **f(a), f(x)** — скалярные функции; **F(x), c(x), ceq(x), K(x, w)** — векторные функции;

• **A**, **Aeq, C, H —** матрицы;

• **b, beq, d, f, w, goal, xdata, ydata —** векторы;

• $\mathbf{x_L}$, $\mathbf{x_U}$ — соответственно, нижняя и верхняя границы области изменения аргумента.

Применяемые алгоритмы

В рамках пакета Optimization Toolbox **все** задачи оптимизации делятся на две группы: задачи малой и средней размерности и задачи большой размерности. Соответственно, такое же деление принято для алгоритмов решения данных задач. Разумеется, это не означает, что для решения задач средней размерности нельзя применять алгоритмы большой размерности и наоборот. Просто алгоритмы той или иной группы более эффективны для задач своей размерности.

В пакете Optimization Toolbox реализован широкий набор алгоритмов для решения задач оптимизации средней и малой размерности. Основными для задач без ограничений являются симплексный метод Нелдера—Мида и квазиньютоновские методы. Для решения задач с ограничениями, минимакса, достижения цели и полубесконечной оптимизации использованы алгоритмы квадратичного программирования.

Задачи, сводящиеся к нелинейным МНК, решаются с помощью алгоритмов Ньютона—Рафсона и Левенберга—Марквардта.

Вспомогательные процедуры одномерной (скалярной) оптимизации используют алгоритмы квадратичной (параболической) и кубической интерполяции.

Рассмотрим наиболее часто применяемые в задачах оптимизации функции линейного (linprog) и нелинейного (fmincon) программирования.

1. Функция fmincon

furincon — функция поиска минимума скалярной функции многих переменных при наличии ограничений вида

c(x) £ 0, ceq(x) = 0 ,

A x £ b, Aeq x = beq,

lb < x < ub

- задача нелинейного программирования.

Функция записывается в виде

x = fmincon(fun, x0, A, b)
x = fmincon(fun, x0, A, b, Aeq, beq)
x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub)
x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon)
x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, options)
x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, opt1ons, Pl, P2, ...)
[x, fval] = fmincon(...) – возвращается не только оптимальное значение векторного аргумента, но и значение целевой функции в точке минимума **fval** ;
[x, fval, exitflag] = fmincon(…) – то же, что и предыдущая функция, но возвращается ещё информация о характере завершения вычисления **exitflag** ;
[x, fval, exitflag, output] = fmincon(…) – то же, что и предыдущая функция, но возвращается ещё информация о результатах оптимизации (выходная структура) **output** ;
[x, fval, exitflag, output, lambda] = fmincon(...) – то же, что и предыдущая функция, но возвращаются ещё множители Лагранжа **lambda** ;
[x, fval, exitflag, output, lambda, grad] = fmincon(…) – то же, что и предыдущая функция, но возвращается ещё величина градиента функции в точке минимума **grad** ;
[x, fval, exitflag, output, lambda, grad, hessian] = fmincon(…) – то же, что и предыдущая функция, но возвращается ещё величина гессиана **H** функции в точке минимума.

Аргументы функции:

fun — векторная функция векторного аргумента. Должна быть задана либо с помощью функции inline, например:
» fun = inline('sin(x.*x)'):
либо как m -файл, например:
function F = myfun(x)
F = …
nonlcon — функция, возвращающая значения функций-ограничений, а при необходимости (если задано options = optimset('Grad-Constr', 'on')) и их градиентов; должна быть оформлена в виде m-файла, например:
function [c, ceq] = mycon(x)
c = ... % Вычисление левых частей нелинейных неравенств
ceq =... % Вычисление левых частей нелинейных равенств
function [c, ceq, GC, GCeq] = mycon(x)
c = ... % Вычисление левых частей нелинейных неравенств
ceq = ... % Вычисление левых частей нелинейных равенств
GC = ... % Градиенты неравенств

GCeq = ... % Градиенты равенств

∗ **Options** – опции (их можно изменять, используя функцию optimset):

o DerivativeCheck — задает проверку соответствия производных, определенных пользователем, их вычисленным оценкам в виде первых разностей;

o Diagnostics — вывод диагностической информации о минимизируемой функции;

o DiffMaxChange — максимальные значения изменений переменных при определении первых разностей;

o DiffMinChange — минимальные значения изменений переменных при определении первых разностей;

o Display — уровень отображения: 'off' — вывод информации отсутствует, 'iter' — вывод информации о поиске решения на каждой итерации, 'final' — вывод только итоговой информации;

o Goal ExactAchieve — определяет количество целей, которые должны быть достигнуты «точно»;

o GradConstr — использование градиентов для ограничений (опция имеет смысл в случае применения аргумента nonlcon), возможные значения — 'off и 'on';

o GradObj — использование градиента для целевой функции, определяемого пользователем (возможные значения — 'off и 'on');

o MaxFunEvals — максимальное число вычислений функции;

o Maxlter — максимальное допустимое число итераций;

o MeritFunction — устанавливает вид функции оценки качества достижения цели (возможные значения 'multiobj' или 'singleobj');

o TolCon — допуск останова вычислений при нарушении ограничений;

o TolFun — допуск останова вычислений по величине изменений функции;

o TolX — допуск останова вычислений по величине изменений x**;**

• exitflag — информация о характере завершения вычислений: если эта величина положительна, то вычисления завершились нахождением решения x, если она равна нулю, то останов произошел в результате выполнения предельного числа итераций, если данная величина отрицательна, то решение не найдено;

• lambda — множители Лагранжа, соответственно, для различных типов ограничений:

o 1ambda. lower — для нижней границы lb;

o 1ambda. upper — для верхней границы ub;

o 1ambda. ineqlin — для линейных неравенств;

o 1ambda. eqlin — для линейных равенств;

o 1ambda. ineqnonlin — для нелинейных неравенств;

o 1ambda. eqnonlin — для нелинейных равенств;

• output — информация о результатах оптимизации:

o output. iterations — число выполненных итераций;

o output. funcCount — число вычислений функции;

o output. algorithm — используемый алгоритм;

o output. cgiterations — число PCG-итераций (только при использовании алгоритма большой размерности);

o output. stepsize — величина конечного шага поиска (только при использовании алгоритма средней размерности);

o output. firstorderopt — мера оптимальности первого порядка (норма вектора градиента в точке минимума) — только при использовании алгоритма большой размерности)

o LargeScale — может принимать значения '**off**' (по умолчанию) и '**on**'. В первом слу;чае используется алгоритм средней размерности, во втором — алгоритм большой размерности.

Следующие опции используются только при работе с алгоритмом средней размерности (описание **см.** выше):

o DerivativeCheck;

o DiffMaxChange;

o DiffMinChange;

o LineSearchType — задание вида алгоритма одномерной оптимизации.

Опции, используемые только в алгоритме большой размерности:

o Hessian — гессиан (в случае матрицы Гессе, задаваемой пользователем, — см. выше);

o HessPattern — задание гессиана как разреженной матрицы (это может привести к существенному ускорению поиска минимума);

o MaxPCGIter — максимальное число итераций PCG-алгоритма (preconditioned conjugate gradient, см. выше);

o PrecondBandWidth — верхняя величина начальных условий для PCG-алгоритма;

o TolPCG — допуск на завершение PCG-итераций;

o TypicalX — типовые величины x;

5) возвращаемая величина output в данном случае имеет дополнительные компоненты:

Пример. Пусть требуется найти минимум функции $f(x) = - x_1 x_2 x_3$ при начальном значении $x = [10; 10; 10]$ и при наличии ограничений

$$0 \le x_1 + 2 x_2 + 2 x_3 \le 72.$$

Решение. Вначале создадим m-файл, определяющий целевую функцию:

```
function f = myfun(x)
f =- x(l)*x(2)*x(3) ;
```

Затем запишем ограничения в виде неравенств:

$$- x_1 - 2 x_2 - 2 x_3 \le 0$$

$$x_1 + 2 x_2 + 2 x_3 \le 72,$$

или в матричной форме:

A x £ b,

где $A = \begin{bmatrix} -1 & -2 & -2 \\ 1 & 2 & 2 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 72 \end{bmatrix}$

Теперь нахождение решения представляется следующим образом:

```
» A = [-1 -2 -2; 1 2 2];
» b = [0; 72];
» x0 = [10; 10; 10];   % Стартовое значение
» [x, fval] = fmincon('myfun', x0, A, b)
x =
   24.0000
   12.0000
   12.0000
fval =
 -3.4560e+003
```

2. Функция linprog

Функция linprog обеспечивает решение задачи линейного программирования.

Функция записывается в виде

```
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
[x,fval] = linprog(...)
[x,fval,exitflag] = linprog(...)
[x,fval,exitflag,output] = linprog(...)
[x,fval,exitflag,output,lambda] = linprog(...)
```

Аргументы и возвращаемые величины здесь аналогичны рассмотренным ранее для функции fmincon одним исключением: здесь введен дополнительный аргумент f — вектор коэффициентов линейной целевой функции. Функция может использовать алгоритм большой размерности lipsol или алгоритм средней размерности (метод проекций).

Рассмотрим примеры задач оптимизации.

Пример 1. Требуется найти решение задачи, описывающейся соотношениями (задача линейного программирования)

$$f(x) = -5x_1 - 4x_2 - 6x_3,$$

$$x_1 - x_2 + x_3 \le 20,$$

$$3x_1 + 2x_2 + 4x_3 \le 42$$

$$3x_1 + 2x_2 \le 42$$

$$0 \le x_1,\ 0 \le x_2,\ 0 \le x_3.$$

Решение.

```
>> f = [ -5, -4, -6];  % Вектор коэффициентов линейной целевой функции
>> % Матрица коэффициентов ограничений – неравенств
>> A = [1 -1 1; 3 2 4; 3 2 0];
>> b = [20; 42; 30]; % Вектор ограничений – неравенств
>> % Задание нижних границ переменных (нулей)
>> lb = zeros(3,1);
>> % Поиск решения
>> [x, fval, exitflag, output, lambda] = linprog(f, A, b, [ ], [ ], lb)
Optimization terminated successfully
x =
    0.0000
   15.0000
    3.0000
fval =
   -78.0000
exitflag =
     1
```

```
output =
        iterations:  6
        cgiterations:  0
        algorithm:   'lipsol'
lambda =
       ineqlin:  [3x1 double]
       eqlin   [0xl double]
       upper  [3x1 double]
       lower  [3x1 double]
```

Из возвращенной информации, в частности, следует:

- что оптимальное решение x = [0.0000 15.0000 3.0000];
- минимальное значение целевой функции равно -78.0000;
- вычисления завершились нахождением решения (exitflag>0);
- всего было выполнено 6 итераций;
- был использован алгоритм lipsol

Пример 2. Цех малого предприятия должен изготовить 100 изделий трёх типов. Каждого изделия нужно сделать не менее 20 штук. На изделия уходит соответственно 4 , 3,4 и 2 кг металла при его общем запасе 340 кг, а также по 4,75 , 11 и 2 кг пластмассы при её общем запасе 700 кг. Сколько изделий каждого типа x_1 , x_2 и x_3 надо выпустить для получения максимального выпуска в денежном выражении, если цена изделия составляет по калькуляции 40, 30 и 20 грн ?

Данная задача сводится к вычислению максимума функции

$$f(x_1,x_2,x_3) = 40\,x_1 + 30\,x_2 + 20\,x_3$$

при наличии ограничений

$$x_1 \geq 20,\ \ x_2 \geq 20,\ x_3 \geq 20$$

$$4\,x_1 + 3{,}4\,x_2 + 2\,x_3 \leq 340$$

$$4{,}75\ \ x_1 + 11\,x_2 + 2\,x_3 \leq 700$$

$$x_1 + x_2 + x_3 = 100$$

Имеем задачу линейного программирования.

```
f=[40;30;20];% Вектор коэффициентов линейной целевой функции
% Матрица коэффициентов ограничений – неравенств
A=[4 3.4 2; 4.75 11 2];
b=[340;700]; %вектор ограничений – неравенств
Aeq=[1 1 1]; %матрица коэффициентов ограничений – равенств
beq=[100]; %вектор ограничений – равенств
% Задание нижних границ переменных (нулей)
lb=[20;20;20];
% Поиск решений, определение максимум целевой функции
[x,fval,exitflag,output]=linprog(-f,A,b,Aeq,beq,lb)
Optimization terminated successfully
x =
  56.0000
```

```
   20.0000
   24.0000
fval =
 -3.3200e+003
exitflag =
     1
output =
     iterations: 5
     cgiterations: 0
     algorithm: 'lipsol'
```

Замечание. Максимизация с помощью функций fminbnd, fminunc, fminsearch, linprog, fmincon, fgoalattain, fminimax, lsqcurvefit и lsqnonlin достигается путём замены целевой функции **f(x)** на **- f(x)** (т. е. изменить знак). Аналогично в задачах квадратичного прогаммирования: там необходимо заменить матрицу **H** и вектор **f** соответственно на **- H** и **-f** .

Пример 3. Оптимальный наклон плоского солнечного коллектора [9].

Расчёт интенсивности солнечной радиации проводится по следующей формуле [Даффи, Бекма]

$$S = (H_b R_b + H_d)\cdot(\tau\alpha) , \qquad (1)$$

где H_b , H_d – прямая и рассеянная составляющие солнечной радиации на горизонтальной поверхности, Вт ч / м^2 ;

$(t\,a)$ - приведенная поглощательная способность

$$(\tau\alpha) = \frac{\tau\alpha}{1-(1-\alpha)\rho_d} , \qquad (2)$$

ρ_d - диффузная отражательная способность. Для системы покрытий из одного, двух, трёх и четырёх листов стекла ρ_d приблизительно равна 0,16 , 0,24, 0,29, 0,32 соответственно;

α - направленная поглощательная способность поглощающей пластины α = 0,94 ÷ 0,95 ;

τ - пропускательная способность

$$\tau = \tau_r\, \tau_a \qquad (3)$$

τ_r - пропускательная способность без учёта поглощения

Для одного покрытия
$$\tau_r = \frac{1-\rho}{1+\rho} \qquad (4)$$

$$\rho = \frac{1}{2}\left[\frac{sin^2(\theta_2-\theta_1)}{sin^2(\theta_2+\theta_1)} + \frac{tg^2(\theta_2-\theta_1)}{tg^2(\theta_2+\theta_1)}\right] \qquad (5)$$

$$\frac{n_1}{n_2} = \frac{sin\,\theta_2}{sin\,\theta_1} \qquad (6)$$

n_1 , n_2 - показатели преломления;

q_1 , q_2 - углы падения и преломления (рис.1).

Если среда 1 воздух, то n_1 = 1 , а показатель преломления стекла n_2 = 1,526 .

τ_a - пропускательная способность, учитывающая только поглощение;

$$\tau_a = e^{-K\,L} \qquad (7)$$

L - толщина стекла, см;
К - коэффициент ослабления стекла, см $^{-1}$.

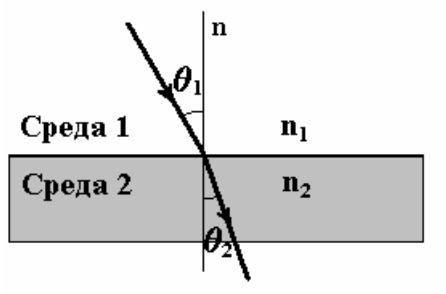


Рис. 1. Преломление луча на границе двух сред

Значение К изменяется от ~ 0,04 см $^{-1}$ для высокопрозрачного стекла (с бесцветной кромкой) до ~ 0,32 см $^{-1}$ для плохого (с зеленоватым оттенком кромки) стекла. Для обычного оконного стекла L = 0,32 см, К = 0,161 см $^{-1}$.

Подставляя соответствующие значения α = 0,94 , ρ_d = 0,16 (для одиночного покрытия), τ_a = 0,9498 в (3) и (2), для приведенной поглощательной способности получим

$$(ta) = 0{,}9014\,\frac{1-\rho}{1+\rho}. \qquad (8)$$

Величина R_b в выражении (1) определяется следующим соотношением

$$R_b = \frac{\cos\theta_1}{\cos\theta_z} = \frac{\cos(\varphi-\beta)\cos\delta\cos\omega + \sin(\varphi-\beta)\sin\delta}{\cos\varphi\cos\delta\cos\omega + \sin\varphi\sin\delta}, \qquad (9)$$

здесь θ_1 , θ_z - углы падения соответственно для наклонной и горизонтальной поверхностей;

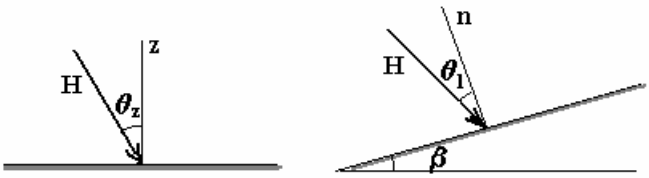


Рис. 2. Ход лучей для горизонтальной и наклонной поверхностей.

φ - широта местности (для Одессы φ = 46,5^0);
β - угол между рассматриваемой плоскостью и горизонтальной поверхностью (наклон коллектора);
δ - склонение (угловое положение Солнца);

$$\delta = 23{,}45^0 \sin\left(360\,\frac{284+n}{365}\right) \qquad (10)$$

n - порядковый номер дня года;
ω - часовой угол;

$$\omega = 180^0 - 15^0\tau \qquad (11)$$

τ - время суток в часах.

Таким образом, интенсивность солнечной радиации, определяемая формулой (1) с учётом зависимостей (5), (6), (8) – (11) представляет собою функцию времени года ***n***, времени суток t и угла наклона b солнечного коллектора

$$S = S(n,\tau,\beta)$$

При расчёте оптимального значения угла наклона b рассматривается усреднённое значение по времени интенсивности солнечной радиации за период использования системы горячего водоснабжения (в течение года апрель – октябрь и в течение суток с 6 часов утра до 19 часов вечера).

$$S_{cp}(\beta) = \frac{1}{7 \cdot 14} \sum_{i=1}^{7} \sum_{k=1}^{14} S(i,k,\beta) \ , \qquad (12)$$

здесь i – месяцы года (апрель – октябрь); k – время суток (от 6 до 19 часов).

Для условий города Одессы (φ = 46,5 градусов северной широты) прямая H_b и рассеянная H_d составляющие солнечной радиации на горизонтальной поверхности определялись по метеорологическим данным. Результаты расчётов средних значений солнечной инсоляции (12) в зависимости от наклона β коллектора представлены на рис. 3

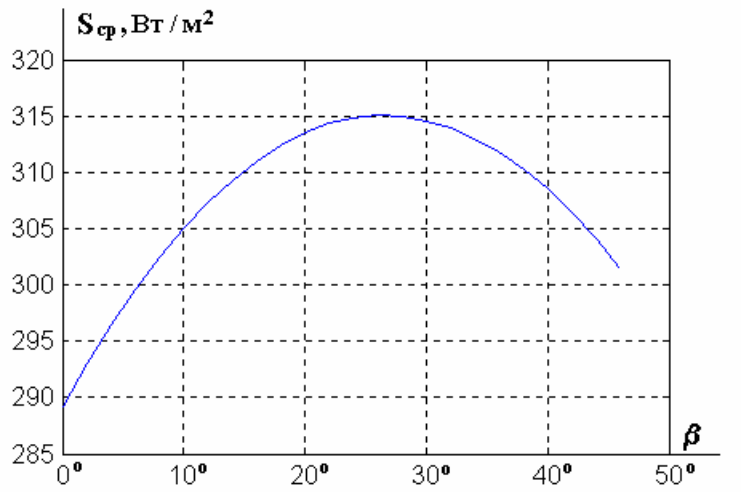


Рис. 3. Зависимость средней интенсивности солнечной радиации от угла наклона приёмной пластины коллектора $S_{cp}(\beta)$

Функция $S_{cp} = S(\beta)$ представляется следующей зависимостью

$$S = \exp((a+cx+ex^2)/(1+bx+dx^2+fx^3)).$$

По данной зависимости находим значение угла наклона солнечного коллектора β, при котором средняя солнечная инсоляция имеет наибольшее значение, используя функцию **fminbnd**.

Вначале создаётся m – файл:

```
%Вычисление максимального угла наклона солнечного коллектора
%Суммарная инсоляция
function S=maxins(x)
```

```
a=5.6667347;b=0.0018194204;c=0.017329411;
d=-0.00014610493;e=-0.0009964058;f=-1.3104374e-07;
m=a+c*x+e*x.^2;
n=1+b*x+d*x.^2+f*x.^3;
S=exp(n/m);
```

Теперь определяются значения угла:

```
>> x=fminbnd('maxins',0,48)

x =
   27.5103
```

Для данного угла наклона $\beta^* = 27{,}5^0$ интенсивность солнечной радиации для плоскости приёмной пластины коллектора за весь период эксплуатации системы горячего водоснабжения (апрель – октябрь с 6 утра до 19 часов вечера) имеет наибольшее значение.

Пример 4. Расчёт оптимальной конструкции противоточной вентиляторной градирни [10].

Выбор рациональной конструкции градирни и режимам её работы как тепломассообменного аппарата должен опираться на достаточно большом числе вариантов принятия решений. Эта задача не однозначна и её решение зависит от целевой оптимизации. Однако в большинстве известных задач оптимизации тепломассообменных аппаратов предпочтение отдаётся экономическим критериям, как наиболее объективным.

В общем случае целевая функция оптимальности **П** тепломассообменного аппарата имеет вид

$$\text{Ï}\ (x, y,..., k) = n_F F(x, y,..., k) + n_{Ni} N_H (x, y,..., k) + n_{Nâ} N_B (x, y,..., k) +$$
$$+ S_{Tn}\tau + S_{Ti}\,\tau_â G_H (x, y,...k) + S_{Ti}\,\tau_â G_B (x, y,...k)$$

здесь площадь тепломассообменной поверхности $F(x,y,\ldots,k)$, мощности $N_H\ (x,y,\ldots,k)$, $N_B\ (x,y,\ldots,k)$ насоса и вентилятора, $G_H(x,y,\ldots,k)$, $G_B(x,y,\ldots,k)$ – массовые потери теплоносителей зависят от ряда независимых величин $x,y,\ldots,k$.

Для определения эксплуатационных и конструктивных параметров градирни экономическая эффективность может быть записана в виде

$$П = Э + \frac{k}{\tau_{но}} = n_F F + n_{N_н} N_H + n_{N_в} N_B + G_H S_{Tн}\tau_г = 0{,}35\, K_{ап}\, n\, L\, H\, d\, r_{Al} / 1000 +$$
$$+ N_в\,[\,0{,}35\ a_в N_в^{b_в} + S_э\ \tau_г\,] + N_н\,[\,0{,}35\,(c_н + a_{1н} N_н^{b_н}) + S_э\ \tau_г\,] + 0{,}05\, G_ж\, S_{Tн}\ \tau_г\,, \quad (1)$$

где Э – годовые эксплуатационные расходы, грн / год; $\frac{k}{\tau_{но}}$ – капиталовложения, приходящиеся на один год нормативного срока окупаемости $\tau_{но}$, грн / год;

Минимизацию целевой функции П следует проводить при ограничении, устанавливающем зависимость температуры жидкости на выходе из аппарата от независимых переменных задачи

$$t_{вых} = t\,(\ n\,L\,,\ H\,,\ G_г^m\,).$$

В работе [ ] температура жидкости на выходе из аппарата представляется следующим образом:

$$t\,(\,x\,) = t_{вых} = C_1\, exp(l_1 H) + C_2\, exp(l_2 H) + t_* \ . \qquad (2)$$

Данная задача оптимизации П при ограничении (2), как задача нелинейного программирования, решается с помощью функции **fmincon**:

формирование целевой функции:

```
function f=zfun1(x)
%задание целевой функции
%стоимость градирни за год эксплуатации
% x(1)- nL (произведение количества листов и длины насадки),м
% x(2) - H (высота насадки),м, x(3)- Gmg (массовый расход воздуха),кг/ч
Se=0.15;      %стоимость электроэнергии, грн/(кВт ч)
Sv=1.0;       %стоимость воды, грн/м3
tg=7000;      %годовое число часов работы оборудования
tno=5;        %нормативный срок окупаемости, 5 лет
del=0.0002;  %толщина алюминиевого листа насадки - 0,2 мм
ro=2710;        %плотность алюминия, кг/м3
Kcto=23800;   %цена единицы массы аппарата, грн/т
Gl=200;                  %объемный расход жидкости, м3/ч
rol=998.2;               %плотность жидкости, кг/м3
Gml=Gl*rol;              %массовый расход жидкости, кг/ч
qul=4.5;                 %плотность орошения, кг/(м2 с)
rog=1.165;               %плотность газа, кг/м3
nug=16e-6;               %кинематическая вязкость газа, м2/с
Sv=Gml/3600/qul;      %площадь горизонтального сечения насадки, м2
Vg=x(3)/3600/rog/Sv;  %скорость газа, м/с
b=Sv/x(1);               %расстояние между листами, м
def=2*b;                 %эквивалентный диаметр, м
Re=x(3)/1800/rog/nug/x(1);            %число Рейнольдса
lam=0.3164/Re^0.25;                   %коэффициент сопротивления
dp=lam*rog*Vg^2*x(2)/2/def;           %гидравлические потери, Па
P1=0.35*Kcto*ro*del*x(1)*x(2)/1000; %стоимость аппарата
av=599.426;              %коэффициенты, определяющие удельные капиталовложения
bv=0.3058;               % вентиляторы в зависимости от его мощности
ett=0.7;                 %к.п.д. вентилятора
Nv=x(3)*dp/3600/rog/ett/1000;         %мощность вентилятора
P3=Nv*(0.35*av*Nv^bv+Se*tg);          %стоимость вентилятора, монтажа и годовой эксплуатации
f=P1+P3;                              %целевая функция
```

формирование ограничения:

```
function [c, ceq]=confuneq2(x)
%нелинейное ограничение
%задана температура жидкости на выходе из аппарата
% x(1)- nL (произведение количества листов и длины насадки)
% x(2) - H (высота насадки), x(3)- Gmg (массовый расход воздуха)
%-------------------------------------------------------------------
%начальные условия
t0=35; tH=30;      %температура жидкости на входе и выходе из аппарата, 0С
tgH=28;            %температура воздуха на входе, 0С
pH=2645.3;         %парциальное давление пара на входе в аппарат, Па
mp=-4062.9;  np=276.1;  %линейная аппроксимация парциального давления насыщенного пара
%другие данные
Gl=200;            %объемный расход жидкости, м3/ч
rol=998.2;         %плотность жидкости, кг/м3
Gml=Gl*rol;        %массовый расход жидкости, кг/ч
qql=Gml/7200/x(1);    %удельный расход жидкости, приходящийся на ед. ширины листа насадки
qqg=x(3)/7200/x(1);   %удельный расход газа - полурасход между листами насадки
qul=4.5;           %плотность орошения, кг/(м2 с)
rog=1.165;         %плотность газа, кг/м3
nug=16e-6;         %кинематическая вязкость газа, м2/с
Sv=Gml/3600/qul;      %площадь горизонтального сечения насадки, м2
Vg=x(3)/3600/rog/Sv;  %скорость газа, м/с
b=Sv/x(1);            %расстояние между листами, м
def=2*b;              %эквивалентный диаметр, м
Re=4*qqg/rog/nug;     %число Рейнольдса
%коэффициенты уравнений ТМО
Nu=0.018*Re^0.8;      %число Нуссельта
a1=0.611e-5*Nu/def/qql;
b1=0.1013e-6*Nu/def/qql;
a2=0.2476e-4*Nu/def/qqg;
```

```
b2=0.2794e-4*Nu/def/qqg;
n1=(a1+b1*np-a2-b2)/2; k=sqrt(a2*b2-a1*b2-a2*b1*np);
lamb1=-n1+sqrt(n1^2-k^2);
lamb2=-n1-sqrt(n1^2-k^2);
d1=tgH+t0;d2=pH-mp-np*t0;
r1=a2*exp(lamb1*x(2))/(a2-lamb1)+1;r2=a2*exp(lamb2*x(2))/(a2-lamb2)+1;
s1=b2*np*exp(lamb1*x(2))/(b2-lamb1)-np;s2=b2*np*exp(lamb2*x(2))/(b2-lamb2)-np;
del=r1*s2-r2*s1;
del1=d1*s2-d2*r2;
del2=d2*r1-d1*s1;
C1=del1/del; C2=del2/del;
%нелинейное ограничение при заданной температуре воды на выходе из аппарата x=H
c=[];
ceq=tH-t0-C1*(exp(lamb1*x(2))-1)-C2*(exp(lamb2*x(2))-1);
```

определение оптимальных параметров:

```
x0=[80,0.1,30000];
options=optimset('LargeScale','off');
[x,fval,exitflag,output]=fmincon('zfun1',x0,[],[],[],[],[180,0.4,200000],[],'confuneq2')
```

Получены следующие значения эксплуатационной ($G_г^m$) и конструктивных (nL , H) переменных (таблица 1), соответствующих минимуму капиталовложений в аппарат и вентиляторы, отнесённых к году окупаемости. При оптимизации функции $П$ кроме (2) накладывалось также ограничение $H \le 4$ м..

Таблица 1

Тип противоточной градирни	Расход воды $G_ж$, м^3 / ч	nL , м	H , м	$G_г^m$, кг / ч	$V_г$,м / с	Годовая стоимость $П_1 + П_3$, грн / год
ГРН - 25	25	80	0,4	28486	3,92	184,5
ГРН - 50	50	100	0,4	59933	4,1	232,0
ГРН - 100	100	120	0,4	124670	4,3	277,5
ГРН - 150	150	140	0,4	189620	4,34	322,2
ГРН - 200	200	180	0,4	253270	4,35	414,4

5. Задачи линейного и квадратичного программирования [11]:

Номер варианта	Задачи	Номер варианта	Задачи
1	$x_1 - x_2 - x_3 \to \max$, $x_1 - x_2 - x_3 = 0$, $x_1 + 15x_2 + x_3 = 2$, $x_j \ge 0$ j = 1,2,3.	**16**	$6x_2 - x_1^2 - 3/2x_2^2 + 2x_1x_2 \to \max$, $3x_1 + 4x_2 \le 12$, $-x_1 + x_2 \le 2$ $x_1 \ge 0$, $x_2 \ge 0$..
2	$6x_1 + 3x_2 + 15x_3 + 24x_4 \to \min$, $5x_1 + 4x_2 + 3x_3 - x_4 \le 10$, $2x_1 + x_2 + 4x_3 + 8x_4 = 3$, $x_j \ge 0$ j = 1,2,3,4.	**17**	$6x_2 - x_1^2 - 3/2x_2^2 + 2x_1x_2 \to \max$, $-x_1 + 2x_2 \le 2$, $x_1 \le 4$, $x_1 \ge 0$, $x_2 \ge 0$.
3	$-3x_1 - 10x_2 - 8x_3 + 6x_4 \to \max$, $3x_1 + 2x_2 + x_3 - x_4 \ge -1$, $x_1 + 3x_2 + 2x_3 - 2x_4 = -1$, $x_j \ge 0$ j = 1,2,3,4.	**18**	$6x_2 - x_1^2 - 3/2x_2^2 + 2x_1x_2 \to \max$, $3x_1 + 4x_2 \le 12$, $-x_1 - 2x_2 \le -2$, $x_1 \ge 0$, $x_2 \ge 0$.

4	$x_1 + x_2 + 4x_3 \to \max,$ $x_1 + x_2 + 4x_3 + 2x_4 = 2,$ $x_1 - x_2 - 2x_3 = 0,$ $x_j \geq 0 \;\; j = 1,2,3,4.$	**19**	$8x_1 + 12x_2 - x_1^2 - 3/2\,x_2^2 \to \max,$ $-2x_1 - x_2 \leq -4,$ $2x_1 + 5x_2 \leq 10,$ $x_1 \geq 0,\; x_2 \geq 0.$
5	$x_1 + x_2 + 2x_4 \to \max,$ $x_1 + x_3 + x_4 = 4,$ $-x_1 - 2x_2 - 3x_3 + x_4 = 0,$ $x_j \geq 0 \;\; j = 1,2,3,4.$	**20**	$8x_1 + 12x_2 - x_1^2 - 3/2\,x_2^2 \to \max,$ $-x_1 + 2x_2 \leq 2,\;\; x_1 \leq 6,$ $x_1 \geq 0,\; x_2 \geq 0.$
6	$2x_1 - 11x_2 + x_3 \to \max,$ $x_1 - x_2 + x_3 - x_4 - 2x_5 \leq 10,$ $2x_1 - 2x_2 + x_3 + 3x_4 + x_5 = 7,$ $x_j \geq 0 \;\; j = 2,4,5.$	**21**	$8x_1 + 12x_2 - x_1^2 - 3/2\,x_2^2 \to \max,$ $-3x_1 + 2x_2 \leq 0,$ $4x_1 + 3x_2 \leq 12,$ $x_1 \geq 0,\; x_2 \geq 0.$
7	$8x_1 + 4x_2 - 5x_3 - 4x_4 + 4x_5 \to \max,$ $-x_1 + 2x_2 - x_3 - x_4 + x_5 \geq 0,$ $2x_1 + x_2 - x_3 - x_4 + x_5 = 3,$ $x_j \geq 0 \;\; j = 2,3.$	**22**	$3x_1 - 2x_2 - 1/2\,x_1^2 - x_2^2 + x_1x_2 \to \max,$ $-2x_1 - x_2 \leq -2,$ $2x_1 + 3x_2 \leq 6,$ $x_1 \geq 0,\; x_2 \geq 0.$
8	$13x_1 - 9x_2 + 2x_3 + 3x_4 - 11x_5 \to \max,$ $x_1 - x_2 + x_4 - 3x_5 \leq 2,$ $4x_1 + x_3 - x_4 + 2x_5 = 8,$ $x_j \geq 0 \;\; j = 1.$	**23**	$6x_1 + 4x_2 - x_1^2 - 1/2\,x_2^2 - x_1x_2 \to \max,$ $x_1 + 2x_2 \leq 2,$ $-2x_1 + x_2 \leq 0,$ $x_1 \geq 0,\; x_2 \geq 0.$
9	$8x_1 + 5x_2 + x_3 - 5x_4 + 7x_5 \to \max,$ $-6x_1 - 3x_2 - 2x_3 + 3x_4 - 4x_5 \geq 5,$ $4x_1 + 2x_2 + x_3 - 2x_4 + 3x_5 = -3,$ $x_j \geq 0 \;\; j = 1,2,5.$	**24**	$6x_1 + 4x_2 - x_1^2 - 1/2\,x_2^2 - x_1x_2 \to \max,$ $2x_1 + x_2 \leq 2,\;\; x_2 \leq 1,$ $x_1 \geq 0,\; x_2 \geq 0.$
10	$-x_1 - x_2 - x_3 + 30x_4 + 8x_5 - 56x_6 \to \min,$ $2x_1 + 3x_2 - x_3 - x_4 + x_5 - 10x_6 \leq 20,$ $x_1 + 2x_2 + 3x_3 - x_4 - x_5 + 7x_6 = -11,$ $x_j \geq 0 \;\; j = 1,2,3,4,5.$	**25**	$6x_1 + 4x_2 - x_1^2 - 1/2\,x_2^2 - x_1x_2 \to \max,$ $3x_1 + 2x_2 \leq 6,$ $-3x_1 - x_2 \leq -3,$ $x_1 \geq 0,\; x_2 \geq 0.$
11	$-20x_1 + x_2 - 18x_3 - 9x_4 + 9x_5 -$ $-81x_6 \to \max,$ $3x_1 + 2x_2 - x_3 + 2x_4 + 7x_5 - x_6 \leq 1,$ $2x_1 - 3x_2 + 2x_3 + x_4 - x_5 + 9x_6 = 41,$ $x_j \geq 0 \;\; j = 1,2,3,4,5,6.$	**26**	$8x_1 + 6x_2 - 2x_1^2 - x_2^2 \to \max,$ $-x_1 + x_2 \leq 1,$ $3x_1 + 2x_2 \leq 6,$ $x_1 \geq 0,\; x_2 \geq 0.$
12	$21x_1 - 9x_2 - 7x_3 - 14x_4 +$ $+ x_5 - 7x_6 \to \max,$ $x_1 + x_2 + 2x_3 + 3x_4 - x_5 - 2x_6 \leq 21,$ $3x_1 - x_2 - x_3 - 2x_4 + 4x_5 - x_6 = 0,$ $x_j \geq 0 \;\; j = 1,2,3,4,5.$	**27**	$8x_1 + 6x_2 - 2x_1^2 - x_2^2 \to \max,$ $-x_1 + x_2 \leq 1,\;\; x_1 \leq 3,$ $x_1 \geq 0,\; x_2 \geq 0.$
13	$2x_1 - 2x_2 - 4x_3 - 6x_4 + 3x_5 +$ $+ 2x_6 \to \max,$ $-x_1 - 2x_2 - 3x_3 + 2x_4 + x_5 + 2x_6 \geq -15,$ $2x_1 - x_2 - 2x_3 - 3x_4 + 2x_5 + x_6 = -5,$ $x_j \geq 0 \;\; j = 1,2,3,4,5,6.$	**28**	$8x_1 + 6x_2 - 2x_1^2 - x_2^2 \to \max,$ $-x_1 + x_2 \leq 2,$ $3x_1 + 4x_2 \leq 12,$ $x_1 \geq 0,\; x_2 \geq 0.$

14	$x_1 + x_2 + x_3 + x_4 \to \max,$ $x_1 + x_2 + x_3 + 3x_4 = 3,$ $x_1 + x_2 - x_3 + x_4 = 1,$ $x_1 - x_2 + x_3 + x_4 = 1,$ $x_j \geq 0 \;\; j = 1,2,3,4.$	**29**	$2x_1 + 2x_2 - x_1^2 - 2x_2^2 + 2x_1x_2 \to \max,$ $4x_1 + 3x_2 \leq 12, \quad x_2 \leq 3,$ $x_1 \geq 0, \; x_2 \geq 0.$
15	$x_1 + 10x_2 - x_3 + 5x_4 \to \max,$ $x_1 + 2x_2 - x_3 - x_4 = 2,$ $-x_1 + 2x_2 + 3x_3 + x_4 = 2,$ $x_1 + 5x_2 + x_3 - x_4 = 5,$ $x_j \geq 0 \;\; j = 1,2,3,4.$	**30**	$2x_1 + 2x_2 - x_1^2 - 2x_2^2 + 2x_1x_2 \to \max,$ $2x_1 + x_2 \leq 4,$ $-x_1 + x_2 \leq 2,$ $x_1 \geq 0, \; x_2 \geq 0.$

4.3. Simulink – система визуального моделирования динамических систем

Назначение пакета Simulink [12]

Пакет Simulink служит для блочного моделирования различных систем и устройств. Для построения функциональной блок-схемы моделируемых устройств Simulink имеет обширную **библиотеку** блочных компонентов и удобный **редактор блок-схем.** Последний основан на использовании возможностей графического интерфейса пользователя и по существу является типичным средством **визуального программирования.** Используя **палитры компонентов** (наборы) блок-схем, пользователь с помощью мышки переносит нужные компоненты с палитр на рабочий стол пакета Simulink и соединяет линиями входы и выходы блоков. Таким образом, готовится блок-схема моделирования нужной системы или устройства.

Но этим уникальность пакета не исчерпывается, так как Simulink автоматизирует следующий наиболее трудоемкий этап моделирования: он автоматически составляет и решает сложные системы алгебраических и дифференциальных уравнений, описывающих заданную функциональную схему (модель), обеспечивая удобный и наглядный визуальный контроль за поведением созданного пользователем **виртуального устройства.** Вам достаточно уточнить (если нужно) вид анализа и пустить Simulink в режим **симуляции** (откуда и название пакета – Simulink) созданной модели системы или устройства. Изучение работы устройств и систем по моделям принято называть **моделированием.** Средства

визуализации результатов моделирования в пакете Simulink настолько наглядны, что порою создаётся ощущение, что созданная в виде блок-схемы модель и впрямь работает как "живая". Более того, Simulink почти мгновенно меняет математическое описание модели по мере ввода ее новых блоков даже в том случае, когда это изменяет порядок системы уравнений и ведет к существенному качественному изменению поведения системы. Впрочем, именно это и является одной из важнейших целей применения пакета Simulink.

Ценность Simulink заключается и в обширной открытой для изучения и модификации библиотеке компонентов. Она включает в себя источники сигналов с практически любыми временными зависимостями (от скачка до произвольной функции, задаваемой пользователем), масштабирующие, линейные и нелинейные преобразователи с разнообразными формами передаточных характеристик, квантующее устройство, интегрирующие и дифференцирующие блоки и так далее.

В библиотеке имеется и приличный набор регистрирующих устройств – от простых измерителей типа вольтметра или амперметра до универсальных осциллоскопов, позволяющих с удобством просматривать временные зависимости выходных параметров моделируемых систем, например, токов и напряжений, перемещений, давления и тому подобное. Есть даже графопостроитель для построения фигур в полярной системе координат, например, фигур Лиссажу или фазовых портретов колебаний. Simulink имеет обширные средства анимации и даже звукового сопровождения. А в дополнительных библиотеках можно отыскать и такие "дорогие приборы", как анализаторы спектра сложных сигналов, многоканальные самописцы и средства анимации графиков.

Интеграция одной из самых быстрых математических систем MATLAB с удобным и мощным пакетом Simulink открывает новые возможности использования самых современных математических методов для решения задач динамического и ситуационного моделирования сложных систем и устройств. А средства анимации Simulink позволяют строить на его основе виртуальные физические лаборатории с очень наглядным представлением результатов модели-

рования. Возможности Simulink позволяют выполнять математическое моделирование сложных динамических систем в физике, электро- и радиотехнике, в биологии и химии, словом, во всех областях науки и техники. Это объясняет популярность данного пакета как в крупных университетах и институтах, так и научных лабораториях всего мира.

И, наконец, важным достоинством пакета является возможность задания в блоках произвольных математических выражений, что позволяет подчас решать типовые задачи, пользуясь примерами пакета Simulink и просто задавая новые выражения, описывающие работу моделируемых пользователем систем и устройств. Полезным свойством пакета является и возможность задания системных S-функций и полноценное их включение в состав библиотек Simulink. К этому надо добавить и возможность моделирования устройств и систем в реальном масштабе времени.

Как программное средство Simulink – типичный представитель визуально-ориентированного языка программирования. На всех этапах работы, особенно при подготовке схем моделей, пользователь вообще не имеет дела с программированием как таковым. Нужная программа автоматически генерируется по мере ввода выбранных блоков компонентов, их соединений друг с другом и по мере задания параметров компонентов.

Важное достоинство Simulink – интеграция как с самой системой MATLAB, так и рядом других пакетов расширения, что обеспечивает по существу неограниченные возможности в применении Simulink для решения практически любых задач имитационного моделирования, причем как простых, так и самых сложных. Совместно с Simulink используется ряд пакетов. Ниже приведены краткие сведения о наиболее важных из них:

Communications – пакет моделирования коммуникационных устройств (более 100 функций и 150 компонентов);

Signal Processing – пакет для проектирования сложных устройств обработки сигналов;

Real Time Workshop – пакет для моделирования в реальном масштабе времени, обеспечивающий создание кода на языке программирования С;

Stateflow – пакет для моделирования событийно-управляемых систем на базе теории конечных автоматов;

DSP BIockset – библиотека моделирования цифровых сигнальных процессоров *DSP (Digital Signal Processor) - цифровых фильтров.*

Интеграция пакета Simulink с системой MATLAB

После инсталляции Simulink (отдельно от MATLAB или в составе этой системы) он интегрируется с ней и не требует никаких дополнительных операций для работы. Внешне интеграция проявляется появлением кнопки New **Simulink Model** в конце панели инструментов (перед кнопкой справочной системы ?). Нажатие этой кнопки открывает окна редактора функциональных блок-схем и оглавления библиотеки компонентов - рисунке 4.11.

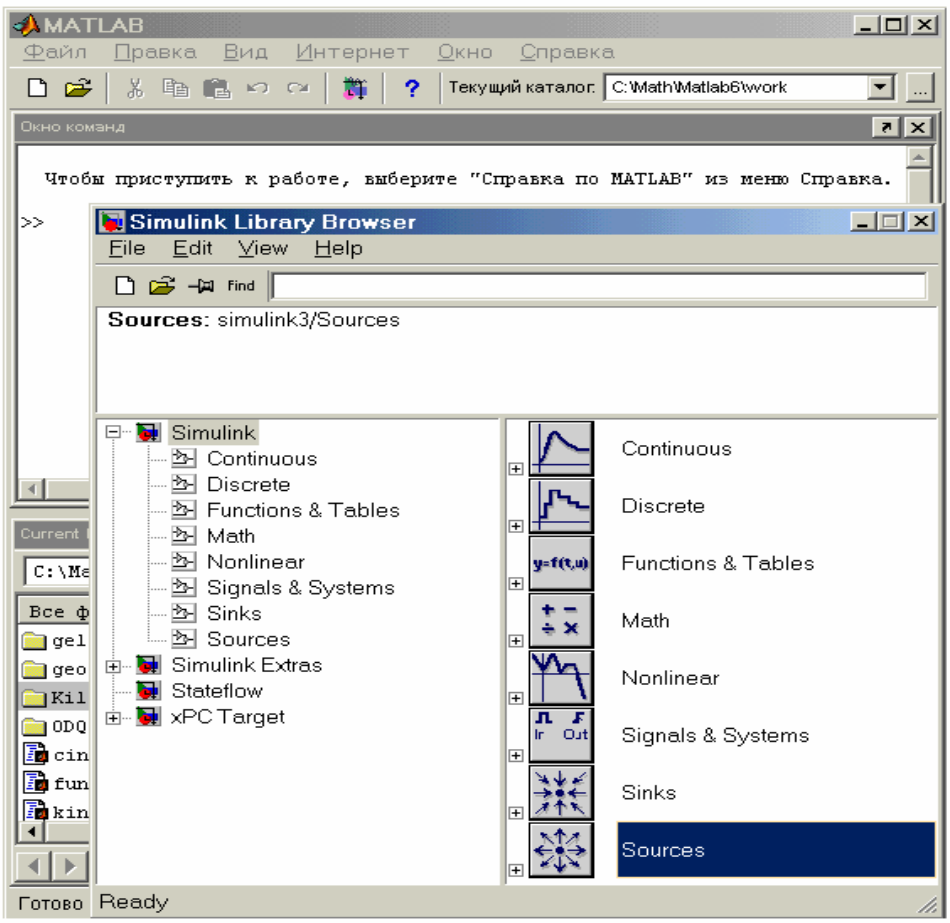


Рис. 4.11. Интеграция средств пакета Simulink с системой MATLAB

Для решения автоматически составленной системы нелинейных дифференциальных уравнений первого порядка (ODE) Simulink использует решатель дифференциальных уравнений, построенный в виде программного цифрового интегратора. Решатель работает в двух основных режимах:

Variable-step solvers – решение с переменным шагом. **Fixed-step solvers** - решение с фиксированным шагом. Как правило, лучшие результаты дает решение с переменным шагом (обычно по времени, но не обязательно). В этом случае шаг автоматически уменьшается, если скорость изменения результатов в процессе решения возрастает. И, напротив, если результаты меняются слабо, шаг решения автоматически увеличивается. Это исключает (опять-таки, как правило) накапливание ошибки, которое нередко случается при фиксированном шаге.

Параметры решателя устанавливаются с помощью окна, которое появляется при исполнении команды Parameters в позиции Simulation главного меню пакета Simulink. В нем же можно установить конкретный метод решения дифференциальных уравнений: ode45, ode23, rk45 (метод Рунге-Кутта), odel13 (метод Адамса), ode15s и odel (метод Эйлера). Своими возможностями пакет Simulink во многом обязан специальному аппарату создания и применения так называемых системных **S-функций** (System Functions). Эти функции позволяют в ходе решения осуществлять сложные функциональные преобразования по различным математическим алгоритмам, например алгоритмам решения систем дифференциальных уравнений. Для подготовки S-функций Simulink имеет специальный редактор. Создав свою S-функцию, пользователь фактически создает блок библиотеки, который может использоваться по всем правилам применения блоков. Это значит, что блок можно переносить в окно редактирования мышкой, менять заданную S-функцию, использовать необходимые связи между блоками и так далее.

Особенности интерфейса Simulink

Интерфейс новой версии Simulink 4 полностью соответствует стилю интерфейса типовых приложений под Windows 95/98, в том числе интерфейсу

системы MATLAB. В тоже время он концептуально строг, дабы не досаждать пользователю многочисленными "излишками" стандартного интерфейса Windows 95/98. Главное меню системы имеет следующие позиции:

File – работа с файлами моделей и библиотек (их создание, сохранение, считывание и печать).

Edit – операции редактирования, работа с буфером промежуточного хранения и создание субсистем.

View – вывод или удаление панелей инструментов и статуса.

Simulation – управление процессом моделирования (старт, ввод паузы и вывод окна настройки параметров симуляции).

Format – операции форматирования модели (смена фонтов, редактирование надписей, повороты блоков, использование тени от блоков операции с цветами линий блоков, их фоном и общим фоном).

Tools – управление видом анализа (в линейной области и в режиме реального времени RTW).

Первые три позиции главного меню содержат общепринятые для Windows-приложений команды и операции, так что мы не будем их обсуждать. Остальные позиции будут рассмотрены по мере знакомства с системой Simulink.

Как уже отмечалось, вместе с рабочим окном Simulink выводится окно с перечнем разделов основной библиотеки компонентов. Это окно – важная специфическая часть интерфейса Simulink. Оно открывает доступ к множеству других подобных по оформлению окон, дающих доступ к новым обширным пакетам компонентов (Blocksets&Toolboxes) и примеров их применения (Demos). Это позволяет пользователю постепенно знакомиться все с новыми и новыми областями применения Simulink.

Демонстрация возможностей Simulink

Даже не знакомый с Simulink пользователь может быстро оценить возможности этого пакета, воспользовавшись множеством интересных и поучи-

тельных примеров его применения. Они находятся в довольно далекой директории

MATLAB/TOOLBOX/SIMULINK/SIMDEMO10.

Для загрузки примеров используется (как обычно) команда **Open** в позиции **File** главного меню основного окна пакета Simulink. В уже знакомом вам окне, правда, с названием Open Model (Открыть Модель), надо войти в указанную директорию с примерами и выбрать подходящий пример.

Выбрав нужный пример, нужно нажать кнопку **Открыть** окна. Появится графическая модель моделируемого устройства. На рисунке 2.3 показана выбранная нами модель, позволяющая проанализировать поведение массивного кубика, который через пружину привязан к палочке (файл и пример **onecart).** Кубик перемещается по плоскости с трением, поэтому при возбуждении системы рывком палочки можно наблюдать характерные затухающие колебательные движения кубика. Таким образом, в этом примере решается типичная физическая задача на колебания реального механического маятника под действие внешней силы.

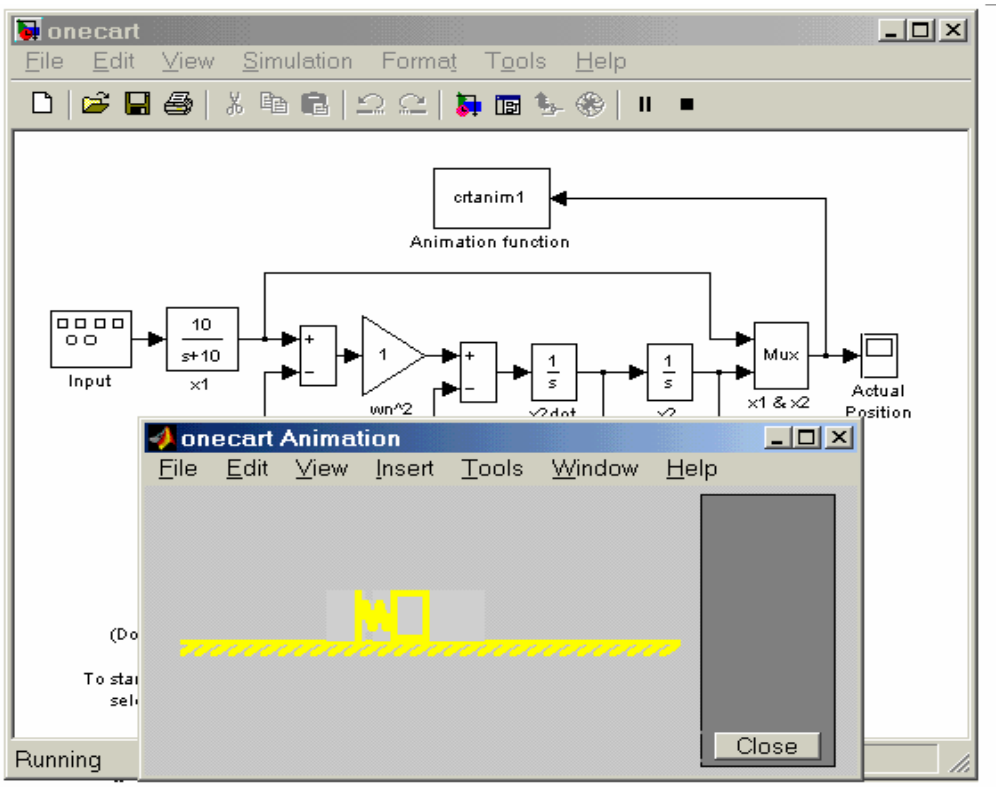


Рис. 4.12. Графическая модель колебательной системы – массивного кубика на пружине

Нетрудно заметить, что графическая модель этого примера содержит 11 блоков. Каждый блок имеет наглядное общепринятое обозначение в виде прямоугольника, треугольника и так далее. Блоки имеют входы и выходы и описываются различными математическими зависимостями. Блоки соединяются друг с другом линиями – стрелками, причем стрелка указывает направление от выходов одних блоков к входам других блоков. Имеются также текстовые комментарии и средства для вывода подсказок и открытия окон системы помощи.

В конце инструментальной панели Simulink находятся две важные кнопки управления симулятором. Одна – в виде черного треугольника (Start/Pause **Simulation)** запускает или приостанавливает начатый процесс моделирования", а другая – в виде черного квадратика (Stop) – останавливает его. Так что все, что надо для стимуляции выбранной модели, – это нажать кнопку с изображением треугольника. На рисунке 4.12 внизу показан результат моделирования для выбранной модели. Вместо кнопок можно использовать команды **Start и Pause** в позиции **Simulation** главного меню симулятора.

В данном случае результат моделирования отображается в виде анимационного видеоклипа (см. изображение физической модели кубика в окне анимации под графической моделью анализируемого физического устройства). Наглядность представления результатов поведения устройства в данном случае вполне очевидна.

Обычно Simulink запускается соответствующей кнопкой из панели инструментов, после чего все последующие действия выполняются в его Windows-среде. Для вывода полного перечня команд Simulink надо исполнить команду

» help simulink

Дополнительную информацию можно получить, используя команды

help blocks и **help simdemos**

Мы не будем рассматривать эти многочисленные команды, поскольку на практике отказ от работы с очень удобным и наглядным Windows-интерфейсом для пакета Simulink трудно чем-то оправдать. При работе с этим интерфейсом данные команды знать не нужно.

Обзор библиотеки компонентов пакета Simulink

Новая версия симулятора Simulink имеет существенно обновленную и пополненную библиотеку компонентов. Она размещается в директории MATLAB/TOOLBOX/SIMULINK/BLOCKS. Основная палитра компонентов представлена файлом simulink.mdi. Как основная, так и дополнительные библиотеки Simulink представлены файлами разного формата – с расширением dll, в виде m-файлов и файлов с расширением m. Последние могут при необходимости редактироваться и модифицироваться.

Окно основной палитры компонентов пакета Simulink появляется при запуске пакета. Оно было показано на рисунке 4.11. Каждый рисунок в данном случае носит обобщающий характер и представляет группу компонентов определенного класса. Можно считать, что эта группа рисунков представляет собой образное оглавление стандартной библиотеки графических элементов для набора функциональных схем.

Как видно на рисунке 4.11, в состав библиотеки графических элементов входят следующие их наборы-

Sources – открытие окна с перечнем источников сигналов и воздействий.

Sinks – открытие окна с перечнем регистрирующих компонентов.

Continuous – линейные компоненты.

Discrete – открытие окна с перечнем дискретных компонентов.

Math – математические компоненты.

Function&Tables – функции и таблицы.

Nonlinear – открытие окна с перечнем нелинейных компонентов.

Signals&Systems – сигналы и системы.

Окно библиотек является обычным окном Simulink и имеет соответствующее главное меню, панель инструментов и панель статуса. С его помощью можно вводить и загружать из файла модели симулируемых устройств и систем. Окно можно свернуть или закрыть. Если при закрытии окна основной библиотеки оно понадобилось вновь, то из основного окна библиотеки надо загрузить файл Simulink из директории MATLAB/TOOLBOX/ SIMULINK/BLOCK10.

Активизация каждого раздела библиотеки открывает окно с пиктограммами каждого компонента. На рисунке 4.13, к примеру, открыто окно источников воздействий Sources

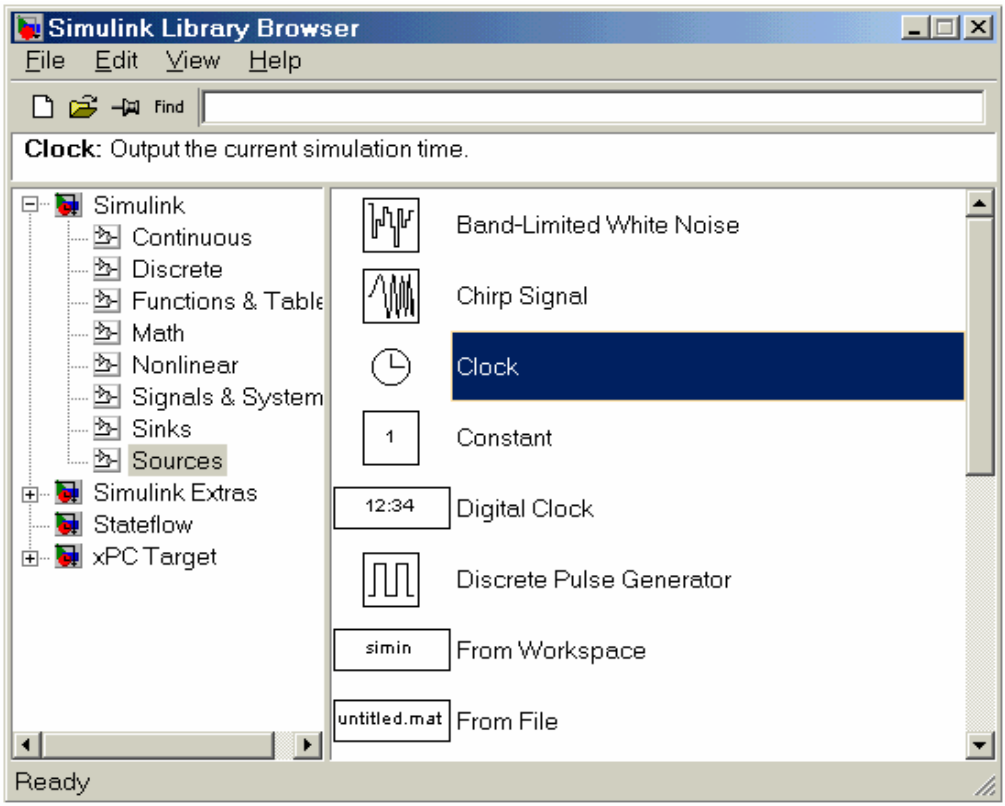


Рис. 4.13. Окно с перечнем источников сигналов

Графические элементы источников сигналов имеют настолько очевидные обозначения, что не имеет смысла приводить даже их названия (значительная часть фирменного электронного справочника по Simulink тем не менее посвящена подробному описанию компонентов и их параметрам). Названия блоков приведены в них и справа. Большинство элементов содержат рисунок, представляющий временную зависимость.

Набор блоков содержит практически все часто используемые при моделировании источники сигналов с самой различной функциональной и временной зависимостью. Возможно задание произвольного воздействия из файла – элемент From File. Есть и случайные воздействия для моделирования систем и устройств методом Монте-Карло.

С каждым графическим элементом связана панель его настроек. Для открытия этого окна достаточно выполнить двойной щелчок левой клавишей

мышки при ее маркере, установленном на нужное изображение элемента. Оно выделяется характерными маленькими кружками по углам.

При вызове окон параметров активизацией графических элементов в окнах библиотек отображаются установки параметров по умолчанию. Как правило, они нормализованы – например, задана единичная частота, единичная амплитуда, нулевая фаза и так далее. Возможность изменения параметров в этом случае отсутствует. Она появляется после переноса графических элементов в окно подготовки и редактирования функциональных схем. Как правило, установки параметров блоков по умолчанию позволяют уверенно начать моделирование и затем уточнить эти параметры.

Регистрирующие элементы – важное средство высококачественной визуализации результатов блочного моделирования. Некоторые регистраторы в Simulink выполнены в виде, весьма похожем на реальные приборы, что нередко создает ощущение практической работы симулируемых систем. В состав виртуальных регистраторов входят:

Scope - осциллограф для наблюдения временных и иных зависимостей.

XY Graph - графопостроитель в системе полярных координат.

Display - дисплей.

To File - устройство, описанное в файле.

To Workspace - устройство обзора рабочего пространства.

Stop Simulation - остановка симуляции.

Важно отметить, что виртуальные регистраторы фиксируют параметры любого типа, а не только электрические. Это придает некоторым виртуальным регистраторам (приборам) уникальный характер. Например, осциллоскоп, фиксирующий не только электрические сигналы, но и перемещения механических объектов, изменения температуры или давления и вообще изменения любых физических величин, можно создать только в памяти компьютера.

Дискретные компоненты включают в себя устройства задержки, дискретно-временной интегратор, дискретный фильтр и иные. Линейные компоненты играют важную роль в создании математических моделей многих устройств.

Имеются следующие типы линейных компонентов: Gain – аналоговый усилитель (масштабирующее устройство), Sum – аналоговый сумматор,

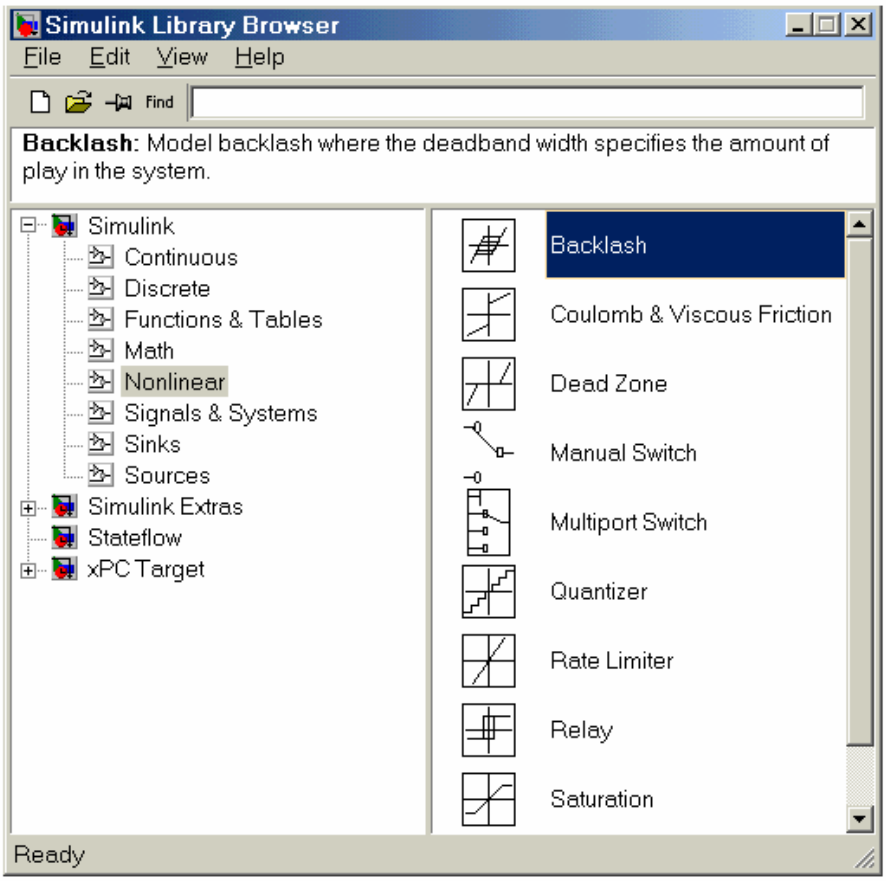


Рис. 4.14. Окно библиотеки с нелинейными компонентами

Integrator – аналоговый интегратор, Derivative – аналоговое дифференцирующее устройство и ряд других (в основном матричных) устройств.

Поскольку Simulink предназначен главным образом для моделирования нелинейных динамических систем, раздел, посвященный нелинейным компонентам, впечатляет уже одним только их обилием – см. рисунок 4.14. Окно с этими элементами настолько велико, что даже не поместилось на рисунке.

Среди нелинейных компонентов следует отметить компоненты с типичными нелинейностями, например вида abs(u), с характеристиками, описанными типовыми математическими функциями, компонентами вида идеальных и неидеальных ограничителей и так далее. Достойно представлены и такие сложные компоненты, как квантователи, блоки нелинейности, моделирующие нелинейные петли гистерезиса, временные задержки и ключи-переключатели. Естественно, что все нелинейные блоки имеют установку своих параметров.

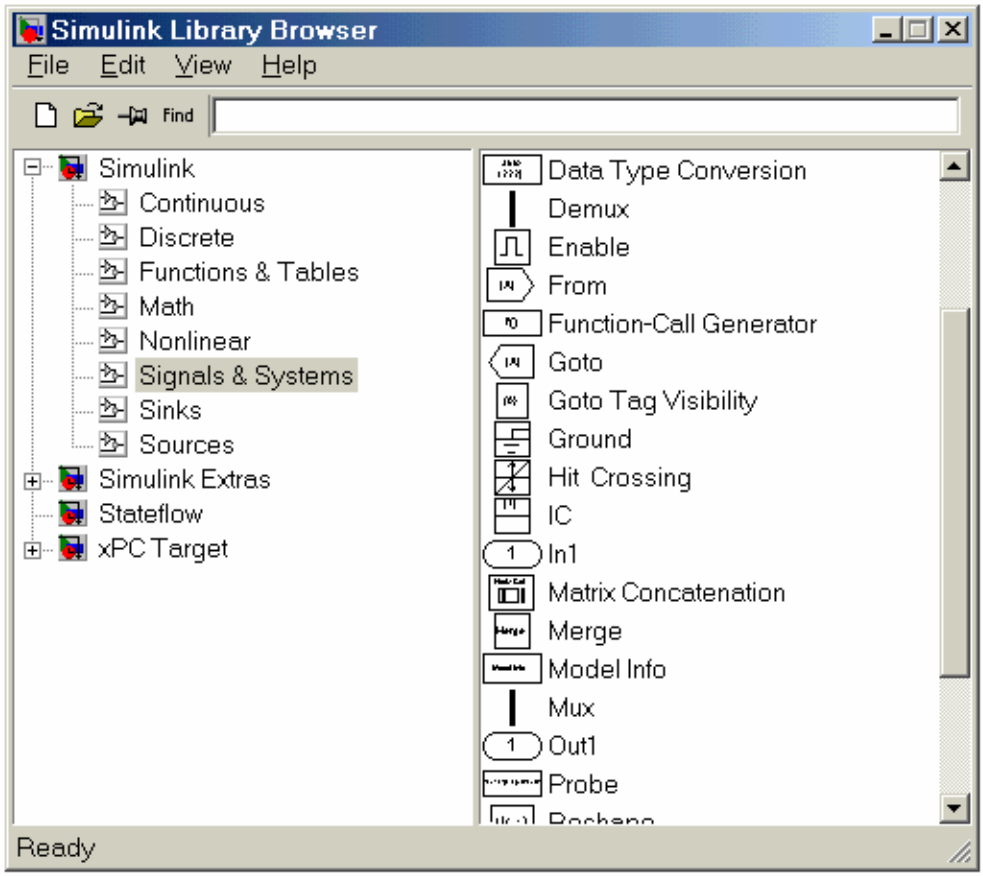


Рис. 4.15. Окно библиотеки Signal&Systems с подключающими компонентами

Окно библиотеки Signal&Systems с подключающими компонентами показано на рисунке 4.15. Оно также содержит впечатляющий выбор таких компонентов – от портов входа In, выхода Out и заземления Ground до компонент, имитирующих работу триггера Trig, и даже задания субсистем Subsystem.

Последняя компонента представляет собой пустое окно, в котором можно создать функциональную схему, рассматриваемую как подсистему (блок). Такая подсистема может многократно использоваться различными моделями. Каждый компонент, как и ранее, имеет окно установки своих параметров.

Представленные в этих разделах компоненты относятся к достаточно серьезным приложениям системы MATLAB, причем подчас вполне законченным.

Ряд примеров применения пакета Simulink представлен и в справочной системе MATLAB - Demo 10. Выход в нее возможен из окон прочих компонентов, в которых также есть разделы с именем Demo 10.

В директории MATLAB/TOOLBOX/SIMULINK/BLOCKS помимо файла основной библиотеки находится ряд файлов дополнительных библиотек.

В разделе MATLAB/TOOLBOX/SIMULINK/DEE находится редактор и решатель систем дифференциальных уравнений с примерами его применения.

Таким образом, в поставку Simulink 4 входит множество библиотечных компонентов, с лихвой удовлетворяющих большинство пользователей, всерьез занятых математическим моделированием различных систем и устройств. При этом есть возможность корректировать описания компонентов и вводить новые компоненты, в том числе в виде функциональных схем отдельных подсистем.

Пример подготовки и решения конкретной задачи

Решение любой задачи в системе Simulink должно начинаться с постановки задачи. Чем глубже продумана постановка задачи, тем больше вероятность успешного ее решения. В ходе постановки задачи нужно оценить, насколько решение задачи отвечает возможностям пакета Simulink и какие компоненты последнего могут потребоваться для решения заданной задачи.

В качестве примера рассмотрим тривиальную задачу моделирования работы линейного осциллятора. Осциллятором в механике называют механическую систему с одной степенью свободы, совершающую колебания около положения устойчивого равновесия (например, маятник, груз на пружине). Колебания осциллятора описываются следующим дифференциальным уравнением [8]

$$x'' + 2nx' + k^2 x = h\sin(pt), \qquad (1)$$

здесь x'' - обобщённое ускорение осциллятора ; $2nx'$ - обобщённая сила линейного сопротивления; $k^2 x$ - обобщённая квазиупругая сила; $h\sin(pt)$ – обобщённая гармоническая возмущающая сила.

Данное уравнение применяется также и для исследования колебаний немеханических систем. В частности, колебательный контур является осциллятором. Колебания напряжённостей электрического и магнитного полей в плоской электромагнитной волне вполне можно описывать предыдущим дифференциальным уравнением. В квантовой механике задача о линейном осцилляторе решается с помощью уравнения Шредингера. Законы колебаний линейного осциллятора играют важную роль в теории твёрдого тела, электромагнитного излучения, колебательных спектров молекул.

Запишем уравнение (1) в форме

$$x'' = -2nx' - k^2 x + h\sin(pt). \qquad (2)$$

В основу воплощения уравнения (1) в блок-схему положим следующую идею последовательного итерационного процесса:

- если сформировать правую часть уравнения (1) зависящую от x , x' , считая их известными, то станет известно ускорение x'' осциллятора. Проинтегрировав ускорение, можно получить скорость x' , а последующее интегрирование даёт закон изменения перемещения осциллятора $x(t)$. Полученные значения скорости и перемещения можно теперь использовать для формирования правой части.

Итак, для формирования блок-схемы, осуществляющей численное интегрирование, можно сделать следующее:

1. В основу блок-схемы положено два последовательно соединённых интегратора (блоки Integrator) с внешне задаваемыми начальными условиями в соответствии с системой дифференциальных уравнений

$$\begin{cases} \dfrac{dV}{dt} = h\sin(pt) - 2nx' - k^2 x \\ \dfrac{dx}{dt} = V \end{cases} \qquad (3)$$

и начальными условиями при $t = 0$ $x = x_0$, $x' = x'_0$.

На вход первого интегратора подаётся ускорение V' , а в качестве начального условия используется начальное значение скорости $V(0) = x'_0$; выходом этого блока будет текущая скорость осциллятора $V = V(t)$; эту величину следует подать на вход второго интегратора с начальным условием в виде начального значения отклонения $x(0) = x_0$; выход из блока будет представлять собой искомый закон движения $x = x(t)$.

2. Сформулировать отдельным блоком в виде подмодели (блок **Subsystem**) функцию правой части $f(t, x, x') = h \sin(pt) - 2nx' - k^2 x$, используя в качестве входных параметров полученные значения $x(t)$ и $x'(t)$ в качестве входных величин - h, p, n, k .

3. Используя созданную подмодель, в основной модели связать сформулированные текущие значения $x(t)$ и $x'(t)$ с соответствующими входами подмодели, а выход подмодели связать с сумматором; сигнал с выхода сумматора подать на вход первого интегратора, замыкая цепь интегрирования.

4. Для отображения результатов интегрирования в графической форме подсоединить блок **Scope** к выходу системы – сформированному сигналу $x(t)$; для отображения фазового портрета осциллятора использовать блок **XY Graph** , на входы которого направить сигналы $x(t)$ и $x'(t)$.

Блок-схема, реализующая указанные идеи приведена на рисунке 4.17.

Отметим также, что при построении блок-схемы подсистемы (рисунке 4.16) связь подсистемы с основной системой осуществляется путём ввода в подсистему стандартных блоков типа **In** (Вход) и **Out** (Выход). Все величины, которые формируются в основной модели, а затем должны быть использованы в подмодели, должны "проникать" в подсистему через блоки **In ,** а величины, сформированные в подсистеме и затем используемые в основной системе, должны "выходить" из подсистемы через блоки **Out** .

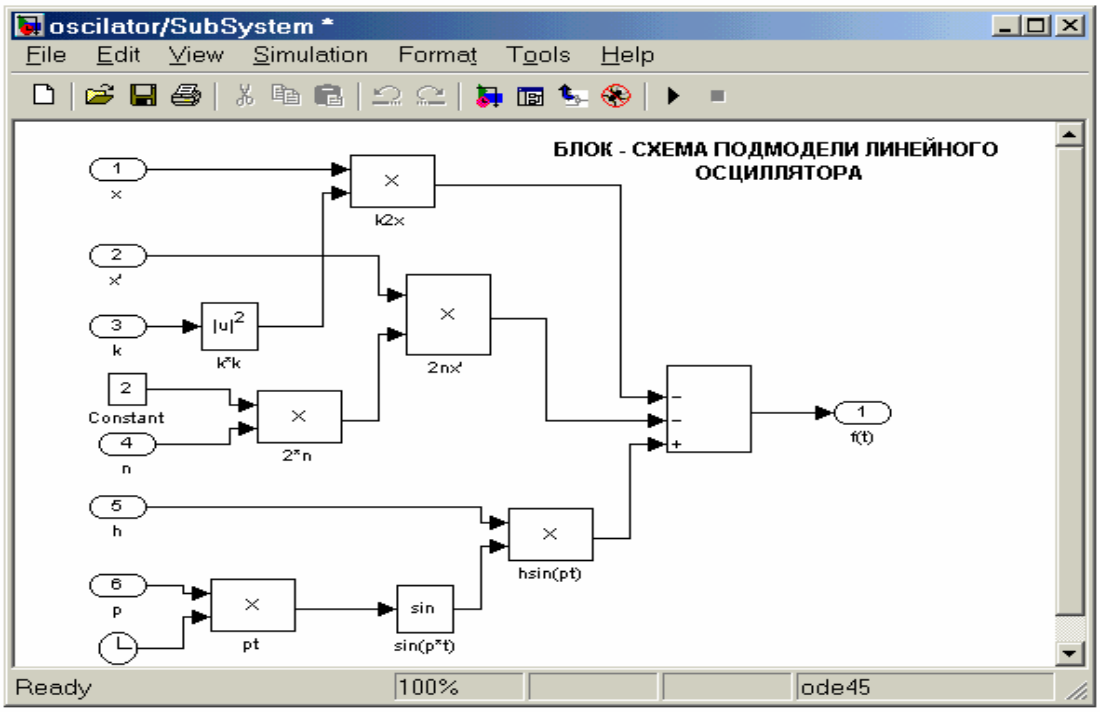


Рис. 4.16. Блок-схема подмодели линейного осциллятора.

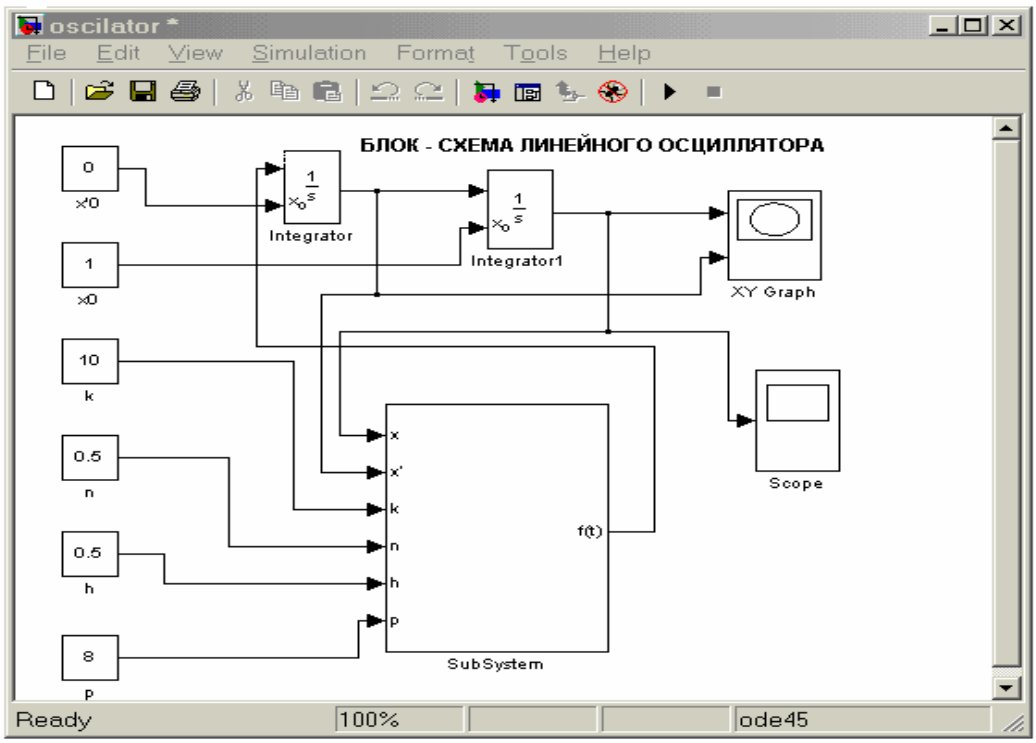


Рис. 4.17. Блок-схема основной модели линейного осциллятора

При этом на изображении блока подсистемы в блок-схеме основной модели автоматически появляется такое количество входов, которое совпадает с числом введенных в подсистеме блоков **In ,** и выходов, равных числу блоков **Out** , использованных в подсистеме.

Вызвав команду **Start** из меню **Simulation** окна основной блок-схемы, можно активизировать процесс моделирования созданной S – модели осциллятора. По окон-

чании этого процесса появится дополнительное графическое окно, созданное блоком **XY Graph** , на котором изображается фазовый портрет осциллятора при выбранных параметрах. Если теперь дважды щёлкнуть мышью на блоке **Scope** в блок-схеме осциллятора, то появится ещё одно графическое окно (рисунок 4.18) с графиком зависимости координаты x от времени.

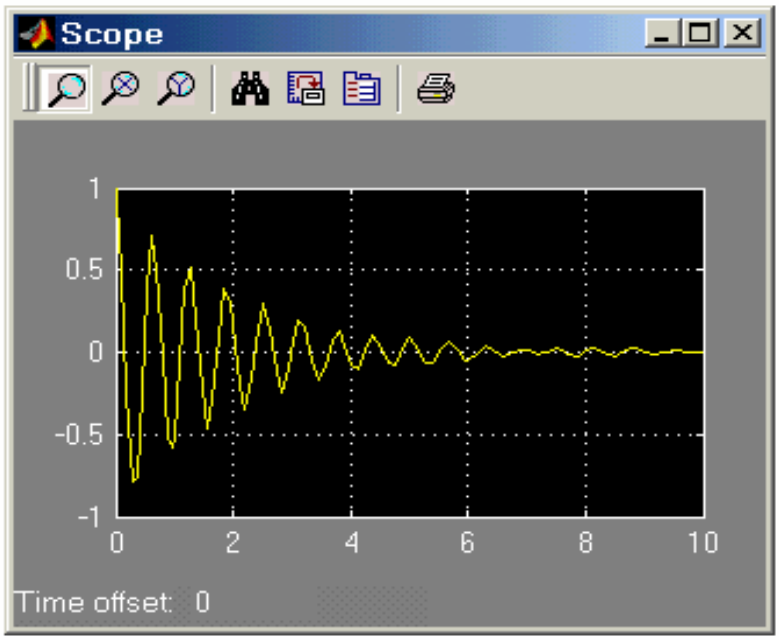


Рис. 4.18. Колебания линейного осциллятора

Изменяя данные настройки входных блоков **Constant,** можно проводить исследования поведения осциллятора при произвольных значениях входных параметров.

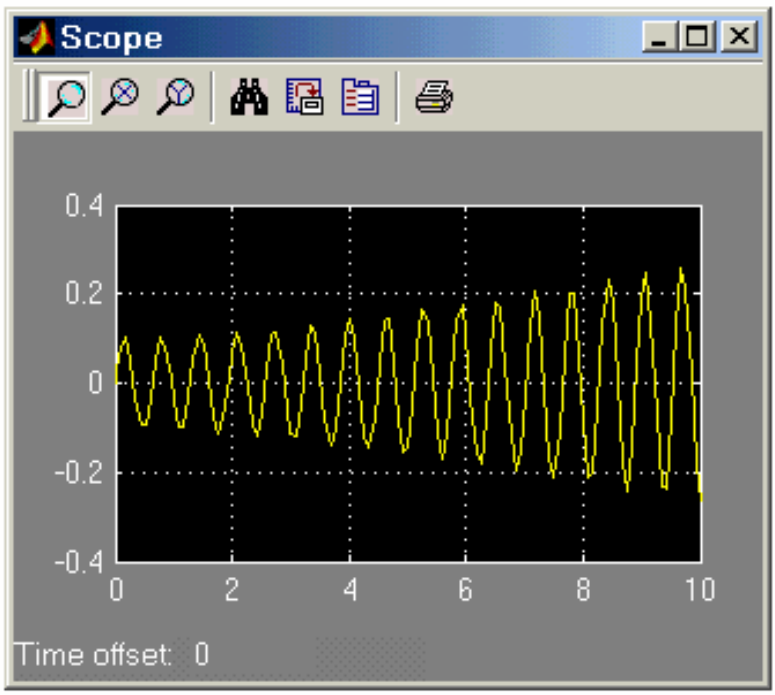


Рис. 4.19. Резонансные колебания осциллятора (при p = k)

В качестве заданий применения пакета расширений Simulink рассматривается задача об определении управляющих факторов, обеспечивающих программное движение манипулятора [7].

Манипулятор (рис. 4.24 – 4.26), состоящий из звеньев 1, 2 и захвата **D**, приводится в движение приводами A и B. Захват **D** перемещается вдоль прямой ON. Со стороны привода A к звену 1 прикладывается либо управляющий момент $\boldsymbol{M_A}$ (варианты 2, 4, 7, 8, 12, 22, 24-26, 29), либо управляющее усилие $\boldsymbol{P_A}$ (варианты 1, 3, 5, 6, 9-11, 13-21, 23, 27, 28, 30). Привод B воздействует на звено 2 либо моментом $\boldsymbol{M_B}$ (варианты 1-3, 5, 6, 8-11, 13-21, 23, 27), либо управляющим усилием $\boldsymbol{P_B}$ (варианты 4, 7, 12, 22, 24-26, 28-30).

Перемещение звена 1 (варианты 3, 4, 7, 12, 22, 24-26, 28-30) или звена 2 (варианты 1, 2, 5, 6, 8-11, 13 - 21, 23, 27) манипулятора ограничено препятствиями ***K*** и ***L***, поэтому изменение угла поворота $\varphi = \varphi(t)$ этого звена возможно лишь в интервале $[\varphi(0), \varphi(\tau)]$, где τ — время движения звена. Технические условия работы манипулятора требуют, чтобы указанное звено сошло со связи ***K*** при $t = 0$ и «мягко» коснулось препятствия ***L*** при $t = \tau$, т. е. так, чтобы были удовлетворены условия

$$\frac{d}{dt}\varphi(t)\big|_{t=0,t=\tau} = 0 \ ; \qquad \frac{d^2}{dt^2}\varphi(t)\big|_{t=0,t=\tau} = 0 \ .$$

Программные движения звена 1, удовлетворяющие требованиям «мягкого» касания, приняты в таком виде:

1) $\varphi(t) = \varphi(0) + [\varphi(\tau) - \varphi(0)]\,(10 - 15\,t/\tau + 6\,t^2/\tau^2)$ (варианты 2, 4, 6, 7, 11, 12, 16, 19, 22, 24-26, 28-30);

2) $\varphi(t) = \varphi(0) + [\varphi(\tau) - \varphi(0)]\,[t/\tau - (1/(2\pi))\sin(2\pi t/\tau)]$ (варианты 1, 3, 5, 8-10, 13-15, 17, 18, 20, 21, 23, 27).

Значения $\varphi(0)$ и $\varphi(\tau)$ заданы в таблице. Силами сопротивления движению пренебречь. Механизм расположен в горизонтальной плоскости. Движением захвата относительно звена 1 пренебречь.

В задании приняты следующие обозначения:

m_1 — масса первого звена, захвата и переносимого в захвате объекта;

ш$_2$ — масса второго звена;

J_1 - момент инерции звена 1, захвата и переносимого в захвате объекта относительно

главной центральной оси инерции;

J_2 - момент инерции звена 2.

Центр тяжести звена 1 находится в точке ***C*** (варианты 1- 4, 6-8, 11-13, 16, 18-20, 22-30) или в точке ***A*** (варианты 5, 9, 10, 14, 15, 17, 21).

Требуется:

1. Вычислить значения управляющих сил и моментов в начале торможения звена 1. Считать, что торможение звена 1 начинается в тот момент, когда угловое ускорение звена обращается в ноль.

2. Построить графики зависимости управляющих моментов и сил от времени.

Пример выполнения задания. Дано: m_1 = 2 кг; m_2 = 3 кг; J_1 = 0,8 кг м^2; l = 1 м; φ (0) = 0, φ (τ) = π / 6 рад; τ = 0,5 с.

Центр тяжести звена 1 находится в точке *C* (рис. 4.20):

$$\varphi(t) = \varphi(0) + [\varphi(\tau) - \varphi(0)]\,[t/\tau - (1/(2\pi))\sin(2\pi t/\tau)]$$

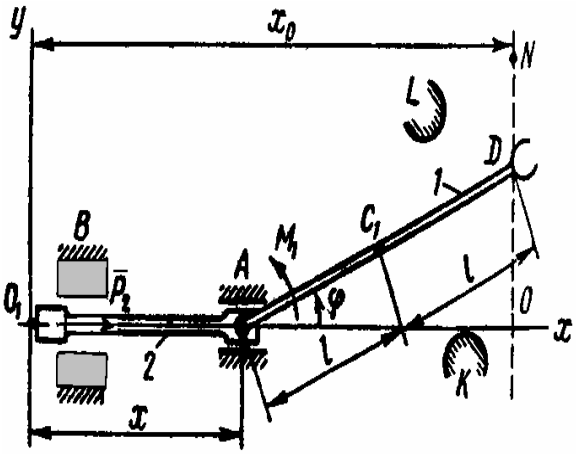


Рис. 4.20

Найти управляющий момент ***M*** и управляющую силу ***P***.

Р е ш е н и е. Для решения задачи применим уравнения Лагранжа II рода. Будем рассматривать механическую систему как систему с двумя степенями свободы, приняв за обобщенные координаты угол φ поворота звена 1 и смещение *x* звена 2 (рис.4.20)

Для рассматриваемой механической системы можно записать:

$$y_{C2} = \text{const}; \qquad x_{C1} = x + l\cos\varphi; \qquad y_{C1} = l\sin\varphi. \qquad (1)$$

Эти равенства играют роль уравнений связей.

В соответствии с выбранными обобщенными координатами имеем

$$\begin{cases} \dfrac{d}{dt}\left(\dfrac{\partial T}{\partial \dot{\varphi}}\right) - \dfrac{\partial T}{\partial \varphi} = Q_\varphi \\ \dfrac{d}{dt}\left(\dfrac{\partial T}{\partial \dot{x}}\right) - \dfrac{\partial T}{\partial x} = Q_x \end{cases} \qquad (2)$$

Совокупность уравнений (1) и (2) позволяет составить дифференциальные уравнения движения механической системы.

Составим выражение для кинетической энергии системы T как функцию обобщенных скоростей $\dot{\varphi}$ и $\dot{x}$ и обобщенных координат φ и x. Кинетическая энергия системы равна сумме кинетической энергии T_1 звена 1 и T_2 звена 2

$$T_1 = \frac{m_1 V_1^2}{2} + \frac{J_1 \omega_1^2}{2} .$$

Кинетическая энергия звена 1, совершающего плоское движение,

$$T_2 = \frac{m_2 V_2^2}{2}$$

где $\omega_1 = \dot{\varphi}$; $V_2 = \dot{x}$.

Продифференцировав (1) по времени, будем иметь

$$\dot{x}_{C1} = \dot{x} - l\dot{\varphi}\sin\varphi \; ; \qquad \dot{y}_{C1} = l\dot{\varphi}\cos\varphi ,$$

откуда $V_1^2 = \dot{x}_{C1}^2 + \dot{y}_{C1}^2 = \dot{x}^2 + l^2\dot{\varphi}^2 - 2l\dot{\varphi}\dot{x}\sin\varphi$.

Таким образом, $T = \dfrac{m_1\dot{x}^2}{2} + \dfrac{m_1 l^2\dot{\varphi}^2}{2} - \dfrac{2m_1 l\dot{x}\dot{\varphi}}{2}\sin\varphi + \dfrac{J_1\dot{\varphi}^2}{2} + \dfrac{m_2\dot{x}^2}{2} =$

$$= (m_1 + m_2)\frac{\dot{x}^2}{2} + (m_1 l^2 + J_1)\frac{\dot{\varphi}^2}{2} - m_1 l\dot{x}\dot{\varphi}\sin\varphi .$$

Определим обобщенные силы Q_φ и Q_x .

Обобщенные силы Q_x и Q_φ можно определить и из выражения работы сил на элементарных перемещениях системы, соответствующих вариации каждой обобщенной координаты:

$$\delta A_x = Q_x \delta x = P\,\delta x; \qquad \delta A_\varphi = Q_\varphi \delta\varphi = M\,\delta\varphi ,$$

отсюда $Q_x = P$ и $Q_\varphi = M$.

Подставляя полученные значения кинетической энергии и обобщённых сил в уравнения Лагранжа II рода (2), получим

$$M(t) = \left(m_1 l^2 + J_1\right) \ddot{\varphi} - 2l^2 m_1 \ddot{\varphi} \sin^2\varphi - l^2 m_1 \dot{\varphi}^2 \sin 2\varphi$$
$$P(t) = \left(m_1 + 2m_2\right)\left(\ddot{\varphi} \sin\varphi + \dot{\varphi}^2 \cos\varphi\right) l.$$

Эти равенства представляют собой зависимость управляющего момента **M** и управляющего усилия **P** от известных функций φ, $\dot{\varphi}$ è $\ddot{\varphi}$.

Данные управляющие факторы находим с помощью пакета расширения Simulink , составляя блочную модель механизма манипулятора

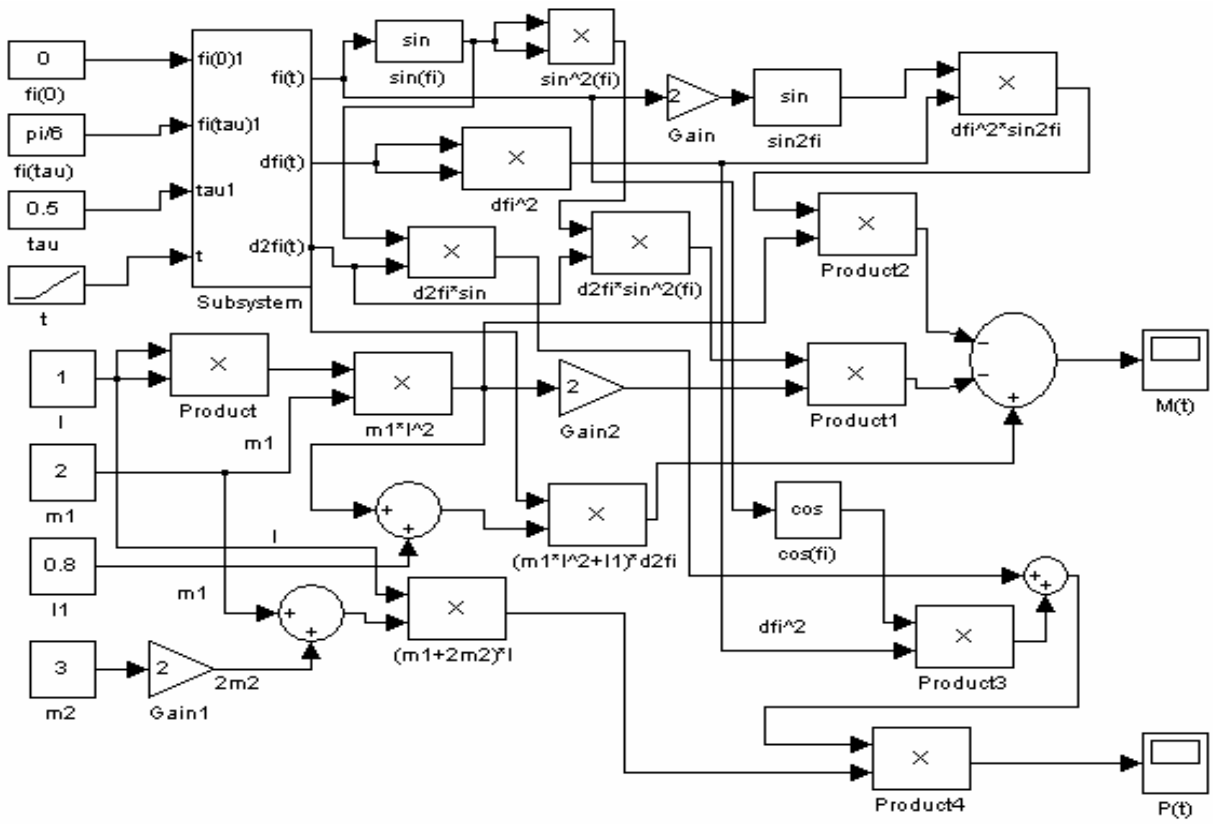


Рис. 4.21. Модель механизма манипулятора

Данная модель содержит подмодель (рис. 4.22), определяющая изменение φ, $\dot{\varphi}$ è $\ddot{\varphi}$ с течением времени

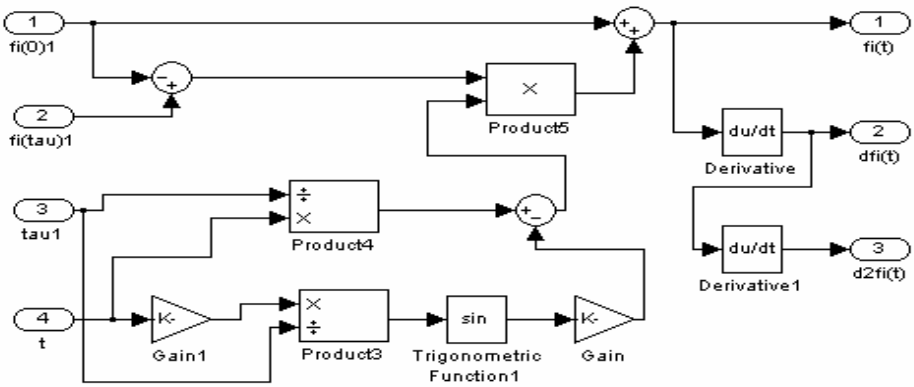


Рис. 4.22. Субмодель механизма манипулятора

Результат такого имитационного моделирования для управляющих факторов представляется в виде двух осциллограмм

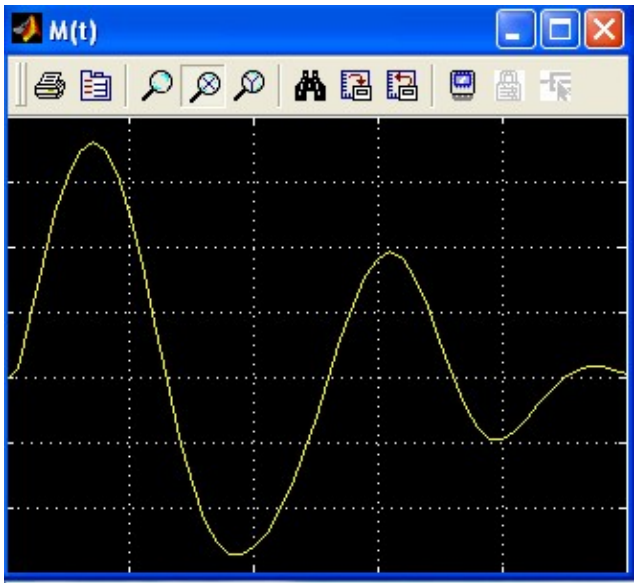


Рис. 4.23, a. Программное изменение управляющего момента M (t)

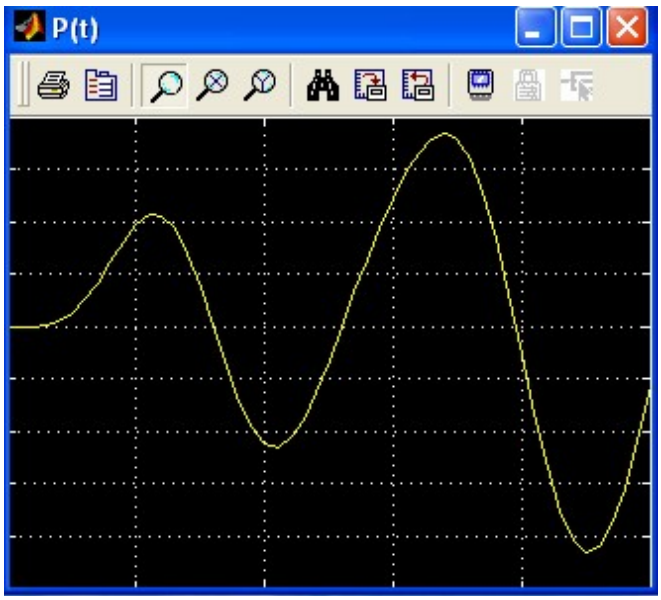


Рис. 4.23, b. Программное изменение управляющего усилия P (t)

Варианты расчётов управляющих моментов M (t) и сил P (t), обеспечивающих программное движение манипулятора приведены на рисунках 4.24 – 4.26, соответствующие числовые данные приведены в таблице.

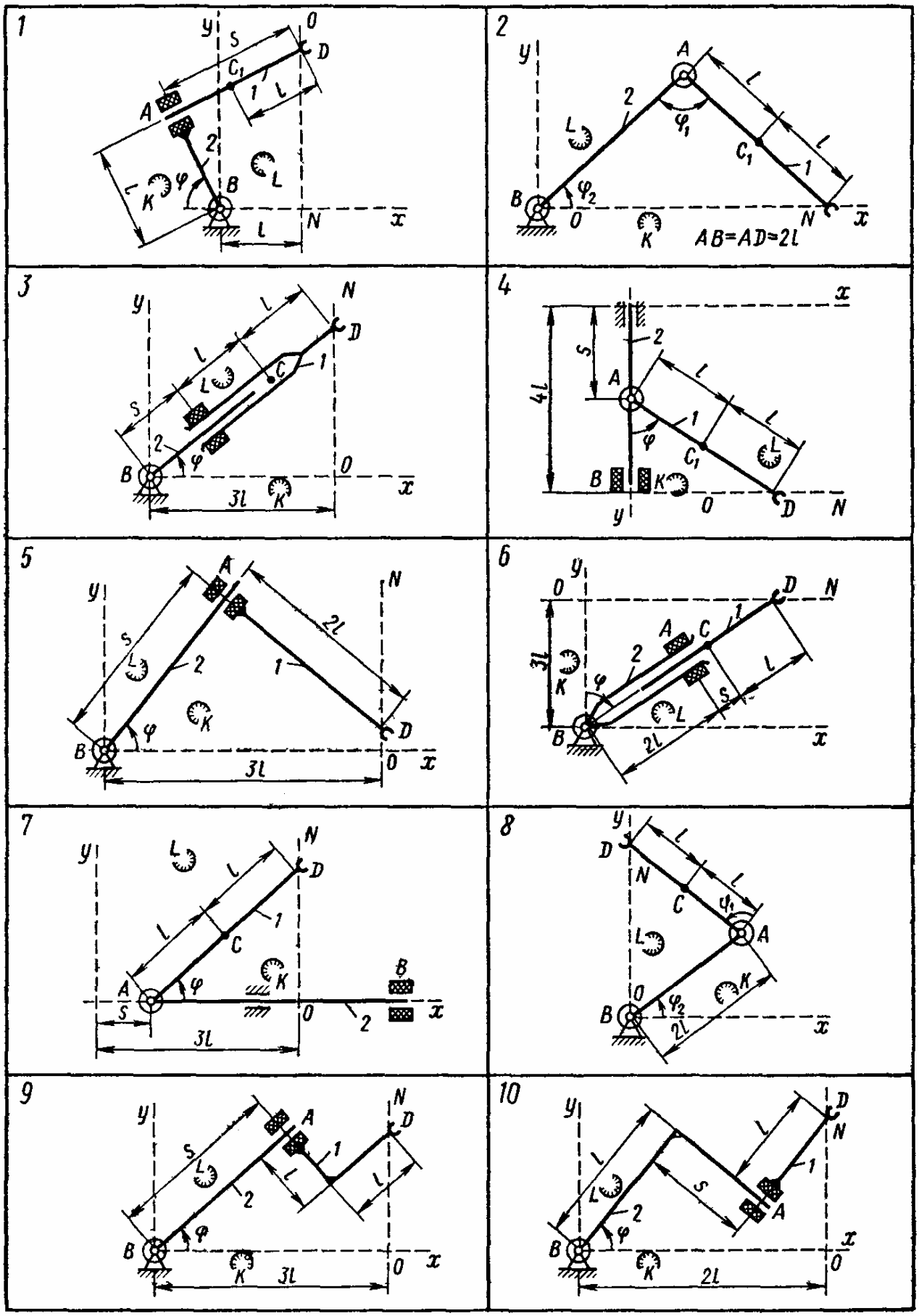


Рис. 4.24

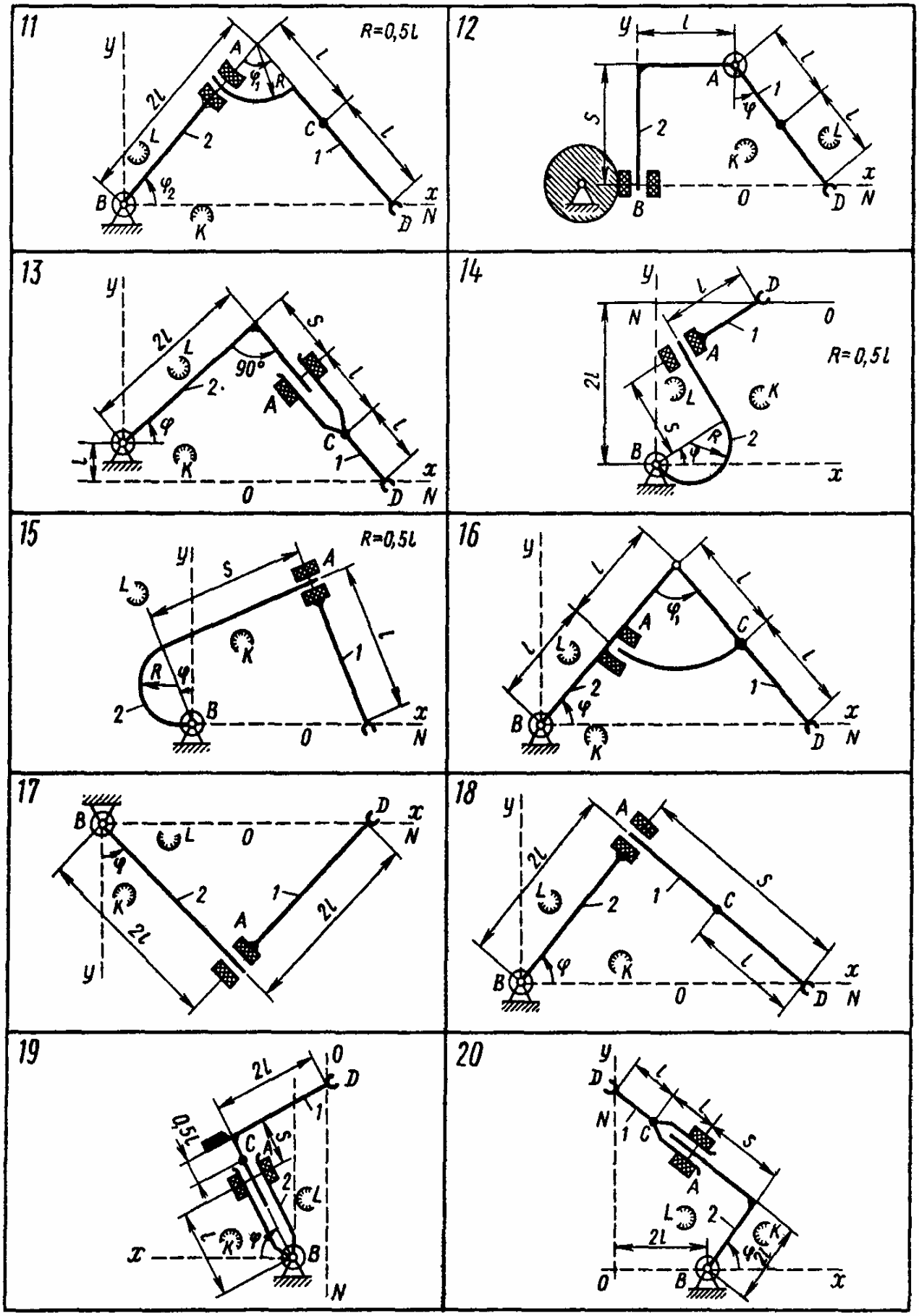


Рис. 4.25

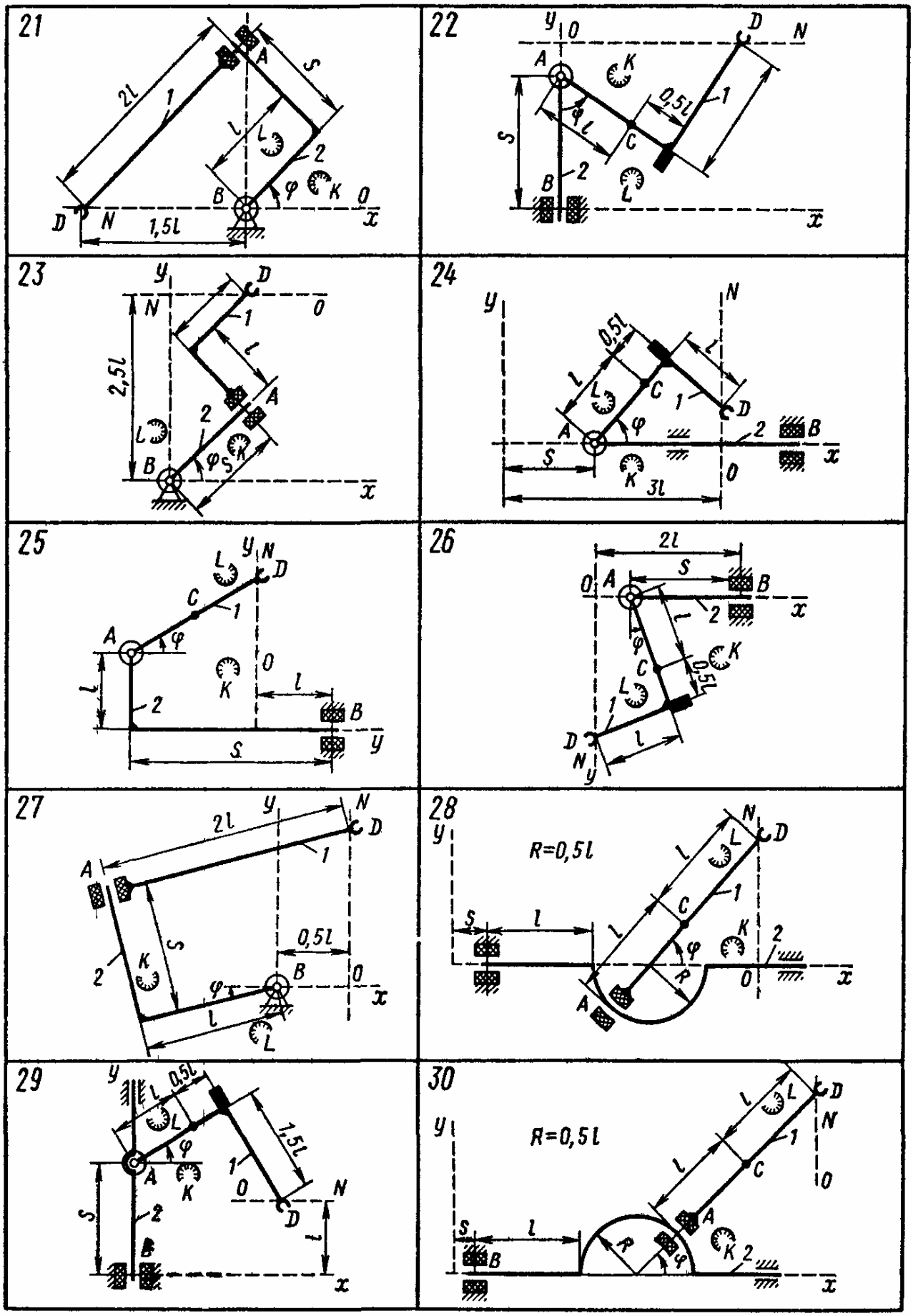


Рис. 4.26

Номер варианта	m_1	m_2	J_1	J_2	*l, м*	τ, с	φ(0)	φ(τ)
	кг	кг	кг м^2	кг м^2			рад	рад
1	3	-	0,5	0,6	0,5	1	π / 3	2 π / 3
2	4	-	1	2	0,3	0,25	0	π/3
3	2	-	1	2	0,2	1	0	π/3
4	2,5	2,5	0,8	-	0,8	0,5	π/6	π/3
5	4	-	2	2	0,5	2	π/6	π/3
6	2,5	-	1,2	1,5	0,3	0,9	0	π/3
7	2	3	0,6	-	0,7	0,5	π/6	π/3
8	3	-	0,9	2	0,4	0,3	π/6	π/3
9	2	-	0,8	1,4	0,3	1	0	π/3
10	3,5	-	1,4	1,6	0,4	2	0	π/6
11	2,5	-	1	1,4	0,5	0,5	0	π/4
12	3	3	0,7	-	0,7	0,4	π/6	π/3
13	4	-	1,5	2	0,3	0,5	0	π/6
14	3	-	1,4	1,8	0,4	1,5	π/6	π/3
15	2,5	-	1	1,2	0,5	0,5	π/6	π/3
16	3	-	1,2	2	0,3	0,3	0	π/4
17	3	-	1,2	2	0,3	1,8	π/6	π/3
18	2	-	0,6	0,8	0,4	0,8	π/6	π/3
19	3	-	1,7	2	0,3	0,6	π/6	π/3
20	3,5	-	1,2	1,6	0,4	0,6	π/4	π/2
21	4	-	2	2	0,5	0,5	π/6	π/4
22	2,5	4	1,2	1,8	0,3	0,8	π/6	π/2
23	4	-	1,1	1,7	0,4	1	π/3	π/2
24	3	3	1,4	-	0,6	1	0	π/3
25	2,5	3	1	-	0,8	0,5	0	π/3
26	3,5	4	1,6	2	0,4	0,9	0	π/6
27	2	-	1,1	1,5	0,4	0,7	0	π/3
28	3	2,5	0,7	-	0,7	0,4	π/6	π/3
29	2	3	1,3	-	0,5	1,2	0	π/3
30	2,5	4	0,8		0,6	0,4	π/6	π/4

ЛИТЕРАТУРА

1. Дьяконов В.П. MATLAB 6. Учебный курс. – СПб.: Питер, 2001. – 592 с.
2. Баженов В.А., Дащенко А.Ф., Коломиец Л.В., Оробей В.Ф. Строительная механика. Специальный курс. Применение метода граничных элементов. – Одесса: Астропринт, 2001. – 288 с.
3. Строительная механика. Динамика и устойчивость сооружений / Под ред. А.Ф. Смирнова. – М. : Стройиздат, 1984. – 415 с.
4. Турчак Л.И. Основы численных методов. – М.: Наука, 1987. – 320 с.
5. Мэтьюз Дж.Г., Финк К.Д. Численные методы. Использование MATLAB. Пер. с англ. – М.: Изд. Дом «Вильямс», 2001. – 720 с.
6. Дьяконов В., Круглов В. Математические пакеты расширения MATLAB. Специальный справочник. – СПб.: Питер, 2001. – 480 с.
7. Сборник задач для курсовых работ по теоретической механике: Учебное пособие для технических вузов / Яблонский А.А. и др. М.: Высш. шк., 1985. – 367 с.
8. Кириллов В.Х., Лещенко Д.Д. Курс теоретической механики. Учебное пособие для студентов технических вузов. – Одесса: ОГАСА, 2002. – 263 с.
9. Даффи Д.А., Бекман У.А. Тепловые процессы с использованием солнечной энергии. – М.: Мир, 1977. – 305 с.
10. Кириллов В.Х. Гидродинамика и тепломассообмен в двухфазных потоках плёночных аппаратов для холодильной техники: Дис…док. техн. наук. – Одесса, 1993. – 342 с.
11. Капустин В.Ф. Практические занятия по курсу математического программирования. – Л.: Изд. Ленинград. ун-та, 1976. – 192 с.
12. Дьяконов В. SIMULINK 4. Специальный справочник. – СПб.: Питер, 2002. – 528 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	……	3
ГЛАВА 1.	Основные элементы языка программирования и визуализации расчетов в системе MATLAB…….	6
1.1.	Алфавит языка программирования……	6
1.2.	Арифметические и логические операторы……	6
1.3.	Элементарные функции……..	8
1.4.	Понятие о файлах-сценариях и файлах-функциях……...	9
1.5.	Основы программирования……	11
1.5.1.	Оператор присваивания…….	11
1.5.2.	Перенос строки……	12
1.5.3.	Ввод и вывод данных……..	12
1.5.4.	Форматы чисел…….	13
1.5.5.	Формирование векторов и матриц…….	14
1.5.6.	Типы данных……	16
1.5.7.	Оператор двоеточие:...……	17
1.5.8.	Оператор разветвления *if* …….	17
1.5.9.	Операторы циклов……...	18
1.5.10.	Сообщения об ошибках и исправление ошибок……...	19
1.5.11.	Вычисление определителя квадратной матрицы……..	19
1.5.12.	Обращение матриц……..	20
1.6.	Основы графической визуализации вычислений……	20
1.6.1.	Построение графиков отрезками прямых……	21
1.6.2.	Графики дискретных отсчетов функции……..	22
1.6.3.	Создание массивов данных для трехмерной графики…..	22
1.6.4.	Построение графиков поверхностей……..	23
1.6.5.	Включение и выключение масштабной сетки……..	24
1.6.6.	Представление нескольких графиков в одном окне…….	24
1.6.7.	Ввод текста на график с помощью мыши…….	25
1.6.8.	Управление свойствами осей графиков…….	25
ГЛАВА 2.	MATLAB в задачах вычислительной математики……...	26
2.1.	Табулирование функций…….	26
2.2.	Решение систем линейных алгебраических уравнений методом исключения Гаусса……	30
2.3.	Аппроксимация функций……	35
2.4.	Численное решение обыкновенных дифференциальных уравнений……...	40
2.5.	Приближенное вычисление определенных интегралов...	44
2.6.	Численное решение нелинейных уравнений……	47
2.7.	Численное решение оптимизационных задач……..	49
ГЛАВА 3.	Применение MATLAB для решения задач строительной механики стержневых систем……	56
3.1.	Основные положения метода граничных элементов…	57
3.2.	Расчет неразрезной балки……..	62

3.2.1. Определение граничных параметров при статической нагрузке…….. 63
3.2.2. Построение эпюр прогибов, углов поворота, поперечных сил и изгибающих моментов……. 65
3.2.3. Определение частот собственных колебаний……... 69
3.2.4. Построение форм собственных колебаний……... 72
3.2.5. Расчет на вынужденные колебания…… 78
3.2.6. Построение эпюр напряженно-деформированного состояния балки при вынужденных колебаниях…….. 82
3.2.7. Определение критических сил потери устойчивости….. 87
3.2.8. Построение форм потери устойчивости…… 90
3.2.9. Варианты заданий…… 94
3.3. Расчет плоской рамы……... 98
3.3.1. Определение граничных параметров при статической нагрузке…….. 98
3.3.2. Построение эпюр напряженно-деформированного состояния элементов рамы…….. 103
3.3.3. Определение частот собственных колебаний…….. 106
3.3.4. Построение форм собственных колебаний…….. 110
3.3.5. Расчет на вынужденные колебания…… 115
3.3.6. Построение эпюр напряженно-деформированного состояния при вынужденных колебаниях……. 120
3.3.7. Определение критических сил рамы……. 125
3.3.8. Построение форм потери устойчивости рамы……. 127
3.3.9. Построение эпюр напряженно-деформированного состояния с разрывами 1-го рода……... 132
3.3.10. Варианты заданий…… 136
ГЛАВА 4. Применение пакетов расширений MATLAB в инженерно-научных задачах…… 140
4.1. Пакет расширений Symbolic Math……. 140
4.2. Пакет оптимизации Optimization Toolbox……. 168
4.3. Simulink – система визуального моделирования динамических систем……..…… 184
ЛИТЕРАТУРА 210

Наукове видання

ДАЩЕНКО Олександр Федорович
КИРИЛЛОВ Володимир Харитонович
КОЛОМІЄЦЬ Леонід Володимирович
ОРОБЕЙ Віктор Федорович

MATLAB в інженерних та наукових розрахунках

Монографія

Російською мовою

Книга видана в авторській редакції

Зав. редакцією *Т. М. Забанова*
Технічний редактор *М. М. Бушин*

Здано у виробництво 13.01.2003. Підписано до друку
Формат 60х84/16. Папір офсетний. Гарнітура
Друк офсетний. Ум. друк. арк. . Тираж 300 прим.
Зам. № 15/2.

Видавництво і друкарня “Астропринт”
(Свідоцтво ДК № 132 від 28.07.2002 р.)
65026, м. Одеса, вул. Преображенська, 24.
Тел.: (0482) 26-98-82, 26-96-82, 37-14-25.
www.astroprint.odessa.ua

Дащенко О. Ф., Кириллов В. Х., Коломієць Л. В., Оробей В. Ф.

Д MATLAB в інженерних та наукових розрахунках: Монографія. – Одеса: Астропринт, 2003. – 214 с.

Рос. мовою.

ISBN 966–549–

Монография посвящена иллюстрации возможностей одной из самых эффективных систем компьютерной математики MATLAB в решении ряда научных и инженерных проблем. Рассмотрены примеры решения задач математического анализа. Классические численные методы дополнены примерами более сложных инженерных и научных задач математической физики. Подробно изложены алгоритмы краевых задач статики, динамики и устойчивости упругих систем в методе граничных элементов. Рассмотрены пакеты расширений системы MATLAB символьных вычислений, оптимизации и моделирования. Применение компьютерных технологий решения задач посредством системы MATLAB иллюстрируется большим количеством упражнений и примеров.

Предназначена для студентов и аспирантов высших технических учебных заведений, будет полезна специалистам и научным работникам широкого профиля.

Д $\frac{-016}{549-2003}$ Без объявл.

ББК

УДК