

П. Сосис

624.04

С66

Статически неопределимые системы

624.01

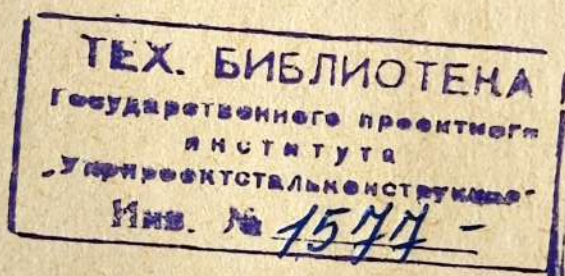
Зональный научно-исследовательский
и проектный институт типового и экспериментального проектирования
жилых и общественных зданий Государственного комитета
по гражданскому строительству и архитектуре при Госстрое СССР

1980
П. М. СОСИС

624.04
С66

СТАТИЧЕСКИ НЕОПРЕДЕЛИМЫЕ СИСТЕМЫ

Построение и развитие алгоритмов



092

8477

ИЗДАТЕЛЬСТВО «БУДІВЕЛЬНИК»
Киев — 1968

624.01.

Статически неопределимые системы. Сосис П. М., 1968, стр. 309.

Монография является изложением докторской диссертации покойного П. М. Сосиса. Подготовлена к изданию им лично и сотрудниками лаборатории программирования и механизации расчетов, руководителем которой он был на протяжении последних лет.

Обобщены многолетние исследования автора по созданию эффективных алгоритмов расчетов сооружений. Рассмотрены алгоритмы перераспределения для оперативного выполнения расчетов на логарифмической линейке, компактные схемы алгоритмов для клавишных автоматов, машинно-аналогические алгоритмы для электро моделирующих машин, алгоритмы с развитой логической схемой для электронных цифровых вычислительных машин. Описаны системы алгоритмов и соответствующие библиотеки программ, составленные в КиевЗНИИЭПе. Рассмотрены вопросы автоматизации комплекса работ по расчету зданий и сооружений на основе информации о их зрительном образе, моделирования в машине элементов творческой деятельности человека.

Рисунков 57, таблиц 14, библиография состоит из 142 позиций.

ПРЕДИСЛОВИЕ

Проблемы строительной механики, как науки о расчете конструкций, в состав которых могут входить как дискретные, так и непрерывные элементы, решаются в настоящее время на четырех последовательных уровнях: формулировка гипотез, являющихся фундаментом теории; доказательство теорем и вывод формул теории; разработка метода, в котором намечены основные этапы расчета и указаны возможные пути для их выполнения; алгоритм, в котором детально разрабатывается вычислительная процедура. Если метод применяется для обучения инженера и допускает различные формы реализации, то алгоритм однозначен и служит для непосредственного исполнения человеком или автоматом.

До недавнего времени в практике инженерных расчетов не всегда проводилась грань между последними двумя этапами. Это нередко приводило к путанице: достоинства алгоритма, вытекающие из его природы—ясной и недвусмысленной инструкции для расчета, противопоставлялись основному методу. Одни говорили о принципиальной новизне (например, о возможности не решать уравнения), другие, наоборот, сомневались, можно ли применять «новый метод» (например, ставили вопрос о доказательстве сходимости процесса итераций и т. п.).

В настоящей работе излагаются принципы построения алгоритмов для методов строительной механики. Внимание концентрируется на разработке оптимальных вычислительных процедур, моделировании в машине элементов мышления человека.

Понятие алгоритма — одно из основных в математике. Алгоритмом принято называть систему предписаний (приказов) о выполнении в твердо установленном порядке вполне определенных вычислительных и логических операций для решения всех задач данного типа. Практическое значение алгоритмов в науке и технике огромно, оно особенно возросло с появлением универсальных электронных цифровых вычислительных машин (ЭЦВМ),

позволяющих реализовать общие алгоритмы для широких классов задач, объединять посредством кибернетических абстракций задачи из различных областей знания. Более точное определение алгоритма, применяемое в теории алгоритмов [18, 66, 118], отождествляет его с алфавитным оператором, рекурсивной функцией, функциональной схемой для идеализированной вычислительной машины (так называемой машины Тьюринга-Поста).

Эффективность алгоритма в значительной степени зависит от степени его соответствия применяемым средствам вычислений, которые развиваются очень быстро. Поэтому из рассмотренных уровней решения проблем алгоритмический уровень наиболее подвижен и изменчив. Под этим углом зрения в работе проводится оценка существующих и построение новых алгоритмов.

Гл. I носит вводный характер — в ней рассмотрены принципы и закономерности построения алгоритмов расчета статически неопределимых систем. В последующих II—VII главах излагаются сами алгоритмы, объединенные в группы в зависимости от применяемых вычислительных средств: алгоритмы перераспределения, предназначенные для выполнения расчетов на логарифмической линейке (гл. II), машинно-аналитические алгоритмы — для электро моделирующих машин (гл. II), компактные алгоритмы — для клавишных автоматов (гл. III и IV), алгоритмы с развитой логической схемой — для электронных цифровых вычислительных машин (гл. V—VII).

С появлением больших электронных цифровых вычислительных машин алгоритмы и основанные на них программы стали принимать все более общий характер. В работе применяются два встречных процесса построения алгоритмов: дедукции (от общего к частному) и индукции (от частного к общему). В алгоритме, имеющем общий характер, предусмотрено обращение к некоторым абстрактным процедурам, для которых определено лишь назначение, но не содержание. Разработка содержательной части процедуры приспособливает общий алгоритм к определенному более узкому классу задач, а замена процедуры качественно меняет алгоритм. Данный алгоритм, в свою очередь, может входить в качестве процедуры в другой, более общий алгоритм. Этот процесс можно продолжить в обоих направлениях, внутреннем и внешнем, и получить рекурсивную цепочку алгоритмов — систему алгоритмов, охватывающую ту или иную проблему с многочисленными ее приложениями, или группу проблем, или даже ту или иную область науки в целом.

По этому принципу в КиевЗНИИЭПе разработана система

алгоритмов и соответствующая библиотека программ (для ЭЦВМ М-20 и БЭСМ 2М) расчета плоских и пространственных статически неопределимых систем различных видов. На левом конце дедуктивной цепочки находится универсальная программа метода деформаций с обращением к некоторой процедуре расчета элемента, содержание которой не оговорено. Это может быть расчет стержня в плоских или пространственных системах, расчет пластинки или пространственного блока и т. д. Остальные программы библиотеки получены прибавлением к универсальной программе той или иной процедуры расчета элемента, учитывающей его тип, возможные регулярности (ортогональность, циклическая или сферическая симметрия, общие топологические свойства, повторяемость) и характер работы (упругая или упруго-пластическая стадия). Универсальная программа метода деформаций и все ее варианты могут входить в состав более сложных программ, управляющих, например, процессом линеаризации в нелинейных системах или процессом вариации при непрерывных связях по линиям сопряжения элементов.

Аналогичный подход применяется при составлении библиотеки программ для метода сил, но здесь дополнительной процедурой является расчет той или иной основной системы. В сложных программах присутствуют блоки как метода деформаций, так и метода сил.

Указанный аппарат можно приложить также к расчету непрерывных систем, если для них разработан и обоснован метод дискретной аппроксимации; по этому вопросу уже имеется обширная литература [33; 37; 59; 61; 82; 87; 125—127; 136; 137].

В заключение рассматриваются логические построения и процедуры формирования основной системы путем анализа зрительного образа расчетной схемы, отображенной на сетках или решетках различных типов. Разработка этих алгоритмов является одним из этапов подготовки автоматизированной системы, охватывающей совокупность расчетов конструкций здания на основе информации о проектном задании, а в перспективе — проектирование и расчет как единый комплекс.

Оценка эффективности алгоритма является относительной. Применение новых вычислительных машин, расширение их возможностей неизбежно вызывает ломку прежних представлений, требует качественного изменения алгоритмов. Уяснение этого вопроса очень важно, так как всегда по инерции стараются применить старые, хорошо знакомые алгоритмы к новым вычислительным средствам. Это возможно, но, как правило, неэффек-

тивно. В проектной практике широко применяются самые различные средства вычислений: в небольших организациях — простые, в крупных организациях — более сложные и совершенные. Необходимо ясно представлять, в каких условиях и для каких задач следует использовать ту или иную группу алгоритмов.

Несколько слов о построении книги. Для удобства ссылок каждая глава разбита на большое число параграфов, не имеющих заглавия и обозначаемых двумя цифрами (знак § — опущен): первая цифра — номер главы, вторая — номер параграфа в главе. Кроме этого, в главах имеются подзаголовки, относящиеся к той или иной группе вопросов и указанные в оглавлении. В каждом параграфе принята своя нумерация формул, начиная с первого номера. При ссылках указывается номер параграфа и номер формулы в параграфе. Всякого рода иллюстративные таблицы обозначены двумя номерами: первая цифра — номер главы, вторая — номер таблицы в главе. Стандартные таблицы, относящиеся к программам, не связаны с этой нумерацией и имеют свои обозначения в виде цифр (для каждой программы начинается с первого номера) или прописных букв. Нумерация рисунков сквозная; такая же нумерация принята и для табличных схем.

Настоящая монография является обобщением работ автора, выполненных в КиевЗНИИЭПе (1960—1967 гг.), и в предшествующий период (1951—1960 гг.) в ГПИ Укрпроектстальконструкция.

В рецензировании рукописи принимали участие акад. В. М. Глушков и канд. физ.-мат. наук А. А. Летичевский.

РАЗВИТИЕ АЛГОРИТМОВ

1.1. В обзорной книге «Строительная механика в СССР. 1917—1957 гг.» [114] рассмотрена история развития основных методов и различных их модификаций, точных и приближенных, представляющих в сущности те или иные алгоритмы, предназначенные для упрощения и облегчения численного расчета сооружений.

С тех пор, как была написана эта книга, прошло всего 10 лет, а в расчетной практике произошли коренные изменения: в ведущих проектных институтах для расчетов сложных сооружений применяются быстродействующие электронные вычислительные машины, электро моделирующие машины и клавишные автоматы. Роль и значение основных методов строительной механики — сил и деформаций — неизмеримо возросла, изменился взгляд на многие упрощенные и приближенные методы расчета, еще недавно весьма популярные. Наряду с этим нужно сказать, что у многих инженеров, применяющих машинную технику, нет ясности в вопросе о том, какие алгоритмы из богатого арсенала существующих нужно применять, в каких случаях следует разрабатывать новые алгоритмы и т. д. Прежде всего необходимо уяснить, что эффективность того или иного алгоритма зависит от характера решаемых задач, от того, кем и с помощью каких средств, приборов или машин они решаются. Некоторые алгоритмы, приспособленные для одних задач, неэффективны или просто неприменимы для других. Один алгоритм необходим при решении задачи квалифицированным специалистом — инженером, другой, — если расчет выполняется оператором клавишного автомата, третий — при автоматическом расчете на ЭЦВМ. При построении алгоритмов следует учесть возможности и специфические особенности тех машин, на которые они рассчитаны. Этот подход является определяющим в данной работе.

Рассмотрим кратко принципиальные предпосылки групп алгоритмов, предназначенных для определенных видов вычислительных средств. Сами алгоритмы будут изложены в последующих главах.

Алгоритмы перераспределения

1.2. В 30—40 гг. большим распространением в проектной практике пользовался метод распределения моментов защемления Бернадского-Кросса (предложен в 1929 г. Н. М. Бернадским [4] и, в несколько другой форме, Харди Кроссом [138]). Развитию и популяризации этого метода в СССР способствовали работы С. А. Рогицкого [83], Ш. М. Гофмана [24, 26], П. П. Шагина [129], И. К. Снитко [94], Н. К. Снитко [95] и др. Этот метод, применяющийся и сейчас довольно широко, представляет собой по существу определенную форму решения уравнений метода деформаций способом последовательных приближений и имеет следующие особенности:

в процессе последовательных приближений определяются не углы поворота, а моменты защемления по концам стержней, сходящихся в узлах;

расчет интерпретируется не математически, как решение уравнений, а статически — как последовательное уравнивание узлов рамы, что позволяет инженеру как бы наблюдать «игру сил» в системе;

для наглядности расчет выполняется на схеме рамы;

процесс расчета разбивается на элементарные операции, которые легко запомнить.

Своеобразие метода многие инженеры восприняли как нечто принципиально новое: говорили о том, что метод Кросса позволяет не решать системы уравнений, резко сокращает количество вычислений по сравнению с классической формой метода деформаций и т. п.

Между тем, простейший анализ показывает, что количество вычислений не сокращается, а наоборот увеличивается. В самом деле, вместо того, чтобы в каждом цикле расчета определять для каждого узла одну величину — угол поворота, вычисляется несколько величин — моменты на концах стержней. Эта операция выполняется многократно (!) по числу циклов, что приводит к громоздкости расчета. По существу, этим сводится на нет основное преимущество метода деформаций перед методом сил, состоящее в резком сокращении числа неизвестных в многопролетных рамах. Очевидно, что обычное решение уравнений метода деформаций итерационным способом более экономно, поскольку многократно определяется в несколько раз меньше неизвестных — углов поворота, заключительная же стадия определения моментов выполняется только один раз. Недостатки метода Крос-

са уменьшаются, если воспользоваться различными приемами ускорения сходимости.

I. 3. При анализе причин широкого распространения метода Кросса можно сделать вывод, что они заключаются в наглядном и четком алгоритме, подкупающей инженера простой физической трактовке, в применении приемов записи исходных данных и результатов на схеме рамы. Отсюда возникла мысль использовать указанные приемы, но возвратиться к классической форме решения уравнений метода деформаций итерационными методами. Это было осуществлено в методе перераспределения узловых деформаций, сформулированного в 1949—1950 гг. [96, 97], и в другой форме — в работе Клоучека [139]. Указанные недостатки метода Кросса * были устранены впоследствии Э. Кани, метод которого [140] имеет сходные черты с методом перераспределения узловых деформаций: объединяются операции распределения и переноса (применяется коэффициент поворота), окончательное определение моментов выполняется на отдельной схеме. Однако стремление автора сохранить при этом распределение моментов защемления (а не деформаций) потребовало введения ряда искусственных приемов для различных типов рам, из-за которых теряется простота и единообразие метода.

Широкое развитие итерационные методы расчета рам получили в работе Н. К. Снитко [95], как в классической форме решения систем уравнений, так и в форме распределения узловых и поэтажных моментов.

Чрезмерное увлечение методом Кросса привело к тому, что с его помощью стали решать многие задачи, для которых целесообразнее использовать метод сил. В этих случаях механизм перераспределения лучше применять при основной системе метода сил и перераспределять моменты в шарнирах или другие усилия.

I. 4. Значительно раньше метода Кросса появились методы моментных и угловых фокусных отношений. Здесь также производится распределение (разгонка) усилий, но на другой основе. По существу, метод фокусов представляет собой интерпретацию точного решения систем трехчленных уравнений. Этот метод имеет серьезные недостатки. Схема поочередного нагружения отдельных стержней и формулы для моментов или углов поворота на концах нагруженного пролета (полученные в результате решения в общем виде двух уравнений с двумя неизвестными)

* Слово «метод» употребляется, как было принято в литературе, но речь идет о том или ином алгоритме выполнения расчетов методом деформаций, о чем сказано в предисловии.

достаточно громоздки и неудобны во многих случаях. Попытка распространить метод фокусов для неразрезной балки на упруго податливых опорах [85] привела к еще более громоздким формулам, полученным в результате решения в общем виде четырех уравнений с четырьмя неизвестными. Для рам с замкнутыми контурами метод фокусов уже не является точным и применение его неразрывно связано со способом последовательных приближений (последовательный обход контуров) как при определении фокусных отношений, так и при нахождении искомых величин моментов (или углов поворота).

Уже давно было замечено, что при решении трехчленных уравнений методом Гаусса в неявном виде определяются фокусные отношения, но схема расчета получается более простой. Однако метод фокусов имеет и определенные достоинства, обусловившие его широкое применение в свое время (затем он уступил место методу Кросса) — это наглядность и медленное накопление погрешности, что позволило производить расчет на логарифмической линейке. Последнее обстоятельство является очень важным, если расчет выполняется инженером. По этой же причине метод фокусов служил долгое время источником для многих хороших приближенных методов (методы Погожева [71], Лезера и др.).

1.5. Детальный анализ показал, что достоинства метода фокусов можно сохранить, устранив многие из его недостатков, если воспользоваться точным (одношаговым) вариантом алгоритма перераспределения начальных значений неизвестных, по своему механизму близким к методу последовательных приближений. Разумеется, практически вполне точных методов не существует. Выражение «точный алгоритм» употребляется в математическом смысле: если вычисления производить точно (например, пользуясь простыми дробями), то точный результат получается в этом случае при конечном числе вычислений. Итерационные же методы теоретически всегда дают приближенный результат, связанный с пренебрежением остаточным членом, хотя и весьма малым для быстросходящихся процессов.

Точный алгоритм перераспределения применим для любых систем уравнений ленточной, или кодиагональной структуры (коэффициенты расположены полосой вдоль главной диагонали), частным случаем которой являются трех-, пяти-, семи-, девяти-... членные системы уравнений [99]. Формулы для вычислений начальных значений неизвестных более просты, чем по методу фокусов, так как получаются в результате решения элементарных

систем уравнений с вдвое меньшим числом неизвестных; не требуется производить поочередное загрузку стержней системы, нагрузка может быть произвольной. В применении к трехчленным уравнениям к этой схеме близки приемы, изложенные в работах А. Ф. Папковича [70], А. А. Уманского [119].

1. 6. Таким образом, был сформулирован обобщенный алгоритм перераспределения начальных значений неизвестных, применяемый с некоторыми изменениями при различных основных системах (метода сил, деформаций и смешанного) и при различных способах решения задачи (точном и итерационном). Краткая, но исчерпывающая характеристика алгоритма перераспределения начальных значений неизвестных дана в обзоре [114].

Этот алгоритм, как и те методы решения систем уравнений, на которых он основан, имеет определенные ограничения: итерационный вариант возможен при условии, что процесс достаточно быстро сходится; точный вариант целесообразен для симметричных ленточных матриц с небольшим количеством кодиагоналей. В некоторых случаях хорошие результаты дают различные сочетания точного и итерационного вариантов, математическая суть которых состоит в выделении трех- и пятичленных групп уравнений.

Компактные схемы алгоритмов

1. 7. Достоинства алгоритмов перераспределения теряют свое значение при переходе к другому виду вычислительной техники — клавишным электромеханическим автоматам типа Рейн-металл-Суперметалл, на базе которых во многих проектных институтах созданы специальные машиносчетные бюро. На этих машинах вычисления могут быть выполнены с большим количеством верных значащих цифр, поэтому требование о медленном накоплении погрешностей становится менее жестким. Поскольку исполнителями расчета являются операторы-вычислители (лица, умеющие только механически считать на машине и совершенно не знакомые со строительной механикой), теряют смысл приемы, разъясняющие статическую суть выполняемых операций, зато возникает другая проблема, в каком виде передавать задание и инструкцию исполнителю для выполнения расчета, иными словами, как передать информацию от инженера к вычислителю. Здесь можно предложить два способа: объединение в одной группе инженеров и операторов; использование специального формализованного языка, понятного оператору.

1.8. При первом способе числа записываются в простейшие табличные бланки, в заглавных строках которых указана последовательность вычислений. В более сложных случаях, не укладывающихся в простейшие формы, инженеру приходится обучать оператора той или иной методике и наблюдать за его работой, а на стыке между различными этапами расчетов обрабатывать полученные результаты и выдавать новое задание и т. д. Недостаток указанной организации вычислительного процесса состоит в необходимости обучения операторов инженерным методам расчета, что не всегда возможно из-за неподготовленности операторов. Кроме того, предполагается создание жесткой технологии, основанной на определенном методе расчета. Если алгоритм уже устарел, приходится ломать налаженную технологию (на это идут крайне неохотно), что приводит к ограниченности. Недостатком является также необходимость постоянного наблюдения и контроля над оператором со стороны инженера.

1.9. Второй способ лишен этих недостатков. Пользуясь формализованным языком, можно описать любой алгоритм и представить его в виде стандартной программы вычислений. Такой язык был предложен в 1959—1961 гг. [100, 101, 102] и нашел применение во многих проектных институтах. В основу этого алгоритмического языка положен принцип адресности по аналогии с программированием для электронных цифровых машин. Положение числа в таблице (матрице) указывается по координатному принципу; введены специальные символы, обозначающие строки или графы (столбцы) в заданных пределах; знак суммирования, обозначения логических операций по типу — **если то иначе** и т. д. Несмотря на исключительную простоту, язык инженерного программирования (вычислители усваивают его за несколько часов) является универсальным языком и пригоден для записи произвольных численных алгоритмов.

Формула инженерной программы записывается в виде оператора засылки, в левой части которого указывается, откуда взять числа (их адреса) и какие действия над ними произвести, в правой — куда записать результаты вычислений (адреса результатов). Программа вычислений, описывающая тот или иной алгоритм, состоит из набора формул, разделяемых точками с запятыми, включая проверочные формулы.

Предложенная символика исходит от матричного исчисления и имеет много общего с ним, но использует термины более привычные для вычислителя и благодаря адресной форме записи позволяет оперировать с любыми группами чисел с указанием пре-

делов для строк, граф, массивов в разных таблицах, что дает возможность избежать во многих случаях формальных преобразований матриц.

I. 10. Эффективный алгоритм для клавишных автоматов (и построенная на его основе программа) должен отвечать следующим основным требованиям:

быть простым и доступным для вычислителя. В ряде случаев для простоты программы можно несколько увеличить объем вычислений. Следует иметь в виду, что трудоемкость расчета зависит не только от количества операций, но и от таких факторов, как единообразие расчета, создающего определенный ритм работы, удобство нахождения в таблицах всех необходимых данных и т. д.;

обладать свойством цикличности, повторяемости. Следует стремиться к тому, чтобы вычисления выполнялись по небольшому числу формул с изменением входящих в них адресов. Однако при этом должно соблюдаться предыдущее требование;

включать построчные (пографные) проверки. Если проверочная формула не соблюдается, то вычисления должны быть повторены для отрезка программы, расположенного между двумя проверочными формулами; переход к следующему этапу производится лишь после того, как проверка будет удовлетворена. Такое построение алгоритма обеспечивает значительно лучший контроль, чем так называемая проверка «второй рукой», при которой возможно повторение тех же или даже появление новых ошибок. Включение независимых самопроверок в алгоритм резко уменьшает вероятность ошибок в конечных результатах, отпадает также необходимость в дублировании расчетов;

использовать возможности клавишных автоматов по автоматическому накоплению сумм произведений чисел, что позволяет уменьшить количество записей и вводов данных. Предпочтение должно быть отдано так называемым компактным схемам алгоритмов, в которых числа в таблицах являются, как правило, результатами автоматических групповых операций;

быть типовым, стандартным, т. е. описывать расчет не одной какой-либо задачи, а целого класса задач. Это дает возможность использовать алгоритм многократно для любых задач данного класса. При записи алгоритма в форме стандартной программы подготовка конкретной задачи к решению сводится к привязке табличных бланков, в которых нумерация строк и граф должна соответствовать нумерации всех элементов или усилий системы.

I. 11. Нетрудно убедиться, что эти требования не удовлетворяются, если использовать для составления программы алгоритмы, приспособленные для логарифмической линейки, например алгоритмы перераспределения. Хотя составление программы на этой основе и возможно (как уже говорилось, на алгоритмическом языке может быть описан любой алгоритм расчета), но явно нецелесообразно. Сами достоинства алгоритма перераспределения, проявлявшиеся при расчете на логарифмической линейке, превращаются в недостатки при использовании таких вычислительных средств, как клавишные автоматы.

В самом деле, вычисления на логарифмической линейке производятся только над двумя числами. Промежуточные результаты можно не записывать только при многократных умножениях или делениях. Если же нужно сложить ряд произведений (скалярное произведение — самая распространенная операция в линейной алгебре), то все произведения предварительно следует где-то записать. Запись этих чисел необходимо производить в табличных бланках, что упорядочивает расчет и облегчает последующую проверку.

При выполнении же вычислений на клавишной машине такая схема обычно приводит к большому количеству ненужных записей, которые занимают больше времени, чем сам счет, выполняемый автоматически. Вот почему, например, при решении систем уравнений вместо обычного алгоритма Гаусса, в котором записываются все промежуточные результаты, применяются теперь более эффективные для клавишных машин компактные схемы. Компактностью отличаются также известные матричные операции.

I. 12. При использовании инженерных программ для клавишных машин вся информация задается и перерабатывается в таблицах (матрицах). Поэтому инженерное программирование можно в известном смысле назвать также матричным программированием, хотя для овладения им и не требуется знать матричную теорию.

Нами составлен ряд стандартных инженерных программ расчета стержневых систем методами сил и деформаций, определения частот и форм свободных колебаний, расчета пластин и оболочек методом конечных разностей [102]. Работа по составлению инженерных программ продолжена также другими авторами [13, 22, 73].

Благодаря стандартным программам клавишные автоматы превращаются как бы в машины с программным управлением.

с тем отличием от ЭЦВМ, что только счет выполняется автоматически (причем отсутствует быстроедействие), а набор данных на клавиатуре и запись результатов арифметических операций осуществляются оператором вручную. Роль внешней памяти для клавишных машин играют таблицы (матрицы), количество которых может быть неограниченным.

I. 13. Несмотря на относительно (по сравнению с электронными вычислительными машинами) медленное выполнение расчетов, клавишные автоматы сохраняют свое значение, по-видимому, в течение длительного времени. Решение задач на ЭЦВМ пока еще требует дополнительной многоступенчатой работы (кодировка исходных данных, перфорирование и т. п.), которая не только занимает время, но и должна быть выполнена весьма тщательно, так как любая даже самая ничтожная ошибка приводит к неверному результату. Многие ЭЦВМ имеют еще сильные ограничения в отношении размеров оперативной памяти, их внешние запоминающие устройства работают не всегда устойчиво, а устройства ввода и вывода — недостаточно быстро.

Не следует забывать также, что ЭЦВМ имеются еще далеко не везде, тогда как недорогие клавишные автоматы повсеместно распространены и продолжается их совершенствование. Нашей промышленностью созданы и в ближайшее время пойдут в серию электронные машины на полупроводниках типа ЛАДА, ВЕГА и др., которые выгодно отличаются от электромеханических машин скоростью и бесшумностью.

Можно поэтому считать, что клавишные автоматы (электромеханические и особенно электронные) весьма перспективные для тех задач строительной механики, решение которых на этих машинах не слишком трудоемко и занимает от нескольких часов до нескольких дней.

Машинно-аналитические алгоритмы для электро моделирующих машин

I. 14. Как известно, электро моделирующие машины основаны на аналогиях, существующих для определенных задач, между распределением физических величин в исследуемых системах и распределением электрических величин в специально подобранных схемах. Если цифровые вычислительные машины (клавишные автоматы или ЭЦВМ) являются машинами универсальными, то электро моделирующие машины относятся к разряду спе-

циализированных, т. е. каждая из них пригодна для решения определенного более или менее узкого класса задач.

Отличительной особенностью является также то, что интересующие нас величины получаются не алгоритмически, не с помощью вычислений, занимающих определенное время в зависимости от их количества, а как практически мгновенный процесс распределения токов и напряжений в электрической схеме с подведенными к ней данными в виде токов и напряжений. Расчет на электро моделирующей машине состоит обычно из следующих операций: набор схемы на так называемом наборном поле машины, ввод характеристик системы, ввод исходных данных, снятие результатов с приборов. При использовании прямых аналогов аналитическая часть расчета состоит лишь в переводе механических величин в электрические и обратно. Совершенствуя конструкцию машины, можно добиться упрощения каждой из указанных операций и уменьшения требуемого для ее выполнения времени. В идеале можно представить себе клавишную коммутацию и ввод характеристик и исходных данных, а также вывод результатов на световое табло или печать в виде цифр или графических изображений.

На практике, чтобы не слишком удорожать машину, идут на определенный компромисс: выполнение указанных операций не всегда удобно (например, ручная коммутация с помощью проводов и штекеров, поворот ручек, определение величин на приборах, установка индикатора на нуль и т. д.), но конструкция машины при этом значительно упрощается. Обычно стремятся к некоторому оптимуму, учитывающему целевое назначение, удобство эксплуатации и стоимость машины.

I. 15. До недавнего времени внедрение электро моделирующих устройств сдерживалось тем, что прямые электрические аналоги были известны для сравнительно ограниченного круга задач строительной механики. Для строительной механики стержневых систем — это несвободные рамы, для прикладной теории упругости — задачи, описываемые дифференциальными уравнениями типа Лапласа-Пуассона. Разработка и широкое применение электро моделирующих машин связано с появлением теории квазианалогового * моделирования, сообщившей аналоговому методу мощный толчок для развития.

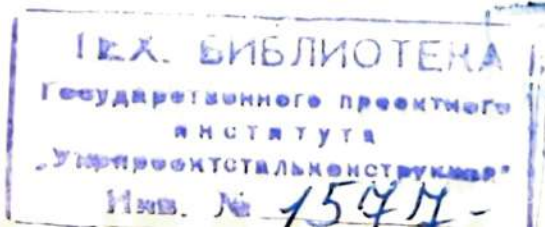
Автор теории квазианалоговых систем член-корр. АН УССР Г. Е. Пухов так определяет ее суть: «В основе теории лежит не

* *Cvasi* по-латински означает «как бы» (как будто). Этим подчеркивается, что в данном случае речь идет о не прямых аналогиях.

принцип подобия уравнений объекта и модели, а более общий принцип эквивалентности в отношении получаемых результатов... под квазианалоговой моделью каких-либо уравнений (а) понимается аналоговая модель иных уравнений (б), не подобных (а) и таких, чтобы при выполнении условий эквивалентности все или часть неизвестных (б) совпали с точностью до постоянных множителей с неизвестными исходных уравнений (а)».

Нетрудно заметить здесь родство с широко применяемой в строительной механике идеей использования «изученных» основных статически неопределимых систем, решение для которых заранее известно. Так, например, в качестве основной системы при расчете свободной рамы при аналитическом расчете можно принять раму несвободную и составить затем дополнительную систему уравнений равновесия ярусов. При машинно-аналитическом решении можно поступить аналогичным образом: с помощью аналоговой машины рассчитать основную статически неопределимую систему и затем составить и решить дополнительную систему уравнений. Однако более удобен другой путь, основанный на методе последовательных приближений. Использование как вспомогательного средства электромоделлирующей машины открывает новые возможности для построения аналитических итерационных схем. Нередко оказываются целесообразными те итерационные способы, которые при чисто аналитическом решении являются слишком громоздкими. Это объясняется тем, что электрическая модель дает возможность мгновенно получить любую величину для многократно статически неопределимой основной системы, аналитический расчет которой весьма трудоемок.

1.16. При квазианалоговом моделировании эквивалентная система — прямой электрический аналог соединяется со специальными уравнивающими устройствами. Уравнивание заменяет аналитическую часть расчета в машинно-аналитическом методе. Если для прямого электрического аналога достаточно ввести в машину исходные данные и снять результаты, то для квазианалога нужно еще выполнить дополнительный «расчет», заключающийся в уравнивании системы. Таким образом, квазианалоговое моделирование можно трактовать как использование изученных основных статически неопределимых систем, но ввиду особых возможностей, представляемых аналоговыми машинами, идея эта значительно глубже. Эквивалентная система уравнений (с точки зрения представлений инженера — основная система), имеющая прямой электрический аналог, может иметь как меньшее, так и большее количество не-



2477

известных, чем у исходной системы. Далее подбором некоторого вектора (либо вручную — поворотом ручек, либо с помощью дополнительного автоматического устройства) «уравновешивают» основную систему, т. е. добиваются того, чтобы результирующие величины, получаемые в эквивалентной системе, совпадали с соответствующими величинами исходной системы. Принципиально возможен и такой подбор эквивалентной системы с большим количеством неизвестных, чтобы часть получаемых величин неизвестных совпадала с искомыми величинами в исходной системе и тем самым исключалась необходимость в уравновешивании. Поэтому квазианалоговые модели могут быть двух типов: уравновешиваемые и неуравновешиваемые. Созданные до сего времени квазианалоговые модели являются, как правило, уравновешиваемыми.

Подобно тому, как в строительной механике возможности выбора рациональных основных систем в тех или иных частных случаях не ограничены, так и здесь существует большой простор для построения моделей эквивалентных систем, имеющих прямые электрические аналоги.

Более того, появляются принципиально новые возможности. В самом деле, что значит выбрать основную систему с большим количеством неизвестных, чем у исходной системы? Такая постановка вопроса не имеет смысла при аналитических расчетах, так как редко бывает известно решение для более сложной системы. В квазианалоговых моделях, наоборот, такое решение является обычным, ибо какой бы сложной не была эквивалентная система, она является прямым аналогом, и решение получается мгновенно.

I. 17. В настоящее время создан ряд эффективных электромоделирующих машин и устройств для расчета балок, рам, некоторых видов пластин и оболочек. Особенно хорошо себя зарекомендовали удобные и недорогие машины ЭМСС-7 и ЭМСС-7М (электрическая модель стержневых систем), серийно выпускаемые нашей промышленностью и имеющиеся во многих проектных институтах. Здесь эквивалентная система имеет такое же количество неизвестных, как и исходная, но отличается от нее. Если ввести нагрузку, то полученные результаты не будут отвечать действительности. Это противоречие устраняется последовательным уравновешиванием перекосов. Машина предназначена для расчета рам определенного вида: стойки должны быть вертикальными, ригели без разрывов. Однако то обстоятельство, что в конструкции машины предусмотрен ввод любого заданного де-

формационного воздействия, делает возможным разработку эффективных итерационных машинно-аналитических методов.

Автором при участии конструкторов ЭМСС-7 были разработаны методы основных жесткостей и групповых перемещений [103, 104], получившие распространение в проектных институтах для расчета так называемых нерегулярных рам — со стойками разной высоты и наклонными, ригелями на разных уровнях, в виде безраскосных ферм и т. п. В качестве основной системы служит электрическая модель несвободной рамы, подвергаемая грузовому и последовательным деформационным воздействиям. Небольшая аналитическая часть расчета выполняется с помощью простого итерационного алгоритма, близкого по природе к алгоритмам перераспределения. Машинно-аналитические алгоритмы этой группы характеризуют подход к решению вопроса о преодолении ограничений электро моделирующих машин, если они имеются *.

1. 18. Следующим шагом вперед было создание в институте кибернетики АН УССР машины ЭМСС-8 (система «альфа»). Эта машина [116, 117] дает решение для рам произвольного вида: плоских и пространственных, из стержней переменного сечения, с любыми структурными нерегулярностями. Очень ценно, что на ней одновременно можно производить расчет нескольких рам. Единственное ограничение состоит в том, что не учитываются продольные деформации стержней. Для учета этого фактора можно использовать машинно-аналитический метод.

Машина ЭМСС-8 интересна также тем, что уравнивание этой системы происходит не алгоритмически, а благодаря специальным усилителям как практически мгновенный процесс (решение является не итерационным, а точным).

Заметим, что с появлением машины ЭМСС-8 предыдущая модель ЭМСС-7 не теряет все же своего значения, так как она стоит во много раз дешевле, коммутация и ввод данных для распространенных типов рам производятся в ней проще.

Дальнейшее совершенствование электро моделирующих машин будет, по-видимому, происходить в направлении автоматизации подготовительных процессов.

* Помимо стержневых систем, квазианалоговый метод был успешно применен к расчету определенных типов оболочек и пластин (работы О. В. Лукина, А. Е. Степанова, В. М. Самуся, В. Д. Шайкевича, Ю. М. Почтмана).

Алгоритмы для электронных вычислительных машин

I. 19. Так же как рассмотренные выше клавишные автоматы, электронные вычислительные машины являются цифровыми и универсальными. Но на этом сходство кончается. Это качественно новый вид вычислительных машин, и возможности его несравненно шире, прежде всего, благодаря устройству автоматического программного управления. Расчет произвольной сложности, если для него составлена программа, выполняется машиной автоматически от начала и до конца без вмешательства человека. Скорость счета чрезвычайно велика, например в серийных машинах типа М-20, БЭСМ-3М, БЭСМ-4 — 20 тысяч арифметических операций в секунду. Это позволяет не только резко сократить время на выполнение сложных расчетов, но и решать задачи, практически не выполнимые на клавишных автоматах. В частности, оказывается возможным быстро решать сложные задачи прикладной теории упругости (строительной механики в широком смысле), которые приводятся к системам уравнений весьма высших порядков, тригонометрическим рядам и т. д. Вычисление коэффициентов и решение систем уравнений со многими десятками или сотнями неизвестных на клавишных автоматах практически невозможно, так как требует многих месяцев или даже лет напряженного труда. Пока не было быстродействующих электронных машин, приходилось в этих случаях либо ограничиваться примитивными методами расчета, которым инженер не вполне доверял и поэтому допускал излишние запасы прочности, либо отказываться (что нередко имело место) от применения новых конструкций в пользу менее рациональных, но изученных решений.

I. 20. Для ЭЦВМ понятия алгоритм и программа по существу тождественны. В последнее время под программой стали понимать алгоритм, написанный на специальном алгоритмическом языке.

Программа в кодах машины, ранее составлявшаяся вручную специалистом особой квалификации — программистом, все более становится делом чисто техническим и может быть получена с помощью формального перевода (трансляции) программы, изложенной на алгоритмическом языке, на язык машины. Этот перевод может быть выполнен самой машиной с помощью программирующей программы (транслятора) либо техником, механически по определенным правилам кодирующим программу, записанную на алгоритмическом языке. В связи с этим освоение

и широкое применение алгоритмического языка (в частности, для задач строительной механики) имеет первостепенное значение.

Для алгоритмических языков характерна близость к математике. Отличие состоит в использовании элементов логики, которые позволяют записывать не только отдельные формулы, но и любые алгоритмы, не прибегая к промежуточным словесным объяснениям. Это и делает язык особенно удобным для программирования задач, понимая под программой такую запись алгоритма в символах, чтобы машина могла сама преобразовать ее в рабочую программу с помощью транслятора. Но это не единственное назначение алгоритмического языка. *На этом языке можно думать*, он может быть с успехом использован для обучения алгоритмам и для обмена информацией (краткой или развернутой) между инженерами и учеными. В частности, алгоритмический язык можно использовать как средство обучения студентов. Недалеко то время, когда написание статьи с использованием не только математического, но и алгоритмического языка станет привычным и обычным делом.

Для сложных алгоритмов описание вычислительных процессов без применения алгоритмического языка может оказаться вообще практически невозможным. Уже сейчас специалисты сталкиваются с трудностями на этой почве, а слабое внедрение многих замечательных методов в значительной мере объясняется тем, что они не доведены их авторами до вычислительного алгоритма.

В различных странах создан ряд алгоритмических языков «местного» значения: ФОРТРАН, ИТ, Мат-Матик в США, операторный и адресный языки в СССР и др.

В настоящее время на основе этих языков создан международный язык АЛГОЛ-60, представляющий как бы «эсперанто» алгоритмических языков.

Отметим, что упомянутый выше язык инженерного программирования также является частным случаем АЛГОЛа, приспособленным к выполнению расчета на малых вычислительных машинах. Адресный язык института кибернетики АН УССР [19] — весьма близкий предшественник АЛГОЛа.

Сходство между алгоритмическими языками, даже если они образовались независимо, объясняется общностью целей, которые ставят их создатели, что приводит, в конечном счете, к аналогичным закономерностям. Отличия обычно заключаются в принятых обозначениях и степени развития языка. АЛГОЛ-60 является

наиболее развитым из существующих алгоритмических языков, в его создании принимали участие многие видные ученые, в том числе и авторы предшествовавших языков. Официальная редакция языка была принята на международной конференции в Париже (этому предшествовал целый ряд подготовительных конференций и совещаний в Европе и США) и дополнена затем на Римском совещании 1962 г. В настоящее время как у нас, так и за рубежом продолжается работа по дальнейшему совершенствованию языка *. Известные трансляторы, созданные в СССР, построены на основе языка АЛГОЛ-60.

1. 21. Теме алгоритмических языков посвящен ряд работ автора. В работе [37], написанной совместно с Л. Г. Дмитриевым, алгоритмические языки (эскизный, адресный, АЛГОЛ-60) используются для формулировки алгоритмов расчета разнообразных пространственных конструкций — пологих оболочек, пластин, висячих и стержневых систем.

Последующая книга [109] целиком посвящена языку АЛГОЛ-60, его применению в строительной механике, разработке правил построения программ на алгоритмическом языке (своего рода алгоритма для построения алгоритмов).

Для изучения языка АЛГОЛ-60 в полном объеме требуется определенное время. Как уже указывалось, это развитый язык с формализацией пунктуации и синтаксиса, более сложный, чем предшествующие языки, что затрудняет быстрое внедрение языка в практику.

Поэтому целесообразно выделить более простую форму АЛГОЛа, названную основной, усваиваемую достаточно быстро и являющуюся вместе с тем универсальной, пригодной для любых задач. После усвоения основной формы можно перейти к изучению расширенной формы языка. Принципиальной разницы между этими двумя формами нет, отличия не качественные, а количественные — в расширенной форме применяются дополнительные средства, с помощью которых можно формулировать программы более кратко, чем в основной форме. Нужно сказать, что эта краткость формулировки, делающая программу на алгоритмическом языке более наглядной и обозримой и облегчающая ее чтение, не приводит к сокращению рабочей программы, полученной с транслятора. Наоборот, трансляция программы,

* В последних номерах «АЛГОЛ-бюллетеня» (этот журнал был основан в Копенгагене в 1959 г.) обсуждается проект нового издания АЛГОЛа, а также упрощенного варианта этого языка.

написанной более длинно в основной форме, нередко дает более краткую рабочую программу.

1. 22. Эффективная программа для ЭЦВМ должна удовлетворять следующим общим требованиям:

- выражать алгоритм задачи с помощью минимального количества команд и, следовательно, занимать сравнительно небольшой объем оперативной памяти, т. е. длина программы должна быть минимальной. В некоторых случаях целесообразно разбить программу на несколько независимых частей, поместить их во внешней памяти (барабанах или лентах) и затем по мере необходимости вызывать по одной в оперативную память;

- обладать свойством цикличности. Это требование для ЭЦВМ более сильное, чем для электрических автоматов, так как от него зависит выполнение предыдущего пункта;

- обеспечивать экономное размещение материала в оперативной памяти путем замещения одних чисел другими в тех же ячейках, исключения элементов матриц, постоянно остающихся нулевыми, использования структурных особенностей матриц и т. д. Для этого применяются специальные приемы рациональной организации памяти машины;

- предусматривать минимальный ввод чисел, заменяя (в тех случаях, когда это возможно) ввод матрицы вводом краткой, но исчерпывающей информации о ней; нужно стремиться уменьшить объем вводимого числового материала для больших задач, так как подготовка данных в пригодном для машины виде (перфокарты) занимает относительно много времени и требует тщательного контроля;

- обеспечивать комплексную автоматизацию расчета, охватывая все его стадии. Подготовка исходных данных должна быть простой и либо полностью исключать ручной счет, либо сводить его к минимуму. Первое предпочтительно, так как подготовительные расчеты, проведенные вручную, даже самые простейшие, резко снижают эффект механизации и увеличивают вероятность ошибок;

- для использования внешней памяти (барабанов, лент) в алгоритме следует предусмотреть возможность деления числового материала на блоки и последовательного вызова их в оперативную память. Блоки должны быть по возможности крупными, чтобы уменьшить количество обменов между оперативной и внешней памятью;

- быть типовой, т. е. описывать решение не одной какой-либо задачи, а целого класса задач. Нередко бывает так, что добав-

лением небольшого количества приказов мы делаем программу более общей. В этом случае целесообразно идти на некоторое (небольшое) увеличение программы, разветвляя процесс для двух-трех обширных групп;

обеспечивать контроль вычислений. Вследствие быстроты счета проверки не следует дробить на построчные (как это делается в электрических автоматах), а охватывать ими сразу большие массивы чисел.

Для того, чтобы удовлетворять этим требованиям, нужно правильно понимать принципиальные возможности ЭЦВМ и иметь представление об их общих свойствах, изложенных в известных учебных курсах, детали знать необязательно. Однако все же следует учитывать для какого типа машин составляется программа. Для существующих ЭЦВМ характерна разнотипность, они отличаются (иногда очень резко) быстродействием, способом задания и выдачи информации, адресностью, объемом оперативной памяти, составом и видом внешних запоминающих устройств, режимом работы (плавающая или фиксированная запятая) и т. д. Поэтому, хотя и можно говорить о требованиях к алгоритмам для электронных вычислительных машин вообще, но необходимо учитывать, что они изменяются для различных типов машин.

Например, если машина обладает большим быстродействием, но малой памятью, то на первый план выдвигаются вопросы рациональной организации памяти и для большей ее экономии можно идти даже на значительное увеличение количества вычислений (в некоторых случаях даже в два-три раза). Как видим, подход принципиально другой, чем при использовании клавишных автоматов.

Некоторые математические методы, весьма изящные и удобные для самого процесса программирования, могут оказаться неприемлемыми для машин такого типа.

1. 23. В этом отношении весьма показательное рассмотрение матричного аппарата как средства для программирования. Матричные методы в строительной механике достигли большого развития [25, 27, 68, 92, 93, 127, 130, 136, 137, 141]. Матричный язык представляет также значительные удобства при формулировке задач для электронных вычислительных машин. Для матричных операций созданы стандартные подпрограммы и на их основе матричная интерпретирующая система, которая позволяет весьма кратко записывать программы с помощью матричных команд. Например, в системе МАК, разработанной в ГИПРОТИСе [81],

имеются команды сложения, вычитания, умножения, обращения матриц и др., закодированные по аналогии с обычными арифметическими операциями кодами 01, 02, 03, 04 и т. д. Однако при практическом применении матричных программ для решения задач большого объема возникают серьезные трудности, связанные с тем, что матричные системы рассчитаны на прямоугольную форму матриц, а это вынуждает резервировать в оперативной памяти машины прямоугольные массивы. Это значит, что для матриц строительной механики, для которых характерна малая заполненность, приходится постоянно хранить в памяти огромное количество нулевых элементов. Поэтому большая задача нередко не вмещается в оперативную память, ее приходится разбивать на части и решать последовательно, обрабатывая вручную промежуточные массивы и вновь вводя в машину. Конечно, возможно привлечение внешней памяти, но это усложняет программу и замедляет счет*.

Отсюда следует, что при составлении оптимальных алгоритмов для ЭЦВМ, имеющих ограниченную память, необходимо вносить существенные изменения в матричные операции.

Нужно иметь в виду, что память машины линейная, а не матричная. Поэтому в трансляторе предусмотрен перевод многомерных массивов в одномерные. Но можно с самого начала при составлении программы исходить из линейного расположения чисел, разворачивая матрицу по строкам или графам. Это дает возможность учесть любые структурные особенности матриц (например, ленточную или ленточную окаймленную структуру) и таким образом исключить нулевые элементы, попросту не нумеруя их. Этой же цели служат логические шкалы. Для уменьшения количества вводимых чисел вместо матрицы вводится краткая информация о ней, а затем уже машиной по специальной программе последовательно формируются элементы матриц, участвующие в операциях. Преимущество таких программ, позволяющих размещать в небольшой памяти весьма большие матрицы, несомненно.

Для машин, имеющих оперативную память и быстродействие примерно на два порядка выше указанных в I. 19, матричные

* При больших матрицах наиболее надежное внешнее запоминающее устройство — барабаны не всегда достаточно. В самом деле, для помещения матрицы, например, двухсотого порядка, требуется $200 \times 200 = 40000$ ячеек. В этом случае приходится использовать магнитные ленты, что сильно замедляет счет и возможно не на всех машинах.

программы могут оказаться приемлемыми для относительно больших задач*.

Применение ЭЦВМ неразрывно связано с организацией, упорядочением числовых массивов — исходных, промежуточных и результатов. Это упорядочение соответствует преобразованиям линейной алгебры, но не обязательно должно проводиться в традиционной матричной форме — ее часто приходится изменять, чтобы иметь возможность оптимальным образом учесть характер переработки информации в машине и особенности структур преобразуемых матриц.

1. 24. Из изложенного видно, что составление алгоритмов и программ для ЭЦВМ требует особого подхода. Алгоритмы расчета, разработанные для клавишных автоматов, хотя и могут быть в неизменном виде реализованы на ЭЦВМ, но это, как правило, нецелесообразно. Компактные схемы, столь удобные для клавишных автоматов, не имеют преимуществ для ЭЦВМ. Разница между компактной и обычной схемами счета (например, компактной и обычной схемами Гаусса для решения систем уравнений) будет состоять лишь в характере записи чисел в ячейках памяти: в первом случае суммы произведений будут накапливаться в дополнительных рабочих ячейках, во втором — во всех ячейках матрицы. В том и другом случаях запись производится автоматически, количество вычислений одинаковое.

Число необходимых ячеек при обычной схеме счета не увеличивается (даже несколько уменьшается благодаря исключению дополнительных рабочих ячеек), а алгоритм решения упрощается. Поэтому при использовании ЭЦВМ компактные схемы обычно не нужны.

В ряде научно-исследовательских, учебных и проектных институтов страны ведутся обширные работы по созданию оптимальных алгоритмов и программ для решения задач строительной механики на ЭЦВМ. Институт ГИПРОТИС представляет собой центр, координирующий эти работы.

* Такие мощные машины уже существуют. В работе [8] даны сведения о ряде зарубежных машин, имеющих оперативную память в несколько сот тысяч ячеек и выполняющих свыше 1 млн. вычислений в секунду. Аналогичные данные имеет отечественная вычислительная машина БЭСМ-6. Все же, даже при использовании таких уникальных машин экономная организация памяти потребуется при решении весьма больших задач: расчете пространственных систем (если отказаться от упрощений), решении дифференциальных уравнений в частных производных методом сеток [9] и т. д. По мере уточнения расчетных предположений такие задачи будут возникать все чаще.

В КиевЗНИИЭПе под руководством автора была создана библиотека программ для расчета плоских и пространственных систем различных видов. Для развития этой библиотеки используются встречные дедуктивный и индуктивный процессы, о которых говорилось в предисловии, и эффективные приемы задания информации, в частности, кодирование зрительного образа зданий и сооружений на плоских сетках и пространственных решетках.

Корнями деревьев указанной библиотеки являются универсальные программы методов сил и деформаций для расчета дискретных систем, т. е. систем из произвольных элементов (в том числе непрерывных), соединенных дискретными связями. В библиотеку легко вписываются разнообразные предложения по расчету пространственных конструкций.

1. 25. Таким образом рассмотрены закономерности развития алгоритмов строительной механики и расширения их возможностей в связи с совершенствованием математических машин. Каждая из рассмотренных групп алгоритмов имеет определенную область рационального применения; границы этой области, однако, со временем изменяются.

Еще преждевременно объявлять устаревшими алгоритмы, приспособленные для ручного исполнения, так как они имеют широкое применение. Такие разговоры пока беспредметны, однако тенденция наметилась вполне определенная: постепенное расширение автоматизации инженерных расчетов при проектировании и соответствующее сокращение работ, выполняемых вручную.

Параллельно с совершенствованием и распространением вычислительных машин неизбежно будет происходить перестройка системы алгоритмов строительной механики: одни устареют и будут забыты, другие, наоборот, получают толчок для развития и модернизации.

$$\begin{aligned} x_1^{(v+1)} &= -\frac{1}{a_{11}} (a_{12}x_2^{(v)} + a_{13}x_3^{(v)} + \dots + a_{1n}x_n^{(v)} + a_{1q}); \\ x_2^{(v+1)} &= -\frac{1}{a_{22}} (a_{21}x_1^{(v)} + a_{23}x_3^{(v)} + \dots + a_{2n}x_n^{(v)} + a_{2q}); \\ . &. \\ x_n^{(v+1)} &= -\frac{1}{a_{nn}} (a_{n1}x_1^{(v)} + a_{n2}x_2^{(v)} + \dots + a_{n,n-1}x_{n-1}^{(v)} + a_{nq}), \end{aligned} \quad (3)$$

$$x_2^{(v+1)} = - \frac{1}{a_{22}} (a_{21}x_1^{(v)} + a_{23}x_3^{(v)} + \dots + a_{2n}x_n^{(v)} + a_{2q});$$

$$x_n^{(v+1)} = -\frac{1}{a_{nn}} (a_{n1}x_1^{(v)} + a_{n2}x_2^{(v)} + \dots + a_{n,n-1}x_{n-1}^{(v)} + a_{nq}),$$

где ν — номер приближения.

В качестве начального, нулевого ($\nu=0$) приближения можно принять все неизвестные равными нулю и получить первые приближения ($\nu+1=1$) для неизвестных:

$$x_1^{(1)} = -\frac{a_{1q}}{a_{11}}; \quad x_2^{(1)} = -\frac{a_{2q}}{a_{22}}; \quad \dots; \quad x_n^{(1)} = -\frac{a_{nq}}{a_{nn}}.$$

Далее подставляя в правую часть формул (3) значения неизвестных, равные первым приближениям ($\nu=1$), получают вторые приближения; подставляя вторые приближения ($\nu=2$), получают третьи приближения и так до тех пор, пока величины неизвестных в ν и $\nu+1$ -ом приближениях не станут равны (в пределах заданной точности).

При итерации по Зейделю, каждое найденное неизвестное сразу же включается в итерационный процесс, что для задач строительной механики приводит к более быстрой сходимости. В этом случае последовательные приближения определяются по формулам:

$$\left. \begin{aligned} x_1^{(v+1)} &= -\frac{1}{a_{11}} (a_{12}x_2^{(v)} + a_{13}x_3^{(v)} + \dots + a_{1,n-1}x_{n-1}^{(v)} + \\ &\quad + a_{1,n}x_n^{(v)} + a_{1q}); \\ x_2^{(v+1)} &= -\frac{1}{a_{22}} (a_{21}x_1^{(v+1)} + a_{23}x_3^{(v)} + \dots + a_{2,n-1}x_{n-1}^{(v)} + \\ &\quad + a_{2,n}x_n^{(v)} + a_{2q}); \\ x_3^{(v+1)} &= -\frac{1}{a_{33}} (a_{31}x_1^{(v+1)} + a_{32}x_2^{(v+1)} + \dots + a_{3,n-1}x_{n-1}^{(v)} + \\ &\quad + a_{3,n}x_n^{(v)} + a_{3q}); \\ &\vdots \\ x_n^{(v+1)} &= -\frac{1}{a_{nn}} (a_{n1}x_1^{(v+1)} + a_{n2}x_2^{(v+1)} + \dots + a_{n,n-1}x_{n-1}^{(v+1)} + a_{nq}). \end{aligned} \right\} (4)$$

$$x_2^{(v+1)} = -\frac{1}{a_{22}} (a_{21}x_1^{(v+1)} + a_{23}x_3^{(v)} + \dots + a_{2,n-1}x_{n-1}^{(v)} + a_{2,n}x_n^{(v)} + a_{2g});$$

$$x_3^{(v+1)} = -\frac{1}{a_{33}} (a_{31}x_1^{(v+1)} + a_{32}x_2^{(v+1)} + \dots + a_{3,n-1}x_{n-1}^{(v)} + a_{3,n}x_n^{(v)} + a_{3\sigma});$$

$$x_n^{(v+1)} = -\frac{1}{a_{nn}} (a_{n1}x_1^{(v+1)} + a_{n2}x_2^{(v+1)} + \dots + a_{n, n-1}x_{n-1}^{(v+1)} + a_{nq}).$$

Принимая начальные ($v=0$) значения неизвестных равными нулю, получим первые приближения:

$$x_1^{(1)} = -\frac{a_{1q}}{a_{11}}; \quad x_2^{(1)} = -\frac{1}{a_{22}} (a_{2q} + a_{21}x_1^{(1)});$$

$$x_3^{(1)} = -\frac{1}{a_{33}} (a_{3q} + a_{31}x_1^{(1)} + a_{32}x_2^{(1)}) \text{ и т. д.}$$

Подставляя в правую часть формулы (4) первые приближения ($v=1$), найдем второе приближение и т. д., пока не будет удовлетворено условие $|x_i^{(v+1)} - x_i^{(v)}| \leq \varepsilon$, где ε — допустимая погрешность расчета.

Процесс можно вести не в приближениях, а в поправках, т. е. вместо второго приближения находить первую поправку к первому приближению, вместо третьего приближения — вторую поправку и т. д. до тех пор, пока поправки не станут близки к нулю. Искомые значения неизвестных равны первым приближениям и сумме последующих поправок.

Процесс в приближениях обладает свойством самоисправляемости, так как в качестве исходных, начальных приближений могут быть приняты любые значения неизвестных; в результате последовательных итераций они будут уточнены. Чем грубее начальные значения неизвестных, тем больше потребуется выполнить итераций для их уточнения. Поэтому, если где-либо в вычислениях была допущена ошибка, то она приведет лишь к затягиванию процесса вычислений, но не отразится на конечных результатах (разумеется, в том случае, если процесс итераций сходится).

Процесс в поправках не обладает свойством самоисправляемости, но имеет другие достоинства: для быстросходящихся процессов поправки быстро уменьшаются и их можно вычислять с небольшой точностью на линейке, а последние поправки даже в уме; вычисления со свободными членами выполняются только один раз в первом приближении, благодаря чему уменьшается количество операций. Проф. Б. Н. Жемочкин [42] рекомендует процесс вести в поправках, а для контроля последние две итерации — в приближениях.

II. 2. Изменение неизвестных в процессе последовательных приближений очень напоминает поведение итеративных систем, исследуемых в работах по кибернетике. Сходство прежде всего состоит в том, что для сходящихся процессов неизвестные системы уравнений изменяются динамически, во времени, и от шага

к шагу стремятся к некоторому устойчивому состоянию. Происходит как бы поиск цели, т. е. состояния, в котором значения неизвестных остаются постоянными, что является искомым решением.

Эшби [133] приводит следующую схему взаимодействия организма и среды (рис. 1). Реакция организма R на воздействия внешней среды должна быть такова, чтобы обеспечить сохранение в определенных границах существенных переменных, имеющих жизненно важные функции. Совокупность существующих переменных на рис. 1 обозначена циферблатом со стрелкой и двумя отметками, означающими допустимые границы их изменения. Между организмом и средой существует обратная связь. Кроме того, существует вторая обратная связь через S — ступенчатые механизмы, вырабатывающие параметры (входные величины), определяющие реакцию организма, если существенные переменные выходят из допустимых границ.

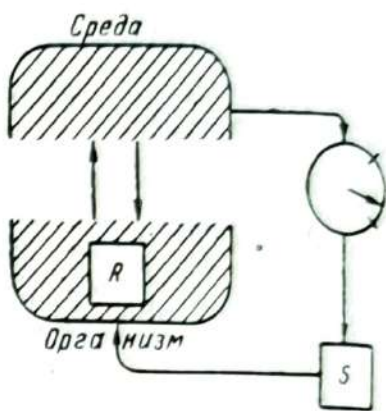


Рис. 1

Вторая обратная связь обеспечивает адаптацию (приспособляемость) организма к окружающей среде. Предполагается, что среда имеет тип черного ящика с неизвестными вначале свойствами; необходимые сведения о среде организм получает методом проб и ошибок, которые последовательно исправляются. Закрепляются такие формы поведения, которые приводят к достижению цели — удержанию существенных переменных в физиологических пределах. Систему с таким поведением называют ультраустойчивой.

Рассматривая системы с большим количеством неизвестных и связей между ними, Эшби показывает, что для таких систем процесс достижения Большого Успеха (Большим Успехом он называет достижение конечной цели — состояния равновесия) через частичные успехи может потребовать слишком много времени и организм погибнет раньше, чем это произойдет *. Отсюда делается вывод: «Для того, чтобы метод ультрастабильности приводил к успеху в достаточно короткое время, необходимо, чтобы частичные успехи закреплялись. А для этого необходимо,

*Ориентировочно, для сравнительной оценки можно считать, что для полностью связанной системы количество проб и ошибок близко к $2N$, где N — число неизвестных.

чтобы определенные части системы не сообщались с определенными другими частями или не влияли на них» [133, стр. 229].

Если система состоит из частей, обладающих большой внутренней связностью, но соединенных относительно слабыми связями, то адаптация может быть достигнута относительно быстро «благодаря частичному, временному «выключению» зависимостей внутри целого, так что целое становится комплексом подсистем, внутри которых связи сильно развиты, но между которыми они более ограничены» (там же, стр. 232).

II. 3. Рассмотрим с этой точки зрения решение систем уравнений, выражающих в той или иной форме зависимости реального мира.

Средой в данном случае являются коэффициенты системы уравнений, реагирующей частью организма R — алгоритм, применяемый для решения системы; роль механизма S , вырабатывающего параметры для R , выполняет человек, выбирающий такой алгоритм, чтобы в результате его применения невязки, получаемые при подстановке неизвестных в уравнения и играющие роль существенных переменных, не превосходили заданных пределов.

Будем считать также, что система состоит из элементарных подсистем. В простейшем случае каждая подсистема имеет одно неизвестное с коэффициентом на главной диагонали. Связи между подсистемами выражаются с помощью побочных коэффициентов. При решении задачи методом итерации процесс полностью соответствует описанному Эшби — Большой Успех достигается путем накопления частичных успехов, состоящих в последовательном уравнивании элементарных подсистем. Процесс продолжается до тех пор, пока система в целом не придет в равновесие — все существенные переменные (невязки) окажутся в заданных пределах, а неизвестные практически перестанут изменяться.

Как известно, итерационный процесс быстро сходится, если в каждом уравнении системы главный коэффициент по абсолютному значению существенно больше суммы побочных коэффициентов, иными словами, главная связь значительно сильнее побочных связей. Если это условие не соблюдается, но выполняются другие достаточные условия сходимости, рассматриваемые в линейной алгебре, то процесс будет сходиться медленно.

Для ускорения процесса потребуется применить механизмы S — изменить алгоритм решения. Эти вопросы будут рассмотрены в VI. 11.

Пока будем предполагать, что система такова, что данный алгоритм обеспечивает быструю сходимость.

II. 4. Способ перераспределения начальных значений неизвестных по существу совпадает с рассмотренной в II. 3. трактовкой решения систем линейных уравнений.

Этот способ в применении к расчету рамных систем имеет следующие особенности:

1. Вместо системы уравнений пользуются следующими формулами для определения величин неизвестных:

$$x_i = x_i^0 + \sum_m x_m k_{im}; \quad (1)$$

$$x_i^0 = - \frac{a_{iq}}{a_{ii}}; \quad (2)$$

$$k_{im} = - \frac{a_{im}}{a_{ii}}. \quad (3)$$

Индекс i последовательно принимает значения $1, 2, \dots, n$. Индекс m в общем случае принимает все значения $(1, 2, \dots, n)$, кроме $i=m$, в частных случаях — только некоторые значения, зависящие от i .

Нетрудно заметить, что элементарные формулы (1), (2) и (3) заменяют систему уравнений (2) из II. 1. Для определенных типов статически неопределимых систем в формулы (1) — (3) вместо коэффициентов входят жесткостные характеристики стержней.

Таким образом, вместо уравнений используются готовые формулы, представляющие часть решения системы уравнений, выполненную в общем виде.

2. Дается наглядная статическая трактовка величин, входящих в формулы (1) — (3). Величины x_i^0 названы начальными значениями неизвестных — это величины неизвестных усилий или деформаций в элементарных один раз статически неопределимых системах от внешней нагрузки. Величины k_{im} названы коэффициентами передачи — это величины неизвестных усилий или деформаций в элементарных системах от единичных усилий или деформаций смежных узлов. Заметим, что они совпадают с обратными фокусными отношениями для элементарных систем. Коэффициенты передачи определяются на схеме статически неопределимой системы.

3. Процесс итерации трактуется как перераспределение величин неизвестных в процессе последовательного снятия (при ме-

тоде деформаций) или восстановления (при методе сил) связей основной системы и разложения сложной системы на элементарные подсистемы. Статический подход к задаче дает возможность получить решение, даже не зная вида исходной системы уравнений, и построить в некоторых случаях интересные итерационные схемы (особенно это относится к смешанным алгоритмам, о кото-

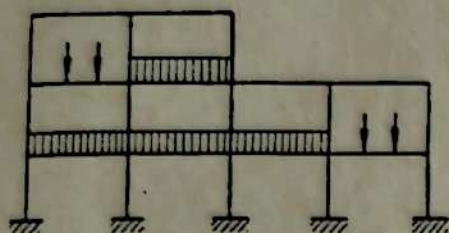


Рис. 2.

рых будет сказано дальше, в II. 13). Циклическая последовательность устранения (восстановления) связей — $1, 2, \dots, n, 1, 2, \dots, n, \dots$ соответствует итерации по Зейделю. Нами была предложена [96] непрерывная последовательность $1, 2, \dots, n-1, n, n-1, \dots, 2, 1, 2, \dots$, являющаяся логическим продолжением идеи способа Зейделя.

4. Алгоритм расчета представляет собой инструкцию в виде последовательности словесных указаний к числовому примеру, характерному для задач определенного узкого класса. Для удобства и наглядности расчет выполняется частично на схеме сооружения, частично в таблице, повторяющей эту схему.

Руководство по расчету рам и неразрезных балок методом перераспределения начальных значений неизвестных состоит из ряда таких алгоритмов — инструкций для типовых примеров. Эти алгоритмы предназначены для инженера и не носят еще формализованного характера*.

II. 5. Покажем построение подобного алгоритма на примере расчета многоэтажной многопролетной рамы на вертикальную нагрузку (рис. 2).

Ввиду большой жесткости этой рамы в горизонтальном направлении можно принять для расчета на вертикальную нагрузку, что линейные смещения узлов равны нулю, иными словами считать раму «несвободной». Основная система метода деформаций с наложенными защемлениями изображена на рис. 3.

Расчет методом итераций эквивалентен последовательному снятию и наложению защемлений и разложению исходной рамы на ряд элементарных подсистем «крестообразных» рам. Устраняя защемление из узла б, получаем элементарную систему (рис.

* Необходимость в формализации возникает при переходе к механизированному расчету на цифровых машинах.

4, а), рассчитав которую на внешнюю нагрузку, находим x_6 — угол поворота шестого узла. Зафиксируем найденный угол поворота при помощи заземления и удалим заземление из следующего узла 7. Рассчитав элементарную систему (рис. 4, б) на действие внешней нагрузки и заданного угла поворота x_6 , найдем x_7 . Далее расчет продолжается аналогично в непрерывной последовательности.

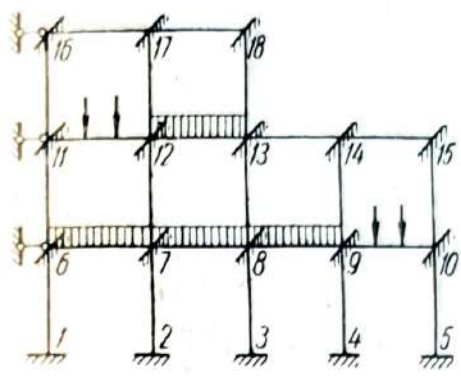


Рис. 3.

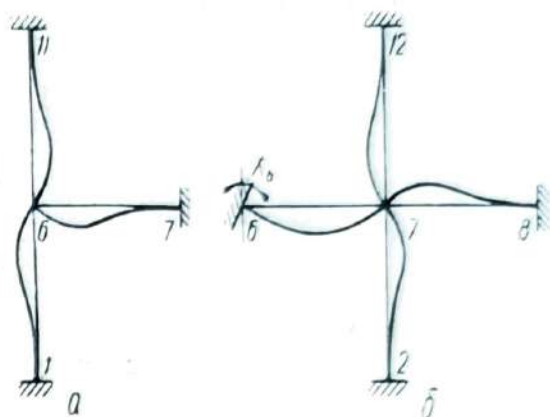


Рис. 4.

Расчетные формулы следующие:

$$x_i^0 = - \frac{2 \sum M_{im}^p}{R_i^0}; \quad (1)$$

$$k_{im} = - \frac{T_{im}}{R_i^0}; \quad (2)$$

$$R_i^0 = 2 \sum T_{im}; \quad (3)$$

$$T_{im} = \frac{4EI}{l_{im}}, \quad (4)$$

где x_i^0 — угол поворота узла i ;

M_{im}^p — момент в заземлении i стержня $i-m$ от внешней нагрузки;

R_i^0 — жесткость узла.

Суммирование в формулах (1) и (3) относится ко всем стержням, сходящимся в узле.

k_{im} — коэффициент передачи угла поворота;

T_{im} — жесткость стержня, представляющая собой момент в заземлении i от угла поворота $x_i = 1$ (на противоположном конце при этом возникает момент равный $0,5 T_{im}$);

EI — жесткость поперечного сечения стержня на изгиб;
 E — модуль упругости материала;
 I — момент инерции поперечного сечения;
 l_{st} — длина стержня.

Данные формулы относятся к наиболее частному случаю, когда стержень рамы имеет постоянное по длине сечение. Для

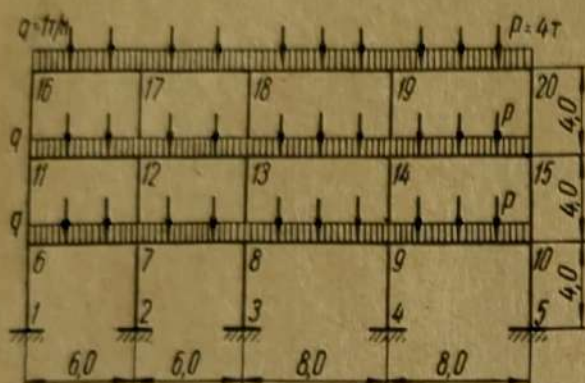


Рис. 5.

стержня ступенчато-переменного сечения следует пользоваться видоизмененными формулами и известными таблицами (см. [97], гл. II).

После того, как углы поворота узлов системы найдены, моменты на концах каждого стержня определяются (и выписываются на схеме рамы) по формулам:

$$\left. \begin{aligned} M_{ab} &= M_{ab}^p + M'_{ab} + M''_{ab}; \\ M_{ba} &= M_{ba}^p + M'_{ba} + M''_{ba}, \end{aligned} \right\} \quad (5)$$

где

$$M'_{ab} = x_a T_{ab}; \quad M'_{ba} = x_b T_{ab}; \quad (6)$$

$$M''_{ab} = \frac{1}{2} M'_{ba}; \quad M''_{ba} = \frac{1}{2} M'_{ab}. \quad (7)$$

Проследим алгоритм расчета на числовом примере.

Рассчитать раму, изображенную на рис. 5.

Нагрузки: $q = 1$ т/м, $P = 4$ т.

Относительные значения жесткостей T :

для ригелей $T-4$;

для стоек 1-го и 2-го этажей $T-2$;

для стоек 3-го этажа $T-1$.

Решение

1. На стержнях схемы рамы (вычерченной не в масштабе) записываем (в скобках) относительные значения жесткостей T , а в узловых кружках — их удвоенные суммы R^0 (рис. 6).

2. Разделив жесткость T каждого из стержней, сходящихся в узле, на R^0 этого узла, определяем коэффициенты передачи k и записываем их в заглавные строки табл. II. 1 на схемах узлов (таблица повторяет собой схему рамы). Знаки минус опускаем.

Проверкой является вычисление суммы коэффициентов k для всех стержней, сходящихся в узле. Эта сумма должна равняться 0,5.

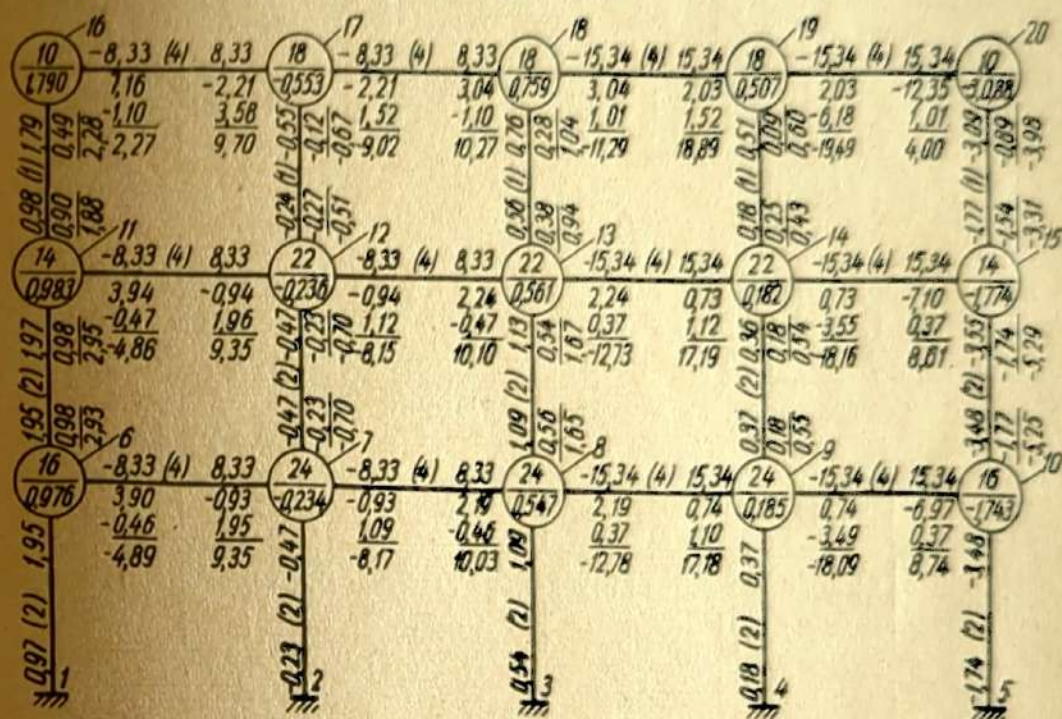


Рис. 6.

3. Определяем реактивные моменты в защемлениях M^p и записываем их на стержнях рамы у узлов (см. рис. 6):

$$-M_{67}^p = M_{76}^p = \frac{1 \cdot 6^2}{12} + 0,222 \cdot 4 \cdot 6 = 3,0 + 5,33 = 8,33;$$

$$-M_{89}^p = M_{98}^p = \frac{1 \cdot 8^2}{12} + 0,313 \cdot 4 \cdot 8 = 5,33 + 10,01 = 15,34.$$

Для остальных стержней значения M^p повторяются.

4. Определяем начальные углы поворота и записываем их в строке 0 табл. II. 1.

$$x_6^0 = \frac{2 \cdot 8,33}{16} = 1,04; \quad x_{11}^0 = \frac{2 \cdot 8,33}{14} = 1,183;$$

$$x_{16}^0 = \frac{2 \cdot 8,33}{16} = 1,04; \quad x_7^0 = x_{12}^0 = x_{17}^0 = 0;$$

$$x_8^0 = \frac{-2(8,33 - 15,34)}{24} = \frac{14,02}{24} = 0,584;$$

$$x_{13}^0 = \frac{14,02}{22} = 0,637; \quad x_{18}^0 = \frac{14,02}{18} = 0,780;$$

$$x_9^0 = x_{14}^0 = x_{19}^0 = 0; \quad x_{10}^0 = -\frac{2 \cdot 15,34}{16} = -1,918;$$

$$x_{15}^0 = -\frac{2 \cdot 15,34}{14} = -2,191; \quad x_{20}^0 = -\frac{2 \cdot 15,34}{10} = -3,064.$$

5. Производим перераспределение начальных углов поворота в табл. II. 1, применяя непрерывную последовательность обхода узлов. При передаче угла поворота знак изменяется на обратный. Порядок составления таблицы следующий.

Таблица II. 1

16	17	18	19	20	
$\begin{array}{l} -0,4 \\ \downarrow \\ 0,1 \end{array}$	$\begin{array}{l} 0,222 \uparrow 0,222 \\ 0,056 \end{array}$	$\begin{array}{l} 0,222 \uparrow 0,222 \\ 0,056 \end{array}$	$\begin{array}{l} 0,222 \uparrow 0,222 \\ 0,056 \end{array}$	$\begin{array}{l} 0,4 \downarrow \\ 0,1 \end{array}$	Номер строки
$\begin{array}{l} 1666 \\ -103 \\ \rightarrow \end{array}$	$\begin{array}{l} - \\ 10 \\ -348 \end{array}$	$\begin{array}{l} 780 \\ -34 \\ 75 \end{array}$	$\begin{array}{l} - \\ 6 \\ -182 \end{array}$	$\begin{array}{l} -3064 \\ 189 \\ 70 \end{array}$	0
1563	-338	821	-176	-2805	1
199	-160	-100	624	$\leftarrow$	2
$\begin{array}{l} 5 \\ \rightarrow \end{array}$	$\begin{array}{l} 4 \\ -45 \end{array}$	$\begin{array}{l} 3 \\ 45 \end{array}$	$\begin{array}{l} -15 \\ 17 \end{array}$	$\begin{array}{l} -10 \\ -248 \end{array}$	3
20	-8	-12	57	$\leftarrow$	4
$\begin{array}{l} 1 \\ \rightarrow \end{array}$	$\begin{array}{l} - \\ -5 \end{array}$	$\begin{array}{l} - \\ 3 \end{array}$	$\begin{array}{l} -1 \\ 2 \end{array}$	$\begin{array}{l} -2 \\ -23 \end{array}$	5
2	-	-1	5	$\leftarrow$	6
1790	-553	759	507	-3088	Σ

11	12	13	14	15	
0,071 +0,286 0,143	0,045 0,182+0,182 0,091	0,045 0,182+0,182 0,091	0,045 0,182+0,182 0,091	0,071 0,286+ 0,143	Номер строки
1183 -149 —→	— 16 -188	637 -56 31	— 9 -111	-2191 271 29	0
1034	-172	612	-102	-1891	1
-126 71	23 -99	-33 -34	-20 308	200 ←—	2
9 —→	5 8	6 11	-25 9	-20 -78	3
-2 -3	2 -3	-2 1	-2 15	18 ←—	4
—→	—	—	-1	-3	5
983	-236	561	182	-1774	Σ
6	7	8	9	10	
0,125 +0,250 0,125	0,083 0,167+0,167 0,083	0,083 0,167+0,167 0,083	0,083 0,167+0,167 0,083	0,125 0,250+ 0,125	Номер строки
1040 —→	— -173	584 29	— -102	-1918 25	0
1040	-173	613	-102	-1893	1
-122 60	21 -90	-45 -27	-15 281	212 ←—	2
—→	10	10	10	-69	3
— -2	-1 -1	-1 -3	— 10	10 ←—	4
—→	—	—	1	-3	5
976	-234	547	185	-1743	Σ

Примечание. Значения x умножены на 1000, знаки минус при x опущены.

Прямой ход. Обходим узлы в порядке возрастания их номеров, углы поворота передаем направо и вверх:

с узла 6 на узлы 7 и 11 $x_7 = -1040 \cdot 0,167 = -173$;

$$x_{11} = -1040 \cdot 0,143 = -149;$$

» 7 » 8 и 12 $x_8 = 173 \cdot 0,167 = 29$;

$$x_{12} = 173 \cdot 0,091 = 16;$$

» 8 » 9 и 13 $x_9 = -613 \cdot 0,167 = -102$;

$$x_{13} = -613 \cdot 0,091 = 56 \text{ и т. д.}$$

Определив первые приближения для углов поворота, запишем их в строке 1.

В дальнейшем будем определять поправки для углов поворота, поэтому вычисляемые суммы далее заносить в таблицу не будем.

Обратный ход. Передаем углы поворота, записанные в строке 1 и ниже, налево и вниз:

с узла 20 на узлы 19 и 15 $x_{19} = 2805 \cdot 0,0222 = 624$;

$$x_{15} = 2805 \cdot 0,071 = 200;$$

» 19 » 18 и 14 $x_{18} = -(-176 + 624) 0,222 = -100$;

$$x_{14} = -(-176 + 624) 0,045 = -20;$$

» 18 » 17 и 13 $x_{17} = -(821 - 100) 0,222 = -160$;

$$x_{13} = -721 \cdot 0,045 = -33 \text{ и т. д.}$$

Заканчивая обратный ход, проводим линию.

Прямой ход. Передаем углы поворота, записанные в строке 2 и ниже, направо и вверх:

с узла 6 на узлы 7 и 11 $x_7 = -(-122 + 60) 0,167 = 10$;

$$x_{11} = -(-122 + 60) 0,143 = 9;$$

» 7 » 8 и 12 $x_8 = -(21 - 90 + 10) 0,167 = 10$;

$$x_{12} = 59 \cdot 0,091 = 5;$$

» 8 » 9 и 13 $x_9 = -(-45 - 27 + 10) 0,167 = 10$;

$$x_{13} = 62 \cdot 0,091 = 6 \text{ и т. д.}$$

Заканчивая прямой ход, проводим линию.

Обратный ход. Передаем углы поворота, записанные в строке 3 и ниже, налево и вниз:

с узла 20 на узлы 19 и 15 $x_{19} = -(-10 - 248) 0,222 = 57$;

$$x_{15} = 258 \cdot 0,071 = 18;$$

» 19 » 18 и 14 $x_{18} = -(-15 + 11 + 57) 0,222 = -12$;

$$x_{14} = -53 \cdot 0,45 = -2;$$

» 18 » 17 и 13 $x_{17} = -(3 + 45 - 12) 0,222 = -8$;

$$x_{13} = -36 \cdot 0,045 = -2 \text{ и т. д.}$$

Заканчивая обратный ход, проводим линию.

На этом вычисления можно оборвать.

Отметим, что практически вполне можно ограничиться двумя или даже одним обходом узлов в прямом и обратном направлениях. Суммируя числа в столбцах в табл. II. 1, начиная со строки 1, получим искомые значения углов поворота, которые записываются в нижней половине узловых кружков на схеме рамы (рис. 6).

6. Определяем значения моментов на схеме рамы:

умножая каждый угол поворота на жесткости T стержней, сходящихся в узле, определяем M' .

Так например, для узла 8:

$$M'_{87} = 0,547 \cdot 4 = 2,19 \text{ тм}; M'_{89} = 0,547 \cdot 4 = 2,19 \text{ тм};$$

$$M'_{8,13} = 0,547 \cdot 2 = 1,09 \text{ тм}; M'_{83} = 0,547 \cdot 2 = 1,09 \text{ тм};$$

передавая полученные моменты M' на противоположные концы стержней с коэффициентом $1/2$, определяем значения M'' .

Так, например:

$$M''_{78} = \frac{2,19}{2} = 1,09 \text{ тм}; M''_{38} = \frac{1,09}{2} = -0,54 \text{ тм и т. д.};$$

суммируя моменты M^p , M' и M'' , определяем действительные значения моментов на концах стержней.

Все вычисления в примере производились на логарифмической линейке. Несмотря на это, невязки моментов в узлах составляют 0,01—0,03 тм.

Для неразрезных балок и некоторых типов рам (например, многоэтажной полурамы) основную систему целесообразно выбрать по методу сил — с отброшенными связями.

Расчет методом итераций сводится к последовательному восстановлению и устранению связей (замыканию и размыканию шарниров) и разложению исходной системы на ряд элементарных подсистем — двухпролетных балок (двухэтажных рам и т. п.). Алгоритм расчета подобен рассмотренному выше, изменяются расчетные формулы. Соответствующие инструкции приведены в [97, 98].

Точное решение

II. 6. Если в методе итераций вначале определяются приближенные значения неизвестных и затем они последовательно уточняются, то в точном способе определяются соотношения между

неизвестными (раскрывается статическая неопределимость), после чего система уравнений распадается и значения неизвестных найти уже нетрудно. Например, в методе исключения (Гаусса) из уравнений последовательно исключаются: первое неизвестное (x_1 выражается через $x_2, x_3, \dots, \Delta_{1q}$), второе неизвестное (x_2 выражается через $x_3, x_4, \dots, \Delta_{2q}$), третье неизвестное (x_3 выражается через $x_4, x_5, \dots, \Delta_{3q}$) и т. д. Последнее n -ое неизвестное выражается только через свободный член Δ_{nq} . Здесь $\Delta_{1q}, \Delta_{2q}, \dots, \Delta_{nq}$ есть линейные функции свободных членов, получаемые в процессе преобразования уравнений. В результате получаем преобразованную треугольную матрицу.

Теперь нетрудно простой подстановкой получить значения неизвестных: подставляя значение x_n в $n-1$ -ое уравнение, находим x_{n-1} , подставляя x_{n-1} и x_n в $n-2$ -ое уравнение, находим x_{n-2} и т. д. до первого уравнения.

В методе Гаусса с увеличением количества неизвестных возрастает накопление погрешностей. Поэтому вычисления должны быть выполнены достаточно точно, с относительно большим числом верных значащих цифр, чего нельзя добиться на логарифмической линейке (погрешность уже в третьем знаке). Это стимулировало развитие специальных методов, позволяющих уменьшить накопление погрешностей, иногда ценой некоторого увеличения количества вычислений. К таким методам относятся методы фокусов для решения трехчленных уравнений. К этой же группе методов принадлежит и точный вариант метода перераспределения начальных значений неизвестных, обладающих большей общностью и некоторыми преимуществами вычислительного характера.

II. 7. Как известно, решение систем линейных уравнений строительной механики методом исключения может быть истолковано статически как введение в расчет постепенно усложняющихся статически неопределимых систем.

Применим эту трактовку к решению систем линейных уравнений ленточной структуры, в которой коэффициенты располагаются диагональной полосой, лентой (рис. 7). Количество диагоналей характеризует степень заполнения матрицы: для трехчленных уравнений имеем 3 диагонали (из них одна главная), для пятичленных уравнений — 5 диагоналей и т. д.

Использование особенностей ленточной структуры, характерной для многих задач строительной механики, дает возможность составить рекуррентные зависимости для общего случая (не только для трехчленных матриц) и применить алгоритм перерас-

пределения, сходный с тем, который рассматривался выше для итерационного решения.

Сходство, впрочем, только внешнее, так как речь идет о точном, замкнутом решении, когда количество вычислений является вполне определенным и не зависит, как в итерационном способе, от требуемой точности результатов.

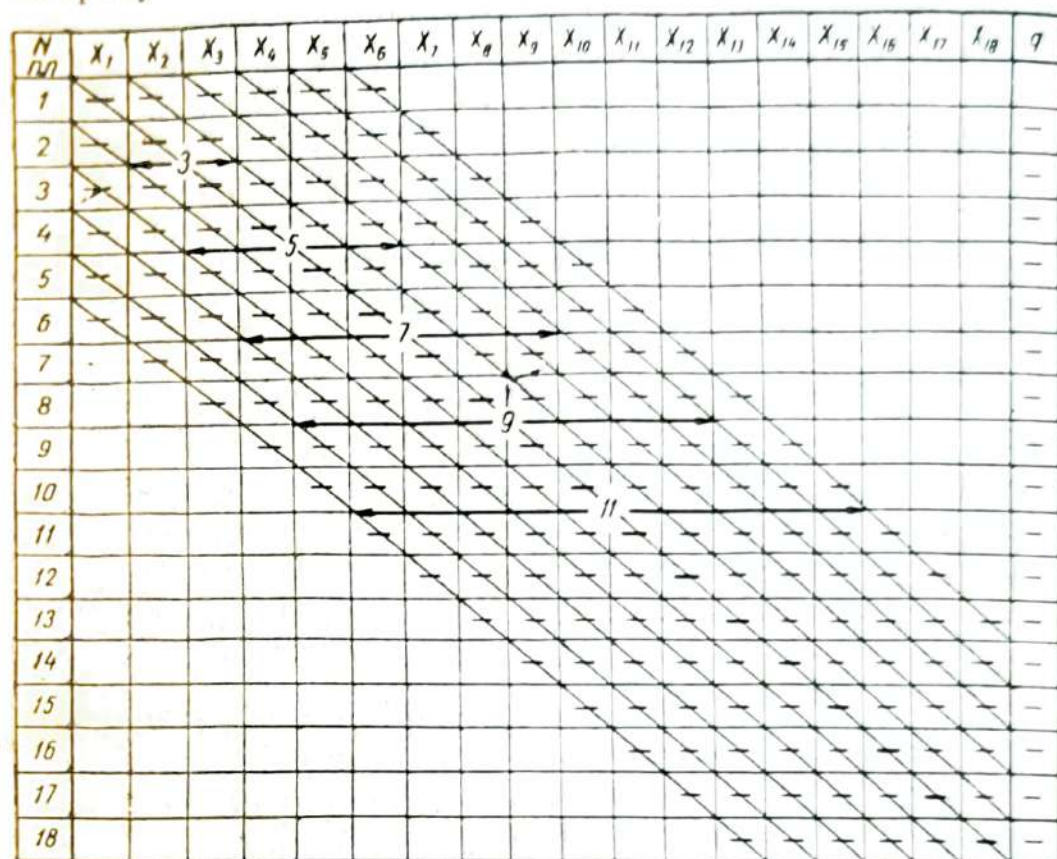
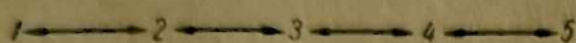


Рис. 7.

Связи между неизвестными в системе удобно изображать с помощью схемы непосредственных воздействий. Так, например, на рис. 8, а дана схема непосредственных воздействий для системы трехчленных уравнений. В этом случае неизвестное x_3 непосредственно связано только с неизвестными x_2 и x_4 , так как побочные коэффициенты δ_{13} и δ_{15} равны нулю. На рис. 8, б изображена схема непосредственных воздействий для системы пятичленных уравнений. Здесь неизвестное x_4 непосредственно связано с неизвестными x_2 , x_3 , x_5 , x_6 , так как побочные коэффициенты δ_{41} и δ_{47} равны нулю.

Кроме схемы непосредственных воздействий, можно построить еще схему окончательных воздействий, в которой указаны все связи как непосредственные, так и вторичные или более отдаленные. Так, например, для рис. 8, а x_1 связано с x_3 через посредство x_2 . В рассматриваемых системах все неизвестные в окончательной схеме связаны друг с другом; это бывает не так только



а

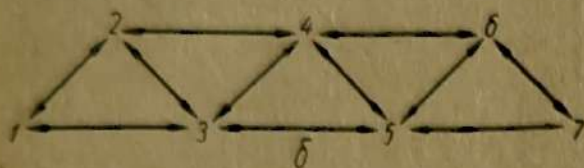


Рис. 8.

в распадающихся системах.

Однако чем ограниченнее схема непосредственных воздействий, тем легче разбить ее на отдельные части, разорвав те или иные связи. Этим обстоятельством можно воспользоваться для упрощения

решения (и здесь имеется аналогия с методом Эшби, хотя исследуемые области относятся к различным предметам).

Статически неопределимая система, для которой матрица имеет ленточную структуру, или, другими словами, ограниченную схему непосредственных воздействий, обладает определенными свойствами. Если m неизвестных равно нулю и $m = \frac{c-1}{2}$,

где c — количество кодиагоналей матрицы, то система разбивается на две независимые части — левую I и правую II (рис. 9). Для трехчленных уравнений достаточно принять одно неизвестное равным нулю: $x_n = 0$ (рис. 9, а), для пятичленных уравнений должны быть равны нулю два неизвестных: $x_{n-1} = 0$ и $x_n = 0$ (рис. 9, б), для семичленных уравнений — три неизвестных: $x_{n-1} = 0$, $x_n = 0$, $x_{n+1} = 0$ (рис. 9, в) и т. д. Это равносильно разрыву части связей.

Системы, изображенные на рис. 9, выбираются в качестве основных. Если принять все свободные члены, кроме свободных членов данной группы, равными нулю, то можно выразить все неизвестные системы через неизвестные данной группы. В этом состоит другое важное свойство рассматриваемых систем. Эта связь, выражаемая простыми зависимостями — коэффициентами передачи, позволяет разбить систему со многими неизвестными на ряд независимых групповых систем с относительно небольшим количеством неизвестных в каждой [99]. Разумеется, такой подход имеет смысл, если величина c значительно меньше порядка системы уравнений.

II. 8. Коэффициенты передачи определяются по индукции в процессе введения в расчет последовательно усложняющихся основных статически неопределимых систем или, что то же, в процессе прямого хода решения системы линейных уравнений.

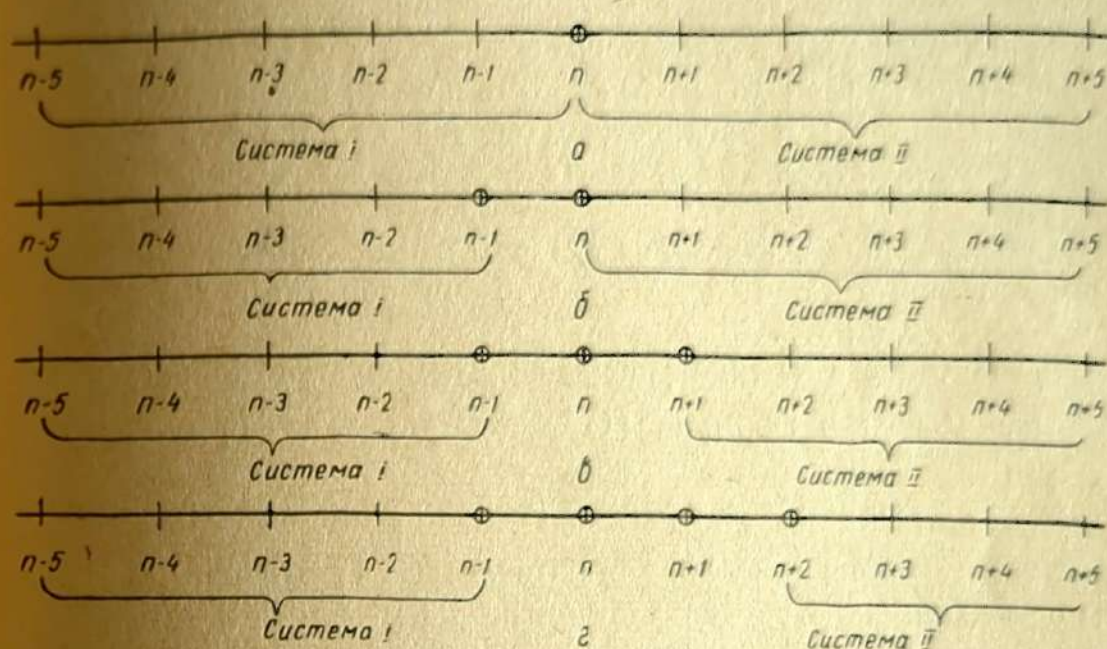


Рис. 9.

Опуская вывод, приведем формулы для определения коэффициентов передачи симметричной системы уравнений *:

$$k_{bc} = - \frac{\delta'_{bc}}{\delta'_{bb}}; \quad (1)$$

$$\delta'_{bc} = \delta_{bc} + \delta_{bc(a)}; \quad (2)$$

$$\delta'_{bb} = \delta_{bb} + \delta_{bb(a)}; \quad (3)$$

$$\delta_{bc(a)} = \sum k_{ab} \delta'_{ac}; \quad (4)$$

$$\delta_{bb(a)} = \sum k_{ab} \delta'_{ab}. \quad (5)$$

Здесь через a обозначены узлы, предшествующие узлу b ; через c — узлы, следующие за узлом b .

δ_{bc} , δ_{bb} , δ_{ab} — коэффициенты исходной системы уравнений;

* Формулы удалось привести к простому виду благодаря использованию взаимности коэффициентов системы ($\delta_{ik} = \delta_{ki}$) и ряду поставок и сокращений.

$\delta'_{bc}, \delta'_{bb}, \delta'_{ab}$ — коэффициенты преобразованной системы уравнений;

k_{bc} — коэффициент передачи представляет собой величину x_b , возникшую от $x_c = 1$. При этом все неизвестные, следующие после неизвестного x_b (кроме x_c , равного единице), равны нулю. Если узлы a расположены слева от узла b , а узлы c справа (при прямом ходе), то k_{bc} есть правый коэффициент передачи. Если узлы a расположены справа от узла b , а узлы c слева (при обратном ходе), то k_{bc} есть левый коэффициент передачи.

Суммирование в формулах (4) и (5) выполняется при прямом ходе для $a=b-1, b-2, b-3, \dots$, при обратном ходе для $a=b+1, b+2, b+3, \dots$.

В зависимости от типа уравнений суммируется больше или меньше членов: в формуле (4) — $\frac{c-3}{2}$ члена, в формуле (5) — $\frac{c-1}{2}$. В соответствии с этим величина $\delta_{bc(a)}$ для трехчленных уравнений равна нулю, для пятичленных уравнений состоит из одного члена и т. д. Величина $\delta_{bb(a)}$ для трехчленных уравнений состоит из одного члена, для пятичленных — из двух членов и т. д.

Составим теперь формулы для определения коэффициентов групповой системы с m неизвестными.

Основные статические неопределимые системы для 3, 5, 7, 9...-членных уравнений схематически изображены на рис. 9. Неизвестные группы узлов приняты равными нулю. Как уже указывалось, левая часть основной системы изолируется от правой. Обозначим δ_{ij}^0 уточненный коэффициент для узла i от $x_j = 1$ и соответствующих величин неизвестных систем I и II. Обозначим a узлы системы I, b — узлы системы II («узел», в данном случае, — отвлеченное понятие).

Коэффициенты групповой системы уравнений определяются по формуле

$$\delta_{ij}^0 = \delta_{ij} + \delta_{ij(a)} + \delta_{ij(b)}, \quad (6)$$

где $\delta_{ij(a)}$ отражает влияние левой части системы (определяется при прямом ходе), $\delta_{ij(b)}$ — правой части системы (определяется при обратном ходе).

При вычислении δ_{ij} величины i и j принимают значения, равные номерам неизвестных групповой системы. Решая уравнения, определяют начальные величины неизвестных x^0 .

II. 9. На основе полученных формул можно наметить следующий общий алгоритм решения систем s -членных уравнений. Сначала определяем коэффициенты передачи, выражающие связи между неизвестными. Затем неизвестные разбиваются на группы и составляются групповые системы уравнений. Решая их, находим начальные значения неизвестных. Далее производится перераспределение начальных значений неизвестных последовательным умножением их на коэффициенты передачи. При передаче неизвестных из группы на группу учитывается не только их начальное значение, но и вторичные величины, выражающие влияние предшествующих групп. Так как эта операция выполняется последовательно, то тем самым автоматически учитывается влияние всех предшествующих групп.

Перераспределение неизвестных производится при обходе узлов слева направо (прямой ход) и независимо справа налево (обратный ход). Суммируя затем все величины неизвестных — начальные и вторичные, определяем искомые величины неизвестных, удовлетворяющие исходной системе уравнений.

Так как большая система уравнений разбивается на независимые групповые системы, коэффициенты которых определяются с помощью двух независимых процессов исключения в прямом и обратном направлениях, то для определенных задач накопление погрешностей по сравнению с алгоритмом Гаусса резко уменьшается. Это очень существенно при выполнении вычислений на логарифмической линейке, удобна также форма записи.

Однако нетрудно заметить, что количество вычислений увеличивается по сравнению с алгоритмом Гаусса, так как приходится дважды производить исключение неизвестных и определять неизвестные, решая групповые системы уравнений (или в простых случаях, по готовым формулам).

Рассмотрим применение точного варианта алгоритма перераспределения для распространенных частных случаев.

Системы трехчленных уравнений

II. 10. В этом случае в формулах из II. 8 следует принять:

$$\delta_{bc(a)} = 0; \quad \delta'_{bc} = \delta_{bc}; \quad \delta_{bb(a)} = k_{ab}\delta_{ab}.$$

Отсюда получим следующие формулы для коэффициентов передачи:

$$k_{bc} = - \frac{\delta_{bc}}{\delta_{bb} + \delta_{bb(a)}}; \quad (1)$$

$$\delta_{bb(a)} = k_{ab}\delta_{ab}. \quad (2)$$

Номера a, b, c следуют при прямом ходе в порядке возрастания ($b=a+1, c=b+1$), при обратном ходе — убывания ($b=a-1, c=b-1$).

В данном случае коэффициенты передачи совпадают с обратными фокусными отношениями системы трехчленных уравнений.

Уравнение для определения начальных значений неизвестных:

$$x_b^0 \delta_{bb}^0 + \delta_{bp} = 0; \quad (3)$$

$$\delta_{bb}^0 = \delta_{bb} + \delta_{bb(a)} + \delta_{bb(c)}, \quad (4)$$

где $\delta_{bb(a)} = k_{ab} \delta_{ab}$ определяется при прямом ходе, $\delta_{bb(c)} = k_{cb} \delta_{bc}$ при обратном ходе.

Из формулы (3) находим

$$x_b^0 = - \frac{\delta_{bp}}{\delta_{bb}^0}. \quad (5)$$

Определение коэффициентов k и δ' удобно выполнять на схеме, изображенной на рис. 10.

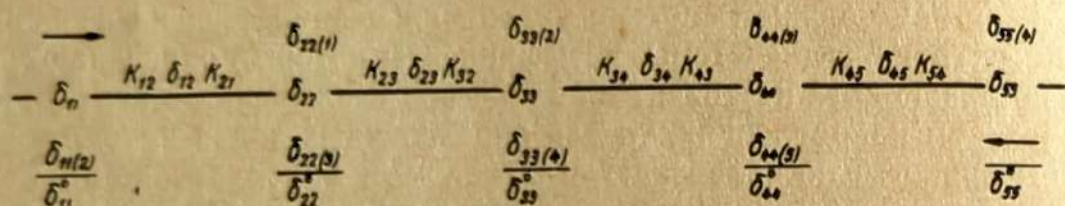


Рис. 10.

Прямой ход:

$$k_{12} = - \frac{\delta_{12}}{\delta_{11}}; \quad \delta_{22(1)} = k_{12} \delta_{12};$$

$$k_{23} = - \frac{\delta_{23}}{\delta_{22} + \delta_{22(1)}}; \quad \delta_{33(2)} = k_{23} \delta_{23} \text{ и т. д.}$$

Обратный ход:

$$k_{54} = - \frac{\delta_{45}}{\delta_{55}}; \quad \delta_{44(5)} = k_{54} \delta_{45};$$

$$k_{43} = - \frac{\delta_{34}}{\delta_{44} + \delta_{44(5)}}; \quad \delta_{33(4)} = k_{43} \delta_{34} \text{ и т. д.}$$

Суммируя числа, записанные на схеме, находим δ_{nn}^0 :

$$\delta_{11}^0 = \delta_{11} + \delta_{11(2)}; \quad \delta_{22}^0 = \delta_{22} + \delta_{22(1)} + \delta_{22(3)} \text{ и т. д.}$$

Контрольная проверка, следующая из принципа взаимности *, записывается так:

$$\delta_{11}^0 k_{12} k_{23} k_{34} k_{45} = \delta_{55}^0 k_{54} k_{43} k_{32} k_{21}.$$

Перераспределение начальных значений неизвестных выполняется в табл. II.2.

Таблица II.2

1		2		3		4		5		Номер строки
—	K_{12}	K_{21}	K_{23}	K_{32}	K_{34}	K_{43}	K_{45}	K_{54}	—	
→		x_2'		x_3'		x_4'		x_5'		1
x_1^0		x_2^0		x_3^0		x_4^0		x_5^0		2
x_1''		x_2''		x_3''		x_4''		←		3
x_1		x_2		x_3		x_4		x_5		Σ

Прямой ход: $x_2' = x_1^0 k_{21}$; $x_3' = (x_2^0 + x_2') k_{32}$ и т. д.

Обратный ход: $x_4'' = x_5^0 k_{45}$; $x_3'' = (x_4^0 + x_4'') k_{34}$ и т. д.

Суммируя начальные значения и поправки к ним, находим искомые значения неизвестных: $x_1 = x_1^0 + x_1'$; $x_2 = x_2^0 + x_2' + x_2''$ и т. д.

Как видим, механизм расчета сходен с тем, который применяется в методе последовательных приближений. Однако количество вычислений — конечное, число строк в таблице перераспределения всегда равно 3.

II. 11. Для определенных задач можно конкретизировать формулы, подставив вместо величин δ геометрические и жесткостные характеристики (приведенные длины, погонные жесткости и т. п.).

Например, для несвободных рам без замкнутых контуров формулы примут следующий вид:
для прямого хода

$$k_{bc} = - \frac{T_{bc}}{R_b^0 + \rho_b}; \quad \rho_b = k_{ab} T_{ab}; \quad (1)$$

* О контрольных равенствах для фокусных отношений см. [76, 97].

для обратного хода

$$k_{ba} = - \frac{T_{ab}}{R_b^0 + \rho_b}; \quad \rho_b = k_{cb} T_{bc}; \quad (2)$$

для начальных значений неизвестных

$$x_b^0 = - \frac{2 \Sigma M^p}{R_b}; \quad R_b = R_b^0 + \rho_b + \rho_b'; \quad (3)$$

для контроля

$$R_1 k_{12} k_{23} \dots k_{n-1, n} = R_n k_{n, n-1} \dots k_{32} k_{21}. \quad (4)$$

Для R^0 и T см. формулы (3) и (4) из II. 5.

Пример расчета дан в II. 15. В том примере рама без замкнутых контуров является подсистемой рамы с замкнутыми контурами.

Системы пятичленных уравнений

II. 12. Для систем пятичленных уравнений формулы для определения коэффициентов передачи удобно представить в следующем виде:

$$k_{cd} = - \frac{\delta'_{cd}}{\delta'_{cc}}; \quad (1)$$

$$k_{ce} = - \frac{\delta_{ce}}{\delta'_{cc}}; \quad (2)$$

$$\delta'_{cc} = \delta_{cc} + \delta_{cc(a)} + \delta_{cc(b)}; \quad (3)$$

$$\delta_{cd} = \delta_{cd} + \delta_{cd(b)}; \quad (4)$$

$$\delta_{cc(a)} = k_{ac} \delta_{ac}; \quad (5)$$

$$\delta_{cc(b)} = R_{bc} \delta'_{bc}; \quad (6)$$

$$\delta_{cd(b)} = k_{bd} \delta'_{bc}. \quad (7)$$

Номера a, b, c, d, e следуют при прямом ходе в порядке возрастания ($b=a+1, c=b+1, d=c+1, e=d+1$), при обратном ходе — убывания ($b=a-1, c=b-1, d=c-1, e=d-1$). В последнем случае вместо δ' пишется δ'' .

Уравнения для определения начальных значений неизвестных запишем так:

$$x_c^0 \delta_{cc}^0 + x_d^0 \delta_{cd}^0 + \delta_{cp} = 0; \quad (8)$$

$$x_c^0 \delta_{cd}^0 + x_d^0 \delta_{dd}^0 + \delta_{dp} = 0; \quad (9)$$

$$\delta_{cc}^0 = \delta'_{cc} + k_{ec} \delta_{ec}; \quad (10)$$

$$\delta_{dc}^0 = \delta_{dc}'' + k_{bc}\delta_{bd}; \quad (11)$$

$$\delta_{dd}^0 = \delta_{dd}'' + k_{bd}\delta_{bd}; \quad (12)$$

$$\delta_{cd}^0 = \delta_{cd}' + k_{ed}\delta_{ec}. \quad (13)$$

Величины δ' определяются при прямом ходе, δ'' — при обратном ходе, номера b, c, d, e следуют в порядке возрастания.

Проверкой служит равенство, следующее из принципа взаимности $\delta_{cd}^0 = \delta_{dc}^0$.

Данный алгоритм и его применение для расчета неразрезных балок на упругих опорах подробно рассмотрены в работе [98]. Как и для трехчленных уравнений коэффициенты k, δ' и δ'' определяются на специальной схеме, а искомые величины неизвестных — в таблице перераспределения.

II. 13. В работе [99] приведен алгоритм перераспределения для систем восьмичленных уравнений цилиндрических оболочек, которые после переименования неизвестных превращаются в неполные девятичленные уравнения ленточной структуры. Однако в таких системах соотношение между коэффициентами таково, что их нужно решать с большой точностью на цифровых машинах, а в этом случае требования к алгоритмам другие и ленточную структуру целесообразно использовать в иной форме (см. II. 2).

Смешанные решения

II. 14. На основе алгоритма перераспределения можно получить разнообразные смешанные решения посредством: сочетания точного и итерационного решения, окаймления дополнительными уравнениями, сочетания основных систем метода сил и метода деформаций.

II. 15. Приведем ряд примеров. Рассмотрим расчет несвободных многопролетных многоэтажных рам. В II. 5. излагался итерационный способ расчета многоэтажных рам, статический смысл которого состоял в разложении сложной рамы на ряд простейших крестообразных. Здесь рассмотрен другой способ расчета, при котором защемления устраняются не из отдельных узлов основной системы, а поочередно из всех узлов горизонтальных уровней. Так, например, для основной системы, изображенной на рис. 11, а, будем последовательно устранять защемления из уровней А, В и С. При этом получим ряд подсистем — двухэтажных рам I—II, II—III и III (рис. 11, б). Так как эти системы

не имеют замкнутых контуров, то для расчета каждой из них целесообразно применить точный вариант алгоритма перераспределения II. 10. Передачу же углов поворота по вертикали с узлов одного уровня на узлы другого будем производить последовательными приближениями.

В первом цикле расчет производится на внешнюю нагрузку и на действие углов поворота узлов, определяемых в процессе

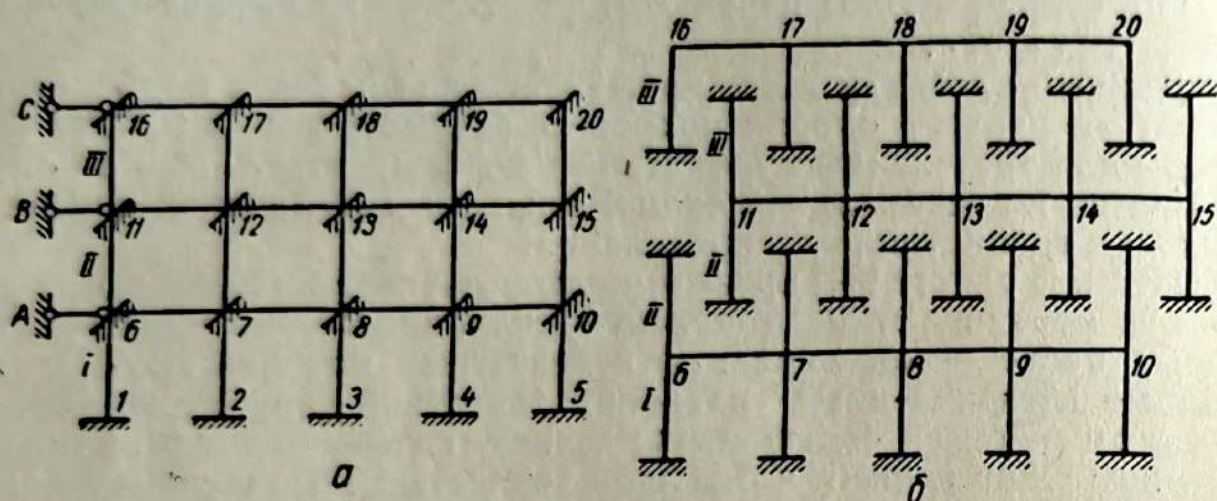


Рис. 11.

последовательных приближений. Из расчета рамы I—II на внешнюю нагрузку определяем первые приближения для углов поворота узлов уровня A. Из расчета рамы II—III на внешнюю нагрузку и на действие заданных углов поворота узлов уровня A определяем первые приближения для углов поворота узлов уровня B. Наконец из расчета рамы III на внешнюю нагрузку и на действие заданных углов поворота узлов уровня B определяем первые приближения для углов поворота узлов уровня C.

Во втором цикле расчет производится аналогично, но на действие углов поворота, не учтенных в первом цикле, и т. д.

В тех случаях, когда жесткости стоек больше жесткостей ригелей, быстрее ведет к цели устранение защемлений последовательно из узлов рамы по вертикалям 6, 11, 16, затем 7, 12, 17 и т. д. Порядок расчета тот же, что и в предыдущем случае.

Математическая суть рассмотренного варианта метода последовательных приближений состоит в выделении трехчленных групп из системы уравнений. Поскольку значительная часть расчета выполняется точно, сходимость процесса итерации ускоря-

ется. Первый же цикл обеспечивает удовлетворительную точность.

Рассмотрим решение примера, приведенного в II. 5, смешанным способом. Механизмы точного и итерационного алгоритмов перераспределения сходные и легко объединяются в один общий алгоритм.

Расчет выполняется в следующей последовательности.

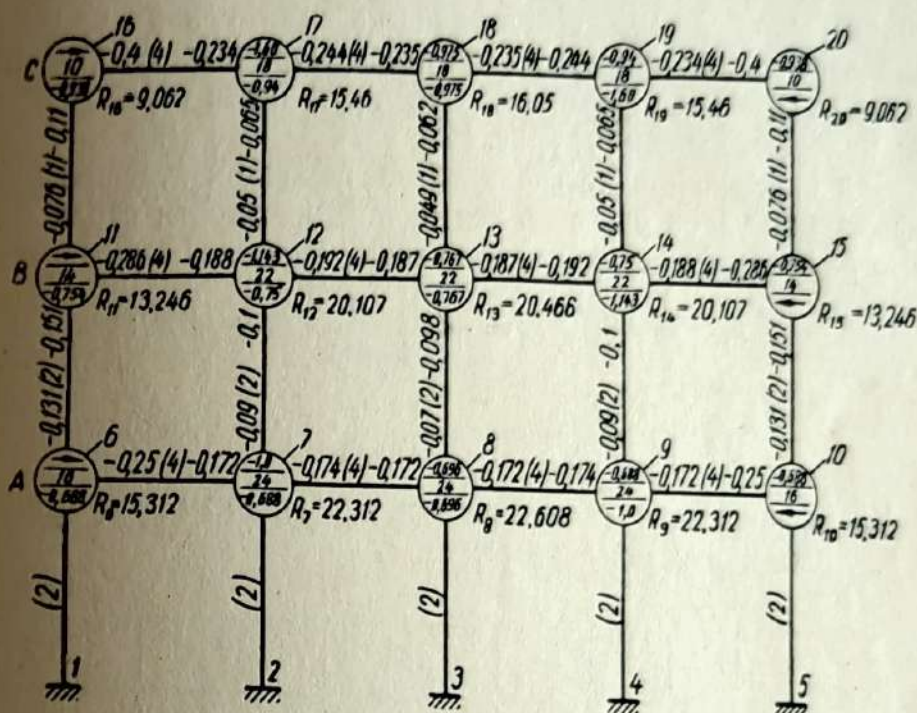


Рис. 12.

На схеме рамы (рис. 12) в скобках записываются относительные величины жесткостей T , а в узловых кружках — значения $R^0 = 2 \sum T$.

Для каждого горизонтального уровня вычисляются коэффициенты k и R , как в точном способе. Например, для уровня А при прямом ходе определяем: $k_{67} = -\frac{T_{67}}{R_6^0} = -\frac{4}{16} = -0,25$;

$$p_7 = K_{67} \cdot T_{67} = -0,25 \cdot 4,0 = -1,0; \quad k_{78} = -\frac{T_{78}}{R_7^0 + p_7} =$$

$$= -\frac{4,0}{24,0 - 1,0} = -0,174 \text{ и т. д.};$$

$$\text{при обратном ходе: } k_{10,9} = - \frac{T_{9,10}}{R_{10}^0} = - \frac{4,0}{16,0} = -0,25;$$

$$\rho_9 = -k_{10,9} \cdot T_{9,10} = -0,25 \cdot 4,0 = -1,0;$$

$$k_{98} = - \frac{T_{89}}{R_9^0 + \rho_9} = - \frac{4}{24,0 - 1,0} = -0,174 \text{ и т. д.}$$

В данном случае, ввиду симметрии рамы, левые коэффициенты можно было без вычислений записать симметрично правым или использовать как проверку (для несимметричных систем используется контрольная формула). Суммируя числа в узловых кружках, находим значения R (рис. 12).

Для определения коэффициентов передачи k , связывающих деформации узлов рамы по вертикали, жесткости стоек необходимо разделить на R соответствующих узлов. Например:

$$k_{8,13} = - \frac{T_{8,13}}{R_8} = - \frac{2,0}{22,608} = -0,070;$$

$$k_{13,8} = - \frac{T_{8,13}}{R_{13}} = - \frac{2,0}{20,465} = -0,098.$$

Коэффициенты k записываются в заглавных строках табл. II. 3. Знак минус опускается.

Таблица II. 3

	16	17	18	19	20	Но- мер стро- ки
Уровень С	$\begin{array}{c} -0,4 \\ 0,11 \end{array}$	$\begin{array}{c} 0,234 \\ 0,065 \end{array}$	$\begin{array}{c} 0,235 \\ 0,062 \end{array}$	$\begin{array}{c} 0,244 \\ 0,065 \end{array}$	$\begin{array}{c} 0,4 \\ 0,11 \end{array}$	
	→	-401	90	-226	96	1—С
	$\begin{array}{c} 1840 \\ -123 \end{array}$	$\begin{array}{c} - \\ 18 \end{array}$	$\begin{array}{c} 875 \\ -37 \end{array}$	$\begin{array}{c} - \\ -15 \end{array}$	$\begin{array}{c} -3385 \\ 218 \end{array}$	2—С
	58	-162	-171	742	←	3—С
Первое при- ближение	1775	-545	757	501	-3071	4—С
	11	12	13	14	15	Но- мер стро- ки
Уровень В	$\begin{array}{c} 0,076 \\ -0,286 \\ 0,151 \end{array}$	$\begin{array}{c} 0,05 \\ 0,188+0,192 \\ 0,1 \end{array}$	$\begin{array}{c} 0,049 \\ 0,187+0,187 \\ 0,098 \end{array}$	$\begin{array}{c} 0,05 \\ 0,192+0,188 \\ 0,1 \end{array}$	$\begin{array}{c} 0,076 \\ 0,286- \\ 0,151 \end{array}$	

	→	-205	33	-127	43	1-B
	1255 -168	- 28	685 -58	- -23	-2312 298	2-B
	22	-107	-68	380	←	3-B
Первое при- ближение	1109	-284	592	230	-1971	4-B
Первая поправка	-135	27	-37	-25	235	6-B
Уровень А	6	7	8	9	10	Но- мер стро- ки
	0,131 -0,25	0,09 0,172+0,174	0,07 0,174+0,172	0,09 0,174+0,172	0,131 0,25-	
	→	-187	32	-113	28	1-A
	1087	-	618	-	-2004	2-A
	24	-97	-59	344	←	3-A
Первое при- ближение	1111	-284	591	231	-1976	4-A
Первая поправка	-146	26	-42	20	259	6-A

Примечание. Значения x умножены на 1000.

На новой схеме рамы (см. рис. 6) записываем значения M^p . Начальные углы поворота определяются по формуле

$$x_i^0 = - \frac{2 \Sigma M_i^p}{R_i}.$$

Например, для узла 6 $x_6^0 = - \frac{-2 \cdot 8,33}{15,312} = 1,087$.

* Для каждого нагружения вычеркивается отдельная схема. Что касается схемы на рис. 12, то она является общей для всех нагружений. Целесообразно заготовить предварительно специальные бланки с нанесенной сеткой стержней.

Значения x^0 записываются в табл. II. 3, в строках 2—А, 2—В и 2—С для уровней А, В и С и производится перераспределение углов поворота.

По горизонтали А. Полученные в строке 4—А углы поворота передаем по вертикали вверх и результаты записываем в строку 2—В.

Например:

$$x_{11} = -1111 \cdot 0,151 = -168.$$

По горизонтали В. Полученные в строке 4—В углы поворота передаем по вертикали вверх в строку 2—С и вниз — в строку 6—А.

Например: $x_{16} = -1109 \cdot 0,11 = -123$; $x_6 = -1109 \cdot 0,131 = -146$.

Далее перераспределение продолжается аналогично, сколько бы ни было горизонталей.

Величины, полученные в строках 6—А, 6—В, ..., являются начальными для второго цикла расчета, но дальнейшие вычисления не нужны и после первого цикла процесс можно оборвать.

Найденные значения углов поворота (первое приближение плюс первая поправка) записываем в узловые кружки схемы рамы и находим моменты на концах стержней так же, как в II. 5. Расхождения с моментами. (см. рис. 6) составляют до 3% (погрешности более 3% имеются только в тех моментах, которые ввиду их малых величин практического значения не имеют).

II. 16. Рассмотрим расчет свободных рам с небольшим числом степеней свободы. Для расчета свободной рамы в качестве основной системы выбираем раму несвободную, т. е. лишенную линейных смещений. Несвободная рама рассчитывается с помощью точного (если рама не имеет замкнутых контуров) или итерационного алгоритма перераспределения на единичные смещения и внешнюю нагрузку. Затем составляются дополнительные уравнения, выражающие отсутствие суммарных реакций в добавленных фиктивных стержнях, решая которые определяют значения линейных смещений.

При небольшом числе дополнительных уравнений такая схема является достаточно экономной. При большом числе независимых линейных смещений применяются другие способы, описанные в II. 18, II. 19.

Перераспределение углов поворота выполняется в общей табл. II. 4.

Таблица II.4

Воздействие	Номера узлов
	Коэффициенты передачи
$x_I = 1$	
$x_{II} = 1$	
...	...
p	
x_I	
x_{II}	
...	
Σ	

Порядок расчета следующий:

расчет выполняется на действие единичных смещений $x_I = 1$, $x_{II} = 1, \dots$ и внешней нагрузки p ;

составляется система дополнительных уравнений, определяются величины x_I , x_{II} , ...;

величины x_I , x_{II} , ... умножаются на углы поворота от единичных смещений и определяются углы поворота, соответствующие смещениям x_I , x_{II} , ...;

суммируются строки для x_I , x_{II} , ... и внешней нагрузки p и в строке Σ получаются искомые величины углов поворота;

в узловые кружки на схеме рамы записываются найденные углы поворота, а возле соответствующих стержней (или групп стержней) — величины x_I , x_{II} , ...;

определяются моменты на схеме рамы, как было описано в II. 5. Для стержней, имеющих смещения, помимо моментов от углов поворота, на схеме выписываются моменты от смещений —

произведения величины $x_1, x_2, \dots$ на моменты защемления стержней от единичных смещений.

II. 17. Рассмотрим расчет свободных многоэтажных многопролетных рам с параллельными стойками. Для расчета многоэтажных многопролетных рам методом уравнивания известна следующая схема;

система рассчитывается как несвободная методом уравнивания, т. е. защемления последовательно поворачиваются на такие углы, чтобы узлы находились в равновесии;

удаляются фиктивные стержни, от сдвигающих сил система получает смещения;

вновь накладываются фиктивные связи и производится уравнивание узлов.

Далее циклы вычислений повторяются до тех пор, пока система не окажется в равновесии.

Схема весьма проста, но оправдывает себя только в тех случаях, когда жесткости ригелей существенно больше жесткостей стоек, что близко к принятой основной системе. Это и понятно, так как чем ближе основная система к расчетной, тем быстрее сходится процесс последовательных приближений. Наоборот, в тех случаях, когда жесткости стоек намного больше жесткостей ригелей (именно этот случай имеет место в высотных зданиях), процесс сходится медленно.

Поэтому разными авторами предлагались те или иные приемы ускорения сходимости. И. К. Снитко предлагает учесть сумму поправок к начальным неуравновешенным моментам с помощью коэффициентов накопления, определяемым приближенно как суммы членов убывающей геометрической прогрессии [94]. С. А. Ривкин [88] применяет этот прием только для учета смещений. Знаменатель прогрессии, так называемый корректирующий множитель, находится приближенно на некотором этапе как отношение последующей и предыдущей поправок для сдвигающих сил. Идея прерывания циклов и определения суммарной поправки как суммы членов некоторого ряда в своеобразной форме разработана П. П. Шагиным [129]. С. А. Рогицкий [84] вводит поправки от смещения при уравнивании моментов одного узла и полученные моменты сразу же включает в процесс последовательных приближений, т. е. применяет итерацию по Зейделю. Г. Кани [140] определяет поправки от смещения для каждого яруса. Указанные авторы используют алгоритм распределения моментов защемления. При тех же предпосылках можно применить и алгоритм перераспределения начальных деформаций.

Другой путь состоит в комбинации точного, одношагового решения и метода последовательных приближений, что позволяет получить практически точное решение после двух циклов расчета при произвольных соотношениях между жесткостями ригелей и стоек. Интересны два способа решения этой задачи.

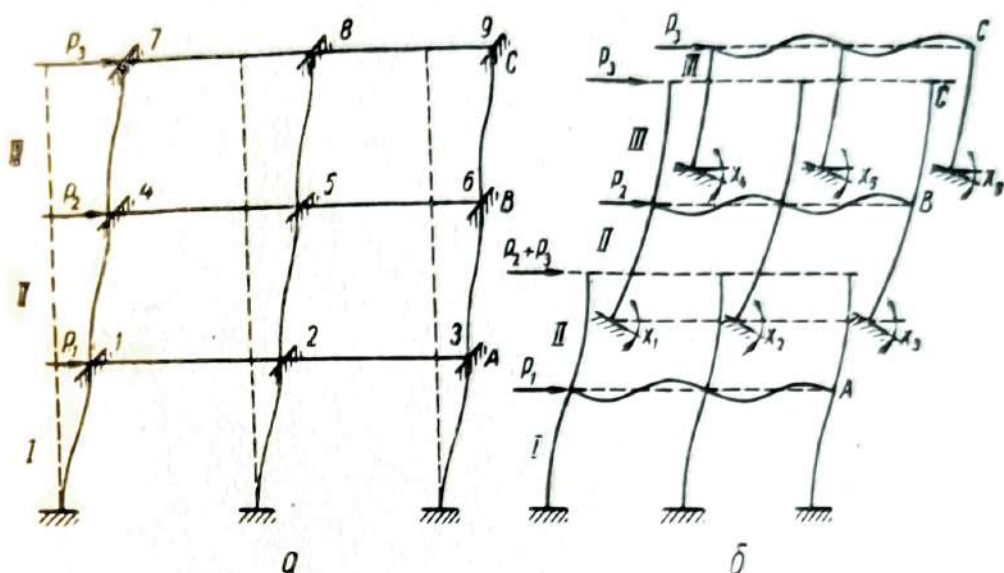


Рис. 13.

II. 18. В первом способе за основную примем систему с наложенными на узлы защемлениями, но без горизонтальных связей (рис. 13, а). Под действием горизонтальной нагрузки основная система получит некоторые смещения, причем деформация каждого яруса происходит независимо от соседних.

Устраним защемления из узлов уровня А. Деформация двухэтажной системы I—II (рис. 13, б) также происходит независимо от вышележащей части рамы. Точно так же, устранив защемления из уровня В, получим систему II—III и т. д.

Определяемые из расчета каждой элементарной системы углы поворота прикладываем в качестве внешнего воздействия к следующей элементарной системе.

Таким образом, расчет многэтажной свободной рамы производится так же, как несвободной рамы смешанным способом (II. 15). Отличие состоит только в том, что каждая двухэтажная система рассчитывается теперь с учетом смещений. Так как элементарная двухэтажная система имеет две степени свободы, то для определения линейных смещений нижнего и верхнего этажа

надо составить и решить два дополнительных уравнения. Решение этих уравнений можно представить также в виде готовых формул.

В данном способе узлы среднего уровня двухэтажной рамы свободны от защемлений, при определении линейных смещений

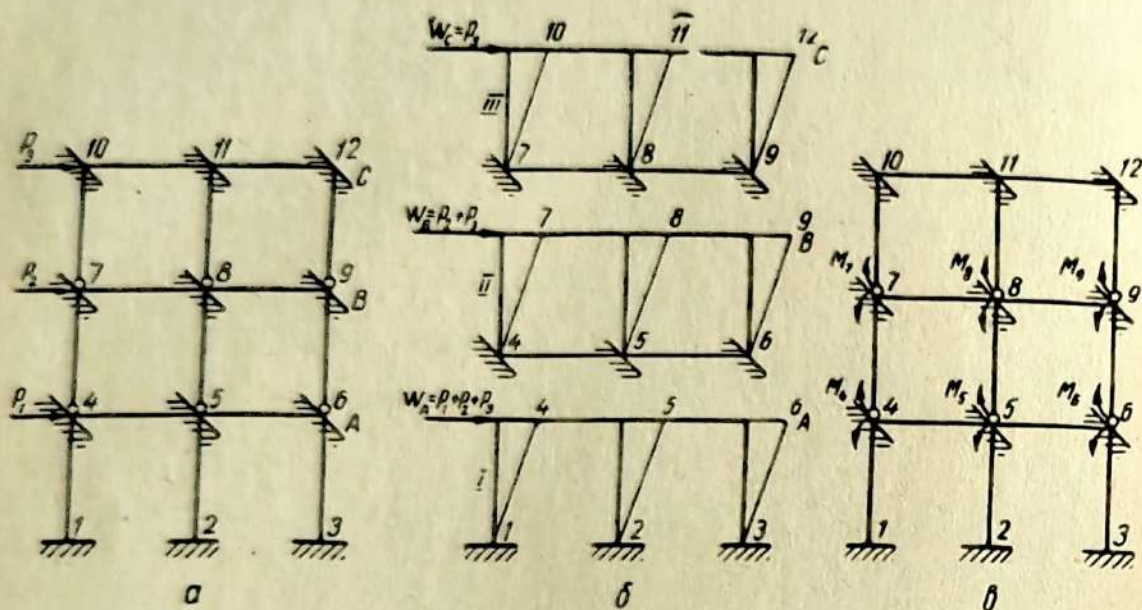


Рис. 14.

учитываются упругости этих узлов, вследствие чего процесс последовательных приближений для многоэтажной рамы сходится очень быстро. Первый же цикл дает удовлетворительное решение, а второй — практически точное.

II. 19. Во втором способе основной будет смешанная система с защемлениями в узлах и шарнирами внизу стоек (рис. 14, а) *.

Противоречия между основной и расчетными системами ликвидируем, устраняя защемления и одновременно закрывая шарниры.

* Такая расстановка шарниров целесообразна в тех случаях, когда коэффициенты $\eta = T/l$ для стоек убывают от нижних этажей к верхним. Это наиболее распространенный случай. Если же величины η возрастают от нижних этажей к верхним, то шарниры лучше располагать вверх стоек (получаются перевернутые одноэтажные рамы). Если многоэтажная рама имеет средний этаж с резко увеличенными по сравнению со смежными этажами значениями η стоек (например, технический этаж), то применять второй способ не следует. Для такого этажа предположение о наличии шарниров не может служить основой для итерационного способа, так как отбрасываемые связи не являются второстепенными.

Устраняя защемления из уровня A , получим одноэтажную раму I (рис. 14, б). Деформация этой системы происходит независимо, изолированно от вышележащей части рамы. Последнюю можно отбросить, приложив в уровне A сумму вышележащих горизонтальных сил

$$W_A = P_1 + P_2 + P_3.$$

Рассчитав одноэтажную раму I, найдем смещение x_I и углы поворота узлов уровня A . Закроем далее шарниры уровня A и устраним одновременно защемления из уровня B . При этом получим изолированную раму II. В уровне B приложена сумма вышележащих горизонтальных сил $W_B = P_2 + P_3$. В качестве внешнего воздействия к раме II приложены также углы поворота узлов уровня A .

Рассчитав эту систему, определим смещение x_{II} и углы поворота узлов уровня B . Точно так же, закрывая шарниры уровня B и устраняя защемления из узлов уровня C , получим раму III, рассчитав которую, найдем x_{III} и углы поворота узлов уровня C .

На этом заканчивается первый цикл расчета.

Включим теперь шарниры внизу стоек. В шарнирах будут действовать некоторые моменты, которые служат «нагрузкой» для второго цикла расчета (рис. 14, в). Второй цикл производится аналогично первому. Последовательно рассчитывая рамы I, II и III, получим ряд поправок для линейных смещений и углов поворота.

Практически после второго цикла процесс можно оборвать. Таким образом, расчет многоэтажной рамы сводится к расчету ряда одноэтажных рам.

Как по первому, так и по второму способу расчет выполняется в таблице перераспределения деформаций, а определение моментов по известным деформациям — на схеме рамы. Второе решение проще первого, так как расчет одноэтажной рамы проще, чем двухэтажной, но имеет некоторые ограничения*.

Интересно отметить, что получить второй способ непосредственно из системы уравнений было бы трудно.

Это пример того, что статический анализ не всегда является только трактовкой известного математического решения.

Разнообразные примеры расчета рам, неразрезных и перекрестных систем с помощью алгоритмов перераспределения приведены в работах [97, 98].

* См. сноску на стр. 60.

Расчет рам произвольного вида на машине ЭМСС-7 (7М) машинно-аналитическими методами

II. 20. Машина ЭМСС-7 (7М) непосредственно моделирует регулярные рамы, у которых стойки должны быть вертикальными и одинаковой высоты (в пределах яруса), а ригели в каждом уровне не имеют пропущенных стержней. Подробные сведения об

этих машинах даны в работе [75] и заводских инструкциях.

В машине имеется возможность задавать в качестве внешнего воздействия не только неуравновешенные моменты в узлах (токи), но и любые деформационные факторы — углы поворота (напряжения) и перекосы стержней. Последним обстоятельством можно воспользоваться для построения итерационных схем, близких

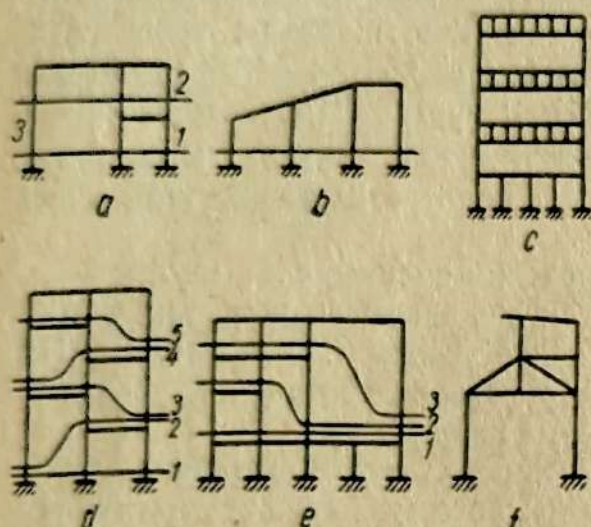


Рис. 15.

к тем, которые рассматривались в II. 18, но более эффективных. В данном случае в качестве системы, решаемой точно, принимается не двухэтажная рама без замкнутых контуров, а вся система в целом, узлы которой являются упругими, а смещения (перекосы) стержней равны нулю или имеют фиксированные значения. Таким образом, в качестве основной системы принимается несвободная рама с заданными линейными смещениями узлов, которая непосредственно моделируется на машине.

Излагаемые ниже (II. 21) итерационные методы групповых перемещений и основных жесткостей позволяют достаточно просто рассчитывать на указанной машине рамы с разнообразными нерегулярностями: произвольным расположением ригелей на разных уровнях; стойками неравной длины или наклонными; включенными элементами в виде раскосных или безраскосных ферм (рис. 15).

Принятые обозначения.

Статические и геометрические величины.

M — изгибающий момент,

M — момент в защемлении от нагрузки на стержне,

N — нормальная сила,
 Q — поперечная сила,
 E — модуль упругости материала,
 I — момент инерции сечения стержня.

δ — линейное смещение,
 $l(h)$ — длина (высота) стержня,

$\psi = \frac{\delta}{l}$ — угол перекоса,

φ — угол поворота узла,

ΔM — сумма моментов в верхнем и нижнем сечении стойки (или группы стоек) за вычетом моментов защемлений (т. е. сумма моментов от деформационных воздействий).

$$\Delta M = M_A + M_B - \bar{M}_A - \bar{M}_B.$$

Электрические величины и их связь со статическими величинами.

Токи $I = \gamma_M M$; $\Delta I = \frac{\gamma_M}{3} \Delta M$.

Сопротивление $r = \gamma_r \frac{l}{2EJ}$.

Напряжение $\sigma = \gamma_\varphi \cdot \rho$.

Масштабные коэффициенты γ_M , γ_r , γ_φ .

Связь между масштабными коэффициентами

$$\gamma_M = \frac{\gamma_\varphi}{\gamma_r}.$$

II. 21. Сущность метода групповых перемещений состоит в последовательном уравнивании разрезов рамы при определенных групповых смещениях стоек.

Назовем r_{ip} сумму горизонтальных проекций всех сил, внутренних и внешних, действующих на отсеченную часть рамы:

$$r_{ip} = W_i - \sum_i Q, \quad (1)$$

где W_i — сумма вышележащих горизонтальных нагрузок;
 $\sum_i Q$ — сумма поперечных сил, пересекаемых i -тым разрезом.

Например, для рис. 16:

$$W_i = P_1 + P_2 + P_3 + P_4;$$

$$\sum_i Q = Q_1 + Q_2 + Q_3.$$

Силы $P_1, P_2, \dots$ считаются положительными, когда они направлены слева направо, поперечные силы $Q_1, Q_2, \dots$, — когда вращают стержни по часовой стрелке.

Поперечные силы Q_i в стойках определяются из расчета рамы. Если они найдены точно, то условие равновесия сил, действующих на отсеченную часть рамы, выполняется, т. е. $r_{ip} = 0$.

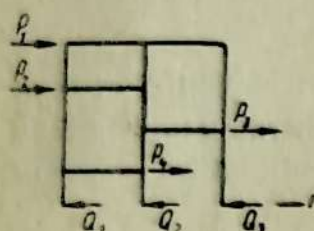


Рис. 16

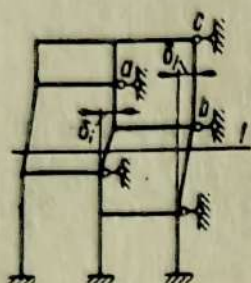


Рис. 17.

Если же величины поперечных сил найдены приближенно, то появится неуравновешенная (сдвигающая) сила $r_{ip} \neq 0$, которая вызовет горизонтальные смещения стоек. Предположим, что от силы r_{ip} сместятся только стойки i -го разреза, причем на одинаковую величину.

Это равносильно смещению группы фиктивных стержней (например, на рис. 17 — фиктивных стержней a, b, c).

Такая предпосылка дает возможность определить величину смещения из одного условия равновесия:

$$r_{ip} - k_i r_{ii} = 0. \quad (2)$$

$r_{ii} = \sum Q_i \bar{\delta}_i$ — суммарная поперечная сила в стойках i -го разреза от некоторого «единичного» смещения $\bar{\delta}_i^*$. Знак минус во втором члене формулы (2) принят потому, что положительная поперечная сила (действующая по часовой стрелке) направлена справа налево (рис. 16);

k_i — коэффициент пропорциональности, равный отношению искомого смещения к «единичному».

Из формулы (2) следует: $k_i = \frac{r_{ip}}{r_{ii}}$.

Искомое смещение (перекос n -го стержня) определяется по формуле

$$\delta_n = k_i \bar{\delta}_i; \quad \psi_n = k_i \bar{\psi}_i. \quad (3)$$

* Смещение $\bar{\delta}_i$ не обязательно должно быть равно единице; оно может быть равно любому заданному числу.

Переко́с стержня ψ_n равен смещению одного конца относительно другого, деленного на длину:

$$\psi_n = \frac{\delta_n}{l_n} \quad (4)$$

II. 22. Имея в виду, что перерезывающая сила в стойке равна сумме концевых моментов, разделенных на длину, и выражая моменты через токи, получим после ряда подстановок и сокращений следующие расчетные формулы:

$$r_{ii} = a_i = - \sum \frac{\Delta I^{\bar{\psi}}}{l} ; \quad (1)$$

$$A_i^0 = \frac{r_M}{3} \left(W_i - \sum \bar{Q} \right) ; \quad (2)$$

$$r_{ip} = A_i = A_i^0 + \sum \frac{\Delta I}{l} ; \quad (3)$$

$$k_i = \frac{A_i}{a_i} ; \quad (4)$$

$$\psi_n = k_i \bar{\psi}_n. \quad (5)$$

Уравновешивание разрезов выполняется последовательно до тех пор, пока неуравновешенные силы A_i не станут близкими к нулю.

Проследим ход расчета на примере рамы, изображенной на рис. 15, d.

В начале расчета определяем по формуле (1) величины a от «единичных» переко́сов $\bar{\psi}$. Сообщая некоторые переко́сы

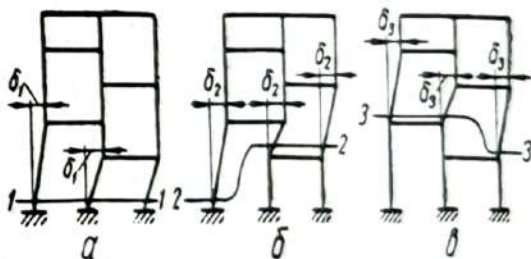


Рис. 18.

стойкам первого разреза, находим a_1 , второго разреза — a_2 , третьего разреза — a_3 и т. д. (рис. 18).

Переко́сы группы стоек i -го разреза обратно пропорциональны их длинам. Например, если принять переко́с первой стойки равным единице, то переко́с второй стойки будет равен l_1/l_2 , тре-

твей стойки l_1/l_3 и т. д., где $l_1, l_2, l_3, \dots$ — длины соответствующих стоек. Если группа стоек имеет одинаковую длину, то вводится перекося сразу для всей этой группы.

Далее выполняется процесс уравнивания разрезов.

Уравнивая первый разрез ($i=1$), находим по формулам (2) — (5) перекося ψ_n стоек этого разреза и прикладываем их в качестве внешнего воздействия, т. е. фиксируем в машине с помощью ручек соответствующие перекося (при переключателе в положении ψ). Таким же образом уравниваем последовательно второй, третий и последующий разрезы ($i=1, 2, 3, \dots$), фиксируя каждый раз полученные перекося. На этом заканчивается первый цикл расчета.

Возвращаемся снова к первому разрезу: в нем будет некоторая, теперь уже небольшая, неуравновешенная сила, вызванная уравниванием других разрезов. После уравнивания первого разреза и фиксирования новых, исправленных величин перекося переходим ко второму разрезу, затем к третьему и т. д.

Практика показала, что при любых соотношениях жесткостей рамы процесс последовательных приближений сходится очень быстро — обычно можно ограничиться двумя циклами расчета. Третий цикл дает поправки меньшие, чем точность машины (машинный нуль).

Аналогично выполняется расчет для рам других конфигураций. Так, для рамы, изображенной на рис. 15, а, последовательно уравниваются разрезы 1, 2, 3. Что касается нижней, регулируемой части рамы, то ее уравнивание выполняется поворотом ручки группового перекося при положении переключателя на ΔM .

Изложенный способ групповых перемещений хорошо приспособлен к машине ЭМСС-7 (7М), которая дает возможность находить моменты или величины ΔM в одной группе стержней, не интересуясь остальными. Это значительно облегчает расчет, так как для получения величины a_i и A_i достаточно знать значения ΔM только для i -го разреза, измерять остальные моменты нет необходимости. Если в состав разреза входит группа стоек одинаковой высоты, то мы получаем ΔM сразу для всей группы. Способ групповых перемещений можно было применить и при чисто аналитическом методе расчета, но без машины расчет получился бы весьма громоздким: требовалось бы полностью рассчитать раму на единичные смещения, определяя моменты (или деформации) во всех стержнях.

Для расчета нерегулярных рам на машине ЭМСС-7 могут применяться и другие способы, являющиеся частными случаями более общего метода групповых перемещений и приспособленные для расчета рам определенных видов. Для рамы, у которой можно выделить часть с большой относительной жесткостью, целесообразно применить способ основных жесткостей. Например, для рамы, изображенной на рис. 15, а, двухэтажная однопролетная часть имеет обычно неизмеримо большую жесткость, чем одиночная левая стойка. Поэтому при расчете рамы на машине можно поступить так: на поле изгиба собрать всю раму, левую стойку вывести на отдельную схему перекоса. В начале принимаем перекося левой стойки равным нулю и уравниваем первый и второй разрезы при условии, что смещаются первый и второй ярусы. Так как двухэтажная часть является регулярной, то такое уравнивание легко выполняется поворотом ручек соответствующих перекосов при положении переключателя на ΔI . Смещение левой стойки 3 должно быть равно сумме смещений ярусов 1 и 2 двухэтажной рамы, исходя из чего, определяем величину перекоса левой стойки и фиксируем ее на машине. Если теперь включить в первый и второй разрезы также и левую стойку, то получим некоторые неуравновешенные силы, которые вновь уравниваются с помощью ручек перекосов ярусов 1 и 2 , после чего вносится поправка для перекоса левой стойки. При желании, можно определить новые значения неуравновешенных сил и продолжить процесс, но обычно их величина после второго цикла является машинным нулем. Величина ΔI для уравнивания основной, регулярной части рамы определяется по формуле

$$\Delta I = A_i h_i, \quad (6)$$

где A_i — находится по формуле (3); h_i — высота яруса.

Метод основных жесткостей удобен тем, что расчет основной части рамы можно выполнить на машине без дополнительных аналитических вычислений. Однако применение этого метода ограничивается случаем, когда возможно выделить основную, наибольшую жесткость.

Для расчета рамы, в состав которой входят элементы в виде раскосных ферм (рис. 15, в), необходимо рассчитать раму при условии, что узлы фермы линейно неподвижны. При определении же перемещений узлов фермы можно пренебречь, как это обычно делается, влиянием узловых моментов. Вычисление перемещений в статически определимой ферме с шарнирами в узлах выполня-

ется аналитически или с помощью диаграммы перемещений. После того как перекосы всех стержней фермы найдены, они вводятся в машину в качестве внешнего воздействия. Расчет рам, в состав которых входят безраскосные фермы (рис. 15, с), по существу не отличается от расчета регулярных рам. Перекосы панелей ригелей уравниваются по величинам ΔI , найденным из рассмотрения отдельной безраскосной фермы.

В работе автора [104], перепечатанной в [45], дано много примеров расчета нерегулярных рамных систем различных видов на машине ЭМСС-7. Несложная аналитическая часть расчета выполняется в стандартной таблице уравнивания.

АЛГОРИТМИЧЕСКИЙ ЯЗЫК ИНЖЕНЕРНОГО ПРОГРАММИРОВАНИЯ КОМПАКТНЫЕ АЛГОРИТМЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

III. I. Для списания алгоритмов, предназначенных для выполнения на клавишных машинах, могут быть применены обычные словесные указания, такие, например, как в предыдущей главе. Однако такое описание имеет существенные недостатки: обучающийся должен быть знаком с основами предмета, к которому относится алгоритм; словесное объяснение редко бывает точным и может быть неверно понято, а для сложных алгоритмов — громоздким; требуется запоминание и т. д. В связи с этим нами был разработан специальный алгоритмический язык, ориентирующий на табличную (матричную) запись цифрового материала. Первая публикация об этом языке [100] отражала опыт ГПИ Укр-проектстальконструкция по выполнению расчетов на машино-счетной станции (1958—1960 гг.). Более подробное описание языка инженерного программирования (под этим названием он получил распространение в проектной практике) и применение его к расчету сооружений дано в работах автора [101, 102]. Основное внимание уделено построению типовых программ из стандартных блоков.

В дальнейшем В. Н. Гордеев предложил специальную индексную запись формул и способ представления их в виде блоков инженерной программы. Появились первые сообщения о международном алгоритмическом языке АЛГОЛ-60 для электронных цифровых машин. При ближайшем рассмотрении была установлена идейная близость языка инженерного программирования и языка АЛГОЛ-60.

Специфика первого, более простого, языка вызвана особенностями работы человека на клавишных автоматах, необходимостью краткой записи компактных схем. Вместе с тем было признано целесообразным внести в язык инженерного программирования некоторые элементы АЛГОЛа, главным образом относящиеся к описанию логических условий. Эта последняя редакция алгоритмического языка для клавишных автоматов и использована в настоящей главе.

Общие сведения о работе клавишных вычислительных машин

III. 2. Общий вид клавишного автомата типа *Soemtron* (ГДР), изображен на рис. 19. Арифметическое устройство этой машины автоматически выполняет четыре арифметических действия — сложение, умножение, вычитание, деление.

Числа набираются на клавиатуре 11, движок 2 служит для установки запятой между целой и дробной частями числа. Правильность набора проверяется по показанию контрольной шкалы 4.

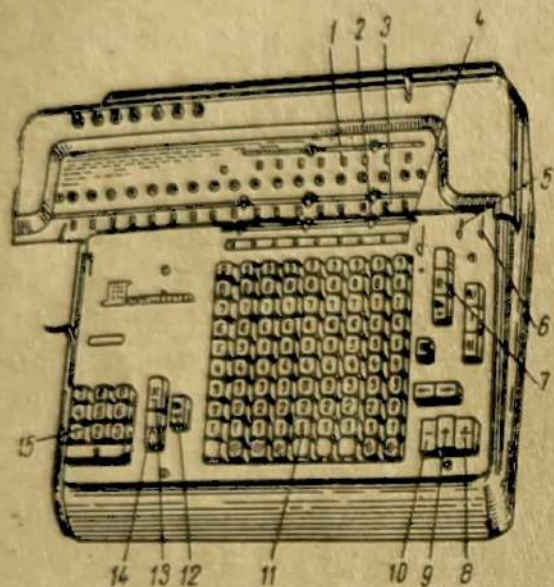


Рис. 19.

Для выполнения операции сложения (вычитания) набираем сначала первое слагаемое (уменьшаемое) на клавиатуре 11 и нажимаем на клавишу 9 со знаком +, число переносится в счетчик результатов 3. Клавиатура при этом автоматически гасится (клавиши возвращаются в первоначальное положение), если не нажата клавиша 7 с надписью *R*. Далее набираем на клавиатуре 11 второе слагаемое (или вычитаемое), нажимаем клавишу +

(при вычитании —) и результат переносится с суммировкой в счетчик результатов 3. Аналогично набираются последующие слагаемые (вычитаемые) и в счетчике 3 получаем результаты последовательного накопления суммы чисел.

Для выполнения операции умножения набираем множимое на клавиатуре 11, множитель — на десятиклавишной клавиатуре 15 и нажимаем клавишу 14 со знаком $\times$. Произведение набранных чисел читается в счетчике 3.

Для выполнения операции деления набираем делимое на клавиатуре 11 и нажимаем на клавишу 12, число переносится в счетчик результатов 3. Затем набираем делитель на клавиатуре 11 и нажимаем на клавишу 8 со знаком $\div$. Результат читается в счетчике частного 1.

Очень важным свойством клавишного автомата является возможность автоматически выполнять групповую операцию суммирования попарных произведений вида:

$$S = a_1 \times b_1 \pm a_2 \times b_2 \pm a_3 \times b_3 \pm \dots$$

Для получения суммы произведений рычаг 5 с надписью II ставим в верхнюю позицию (благодаря этому выключается автоматическое погашение показаний счетчика результатов 3), набираем на клавиатуре 11 число a_1 , на клавиатуре 15 число b_1 , нажимаем на клавишу 14 со знаком $\times$, результат переносится в счетчик 3. Затем набираем на клавиатуре 11 число a_2 , на клавиатуре 15 — число b_2 ; нажимаем на клавишу 14 со знаком $\times$ (при разности произведений — на клавишу 13 со знаком $\times$), результат суммы попарных произведений переносится в счетчик 3. Аналогично набираются последующие числа $a_3, a_4, b_3, b_4, \dots$ и в счетчике 3 автоматически получают результаты последовательного накопления сумм произведений.

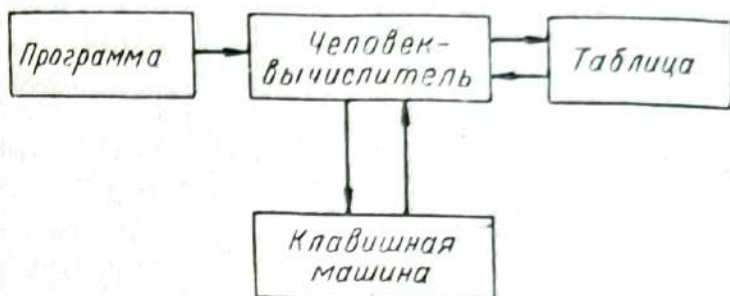


Рис. 20.

Аналогично накапливаются суммы частных при верхнем положении рычага 6 с надписью I.

Таким образом, машина дает возможность автоматически производить любые арифметические действия. Управление работой машины выполняется вручную — ввод исходных данных и запись получаемых результатов производится человеком. После выполнения каждого арифметического или группового действия необходимо записать результат в таблицу, так как для запоминания чисел в машине имеется только один счетчик результатов, как бы одна ячейка оперативной памяти. Внешней памятью для машины служат таблицы, количество и размеры которых не ограничены.

Схему работы машины можно представить следующим образом (рис. 20). Техник-вычислитель получает от инженера задание (это табличные бланки с пронумерованными строками и графами, в которых записаны исходные данные) и программу на алгоритмическом языке. Управляет вычислительной работой программа (требования, предъявляемые к программам, указаны в I. 10). С этой точки зрения весь комплекс, представленный на рис. 20, можно трактовать как программно управляемый, но через вычислителя, имеющего прямую и обратную связь с клавишной машиной и таблицей (памятью).

В соответствии с предписаниями программы вычислитель вводит числа из таблицы в машину, нажимает соответствующие клавиши и, после выполнения машиной вычислений, читает их и записывает в указанные программой клетки (ячейки) таблиц. Через определенные интервалы программа предписывает выполнять контрольные вычисления и, если проверка не удовлетворяется, — повторять заново некоторую часть расчета. Работа выполняется до тех пор, пока не закончится программа.

Научиться считать на машине нетрудно, но скорость работы натренированного вычислителя и человека, пользующегося машиной от случая к случаю, находятся примерно в таком же отношении, как скорости печати на пишущей машинке опытной машинистки и человека, печатающего одним пальцем. Опытный вычислитель за один час выполняет до 400 вычислений типа сложения и 300 — типа умножения-деления, а всего за смену, включая групповые операции с накоплением, до 2000 вычислений.

Переход на электронные клавишные автоматы типа ЛАДА, ВЕГА (по внешнему виду они похожи на электромеханические, но в счетчике результатов цифры более крупные и светящиеся *) позволит еще более ускорить расчет, но не очень значительно, так как основное время занимает не счет, а неавтоматизированные операции: чтение, ввод, запись результатов в таблицу и исправление ошибок.

Качественный скачок мы получаем только при переходе на быстродействующие ЭЦВМ с автоматическим программным управлением (гл. V).

Символы алгоритмического языка для операторов клавишных автоматов

III. 3. В общем случае расчет сооружения выполняется во многих таблицах. Поэтому обозначения элементов таблиц должны быть такими, чтобы с их помощью можно было точно указать местоположение (адрес) любого числа или ряда чисел в разных табличных бланках.

В соответствии с этим прием следующие обозначения. Расчетную таблицу обозначим символом T_N . T — таблица, индекс N указывает номер таблицы и может быть заменен цифрами или буквами 1, 2 ..., А, Б и т. д. Например, T_1 — табл. 1, T_A — табл. А.

* Промежуточное положение занимают электрические автоматы типа «Вильнюс», у которых развита электрическая схема (каретка неподвижна, но счетчики такие же, как на электромеханических машинах).

Бланк таблицы состоит из горизонтальных строк, вертикальных граф (столбцов) и клеток.

Обозначение строки: $|\text{стр. } a T_N|_{n_1}^{+\nu}$ читается так — строка a табл. N в пределах от графы n_1 до графы $n_1 + \nu$ (для сокращения записи индекс n_1 в верхнем пределе не повторяется). Количество клеток в данном отрезке строки равно $\nu + 1$.

Например, если $a=2$, $N=1$, $n_2=1$, $\nu=3$, то получим: $|\text{стр. } 2T_1|_1^4$.

Для схемы 1 это числа: 5,0; 10,0; — 5,0; 8,0.

Схема 1

T ₁						T ₂	
	1	2	3	4	5	1	2
1	20,0	—5,0	3,0	—2,0	16,0		
2	5,0	10,0	—5,0	8,0	18,0		
3	4,0	—6,0	15,0	—4,0	9,0	19,0	—10,0
4	1,0	5,0	—6,0	20,0	20,0	—5,0	25,0
5	30,0	4,0	7,0	22,0			

1	2	3	4	5
5,0	7,0	1,0	2,0	3,0

Обозначение графы: $|\text{гр } a T_N|_{m_1}^{+\nu}$ читается так — графа a табл. N .

в пределах от строки m_1 до строки $m_1 + \nu$. Например, если $a=3$, $N=1$, $m_1=1$, $\nu=3$, то получим: $|\text{гр } 3T_1|_1^4$.

Для схемы 1 это числа: 3,0; — 5,0; 15,0; — 6,0.

Когда говорится о всей строке таблицы, то обозначения пределов можно опустить. Так, для схемы 1 обозначения стр $2T_1$ без пределов охватывают числа 5,0; 10,0; — 5,0; 8,0; 18,0:

Обозначения таблиц можно ставить как внутри прямых скобок, так и вне их. Например, можно написать $|\text{стр } 2|_1^4 T_1$ или $|\text{стр } 2T_1|_1^4$. В некоторых случаях удобно выделить часть строки, расположенную слева или справа от главной диагонали таблицы.

Схема 2

T₂

	1	2	3	4	5	6
1	20,0	3,0	2,0	1,0		
2	3,0	10,0	3,0	2,0	1,0	
3	2,0	3,0	2,0	3,0	2,0	1,0
4	1,0	2,0	3,0	15,0	3,0	2,0
5		1,0	2,0	3,0	20,0	3,0
6			1,0	2,0	3,0	15,0

В этих случаях у номера строки ставится индекс л и п. Например, стр 3_л T₂ — часть строки 3, расположенная слева от главной диагонали табл. 2 (для схемы 2 это числа 2,0 и 3,0); стр 3_п T₂ — часть строки 3, расположенная справа от главной диагонали (для схемы 2 это числа 3,0; 2,0 и 1,0).

Клетка таблицы находится на пересечении строки и графы и обозначается двумя индексами (буквами или цифрами), разделенными точкой *; первая буква (цифра) указывает номер строки, вторая — номер графы. Например:

$i.kT_N$ — клетка на пересечении строки i и графы k табл. N ,

$k.iT_N$ — клетка на пересечении строки k и графы i табл. N .

Так, для схемы 1 в клетке 2.3T₁ находится число — 5,0, а в клетке 3.2T₁ — число — 6,0.

Если таблица одномерная и состоит из одной строки или графы, то она обозначается одной буквой, заменяющей также обозначение таблицы (символ T не пишется). Например, отдельная строка a на схеме 1 обозначается |стр a |₁⁵. В клетке $a.2$ этой таблицы содержится число 7,0.

Введем также понятие отрезка диагонали.

$|DT_N|_{m_1 \cdot a_1}^{+v}$ — отрезок диагонали от клетки $m_1.n_1$ до клетки $m_1 + v.p_1 + v$.

* В языке АЛГОЛ-60 индексы разделяются запятой, так как в десятичных дробях в западных странах вместо запятой ставится точка. Наши обозначения соответствуют принятым в СССР.

Если речь идет о главной диагонали, то $m_1 = n_1$ и один из индексов в нижнем пределе можно убрать.

$DT_N|_{m_1}^{+v}$ — отрезок главной диагонали от строки (графы) m_1 до строки (графы) $m_1 + v$.

Например, если $N=2$, $m_1=2$ и $v=3$, то получим отрезок главной диагонали $|DT_2|_2^5$. Для схемы 2 это числа 10,0; 2,0; 15,0; 20,0.

Для выделения части таблицы (групповой клетки) применяется обозначение: $|m_1 : m_2 . n_1 : n_2|T_N$. Читается так — часть таблицы, находящаяся в пределах от строки m_1 до строки m_2 и от графы n_1 до графы n_2 . Например, если $N=2$, $m_1=3$, $m_2=5$, $n_1=2$, $n_2=6$, то получим: $|3 : 5 . 2 : 6|T_2$. Для схемы 2 это выделенная часть таблицы. Символы арифметических и логических операций такие же, как в математике.

Заметим, что принятые обозначения обладают наглядностью, они рассчитаны на человека, зрительно воспринимающего таблицы; в этом их особенность и отличие от обозначений алгоритмических языков для ЭЦВМ.

Арифметические и логические выражения

III. 4. Символы языка, объединенные знаками арифметических операций, образуют арифметическое выражение. Основные типы арифметических выражений даны в табл. III. 1.

Рассмотренные групповые действия описывают матричные операции линейной алгебры (операции над векторами). Однако сокращенные обозначения, применяемые в линейной алгебре, не всегда могут быть непосредственно использованы в программах вычислений, так как в них отсутствует информация о том, какой вектор (столбец или строка) изображается данным символом, к какой таблице он относится и каковы его размеры и расположение. Достаточная информация содержится лишь в полных изображениях векторов, которые из-за их громоздкости неудобно использовать в программах.

III. 5. Символы, языка, объединенные знаками логических операций ($=$, $\neq$, $<$, $\leq$, $>$, $\geq$), образуют логическое выражение. Например: $A=B$, $A \neq B$, $A < B$ и т. д. A и B — арифметические выражения.

Возможны также более сложные логические высказывания, но в программах, предназначенных для техника-вычислителя, их применяют редко.

Таблица III. 1

№ пп.	Арифметическое выражение	Содержание
1	$\Sigma \text{ стр } i T_N \big _{n_1}^{+v}$	Суммирование чисел в заданных пределах строки i
2	$\Sigma_{\text{по}+} \text{ стр } i T_N \big _{n_1}^{+v}$	Суммирование положительных чисел в строке i
3	$\Sigma_{\text{по}-} \text{ стр } i T_N \big _{n_1}^{+v}$	Суммирование отрицательных чисел в строке i
4	$\Sigma_{\text{по}+\text{стр}k} \text{ стр } i T_N \big _{n_1}^{+v}$	Суммирование чисел строки i , находящихся в одних графах с положительными числами строки k
5	$\max_{(\min)} \text{ стр } i T_N \big _{n_1}^{+v}$	Максимальное (минимальное) число в заданных пределах строки i
6	$\text{absmax}_{(\text{absmin})} \text{ стр } i T_N \big _{n_1}^{+v}$	То же, по абсолютному значению
7	$\text{стр } \alpha T_A \big _{n_1}^{+v} \times \text{стр } \beta T_B \big _{n_2}^{+v}$	Умножение соответствующих чисел строки α табл. А и строки β табл. В
8	$\text{стр } \alpha T_A \times \text{стр } \beta T_B \big _{n_2}^{+v}$	То же, при одинаковых пределах для строки α табл. А и строки β табл. В
9	$\Sigma \text{ стр } \alpha T_A \times \text{стр } \beta T_B \big _{n_1}^{+v}$	Суммирование попарных произведений соответствующих чисел строки α табл. А и строки β табл. В

Примечание. Если в выражениях 1—9 заменить обозначение строки (стр.) на обозначение графы (гр.), те же действия будут выполняться для чисел, расположенных в графах.

Операторы засылки и цикла

III. 6. Основной единицей языка, выражающей законченное действие (приказ), является оператор (или формула) засылки:

$$A \rightarrow b,$$

где A — арифметическое выражение,

b — адреса элементов, куда должны быть записаны результаты вычислений.

Ряд характерных операторов приведен в табл. III. 2.

Таблица III.2

№ пп.	Оператор	Содержание	Примечания																														
	$2.5 T_2 \rightarrow 5.2 T_2$	Записать число из клетки 2.5 табл. 2 в клетку 5.2 той же таблицы	Для схемы 2 это число равно 1,0																														
2	$ \text{стр } 2 T_2 _1^4 \rightarrow \text{гр } 2 T_2 _4^1$	Записать числа из строки 2 табл. 2 в графу 2 той же таблицы	Для схемы 2: числа из строки 2 перенесены в симметричные клетки графы 2																														
3	$\Sigma \text{стр } 3 T_1 _1^4 \rightarrow 3.5 T_1$	Просуммировать числа строки 3 табл. 1 в заданных пределах и результаты записать в клетку 3.5 той же таблицы	Для схемы 1: $4,0-6,0+15,0-4,0=9,0$																														
4	$ \text{стр } 1+\text{стр } 2+\text{стр } 3+\text{стр } 4 _1^4 T_1 \rightarrow \text{стр } 5 T_1$	Просуммировать соответствующие числа строк 1, 2, 3 и 4 табл. 1 и результаты записать в строку 5 той же таблицы	Для схемы 1: <table><tr><td>20,0</td><td>—</td><td>5,0</td><td>3,0</td><td>—</td><td>2,0</td></tr><tr><td>+5,0</td><td>—</td><td>+10,0</td><td>—</td><td>5,0</td><td>+ 8,0</td></tr><tr><td>+4,0</td><td>—</td><td>— 6,0</td><td>+15,0</td><td>—</td><td>— 4,0</td></tr><tr><td>+1,0</td><td>—</td><td>+ 5,0</td><td>— 6,0</td><td>—</td><td>+20,0</td></tr><tr><td>30,0</td><td></td><td>— 4,0</td><td>7,0</td><td></td><td>22,0</td></tr></table>	20,0	—	5,0	3,0	—	2,0	+5,0	—	+10,0	—	5,0	+ 8,0	+4,0	—	— 6,0	+15,0	—	— 4,0	+1,0	—	+ 5,0	— 6,0	—	+20,0	30,0		— 4,0	7,0		22,0
20,0	—	5,0	3,0	—	2,0																												
+5,0	—	+10,0	—	5,0	+ 8,0																												
+4,0	—	— 6,0	+15,0	—	— 4,0																												
+1,0	—	+ 5,0	— 6,0	—	+20,0																												
30,0		— 4,0	7,0		22,0																												
5	$ \text{гр } 1+\text{гр } 2+\text{гр } 3+\text{гр } 4 _1^4 T_1 \rightarrow \text{гр } 5 T_1$	Просуммировать соответствующие числа граф 1, 2, 3 и 4 табл. 1 и результаты записать в графу 5 той же таблицы	Для схемы 1: <table><tr><td>20,0</td><td>—</td><td>5,0</td><td>+ 3,0</td><td>—</td><td>2,0=16,0</td></tr><tr><td>5,0</td><td>+10,0</td><td>— 5,0</td><td>+ 5,0</td><td>+ 8,0</td><td>=18,0</td></tr><tr><td>4,0</td><td>— 6,0</td><td>+15,0</td><td>+15,0</td><td>— 4,0</td><td>= 9,0</td></tr><tr><td>1,0</td><td>+ 5,0</td><td>— 6,0</td><td>— 6,0</td><td>+20,0</td><td>=20,0</td></tr></table>	20,0	—	5,0	+ 3,0	—	2,0=16,0	5,0	+10,0	— 5,0	+ 5,0	+ 8,0	=18,0	4,0	— 6,0	+15,0	+15,0	— 4,0	= 9,0	1,0	+ 5,0	— 6,0	— 6,0	+20,0	=20,0						
20,0	—	5,0	+ 3,0	—	2,0=16,0																												
5,0	+10,0	— 5,0	+ 5,0	+ 8,0	=18,0																												
4,0	— 6,0	+15,0	+15,0	— 4,0	= 9,0																												
1,0	+ 5,0	— 6,0	— 6,0	+20,0	=20,0																												

Продолжение табл. III.2

№ пп.	Оператор	Содержание	Примечания
6	$T_1 + T_2 \rightarrow T_3$ ($\times$)	Просуммировать (вычесть, умножить, разделить) соответствующие числа табл. 1 и 2 и результаты записать в табл. 3	При такой записи размеры таблицы 1, 2 и 3 должны быть одинаковыми, в противном случае следует указать размеры частей таблиц
7	$\text{гр } 1 \text{ } T_1 \times \text{гр } 3 \text{ } T_2 \mid_1^6 \rightarrow \text{гр } 1 \mid_1^6 T_3$	Графу 1 табл. 1 совместить с графой 3 табл. 2 и попарные произведения рядом расположенных чисел записать в графу 1 табл. 3	Для схемы 3: $1,0 \times 3,0 = 3,0$; $2,0 \times 4,0 = 8,0$; $5,0 \times (-1,0) = -5,0$ $7,0 \times 2,0 = 14,0$; $4,0 \times 3,0 = 12,0$; $-6,0 \times 5,0 = -30,0$
8	$\Sigma \mid \text{гр } 1 \text{ } T_1 \times \text{гр } 3 \text{ } T_2 \mid_1^6 \rightarrow 1.3 T_5$	Просуммировать попарные произведения соответствующих чисел графы 1 табл. 1 и графы 3 табл. 2 и результаты записать в клетку 1.3 табл. 5	Для схемы 3: $1,0 \times 3,0 + 2,0 \times 4,0 + 5,0 \times (-1,0) +$ $+ 7,0 \times 2,0 + 4,0 \times 3,0 - 6,0 \times 5,0 = 2,0$
9	$\Sigma \mid \text{стр } 1 \text{ } T_B \times \text{стр } 4 \text{ } T_A \mid_4^7 \rightarrow 4.8 T_B$	Просуммировать попарные произведения соответствующих чисел строки 1 табл. В и строки 4 табл. А и результаты записать в клетку 4.8 табл. А	Для схемы 4: $0 \times 10,0 + 3,2 \times 1,0 - 1,5 \times 2,0 +$ $+ 2,1 \times 1,0 = 2,3$
10	$- \mid \text{стр } 3 \text{ } T_6 \mid_3^7 / 3.3 T_6$ $\rightarrow \mid \text{стр } 3 \text{ } T_5 \mid_3^7$	Разделить каждое число строки 3 табл. 6 на число в клетке 3.3 той же таблицы и результаты записать с обратным знаком в строку 3 табл. 5	Для схемы 5: $-20,0/20,0 = -1,0$; $-(-60,0)/20,0 = 3,0$; $-20,0/20,0 = -1,0$; $-(-80,0)/20,0 = 4,0$; $-(-100,0)/20,0 = 5,0$

Изложенные выше символы достаточны для записи произвольных вычислительных процедур. Однако словарный состав языка может быть пополнен в тех случаях, когда вносимые дополнения приводят к существенному упрощению или сокращению записи.

Например, для записи численного интегрирования оказалось целесообразным ввести специальную операцию умножения граф по диагонали, которая не имеет аналога в матричном исчислении. Каждая пара строк таблицы выделена линиями (строки 1—2, 3—4, 5—6... на схеме 3), числа в графах умножаются крест-накрест (по диагонали) между выделенными линиями и результаты суммируются.

Схема 3

	$1T_1$	$3T_2$	$1T_3$	$3T_5$
1	1,0	3,0	3,0	2,0 (15,0)
2	2,0	4,0	8,0	
3	5,0	-1,0	-5,0	
4	7,0	2,0	14,0	
5	4,0	3,0	12,0	
6	-6,0	5,0	-30,0	
7	2,0	3,0		
8	—	4,0		

Схема 4

	4	5	6	7	8
$4T_A$	10,0	1,0	2,0	1,0	2,3
$1T_B$		3,2	-1,5	2,1	

Схема 5

	3	4	5	6	7
$3T_5$	-1,0	3,0	-1,0	4,0	5,0
$3T_6$	20,0	60,0	20,0	-80,0	-100,0

Запись

$$\Sigma | \text{gr} 1T_1 \times \text{gr} 3T_2 |_{\text{диаг}}^6 \rightarrow 1.3T_5$$

для схемы 3 реализуется следующим образом:

$$(1,0 \times 4,0 + 3,0 \times 2,0) + (5,0 \times 2,0 - 1,0 \times 7,0) + \\ + (4,0 \times 5,0 - 3,0 \times 6,0) = 15,0.$$

Результат записан в клетке 1.3 табл. 5 (в скобках).

Данная групповая операция несколько необычна, но легко воспринимается зрительно и запоминается. Применять ее очень удобно, так как существенно сокращаются промежуточные записи.

III.7. Номера элементов матриц (клеток, строк, граф, таблиц) могут входить в вычислительную формулу как переменные величины, характер изменения которых указывается отдельно. По этому принципу строится так называемый оператор цикла. В заголовке оператора цикла указывается список значений переменных, называемых параметрами цикла, затем следует сам оператор.

Пример оператора цикла:
 для $i := 1, 2, \dots, n$ выполнить
 для $k := i, i+1, \dots, n$ выполнить

$$\Sigma | \text{гр } i \text{ } T_A \times \text{гр } k \text{ } T_B |_1^m \rightarrow i.k T_B \rightarrow k.i T_B.$$

Это так называемый цикл в цикле: цикл для k входит в цикл для i .

Вместо i в формулу подставляются значения, указанные в первом списке, причем для каждого значения i величина k принимает последовательно все значения, указанные во втором списке.

Знак $(:=)$ заимствован из языка АЛГОЛ и означает присваивание. Между знаками равенства $(=)$ и присваивания $(:=)$ имеется смысловое различие. Выражение $i=1$ означает утверждение в том, что величина i равна единице. Выражение $i:=1$ означает предписание, действие — величине i присвоить значение, равное единице (величину i принять равной единице).

В самом деле, величина i не может быть одновременно равна $1, 2, \dots, n$, но величине i может быть поочередно присвоено значение $1, 2, \dots, n^*$.

Для $i:=1, K:=1$ получим

$$\Sigma | \text{гр } 1 \text{ } T_A \times \text{гр } 1 \text{ } T_B |_1^m \rightarrow 1.1 T_B.$$

Для $i:=1, k:=2$, получим

$$\Sigma | \text{гр } 1 \text{ } T_A \times \text{гр } 2 \text{ } T_B |_1^m \rightarrow 1.2 T_B \rightarrow 2.1 T_B \text{ и т. д.}$$

Переменными величинами могут быть также наименования таблиц, пределы изменения граф (строк) и т. д.

Возможно также, как это делается на языке АЛГОЛ, заголовок записать так:

для $i := 1$ шаг 1 до n выполнить
 для $k := i$ шаг 1 до n выполнить

* В языке АЛГОЛ-60 вместо знака ссылки также применяется знак присваивания, левая и правая части формулы меняются местами. В упрощенном языке инженерного программирования сохранен более наглядный символ ссылки. Операция присваивания указывается только в заголовке цикла.

Однако, поскольку программа предназначена для человека, то перечисление с применением многоточия кажется более наглядным.

Рассмотренный оператор цикла имеет тип арифметической прогрессии. Шаг изменения параметра может быть равен не единице, а любому другому числу, но последнее в инженерных программах бывает редко.

Для описания итерационных процессов будем применять оператор цикла типа пересчета, который записывается так:

для $i := \dots$ пока $\alpha > \varepsilon$ выполнить <оператор>;

для значений параметра i , указанного в заголовке, выполнять оператор цикла до тех пор, пока соблюдается условие $\alpha > \varepsilon$ (например, пока погрешность будет больше требуемой точности расчета).

III. 8. При составлении программ для вычислителей клавишных машин имеются две противоположные тенденции. С одной стороны, наибольшей простотой отличается развернутая программа, состоящая из относительно большого числа простейших приказов и проверок, что обеспечивает автоматизм вычислений. Этот путь выбран в работе [102]. С другой стороны, желательно выразить программу по возможности более кратко [101, 105], так как длинной программой не всегда удобно пользоваться, ее труднее написать, издать и т. д. * Для краткого выражения программы приходится применять сокращенные обозначения, в частности, оператор цикла. При выполнении программы вычислитель должен в уме осуществлять логику счета, подставляя вместо индексов их числовые значения в соответствии с закономерностями, указанными в заголовке. Это требует от вычислителя определенных навыков.

Сравнительно простые циклы (одно-, двухкратные) усваиваются вычислителями достаточно быстро и пользование ими не замедляет счета. Лучшая обозримость отдельных более или менее крупных блоков даже облегчает чтение программы.

Попытка использовать при работе на клавишных автоматах более сложные понятия АЛГОЛа (например, сложные индексные выражения) привела к тому, что даже опытный вычислитель больше времени тратил на обдумывание разных чисел (указанных сложными индексными выражениями), чем на сам счет. Хотя количество операций при этом иногда удавалось уменьшить, но

* Однако этот вопрос стоит менее остро, чем для ЭЦВМ, где необходимо экономить память; в данном случае экономится бумага.

в целом сроки выполнения расчетов резко возросли. Таким образом, усложнение языка для клавишных машин имеет определенные пределы. При разработке языка были по возможности учтены особенности работы на этих машинах.

Строение программы, операторы перехода и проверки, составной оператор

III. 9. Программа представляет собой набор вычислительных операторов, разделенных точками с запятыми и выполняемых в той последовательности, в которой они записаны. В начале программы пишется слово **начало**, в конце программы слово **конец**.

В некоторых случаях естественный порядок выполнения операторов должен быть нарушен, так как необходимо перейти не к следующему по порядку, а к другому оператору программы. Для этого оператор, на который следует перейти, снабжается меткой, представляющей собой букву, цифру или любое их сочетание, за которыми следует двоеточие. Например:

M1 : <оператор>.

В качестве метки можно использовать фразу, имеющую mnemonicский смысл:

контрольное вычисление: <оператор>.

В данном случае меткой служат слова «контрольное вычисление», разъясняющие смысл выполняемых затем действий.

Указание о переходе на данную метку осуществляется с помощью оператора перехода. Например: на M1 (или на контрольное вычисление).

Если переход на ту или иную формулу ставится в зависимость от выполнения определенного условия, то применяется условный оператор перехода, который записывается, например, так:

если $a \leq b$ то на M1 иначе на M2;

либо: если $a \neq b$ то на M1;

Последнее выражение означает: если $a \neq b$, то перейти на метку M1, в противном случае — на следующий оператор программы.

III. 10. При составлении программы для клавишных машин очень важно организовать последовательную проверку небольших вычислительных этапов, что дает возможность в процессе счета своевременно обнаружить и исправить ошибки; без этого практически нельзя реализовать сложный алгоритм. Примером могут служить построчные проверки в алгоритме Гаусса, без ко-

торых практически невозможно решать вручную системы уравнений со многими неизвестными.

Проверки можно записать в виде условного оператора по следующей схеме:

```
М1 : . . . . . ;  
      . . . . . ;  
если  $a \neq b$  то на М1;  
      . . . . . ;
```

Вычислительные формулы обозначены многоточиями. После выполнения ряда формул следует проверка. Если $a = b$, переходим к следующей по порядку формуле, т. е. продолжаем вычисления по программе. Если $a \neq b$, возвращаемся к метке М1, повторяем заново предыдущий этап вычислений и исправляем ошибку.

В тех случаях, когда условные операторы служат для контроля последовательных этапов вычислений, их можно заменить более наглядной конструкцией с применением слова **проверка** и исключить метки:

```
. . . . . ;  
. . . . . ;  
Проверка  $a = b$  ;  
. . . . . ;  
Проверка  $c = d$  ;  
. . . . . ;
```

Такая запись означает: если условие, следующее после слова **проверка**, не соблюдается, необходимо повторить предшествующий этап вычислений (между двумя проверками).

В некоторых случаях целесообразно группу операторов объединить и рассматривать как один оператор по отношению к другим операторам программы. Такую группу операторов, заключенную в операторные скобки из слов **начало** и **конец** или фигурные скобки, называют составным оператором. В частности, составным оператором может быть оператор цикла. Это значит, что указанные в заголовке значения параметров цикла относятся не к одной, а к группе формул, заключенной в фигурные (или операторные) скобки.

III.11. В языке инженерного программирования используются служебные слова, заимствованные из языка АЛГОЛ: **начало** (по-английски **begin**), **конец** (**end**), **шаг** (**step**), **до** (**until**), **выполнить** (**do**), **если** (**if**), **то** (**then**), **иначе** (**else**), **пока** (**while**). Эти слова подчеркиваются (или печатаются полужирным шрифтом).

Введены дополнительно слово **проверка (control)** и символы: стр (строка — line), гр (графа — column или сокращенно col), Т (таблица — table или сокращенно T).

Если необходимо дать какие-либо пояснения к программе, то их будем писать после служебного слова **примечание (comment)** или после слова **конец** в конце программы. После окончания текста примечаний ставится точка с запятой.

Таким образом, приближаем конструкцию языка инженерного программирования к простым формам АЛГОЛа, сохранив специфические особенности, связанные с выполнением вычислений на клавишных машинах.

Правила составления программы для клавишных автоматов

III. 12. Очень важно сформулировать правила составления программ и определить последовательность их применения. Укажем наиболее важные из этих правил.

1. Необходимо выбрать метод расчета и записать расчетные формулы, ориентируясь на табличное размещение числового материала.

Эту задачу можно решить различными способами.

Обычный способ состоит в том, чтобы выписать расчетные формулы в той последовательности, в которой их надлежит выполнять.

Другой способ использует матричные формулы, в которых уже определены таблицы (матрицы) для выполнения вычислений. Такая запись позволяет весьма кратко описать схему алгоритма.

В третьем способе [22] применяются специальные индексные обозначения в расчетных формулах, приспособленных к составлению программы.

Наконец, алгоритм может быть полностью или частично описан словесно. Такое описание применяется, когда в алгоритме много неарифметических операций.

2. Нумеруются табличные бланки и их элементы — строки и графы. Составляются числовые формулы программы (операторы засылки).

Если заданием для составления программы являются матричные формулы, то операции умножения строк на графы должны быть заменены умножением одноименных элементов (строк на строки или граф на графы), так как последние операции для вычислителя удобнее. Это нетрудно сделать, соответствующим

образом расположив числовой материал в таблицах. Умножение вектора (графы или строки) на диагональную матрицу заменяется умножением графы (или строки) на диагональ (III. 3) без накопления. Такая замена позволяет сократить размеры табличных бланков и их число.

В ряде случаев оказывается возможным несколько матричных операций заменить одной. Например, численное интегрирование удобно описывается специальной операцией суммирования произведений чисел в графах по диагоналям (III. 6). Возможно введение и других специальных операций с соответствующими обозначениями.

3. Продумывается система проверок. Наиболее распространенный вид проверок — с помощью контрольных сумм. Для этого в табличных бланках предусматриваются суммарные строки или графы, обозначаемые обычно буквой *S*. Помимо таких построчных (пографных) проверок желательно иметь еще интегральную, смысловую проверку расчета. Для выполнения проверок и исправления ошибок при их обнаружении используются условные операторы.

4. Операторы программы располагаются в той последовательности, в которой их следует выполнять, и разделяются точками с запятыми. Над программой пишется ее наименование или указывается номер. Программа ограничивается словами **начало** и **конец**.

В программе, если нужно, даются словесные пояснения, помещаемые в любом месте после служебного слова **примечание** либо после слова **конец**.

5. В некоторых случаях словесные указания могут оказаться для вычислителя более простыми, чем формульная запись; это относится, главным образом, к указаниям о подготовке исходных данных, использования таблиц из книг и т. д. В таких случаях дается соответствующая словесная инструкция — словесный оператор, отделяемый от других операторов точкой с запятой.

6. Для составления больших программ весьма эффективен метод компоновки. Этот метод предполагает предварительное составление достаточно большой библиотеки стандартных программ, описывающих те или иные матричные и другие процедуры. При наличии такой библиотеки сложная программа будет состоять из расположенных в определенном порядке наименований процедур (операторов процедур) и некоторых компоновочных операторов — перехода и условных.

Сопоставление инженерного программирования и матричного исчисления

III. 13. Инженерное программирование можно назвать также матричным программированием, так как с его помощью описываются любые матричные операции. Однако дословное повторение нецелесообразно.

Инженерная программа имеет ряд средств, обусловленных спецификой работы на машинах: введение пределов, неарифметических операторов, организация проверок и т. д. Эти средства крайне необходимы при выполнении больших объемов вычислений, но они не нужны в теоретических исследованиях. Например, при преобразовании матричных формул, для доказательства теорем и т. п. незачем указывать пределы числовых массивов, не нужны также проверки вычислений и т. д.

Таким образом, матричное исчисление и инженерное программирование близки по содержанию, но выполняют разные функции, которые диктуют и различные формы записи. Сопоставление будет продолжено после рассмотрения ряда примеров.

III. 14. Приведем пример несложной программы преобразования матрицы к симметричному виду (так называемой первой трансформации Гаусса).

В матричных обозначениях указанная программа записывается кратко:

$$A' \times \bar{A} \rightarrow B$$

— умножить расширенную матрицу $\bar{A}$ (дополненную графой свободных членов) слева на транспонированную матрицу A' и результаты записать в матрицу B . Если A есть матрица системы уравнений, то преобразованная матрица B становится симметричной и положительно определенной.

О замечательных свойствах симметричных матриц положительно определенной формы и использовании этих свойств при решении систем линейных уравнений можно прочесть в работе [121]. Трансформация Гаусса находит применение не только для решения систем уравнений. В частности, как будет показано в дальнейшем, составление системы линейных уравнений, метода сил сводится к трансформации матрицы усилий, подобной трансформации Гаусса. Поэтому рассмотрим общий случай преобразования прямоугольной матрицы A с размерами $m \times n$. Для частного случая квадратной матрицы $m = n$. Расчетные бланки даны на схеме 6.

	1	2	...	n	$n+1$	$n+2$	...	$n+q$	S
1									
2									
3									
...									
m									

Таблица В

	1	2	...	n	$n+1$	$n+2$	...	$n+q$	S
1									
2									
...									
n									

Графы $1—n$ относятся к матрице коэффициентов, графы $n+1, \dots, n+q$ — к свободным членам, графы S — контрольные. В табл. А записываются исходные данные (переписываются или вычисляются по предшествующей программе), в графу S — суммы чисел в строках. Для операции заполнения контрольной графы (строки) S программа обычно не дается: вычислитель знает, что если в бланке имеется пустая графа (строка) S и дополнительных указаний не дано, то ее следует заполнить, суммируя все предшествующие числа.

Программа первой трансформации Гаусса:

начало

для $i := 1, 2, \dots, n$ выполнить

для $k := i, i+1, \dots, S$ выполнить

$\{\Sigma | \text{гр } i \times \text{гр } k |_i^m T_A \rightarrow i.k T_B;$

если $i \leq n$ то $i.k T_B \rightarrow k.i T_B$ };

проверка $\Sigma | \text{стр } i |_i^{n+q} T_B = i.S T_B$

конец программы.

III. 15. Рассмотрим выполнение этой программы на простом числовом примере (схеме 7); в этом состоит так называемая отладка программы.

Схема 7

Таблица А							Таблица В						
n							n						
	1	2	3	4	5	S		1	2	3	4	5	S
1	1	2	3	1	4	11	1	15	19	10	15	15	74
2	1	3	2	1	3	10	2	19					
3	2	4	1	2	1	10	3	10					
$m=4$	3	2	1	3	2	11	$n=3$	10					

В данном примере $n=3$, $m=4$, $q=2$. В графе S табл. А записаны суммы чисел в строках. Выполняя цикл, будем последовательно присваивать параметрам i и k значения, указанные в заголовке. Для $i := 1$ последовательно находим:

$$k := i = 1; \Sigma | \text{гр } 1 \times \text{гр } 1 |_1^m T_A = 1 \times 1 + 1 \times 1 + 2 \times 2 + 3 \times 3 = 15 \rightarrow 1.1 T_B;$$

так как $k \leq n$, то $1.1 T_B \rightarrow 1.1 T_B$;

Эта операция получает смысл в последующих вычислениях.
 $k := i+1 = 2; \Sigma | \text{гр } 1 \times \text{гр } 2 |_1^n T_A = 1 \times 2 + 1 \times 3 + 2 \times 4 + 3 \times 2 = 19 \rightarrow 1.2 T_B;$

так как $k \leq n$, то $1.2 T_B \rightarrow 2.1 T_B$;

$$k := 3; \Sigma | \text{гр } 1 \times \text{гр } 3 |_1^m T_A = 1 \times 3 + 1 \times 2 + 2 \times 1 + 3 \times 1 = 10 \rightarrow 1.3 T_B;$$

так как $k \leq n$ то $1.3 T_B \rightarrow 3.1 T_B$;

$$k := 4; \Sigma | \text{гр } 1 \times \text{гр } 4 |_1^m T_A = 1 \times 1 + 1 \times 1 + 2 \times 2 + 3 \times 3 = 15 \rightarrow 1.4 T_B.$$

Условие $k \leq n$ не соблюдается ($k > n$);

$$k := 5; \Sigma | \text{гр } 1 \times \text{гр } 5 |_1^m T_A = 1 \times 4 + 1 \times 3 + 2 \times 1 + 3 \times 2 = 15 \rightarrow 1.5 T_B;$$

$$k := S; \Sigma | \text{гр } 1 \times \text{гр } |_1^m T_A = 1 \times 11 + 1 \times 10 + 2 \times 10 + 3 \times 11 = 74 \rightarrow 1.5 T_B.$$

Цикл для k закончен.

$$\text{Проверка: } \Sigma | \text{стр } 1 |_1^S T_B = 15 + 19 + 10 + 15 + 15 = 74 = 1.5 T_B.$$

Поскольку проверка соблюдается, то принимаем $i=2$ и выполняем цикл для k , начиная с $k=i=2$. Если допущена ошибка, то проверка не удовлетворится * и вычисления в строке 1 придется повторить. Далее процесс продолжается аналогично.

В данной программе предусмотрен учет симметрии в матрице B (в пределах $n \times n$). Это потребовало введения еще одного условного оператора.

III. 16. Приведем для сравнения развернутую запись программы из III.14, которую следует применять при начальном обучении вычислителей.

Программа первой трансформации Гаусса (развернутая запись):

$$\Sigma | \text{гр } 1 \times \text{гр } 1 |_1^m T_A \rightarrow 1.1 T_B;$$

$$\Sigma | \text{гр } 1 \times \text{гр } 2 |_1^m T_A \rightarrow 1.2 T_B \rightarrow 2.1 T_B;$$

$$\Sigma | \text{гр } 1 \times \text{гр } S |_1^m T_A \rightarrow 1.S T_B;$$

$$\text{проверка } \Sigma | \text{стр } 1 |_1^{n+q} T_B = 1.S T_B;$$

примечание если проверка не удовлетворяется, то вычисление повторить;

* В самом деле, над всеми числами строки 1 табл. A , в том числе и над контрольной суммой $1.S T_A$ выполнялись одни и те же действия, и результаты записывались в строку 1 табл. B . Поэтому и в строке 1 табл. B , если вычисления выполнены верно, также должно соблюдаться равенство $\Sigma \text{стр } 1 T_B = 1.S T_B$.

$$\Sigma | \text{гр } 2 \times \text{гр } 2 |_1^m T_A \rightarrow 2.2 T_B ;$$

$$\Sigma | \text{гр } 2 \times \text{гр } 3 |_1^m T_A \rightarrow 2.3 T_B \rightarrow 3.2 T_B ;$$

$$\Sigma | \text{гр } 2 \times \text{гр } S |_1^m T_A \rightarrow 2.S T_B ;$$

$$\text{проверка } \Sigma | \text{стр } 2 |_1^{n+q} T_B = 2.ST_B ;$$

и т. д. до гр n .

$$\Sigma | \text{гр } n \times \text{гр } n |_1^m T_A \rightarrow n.n.ST_B ;$$

$$\Sigma | \text{гр } n \times \text{гр } (n+1) |_1^m T_A \rightarrow n.n+1 T_B ;$$

$$\Sigma | \text{гр } n \times \text{гр } S |_1^m T_A \rightarrow n.ST_B ;$$

$$\text{проверка } \Sigma | \text{стр } n |_1^{n+q} T_B = n.ST_B ;$$

примечание симметричное заполнение таблицы B производится в пределах бланка (до строки n).

III. 17. Рассмотрим теперь отрезок программы, описывающей действия над частями таблиц A и B , заштрихованными на схеме 8:

для $i := 2, 3, 4$ выполнить

для $k := 6, 7, 8, 9$ выполнить

$$\Sigma C | \text{гр } i T_A |_2^5 \times | \text{гр } k T_B |_3^6 \rightarrow i-1.k-5 T_C ;$$

Схема 8

Таблица А

	1	2	3	4	5	...
1						
2						
3						
4						
5						
6						
7						
...						

Таблица В

	...	6	7	8	9	...
1						
2						
3						
4						
5						
6						
7						
...						

Таблица С

	1	2	3	4
1				
2				
3				

В результате расчета заполняется таблица C .

III. 18. Продолжим сопоставление с матричной записью, начатое в III. 13. Формула из III. 14 очень проста, но для выполнения расчета вычислителю нужно будет устно объяснить, что она собой представляет и как ее реализовать, ибо в самом обозначении это не отображено.

Более того, если формально следовать формуле, то потребуется: транспонировать матрицу a ; совершать неудобные операции умножения строк на графы, т. е. произвести ряд ненужных действий. Помимо этого, в выражении не содержится никаких указаний, как выполнять проверки.

Эти трудности еще более возрастают для примера из III. 17., в котором надлежит выполнить операции над произвольными частями таблиц. В матричных обозначениях нет средств для непосредственного описания таких действий, поэтому для выполнения их следует произвести ряд дополнительных преобразований. Это и понятно, так как в линейной алгебре исследуются общие закономерности, а не частные случаи. В вычислительной же практике подобные вычисления могут быть, хотя обычно части массивов более упорядочены, например, занимают все поле таблицы по вертикали и часть поля по горизонтали.

Матричные обозначения можно применить как сокращенную информацию для составителя программы, который должен поставить им в соответствии тот или иной блок программы на алгоритмическом языке, пользуясь при этом каталогом имеющейся библиотеки блоков.

Компактные алгоритмы решения систем линейных уравнений методом исключения

III. 19. Одной из наиболее распространенных и важных процедур в алгоритмах расчетов сооружений является решение систем линейных уравнений метода сил или метода деформаций (см. систему (1) из II. 1).

О сути метода исключения было сказано в II. 6. Формулы (1) — (5) из II. 8 описывают прямой ход решения системы уравнений — исключение неизвестных; после подстановки формул (4) и (5) в (2) и (3), получим:

$$k_{bc} = - \frac{\delta'_{bc}}{\delta'_{bb}} ; \quad (1)$$

$$\delta'_{bc} = \delta_{bc} + \Sigma k_{ab} \delta'_{ac} ; \quad (2)$$

$$\delta'_{bb} = \delta_{bb} + \Sigma k_{ab} \delta'_{ab} . \quad (3)$$

Так как в данном случае рассматривается полное заполнение матрицы, то суммирование в формулах (2), (3) относится ко

всем предыдущим номерам неизвестных

$$a=b-1, b-2, b-3, \dots, 2, 1.$$

δ_{bb}, δ_{bc} — главные и побочные коэффициенты исходной симметричной матрицы системы уравнений;

$\delta'_{bb}, \delta'_{bc}$ — соответствующие коэффициенты преобразованной треугольной матрицы.

Как видно из формул (2) и (3), коэффициенты δ' получаются прибавлением к исходным коэффициентам δ сумм произведений, которые можно вычислять на клавишной машине автоматически. Это дает возможность разработать компактную схему решения систем уравнений, при которой значительно сокращается количество записей, уменьшаются размеры расчетных таблиц n , в результате, уменьшается необходимое для решения время.

В обычной, некомпактной схеме Гаусса все промежуточные произведения записываются. Вспомним, как это делается: сначала коэффициенты первого уравнения делятся на коэффициент при первом неизвестном и результаты с обратным знаком записываются в таблицу; полученная строка поочередно умножается на коэффициенты при первом неизвестном в 2, 3 ..., n -ом уравнениях и результаты записываются в предусмотренные строки таблицы; затем производится сложение соответствующих строк и исключение первого неизвестного; получаем преобразованную систему с $n-1$ неизвестными. Далее таким же образом исключается 2, 3, ..., $n-1$ неизвестное. На каждом шаге к предыдущим значениям коэффициентов при неизвестных прибавляется дополнительный член (один из членов суммы в формулах (2) и (3)).

В компактном алгоритме метода исключения выполняются те же действия, но иначе: вместо последовательного вычисления и записи каждого члена суммы, на автомате сразу вычисляется сумма произведений и записывается одно число.

Существуют различные компактные схемы, отличающиеся друг от друга формой таблиц и характером записи чисел. Например, если поместить коэффициенты δ_{ik} в строки $i=1, 2, \dots, n$ табл. A , k — записывать в строки $i'=1', 2', \dots, n'$ табл. A (см. схему 9), δ'_{ik} — в строки i' вспомогательной табл. B , у которой в клетках 1.1, 2.2, ..., $S_1.S_1$ записаны единицы, то получим следующие вычислительные формулы на алгоритмическом языке: для вычисления δ'_{ik} —

$$\Sigma | \text{гр } i \text{ Т}_A \times \text{гр } k \text{ Т}_B | \rightarrow i, k \text{ Т}_B;$$

для вычисления коэффициентов k —

$$- | \text{стр } i' T_B | / i' . i T_B \rightarrow \text{стр } i' T_A .$$

Такое размещение чисел в таблицах и основанная на нем компактная схема предложены в институте механики АН УССР Л. С. Ямпольским и И. О. Губерман [30]. Процесс вычислений состоит в определении сумм попарных произведений чисел в графах основной и вспомогательной таблиц A и B и делении получаемых преобразованных уравнений на главные коэффициенты. Величины неизвестных определяются обратной подстановкой и записываются в табл. B .

В работе [30] указанный алгоритм описан словесно. Запись того же алгоритма на алгоритмическом языке короче и проще и, что главное, не требует от вычислителя изучения и запоминания (запоминание если и происходит, то естественно, как следствие многократного повторения).

III. 20. Процедура (1). Решение симметричных систем уравнений по компактной схеме метода Гаусса:

начало примечание бланки табл. A , B , V даны на схеме 9, n — порядок системы, q — число граф свободных членов. В строках $1, 2, \dots, n$ табл. A записаны коэффициенты и свободные члены исходной системы уравнений, в графе S_1 табл. A перед началом расчета выполняется контрольное суммирование. Клетки таблиц в различных бланках (A , B , V) должны иметь строго одинаковые размеры (стандартная сетка);

Симметричное заполнение нижней части табл. A :
для $i := 1, 2, \dots, n$ выполнить

$$| \text{стр } i |_{n+1}^{S_1} T_A \rightarrow | \text{гр } i |_{n+1}^{S_1} T_A ;$$

Дополнительное контрольное суммирование:
для $i := 1, 2, \dots, n$ выполнить

$$\Sigma | \text{гр } i |_1^n T_A \rightarrow S . i T_A ;$$

проверка $| \text{стр } S + \text{стр } n+1 + \dots + \text{стр } n+q |_1^n = | \text{стр } S_1 |_1^n ;$

примечание если проверка не удовлетворяется, то это значит, что контрольные суммы были вычислены неправильно;

Прямой ход:

для $i := 1, 2, \dots, n$ выполнить

{ для $k := i, i+1, \dots, S_1$ выполнить

$$\Sigma | \text{гр } i T_A \times \text{гр } k T_B |_1^{S_1} \rightarrow i' . k T_B ;$$

проверка $\Sigma | \text{стр } i' T_B |_1^{n+q} = i' . S_1 T_B ;$

$$- | \text{стр } i' T_B |_1^{S_1} / i' . i T_B \rightarrow | \text{стр } i' T_A |_1^{S_1} ;$$

проверка $\sum |\text{стр } i' T_A|_1^{n+q} = i' S_1 T_A$;

Обратный ход:

для $i := 1, 2, \dots, q$ выполнить

[для $k := n, n-1, \dots, 1$ выполнить

$$\Sigma | \text{стр } iT_B \times \text{стр } k'T_A |_1^n + k'.n + iT_A \rightarrow i.kT_B;$$

проверка $\sum \text{стр } iT_B \times \text{стр } ST_A |1^n + S.n + iT_A = 0|$

конец процедуры.

Начальное содержание строки 2 табл. В равно нулю и первый член формулы обратного хода обращается в нуль, затем строка табл. В постепенно заполняется числами.

III. 21. Для решения систем уравнений при большом числе граф свободных членов целесообразно использовать числа влияния. Числом влияния β_{ik} принято называть значение неизвестного x_i при условии, что свободный член для k -го уравнения равен единице ($\delta_{kp} = 1$), а остальные свободные члены равны нулю. Так, например, от свободного члена $\delta_{3p} = 1$ получим числа влияния $x_1 = \beta_{13}$, $x_2 = \beta_{23}$, $x_3 = \beta_{33}$, $x_4 = \beta_{43}$, ...

Система уравнений в матричных обозначениях записывается так:

$$A\mathbf{x} + \mathbf{q} = 0, \quad (1)$$

где A — матрица коэффициентов; q — вектор свободных членов.

Матрица, составленная из чисел влияния, называется в линейной алгебре обратной матрицей и обозначается A^{-1} *.

Для определения чисел влияния следует решить систему уравнений по компактной схеме, порознь для единичных свободных членов. Для симметричной системы уравнений обратная матрица также получается симметричной и это может быть учтено в программе: вычисления достаточно провести только для половины обратной матрицы, расположенной ниже главной диагонали, остальные коэффициенты определяются по симметрии.

Умножая действительные значения свободных членов на соответствующие числа влияния, нетрудно получить величины неизвестных от произвольной комбинации свободных членов.

$$\begin{aligned} x_1 &= \beta_{11}\delta_{1p} + \beta_{12}\delta_{2p} + \dots + \beta_{1n}\delta_{np}; \\ x_2 &= \beta_{21}\delta_{1p} + \beta_{22}\delta_{2p} + \dots + \beta_{2n}\delta_{np}; \\ &\vdots \\ x_n &= \beta_{n1}\delta_{1p} + \beta_{n2}\delta_{2p} + \dots + \beta_{nn}\delta_{np}. \end{aligned} \quad (2)$$

* $AA^{-1}=E$, где E — единичная матрица; отсюда название — обратная матрица.

Если свободные члены находятся в левой части уравнений, то числа влияния отрицательны.

В матричных обозначениях выражение (2) имеет вид:

$$x = A^{-1} q, \quad (3)$$

где q — вектор свободных членов; x — вектор неизвестных.

На алгоритмическом языке формула (3) записывается так:

для $i := 1, 2, \dots, q$ выполнить

для $k := 1, 2, \dots, n$ выполнить

$$\Sigma | \text{гр } iT_p \times \text{гр } kT_{B_i} | \rightarrow i.kT_B.$$

Схема 9

Таблица А

	1	2	...	n	n+1	n+2	...	n+q	S_i
1									
1'									
2									
2'									
...									
...									
n									
n'									
n+1									
n+2									
n+3									
n+4									
...									
n+q									
S_1									
S									

Таблица Б

	1	2	...	n	$n+1$	$n+2$	...	$n+q$	S_1
1	1								
1'									
2		1							
2'									
...			1						
...									
n				1					
n'									
$n+1$					1				
$n+2$						1			
...							1		
$n+q$								1	
S_1									

Таблица В

	1	2	...	n
1				
2				
...				
q				

В графах 1, 2, ..., q табл. P помещаются свободные члены, в графах 1, 2, ..., n табл. B_1 — числа влияния (обратная матрица A^{-1}), в строках 1, 2, ..., q табл. B записываются величины неизвестных соответственно от 1, 2, ..., q граф свободных членов (см. бланки на схеме 10).

Таблица А

	1	2	...	n	S
1					
1'					
2					
2'					
...					
...					
n					
n'					
S					

Таблица Б

	1	2	...	n	S
1	1				
1'					
2		1			
2'					
...			1		
...					
n				1	
n'					
S					1

Таблица Р

	1	2	...	q
1				
2				
...				
n				

Таблица B_1

	1	2	...	n	S
1					
2					
...					
n					

Таблица В

	1	2	...	n	S
1					
2					
...					
q					

В программу нужно, кроме того, включить проверочные равенства.

III. 22. Процедура (2). Решение симметричных систем уравнений по компактной схеме метода Гаусса с применением обратной матрицы;

начало примечание бланки табл. A, B, P, B_1, B даны на схеме 10. n — порядок системы, q — число граф свободных членов. В табл. A записаны коэффициенты исходной системы уравнений, в строке и графе S вычисляются контрольные суммы этих коэффициентов. В $1, 2, \dots, q$ графах табл. P даны свободные члены.

В бланках табл. A и B графы $n+1, n+2, \dots, n+q$ отсутствуют (см. схему 9). Если эти графы были получены в табл. A на предшествующем этапе расчета, то их следует перенести в табл. P .

В табл. B_1 получаем обратную матрицу, в строках табл. B — значения неизвестных от $1, 2, \dots, q$ граф свободных членов;

Прямой ход:

для $i := 1, 2, \dots, n$ выполнить

{ для $k := i, i+1, \dots, S$ выполнить

$$\Sigma | \text{гр } iT_A \times \text{гр } kT_B |_i^S \rightarrow i'_k T_B;$$

проверка $\Sigma | \text{стр } i'T_B |_1^n = i'.ST_B;$

$$- | \text{стр } i'_0 T_B |_1^S / i'_0 iT_B \rightarrow | \text{стр } i'_0 T_A |_1^S;$$

проверка $\Sigma | \text{стр } i'_0 T_A |_1^n = i'_0.ST_A;$

примечание прямой ход выполняется только для матрицы A (без свободных членов), в этом отличие от предыдущей процедуры;

Обратный ход для вычисления обратной матрицы:

для $i := n, n-1, \dots, 1$ выполнить

{ для $k := i, i-1, \dots, 1$ выполнить

{ если $k=i$ то $-m/i' iT_B \rightarrow a$ иначе $0 \rightarrow a$

$$\Sigma | \text{стр } iT_{B_1} \times \text{стр } k'T_A |_1^n + a \rightarrow i.kT_{B_1} \rightarrow k.iT_{B_1};$$

проверка $\Sigma | \text{стр } iT_{B_1} \times \text{стр } ST_A |_1^n + m = 0;$

примечание текущие значения a записываются на отдельном листе бумаги, предыдущие значения зачеркиваются.

m — масштабный множитель — принимается таким, чтобы порядок чисел в обратной матрице был удобен для вычислений и для записи ($m=1, 10, 100, \dots$);

$$\Sigma | \text{стр } iT_{B_1} |_1^n \rightarrow i.ST_{B_1};$$

Определение величин неизвестных для q граф свободных членов по обратной матрице:

для $i := 1, 2, \dots, q$ выполнить
 для $k := 1, 2, \dots, n, S$ выполнить

$$\Sigma | \text{гр } iT_P \times \text{гр } kT_B |_1^n \rightarrow i.kT_B;$$

проверка $\Sigma | \text{стр } iT_B |_1^n = i.ST_B$

конец процедуры.

III. 23. В правильно поставленных задачах строительной механики системы уравнений должны получаться симметричными — это следует из принципа взаимности работ. Однако при решении задач теории упругости численными методами могут быть отклонения. Например, при решении методом сеток для уточнения расчета нередко принимают на краях области более густую сетку, чем для внутренней зоны [10], что приводит к нарушению симметрии в системе уравнений. Поэтому в библиотеку программ вошло решение не только симметричных, но и несимметричных систем уравнений. В последнем случае количество вычислений существенно увеличивается.

Схема 11

Таблица А

	1	2	...	n	n+1	n+2	...	n+q	S
1									
1'									
2									
2'									
...									
...									
n									
n'									
S									

Таблица Б

	1	2	...	n	S
1	1				
1'					
2		1			
2'					
...			1		
...					
n				1	
n'					
S					1

Таблица В

	2	2	...	n
1				
2				
...				
n				

Процедура (3). Решение несимметричных систем уравнений по компактной схеме метода Гаусса;
начало примечание n — порядок системы, q — число граф свободных членов. Бланки табл. A , B , V даны на схеме 11. Перед началом расчета выполняются стандартные вычисления контрольных сумм исходных данных в строке и графе S табл. A ;

Прямой ход: для $i := 1, 2, \dots, n$ выполнить
 { для $k := i, i+1, \dots, n, S$ выполнить

$$\Sigma | \text{гр } iT_A \times \text{гр } kT_B |_1^S \rightarrow i'.kT_B;$$

проверка $\Sigma | \text{стр } i' T_B |_1^n = i'.ST_B;$

для $k := i, i+1, \dots, n, n+1, \dots, n+q, S$ выполнить

$$- \Sigma | \text{гр } iT_B \times \text{гр } kT_A |_1^S / i'.iT_B \rightarrow i'.kT_A;$$

проверка $\Sigma | \text{стр } i' T_A |_1^{n+q} = i'.ST_A \}$

Обратный ход:

для $i := 1, 2, \dots, q$ выполнить

{ для $k := n, n-1, \dots, 1$ выполнить

$$\Sigma | \text{стр } iT_B \times \text{стр } k' T_A |_1^n + k'.n + iT_A \rightarrow i.kT_B;$$

проверка $\Sigma | \text{стр } iT_B \times \text{стр } ST_A |_1^n + S.n + iT_A = 0 \}$

конец процедуры.

Для получения по данной программе обратной матрицы следует принять n граф свободных членов (т. е. принять $q=n$) с единицами в клетках $1.n+1, 2.n+2, \dots, n.n+q$ и нулями в остальных клетках, тогда в табл. B_1 (вместо табл. B) получим обратную матрицу. Для удобства вместо единицы можно поместить произвольный масштабный множитель m , при этом обратная матрица получится умноженной на m . Составление табл. B по табл. B_1 и P выполняется так же, как в предыдущей процедуре (2).

Компактный алгоритм

для метода последовательных приближений

III.24. В II.1 была описана сущность метода последовательных приближений для решения систем уравнений. В II.4 этот метод был представлен в форме алгоритма перераспределения, удобного для инженера, выполняющего расчет на логарифмической линейке. При выполнении расчета на клавишной машине эта форма уже не является удобной, — нужна компактная схема. Ниже приведена соответствующая процедура на алгоритмическом языке, являющаяся модернизацией алгоритма, разра-

ботанного в работе [100]. В программу включены построчные проверки, аналогичные тем, которые применяются в алгоритме Гаусса. Данная процедура интересна применением оператора цикла типа пересчета: **для... пока... выполнить**.

Вычисления выполняются в расчетных бланках, представленных на схеме 12. В исходной табл. А даны коэффициенты и свободные члены системы уравнений. В табл. Б получается преобразованная система после деления на главные коэффициенты с обратным знаком. Для каждой графы свободных членов составляется отдельная таблица В, итераций ($v=1, 2, \dots, q$). Процесс продолжается до тех пор, пока значения неизвестных практически перестанут изменяться (в пределах заданной точности). Значения неизвестных, полученные в последних строках таблиц $B_1, B_2, \dots, B_q$, переписываются в строки 1, 2, ..., q табл. В, которая является конечной целью расчета как в методе исключения, так и в методе итераций.

III. 25. Процедура (4). Решение систем уравнений по компактной схеме метода последовательных приближений (итерация по Зейделю); **начало примечание** n — порядок системы, q — число граф свободных членов, v — порядковый номер графы свободных членов ($v := 1, 2, \dots, q$), i — порядковый номер итерации, k — порядковый номер строки матрицы.

Процесс прекращается, когда величины неизвестных в $i-1$ и i -том приближениях станут практически равны друг другу (в пределах допускаемой погрешности);

Преобразование матрицы А к виду, удобному для итерации: **для** $k := 1, 2, \dots, n$ **выполнить**

$$\{ - | \text{стр } kT_A |_1^S / k.kT_A \rightarrow | \text{стр } kT_B |_1^S ;$$

проверка $\Sigma | \text{стр } kT_B |_1^{n+q} = k.ST_B ; \}$

Дополнительное контрольное суммирование в табл. Б: **для** $k := 1, 2, \dots, n$ **выполнить**

$$\{ \Sigma | \text{гр } k_n |_1^n \rightarrow S_n.k ; \Sigma | \text{гр } k_n |_1^n \rightarrow S_n.k ; S_n.k + S_n.k \rightarrow S_1.k \} ;$$

примечание в соответствии с обозначениями III. 3: $\text{гр } k_n T_B$ — часть графы k , расположенная слева от главной диагонали табл. Б, $\text{гр } k_n T_B$ — часть графы k , расположенная справа от главной диагонали табл. Б. Для заполнения строки S табл. Б действует обычное правило: если имеется пустая строка (графа) S , то ее следует заполнить, суммируя предыдущие числа в графах (строках). Такое правило избавляет от необходимости писать каждый раз операторы контрольного суммирования, их приводят только в случаях других обозначений ($S_1, S_n, S_n, \dots$);

Схема 12

Таблица А										Таблица В				
	1	2	...	n	n+1	n+2	...	n+q	S		1	2	...	n
1										1				
2										2				
...										...				
n										q				
S														

Таблица Б										Таблица В _v				
	1	2	...	n	n+1	n+2	...	n+q	S		1	2	...	n
1				n						1				
2										2				
...										...				
n	л									i				
S														
S _л														
S _п														
S ₁														

Проверка контрольных сумм: стр $S_1 = \text{стр } S - D$;

Процесс итерации:

для $v := 1, 2, \dots, q$ выполнить начало $i := 0$;

для $i := i + 1$ пока стр $i T_{B_v} \neq \text{стр } i - 1 T_{B_v}$, выполнить

{ для $k := 1, 2, \dots, n, S$ выполнить

$$\Sigma | \text{стр } i T_{B_v} \times \text{стр } k_n T_B |_1^n + \Sigma | \text{стр } i - 1 T_{B_v} \times \text{стр } k_n T_B |_1^n + k.n + v T_B \rightarrow i.k T_{B_v};$$

проверка $\Sigma | \text{стр } i T_{B_v} |_1^n = i.S T_{B_v}$;

стр $i T_{B_v} \rightarrow \text{стр } v T_B$ конец v

конец процедуры.

Если строки, указанные в программе, в бланках отсутствуют, то соответствующие члены в операторах обращаются в нуль.

Так, например, при $k=1$ отсутствует $| \text{стр } 1, |_1^n T_B$, при $k=n$ — $| \text{стр } k, |_1^n T_B$, при $i=1$ отсутствует $\text{стр } i-1 T_B$.

В последних строках i таблиц B — получаем искомые величины неизвестных для $v=1, 2, \dots, q$.

В данной процедуре применен сложный оператор цикла с тремя параметрами v, i и k , подстановка которых в уме требует некоторого навыка. Для удобства рекомендуется записывать текущие значения индексов и индексных выражений во вспомогательной табличке.

Например, в табличке

$n+v$	i	$i-1$	k
11	2	1	1, 2, 3, 4, 5, 6, 7
	3	2	1, 2, 3, 4

последние значения индексов будут $n+v=11, i=3, i-1=2, k=4$.

Для менее подготовленных операторов можно дать развернутую, более длинную, но простую запись (III. 16).

В приведенном алгоритме итерация выполняется в приближениях (а не в поправках, как в работе [100]). Поэтому процесс последовательных приближений обладает свойством самоисправляемости.

Однако в алгоритм все же введены построчные проверки, чтобы исключить возможные ошибки и связанные с ними дополнительные вычисления, необходимые для рассасывания (релаксации) ошибок.

III. 26. Интересно сопоставить метод исключения и метод последовательных приближений. Метод Гаусса (метод исключения по схеме единственного деления) является наиболее общим методом решения уравнений строительной механики, пригодным для большинства систем, встречающихся в практике расчета сооружений*. Для симметричных систем уравнений ленточной структуры вычисления выполняются над числами, расположенными выше главной диагонали и только в пределах ленты.

Метод последовательных приближений (вариант Зейделя) имеет существенные ограничения. Хотя теоретически для симмет-

* Плохо обусловленные матрицы при правильной постановке задачи не должны иметь места.

ричных матриц положительно определенной формы, характерных для строительной механики, процесс обладает неизбежной сходимостью, но его далеко не всегда можно реализовать на практике. Если главные коэффициенты меньше по модулю суммы побочных коэффициентов, то процесс часто сходится настолько медленно, что его трудно осуществить не только на клавишных автоматах, но и на быстродействующих электронных вычислительных машинах. Правда, в ряде случаев возможно применить то или иное преобразование исходных уравнений или определенные приемы ускорения сходимости. Однако эти преобразования и приемы не носят общего характера, они сугубо локальны и применимы лишь в определенных частных случаях.

Другие циклические методы (релаксация, скорейший спуск, сопряженные градиенты и др.) также капризны и в большинстве случаев приводят к значительно большему количеству вычислений, чем по методу Гаусса. При решении систем уравнений на клавишных автоматах метод последовательных приближений можно применять лишь в тех случаях, когда процесс сходится достаточно быстро, скажем за 3—5 циклов. Однако, даже в этих наиболее благоприятных случаях преимущества итерационного метода не всегда очевидны. В методе Гаусса наиболее трудоемкая часть расчета (прямой ход) выполняется сразу для матрицы и произвольного числа свободных членов. Для каждой графы свободных членов дополнительные вычисления невелики. Для метода итерации инвариантная (не зависящая от свободных членов) часть расчета незначительна и почти весь расчет нужно выполнять заново для каждой новой графы свободных членов (в III. 25. составлялись q таблиц итераций).

Поэтому при большом числе правых частей (граф свободных членов) или при отыскании обратной матрицы не слишком высокого порядка метод Гаусса может оказаться более удобным. Особенно это относится к ленточным матрицам с относительно узкой лентой, для которых решение может быть получено вручную на клавишной машине за несколько часов даже при большом числе неизвестных (20—30).

Только при очень большом числе неизвестных (40—60) и быстро сходящемся процессе преимущества метода итерации для счета на клавишных машинах очевидны.

КОМПАКТНЫЕ АЛГОРИТМЫ СТРОИТЕЛЬНОЙ МЕХАНИКИ

IV. 1. В главе изложены компактные алгоритмы и построенные на их основе инженерные программы для двух основных методов строительной механики — сил и деформаций; даны рекомендации по привязке типовых программ к расчету тех или иных конкретных сооружений, подготовке табличных бланков и исходных данных.

Далее приводится программа определения частот и форм свободных колебаний произвольной системы, приемы автоматизации составления систем уравнений метода сеток, краткие сведения о ряде других программ.

Рассмотренные программы характеризуют подход к построению алгоритмов. По этому образцу могут быть составлены и изложены на алгоритмическом языке компактные схемы алгоритмов для разнообразных задач строительной механики и прикладной теории упругости.

Построение универсальной программы метода сил

IV. 2. Предшествующий расчет статически определимой системы рассматриваться не будет. Предлагается, что он выполняется по одной из программ для того или иного частного вида конструкций * и результаты записываются в стандартную матрицу усилий.

Расчет статически неопределимых систем методом сил состоит из трех этапов: составления системы уравнений, решения системы уравнений, определения искомых усилий в стержнях статически неопределимой системы.

* Составить общую программу для этой стадии трудно, ввиду большого разнообразия возможных основных систем, составление же частной программы для конкретного типа основной системы обычно не представляет затруднений и может быть выполнено оптимальным образом.

Программы решения систем уравнений были рассмотрены выше. Остается составить процедуры для двух других этапов.

IV.3. Формулу Мора для определения перемещений в пространственной стержневой системе запишем в следующем обобщенном виде:

$$\delta_{ik} = \sum \int \frac{S_i S_k dl}{c}, \quad (1)$$

где S_i, S_k — обобщенные усилия в статически определимой системе могут быть изгибающими и крутящими моментами, продольными и поперечными силами;

i — индекс усилий от единичных неизвестных;

k — индекс усилий от единичных неизвестных и от нагрузки;

c — характеристика жесткости, которой могут быть величины EI (для изгибающих моментов), EF (для продольных сил), $\frac{GF}{\mu}$ (для поперечных сил), GT (для крутящих моментов);

E — модуль упругости материала, G — модуль упругости второго рода;

I, F — момент инерции и площадь поперечного сечения;

μ — коэффициент, зависящий от формы поперечного сечения;

dl — дифференциал длины участка интегрирования.

Пределы интегрирования в формуле (1) не указаны — для каждого участка они ограничены его длиной l (от 0 до l). Суммирование распространяется на все участки системы.

Развернутый вид формулы Мора следующий ([78], стр. 371):

$$\begin{aligned} \delta_{ik} = & \sum \int \frac{M_i^y M_k^y dl}{EI_y} + \sum \int \frac{M_i^z M_k^z dl}{EI_z} + \sum \int \frac{M_i^x M_k^x dl}{GT_k} + \\ & + \sum \int \frac{N_i N_k dl}{EF} + \sum \int \frac{\mu_y Q_i^y Q_k^y dl}{GF} + \sum \int \frac{\mu_z Q_i^z Q_k^z dl}{GF}. \end{aligned} \quad (2)$$

Величина T имеет следующие приближенные значения. Для квадрата со стороной a : $T = 0,1426 a^4$; для узкого прямоугольника со сторонами a, b , где $a > b$: $T = \frac{b^3}{3} (a - 0,63 b)$; для кругового сечения: $T = \frac{\pi r^4}{2}$.

Для сечений, состоящих из узких прямоугольников, например, для тавровых и двутавровых, приближенно можно принять: $T = \frac{1}{3} \Sigma d^3 l$, где d — длина коротких сторон, l — длина больших сторон прямоугольника. Отношение E/G можно приближенно принять для металлических конструкций 2,57, для железобетонных 2,33. В дальнейшем будем пользоваться более краткой и удобной для машинного счета формулой (1).

Криволинейные эпюры будем аппроксимировать ломаной линией*. Таким образом, предполагается, что каждый стержень системы делится на определенное количество участков так, что в пределах участка все интегрируемые эпюры ограничены прямыми линиями.

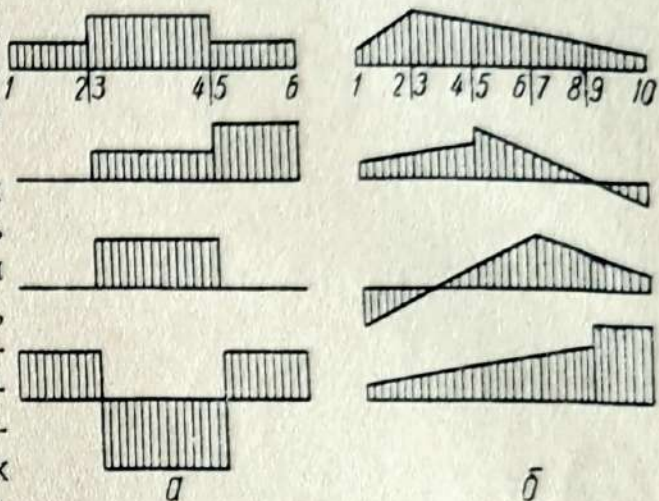


Рис. 21.

Эпюры усилий в статически определимой системе разобьем на две группы: в первой группе все эпюры усилий ограничены линиями, параллельными продольной оси стержня, во второй группе — наклонными линиями. Переломы или скачки в пределах участка не разрешаются. Первая группа является более простым, частным случаем второй группы.

Для каждой группы принимается свое разделение на участки. Начало и конец каждого участка нумеруются. Например, на рис. 21, а представлена первая группа эпюр с нумерацией участков 1—2, 3—4, 5—6; на рис. 21, б — вторая группа эпюр с нумерацией участков 1—2, 3—4, 5—6, 7—8, 9—10. Более редко раз-

* Обычно достаточно разбить криволинейную эпюру на стержне на 4—5 участков. При более точной аппроксимации количество участков можно уменьшить, но это приведет к неравноценному усложнению формул, что очень нежелательно при механизированном расчете, для которого простота и стандартность имеют первостепенное значение. Для многих сооружений местную равномерно распределенную нагрузку на стержне в проектной практике принято приводить к узловой (ветровая нагрузка) или сосредоточенной. Например, для построения линий влияния усилий производится расчет на единичные грузы, прилагаемые через определенные интервалы, а усилия от равномерно распределенной нагрузки определяются как площади соответствующих линий влияния.

бить участки нельзя, так как при этом появились бы переломы или скачки эпюры в пределах отдельных участков.

Интегрирование двух эпюр первой группы в пределах одного участка выполняется по формуле

$$\int \frac{S_i S_k dl}{c} = \frac{i_1 \times k_1 \times l}{c}, \quad (3)$$

где i_1 и k_1 — ординаты эпюр усилий на участке в i -том и k -том состояниях основной системы (рис. 22, а); i — длина участка; c — характеристика жесткости участка.

Для совокупности участков:

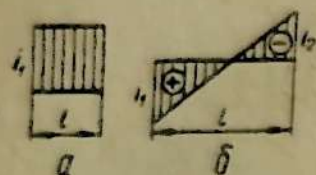


Рис. 22.

где v — номер участка; m_1 — число участков.

Расположим данные о первой группе участков (величины i_{1v} , k_{1v} , $\alpha_v = \frac{i_v}{c_v}$) в графах i , k , α табл. 1 на схеме 13. Тогда величины $i_{1v} \times \alpha_v$ можно вычислить по формуле для $v := 1, 2, \dots, m_1$ выполнить

$$|\text{стр } v|_1^n T_1 \times v. \alpha T_1 \rightarrow |\text{стр } v|_1^n T_3,$$

где m_1 — количество строк в табл. 1 и 3, равное числу участков. В табл. 3 зачисляются величины $i_{1v} \times \alpha_v$.

Теперь нетрудно получить формулу для δ_{ik} :

$$\Sigma |\text{гр } iT_3 \times \text{гр } kT_1|_1^m \rightarrow i.kT_A. \quad (5)$$

Содержание формул (4) и (5) одинаковое, в табл. А (см. схему 9) зачисляются величины δ_{ik} — коэффициенты и свободные члены системы уравнений. Индекс i изменяется от 1 до n , индекс k — от 1 до $n+q$. В графах от 1 до n табл. А получаем симметричную матрицу коэффициентов, в графах от $n+1$ до $n+q$ — свободные члены для 1, 2, ..., q комбинаций нагружения. Выполняемое преобразование аналогично трансформации Гаусса (III. 14). Интегрирование эпюр второй группы в пределах одного участка выполняется по более сложной формуле

$$\int \frac{S_i S_k dl}{c} = \frac{l}{6 \times c} [2 \times (i_1 \times k_1 + i_2 \times k_2) + (i_1 \times k_2 + i_2 \times k_1)], \quad (6)$$

где i_1, i_2, k_1, k_2 — ординаты эпюр усилий на участке в i -том и k -том состояниях основной системы* (рис. 22, б). Первый член формулы представляет собой удвоенную сумму произведений соответствующих ординат, второй — сумму произведений ординат по диагонали.

Выражения (6) следует просуммировать для всех участков:

$$\delta_{ik} = \sum_{v=1}^{m_2} \frac{l_v}{6 \times c_v} [2 \times (i_{1v} \times k_{1v} + i_{2v} \times k_{2v}) + (i_{1v} \times k_{2v} + i_{2v} \times k_{1v})], \quad (7)$$

где v — номер участка, m_2 — число участков.

Расположим данные о второй группе участков (величины $i_{1v}, i_{2v}, k_{1v}, k_{2v}, \alpha_v = \frac{i_v}{6 \times c_v}$) в графах i, k, α табл. 2 на схеме 13. Тогда

величины $i_{1v} \times \alpha_v, i_{2v} \times \alpha_v$ можно вычислить по формуле для $v = 1, 2, \dots, 2 \times m_2$ выполнить

$$|\text{стр } v|_1^{m_2} T_2 \times v. \alpha T_2 \rightarrow |\text{стр } v|_1^{2 \times m_2} T_1,$$

где $2 \times m_2$ — количество строк в табл. 2 и 4, равное удвоенному числу участков. В табл. 4 засылаются величины $i_{1v} \times \alpha_v$ и $i_{2v} \times \alpha_v$.

Теперь можно получить вычислительную формулу для δ_{ik} —

$$2 \times \Sigma |\text{гр } iT_4 \times \text{гр } kT_2|_1^{2 \times m_2} + \Sigma |\text{гр } iT_4 + \text{гр } kT_2|_{\text{диаг}}^{2 \times m_2} \rightarrow i.kT_A. \quad (8)$$

В табл. А засылаются δ_{ik} — коэффициенты и свободные члены системы уравнений.

Если имеются обе группы эпюр усилий, то полная формула для вычисления δ_{ik} будет:

$$\Sigma |\text{гр } iT_3 \times \text{гр } kT_1|_1^{m_1} + 2 \times \Sigma |\text{гр } iT_1 \times \text{гр } kT_2|_1^{2 \times m_2} + \Sigma |\text{гр } iT_4 \times \text{гр } kT_2|_{\text{диаг}}^{2 \times m_2} \rightarrow i.kT_A. \quad (9)$$

Эта формула является основным вычислительным оператором в процедуре составления системы уравнений. Остальные операторы относятся к контрольному суммированию и проверкам.

IV. 4. Процедура составления системы уравнений метода сил;
начало примечание бланки табл. 1, 2, 3, 4 даны на схеме 13.

* Поэтому и имеет смысл выделение первой группы. К первой группе обычно относятся эпюры нормальных сил, поперечных сил и крутящих моментов, ко второй группе — эпюры изгибающих моментов.

Схема 13

Таблица 1

	1	2	...	n	n+1	...	n+q	S	S ₁	α
1										
2										
...										
m ₁										

Таблица 2

	1	2	...	n	n+1	...	n+q	S	S ₁	α
1										
2										
3										
4										
...										
2×m ₂										

Таблица 3

	1	2	...	n	S ₁
1					
2					
...					
m ₂					

Таблица 4

	1	2	...	n	S ₁
1					
2					
3					
4					
...					
2×m ₂					

В бланках 1, 2 помещаются ординаты эпюр усилий первой и второй групп в статически определимой основной системе. В графе α табл. 1 даны величины $\frac{l}{c}$, в графе α табл. 2 — величины $\frac{l}{6 \times c}$.

В табл. А (см. схему 9) получается матрица системы уравнений метода сил.

m₁ — количество сечений для участков первой группы (по одному сечению на каждый участок), 2×m₂ — количество сечений для участков второй группы (по два сечения на каждый участок в начале и конце), n — порядок системы, q — число граф свободных членов.

В графе S табл. 1 записаны контрольные суммы предшествующих чисел в строках;

Дополнительное контрольное суммирование:

для $i := 1, 2, \dots, m_1$ выполнить

$$\Sigma | \text{стр } i_1^n \rightarrow i.S_1 T_1;$$

заполнение табл. 3 и 4;

для $i := 1, 2, \dots, m_1$ выполнить

$$\{ | \text{стр } i_1^{n, S_1} T_1 \times i.x T_1 \rightarrow \text{стр } i_1^{n, S_1} T_3;$$

проверка $\Sigma | \text{стр } i_1^n T_3 = i.S_1 T_3$;

для $i := 1, 2, \dots, m_2$ выполнить

$$\{ | \text{стр } i_1^{n, S_1} T_2 \times i.x T_2 \rightarrow | \text{стр } i_1^{n, S_1} T_4;$$

проверка $\Sigma | \text{стр } i_1^n T_4 = i.S_1 T_4$;

примечание запись n, S_1 в верхнем пределе означает, что вычисление нужно выполнить для отрезка строки табл. 1 от 1 до n и для контрольной суммы S_1 (которая расположена не сразу за этим отрезком, а дальше);

для $i := 1, 2, \dots, n$ выполнить

{ для $k := i, i+1, \dots, n+q, S$ выполнить

$$\{ \Sigma | \text{гр } iT_3 \times \text{гр } kT_1 |_1^{m_1} + 2 \times \Sigma | \text{гр } iT_4 \times \text{гр } kT_2 |_1^{2 \times m_2} + \\ + \Sigma | \text{гр } iT_4 \times \text{гр } kT_2 |_{\text{диаг}}^{2 \times m_2} \rightarrow i.kT_A;$$

если $i < n$ то $i.kT_A \rightarrow k.iT_A$ };

проверка $\Sigma | \text{стр } iT_A |_1^{n+q} = i.ST_A$ }

конец процедуры.

В процедуре использована симметрия матрицы A относительно главной диагонали — коэффициенты δ_{ki} записываются симметрично коэффициентам δ_{ik} .

IV. 5. Получены две процедуры: составления и решения системы уравнений метода сил. Остается еще составить процедуру определения искомых усилий в статически неопределимой системе по следующей формуле

$$S_i = S_i^0 + \sum_{k=1}^n x_k S_{ik}, \quad (1)$$

где S_i^0 — усилие в i -том сечении статически определимой системы от нагрузки;

S_{ik} — усилия в сечении i от неизвестных $x_k = 1$ ($k = 1, 2, \dots, n$).

Учитывая, что S_{ik} даны в графах 1— n , усилия S_i^0 — в графах $n+t$ ($t = 1, 2, \dots, q$) табл. 1 и 2, x_k — в графах 1— n табл. В (схема 9), получим следующую запись формулы (1) на алгоритмическом языке.

Для группы усилий 1 —

$$\Sigma | \text{стр } tT_B + \text{стр } kT_1 |_1^n + k.n + tT_1 \rightarrow k.tT_5. \quad (2)$$

Для группы усилий 2 —

$$\Sigma | \text{стр } tT_B \times \text{стр } kT_2 |_1^n + k.n + tT_2 \rightarrow k.tT_6; \quad (3)$$

t — порядковый номер загрузки, изменяется от 1 до q .

k — порядковый номер строки в табл. 1 (изменяется от 1 до m_1) и в табл. 2 (изменяется от 1 до $2m_2$).

На основе формул (2) и (3) получена следующая процедура.

Схема 14

Таблица 5

	1	2	...	q
1				
2				
...				
m_1				
S				

Таблица 6

	1	2	...	q
1				
2				
...				
$2m_2$				
S				

IV. 6. Процедура определения усилий в статически неопределимой системе (метод сил);

начало примечание бланки табл. 5 и 6 даны на схеме 14. В графах 1, 2, ..., q этих таблиц получают искомые усилия первой и второй групп для 1, 2, ..., q загрузений.

Бланки табл. 1 и 2 см. схему 13, бланки табл. B — в одной из процедур решения уравнений (гл. III);

для $t := 1, 2, \dots, q$ выполнить

{ для $k := 1, 2, \dots, m_1, S$ выполнить

$$\Sigma | \text{стр } tT_B \times \text{стр } kT_1 |_1^n + k.n + tT_1 \rightarrow k.tT_5;$$

проверка $\Sigma | \text{гр } tT_5 |_1^{m_1} = S.tT_5;$

для $k := 1, 2, \dots, 2m_2, S$ выполнить

$$\Sigma | \text{стр } tT_B \times \text{стр } kT_2 |_1^n + k.n + tT_2 \rightarrow k.tT_6;$$

проверка $\Sigma | \text{гр } tT_6 |_1^{2m_2} = S.tT_6 \}$

конец процедуры.

IV. 7. Теперь можно записать компоновочную программу, состоящую из операторов процедур.

В операторе процедуры указывается ее название (без слова **процедура**) и, когда требуется, сообщаются сведения об изменениях в стандартной процедуре для данного частного случая.

Вычислителю вместе с компоновочной программой передаются участвующие в ней процедуры, оформленные на отдельных листах.

Компоновочная программа расчета статически неопределимых стержневых систем методом сил:

начало примечание программа применима для расчета произвольных плоских и пространственных систем;

составление системы уравнений;

решение системы уравнений;

определение усилий в статически неопределимой системе

конец программы.

В данном случае в качестве процедуры решения систем уравнений могут быть выбраны процедуры (1) и (2), использующие метод Гаусса. Процедуру (2) целесообразно применять при большом числе загрузений. Возможно также применение процедуры (4), использующей метод последовательных приближений, но только при благоприятных соотношениях коэффициентов (главный коэффициент в каждом уравнении должен быть по абсолютному значению значительно больше суммы побочных коэффициентов).

В компоновочную программу, если нужно, включаются другие операторы, например, безусловного или условного перехода.

Рассмотренная программа метода сил носит общий характер. Для отдельных классов сооружений автоматически получаются нужные упрощения. Так, для статически неопределимых ферм, отпадают табл. 2, 4 и 6 и связанные с этими таблицами члены в операторах программы. Для рам отпадают табл. 1, 3 и 5 и соответствующие члены в операторах программы. Только при расчете комбинированных систем фигурируют усилия обеих групп и программа реализуется полностью.

Программа расчета стержневых систем методом сил нашла применение в проектной практике для расчета статически неопределимых ферм, одноэтажных рам промышленных зданий, различных видов комбинированных систем, пролетных строений мостов и др. сооружений (ряд примеров приведен в работе [102]).

При изложении алгоритмов, предназначенных для их реализации инженерами на логарифмической линейке (например, алго-

ритмов перераспределения, рассмотренных в гл. II), необходимо дать подробные словесные указания к примерам расчета, иллюстрирующим вычислительные схемы для тех или иных видов конструкций (в этом, собственно, и состоял алгоритм); при механизированном же выполнении расчетов по стандартным программам уже не нужно сопровождать численный расчет пояснениями, разъясняющими суть выполняемых арифметических действий. Вместо этого достаточно разъяснить инженеру как готовить исходные данные и табличные бланки для той или иной частной задачи, т. е. как производить привязку к типовой программе. Сам же расчет всегда выполняется стандартно, его алгоритм ясно и однозначно описан на алгоритмическом языке.

IV. 8. Приведем ряд рекомендаций по подготовке исходных данных для расчета по универсальной программе метода сил.

Для статически неопределимых ферм:

1. Рассчитывается статически определимая основная система, определяются продольные усилия в стержнях от единичных неизвестных и нагрузок, а также ориентировочно площади сечений. Для выполнения этой работы может быть составлена дополнительная программа *.

2. Система разбивается на участки так, чтобы в пределах отдельного участка сечение и продольное усилие не изменялись. Стержень постоянного по длине сечения при узловой нагрузке имеет один участок. Сечения нумеруются по одному номеру на каждый участок, для удобства номера записываются на схеме фермы (в кружках). Вычисляются относительные величины

$$\frac{l}{c} = \frac{F}{EF}.$$

3. В бланке табл. 1 ** номера строк $1, 2, \dots, m_1$ соответствуют номерам сечений (m_1 — число сечений), номера граф $1, 2, \dots, n$ — номерам неизвестных, номера граф $n+1, n+2, \dots, n+q$ — номерам загружений, $1, 2, \dots, q$.

В табл. 1 записываются продольные усилия в сечениях участков от единичных неизвестных (графы $1-n$) и нагрузки (графы $n+1, \dots, n+q$), а также величины $\frac{l}{EF}$ (графа a).

* Для распространенных случаев такие программы могут быть оформлены как стандартные, т. е. с использованием переменных (буквенных) параметров.

** Рекомендуется размножить на ротапринте или типографским способом бланки с нанесенной на них сеткой таблицы. Удобный размер клетки $0,7 \times 1,8$ см.

Нумеруются графы и строки в бланках остальных таблиц 3, А, Б, В (или В₁, В) и 5 для конкретных значений параметров n , m_1 и q .

Для данной задачи бланки 2, 4, 6 отсутствуют.

На этом подготовка задачи заканчивается. После выполнения расчета вычислителем усилия в стержнях статически неопределимой системы получаем в табл. 5.

Для плоских рам:

1. Рассчитывается статически определимая основная система, определяются ординаты эпюр изгибающих моментов и ориентировочно величины I — моменты инерции сечений.

2. Система разбивается на участки так, чтобы в пределах отдельного участка сечение было постоянным, а эпюра моментов ограничена прямой линией (в общем случае наклонной) и не имела переломов и скачков*. Каждый участок имеет два сечения — в начале и конце. Сечения нумеруются в порядке 1, 2, ..., $2m_2$ (m_2 — число участков), номера записываются на схеме рамы.

Для участков вычисляются относительные величины $\frac{l}{6EI}$.

3. В бланке табл. 2 номера строк 1, 2, ..., $2m_2$ соответствуют номерам сечений, номера граф 1, 2, ..., n — номерам неизвестных, номера граф $n+1$, $n+2$, ..., $n+q$ — номерам нагрузок 1, 2, ..., q .

В табл. 2 записываются ординаты эпюры моментов от единичных неизвестных и нагрузок в сечениях основной системы в начале и конце каждого участка (по две строки для каждого участка), а также величины $\frac{l}{6EI}$ (графа α).

4. Нумеруются графы и строки в остальных табличных бланках 4, А, Б, В (либо В₁ и В), 6 для конкретных значений n , m_2 и q . Для данной задачи бланки 1, 3 и 5 отсутствуют. После выполнения расчета вычислителем изгибающие моменты в сечениях статически неопределимой системы получаем в табл. 6.

Например, на рис. 23 представлена разбивка на участки поперечника цеха в соответствии с приложенными нагрузками (главным образом крановыми) и формой эпюр моментов. Основная система образована разрезами ригелей по середине, неизвестные обозначены на рис. 23. Оси стоек в местах перемены сечений у

* Криволинейная эпюра приближенно разбивается на ряд прямолинейных участков, вут — на ряд участков с одинаковым сечением.

подкрановых ступеней имеют переломы (этот существенный фактор часто для упрощения во внимание не принимают, что приводит к ощутимой погрешности). Для уменьшения количества участков (ввиду их малости) можно пренебречь площадями эпюр моментов от тормозных сил на участках, расположенных выше подкрановых ступеней, и не нумеровать сечения в местах приложения тормозных сил.

Для данного примера значения параметров следующие: $n=4$; $2 \times m_2=30$; $q=14$. В соответствии с этим готовятся расчетные бланки. Так

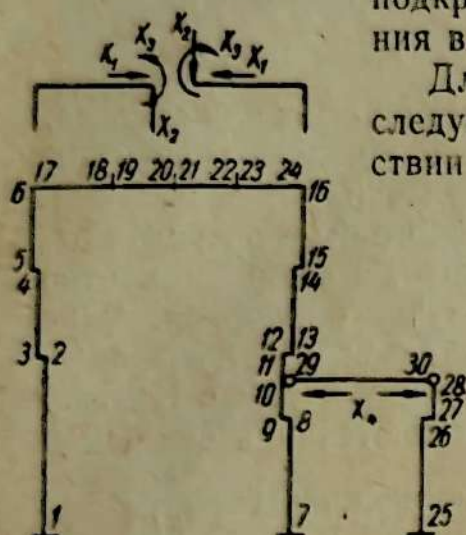


Рис. 23.

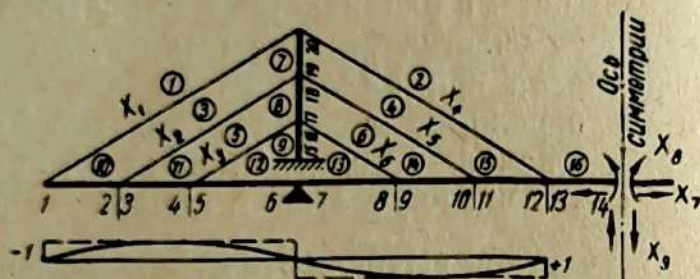


Рис. 24.

как количество загрузений очень значительно, то решение уравнений целесообразно выполнять с использованием обратной матрицы (III. 22) в бланках А, Б, В₁ и В.

Для комбинированных систем:

1. Рассчитывается статически определимая основная система, определяются нормальные силы, изгибающие моменты и ориентировочно величины F и I .

2. Система разбивается на участки: отдельно для нормальных сил (нумерация сечений $1, 2, \dots, m_1$), отдельно для изгибающих моментов (нумерация сечений $1, 2, \dots, 2 \times m_2$).

3. Независимо заполняются табл. 1 и 2, как было указано выше.

4. Подготавливаются остальные бланки 3, 4, А, Б, В (или В₁ и В), 5, 6. Результаты получаются: в табл. 5 — нормальные силы, в табл. 6 — изгибающие моменты.

Например, для схемы вантового моста, изображенного на рис. 24, нумерация сечений для продольных сил указана в кружках, для изгибающих моментов в балке жесткости и пилоне — цифрами в начале и конце каждого участка. Основная система образуется разрезом вант 1—6 и балки жесткости посередине.

Расчет выполняется порознь на отдельные симметричные и обратнoсимметричные грузы $P=0,5$, расположенные в узлах. Суммируя табл. 5с и 5ос, 6с и 6ос для симметричного и обратнoсимметричного состояний, получаем в табл. 5 и 6 искомые ординаты линий влияния M и N . Ординаты линий влияния моментов посередине панелей балки жесткости определяются с помощью специального приема *.

В данном примере для симметричного состояния $n=8$ ($x_1—x_8$), для обратнoсимметричного состояния $n=7$ ($x_1—x_5, x_3$). Остальные параметры для обоих состояний следующие: $m_1=16$; $2 \times m_2=20$; $q=7$. В соответствии с этими значениями подготавливаются бланки 1, 2, 3, 4, А, Б, В₁, В, 5с, 5ос, 6с, 6ос, 5, 6.

Для пространственных систем произвольного вида:

1. Усилия классифицируются по группам: для первой группы в пределах участка усилия являются постоянными (обычно нормальные силы и крутящие моменты), для второй группы эпюры усилий ограничены наклонными линиями (обычно изгибающие моменты).

Рассчитывается статически определимая система и определяются величины усилий для каждой группы.

2. Система разбивается на участки для усилий первой группы, причем для каждой подгруппы отдельно (например для M , $M_{кр}$), и соответствующие номера сечений записываются на отдельных схемах. Нумерация сечений принимается сквозная: сначала для $M_{кр}$, затем по порядку номеров для M . Для каждого участка нумеруется одно сечение. В том же порядке усилия записываются в табл. 1, причем части таблицы, занимаемые подгруппами, выделяются двойными линиями. В графу α для первой подгруппы записываются величины $\alpha_i = \frac{l_i}{EF}$, для второй подгруппы $\alpha_i = \frac{l_i}{GT}$ и т. п.

3. Аналогично система разбивается на участки для усилий второй группы, причем для каждого участка нумеруются два сечения. Если вторая группа состоит из нескольких подгрупп (например, помимо изгибающих моментов ко второй группе можно отнести поперечные силы, если их влиянием нельзя пренебречь), то принимается сквозная нумерация сечений, как было указано в п. 2.

* Вып. 32. Укрпроектстальконструкция, 1960.

Длины участков в разных подгруппах могут быть разными, т. е. не нужно никакого согласования в разбивке участков в разных группах или подгруппах. Для подгруппы изгибающих моментов — $\alpha_i = \frac{l_i}{6EI}$, поперечных сил — $\alpha_i = \frac{l_{ii}}{GF}$.

Построение универсальной программы метода деформаций

IV. 9. Основная система метода деформаций образуется наложением на все узлы системы фиктивных закреплений, препятствующих линейным и угловым перемещениям (пример см. в II. 4). Благодаря этому сложная система распадается на ряд отдельных стержней, закрепленных на концах. В общем случае на каждый узел плоской системы нужно наложить три закрепления, препятствующие в направлениях выбранных осей двум линейным смещениям и одному углу поворота, и на каждый узел пространственной системы — шесть закреплений, препятствующих трем линейным смещениям и трем углам поворота. Система канонических уравнений метода деформаций (II. 1) выражает условия

равенства нулю сумм реактивных усилий по направлению добавленных закреплений от неизвестных деформаций (линейных смещений и углов поворота узлов) и внешней нагрузки.

В частных случаях некоторые из неизвестных равны нулю, другие неизвестные равны по величине. Например, если для рамы на рис. 25 пренебречь продольными деформациями элементов, то все вертикальные смещения узлов будут равны нулю, а горизонтальные смещения узлов каждого уровня равны друг другу.

В работе [102] рассмотрена частная программа метода деформаций, предназначенная для расчета рам с параллельными стой-

ками и непрерывными ригелями в каждом уровне. Составлялось два типа уравнений: первый тип уравнений выражал условие равенства нулю суммы реактивных моментов в i -том узле, второй тип — равенство нулю суммы поперечных сил в стойках j -того яруса. В каждое из уравнений второго типа входило только одно линейное смещение. Задача была решена методом последова-

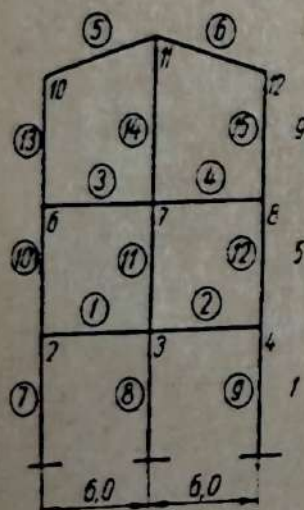


Рис. 25.

тельных приближений. Предварительно по программе выполнялось несложное преобразование — линейные смещения ярусов в уравнения второго типа были выражены через углы поворота узлов и подставлены в уравнения первого типа. Преобразованная система уравнений, содержащая только углы поворота, оказывалась весьма благоприятной для метода последовательных приближений, так как главные коэффициенты были значительно больше побочных при самых различных соотношениях жесткостей стоек и ригелей. Благодаря этому процесс сходился очень быстро*.

Для построенной программы, состоящей из ряда завязанных проверками компактных процедур (составление уравнений первого и второго типов, исключение линейных смещений, решение преобразованной системы методом итерации и определение углов поворота, определение линейных смещений, вычисление моментов, соответствующих найденным деформациям), характерно использование в качестве наглядного зрительного образа схемы рамы. На ней записываются исходные данные и выполняется заключительный этап расчета — определение моментов.

Эти приемы, взятые из алгоритмов перераспределения, могут быть полезными (если их применять ограниченно) и при компактных схемах расчета — запись части величин на схеме рамы может оказаться удобнее, а в некоторых случаях и экономнее табличной записи. Однако имеются и недостатки: отсутствие универсальности (такие алгоритмы предназначены для определенных узких классов задач), появление нестандартных операций с таблицами и графическими схемами, нарушающих ритм вычислительной работы и требующих определенных навыков.

Для того, чтобы избежать этих недостатков, следует задачу поставить абстрактно, не связывая алгоритм с конкретным типом стержневой системы. Это даст возможность построить программу расчета стержневых систем произвольного вида методом деформаций, обладающую такой же универсальностью, как и рассмотренная выше программа метода сил.

Перейдем к построению подобного алгоритма. За основу примем стержень абстрактной стержневой системы, произвольно расположенный в пространстве и закрепленный на концах. Пол-

* Если не выполнять указанное преобразование, то в тех случаях, когда жесткости стоек больше жесткостей ригелей, главные коэффициенты в уравнениях второго типа получаются меньше побочных, процесс сходится чрезвычайно медленно, его практически нельзя применить.

ную информацию о положении N -го стержня в системе и о его упругих свойствах дает табл. C_N .

Таблица C_N

	a	b	...	v	$n+1$	...	$n+q$
a	r_{aa}	r_{ab}		r_{av}	$r_{a, n+1}$		$r_{a, n+q}$
b	r_{ba}	r_{bb}		r_{bv}	$r_{b, n+1}$		$r_{b, n+q}$
...							
v	r_{va}	r_{vb}		r_{vv}	$r_{v, n+1}$		$r_{v, n+q}$

Буквами $a, b, \dots, v$ обозначены номера неизвестных, оказывающих влияние на данный стержень.

В клетках табл. C_N записаны реактивные усилия в закреплениях стержня от единичных кинематических факторов $x_a = 1$, $x_b = 1, \dots, x_v = 1$ и местной нагрузки на стержне.

Неизвестными можно считать углы поворота и линейные смещения узлов системы, а также любые их линейные комбинации. В качестве «стержня» можно принять любую статически неопределимую систему: брус (прямой или изогнутый), систему брусьев или сплошные конструкции, такие как панель, диафрагма, тем или иным способом закрепленные в пространстве.

Как известно, идея метода деформаций состоит в том, что основная статически неопределимая система рассматривается состоящей из стержней, каждый из которых «изучен», т. е. табл. C_N для него известна. Величины реактивных усилий r_{ik} от единичных деформаций (в гл. II они были названы жесткостями стержня) могут определяться по готовым формулам или таблицам. Составление табл. C_N для стержней имеет смысл выделить в отдельную процедуру, связанную с частным типом задачи. Раскрытие же статической неопределимости для системы в целом выполняется во всех случаях одинаково и может быть представлено в виде стандартной программы вычислений, состоящей из трех процедур: засылки коэффициентов из табл. C_N в общую матрицу A системы уравнений, решения системы уравнений и определение перемещений узлов, определение реактивных усилий на концах стержней от нагрузки и перемещений узлов. Первая и третья процедуры даны ниже.

IV. 10. Процедура засылки коэффициентов и свободных членов из матриц C_N в матрицу A ;

начало примечание бланк табл. A см. в III. 19 (схема 9);

для N : = номера стержней выполнить

для i : = $a, b, \dots, v$ выполнить

для k : = $a, b, \dots, v, n+1, \dots, n+q$ выполнить

$M1 : i.kT_A + i.kT_{C_N} \rightarrow i.kT_A$;

проверка симметрия коэффициентов относительно главной диагонали табл. A ;

проверка $\{S_1=S_2; S_3=S_4\}$

конец процедуры

S_1 — сумма главных коэффициентов во всех таблицах C_N ;

S_2 — сумма главных коэффициентов в табл. A ; S_3 — сумма свободных членов во всех таблицах C_N ; S_4 — сумма свободных членов в табл. A .

Номера $a, b, \dots$, строк и граф указаны в табл. C_N .

Один и тот же символ $i.kT_A$ в левой и правой частях оператора засылки указывает, что происходит суммирование с предыдущим значением (накопление). Обычно такое накопление выполняется лишь для главных коэффициентов и свободных членов.

Суммирование последних выполняется во вспомогательной таблице, после чего величины $i.i$ и $i.p$ переносятся в табл. A ($i=1, 2, \dots, n$; $p=n+1, n+2, \dots, n+q$).

Вспомогательные таблицы для $i.i$ и $i.p$.

i	1	2	...	n
$i.i$				
Σ				

i	1	2	...	n
$i.p$				
Σ				

В результате выполнения процедуры получаем в табл. A симметричную систему уравнений метода деформаций, каждая строчка которой выражает равенство нулю суммарного реактивного усилия в i -том фиктивном закреплении от деформаций узлов и нагрузки.

IV. 11. Процедура вычисления реактивных усилий от нагрузки и найденных деформаций;

начало примечание бланки табл. C_N и R_N помещены на схеме 15.

Схема 15

Таблица C_N

	a	b	...	v	$n+1$	...	$n+q$
a							
b							
...							
v							
S							

Таблица R_N

a	b	...	v	S
1				
2				
...				
q				

x^p				
-------	--	--	--	--

Исходные данные помещены в табл. C_N , строка S — для контрольных сумм. Во вспомогательную строку x^p последовательно переносятся значения неизвестных $x_a, x_b, \dots, x_v$, полученные при решении уравнений в строках $p=1, 2, \dots, q$ табл. B (бланк табл. B , см. схему 9). Результаты — реактивные усилия — получают в табл. R_N (N — номер стержня);

для $p := 1, 2, \dots, q$ выполнить

для $i := a, b, \dots, v, S$ выполнить

$$\sum \text{стр } x^p \times \text{стр } iT_{CN} \big|_a + i.n + pT_{CN} \rightarrow p.iT_{RN};$$

проверка $\sum \text{стр } iT_{RN} \big|_a = i.ST_{RN}$

конец процедуры

Теперь можно записать полную программу в процедурах.

IV. 12. Компоновочная программа расчета произвольных стержней систем методом деформаций:

начало засылка коэффициентов и свободных членов из матриц C_N в матрицу A ;

решение системы уравнений по одной из процедур (выбирается инженером);

вычисление реактивных усилий от нагрузки и найденных деформаций

конец программы.

Если количество уравнений не очень велико, то удобнее воспользоваться для решения систем уравнений программой метода исключения.

Опасения в отношении метода Гаусса, которые существуют у многих проектировщиков, связаны с представлением о трудностях решения на логарифмической линейке. Эти трудности в значительной мере отпадают при переходе на клавишные машины. Кроме того, подсчет трудоемкости, применяемый при оценке метода Гаусса для полностью заполненных матриц, является неправильным в отношении уравнений рассматриваемого типа. Обычно система уравнений имеет ленточную или ленточную окаймленную структуру (II.7) с узкой лентой. В этом случае трудоемкость расчета зависит линейно от количества неизвестных и приблизительно в квадратичной степени от ширины ленты (VI.9), поскольку вычисления выполняются для элементов матрицы только в пределах ленты; к тому же используется симметрия коэффициентов.

Решение таких систем даже с большим количеством неизвестных (20—30) по компактной схеме Гаусса на клавишных автоматах выполняется в течение нескольких часов. О других преимуществах метода Гаусса было сказано в III.26.

IV.13. Данная программа реализует по существу матричный алгоритм метода деформаций и обладает как его достоинствами, так и некоторыми недостатками: достоинства — универсальность и стандартность, рассчитывать можно произвольные стержневые системы как плоские, так и пространственные; недостаток — большое количество записей в таблицах C_N (другие таблицы заполняются компактно).

В самом деле, в квадратных матрицах C_N для отдельных стержней содержится, как видно, избыточная информация: симметрия не используется и одинаковые побочные коэффициенты $r_{ik} = r_{ki}$ записаны дважды; для стержня постоянного сечения некоторые главные коэффициенты имеют одинаковые значения, они также записываются дважды. Посмотрим, например, к чему сведется подготовка исходных данных для плоской рамы, изображенной на рис. 25.

Для получения ленточной структуры матрицы A неизвестные нумеруются в следующем порядке: 1 — смещение первого яруса, 2, 3, 4 — углы поворота узлов первого уровня, 5 — смещение второго яруса, 6, 7, 8 — углы поворота узлов второго уровня и т. д. Номера стержней записаны на схеме рамы в кружках.

Матрицы C_N для ригеля 1 (аналогично для остальных ригелей) и стоек 7 и 10 (аналогично для остальных стоек) даны ниже.

$$C_1$$

	2	3
2	r_{22}	r_{23}
3	r_{32}	r_{33}

$$C_7$$

	1	2
0	r_{01}	r_{02}
1	r_{11}	r_{12}
2	r_{21}	r_{22}

$$C_{10}$$

	2	5	6
2	r_{21}	r_{25}	r_{26}
5	r_{52}	r_{55}	r_{56}
6	r_{62}	r_{65}	r_{66}

На ригель 1 влияют два неизвестных угла поворота x_2 и x_3 .

На стойку 7 влияют два неизвестных: x_1 — смещение первого яруса и x_2 — угол поворота второго узла. В табл. C_7 прибавлена строка 0 для реактивных моментов в защемлении.

На стойку 10 влияют три неизвестных: x_2 — угол поворота второго узла; x_5 — смещение второго яруса и x_6 — угол поворота шестого узла.

Таблицы C_N следует дополнить графами свободных членов от местной нагрузки, действующей на стержень, для 1, 2, ..., q комбинаций загрузки. Свободные члены для узловой нагрузки непосредственно записываются в табл. А. Соответствующая графа в табл. C_1 остается пустой.

Для стержня постоянного сечения значения коэффициентов табл. C_1 будут:

$$r_{22} = r_{33} = m_1, \quad r_{23} = r_{32} = 0,5m_1,$$

где $m_1 = \frac{4EI}{l}$.

Значения коэффициентов табл. C_{10} :

$$r_{22} = r_{66} = m_{10}; \quad r_{26} = r_{62} = 0,5m_{10};$$

$$r_{25} = r_{52} = r_{65} = r_{56} = v_{10}; \quad r_{55} = w_{10},$$

где $m = \frac{4EI}{l}$, $v = -\frac{6EI}{l^2} = -1,5 \frac{m}{l}$, $w = \frac{12EI}{l^3} = -\frac{2v}{l}$

так называемые первая, вторая и третья жесткости стержня.

Таким образом, в данном случае по универсальной программе для ригелей записываются по четыре исходных величины (вместо одной — m , в частной программе из работы [102]), а для стоек — по девять величин (вместо трех m , v , w). Для стержней переменного сечения лишних записей будет меньше — для дан-

ного примера, соответственно, четыре величины вместо трех и девять величин вместо шести.

Выбор основной системы и неизвестных, а также заполнение таблиц C_n производится инженером.

В частных случаях возможны упрощения: использование регулярности, монотонности структуры, одностипности стержней и т. д.

IV. 14. То обстоятельство, что в универсальной программе не экономится запись исходных данных, нельзя рассматривать только как недостаток — именно это дает возможность придать программе исключительно простую форму и облегчить пользование ею, особенно для сложных задач.

Конечно, можно было построить программу так, чтобы избежать лишних записей, предложив вычислителю в процессе счета выбирать числа из сокращенных таблиц в соответствии с определенными логическими условиями. Однако это требует от человека большого умственного напряжения и в итоге не ускоряет, а замедляет счет.

Забегая вперед, укажем, что такая выборка чисел легко выполняется электронной вычислительной машиной при самых сложных закономерностях в их расположении. Поэтому программа для ЭЦВМ построена на другой основе, с максимальным сокращением ввода и экономией памяти.

Здесь мы сталкиваемся с различными возможностями человека и машины. Человек легко воспринимает зрительные образы и монотонные закономерности, например, упорядоченные массивы, отрезки строк или граф в разных таблицах, которые можно совместить при вычислениях и т. п. Усложнение логики счета в программах хотя и возможно, но неудобно для вычислителя, требует напряжения и неизбежно приводит к ошибкам. Для программно управляемой же машины нет неудобных действий, она безотказно выполняет любые логические построения, заданные программой.

Программа определения частот и форм собственных колебаний произвольной системы

IV. 15. В работе [102] приведена инженерная программа определения основной (низшей) частоты свободных колебаний системы с n — степенями свободы методом последовательных приближений, которая использовалась для динамического расчета мостов через Днепр в Киеве.

Для решения задачи методом последовательных приближений представим уравнения (1) в следующем виде:

$$y_{ij}^{(v+1)} = \frac{\rho_j^2}{g} \sum_{k=1}^n \delta_{ik} Q_k y_{kj}^{(v)}, \quad (3)$$

где i — принимает значения $1, 2, 3, \dots, n$; v — номер приближения; вместо m_k подставлена величина Q_k/g , где Q_k — вес сосредоточенного груза, $g=9,81$ м/сек² — ускорение силы тяжести.

Так как величина ρ_j^2 нам неизвестна, то будем пользоваться другой формулой:

$$y_{ij}^{(v+1)} = \sum_{k=1}^n \delta_{ik} Q_k y_{kj}^{(v)}. \quad (4)$$

При этом форма деформации сохранится, но на каждом шаге значения будут делиться на постоянное число ρ_j^2/g . Подставляя в первую часть формулы (4) значения $y_{ki}^{(v)}$ для v -го приближения ($v=1, 2, 3, \dots$), получаем величины $y_{ij}^{(v+1)}$ для последующего $v+1$ -го приближения.

Первые приближения принимаются в соответствии с ожидаемой формой (IV.19). Процесс итерации продолжается до тех пор, пока в двух последовательных приближениях упругие линии станут подобными, т. е. для всех точек системы будет соблюдаться равенство:

$$y_{ij}^{(v)}/y_{ij}^{(v+1)} = \text{const} \quad (i=1, 2, \dots, n).$$

Как было показано выше, для формулы (4) это постоянное число равно ρ^2/g . Поэтому

$$\frac{y_{ij}^{(v)}}{y_{ij}^{(v+1)}} = \frac{\rho^2}{g}.$$

Отсюда $\rho = \sqrt{\frac{y_{ij}^{(v)} g}{y_{ij}^{(v+1)}}} = 3.14 \sqrt{\frac{y_{ij}^{(v)}}{y_{ij}^{(v+1)}}}. \quad (5)$

Для определения высших форм колебаний воспользуемся методом ортогонализации последовательных итераций ([121], стр. 314), основанном на том, что каждая форма колебаний должна быть ортогональна другим формам.

Физический смысл указанного свойства состоит в том, что сумма работ инерционных сил одной формы на соответствующих перемещениях другой формы должна быть равна нулю.

Условие взаимной ортогональности первой и второй формы запишется так:

$$\sum_{k=1}^n Q_k y_{k1} y_{k2}^{(v)} = 0. \quad (6)$$

Вместо массы m_k записан вес Q_k (g — сократилось).

Примем следующее выражение для второй формы колебаний:

$$y_{k2}^{(v)} = y_{k2}^{(0, v)} - \psi_{21} y_{k1}, \quad (7)$$

где $y_{k2}^{(v)}$ — v -тое приближение для ординат второй формы;

$y_{k2}^{(0, v)}$ — ординаты второй формы до ортогонализации; y_{k1} — найденные ранее ординаты первой формы колебаний; $k = 1, 2, \dots, n$ — точки приложения грузов; ψ_{21} — коэффициент, определяемый из условия ортогональности первой и второй формы.

Подставив выражение (7) в (6), получим

$$\sum_{k=1}^n Q_k y_{k1} (y_{k2}^{(0, v)} - \psi_{21} y_{k1}) = 0. \quad (8)$$

Отсюда коэффициент ортогонализации

$$\psi_{21} = \frac{\sum_{k=1}^n Q_k y_{k1} y_{k2}^{(0, v)}}{\sum_{k=1}^n Q_k y_{k1}^2}. \quad (9)$$

Процесс итерации состоит из следующих этапов:

а) задаемся подходящими начальными значениями y_{k2}^0 второй формы колебаний (IV. 19);

б) по формуле (9) определяем коэффициент ψ_{21} ;

в) находим $y_{k2}^{(v)}$ по формуле (7);

г) подставляем $y_{k2}^{(v)}$ в формулу (4) и находим следующее приближение $y_{k2}^{(0, v+1)}$ для ординат второй формы.

Повторяем циклически пункты б, в и г до тех пор, пока не выполнится условие $y_{ij}^{(v)}/y_{ij}^{(v+1)} = \text{const}$. Затем по формуле (5) определяем частоту второй формы.

Для определения колебаний по третьей форме составляем условие ортогональности с первой и второй формой:

$$\sum_{k=1}^n Q_k y_{k1} y_{k3}^{(v)} = 0; \quad (10)$$

$$\sum_{k=1}^n Q_k y_{k2} y_{k3}^{(v)} \parallel 0. \quad (11)$$

Примем следующее выражение для третьей формы колебаний:

$$y_{k3}^{(v)} = y_{k3}^{(0, v)} - \psi_{31} y_{k1} - \psi_{32} y_{k2}, \quad (12)$$

где ψ_{31} и ψ_{32} определяются из условий (10) и (11):

$$\psi_{31} = \frac{\sum_{k=1}^n Q_k y_{k1} y_{k3}^{(0, v)}}{\sum_{k=1}^n Q_k y_{k1}^2}; \quad (13)$$

$$\psi_{32} = \frac{\sum_{k=1}^n Q_k y_{k2} y_{k3}^{(0, v)}}{\sum_{k=1}^n Q_k y_{k2}^2}. \quad (14)$$

Процесс последовательных приближений производится в той же последовательности, что и для второй формы.

Далее по индукции можно получить формулы для определения любой j -той формы и частот свободных колебаний.

Этот вопрос ставится в VII. 11 при построении соответствующей программы для ЭЦВМ. В программе же для клавишных машин ограничимся тремя первыми формами.

IV. 17. Для получения компактной схемы алгоритма несколько видоизменим расчетные формулы и выпишем их в той последовательности, в которой они выполняются.

Для всех трех форм:

$$P_{kj} = \mu Q_k y_{kj}^{(v)}; \quad (1)$$

$$y_{ij}^{(v+1)} = \sum_{k=1}^n P_{kj} \delta_{ik}; \quad (2)$$

$$\lambda_{ij} = y_{ij}^{(v)} / y_{ij}^{(v+1)}; \quad (3)$$

$$\varphi_j = 3,14 \sqrt{\mu \lambda_j}. \quad (4)$$

Для первой формы $j=1$, второй формы $j=2$, третьей формы $j=3$. Масштабный коэффициент μ выбирается таким, чтобы порядок чисел в таблицах был удобен для счета ($\mu = 10^2, 10^3, \dots$).

Для второй формы:

$$a_{11} = \sum_{k=1}^n P_{k1} y_{k1}; \quad (5)$$

$$a_{21} = \sum_{k=1}^n P_{k1} y_{k2}^{(0, \nu)}; \quad (6)$$

$$\psi_{21} = a_{21}/a_{11}; \quad (7)$$

$$y_{k2}^{(\nu)} = y_{k2}^{(0, \nu)} - \psi_{21} y_{k1}; \quad (8)$$

$$\sum_{k=1}^n P_{k1} y_{k2}^{(\nu)} = 0. \quad (9)$$

Для третьей формы:

$$a_{31} = \sum_{k=1}^n P_{k1} y_{k3}^{(0, \nu)}; \quad (10)$$

$$a_{22} = \sum_{k=1}^n P_{k2} y_{k2}; \quad (11)$$

$$a_{32} = \sum_{k=1}^n P_{k2} y_{k3}^{(0, \nu)}; \quad (12)$$

$$\psi_{31} = a_{31}/a_{11}; \quad \psi_{32} = a_{32}/a_{22}; \quad (13)$$

$$y_{k3}^{(\nu)} = y_{k3}^{(0, \nu)} - \psi_{31} y_{k1} - \psi_{32} y_{k2}; \quad (14)$$

$$\sum_{k=1}^n P_{k1} y_{k3}^{(\nu)} = 0; \quad \sum_{k=1}^n P_{k2} y_{k3}^{(\nu)} = 0. \quad (15)$$

Формулы (9) и (15) используются для проверки.

На основании формул (1) — (15) ниже составлена программа определения первых трех форм и частот собственных колебаний.

Схема 16

Таблица О

	1	2	...	n	S	Q
1						
2						
...						
n						

Продолжение схемы 16

$\nu=1$ $\nu=2$ $\nu=3$ Таблица 1

	y	P	λ	y	P	λ	y	P	λ		
	1	2	3	1	2	3	1	2	3	...	y P
1											
2											
...											
n											
S											

$$\rho_1 = 3,14 \sqrt{\mu \times \lambda_1}$$

$$\rho_2 = 3,14 \sqrt{\mu \times \lambda_2}$$

$$\rho_3 = 3,14 \sqrt{\mu \times \lambda_3}$$

$\nu=1$ $\nu=2$ Таблица 2

	y^0	P	—	y	y^0	P	λ	y		
	1	2	3	4	1	2	3	4	...	y P
1			—							
2			—							
...			—							
n			—							
S			—							

Таблица 4

	a_{11}	a_{12}	ψ_{12}
$\nu \downarrow$	1	2	3
1			
2			
...			

$\nu=1$ $\nu=2$ Таблица 3

	y^0	P	—	y	y^0	P	λ	y		
	1	2	3	4	1	2	3	4	...	y
1			—							
2			—							
...			—							
n			—							
S			—							

Таблица 5

	a_{13}	a_{23}	a_{33}	ψ_{13}	ψ_{23}
$\nu \downarrow$	1	2	3	4	5
1					
2					
...					

Табличные бланки помещены на схеме 16. В графе S исходной табл. 0 записаны суммы всех предшествующих чисел. Для удобства изучения программы сверху таблиц даны буквенные обозначения (которыми вычислитель не пользуется). Индекс ν в таблицах указывает номер приближения. Так, например, обозначение $T_{1,2}$ указывает, что заполняется поле табл. 1 для приближения $\nu=2$.

IV. 18. Программа определения первых трех форм и частот колебаний произвольной системы:

начало примечание программа может быть использована для расчета любой системы (как дискретной, так и непрерывной), для которой известна матрица перемещений T_0 от единичных грузов;

Определение первой формы:

$$\Sigma | \text{гр } 1T_{1,1} |_1^n \rightarrow S.1T_{1,1}; \quad \Sigma | \text{гр } 1T_{2,1} |_1^n \rightarrow S.1T_{2,1}; \\ \Sigma | \text{гр } 1T_{3,1} |_1^n \rightarrow S.1T_{3,1};$$

для $\nu := 1, 2, \dots$ пока $\lambda \neq \text{const}$ выполнить
начало

$$\mu \times | \text{гр } QT_0 \times \text{гр } 1T_{1,\nu} |_1^n \rightarrow | \text{гр } 2T_{1,\nu} |_1^n; \\ \Sigma | \text{гр } 2T_{1,\nu} |_1^n \rightarrow S.2T_{1,\nu};$$

проверка $\mu \times \Sigma | \text{гр } QT_0 \times \text{гр } 1T_{1,\nu} |_1^n = S.2T_{1,\nu};$
для $i := 1, 2, \dots, n$, S выполнить

$$\{ \Sigma | \text{гр } 2T_{1,\nu} \times \text{гр } iT_0 |_1^n \rightarrow i.T_{1,\nu+1};$$

проверка $\Sigma | \text{гр } 1T_{1,\nu+1} |_1^n = S.1T_{1,\nu+1} \}$:

$$| \text{гр } 1T_{1,\nu} |_1^n / | \text{гр } 1T_{1,\nu+1} |_1^n \rightarrow \text{гр } 3T_{1,\nu} |_1^n;$$

проверка $\Sigma | \text{гр } 1T_{1,\nu+1} \times \text{гр } 3T_{1,\nu} |_1^n = S.1T_{1,\nu}$
конец ν ;

примечание после окончания процесса последовательных приближений определяем форму свободных колебаний делением всех ординат на величину первой ординаты, при этом ν имеет последнее значение;

$$| \text{гр } 1T_{1,\nu} |_1^S / | \text{гр } 1T_{1,\nu} |_1^S \rightarrow | \text{гр } yT_{1,\nu} |_1^S; \quad \text{проверка } \Sigma | \text{гр } yT_{1,\nu} |_1^S = S.yT_{1,\nu}; \\ | \text{гр } 2T_{1,\nu} |_1^S / | \text{гр } 1T_{1,\nu} |_1^S \rightarrow | \text{гр } PT_{1,\nu} |_1^S; \quad \text{проверка } \Sigma | \text{гр } PT_{1,\nu} |_1^S = S.PT_{1,\nu};$$

Определение второй формы:

для $\nu := 1, 2, \dots$ пока $\lambda \neq \text{const}$ выполнить начало

$$\Sigma | \text{гр } P \times \text{гр } y |_1^n T_1 \rightarrow \nu.1T_4;$$

$$\Sigma | \text{гр } PT_1 \times \text{гр } 1T_{2,v} |_1^n \rightarrow v.2T_4;$$

$$v.2T_4/v.1T_4 \rightarrow v.3T_4;$$

$$| \text{гр } 1T_{2,v} |_1^n - v.3T_4 \times | \text{гр } yT_1 |_1^n \rightarrow | \text{гр } 4T_{2,v} |_1^n;$$

проверка $\Sigma | \text{гр } PT_1 \times \text{гр } 4T_{2,v} |_1^n = 0;$

$$\mu \times | \text{гр } QT_0 \times \text{гр } 4T_{2,v} |_1^n \rightarrow | \text{гр } 2T_{2,v} |_1^n;$$

$$\Sigma | \text{гр } 2T_{2,v} |_1^n \rightarrow S.2T_{2,v};$$

проверка $\mu \times \Sigma | \text{гр } QT_0 \times \text{гр } 4T_{2,v} |_1^n = S.2T_{2,v};$

для $i := 1, 2, \dots, n$, **S** **выполнить**

$$\Sigma | \text{гр } 2T_{2,v} \times \text{гр } iT_0 |_1^n \rightarrow i.1T_{2,v+1};$$

проверка $\Sigma | \text{гр } 1T_{2,v+1} |_1^n = S.1T_{2,v+1};$

если $v > 1$ **то**

$$| \text{гр } 1T_{2,v} |_1^n / | \text{гр } 1T_{2,v+1} |_1^n \rightarrow | \text{гр } 3T_{2,v} |_1^n;$$

проверка $\Sigma | \text{гр } 1T_{2,v+1} \times \text{гр } 3T_{2,v} |_1^n = S.1T_{2,v}$

конец v ;

$$| \text{гр } 1T_{2,v} |_1^S / 1.1T_{2,v} \rightarrow | \text{гр } yT_2 |_1^S; \text{ проверка } \Sigma | \text{гр } yT_2 |_1^n = S.yT_2;$$

$$| \text{гр } 2T_{2,v} |_1^S / 1.1T_{2,v} \rightarrow | \text{гр } PT_2 |_1^S; \text{ проверка } \Sigma | \text{гр } PT_2 |_1^n = S.PT_2;$$

Определение третьей формы:

для $v := 1, 2 \dots$ **пока** $\lambda \neq \text{const}$ **выполнить**

$$\Sigma | \text{гр } PT_1 \times \text{гр } 1T_{3,v} |_1^n \rightarrow v.1T_5;$$

$$\Sigma | \text{гр } P \times \text{гр } y |_1^n T_2 \rightarrow v.2T_5;$$

$$\Sigma | \text{гр } PT_2 \times \text{гр } 1T_{3,v} |_1^n \rightarrow v.3T_5;$$

$$v.1T_5/v.1T_4 \rightarrow v.4T_5;$$

$$v.3T_5/v.2T_5 \rightarrow v.5T_5;$$

$$| \text{гр } 1T_{3,v} |_1^n - v.4T_5 \times | \text{гр } yT_1 |_1^n - v.5T_5 \times | \text{гр } yT_2 |_1^n \rightarrow \text{гр } 4T_{3,v} |_1^n;$$

проверка $\{ \Sigma | \text{гр } PT_1 \times \text{гр } 4T_{3,v} |_1^n = 0;$

$$\Sigma | \text{гр } PT_2 \times \text{гр } 4T_{3,v} |_1^n = 0 \};$$

$$\mu \times | \text{гр } QT_0 \times \text{гр } 4T_{3,v} |_1^n \rightarrow | \text{гр } 2T_{3,v} |_1^n;$$

$$\Sigma | \text{гр } 2T_{3,v} |_1^n \rightarrow S.2T_{3,v};$$

проверка $\mu \times \Sigma | \text{гр } QT_0 \times \text{гр } 4T_{3,v} |_1^n = S.2T_{3,v};$

для $i := 1, 2, \dots, n$, **S** **выполнить**

$$\Sigma | \text{гр } 2T_{3,v} \times \text{гр } iT_0 |_1^n \rightarrow i.1T_{3,v+1};$$

проверка $\Sigma | \text{гр } 1T_{3, v+1} |_1^n = S \cdot 1T_{3, v+1}$;

если $v > 1$ то $| \text{гр } 1T_{3, v} / \text{гр } 1T_{3, v+1} |_1^n \rightarrow \text{гр } 3T_{3, v}$;

проверка $\Sigma | \text{гр } 1T_{3, v+1} \times \text{гр } 3T_{3, v} |_1^n = S \cdot 1T_{3, v}$;

конец v ;

$| \text{гр } 1T_{3, v} |_1^S / 1 \cdot 1T_{3, v} \rightarrow | \text{гр } yT_{3, v} |_1^S$; проверка $\Sigma | \text{гр } yT_{3, v} |_1^n = S \cdot yT_{3, v}$;
конец программы.

IV. 19. Для расчета по программе, изложенной в предыдущем параграфе, необходимо задаться начальной формой колебаний. Чем ближе форма к действительной, тем быстрее сходится процесс последовательных приближений.

Если целью расчета является определение частот свободных колебаний, то при вычислении перемещений y большая точность обычно не нужна; к тому же при извлечении корня по формуле (5) из IV. 16 погрешность уменьшается. Например, в работе [102] при определении основной частоты колебаний вантового моста принята простая начальная форма, изображенная пунктиром на рис. 24, — прогибы всех узлов балки равны -1 , консоли $+1$. Эта форма далека от действительной, но все же качественно соответствует ей. Поэтому для получения практически точного результата достаточно было выполнить шесть приближений. При этом $k = 0,716$ гц. После четырех приближений $k = 0,739$ гц. Таким образом, вполне можно было ограничиться четырьмя приближениями, что дает погрешность всего 3%.

Если целью расчета является определение не только частот, но и форм колебаний (например, при сейсмических воздействиях), то требования к точности расчета должны быть увеличены (особенно при определении высших форм колебаний) что удлиняет процесс итерации. Здесь удачный выбор начальной формы упругой линии имеет уже существенное значение.

При расчете сооружений по консольной или балочной схеме можно воспользоваться рекомендациями проф. И. Л. Корчинского [53] и вычислять начальные значения y_{ij} по формуле:

$$y_{ij} = \sin \left(\alpha_j \frac{h_i}{H} \right), \quad (1)$$

где α_j — для первых трех тонов принимает значения: для консоли — $\alpha_1 = \pi/2$, $\alpha_2 = 3\pi/2$, $\alpha_3 = 5\pi/2$ (рис. 26, а); для балки — $\alpha_1 = \pi$, $\alpha_2 = 2\pi$, $\alpha_3 = 3\pi$ (рис. 26, б).

В данном случае к консоли или к балке приводится сложное сооружение переменной жесткости с неодинаковыми сосредото-

ченными массами, но в качестве первого приближения такая кривая годится.

Формула (1) определяет лишь общий характер деформаций, соответствующий данной форме колебаний. Так, для консоли противоречие состоит в том, что угол поворота в защемлении получается не равным нулю.

Указав на это обстоятельство, проф. Н. К. Снитко предложил другое выражение для приближенного определения первой формы

$$y_{i1} = 1 - \cos \left(\frac{\pi}{2} \frac{h_i}{H} \right). \quad (2)$$

В защемлении равны нулю как прогиб, так и угол поворота.

Чем ближе начальная форма к действительной, тем быстрее сходится процесс итераций.

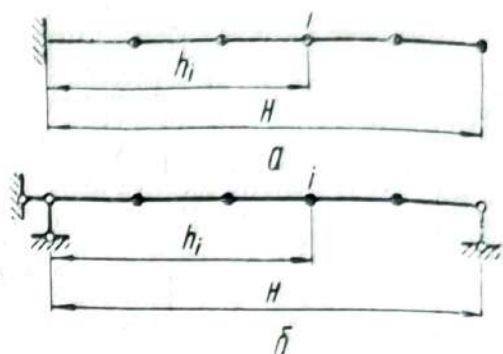


Рис. 26.

Прием построения программ метода сеток *

IV. 20. В данном параграфе рассматривается общий прием составления уравнений метода сеток, который нашел применение при построении программ для расчета пластин на клавишных автоматах и впоследствии — для расчета оболочек на ЭЦВМ. Составление системы уравнений метода сеток является сравнительно нетрудоемкой операцией — используются те или иные шаблоны для внутренних, предконтурных, предугловых, угловых и контурных узлов сетки.

Коэффициенты для этих шаблонов записываются в матрицу системы уравнений. При выполнении этой, казалось бы, простой работы на практике возникают две трудности.

Во-первых, пользование шаблонами для составления системы уравнений при разнообразных граничных условиях требует определенной квалификации, во-вторых, даже квалифицированный специалист при отсутствии организации проверок почти неизбежно

* С применением метода сеток для расчета пластинок можно ознакомиться по известной книге проф. П. М. Варвака [10]. Дальнейшему развитию метода сеток для решения смешанной задачи теории упругости посвящена работа М. И. Длугача [36]. Популярное изложение метода сеток в общематематическом плане дано в книге [89].

но делает ошибки как в вычислениях (при составлении нестандартных шаблонов), так и при переписывании коэффициентов в матрицу системы уравнений.

Между тем имеется возможность построить общий алгоритм составления уравнений метода сеток при произвольных граничных условиях, сделать процесс составления уравнений автоматическим и контролируемым системой построчных самопроверок.

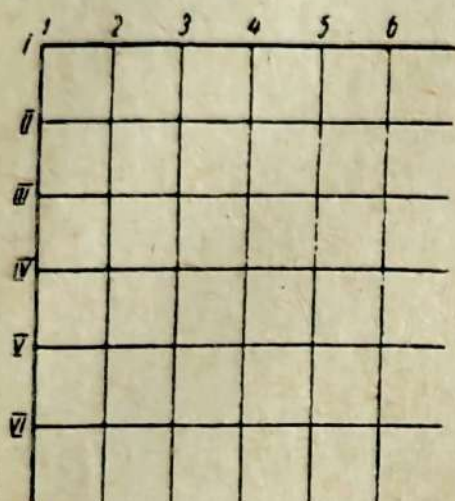


Рис. 27.

Суть алгоритма состоит в получении различных частных систем уравнений из некоторой обобщенной (абстрактной) матрицы, инвариантной по отношению к граничным условиям, с помощью определенного преобразования и вычеркивания «лишних» строк и граф, отвечающих заданным граничным условиям.

В виде примера рассмотрим предпосылки программы для расчета изгибаемых прямоугольных плит.

Для того, чтобы иметь возможность варьировать количество шагов сетки, обобщенная система уравнений сформулирована для части области, начиная от левого верхнего угла (рис. 27). Горизонтальные линии сетки обозначены римскими цифрами I, II, III..., вертикальные линии — арабскими цифрами 1, 2, 3....

Неизвестными системы уравнений являются прогибы узлов w_i и введенные М. И. Длугачем [34] величины k_i — разности прогибов законтурного $i+1$ и предконтурного $i-1$ узлов сетки

$$k_i = w_{i+1} - w_{i-1}.$$

Для общего случая свободного края вместо w_{i+1} подставляют их значения, выраженные через w_m для контурных и внутренних узлов сетки, и получают выражения для k_i , в которых законтурные узлы отсутствуют.

Функция прогибов k_i удобна тем, что в частном случае защемления в узле i имеем:

$$w_{i+1} = w_{i-1}; \quad k_i = 0.$$

Конкретные, довольно громоздкие, выражения для уравнений приводить не будем; нас в данном случае интересует подход к решению задачи, а не сама задача. Аналогичный подход может быть и в тех случаях, когда область и сетка непрямоугольны,

дифференциальное уравнение с частными производными имеет другой тип и т. д.

Вместо одной таблицы коэффициенты обобщенной системы уравнений для узлов сетки помещены в трех таблицах (см. схему 17).

В строках табл. 1 записаны коэффициенты при ω . Табл. 1 представляет собой пятиклеточную матрицу, все клетки которой также являются матрицами (таблицами), обозначенными буквами. Клетки табл. 1 выражают связи между группами узлов в уравнениях.

Так, клетка I.I табл. 1 содержит коэффициенты, связывающие узлы горизонтали I с этими же узлами, клетка I.II содержит коэффициенты, связывающие узлы горизонтали I с узлами горизонтали II и т. д.

В строках табл. 2 аналогично записаны коэффициенты при ω в выражениях для k . Табл. 2 представляет собой трехклеточную матрицу.

	1	2	3	4	5
I					
II					
III					
IV					
V					

Рис. 28.

Наконец, в графах табл. 3 (матрица транспонирована) записаны коэффициенты при k в уравнениях.

Программа предусматривает получение по стандартным табл. 1, 2 и 3 для части плиты одноименных таблиц для всей плиты при любой заданной сетке. Пусть, например, нужно компоновать табл. 1, 2 и 3 для сетки, изображенной на рис. 28. Для этого соответствующие клетки строк I, II, III переписываются из стандартных таблиц при заданных параметрах, строка IV обратна строке II, строка V обратна строке I.

Задания на компоновку таблиц выдаются в виде пустых бланков с пронумерованными указанным образом строками и графами.

Табл. 1, 2, 3 абстрагированы от граничных условий. В данном случае это дает возможность использовать при компоновке таблиц симметрию сетки, хотя граничные условия могут быть не симметричными.

Правильность составления табл. 1, 2, 3 в числовом виде (после подстановки параметров) контролируется простой проверкой сумм коэффициентов в строках, которые должны быть равны нулю.

Теперь нужно «привязать» таблицы к данным граничным условиям. Сначала учитываем наличие заземлений: если в узле имеется заземление, то одноименные строки табл. 2 и 3 зачер-

Обобщенная таблица 1

	I	II	III	IV	V	VI	VII	...
I	F_1	f_1	f_2					
II	f_1	F_2	f_1	f_2				
III	f_2	f_1	F_3	f_1	f_2			
IV		f_2	f_1	F_3	f_1	f_2		
V			f_2	f_1	F_3	f_1	f_2	
VI				f_2	f_1	F_3	f_1	
VII					f_2	f_1	F_3	
...								

Обобщенная таблица 2

	I	II	III	IV	V	VI	VII	...
I	F_4	f_3						
II	f_4	F_5	f_4					
III		f_4	F_5	f_4				
IV			f_4	F_5	f_4			
V				f_4	F_5	f_4		
VI					f_4	F_5	f_4	
VII						f_4	F_5	
...								

квиваются (исключаются), так как коэффициент k в этом случае равен нулю.

Для того, чтобы получить обобщенную матрицу A системы уравнений, следует подставить значения k , выраженные через ω

Обобщенная таблица 3

	I	II	III	IV	V	VI	VII	...
I	F_6	f_5						
II	f_6	F_7	f_6					
III		f_6	F_7	f_6				
IV			f_6	F_7	f_6			
V				f_6	F_7	f_6		
VI					f_6	F_7	f_6	
VII						f_6	F_7	
...							f_6	

Таблицы 1, 2, 3

	I	II	III	IV	V
I					
II					
III					
IV-II ₀					
V-I ₀					

(из табл. 2), в табл. 3 и сложить с числами табл. 1. Это преобразование выражается следующей процедурой.

Процедура составления матрицы A ;

начало для $i := 1, 2, \dots, n$ выполнить

{ для $k := i, i+1, \dots, n$ выполнить

$$\Sigma | \text{гр } iT_2 \times \text{гр } kT_3 | + i.kT_1 \rightarrow i.kT_A \rightarrow k.iT_A ;$$

проверка $\Sigma | \text{стр } iT_A |^2 = 0$

конец процедуры.

Правильность составления табл. A контролируется такой же проверкой, как и табл. 1, 2, 3, т. е. суммы коэффициентов в строках

табл. А должны быть равны нулю. Другой проверкой служит симметрия относительно главной диагонали.

Теперь нужно учесть граничные условия для узлов, имеющих неподвижные опоры. Уравнения для этих узлов отпадают и поэтому соответствующие строки и графы табл. А исключаются (вычеркиваются).

Таким образом, составление уравнений для разнообразных граничных условий может быть выполнено вычислителем по стандартной программе с построчным контролем.

По этому принципу была построена программа расчета изгибаемых пластин на клавишных автоматах, которая применялась в практике для расчета плит, опирающихся в отдельных точках (для таких плит отсутствуют готовые таблицы).

Указанный прием (составление обобщенной матрицы системы уравнений метода сеток, привязка граничных условий путем записи нулей в указанные строки и графы) использован Г. Б. Гильманом для составления программы расчета на ЭЦВМ прямоугольных в плане пологих оболочек с переменными кривизнами при произвольных граничных условиях и нагрузках. Поскольку эта программа предназначалась для ЭЦВМ, применялись специальные приемы экономии памяти — учет симметрии, ленточности и т. п. Эти приемы не нужны при использовании клавишных автоматов, для которых язык таблиц (матриц) действительно является лучшим.

Краткие сведения о других программах

IV. 21. Инженерные программы могут служить непосредственно для расчета сооружений на клавишных машинах, а также для численной проверки алгоритма перед составлением программы для ЭЦВМ, что особенно необходимо для новых еще не проверенных на практике методов. В последнем случае облегчается и программирование для ЭЦВМ, поскольку арифметизация задачи выполнена на стадии инженерной программы.

Ниже даны краткие сведения об инженерных программах, составленных в КиевЗНИИЭПе под руководством автора.

Программа расчета каркасных зданий с заполнением разработана инж. Л. И. Григоренко. В качестве расчетной схемы рамного каркаса принимается эквивалентная полурама.

Программа включает процедуры определения характеристик деформативности и несущей способности панелей заполнения;

жесткостных характеристик эквивалентной полурамы; перемещений расчетной схемы от единичных горизонтальных нагрузок по приближенным формулам, рекомендуемым нормами; первых трех форм и частот колебаний (IV. 18); сейсмических нагрузок; усилий в стержнях эквивалентной полурамы по программе метода сил; нагрузок, воспринимаемых панелями заполнения и непосредственно каркасом с учетом возможного разрушения панели; усилий в элементах каркаса.

Программа расчета панельных зданий на горизонтальную нагрузку разработана А. И. Козачевским и Е. Г. Гутман. Вертикальные диафрагмы пространственного блока разбиваются на две категории:

сплошные, рассчитываемые по схеме консоли, и ослабленные проемами, рассчитываемые по схеме полурамы с абсолютно жесткими узловыми вставками. В последнем случае учитываются изгибные и продольные деформации. Программа состоит из следующих процедур: расчет диафрагм и определение их жесткостных характеристик, обратных по величине прогибам от обобщенной горизонтальной нагрузки (обычно суммарная ветровая нагрузка); распределение горизонтальной нагрузки пропорционально жесткостям; определение усилий в диафрагмах.

Программа определения жесткостных характеристик крупнопанельного здания по методике Б. В. Косицына составлена А. С. Вайнбергом. После того, как жесткостные характеристики в различных сечениях панельного здания найдены, последнее рассчитывается как балка на упругом основании.

Ряд программ составлен И. Д. Гликиным для расчета нелинейных (в основном вантовых) систем. Используются предложенные Л. Г. Дмитриевым методы: многоступенчатого нагружения; вариационный; приведения нагрузки к равновесной. Инженерные программы первых двух методов применялись для численной проверки алгоритма перед разработкой программы для ЭЦВМ, программа третьего метода — непосредственно для расчета на клавишных автоматах.

И. Д. Гликиным разработана также инженерная программа составления характеристического полинома по методу Крылова и решения этого полинома по методу Лобачевского [17].

Работа, начатая автором в 1958—1960 гг. в ГПИ Укрпроект-стальконструкция, была продолжена в последующие годы В. Н. Гордеевым, Ю. С. Борисенко, М. Л. Гринбергом и др. Разрабатывались программы расчета поперечников промышленных зданий с учетом пространственной работы, антенно-мачтовых

сооружений, плоских нитяных сетей и тканевых оболочек, комбинированных систем. В Приднепровском Промстройпроекте В. Е. Манин разработал программу расчета двухветвевых стоек с учетом деформаций от продольных сил, в Сибирском отделении ПромтрансНИИпроекта А. Д. Коновалов составил программу расчета решетчатых конструкций, ряд программ составлен в Гипростроме, Ростовском промстройпроекте и др. Перечисленные программы оформлены в виде выпусков указанных институтов.

Из изложенного видно, что инженерное программирование получило в нашей стране определенное развитие и распространение. Это нашло отражение в обобщающей работе Гипротиса [73].

Дальнейшая формализация и уточнение алгоритмического языка, выполненные в настоящей работе и внесение в него элементов АЛГОЛа, существенно расширяют возможности инженерного программирования.

Запись алгоритмов для ЭЦВМ на языке инженерного программирования

IV. 22. Программы для клавишных автоматов образуют фонд проверенных алгоритмов, которые можно использовать при программировании для ЭЦВМ. Однако предварительно алгоритмы должны быть записаны более кратко, с учетом некоторых особенностей ЭЦВМ.

Эта запись отличается тем, что для экономии памяти производится (в тех случаях, когда это возможно) замещение одних чисел другими в одних и тех же элементах матриц. Например, запись:

$$\text{гр } c - a \times \text{гр } b \rightarrow \text{гр } c$$

обозначает, что из чисел графы c вычитаются умноженные на постоянный коэффициент a числа графы b и в результаты засылаются в ту же графу c . При этом числа, записанные ранее в графе c , автоматически стираются.

В инженерной программе, рассчитанной на прямоугольную форму числовых массивов, не учитываются более тонкие структурные особенности непрямоугольных матриц. Это упрощает составление и чтение программы, но она оказывается менее эффективной в отношении возможности размещения больших массивов в памяти, т. е. имеется характерный недостаток традиционной матричной записи.

Поэтому подобная запись используется нами как первая, эскизная стадия, своего рода проектное задание на программи-

рование. Последующая детальная разработка программы для ЭЦВМ выполняется на более гибком языке АЛГОЛ-60; эту стадию можно назвать стадией технического проекта. Наконец, рабочее программирование в кодах машины соответствует стадии рабочего проекта и выполняется с помощью транслятора либо ручной кодировки.

Так же, как и в проектировании, это разделение не является абсолютным и не всегда соблюдается. Так, в некоторых случаях от эскизной записи можно перейти непосредственно к программированию в кодах. В других случаях проектное задание может представлять собой одну из форм записи на АЛГОЛе, а запись в форме инженерной программы — отсутствовать.

ЭЦВМ состоит из следующих основных частей: устройства ввода исходных данных, запоминающего устройства, арифметического устройства, устройства управления, устройства ввода результатов, пульта управления.

Остановимся на особенностях работы этих устройств, проводя аналогию с клавишными машинами.

Ввод исходных данных в ЭЦВМ производится обычно с перфокарт (или перфолент), пробитых предварительно на вспомогательном устройстве — перфораторе. Каждому числу соответствует определенное сочетание отверстий в строке перфокарты. Обычно на одной перфокарте можно поместить двенадцать чисел. Вся вводимая в машину информация — программа и исходные данные — представлена в виде сложенной в определенном порядке колоды перфокарт.

Устройство ввода с помощью фотоэлемента или механических щупов воспринимает эти пробивки и преобразует их в электрические сигналы, которые поступают затем в запоминающее устройство машины. Скорость считывания в среднем колеблется для разных машин от 20 до 100 кодов (чисел) в секунду, т. е. в десятки или сотни раз быстрее ручного набора чисел на клавиатуре автомата.

Запоминающее устройство машины состоит из ряда пронумерованных ячеек, в каждой из которых помещается число или, как говорят, «слово» машинного языка. Номер ячейки называется ее адресом.

IV. 23. В виде примера рассмотрим построение инженерной программы решения систем линейных уравнений методом Гаусса с учетом экономии памяти. При составлении программы для ЭЦВМ откажемся от компактной схемы (III. 19—III. 23) и будем записывать преобразованные на каждом шаге коэффициенты на

месте нахождения исходных коэффициентов *. Преобразование свободных членов (при прямом ходе) и определение неизвестных (при обратном ходе) выполняется на месте исходного массива свободных членов.

Расположим исходную матрицу системы уравнений в табл. А, свободные члены — в отдельной графе В (схема 18). Будем считать, что в графу В поочередно вводятся комбинации свободных членов для загрузений $\nu = 1, 2, \dots, q$. Ввод свободных членов для каждого последующего загрузения выполняется программно (т. е. автоматически) после вывода результатов — значений неизвестных для предыдущего загрузения. Первый этап прямого хода (для матрицы) обозначим — прямой ход 1, второй этап прямого хода (для свободных членов) обозначим — прямой ход 2.

Схема 18
Таблица А

	1	2	...	n
1				
2				
...				
n				

B

Программа решения несимметричных систем уравнений методом исключения:

начало

Прямой ход 1:

для $i := 1$ шаг 1 до $n-1$ выполнить

для $k := i+1$ шаг 1 до n выполнить

$$(|\text{стр } k|_{i+1}^n - |\text{стр } i|_{i+1}^n \times k.i/i.i) T_A \rightarrow |\text{стр } k|_{i+1}^n T_A;$$

для $\nu := 1$ шаг 1 до q выполнить

начало ввод ν -той комбинации свободных членов в графу В;

* При расчете на клавишных машинах в качестве рабочей, накапливающей сумму произведений ячейки используется счетчик 3 (см. III. 2). При расчете на ЭЦВМ выделение операций суммирования попарных произведений с накоплением в рабочей ячейке не дает преимуществ, так как не только эти, но все вычисления выполняются автоматически; в качестве накапливающих ячеек можно использовать все ячейки матрицы, надобность во внешних рабочих ячейках отпадает.

Прямой ход 2:

для $i := 1$ шаг 1 до $n-1$ выполнить

для $k := i+1$ шаг 1 до n выполнить

$$k.B - k.iT_A \times i.B / i.iT_A \rightarrow k.B;$$

Обратный ход:

для $i := n$ шаг -1 до 1 выполнить

$$\{-i.B / i.iT_A \rightarrow i.B; \text{ если } i=1 \text{ то на } M1;$$

$$|\text{гр } B|_1^{i-1} + i.B \times |\text{гр } iT_A|_1^{i-1} \rightarrow |\text{гр } B|_1^{i-1};$$

M1: вывод графы B конец v конец программы.

Из данной программы можно легко получить программу для решения симметричных систем. Для этого достаточно заменить коэффициент $k.i$ в операторах прямого хода равным ему коэффициентом $i.k$. В этом случае в программе будут фигурировать только те элементы, которые расположены выше главной диагонали. В дальнейшем при составлении программы на АЛГОЛе мы используем это обстоятельство для того, чтобы для симметричных матриц не записывать в память элементы, расположенные ниже главной диагонали.

ОБЩИЕ ВОПРОСЫ ПРИМЕНЕНИЯ АЛГОРИТМИЧЕСКОГО
ЯЗЫКА АЛГОЛ-60

V. 1. Алгоритмический язык для ЭЦВМ сходен с языком инженерного программирования (гл. III и IV), так как строй языка, его идейные основы по существу остаются теми же. Изменяется форма операторов, ориентированная на автоматизацию программирования для ЭЦВМ, язык получает более развитую, гибкую структуру, первостепенное значение приобретает формализм в синтаксических описаниях. Если понимать суть языка, то эти усложняющие элементы можно усвоить достаточно быстро.

Программа на языке АЛГОЛ-60 делится на две части — описательную и исполнительную. Исполнительная часть состоит из операторов, разделенных точками с запятыми. Сущность операторов та же, что и в инженерном программировании — это приказы о выполнении определенных действий, но обозначения другие. Что касается описательной части, то необходимость в ней вызвана большей формализацией языка. В примитивной форме мы сталкивались с описаниями уже в инженерном программировании — это были расчетные бланки пронумерованных таблиц с заданными границами, а также параметры задачи. В языке АЛГОЛ подобные сведения указываются в разделе описаний: перечисление обозначений массивов с указанием их границ, обозначений переменных, описания процедур. Новым является описание типов массивов и переменных.

В соответствии с описаниями распределяется память машины для размещения числового материала. Поэтому ошибки в описательной части также недопустимы, как и в исполнительной.

АЛГОЛ-60 имеет три различных уровня представления:

- а) эталонный, или определяющий язык, являющийся образцом для всех конкретных представлений, основой и руководством для создания программирующих программ;
- б) язык публикаций, допускающий изменения эталонного языка, связанные с удобством печати или написания, применяемый для взаимного обмена информацией;

в) конкретные представления языка, каждое из которых использует набор знаков конкретной машины, является входным языком программирующей программы для этой машины и сопровождается правилами для перевода с языка публикаций или эталонного языка.

В последние годы опубликован ряд общематематических работ и учебных пособий по АЛГОЛу [1, 5, 18, 58, 65]. Предыдущая книга автора [109] целиком посвящена применению языка АЛГОЛ-60 в строительной механике: излагаются основная и расширенная формы языка (обе они универсальны, но первая более проста и ближе к машинному языку), опыт автоматизации программирования с помощью транслятора ТА-2 и алгоритм ручного кодирования алгольной программы; даны комментарии к официальному «Сообщению об алгоритмическом языке» [2], разъясняющие суть несложного языка описаний (так называемого мета-языка), на котором написано «Сообщение». После ознакомления с основами языка (в неформальном изложении) и с указанными комментариями инженеры в дальнейшем самостоятельно пользуются «Сообщением» как очень удобным и, в конечном счете, простым (если понимать мета-язык) справочным, нормативным пособием, в котором благодаря делению текста на разделы и подразделы и наличию алфавитного указателя можно быстро найти исчерпывающий ответ на любой вопрос, возникающий при программировании. С этой точки зрения никакая даже самая лучшая популяризация не может заменить текст «Сообщения».

В данной главе рассмотрен ряд дополнительных материалов, развивающих работу [109].

Программы на языке АЛГОЛ-60 абстрагируются от конкретной машины, для их составления не обязательно быть специалистом по программированию в кодах, достаточно иметь лишь общее представление об ЭЦВМ, принципах их действия, типах машин и их возможностях. Эти сведения кратко даны в начале главы *. Подробное же описание ЭЦВМ и программирования в кодах можно найти в известной учебной литературе [19, 31, 46, 47, 60, 63, 64].

* Излагаемые в V.2—V.5 сведения обладают даже некоторой (необходимой) избыточностью в смысле эффективного программирования на языке АЛГОЛ-60. Изучение материалов для конкретной машины (система команд, инструкции по их использованию и т. п.) требуется лишь на заключительной стадии при оформлении программы на входном языке транслятора либо при ручном кодировании алгольной программы.

Краткие сведения об электронных цифровых вычислительных машинах

V. 2. Отличительными особенностями ЭЦВМ являются автоматическое управление всем процессом вычислений, способность запоминания массивов чисел, огромная скорость выполнения арифметических и логических операций.

Серийные отечественные машины среднего класса имеют память, состоящую из 2048—4096 ячеек. Существуют также машины со значительно большим объемом памяти. Например, машина «Раздан-3» имеет память до 32768 ячеек. Наряду с этим применяются малогабаритные машины с небольшим объемом памяти от 200 до 1024 ячеек, обычно для массовых небольших расчетов.

Ячейка памяти машины имеет то свойство, что при засылке (записи) в нее нового числа предыдущее число, хранившееся раньше, стирается. Если же из ячейки берется (считывается) число, то содержимое ее остается без изменений.

Кроме быстродействующей оперативной памяти во многих машинах имеется еще внешняя, чаще всего магнитная память на барабанах и лентах, по принципу устройства сходная с магнитофоном. Внешняя память обладает большой емкостью (десятки и сотни тысяч кодов), но меньшей скоростью выборки и записи чисел.

Некоторые сведения о ряде отечественных ЭЦВМ даны в табл. V. 1.

Внешняя память не связана непосредственно с арифметическим устройством машины. Для того, чтобы выполнить какое-либо действие над числами, хранящимися во внешней памяти, нужно их предварительно переписать во внутреннюю память, что замедляет время счета.

Можно заметить здесь некоторое сходство с памятью человека, которая также имеет несколько уровней. Определенные, ограниченные сведения мы всегда помним и легко извлекаем из нашей активной памяти. Другие сведения находятся глубже, в пассивной части памяти, и чтобы вспомнить, нужно сделать определенное усилие. Нередко обращаемся к внешней памяти — записной книжке или справочнику, которые помогают вспомнить забытое.

Арифметическое устройство ЭЦВМ способно выполнять арифметические и простейшие логические операции. Числа (коды), с которыми оперирует арифметическое устройство, выбираются из запоминающего устройства. Арифметическое устройство ЭЦВМ

в принципе аналогично арифметическому устройству клавишного автомата (речь идет только об арифметических операциях, логические операции на клавишном автомате не предусмотрены), но благодаря применению электронных схем обладает большим быстродействием — тысячи и десятки тысяч операций в секунду, а наиболее совершенные модели — порядка миллиона вычислений в секунду (см. табл. V. I).

Заметим, что устройство ввода, запоминающее и арифметическое устройства имеют, правда, в простейшем виде, в предшественниках ЭЦВМ — в электрических и механических арифмометрах.

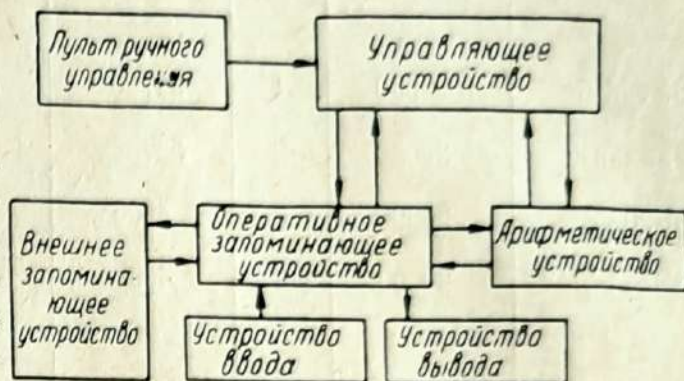


Рис. 29.

Качественно новым является устройство автоматического управления, по сигналам которого выполняются команды программы, в закодированном виде хранящейся в оперативном запоминающем устройстве. Таким образом, устройство управления заменяет человека — оператора вычислительной машины и процесс выполнения алгоритма полностью автоматизируется.

Программа представляет собой совокупность команд, выполняемых в порядке их следования. Команды — это закодированные приказы о том, откуда, из каких ячеек брать исходные числа, какие действия над ними выполнять, в какие ячейки помещать результаты. Помимо команд вычислительного характера, имеются команды управляющие, — реализующие обращение к внешним устройствам (ввода, вывода, внешней памяти), и переходы (условный или безусловный) к заданным командам, если требуется изменить естественный (по порядку следования) ход выполнения команд.

Вывод результатов осуществляется на печать с помощью печатающего устройства или на перфокарты с помощью перфоратора. Хотя работа машины полностью автоматизирована и происходит без участия человека, но на каждой машине имеется пульт ручного управления, позволяющий человеку в случае необходимости вмешиваться в ход вычислительного процесса. Принципиальная схема действия ЭЦВМ изображена на рис. 29.

ЭЦВМ	Быстродей- ствие (опера- ций в сек)	Адрес- ность	Емкость опе- ративного запоминаю- щего устрой- ства	Емкость внешних запомина- ющих устройств		Коли- чество двоич- ных разря- дов
				Магнитных барабанов	Магнитных лент	
БЭСМ-6	1000000	1	32768	131072	32000000	48
Раздан-3*	20000	2	32768	—	6240000	48
М-20	20000	3	4096	12288	300000	45
БЭСМ-3М*	20000	3	4096	32768	2400000	45
БЭСМ-4*	20000	3	8192	65536	4000000— 8000000	45
М-220	25000	3	4096—16384	65536	16000000	45
БЭСМ-2М	10000	3	2048	12288	200000	39
Урал-2	5000—6000	1	2048	65535	100000	40
Урал-4	5000—6000	1	2048	131072	3145728	40
Урал 11*	25000—50000	1	4096—16384	98000— 399000	1000000— 8000000	12—24
Минск-2*	5000—6000	2	4096	—	400000	37
Минск-22*	5000—6000	2	8192	—	1600000	37

* Полупроводниковая машина.

** АЦПУ — алфавитно-цифровое печатающее устройство.

Таблица V.1

Форма представленных чисел		Скорость ввода (кодов в 1 мин)	Наличие ввода и вывода на перфокартах	Скорость обмена с внешним запоминающим устройством	Примечание
с плавающей запятой	с фиксированной запятой				
Да	Нет	8400	Есть	С магнитными лентами 8000 код/сек, с магнитным барабаном 25600 код/сек	Имеет АЦПУ**
"	"	3300	"	С магнитными лентами 3500 код/сек, с магнитным барабаном 6500 код/сек	"
"	"	1200	"	С магнитными лентами 3000 код/сек, с магнитным барабаном 5000 код/сек	"
"	"	1200	"	С магнитными лентами 3000 код/сек, с магнитным барабаном 6400 код/сек	Система команд та же, что и у М-20
"	"	8400	"	С магнитными лентами 5000 код/сек, с магнитным барабаном 12000 код/сек	Имеет АЦПУ
"	"	8400	"	С магнитными лентами 5000 код/сек, с магнитным барабаном 50000 код/сек	"
"	"	1200	"	С магнитными лентами 400 код/сек, с магнитным барабаном 800 код/сек	"
"	Да	9000	Нет	С магнитными лентами 2000 код/сек, с магнитным барабаном 5000 код/сек	"
"	"	3600	Есть	С магнитными лентами 2000 код/сек, с магнитным барабаном 4500 код/сек	Имеет АЦПУ
Нет	"	3600	"		Одновременно можно подключить до 16 устройств для ввода и вывода информации
Да	Да	3000	Нет	С магнитной лентой 400 код/сек	
"	"	3000	Есть	С магнитной лентой 2500 код/сек	Имеет устройство ввода и вывода алфавитно-цифровой информации с пишущей машинки

Управляющее устройство (сокращенно УУ) в соответствии с программой руководит работой машины и производит обмен между всеми устройствами машины: ввод чисел с перфокарт в оперативное запоминающее устройство (ОЗУ), пересылка чисел из ОЗУ в арифметическое устройство (АУ) и переработка их, пересылка результатов снова в ОЗУ и, если нужно, дальнейшая пересылка во внешние запоминающие устройства, считывание чисел в ОЗУ с внешних запоминающих устройств, вывод результатов, останов машины *. Между УУ, АУ и ОЗУ имеются не только прямые, но и обратные связи. Это позволяет в соответствии с программой анализировать результаты и менять по сигналам УУ ход вычислительного процесса.

V. 3. Вся циркулирующая в машине информация имеет вид чисел: при вводе и выводе — в десятичной или восьмеричной системе исчисления, при переработке в самой машине — в двоичной системе (или, как говорят, в двоичном коде).

Хотя в алгоритмическом языке фигурируют только десятичные числа, но нередко приходится применять некоторые процедуры в машинном коде и использовать восьмеричные, четверичные, двоичные числа.

В машинах с фиксированной запятой положение запятой точно указано. Например, в некоторых машинах («Урал-1», «Киев» и др.) запятая фиксируется после нуля. Это значит, что все числа, вводимые в машину и получаемые в результате вычислений, должны быть меньше единицы, т. е. вида $0, a_1 a_2, a_3, \dots$, где $a_1, a_2, a_3, \dots$ — цифры, образующие число.

В противном случае произойдет переполнение разрядной сетки и так называемый Авост (аварийный останов машины). Поэтому все числа вводятся в машину с определенным масштабом с тем, чтобы не произошло переполнения разрядной сетки.

Необходимость в масштабировании во многих случаях усложняет подготовку задачи к счету, сильно ограничивает возможности машины. Поэтому большинство современных машин имеет режим плавающей запятой, при котором положение запятой в процессе вычислений изменяется (плавает). В связи с тем, что при фиксированной запятой арифметические операции моделируются в машине проще (как следствие увеличивается быстродействие и уменьшается время счета), ряд машин имеет оба режима — с фиксированной и плавающей запятой (табл. V. 1).

* В случае необходимости те же действия можно выполнять вручную с пульта. Подобное вмешательство обычно бывает при отладке программы (проверочном расчете).

меричная цифра. Например, запись одной и той же команды на бланке и в машине выглядит так:

0	1	1	4	1	0	0	1	1	2	3	7	0	0
000	001	001	100	001	000	000	001	001	010	011	111	000	000

01 — код операции сложения, заменяющий знак +. Смысл данной команды — содержимое ячейки 1410 сложить с содержимым ячейки 0112 и результат поместить в ячейку 3700. Аналогично для вычитания код операций 02, деления 04, умножения 05 и т. д.

Для составления программы в кодах нужно знать систему команд данной машины. Строение командной ячейки для М-20 приведено лишь для иллюстрации. Подробные сведения о системе команд и программировании для конкретных машин содержатся в работах [63, 64] и др.

В одноадресной машине командная ячейка имеет следующий вид:

КОП	Адрес
-----	-------

В адресной части записан адрес одной ячейки, содержимое которой участвует в операции: в качестве промежуточного звена, заменяющего другие адреса, используется сумматор.

Например, команда трехадресной машины

+	<i>a</i>	<i>b</i>	<i>c</i>
---	----------	----------	----------

реализуется на одноадресной машине следующим образом:

Номер ячеек	КОП	Адрес
$N + 1$	$OЗУ \leftarrow S$	<i>a</i>
2	+	<i>b</i>
3	$S \rightarrow OЗУ$	<i>c</i>

Приняты обозначения [93]:

$OЗУ \rightarrow S$ — вызов на сумматор из оперативного запоминающего устройства;

$S \rightarrow OЗУ$ — посылка из сумматора в оперативное запоминающее устройство.

По команде $N+1$ на сумматор вызывается содержимое ячейки a , по команде $N+2$ к содержимому сумматора прибавляется содержимое ячейки b , по команде $N+3$ число из сумматора посылается в ячейку c . Таким образом, арифметические действия ($+$, $-$, $\times$) реализуются для одноадресной машины с помощью трех команд, вместо одной для трехадресной машины. Простейший оператор засылки ($a \rightarrow b$) для одноадресной машины занимает две ячейки (команды $N+1$ и $N+2$ в приведенной выше записи) вместо одной для трехадресной машины.

Вследствие этого длина программы для одноадресных машин существенно увеличивается. Однако при большом объеме ОЗУ (несколько десятков или сотен тысяч ячеек) длина одного адреса может получиться столь значительной, что для его размещения нужна будет целая ячейка, если считать, что размер ячейки определяется из условия размещения нормализованного числа. Это одна из причин того, что наиболее мощные машины являются одноадресными*.

Промежуточное положение занимают машины двухадресные, для которых ячейка имеет следующий вид:

КОП	IA	IIA
-----	----	-----

В первом и втором адресе указываются номера ячеек, участвующих в операции. Результат вычисления, предписываемого кодом операции, засылается по второму адресу. При этом предыдущее содержимое второго адреса вытесняется. Если вытесняемое число для дальнейшего расчета не нужно, то больше команд не потребуется; в противном случае, данной команде должна предшествовать команда пересылки числа из второго адреса в некоторую запасную (рабочую) ячейку.

Длина программы для двухадресной машины обычно получается не на много больше, чем для трехадресной, так как вытеснение промежуточных результатов весьма широко применяется в программировании.

V. 5. В практике проектирования наряду со средними и большими машинами все большее применение для массовых расче-

* Имеются, впрочем, сведения о мощных трехадресных машинах (фирма «Контрол дейта корп», США), использующих два вида оперативной памяти: небольшого и большого объема. Первый и третий адреса сделаны короткими и рассчитаны на первый вид памяти, второй — длинный адрес рассчитан на второй вид памяти.

тов находят малогабаритные машины типа «Проминь», «Мир», «Наири» (табл. V. 1). Например, машина «Наири» является двух-адресной машиной с весьма рациональной системой команд. Программа может быть изготовлена вручную или с помощью автокода, правда, не вполне совершенного. Имеется ДЗУ (долговременное запоминающее устройство) для размещения стандартных программ (тогда они не занимают место в памяти). Числовой материал можно размещать только в оперативной памяти, так как обмен между ОЗУ и ДЗУ отсутствует.

Характеристики машин «Наири» являются в общем удовлетворительными для решения распространенных инженерных задач не очень большого объема (расчеты элементов конструкций, статически неопределимых систем, например, рам не выше 6—7 этажей). Препятствием для высокой производительности машины служит другое — медленная работа устройств ввода и вывода (особенно ввода), которые необходимо заменить на более совершенные.

Достоинством машины «Мир» является развитое автоматическое программирование на основе АЛГОЛа; объем памяти 4096 символов, имеется возможность производить расчет с увеличенной точностью.

Операторы процедур ввода-вывода на алгоритмическом языке

V. 6. Операция ввода-вывода числового материала в ОЗУ является необходимой составной частью любой программы, независимо от ее структуры. В издание [2] включено помимо официального описания языка АЛГОЛ-60 также сообщение о процедурах ввода-вывода, принятое в 1964 г. рабочей группой при Международной федерации по обработке информации (IFIP/WG 2.1).

Другая система процедур обмена предложена М. Р. Шура-Бура и Э. З. Любимским для входного языка транслятора ТА-2 [131] и в отвлеченном виде (с буквенными обозначениями внешних носителей) используется в работе [109].

Целью последующих глав настоящей книги является изложение алгоритмов строительной механики, предназначенных для ЭЦВМ, но в обобщенном виде без детализации тех процедур, которые связаны с оборудованием определенной машины и конкретным представлением языка. Поэтому для операторов процедур ввода-вывода будем применять отвлеченные обозначения,

достаточные для общего описания процесса обмена между внешними носителями и оперативной памятью, а именно:

input ($a, b, \dots, c$) — ввод чисел или логических значений извне в ячейки оперативной памяти машины, соответствующие параметрам $a, b, \dots, c$. В качестве параметров могут быть переменные (простые или с индексами) идентификаторы массивов того типа (**integer, real, boolean**), который указан в разделе описаний;

output ($a, b, \dots, c$) — то же для вывода чисел из оперативной памяти;

read ($a, b, \dots, c$) — чтение чисел из внешней памяти, т. е. пересылка чисел с магнитных барабанов, лент и т. п. в ячейки оперативной памяти, соответствующие параметрам $a, b, \dots, c$;

write ($a, b, \dots, c$) — запись чисел во внешнюю память, т. е. пересылка чисел из ячеек оперативной памяти, соответствующих параметрам $a, b, \dots, c$, на магнитные барабаны, ленты и т. д.

Операторы процедур *input-output* определяют только, что данная информация вводится в машину или выводится из нее, но не указывают способа передачи информации (перфокарты, перфоленты, печать на бумажной ленте и т. д.). Точно также операторы процедур *read-write* определяют только то, что данная информация пересылается из внешней памяти в оперативную и обратно, но не указывают конкретно ни типа запоминающего устройства, ни адресов хранящихся в них чисел. Эти детали разрабатываются на стадии рабочего программирования — при уточнении записи алгоритма на входном языке транслятора для данной машины либо при ручной кодировке алгоритма.

Пример построения программы

V. 7. Процесс построения программы на алгоритмическом языке можно разбить на ряд последовательных этапов. Проиллюстрируем этот процесс на простом примере составления процедуры вычисления ординат эпюры моментов и поперечных сил в балке на двух опорах от произвольной нагрузки.

1. Составление задания. Начнем с того, что запишем формулы для определения опорных реакций A и B , моментов и поперечных сил от отдельных видов нагрузок.

Для распределенной нагрузки (рис. 30, а):

$$B = \frac{q_1 b}{l} \left(a + \frac{b}{2} \right) + \frac{b(q_2 - q_1)}{2l} \left(a + \frac{2}{3} b \right); \quad (1)$$

$$A = \frac{(q_1 + q_2)b}{2} - B; \quad (2)$$

$$M_2 = M_1 + Q_1 \times b - \frac{b^2}{6} (2q_1 + q_2); \quad (3)$$

$$Q_2 = Q_1 - \frac{(q_1 + q_2)b}{2}. \quad (4)$$

Здесь момент M_2 и поперечная сила Q_2 в конце участка выражены через момент M_1 , поперечную силу Q_1 в начале участка и нагрузку q_1, q_2 на участке.

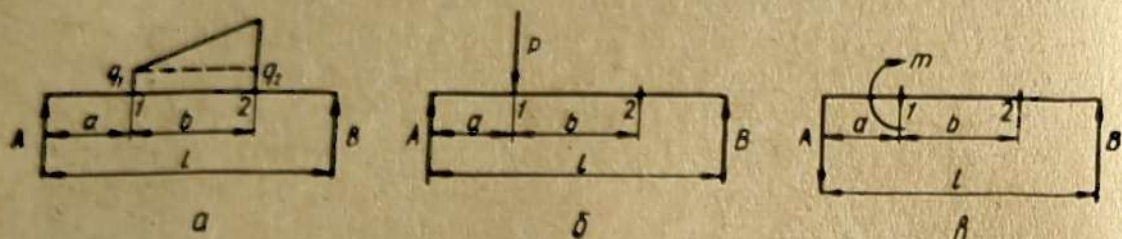


Рис. 30.

Для равномерно распределенной нагрузки следует принять $q_1 = q_2$ и подставить в формулы (1) — (4):

$$B = \frac{q_1 b}{l} \left(a + \frac{b}{2} \right); \quad (5)$$

$$A = q_1 \times b - B; \quad (6)$$

$$M_2 = M_1 + Q_1 \times b - \frac{q_1 \times b^2}{2}; \quad (7)$$

$$Q_2 = Q_1 - q_1 \times b. \quad (8)$$

Для сосредоточенной нагрузки (рис. 30, б) *:

$$B = \frac{Pa}{l}; \quad A = P - B; \quad (9)$$

$$M_2 = M_1 + Q_1 b; \quad Q_2 = Q_1 - P. \quad (10)$$

Для сосредоточенных моментов (рис. 30, в) *:

$$B = -A = \frac{m}{l}; \quad (11)$$

$$M_2 = M_1 + Q_1 b + m; \quad Q_2 = Q_1. \quad (12)$$

* Балка разбивается на участки так, чтобы в пределах участка сосредоточенных воздействий не было.

Правило знаков общепринятое — опорные реакции считаются положительными, когда они направлены вверх; изгибающие моменты положительны, когда они изгибают балку выпуклостью вниз; поперечные силы, — когда они сдвигают левую часть балки вверх относительно правой. Вертикальная нагрузка положительна, когда она направлена вниз; сосредоточенные моменты положительны, когда они действуют по часовой стрелке.

Составить программу для расчета балки на отдельную нагрузку нетрудно, но такая программа будет слишком частной и не будет иметь ценности. Поэтому ставится более общая задача расчета балки на произвольное число указанных элементарных нагрузок при любом их сочетании*.

Эту задачу трудно представить в замкнутом формульном виде, так как заранее неизвестно количество членов в формулах, да и сами члены не определены. Обычно в таких случаях в формулах ставят многоточие, но это лишает их определенности.

Такую задачу, однако, нетрудно разрешить алгоритмически, разработав циклический процесс вычислений, управляемый заданными параметрами.

Формула также представляет собой алгоритм (правило последовательных вычислений), но всегда выполняемый одинаково. Формулу можно назвать безусловным алгоритмом.

Алгоритм в широком смысле включает в себя не только вычислительные формулы, но и проверки логических условий, изменяющих ход вычислительного процесса и определяющих его протяженность. Это и дает возможность создать общий алгоритм для всех задач данного класса, в котором объединяется множество алгоритмов, причем реализуется тот из них, который соответствует заданным значениям входных параметров. Алгоритм, как известно, уподобляют машине со входом, характер работы которой определяется значениями входных параметров.

Итак можно сформулировать задание: с помощью формул (1) — (12) составить алгоритм (программу) определения опорных реакций и ординат эпюр моментов и поперечных сил в балке от произвольного числа нагрузок различных видов.

Заметим, что при построении формул для отдельных нагрузок эта цель имела в виду — структура формул такова, что исходными для определения усилий в данном сечении служат усилия.

* Если нагрузка распределена по криволинейному закону (это бывает редко), то ее можно заменить с небольшой погрешностью многоугольником, так что в пределах одного участка нагрузка будет ограничена прямой линией.

найденные ранее в предыдущем сечении, т. е. на каждом шагу расчета достаточно помнить только то, что находится на одном участке балки.

2. Определяем входные параметры задачи. Разделим балку на участки таким образом, чтобы в пределах одного участка отсутствовали сосредоточенные воздействия или скачки в распределенной нагрузке. Например, для рис. 31 получаем участки 1—2, 2—3, 3—4, 4—5.

В качестве входных параметров примем величину N — число участков балки и признаки — переменные $c1, c2, c3, c4$, информирующие о типе нагрузки.

$c1$ — признак равномерно распределенной нагрузки;

$c2$ — признак нагрузки, распределенной по линейному закону;

$c3$ — признак сосредоточенной нагрузки;

$c4$ — признак сосредоточенных моментов.

Условимся, что значение признака равно 1, если нагрузка данного типа имеется, и 0, если ее нет.

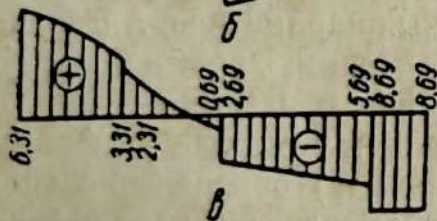
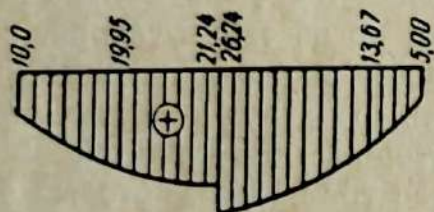
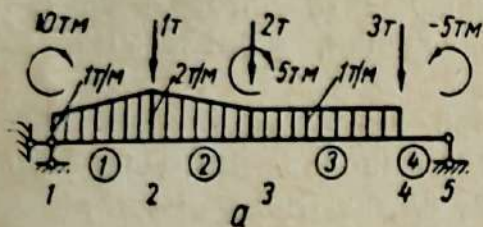


Рис. 31.

Введем еще один параметр μ (заменяет греческую букву μ): если его значение равно 1, то результаты выводятся на печать, в противном случае — используются в другой части программы, в которую входит данная процедура. Величину μ будем рассматривать как формальный параметр процедуры.

3. Определение исходных данных и их размещение. Для данной задачи исходными данными являются длины участков и значения нагрузок, приложенных в точках, разделяющих участки (для сосредоточенных воздействий), или в пределах участков (для распределенных воздействий).

Количество этих величин зависит от одного параметра N , поэтому их следует располагать в линейных массивах:

$L[1 : N]$ — для длин участков;

$g1[1 : N]$ — для равномерно распределенных нагрузок (для каждого участка своя интенсивность);

$g2[1 : 2 \times N]$ — для нагрузок, распределенных по линейному

закону (для каждого участка по две интенсивности);

$g3[1:N]$ — для сосредоточенных сил;

$g4[1:N]$ — для сосредоточенных изгибающих моментов.

Сосредоточенные силы или моменты приложены в точках 2, 3, ..., N , разделяющих участки (например, на рис. 31 — в точках 2, 3, 4). Первые элементы массивов $g3$ и $g4$ нулевые (эти элементы добавлены для того, чтобы параметр цикла для всех четырех массивов изменялся одинаково — от 1 до N).

Если нагрузка данного типа на балке имеется (соответствующий признак равен 1), то вводятся значения нагрузок для всех участков, в том числе нулевые значения*. К исходным данным относятся также значения моментов на концах балки MA и MB , которые вводятся извне или могут быть получены по другой программе (например, из расчета статически неопределимой системы).

4. Назначение переменных для промежуточных и конечных результатов. В машине этим переменным соответствуют рабочие ячейки, в которые помещаются текущие значения величин.

Для данной задачи назначаем следующие переменные:

A — реакция левой опоры;

B — реакция правой опоры;

a — расстояние от начала участка до левой опоры;

b — длина участка;

l — длина балки;

$Q1, Q2$ — поперечные силы в начале и конце участка;

$M1, M2$ — моменты в начале и конце участка.

5. Назначение индексов и переменных с индексами. Порядковый номер участка (и точки, разделяющей участки) назовем j . Порядковые номера (индексы) сечений в начале и конце j -го участка:

$$j1 = 2 \times (j-1) + 1 \text{ и } j1 + 1.$$

В соответствии с этим получим следующие значения переменных с индексами.

$L[j]$ — длина j -го участка;

$g1[j]$ — интенсивность равномерно распределенной нагрузки для j -го участка;

* Можно дать дифференцированные признаки для участков балки, помещая 1 или 0 в разряды ячейки (шкалы), если на соответствующем участке имеется или отсутствует нагрузка. При такой информации используется процедура — код.

$g2[j1], g2[j1+1]$ — интенсивности, распределенные по линейному закону нагрузки в начале и конце участка;

$g3[j]$ — сосредоточенная сила в j -той точке;

$g4[j]$ — сосредоточенный момент в j -той точке.

Индексам j и $j+1$ в программе соответствуют счетчики.

6. Вычислительные операторы легко получить из формул (1) — (12), приняв для распределенной по линейному закону нагрузки: $q_1 = g2[j1]$; $q_2 = g2[j1+1]$; для равномерно распределенной нагрузки: $q = g1[j]$; для сосредоточенной нагрузки: $P = g3[j]$, для сосредоточенных моментов: $m = g4[j]$.

Величине b присваивается значение $b := L[j]$.

Для организации цикла по j оператор присваивания записывается так, чтобы в левой и правой его частях стояла одна и та же переменная ($M2, Q2$), т. е. происходило накопление ее значений от всех нагрузок. Например, формулы (5) — (8) для равномерно распределенной нагрузки преобразуются в операторы следующим образом:

$$r := q1[j] \times (a + b/2) \times b/l;$$

$$B := B + r; A := A + g1[j] \times b - r;$$

$$M2 := M1 + Q1 \times b;$$

$$M2 := M2 - g1[j] \times b \uparrow 2/2;$$

$$Q2 := Q1;$$

$$Q2 := Q2 - g1[j] \times b.$$

7. Составляется логическая схема программы. Организуются два цикла с параметром j : один — для определения реакций A и B от нагрузок различных типов в соответствии с признаками $c1, c2, c3, c4$, второй — для определения усилий M и Q (см. процедуру в V. 9). В результате расчета на печать выдаются величины $A, B, M1, M2, Q1, Q2$.

8. Выполняется отладка логической схемы счета на небольшом числовом примере. Отладка является обязательной, так как только с ее помощью можно выявить ошибки, почти неизбежные при первой записи алгоритма. Для отладки использован пример, изображенный на рис. 31, а, номера участков указаны в кружках. Отладку удобно выполнять в табличной форме, как указано в работе [109].

Исходные данные — массивы $g1, g2, g3, g4$ и L , составленные по геометрическим размерам и нагрузкам, представим в следующем виде:

<i>j</i>	1	2	3	4				
<i>g1</i>	—	—	1,0	—				
	1	2	3	4	5	6	7	8
<i>g2</i>	1,0	2,0	2,0	1,0	—	—	—	—
<i>g3</i>	—	1,0	2,0	3,0				
<i>g4</i>	—	—	5,0	—				
<i>L</i>	2,0	2,0	3,0	1,0				

В данном примере имеются нагрузки всех четырех типов, поэтому $c1=c2=c3=c4=1$.

Для контроля на рис. 31, б и в даны эпюры моментов и поперечных сил, построенные независимо от программы.

9. Процедура оформляется в виде блока, добавляются описания и операторы ввода-вывода.

10. Производится тщательная проверка правильности описательной части (желательно другим лицом).

V. 8. **procedure** расчет шарнирно опертой балки на произвольную нагрузку (*mu*); **value** *mu*; **integer** *mu*; **begin** **integer** *j*, *j1*; **real** *r*, *a*, *b*, *l*; **comment**

во внешнем (охватывающем процедуру) блоке даны описания переменных: **integer** *N*, *c1*, *c2*, *c3*, *c4* (эти величины вводятся как исходные данные), **real** *MA*, *MB* (вычисляются или вводятся), *A*, *B*, *M1*, *M2*, *Q1*, *Q2*, описания массивов: **array** *L*, *g1*, *g3*, *g4* [*1* : *N*], *g2* [*1* : $2 \times N$] (вычисляются или вводятся). Если данная процедура входит в состав программы расчета статически неопределимой системы, то массивы *L*, *g1*, *g2*, *g3*, *g4* и переменные *MA*, *MB* формируются по специальной процедуре в соответствии с информацией о геометрии и нагрузках для системы;

l := 0; **for** *j* := 1 **step** 1 **until** *N* **do** *l* := *l* + *L* [*j*];

a := 0; *B* := (*MA* + *MB*) / *l*; *A* := -*B*;


```

for j := 1 step 1 until N do begin b := L[j];
if c1 = 1 then begin r := g1[j] × (a + b/2) × b/l;
B := B + r; A := A + g1[j] × b - r end; if c2 = 1
then begin j1 := 2 × (j - 1) + 1; r := b/l × (g2[j1] × (a + b/2) +
(g2[j1 + 1] - g2[j1]) × (a/2 + b/3));
B := B + r; A := A + (g2[j1] + g2[j1 + 1]) × b/2 - r end;
if c3 = 1 then begin r := g3[j] × a/l; B := B + r;
A := A + g3[j] - r end; if c4 = 1 then begin r := g4[j]/l;
B := B + r; A := A - r end; a := a + b end j и вычисления реакций;
Q1 := A; M1 := MA; M2 := 0;
if mu = 1 then output (A, B);
for j = 1 step 1 until N do begin
b := L[j]; M2 := M1 + Q1 × b; Q2 := Q1;
if c1 = 1 then begin M2 := M2 - g1[j] × b ↑ 2/2;
Q2 := Q2 - g1[j] × b end;
if c2 = 1 then begin j1 := 2 × (j - 1) + 1;
M2 := M2 - b ↑ 2 × (2 × g2[j1] + g2[j1 + 1])/6;
Q2 := Q2 - (g2[j1] + g2[j1 + 1]) × b/2 end;
if mu = 1 then output (M1, M2, Q1, Q2);
M1 := M2; Q1 := Q2;
if j = N then go to k1;
if c3 = 1 then Q1 := Q1 - g3[j + 1];
if c4 = 1 then M1 := M1 + g4[j + 1];
k1: end j и определения M, Q. По мере вычисления печатаются
моменты и поперечные силы в начале и конце каждого участка
end процедуры.

```

Данная процедура описывает простую задачу, но имеет развитую логическую схему, которая дает возможность произвести расчет как на нагрузку определенного типа, так и на произвольную комбинацию различных типов нагрузок.

Правила составления программ на языке АЛГОЛ-60

V.9. В V.7 на примере было показано, что процесс составления программы можно разбить на последовательные этапы и подчинить определенным правилам.

Подобные правила, сформулированные даже в самом общем виде, могут оказаться весьма полезными. Можно будет действовать по определенному плану и для программирования отдель-

ных вычислительных этапов применять испытанные приемы, а не разрабатывать их каждый раз заново и не всегда в лучшем варианте. Останется больше времени для продумывания индивидуальных особенностей решаемой задачи, которых обычно бывает не так уж много. Даже для лиц, имеющих определенный опыт в программировании, такая направляющая нить облегчит процесс разработки; еще более она будет полезна для инженеров, только недавно освоивших алгоритмический язык.

По мере усложнения задач развивается и система правил. Для строительной механики эти правила сводятся в основном к следующему:

1. Изучается класс программируемых задач и определяются основные параметры, характеризующие этот класс. Для разветвляющихся процессов назначаются признаки, в зависимости от значения которых вычисления выполняются по одним или другим формулам (например, признаки нагрузки в V.7). Одновременно записываются условия для признаков.

Расчетные формулы выписываются в той последовательности, в которой их следует выполнять. Полезно составить инженерную программу и проверить алгоритм с помощью численного расчета на клавишных автоматах. Это первая эскизная запись алгоритма.

2. Продумывается расположение информации, исходной и перерабатываемой. Назначаются идентификаторы переменных и массивов. Если алгоритм имеет большой объем, то его желательно разбить на отдельные части, более или менее изолированные в виде блоков или процедур. Локализованные величины записываются в разделе описаний соответствующих блоков, глобальные величины — в разделе описаний программы. Это первый набросок описаний будущей программы. В дальнейшем он будет неоднократно корректироваться. Размеры массивов пока не определены. Переменные разделяются на входные величины (параметры, признаки и исходные данные), промежуточные величины и результаты.

3. Изучается структура принятых массивов — матриц. Выясняется, какие элементы матриц постоянно остаются нулевыми. Желательно характер структуры массивов описать с помощью небольшого количества параметров, позволяющих автоматически исключать нулевые элементы.

4. Решается вопрос о том, в каком виде представлять массивы — в виде двухмерных (или даже многомерных) матриц или в виде линейных последовательностей ячеек. Как известно, память машины является линейной (ячейки идут в последовательности

1, 2, 3, ..., S), поэтому формулы АЛГОЛа для элементов многомерных массивов должны быть переведены транслятором (или вручную программистом) для машины так, чтобы переменная с индексами определяла элемент развернутого линейного массива. Если исходить из прямоугольных матриц, то алгоритм перевода может быть реализован транслятором. В этом случае легче найти закономерности, и программа на АЛГОЛе записывается наиболее просто. Казалось бы можно принять матричное задание информации. Но здесь имеется одна неприятная деталь — приходится нумеровать все элементы двумерного массива и, следовательно, резервировать для них ячейки памяти. А для матриц строительной механики как раз характерна малая заполненность, симметрия, расположение коэффициентов в виде диагональной полосы (ленты) и т. д. Очень нежелательно для нулевых элементов (по существу паразитических) отводить массивы памяти.

При ограниченном объеме оперативных запоминающих устройств это может явиться серьезным препятствием для реализации программы. При линейном представлении массивов проще учесть структурные особенности получаемых матриц для исключения нулевых элементов.

Таким образом, требуется анализ этого вопроса и выбор, в зависимости от возможного объема и особенностей задачи, оптимального решения. После этого определяются граничные пары в описаниях массивов.

5. Выявляются возможности совмещения идентификаторов, обозначающих разные переменные (или совокупности переменных с индексами в массивах), но одного типа. Например, если массив A преобразуется в массив B и содержимое массива A дальше нигде не используется (область действия переменных массива A не выходит за пределы B), то идентификаторы A и B можно совместить в один идентификатор A , имеющий размеры большего из массивов; граница у обоих массивов должна быть одинакова. При этом преобразование массива A выполняется на месте его нахождения и достигается экономия памяти. Так, построены, в частности, программы решения систем уравнений (VI. 3).

Второй пример: если массив A не используется для преобразования массива B , но область действия A заканчивается до ввода и (или) обработки B , после чего A очищается, то идентификаторы A и B можно заменить одним идентификатором A . В отношении типов и размеров массивов ограничения те же, что для первого примера.

После совмещений идентификаторов вносятся изменения в раздел описаний.

6. Составляются операторы присваивания, соответствующие расчетным формулам, определенным в пункте 1. Оператор присваивания отличается от описываемой им формулы тем, что он имеет динамическую природу — в арифметическом выражении правой части может быть переменная, указанная в левой части, т. е. рассматриваются два состояния переменной: до и после выполнения оператора присваивания, или, иными словами, до и после преобразования переменной.

7. В операторы присваивания могут входить как простые, так и индексные переменные. В последнем случае в качестве индексов вводятся специальные переменные (в рабочих программах им соответствуют счетчики), определяющие положение переменной в массиве. Это обычно порядковые номера строк (граф), порядковые номера ячеек в строке (графе), номера диагональных элементов в матрице, номера фиксирующих элементов в массиве исходных данных. Одновременно определяются ограничения для индексов в виде равенств или неравенств, в которые входят параметры задачи. Например, номера i строк в матрице системы уравнений ограничены порядком n системы уравнений ($i \leq n$).

8. Нужно стремиться к тому, чтобы вычисления имели циклический характер и выполнялись по минимальному количеству операторов. В сложных задачах количество формул, преобразуемых в операторы, может все же оказаться большим. Для уменьшения числа операторов соответствующие формулы следует попытаться разбить на группы таким образом, чтобы в каждой группе они имели подобную структуру. Затем анализируются отличия формул в пределах каждой группы. Нередко удается установить определенные закономерности в изменении формул. Тогда все формулы данной группы можно представить одним вычислительным оператором с применением переменных с индексами, изменение которых соответствует найденным закономерностям.

Текущие значения индексов указывают элементы рабочего массива, в которые помещаются коэффициенты, отличающие формулы друг от друга. Применение данного правила подробно показано на примерах в VII. 4, VII. 8.

По мере введения новых переменных и массивов они включаются в разделы описаний соответствующих блоков, а для глобальных величин — в общий раздел описаний для всей программы.

9. В тех случаях, когда вычисляемые элементы матриц независимы друг от друга (т. е. в процессе решения состояние соответствующего массива не меняется) и могут быть вычислены по одной или нескольким обобщенным формулам (см. предыдущий пункт), эти элементы можно помещать последовательно в одни и те же рабочие ячейки с вытеснением предыдущих чисел. Такая матрица является воображаемой (в том смысле, что она не хранится в памяти машины целиком) и используется лишь для того, чтобы установить те или иные закономерности процесса вычислений.

10. Составляется исполнительная часть программы в виде некоторой последовательности операторов: присваивания, перехода, условных и т. п. Полученные выше ограничения для индексов и других переменных, условия для признаков, для допустимой погрешности (в итерационных процессах) и т. д. являются основой для построения логической схемы алгоритма. После записи исполнительной части программы она отлаживается на небольшом примере на бумаге и если отладка показывает, что алгоритм выполняется, то переходим к следующему этапу — завершению программы. Но чаще всего первая отладка обнаруживает ошибки: закономерности алгоритма в одних местах соблюдаются, в других нарушаются. Это вынуждает исправлять условия, иногда вводить новые ограничения в переменные. Затем отладка, частичная или полная, повторяется. Таким образом составление исполнительной части программы проводится последовательными попытками до тех пор, пока отладка не даст удовлетворительных результатов.

11. Завершаются описания для отдельных блоков, процедур и для программы в целом. Это сделать нетрудно, так как все переменные и их типы определились при разработке предшествующих этапов. Нужно лишь проверить, нет ли упущений, все ли идентификаторы, фигурирующие в исполнительной части, описаны и нет ли лишних идентификаторов (исключенных на каком-либо этапе, но по недосмотру оставшихся в разделе описаний).

12. Готовая программа еще раз тщательно проверяется. Вся программа в целом отлаживается на бумаге, причем желательно отладить не только логическую схему алгоритма, но и сам счет, т. е. проверить правильность получаемых числовых результатов на небольшом примере, для которого есть готовый ответ. Такая отладка резко уменьшает возможность случайной ошибки.

Проверяется также (лучше другим лицом) синтаксис программы — нет ли неописанных идентификаторов, все ли гранич-

ные пары массивов, описанных в блоке, определены до входа в блок, не пропущена ли где-нибудь точка с запятой, соответствует ли каждому **begin** свое **end**. При проверке алгоритма с помощью отладки подобные «непринципиальные» ошибки легко пропустить; они могут сильно задержать получение программы с транслятора, который будет возвращать программу с указанием, где находится ошибка. При подготовке программы на языке АЛГОЛ-60 для ручного перевода на язык машины ошибки в синтаксисе не так опасны, поскольку кодировку выполняет человек, который по смыслу заметит и устранил явные опечатки*.

V. 10. Изложенная система правил определяет последовательные этапы составления алгоритмов на языке АЛГОЛ и дает некоторые рекомендации по выполнению этих этапов. Например, по правилу 8 предлагается установить подобие структур в разных формулах, найти закономерности изменения формул, использовать эти закономерности для составления обобщенного оператора, заменяющего группу формул. Следовательно, рекомендуется применение определенного принципа, но не указано точно, как его реализовать. В дальнейшем (гл. VII) приводятся примеры, показывающие применение этого принципа для ряда задач, но предполагается, что в других случаях могут быть найдены структурное сходство и закономерности иного типа.

Такое сочетание направляющего начала и свободы в выборе решения для конкретной задачи является плодотворным, если разработку алгоритма ведет человек.

Естественно, возникает вопрос: нельзя ли автоматизировать самый процесс составления алгоритмов? Задача эта весьма сложная, так как потребуются формализовать не только вычислительные, но некоторые творческие процессы, такие, например, как распознавание подобия формул, поиск закономерностей их изменения (правило 8). Это неизбежно приведет к общей проблеме машинной выводимости формул, обсуждение которой выходит за рамки данной работы.

* Бывает так, что допущенная опечатка неожиданно меняет смысл алгоритма, но не нарушает правил синтаксиса. Такие «смысловые» ошибки транслятор обнаружить не может. Например, если арифметическое выражение в операторе присваивания $a := b(c+d)$ записать, пропустив знак умножения, то новый оператор $a := b(c+d)$ будет воспринят транслятором как присваивание переменной a значения функции b с фактическим параметром $c+d$. Но этот оператор будет воспринят правильно человеком-кодировщиком, который мысленно восполнит знак умножения, как это обычно делается в алгебре, ибо он знает, что такой функции в алгоритме нет.

О формах публикации программы

V. 11. В соответствии с определением «Сообщения» знаки языка публикаций могут быть различными в разных странах, но должно быть сохранено однозначное соответствие с эталонным языком, т. е. отличаться могут только символы, а структура и содержание для всех представлений должны быть одними и теми же.

Допускается, например, написание служебных слов на национальном языке, обычная запись индексов ниже идентификатора переменной и удаление квадратных скобок, в выражениях — любая форма скобок (круглые, квадратные, фигурные). Но подобными изменениями лучше не злоупотреблять и если вводить их, то очень ограниченно. В самом деле, каждый грамотный человек знает латинский алфавит и может легко запомнить двадцать служебных слов, для этого вовсе не нужно знать английский язык. Желательно, чтобы программа была изложена на международном языке и не требовался ее перевод на национальные языки. По-видимому, по этой причине входные языки известных в нашей стране трансляторов с АЛГОЛа используют за основу эталонный язык.

Значительно важнее, чем в обозначениях, иметь некоторую свободу в формах публикации программ.

Задачи строительной механики характеризуются обычно большим объемом перерабатываемой информации. Вопрос экономии ячеек памяти весьма существен. Поэтому часто возникает необходимость поместить в одной ячейке (например, в первом, втором и третьем адресах или в отдельных разрядах) несколько величин. Для того, чтобы оперировать в программе с этими величинами, приходится прибегать к выделению содержимых частей ячеек, к сдвигам и т. д. Эти операции на языке АЛГОЛ-60 не описываются и для них должны применяться процедуры — коды для соответствующих машин. В этих случаях вместо тела процедуры будем в описаниях ставить многоточие и называть такие процедуры неописанными.

Остальная запись производится в соответствии с формальными правилами АЛГОЛа: перечисление неописанной процедуры в описаниях, формальные параметры, спецификации, обращение к процедуре (оператор процедуры) в исполнительной части программы с указанием фактических параметров (если процедура имеет параметры) и т. д. Многоточие вместо тела процедуры означает, что его нужно разработать дополнительно. При доработ-

ке программы неописанную процедуру можно заменить также другим оператором (вместо оператора процедуры записать блок или составной оператор) и исключить описание процедуры. Если при этом появляются новые идентификаторы, то их следует описать. Может возникнуть также необходимость в дополнительных операторах ввода. Применение неописанных процедур дает возможность публиковать программы с большей или меньшей степенью детализации, соблюдая при этом формальные правила языка АЛГОЛ-60.

Рассмотрим ряд примеров: 1. Общеизвестна важная роль, которую играет при передаче информации блок-схема программы, где намечены ее общие контуры; частично разработана логика разветвляющихся процессов, но отсутствует детализация отдельных блоков. Блок-схема программы изображается обычно в виде совокупности обозначенных тем или иным образом прямоугольников-блоков, соединенных идущими в различных направлениях стрелками. Разобраться в этом переплетении стрелок для сложной задачи без помощи автора программы бывает трудно.

Эквивалентом блок-схемы может служить программа, состоящая из неописанных процедур и ряда связывающих их операторов перехода (условных или безусловных). В разделе описаний помещаются названия процедур (вместо их тел — многоточия) и те идентификаторы, которые фигурируют в операторах перехода; если опустить раздел описаний, то получим неформальную запись алгольной блок-схемы.

2. Такая трактовка дает возможность последовательно увеличивать степень детализации блок-схемы. Следующая ступень — это программа, в которой не описаны только отдельные процедуры, а в остальном могут использоваться произвольные структуры АЛГОЛа.

Действительно, нередко бывает так, что для построения логической схемы оператор процедуры необходимо поместить в исполнительную часть программы, а тело процедуры в описательной части давать не имеет смысла (даже если оно есть). Как уже говорилось, для составления оптимальной программы приходится применять процедуры-коды, соответствующие особенностям конкретной машины: помещать в группах разрядов или адресах одной ячейки различную информацию — номера неизвестных, признаки различных видов * и т. д. При публикации программы

* Этого можно избежать, если каждый признак помещать в отдельную ячейку, но это сильно увеличивает объем исходной информации в памяти.

на алгоритмическом языке лучше не привязывать ее к конкретной машине, достаточно поместить обращение к некоторой процедуре-коду и в примечании указать, что от нее требуется (можно также дать рекомендацию, как составить подобную процедуру для тех или иных типов машин).

В тексте программы на языке АЛГОЛ-60, предназначенной для публикации, отдельные блоки программы, включающие процедуры-коды, представляются нами в виде неописанных процедур. Таким же образом удобно представить известные блоки программы (например, блоки решения систем уравнений тем или иным методом), которые не имеет смысла повторять при публикации каждой новой программы, включающей эти блоки.

3. Особенно эффективно применение неописанных процедур для обозначения совокупности алгоритмов некоторого класса. Например, в универсальную программу метода деформаций входит неописанная процедура расчета абстрактного элемента. Разрабатывая процедуры для расчета элемента в системе определенного типа и помещая в универсальную программу описание тела этой процедуры, получаем специализированную программу для сооружений указанного типа.

Несколько слов о примечаниях к программе. Согласно «Сообщению» примечания помещаются в тексте программы после служебного слова **comment**. Однако это не всегда удобно. Если примечания даются к группе операторов, то трудно указать, к какому из операторов относится то или иное примечание (если операторы не имеют меток). Давать примечания к каждому оператору тоже неудобно, этим разбивается текст программы и затрудняется ее восприятие как единого целого. Между тем, при публикациях подробные примечания очень полезны — можно разъяснить назначение идентификаторов, смысловое значение тех или иных операторов и т. д.

В тех случаях, когда много коротких примечаний, они будут размещены на полях справа от программы. При такой записи можно дать, если нужно, краткие пояснения к любому оператору программы, не разбивая ее текста, а также уменьшается возможность ошибки при подготовке к трансляции (техник, набирающий программу на телетайпе, может механически записать какое-либо из примечаний). Длинные примечания помещаются, как обычно, в тексте программы после слова **comment**.

V. 12. В нашей стране автоматизация программирования в последние годы получила значительное развитие. Применяются как большие, универсальные, так и более короткие трансляторы (программирующие программы, рассчитанные на автоматизацию программирования массовых задач); для малых машин создаются простейшие трансляторы-автокодировщики.

Современные автоматизированные системы работают с полной нагрузкой. В виде примера можно сослаться на практику КиевЗНИИЭПа, являющегося, по-видимому, первой организацией строительного профиля, предпринявшей значительный эксперимент по автоматизации программирования с помощью транслятора ТА-2: десятки программ, малых (несколько сот кодов) и больших (несколько тысяч кодов) были изготовлены в короткий срок на машине М-20.

Пока еще программы, полученные с помощью транслятора, имеют большую длину и менее эффективны (увеличивается время счета по программе), чем полученные вручную опытными программистами.

Эта разница невелика для относительно простых задач (например, длина программ решения систем уравнений из гл. VI увеличивается на 15—20%) и резко возрастает для задач сложных, для которых характерны обмены между оперативной и внешней памятью и применение процедур (особенно рекурсивных).

Поэтому проводится большая работа по дальнейшему совершенствованию трансляторов и устранению их недостатков: предусматривается динамическое распределение памяти (транслятор «Альфа» [41]), приближение к языку машин (транслятор с адресного языка [3]), вводятся так называемые комментарии для исполнения (транслятор ТА-2 [131]), с помощью которых можно предписать желательное распределение памяти. Дальнейшее развитие этого принципа позволит сочетать опыт, знания и изобретательность человека с преимуществами автоматического изготовления программ на машине.

Все же пока еще рано говорить о полной и повсеместной автоматизации программирования хотя бы потому, что для многих машин трансляторы еще не созданы. Кроме того, по изложенным выше причинам типовые программы, предназначенные для многократного применения (тем более стандартные программы, входящие в интерпретирующие системы), все еще имеет смысл со-

ставлять вручную с оптимальной организацией памяти и учетом индивидуальных особенностей задачи.

Но поставим вопрос иначе: можно ли считать, что в ближайшем будущем автоматизация программирования станет нормой, а ручное кодирование редкостью?

На этот вопрос, по-видимому, можно ответить утвердительно. Уже сейчас для машин с большим объемом памяти и большим быстродействием некоторое уменьшение или увеличение длины программы и времени счета не имеет практического значения. Что значит сэкономить несколько сот ячеек для машины с емкостью оперативной памяти в 100000 ячеек? Это не то, что для машины с памятью в 1024 или 2048 ячеек. И действительно, для больших машин автоматизированная система программирования становится необходимостью и нормой (в США большинство трансляторов построено на языке ФОРТРАН, близком предшественнику АЛГОЛа).

Однако имеется другая точка зрения, прямо противоположная и разделяемая авторитетными специалистами.

Транслятор нивелирует потребителей ЭЦВМ и опытных программистов. Первые видят в трансляторе возможность в короткие сроки получить программу без совместной работы с программистом обычно очень длительной. Вторые имеют опыт быстрого получения вручную коротких и изящных программ, сравнивают и видят недостатки получаемых с транслятора программ, а также самого процесса трансляции, отладки и контроля.

Это вызывает раздражение, мешает объективно разобраться в причинах и найти пути для преодоления недостатков. После одной-двух мало удачных попыток нередко теряют терпение и приходят к выводу, что дефект находится в самом принципе автоматизации программирования.

Р. С. Гутер, автор хороших книг по программированию, в одной из них пишет следующее (см. [31], заключение, стр. 205—211): «Если сравнивать АЛГОЛ с системой программирования в содержательных обозначениях, то он экономит только кодирование программы, которое при правильной организации работы все равно делается не программистом.... Стремление создать язык программирования, независимый от конкретной машины, на которой данный алгоритм должен быть реализован, как было показано выше, само по себе неразумно». Автор далее утверждает, что достаточно иметь трех кодировщиц, чтобы обеспечить всю кодировку программ для большой электронной счетной машины, и вспоминает то место из «Одноэтажной Америки» И. Ильфа и

Е. Петрова, где описывается завод Форда и рабочий, вручную кисточкой проводящий цветную полосу на борту готового автомобиля, поскольку автоматизировать эту операцию бессмысленно. Вот как расценивается огромная работа, проведенная во всем мире по автоматизации программирования.

Такое заявление обескуражило многих читателей этой книги, вышедшей большим тиражом. В высказывании Р. С. Гутера имеется однако по крайней мере две натяжки.

Во-первых, автор сам применяет алгоритмический язык, не может без него обойтись, ибо содержательные обозначения — не что иное как одна из зачаточных форм алгоритмического языка, которую можно считать предшествующей адресному языку Института кибернетики АН УССР*.

Во-вторых, трудно согласиться с тем, что три кодировщицы могут полностью обслуживать большую электронную машину. Но если бы это и было так (в конце концов, три ли будет кодировщицы или несколько больше, не меняет дела), то кто будет готовить для них информацию в содержательных обозначениях? Для этого все заказчики должны хорошо знать машину и быть опытными программистами, либо обращаться к программистам, чтобы те составляли для них алгоритмы. А вот последнее нежелательно. Подобная совместная работа мало эффективна. Может ли специалист данной области знания сообщить малосведущему в ней программисту, в чем состоит алгоритм, если ему самому многое неясно? Ведь в процессе составления алгоритма выясняются закономерности, которые заранее знать невозможно.

V. 13. Здесь приходит на помощь такой развитый алгоритмический язык, как АЛГОЛ-60, отвлеченный от конкретной машины (в котором, как признает Р. С. Гутер, «Формулы выглядят даже более привычно, чем в содержательных обозначениях»), и в этой отвлеченности глубокий смысл. Благодаря этому АЛГОЛ может быть использован и в тех случаях, когда не применяется транслятор, как язык-посредник между специалистами различных областей знаний и программистами. Возможны два способа передачи информации.

Первый способ рассчитан на квалифицированного программиста. Алгольная программа составляется инженером сжато, без детализации. К такой программе желательно иметь комментарии

* Вопрос о приоритете здесь не обсуждается. Вполне возможно, что А. Л. Брудно, автор системы содержательных обозначений [7], предложивший ее в 1953 г., является одним из первых создателей алгоритмического языка.

для исполнения, в которых автор указывает частные особенности задачи и намечает возможные пути экономии памяти, вычислений и т. п.

Главное в такой программе — исчерпывающе описать алгоритм. Все остальное (выбор рационального способа задания информации, расположение массивов в памяти и их замещение, переработка алгоритма с целью его оптимизации) выполняется программистом, которому представляется значительная инициатива.

Второй способ рассчитан на техника-кодировщика. Составляется развернутая алгольная программа с полным решением всех вопросов. Задача кодировщика в том, чтобы формально перевести эту программу на язык данной машины. Здесь может быть разработан четкий алгоритм перевода. Подобно тому, как это делается в трансляторе, при ручном переводе также можно использовать различные вспомогательные таблицы: перевод характерных формул АЛГОЛа, распределение памяти констант и т. д. Следовательно, осуществляется частичная автоматизация программирования, которая заключается в механической кодировке алгольной программы. Как видим, в данном случае программа на АЛГОЛе выполняет те же функции, что и программа в содержательных обозначениях.

В работе [109] описан алгоритм ручного перевода для машины БЭСМ-2М, составленный программистом Ж. М. Ременник по плану и под руководством автора. Аналогичные алгоритмы могут быть построены для любых других машин: это может сделать квалифицированный программист, хорошо знающий машину.

ПОСТРОЕНИЕ СИСТЕМ АЛГОРИТМОВ СТРОИТЕЛЬНОЙ МЕХАНИКИ

VI. 1. Система алгоритмов, излагаемая в данной и последующей главах, состоит из следующих частей *:

1. Процедуры решения систем линейных уравнений определенных структур методом исключения и циклическими методами.

2. Универсальные программы расчета статически неопределимых систем методами сил и деформаций с применением абстрактных (переменных) процедур расчета элемента. В методе сил элементом является прямолинейный участок стержневой системы (криволинейный участок заменяется рядом прямолинейных). В методе деформаций под элементом понимается произвольная стержневая или непрерывная система (отдельный стержень, пластинка, набор стержней и пластинок, пространственный блок и т. д.), тем или иным образом закрепленная в пространстве.

3. Множество процедур расчета элемента в статически неопределимых плоских и пространственных системах различных видов, каждая процедура этого множества является конкретным воплощением абстрактной процедуры («значением» переменной процедуры).

4. Процедуры расчета элемента в пространственных системах со структурной симметрией различных видов. Первоначально работы проводились в рамках основной библиотеки метода деформаций, но затем было решено создать специальную библиотеку программ для расчета систем с циклической и сферической симметрией.

5. Процедура определения частот и форм собственных колебаний произвольной системы. В строительном проектировании эта процедура применяется главным образом при расчете зданий и сооружений на сейсмические воздействия.

* В работе [109] приводились отдельные процедуры этой системы; как оформленное целое она излагается впервые.

6. Процедуры линеаризации нелинейных задач. Кратко рассматриваются обобщенные алгоритмы расчета дискретных систем с односторонними (ограниченными) связями, стержневых систем на устойчивую прочность, рамных систем с учетом действительной работы железобетона. Даны краткая характеристика шаговых алгоритмов и их сравнение с итерационными.

7. Процедуры формирования основной системы метода деформаций и построения операторов на основе анализа зрительного образа расчетной схемы, отображенной на сетках или решетках различных видов. Обсуждаются перспективы автоматизации комплекса расчетов при проектировании.

При построении системы алгоритмов и соответствующей библиотеки программ применяется принцип последовательного рекурсивного развития, о котором говорилось в предисловии к книге; имеются потенциальные возможности для дальнейшего расширения и углубления системы.

Решение систем линейных уравнений ленточной и ленточной окаймленной структур

VI. 2. Для многих задач строительной механики характерна ленточная (кодиагональная) структура получаемых систем уравнений, в которой значащие числа располагаются на некотором количестве диагоналей по обе стороны от главной диагонали.

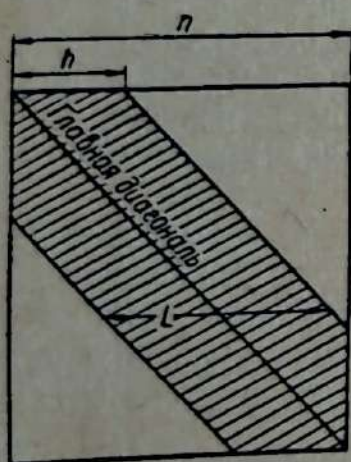


Рис. 32.

На рис. 32 поле значащих чисел ленты заштриховано, вне ленты все элементы матрицы нулевые. Особенностью этих систем является то, что в процессе их решения методом исключения элементы, расположенные вне ленты, постоянно остаются нулевыми и в процессе решения не участвуют. Это дает возможность исключить эти элементы и не записывать их в оперативную память.

Нулевые элементы, расположенные внутри ленты, при решении системы заполняются значащими числами и их следует записывать в память.

Упорядоченный характер структуры позволяет описать ее одним параметром h — порядком ленты, представляющим собой число значащих диагоналей, расположенных справа от главной, включая последнюю.

При $h=1$ имеем диагональную матрицу, $h=2$ — трехчленную матрицу, $h=3$ — пятичленную матрицу и т. д. При условии, что порядок ленты h равен порядку системы n ($h=n$), получаем полностью заполненную матрицу. Таким образом, любая система уравнений, в том числе и полностью заполненная, может трактоваться как ленточная.

$n=9$

	$h=4$								
0	1	2	3	4					
1	5	6	7	8	9				
2	10	11	12	13	14	15			
3	16	17	18	19	20	21	22		
4		23	24	25	26	27	28	29	
5			30	31	32	33	34	35	36
6				37	38	39	40	41	42
7					43	44	45	46	47
8						48	49	50	51

Рис. 33.

$n=9$

	$h=4$								
0	1	2	3	4					
1		5	6	7	8				
2			9	10	11	12			
3				13	14	15	16		
4					17	18	19	20	
5						21	22	23	24
6							25	26	27
7								28	29
8									30

Рис. 34.

ваться как ленточная. Форма матрицы в рассматриваемой задаче получается сложной, и запись ее в виде двухмерного массива, рассчитанного на прямоугольную форму, здесь не применима. Поэтому расположим матрицу в одномерном массиве со сквозной нумерацией ячеек по строкам. Отдельно расположим массив свободных членов. Нумерация ячеек показана на простейших примерах: для несимметричной матрицы — на рис. 33, для симметричной — на рис. 34.

Элементы матрицы, расположенные вне ленты, не нумеруются и для них не нужно предусматривать ячейки памяти.

С помощью одного или нескольких параметров можно описать также и другие упорядоченные структуры, если известны начальная и преобразованная формы матрицы. Например, ленточную окаймленную матрицу можно описать с помощью двух параметров: порядка ленты h и порядка пустого треугольника $n-1$. Нумерация ячеек для этого случая показана на рис. 35. Форма исходной и преобразованной матрицы здесь также совпадает.

Матрица, состоящая из ряда накладывающихся квадратов, которая при преобразовании также сохраняет свою структуру, изображена на рис. 36. Вводятся параметры $n[i]$ — размер квадрата, $n[i+1]$ — размер наложения данного квадрата на после-

дующий. Здесь i — порядковый номер квадрата. Если обозначить t — количество квадратов, то параметры помещаются в массиве $n[1 : 2 \times t]$. В том случае, когда квадраты имеют ленточную или ленточную окаймленную структуру, вводятся дополнительные параметры. Если все квадраты одинаковы, то можно ограничить-

	$n=9$								
	$h=3$			$n_i=4$					
0	1	2	3					4	5
1		6	7	8				9	10
2			11	12	13			14	15
3				16	17	18		19	20
4					21	22	23	24	25
5						26	27	28	29
6							30	31	32
7								33	34
8									35

Рис. 35.

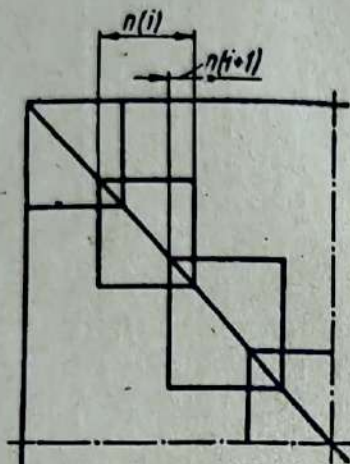


Рис. 36.

ся параметрами для одного квадрата и указать их количество. Во всех рассмотренных случаях применяется обычный метод однократного исключения неизвестных, параметры входят в логические условия алгоритма.

Если форма матрицы меняется в процессе исключения неизвестных, то особенности структуры можно использовать для экономии ввода и при итерационных методах. (VI. 12).

Нумерация ячеек для рассмотренных матриц следует определенным закономерностям. Покажем, как их находить на примере симметричной ленточной матрицы (рис. 34). Прежде всего установим, что определяющим в каждой строке является номер диагонального элемента, номера всех остальных элементов получаются прибавлением к диагональному элементу числа m , изменяющегося от 1 до $h-1$ с шагом 1. Так, на рис. 34 для первой строки номер диагонального элемента равен 1, номера следующих элементов будут 2, 3, 4; для второй строки номер диагонального элемента равен 5, а последующих элементов — 6, 7, 8 и т. д. Нетрудно заметить, что номера диагональных элементов отличаются друг от друга на величину h ; первый диагональный элемент имеет номер $d=1$, второй диагональный элемент — $d=1+$

$+h=1+4=5$, третий диагональный элемент $-d=5+h=5+4=9$ и т. д.

Эта закономерность нарушается в нижней треугольной части матрицы. Если нумеровать строки $0, 1, 2, \dots, n-1$ (первая строка имеет нулевой номер, последняя номер, $n-1$), то после строки $n-h$ каждая последующая строка содержит на одну клетку меньше, чем предыдущая. Пятая строка ($n-h=5$) на рис. 34 содержит $h=4$ клетки, шестая строка — 3 клетки, седьмая — 2 клетки и последняя восьмая ($n-1$) строка — 1 клетку. Для того, чтобы учесть эту особенность, введем переменную $h1$, которая для $n-h$ строк равна h , а затем последовательно уменьшается на единицу. Тогда формула для определения номера диагонального элемента будет $d := d + h1$. Матрицу системы уравнений расположим в линейном массиве A , свободные члены — в массиве B .

Теперь мы имеем все необходимое для составления вычислительных операторов, в которые входят переменные с индексами, и условных операторов, в которые входят ограничения для индексов.

Обратимся к эскизной (инженерной) программе прямого хода для симметричной матрицы (IV. 23). Номер текущего элемента строки k обозначим $d1 + m$, где $d1$ — номер диагонального элемента, m — переменная величина. Диагональному элементу $i.i$ соответствует номер d , а элементу $i.k$ — номер $d + m1$. Текущий элемент строки i отличается от элемента $i.k$ на величину m , поэтому соответствующий номер определяется выражением $d + m1 + m$. Указанные номера являются индексами переменных в массиве A . Перепишем вычислительную формулу эскизной программы в этих обозначениях, поменяв местами левую и правую части и заменив операцию засылки операцией присваивания:

$$A[d1 + m] := A[d1 + m] - A[d + m1] \times A[d + m1 + m] / A[d]$$

элемент	элемент	элемент	элемент	элемент
строки k	строки k	$i.k$	строки i	$i.i$

По этой формуле преобразовываем матрицу без свободных членов.

Преобразование свободных членов (массив B) выделим в отдельную формулу:

$$B[i + 1 + m1] := B[i + 1 + m1] - A[d + m1] \times B[i + 1] / A[d].$$

Эти формулы аналогичны по структуре. Разделение их необходимо для того, чтобы реализовать принятую последовательность выполнения процедур прямого и обратного хода (об этом будет сказано дальше).

Определяем основные ограничения для индексов. Поскольку первая строка матрицы имеет нулевой номер, то вместо последовательности $i=1, 2, 3, \dots, n-1$ (см. эскизную программу) получаем последовательность $i=0, 1, 2, \dots, n-2$. Для i можно организовать цикл:

for $i := 0$ **step** 1 **until** $n-2$ **do**...

Индекс $d+m1$, указывающий номер элемента в строке, очевидно, принимает значения $d+1, d+2, \dots, d+h1-1$. Например, по рис. 34 для строки 0 получаем числа 2, 3, 4 (последнее число равно $d+h1-1=1+4-1=4$). Для последовательных значений $m1=1, 2, \dots, h1-1$ можно организовать цикл:

for $m1 := 1$ **step** 1 **until** $h1-1$ **do**...

В индексе $d+m1+m$ величина m последовательно принимает значения $0, 1, 2, \dots, h1-m1-1$. Заголовок соответствующего цикла будет иметь вид:

for $m := 0$ **step** 1 **until** $h1-m1-1$.

Как уже указывалось, для определения номера диагонального элемента d применяется формула:

$$d := d + h1,$$

а для определения $h1$ составляется условие:

if $i \geq n-h1$ **then** $h1 := h1-1$.

Если $i \geq n-h1$, то значение $h1$ при переходе к последующей строке уменьшается на единицу; если $i < n-h1$, то $h1$ сохраняет начальное значение, равное h . По существу та же зависимость должна иметь место и для $d1$, но особенность состоит в том, что $d1$ изменяется в зависимости не только от i , но и от $m1$. Поэтому формулу для определения $d1$ представим в виде

$$d1 := d1 + h1 - j.$$

Переменная j необходима в связи с тем, что при расположении диагонального элемента в зоне треугольной части матрицы величину $d1$ нужно последовательно уменьшать на единицу.

Для определения j составим условие:

if $i+m1-1 \geq n-h1$ **then** $j := j+1$.

Тем самым проверяем, в какой части матрицы находится $d1$, соответствующее данному $m1$: если в нижней треугольной части матрицы, то j увеличивается на единицу и $d1$ соответственно уменьшается, если в зоне параллелограмма, то j равно начальному значению — нулю. Например, на рис. 34 при $n=9, i=4$ и $h1=4$ получим начальное значение $j=0, d1=d=17$.

$$m1 := 1; d1 := d1 + h1 - j = 17 + 4 - 0 = 21;$$

условие $\text{if } 4+1-1 \geq 9-4$ не удовлетворяется и $j=0$.

$$m1 := 2; d1 := d1 + h1 - j = 21 + 4 - 0 = 25;$$

условие $\text{if } 4+2-1 \geq 9-4$ удовлетворяется и $j=0+1=1$.

$$m1 := 3; d1 := d1 + h1 - j = 25 + 4 - 1 = 28.$$

Это значение $m1$ является последним, поскольку $h1-1=3$.

Иногда в принятое вначале условие требуется внести коррективы. Это выясняется при непосредственной проверке на простейшем примере. В данном случае вначале условие было записано так:

$$\text{if } i + m1 \geq n - h1 \text{ then } j := j + 1.$$

Хотя это условие качественно и определяет характер изменения величины $d1$, которая, начиная с некоторой строки, последовательно должна уменьшаться на 1, 2, 3, ..., но количественно оно не приводит к нужному результату. При проверке выяснилось, что в левой части неравенства нужно поставить -1 , т. е. изменение величины j начать раньше.

Для процедуры прямого хода выбираем формальный параметр b . Фактическими значениями этого параметра могут быть: $b=0$ — признак выполнения прямого хода для матрицы, $b=1$ — признак выполнения прямого хода для свободных членов (соответствующие вычислительные операторы приведены выше).

После выполнения прямого хода, реализующего последовательное исключение неизвестных, на месте нахождения исходной получаем преобразованную матрицу, в которой в n -ом уравнении имеется одно неизвестное, $n-1$ -м уравнении — два неизвестных, $n-2$ -м уравнении — три неизвестных, ..., 1-м уравнении — n неизвестных.

Величины неизвестных определяем при обратном ходе путем последовательной подстановки неизвестных: из n -го уравнения находим неизвестное x_n , подставляем x_n в уравнение $n-1$ и находим x_{n-1} , подставляем x_n, x_{n-1} в уравнение $n-2$ и находим x_{n-2} и т. д. до первого уравнения. Определяемые величины неизвестных можно помещать на места свободных членов с вытеснением последних.

Формулы для определения неизвестных при обратном ходе можно записать так:

$$x_i = - \frac{a'_{iq}}{a_{ii}} \text{ и } a'_{iq} = a_{iq} + \sum a_{ik} x_k,$$

где x_i — вычисляемое неизвестное;

a'_{iq} — результат подстановки неизвестных в i -тое уравнение плюс свободный член;

a_{ii}, a_{ik} — преобразованные коэффициенты i -го уравнения;

a_{iq} — преобразованный свободный член i -го уравнения;

x_k — предыдущие неизвестные.

Ввиду ленточной структуры матрицы суммирование выполняется от $k=i+1$ до $k=i+hl-1$. Напомним, что элементы, расположенные вне ленты, равны нулю и не нумеруются. Поэтому они не фигурируют во втором члене второй формулы. Первую формулу в терминах АЛГОЛа с применением переменных с индексами можно записать следующим образом:

$$B[i] := -B[i]/A[d].$$

Значение вычисляемого неизвестного x_i помещаем в ячейку $B[i]$ массива B с вытеснением прежнего содержимого этой ячейки, равного a'_{ig} . $A[d]$ есть диагональный элемент a_{ii} . Индекс i изменяется от n до 1. Вторая формула в терминах АЛГОЛа записывается так:

$$B[i] := B[i] + A[d+m] \times B[i+m].$$

Один и тот же член в левой и правой части формулы указывает, что в ячейке $B[i]$ накапливается сумма произведений. $A[d+m]$ соответствует коэффициенту a_{ik} , $B[i+m]$ — неизвестному x_{i+m} , найденному ранее. Величина m изменяется от 1 до $hl-1$ и заголовок соответствующего цикла запишется в виде:

for $m := 1$ step 1 until $hl-1$ do...

После выполнения прямого хода величина hl равняется 1. При обратном ходе hl последовательно увеличивается на единицу до тех пор, пока не достигнет величины h , и далее остается неизменной. Этот процесс выражается условием

if $h > hl$ then $hl := hl + 1$.

Если $h > hl$, то величина hl увеличивается на единицу, в противном случае не увеличиваем hl и переходим к следующему оператору программы. Изменение величин d и i выражается формулами $d := d - hl$; $i := i - 1$. Окончание обратного хода регламентируется условием $d = 1$, указывающим, что последнее неизвестное уже вычислено.

Процедуры прямого и обратного хода для матриц рассматриваемого типа помещены в VI. 4.

Обратный ход оформлен как процедура без параметров*.

Аналогично составляются вычислительные операторы и ограничения для индексов матриц, изображенных на рис. 33 и 35. Для этих систем закономерности изменения индексов сложнее, что потребовало введения ряда дополнительных переменных и соответствующих им ограничивающих неравенств, которые нетрудно найти в процедурах, помещенных в VI. 5, VI. 6.

Имея процедуры прямого и обратного хода, остается объединить их в одно целое — программу решения систем линейных уравнений. Поставим следующую задачу: программа должна обеспечивать возможность решения системы для произвольного количества граф свободных членов при стабильном объеме массива B и в случае необходимости выдавать обратную матрицу**. В соответствии с такой постановкой задачи примем следующую последовательность вычислений:

- а) ввод матрицы в массив A оперативной памяти;
- б) первый этап прямого хода для массива A ;
- в) ввод j -й графы свободных членов в массив B оперативной памяти;
- г) второй этап прямого хода для массива B ;
- д) обратный ход, получение неизвестных в массиве B и вывод их на печать.

Пункты в, г, и д образуют цикл вычислений, который повторяется для $j = 1, 2, \dots, q$ (первой, второй, ..., q -й графы свободных членов) с замещением предыдущего содержимого массива B — величин неизвестных — новой графой свободных членов. Весь этот процесс автоматизирован: ввод первой графы свободных членов, счет, печать неизвестных, ввод второй графы свободных членов, счет, печать неизвестных и т. д. до последней q -й графы. Эта схема дает возможность экономить память (для произвольного числа свободных членов требуется стабильный массив $B[1:n]$) и особенно удобна для машин, имеющих буферное устройство, в которых счет и печать могут производиться одновременно.

q — число граф свободных членов — является параметром задачи. Если $q=0$, то вычисляется обратная матрица в следующем порядке:

* Для некоторых задач целесообразно в качестве формальных параметров процедур прямого и обратного хода принять идентификаторы массивов A , B и порядок системы n .

** В строительной механике обратную матрицу называют еще матрицей чисел влияния — в ней даны величины неизвестных при i -том свободном члене, равном единице, а остальных — нулю.

очищается массив B : переменной $B[j1]$ присваивается значение 1, 0, где $j1$ — порядковый номер элемента в массиве B ;

выполняются вычисления — второй этап прямого хода и обратный ход, печатается строка $j1$ обратной матрицы.

Цикл вычислений выполняется для $j1 = 1, 2, \dots, n$. Поскольку обратная матрица печатается построчно, то ее не нужно хранить в памяти.

Программа, которую предстоит составить, является компоновочной и состоит из операторов ввода и вывода, операторов процедур прямого и обратного хода и некоторых условных операторов. Структурные особенности матриц отражаются только на описаниях массива A , некоторых переменных, процедур прямого и обратного хода. Поэтому имеет смысл в общей программе эти описания не давать и изобразить многоточиями (см. неописанные процедуры, V. 11). Помещая вместо многоточий соответствующие описания (они даны отдельно), получим специализированные программы определенного вида. Этот прием будет нами и в дальнейшем широко применяться.

VI. 3. Программа решения систем линейных уравнений:

```
begin integer  $n, h, q, h1, d, d1, c, m, m1, i, j, j1, \dots$ ; input ( $n, h,$   
 $q, \dots$ );  
begin array  $A[1 : \dots], B[1 : n]$ ;  
procedure прямой ход ( $b$ ); ...;  
procedure обратный ход; ...;  
input ( $A$ ); прямой ход (0);  
if  $q=0$  then  $c := n$  else  $c := q$ ;  
for  $j1 := 1$  step 1 until  $c$  do  
begin if  $q=0$  then go to M1;  
input ( $B$ ); go to M2;  
M1: for  $m := 1$  step 1 until  $n$  do  
     $B[m] := 0$ ;  $B[j1] := 1, 0$ ;  
M2: прямой ход (1); обратный ход;  
output ( $B$ );  
end  $j1$  end end программы.
```

Количество ячеек в массиве A (верхняя граница) определяется по формуле: для симметричных ленточных матриц —

$$u = n \times h - h \times (h-1)/2;$$

для несимметричных ленточных матриц —

$$u = (2 \times h - 1) \times n - h \times (h-1);$$

для симметричных ленточных окаймленных матриц

$$u = (n \times (n+1) - n1 \times (n1+1))/2.$$

В разделе описаний указаны глобальные величины (n , h , q и т. д.) для программы *.

VI. 4. Описания процедур для симметричных ленточных матриц:

```

..., input (n, h, q); begin array A[1 : n × h — h × (h — 1) / 2],
B[1 : n]; procedure прямой ход (b); value b; integer b;
begin
M1 : h1 := h; d := 1;
for i := 0 step 1 until n — 2 do begin M2 : j := 0; d1 := d;
for m1 := 1 step 1 until h1 — 1 do begin M3 : if b = 0 then
go to M4; B[i + 1 + m1] := B[i + 1 + m1] — A[d + m1] ×
× B[i + 1] / A[d];
go to M5; M4 : d1 := d1 + h1 — j;
for m := 0 step 1 until h1 — m1 — 1 do
A[d1 + m] := A[d1 + m] — A[d + m1] × A[d + m1 + m] / A[d];
if m1 + i — 1 ≥ n — h1 then j := j + 1;
M5 : end m1;
d := d + h1; if i ≥ n — h1 then h1 := h1 — 1 end i end прямой ход;
procedure обратный ход;
begin i := n;
M1 : B[i] := —B[i] / A[d];
if d = 1 then go to M2;
if h > h1 then h1 := h1 + 1; d := d — h1;
i := i — 1; for m := 1 step 1 until h1 — 1 do
B[i] := B[i] + A[d + m] × B[i + m]; go to M1;
M2 : end обратный ход; ...

```

Данный отрезок должен быть включен в основную программу в соответствующем месте (в VI. 3 отчеркнуто слева).

VI. 5. Описания процедур для несимметричных ленточных матриц:

```

..., h2, r, j2, input (n, h, q); begin array A[1 : (2 × h — 1) ×
× n — h × (h — 1)], B[1 : n];
procedure прямой ход (b); value b; integer b, j2; begin
M1 : h1 := h; h2 := d := 1;
for i := 0 step 1 until n — 2 do begin

```

* Предполагается, что эти величины входят во все специализированные программы, порождаемые данной общей программой. Однако некоторые из описанных глобальных величин могут и не использоваться в какой-либо из специализированных программ, это не приведет к ошибке. С другой стороны, могут потребоваться дополнительные глобальные переменные в специализированной программе. Многоточие за последним идентификатором $j1$ указывает на такую возможность. При дополнении программы многоточия заменяются соответствующими идентификаторами либо пустым символом.


```

M2: j := j2 := 0; d1 := d;
    for m1 := 1 step 1 until h1-1 do begin
        M3: d1 := d1 + h1 + h2 + j - j2; r := d1 - m1;
        if b = 0 then go to M4;
        B[i+1+m1] := B[i+1+m1] - A[r] × B[i+1]/A[d]; go to M5;
M4: for m := 1 step 1 until h1-1 do
    A[r+m] := A[r+m] - A[r] × A[d+m]/A[d];
M5: if h > i+m1+1 then j := j+1;
    if i+m1 > n-h then j2 := j2+1 end m1;
    d := d+h1+h2;
    if h-1 > i+1 then h2 := h2+1;
    if i+1 > n-h1 then h1 := h1-1
    end i end прямой ход;
procedure обратный ход;
begin r := n; i := 1;
M1: B[r] := -B[r]/A[d];
    if d = 1 then go to M2;
    if h1 ≠ h then h1 := h1+1;
    d := d-h1-h2; i := i+1;
    if i > n-h+1 then h2 := h2-1; r := r-1;
    for m := 1 step 1 until h1-1 do
        B[r] := B[r] + A[d+m] × B[r+m]; go to M1;
M2: end обратный ход; ...

```

VI. 6. Описания процедур для симметричных окаймленных ленточных матриц:

```

..., nu, r, r1; input (n, h, q, n1);
begin array A[1 : (n × (n+1) - n1 × (n1+1)) / 2], B[1 : n];
procedure прямой ход (b); value b; integer b;
begin
M1: h1 := n - n1; d := 1;
    for i := 0 step 1 until n-2 do begin
M2: j := 0; d1 := d; nu := 0;
        for m1 := 1 step 1 until h1-1 do begin
M3: if b = 0 then go to M4;
            if m1 ≥ h then begin if n1 ≥ i then nu := n1 - i end;
            r1 := 1 + m1 + nu + i;
            B[r1] := B[r1] - A[d+m1] × B[i+1]/A[d];
            go to M7;
M4: d1 := d1 + h1; if i+m1+nu ≤ n1 then go to M5;
            j := j+1; d1 := d1 - j+1;
M5: if m1 ≤ h-1 then go to M6; nu := nu+1;
            if nu+i ≤ n1 then go to M4;

```



```

M6: for  $m := 0$  step 1 until  $h1 - m1 - 1$  do
  begin  $r := d1 + m$ ; if  $m1 + m > h - 1$  then
     $r := r + m1 - j$ ;
     $A[r] := A[r] - A[d + m1] \times A[d + m1 + m] / A[d]$ 
  end  $m$ ;
M7: end  $m1$ ;
 $d := d + h1$ ; if  $i \geq n1$  then  $h1 := h1 - 1$ ;
end  $i$  end прямой ход;
\ procedure обратный ход;
begin  $r := n$ ;  $nu := 0$ ;
M1:  $B[r] := -B[r] / A[d]$ ;
  if  $d = 1$  then go to M2;
  if  $n - n1 \neq h1$  then  $h1 := h1 + 1$  else  $nu := nu + 1$ ;
   $d := d - h1$ ;  $r := r - 1$ ;
  for  $m := 1$  step 1 until  $h1 - 1$  do begin;
     $r1 := r + m$ ; if  $h1 \geq n - n1$  then begin
      if  $m > h - 1$  then  $r1 := r1 + nu$  end;
       $B[r] := B[r] + A[d + m] \times B[r1]$  end  $m$ ;
  go to M1;
M2: end обратный ход; ...

```

VI. 7. Изложенные выше программы позволяют во многих случаях разместить в оперативной памяти большие системы уравнений. По формулам, помещенным в конце VI. 3, можно вычислить возможный порядок системы n , зная полезный объем оперативной памяти v (за вычетом программы) и порядок ленты h (для окаймленных матриц порядок пустого треугольника $n1$). Например, на машине М-20 при порядке ленты $h = 25$ (соответствует строке из 49 клеток) можно решить симметричную систему уравнений 160 порядка. При более узких лентах возможный порядок системы достигает нескольких сот *.

Однако и при весьма широких лентах можно получить значительный эффект, если использовать внешнюю память машины. В этом случае матрицу системы уравнений и все графы свободных членов лучше расположить подряд в одном массиве (рис. 37).

Особенность ленточной матрицы ** состоит в том, что после

* Даже в тех случаях, когда полностью заполненная матрица помещается в памяти, учет ленточной структуры полезен, так как уменьшается объем подготавливаемой и вводимой информации.

** Рассматриваются характерные для строительной механики симметричные матрицы. Для несимметричных ленточных матриц, а также ленточных окаймленных матриц могут быть построены аналогичные алгоритмы.

исключения при прямом ходе группы неизвестных n_2 преобразовываются два блока чисел: u_2 и частично u_3 , остальное поле матрицы ниже заштрихованной части остается без изменения. Это позволяет вводить матрицу в оперативную память машины блоками. Сначала вводится блок $u_1 = u_2 + u_3$. После выполнения

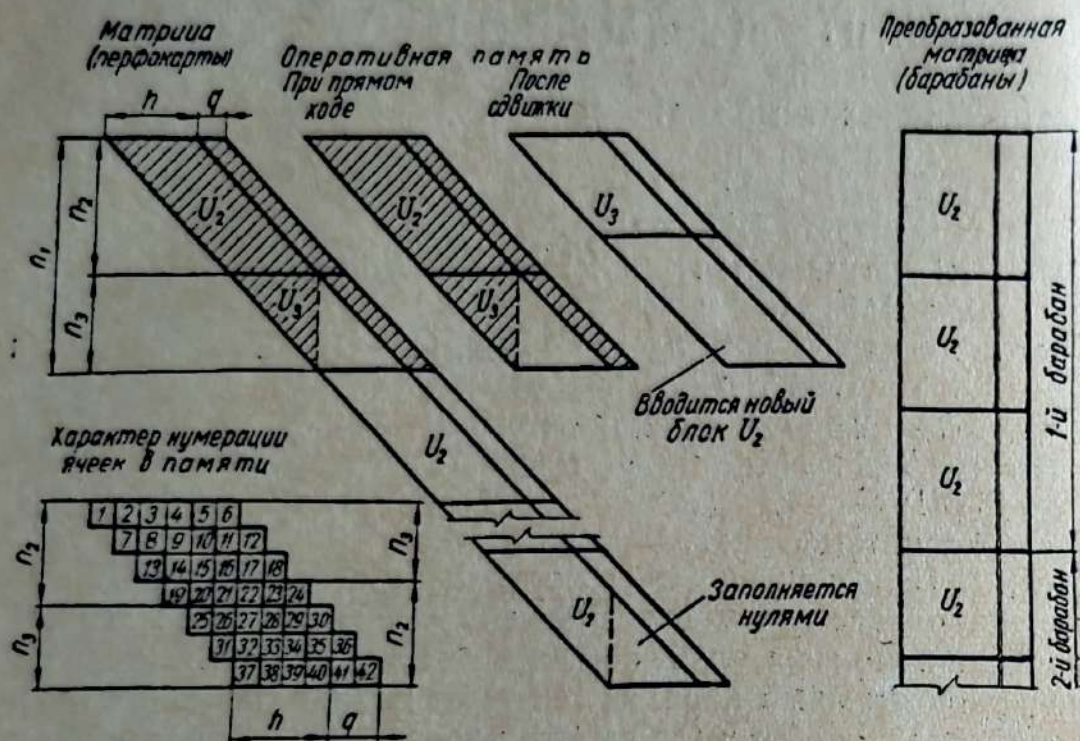


Рис. 37.

прямого хода и исключения n_2 неизвестных блок u_2 направляется в барабаны. Затем блок u_3 сдвигается, на освободившееся место вводится новая порция чисел u_2 и снова выполняется прямой ход. В результате последовательного выполнения прямого хода в оперативной памяти и записи на барабаны блоков u_2 вся преобразованная матрица оказывается на барабанах. Далее матрица направляется частями из барабанов в оперативную память, выполняется обратный ход и вычисляются неизвестные. После печати каждая порция неизвестных сдвигается в памяти подобно тому, как это делалось при прямом ходе, но не вверх, а вниз.

Величина $n_3 = h - 1$; n_1 — максимальный порядок матрицы, который можно поместить в оперативную память. Отсюда $n_2 = n_1 - n_3 = n_1 - (h - 1)$. Следовательно, если порядок ленты h боль-

ше $n1$, то воспользоваться предлагаемым способом нельзя. По изложенной схеме в оперативную память вводится не только заштрихованная, но и дополнительная часть матрицы так, чтобы блок $u3$ имел форму параллелограмма.

Максимальный порядок ленты при одной графе свободных членов можно определить по формуле $v = n1(h+1)$, если вместо $n1$ подставить h , а вместо v — полезный объем оперативной памяти. Для машины М-20 максимальная величина h будет равна 61 (120 чисел в уравнении).

Был разработан также второй вариант программы, схема которого сводится к следующему. Матрица системы уравнений (рис. 38) разбивается на равные блоки $u2$, а последний блок $u3$ представляет собой треугольник. Если разделение на равные блоки невозможно,

то добавляется несколько уравнений в начале системы, состоящих только из главного коэффициента (остальные коэффициенты — нули) и не оказывающих влияния на решение системы.

Блок $u2$ и блок свободных членов $u2q$ считываются в оперативную память машины (или формируются в оперативной памяти по специальной программе); нижняя часть матрицы в оперативной памяти — блок $u3$ и блок свободных членов $u3q$ — очищаются. Выполняется прямой ход до строки $n2$; при этом заполняется поле блоков $u3$ и $u3q$, т. е. получают те числа, которые должны быть сложены с соответствующими числами следующих блоков $u2$ и $u2q$. Преобразованные блоки $u2$ и $u2q$ записываются на магнитный барабан, а на их место вводятся с перфокарт новые блоки $u2$ и $u2q$, к которым прибавляются числа блоков $u3$ и $u3q$, после чего блоки $u3$ и $u3q$ в оперативной памяти очищаются и повторяется цикл вычислений прямого хода.

Такой процесс происходит до тех пор, пока не будут записаны на барабаны все блоки $u2$ и $u2q$. Последний треугольник матрицы вводится полностью, причем, если блок $u3$ занимает больше

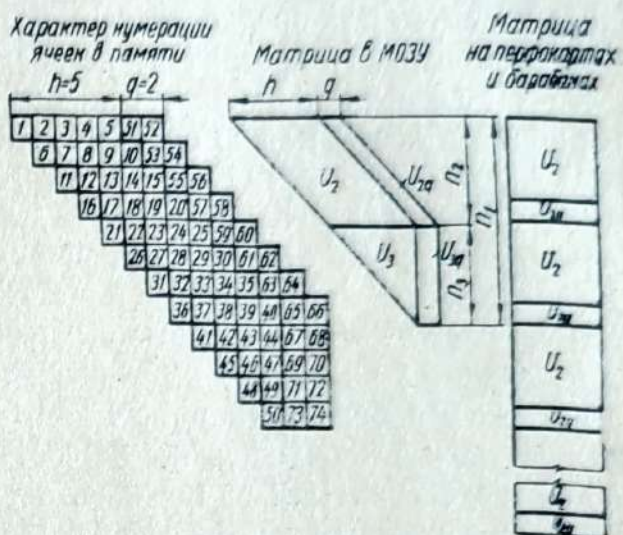


Рис. 38.

места в оперативной памяти, чем блок v_2 , то ввод блока v_3 производится частями по количеству чисел в блоке v_2 .

После ввода последнего треугольного блока для него производится прямой ход от строки $n_2 + 1$ до n_1 .

Затем производится обратный ход для поочередно считывающихся с барабана блоков v_2 и v_{2q} , начиная с последних. Неизвестные помещаются на места свободных членов и сразу выдаются на печать. Сначала печатаются неизвестные для последнего блока v_2 , затем для предпоследнего и т. д.

Второй вариант программы несколько сложнее первого, но допускает введение в машину более широких лент. Максимальный порядок ленты можно определить по формуле $v = n_1 \times (h + 1) - 0,5h \times (h - 1)$, если вместо n подставить h , а вместо v — полезный объем оперативной памяти. Для машины М-20 максимальная величина h при этом получается равной 84 (167 чисел в уравнении).

Надо сказать, что указанные ограничения величины порядка ленты значительно слабее ограничений, связанных с емкостью магнитных барабанов. Например, для бигармонических уравнений предельный порядок ленты 84 соответствует сетке 41×41 и системе уравнений 1681-го порядка, а такие системы даже при ленточной записи не вмещаются в барабаны.

Для того чтобы сделать возможный порядок решаемой системы практически неограниченным, необходимо использовать магнитные ленты. В этом случае целесообразно видоизменить схему расчета с тем, чтобы снять также ограничения в отношении порядка ленты. Примем следующую последовательность решения системы линейных уравнений с использованием магнитных лент:

а) расширенная матрица системы уравнений делится на заданное количество s одинаковых блоков v_2 таким образом, чтобы в оперативной памяти можно было разместить два таких блока. Если порядок системы n на s не делится, то всегда можно добавить в начале системы некоторое число фиктивных нулевых уравнений (с единицами на главной диагонали) с таким расчетом, чтобы увеличенный порядок системы делился на s ;

б) матрица системы уравнений записывается на магнитные ленты;

в) в оперативную память вызывается с магнитной ленты первый блок v_2 , который назовем исключаемым блоком.

При полном заполнении системы уравнений потребовалось бы последовательно вызывать в память другие блоки v_2 , которые назовем сопрягаемыми, и выполнять прямой ход, отправляя

затем преобразованные части матриц снова на магнитные ленты. Для ленточных матриц сопрягаемые блоки вызываются до тех пор, пока соблюдается условие $N \times v_2 \leq v_3$, где N — счетчик вызываемых блоков. Дальнейший вызов блоков при $N \times v_2 > v_3$ не нужен, так как остальное поле матрицы остается без изменения (рис. 37). После окончания цикла для N исключаемый блок отправляется на соответствующую зону ленты. Пункт v циклически повторяется для второго исключаемого блока (сопрягаются третий, четвертый, ..., s -й блоки), третьего исключаемого блока (сопрягаются четвертый, пятый, ..., s -й блоки) и т. д. до последнего s -го исключаемого блока.

Такая схема расчета позволяет использовать программу для общего случая — полностью заполненных матриц, а для ленточных матриц получить значительное сокращение числа обращений к магнитным лентам и объема вычислительной работы. Для матриц очень высоких порядков резкое уменьшение количества вычислений имеет первостепенное значение даже при использовании быстродействующих вычислительных машин.

Заметим, что логика счета почти не усложняется, добавляется лишь одна проверка `if  $N \times v_2 \leq v_3$  then ... else ...`

Первый вариант программы с использованием магнитных барабанов является частным случаем последней более общей программы. При этом сопрягается только один блок, что при барабанах вполне достаточно (лимитирует емкость барабанов, а не другие ограничения).

Наряду с описанными был изготовлен также вариант программы с помощью транслятора ТА-2 с использованием барабанов и магнитных лент. Для этого достаточно поместить в программах из VI.3 — VI.6, переписанных на входном языке, соответствующий комментарий для исполнения, указав максимальную длину массива, скажем, несколько десятков тысяч. Но в этом случае распределение и обмены между различными видами памяти организует транслятор стандартным образом, не приспособленным к индивидуальным особенностям данной задачи. Поэтому счет по программе, полученной автоматически, производится значительно медленнее, чем по оптимальным программам, описанным выше.

Среди других программ для метода исключения укажем программу блочного исключения неизвестных для матриц трехклеточной структуры (так называемый метод прогонки, широко применяемый при решении систем уравнений метода сеток) с организацией обмена между оперативной памятью и магнитными

барабанами. Эта эффективная программа, разработанная в Институте механики АН УССР М. И. Длугачем и А. И. Шинкарем [35], применялась ими для расчета методом сеток пластин и оболочек с отверстиями; при этом составлялись и решались системы уравнений с сотнями и тысячами неизвестных *.

В сборнике [143] сообщается о составлении эффективных американских программ для решения систем уравнений весьма высоких порядков методом исключения (статьи С. У. Мак-Кормика и Р. Н. Уайта).

VI. 8. Инженеру-проектировщику пока редко приходится иметь дело с очень большими системами уравнений. Главным образом потому, что для упрощения расчета сложная пространственная система обычно разлагается на плоские. Если же такое разложение затруднено, то пользуются различными упрощенными методами расчета пространственных систем. В большинстве случаев представление о допущенной погрешности носит предположительный характер; ввиду отсутствия точного расчета не с чем сравнивать.

Теперь, с появлением мощных ЭЦВМ возникает реальная возможность точного (в пределах принятых исходных предпосылок) расчета сложных пространственных конструкций, в связи с чем появляется необходимость в решении весьма больших систем — с сотнями и тысячами неизвестных.

Точный метод расчета можно использовать для двух целей: во-первых, для определения усилий в элементах пространственной системы, если отсутствует надежный приближенный метод расчета; во-вторых, для использования приближенных методов расчета и установления связанных с ними погрешностей.

Хотя применение все более совершенных ЭЦВМ и увеличивает эффективность точных методов для расчета пространственных систем, но построение простых и надежных приближенных методов никогда, по-видимому, не потеряет своего значения.

При решении систем линейных уравнений важно существенно упростить подготовку исходных данных и ввод их в машину. Желательно, а для больших задач необходимо, чтобы система фор-

* Заметим, что для многоклеточных или коквазидиагональных матриц (под клетками понимаются квадратные матрицы меньшего размера, входящие в большую матрицу, квазидиагонали — диагонали клеток) можно разработать также общую программу, подобную рассмотренным программам для ленточных матриц. В этом случае порядок ленты равен числу коквазидиагоналей, расположенных справа от главной диагонали, включая последнюю. Для частного случая трехклеточных матриц $h=2$, пятиклеточных — $h=3$ и т. д.

мировалась автоматически. Это осуществляется в программах метода сил и деформаций, которые будут рассмотрены дальше.

Существуют также многообразные приемы формирования систем уравнений для разностных (сеточных) аппроксимаций различных типов дифференциальных уравнений в частных производных [35, 37].

Для малозаполненных матриц применяются специальные приемы сокращения исходной информации (VI. 12).

О решении систем линейных уравнений циклическими методами

VI. 9. В III. 26 рассматривалось сравнение методов исключения и итерационных при ручном выполнении расчета на клавишных автоматах. Достоинства метода исключения возрастают при использовании ЭЦВМ, его можно считать универсальным для расчета статически неопределимых систем: преобразование матрицы системы уравнений (прямой ход) выполняется только один раз и используется при произвольном количестве правых частей (тогда как в итерационных методах процесс расчета выполняется заново для каждой правой части); при той большой точности вычислений, которая обеспечивается ЭЦВМ (для М-20 — девять верных значащих цифр), погрешности округления практически не сказываются на конечных результатах (если основная система выбрана относительно правильно) даже при очень высоких порядках системы; наконец, время счета, как правило, меньше, чем в итерационных методах.

Эти преимущества отмечены в работе [9] и для конечно-разностных аппроксимаций, хотя основное внимание отведено в ней итерационным методам: «Такие прямые методы очень желательны по двум причинам. Во-первых, можно ожидать, что они намного быстрее итерационных методов, так как никакой сходимости не требуется. Во-вторых, они позволяют решать системы $AU=B$, с различными правыми частями, не проделывая всю работу заново для каждого B_i » (стр. 233). Аналогичная точка зрения высказывается в работах [20, 36, 143].

Недостаток методов исключения в применении к ЭЦВМ по существу только один: возможности экономии памяти для малозаполненных (редких) матриц ограничены, поскольку многие нулевые элементы в процессе прямого хода заполняются числами.

Из вышеизложенного видно, что для матриц определенных структур, распространенных в строительной механике, имеется возможность эффективно экономить память и уменьшить объем

вычислений. Если в полностью заполненных несимметричных матрицах объем вычислительной работы грубо оценивается величиной $\frac{n^3}{3}$ ([9], стр. 232), где n — порядок систем, то в симметричных ленточных матрицах аналогичная грубая оценка даст $\frac{nh(h+1)}{2}$; с уменьшением порядка ленты h количество вычислений резко убывает. Однако для систем, не имеющих благоприятной структуры, этот эффект отсутствует.

Особенно резко возрастает порядок ленты для пространственных задач. Например, для пространственного рамного каркаса, рассчитываемого методом деформаций, $h=6(m_1 \times m_2 + 1)$, где $m_1 \times m_2$ — количество узлов в горизонтальной плоскости (m_1 и m_2 — количество узлов в одном и другом направлении). То же имеет место и для уравнений Ламе в пространственной задаче теории упругости, если ее аппроксимировать прямоугольной решеткой. Даже при небольшом числе узлов в каждом направлении величина h становится большой. Так, при $m_1 = m_2 = 10$, получим $h=6(10 \times 10 + 1) = 606$; в оперативной памяти машины М-20 можно будет поместить всего 6 строк такой матрицы.

То обстоятельство, что в пределах ленты имеется много нулевых элементов, в методе исключения не используется — для этих элементов приходится резервировать ячейки памяти, так как при преобразовании матрицы (прямом ходе) они заполняются числами.

Преимущество итерационных методов состоит в том, что имеется возможность «сжать» матрицу полностью, записав в память только значащие числа и сведения об их расположении в матрице (VI. 12).

Этим свойством обладает также более широкий класс методов (как итерационных, так и точных), которые будем называть циклическими методами.

Циклические методы характеризуются следующими особенностями:

начиная процесс, исходят из некоторых начальных значений неизвестных, которые затем последовательно изменяются. Как было показано в II. 2, этот процесс можно трактовать как динамическое изменение величин неизвестных во времени. Здесь имеется принципиальное отличие от «статического» метода исключения, в котором сначала устанавливаются зависимости между неизвестными (преобразование матрицы), а неизвестные определяются только в заключительной стадии;

процесс расчета — изменение неизвестных — выполняется в отдельном от матрицы массиве. Сама матрица в процессе расчета остается без изменения — это внешняя среда, к которой должна приспособиться система неизвестных в своем стремлении к равновесному состоянию;

процесс является циклическим, т. е. состоит из определенного количества однотипных шагов. Через некоторые интервалы с помощью тех или иных приемов ускорения сходимости характер процесса может измениться, но это изменение, если оно имеет место, происходит скачком (соответствует ступенчатым механизмам на схеме в II. 2), после чего процесс снова приобретает равномерный, циклический характер.

Для удобства рассмотрения циклические методы разделим на три основные группы.

К первой группе отнесем итерационные методы (их еще называют методами смещений, релаксации и т. д.), в которых процесс последовательных приближений направляется локальными оценками.

К этой группе относятся рассмотренные в II. 1 методы простой итерации и итерации по Зейделю — неизвестное в i -том уравнении изменяется так, чтобы привести к нулю (подавить) невязку в i -том же уравнении. В так называемом методе простой релаксации на каждом шаге уничтожается наибольшая по абсолютному значению невязка. Цикл вычислений состоит в отыскании наибольшей невязки, подавлении ее, определении невязок в остальных уравнениях. Этот метод дает эффект в тех случаях, когда на каждом шаге имеется преобладающая невязка *. Если же этого нет, то такой механизм не ускоряет сходимости, а в некоторых случаях может привести к обратным результатам — подавление наибольшей невязки может привести к пропорциональному увеличению других невязок. В методе групповой релаксации Саусвелла [89, 142] на каждом шаге изменяют группу неизвестных таким образом, чтобы невязки во всех уравнениях имели тенденцию к уменьшению в большей или меньшей степени, а если

* Вычислений в одном цикле значительно больше, чем в методе Зейделя, так как для подавления одной невязки нужно предварительно найти невязки во всех уравнениях и выбрать наибольшую. Это не компенсируется возрастанием темпа сходимости. Поэтому в работе [9] указанный метод не рекомендуется для ЭЦВМ (стр. 267). Этот метод больше приспособлен для ручного счета — определение уравнения, в котором имеется наибольшая невязка, обычно выполняется вычислителем «на глаз», формальная подстановка во все уравнения не производится.

какая-либо из невязок увеличивается, то незначительно (существенно меньше, чем другие невязки уменьшаются). Метод групповой релаксации был ориентирован на ручное выполнение и полагался на интуицию вычислителя. Известны случаи, когда с его помощью были быстро рассчитаны системы с несколькими десятками неизвестных. Г. С. Пропасытина (Киевский автомобильно-дорожный институт), решавшая с помощью групповой релаксации системы более чем из сорока уравнений (для метода сеток), так объясняла процесс своих вычислений: вначале я просматриваю систему уравнений, делю уравнения на характерные группы, пробую различные соотношения между неизвестными, прикидываю (в уме), какие они могут дать невязки, останавливаюсь на определенном соотношении, некоторое время его придерживаюсь, затем.... я «чувствую» систему и когда вижу, что это соотношение между поправками для неизвестных уже не годится, меняю его. Если вспомнить трактовку из II. 3, то видим здесь характерный процесс приспособления (адаптации) расчетного механизма к системе. В нашей практике для некоторых систем удавалось установить приближенное соотношение между неизвестными (названное подавляющим оператором), придерживаясь которого на протяжении всего процесса итерации получали резкое ускорение сходимости. В этих случаях нетрудно было разработать соответствующий алгоритм расчета. Однако хотелось бы иметь более гибкий аппарат, который бы заменял фразу — «чувствую систему», но об этом позже.

Ко второй группе отнесем разнообразные методы спуска (их еще называют методами скорейшего спуска, хотя на практике определение «скорейший» не всегда оправдывается), в которых процесс последовательных приближений направляется интегральными оценками. Обычно используются экстремальные свойства некоторого выражения, составленного для системы в целом; например, условия минимума потенциальной энергии системы, среднеквадратичного отклонения невязок и т. д. Сходимость процесса в этом методе (имеющая тип геометрической прогрессии) замедляется с увеличением порядка системы, а для весьма больших систем и при не очень благоприятных соотношениях между коэффициентами решение трудно получить даже на быстродействующих вычислительных машинах.

С точки зрения теории динамических систем Эшби [133] это понятно: с помощью жесткого простого механизма, учитывающего лишь интегральные, самые общие свойства, трудно подчинить сложную, большую систему. В данном случае возможны ложные

минимумы, которым соответствуют значения неизвестных, не удовлетворяющие системе уравнений.

Первая и вторая группа методов являются итерационными: даже при отсутствии ошибок округления всегда имеется некоторая погрешность, связанная с пренебрежением остаточным членом; точное же решение является пределом бесконечной последовательности.

К третьей группе отнесем точные методы сопряженных градиентов, основанные на последовательной ортогонализации векторов*.

Вычислительный механизм их близок к итерационному, но точное решение получается в результате конечного числа шагов (если считать, что ошибки округления отсутствуют). Несмотря на то, что эти методы (их иногда называют методами конечных итераций) являются точными, применение их для решения больших систем затруднительно: числа, получаемые в процессе расчета, могут резко отличаться по величине, в результате чего малые числа могут превратиться в машинный нуль или большие числа — выйти из разрядной сетки машины; количество вычислений является большим — в несколько раз больше, чем в методе Гаусса, в памяти требуется хранить не один вектор неизвестных, а несколько векторов.

Хотя метод сопряженных градиентов непосредственно предназначен для решения симметричных систем, но его можно применить и для решения несимметричных систем, выполняя в неявном виде трансформацию Гаусса.

Ряд алгоритмов для циклических методов, записанных на языке инженерного программирования, приведен в работе [108] (методы проекций, ** скорейшего спуска, сопряженных градиентов).

Эти алгоритмы составлены также на языке АЛГОЛ-60 и реализованы на машине БЭСМ-2М.

* В работе [121] дается следующее определение: «Методами сопряженных направлений для решения систем линейных уравнений $Ax=F$ называются методы, в которых решение x представляется в виде линейной комбинации векторов, ортогональных в некоторой метрике, так или иначе связанных с метрикой системы. ...Каждый отдельный метод характеризуется выбором метрики и выбором системы векторов, подвергающихся ортогонализации» (стр. 433).

** В работах [43, 62] метод проекций трактуется как метод спуска (используется экстремальная теорема для гиперэллипсоида). Вместе с тем можно показать, что метод проекций эквивалентен методу Зейделя для системы, образованной из данной с помощью второй трансформации Гаусса, по существу выполняемой в неявном виде. Подобные связи между различными по форме методами встречаются нередко, но они не всегда видны.

VI. 10. Здесь ограничимся тем, что приведем запись на АЛГОЛе программы для метода Зейделя с включением так называемого релаксационного параметра ω , смысл которого будет разъяснен в следующем параграфе.

Примем следующие обозначения:

n — порядок системы;

$x[1 : n]$ — массив, в который записываются последовательно вычисляемые значения неизвестных;

$S[1 : L]$ — массив, в который вводят исходные данные для формирования матрицы;

L — длина массива S ;

$A[1 : n]$ — рабочий массив, в котором поочередно формируется i -тая строка матрицы коэффициентов ($i=1, 2, \dots, n$);
procedure формирование A ; ...; — неописанная процедура формирования массива A . Описание этой процедуры зависит от выбранного способа сжатия матрицы (VI. 12);

$B[1 : n]$ — массив для свободных членов *;

m — счетчик циклов расчета;

i — счетчик строк матрицы;

k — счетчик элементов в строке;

$eps(\epsilon)$ — допустимая погрешность;

$omega(\omega)$ — релаксационный параметр (VI. II).

Если $\omega=1$, то получим программу для обычного метода Зейделя;

c — признак, вырабатываемый в процессе расчета. Если после выполнения цикла расчета $c=0$, то это значит, что найденные значения неизвестных удовлетворяют всем уравнениям и процесс прекращается;

b — результат подстановки неизвестных в i -тое уравнение.

Имея в виду приведенные обозначения, читателю будет нетрудно проследить логическую схему помещенной ниже процедуры (или отладить ее на простом примере).

Procedure решение систем линейных уравнений методом Зейделя; **begin comment** в процедуре используется релаксационный параметр ω , вводимый как исходная величина. Начальные значения неизвестных вводятся как исходные данные, в простейшем случае они равны нулю;

integer n, l, m, i, k, c ; **input** (n, L);

begin real $b, eps, omega$;

* Предполагается, что свободные члены находятся в левой части системы уравнений.


```

array A, B, x[1 : n], S[1 : L];
procedure формирование A; ...;
input (eps, omega, B, x, S);
m := 0; c := 1;
for m := m + 1 while c = 1 do begin
  for i := 1 step 1 until n do begin
    формирование A; b := 0;
    for k := 1 step 1 until n do;
      b := b + x[k] × A[k];
    if b + B[i] > eps then c := 1;
    else begin c := 0; go to M1 end;
    x[i] := x[i] - (B[i] + omega × b) / A[i];
  M1 : end i end m;
output (x, m);
end end процедуры.

```

VI. 11. Для практического применения циклических методов необходимо иметь ясные ответы на два вопроса. Во-первых, нужно знать условия, при которых итерационный процесс сходится, а для точного метода быть уверенным в его надежности — возможно ли пренебречь ошибками округления, не происходит ли потеря значащих цифр из-за большой разницы в числах и т. д. Во-вторых, нужно быть уверенным в том, что процесс сходится достаточно быстро и что решение можно получить в приемлемые для практики сроки.

Первому вопросу посвящено много математических исследований [6, 121]. Было доказано, что существуют универсальные итерационные алгоритмы, обладающие неизбежной сходимостью. Например, процесс Зейделя обладает неизбежной сходимостью для симметричных положительно определенных матриц, а этим свойством обладают матрицы методов сил и деформаций (при правильной постановке задачи). Однако, если доверчивый инженер попытается применить метод Зейделя для систем, в которых отсутствует диагональное преобладание (т. е. главные коэффициенты равны или меньше побочных), то будет разочарован — процесс будет сходиться настолько медленно, что решение практически невозможно будет получить даже на быстродействующих вычислительных машинах. Известно, например, что методом Зейделя не удалось получить решение системы уравнений для разностной аппроксимации бигармонической задачи, хотя при вариационной постановке матрица получается симметричной и положительно определенной (в работе [33] показано, что разностная аппроксимация бигармонической задачи эквивалентна ре-

шению некоторой стержневой системы методом сил, в работе [37] — методом деформаций), а соотношение коэффициентов относительно благоприятно (главный коэффициент больше каждого из побочных).

Доказано, что итерационный процесс в методах скорейшего спуска имеет характер геометрической прогрессии и обладает неизбежной сходимостью для любых систем уравнений. Если проверить это утверждение на системах с несколькими десятками неизвестных, большей частью получим удовлетворительное решение. Но попробуйте решить неблагоприятную систему (отсутствие диагонального преобладания) очень высокого порядка (с сотнями неизвестных). В лучшем случае, придется затратить во много раз больше машинного времени, чем при решении той же системы методом исключения с использованием внешней памяти, в худшем же случае, решение вообще невозможно будет получить — знаменатель прогрессии будет близок к единице.

В методе сопряженных градиентов решение теоретически должно быть получено за конечное число шагов, но картина получается та же, что и для методов спуска: для сравнительно небольшой системы все обстоит благополучно, для больших же систем время счета резко возрастает, требуется повышенная точность и т. п.

Для этих методов известны решения больших систем, но определенного вида, подходящих для данного расчетного механизма.

Рассмотрим теперь второй вопрос, представляющий наибольший интерес для инженера — выяснение скорости сходимости, возможности быстро получить решение по тому или иному циклическому методу. Для этого разработан специальный аппарат численного анализа матриц и их собственных значений. Так как определение собственных значений представляет не менее сложную задачу, чем решение систем уравнений, то этот аппарат используется для ориентировочной оценки скорости сходимости того или иного класса матриц. Эти оценки сугубо приближенные. В пределах данного класса собственные значения для отдельных числовых примеров могут сильно отличаться, и вполне определенные общие выводы сделать трудно — слишком много может быть исключений из правила.

И здесь выясняется, что универсальных итерационных алгоритмов по существу нет; каждый алгоритм имеет определенную относительно узкую область применения и его еще нужно обычно облагородить приемами ускорения сходимости, приспособ-

собленными для данной конкретной задачи*. Возвращаясь к трактовке из II.3 можно сказать, что нужен регулятор, приспособляющий данный механизм к конкретной системе уравнений (внешней среде).

И действительно, наиболее удачными оказывались те методы, в которых удавалось найти подобный регулятор. Например, для распространенной в теории упругости задачи Лапласа, при ее решении методом конечных разностей, в качестве регулятора используется релаксационный параметр ω [9], значение которого берется в пределах $0 < \omega < 2$. Действие параметра ω можно проследить в процедуре, приведенной в предыдущем параграфе:

при изменении неизвестного соответствующая невязка не приводится к нулю, а лишь уменьшается по абсолютному значению так, чтобы остальные невязки не выходили из определенных пределов, а в целом все невязки имели тенденцию к уменьшению. Если $\omega = 1$, имеем обычный процесс Зейделя, $0 < \omega < 1$ соответствует так называемой нижней релаксации, $1 < \omega < 2$ — верхней релаксации. В работе [9] приведена библиография математических работ, посвященных методам последовательной верхней релаксации, (которым соответствует процедура из VI.10. при выборе $1 < \omega < 2$), изложены положения теории Янга-Франкела для этих методов и даны рекомендации по определению оптимальной величины релаксационного параметра.

Если использовать оптимальное значение параметра ω_{opt} или величину, близкую к нему, то скорость сходимости увеличивается в десятки раз.

Приведены сведения о решении задачи Лапласа с двумя тысячами внутренних точек (две тысячи неизвестных!) с сокращением невязки в 10^{-6} раз. При $\omega = 1$ решение было получено за 5 ч, а при ω , близкой к оптимальной (1.87), — всего за 13 мин. Речь идет об ускорении сходимости сходящегося процесса, т. е. нужно быть уверенным в том, что процесс теоретически сходится. Поэтому формальное исследование сходимости процесса итерации, даже безотносительно к его скорости, имеет важное практическое значение — это почва, на которой могут развиваться различные варианты с ускоренной сходимостью.

В работе [32] и в диссертации Ф. А. Романенко, выполненной

* Только для определенного узкого класса матриц, в которых абсолютное значение главного коэффициента значительно больше суммы абсолютных значений побочных коэффициентов, процесс сходится настолько быстро, что для решения можно применить даже столь несовершенное вычислительное средство, как логарифмическая линейка (см. гл. II).

под руководством проф. Д. В. Вайнберга, методы верхней и нижней релаксации применены для решения бигармонической задачи, заменяемой двумя задачами: Лапласа — для нее применена верхняя релаксация и Пуассона — для нее применена нижняя релаксация. Найдены оптимальные параметры верхней релаксации для многих видов односвязных и неодносвязных областей при разной густоте сетки, при этом скорость сходимости итерационного процесса возросла от 5 до 20 раз. Максимальный порядок решенной системы был равен 700 (программа допускает решение системы с 1200 неизвестными).

Итак, можно добиться резкого ускорения сходимости процесса, но для этого нужно уметь находить с достаточной точностью релаксационный параметр ω .

Вазов и Форсайт предложили три метода, в том числе метод проб, определения ω_{opt} ([9], стр. 406—410). Хотя указывается, что «ни один из этих методов нельзя считать окончательным решением вопроса», но для ряда задач удалось составить программы автоматического определения ω_{opt} и реализовать их на машинах.

Можно указать, в частности, на составленную Ф. А. Романенко на основе метода проб программу определения оптимального релаксационного параметра при решении на машине М-20 уравнений Лапласа для прямоугольных областей. Таким образом, по крайней мере для данного узкого класса задач, осуществлена автоматизация процесса приспособления (схема на рис. 1) и выработки регулирующего механизма.

Так обстоит дело с наиболее изученной и простой задачей теории упругости — уравнением Лапласа. Для более сложных задач уже не удастся, по-видимому, все свести к одному релаксационному параметру (впрочем, и для задачи Лапласа один релаксационный параметр не всегда удовлетворяет, делаются попытки ввести два параметра); возможно придется ввести несколько параметров для различных частей матриц, на различных этапах процесса использовать групповую релаксацию и другие пути.

Очевидно, на данной стадии изученности вопроса инженеру можно рекомендовать пользоваться циклическими методами только в тех случаях, когда их применение не вызывает сомнений, т. е. имеются не только общие теоретические исследования, но значительный опыт решения на ЭЦВМ по соответствующим программам систем высоких порядков именно такого вида, которые предстоит решить в данном, конкретном случае.

В некоторых задачах целесообразно использовать смешанное решение, аналогичное тем, которые рассматривались в гл. II. В методе сеток смешанному решению соответствует релаксация по линиям.

Для более широкого применения циклических методов при решении систем линейных уравнений высоких порядков на ЭЦВМ, по-видимому, целесообразно использовать два пути.

Первый путь состоит в разработке специальной интерпретирующей системы, в которую будут собраны проверенные циклические методы с указанием границ применения для каждого из них. Такая система создается для машины М-20 в лаборатории тонкостенных и пространственных конструкций Киевского инженерно-строительного института под руководством проф. Д. В. Вайнберга [12]. В работе [9] сообщается о создании библиотеки программ на языке ФОРТРАН-II и машинных программ для решения систем эллиптических и параболических уравнений в частных производных методом сеток.

Второй путь заключается в разработке алгоритма поиска регулирующего механизма для различных циклических методов; задача эта весьма сложная, требующая обширных исследований.

VI. 12. Как указывалось, циклические методы позволяют максимально сжимать информацию, не записывая в память нулевые элементы матриц. В соответствии с терминологией работы Вазова и Форсайта будем называть плотностью матрицы α отношение количества ненулевых элементов к общему числу N элементов матрицы. У квадратных матриц $N=n^2$, где n — порядок матрицы. Для полностью заполненных матриц $\alpha=1$. Если матрица вводится в память машины полностью, то она называется «хранящейся»; если матрица формируется в памяти в соответствии с более краткой информацией — «порождающейся».

Большие системы уравнений для методов строительной механики (и родственного им метода сеток в задачах теории упругости) имеют обычно малую плотность (порядка 0,01—0,1); тем меньшую, чем выше порядок системы. Информацию о редкой матрице можно сжать и поместить в относительно небольшом количестве ячеек. Затем на основе этой информации в рабочих ячейках последовательно формируются те элементы матриц, которые участвуют в расчете. В программе из VI. 10 применена неопи-санная процедура последовательного формирования строк матрицы в одном и том же рабочем массиве $A[1:n]$ (для ленточных матриц $A[1:h]$). Характер этой процедуры зависит от способа представления в машине матрицы S .

Различные способы сжатия матриц строительной механики изложены в работах [37, 105—109]. В основном они сводятся к следующим:

1. Информация о расположении чисел в матрице задается с помощью так называемых логических шкал — это ячейки, в двоичных разрядах которых записываются единицы (соответствуют значащим числам) или нули (соответствуют нулевым элементам). Отдельным массивом задаются значащие числа. Специальная подпрограмма просматривает шкалу для данной строки (используются процедуры — коды выделения и сдвига) и формирует рабочий массив $A[1:n]$, в соответствующие ячейки которого записываются значащие числа и нули.

Для шкал требуется запасти n^2/c ячеек, где c — количество разрядов в одной ячейке (например, для машины М-20 величина $c=45$). Если учесть то обстоятельство, что многие строки матрицы подобны (при сдвиге совпадают), то шкалы можно задать только для разнотипных строк. Тогда количество ячеек для шкал будет равно $n \times m/c$, где m — число различных типов строк. Отдельно даны указатели, в которых записаны номера шкал для строк матрицы.

Недостатком этого способа является трудоемкость подготовки шкал — их сначала представляют как двоичные числа, а при записи на бланки кодируют восьмеричными цифрами (триады двоичных чисел заменяют восьмеричными числами, см. V.3). Затем необходимо записать указатели — адреса шкал для всех строк.

2. Для каждой строки (или графы) матрицы задается оператор — это несколько ячеек, в которых записываются номера неизвестных для нулевых элементов. В одной ячейке можно записать несколько номеров неизвестных (например, для трехадресных машин — в первом, втором и третьем адресах). Для повторяющихся строк в операторе указывается признак повторения (например, единица в 45-м разряде указывает, что числа для данного оператора такие, как для предыдущего). Это дает возможность поместить в памяти значащие числа только для неповторяющихся строк. Этот способ задания информации широко применяется нами в программах метода деформаций (сил) для указания номеров неизвестных, влияющих на стержень (участок) статически неопределимой системы, и будет подробнее описан в дальнейшем. Операторный способ тем эффективнее, чем меньше значащих чисел в строке (графе) матрицы. Подготовка операторов проще, чем шкал в первом способе.

Для уменьшения количества операторов можно использовать зависимости между ними, если они существуют. Например, во многих задачах группа операторов может быть порождена одним оператором прибавлением заданного параметра сдвига.

3. Следующий способ состоит в использовании структурных параметров матриц. Структурные параметры использовались нами в методах исключения (порядок ленты h , порядок пустого треугольника $n1$), но они описывали структуру преобразованной матрицы и это ограничивало возможности экономии памяти.

Для циклических же методов матрица остается без изменения и достаточно описать структуру исходной матрицы. Например, в строительной механике (и в методе сеток) часто встречаются матрицы полиленточной структуры (рис. 39). Для исчерпывающего описания такой матрицы достаточно задать параметры $h_1, n_1, h_2, n_2, h_3, n_3, \dots, h_t, n_t$ (эти величины удобно поместить в массиве $h[1:2 \times t]$, где t — число лент) и значащие числа в пределах всех лент (ленты обычно полностью заполнены). Значащие числа нумеруются по строкам аналогично рис. 35. Описание матрицы с помощью структурных параметров значительно проще, чем в предыдущих способах. Заметим, что в методе исключения использование полиленточной структуры для экономии памяти мало эффективно, так как все поле матрицы между лентами, за исключением треугольников в верхней части, ограниченных пунктиром на рис. 39, в процессе прямого хода заполняется числами. Однако эту структуру можно использовать для экономии ввода с последующим формированием матрицы во внешней памяти.

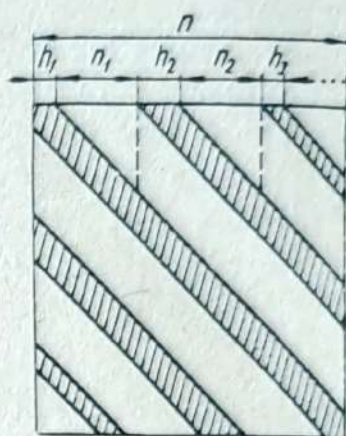


Рис. 39.

4. Наиболее экономным и простым является способ задания информации на основе зрительного образа стержневой системы (VII. 21), а для метода сеток — информации, необходимой для построения сетки [9, 35].

Обобщенная программа расчета стержневых систем методом сил

VI. 13. Предпосылки алгоритма метода сил были изложены в IV. 3 — для определения перемещений (коэффициентов системы уравнений) применялась обобщенная формула (1). Можно бы-

ло бы, конечно, компактную схему алгоритма, разработанную для клавишных автоматов, без всяких изменений переписать на языке АЛГОЛ-60 и реализовать на ЭЦВМ. Однако такая программа не будет оптимальной, так как придется хранить в памяти много двумерных массивов чисел (1, 2, 3, 4, A, B, B₁, B, 5, 6), вследствие чего даже относительно небольшие задачи нельзя будет разместить в оперативной памяти. Поэтому применим другой подход: откажемся от компактной схемы вычислений, не дающей преимущества для ЭЦВМ, и от матричной записи усилий в оперативной памяти.

Для определения слагаемых перемещений применим формулу (6) из IV. 3:

$$v = \int \frac{S_i S_k dl}{c} = \frac{l}{6 \times c} [2 \times (i_1 \times k_1 + i_2 \times k_2) + (i_1 \times k_2 + i_2 \times k_1)]. \quad (1)$$

Здесь i_1 , i_2 и k_1 , k_2 — ординаты эпюр усилий на концах участков интегрирования в i -том и k -том состояниях основной системы, l — длина участка, c — жесткость участка (EI — для изгибающих моментов, EF — для продольных сил, GF/μ — для поперечных сил, GT — для крутящих моментов). В пределах участка любая из эпюр ограничена прямой наклонной линией. Будем считать также, что номера неизвестных, вызывающих усилия на данных участках, засылаются в процессе расчета в рабочий массив R . Задача состоит в том, чтобы составить процедуры для формирования массива R и вычисления величин i_1 , i_2 , k_1 , k_2 от единичных значений неизвестных (номера которых указаны в массиве R) и нагрузки для различных типов основных систем.

Будем считать, что эти процедуры являются переменными и представим их в виде неописанных процедур, для которых в разделе описаний даны только названия, т. е. определено лишь назначение, но не содержание. Информацию об основной системе в общем случае задаем в двух массивах: $H1 [1 : c1]$ — для восьмеричных чисел (описывается типом `integer`) и $H2 [1 : c2]$ — для десятичных чисел (описывается типом `real`), информацию о нагрузках — в массиве $P [1 : c3]$.

В программе представляется свобода в выборе способа задания этой информации.

В обобщенную программу, помимо переменных процедур, входит постоянный основной блок, включающий определение параметров n и h (порядка системы и порядка ленты), составление системы уравнений, процедур прямого и обратного хода

решения системы, вычисление искомых усилий в статически неопределимой системе.

Запись программы в виде основного блока и переменных процедур заменяет блок-схему и является планом для построения библиотеки программ расчета статически неопределимых систем различных видов методом сил. Наиболее общий, универсальный вариант можно получить, приложив к основному блоку процедуру выборки величин i_1, i_2, k_1, k_2 из заданного массива.

Основной блок может быть оформлен независимо в кодах машины. Обращения к неописанным процедурам производятся в соответствии с принятым распределением памяти. В случае необходимости эти обращения впоследствии могут быть изменены. Специализированная программа компилируется из перфокарт основного блока и подпрограмм (процедур) для данного типа систем.

В обобщенную программу метода сил входят следующие этапы:

1) ввод массивов $H1$ и $H2$ исходных данных об основной системе;

2) определение параметров — порядка системы уравнений n и порядка ленты h ;

3) очистка массива A для матрицы системы уравнений;

4) определение поочередно для всех участков усилий на их концах от единичных значений неизвестных и вычисление по формуле (1) слагаемых перемещений с накоплением их в ячейках массива A (накопление матрицы системы уравнений);

5) прямой ход для матрицы с получением преобразованной матрицы A на месте исходной (для экономии памяти используется ленточная структура);

6) ввод массива P исходных данных по очередному загрузению;

7) очистка массива B для свободных членов;

8) определение поочередно для всех участков усилий на их концах от единичных значений неизвестных и от нагрузки, вычисление по формуле (1) слагаемых «грузовых» перемещений с накоплением их в ячейках массива B (накопление свободных членов);

9) прямой ход для свободных членов и обратный ход решения системы уравнений с получением значений неизвестных на месте свободных членов в массиве B , печать значений неизвестных;

10) определение поочередно для всех участков усилий на их концах от единичных значений неизвестных, умножение их на

найденные значения неизвестных с последовательным накоплением в ячейках рабочего массива $S1$; вычисление усилий от нагрузки и суммирование их с усилиями от неизвестных; печать полученных величин усилий.

Этапы 6—10 выполняются для каждого из загрузжений.

Как видим, операция определения усилий на концах участков от единичных значений неизвестных и нагрузки выполняется многократно. Это выгоднее, чем хранить в памяти машины полную матрицу усилий S статически определимой основной системы, т. е. подход другой, чем при составлении аналогичной программы для ручного выполнения расчетов на клавишных автоматах.

Указанные этапы расчета организуются в циклы и объединяются рядом логических условий.

VI. 14. В программе приняты следующие обозначения.

Параметры:

- $e1$ — количество участков интегрирования;
- $e2$ — заданное число групп усилий;
- $c1, c2, c3$ — длины массивов, соответственно, $H1, H2$ и P ;
- h — порядок ленты;
- n — порядок системы уравнений;
- q — число комбинаций загрузки (число граф свободных членов);
- t — максимально возможное число неизвестных, влияющих на участок системы (вызывающих в нем усилия).

Параметры $e1, e2, c1, c2, c3, q$ и t — вводятся, параметры n и h вычисляются во внешнем блоке программы.

Массивы и переменные:

- $H1 [1 : c1]$ и $H2 [1 : c2]$ — для информации об основной системе (целые и вещественные числа);
- $P [1 : c3]$ — для информации о нагрузках;
- $H3 [1 : e1]$ — для длин участков;
- $H4 [1 : e1 \times e2]$ — для жесткостей участков;
- $A [1 : u]$ — для коэффициентов системы уравнений ($u = n \times h - h \times (h - 1) / 2$);
- $B [1 : n]$ — для свободных членов и величин неизвестных;
- $R [1 : t]$ — для номеров неизвестных, влияющих на один участок системы;

$S1 [1 : 2 \times e2]$ — для величин усилий в начале и конце участка (для каждой группы — по два усилия, например: $N_1, N_2, M_{1x}, M_{2x}, \dots$);

$i1, i2$ — для усилий в начале и конце участка в i -том единичном состоянии (в машине переменным $i1$ и $i2$ соответствуют рабочие ячейки, в которые последовательно помещаются усилия разного вида и для различных участков);

$k1, k2$ — для усилий в начале и конце участка в k -том единичном или в грузовом состояниях;

v — для вычисления слагаемых перемещений.

Признаки:

$a1, a2 [1 : e2]$ — для признаков, используемых в процедурах;

$b=0$ — формирование массива A , прямой ход для матрицы;

$b=1$ — формирование массива B , прямой ход для свободных членов, обратный ход;

$b=2$ — определение искомым величин усилий в статически неопределимой системе.

Признак b формируется в программе.

Индексы (счетчики):

$j1$ — порядковый номер комбинации загрузки;

nu — порядковый номер участка;

tu — порядковый номер группы усилий;

$g1, g2$ — порядковые номера ячеек в массиве R ;

i, k — номера граф матрицы A (матрица A — порождающаяся, поскольку используется ленточная структура) или номера неизвестных;

d — номер диагонального элемента матрицы A ;

$h1, d1, m, m1, j$ — дополнительные индексы, используемые в процедурах прямого и обратного хода решения системы уравнений (VI. 4);

$N1, N2$ — переменные, которые служат для связи основной программы с процедурой выборки (VI. 16);

$r1, r2$ — рабочие переменные.

Обобщенная программа метода сил

```
begin integer e1, e2, n, h, q, c1, c2, c3, u,
b, j1, nu, tu, g1, g2, i, k, d, h1, d1, m, m1,
j, r1, r2, N1, N2, t, ...;
real i1, i2, k1, k2, v, ...;
input (c1, c2, c3, e1, e2, q, t);
begin integer array H1[1 : c1], R[1 : t], a1,
a2[1 : e2];
```

Примечания

Значение многоточий
см. V.12;

Ввод параметров;


```

real array H2[1 : c2], H3[1 : e1]
H4[1 : e1 × e2], S1[1 : 2 × e2], P[1 : c3];

```

```

procedure построение массива R; ...;
procedure определение i1 и i2; ...;
procedure определение k1 и k2; ...;
procedure прямой ход (b); ...;
procedure обратный ход; ...;
input (H1, H2, H3, H4, a1, a2);
n := 1; h := 1; b := 0;
for nu := 1 step 1 until e1 do begin,
for g1 := 1 step 1 until t do R[g1] := 0;
построение массива R;
for g1 := 1 step 1 until t do begin
if R[g1] = 0 then go to M1;
if n < R[g1] then n := R[g1];
for g2 := g1 step 1 until t do if R[g2] ≠ 0;
then begin r1 := abs (R[g2] - R[g1]) + 1
if h < r1 then h := r1 end;
M1: end g1 end nu;
u := n × h - h × (h - 1) ÷ 2;
begin array A[1 : u], B[1 : n];
for i := 1 step 1 until u do A[i] := 0;
r1 := n - h + 2;
for j1 := 1 step 1 until q do begin
if b = 2 then b := 1;
for i := 1 step 1 until n do B[i] := 0;
M2: N1 := 0; N2 := 0;
if b = 1 then input (P, a2);
for nu := 1 step 1 until e1 do begin;
for g1 := 1 step 1 until t do R[g1] := 0;
построение массива R;
for mu := 1 step 1 until e2 do begin
if b ≠ 0 then определение k1 и k2;
if b = 2 then begin
r1 := 2 × (mu - 1); S1[r1 + 1] := k1;
S1[r1 + 2] := k2 end;
for g1 := 1 step 1 until t do begin
if R[g1] = 0 then go to M8;
i := R[g1];
if b ≠ 0 then go to M3;

```

Переменные для
 верхних границ мас-
 сивов описаны и вве-
 дены вне блока;
 Перечисление неопи-
 санных процедур;
 Описания процедур
 даны в VI. 4;

Очистка массива R;

Определение пара-
метров n и h;

Верхние границы A
и B вычислены вне
блока;

Очистка массива B;

Ввод информации о
нагрузках;

Определение усилий
от нагрузки и запись
их в массив S1;


```

r2 := 0; if i > r1 then begin
r2 := i - r1; r2 := r2 × (r2 + 1) ÷ 2 end;
d := (i - 1) × h - r2 + 1;

```

Определение номера диагонального элемента матрицы A ($r2$ — поправка для нижней треугольной части матрицы);

```

M3: определение i1 и i2;
    if b = 1 then go to M5 else
    if b = 2 then go to M4;
    for g2 := 1 step 1 until t do begin
    if R[g2] = 0 then go to M6;
    k := R[g2]; if k < i then go to M6;
    определение k1 и k2; go to M5;

```

Усилия от единичных значений неизвестных;

```

M4: S1[r1 + 1] := S1[r1 + 1] + i1 × B[i];
    S1[r1 + 2] := S1[r1 + 2] + i2 × B[i];
    go to M8;

```

Усилия от единичных значений неизвестных для $b = 0$;

```

M5: v := (H3[nu] / (6 × H4[(nu - 1) × e2 +
mu])) × (2 × (i1 × k1 + i2 × k2) + i1 × k2 +
i2 × k1);

```

Определение усилий на концах участка от нагрузки и найденных в массиве B величин неизвестных; Вычисление слагаемых перемещений;

```

    if b = 0 then A[d + k - i] := A[d + k - i] +
v else if b = 1 then go to M7;

```

```

M6: end g2; go to M8;

```

```

M7: B[i] := B[i] + v;

```

```

M8: end g1

```

```

    end mu;

```

```

    if b = 2 then output (S1);

```

```

    N2 := N2 + e2

```

```

    end nu;

```

```

    if b = 0 then begin прямой ход (0);

```

```

    b := 1; go to M2 end;

```

```

    if b = 1 then begin прямой ход (1);

```

```

    Обратный ход; output (B);

```

```

    b := 2; go to M2 end

```

```

    end j1 end end

```

```

    end программы.

```

Формирование массива A (матрицы системы уравнений);

Заполнение массива B свободными членами;

Печать величин усилий на концах участка;

Печать величин неизвестных из массива B .

Некоторые из входных величин ($e1, e2, c1, c2, c3, t$) в специализированных программах могут вычисляться по особым процедурам.

Для проверки логической схемы выполняется отладка программы, например, при следующих значениях входных параметров:

$e1=10, e2=2, t=5, q=2$. Пусть состояния массива R (для номеров неизвестных) при этом будут:
для первого участка ($ni=1$) —

3	5	4		
---	---	---	--	--

для десятого участка ($ni=10$) —

7	8	6	9	
---	---	---	---	--

В отладочном примере состояния массива R для остальных участков можно не задавать, так как обработка информации производится аналогично. Состояния переменных в процессе отладки записываются в таблицу.

О способах задания исходной информации для расчета методом сил

VI. 15. Изложенная программа является абстрактной и изображает собою множество возможных программ библиотеки метода сил. Для получения конкретной программы нужно дать описания переменных процедур (как бы присвоить значения переменным процедурам), которые зависят от выбранного способа задания информации.

Универсальную программу метода сил получим, если в качестве исходных данных примем усилия в основной системе от единичных неизвестных и нагрузки. Достоинство этой программы состоит в том, что она не зависит от вида решаемой системы, недостаток — в необходимости производить расчет основной системы вручную. Несмотря на последнее обстоятельство, универсальная программа метода сил широко применяется на практике для расчета нестандартных задач, если отсутствует специализированная программа, в частности, в ГПИ Укрпроектстальконструкция — для расчета башенных телевизионных опор, сложных систем покрытий, рам с ломаными очертаниями стержней и т. п.

Информация о неизвестных, влияющих на участки системы, представляет собой по сути информацию о структуре матрицы S усилий в основной статически определимой системе. При этом могут быть использованы те же приемы экономии памяти, что и для матрицы системы уравнений (VI. 12).

Если не заботиться об экономии памяти, то можно задать всю матрицу целиком; при этом нулевое усилие кодируется как десятичное число, для него отводится отдельная ячейка.

Можно задавать информацию о значащих числах матрицы S с помощью шкал — ячеек, в разрядах которых записываются единицы (соответствуют значащим числам матрицы S или номерам неизвестных, влияющих на участок) или нули (соответствуют нулевым элементам матрицы S или номерам неизвестных, вызывающих нулевые усилия на концах участков). Можно применить специальный признак повторения строк матрицы S , позволяющий сократить информацию.

Такая форма записи позволяет исключить нулевые элементы в редких матрицах, но для плотных матриц дается лишняя информация: помимо массива усилий, вводится массив шкал. Имеется и другой недостаток — кодирование шкал в двоичной системе и перевод при записи на бланки в восьмеричную систему, что непривычно для инженера и требует большой внимательности.

Более удобно непосредственно указывать номера неизвестных, влияющих на участок, в так называемом операторе — это группа ячеек, в каждой из которых может быть помещено несколько номеров (например, для трехадресных машин — в первом, втором и третьем адресе). Номера неизвестных даются в восьмеричном исчислении*, на окончание оператора указывает условное число u , заведомо превосходящее возможный наибольший номер неизвестного. Например, для машины БЭСМ-2М число $u=3776$, для машины М-20 $u=7776$.

Операторный способ задания неизвестных широко применялся в программах метода сил и особенно метода деформаций. Он удобен для редких матриц. В матрицах же большой плотности для размещения операторов потребуется много ячеек в памяти. В пределе для полностью заполненных матриц каждый оператор будет состоять, например, для трехадресных машин из $n \div 3$ ячеек, тогда как в этом случае массив операторов вообще не нужен.

* Можно записывать их и в десятичной системе, но тогда нужна специальная программа перевода.

Для уменьшения количества операторов можно в операторе участка указывать номера групп неизвестных; например, если основная система образована с помощью разрезов, то указывать номера разрезов. Номера неизвестных, соответствующие разрезам, помещаются в отдельном массиве.

Если количество неизвестных в разрезах является постоянной величиной (для плоских систем — три неизвестных, пространственных систем — шесть неизвестных), то номера неизвестных для разрезов могут быть вычислены по формуле, и дополнительного массива давать не нужно.

Для других задач целесообразно применить следующий прием задания информации о неизвестных. В операторе участка будем записывать номер меньшего неизвестного и количество следующих за ним неизвестных, номера которых изменяются на единицу. Например, если на участок влияют неизвестные $x_5, x_6, x_7, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{25}, x_{26}, x_{27}, x_{30}, x_{31}, x_{32}, x_{33}$, то запись оператора будет иметь следующий вид (для трехадресных машин):

	5	7	25
	6	7776	

Номер меньшего неизвестного 5, за ним следуют 7 номеров, изменяющихся с шагом единица (6, 7, 10, 11, 12, 13, 14), затем разрыв, номер следующего меньшего неизвестного 25, за ним следует 6 номеров, изменяющихся на единицу.

В принципе шаг изменения может быть равен не 1, а любому другому числу, но это должно быть учтено в программе, шаг может быть задан в качестве параметра.

Для того, чтобы еще больше сократить информацию, в коде операции может быть дан специальный признак (например, единица), указывающий, что в операторе за условным числом 7776 даны дополнительные сведения в виде двух чисел, назовем их k_1 и k_2 .

1	...	...	...
	...	7776	k_1
	k_2		

k_1 — количество операторов, следующих за данным и имеющих с ним одинаковую структуру, т. е. одно и то же количество неизвестных с теми же интервалами между ними, k_2 — шаг изменения неизвестных от оператора к оператору. Многоточиями на схеме указаны номера неизвестных *.

Последний способ дает возможность экономно записывать не только редкие, но и плотные матрицы. При полном заполнении достаточно дать для всей матрицы один оператор из двух ячеек следующего вида:

КОП	IA	IIA	IIIA
1	1	$n-1$	7776
	$n-1$	0	

Единица в первом адресе — номер меньшего неизвестного x_1 , $n-1$ во втором адресе — количество следующих за ним номеров неизвестных, изменяющихся с шагом 1 ($x_2, x_3, \dots, x_n$), число 7776 в третьем адресе указывает на окончание номеров неизвестных в операторе. По признаку 1 в коде операции в следующей ячейке помещена информация о других операторах: $k_1 = n-1$ указывает количество следующих за данным однотипных операторов (все операторы системы одного типа); 0 — шаг изменения неизвестных от оператора к оператору (в данном случае все операторы одинаковые).

Можно применить и другие варианты задания операторов, учитывающие те или иные закономерности между номерами неизвестных. Операторы помещаются в массиве $H1$.

По информации, указанной в операторе, формируется рабочий массив R для одного участка, имеющий длину t . В каждой ячейке этого массива помещается номер одного неизвестного. t — входной параметр — максимально возможное для данной задачи количество неизвестных, влияющих на один участок. После обработки каждого участка массив R очищается и в него записываются номера неизвестных для следующего участка (см. программу в VI. 14).

Поскольку в операторе в одной ячейке записано несколько неизвестных, то при извлечении соответствующих чисел и запол-

* Заметим, что этот прием может быть эффективно использован для сжатия матриц систем уравнений метода сеток, для которых характерны подобные закономерности между неизвестными.

нении массива R приходится выполнять операции выделения части кода, сдвига и т. п., зависящие от конкретной машины. Для описания подобных операций на языке АЛГОЛ-60 применяется процедура-код. Поэтому процедура построения массива R оформляется при изложении программы на входном языке для данной машины. В ствлеченной же программе, предназначенной для публикации, достаточно привести неописанный оператор этой процедуры.

Заметим, что в машинном коде имеет смысл составить несколько вариантов процедуры заполнения массива R для различных способов задания информации и применять тот из них, который более всего соответствует характеру задачи. Предписание машине о выборе процедуры-кода (хранящейся во внешней памяти) можно задавать с помощью признака. Так, можно условиться, что значение признака $\lambda = 0$ соответствует использованию шкал, $\lambda = 1$ — операторов участков, $\lambda = 2$ — операторов с учетом существующей закономерности между ними и т. д. Содержимое ячейки λ задается как исходная величина.

Процедуры выборки величин $i1$ и $i2$ ($k1$ и $k2$)

VI.16. Рассмотрим процедуры определения усилий $i1$ и $i2$ ($k1$ и $k2$) в универсальной программе, т. е. выборку чисел из заданного массива $H2$ (усилия на концах участков от единичных неизвестных) и P (усилия на концах участков от нагрузок)*. Для определения порядковых номеров выбираемых из массивов чисел составлены формулы, в которые помимо параметров и переменных основной программы входят счетчики $N1$, $N2$, начальные значения которых присваиваются в основной программе.

Для составления процедуры выборки необходимо прежде всего условиться о порядке записи чисел в массивы $H2$ и P . Усилия данной группы будем записывать парами (в начале и конце участка): в массив $H2$ — от единичных значений неизвестных,

* Заметим, что в предыдущих наших работах [37, 109] изложен другой вариант программы метода сил, более близкий к матричной записи: в рабочих массивах $B1$ и $B2$ поочередно формируются и обрабатываются строки матрицы S усилий статически определимой системы, для экономии памяти используются шкалы и признаки повторения. Исходной информацией служат массивы шкал и значащих чисел матрицы (от единичных неизвестных) и P (от нагрузок). Нетрудно увидеть, что в новом изложении массив шкал заменяется массивом $H1$ операторов, массивы значащих чисел S и P — массивами $H2$ и P и массивы $B1$ и $B2$ — четырьмя рабочими ячейками $i1$, $i2$, $k1$, $k2$.

влияющих на данный участок, в массив P — от нагрузки. Сначала производится запись усилий первой группы, затем второй, третьей и т. д.

Нумерация групп может быть выбрана любой, но ее следует придерживаться на протяжении всего расчета, как при вводе, так и при выводе. Покажем, например, характер записи для следующих данных: число групп усилий $e2=4$; N — нормальные силы — первая группа ($tu=1$); M_x — изгибающие моменты относительно оси x поперечного сечения участка — вторая группа ($tu=2$); M_y — изгибающие моменты относительно оси y поперечного сечения участка — третья группа ($tu=3$); M_z — крутящие моменты относительно продольной оси z участка — четвертая группа ($tu=4$). Для первых двух участков нумерация усилий в массиве $H2$ показана в табл. VI. 1, в массиве P — в табл. VI. 2. Принято, что на первый участок влияют два неизвестных ($g1=1, 2$), на второй — три неизвестных ($g1=1, 2, 3$). Номера неизвестных даются в операторах стержней (массив $H1$) и последовательно переносятся в массив R .

Для каждой группы усилий введены специальные признаки $a1[tu]$ (для табл. VI.1) и $a2[tu]$ (для табл. VI. 2). Если соответствующий признак не равен нулю, то это значит, что усилия в начале и конце участка одинаковы и вместо пары чисел в табл. VI. 1 (VI. 2) записывается одно число. Эти признаки являются интегральными и относятся ко всем участкам системы. Обычно таким свойством обладают группы нормальных сил и крутящих моментов. Что касается изгибающих моментов, то их значения бывают одинаковыми только на отдельных участках, и чтобы учесть это, требуется введение дифференцированных признаков для участков. В данном случае это нецелесообразно, так как сильно усложнило бы (без существенного эффекта) исходную информацию*.

Для рассматриваемого примера примем следующее содержание массивов $a1[1:e2]$ и $a2[1:e2]$.

$a1$	1	0	0	1
$a2$	1	0	0	1

* Дифференцированные признаки введены для стержней в методе деформаций (VI. 20).

В соответствии с этой информацией для каждого участка для первой и четвертой групп будет введено по одному усилию от единичных и грузовых воздействий. В общем случае все признаки можно принять равными нулю и информацию задавать единообразно для всех групп усилий (как во 2 и 3 группе в рассмотренном примере).

procedure определение $i1$ и $i2$;

begin $N1 := N1 + 1$; $i1 := H2[N1]$;

if $a1[mu] = 0$ **then** $N1 := N1 + 1$; $i2 := H2[N1]$

end процедуры;

procedure определение $k1$ и $k2$;

begin **integer** $r10, r20$; **if** $b = 0$ **then** **begin**:

$r10 := g2 - g1$;

if $a1[mu] = 0$ **then** **begin** $r10 := 2 \times r10 - 1$;

$r20 := r10 + 1$ **end** **else** $r20 := r10$;

$k1 := H2[N1 + r10]$; $k2 := H2[N1 + r20]$;

go to $M1$ **end**;

$N2 := N2 + 1$;

$k1 := P[N2]$; **if** $a2[mu] = 0$ **then**

$N2 := N2 + 1$; $k2 := P[N2]$;

$M1$: **end** процедуры.

Логическую схему процедуры определения $i1$ и $i2$ ($k1$ и $k2$) совместно с основной программой удобно проследить на отладочном примере. Условия даны в табл. VI. 1 и VI. 2.

Описанная выше универсальная программа с включением процедур выборки незаменима в нестандартных случаях.

В специализированных программах процедуры определения $i1$ и $i2$ ($k1$ и $k2$) заключают расчет основной системы на основе использования регулярностей структуры данного типа сооружений.

В виде примера рассмотрим построение по изложенному принципу специализированной программы расчета пространственных трубопроводов на температурные воздействия (составленной в КиевЗНИИЭПе инж. Е. Г. Гутман по заказу Укрводканалпроекта). Основная консольная система образуется разрезами у опорных сечений, к которым прикладываются группы по 6 неизвестных усилий ($m_x, m_y, m_z, n_x, n_y, n_z$).

Исходная информация:

а) нумеруются характерные точки системы; номера точек, определяющих ориентацию каждого участка и его сечения, записываются в сператоре участка. Влияние каждой группы неизвестных на участки указывается в отдельной шкале. Операто-

Таблица VI.1

ni	mi	$g1 (g2)$	Усилия	Номер ячейки в H2
1	1	1	N_1, N_2	1
		2	N_1, N_2	2
	2	1	M_{1x}	3
			M_{2x}	4
		2	M_{1x}	5
			M_{2x}	6
		1	M_{1y}	7
			M_{2y}	8
	3	2	M_{1y}	9
			M_{2y}	10
	4	1	M_{1z}, M_{2z}	11
		2	M_{1z}, M_{2z}	12
2	1	1	N_1, N_2	13
		2	N_1, N_2	14
		3	N_1, N_2	15
	2	1	M_{1x}	16
			M_{2x}	17
		2	M_{1x}	18
			M_{2x}	19
			M_{1x}	20
		3	M_{2x}	21
			M_{1y}	22
	3	1	M_{2y}	23
			M_{1y}	24
		2	M_{2y}	25
			M_{1y}	26
		3	M_{2y}	27
			M_{1y}	28
	4	1	M_{1z}, M_{2z}	28
		2	M_{1z}, M_{2z}	29
		3	M_{1z}, M_{2z}	30

Таблица VI.2

ni	mi	Усилия	Номер ячейки в P
1	1	N_1, N_2	1
		M_{1x}	2
	2	M_{2x}	3
		M_{1y}	4
	3	M_{2y}	5
		M_{1z}, M_{2z}	6
	4	N_1, N_2	7
		M_{1x}	8
	2	M_{2x}	9
		M_{1y}	10
2	3	M_{2y}	11
		M_{1z}, M_{2z}	12
	4	M_{1z}, M_{2z}	12
		M_{1z}, M_{2z}	12

ры участков и шкалы влияния — это массив H1 (восьмеричные числа);

б) выбирается общая неподвижная система координат xuz .

Координаты точек помещаются в массиве H2 (десятичные числа);

в) характеристики сечений (площадь F , осевой и полярный моменты инерции I, T) записываются в массив H4;

г) вводится перепад температур и коэффициент линейного расширения, по которым в программе вычисляются свободные члены системы уравнений.

Неописанные процедуры общей программы получают конкретное содержание: в «procedure построение массива R » последовательно вычисляются номера неизвестных для разрезов и записываются в массив R в соответствии со шкалами влияния; две процедуры вычисления ординат эпюр $i1$ и $i2$ ($k1$ и $k2$) заменены одной процедурой с формальными параметрами $a1$, $a2$ и a , которые заменяются фактическими параметрами $i1$, $i2$, i или $k1$, $k2$, k . Эта процедура состоит из двух этапов.

Первый этап. По координатам точек, указанных в операторе участка, вычисляются его длина и направляющие косинусы (массив L [1 : 9]) осей собственной системы координат участка x_1 , y_1 , z_1 (продольная ось и главные оси инерции сечения) в неподвижной системе координат x , y , z .

Второй этап. Начало системы координат xuz при переходе к новой группе неизвестных усилий переносится в точку ее приложения (заделку). Усилия M_x , M_y , M_z , N_x , N_y , N_z в любой точке от каждой группы единичных значений неизвестных выражаются через координаты этой точки x , y , z (в системе координат с началом в соответствующей заделке) по следующей матрице $T1$:

	$m_x=1$	$m_y=1$	$m_z=1$	$n_x=1$	$n_y=1$	$n_z=1$
M_x	1				z	$-y$
M_y		1		$-z$		x
M_z			1	y	$-x$	
N_x				1		
N_y					1	
N_z						1

При переходе к системе координат участка эти усилия умножаются на направляющие косинусы осей x_1 , y_1 , z_1 . Таким образом, при определении перемещений ординаты эпюр характеристики сечения взяты в собственной системе координат участка.

Первый этап выполняется при первом обращении к процедуре для данного участка, а при последующих обращениях в цикле для mu и gl выполняется только второй этап. Ниже приведен второй этап процедуры с комментариями. Индексы 1 и 2 в координатах и усилиях относятся к началу и концу участка. Описания массивов и переменных, а также **procedure** Fa выборки из массива координат точки помещены в начале первого этапа процедуры.

...; Второй этап: **begin** switch $S := M3, M4, M5, M6, M7, M8$;

$M1: ai := (a-1) \div 6 + 1; Fa(xa, ya, za, ai);$

$x1 := x1 - xa; y1 := y1 - ya; z1 := z1 - za;$

$x2 := x2 - xa; y2 := y2 - ya; z2 := z2 - za;$

comment по метке $M1$ выполняется обращение к процедуре Fa выборки координат точки ai приложения группы усилий и пересчет координат концов участка относительно этой точки (параллельный перенос осей координат);

$M2: Mx1 := My1 := Mz1 := Mx2 := My2 := Mz2 :=$

$Nx := Ny := Nz := 0;$

$a := a - (ai-1) \times 6; \text{go to } S[a];$

comment при каждом обращении к процедуре переключатель S отсылает к одной из следующих 6 формул, записанных в графах матрицы $T1$ в соответствии с номером a неизвестного в группе

$M3: Mx1 := Mx2 := 1; \text{go to } M9;$

$M4: My1 := My2 := 1; \text{go to } M9;$

$M5: Mz1 := Mz2 := 1; \text{go to } M9;$

$M6: Nx := 1; My1 := -z1; Mz1 := y1; My2 := -z2; Mz2 := y2;$
 $\text{go to } M9;$

$M7: Ny := 1; Mx1 := z1; Mz1 := -x1; Mx2 := z2; Mz2 := -x2;$
 $\text{go to } M9;$

$M8: Nz := 1; Mx1 := -y1; My1 := x1; Mx2 := -y2; My2 := x2;$

$M9: \text{if } mu=4 \text{ then go to } M10; mu1 := 3 \times mu; \text{comment } mu \text{ определяется в общей программе: } mu=1, 2, 3 \text{ относится к моментам } Mx, My, Mz, \text{ а } mu=4 \text{ — к продольной силе на участке.}$

Далее вычисляются усилия $a1$ и $a2$ на концах участка;

$a1 := Mz1 \times L[mu1] + My1 \times L[mu1-1] + Mx1 \times L[mu1-2];$

$a2 := Mz2 \times L[mu1] + My2 \times L[mu1-1] + Mx2 \times L[mu1-2];$

$M10: a1 := a2 := Nx \times L[7] + Ny \times L[8] + Nz \times L[9] \text{ end}$

end процедуры.

Обобщенная программа расчета дискретных систем методом деформаций

VI. 17. Наибольшее применение метод основного блока и переменных процедур получил в рекурсивных построениях библиотеки программ метода деформаций, важным преимуществом которого является сочетание стандартности, облегчающей автоматизацию, и общности — элементы дискретной (не только стержневой) системы могут быть произвольными. По существу расчет методом деформаций начинается тогда, когда все частные вопросы, связанные с видом сооружения и сведенные к расчету элемента определенного типа, уже решены. Специфика сооружения того или иного вида переносится целиком на стадию расчета элемента, для которого разрабатывается серия процедур.

Следующие далее рассуждения во многом повторяют сказанное для метода сил. Форма изложения сохраняется с тем, чтобы легче было увидеть как сходство, так и различие в построении программ для этих двух основных методов строительной механики.

Предпосылки алгоритма метода деформаций были изложены в IV. 9. Можно было бы компактную схему алгоритма без всяких изменений переписать на языке АЛГОЛ-60 и реализовать затем на ЭЦВМ. Однако такая программа не будет оптимальной, так как придется хранить в памяти множество двумерных матриц для коэффициентов отдельных элементов и матрицы для решения системы уравнений. Поэтому применим другой подход: откажемся от компактной схемы и от матричной записи исходной информации в оперативной памяти. Если и будем пользоваться матричным представлением, то, главным образом, для выявления закономерностей и составления вычислительных формул: речь будет идти не о хранящихся, а о порождающихся матрицах (VI. 12).

Будем считать, что номера неизвестных, вызывающих реактивные усилия в закреплениях данного элемента, засылаются в процессе расчета в рабочий массив R . Для определения реактивных усилий от единичных значений неизвестных (номера которых указаны в массиве R) используем переменную $F1$, от нагрузки — $F2$. В машине этим двум переменным соответствуют рабочие ячейки, в которые последовательно, с замещением, засылаются вычисляемые величины. Алгоритм вычисления $F1$ и $F2$ зависит от типа элементов, из которых составлено сооружение.

Будем считать, что процедуры вычисления указанных вели-

чин являются переменными, и, следуя нашему правилу, представим их в виде неописанных процедур-функций, обозначенных $F1$ и $F2$, для которых в разделе описаний даны только названия, т. е. определено лишь назначение, но не содержание. Речь идет именно о процедуре-функции, так как результатом ее выполнения является одна величина.

Информацию об основной системе будем задавать в общем случае в двух массивах: $H1 [1 : c1]$ — для целых и $H2 [1 : c2]$ — вещественных чисел, информацию о нагрузках — в массиве $P [1 : c3]$. Границы массивов $c1$, $c2$, $c3$ рассматриваются как входные величины. В программе представляется свобода в выборе способа задания этой информации.

В обобщенную программу помимо переменных процедур входит постоянный основной блок, включающий определение параметров n и h (порядка системы и порядка ленты), составление системы уравнений, процедур прямого и обратного хода, решения системы, вычисления реактивных усилий в закреплениях элемента от найденных деформаций и нагрузки. На этом заканчивается собственно расчет статически неопределимой системы (раскрытие статической неопределимости). Дальнейшие расчеты носят локальный характер и состоят в определении усилий в сечениях элемента при известных усилиях в закреплениях.

Запись программы в виде основного блока и переменных процедур заменяет блок-схему и является планом для построения библиотеки программ для расчета статически неопределимых систем различных видов методом деформаций. Наиболее общий, универсальный вариант можно получить, приложив к основному блоку процедуры-функции выборки из заданных массивов величин реактивных усилий от единичных деформаций и от нагрузки.

Основной блок может быть оформлен независимо в кодах машины. Обращения к неописанным процедурам производятся в соответствии с принятым распределением памяти. В случае необходимости эти обращения могут быть изменены. Специализированная программа компилируется из перфокарт основного блока и подпрограмм (процедур) для данного типа систем.

В обобщенную программу метода деформаций входят следующие этапы расчета:

- 1) ввод массивов $H1$ и $H2$ — исходных данных об основной системе;
- 2) определение параметров — порядка системы уравнений n и порядка ленты h ;
- 3) очистка массива A для матрицы системы уравнений;

4) определение поочередно для всех элементов реактивных усилий в закреплениях от единичных значений неизвестных деформаций с накоплением (суммированием) одноименных величин в ячейках массива A (матрице системы уравнений);

5) прямой ход для матрицы с получением преобразованной матрицы A на месте нахождения исходной (для экономии памяти используется ленточная структура);

6) ввод массива P исходных данных по очередному загрузению;

7) очистка массива B для свободных членов;

8) определение поочередно для всех элементов системы реактивных усилий в закреплениях от нагрузки с накоплением их в ячейках массива B (графе свободных членов);

9) прямой ход для свободных членов и обратный ход решения системы уравнений с получением значений неизвестных на месте свободных членов в массиве B (печать значений неизвестных);

10) определение поочередно для всех элементов реактивных усилий S от суммарного воздействия внешней нагрузки и найденных значений неизвестных (печать величин S). Для определения каждого реактивного усилия выполняются следующие операции: вычисление реактивного усилия от нагрузки и присвоение его значения переменной S , вычисление реактивных усилий от единичных значений неизвестных, умножение их на найденные значения неизвестных с накоплением в S , печать S .

Этапы 6—10 необходимо выполнить для каждого из загрузений.

Как видим, операция определения реактивных усилий в закреплениях элементов от единичных значений неизвестных и нагрузки выполняется многократно. Это выгоднее, чем хранить в памяти машины множество двумерных матриц реактивных усилий для всех элементов — подход другой, чем при составлении аналогичной программы для ручного выполнения расчетов на клавишных автоматах.

Указанные этапы расчета организуются в циклы и объединяются рядом логических условий.

VI. 18. В программе приняты следующие обозначения.

Параметры:

$e1$ — количество элементов в основной системе;

$c[1], c[2], c[3], c[4]$ — длины массивов, соответственно $H1, H2, H3, P$. Остальные переменные массива c являются резервными и могут быть использованы в переменных процедурах;

n — порядок системы уравнений (количество фиктивных закреплений);

h — порядок ленты;

q — число комбинаций загрузки (число граф свободных членов);

t — максимально возможное количество неизвестных, влияющих на элемент основной системы (вызывающих реактивные усилия в его закреплениях).

Параметры $e1$, $c[1]$, $c[2]$, $c[3]$, $c[4]$, t вводятся, параметры n и h вычисляются во внешнем блоке.

Массивы, переменные и функции:

$c[1 : 10]$ — массив для параметров;

$H1[1 : c[1]]$ и $H2[1 : c[2]]$ — для информации об основной системе (целые и вещественные числа);

$P[1 : c[3]]$ — для информации о нагрузках;

$H3[1 : c[4]]$ — массив для признаков, используемый в процедурах-функциях;

$A[1 : u]$ — для коэффициентов системы уравнений ($u = n \times h - h \times (h-1)/2$);

$B[1 : n]$ — для свободных членов и величин неизвестных;

$R[1 : t]$ — для номеров неизвестных, влияющих на один элемент системы;

$S[1 : t]$ — для определения реактивных усилий в закреплениях элемента от найденных величин неизвестных и нагрузки;

$F1$ — функция для определения реактивных усилий в закреплениях элемента от единичных значений неизвестных;

$F2$ — функция для определения реактивных усилий в закреплениях элемента от нагрузки.

Признаки:

$b=0$ — формирование массива A , прямой ход для матрицы;

$b=1$ — формирование массива B , прямой ход для свободных членов, обратный ход;

$b=2$ — определение реактивных усилий в закреплениях от найденных неизвестных и нагрузки.

Признак b вырабатывается в программе.

$\omega=0$ — отсутствие узловой нагрузки;

$c[3]=1$ — отсутствие нагрузки на элементах (массив $H3$ состоит из одной ячейки). Узловая нагрузка (при $\omega \neq 0$) вводится в виде свободных членов. ω и $c[3]$ — входные величины.

$p1$, $p2$ — признаки, используемые в процедурах-функциях.

Индексы (счетчики):

$j1$ — порядковый номер комбинации загрузки;

nu — порядковый номер элемента;
 $g1, g2$ — порядковые номера ячеек в массиве R ;
 i, k — номера граф матрицы A ;
 $n1$ — количество неизвестных в массиве R , включая нулевые неизвестные;
 d — номер диагонального элемента;
 $h1, d1, m, m1, j$ — дополнительные индексы, используемые в процедурах прямого и обратного хода;
 us — условное число, означающее окончание массива R , если количество неизвестных для данного элемента меньше максимального значения t .

Условное число будем применять и в тех случаях, когда данную ячейку по тем или иным причинам нужно исключить из процесса вычислений. Использовать для этой цели нулевое значение (как в методе сил) уже нельзя, поскольку нулевое значение в массиве R означает действительную (а не фиктивную) связь, в которой по признаку $b=2$ должно быть найдено реактивное усилие.

$a, N1, N2, N3, N4, n1, n2$ — переменные, которые служат для связи основной программы с процедурами-функциями.

Обобщенная программа метода деформаций

Примечание

```

begin integer  $e1, n, h, q, w, p1, p2, u, b, a,$ 
 $t, j1, nu, g1, g2, i, k, n1, n2, d, h1, d1, m,$ 
 $m1, j, us, r1, r2, N1, N2, N3, N4; \dots$ 
real  $s \dots$ ; integer array  $c[1:10]$ ;
input ( $c, w, e1, q, t$ );
begin integer array  $H1[1:c[1]], R[1:t],$ 
 $H3[1:c[4]]$ ; real array  $H2[1:c[2]], P[1:$ 
 $:c[3]], S[1:t]$ ;
procedure построение массива  $R; \dots$ ;
real procedure  $F1; \dots$ ;
real procedure  $F2; \dots$ ;
procedure прямой ход ( $b$ ); ...
procedure обратный ход; ....
input ( $H1, H2, H3, P, c$ );
 $n := 1; h := 1; b := 0$ ;
for  $nu := 1$  step 1 until  $e1$  do begin
for  $g1 := 1$  step 1 until  $t$  do  $R[g1] := 0$ ;
построение массива  $R$ ;
for  $g1 := 1$  step 1 until  $t$  do begin

```

Ввод параметров;

Перечисление неописанных процедур;

Описания процедур даны в VI. 4;

Очистка массива R ;


```

if  $R[g1] \neq 0 \vee R[g1] = us$  then go to M1;
if  $n < R[g1]$  then  $n := R[g1]$ ;
for  $g2 := g1$  step 1 until  $t$  do
if  $R[g2] \neq 0 \vee R[g2] \neq us$  then begin
 $r1 := \text{abs}(R[g2] - R[g1]) + 1$ ;
if  $h < r1$  then  $h := r1$  end;
M1: end  $g1$  end  $nu$ ;
 $u := n \times h - h \times (h - 1) \div 2$ ;
begin array  $A[1 : u], B[1 : n]$ ;
for  $i := 1$  step 1 until  $u$  do  $A[i] := 0$ ;
 $r1 := n - h + 2$ ;
for  $j1 := 1$  step 1 until  $q$  do begin;
if  $b = 2$  then  $b := 1$ ;
for  $i := 1$  step 1 until  $n$  do  $B[i] := 0$ ;
M2:  $N2 : N4 := 1$ ;
if  $b = 1$  then begin if  $w \neq 0$  then
input ( $B$ ); if  $c[3] \neq 1$  then
input ( $P$ ) else go to M6 end;
for  $nu := 1$  step 1 until  $e1$  do begin
 $a := 0$ ;  $n2 := 1$ ;
for  $gl := 1$  step 1 until  $t$ 
do begin  $R[gl] := 0$ ;  $S[gl] := 0$  end;
построение массива  $R$ ;  $n1 := 0$ ;
for  $gl := 1$  step 1 until  $t$ 
do if  $R[gl] \neq us$  then  $n1 := n1 + 1$ ;
for  $gl := 1$  step 1 until  $t$  do begin
 $i := R[gl]$ ; if  $i = us \vee (i = 0 \wedge b \neq 2)$ 
then go to M5;
if  $b = 1$  then begin if  $c[3] \neq 1$  then
 $B[i] := B[i] + F2$ ; go to M5 end;
if  $b = 2$  then begin if  $c[3] \neq 1$  then
 $S[gl] := S[gl] + F2$ ; go to M3 end;
 $r2 := 0$ ; if  $i > r1$  then begin
 $r2 := i - r1$ ;  $r2 := r2 \times (r2 + 1) \div 2$  end;
 $d := (i - 1) \times h - r2 + 1$ ;
M3: for  $g2 := 1$  step 1 until  $t$  do begin
 $k := R[g2]$ ; if  $k = us \vee k = 0$  then go to M4;
if  $b = 2$  then begin if  $g2 < gl$  then
go to M4;  $S[gl] := S[gl] + B[k] \times F1$ ;
if  $g2 = gl$  then go to M4;
 $S[g2] := S[g2] + B[i] \times F1$ ;
go to M4 end;

```

Определение параметров n и h ;

Верхние границы массивов A и B вычислены вне блока;

Очистка массива B ;

Ввод свободных членов и (или) нагрузок на элементах;
Очистка массивов R и S ;

Извлечение номера i -го неизвестного из массива;
Формирование массива B (свободных членов); начало формирования массива S ;
Определение номера диагонального элемента;

if $k > i$ then $A[d+k-i] := A[d+k-i] + F1$;

M4: if $g2 > g1$ then $n2 := n2 + 1$
end $g2$;

M5: end $g1$;

if $b=2$ then begin

for $g1 := 1$

step 1 until t do if $R[g1] \neq us$ then

output ($S[g1]$) end

end nu ;

comment после печати следует пробел, т. е. при выводе на печать группы усилий для различных стержней отделены друг от друга пробелами. Если в машине имеется промежуточное буферное запоминающее устройство, то массивы S последовательно выводятся на буфер. Печать усилий от данной комбинации загрузки производится с буфера, одновременно выполняется расчет на следующее загрузке;

if $b=0$ then begin прямой ход (0);

$b := 1$; go to M2 end;

M6: if $b=1$ then begin прямой ход (1);
обратный ход; output (B);

$b := 2$; go to M2 end

end $j1$ end end

end программы.

Формирование массива A (матрицы системы уравнений);

Печать реактивных усилий в закреплениях стержня;

Печать величин неизвестных.

При формировании массива A можно было применить сокращенный цикл for $g2 := g1$ step 1 until t и несколько уменьшить число проверочных операций, но при этом пришлось бы многократно производить вычисление d в цикле для $g2$ (менять места i и k , если $i > k$), что не выгодно. Некоторые из входных величин ($e1, c1, c2, c3, t$) в специализированных программах могут вычисляться по особым процедурам.

Пример, приведенный в VI. 14 для метода сил, полезно отладить для метода деформаций.

Повторим данные для отладки: $e1=10, t=5, q=2$. Состояния массива R (для номеров неизвестных) следующие:

для первого элемента ($ni=1$) —

R	3	5	4		us
-----	---	---	---	--	------

для десятого элемента ($ni=10$) —

R	7		6	9	us
-----	---	--	---	---	------

Помимо нулевых ячеек, введены ячейки с условными числами (us). Состояния массива для остальных участков можно не задавать, так как обработка информации производится аналогично. Состояния переменных в процессе отладки записываются в таблицу.

О способах задания исходной информации для расчета методом деформаций

VI. 19. Просматривая обобщенные программы метода сил и метода деформаций, можно заметить их глубокое сходство: одинаковую организацию процесса в циклах, совпадение больших отрезков программ. Это наталкивает на мысль о создании общей программы, охватывающей оба метода.

Однако в нашей практике эти программы пока применялись раздельно (если не считать того, что некоторые процедуры-функции F и $F1$ включают упрощенные блоки метода сил).

Приемы задания исходной информации, рассмотренные в VI. 15, применимы (и, пожалуй, более эффективно) для метода деформаций.

Сведения об основной системе и о связи между элементами в системе будем задавать с помощью операторов. Оператор элемента представляет собой одну или несколько ячеек в первом, втором и третьем адресах*, в которых записываются номера неизвестных $a, b, c, d, e \dots$ (рис. 40), оказывающих влияние на данный элемент (условное число $us=7776$ является признаком окончания оператора), или более краткая информация, на основании которой можно вычислить эти номера. Например, в операторе стержня пространственной системы можно указать номера ограничивающих его узлов, а в отдельном массиве поместить операторы узлов — группы ячеек, в которых указывают-

* Здесь и в дальнейшем будем ориентироваться на трехадресные машины. Для других типов машин ячейка делится на части так, чтобы в каждой из них можно было записать максимальный номер неизвестного, возможный в задачах данного класса.

ся номера неизвестных, соответствующие каждому узлу. Под узлом можно понимать также любую группу неизвестных, общую для ряда стержней.

Для дальнейшего сокращения информации можно использовать группировку стержней, например, для группы стержней

	<i>a</i>	<i>b</i>	<i>c</i>
	<i>d</i>	<i>e</i>	<i>f</i>
	<i>g</i>	<i>i</i>	<i>777</i>

$a-b, a-c, a-d, \dots$ в соответствующем операторе указаны номера узлов $a, b, c, d, \dots$. При такой форме записи номер узла a указывается в данной группе стержней только один раз.

Рис. 40.

По информации, указанной в операторе элемента, формируется рабочий массив R для номеров неизвестных, фигурирующий в обобщенной программе; для этой цели применяется специальная процедура построения массива R . Поскольку в операторе в одной ячейке записано несколько неизвестных, то при извлечении соответствующих чисел и заполнении массива R приходится выполнять операции выделения части кода, сдвига и т. п., зависящие от конкретной машины и оформляемые как процедуры-коды.

Заметим, кстати, что слово «оператор» употребляется нередко для обозначения разных понятий: оператор АЛГОЛа выражает единицу действия алгоритмического языка; структурный оператор системы указывает номера неизвестных, оказывающих влияние на данный элемент структуры; наконец, слово оператор применяется по отношению к человеку, работающему на вычислительной машине. Во всех этих понятиях общим является некоторое действие (влияние).

Указанный способ задания информации и построенный на его основе алгоритм абстрагируются от того, что представляют собой неизвестные статически неопределимой системы, какова их физическая природа, — это могут быть углы поворота узлов, линейные смещения или любые их линейные комбинации.

Покажем, например, к чему сводится информация о неизвестных для простейшего частного случая — плоской многоэтажной рамы, представленной на рис. 41. Номера стержней записаны на схеме в кружках, номера углов поворота — у узлов, номера смещений ярусов — справа от каждого яруса. Нумерация неизвестных выбрана так, чтобы получить благоприятную ленточную структуру системы уравнений:

x_1 — смещение первого яруса, x_2-x_5 — углы поворота узлов первого уровня, x_6 — смещение второго яруса, x_7-x_{12} — углы поворота узлов второго уровня и т. д. Для рам данного типа поря-

док ленты определяется при такой нумерации по формуле $h = m + 2$, где m — максимальное число узлов в уровне рамы. Если нумеровать подряд все углы поворота узлов, затем смещения ярусов, то получится менее благоприятная ленточная окаймленная структура.

Таким образом, от принятого порядка нумерации неизвестных зависит характер структуры матрицы системы уравнений. Для метода деформаций всегда можно получить ленточную структуру, правильно выбирая порядок нумерации неизвестных.

Для данной задачи в операторах стоек в восьмеричном коде записываются номера a, b, c трех неизвестных, в операторах ригелей — номера a, b , двух неизвестных. Так, в операторе стойки 6 (рис. 42) указаны номера неизвестных 3, 6 и 10 (x_3 и x_{10} — углы поворота соответствующих узлов, x_6 — смещение второго яруса); в операторе стойки 2 — номера неизвестных 0, 1 и 3 (0 — защемление, x_1 — линейное смещение первого яруса, x_3 — угол поворота); в операторе ригеля 21 указаны номера 10, 11 неизвестных x_{10}, x_{11} (углы поворота). В рассматриваемом примере номера неизвестных идут в возрастающем порядке, но это не обязательно.

Максимальное количество неизвестных, влияющих на стержень пространственной системы, может быть равно 12 (по шесть неизвестных на каждый узел); обычно же меньше, так как некоторые неизвестные равны нулю или связаны определенными зависимостями.

Для стойки 6	3	6	10
Для стойки 2	0	1	3
Для ригеля 21	10	11	3776

Рис. 42.

Если система состоит из элементов более сложного вида, чем стержень, то количество неизвестных может значительно увеличиваться. Например, в панельном здании с поперечными несущими стенами в качестве элементов рассматриваются вертикальные диафрагмы (поперечные стены) и горизонтальные диафрагмы (диски перекрытий), неизвестными являются деформации по линиям сопряжений вертикальных и горизонтальных диафрагм.

Помимо информации об основной системе и неизвестных (операторы), для расчета должны быть заданы геометрия системы,

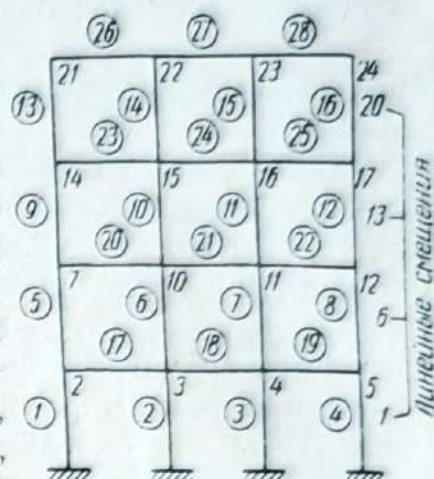


Рис. 41.

жесткостные характеристики элементов и нагрузки. Способы задания этих величин будут рассмотрены при изложении процедур-функций для расчета элементов различных типов. Эти процедуры являются взаимозаменяемыми в общей программе и поэтому обозначены одним идентификатором $F(F1)$.

Функция-выборка реактивных усилий из заданного массива

VI. 20. Будем считать, что реактивные усилия в добавленных закреплениях данного элемента от единичных значений неизвестных заданы как исходные величины и расположены в числовом массиве $H2$ в последовательности $r_{aa}, r_{ab}, r_{ac}, \dots, r_{bb}, r_{bc}, r_{bd}, \dots, r_{cc}, r_{cd}, \dots$, и т. д., где $a, b, c, d, \dots$ — номера неизвестных в массиве R , представляющем собой развернутую запись оператора (см. рис. 40), выбранного из массива $H1$. Таким образом, коэффициенты для отдельного элемента (составляющие коэффициентов системы уравнений) образуют некоторую треугольную матрицу (рис. 43), вытянутую в одну линию. Если в результате расчета желательно получить усилие в каком-либо действительном закреплении стержня, то оно представляется в операторе как нулевое неизвестное, а в массиве $H2$ для него записываются реактивные усилия от единичных значений всех других неизвестных, кроме нулевых и условных (0 и us). В массиве $H2$ записаны подряд реактивные усилия в закреплении для $1, 2, \dots, e$ элементов системы от единичных значений неизвестных, в массиве P — от нагрузки. Группы реактивных усилий для ряда стержней нередко бывают одинаковыми. Для того, чтобы учесть это обстоятельство, применяются специальные признаки, но не интегральные, как в методе сил (VI. 16), а дифференцированные — для каждого стержня свой признак.

r_{aa}	r_{ab}	r_{ac}	r_{ad}	...
	r_{bb}	r_{bc}	r_{bd}	...
		r_{cc}	r_{cd}	...
			r_{dd}	...
				...

Рис. 43.

Условимся, что признак $p1 \neq 0$ ($p2 \neq 0$) указывает, что группа реактивных усилий в массиве $H2$ (P) для данного стержня такая же, как для предыдущего, и $p1=0$ ($p2=0$), если эти группы различные.

Для экономии памяти значения $p1$ ($p2$) можно размещать не в целых ячейках, а в разрядах ячейки. Величины $p1$ удобно размещать в кодовых разрядах ячейки оператора, величины $p2$ — в разрядах шкалы, для каждой нагрузки своя шкала. Для ячеек шкалы резервируем массив $H3$ $[1:c[4]$, где длина массива c обычно небольшая — 3-4 ячейки. При вводе очеред-

Таблица VI.3

$ni (p-1)$	$g1$	$g2$	Номер ячейки в $H-2$
1 (0)	1	1	1
		2	2
		3	3
	2	2	4
		3	5
		3	6

2 (0)	1	1	7
		2	8
		3	9
		4	10
3 (1)	2	2	11
		3	12
		4	13
	3	3	14
		4	15
		4	16

Таблица VI.4

$ni (p-2)$	$g1$	Номер ячейки в P
1 (0)	1	1
	2	2
	3	3
2 (0) 3 (1)	1	4
	2	5
	3	6
	4	7

ной комбинации нагрузок вводится и шкала для нее. Таким образом, массив $H3$ (как и массив P) имеет постоянную длину в памяти, сколько бы ни было нагрузок. Для получения признаков $p1$ и $p2$ используется процедура-код, которая зависит от способа задания информации о признаках, системы команд и длины слова конкретной машины. Ввиду этого «procedure выде-

ление признака» оформим как неописанную процедуру.

Рассмотрим пример записи данных в массивы $H2$ и P . Пусть на первый стержень ($ni=1$) влияют три неизвестных ($g1=1, 2, 3$), на второй и третий стержень ($ni=2, 3$) — по четыре неизвестных ($g1=1, 2, 3, 4$). Номера неизвестных определяются по информации, изложенной в операторах и перенесенных в массив R . Будем считать, что реактивные усилия от единичных значений неизвестных в третьем стержне такие же, как во втором,* хотя

* Пример подобной ситуации — ригели (или стойки) ортогональной плоской рамы, имеющие одинаковые длины и моменты инерции.

номера неизвестных и разные, повторяются также реактивные усилия от нагрузок. Таким образом, будем иметь следующие значения признаков: для первого и второго стержней $p1=0$, $p2=0$ для третьего стержня $p1 \neq 0$, $p2 \neq 0$.

Для данного примера массив $H2$ помещен в табл. VI. 3, массив P — в табл. VI. 4. Условимся, что номера неизвестных в операторах (и в массиве R) идут в порядке возрастания. Хотя в основной программе нет подобного ограничения, но когда реактивные усилия являются исходными данными, их всегда можно расположить в порядке, соответствующем такой нумерации, что более удобно* (нулевые неизвестные располагаются в начале). В массиве $H2$ помещены реактивные усилия в закреплениях стержня $R[g1]$ от единичных значений неизвестных, имеющих номера $R[g2]$ (на рис. 43 они обозначены $r_{aa}, r_{ab}, \dots, r_{bb}, r_{bc}, \dots$), в массиве P — реактивные усилия в закреплениях $R[g1]$ от нагрузки (или иначе $r_{ap}, r_{bp}, \dots$).

В данном примере было принято, что нулевых закреплений (нулевых неизвестных) нет. Если, например, принять, что для первого стержня $R[1]=0$, то нумерация в табл. VI. 3 изменится: будут записаны реактивные усилия в закреплении $R[1]$ ($g1=1$) от неизвестных $R[2]$ и $R[3]$ ($g2=2, 3$, но отсутствует $g2=1$).

real procedure F1;

begin integer g3;

procedure выделение $p1; \dots$;

comment в процедуре выполняется выборка из массива $H2$ реактивных усилий от единичных значений неизвестных;

if $a=0$ **then** **begin** $a := 1$ выделение $p1$;

if $p1 \neq 0$ **then** $N2 := N1$ **else** $N1 := N2$;

if $g1 \neq 1 \wedge R[g1-1]=0$

$\wedge b=0$ **then**

for $g3 := g1$ **step** 1 **until** t **do**

if $R[g3] \neq 0 \wedge R[g3] \neq us$ **then** $N2 := N2 + 1$ **end**;

comment

последний цикл нужен для того, чтобы обойти при $b=0$ группу чисел в массиве $H2$, соответствующую реактивным усилиям в нулевом закреплении;

$F1 := H2[N2]$; $N2 := N2 + 1$

end процедуры;

real procedure F2;

begin **comment**

* Заметим, что в других процедурах это не всегда возможно и подобное ограничение могло бы вызвать затруднения.

в процедуре выполняется выборка из массива P реактивных усилий от нагрузки;

if $a=0$ **then begin** $a := 1$; выделение p_2 ; **if** $p_2 \neq 0$ **then**
 $N_4 := N_3$ **else** $N_3 := N_4$; **if** $g_1 \neq 1 \wedge R[g_1-1]=0 \wedge b=1$
then $N_4 := N_4 + 1$ **end**;

comment последняя проверка нужна для того, чтобы обойти при $b=1$ (вычисление свободных членов) число в массиве P , соответствующее реактивному усилию в нулевом закреплении;

$F_2 := P[N_4]$; $N_4 := N_4 + 1$

end процедуры.

Для отладки процедур F_1 и F_2 можно использовать пример, приведенный в табл. VI.3 и VI.4, а также варианты этого примера при различных сочетаниях признаков p_1 , p_2 , c_1 .

ХАРАКТЕРНЫЕ ПРОЦЕДУРЫ СИСТЕМЫ АЛГОРИТМОВ
СТРОИТЕЛЬНОЙ МЕХАНИКИПроцедуры-функции для расчета элементов
статически неопределимых систем различных видов

VII. 1. Изложенная в VI. 18—VI. 20 программа расчета статически неопределимых систем с включением процедуры-функции выборки является универсальной, с ее помощью можно рассчитать произвольную дискретную систему. Операторный способ записи номеров неизвестных и разного рода признаки сообщают программе гибкость, делают возможным учесть частные особенности той или иной задачи. Элемент системы, произвольно расположенный в пространстве, может испытывать различные деформации: продольные, изгибные, крутильные, сдвигающие. Однако достаточно принять некоторые из неизвестных равными нулю или установить соотношения между ними (задавать функцию неизвестных), либо принять некоторые реактивные усилия равными нулю или очень большой величины (моделирующей бесконечную жесткость), чтобы автоматически получить соответствующие упрощения. Имеется возможность учесть регулярности некоторых видов. При небольшом видоизменении функции выборки можно задавать, например, признаки повторения не для одного стержня, а для группы стержней (если элементом является стержень). В некоторых случаях для унификации вводятся дополнительные фиктивные стержни, не участвующие в вычислениях. Так, если характеристики стержней 4—7 и 4—10 (рис. 44, б) такие же, как стержней 3—6 и 3—7 (рис. 44, а), то для того, чтобы группа стержней повторяла предыдущую, необходимо добавить два фиктивных стержня, противоположные концы которых отмечаются условными номерами (us). Реактивные усилия, относящиеся к фиктивным стержням, при выборке пропускаются. В качестве условного номера можно принять, например, максимальный адрес, для машины М-20 — число 7777.

По существу, благодаря развитой логической схеме программа реализует те же операции, которые выполняет инженер: анализирует структуру системы и упрощает решение в том или ином частном случае в соответствии с информацией, характеризующей

задачу и заданной операторами и характеристиками (в общей программе ими являются реактивные усилия в закреплениях стержня от единичных значений неизвестных) типовых стержней или групп стержней *.

Однако достоинство рассмотренной программы — ее универсальность является в известном смысле и недостатком, так как

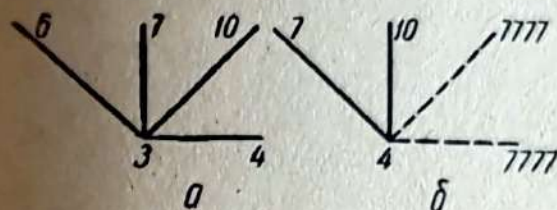


Рис. 44.

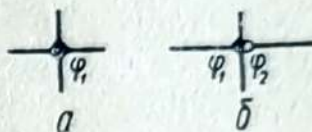


Рис. 45.

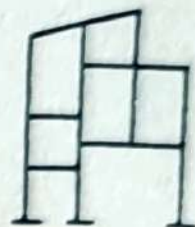


Рис. 46.

исходные данные должны быть заданы в общем виде как реактивные усилия от единичных значений неизвестных, которые приходится заранее вычислять вручную для каждого элемента. Это затрудняет подготовку исходных данных, особенно при большом количестве закреплений, и требует внимательности при определении знаков усилий.

Для сложных пространственных задач подобная работа может оказаться в условиях проектной организации практически невыполнимой.

Вообще опыт показал, что всякое упрощение подготовки исходных данных имеет большое практическое значение и, наоборот, выполнение даже сравнительно небольшой и несложной части расчета вручную сильно снижает эффект механизации. В процессе развития системы программ должно быть достигнуто последовательное и планомерное сокращение неавтоматизированных этапов расчета. Процедуры-функции, рассмотренные ниже, преследуют эту цель: каждая из них является менее общей, чем процедура выборки, и относится к определенному классу систем, но зато реактивные усилия не выбираются из заданного массива, а вычисляются на основе более простой информации о элементах в виде длин (координат узлов), жесткостей и т. п.

Расчетные формулы в процедурах даны для стержня с жесткими закреплениями на концах (рис. 45, а). Если какая-либо

* Дальнейшее развитие этой идеи дано в программах анализа зрительного образа стержневой системы (VII.20—VII.25).

связь отсутствует, то она заменяется фиктивной связью. Например, если на конце стержня имеется шарнир, то на него накладывается фиктивное защемление и добавляется еще одно неизвестное — угол поворота в шарнире (рис. 45, б). Благодаря этому в основной системе шарниров нет и расчетные формулы не изменяются.

VII. 2. Начнем с простейшей задачи — процедуры-функции расчета стержня постоянного сечения, входящего в состав плоской рамы.

Ограничения: вертикальность стоек и отсутствие учета продольной деформативности стержней. Система не регулярна — могут отсутствовать стойки или ригели в любом месте системы (рис. 46).

Исходную информацию будем располагать: операторы — в массиве $H1$; жесткости EI и длины L стержней — в массиве $H2$ (для каждого стержня сначала жесткость, затем длина). В основу программы положены следующие формулы вычисления реактивных усилий:

$r_{11} = \frac{4EI}{L}$ — момент в защемлении стержня от угла поворота, равного единице;

$r_{12} = -\frac{6EI}{L^2}$ — момент в защемлении стержня от смещения одного конца относительно другого на единицу, равный по взаимности реакции по направлению смещения от единичного угла поворота;

$r_{22} = \frac{12EI}{L^3}$ — реакция в закреплении стержня, направленном перпендикулярно к оси, от смещения одного конца относительно другого на единицу.

В рабочих ячейках $r1$, $r2$, $r3$ расположим величины: $4EI/L$, $6EI/L^2$, $12EI/L^3$. Определение величин $r1$, $r2$, $r3$ представим в виде операторов присваивания:

$$r1 := 4 \times H2[N2] / H2[N2 + 1];$$

$$r2 := 1,5 \times r1 / H2[N2 + 1];$$

$$r3 := 2 \times r2 / H2[N2 + 1].$$

Как и в предыдущей процедуре, счетчик $N2$ — порядковый номер числа в массиве $H2$, при переходе к следующему стержню он изменяется на 2 (если нет признака повторения).

Рассмотрим три случая вычисления реактивных усилий для стержня рамы.

Первый случай — ригель рамы, не имеющий линейных смещений. В операторе соответствующего стержня указываются номера двух неизвестных углов поворота a и b , в третьем адресе записано условное число us (в М-20 — число 7776).

	a	b	us
--	-----	-----	------

Второй случай — стойка рамы, имеющая одно неизвестное линейное смещение. В операторе соответствующего стержня указываются номера трех неизвестных — двух углов поворота нижнего и верхнего узлов a, c и одного линейного смещения b :

	a	b	c
--	-----	-----	-----

Этот случай относится к рамам, не имеющим пропущенных ригелей (рис. 41), когда для каждого яруса получаем одно групповое перемещение.

Третий случай — стойка рамы, имеющая два неизвестных смещения для нижнего и верхнего узлов. В операторе соответствующего стержня указываются номера четырех неизвестных — двух углов поворота a, b и двух линейных смещений c, d :

	a	b	c
	d	us	

В кодовых разрядах ячейки оператора записываются также различные признаки. Например, единица в первом разряде означает, что характеристики данного стержня (длина, жесткость) такие же, как у предыдущего, нуль — в противном случае*; единица во втором разряде указывает, что запись номеров неизвестных продолжается в следующей ячейке, нуль — что данная ячейка последняя в операторе и т. п.

Указанным трем случаям соответствуют следующие матрицы характеристик $r_{aa}, r_{ab}, r_{ac}, \dots, r_{bb}, r_{bc}, \dots$:

* В другом варианте даны дифференцированные признаки повторения — отдельно для длины, отдельно для жесткости.

Первый случай

$4EI/L$	$2EI/L$
	$4EI/L$

Второй случай

$4EI/L$	$-6EI/L^2$	$2EI/L$
	$12EI/L^3$	$-6EI/L^2$
		$4EI/L^2$

Третий случай

$4EI/L$	$2EI/L$	$6EI/L^2$	$-6EI/L^2$
	$4EI/L$	$6EI/L^2$	$-6EI/L^2$
		$12EI/L^3$	$-12EI/L^3$
			$12EI/L^3$

Если обратиться к основной программе, то номера строк этих матриц равны значениям $g1$, номера граф — значениям $g2$.

В основной программе вычисляются величины $n1$ и $n2$. $n1$ — количество неизвестных в операторе: в первом случае оно равно двум, во втором случае — трем и в третьем случае — четырем. Эти равенства используются в операторах перехода на ту или другую группу формул. $n2$ — порядковый номер ячейки в матрице реактивных усилий стержня — используется для организации переключателя.

real procedure F1;

begin integer i1, k1; real r1, r2, r3;

procedure выделение p1; ...;

switch S1 := M3, M4, M3;

switch S2 := M3, M7, M4, M6, M7, M3;

switch S3 := M3, M4, M5, M7, M3, M5, M7, M6, M8, M6;

if a=0 then begin a := 1; выделение p1;

if p1 ≠ 0 then N2 := N1 else N1 := N2;

r1 := 4 × H2[N2]/H2[N2+1];

r2 := 1,5 × r1/H2[N2+1];

r3 := 2 × r2/H2[N2+1]; N2 := N2+2 end;

comment вычисление r1, r2, r3 производится при первом обращении к процедуре (a=0);

if g1 > g2 then go to M1;

i1 := g1; k1 := g2; go to M2;

M1: i1 := g2; k1 := g1;

M2: if n1 = 2 then go to S1[n2];


```

if  $n1 = 4$  then go to  $S3[n2]$ ;
if  $n1 = 3$  then go to  $S2[n2]$ ;
M3:  $F1 := r1$ ; go to M9;
M4:  $F1 := r^{1/2}$ ; go to M9;
M5:  $F1 := r2$ ; go to M9;
M6:  $F1 := r3$ ; go to M9;
M7:  $F1 := -r2$ ; go to M9;
M8:  $F1 := -r3$ ;
M9: end процедуры.

```

VII. 3. Перейдем к процедуре-функции расчета стержня постоянного сечения, входящего в состав плоской стержневой системы произвольного вида. Постараемся составить эту процедуру так, чтобы можно было автоматически получить упрощения для практически важных подклассов, входящих в этот обширный класс задач.

В общем случае для стержня будем вводить четыре величины: изгибную жесткость EI , продольную жесткость EF , длину L , тангенс угла наклона: для ригелей — к горизонтали, для стоек — к вертикали.

Подразделение стержней на ригели и стойки сделано для того, чтобы можно было получить упрощения для подкласса рамных систем. В других случаях оно является условным. Например, раскосы в фермах можно по желанию отнести как к стойкам, так и к ригелям.

Номера неизвестных будем записывать в операторе стержня в следующей последовательности: $\varphi_n, x_n, v_n, \varphi_k, x_k, y_k$, где индексы n и k означают узел в начале и конце стержня, φ — угол поворота, x — горизонтальное смещение для стойки или вертикальное смещение для ригеля (для ортогональных систем смещение x будет направлено перпендикулярно к оси стержня), y — вертикальное смещение для стойки или горизонтальное смещение для ригеля (для ортогональных систем смещение y направлено вдоль стержня). Таким образом, одно и то же неизвестное — вертикальное смещение узла для ригеля записывается как x (во втором адресе ячейки оператора), а для стойки как y (в третьем адресе); для горизонтального смещения — наоборот. Только в этом состоит отличие в задании информации для ригелей и стоек.

Для рам с вертикальными стойками вводится интегральный признак ортогональности s [5] (используется резервная ячейка массива s из обобщенной программы, VI. 18). В этом случае для каждого стержня даются три величины — EI , EF и L . Если не учитывается влияние изгиба (шарниры на двух концах), то вме-

сто номеров углов поворота записываются условные числа и вводятся величины EF и L . Если продольная деформативность не учитывается, то вместо номеров соответствующих неизвестных записываются условные числа и вводятся величины EI и L . В операторе стержня (и рабочем массиве R) записываются в указанном выше порядке шесть номеров неизвестных ($t=6$), им соответствует матрица реактивных усилий, приведенная в табл. VII. 1. Матрица эта в памяти не хранится, используется промежуточный рабочий массив r из двенадцати ячеек, в которые последовательно записываются вычисляемые величины.

Процедура-функция $F1$ определяет реактивное усилие, соответствующее значениям ni (порядковый номер стержня), $g1$ и $g2$ (порядковые номера строки и графы табл. VII. 1), вычисляемым в основной программе (VI. 18).

Возможны пять случаев задания исходной информации для стержня:

- 1) $EI, EF, L, \text{tg } \alpha$; 2) $EF, L, \text{tg } \alpha$; 3) EI, EF, L ; 4) EI, L ;
- 5) EF, L . Эти величины располагаются в массиве $H2$.

```
real procedure F1; begin integer i1, k1, u;
array r[1:12]; procedure выделение p1; ...;
if a=0 then begin a:=1; выделение p1;
if p1≠0 then N2:=N1 else N1:=N2;
if c[5]=1 then begin r[1]:=1; r[2]:=0 end;
go to M1; if R[1]=us then u:=2 else u:=3;
alpha:=arctan(H2[N2+u]); r[1]:=cos(alpha);
r[2]:=sin(alpha);
```

```
M1: if R[1]=us then r[3]:=0 else r[3]:=H2[N2];
if R[3]=us then r[4]:=0 else if R[1]=us then r[4]:=H2[N2]
else r[4]:=H2[N2+1];
if R[1]=us ∨ R[3]=us then r[5]:=H2[N2+1]
else r[5]:=H2[N2+2];
r[6]:=r[3]/r[5]; r[7]:=r[4]/r[5]; r[8]:=6×r[6]/r[5];
r[9]:=r[8]×r[1]; r[8]:=r[8]×r[2];
r[10]:=2×r[8]×r[2]/r[5]+r[7]×r[1]×r[1];
r[10]:=2×r[8]×r[1]/r[5]-r[7]×r[1]×r[2];
r[12]:=2×r[9]×r[1]/r[5]+r[7]×r[2]×r[2];
N2:=N2+4; if c[5]=1 then N2:=N2-1;
```

```
if R[1]=us then N2:=N2-1; if R[3]=us then N2:=N2-1 end;
comment формирование массива r[1:12] производится при первом обращении к процедуре F1 для данного стержня (по признаку a=0);
```

```
if g1>g2 then go to M2; i1:=g1; k1:=g2; go to M3;
```


M2: $i1 := g2; k1 := g1;$
 M3: if $i1 = 1$ then begin if $k1 = 1$ then go to M4;
 if $k1 = 2$ then go to M12; if $k1 = 3$ then go to M6;
 if $k1 = 4$ then go to M5; if $k1 = 5$ then go to M7;
 if $k1 = 6$ then go to M11 end;
 if $i1 = 2$ begin if $k1 = 2$ then go to M10;
 if $k1 = 3$ then go to M14; if $k1 = 4$ then go to M12;
 if $k1 = 5$ then go to M15; if $k1 = 6$ then go to M9 end;
 if $i1 = 3$ then begin if $k1 = 3$ then go to M8;
 if $k1 = 4$ then go to M6; if $k1 = 5$ then go to M9;
 if $k1 = 6$ then go to M13 end;
 if $i1 = 4$ then begin if $k1 = 4$ then go to M4;
 if $k1 = 5$ then go to M7; if $k1 = 6$ then go to M11 end;
 if $i1 = 5$ then begin if $k1 = 5$ then go to M10;
 if $k1 = 6$ then go to M14 end; go to M8;
 M4: $F1 := 4 \times r[6];$ go to M16;
 M5: $F1 := 2 \times r[6];$ go to M16;
 M6: $F1 := r[8];$ go to M16;
 M7: $F1 := r[9];$ go to M16;
 M8: $F1 := r[10];$ go to M16;
 M9: $F1 := r[11];$ go to M16;
 M10: $F1 := r[12];$ go to M16;
 M11: $F1 := -r[8];$ go to M16;
 M12: $F1 := -r[9];$ go to M16;
 M13: $F1 := -r[10];$ go to M16;
 M14: $F1 := -r[11];$ go to M16;
 M15: $F1 := -r[12];$
 M16: end процедуры.

VII. 4. Рассмотрим процедуру-функцию расчета стержня пространственной фермы произвольного вида. Предполагается, что стержни пространственной системы работают только на продольные усилия. Исходными данными для расчета стержня являются четыре величины:

$$\begin{aligned}
 \Delta x &= (x_2 - x_1)/L; \Delta y = (y_2 - y_1)/L; \\
 \Delta z &= (z_2 - z_1)/L; EF_c/L,
 \end{aligned}$$

где $x_1, y_1, z_1, x_2, y_2, z_2$ — координаты узлов 1 и 2 стержня 1—2; F_c — площадь поперечного сечения стержня; L — длина стержня; $\Delta x, \Delta y, \Delta z$ — косинусы углов между осью стержня и принятыми координатными осями.

На стержень пространственной фермы в общем случае влияют шесть неизвестных: три смещения для каждого узла по на-

Таблица VII. I

	1	2	3	4	5	6
1	$\frac{4EI}{L}$	$-\frac{(6EI/L^2) \times}{\times \cos \alpha}$	$\frac{(6EI/L^2) \times}{\times \sin \alpha}$	$\frac{2EI}{L}$	$\frac{(6EI/L^2) \times}{\times \cos \alpha}$	$-\frac{(6EI/L^2) \times}{\sin \alpha}$
2		$\frac{(12EI/L^3) \times}{\times \cos^2 \alpha} + \frac{(EF/L) \times}{\times \sin^2 \alpha}$	$-\frac{(12EI/L^3) \times}{\times \sin \alpha \cos \alpha} + \frac{(EF/L) \times}{\times \sin \alpha \cos \alpha}$	$-\frac{(6EI/L^2) \times}{\times \cos \alpha}$	$-\frac{(12EI/L^3) \times}{\times \cos^2 \alpha} - \frac{(EF/L) \times}{\times \sin^2 \alpha}$	$\frac{(12EI/L^3) \times}{\sin \alpha \cos \alpha} - \frac{(EF/L) \times}{\sin \alpha \cos \alpha}$
3			$\frac{(12EI/L^3) \times}{\times \sin^2 \alpha} + \frac{(EF/L) \times}{\times \cos^2 \alpha}$	$\frac{(6EI/L^2) \times}{\sin \alpha}$	$\frac{(12EI/L^3) \times}{\times \sin \alpha \cos \alpha} - \frac{(EF/L) \times}{\times \sin \alpha \cos \alpha}$	$-\frac{(12EI/L^3) \times}{\times \sin \alpha} - \frac{(EF/L) \times}{\times \cos^2 \alpha}$
4				$\frac{4EI}{L}$	$\frac{(6EI/L^2) \times}{\times \cos \alpha}$	$-\frac{(6EI/L^2) \times}{\times \sin \alpha}$
5					$\frac{(12EI/L^3) \times}{\times \cos^2 \alpha} + \frac{(EF/L) \times}{\times \sin^2 \alpha}$	$-\frac{(12EI/L^3) \times}{\times \sin \alpha \cos \alpha} + \frac{(EF/L) \times}{\times \sin \alpha \cos \alpha}$
6						$\frac{(12EI/L^3) \times}{\times \sin^2 \alpha} + \frac{(EF/L) \times}{\times \cos^2 \alpha}$

	1	2	3	4	5	6
r	$\cos \alpha$	$\sin \alpha$	EI	EF	L	EI/L
	7	8	9	10	11	12
	$\frac{EF}{L}$	$\frac{(6EI/L^2) \times}{\times \sin \alpha}$	$\frac{(6EI/L^2) \times}{\times \cos \alpha}$	$\frac{(12EI/L^3) \times}{\times \sin^2 \alpha} + \frac{(EF/L) \times}{\times \cos^2 \alpha}$	$\frac{(12EI/L^3) \times}{\times \sin \alpha \cos \alpha} - \frac{(EF/L) \times}{\times \sin \alpha \cos \alpha}$	$\frac{(12EI/L^3) \times}{\times \cos^2 \alpha} + \frac{(EF/L) \times}{\times \sin^2 \alpha}$

правлению координатных осей x, y, z . В операторе стержня записываются номера левого и правого узлов стержня, а в операторах соответствующих узлов — по три номера неизвестных для каждого узла (для каждого оператора требуется одна ячейка). Если какое-либо неизвестное равно нулю, то оно не нумеруется и соответствующий адрес остается пустым.

Матрица характеристик $r_{aa}, r_{ab}, \dots, r_{bb}, r_{bc}, \dots$ для стержня имеет следующий вид (все значения, указанные в табл. VII. 2, умножаются на EF_c/L).

Таблица VII. 2

$\Delta x \times \Delta x$	$\Delta x \times \Delta y$	$\Delta x \times \Delta z$	$-\Delta x \times \Delta x$	$-\Delta x \times \Delta y$	$-\Delta x \times \Delta z$
	$\Delta y \times \Delta y$	$\Delta y \times \Delta z$	$-\Delta y \times \Delta x$	$-\Delta y \times \Delta y$	$-\Delta y \times \Delta z$
		$\Delta z \times \Delta z$	$-\Delta z \times \Delta x$	$-\Delta z \times \Delta y$	$-\Delta z \times \Delta z$
			$\Delta x \times \Delta x$	$\Delta x \times \Delta y$	$\Delta x \times \Delta z$
				$\Delta y \times \Delta y$	$\Delta y \times \Delta z$
					$\Delta z \times \Delta z$

В данном случае все формулы табл. VII. 2 можно привести к одной формуле с применением переменных с индексами и циклизированием вычислений (правило 8 из V. 10). Для этого данные для стержня будем засылать в промежуточную рабочую строку

r	Δx	Δy	Δz	$-\Delta x$	$-\Delta y$	$-\Delta z$
-----	------------	------------	------------	-------------	-------------	-------------

Величины EF_c/L поместим в рабочую ячейку G .

Нетрудно проверить справедливость следующей формулы для вычисления коэффициентов табл. VII. 2:

```
for i1 := 1 step 1 until 6 do
for k1 := i1 step 1 until 6 do
F1 := G × r[i1] × r[k1].
```

Для вычисления величин $\Delta x, \Delta y, \Delta z$ достаточно знать координаты узлов системы относительно выбранных осей. Если система обладает регулярностью того или иного рода, то исходную информацию можно упростить. Для того, чтобы иметь свободу в этом отношении, формирование r и G представим в виде неопи-
санной процедуры.

Приведенная ниже процедура-функция расчета стержня пространственной фермы произвольного вида получилась весьма

компактной благодаря использованию закономерностей в образовании формул *.

real procedure $F1$;

begin integer $i1, k1$; real G ; array $r[1:6]$;

procedure формирование r и G ; ...;

if $a=0$ then begin $a:=1$; формирование r и G end;

for $i1:=1$ step 1 until 6 do

for $k1:=i1$ step 1 until 6 do begin

if $g1 > g2$ then begin if $i1 \neq g2 \vee k1 \neq g1$

then go to $M1$ end;

if $i1 \neq g1 \vee k1 \neq g2$ then go to $M1$;

$F1 := G \times r[i1] \times r[k1]$ end;

$M1$: end процедуры.

VII. 5. Процедура-функция расчета стержня пространственной системы произвольного вида предназначена для широкого класса задач.

Предполагается, что стержни пространственной системы воспринимают как продольные усилия, так и изгибающие моменты. Принимаем две системы координат. Первая система x_1, y_1, z_1 — постоянная для всех стержней; вторая система x, y, z — текущая и относится к отдельному стержню. Оси x и y совпадают с главными осями поперечного сечения, ось z — направлена вдоль стержня, от начала к концу.

Исходными данными для стержня являются следующие величины:

I_1, I_2, I_3 — косинусы углов между осями x, y, z и осью x_1 ;

m_1, m_2, m_3 — косинусы углов между осями x, y, z и осью y_1 ;

n_1, n_2, n_3 — косинусы углов между осями x, y, z и осью z_1 ;

EI_x, EI_y — жесткости стержня на изгиб относительно осей x и y ;

GT — жесткость стержня на кручение относительно оси z ;

EF — жесткость стержня на продольную деформацию (сжатие, растяжение);

L — длина стержня.

В частных случаях некоторые из этих величин обращаются в нуль. Например, для пространственной ортогональной рамы часть косинусов равна нулю, и если не учитываются продольные деформации, то можно не задавать EF .

* Интересно отметить, что преобразование строки r эквивалентно трансформации Гаусса для матрицы-строки.

Данная задача значительно сложнее предыдущих. На стержень пространственной системы в общем случае влияют двенадцать неизвестных:

$\varphi_{nx_1}, \varphi_{ny_1}, \varphi_{nz_1}$ — углы поворота узла в начале стержня относительно осей x_1, y_1, z_1 ;

$\delta_{nx_1}, \delta_{ny_1}, \delta_{nz_1}$ — смещения того же узла по направлению осей x_1, y_1, z_1 ;

$\varphi_{kx_1}, \varphi_{ky_1}, \varphi_{kz_1}$ — углы поворота узла в конце стержня относительно осей x_1, y_1, z_1 ;

$\delta_{kx_1}, \delta_{ky_1}, \delta_{kz_1}$ — смещения того же узла по направлению осей x_1, y_1, z_1 .

В операторе стержня записываются номера узлов на концах стержня, а в операторах соответствующих узлов — по шесть неизвестных для каждого узла (для каждого оператора требуется две ячейки). Если какое-либо неизвестное равно нулю, то оно не нумеруется и соответствующий адрес остается пустым. При этом автоматически получаются такие же упрощения, как для предыдущей процедуры. Матрица характеристик $r_{aa}, r_{ab}, \dots, r_{bb}, r_{bc}, \dots$ для стержня приведена в табл. VII. 3. Большинство формул имеет по три члена, остальные — по два.

Поскольку в табл. VII. 3. количество разных формул очень велико, прямой путь — указание по именуемому выражению, какой формулой следует пользоваться, — совершенно неприемлем. Поэтому попытаемся в соответствии с правилом из V. 10 разделить формулы на группы с одинаковой структурой и для каждой группы составить одну обобщенную формулу с применением переменных с индексами и циклированием вычислений.

В данном случае лучше разбить формулы на две группы: первая группа имеет трехчленную структуру, вторая — двухчленную.

Например, в клетке 1. 3 табл. VII. 3 расположена трехчленная формула, соответствующая значению функции,

$$F1 = 4EI_x l_1 n_1 / L + 4EI_y l_2 n_2 / L + GT l_3 n_3 / L,$$

а в клетке 1. 4 — двухчленная формула

$$F1 = -6EI_x l_1 l_2 / L^2 + 6EI_y l_2 l_1 / L^2.$$

Знаки умножения в формулах опущены.

Исходную информацию располагаем в рабочем массиве r .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
e	l_1	l_2	l_3	m_1	m_2	m_3	n_1	n_2	n_3	EI_x	EI_y	GT	EF	L

Для вычисления направляющих косинусов достаточно знать координаты узлов относительно выбранных осей и угол чистого вращения стержня *. Соответствующая процедура для общего случая, позволяющая учесть структурные особенности системы (циклическую или сферическую симметрию и т. п.), будет рассмотрена в VII. 9. Здесь же представим формирование массива r в виде неописанной процедуры.

Для того, чтобы легче было установить зависимости между индексами, предусмотрим еще рабочую строку $r1$ из трех ячеек, которая может иметь два состояния:

$$r1 \left[\begin{array}{|c|c|c|} \hline EI_x & EI_y & GT \\ \hline \end{array} \right] \text{ и } r1 \left[\begin{array}{|c|c|c|} \hline EI_y & EI_x & EF \\ \hline \end{array} \right]$$

В ячейке массива $r1$ числа последовательно засылаются из соответствующих ячеек массива r .

Далее предусмотрим рабочий массив $r2$ также из трех ячеек:

$$r2 \left[\begin{array}{|c|c|c|} \hline EI_x & EI_y & GT \\ \hline \end{array} \right]$$

В этот массив засылаются в различных сочетаниях вспомогательные величины: $4/L$, $2/L$, $1/L$, $-1/L$, $6/L^2$, $12/L^3$.

С помощью полученных рабочих массивов r , $r1$, $r2$ были составлены две обобщенные формулы:

трехчленные формулы заменяются циклом

$a := 0$; for $u := 1$ step 1 until 3 do

$a := a + r1[u] \times r2[u] \times r[v1 + u] \times r[v2 + u]$; $F1 := a$.

Все три члена последовательно вычисляются по одной и той же формуле и результат накопления a присваивается функции $F1$;

двухчленные формулы заменяются выражением

$$F1 := r1[1] \times r2[1] \times r[v1 + 1] \times r[v2 + 2] + \\ r1[2] \times r2[2] \times r[v1 + 2] \times r[v2 + 1].$$

Порядок засылки чисел в массивы $r1$, $r2$ определяется в зависимости от ряда логических условий, учитывающих закономерности изменения множителей в вычислительных формулах. Индексы $v1$ и $v2$, указывающие, какие из величин (косинусов) массива r следует подставлять в формулы, также изменяются по определенным закономерностям.

* Угол чистого вращения (один из углов Эйлера) представляет собой угол между осью x и горизонталью, проведенной в плоскости поперечного сечения. Для вертикального стержня задается угол между осями x и x_1 .

Эти закономерности используются в помещенной ниже процедуре-функции.

Приняты обозначения:

$g1, g2$ — номера строки и графы табл. VII. 3, определяющие вычислительную формулу;

a — признак начала обработки стержня. Величины $g1, g2, a$ вычисляются в основной программе.

$i1$ и $i2$ — порядковые номера строки и графы табл. VII. 3, равные соответственно $g1$ и $g2$ при условии, что $g1 < g2$ или $g2$ и $g1$ при $g1 > g2$.

$r[1:14]$ — массив исходных данных.

$r1, r2[1:3]$ — вспомогательные массивы.

Остальные переменные служат для формирования двух вычислительных формул, заменяющих матрицу VII. 3.

real procedure F1; begin integer $i1, k1, s1, s2, w, v1, v2$;

real $L, c1, c2, c3$; array $r[1:14], r1, r2[1:3]$;

procedure формирование r ; ...;

if $a=0$ then begin $a:=1$; формирование r ;

$L:=r[14]$; $c1:=1/L$; $c2:=6 \times c1/L$; $c3:=2 \times c1 \times c2$ end;

if $g1 > g2$ then go to M1;

$i1:=g1$; $k1:=g2$; go to M2;

M1: **$i1:=g2$; $k1:=g1$;**

M2: **$s1:=i1$; $w:=0$; $s2:=4$;**

M3: **if $s1 \leq 3$ then go to M4;**

$s1:=s1-3$; $s2:=s2+3$; go to M3;

M4: **if $w=1$ then go to M5;**

$v1:=(s1-1) \times 3$;

if $k1 < s2$ then begin $v2:=(k1-i1) \times 3$;

go to M6 end;

$s1:=k1$; $w:=1$; go to M3;

M5: **$v2:=(s1-1) \times 3$;**

M6: **if $i1 \leq 3$ then begin if $k1 \leq 3$ then go to M7**

else if $k1 \leq 6$ then go to M14 else if $k1 \leq 9$

then go to M8 else go to M16 end else

if $i1 \leq 6$ then begin if $k1 \leq 6$ then go to M9

else if $k1 \leq 9$ then go to M17 else go to M10 end else

if $i1 \leq 9$ then begin if $k1 \leq 9$ then go to M7

else go to M16 end else go to M9;

comment в этом полном условном операторе отчетливо видна закономерность изменения членов и индексов в вычислительных формулах. Эти изменения происходят при значениях $i1$ и $k1$, равных 4, 7, 10;

M7: $r2[1] := r2[2] := 4 \times c1$; $r2[3] := c1$;
 $v2 := v2 + v1$; go to M11;
 M8: $r2[1] := r2[2] := 2 \times c1$; $r2[3] := -c1$; go to M11;
 M9: $r2[1] := c3$; $r2[2] := c3$; $r2[3] := c1$;
 $v2 := v2 + v1$; go to M12;
 M10: $r2[1] := -c3$; $r2[2] := -c3$; $r2[3] := -c1$; go to M12;
 M11: $r1[1] := r[10]$; $r1[2] := r[11]$;
 $r1[3] := r[12]$; go to M13;
 M12: $r1[1] := r[11]$; $r1[2] := r[10]$; $r1[3] := r[13]$;
 M13: $a := 0$; for $u := 1$ step 1 until 3 do
 $a := a + r1[u] \times r2[u] \times r[v1 + u] \times r[v2 + u]$;
 $F1 := a$; go to M19;
 M14: $r2[1] := -c2$; $r2[2] := c2$;
 M15: $r1[1] := r[10]$; $r1[2] := r[11]$; go to M18;
 M16: $r2[1] := c2$; $r2[2] := -c2$; go to M15;
 M17: $r2[1] := c2$; $r2[2] := -c2$;
 $r1[1] := r[11]$; $r1[2] := r[10]$;
 M18: $F1 := r1[1] \times r2[1] \times r[v1 + 1] \times r[v2 + 2] + r1[2] \times r2[2] \times r[v1 + 2] \times r[v2 + 1]$;
 M19: end процедуры.

VII. 6. В библиотеку программ метода деформаций входит ряд других процедур, которые здесь не приводятся:

— процедуры расчета стержня переменного сечения и с абсолютно жесткими вставками, используется упрощенный алгоритм метода сил;

— процедуры $F2$ определения реактивных усилий от местной нагрузки на стержне, алгоритм подобен рассмотренному в V. 8;

— процедуры определения ординат эпюры усилий по длине стержня через заданные интервалы.

Ряд процедур представляет упрощение процедуры-функции для распространенных частных случаев: пространственных ортогональных рам, пространственных многоэтажных рам с жесткими дисками и т. п.

Построение алгоритмов для расчета пространственных систем со структурной симметрией

VII. 7. Процедура-функция, изложенная в VII. 5, пригодна для расчета пространственных систем произвольного вида, но не приспособлена для эффективного использования структурных особенностей регулярных систем. В связи с этим в работе [109]

был поставлен вопрос об изменении текущих систем координат для стержня с тем, чтобы можно было использовать циклическую или сферическую симметрию сооружения для сокращения исходной информации и упрощения расчета.

Развитие этой темы привело к созданию целой библиотеки процедур расчета циклических и сферических систем * [111, 112]. Согласно принятой классификации [49], циклическими называются конструктивные схемы, полярно-симметричные относительно центральной оси не только полностью, но и частично, т. е. одни элементы следуют циклической закономерности, другие нет; введено понятие степени цикличности системы. Такое расширение понятия «циклические системы» по сравнению с традиционным позволяет включить в этот класс самые разнообразные сооружения, как-то: сетчатые куполы, башни для радио и телевидения, сквозные цилиндрические оболочки [115], остовы дирижаблей, турбины, градирни, водонапорные башни различных типов, комбинированные сооружения, составленные из фрагментов поверхностей вращения разных видов, и т. д.

Геометрические характеристики стержней будем относить теперь не к одной неподвижной прямоугольной декартовой системе координат, а к двум системам координат определенного вида. Первая из них — подвижная система прямоугольных декартовых координат — поочередно помещается в узлах сооружения (сопровождающий трехгранник). Направление осей этой системы может быть произвольным и меняется от узла к узлу таким образом, чтобы получить единообразное расположение в этих осях всех или, во всяком случае, большинства стержней данного сооружения и тем самым уменьшить количество разнотипных стержней.

Другая система — неподвижная, общая для всего сооружения. В этом отношении она сходна с системой координат из VII. 5, но в отличие от последней тип ее обусловлен видом структуры рассматриваемого сооружения.

Обе системы (подвижную и соответствующую ей неподвижную), обеспечивающие в каждом конкретном случае минимум исходной информации, в соответствии с установившейся терминологией [11] назовем естественными. Выбор типа естественных систем координат диктуется структурными особенностями рассматриваемой конструкции.

* Параграфы VII.7—VII.10 подготовлены И. Э. Кимельфельдом под руководством автора.

Для осециклического сооружения (рис. 47) естественной подвижной является система прямоугольных декартовых координат x_1, y_1, z_1 . Начало ее помещается в каждый узел сооружения; во всех узлах ось z_1 параллельна центральной оси и направлена в сторону увеличения нумерации ярусов системы; ось x_1 в каждом узле направлена к центральной оси сооружения, а ось y_1 касательна к окружности, проходящей через узлы рассматриваемого яруса конструкции.

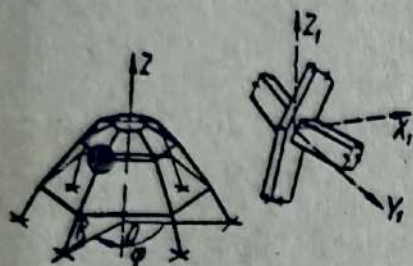


Рис. 47.



Рис. 48.

В качестве неподвижной естественной системы примем цилиндрическую систему координат Z, R, φ ; ось z совпадает с центральной осью системы и направлена в ту же сторону, что и ось z_1 подвижной естественной системы.

При расчете сферических систем, например, геодезических куполов (рис. 48), в качестве естественной подвижной лучше принять прямоугольную систему координат x_1, y_1, z_1 с таким направлением осей: ось x_1 направлена к центру описанной сферы, оси x_1 и z_1 лежат в диаметральной плоскости, проходящей через центр сферы и рассматриваемый узел, а ось y_1 — перпендикулярна к этой плоскости.

В практике встречаются сооружения, расчетная схема которых представляет собой комбинацию стержневых систем, отличающихся одна от другой видом структуры. В этом случае на отдельных участках сооружения применяются естественные системы различных типов — в соответствии с особенностями структуры рассматриваемых частей сооружения. Для схемы, приведенной на рис. 49, на участке II принимаются естественные системы координат, как для ортогональной, а на участках I и III — как для циклической системы. На рис. 50 дана схема ребристого покрытия стадиона «Динамо» в Москве [113], предложенного институтом Моспроект. При расчете его в качестве естественных неподвижных систем отсчета следует принять четыре цилиндрических системы координат, вертикальные оси которых должны проходить через точки O_1, O_2, O_3 и O_4 .

VII. 8. Для пространственных циклических систем была составлена более гибкая, чем в VII. 5, процедура-функция расчета стержня. Будем считать, что в начале и конце стержня, произвольно расположенного в пространстве, помещены разные декартовы прямоугольные системы координат x_1, y_1, z_1 и x_2, y_2, z_2 (рис. 51). Положение этих систем относительно стержня определяется направляющими косинусами углов между осями $x_1, y_1, z_1, x_2, y_2, z_2$ и собственными осями стержня x, y, z .

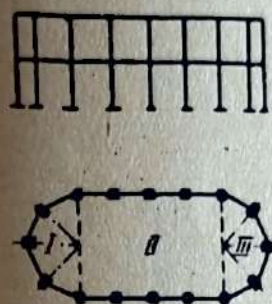


Рис. 49.

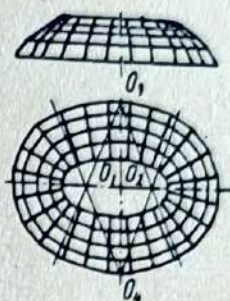


Рис. 50.

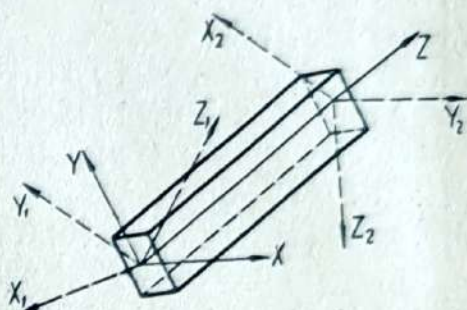


Рис. 51.

Исходными данными для расчета стержня являются следующие величины:

$l_1, m_1, n_1; l_2, m_2, n_2; l_3, m_3, n_3$ — косинусы углов между осями x_1, y_1, z_1 , соответственно, с осями x, y, z ;

$l'_1, m'_1, n'_1; l'_2, m'_2, n'_2; l'_3, m'_3, n'_3$ — косинусы углов между осями x_2, y_2, z_2 , соответственно, с осями x, y, z . Эти величины располагаются в рабочем массиве $r [1 : 23]$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
r	l_1	l_2	l_3	m_1	m_2	m_3	n_1	n_2	n_3	l'_1	l'_2	l'_3	m'_1	m'_2	m'_3	n'_1	n'_2	n'_3	L	EI_x	EI_y	GT	EF

Построение процедуры-функции производится аналогично приведенному в VII. 5, промежуточные массивы 1 и 2 те же. Матрица реактивных усилий получается из табл. VII. 3 следующим образом: в пределах строк 1—6 и граф 1—6 подставляются направляющие косинусы одного конца стержня (эта часть таблицы записывается также, как в табл. VII. 3), в пределах строк 7—12 и граф 7—12 — направляющие косинусы другого конца (косинусы l, m и n из табл. VII. 3 отмечаются штрихами), в пределах строк 1—6 и граф 7—12 — направляющие косинусы обоих концов (вторые косинусы табл. VII. 3 отмечаются штрихами).


```

real procedure F1; begin integer i1, k1, s1, w, p, e, v1, v2, u;
real a1; array r[1:23], r1, r2[1:3];
procedure F6 формирование r; ...;
if a=0 then begin a:=1; F6;
L:=r[19]; c1:=1/L; c2:=6×c1/L;
c3:=2×c1×c2 end;
if g1>g2 then go to M1;
i1:=g1; k1:=g2; go to M2;
M1: i1:=g2; k1:=g1;
M2: s1:=i1; w:=0;
M3: if s1≤6 then begin p:=0;
go to if s1≤3 then M5 else M4;
end else p:=3; if s1≤9 then go to M5
M4: s1:=s1-3;
M5: e:=(s1-(1+p))×3; if w=0 then begin
v1:=e; go to M6 end else v2:=e; go to M7;
M6: s1:=k1; w:=1; go to M3;
M7: if i1≤3 then begin if k1≤3 then
go to M8 else if k1≤6 then go to M16
else if k1≤9 then go to M9 else go to
M15 end else if i1≤6 then begin
if k1≤6 then go to M10 else if k1≤9
then go to M15 else go to M11 end else
if i1≤9 then begin if k1≤9 then go to
M8 else go to M15 end else go to M10;
M8: r1[1]:=r1[2]:=4×c1; r1[3]:=c1; go to M12;
M9: r1[1]:=r1[2]:=2×c1; r1[3]:=-c1; go to M12;
M10: r1[1]:=r1[2]:=c3; r1[3]:=c1; go to M13;
M11: r1[1]:=r1[2]:=-c3; r1[3]:=-c1; go to M13;
M12: r2[1]:=r[20]; r2[2]:=r[21]; r2[3]:=r[22]; go to M14;
M13: r2[1]:=r[21]; r2[2]:=r[20]; r2[3]:=r[23];
M14: a1:=0; for u:=1 step 1 until 3 do
a1:=a1+r1[u]×r2[u]×r[v1+u]×r[v2+u];
F1:=a1; go to M19;
M15: r1[1]:=c2; r1[2]:=-r1[1];
if k1>9 then go to M17;
r2[1]:=r[21]; r2[2]:=r[20]; go to M18;
M16: r1[2]:=c2; r1[1]:=-r1[2];
M17: r2[1]:=r[20]; r2[2]:=r[21];
M18: F1:=r1[1]×r2[1]×r[v1+1]×r[v2+2]+r1[2]×r2[2]×r[v1
+2]×r[v2+1];
M19: end процедуры.

```


VII. 9. Для определения направляющих косинусов достаточно знать координаты узлов в начале и конце стержня и угол чистого вращения.

Из известных выражений для определения углов Эйлера (см. например, И. Н. Бронштейн и К. А. Семендяев «Справочник по математике») получены следующие расчетные формулы:

$$l_1 = - \frac{1}{\sqrt{x_1^2 + y_1^2}} \left(y_1 \cos \rho + \frac{z_1 x_1 \sin \rho}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \right); \quad (1)$$

$$l_2 = - \frac{1}{\sqrt{x_1^2 + y_1^2}} \left(y_1 \sin \rho - \frac{z_1 x_1 \cos \rho}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \right); \quad (2)$$

$$l_3 = \frac{x_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}}; \quad (3)$$

$$m_1 = \frac{1}{\sqrt{x_1^2 + y_1^2}} \left(x_1 \cos \rho - \frac{z_1 y_1 \sin \rho}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \right); \quad (4)$$

$$m_2 = - \frac{1}{\sqrt{x_1^2 + y_1^2}} \left(x_1 \sin \rho + \frac{z_1 y_1 \cos \rho}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \right); \quad (5)$$

$$m_3 = \frac{y_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}}; \quad (6)$$

$$n_1 = \sqrt{\frac{x_1^2 + y_1^2}{x_1^2 + y_1^2 + z_1^2}} \cdot \sin \rho; \quad (7)$$

$$n_2 = \sqrt{\frac{x_1^2 + y_1^2}{x_1^2 + y_1^2 + z_1^2}} \cdot \cos \rho; \quad (8)$$

$$n_3 = \frac{z_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}}, \quad (9)$$

где x_1, y_1, z_1 — координаты конца стержня в естественной подвижной системе координат, помещенной в начале стержня;

ρ — угол чистого вращения.

Для получения направляющих косинусов другого конца ($l'_1 — n'_3$) в формулы (1) — (9) вместо величин x_1, y_1, z_1 следует подставить величины x_2, y_2, z_2 , т. е. координаты начала стержня в естественной системе координат, помещенной в конце стержня.

В рабочий массив $R2$ поочередно для каждого стержня засылаются величины:

	0	1	2	3	4	5	6
$R2$	ρ	x_1	y_1	z_1	x_2	y_2	z_2

Значения величин $x_1, y_1, z_1, x_2, y_2, z_2$ вычисляются по одной из процедур перехода от общей системы координат (естественной неподвижной) к естественным подвижным системам (об этом будет сказано дальше). Для частного случая постоянного направления осей (VII. 5) достаточно иметь координаты x_1, y_1, z_1 , которые равны разности координат концов стержня в общей декартовой ортогональной системе координат.

Этому случаю в помещенной ниже процедуре соответствует признак $c=1$.

Вычисляемые косинусы и длина засылаются в соответствующие ячейки рабочего массива $r[1:23]$ — в общем случае (признак $c=0$), либо $r[1:14]$ (признак $c=1$).

procedure F6; comment в процедуре выполняется формирование рабочей строки r исходных геометрических данных, величина c описывается и вводится в общей программе;

```

begin integer  $u$ ;
  real  $r1, r2, r3, r4, r5, r6, r7, r8, L$ ;
  array  $R2[0:6-3 \times c], r[1:23-9 \times c]$ ;
  begin  $r1 := R2[0]; r2 := \cos(r1); r3 := \sin(r1)$ ;
   $u := 0$ ;
   $r4 := R2[1]; r5 := R2[2]; r6 := R2[3]$ ;
M1:  $r7 := \text{sqrt}(r4 \uparrow 2 + r5 \uparrow 2)$ ;
   $L := \text{sqrt}(r7 \uparrow 2 + r6 \uparrow 2)$ ;
   $r8 := r6/L$ ;
  if  $r7=0$  then begin  $r[u+1] := r[u+5] := r2$ ;
   $r[u+2] := r3; r[u+4] := r3$ ; go to M2 end;
   $r[u+1] := -(r5 \times r2 + r8 \times r4 \times r3)/r7$ ;
   $r[u+2] := (r5 \times r3 - r8 \times r4 \times r2)/r7$ ;
   $r[u+4] := (r4 \times r2 - r8 \times r5 \times r3)/r7$ ;
   $r[u+5] := -(r4 \times r3 + r8 \times r5 \times r2)/r7$ ;
M2:  $r[u+3] := r4/L$ ;
   $r[u+6] := r5/L$ ;
   $r[u+7] := r7 \times r3/L$ ;
   $r[u+8] := r7 \times r2/L$ ;
   $r[u+9] := r8$ ;
  if  $c=1$  then begin  $r[10] := L$ ; go to M3 end;

```



```

if  $u = 0$  then begin  $u := 9$ ;
 $r4 := R2[4]$ ;  $r5 := R2[5]$ ;  $r6 := R2[6]$ ;
go to M1 end;  $r[19] := L$ ;
M3: end
end процедуры.

```

VII. 10. Выше были приведены две процедуры из библиотеки программ расчета пространственных систем с циклической симметрией. В библиотеку входят взаимозаменяемые (переменные) процедуры типа Q , P , PR и основной постоянный блок программы метода деформаций (VI. 18) с включенными в него процедурами $F1$ и $F6$.

Входными данными служат: для регулярных систем типа ребристых куполов, башен и т. п. — высоты и диаметры ярусов, их количество, жесткости и углы чистого вращения стержней, сведения о нагрузках; для систем с нарушениями регулярности, неоднородных и т. п. — более подробная информация: операторы стержней и узлов, координаты узлов, жесткости и углы чистого вращения типовых стержней, нагрузки в узлах.

Процедуры типа Q формируют операторы стержней (в них указываются номера ограничивающих узлов) и операторы узлов (в них указываются номера неизвестных), а также вычисляют координаты узлов по общим геометрическим данным.

Процедуры типа P предназначены для перевода координат узлов из общей системы (цилиндрической, сферической и других типов криволинейных координат) в подвижную систему координат. Эти процедуры охватывают обширные классы сооружений. Так, процедура $P1$ используется для расчета осециклических систем, $P2$ — сферических систем и т. д.

В указанных процедурах типов P и Q структурные особенности циклических систем используются для сокращения исходной информации и упрощения ее подготовки. Для ЭЦВМ это имеет первостепенное значение.

Процедуры типа PR — разложения нагрузок в функциональные ряды — предназначены для установления зависимостей между неизвестными, что позволяет резко уменьшить порядок решаемой системы уравнений. Применение процедур типа PR ограничивается регулярными системами с физической симметрией, в которых циклическим закономерностям подчиняются не только геометрические, но и жесткостные характеристики элементов. Эти процедуры составлены на основе работ [11, 90, 91].

Поскольку процедуры $F1$ и $F6$ не зависят от типа систем координат, выбираемого в каждом конкретном случае, в библиоте-

ку могут быть включены процедуры расчета стержневых систем, узлы которых лежат на поверхностях переноса (торонды, параболонды, гиперболонды и др.).

Процедура определения частот и форм собственных колебаний произвольной системы

VII. 11. В IV. 16—IV. 18 была приведена компактная схема алгоритма определения частот и форм собственных колебаний системы методом ортогонализации последовательных итераций, предназначенная для клавишных автоматов. Этот алгоритм положен также в основу процедуры, предназначенной для ЭЦВМ. Отличие состоит в том, что числовые массивы сильно сжимаются: вместо нескольких матриц (табл. 1, 2, 3...), размеры которых определяются количеством приближений и заранее неизвестны, применяется постоянный массив $y[1:m1 \times t]$, где $m1$ — количество степеней свободы, t — заданное в качестве параметра количество форм колебаний. Для одной формы используется $m1$ ячеек памяти. Кроме того, имеется рабочий массив $R[1:m1]$ для получения $\nu+1$ -го приближения*. Такое распределение памяти не только является экономным, но дает возможность производить вычисления для всех форм колебаний по одним и тем же формулам. Это позволило построить программу для определения любого заданного числа форм и частот колебаний.

Расчетные формулы для ν -го приближения j -той формы ($j=1, 2, \dots, t$) следующие:

$$y_{ij}^{(\nu+1)} = \sum_{k=1}^{m1} Q_k y_{kj}^{(\nu)} \delta_{ik}; \quad (1)$$

$$\lambda_{ij} = y_{ij}^{(\nu)} / y_{ij}^{(\nu+1)}; \quad (2)$$

$$\rho_j = 3,14 \sqrt{\mu \lambda_j}; \quad (3)$$

$$a_{j,j_1} = \sum_{k=1}^{m1} Q_k y_{kj_1} y_{kj}; \quad (4)$$

$$a_{jj_1} = \sum_{k=1}^{m1} Q_k y_{kj}^0 y_{kj_1}; \quad (5)$$

* После выполнения цикла расчета числа из массива R засылаются в ячейки массива y , соответствующие данной форме колебаний, с вытеснением чисел предыдущего приближения.

$$\psi_{ji} = \frac{a_{ji}}{a_{j,j_i}}; \quad (6)$$

$$y_{kj}^{(\nu)} = y_{kj}^0 - \sum_{i_1 < j} \psi_{ji_1} y_{ki_1}. \quad (7)$$

Здесь k, i — порядковые номера сосредоточенных масс (степеней свободы), j, i_1 — номера предшествующих форм колебаний. Остальные обозначения те же, что в IV. 16.

Другая особенность алгоритма состоит в том, что на каждом шаге процесса итерации все ординаты формы колебания делятся на первую ординату, которая таким образом становится равной единице. Благодаря этому устраняется возможность слишком большого увеличения или уменьшения чисел в процессе итерации и отпадает необходимость в масштабировании. На каждом шаге процесса определяется отношение λ — ординат предыдущего и последующего приближений для формы колебаний. Так как первая ордината в обоих приближениях равна единице, то условием подобия форм является их полное совпадение. Поэтому процесс итераций можно прервать, когда λ для всех точек станет близка к единице с заданной погрешностью

$$abs(\lambda - 1) \leq \varepsilon. \quad (8)$$

VII. 12. В процедуре приняты следующие обозначения.

Параметры и константы:

t — заданное число частот и форм колебаний;

$m1$ — число степеней свободы;

$mu(\mu)$ — масштабный коэффициент для жесткостей элементов;

$eps(\varepsilon)$ — заданная погрешность в отклонении от подобия ν и $\nu+1$ приближения для j -той формы колебания.

В скобках указаны соответствующие обозначения в формулах.

Массивы и переменные:

$Q[1:m1]$ — для сосредоточенных грузов (вместо масс);

$A3[1:m1 \uparrow 2]$ — для матрицы перемещений от сил $Q=1$ г, приложенных в направлении исследуемых колебаний;

$y[1:m1 \times t]$ — для ординат форм колебаний (для каждой формы графа размером $1:m1$).

Массивы Q , $A3$ и начальные значения ординат y вводятся или вычисляются во внешнем блоке. Матрицы $A3$ и y представлены в виде линейных массивов.

$R[1:m1]$ — для рабочей графы, в которую последовательно помещаются $\nu+1$ -ые приближения.

$lamb[1:t]$ — массив для определения коэффициентов λ для всех рассчитываемых форм (в этом же массиве получаются частоты ρ);

$psi(\psi)$ — для коэффициента ортогонализации форм;

$a1, a2, (a_{j1}, a_{ji})$ — коэффициенты для определения ψ .

Индексы (счетчики):

j — порядковый номер рассчитываемой формы колебаний;

$j1$ — номера предшествующих форм;

$nu(v)$ — номер приближения;

k, i — порядковые номера сосредоточенных масс (степеней свободы);

$c1, c2$ — номера ячеек, предшествующих в массиве y данной форме колебаний.

Признаки:

$c=0$ — расхождения между приближениями для формы колебаний превосходят заданную точность ε ($c=1$ — в противном случае). Признак c вырабатывается в процедуре.

procedure вычисление частот и форм колебаний; **begin comment** во внешнем блоке описываются переменные **integer** $m1, t$, **real** eps, tu (эти величины вводятся как исходные данные) и массивы

array $A3[1:m1 \uparrow 2], Q, R[1:m1]$,

$y[1:m1 \times z]$;

integer $j, j1, nu, k, i, c, c1, c2$;

real $a1, a2, r, psi$; **array** $lamb[1:t]$;

for $j := 1$ **step** 1 **until** t **do begin**

$c1 := (j-1) \times m1$;

for $i := 1$ **step** 1 **until** $m1$ **do**

$y[c1+i] := y[c1+i]/y[c1+1]$

end j ;

for $r := 1$ **step** 1 **until** t **do begin**

$c1 := (j-1) \times m1$; $nu := 0$;

for $i := 1$ **step** 1 **until** $m1$ **do** $R[i] :=$

$y[c1+i]$;

$M1: j1 := 0$; $c := 1$;

$M2: \text{if } j1 = j-1 \text{ then go to } M3$;

$j1 := j1 + 1$; $c2 := (j1-1) \times m1$; $a1 :=$

$a2 := 0$;

for $k := 1$ **step** 1 **until** $m1$ **do begin**

$a1 := a1 + Q[k] \times y[c2+k] \times y[c2+k]$;

Примечания

Заголовок основного цикла;

$a2 := a2 + Q[k] \times y[c2+k] \times y[c1+k]$ end k ; $psi := a2/a1$; for $k := 1$ step 1 until $m1$ do $y[c1+k] := y[c1+k] - psi \times y[c2+k]$; go to $M2$; $M3$: if $j \neq 1$ then for $k := 1$ step 1 until $m1$ do $y[c1+k] := y[c1+k]/y[c1+1]$; if $j=1 \wedge nu=0$ then go to $M4$; for $i := 1$ step 1 until $m1$ do begin $lamb[j] := R[i]/y[c1+i]$; if $abs(lamb[j]-1) > eps$ then $c := 0$ end i ; if $c \neq 0$ then go to $M5$; $M4$: for $i := 1$ step 1 until $m1$ do begin $c2 := (i-1) \times m1$; $R[i] := 0$; for $k := 1$ step 1 until $m1$ do $R[i] := R[i] + Q[k] \times y[c1+k] \times A3[c2+k]$; if $i=1$ then $a1 := R[1]$; $R[i] := R[i]/a1$ end i ; for $i := 1$ step 1 until $m1$ do begin $r := y[c1+i]$; $y[c1+i] := R[i]$; $R[i] := r$ end i ; $nu := nu + 1$; go to $M1$; $M5$: $lamb[j] := 3,14159 \times sgrt(mu/a1)$ end j end end процедуры.	Вычисление ψ ; вычисление ν -того приближения для ор- динат y ; Определение λ ; Выработка призна- ка r ; Вычисление $\nu+1$ -го приближения для ор- динат $R[i]$; Определение частоты колебаний.
--	---

Достоинством данной процедуры является возможность получения первых форм и частот колебаний (а именно эта задача чаще всего возникает на практике), не определяя последующие формы.

В программе, составленной для расчета каркасных зданий на сейсмические воздействия, используются известные положения норм, и параметр t определяется автоматически.

Алгоритмы расчета систем с односторонними связями

VII. 13. Односторонними называются связи, ограничивающие перемещения в одном направлении и не ограничивающие их в другом. Примерами односторонних связей могут служить опоры без анкеров, воспринимающие сжатие и не воспринимающие растяжения, гибкие связи, воспринимающие растяжение и не вос-

принимающие сжатия и т. п. Если после удаления всех односторонних связей система остается неизменяемой, то она называется системой с односторонними лишними связями. Такие системы были подробно изучены проф. И. М. Рабиновичем [78, 79, 80].

В процессе нагружения некоторые односторонние связи системы замыкаются, другие — размыкаются, вследствие чего расчетная схема сооружения является функцией нагрузки. Система, которая образуется из исходной, если удалить все разомкнутые связи, а замкнутые считать двусторонними, называется рабочей системой. Каждой нагрузке соответствует своя, и притом единственная, рабочая система (доказательство единственности, см. [80], стр. 208). Основная трудность расчета систем с односторонними связями состоит в отыскании рабочей системы.

Рассмотрим сначала видоизменение универсальной программы метода сил для расчета систем с односторонними связями. В. Н. Гордеев предложил [21] применить для расчета систем с односторонними связями аппарат жордановых исключений [44]. В изложенной в VI. 14 программе метода сил при этом меняется блок решения уравнений. На каждом шаге жорданова исключения замыкается одна связь основной системы, т. е. удовлетворяется одно уравнение. Это напоминает механизм последовательных приближений. Однако метод является точным, отличие его состоит в том, что данная связь так и остается замкнутой. Если в процессе расчета в какой-либо из связей возникает «отрицательное» усилие или перемещение (не воспринимаемое односторонней связью), то выполняется шаг жорданова исключения, который размыкает (или замыкает) соответствующую неправильно работающую связь. Как видим, жордановы исключения в некотором смысле напоминают также исключение по Гауссу, так как в расчет вводятся последовательно усложняющиеся статически неопределимые системы. Отличие состоит в том, что процесс этот обратимый, производится не только замыкание, но и размыкание связей. Расчет одновременно выполняет две функции: поиск рабочей системы и ее решение.

Основные этапы, выполняемые по программе расчета системы с односторонними связями методом сил, следующие:

- 1) ввод исходных данных о системе и выбранных неизвестных;
- 2) ввод исходных данных о начальном состоянии системы;
- 3) ввод исходных данных по очередному нагружению;
- 4) очистка массива A для матрицы систем уравнений;

5) загрузка начальных значений свободных членов в массив B ;

6) определение поочередно для всех участков усилий на их концах от единичных значений неизвестных и вычисление по формуле (1) из VI. 13 слагаемых перемещений с накоплением их в ячейках массива A ;

7) определение поочередно для участков усилий на их концах от единичных значений неизвестных и от нагрузки, вычисление по формуле (1) слагаемых «грузовых» перемещений с накоплением их в массиве B ;

8) поиск рабочей системы путем жордановых исключений и получения преобразованной матрицы A на месте нахождения исходной, вычисление неизвестных усилий по направлению связей и получение результатов в массиве B ;

9) печать массива B ;

10) запись нулей в те ячейки массива B , в которых получены перемещения;

11) определение поочередно для всех участков усилий на их концах от единичных значений неизвестных и от нагрузки, вычисление и печать искомых усилий в действительной системе.

Этапы 3—11 выполняются для каждого из последовательных нагружений.

Отличие данной программы от универсальной программы метода сил заключено в этапах 5, 8, 10. Поскольку система с односторонними связями является нелинейно деформируемой, для расчета имеет значение начальное состояние системы (преднапряжение, начальные зазоры в односторонних связях). Поэтому в машину вводится массив начальных значений свободных членов — перемещения по направлению неизвестных в ненагруженной основной системе, которые при расчете системы на каждую из нагрузок записываются в массив B (этап 5). В процессе поиска рабочей системы (этап 8) используется булевский массив $B1$ $[1 : n]$ с числом элементов, равным числу неизвестных n . Каждый элемент этого массива соответствует одной из устраненных связей: если связь замкнута, элемент приобретает значение **true**, если разомкнута — **false**. Перед работой алгоритма все элементы этого массива должны иметь значение **false**.

В результате жордановых исключений матрица A системы уравнений преобразуется так, что ленточную структуру для экономии памяти использовать не удастся. Поэтому для случая системы с односторонними связями порядок ленты принимается равным $h = n$.

Достоинством данной программы является то, что она построена на точном методе и применима для произвольных систем с односторонними связями. Единственное ограничение состоит в том, что односторонние связи должны быть лишними, т. е. при удалении их система должна быть геометрически неизменяемой.

Указанная программа расчета систем с односторонними связями была получена путем небольшой переработки готовой программы метода сил для машины БЭСМ-2М (составленной в соответствии с VI. 14), изменен блок 8 и добавлены команды, соответствующие пунктам 5 и 10. Интересные предложения по применению разработанной Р. А. Резниковым программы СМ-4 метода сил для решения ряда нелинейных задач (в частности, для расчета ферм, включающих стержни, работающие только на растяжение) приведены в книге [81].

VII. 14. Рассмотрим теперь видоизменение обобщенной программы метода деформаций (VI. 18) для систем с односторонними связями.

Проф. И. М. Рабинович предложил находить рабочую систему последовательными попытками ([78], стр. 409). Предполагается, что инженер интуитивно сумеет выбрать рабочую систему, близкую к действительной, и что за одну-две попытки можно будет получить систему, все связи которой замкнуты. Однако для сложных задач это не всегда возможно, потребуется выполнить много приближений и многократно производить расчет статически неопределимой системы.

Указанная схема расчета может быть более эффективна при автоматическом выполнении расчетов на ЭЦВМ. Например, на машине М-20 расчет статически неопределимой системы с 40—50 неизвестными продолжается меньше минуты. Поэтому в неблагоприятных случаях, при многократных расчетах машинное время будет небольшим и исчисляться минутами. Вручную же выполнить многократный расчет больших систем практически невозможно.

Поставим задачу так: разработать общий алгоритм расчета систем с односторонними связями методом деформаций, охватывающий разнообразные случаи, встречающиеся на практике.

Для этого введем понятие ограниченной связи, допускающей свободные перемещения в двух направлениях: t_1 — положительное, t_2 — отрицательное. На рис. 52,а показана связь, ограничивающая вертикальные перемещения, но ограничения могут быть заданы для перемещений в горизонтальном или ином направле-

ниях, для углов поворота и т. п. Пока будем рассматривать опорные ограниченные связи, внешние по отношению к системе.

Любую связь можно трактовать как частный случай ограниченной связи. Назовем us — условное число, заведомо большее возможной упругой деформации по направлению связи, если ее отбросить. us — заменяет ∞ и означает отсутствие связи.

Нетрудно проверить следующие условия:

$t_1 = us$ — односторонняя положительная ограниченная связь (рис. 52, б);

$t_2 = -us$ — односторонняя положительная ограниченная связь (рис. 52, в);

$t_1 = 0$ или (u) $t_2 = 0$ — односторонняя (двусторонняя) связь в обычном понимании, т. е. без зазоров.

Для задания информации об основной системе введем два параметра: c — количество действительных связей (односторонних и двусторонних), n — количество всех закреплений основной системы как действительных, так и фиктивных. Действительные связи могут быть жесткими или упругими с заданными коэффициентами податливости.

Для указания типа действительной связи будем задавать массивы $t_1, t_2 [1 : c]$, в которых записываются соответственно величины зазоров t_1 и t_2 (или нули, если зазоров нет).

Для указания типа закреплений основной системы введем признаки, которые в виде восьмеричных цифр записываются в шкалу, состоящую из нескольких ячеек. Например, в ячейку машины М-20 можно записать 15 признаков — восьмеричных цифр. Различным типам закреплений соответствуют следующие значения признака, который назовем pz :

$pz(i) = 0$ — жесткая связь;

$pz(i) = 1$ — упругая связь;

$pz(i) = 2$ — фиктивное закрепление основной системы.

Под упругой связью понимается связь с заданным коэффициентом податливости (упругая опора, упругое защемление). Тип односторонней жесткой или упругой связи определяется в зависимости от сочетания признаков $pz(i)$ и величин, записанных в массивах t_1 и t_2 :

$t_2[v] = -us \wedge pz(i) = 0$ — положительная односторонняя жесткая связь;

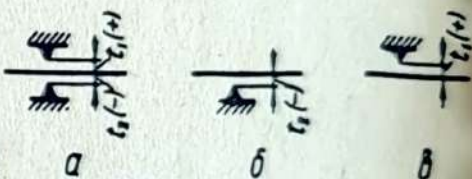


Рис. 52.

$t1[v] = us \wedge pz(i) = 0$ — отрицательная односторонняя жесткая связь;

$t2[v] = -us \wedge pz(i) = 1$ — положительная односторонняя упругая связь;

$t1[v] = us \wedge pz(i) = 1$ — отрицательная односторонняя упругая связь.

Здесь i — порядковый номер закрепления основной системы (нумеруются все связи подряд, независимо от их типа);

v — порядковый номер ограниченной связи (определяется в цикле для тех связей, у которых $pz(i) \neq 2$).

Воспользуемся приемом, который успешно применялся в IV. 20 для автоматизации составления систем уравнений метода сеток. Будем формировать систему уравнений в предположении, что все закрепления основной системы, в том числе и действительные связи, являются фиктивными. В этой системе имеются уравнения для всех односторонних связей, а также двусторонних жестких связей с зазорами — временно эти связи считаются фиктивными. Уравнения не составляются только для двусторонних жестких связей без зазоров ($t_1 = t_2 = 0$).

Подобная абстрактная система уравнений сама по себе смысла не имеет (в общем случае детерминант ее равен нулю), но она очень удобна для анализа ограниченных связей (односторонних и двусторонних) и выбора рабочей системы. Можно сказать, что абстрактная система уравнений содержит в себе множество вероятных рабочих систем. Задача состоит в том, чтобы найти единственную рабочую систему, соответствующую данной нагрузке.

Первое приближение для рабочей системы задается в качестве исходных данных в массиве $t3[1 : c]$ — это свободные перемещения, равные величинам зазоров, которые приводят к контакту со связями рабочей системы. Для разомкнутых односторонних связей в массив $t3$ записываются условные числа us .

Ввод граничных условий в абстрактную систему уравнений производится в зависимости от содержимого массива $t3$.

Если в ячейке $t3[v]$ записано число, не равное us , то это значит, что данное закрепление является действительной связью рабочей системы.

Если эта связь жесткая, то вместо главного коэффициента в массив A записывается 1, вместо свободного члена в массив B записывается величина — $t3[v]$, вместо побочных коэффициентов строки и графы i в массив A записываются нули. Тогда i -тое уравнение будет независимо от остальных, и при решении систе-

мы уравнений неизвестное $B[i]$ будет равно $t3[v]$. К свободным членам остальных уравнений (кроме тех, которые отвечают жестким связям) прибавляются реактивные усилия от заданной величины $t3[v]$. Для этого графу i матрицы A нужно умножить на $t3[v]$ и прибавить к свободным членам, записанным в массиве B .

Если указанная связь упругая, то к соответствующему диагональному члену матрицы A добавляется коэффициент жесткости упругой связи kg (реактивное усилие от смещения по направлению связи на единицу), побочные члены строки остаются без изменения, а массив B преобразуется так же, как для жесткой связи. Коэффициент жесткости kg определяется с помощью процедуры-функции выборки величин из заданного массива.

После решения системы уравнений методом исключения и получения в массиве B значений неизвестных производится их анализ и запись в дополнительный массив $x[1:n]$:

если связь фиктивная, то выполняется присваивание $x[i] := B[i]$;

если связь упругая, то $x[i] = x[i] + B[i]$ (в правой части оператора присваивания — сумма свободного и упругого перемещений);

если связь жесткая, то для нее определяется сумма реактивных усилий от перемещений всех остальных закреплений основной системы, полученных в массиве B , и записывается в ячейку $x[i]$.

Подобные операции выполняются для всех уравнений ($i=1, 2, \dots, n$), в результате чего формируется массив $x[1:n]$.

Далее восстанавливается состояние массивов A и B , каким оно было до решения системы: в массиве A — коэффициенты абстрактной системы уравнений, в массиве B — свободные члены. Для каждой жесткой связи определяется перемещение по ее направлению в предположении, что эта связь отброшена, а перемещения остальных связей заданы, и результат записывается в массив $x[i]$. Эти действия выполняет оператор присваивания:

$$x[i] := -(x[i] + B[i]) / A[d].$$

В правой части оператора записаны: $x[i]$ — сумма реактивных усилий от остальных неизвестных, найденная при предварительном формировании массива x (см. выше), $B[i]$ — свободный член, $A[d]$ — диагональный элемент i -го уравнения.

Значения $x[i]$ для ограниченных связей и значения $t3[v]$ для нового состояния рабочей системы определяются с помощью оператора:


```

if  $x[i] \geq t1[v]$  then begin  $x[i] := t1[v]$ ; go to M1 end
else if  $x[i] \leq t2[v]$  then begin  $x[i] := t2[v]$ ; go to M1 end;
if  $t3[v] = us$  then  $pc := 0$  else  $pc := 1$ ;
 $t3[v] := us$ ; go to M2;
M1: if  $t3[v] = x[i]$  then  $pc := 0$  else  $pc := 1$ ;
 $t3[v] := x[i]$ ;
M2: . . . . . ;

```

В операторе вырабатывается признак pc . Если $pc=0$, то состояние ограниченной связи такое же, как на предыдущем цикле: если она была замкнутой, то и осталась замкнутой и с тем же знаком; если была разомкнутой (этому случаю соответствует условное число us), то и осталась разомкнутой.

Смысл оператора: если неизвестное перемещение $x[i]$ больше или равно положительному зазору, либо меньше или равно отрицательному зазору, то связь замкнута и $x[i]$ присваивается значение соответствующего зазора, после чего вырабатывается признак pc и величине $t3[v]$ присваивается значение $x[i]$; в противном случае вырабатывается признак pc и величине $t3[v]$ присваивается значение us .

Если после обхода всех связей (выполнения цикла для $i := 1, 2, \dots, n$) величина $pc=0$, то это значит, что состояние рабочей системы такое же, как на предыдущем шаге, т. е. получено состояние рабочей системы и неизвестных. Следовательно, если $pc=0$, то процесс обрываем, печатаем величины неизвестных и переходим к завершающему этапу — определению усилий в элементах системы.

Основные этапы расчета дискретной системы с ограниченными, в частном случае — односторонними связями по программе метода деформаций следующие:

1. ввод данных о геометрии, жесткостях элементов и выбранных неизвестных (операторы);
2. ввод массивов $t1, t2, t3 [1 : c]$;
3. формирование в массиве A абстрактной системы уравнений с неопределенными граничными условиями, пересылка массива A во внешнюю память (на барабан);
4. определение порядка системы n и порядка ленты h , параметра c ;
5. ввод исходных данных по очередному загрузению;
6. формирование массива B для свободных членов, пересылка массива B на барабан;
7. введение граничных условий в массивы A и B ;

8. прямой и обратный ход решения системы уравнений, вычисление неизвестных в массиве B ;

9. формирование массива x (первая стадия);

10. запись массивов A и B с барабана в оперативную память;

11. формирование массива x (вторая стадия) и массива $t3$;

12. повторение этапов 7—11, если предыдущее и последующее содержимое массива $t3$ не одинаковы;

13. печать массива x (величин неизвестных);

14. определение суммарных реактивных усилий в закреплении элемента от нагрузки и найденных деформаций, определение усилий в теле элемента (для стержня — в сечениях через заданные интервалы);

15. повторение цикла вычислений (этапы 5—14) для следующего нагружения. Если нагружения исчерпаны, то расчет закончен.

Этапы 5—15 выполняются для каждого из нагружений.

Этапы 1, 3, 4, 5, 6, 8, 12, 15 выполняются так же, как и в обобщенной программе метода деформаций (с включением в нее тех или иных процедур), и описываются теми же операторами. Добавляются операторы обмена между оперативной памятью и барабанами (этапы 3, 6, 10).

Своеобразие программы расчета систем с ограниченными связями заключено в этапах 7 и 9—11, которые ниже представлены в виде отдельных процедур. Процедура-функция выделения признака $pz(i)$ является процедурой-кодом и описывается вне блока; i — формальный параметр — номер неизвестного.

procedure введения граничных условий в матрицу A ;

begin $d := 1$; $h1 := h$; $h2 := h - 1$;

$r1 := n - h2$; $v := 0$;

for $i := 1$ **step** 1 **until** n **do begin**

if $pz(i) = 2$ **then go to** $M5$; $v := v + 1$;

if $t3[v] = us$ **then go to** $M5$;

$r2 := d$; $j := 0$;

if $pz(i) = 1$ **then begin** $A[d] := A[d] + kg$;

go to $M1$ **end**;

$A[d] := 1$; $B[i] := -t3[v]$;

$M1$: **for** $m := 1$ **step** 1 **until** $h2$ **do begin**

if $i + m > n$ **then go to** $M2$; **if** $pz(i + m) = 0$ **then**

go to $M1a$; $B[i + m] := B[i + m] + t3[v] \times A[d + m]$;

Примечания

Описания всех величин даны в основной программе;

в процедуре учитывается симметрия и ленточная структура матрицы A (рис. 32), что несколько усложняет логическую схему;

Преобразование свободных членов;

M1a: if $pz(i) = 0$ then $A[d+m] := 0$;
 M2: if $i-m < 1$ then go to M4;
 if $i > r1+1$ then begin $j := i-r1-m$;
 if $j < 0$ then $j := 0$ end;
 $r2 := r2-h2+j$; if $pz(i-m) = 0$ then
 go to M3;
 $B[i-m] := B[i-m] + t3[v] \times A[r2]$;
 M3: if $pz(i) = 0$ then $A[r2] := 0$;
 M4: end m;
 M5: if $i > r1$ then $h1 := h1-1$; $d := d+h1$;
 end i end процедуры;
 procedure формирование массивов x и $t3$;
 begin switch $S := M2, M4, M7, M8$;
 $h2 := h-1$; $b := 1$; $pc := 0$; $r1 := n-h2$;
 M1: $d := 1$; $h1 := h$; $v := 0$;
 for $i := 1$ step 1 until n do begin
 go to $S[b]$;
 M2: if $pz(i) = 2$ then go to M3; $v := v+1$;
 if $pz(i) = 1$ then $B[i] := B[i] + t3[v]$
 else begin $x[i] := 0$; go to M11 end;
 M3: $x[i] := B[i]$; go to M11;
 M4: if $pz(i) \neq 2$ then begin $v := v+1$;
 $g := 0$ end else $g := 1$;
 $r2 := d$; $j := 0$;
 $v1 := v$; $v2 := v+g$;
 for $m := 1$ step 1 until $h2$ do begin
 if $i+m > n$ then go to M5;
 if $pz(i+m) = 2$ then go to M5;
 $v1 := v1+1$;
 if $pz(i+m) = 0 \wedge t3[v1] \neq us$ then
 $x[i+m] := x[i+m] + B[i] \times A[d+m]$;
 M5: if $i-m < 1$ then go to M6;
 if $pz(i-m) \neq 2$ then $v2 := v2-1$;
 if $i > r1+1$ then begin $j := i-r1-m$;
 if $j < 0$ then $j := 0$ end;
 $r2 := r2-h2+j$;
 if $pz(i-m) = 0 \wedge t3(v2) \neq us$ then
 $x[i-m] := x[i-m] + B[i] \times A[r2]$;
 M6: end m; go to M10;
 M7: if $pz(i) \neq 2$ then $v := v+1$;
 if $pz(i) = 0 \wedge t3[v] \neq us$ then

Очистка строки i для жесткой связи;

Определение номера диагонального элемента;

Описания всех величин даны в основной программе;

Засылка неизвестных из массива B в массив x ;

Определение реактивного усилия в жесткой связи от перемещений других связей;

$x[i] := -(x[i] + B[i]) / A[d];$
go to M10;

M8: **if** $p_z(i) = 2$ **then go to** M11;
 $v := v + 1;$
if $x[i] \geq t1[v]$ **then begin** $x[i] := t1[v];$
go to M9 **end else if** $x[i] \leq t2[v]$ **then**
begin $x[i] := t2[v];$ **go to** M9 **end;**
if $t3[v] = us$ **then** $pc := 0$ **else** $pc := 1;$
 $t3[v] := us;$ **go to** M11;
M9: **if** $t3[v] = x[i]$ **then** $pc := 0$ **else** $pc := 1;$
 $t3[v] := x[i];$ **go to** M11;
M10: **if** $i > r1$ **then** $h1 := h1 - 1;$
 $d := d + h1;$
M11: **end** i
 $b := b + 1;$ **if** $b = 3$ **then**
 $read(A, B);$
if $b \leq 4$ **then go to** M1
end процедуры.

Определение перемещения по направлению жесткой связи, если ее отбросить;
 Второй этап формирования массива x , формирование массива $t3$, выработка x ;

Определение номера диагонального элемента;
 Чтение чисел из внешней памяти в массивы A и B оперативной памяти.

В данном алгоритме можно исключить обмен с внешним запоминающим устройством, но тогда потребуется в каждом цикле заново формировать массив A , т. е. повторять все этапы расчета, начиная с третьего.

Подготовка исходных данных для расчета на машине системы с внешними ограниченными связями отличается следующими особенностями:

1. Нумеруются неизвестные для всех связей, как фиктивных, так и действительных. Вместо нулей для действительных закреплений в ячейки операторов стержней записываются соответствующие номера неизвестных. Порядок системы n теперь получается равным количеству всех закреплений основной системы фиктивных (добавленных) и действительных.

2. Подготавливается исходная информация об ограниченных связях: в ячейки массивов $t1, t2[1 : c]$ записываются величины зазоров, а для односторонних связей — величина зазора в одном направлении и условное число us (или $-us$) — в другом. Условное число принимается таким, чтобы его абсолютное значение во много раз превосходило ожидаемую упругую деформацию. Например, для линейных смещений можно принять $us = 10$ м, для

угла поворота — величину, в тысячу раз больше максимальной приведенной длины стержня l/EI .

3. Подготавливаются данные о первом приближении для рабочей системы: в ячейки массива $I3[1 : c]$ записываются величины свободных перемещений, равные зазорам, которые приводят к контакту с предполагаемыми рабочими связями, либо условные числа (us), если связи разомкнуты.

Достоинством данного алгоритма является использование ленточной структуры матрицы, что позволяет разместить в памяти большие системы уравнений.

Требование, чтобы все односторонние связи были лишними, может быть заменено менее жестким ограничением — формируемые в процессе расчета рабочие системы должны быть неизменяемыми, что по-видимому, будет иметь место во многих реальных задачах (если выбрана надежная конструкция и расчетная схема ей соответствует) даже и в тех случаях, когда не все односторонние связи являются лишними (например, неразрезная балка свободно, без анкеровки, положенная на опоры).

В логическую схему алгоритма нетрудно внести в случае необходимости дополнительные условия. Можно обусловить, например, что реактивные усилия в связях не должны превышать определенной величины, в противном случае связи исключаются из работы. Для этого нужно предусмотреть дополнительную информацию (подобную массивам $I1$ и $I2$) о предельных величинах реактивных усилий, положительных и отрицательных, в действительных связях.

Что же представляет собой рассмотренный алгоритм, является ли он приближенным или точным? Особенность алгоритма состоит в том, что последовательными итерациями определяются не приближенные численные величины неизвестных, а качественные состояния системы, каждое из которых может иметь только два значения: истинное или ложное. После того как истинное состояние рабочей системы найдено, решение системы уравнений выполняется точно методом исключения.

Данный алгоритм является точным, но количество шагов, хотя и конечное, зависит от характера задачи и от того, насколько удачно выбрано первое приближение, т. е. по своему механизму метод является итерационным и относится к группе методов конечных итераций.

VII. 15. Алгоритм, изложенный в предыдущем параграфе, непосредственно применим к системам с ограниченными внешними связями — опорными закреплениями того или иного вида.

В тех случаях, когда ограниченные связи являются внутренними и входят в систему как ее элементы, в алгоритм метода деформаций вносятся изменения, о которых уже говорилось — ограничиваются усилия в связях *. Характерные примеры: стержень, для которого ограничивается величина сжимающего усилия, исходя из условий продольного изгиба, или растягивающего усилия, — из условия предельного растяжения в железобетоне; горизонтальная связь, работающая на трение, усилия в которой не могут превосходить величины вертикальной нагрузки, умноженной на коэффициент трения; вертикальные связи между панелями в панельном здании, работающие только на сжатие.

Номера неизвестных, влияющих на ограниченные внутренние связи, указываются в операторах так же, как и для других элементов системы. Специальный признак (его удобно располагать в коде операции) указывает, что данный оператор описывает ограниченную связь. В массивах $S1$, $S2$ указываются предельные положительные и отрицательные усилия в связях. В массиве $S3$ помещаются данные о рабочей системе — предельное усилие того или иного знака, либо условные числа, если связи не работают. Как видим, подход к решению задачи такой же, как и при внешних ограниченных связях, но вместо перемещений указываются усилия.

В более простом случае, когда внутренние связи могут воспринимать усилия только одного знака, соответствующие условия вводятся в программу и надобность в массивах $S1$ и $S2$ отпадает.

Схема алгоритма аналогична рассмотренной в VII.14 с тем отличием, что в каждом цикле заново формируется система уравнений с исключением тех связей, для которых в массиве $I3$ записано условное число. Процесс расчета продолжается до тех пор, пока не совпадут состояния массива $I3$ для двух последовательных циклов.

Интересной задачей является расчет по указанной схеме крупнопанельного здания, рассматриваемого как система с односторонними связями. Соответствующий алгоритм разработан аспирантом А. Л. Пекарским под руководством автора. Своеобразие задачи состоит в том, что в системе имеются связи различных видов:

* Свободные перемещения вследствие зазоров для внутренних связей обычно отсутствуют. Если, однако, такие зазоры (люфты) имеются, то в случае необходимости их можно учесть, введя в алгоритм дополнительные условия по типу, рассмотренному в VII.14.

1. Связи между панелями по типу I на рис. 53, а. В зависимости от конструкции стыка эти связи могут быть двусторонними (при наличии свариваемых закладных деталей и выпусков арматуры) или односторонними, работающими на сжатие и, за счет трения, на сдвиг *. Если в результате деформаций системы происходит раскрытие шва, то связь полностью выключается из ра-

боты (вертикальная и горизонтальная составляющие усилия равны нулю).

2. Двусторонние связи между стеновыми панелями по типу II, воспринимающие усилия растяжения-сжатия и сдвига.

Связи типов I и II

в зависимости от конструкции стыка могут быть жесткими или упругими. Для рекомендуемого Госстроем замоноличенного стыка связи принимаются жесткими.

3. Односторонние внешние связи между цокольными панелями и грунтовым основанием, рассматриваемым, в соответствии с рекомендациями НИИ оснований и ЦНИИСК, по схеме Винклера (один коэффициент постели) или П. Л. Пастернака (два коэффициента постели на сжатие и сдвиг).

Упругое основание заменяется рядом упруго-податливых односторонних опор. Задача упрощается благодаря тому, что все односторонние связи воспринимают усилия только одного знака — сжатие, а зазоры в этом направлении принимаются равными нулю.

Основная система метода деформаций образована наложением на все узлы, соединяющие панели, и опорные узлы по два фиктивных закрепления, препятствующих вертикальным и горизонтальным перемещениям. При жестких связях типа I и II количество необходимых закреплений уменьшается: если одна из связей типа I замкнута, то в узле имеется два неизвестных перемещения — вертикальное и горизонтальное по направлению закреплений (рис. 53, б); если обе связи I разомкнуты, то вместо

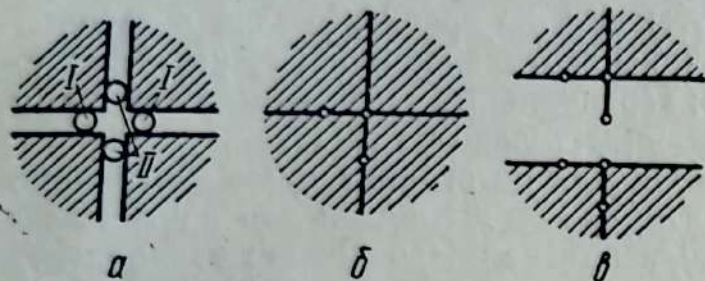


Рис. 53.

* Обычно в расчетах панельных зданий эти связи, независимо от типа стыка, принимаются двусторонними. Однако экспериментально установлено, что в реальных конструкциях возможно раскрытие горизонтальных швов [28].

одного получаем два смежных узла, каждый из которых имеет два перемещения, соответствующие закреплениям (рис. 53, в).

Согласно принятому алгоритму (VII. 14) в массиве A формируется сначала абстрактная система уравнений в предположении, что все связи разомкнуты; затем вводятся граничные условия для рабочей системы, найденной в данном цикле расчета. Если связь I замкнута, то вертикальные (также горизонтальные) перемещения x_i и x_k для двух смежных узлов принимаются одинаковыми. Этому условию соответствует следующее преобразование матрицы A (для определенности $i < k$):

сложить графы и строки i и k и результат поместить в графу и строку k ;

• графу и строку i очистить;

присвоить значения: $A[d] := 1$;

$$A[d+k-i] := -1; B[i] := 0,$$

где d — номер диагонального элемента, $d+k-i$ — номер k -того элемента матрицы A , развернутой в линейный массив (с учетом симметричной ленточной структуры матрицы).

Нетрудно проверить, что для такой матрицы после прямого хода строка i останется без изменения, а после обратного хода получим $x_i = x_k$, что и требуется*.

Для упругой опоры к соответствующему диагональному элементу прибавляется коэффициент жесткости kg . Алгоритм подобен процедуре, рассмотренной в VII. 14.

Рассмотренный способ преобразования матрицы имеет тот недостаток, что приходится хранить большой массив для формирования абстрактной системы уравнений. Этого недостатка можно избежать, если номера неизвестных, соответствующие каждому состоянию рабочей системы, вычислять в программе (можно использовать метод построения операторов, изложенный в VII. 20—VII. 24).

Обобщенный алгоритм расчета стержневых систем на устойчивую прочность

VII. 16. По терминологии проф. Н. В. Корноухова расчетом на устойчивую прочность называется определение усилий в элементах системы по деформированной схеме. Учитываются допол-

* Этого преобразования можно избежать, если жесткую связь I (рис. 53, а) заменить упругой связью с весьма большим коэффициентом жесткости (этот же искусственный прием применим и при внешних жестких односторонних связях), но получаемая при этом система уравнений не всегда хорошо обусловлена.

нительные моменты от нормальных сил, возникающие вследствие упругих деформаций системы, что приводит к нелинейным зависимостям между усилиями и деформациями. Формулы для расчета стержня представлены в привычной форме, но входящие в них величины умножаются на коэффициенты, представляющие собой сложные трансцендентные функции от величины $\nu = \sqrt{\frac{Nl^3}{EI}}$.

В общем случае величины N также зависят от деформаций системы. Однако для упрощения задачи величины N рассматриваются как известные и задаются приближенно (например, определяются из расчета линейной системы), а по ним уже определяются величины ν . Это дает возможность применить принцип наложения с использованием так называемых равносжатых единичных и грузовых состояний. Если величины N , полученные в результате расчета, существенно отличаются от принятых в начале, то расчет предлагается повторить при новых значениях N . Итак, заметим, что в общем случае расчет на устойчивую прочность выполняется последовательными приближениями *.

Книга проф. Н. В. Корноухова [52] была издана в 1949 г., когда электронные вычислительные машины еще не применялись в практике расчета сооружений; естественно поэтому, что излагаемые в ней методы ориентированы на ручное выполнение расчетов и основное внимание обращено на уменьшение их трудоемкости. Отмечается, что для расчета любых систем на устойчивую прочность предпочтительно избрать один метод, именно — метод деформаций (стр. 144), вместе с тем использовано множество специальных способов и приемов, единственная цель которых состоит в уменьшении трудоемкости численного расчета для тех или иных узких классов задач; составлены таблицы значений трансцендентных функций, по которым определяются коэффициенты в расчетных формулах.

Как известно, при выполнении расчетов на ЭЦВМ вместо таблиц функций (например $\sin$, $\cos$ и т. д.) используются стандартные программы по вычислению этих функций. Это выгоднее, чем хранить в памяти большие массивы чисел для таблиц. Поэтому, если бы была поставлена задача запрограммировать для ЭЦВМ в неизменном виде методы расчета на устойчивую проч-

* Применяются термины: моноциклические системы, для которых известны точные значения N ; квази-моноциклические системы, для которых принимаются приближенные значения N , но изменения их после одного цикла расчета незначительны и ими можно пренебречь; полициклические системы, для которых следует выполнить ряд циклов расчета.

ность, разработанные для ручного счета, то пришлось бы в общую программу включить те или иные алгоритмы вычисления трансцендентных функций, изложенные в работе [52]. При этом была бы получена специализированная программа для расчета плоских рам. Это один путь решения задачи.

VII. 17. Вторым путем состоит в том, чтобы использовать быстрое действие ЭЦВМ и ценой увеличения количества вычислений получить обобщенный алгоритм расчета по деформированной схеме, объединяющий произвольные стержневые системы, плоские и пространственные.

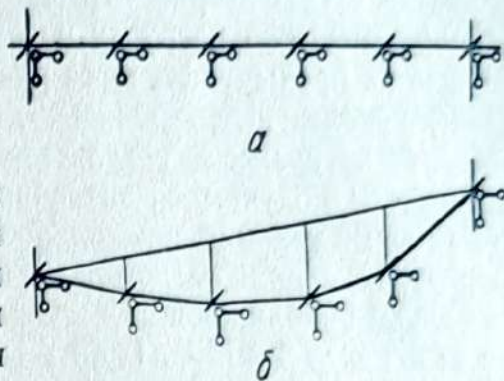


Рис. 54.

При построении основной системы метода деформаций введем фиктивные закрепления не только в узлы системы, но и через определенные интервалы по длине каждого стержня (рис. 54, а). Количество участков выбирается такое, чтобы при расчете по деформированной схеме ось участка можно было принять не криволинейной, а прямой.

Относительная погрешность от такого допущения уже при 4-5 участках будет ничтожной.

В деформированном состоянии расчетная схема стержня имеет полигональную форму (рис. 54, б). Поэтому стержень системы будем называть полигональным. Исходная геометрия стержня также может быть полигональной, причем участки могут иметь разные жесткости. Это дает возможность включить в рассмотрение произвольные системы, состоящие не только из прямолинейных, но и криволинейных стержней и из стержней переменной жесткости.

Укажем следующую итерационную схему расчета:

1. Исходными данными для каждого цикла расчета являются координаты узлов системы и упругие деформации их, найденные на предыдущем шаге. Для основных узлов эти деформации записаны в массиве X , для промежуточных узлов полигонального стержня — в массиве $X1$. Массивы $X1$ по мере их формирования записываются на барабаны в порядке нумерации стержней.

2. Каждый полигональный стержень рассчитывается по заданной деформированной схеме. Для этого с барабана вызывается соответствующий массив $X1$. Система уравнений формирует-

ся в массиве $A1$, свободные члены — в массиве $B1$. Сначала по уточненной геометрии выполняется расчет на единичные деформации крайних узлов. Реактивные усилия крайних узлов от каждой из этих деформаций, найденные после решения системы уравнений в массиве $B1$, являются коэффициентами общей системы уравнений и по мере вычисления записываются в массив A .

Затем стержень рассчитывается на внешнюю нагрузку, включающую местную нагрузку и заданные деформации $X1[i]$, полученные на предыдущем шаге расчета. Таким образом, величины $X1[i]$ используются дважды: для уточнения геометрии и как внешняя нагрузка. В результате решения системы уравнений в массиве $B1$ получаются значения неизвестных деформаций промежуточных узлов полигонального стержня. В массиве $X1$ накапливаются значения деформаций, получаемых в процессе последовательных приближений:

for $i := 1$ **step** 1 **until** $n1$ **do** $X1[i] := X1[i] + B1[i]$; здесь $n1$ — порядок системы уравнений для полигонального стержня.

Определяются реактивные усилия в закреплениях крайних узлов от местной нагрузки и найденных деформаций $X1[i]$ — это свободные члены общей системы уравнений для основных узлов, которые записываются в массив B . После этого массив $X1$ пересылается на барабан.

3. В результате решения общей системы уравнений в массиве B на месте нахождения свободных членов получаем значения неизвестных деформаций основных узлов. В массиве X накапливаются значения деформаций, получаемых в процессе последовательных приближений: **for** $i := 1$ **step** 1 **until** n **do** $X[i] := X[i] + B[i]$. Здесь n — порядок общей системы уравнений.

Этапы 2 и 3 повторяются циклически до тех пор, пока величины, записанные в массивах $B1$ и B , не станут меньше требуемой точности расчета. Если стержневая система удовлетворяет требованиям жесткости, то процесс должен сходиться быстро. Наоборот, медленная сходимость процесса означает, как правило, очень большой, недопустимый рост деформаций по сравнению с найденными по недеформированной схеме. Если процесс расходится, то это означает по существу потерю устойчивости второго рода.

Особенность данного итерационного процесса состоит в том, что каждый шаг расчета выполняется по известному деформированному и напряженному состоянию, найденному на предыдущем шаге. Поэтому отпадает необходимость в приближенном

определении нормальных сил, использовании равносжатых единичных и грузовых состояний, включении в формулы трансцендентных функций. В общем случае можно отказаться также от предпосылки о малых деформациях и считать их конечными.

Все это достигается, правда, ценой значительного увеличения количества неизвестных, что, на первый взгляд, может показаться чрезмерным даже при использовании быстродействующих машин. Однако простой подсчет количества неизвестных для оценки алгоритма будет неправильным. В данном случае большая система уравнений распадается на ряд изолированных систем для полигональных стержней, объединяемых общей системой уравнений для основных узлов. Элементарные системы уравнений и свободные члены для полигональных стержней составляются и решаются поочередно в двух небольших рабочих массивах $A1$ и $B1$, занимающих в памяти немного места. Порядок же общей системы уравнений для основных узлов будет таким же, как при расчете стержневой системы по недеформированной схеме; для каждого узла плоской системы — по три неизвестных, пространственной системы — по шесть неизвестных (или по три неизвестных при шарнирных узлах).

Для ряда частных задач (например, плоских ортогональных рам) можно использовать, как известно, те или иные упрощения и уменьшить количество неизвестных. Но тогда придется отказаться от рассмотренной схемы и пойти по первому пути, о котором говорилось в предыдущем параграфе.

Алгоритм расчета стержневых систем с учетом неупругих свойств железобетона

VII. 18. В НИИЖБе и КиевЗНИИЭПе проводится совместно разработка алгоритмов расчета на ЭЦВМ стержневых систем из железобетона на основе исследований действительной работы конструкций.

А. И. Козачевским под руководством С. М. Крылова и автора настоящей работы разработан алгоритм и программа (для машины М-20) расчета плоских рамных систем с учетом неупругих деформаций железобетона [50, 51, 56]. Для учета физической нелинейности материала применяется метод последовательных приближений, в процессе расчета определяются усилия в эксплуатационной стадии до образования пластических шарниров.

Использована универсальная программа метода деформаций. Своеобразие состоит в разработке процедуры расчета стержня в

упруго-пластической стадии. Формулы для определения жесткостных характеристик составлены на основе рекомендаций СНиП и данных экспериментально-теоретических исследований НИИЖБа.

Исходными данными для расчета являются: размеры и армирование типовых стержней, количество участков, на которое разбивается стержень при определении жесткостных характеристик, величины, характеризующие материалы конструкции (стали и бетона), признак недопущения трещин в тех случаях, когда это необходимо. Остальные данные (операторы стержней, нагрузки и т. д.) те же, что в общей программе метода деформаций.

В каждом цикле выполняется полный расчет системы методом деформаций, для определения жесткостных характеристик используется распределение усилий, полученное в предыдущем цикле. В первом приближении жесткости участков стержней принимаются как для упругой системы, в последующих циклах они уточняются. Исследованиями было установлено, что для практических целей можно ограничиться 5—6 циклами расчета. Поэтому количество циклов задается как исходный параметр.

Количество участков постоянной жесткости, на которые разбивается стержень, принимается большим — 40-50. Довольно точно выявляется изменение жесткостей по длине стержня. Определение реактивных усилий в закреплениях стержня основной системы от единичных деформаций и нагрузки производится по формулам метода сил. В результате расчета по программе метода деформаций на печать выдаются ординаты эпюр моментов и нормальных сил.

Данный алгоритм ввиду огромного количества вычислений (главным образом, при расчете стержней) практически неосуществим при выполнении расчетов вручную, время же для расчета на ЭЦВМ небольшое. Например, расчет пятиэтажной, пятипролетной рамы на машине М-20 занял около 10 мин.

Для проверки алгоритма по программе были выполнены расчеты ряда простейших систем (рам и балок), для которых имеются экспериментальные данные [54, 55]. Результаты получились весьма близкие к экспериментальным.

Выполненные на ЭЦВМ с помощью данной программы исследования рабсты многоэтажных, многопролетных рам с учетом образования трещин выявили ряд интересных закономерностей. Оказалось, что для наиболее распространенных видов загрузки увеличиваются опорные моменты ригелей и соответственно уменьшаются пролетные моменты по сравнению с величинами,

полученными при условии упругой работы конструкции. Перераспределение арматуры между опорными и пролетными сечениями не сопровождается пропорциональным изменением усилий. Это приводит к существенному эффекту при оценке эксплуатационных качеств конструкции.

Дальнейшим развитием рассматриваемого алгоритма является распространение его на стадию работы конструкции, близкую к разрушению. На основе последних исследований НИИЖБа зоны неупругих деформаций арматуры (изображаемые обычно в виде шарниров пластичности) представляются как участки пониженной жесткости, что дает возможность без изменения основной системы проследить ход образования пластических шарниров и возникающее при этом перераспределение усилий. Результаты расчета близки к полученным экспериментально. Таким образом, появляется возможность одним расчетным алгоритмом охватить различные стадии работы железобетона и связать метод расчета упругих систем с методом предельного равновесия.

Возможно также сочетание данного алгоритма с алгоритмом расчета по деформированной схеме, о котором говорилось в предыдущем параграфе. В качестве критерия несущей способности могут быть приняты условия прочности, деформативности и трещиностойкости (ширины раскрытия трещин).

О шаговых алгоритмах расчета нелинейных систем

VII. 19. В последние годы все большее распространение получают шаговые методы решения нелинейных задач; в проектной практике применяются две формы этого метода: метод последовательного изменения жесткостей (Э. Н. Кузнецов [57]) и метод многоступенчатого нагружения (Л. Г. Дмитриев [37]). Исходный параметр задачи, характеризующий деформированное состояние системы (в методе изменения жесткостей — это жесткость, в методе многоступенчатого нагружения — нагрузка), рассматривается как переменная величина, изменяющаяся в процессе расчета *.

Идея метода состоит в следующем: выбирается некоторое исходное состояние, при котором параметр имеет фиксированное значение такое, при котором работу системы можно считать ли-

* В. И. Феодосьев применяет шаговые методы к задачам устойчивости [122, 123, 124], Дж. Аргирис — к расчету двухмерной и трехмерной упругой среды, представленной в виде системы сопряженных четырехгранников [137].

нейной. Далее параметр изменяется ступенями до тех пор, пока не достигнет действительной величины. При каждом изменении параметра выполняется расчет линейной системы в деформированном состоянии, найденном на предыдущем шаге. В методе последовательного изменения жесткостей в начале применяют фиктивную большую жесткость элементов, при которой система работает как линейная, далее жесткости последовательно уменьшаются до требуемой величины. В методе многоступенчатого нагружения в начале принимают малую нагрузку и затем последовательно, ступенями увеличивают ее до заданной величины.

При выводе нелинейной системы уравнений, описывающей задачу, предполагается, что система имеет известные упругие деформации и что неизвестными являются приращения этих деформаций, возникающие от изменения параметра. Система нелинейных уравнений представляется в таком виде (применяется разложение функции в ряды и другие приемы), чтобы можно было выделить и отбросить нелинейные члены, представляющие собой высшие степени от приращений деформаций. Чем меньше приращение деформаций, тем меньше погрешность от отбрасывания этих членов.

Таким образом, на каждом шаге расчета составляется и решается система линейных уравнений, причем исходными данными являются напряженное и деформированное состояния, найденные на предыдущем шаге.

В этом отношении имеется сходство с методом последовательных приближений. Однако природа этих двух методов различна, соответственно разный вид имеют и системы уравнений. В методе последовательных приближений уточнение величин неизвестных выполняется до тех пор, пока невязки в уравнениях не станут меньше требуемой точности расчета. Простым продолжением процесса итерации можно увеличить, при желании, точность расчета. В шаговом методе количество шагов является конечным и задается как исходная величина. Погрешность расчета, допущенная на каждом шаге, зависит от количества шагов, но в процессе последующего расчета не устраняется, т. е. метод является приближенным. Для получения более точного решения нужно увеличить количество шагов (уменьшить приращение параметра) и повторить решение с самого начала. В данном случае речь может идти не о сходимости процесса, а о сходимости метода: решение должно отличаться от точного на величины высшего порядка малости при условии, что изменение параметра становится бесконечно малой величиной. В этом отношении шаговый метод

сходен с методом сеток, в котором точность результатов зависит от густоты сетки.

Для ряда задач шаговый метод имеет существенные преимущества перед методом итерации. Это прежде всего относится к системам из гибких нитей (вантовые системы), для которых применение итерационных методов ограничено и связано с необходимостью применения частных приемов ускорения сходимости. Для таких задач шаговый метод является более общим.

А. С. Городецкий [23] применил метод многоступенчатого загрузки для расчета пространственных железобетонных стержневых систем, в основном изучались непрерывные системы (балки-стенки, оболочки), аппроксимируемые стержневой системой, состоящей из сжато-растянутых стержней, работающих в упруго-пластической стадии.

Применялись зависимости, предложенные Л. И. Онищиком для сжатых стержней и В. И. Мурашевым для растянутых стержней. Использована универсальная программа метода деформаций и процедура-функция расчета стержня в пространственных фермах (VII. 4).

Расчет по программе сводится к многократному расчету пространственной фермы, причем на каждом шаге жесткостные характеристики определяются с учетом перераспределения усилий, полученных на предыдущем шаге. Заметим, что в программу легко могут быть введены другие зависимости для сжато-растянутого стержня; при этом изменится только процедура вычисления жесткостных характеристик.

Расчет по программе имеет несколько вариантов: если необходимо выполнить расчет на заданную нагрузку, исходными данными служат величина этой нагрузки и количество циклов; при расчете по первому предельному состоянию (величина заданной нагрузки заведомо превышает разрушающую) счет прекращается при обращении детерминанта системы в нуль, что служит признаком геометрической изменяемости системы. При расчете по второму (третьему) предельному состоянию задается величина максимально допустимого перемещения узлов (относительного удлинения стержней). Счет прекращается при достижении указанных предельных величин, количество циклов определяет величину нагрузки.

Процедуры построения основной системы метода деформаций и операторов стержней на основе информации о зрительном образе сооружений

VII. 20. Ознакомившись с предыдущими главами, можно заметить, что структура алгоритма в значительной мере зависит от уровня задания исходной информации. Это прежде всего относится к внешним блокам программы, непосредственно перерабатывающим вводимые данные.

Так, в описанном выше универсальном алгоритме метода деформаций для произвольной дискретной системы исходная информация дана в абстрагированном виде — это реактивные усилия в закреплениях каждого элемента от единичных кинематических факторов и местной нагрузки. Это один уровень задания информации. Программа при этом получается предельно простой и короткой. Однако расчет элементов, составляющих систему, требуется выполнить вручную.

Другой уровень задания информации конкретнее и связан с определенным типом сооружения — это координаты узлов, жесткости стержней и нагрузка на них. Для того чтобы вводить информацию на данном уровне, потребовалось разработать целую библиотеку процедур расчета стержня (элемента) в плоских и пространственных системах тех или иных типов.

Таким образом, конкретизация информации, как правило, требует расширения программы, но при этом не автоматизированная часть расчета исключается вовсе или сводится к минимуму.

Но есть и другая сторона вопроса. Дело не только в трудоемкости подготовки исходных данных, но и в том, какова должна быть квалификация работника, готовящего эти данные. Должен ли он быть специалистом по строительной механике, инженером общего профиля или техником?

В рассмотренных выше программах методов сил и деформаций предполагалось, что данные готовит инженер-расчетчик, выбирающий ту или иную основную систему и лишние неизвестные. Операторный способ задания неизвестных или узлов и неизвестных является весьма общим для инженера, знакомого с расчетом статически неопределимых систем; пользование им не составляет труда. Практика показала, что на этой стадии редко бывают ошибки и занимает она мало времени.

Все же весьма желательно иметь программы, в которых эта стадия подготовки выполняется самой машиной. Тогда достаточ-

но будет указать общие геометрические данные и сведения о структуре сооружения, что может сделать и не инженер-расчетчик, а скажем, инженер или архитектор общего профиля или даже техник.

Но дело не только в этом. Представим себе, что программа расчета статически неопределимой системы входит в более общую программу, в которой формируется расчетная схема сооружения на основе информации о проектном задании. Тогда инженер не сможет составить операторы элементов, поскольку ему неизвестна расчетная схема.

Таким образом, программу формирования основной системы и выбора неизвестных можно рассматривать как первый шаг к более широкой автоматизации проектных работ, в которых расчет сооружений будет лишь одним из звеньев.

Попытки к автоматизации расчета основной системы по методу сил делались рядом авторов. Большей частью они идут по пути использования основных систем из консолей с построением логического дерева [48, 135], для одноэтажных рам основная система формируется из трехшарнирных арок [39].

Иным путем идет Р. А. Резников [81], он предлагает вообще отказаться от основной системы и составлять на основе условно экстремального принципа систему из $2M - n$ неизвестных, где M — количество искомых усилий (моментов, нормальных сил и т. д.) на концах прямолинейных участков системы, n — степень статической неопределимости. Часть неизвестных (M) представляет собой усилия, другая часть ($M - n$) — неопределенные множители *.

Такая вариационная постановка задачи позволяет исключить информацию о принятых неизвестных и вводить в машину только данные, относящиеся к геометрии, соединениям узлов системы и нагрузкам.

Количество неизвестных, однако, получается значительно большим, чем при использовании основных систем методов сил и деформаций. Р. А. Резников отмечает, что «к сожалению, все перечисленные преимущества не могут быть в полной мере реализованы на современных ЭЦВМ, так как из-за малого объема оперативной памяти систему уравнений можно эффективно решать только при весьма скромном числе искомых усилий M , в

* Минимизируемая функция представляет собой в этом случае сумму потенциальной энергии системы и левых частей уравнений равновесия, умноженных на неопределенные множители.

несколько раз меньше, чем допустимо по программе СМ-4» ([81], стр. 322).

VII. 21. Автором в 1965 г. был предложен метод задания информации о зрительном образе сооружения на плоской сетке или пространственной решетке. Идея метода проста — задается сетка, состоящая из линий, пересекающихся под некоторыми углами. В первом типе сетки поле, ограниченное четырьмя линиями, ничем не заполняется, во втором типе в каждом поле имеются две перекрещивающиеся линии. Возможны и другие типы сеток. Семейство сеток относится к одному типу, если оно обладает общими топологическими свойствами.

На выбранную сетку наносится рассчитываемая система: на первом типе сетки можно навести произвольную схему рамы, состоящую из стоек и ригелей, на втором типе сетки — комбинированную систему, состоящую из стоек, ригелей и раскосов. Область сетки, в которой находится сооружение, указывается количеством линий в каждом направлении.

Далее зрительный образ кодируется — в специальных шкалах указывается, какой из элементов сетки совпадает со стержнем системы (признак 1 — наличие стержня), а какой нет (признак 0 — отсутствие стержня). Этой информации, а также данных о геометрических и жесткостных характеристиках стержней достаточно для составления логических схем формирования той или иной основной системы метода деформаций, гибкого выбора неизвестных, объединения их в определенных случаях в группы, последовательного формирования операторов стержней. Таким образом, та информация, которая раньше задавалась в качестве исходной — операторы стержней, вырабатывается теперь самой машиной.

Данные о геометрии могут быть заданы одним из трех способов: длинами типовых стержней (для ортогональных систем) или координатами узлов (в общем случае), или по методу Р. А. Резникова * — уравнениями линий (в нашем случае линий сетки).

Тот же принцип применяется для кодировки зрительного образа на пространственной решетке различных форм — ортогональной, цилиндрической, сферической.

На основе указанного принципа был создан ряд развитых программ для расчета плоских и пространственных систем. Итак,

* Этот метод эффективно используется в программе СМ-4 метода сил, [81], стр. 115.

третий уровень задания информации — зрительный образ на сетке. Пока он кодируется числами, можно представить, что в будущем он будет наноситься световым пером на сетке электронно-лучевой трубки [29].

Обращаем внимание на важную особенность: для создания эффективных алгоритмов и программ обязательно нужно найти какие-то закономерности

в изучаемом явлении, какие-то регулярности. Эти регулярности, на первый взгляд, могут быть незаметны, но в некотором отображении они всегда существуют. Если система сама по себе нерегулярна (например, рама, имеющая произвольные разрывы в ригелях и стойках), то следует искать более общие регулярности.

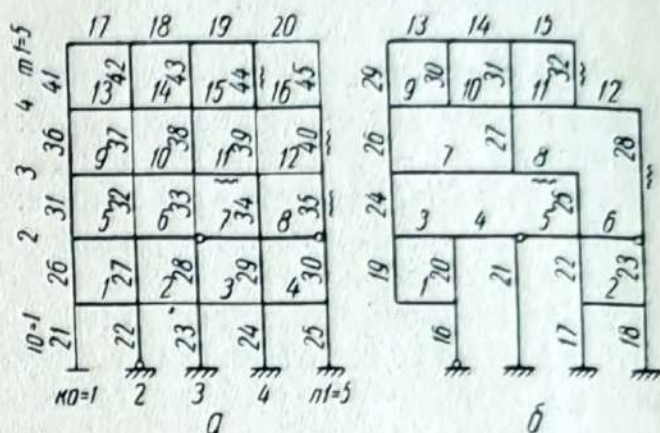


Рис. 55.

Именно это мы и делали, предлагая зрительный образ на сетке — искали регулярное в нерегулярном. В данном случае регулярностью обладает не система, а сетка, на которой она отображается. Общим регулярным свойством большого числа различных систем, с обычной точки зрения нерегулярных, является то, что они отображаются на сетке одного типа.

VII. 22. Рассмотрим плоскую сетку, на которой жирными линиями нанесена рассчитываемая рамная система. На рис. 55, а пронумерованы элементы сетки в последовательности — слева направо и снизу вверх, на рис. 55, б пронумерованы только действительные стержни: сначала ригели (1—15), затем стойки, примыкающие к ригелям первого, второго и последующих уровней. В такой последовательности вводятся исходные данные для стержней и выводятся результаты. На схемах шарниры обозначены кружками, продольно деформируемые стержни — в виде пружинок, помещенных рядом с линиями. Намеренно взят сборный пример, в котором имеются пропущенные ригели и стойки, шарниры, продольно деформируемые стержни (этот пример был использован для отладки программ).

Система не обязательно должна быть ортогональной, хотя этот случай наиболее распространенный; линии сетки могут быть направлены под произвольными углами. Если стойки наклонные,

то все элементы принимаются продольно деформируемыми. Единственное ограничение обусловлено типом сетки: каждое поле сетки должно быть ограничено четырьмя линиями.

Узлы сетки могут быть трех типов: узел внутренней области, крайний узел слева, крайний узел справа. Кроме того, имеются три типа узлов для верхнего уровня — средний узел, крайний узел слева, крайний узел справа.

Область сетки определяется двумя параметрами: $m1$ — количество горизонталей, $n1$ — количество вертикалей. В соответствии с этим вводятся два счетчика: $i0$ — порядковый номер горизонтали, $k0$ — порядковый номер вертикали.

Тип узла определяется условиями:

$i0 = m1$ — узлы верхнего уровня;

$k0 = 1$ — крайние левые узлы;

$k0 = n1$ — крайние правые узлы.

Сочетание этих условий определяют типы узлов, расположенных на границе области, отсутствие этих условий определяет внутренний узел.

Для анализа расчетной схемы введем следующие признаки для элемента сетки:

$psl(v) = 0$ — стержень, отвечающий данному элементу сетки, отсутствует;

$psl(v) = 1$ — стержень продольно недеформируемый;

$psl(v) = 2$ — стержень продольно деформируемый;

v — порядковый номер элемента сетки.

Эти признаки лучше располагать в шкале в виде восьмеричных чисел. В одной ячейке шкалы можно поместить признаки для нескольких следующих подряд элементов сетки, в частности, в ячейке машины М-20 — для пятнадцати элементов. Например, кодировка схемы, представленной на рис. 55, а, выглядит так:

1	0	0	1	1	1	1	1	1	1	2	0	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
1	0	0	1	2	1	0	1	0	2	1	1	1	2	0

В данном случае шкала состоит из трех ячеек: сначала следуют признаки для 1—20-го горизонтальных элементов, затем — для

21—45-го вертикальных элементов. Для наглядности ригели и стойки, относящиеся к одному ярусу, ограничены штрихами.

Введены также признаки для шарниров:

$pc2(v) = 0$ — шарниры отсутствуют;

$pc2(v) = 1$ — шарнир в начале стержня;

$pc2(v) = 2$ — шарнир в конце стержня.

В данном случае v — порядковый номер действительного стержня.

Выделение признаков выполняется с помощью процедур-кодов, которые представим как неописанные булевские процедуры-функции. Будем считать, что в результате выполнения этих процедур в ячейках $pc1$ и $pc2$ будут содержаться нужные нам признаки.

Имея признаки $pc1$ для всех стержней, можно построить процедуры анализа расчетной схемы и последовательного формирования операторов стержней. В самом деле, отсутствие узла и, следовательно, неизвестных соответствует условию, что в узле отсутствуют верхняя и нижняя стойки, левый и правый ригель.

Второе условие: если стержень, соединяющий два узла, продольно недеформируемый, то перемещения этих узлов, направленные вдоль стержня, одинаковы и должны быть обозначены в операторе одним номером.

Третье условие — если нижний уровень яруса не имеет разрывов и состоит из продольно недеформируемых элементов, то в качестве неизвестного может быть принято групповое смещение всех вышележащих ярусов с тем, чтобы перекося получал только данный ярус.

Для формирования номеров неизвестных для узлов двух последовательных уровней рамы резервируем два массива: $B1[1 : 3 \times n1]$ — для нижнего уровня, $B2[1 : 3 \times n1]$ — для верхнего уровня. В этом массиве для каждого узла предусмотрено три ячейки для номеров неизвестных: угла поворота φ , горизонтального смещения x , вертикального смещения y . Если какие-либо из неизвестных равны нулю (нулевое закрепление), то в соответствующих ячейках массивов $B1$ и $B2$ должны быть записаны нули.

Для формирования операторов резервируется один рабочий массив $S1[1 : 6]$, в который последовательно помещаются по шесть номеров неизвестных для стержней (по три неизвестных для каждого конца). При записи неизвестных в операторе ригеля x и y меняются местами. Это дает возможность воспользоваться одной и той же процедурой для расчета ригеля и стойки.

Процедура-функция F из VII. 3 подвергается небольшой переработке с тем, чтобы учесть изменения формул при наличии шарниров. В данном случае вводить для шарниров дополнительные неизвестные нецелесообразно.

Процедуру обработки стержня на всех этапах расчета по признаку $b=0, 1, 2$ (VI. 18) назовем $PR(b)$ и представим как не описанную. После того как оператор стержня сформирован, выполняется процедура $PR(b)$, затем формируется следующий оператор и т. д.

Формирование массивов $B1$ и $B2$ (и одновременно операторов ригелей и стоек) выполняется в следующем порядке:

формируются массивы $B1$ и $B2$ для первого яруса;

величины из массива $B2$ (кроме нулевых) засылаются в массив $B1$;

очищается массив $B2$;

формируются массивы $B1$ и $B2$ для первого яруса;

величины из массива $B2$ засылаются в массив $B1$;

очищается массив $B2$;

формируется массив $B2$ для третьего яруса.

Далее вычисления выполняются аналогично для последующих ярусов. Для этой цели организован цикл с параметром $i0$.

VII. 23. В помещенной ниже процедуре ps построения операторов стержней, помимо указанных выше, приняты следующие обозначения:

$N1$ — количество действительных ригелей;

$N2$ — количество упруго-податливых ригелей.

Будем считать, что эти величины вычисляются во внешнем блоке в соответствии с информацией, указанной в шкалах.

$a1, a2, a3, a4$ — признаки, вырабатываемые в самой процедуре в зависимости от значения признака $ps1(v)$;

$g1, g2$ — порядковые номера действительных стержней — ригелей и стоек;

$g4, g5$ — то же для продольно деформируемых стержней;

$n4, n5, j$ — счетчики;

$r, r1, r2, g3$ — рабочие ячейки;

$t[1:m1]$ — массив признаков горизонталей. Признак $t[i0]=1$ означает, что все ригели нижнего уровня продольно не деформируемы и не имеют разрывов;

procedure pi — процедура, в которой выполняется дополнительная обработка операторов. Если номера неизвестных $s1[3]=s1[6]$, то стержень перемещается поступательно и в нем не возникает никаких усилий. Поэтому процедура pi выполняет в этом

случае присваивание $s1[3] := s1[6] := u$, где u — условное число. Влияние этих неизвестных на стержень учитываться не будет. Если имеется шарнир на одном конце стержня, то в ячейку для соответствующего угла поворота засылается условное число u , в соответствии с чем изменяются коэффициенты в формулах для вычисления реактивных усилий. Если шарниры имеются на двух концах стержня, то жесткость стержня EI принимается равной нулю.

Для лучшего понимания логической структуры данной процедуры, в которой предусмотрено множество возможных ситуаций, целесообразно отладить ее на примере, представленном на рис. 55.

Массивы $B1$, $B2$ и $S1$ для этого примера имеют вид:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$B2$															
$B1$															
$S1$															

Отладка состоит в записи состояний этих массивов в процессе вычислений.

procedure ps ;

begin comment процедура выполняет построение в рабочем массиве $S1 [1 : 6]$ операторов ригелей и стоек плоской рамной системы. Параметры $m1, n1, N1, N2$ и массив $z [1 : m1]$ описываются во внешнем блоке;

integer $n2, n3, n4, n5, g1, g2, g3, g4, i0, k0, r, r1, r2, j, a1, a2, a3, a4$; **integer array** $B1, B2 [1 : 3 \times n1], S1 [1 : 6]$;

boolean procedure $pc1 (v)$; ...;

boolean procedure $pc2 (v)$; ...;

procedure PR ; ...;

procedure $pu (v)$; **integer** v ; **begin if** $S1 [3] = S1 [6]$ **then** $S1 [3] := S1 [6] := u$; **if** $pc2 [v] = 1$ **then** $S1 [1] := u$; **if** $pc2 [v] = 2$ **then** $S1 [4] := u$ **end** процедуры;

Примечание

Процедура, учитывающая шарниры, отсутствие взаимного продольного смещения стержней;


```

 $r := 0; n2 := n1 \times m1; n3 := n2 - m1;$ 
 $g1 := 1; g2 := N1 + 1; g3 := 0; g4 := N2;$ 
for  $i0 := 1$  step 1 until  $m1$  do begin
 $n4 := (i0 - 1) \times n1; n5 := n4 - i0 + 1;$ 
if  $r \neq 0$  then go to M1;
for  $j := 1$  step 1 until  $3 \times n1$  do
 $B2[j] := B1[j] := 0; go to M3;$ 
M1: for  $j := 1$  step 3 until  $3 \times n1$  do begin
if  $B2[j] = 0$  then go to M2;
 $B1[j] := B2[j]; B1[j+1] := B2[j+1];$ 
 $B1[j+2] := B2[j+2]$ 
M2: end j;
for  $j := 1$  step 1 until  $3 \times n1$  do
 $B2[j] := 0;$ 
M3: for  $k0 := 1$  step 1 until  $n1$  do begin
 $i1 := n5 + k0; k1 := n3 + n4 + k0;$ 
 $r1 := (k0 - 1) \times 3 + 1; a1 := a2 := a3 :=$ 
 $= a4 := 2;$ 
if  $k0 = 1$  then begin  $a1 := 0; go to M4$  end;
if  $pc1(i1 - 1) = 0$  then  $a1 := 0$  else
if  $pc1(i1 - 1) = 1$  then  $a1 := 1;$ 
if  $k0 = n1$  then begin  $a2 := 0;$ 
go to M5 end;
M4: if  $pc1(i1) = 0$  then  $a2 := 0;$ 
M5: if  $i0 = m1$  then begin  $a4 := 0; go to M6$ 
end;
if  $pc1(k1 + n1) = 0$  then  $a4 := 0;$ 
M6: if  $pc1(k1) = 0$  then  $a3 := 0$  else;
if  $pc1(k1) = 1$  then  $a3 := 1;$ 
if  $a1 \neq 0$  then go to M7;
if  $a2 \neq 0$  then go to M7; go to M8;
M7: if  $a3 \neq 0$  then go to M9;
if  $a4 \neq 0$  then go to M9;
M8:  $B2[r1] := 0; go to M11;$ 
M9:  $r := r + 1; B2[r1] := r; if a1 = 1$  then
go to M10;
 $r := r + 1; B2[r1 + 1] := r;$ 
M10: if  $a3 = 1$  then go to M11;
 $r := r + 1; B2[r1 + 2] := r;$ 
M11: if  $a1 = 1$  then  $B2[r1 + 1] := B2[r1 - 2];$ 
if  $a3 = 1$  then  $B2[r1 + 2] := B1[r1 + 2];$ 
if  $B2[r1] = 0$  then go to M15;

```

Заголовок основного цикла;

Очистка массивов $B1$ и $B2$

Получение массива $B1$ для следующего яруса

Очистка массива $B2$
Заголовок внутреннего цикла;

Выработка признаков $a1 - a4$;

Операторы условного перехода в зависимости от значений $a1 - a4$;

Формирование массива $B2$, в ячейке r накапливаются номера неизвестных;


```

M12: if  $a1=0$  then go to M14;  $r2 := r1$ ;
M13:  $r2 := r2-3$ ; if  $B2[r2]=0$  then
    go to M13;
     $S1[1] := B2[r2]$ ;  $S1[2] := B2[r2+2]$ ;
     $S1[3] := B2[r2+1]$ ;  $S1[4] := B2[r1]$ ;
     $S1[5] := B2[r1+2]$ ;  $S1[6] := B2[r1+1]$ ;
     $pu(g1)$ ;  $PR$ ;
     $g1 := g1+1$ ; if  $a1=2$  then  $g3 := g3+1$ ;
M14: if  $a3=0$  then go to M15;
    for  $j := 0$  step 1 until 2 do begin
         $S1[j+1] := B1[r1+j]$ ;
         $S1[j+4] := B2[r1+j]$  end;
        if  $t[i0-1] \neq 0$  then  $S1[2] := u$ ;
         $pu(g2)$ ;  $PR$ ;  $g2 := g2+1$ ;
        if  $a3=2$  then  $g4 := g4+1$ ;
M15: end k0
    end i0
end процедуры.

```

Формирование оператора ригеля в массиве $S1$;

Счетчики $g3$ и $g4$ служат для связи с процедурой PR ;

Формирование оператора стойки в массиве $S1$;

Выбор группового смещения.

В данной процедуре все операторы формируются в одном рабочем массиве $S1[1:6]$.

VII. 24. Разработан также второй вариант процедуры ps , в котором формируется весь массив $H1$ операторов без выполнения процедуры PR . Этот вариант менее экономичен, но имеет то достоинство, что выполняется независимо и поэтому может быть включен в имеющиеся готовые программы, в частности в обобщенную программу метода деформаций из VI. 18. До метки M12 второй вариант совпадает с первым. Далее производится засылка номеров неизвестных из массивов $B1$ и $B2$ в массив операторов $H1$ (вместо рабочего массива $S1$). При этом используется процедура-код, поскольку в одну ячейку массива $H1$ помещаются три неизвестных (оператор стержня из 6 неизвестных состоит из двух ячеек).

Условно измененную часть процедуры можно записать следующим образом.

Окончание процедуры ps для второго варианта Примечания

```

    if  $B2[r1]=0$  then go to M15;  $bi := 0$ ;
M12:  $H1[g1]_{1A} := B2[r1]$ ;  $H1[g1]_{2A} :=$ 
     $B2[r1+2]$ ;
     $H1[g1]_{3A} := B2[r1+1]$ ;  $g1 := g1+1$ ;
    if  $a1=0 \vee a2=0 \vee bi=1$  then go to M13;

```

До этого места—совпадение с первым вариантом, процедуры PR и pu из описаний исключаются;

$bi := 1$; go to M12;
 M13: if $a3 = 0$ then go to M15;
 $H1[g2]_{1A} := B1[r1]$; $H1[g2]_{2A} := B1[r1+1]$;
 $H1[g2]_{3A} := B1[r1+2]$; if $t[i0-1] \neq 0$
 then $H1[g2]_{2A} := u$; $g2 := g2 + 1$;

$H1[g2]_{1A} := B2[r1]$; $H1[g2]_{2A} := B2[r1+1]$;
 $H1[g2]_{3A} := B2[r1+2]$; $g2 := g2 + 1$;
 M15: end $k0$ end $i0$;

for $g := 1$ step 1 until $N3$ do
 $g1 := (g-1) \times 2 + 1$;
 if $H1[g1]_{3A} = H1[g1+1]_{3A}$ then
 $H1[g1]_{3A} := H1[g1+1]_{3A} := u$;
 if $pc2(g) = 1$ then $H1[g1]_{1A} := u$;
 if $pc2(g) = 2$ then $H1[g1+1]_{1A} := u$;
 end процедуры.

Индексы 1A, 2A, 3A означают, что речь идет о процедуре-коде, в данном случае — о засылке в 1, 2 и 3 адреса ячейки номеров неизвестных; эта условная запись неалгольного оператора дана для наглядности;
 Метка M14 отсутствует;

$N3$ — общее количество действительных стержней;
 Заключительная обработка массива операторов стержней (заменяет процедуру pu).

VII. 25. По такому же принципу составляются процедуры построения операторов элементов и при других сетках.

Так, для равно-связевых систем целесообразно использовать сетку, в каждом поле которой дополнительно имеется две перекрещивающиеся линии. На рис. 56 приведен сборный пример,

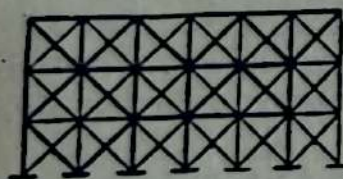


Рис. 56.

в котором жирными линиями отмечены действительные стержни. Как видим, на такой сетке могут быть отображены треугольная, крестовая, ромбическая, полураскосная решетка, рамный каркас. Нумерация элементов сетки принимается слева направо и снизу вверх: сначала ригелей, затем стоек, после чего раскосов (парами — восходящий и нисходящий). В этой последовательности образ кодируется в шкалах.

Отсутствие узла системы (следовательно, и неизвестных) соответствует случаям, когда в узле сходятся только два стержня, расположенные на одной прямой, либо все стержни отсутствуют.

Построение массивов $B1$ и $B2$ для номеров неизвестных нижнего и верхнего уровней яруса $i0$ рамно-связевой системы, формирование операторов ригелей, стоек и раскосов представлены в виде отдельных процедур. В общей программе они работают совместно.

Здесь ограничимся тем, что приведем процедуру PB для построения массивов $B1$ и $B2$. Описание и ввод параметров $m1, n1$, а также присваивание значений $i0$ ($i0 = 1, 2, \dots, m1$) выполняются в охватывающем блоке. В процедуре применяются булевские переменные, признаки те же, что в VII. 22.

Во внешнем блоке вычисляются вспомогательные величины:

$$n2 := n1 \times m1; n3 := n2 - m1; n4 := (i0 - 1) \times n1;$$

$$n5 := n4 - i0 + 1;$$

$$n6 := n3 + n2 + 2 \times n5; n7 := (n1 - 1) \times 2.$$

procedure PB ;

begin integer $k0, i1, k1, k2, k3, j, r, r1, r2$;

integer array $B1, B2[1 : 3 \times n1]$;

boolean $b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15$;

if $i0 = 1$ then begin for $j := 1$ step 1

until $3 \times n1$ do $B1[j] := B2[j] := 0$;

$r := 0$; go to $M1$ end;

for $j := 1$ step 3 until $3 \times n1$ do,

if $B2[j] \neq 0$ then begin $B1[j] := B2[j]$;

$B1[j+1] := B2[j+1]$;

$B1[j+2] := B2[j+2]$ end;

for $j := 1$ step 1 until $3 \times n1$ do $B2[j] := 0$;

$M1$: for $k0 := 1$ step 1 until $n1$ do begin

$i1 := n5 + k0$; $k1 := n3 + n4 + k0$;

$k3 := n6 + 2 \times k0$; $k2 := k3 - 1$;

$b1 := k0 = 1$; $b2 := k0 = n1$;

$b3 := i0 = m1$;

Булевские величины

$b4 := k0 \neq 1 \wedge p_{c1}(i1 - 1) = 0$; $b5 := k0 \neq n1 \wedge p_{c1}(i1) = 0$;

$b6 := p_{c1}(k1) = 0$; $b7 := i0 \neq m1 \wedge p_{c1}(k1 + n1) = 0$;

$b8 := k0 \neq n1 \wedge p_{c1}(k3) = 0$; $b9 := k0 \neq 1 \wedge p_{c1}(k3 - 3) = 0$

Примечания

Очистка массивов $B1$ и $B2$;

Получение массива $B1$ для следующего яруса;

Очистка массива $B2$;

$i1, k1, k2, k3$ — порядковые номера ригеля, стойки, восходящего и нисходящего раскосов;

Условия, определяющие контурные узлы;

Условия, выражающие отсутствие ригелей, стоек, раскосов, сходящихся в узле;

$b10 := k0 \neq n1 \wedge i0 \neq m1 \wedge p1 (k2 + n7)$
 $= 0;$
 $b11 := k0 \neq 1 \wedge i \neq m1 \wedge p1 (k2 + n7 - 1)$
 $= 0;$
 $b12 := k0 \neq 1 \wedge pc (i1 - 1) = 2; b13 :=$
 $p1 (k - 1) = 2;$
 if $b3$ then begin if $b1$ then go to
 $M2$ else if $b2$ then go to $M3$
 else go to $M4$ end else
 if $b1$ then go to $M5$ else if $b2$
 then go to $M6$ else go to $M7;$
 $M2: go to if b5 \wedge b6 \wedge b8 then M12 else M8;$
 $M3: go to if b4 \wedge b6 \wedge b9 then M12 else M8;$
 $M4: go to if b6 \wedge b8 \wedge b9 then M12 else M8;$
 $M5: go to if b4 \wedge b8 \wedge b10 then M12 else M8;$
 $M6: go to if b5 \wedge b9 \wedge b11 then M12 else M8;$
 $M7: b14 := b8 \wedge b9 \wedge b10 \wedge b11;$
 $b15 := b4 \wedge b5 \wedge b6 \wedge b7;$
 go to if $(b6 \wedge b7 \wedge b14) \vee (b4 \wedge b5 \wedge b14)$
 $\vee (b15 \wedge b8 \wedge b11) \vee (b15 \wedge b9 \wedge b10)$ then
 $M12;$
 $M8: r := r + 1; r1 := (k0 - 1) \times 3 + 1;$
 $B2[r1] := r;$
 if $b1 \vee b4 \vee b12$ then begin $r := r + 1;$
 $B2[r1 + 1] := r; go to M11 end;$
 $r2 := r1 - 2;$
 $M9: if B2[r2] \neq 0 then go to M10;$
 $r2 := r2 - 3; go to M9;$
 $M10: B2[r1 + 1] := B2[r2];$
 $M11: if b6 \vee b13 then begin r := r + 1;$
 $B2[r1 + 2] := r end else B2[r1 + 2] :=$
 $= B1[r1 + 2];$
 $M12: end k0$
 end процедуры.

Условия продольной
 деформативности;
 Условия выбора мет-
 ки оператора в зави-
 симости от положе-
 ния узла;

Проверки на отсут-
 ствие узлов (и неиз-
 вестных) верхнего
 уровня;
 Проверки на отсут-
 ствие крайних слева
 или справа узлов;
 Проверка на отсут-
 ствие внутренних уз-
 лов.

VII. 26. Тот же принцип кодирования зрительного образа сох-
 раняется для пространственных задач. Для стержневых оболочек
 сетка наносится на поверхность того или иного вида, для про-
 странственных рамных или рамно-связевых каркасов — исполь-
 зуется решетка, составленная из пересекающихся сеток рассмот-
 ренных выше типов. Для дискретных систем, составленных из
 пластинок, вместо стержней нумеруются поля, а в шкалах коди-
 руется наличие или отсутствие пластинок в полях.

При использовании программами, основанными на информации о зрительном образе, возможна ситуация, при которой одноименные графы и строки матрицы системы уравнений оказываются нулевыми (например, если все стержни в узле имеют шарниры). Чтобы избежать неопределенности (деление нуля на ноль), в процедуру прямого хода решения систем уравнений добавляется оператор —

if $A[d]=0$ then $A[d] := 1$.

Если диагональный элемент матрицы равен нулю, то в соответствующую ячейку засылается единица.

Логическая схема процедуры формирования операторов упрощается, если рассматривать более узкие классы задач. Так, для регулярной плоской рамы (без разрывов ригелей и стоек) была составлена процедура вычисления номеров неизвестных по простым формулам и формирования операторов ригелей и стоек (реализована на машинах М-20, БЭСМ-2М, НАИРИ). Рама указанного типа изображена на рис. 57, пунктиром обозначены элементы, дополняющие контур до прямоугольника. Эти элементы представляются как фиктивные стержни с нулевыми жесткостями $EY=0$. Благодаря этому отпадает необходимость в кодировании зрительного образа. Продольная деформативность стержней в программе не учитывается. Нумерация стержней: слева направо и снизу вверх; сначала ригелей, затем стоек.

В помещенной ниже процедуре *ps* определения номеров неизвестных для стержней (ригелей и стоек) приняты следующие обозначения.

Параметры:

m1 — количество этажей рамы;

n1 — количество пролетов.

Переменные:

N1 — общее количество ригелей;

N2 — общее количество стоек;

uz — порядковый номер стержня (при сквозной нумерации, см. рис. 57);

rig — порядковый номер ригеля;

st — порядковый номер стойки;



Рис. 57.

k — текущий номер яруса (снизу вверх);
 $x1$ — угол поворота на левом конце ригеля (нижнем конце стойки);
 $x3$ — угол поворота на правом конце ригеля (верхнем конце стойки);
 $x2$ — номер группового смещения яруса (для стоек).

Массивы:

$B[1 : m1]$ — для записи номеров линейных смещений ярусов.

Процедуры:

procedure $PR(b)$ — неописанная процедура обработки стержня на всех этапах расчета по признаку $b=0, 1, 2$. После того, как определены номера неизвестных для стержня (сформирован оператор), выполняется процедура $PR(b)$, затем определяются номера неизвестных для следующего стержня и т. д.

```

procedure  $ps$ ;
begin procedure  $PR(b); \dots$ ;
 $N1 := m1 \times n1$ ;  $N2 := N1 + m1$ ;
 $k := 1$ ;  $uz := 0$ ;
for  $rig := 1$  step 1 until  $N1$  do begin
 $uz := uz + 1$ ; if  $rig > n1 \times k$  then  $k := k + 1$ ;
 $x3 := rig + 2 \times k$ ;  $x1 := x3 - 1$ ;
 $PR$ ; end  $rig$ ;
 $k := 1$ ;  $x2 := 1$ ;  $B[k] := x2$ ;
for  $st := 1$  step 1 until  $N2$  do begin
 $uz := uz + 1$ ; if  $st > k \times (n1 + 1)$  then begin
 $k := k + 1$ ;  $x2 := x2 + n1 + 2$ ;  $B[k] := x2$  end;
 $x3 := st + k$ ;
if  $k = 1$  then  $x1 := 0$  else  $x1 := x3 - n1 - 2$ ;
 $PR$  end  $st$  end процедуры.
  
```

Данная процедура очень проста, но относится к расчету рамных каркасов определенного типа.

Для общего случая процедура упрощается, если в каждом узле (за исключением опорных) всегда принимать: для плоской системы — три степени, для пространственной системы — шесть степеней свободы. Однако, в этом случае резко возрастает количество неизвестных. Эти процедуры уступили место более гибким «сеточным» процедурам, изложенным выше.

Зрительный образ на сетке может быть эффективно использован для автоматизации ряда процессов проектирования: может быть дана информация не только об элементах, ограничивающих поля сетки или решетки, но и о самих полях — привязка нагрузок, отверстий, характер заполнения. Разработка этой темы вы-

полняется в настоящее время аспирантом В. В. Артеменко под руководством автора.

Мы придаем этому направлению большое значение, так как расчеты систем составляют лишь небольшую долю от всего комплекса расчетов конструкций здания. Такие расчеты как сбор нагрузок, расчет второстепенных балок и плит, предварительный подбор сечений элементов системы, подбор сечений элементов после выполнения статических (динамических) расчетов относятся обычно к проектированию и выполняются вручную. Если такие расчеты и автоматизируются, то пока отдельно, без связи с расчетом общей системы здания.

ЛИТЕРАТУРА

1. Агеев М. И. Основы алгоритмического языка АЛГОЛ-60., М., изд. Вычислительного центра АН СССР, 1964.
2. Алгоритмический язык АЛГОЛ-60, пересмотренное сообщение. М., «Мир», 1965.
3. Бабенко Л. П., Довгополая Л. И., Корниенко Г. М., Ющенко Е. Л. Система автоматического программирования для машины М-20. Транслятор с адресного языка. К., «Наукова думка», 1965.
4. Бернадский Н. М. Символический расчет жестких стержневых систем. Труды Среднеазиатского опытно-исследовательского института, Ташкент, 1929.
5. Боттенбрух Г. Структура АЛГОЛ-60 и его использование. М., Изд-во иностранной литературы, 1963.
6. Березин И. С., Жидков Н. П. Методы вычислений. Физматгиз, т. 1, 1959, т. 2, 1960.
7. Брудно А. Л. Введение в программирование в содержательных обозначениях. М., «Наука», 1955.
8. Вавилова А. С., Жевелева И. С. Электронные вычислительные машины за рубежом. М., Машгиз, 1962.
9. Вазов В., Форсайт Дж. Разностные методы решения дифференциальных уравнений в частных производных. М., Изд-во иностранной литературы, 1963.
10. Варвак П. М. Развитие и приложение метода сеток к расчету пластинок, Изд-во АН УССР, ч. I, 1949, ч. II, 1952.
11. Вайнберг Д. В., Чудновский В. Г. Расчет пространственных рам. К., Госстройиздат УССР, 1964.
12. Вайнберг Д. В., Синявский А. Л. Дискретный анализ теории пластин и оболочек. Труды 6-й Всесоюзной конференции по теории пластин и оболочек, Баку, 1966.
13. Вайнштейн Ф. А., Левит В. А. Типовая программа механизированного расчета по определению периодов и форм свободных колебаний систем с произвольным числом степеней свободы. «Вычислительная и организационная техника в строительстве», М., изд. Гипротис, вып. 7, 1966.
14. Гильман Г. Б. Расчет тонких упругих пологих оболочек с использованием стержневых аппроксимаций. Сб. «Применение электронных математических машин в строительной механике». К., «Наукова думка», 1968.
15. Геммерлинг А. В. Вопросы прочности и устойчивости строительных конструкций. Труды ЦНИИСК, вып. 7. Расчет конструкций, работающих в упруго-пластической стадии. М., Госстройиздат, 1961.
16. Геммерлинг А. В. О расчете стержневых конструкций, работающих в упруго-пластической стадии. Сб. «Проблемы устойчивости в строительной механике». М., Стройиздат, 1965.

17. Гликин И. Д. Определение собственных частот колебаний вантовых покрытий с применением средств механизации. «Вопросы современного строительства». К., «Будівельник», 1964.
18. Глушков В. М. Введение в кибернетику. К., «Наукова думка», 1964.
19. Гнеденко Б. В., Королюк В. С., Ющенко Е. Л. Элементы программирования. М., Физматгиз, 1961.
20. Годунов С. К., Рябенький В. С. Введение в теорию разностных схем. М., Физматгиз, 1962.
21. Гордеев В. Н. Алгоритмы для расчета систем с односторонними лишними связями. Сб. «Применение электронных математических машин в строительной механике», К., «Наукова думка», 1968.
22. Гордеев В. Н., Гринберг М. Л. Методика подготовки инженерных расчетов к механизированному выполнению. Сб. I (11) «Вычислительная и организационная техника в строительстве и проектировании», М., изд. Гипротис, 1964.
23. Городецкий А. С. К расчету тонкостенных железобетонных конструкций в неупругой стадии. Сб. трудов НИИСК «Строительные конструкции», вып. 3., К., «Будівельник», 1965.
24. Гофман Ш. М. Расчет плоских и пространственных рам способом группового последовательного уравнивания. Труды Института сооружений АН УзССР, вып. 1, Ташкент, 1949.
25. Гофман Ш. М. Матричные формулы для расчета некоторых сложных плоских рамных систем. Труды Института сооружений АН УзССР, Ташкент, 1954, № 4.
26. Гофман Ш. М. Расчет плоских рам способами последовательного уравнивания. Труды Ташкентского института инженеров железнодорожного транспорта, вып. XV, Ташкент, 1962.
27. Гребень Е. С. Вопросы матричного расчета многократно статически неопределимых систем. Прочность и колебания конструкций. Сб. трудов ЛИИЖТ, вып. 172, Л., 1960.
28. Григорьев Г. М., Конструкции жилых зданий на подрабатываемых территориях. К., «Будівельник», 1965.
29. Григорьев Э. П. Средства взаимосвязи человека и машины в автоматизированном процессе проектирования. «Вычислительная и организационная техника в строительстве и проектировании», М., изд. Гипротис, вып. 5, 1965.
30. Губерман И. О. Упрощенная схема решения систем линейных алгебраических уравнений по способу Гаусса. Институт строительной механики АН УССР. Информационные материалы, К., 1957, № 10.
31. Гутер Р. С., Арлазаров В. Л., Усков А. В. Практика программирования. М., «Наука», 1965.
32. Дехтярюк Е. С., Романенко Ф. А., Синявский А. Л. Численное решение плоской задачи для многосвязных областей. Сб. «ЭЦВМ в строительной механике». М.-Л., Стройиздат, 1966.
33. Длугач М. И. Розрахункова модель методу сіток. «Прикладна механіка», вип. 3, т. II, 1962.
34. Длугач М. И. Основні положення розрахунку циліндричних оболонок з прямокутними отворами методом скінчених різниць. «Прикладна механіка», т. VI, вип. 3, 1960.
35. Длугач М. И., Шинкар А. И. Застосування електронних обчислювальних машин при рішенні бігармонічної задачі. «Прикладна механіка», вип. 2, т. VIII, 1962.
36. Длугач М. И. Метод сеток в смешанной плоской задаче теории упругости. К., «Наукова думка», 1964.

37. Дмитриев Л. Г., Сосис П. М. Программирование расчета пространственных конструкций. К., Госстройиздат, 1963.
38. Дмитриев Л. Г., Касилов А. В. Вантовые покрытия, К., «Будівельник», 1968.
39. Дукарский О. М., Лавитман В. С. Расчет рам на электронных машинах. М., Стройиздат, 1965.
40. Егупов В. К. Расчет зданий на прочность, устойчивость и колебания, К., «Будівельник» 1965.
41. Ершов А. П., Кожухин Г. И., Волошин Ю. М. Входной язык для систем автоматического программирования. Сибирское отделение АН СССР, Новосибирск, 1964.
42. Жемочкин Б. Н. Расчет рам. М., Госстройиздат, 1933, изд. 2-е, 1965.
43. Загускин В. Л. Справочник по численным методам решения уравнений. М., Физматгиз, 1960.
44. Зуховицкий С. И., Авдеева Л. И. Линейное и выпуклое программирование. М., «Наука», 1964.
45. Керопян К. К., Чеголин П. П. Электрическое моделирование в строительной механике. М., Госстройиздат, 1963.
46. Китов А. И., Криницкий Н. А. Электронные цифровые машины и программирование. М., Физматгиз, 1959.
47. Китов А. И. Программирование информационно-логических задач. М., «Советское радио», 1967.
48. Китов Ю. П. Опыт программирования расчетов сложных рам на ЭЦВМ УРАЛ-1 с использованием элементов топологии. Сб. «ЭЦВМ в строительной механике». М.-Л., Стройиздат, 1966.
49. Кимельфельд И. Э. Классификация циклических систем в связи с постановкой их расчета на ЭЦВМ. «Промышленное строительство и инженерные сооружения», 1966, № 3.
50. Козачевский А. И. Расчет перераспределения усилий в сложных стержневых системах с учетом неупругих деформаций железобетона. «Вычислительная и организационная техника в строительстве и проектировании», вып. 6, М., изд. Гипротис, 1965.
51. Козачевский А. И., Сосис П. М. Расчет на ЭЦВМ рамных каркасов с учетом неупругих деформаций железобетона. «Промышленное строительство и инженерные сооружения». К., 1966, № 2.
52. Корноухов Н. В. Прочность и устойчивость стержневых систем. М., Госстройиздат, 1949.
53. Корчинский И. Л., Поляков С. В. и др. Основы проектирования зданий в сейсмических районах. М., Госстройиздат, 1961.
54. Красовская Т. А. Исследование работы двухшарнирных железобетонных рам. Труды МИИТ, вып. 171, М., Трансжелдориздат, 1961.
55. Крылов С. М. Перераспределение усилий в статически неопределимых железобетонных конструкциях. М., Стройиздат, 1964.
56. Крылов С. М., Козачевский А. И. Применение ЭЦВМ для расчета сложных стержневых систем с учетом неупругих свойств железобетона. «Бетон и железобетон», 1966, № 1.
57. Кузнецов Э. Н. Некоторые вопросы расчета вантовых покрытий. Сб. «Висячие покрытия», М., Госстройиздат, 1962.
58. Лавров С. С. Универсальный язык программирования АЛГОЛ-60. М., «Наука», 1964.
59. Левин М. А. Построение дискретной модели анизотропного тела в виде регулярной стержневой системы. Сб. «Применение электронных математических машин в строительной механике», К., «Наукова думка», 1968.

60. Ледли Р. С. Программирование и использование цифровых вычислительных машин. М., «Мир», 1966.
61. Лосаберидзе А. А. Расчет арочных плотин. Тбилиси, «Мецниереба», 1966.
62. Лопищев А. М. Экстремальная теорема для гиперэллипсоида и ее применение к решению системы линейных уравнений. Труды семинара по векторному и тензорному анализу, вып. 9, М., 1956.
63. Ляшенко В. Ф. Программирование для цифровых вычислительных машин М-20, БЭСМ-3М, БЭСМ-4, М-220. М., «Советское радио», 1967.
64. Магарик В. А., Нагорный Н. М. Система команд универсальной цифровой автоматической машины БЭСМ-2. М., изд. Вычислительного центра АН СССР, 1960.
65. Мак-Кракен Дж. Программирование на АЛГОЛе. М., «Мир», 1964.
66. Мальцев А. И. Алгоритмы и рекурсивные функции. М., «Наука», 1965.
67. Механизация и автоматизация расчетов в строительном проектировании (из опыта проектных организаций Госстроя СССР). М., изд. Гипротис, 1963.
68. Нарец Л. К. Расчет статически неопределимых систем на малых вычислительных машинах. М., Госстройиздат, 1958.
69. Нарец Л. К. Новые вопросы статики, динамики и устойчивости балок, возникающие в связи с появлением вычислительных машин непрерывного и дискретного действия. Таллин, изд. Таллинского политехнического института, 1963.
70. Папкович А. Ф. Строительная механика корабля, ч. I, т. I, М., «Морской транспорт», 1945.
71. Погужев А. И. Метод последовательного приближения в применении к расчету рам при вертикальной нагрузке. М., Гостехиздат, 1931.
72. Попов Б. Н., Степанов В. А., Стишева А. Г., Травникова Н. А. Программирующая программа. «Журнал вычислительной математики и математической физики», т. 4, 1964, № 1.
73. Применение малых средств вычислительной техники для расчетов строительных конструкций. М., изд. Гипротис, серия 11—50, 1965.
74. Пухов Г. Е. Электрическое моделирование стержневых и тонкостенных конструкций. К., Изд-во АН УССР, 1960.
75. Пухов Г. Е., Васильев В. В., Степанов А. Е., Токарева О. Н. Электрическое моделирование задач строительной механики. К., Изд-во АН УССР, 1963.
76. Рабинович И. М. Контрольные равенства для фокусных отношений, вытекающие из принципа взаимности. Сб. «Исследования по теории сооружений», М., Госстройиздат, 1936.
77. Рабинович И. М. Методы расчета рам, ч. I, М., Госстройиздат, 1933; ч. II, М., Госстройиздат, 1934, ч. III, ОНТИ, 1937.
78. Рабинович И. М. Курс строительной механики стержневых систем, ч. II, М., Госстройиздат, 1954.
79. Рабинович И. М. Некоторые вопросы теории сооружений, содержащих односторонние связи. Инженерный сборник. М., Изд-во АН СССР, т. VI, 1960.
80. Рабинович И. М. К задаче расчета статически неопределимых систем с односторонними связями (доказательство единственности решения). Сб. «Исследования по теории сооружений», вып. X, М., Госстройиздат, 1961.
81. Резников Р. А. Методы решения задач строительной механики на электронных цифровых машинах. М., Стройиздат, 1964.
82. Ржаницын А. Р. Представление сплошного изотропного тела в виде шарнирно-стержневой системы. Сб. «Исследования по вопросам строительной механики и теории пластичности». М., Госстройиздат, 1956.
83. Рогоцкий С. А. Расчет плоских и пространственных рам методом последовательных приближений (способ Харди Кросса). М., ГОНТИ, 1939.

84. *Розицкий С. А.* Расчет рам. М., Машигиз, 1948.
85. *Розенблат Г. И.* Расчет балки на упругих опорах методом фокусных коэффициентов. Сб. «Исследования по теории сооружений», вып. V, М., Госстройиздат, 1951.
86. *Розин Л. А.* Об аналогии метода Максвелла-Мора в теории упругости (плоская задача); формулы Максвелла-Мора в теории оболочек. Исследования по упругости и пластичности. Сб. 3, Л., Изд-во ЛГУ, 1964.
87. *Розин Л. А.* Некоторые вопросы расчленения и дискретизации уравнений теории оболочек. Исследования по упругости и пластичности. Сб. 4, Л., Изд-во ЛГУ, 1965.
88. *Ривкин С. А.* Приближенный метод расчета свободных рам. Новое в строительстве и технике. Сб. «Строительные конструкции», вып. VII, К., Госстройиздат, 1955.
89. *Сальвадори М. Дж.* Численные методы в технике. М., Изд-во иностранной литературы, 1955.
90. *Сегаль А. И.* Упрощение при расчете циклических систем. «Проект и стандарт», 1937, № 8.
91. *Сегаль А. И.* Расчет циклически несимметричных систем. Сб. «Расчет пространственных конструкций». Вып. II, М., Госстройиздат, 1951.
92. *Смирнов А. Ф.* Статическая и динамическая устойчивость сооружений. М., Трансжелдориздат, 1958.
93. *Смирнов А. Ф., Александров А. В., Шапошников Н. Н., Лащенко Б. Я.* Расчет сооружений с применением вычислительных машин. М., Стройиздат, 1964.
94. *Снитко Н. К.* Практические методы расчета статически неопределимых систем. М., Госстройиздат, 1960.
95. *Снитко Н. К.* Расчет рамных сооружений итерационными методами. М., Госстройиздат, 1962.
96. *Сосис П. М.* Розрахунок рам методом розгону вузлових деформацій. Доповіді АН УРСР, К., 1950, № 5.
97. *Сосис П. М.* Расчет рам способом перераспределения начальных значений неизвестных. К., Гостехиздат, изд. 2-е, дополн., 1956.
98. *Сосис П. М., Хакало Б. П.* Расчет неразрезных и перекрестных балок. К., Госстройиздат УССР, 1958.
99. *Сосис П. М.* Розв'язання п-члених рівнянь будівельної механіки методом перерозподілу початкових значень невідомих. «Прикладна механіка», вип. 4, т. V, 1959.
100. *Сосис П. М.* Опыт механизации инженерных расчетов на машиносчетной станции. «Строительная механика и расчет сооружений», 1960, № 3.
101. *Сосис П. М.* Метод составления программ инженерных расчетов. Сб. «Исследования по теории сооружений», вып. X, М., Госстройиздат, 1961.
102. *Сосис П. М.* Механизация расчетов сооружений по типовым программам. К., Госстройиздат, 1961.
103. *Сосис П. М., Васильев В. В., Токарева О. Н.* О расчете нерегулярных рам на машине ЭМСС-7. «Организация и механизация проектно-конструкторских и копировально-множительных работ». М., изд. Гипротис, 1961, № 4.
104. *Сосис П. М.* Краткое руководство по расчету нерегулярных рам на электро моделирующей машине ЭММС-7. К., Изд-во АС и А УССР, 1962.
105. *Сосис П. М., Дмитриев Л. Г.* Программирование и применение электронных вычислительных машин для расчета строительных конструкций. Труды VI сессии АС и А УССР, К., Изд-во АС и А УССР, 1962.
106. *Сосис П. М.* Современные вопросы программирования для вычислитель-

- ных машин применительно к задачам строительной механики в широком смысле. Сб. «ЭЦВМ в строительной механике». М.-Л., Стройиздат, 1966.
107. Сосис П. М. О решении систем линейных уравнений с определенными структурами матриц на электронных вычислительных машинах. «Журнал вычислительной математики и математической физики», т. 3, 1963, № 4.
108. Сосис П. М. О методах решения систем линейных уравнений с малозаполненными матрицами коэффициентов на электронных цифровых машинах. Сб. «Исследования по теории сооружений», вып. XII, М., Госстройиздат, 1963.
109. Сосис П. М., АЛГОЛ-60 и применение его в строительной механике. К., «Будівельник», 1965.
110. Сосис П. М. Принцип построения универсальных алгоритмов для решения задач строительной механики. Сб. «Применение электронных математических машин в строительной механике». К., «Наукова думка», 1968.
111. Сосис П. М., Кимельфельд И. Э. Расчет пространственных стержневых систем по универсальной программе метода деформаций. «Строительная механика и расчет сооружений», 1966, № 5.
112. Сосис П. М., Кимельфельд И. Э. Расчет стержневых систем на электронных вычислительных машинах. Сб. «Исследования по теории сооружений», вып. XV, М., Госстройиздат, 1967.
113. Стрелецкий Н. С. Металлические конструкции, М., Госстройиздат, 1961.
114. Строительная механика в СССР 1917—1957. М., Сб. статей под редакцией члена-корреспондента АН СССР И. М. Рабиновича. М., Госстройиздат, 1957.
115. Таиров В. Д. Сетчатые пространственные конструкции. К., «Будівельник», 1966.
116. Токарева О. Н. Электронное моделирование статически неопределимых систем. К., Изд-во АН УССР, 1965.
117. Токарева О. Н. Новая электронная специализированная математическая машина для расчета рамных систем. Организация и механизация проектно-конструкторских и копировально-множительных работ. Сб. 1(10), М., изд. Гипротис, 1963.
118. Трахтенброт Б. А. Алгоритмы и машинное решение задач. М., Физматгиз, 1960.
119. Уманский А. А. О решении трехчленных уравнений. Сб. «Рамы и фермы, пространственные и плоские», М., Госстройиздат, 1933.
120. Уманский А. А. Специальный курс строительной механики, ч. I, М.-Л., ОНТИ, 1935, ч. II, М., Стройиздат, 1940.
121. Фадеев Д. К., Фадеева В. Н. Вычислительные методы линейной алгебры. М., Физматгиз, 1960.
122. Феодосьев В. И. Об одном способе решения нелинейных задач устойчивости деформируемых сред, ПММ, т. 27, М., 1963, № 2.
123. Феодосьев В. И. Применение шагового метода к анализу устойчивости сжатого стержня. ПММ, т. 27, М., 1963, № 5.
124. Феодосьев В. И. Применение вариационного шагового метода к решению задач об устойчивости стержня за пределами упругости. ПММ, т. 27, М., 1963, № 6.
125. Филлин А. П. Расчет пространственных стержневых конструкций типа системы перекрестных связей и его применение к оболочкам при использовании электронных вычислительных машин. Исследования по строительной механике. Сб. трудов ЛИИЖТ, вып. 190, Л., 1962.
126. Филлин А. П. Проблемы использования современных вычислительных средств в строительной механике. Сб. «ЭЦВМ в строительной механике», М.-Л., Стройиздат, 1966.

127. Филин А. П. Матрицы в статике стержневых систем и некоторые элементы использования ЭЦВМ. М.-Л., Стройиздат, 1966.
128. Хел Д. М. Геодезические купола. «Современная архитектура», 1962, № 1.
129. Шагин П. П. Сокращенные методы расчета рам. М., Госстройиздат, 1951.
130. Шайкевич В. Д. Матричный метод расчета регулярных систем. Расчет пространственных конструкций, вып. IV, М., Госстройиздат, 1958.
131. Шура-Бура М. Р., Любимский Э. З. Транслятор АЛГОЛ-60. «Журнал вычислительной математики и математической физики», т. 4, 1964, № 1.
132. Эшби У. Росс. Введение в кибернетику. М., Изд-во иностранной литературы, 1959.
133. Эшби У. Росс. Конструкция мозга. Происхождение адаптивного поведения. М., Изд-во иностранной литературы, 1962.
134. Ющенко Е. Л. Некоторые вопросы теории алгоритмического языка и автоматизация программирования. К., Изд. института математики АН УССР, 1965.
135. Якобсон Л. С. Автоматический расчет на ЭВМ плоских стержневых систем. «Вычислительная и организационная техника в строительстве и проектировании», изд. Гипротис, М., 1965, вып. 12.
136. Современные методы расчета сложных статически неопределимых систем. Сб. статей иностранных авторов. Составление и общая редакция проф. А. П. Филина. М.-Л., Судпромгиз, 1961.
137. Argyris I. H. Matrix displacement analysis of two— and three — dimensional deformable media. «proc. 2-nd Conf. Dimens. and Strength Calculat. Budapest, 1961. Budapest, 1965.
138. Cross H. Analysis of continuous frames by distributing fixed end moments «Proceedings of the American Society of civil Eng. May. 1930, vol. 96, 1932, «Numerical Methods of Analysis in Engineering», Nev-York, 1949.
139. Cloucek C. Pozvod deformace jako podklad nove početni metody, I, Praha, 1953, SNTL; Rozvod deformace při posuvnich styčnicich, II, Praha, 1957, SNTL; Distribution of deformation, London, 1955.
140. Kani C. Die berechnung mehrstackigen Rahmen. Stuttgart, 1949.
141. Livesley R. K. Matrix method of structural analysis. Oxford, Pergamon Press, 1964.
142. Sauthwell R. V. Relaxation Methods in Theoretical Physics, London, 1946.
143. Расчет строительных конструкций с применением электронных машин. Сб. статей под редакцией проф. А. Ф. Смирнова. М., Стройиздат, 1967.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава I. Развитие алгоритмов	7
Алгоритмы перераспределения	8
Компактные схемы алгоритмов	11
Машинно-аналитические алгоритмы для электро- моделирующих машин	15
Алгоритмы для электронных вычислительных машин	20
Глава II. Алгоритмы перераспределения	28
Итерационное решение	28
Точное решение	41
Системы пятичленных уравнений	50
Смешанные решения	51
Расчет рам произвольного вида на машине ЭМСС-7 (7М) машинно-аналитическими мето- дами	62
Глава III. Алгоритмический язык инженерного програм- мирования. Компактные алгоритмы решения си- стем линейных уравнений	69
Общие сведения о работе клавишных вычисли- тельных машин	70
Символы алгоритмического языка для операто- ров клавишных автоматов	72
Арифметические и логические выражения	75
Операторы засылки и цикла	76
Строение программы, операторы перехода и проверки, составной оператор	82
Правила составления программы для клавиш- ных автоматов	84
Сопоставление инженерного программирования и матричного исчисления	86
Компактные алгоритмы решения систем линей- ных уравнений методом исключения	91
Компактный алгоритм для метода последова- тельных приближений	100

Глава IV.	Компактные алгоритмы строительной механики	105
	Построение универсальной программы метода сил	105
	Построение универсальной программы метода деформаций	118
	Программа определения частот и форм собственных колебаний произвольной системы	125
	Прием построения программ метода сеток	135
	Краткие сведения о других программах	140
	Запись алгоритмов для ЭЦВМ на языке нижнего программирования	142
Глава V.	Общие вопросы применения алгоритмического языка АЛГОЛ-60	146
	Краткие сведения об электронных цифровых вычислительных машинах	148
	Операторы процедур ввода-вывода на алгоритмическом языке	156
	Пример построения программы	157
	Правила составления программ на языке АЛГОЛ-60	164
	О формах публикации программы	170
	О перспективах автоматизации программирования	173
Глава VI.	Построение систем алгоритмов строительной механики	177
	Решение систем линейных уравнений ленточной и ленточной окаймленной структур	178
	О решении систем линейных уравнений циклическими методами	195
	Обобщенная программа расчета стержневых систем методом сил	207
	О способах задания исходной информации для расчета методом сил	214
	Процедуры выборки величин i_1 и i_2 (k_1 и k_2)	218
	Обобщенная программа расчета дискретных систем методом деформаций	224
	О способах задания исходной информации для расчета методом деформаций	231
	Функция-выборка реактивных усилий из заданного массива	234
Глава VII.	Характерные процедуры системы алгоритмов строительной механики	238
	Процедуры-функции для расчета элементов статически неопределимых систем различных видов	238
	Построение алгоритмов для расчета пространственных систем со структурной симметрией	252
	Процедура определения частот и форм собственных колебаний произвольной системы	260

Алгоритмы расчета систем с односторонними связями	263
Обобщенный алгоритм расчета стержневых систем на устойчивую прочность	277
Алгоритм расчета стержневых систем с учетом неупругих свойств железобетона	281
О шаговых алгоритмах расчета нелинейных систем	283
Процедуры построения основной системы метода деформаций и операторов стержней на основе информации о зрительном образе сооружений	286
Л и т е р а т у р а	302

*Зональный научно-исследовательский и проектный институт типового и экспериментального проектирования жилых и общественных зданий
Государственного комитета по гражданскому строительству и архитектуре
при Госстрое СССР*

ПЕТР МОИСЕЕВИЧ СОСИС

СТАТИЧЕСКИ НЕОПРЕДЕЛИМЫЕ СИСТЕМЫ

Редактор *И. М. Лыкова*
Обложка художника *А. М. Шоломий*
Художественный редактор *В. В. Красий*
Технический редактор *Л. А. Серафин*
Корректор *С. Н. Лысенко*

БФ 02466. Сдано в набор 1. III. 1968 г. Подписано к печати 24. VII. 1968 г. Бумага типографская № 2, $60 \times 84 \frac{1}{16} = 9,75$ бумажных, 19,5 физ., 18,13 усл. печ., 17,1 уч.-изд. л. Тираж 5000. Цена 1 руб. 42 коп. Зак. 73.

Издательство «Будівельник», Киев, ул. Владимирская, 24.

Киевская книжная типография № 6, Киев, Выборгская, 84.