

Т. М. Платова,
В. А. Скрипняк

Системы
автоматизированного
проектирования
конструкций
с учетом
динамической
прочности

7 6 ЯНВ 2001

LAURENȚIU ȚIPLEA

și dacă $s/\Lambda \neq t/\Lambda$, conform propoziției
ficabilă. Dacă însă $s/\Lambda = t/\Lambda$, atunci fie
, ..., $s_i, \dots, s_k$), $t = f(t_1, \dots, t_i, \dots, t_k)$. Cum
e $\langle s_i = t_i \rangle$, $i \in \{1, 2, \dots, k\}$, conform ipo-
sînt decidabile. Atunci ecuația $\langle s = t \rangle$

într-un algoritm de unificare generală.
folosește o listă LISTA în care sînt intro-

ritmul AUG

unificarea este terminată, stop; altfel
în LISTA, fie acesta π , și se continuă

și la P4; altfel merg la P9.

Томский ордена Трудового Красного Знамени и ордена
Октябрьской Революции государственный университет им.В.В.Куйбышева

Т.М.Платова, В.А.Скрипник

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ КОНСТРУКЦИЙ
С УЧЕТОМ ДИНАМИЧЕСКОЙ ПРОЧНОСТИ

Учебное пособие



Томск - 1988

1-6278283

УДК 681.3 + 539.3

Платова Т.М., Скрипняк В.А. Системы автоматизированного проектирования конструкций с учетом динамической прочности :

Учебное пособие. - Томск : Изд-во Том. ун-та, 1988 . - 130 с.
25 к., 300 экз. 1003040000

Рассмотрены вопросы организации, структуры, программного обеспечения САПР, организация специального программного обеспечения. Изложена методика разработки модулей пакетов прикладных программ для расчета элементов конструкций на прочность при высокоскоростном деформировании. Приведен пример разработки программного комплекса для расчета высокоскоростного судорення пластин.

Для студентов физико-технических специальностей университетов и других вузов, обучающихся по ЦИПС.

II 1003040000 53 -- 88
177/012/-88

© Издательство Томского университета , 1988

ВВЕДЕНИЕ

В последнее десятилетие теория и практика автоматизированного проектирования развивалась высокими темпами. Созданы системы автоматизированного проектирования, успешно эксплуатирующиеся в ряде отраслей промышленности. В связи с этим возросла потребность народного хозяйства в квалифицированных специалистах-пользователях САПР.

Основой САПР является программное обеспечение, на создание которого уходит, по некоторым оценкам, до 80 % общих затрат. Программное обеспечение определяет эффективность САПР и тенденция увеличения затрат на его разработку сохраняется.

Центральной проблемой в разработке и эксплуатации САПР является построение адекватной модели проектируемого объекта. Проведение огромного числа точных расчетов на ЭВМ с использованием неадекватной модели приводит к результатам с сомнительной практической ценностью и дискредитирует САПР в целом.

Проблемы адекватного описания объекта проектирования и создания специального программного обеспечения САПР усложняются тем, что между потребителем результатов расчетов - пользователем и ЭВМ стоят специалисты, закрепленные за конкретными пакетами прикладных программ и обеспечивающие их функционирование. Эти специалисты, обладающие знаниями в программировании, часто недостаточно понимают смысл решаемых задач, терминологию прикладной области. Поэтому часть важной информации о задаче теряется и в математическую модель закладываются либо завышенные, либо заниженные требования к проектируемому объекту. И то, и другое с точки зрения эффективности САПР нежелательно.

Одним из путей разрешения этой ситуации является подготовка специалистов-прикладников, обладающих знаниями о САПР и о работе ее подсистем. Такие специалисты могли бы не только квалифицированно эксплуатировать проектирующие подсистемы, но и принимать участие в разработке новых.

При современных темпах смены поколений ЭВМ и общесистемного программного обеспечения у специалиста, занимающегося эксплуатацией САПР, возникает проблема выбора направления совершенствования и углубления знаний. При этом необходимо помнить, что основу проектных процедур составляют приемы и методы специальной предметной области знаний.

Особенно остро эти проблемы стоят при разработке проектируемых подсистем, выполняющих расчет конструкций и их элементов на прочность при динамическом нагружении.

Известно, что до настоящего времени не разработана универсальная модель, способная адекватно описывать реакцию материалов в условиях высокоскоростного деформирования. Поэтому от разработчика таких подсистем САПР требуется понимание ограничений применяющихся моделей, методов расчета и способность оценивать достоверность получаемых результатов.

В данном учебном пособии рассмотрен ряд вопросов, которые возникают при создании или эксплуатации подсистем САПР, ориентированных на решение задач прочности конструкций при динамическом нагружении.

Г Л А В А I

СТРУКТУРА И ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

I.1. Комплекс средств автоматизации проектирования

Ускорение научно-технического прогресса обострило противоречие между сроками разработки технических проектов и их качеством. Затянувшаяся разработка проекта неизбежно приводит к моральному старению технических решений, что особенно ощутимо в машиностроении. Сокращение сроков проектирования снижает качество проектов вследствие недостаточно глубокой проработки, что, в свою очередь, вызывает необходимость длительной "доводки" изделия.

Надежды на разрешение этой проблемы связываются с разработкой систем автоматизации процессов конструирования и технологической подготовки производства.

В настоящее время подавляющее большинство проектных и конструкторских организаций выполняет отдельные этапы проектирования с помощью пакетов прикладных программ (ППП). В наиболее крупных проектных организациях разрабатываются и используются системы автоматизированного проектирования (САПР).

Под системой автоматизированного проектирования понимается комплекс средств автоматизации проектирования, взаимосвязанных с необходимыми подразделениями проектной организации или коллективом специалистов, которые выполняют автоматизированное проектирование.

Комплекс средств автоматизации проектирования включает несколько видов обеспечения: техническое, математическое, программное, информационное, лингвистическое, методическое, организационное. Состав САПР показан на рис.1 [1 - 3].

Техническое обеспечение САПР представляет собой совокупность взаимосвязанных и взаимодействующих технических средств,

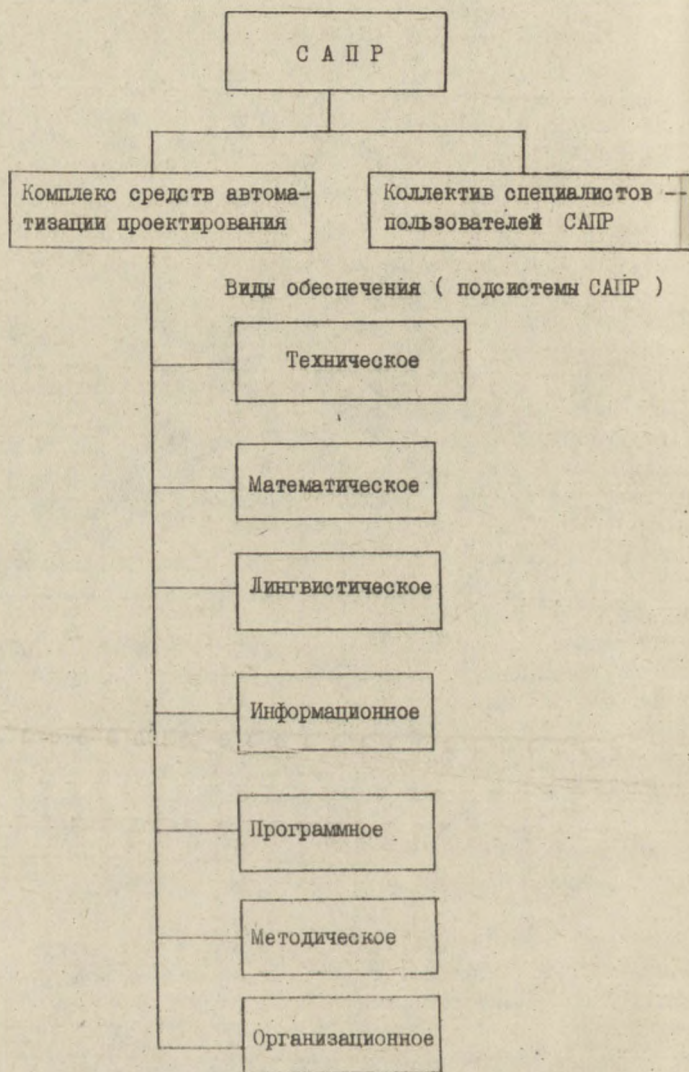


Рис.1. Состав систем автоматизированного проектирования

предназначенных для выполнения автоматизированного проектирования. Состав технических средств САПР показан на рис.2 [4]. Эти средства по своему функциональному назначению делятся на группы :

- а) программной обработки данных ;
- б) подготовки и ввода данных ;
- в) отображения и документирования ;
- г) архива проектных решений и данных ;
- д) передачи данных.

Математическое обеспечение САПР объединяет в себе математические методы, математические модели объектов проектирования, алгоритмы проектирования, необходимые для выполнения автоматизированного проектирования. В математическое обеспечение входят инвариантные элементы (методы решения алгебраических и дифференциальных уравнений, методы постановки и решения экстремальных задач, принципы и методы построения функциональных моделей), а также элементы специального математического обеспечения, в которых находят отражение особенности математических моделей и постановки задач проектирования конкретных изделий.

Математическое обеспечение реализуется в специальном программном обеспечении.

Программное обеспечение САПР (ПО) объединяет собственно программы для обработки данных и программную документацию, необходимую для эксплуатации программ при выполнении автоматизированного проектирования [1-3]. Программное обеспечение подразделяется на общесистемное, базовое и прикладное (специальное), как показано на рис.3.

Общесистемное ПО предназначено для организации функционирования технических средств, управления вычислительным процессом в САПР, подготовки программ из состава пакетов прикладных программ. Эти функции выполняются программами, входящими в состав операционных систем .

В базовое ПО входят программы, обеспечивающие правильное функционирование проектирующих подсистем САПР, в том числе пакетов прикладных программ [3].

В прикладное ПО входят программы, ориентированные на выполнение проектных процедур и решение специальных задач проектирования. Обычно эти программы имеют форму пакетов прикладных программ.

КТС САПР	
Э В М	Центральные процессоры, специализированные процессоры, оперативная память, процессоры ввода - вывода, устройства сопряжения интерфейсов
Внешние запоминающие устройства	Накопители на магнитных дисках, накопители на гибких магнитных дисках, накопители на магнитной ленте
Устройства ввода-вывода информации	Устройства ввода - вывода с перфокарт, устройства ввода - вывода с перфоленты, печатающие устройства, устройства вывода на микрофими
Устройства подготовки данных	Устройства подготовки данных на перфокартах, устройства подготовки данных на перфоленте, устройства подготовки данных на магнитной ленте
Устройства оперативной связи с ЭВМ	Алфавитно-цифровые дисплеи, графические дисплеи, устройства управления курсором, устройства речевого ввода-вывода
Устройства машинной графики	Устройства кодирования графической информации, графопостроители, графические дисплеи, растровые печатающие устройства
Технические средства теледоступа и сетей ЭВМ	Процессоры связи, сетевые контролеры, каналы связи, аппаратура передачи данных, мультиплексоры передачи данных
Устройства связи с технологическим оборудованием	

Рис. 2. Комплекс технических средств систем автоматизированного проектирования

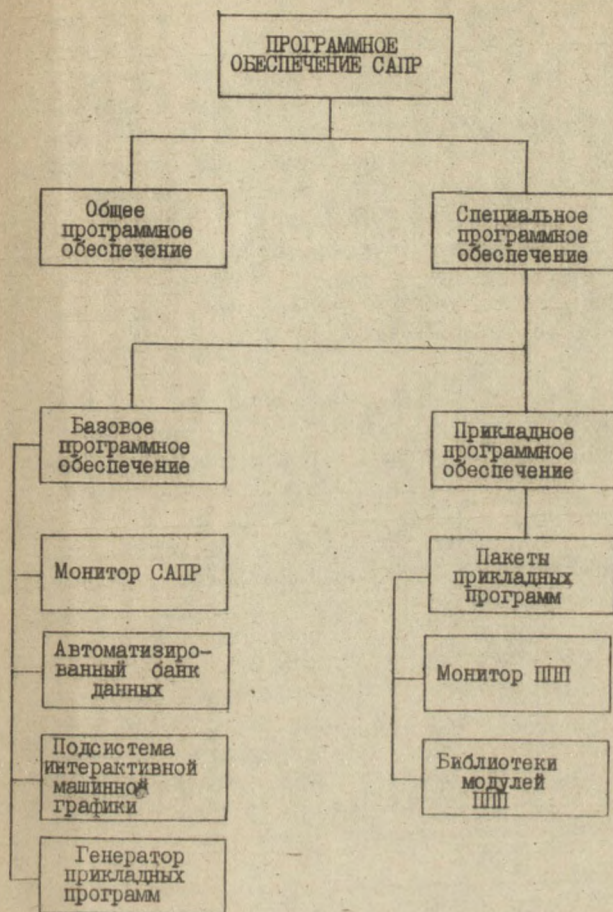


Рис.3. Состав программного обеспечения САПР

Информационное обеспечение САПР объединяет данные, необходимые для выполнения автоматизированного проектирования. Основную часть информационного обеспечения составляет база данных (БД) (рис.4). Она представляет собой информационные массивы, имеющие определенную структуру и записанные на запоминающие устройства ЭМ. В процессе функционирования БД пополняется и корректируется с помощью программ обслуживания БД, которые объединяются в систему управления базой данных (СУБД).

База данных совместно с системой управления базой данных образуют автоматизированный банк данных (АБД) САПР.

Лингвистическое обеспечение САПР представляет совокупность языков, применяемых при общении человека с ЭМ или САПР (рис.5). В этот вид обеспечения входят :

- а) языки программирования (алгоритмические языки – Алгол, Фортран, Кобол, Паскаль, ПЛ-I и т.д.);
- б) языки проектирования (языки описания проектных процедур, языки описания проектных решений, языки описания объектов проектирования).

Лингвистическое обеспечение реализуется в трансляторах, входящих в состав операционных систем и мониторов САПР.

Методическое обеспечение САПР составляет совокупность документов, устанавливающих правила отбора и эксплуатации средств автоматизированного проектирования.

Организационное обеспечение САПР объединяет документы, устанавливающие организационную структуру проектной организации или ее подразделений, где осуществляется проектирование с использованием САПР. Документы, составляющие организационное обеспечение САПР, также определяют форму и состав выпускаемой проектной документации.

1.2. Классификация САПР

Для стандартизации, типизации и унификации типов САПР разработана классификация САПР, которая регламентируется ГОСТ 23501.8-80 "Системы автоматизированного проектирования. Классификация и обозначение". Согласно этой классификации к наиболее важным признакам САПР относятся : а) тип объекта проектирования ; б) комплексность автоматизации проектирования ; в) количество уровней в структуре технического обеспечения [1] .

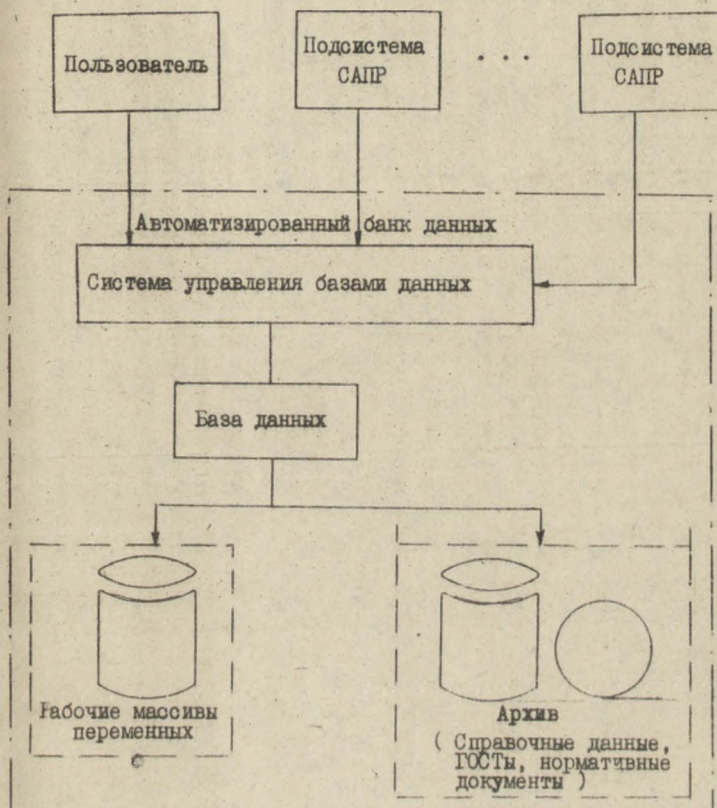


Рис.4. Структура информационного обеспечения САИР

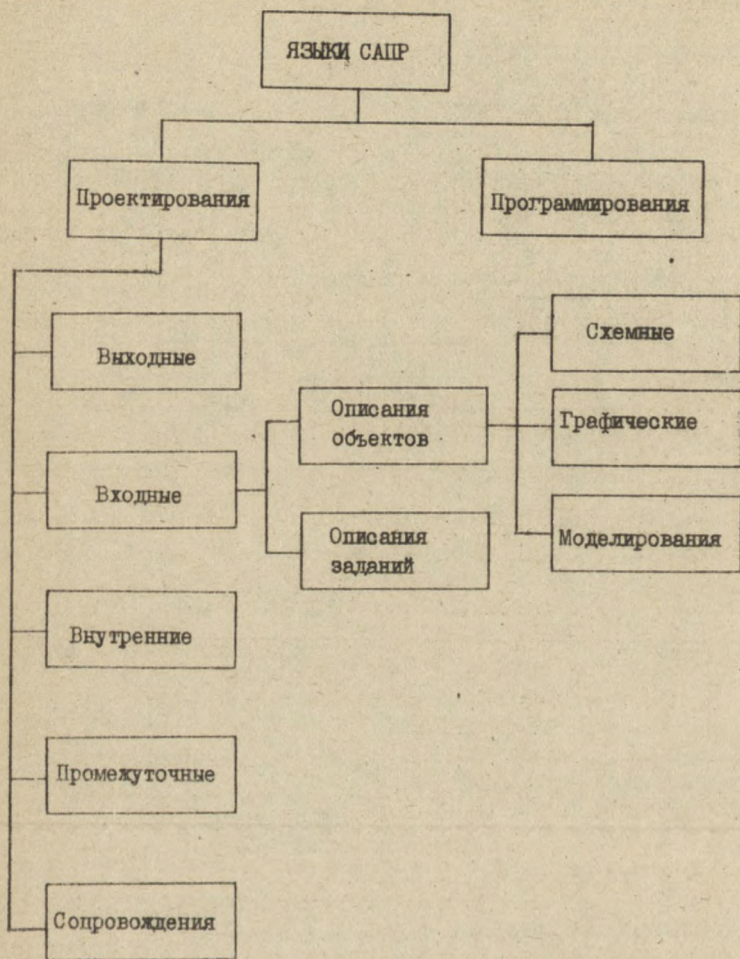


Рис.5. Классификация языков САПР

По типу объекта проектирования выделяют :

- а) САПР изделий машиностроения и приборостроения ;
- б) САПР технологических процессов (САПР ТП) в машиностроении и приборостроении ;
- в) САПР объектов строительства ;
- г) САПР организационных систем.

В зависимости от комплексности автоматизации проектирования выделяют :

- а) одноэтапные САПР, которые выполняют один этап проектирования из всех, установленных для объекта ;
- б) многоэтапные САПР, осуществляющие несколько этапов проектирования ;
- г) комплексные САПР, выполняющие все этапы проектирования, установленные для объекта.

Под этапом проектирования понимается условно выделенная часть процесса проектирования, в результате которой находится проектное решение или их совокупность [1] .

В зависимости от состава используемых в САПР технических средств выделяют САПРы различных уровней : а) одноуровневые ; б) двухуровневые ; в) трехуровневые (рис.6).

Одноуровневая САПР строится на базе ЭЕМ среднего или высокого класса со штатным набором периферийного оборудования, который может быть дополнен средствами обработки графической информации. Это наиболее распространенные САПРы. Как правило, в них используются единая операционная либо мониторная система, АБД, ППП, ориентированные на используемую ЭЕМ.

Двухуровневая САПР строится на базе ЭЕМ высокого или среднего класса и одного или нескольких автоматизированных рабочих мест (АРМ), включающих мини-ЭЕМ.

Трехуровневая САПР строится на базе ЭЕМ высокого класса, связанной с несколькими АРМ и периферийным программно управляемым оборудованием (станками с числовым программным управлением, роботизированными линиями и т.д.).

1.3. Структура САПР

Системы автоматизированного проектирования состоят из взаимосвязанных и взаимодействующих подсистем, обладающих всеми свойствами самостоятельных систем.

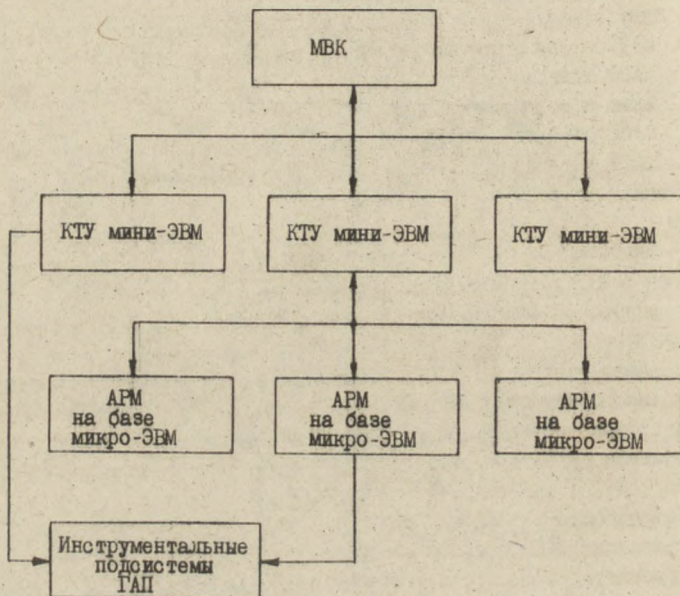


Рис.6. Состав многоуровневой САИР: МБК – комплект технических средств многопроцессорного вычислительного комплекса; АРМ – автоматизированное рабочее место программиста; ГАП – гибкие автоматизированные производства

По назначению подсистемы САПР подразделяют на проектирующие и обслуживающие [I - 4]. К проектирующим относятся подсистемы, выполняющие проектные операции и процедуры. Под проектной операцией понимается действие или формализованная совокупность действий, составляющих часть проектной процедуры, алгоритм которых остается неизменным [I].

Проектная процедура - это формализованная совокупность действий, выполнение которых оканчивается проектным решением [I - 3].

Примером проектирующих подсистем в машиностроительных САПР могут служить подсистема проектирования узлов и деталей, подсистема технологического проектирования.

Подсистемы, предназначенные для обеспечения функционирования проектирующих подсистем и расширения их возможностей, относятся к обслуживающим.

Примером обслуживающих подсистем может служить СУБД, а также подсистема графического отображения объектов проектирования.

Проектирующие подсистемы включают инвариантные (иначе-объектно-независимые) и объектно-ориентированные компоненты. Инвариантные компоненты подсистем выполняют унифицированные проектные процедуры и операции с неизменным для различных объектов алгоритмом. Объектно-ориентированные компоненты подсистем осуществляют одну или несколько проектных процедур или операций, зависящих от конкретного объекта проектирования. Одновременная разработка систем автоматизированного проектирования в одной отрасли народного хозяйства вызывается именно существенными различиями в проектируемых изделиях и необходимо учитывать специфику объекта проектирования в проектирующих подсистемах САПР.

I.4. Принципы построения САПР

Разработка САПР и ее подсистем ведется на основе следующих принципов [I-4]:

- 1) включения ;
- 2) системного единства ;
- 3) развития ;
- 4) комплексности ;
- 5) информационного единства ;
- 6) инвариантности.

Принцип включения состоит в том, что разработка САПР ведется в интересах не только самой организации, но и отрасли. Поэтому разработка САПР должна быть упорядочена, а создаваемые инвариантные подсистемы должны отвечать принятым в отрасли стандартам. Унификация программного, лингвистического, технического и других видов обеспечения открывает широкую возможность внедрения САПР и ее подсистем на различных предприятиях.

Принцип системного единства предусматривает на всех стадиях создания и развития САПР целостность системы, которая обеспечивается связями между подсистемами, а также функционированием подсистемы управления САПР.

Согласно принципу развития САПР должна создаваться и функционировать с учетом пополнения, совершенствования и обновления подсистем и компонентов.

Принцип комплексности означает, что САПР должна обеспечивать связность проектирования отдельных элементов и всего объекта в целом на всех этапах проектирования. Наибольший эффект от САПР достигается при комплексной автоматизации проектирования на всех уровнях.

Принцип информационного единства и совместимости компонентов САПР заключается в том, что символы, коды, условные обозначения информационных и технических характеристик объектов, языки проектирования, способы представления информации должны обеспечивать совместную работу различных подсистем САПР с одной и той же базой данных и единым входным языком.

Принцип инвариантности предусматривает создание подсистем и компонентов САПР таким образом, чтобы они могли функционировать независимо друг от друга. Разрабатываемые подсистемы должны быть по возможности универсальными по отношению к проектируемым объектам и отраслевой специфике.

Г Л А В А 2

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ САПР

2.1. Структура программного обеспечения САПР

Математическое, лингвистическое, методическое и информационное обеспечения САПР реализуются в программном обеспечении. Программное обеспечение САПР представляет собой совокупность программ общего и специального назначения (рис.7). В зависимости от своего функционального назначения программы относятся к общему либо специальному программному обеспечению [4].

В состав ПО САПР обычно входят :

1. Автоматизированный банк данных .
2. Операционная система общего назначения (ОС).
3. Монитор САПР.
4. Пакеты прикладных программ .
5. Подосистема машинной графики .

Варианты организации ПО САПР разнообразны и определяются предметной областью, где функционирует САПР, уровнем и степенью комплексности САПР.

2.2. Общее программное обеспечение САПР

Общее ПО САПР составляют операционные системы, представляющие комплексы программ, предназначенных для организации ввода-вывода, управления данными, распределения ресурсов ЭВМ между программами, подготовки и отладки программ. Все компоненты ПО работают под управлением ОС, которая обеспечивает наиболее рациональное распределение и использование всех ресурсов ЭВМ.

Различают ОС общего и специального назначения. К ОС специального назначения относят системы, предназначенные для поддержки однородных вычислительных структур и сетей, организации и работы с ИД, а также ориентированные на решение задач в реальном времени.



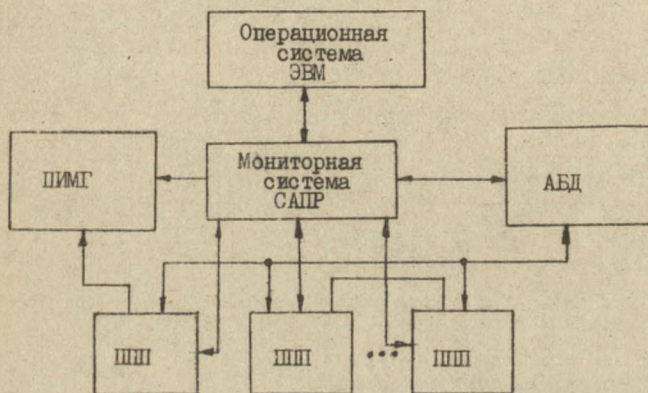


Рис.7. Структура программного обеспечения САПР:

ПИМГ - подсистема интерактивной машинной графики;

АБД - автоматизированный банк данных ;

ППП - проектирующие подсистемы (пакеты прикладных программ)

Операционные системы могут обеспечивать однопрограммный либо мультипрограммный режим обработки задач. При этом взаимодействие пользователя с ОС может идти в режиме пакетной обработки и в режиме диалога.

В САПР применяются операционные системы, которые условно можно объединить в несколько групп [4].

1. Группа ОС общего назначения, обеспечивающих однопрограммный режим обработки задач и диалоговый способ взаимодействия с пользователем. Эти ОС предоставляют пользователю сведения о ходе выполнения задач, обеспечивают работу с библиотеками, а также некоторые другие возможности. Такие ОС обычно называют мониторами и используют только в персональных микро- и мини-ЭВМ.

2. Группа ОС общего назначения, выполняющих пакетную обработку задач в режиме мультипрограммирования. Их применяют в вычислительных системах большой и средней производительности. В условиях потока отладочных задач, когда основное время уходит на многочисленные обмены, ОС этого типа не способствуют эффективному использованию времени центрального процессора. Кроме того, у пользователей отсутствует возможность вмешиваться в процесс решения задачи.

3. Группа ОС разделения времени. В ОС подобного типа пользователи имеют возможность вмешиваться в процесс выполнения программ. Причем ОС реализуют такой режим, при котором каждая задача разных пользователей получает необходимую ей область оперативной памяти ЭВМ, а для выполнения задачи выделяются процессор и другие вычислительные ресурсы в течение определенного, достаточно малого интервала времени — кванта. После окончания отводимого кванта времени процессор переключается на выполнение задачи другого пользователя. Если ресурсов оперативной памяти ЭВМ не хватает для содержания всех программ пользователей, предыдущая задача удаляется из оперативной памяти и размещается на накопителе на магнитных дисках. Когда очередь снова дойдет до выполнения программы, она будет считана с накопителя в оперативную память, а выполнение начнется с момента прерывания. Таким образом, у пользователей только создается иллюзия доступности всех вычислительных ресурсов. Фактическое время исполнения программы пользователя значительно меньше, чем в режиме мультипрограммирования, и зависит от количества одновременно работающих с ЭВМ пользователей.

Операционные системы этой группы наиболее часто применяются в САПР.

Помимо перечисленных выше групп ОС в многоуровневых САПР применяются ОС реального времени и ОС вычислительных сетей. Эти операционные системы используются большей частью в гибких автоматизированных производствах и автоматизированных системах управления.

2.3. Структура операционных систем

Операционные системы состоят из программ, которые по своему функциональному назначению можно отнести к двум группам [1 - 4]:

1. Обработывающие программы, составляющие подсистему подготовки программ или внешнее программное обеспечение.

2. Управляющие программы, составляющие группу исполнения программ или внутреннее программное обеспечение.

Структура ОС показана на рис.8.

К обрабатывающим относятся следующие программы :

- а) трансляторы ;
- б) обслуживающие или сервисные программы ;
- в) библиотеки стандартных программ .

В функции трансляторов входит перевод программы с алгоритмического языка на машинный.

Сервисные программы служат для обработки данных - редактирования, связывания, сортировки, объединения, перезаписи и т.д.

Программы, входящие в состав библиотеки стандартных программ, применяются для выполнения наиболее часто встречающихся при работе ОС задач.

Группа управляющих программ выполняет функции управления обрабатывающими программами. В эту группу входят:

- а) программы управления данными ;
- б) программы управления заданиями ;
- в) программы управления задачами (супервизор) .

В функции программ управления данными входят организация обращений пользователя к данным, а также эффективное планирование и управление обменом данными между оперативной памятью ЭВМ и накопителями на магнитных дисках.

Функции программ управления заданиями составляет обработка



Рис.8. Структура общего программного обеспечения САПР

заданий – считывание и интерпретация команд языка управления заданиями, организация очереди заданий, распределение устройств ввода-вывода, передача управления супервизору.

Программы управления задачами часто называют супервизорами. В их функции входит обработка прерываний, управление передачей управления от одной программы другой, распределение ресурсов времени и оперативной памяти между программами.

2.4. Последовательность обработки задания на ЭВМ

Решение задачи на ЭВМ начинается с записи ее алгоритма на языке программирования. Законченную совокупность операторов или команд, соответствующую алгоритму решения задачи, принято называть программой.

Текст программы на алгоритмическом языке называют исходной программой, или исходным модулем. Алгоритмические языки приближены к человеческому, поэтому исходный модуль понятен программисту, но не понятен ЭВМ. ЭВМ способна воспринимать только язык машинных команд. Операционная система, а вернее ее обслуживающие программы, предоставляет программисту услуги по преобразованию исходного модуля в программу, понятную ЭВМ. При этом исходная программа проходит несколько этапов обработки и претерпевает изменения. На рис. 9 представлены этапы обработки программного комплекса, состоящего из нескольких модулей, которые написаны на различных языках программирования.

На первом этапе – трансляции – тексты исходных модулей преобразуются в тексты на языке машинных команд конкретной ЭВМ [4]. Для этого ОС загружает в оперативную память программу транслятора с соответствующего языка и передает ей управление центральному процессором. Входной информацией для работы транслятора служит исходный модуль, а результатом работы является объектный модуль, который представляет собой текст машинных команд и справочники перекрестных и внешних ссылок.

Программные комплексы, как правило, состоят из ряда программных модулей. Модули могут быть записаны на разных алгоритмических языках. ОС имеет комплект трансляторов, которые преобразуют каждый из исходных модулей в объектный. Следующим этапом обработки является определение связей между модулями и объединение всех объектных модулей в единую программу, которая носит назва-

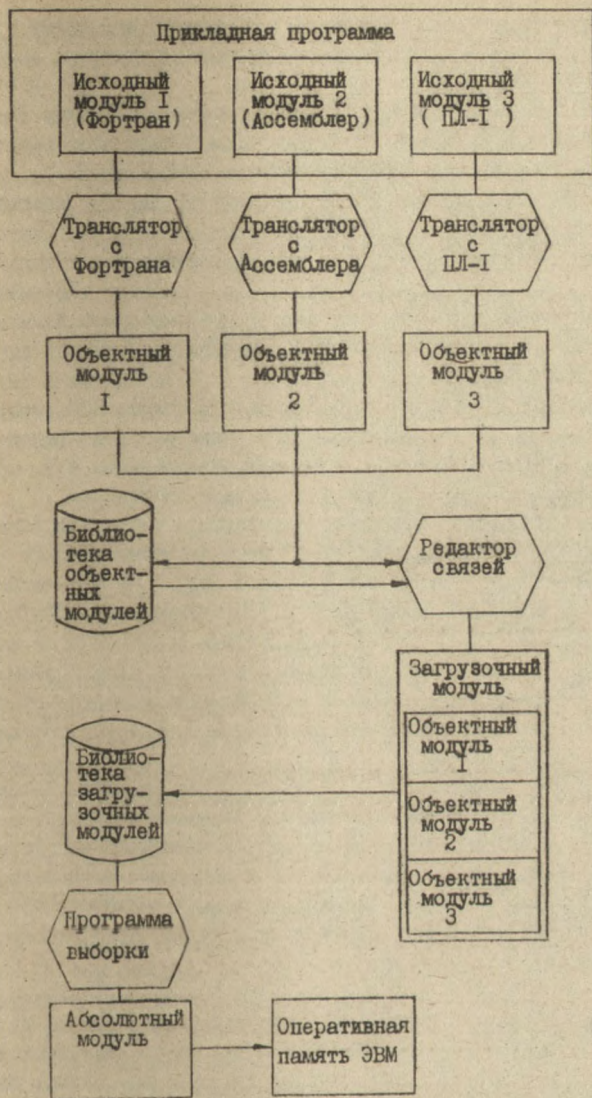


Рис.9. Обработка программы в операционной системе ЕС ЭВМ

ние загрузочного модуля. В ОС такое объединение выполняется с помощью специальной обслуживающей программы - редактора связей и называется редактированием.

Исходной информацией для редактора служат объектные модули, которые могут быть переданы редактору связей либо непосредственно после трансляции, либо из библиотеки объектных модулей, куда они могут быть записаны ранее. Загрузочный модуль в обязательном порядке записывается в библиотеку загрузочных модулей. Как объектные, так и загрузочные модули могут использоваться в дальнейшей работе пользователя неоднократно, что позволяет экономить время центрального процессора на работу транслятора и редактора. Это обстоятельство особенно важно для уменьшения времени на обработку программ в САПР.

Загрузочный модуль еще не может быть выполнен ЭВМ, поскольку записан в относительных адресах. На этапе редактирования осуществляется первичное разрешение межмодульных ссылок, т.е. согласовываются условные адреса входов и выходов из модулей.

На следующем этапе обработки программы - этапе загрузки - специальная обслуживающая программа загрузчик осуществляет настройку относительных адресов загрузочного модуля на абсолютные физические адреса оперативной памяти ЭВМ. Такая настройка носит название вторичного разрешения межмодульных ссылок. После настройки полученная программа, или иначе абсолютный модуль, размещается в оперативной памяти ЭВМ. На этом заканчивается подготовка программы к исполнению на ЭВМ, и управление передается управляющим программам ОС.

Порядок обработки программ может изменяться, если исходные модули были заранее оттранслированы и записаны в библиотеку объектных модулей. В этом случае этап трансляции не выполняется, а сразу осуществляется редактирование и загрузка. Если исходные модули были ранее не только оттранслированы, но и отредактированы и записаны в библиотеку загрузочных модулей, обработка программ будет сводиться к загрузке.

Порядок обработки программ задается ОС пользователем с помощью языка управления заданиями. Следуя терминологии ОС ЕС ЭВМ, заданием называется последовательность вызова обрабатываемых программ для выполнения одной функции по обработке данных на ЭВМ. Каждая обрабатываемая программа выполняет шаг задания. Так, например, обработка с помощью прикладной программы, написанной на языке

Фортран, входных данных является заданием. Это задание состоит из трех шагов – трансляции с Фортрана, редактирования связей и выполнения .

При работе ОС в мультипрограммном режиме обработки заданий пользователь имеет возможность контролировать ход обработки программ по системным сообщениям ОС. В них ОС сообщает, что такой-то шаг задания завершился без ошибок или завершился аварийно.

2.5. Организация ОС ЕС ЭВМ

В настоящее время разработано и используется большое количество ОС. Практически для каждой серии ЭВМ созданы ОС нескольких разновидностей – общего назначения, реального времени, разделения времени, пакетной обработки [4] . Наибольшее распространение получили :

- а) ОС многопроцессорных вычислительных комплексов "Эльбрус" (ОС МВК) ;
- б) ОС ЭВМ единой серии (ОС ЕС ЭВМ) ;
- в) ОС серии малых ЭВМ (ЛОС, ДОС, СРВ, СРВ-М, ДМОС РВ, ОС РВ, ПЛОС и др.) ;
- г) ОС микро-ЭВМ (СПО СМ 1800, ДОС 1800, ОС 1800, МОС РВ СМ 1800, ФОДОС, UNIX и др.) .

Эти ОС используются в САПР и под их управлением функционирует подавляющее большинство ППП САПР. Перечисленные выше ОС являются универсальными и рассчитаны на решение разнообразных задач. Вследствие этого существующие ОС общего назначения не являются оптимальными для применения в конкретной САПР, где круг задач ограничен. Применение их в САПР объясняется тем, что разработка проблемно-ориентированной ОС требует больших затрат времени и средств.

Наиболее развитыми ОС в настоящее время являются версии ОС ЭВМ единой серии (ОС ЕС ЭВМ). В их состав входят практически все характерные компоненты. В зависимости от используемого варианта управляющей программы в ОС ЕС возможны несколько вариантов обработки задач :

- а) однопрограммный режим (РСР) ;
- б) мультипрограммный режим с фиксированным числом задач (МЕТ) ;
- в) мультипрограммный режим с переменным числом задач (МУТ) ;

г) мультипрограммный режим с переменным числом задач и использованием виртуальной памяти (SVS) [4-6].

В режиме РСР ОС обрабатывает только одну программу. В настоящее время ОС этого типа встречается редко.

В режиме МГТ оперативная память ЭВМ до начала обработки программ разбивается на фиксированное число разделов. Каждая из решаемых задач выполняется в отдельном разделе.

В режиме МУТ возможна одновременная обработка до 15 задач. Оперативная память ЭВМ выделяется задачам автоматически из одного общего раздела, который называется динамической областью. После окончания выполнения задачи занимаемая ею область оперативной памяти перераспределяется.

Общая структура ОС ЕС ЭВМ показана на рис.10 [4 - 6]. Обработываемые программы ОС ЕС включают: программы-трансляторы с Ассемблера, Алгола, Фортрана, Кобола, ПЛ-1, РПГ, редактор связей, загрузчик, программы сортировки-объединения и программы обслуживания (утилиты).

Редактор связей ОС ЕС выполняет несколько функций:

а) объединяет несколько объектных модулей в единый загрузочный модуль;

б) выполняет первичное разрешение всех межмодульных ссылок;

в) формирует загрузочный модуль данной структуры;

г) осуществляет более плотную упаковку текстовой и справочной информации в формате загрузочного модуля [4-6].

В ОС ЕС загрузочные модули могут быть простой, оверлейной и динамической структуры. Загрузочный модуль простой структуры загружается в оперативную память целиком и выполняется быстрее по сравнению с модулями другой структуры. Загрузочный модуль оверлейной структуры не может быть загружен в оперативную память целиком из-за нехватки области памяти. Поэтому редактор связей разбивает загрузочный модуль на сегменты, которые автоматически загружаются в оперативную память в соответствии с алгоритмом программы. Время решения задачи увеличивается за счет многочисленных перемещений сегментов между оперативной памятью ЭВМ и внешними запоминающими устройствами на магнитных дисках.

Загрузочные модули динамической структуры применяются тогда, когда программный комплекс состоит из большого числа достаточно автономных частей. Редактор связей после формирования загрузоч-

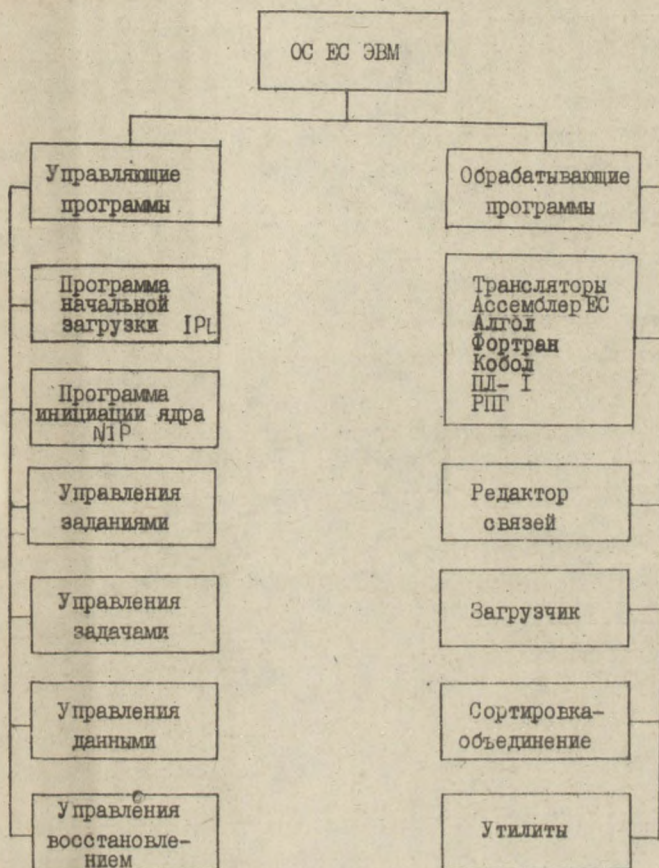


Рис. 10. Структура ОС ЕС

ного модуля помещает его в библиотеку загрузочных модулей. Оттуда по мере необходимости с помощью системных макрокоманд `LINK`, `LOAD`, `XCTL`, `ATTACH` осуществляется загрузка в оперативную память ЭВМ.

Программа сортировки-объединения позволяет сортировать неупорядоченные наборы данных, а также объединять несколько упорядоченных наборов данных в единый.

Программы обслуживания (утилиты) в ОС ЕС служат для разметки томов прямого доступа на магнитных дисках, копирования содержимого магнитных лент на магнитные диски и с магнитного диска на ленту, а также для создания и обслуживания наборов данных.

Управляющие программы ОС ЕС объединяют 6 групп программ. Программы начальной загрузки `IPL` и программа инициации ядра ОС `NIP` служат для первоначального размещения ОС в оперативной памяти ЭВМ после ее включения. Программы, составляющие ядро ОС, т.е. ту часть ОС, которая должна находиться в оперативной памяти ЭВМ, записаны на магнитном диске (системном томе), подключенном на внешнем запоминающем устройстве. Ввод программ `IPL` и `NIP` в оперативную память ЭВМ осуществляется оператором ЭВМ.

Программы управления восстановлением служат для обработки прерываний, вызванных сбоями и отказами технических устройств ЭВМ. Все сбои регистрируются, оператору передаются сообщения о зарегистрированных ошибках и, если они не связаны с серьезными неисправностями устройств, ОС продолжает нормальную работу. При серьезных сбоях система переходит в состояние ожидания.

Программа инициатора-терминатора `INIT` подготавливает каждую задачу к выполнению. Эта программа закреплена за каким-то классом заданий и постоянно находится в оперативной памяти ЭВМ в готовности обработать очередное задание. За входной очередью обычно закреплено несколько инициаторов. Их запуск осуществляет оператор ЭВМ.

Инициатор последовательно выполняет подготовку задачи к решению: запрашивает область оперативной памяти для размещения транслятора, редактора, программы пользователя; отыскивает необходимые наборы данных; передает управление загруженной в оперативную память программе. После выполнения задания `INIT` удаляет информацию из входной очереди и помещает в выходную

очередь все результаты и системные сообщения, полученные при обработке и решении задачи.

Кроме инициатора -терминатора в состав программ управления заданиями входят : программа системного ввода (*RDR*), программа системного вывода (*WTR*), главный планировщик.

С помощью программы системного ввода из входной очереди выбираются задания, состоящие из текстов программ и операторов языка управления заданиями (*JCL*), и направляются на обработку обслуживающими программами ОС. Характеристики и последовательность обработки программ пользователь сообщает ОС с помощью операторов *JCL*. Каждое из заданий поступает в одну из системных входных очередей в соответствии со своим классом. В ОС ЕС допускается 15 классов заданий, различающихся по виду и количеству потребляемых ресурсов [4-6]. Местоположение задания во входной очереди, помимо класса задания, определяется его приоритетом. Приоритет заданий в ОС ЕС определяется цифрой от 0 до 13. Чем выше приоритет (цифра), тем быстрее будет обработано задание.

Программа системного вывода (*WTR*) осуществляет передачу всей итоговой информации на устройство системного вывода. Системных выходных очередей может быть несколько. Класс выходной очереди обозначается буквой и задается пользователем в операторе *DD*. Наличие разных классов выходных очередей позволяет пользователю выводить информацию, полученную при решении одной задачи, на разные выходные устройства (АЦПУ, перфоратор, дисплей и т.д.). В ОС ЕС существует возможность прямого системного вывода, т.е. получение информации во время выполнения задания без предварительной записи на устройство системного вывода. Этот режим реализует программа *DSO*.

Программа главного планировщика выполняет передачу директив от оператора ЭВМ супервизору ОС, с помощью которых пользователь может влиять на организацию вычислительного процесса. Например, прекратить счет заиклившей программы.

Программы управления задачами в ОС ЕС представляют собой набор программ супервизоров, служащих для динамического распределения ресурсов между задачами в процессе их выполнения, а также обработки прерываний. Причиной прерывания может быть, например, обнаружение ошибки в программе пользователя, окончание работы устройств ввода-вывода, вмешательство оператора. В ОС ЕС прерывания могут относиться к одному из пяти типов :

- 1) программные прерывания ;
- 2) внешние прерывания ;

- 3) прерывания ввода-вывода ;
- 4) прерывания при обращении к супервизору ;
- 5) прерывания от схем контроля .

Программист- пользователь обычно имеет дело с прерываниями, вызванными ошибками в программах. Оценить причину прерывания можно с помощью информации, содержащейся в старом слове состояния программы (*OLDPSW*). Это слово выдается пользователю в системном сообщении после обработки ОС прерывания. Восьмая цифра указывает код прерывания, которому отвечает определенная причина. Например, коду *C* соответствует переполнение порядка при выполнении операции с плавающей точкой, коду *8* - переполнение при выполнении операции с фиксированной точкой. [7] .

Программы управления данными в ОС ЕС выполняют обмен информацией между оперативной памятью и внешними устройствами. Этот обмен может осуществляться с помощью нескольких методов [4,7] . Наличие шести методов доступа к данным объясняется тем, что ОС ЕС допускает четыре разных типа организации данных (последовательную, индексно-последовательную, прямую и библиотечную). Методы доступа к данным ориентированы на работу с данными определенных типов организации. Наиболее часто используется способ доступа с очередями. Он позволяет программисту не останавливаться подробно на планировании операций ввода-вывода. Это делает операционная система автоматически. Ввод данных выполняется раньше, чем на него поступит запрос из программы, а вывод данных осуществляется после запроса [6] .

2.6. Язык управления заданиями ОС ЕС ЭВМ

Общение пользователя с ОС ЕС реализуется с помощью языка управления заданиями (ЯУЗ). Каждое задание оформляется в виде совокупности операторов ЯУЗ, текстов программ пользователя и описания входных и выходных данных. Начинается описание задания с оператора *JOB* - идентификации задания. За ним могут следовать операторы описания наборов данных, общих для всего задания (например, описание библиотек пользователя, в которых содержатся программы, подлежащие выполнению). Далее следуют операторы описания шагов задания. Для описания шага задания программист должен записать исполнительный оператор *EXEC*, указывающий, какую программу нужно выполнить операционной системе, и оператор *DD* описания

необходимых этой программе входных и выходных данных.

Задание в общем случае может состоять из нескольких шагов. В языке управления заданиями ОС ЕС девять типов команд. Из них три типа — *JOB*, *EXEC*, *DD* — необходимы в любом задании. Под запись операторов ЯУЗ отводятся 72 колонки. Все операторы начинаются с двух косых черт — "слэшей", записывающихся в двух первых колонках. Если оператор не размещается на одной перфокарте или строке, его можно продолжить на следующих перфокартах. Разрывать оператор можно на запятой, причем в этом случае необходимо оставить хотя бы один пробел до 72 колонки. Продолжить запись оператора на следующей строке или перфокарте следует с любой колонки в интервале от 4 до 16.

Общий формат оператора ЯУЗ имеет вид

//имя — ключевое слово команды — информационная часть оператора. Ключевым словом команды является одна из команд *JOB*, *EXEC*, *DD*. Информационная часть оператора состоит из одного или нескольких параметров. Различают два типа параметров — позиционные и ключевые. Позиционные параметры должны быть записаны в определенном порядке. Ключевые параметры распознаются операционной системой по ключевому слову и знаку равенства после него. В отличие от позиционных эти параметры могут быть записаны в любом порядке. Под информационную часть оператора отводится поле, не имеющее определенной длины. Оно может начинаться с произвольной колонки, но ему должен предшествовать и за ним следовать по крайней мере один пробел. Внутри поля пробелы не допускаются.

Рассмотрим обязательные операторы ЯУЗ.

Оператор *JOB*.

Это первый и единственный оператор в любом задании ОС ЕС имеет формат

//имя задания *JOB* (учетная информация), параметры

Учетная информация включает индекс пользователя и его фамилию.

Параметры сообщают ОС ЕС следующую информацию:

MSGLEVEL = (*m*, *n*) указывает, какую информацию о задании нужно выдавать в выходной поток. При *m* = 0 выводится только оператор *JOB*; при *m* = 1 выводятся все входные управляющие операторы задания, включая операторы каталогизированной процедуры; при *m* = 2 выводятся только входные управляющие операторы задания. При *n* = 0 сообщения о распределении-окончании выводятся только в случае аварийного завершения задания; при *n* = 1 выво-

дятся все сообщения о распределении-окончании.

TIME = (минуты, секунды) задает максимальное время использования центрального процессора для выполнения задания .

PRTY = *п.п.* сообщает ОС приоритет задания .

MSGCLASS = *x* позволяет направить все относящиеся к данному заданию сообщения планировщика заданий в выходной класс, отличный от *A*. *x* может принимать значения от *A* до *Z* или от 0 до 9.

CLASS = класс задания определяет конфигурацию периферийного оборудования, необходимого для выполнения задания . Класс задания принимает буквенные значения от *A* до *O*. Если параметр пропущен, то заданию назначается класс *A*.

REGION = *п.п.п.п.п.п.к* позволяет запросить максимальную область оперативной памяти, которая должна быть выделена заданию. Если параметр пропущен, то заданию выделяется зона стандартных размеров. *п.п.п.п.п.п.к* может принимать значения от 1 до 16383 кбайт.

ROLL = (*m* , *n*) сообщает ОС, что в случае нехватки собственной зоны оперативной памяти для выполнения задания память выделяется за счет другого задания (свертки). Выполнение свернутого задания временно прекращается. Параметр содержит два подпараметра, первый из которых указывает, может ли задание быть свернуто для удовлетворения потребности другого задания в памяти. Первый подпараметр принимает значения либо *YES*, либо *NO* . Второй подпараметр указывает, может ли задание вызывать свертку другого задания, и принимает значения *YES* или *NO* . Если параметр пропущен, то подразумевается *ROLL* = (*YES*, *NO*).

Оператор *JOB* может содержать еще несколько параметров [6], которые применяются программистами реже.

Оператор *EXEC*

Этот оператор предваряет каждый отдельный шаг задания. Появление его во входном потоке означает конец предыдущего шага задания и начало нового шага. За оператором *EXEC* могут располагаться операторы *DD* и данные, необходимые для выполнения этого шага задания. Оператор *EXEC* имеет формат

//имя шага *_EXEC_* параметры

Рекомендуется каждому шагу задания давать свое имя, причем имена в одном задании не должны совпадать.

Основная функция оператора *EXEC* - указать ОС программу или процедуру, которая должна быть выполнена. Программа указыва-

ется с помощью параметра *PGM*, а процедура — с помощью параметра *PROC*. Эти параметры являются взаимоисключающими. Для того чтобы выполнить каталогизированную процедуру, надо записать параметр в виде *PROC* = имя процедуры, а для выполнения программы, записанной в системной библиотеке (либо личной библиотеке пользователя, описанной в *DD*-операторе), необходимо указать имя программы *PGM* = имя программы. Например:

```
// STEP EXEC PGM = IEYF0RT, REGION = 100K
```

или

```
// STEP EXEC PROC = COMPLGO, REGION = 228K
```

Допускается обращение к каталогизированной процедуре путем указания в поле параметров просто имени процедуры. Например:

```
// STEP EXEC COMPLGO, REGION = 228K
```

Оператор *EXEC* может содержать еще ряд параметров, которые позволяют пользователю управлять порядком выполнения шагов задания, передавать программе на шаге задания любую информацию [6].

Оператор *DD*

С помощью оператора *DD* определяются все наборы данных, используемые или создаваемые на шаге задания. Операторы *DD* следуют за оператором *EXEC* и имеют следующий формат:

```
//dd- имя DD = параметры.
```

dd — имя указывает имя используемого набора данных. Если в пункте задания происходит обращение к каталогизированной процедуре, то операторы *DD* этого пункта задания используются для добавления к этой процедуре новых операторов *DD*. Для этого в поле *dd* — имени добавляемого *DD*-оператора имя записывается (имя *ds* пункта процедуры . имя *DD*). Например, пусть требуется выполнить каталогизированную процедуру *FORTCCLG*, которая транслирует, редактирует и выполняет программу, написанную на Фортране. Для того чтобы передать программе на шаге выполнения входные данные, записанные на перфокартах, необходимо добавить процедуре входной набор данных с помощью оператора *//GO.SYSIN DD * * ** и поместить за ним перфокарты с данными. Звездочка (***) указывает системе, что набор данных следует сразу за *DD*-оператором во входном потоке. Когда ОС прочитает символ ***, она будет записывать все следующие за *** карты на буферное устройство до тех пор, пока не обнаружит карту, начинающуюся с *//* или */**. Данные, записанные на буфере, будут использованы как входной набор.

Параметры *DD*-оператора служат для передачи ОС следующей ин-

формации.

DDNAME = *dd*- имя описывает набор данных во входном потоке.

SYSOBT = (имя класса) указывает класс выходного потока. Имя класса принимает значения от А до Z и от 0 до 9. Обычно класс А закреплен за набором данных, выводимым на АЦПУ.

DSNAME = *dd*- имя или *DSN* = *dd*- имя используется, например, при обращении к библиотечному набору данных. Этот параметр служит для указания имени существующих наборов данных, которые будут использоваться на этом шаге, либо указания имени вновь создаваемых наборов.

UNIT = (групповое имя или тип устройства, число устройств) сообщает ОС типы и количество устройств ввода-вывода для размещения наборов данных. Например, *UNIT* = (5056,2) соответствует заказу двух накопителей на сменных магнитных дисках ; *UNIT* = *TAPE* соответствует заказу одного накопителя на магнитной ленте.

VOLUME = (*PRIVATE*, *RETAIN*, порядковый номер тома, число томов, *SER* = (серийный номер)) определяет том (пакет магнитных дисков или бобину магнитной ленты), на котором хранится или будет храниться набор данных. Наличие подпараметра *PRIVATE* означает, что том является личным и не может быть использован ни для какой другой цели, кроме работы с описываемым набором данных. Подпараметр *RETAIN* запрещает снимать том, поскольку он понадобится при выполнении следующих шагов задания. Порядковый номер тома указывает начальный номер тома в наборе данных.

Набор данных может быть большим и записанным на нескольких томах. Их число задается подпараметром, стоящим после порядкового номера тома. *SER* = (серийный номер) указывает серийные номера в головных метках для всех требуемых томов.

DCB = (*BLKSIZE* = *xxx*, *LRECL* = *ууу*, *RECFM* = *xx*) задает информацию для организации блоков в наборе данных. С помощью подпараметра *BLKSIZE* = *xxx* задается длина (*xxx*) блока или физической записи в байтах, *LRECL* = *ууу* - длина логической записи в байтах, *ZZ* - указывает, является ли запись сблокированной (*PB*) или фиксированной длины (*F*).

LABEL = (порядковый номер набора данных, тип метки, защита файла, *EXPD* = *ххууу* либо *RETPD* = *пппп*) служит для описания метки набора данных. Порядковый номер набора данных указывается, если входной набор не первый в томе. Тип метки сообщает ОС, какая метка будет использована. (*SL*) соответствует стандартному типу.

Подпараметр защиты файла принимает значения :

NOPWREAD , означающее необходимость ввода оператором ЭЕМ с консоли пароля, который должен совпадать с паролем в метке набора ;

PASSWORD означает, что пароль требуется только при записи в набор данных ;

EXPD7 = ххууу - устанавливает дату уничтожения набора (хх - год, ууу - день года) ;

RETPD = пппп - определяет срок хранения х набора в днях с момента создания.

DISP = (статус, диспозиция) служит для указания состояния набора данных в момент выполнения шага задания и после его завершения. Статус указывает, что набор данных входной (*old*) или выходной (*new*). Диспозиция указывает ОС, что нужно делать с набором после завершения шага - сохранить набор (*keep*) или уничтожить (*delete*).

Вот пример оформления задания, включающего трансляцию, редактирование и выполнение программы, составленной на языке Фортран:

```
// EXCL_008_(808000), 'ПЕТРОВ П.П.', MSGLEVEL=(2,0),  
// REGION=300K  
// CLASS=C, PRTY=7,  
// TIME=(5,30)
```

Текст исходного модуля на Фортране

```
/*  
//  
даные c  
/*  
//
```


Г Л А В А 3

СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ САПР

3.1. Структура специального программного обеспечения САПР

Специальное программное обеспечение (СПО) САПР представляет собой комплекс программ, реализующих алгоритмы проектных операций и процедур. Общая структура СПО показана на рис. II. Основным требованием к структуре СПО является обеспечение его эффективности при выполнении автоматизированного проектирования, а также уменьшение затрат времени и средств на создание СПО. Строится СПО по модульному принципу. Это ускоряет процесс его создания, т.к. позволяет вести разработку различных модулей независимо друг от друга, делает САПР открытой, а само СПО гибким и легко адаптируемым к изменениям в задачах, решаемых САПР.

Модульная структура СПО – иерархическая, построенная по принципу вложенности. Если совокупность модулей более низкого порядка обладает признаками самостоятельной программы или входит как составная часть более чем в одно сочетание модулей высокого порядка, ее можно считать отдельным модулем.

Модули, предназначенные для совместного выполнения, объединяются в пакеты прикладных программ. Каждый пакет обычно ориентирован на обслуживание отдельной подсистемы САПР и характеризуется определенной специализацией [I-3].

Для организации управления процессом проектирования при автоматизированном выполнении проектных процедур создается обслуживающая подсистема управления – монитор САПР. В функции монитора входят :

- а) загрузка и активизация компонентов СПО, организация и управление маршрутами их выполнения ;
- б) организация взаимодействия САПР с пользователем, в том числе в режиме диалога ;
- в) распределение ресурсов ЭВМ с учетом запросов и приорите-

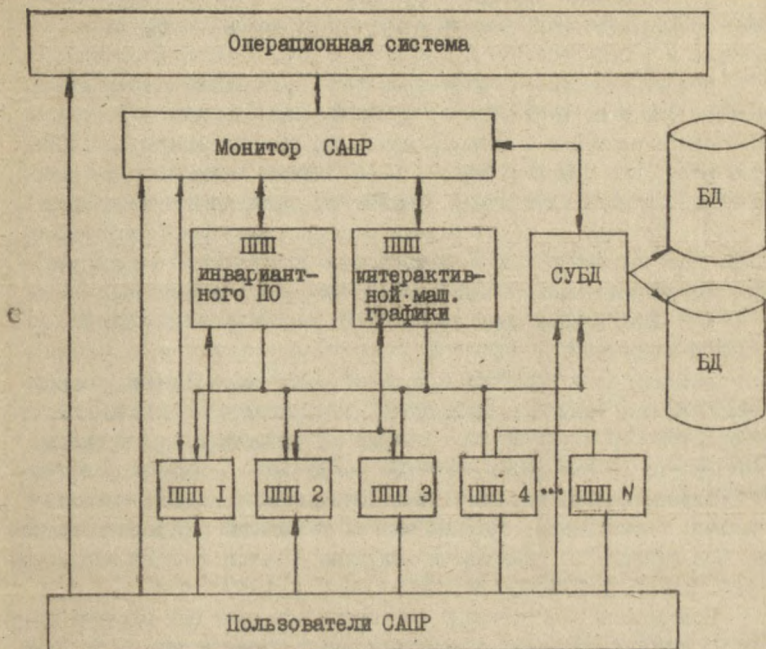


Рис.И. Организация функциональных связей в программном обеспечении САПР.

СУБД - система управления базами данных (БД)

тов пользователей ;

г) выполнение сервисных функций (регистрации пользователей, сбора статистики, защиты ресурсов от несанкционированного доступа) [3].

Мониторная система действует под управлением операционной системы ЭВМ и не дублирует ее функций. Пользователь общается с мониторной системой с помощью языка управления монитором САПР. Этот язык, как правило, прост. В его основе лежат команды вызова проектирующих процедур и задания им управляющих параметров [2, 5]. Получив от пользователя задание, мониторная система вырабатывает структуру рабочей программы для заданного маршрута проектирования, выбирая необходимые для этого модули ППП.

Для организации подсистем в САПР используют простую и сложную структуры.

Простой структуре соответствуют программы или пакеты программ, имеющие узкую направленность и проблемную ориентацию. Пакеты простой структуры состоят только из функциональных модулей, реализующих те алгоритмы, которые могут использоваться различными пользователями. Это могут быть алгоритмы численных методов анализа, оптимизации, решения систем уравнений определенного типа. Все функции по управлению модулями пакетов простой структуры выполняют операционная система и монитор САПР.

Подсистемы САПР сложной структуры образуют ППП сложной структуры, которые включают обрабатывающие и управляющие части. Эти подсистемы используются тогда, когда в САПР либо отсутствует развитая мониторная система, либо ее возможности по управлению модулями пакета недостаточны [5]. В этом случае в состав пакета включается мониторная система ППП.

Она работает под управлением монитора САПР и операционной системы ЭВМ.

Монитор пакета представляет собой набор системных программ, образующих функциональные подсистемы :

1) процессор входного языка пакета, который осуществляет перевод задания с языка пакета на внутренний язык ЭВМ ;

2) планировщик, который составляет последовательность модулей ;

3) исполнитель, который реализует построенную планировщиком цепочку модулей.

Вызов нужных подсистем монитора пакета в процессе сборки рабочей программы осуществляет ядро, или резидент пакета. Ядро пакета постоянно находится в оперативной памяти ЭВМ. В нем анализиру-

ются команды, содержащиеся в задании пользователя, и вырабатываются соответствующие управляющие команды для работы ЭВМ.

Пакеты сложной структуры для общения с пользователем имеют специальный язык – язык пакета, структура которого будет рассмотрена позже.

Пакеты сложной структуры, как правило, являются проблемно-ориентированными и предназначены для выполнения отдельной проектной процедуры или операции.

Помимо объектно-ориентированных ППП в состав СПО входят объектно-независимые и обслуживающие подсистемы, например подсистема изготовления проектной документации и интерактивной машинной графики. Эта подсистема занимает промежуточное положение между проектирующими и обслуживающими подсистемами СПО. Она используется для наглядного представления входной и выходной информации в проектирующих подсистемах, а также для выдачи необходимой проектной документации. Поэтому данная подсистема может состоять из нескольких ППП, которые входят как составные части в проблемно-ориентированные пакеты сложной структуры.

3.2. Принципы организации пакетов прикладных программ

Пакеты прикладных программ составляют основу специального программного обеспечения САПР. Разработка ППП связана с решением ряда проблем:

- 1) выделение модулей и определение их структуры;
- 2) выбор технологии сборки программы из модулей;
- 3) обеспечение интерфейсов модулей (связи модулей по информации);
- 4) создание или использование языка пакета [2,3].

Все эти проблемы должны быть решены в процессе разработки пакета коллективом разработчиков СПО САПР.

Эффективность создаваемого ППП в значительной степени зависит от разбивки программного обеспечения на модули.

Под модулем будем понимать генерируемую или библиотечную программу или ее часть, способную входить в сочетания с другими модулями как самостоятельный элемент [1]. Модулем может быть крупная программа, но возможно оформление в виде самостоятельных модулей отдельных фрагментов программ.

В состав ППП могут входить модули нескольких типов:

- а) функциональные;

- б) библиотечные ;
- в) оригинальные [2] .

Тип модулей отражает принципы, по которым они выделялись либо создавались при разработке СПО САИР. Обычно используется несколько принципов выделения модулей :

- а) функциональный ;
- б) выделения общей части алгоритма ;
- в) выделения части программы, занимающей стандартный объем оперативной памяти ЭВМ ;
- г) уменьшения числа переключений с одного модуля на другой.

Могут использоваться и другие соображения.

Функциональный принцип предполагает выделение части алгоритма или программы, выполняющей законченную функцию. Например, задание начальных или граничных условий, решение уравнений определенного типа .

Функциональные модули могут, как видно из примера, быть инвариантными и проблемно-ориентированными. Инвариантные модули реализуют такие алгоритмы, которые могут использоваться в различных пакетах. Функциональные модули объединяются по какому-либо признаку в библиотеки. Например, это могут быть библиотеки математических моделей элементов конструкций, электрических схем и т.д., библиотеки алгоритмов выполнения отдельных процедур и т.п.

Создавая пакет, необходимо стремиться с меньшим числом модулей решать как можно больший класс задач. В связи с этим в качестве модуля может быть выделена часть алгоритма, не имеющая законченного смысла, но использующаяся при решении разных по своей постановке задач.

Часто модуль выбирается по соображениям объема используемой оперативной памяти при сегментации программ. В этом случае вводят понятия простого, автономного и неавтономного модулей. Простой модуль можно разместить в оперативной памяти ЭВМ вместе с требуемой для его работы информацией. Если этого сделать не удастся, программа разбивается на несколько сегментов-модулей, которые не имеют законченного смысла и называются неавтономными. При использовании ЭВМ единой серии рационально выделение модулей объемом в один лист оперативной памяти, т.е. 1024 байт [9] .

Модули, выделенные не по функциональному принципу, также объединяются в библиотеки, но, в отличие от функциональных модулей, для сборки из них рабочей программы требуется вмешательство мониторной системы пакета.

В ППП сложной структуры применяются модули еще одного типа — оригинальные, которые отражают специфические особенности конкретного проектируемого объекта. Они не являются типовыми и должны создаваться каждый раз заново. Генерацию оригинальных модулей рабочих программ осуществляет по директивам пользователя языковый процессор мониторной системы пакета.

При выборе модулей стремятся к максимально возможному сокращению объема входных, выходных данных и числа переключений с одного модуля на другой.

3.3. Организация рабочих программ в САПР

Для выполнения проектных операций и процедур из модулей, входящих в состав ППП, и оригинальных генерируемых модулей создаются рабочие программы. Получение рабочих программ возможно двумя методами: методом компиляции и методом интерпретации [1,2]. Метод компиляции предполагает, что для каждой операции вычислительного процесса в рабочей программе формируется своя отдельная машинная команда. При компиляции этапы трансляции и выполнения программы четко разделены, а полученная рабочая программа линейна. Она состоит из команд, исключающих передачи управления. Скомпилированные программы наиболее экономичны по затратам машинного времени на их выполнение, но занимают значительно большую область оперативной памяти по сравнению с программами, полученными методом интерпретации.

Метод интерпретации подразумевает, что рабочая программа не создается в окончательном виде до начала этапа выполнения. Программа генерируется по частям при переходе от одной директивы входного языка к последующей. При этом затраты машинного времени возрастают, особенно при многократном обращении к повторяющимся модулям. При итерационном вычислительном процессе, например, приходится многократно повторять выполнение одних и тех же вспомогательных операций по генерации рабочей части программ [1-3,9]. Вместе с тем метод интерпретации существенно уменьшает область оперативной памяти, необходимой для размещения программы.

Наличие в САПР развитых операционных и мониторных систем позволяет использовать при генерации рабочих программ оба метода.

Метод интерпретации чаще применяется при генерации программ более высоких уровней и обеспечения диалогового режима общения

пользователя с САПР и ППП. Метод компиляции применяют при генерации программ самых низких уровней, которые используются наиболее часто.

Для упрощения согласования модулей на этапе сборки, обеспечения информационной и программной совместимости модулей разработаны единые правила построения модулей и выпуска документации сопровождения.

Каждый модуль состоит из тела модуля и паспорта модуля.

Тело модуля — это его содержательная часть или, иными словами, наполнение. Наполнение модулей может быть различным, не только предназначенным для проведения расчетов, но и для управления вычислительным процессом, распределения и обработки информации. Тело модуля может задаваться и храниться в библиотеке в нескольких видах :

а) в текстовом виде на некотором алгоритмическом или специальном языке;

б) записанном на некотором промежуточном языке загрузки в относительных адресах ;

в) записанном в машинных командах и истинных адресах.

В зависимости от вида представления тела модулей в состав ППП будут входить библиотеки исходных, объектных или абсолютных модулей.

Паспорт модуля служит для хранения информации, необходимой при сборке рабочей программы. Он содержит сведения о функциях модуля, размере оперативной памяти, необходимой для размещения модуля, о списке входных и выходных параметров, времени работы модуля, способах обмена информацией с другими модулями, местонахождении в ППП, перечень возможных прерываний.

Паспорт модуля может входить в каталог и храниться в библиотеке, входящей в состав пакета, либо храниться в автоматизированном банке данных.

Модули из состава ППП записаны в библиотеки и не связаны между собой. Для того чтобы собрать модули в единую рабочую программу, необходимо :

а) задать управляющие связи, т.е. установить последовательность работы модулей ;

б) осуществить информационные связи, т.е. вместо условных адресов в разных модулях установить конкретные значения истинных адресов.

Рассмотрим наиболее употребительные варианты сборки.

Ручной способ сборки.

Ручной способ сборки основан на использовании библиотек и существующих операционных систем. Применяется в основном в пакетах простой структуры. Связь между модулями организуется через иерархическую структуру, в вершине которой находится основная программа — единственная изменяемая от задачи к задаче часть программного комплекса. Остальные модули оформлены в виде подпрограмм. Большое распространение получили ППП с ручной сборкой, использующие библиотеки исходных модулей на Фортране. В этих пакетах основная программа обычно содержит четыре элемента:

- 1) описание многомерных массивов и *common*-блоков;
- 2) вызов модуля, управляющего счетной частью;
- 3) вызов модуля, задающего исходные данные;
- 4) вызов модуля обработки результатов.

Вместо вызова модуля управляющего счетной частью могут быть заданы прямые обращения к соответствующим модулям (подпрограммам). Записывая с помощью операторов *CALL* цепочку вызова подпрограмм, пользователь организует связи по управлению.

Связи по информации организуются также при участии пользователя с помощью программы редактора ОС.

В фортрановских программах существуют два способа задания связей по информации:

- а) с помощью списков формальных и фактических параметров;
- б) с помощью *common*-блоков.

На этапе редактирования устанавливается эквивалентность адресов ячеек оперативной памяти, которые будут выделены параметрам, содержащимся в *common*-блоках. Предложения *common* должны быть записаны в этом случае во всех модулях.

Задавая в операторе *CALL* вызова очередного модуля вместо формальных параметров их фактические значения, программист осуществляет передачу данных из одного модуля в другой.

При ручной сборке нет необходимости в использовании монитора ППП. Функции планировщика и исполнителя выполняют обрабатывающие программы ОС. Вместе с тем следует отметить, что этот способ сборки имеет ряд неудобств. Пользователь берет на себя утомительную работу по сборке модулей. Он должен внимательно распределить память в *common*-блоках, везде указать значения фактических параметров, согласовать их по типу, количеству. При большом числе параметров эта процедура достаточно трудоемка.

Кроме того, при любом изменении в одной подпрограмме приходится просматривать все остальные модули для внесения корректив. Перед каждой сборкой пользователь должен выяснить, какие изменения были внесены в модули в процессе совершенствования. Все эти неудобства приводят к тому, что нарушаются принципы разработки САПР — открытость, информационная независимость, независимость разработки. Поэтому в настоящее время при создании ППП чаще используются автоматические способы сборки модулей.

Автоматические способы сборки.

При разработке ППП нашли применение несколько принципов автоматической сборки рабочих программ :

а) сборка с изначальным заданием связей по управлению ;

б) сборка с изначальным установлением информационных связей.

Принцип сборки с изначальным заданием связей по управлению эффективен при небольшом количестве реализуемых схем счета (т.е. последовательностей вызова модулей). Особенно эффективен он в случае, если схема счета практически остается неизменной, а меняются лишь модули. Такая ситуация возникает, когда расчеты проводятся с использованием разных моделей одного и того же объекта.

Простейший вариант реализации этого принципа — генерация программ с использованием каталога задач. Названия всех задач, которые могут быть решены на основе модулей данного ППП, собираются в каталоге. Каждому названию заранее ставится в соответствие цепочка вызовов соответствующих модулей. Эта цепочка может оформляться в виде модуля. По запросу пользователя планировщик осуществляет вызов цепочки задачи и генерацию рабочей программы. Вызов выполняется вспомогательной программой монитора ППП, которая выполняет настройку модулей по информации. С целью повышения качества рабочей программы часто все информационные связи объединяют в модуль памяти.

Хранение фиксированных цепочек задач в каталоге рационально в случае, если их число невелико.

Сборка с изначальным установлением информационных связей применяется при большом количестве модулей. Сборка по такому принципу применяется в ППП, выполняющих конструкторские расчеты, проектирование различных устройств. Суть этого способа сборки заключается в следующем. Пользователь задает последовательность величин, подлежащих определению. Планировщик, обрабатывая эту цепочку, находит последовательность вызова модулей и формирует рабочую программу.

С точки зрения пользователя, более естественным является задание последовательности величин, которые требуется найти. Но в этом случае решение может быть не единственным, т.к. искомые величины могут быть получены с помощью различных модулей. Например, решение системы уравнений может быть получено с помощью различных численных методов, имеющих разную скорость сходимости. Двойственности не возникает, если задавать последовательность входных величин. Так и поступают на практике.

Управление работой планировщика осуществляет пользователь с помощью языка ППП.

3.4. Язык пакета прикладных программ

Язык пакета является языком высокого уровня, с помощью которого пользователь может описывать не алгоритм решения задачи, а только ее постановку в терминах соответствующей предметной области [1-5,9]. Язык пакета обязательно является предметно ориентированным. Обычно он представляет собой продукт синтеза языка описания задачи и языка управления.

В САПР входят разные пакеты, каждый из которых имеет ориентацию на определенную подсистему. Поэтому для работы пользователя с пакетами применяется не один, а несколько проблемно-ориентированных языков. В пакетах простой структуры в качестве языка пакета используется язык управления заданиями операционной системы либо аналогичный язык описания заданий мониторной системы. Эти языки не учитывают специфики задач, решаемых с помощью пакета, и требуют от пользователя САПР достаточно высокой квалификации программиста.

В САПР с развитым специальным программным обеспечением чаще используются пакеты сложной структуры. Их отличает наличие собственного монитора и проблемно-ориентированного входного языка.

Входные языки включают операторы двух типов, образующие язык описания объектов (ЯОО) и язык описания заданий (ЯОЗ).

ЯОО служат для описания исходной информации об объекте проектирования. Поскольку эта информация может представляться в виде схем, чертежей, алгоритмов функционирования, использование языков описания объектов могут быть схемными, графическими и языками моделирования.

Схемные языки широко применяются при конструировании электронных и электрических устройств. Графические языки применяют-

ся в системах машинной графики. Языки моделирования применяются в подсистемах имитационного моделирования.

ЯОЗ служат для описания заданий на выполнение проектных операций и процедур. По структуре и грамматике ЯОЗ напоминает язык управления заданиями ОС ЕС ЭВМ. Задание представляет совокупность операторов, каждый из которых состоит из команды и операнда. В качестве операндов могут использоваться имена модулей ППП, значения параметров проектируемого объекта.

Кроме входных языков, в ППП сложной структуры могут использоваться выходные языки, языки промежуточные, внутренние и сопровождения.

Выходные языки используются для представления результатов выполнения проектных процедур в виде удобном для пользователя или для последующей обработки другими подсистемами САПР.

Языки сопровождения служат для непосредственного общения пользователя с ЭВМ в процессе обработки и исполнения рабочей программы. Язык сопровождения кроме специфических средств редактирования и управления может содержать элементы входного и выходного языка. В зависимости от режима работы операционной системы ЭВМ языки сопровождения могут быть диалоговыми и недиалоговыми.

Промежуточные и внутренние языки ППП предназначены для представления информации на определенных стадиях ее обработки при исполнении рабочей программы. Обычно в их качестве выступают языки программирования, дополненные в некоторых случаях псевдооператорами.

Г Л А В А 4

ТЕХНОЛОГИЯ РАЗРАБОТКИ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ САПР

4.1. Разработка СПО САПР

Специальное программное обеспечение САПР состоит из десятков, а иногда и сотен компонентов. Разработка СПО является сложной научно-технической задачей. На качество СПО оказывает влияние эффективность алгоритмов решения задач математического моделирования, степень развития лингвистического обеспечения САПР, эффективность операционных и мониторинговых систем, а также ряд других факторов. Выполняется разработка СПО САПР в несколько этапов [5,10] .

На первом этапе с учетом требований пользователей формулируются общие требования к СПО САПР и его компонентам. Этот этап заканчивается выработкой полного и непротиворечивого перечня требований к СПО.

На втором этапе на основании этого перечня осуществляется а) выделение проектных процедур и операций, подлежащих автоматизации ;

б) выбирается тип ЭЕМ и операционные системы для обслуживания подсистем, выполняющих автоматизированное проектирование в зависимости от объема и сложности производимых расчетов ;

в) выбирается круг математических моделей объектов и процессов, которые целесообразно использовать в автоматизированном проектировании .

В результате выполнения этого этапа вырабатываются спецификации на программные комплексы и ППП.

Спецификация служит заданием на разработку программ и ее требования обязательны для выполнения разработчиком. Прежде чем разработчик СПО напишет программу для решения задачи, он должен четко представить ее цель, изучить математическую постановку и

методы решения. Эти сведения, а также ряд ограничений на создаваемые программы и должны содержаться в спецификации. Следует однако заметить, что до настоящего времени нет единого мнения о том, являются ли предъявляемые к спецификации требования формальности, однозначности и точности обязательными или желательными. Дело в том, что разработка спецификации не может быть выполнена только специалистами по структуре ПО САПР (системными аналитиками), хотя именно на них и возложена эта задача. В подавляющем большинстве случаев они не в состоянии определить четкие требования к программам и подсистемам, выполняющим конкретные проектные операции или процедуры. Поэтому обычно в спецификации достаточно четко и полно формулируется, что именно должна делать программа, но без уточнения, как и с помощью каких методов она это будет делать.

Помимо функциональных требований (т.е. формулировки задачи) в спецификации содержатся эксплуатационные требования, в которые входят :

- а) ограничения на объем занимаемой оперативной памяти ;
- б) перечень технических средств, предоставляемых для функционирования данной подсистемы САПР ;
- в) указание входного и выходного языка, а также языка программирования ;
- г) указание формы представления входной и выходной информации ;
- д) указание способа информационной связи с другими модулями и подсистемами САПР.

На третьем этапе разработки СПО в соответствии с требованиями спецификации осуществляется создание конкретных ППП.

Начинается разработка программ с выполнения подготовительных работ :

- а) выбора модели процесса или объекта ;
- б) математической формулировки задачи ;
- в) выбора метода решения задачи ;
- г) разработки общего алгоритма решения задачи .

Выполнить их могут только специалисты, обладающие знаниями в соответствующей прикладной области.

После того как будет сформирован алгоритм решения задачи, разрабатывается структура программы, выделяются модули, устанавливаются их иерархические уровни. Решение этих вопросов должно осуществляться в тесном сотрудничестве специалистов в прикладной

области и программировании.

Алгоритмы модулей программ кодируются, отлаживаются и тестируются. Разработка программных комплексов и ППП заканчивается выпуском программной документации и передачей программ для эксплуатации.

Последний этап разработки СПО состоит в отладке и тестировании работы программного обеспечения в целом. Заканчивается он выпуском документации сопровождения и вводом СПО в эксплуатацию.

4.2. Технология программирования модулей

После детальной разработки алгоритма программного модуля начинается этап кодирования (оставления программы для ЭВМ). Учитывая большой объем и трудоемкость этой работы, очень важно выбрать наиболее подходящую для разработчиков технологию программирования. В настоящее время распространены и используются не зависимо от языка программирования четыре технологии:

- а) модульное программирование;
- б) структурное программирование;
- в) нисходящее программирование;
- г) HIPO – технология.

В основе модульного программирования лежит искусство разбиения задачи на некоторое число модулей разной иерархии, часть из которых уже существует. Это позволяет использовать уже готовые отлаженные модули и сократить сроки создания всей программы. Программирование комплекса начинается с кодирования, отладки и тестирования модулей низшего порядка. При конструировании модулей более высокого порядка в этой технологии широко используется техника макрорегионации и автоматизации сборки готовых модулей [5, II].

Разработанные в рамках этой технологии программы легко читаются пользователем и могут быть легко модифицированы. Для таких программ удается дать формальное и строгое обоснование их правильности и соответствия спецификации.

Структурное программирование основано на принципе пошаговой детализации. Главная особенность этой технологии в том, что программисту заранее определяются допустимые и запрещенные структуры. Алгоритм записывается только с помощью допустимых структур, количество которых невелико – 2 или 3 [11].

Наиболее распространенные структуры показаны на рис.12.

Достоинством этой технологии программирования является ясная логика построения программ, которая существенно облегчает их отладку.

Идея нисходящего программирования состоит в том, что программирование модулей осуществляется параллельно с разработкой их алгоритма. Программный комплекс оформляется в виде иерархической схемы подключения модулей. Причем структура программы выбирается такой, что описание работы модулей более высокого порядка не зависит от описания модулей более низкого порядка. Кодирование начинается с модулей высокого порядка, а неразработанные модули более низкого порядка имитируются "пустышками" или, как их еще называют, "заглушками".

Весь программный комплекс кодируется и отлаживается по уровням с самого высокого до самого низкого. Каждый нижний уровень комплекса отлаживается с помощью тестов, уже проверенных на более высоком уровне.

В основе HIPO - технологии лежит пошаговая детализация программного комплекса. Для этой цели разработаны специальные шаблоны, бланки и типовые диаграммы. Обычно в HIPO - технологии три типа диаграмм. Диаграммы первого типа предназначены для записи блок-схемы комплекса на самом высоком уровне иерархии. Заполняются они по окончании разработки всего программного комплекса. Диаграммы второго типа задают иерархию связи и сборки диаграмм третьего типа (IPO - диаграмм), которые являются основными. На IPO-диаграммах три колонки. В первой записывается входная информация, в средней - алгоритм преобразования входных данных в выходные, в последней колонке записывается выходная информация.

HIPO-технология обеспечивает полностью формализованную систему ссылок и ясное описание алгоритма программы на каждом уровне. Каждому уровню программы соответствует одна IPO-диаграмма. На диаграмме содержится не более 5-7 блоков, детализирующих программу на этом уровне.

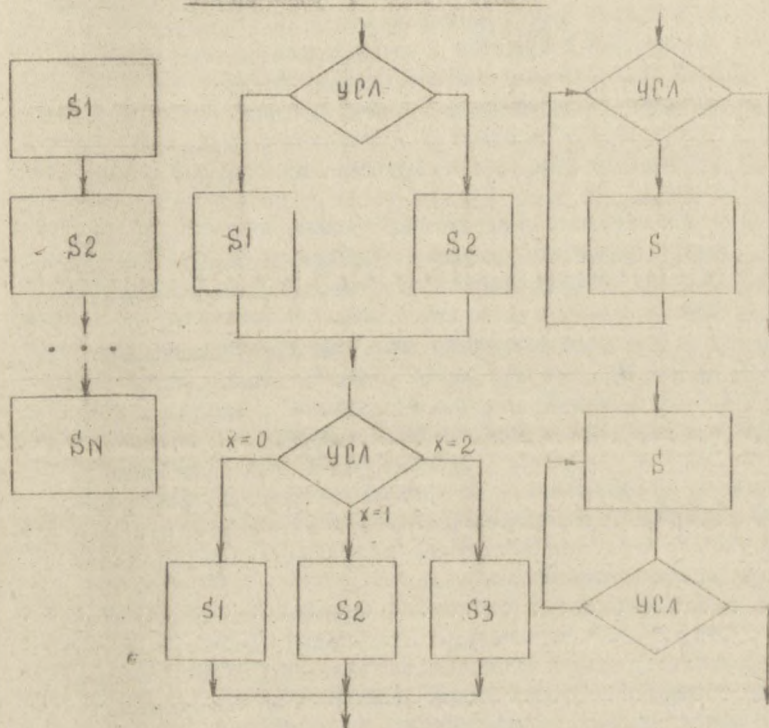
К недостаткам HIPO - технологии можно отнести невозможность проверки и отладки отдельных модулей до полного окончания заполнения всех IPO - диаграмм и их увязки друг с другом. Однако процесс кодирования и выпуска программной документации значительно упрощается и может выполняться менее квалифицированными сотрудниками.

Линейная
структура

Структура
выбора

Циклическая
структура

Обозначение в блок - схемах



Реализация средствами языка Паскаль

S1: $x := x + a$

S2: $P(x, y)$

SN: $x := F(y) + P(x, y)$

if УСЛ then S1 else S2

case УСЛ of (S1, S2, S3)

while УСЛ do S

repeat S until УСЛ

Рис.12. Допустимые элементы структурного программирования

На этапе кодирования алгоритм решения задачи записывается на одном из языков программирования. Если язык программирования не оговорен в спецификации, то выбор его возлагается на программиста - разработчика. Важно, чтобы язык программирования согласовывался с принятой технологией программирования и был удобен для записи алгоритма. Кроме этого, необходимо учитывать:

- а) тип ЭМ и операционную систему, на которой будет эксплуатироваться программный комплекс;
- б) наличие или отсутствие в операционной системе эффективного транслятора с соответствующего языка;
- в) возможность перенесения программ с одного типа ЭМ на другие, с учетом развития технического обеспечения САПР;
- г) возможность кодирования структур алгоритма средствами языка;
- д) эффективность закодированных программ (по объему занимаемой оперативной памяти, межмодульным связям и т.п.);
- е) удобство отладки программ и производительность кодирования.

В настоящее время для кодирования инженерно-технических программ широко используются алгоритмические языки: Алгол-60 и его расширения, Алгол-ЕС, Алгол-ГДР, Алгол-68, Фортран и его расширения - Фортран-IV, Фортран-ДУБА, Фортран-77, Форекс-3, Форекс-5, ПЛ-1 и его версии, Паскаль, ассемблеровские языки (Ассемблер-ЕС, Мадлен и др.). Эти языки имеют развитый аппарат для кодирования алгоритмов преобразования числовой информации. Причем по мере появления более поздних версий и расширений возможности этих языков сближаются.

Пожалуй, самым распространенным алгоритмическим языком является Фортран. Трансляторы с Фортрана имеются практически во всех операционных системах. В ОС ЕС имеется высокоэффективный оптимизирующий транслятор уровня H, в ОС Дубна и ОС ДИПАК имеются эффективные трансляторы с Фортрана-Дубна и Форекса. Недостатком Фортрана является отсутствие языковых средств распределения памяти и возможность использовать только простейшие структуры и типы данных. Расширения Фортрана позволяют ввести ряд новых возможностей, частично компенсирующих его недостатки, но при этом возникают большие трудности при переносе программ с ЭМ одного типа на другие. Некоторые из этих трудностей обсуждаются в [12].

Алгоритмический язык Алгол имеет большие возможности по

сравнению с Фортраном, однако эффективного транслятора с него для ЭВМ единой серии нет.

Алгоритмические языки ПЛ-I и Паскаль имеют средства реализации конструкций структурного программирования. Поэтому их применение целесообразно и эффективно при использовании технологии структурного программирования.

4.3. Отладка и тестирование программных модулей

После окончания кодирования программных модулей основной этап создания программного комплекса заканчивается. Вместе с тем у разработчика и у пользователя нет уверенности в том, что комплекс будет правильно выполнять решение задачи. Ошибки могут закрадываться в программу на любом этапе ее создания, начиная с математической постановки задачи. Поэтому перед началом эксплуатации программных комплексов проводится доказательство их правильности. Доказательство может быть дано либо математическим методом, либо методом тестирования. Математический метод применяется для достаточно коротких программ. Основным методом доказательства правильности программ остается тестирование, хотя и оно не всегда позволяет выявить ошибки. Тестирование программ трудно поддается формализации и его проведение в основном базируется на опыте программиста-разработчика СЮ САПР.

При создании ППП для решения задач динамической прочности элементов конструкций можно учесть рекомендации по отладке и тестированию программ, сформулированные в [13].

Тестирование и отладка обычно проводятся поэтапно. На первом этапе исправляются ошибки трансляции. На втором исследуются реализация численного метода решения задачи, его сходимость и точность. На третьем проверяется работоспособность программы на всем диапазоне изменений входных параметров. Поскольку второй и третий этапы тестирования связаны с большим объемом пусков программы, необходимо стремиться уменьшить время ее работы до получения результатов. Это часто удается сделать, уменьшая размерность массивов переменных (например, узлов сетки).

Для тестирования должна использоваться любая доступная информация о решении (имеющиеся данные физического эксперимента, точные аналитические решения для частных случаев и т.д.).

Наиболее сложной проблемой при тестировании и отладке является локализация и определение ошибки программы. Дело в том, что

в программе может содержаться несколько ошибок, причем разного характера, начиная с нарушения логики алгоритма до неадекватности реализации математической модели. Для локализации ошибок применяются разные приемы. Наиболее легко находятся ошибки, которые приводят к авостомам. Отыскать их помогают сообщения операционных систем. Значительно облегчают поиск ошибок специальные программные средства отладки, с которыми можно познакомиться в [7,14].

Для локализации более тонких ошибок применяют метод выключения программных блоков, суть которого заключается в организации обходов некоторых участков программы или использовании вместо части программных блоков "заглушек-имитаторов". Отладка в этом случае начинается с проверки схемы счета, которая постепенно наполняется блоками, проходящими отладку.

В заключение следует заметить, что после завершения отладки и тестирования программа должна быть полностью освобождена от отладочных вставок и выдач. При этом, вероятно, возникнут новые ошибки. Чтобы избежать этого, следует соблюдать особую тщательность при внесении исправлений в программу. Нельзя допускать, чтобы вносимые изменения имели характер "заплат" и были разбросаны по программе, т.к. в этом случае исчезает обзорность внесенного.

4.4. Выпуск программной документации

Разработка программных комплексов заканчивается выпуском программной документации и вводом их в эксплуатацию. Документация выполняет несколько функций :

- а) является обоснованием для проведения работ, связанных с автоматизированными вычислительными системами ;
- б) является отчетом о проделанной работе ;
- в) служит для сопровождения и эксплуатации готового программного комплекса.

Разработка документации должна вестись параллельно с разработкой программы. Оформление программной документации осуществляется в строгом соответствии с требованиями Единой системы программной документации (ЕСПД), объединяющей 29 стандартов.

Стандарты ЕСПД подразделяются на группы. К нулевой группе относятся ГОСТы, определяющие общие положения ЕСПД . Так, ГОСТ Р 9.001-77 "ЕСПД. Общие положения" устанавливает цели, назначение, область распространения, классификацию стандартов.

ГОСТ 19.003-80 "ЕСПД. Схемы алгоритмов и программ. Правила выполнения" содержит правила выполнения схем. Графические символы для построения схем устанавливает ГОСТ 19.002-80 "ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические". Основные термины, применяемые в программировании, установлены в ГОСТ 19.004-80 "ЕСПД. Термины и определения".

К стандартам первой группы относят :

- ГОСТ 19.101-77 "ЕСПД. Виды программ и программных документов";
- ГОСТ 19.102-77 "ЕСПД. Стадии разработки" (в котором устанавливаются стадии и этапы разработки программ и их содержание);
- ГОСТ 19.103-77 "ЕСПД. Обозначение программ и программных документов" (в котором определяется структура документов);
- ГОСТ 19.104-78 "ЕСПД. Основные надписи" (в котором дается форма, размеры, расположение и порядок основных надписей листа утверждения и титульного листа);
- ГОСТ 19.105-78 "ЕСПД. Общие требования к программным документам" ;
- ГОСТ 19.106-78 "ЕСПД. Правила выполнения программных документов, выполненных печатным способом".

Вторая группа объединяет ГОСТы, устанавливающие правила выполнения документации разработки. К ним относятся :

- ГОСТ 19.201-78 "ЕСПД. Техническое задание" ;
- ГОСТ 19.202-78 "ЕСПД. Спецификация".

Правила выполнения документации по методике и порядку проведения испытаний программ содержатся в стандартах третьей группы, к которой относится ГОСТ 19.301-79 "ЕСПД. Порядок и методика испытаний".

Четвертая группа стандартов включает правила оформления документации сопровождения :

- ГОСТ 19.401-78 "ЕСПД. Текст программы" ;
- ГОСТ 19.402-78 "ЕСПД. Описание программы" ;
- ГОСТ 19.403-79 "ЕСПД. Ведомость держателей подлинников" ;
- ГОСТ 19.404-79 "ЕСПД. Пояснительная записка".

К пятой группе стандартов относят правила выполнения эксплуатационных документов :

- ГОСТ 19.501-78 "ЕСПД. Формуляр" ;
- ГОСТ 19.502-78 "ЕСПД. Общее описание" ;
- ГОСТ 19.503-79 "ЕСПД. Руководство системного программиста";
- ГОСТ 19.504-79 "ЕСПД. Руководство программиста " ;

- ГОСТ 19.505-79 "ЕСПД. Руководство оператора" ;
- ГОСТ 19.506-79 "ЕСПД. Описание языка " ;
- ГОСТ 19.507-79 "ЕСПД. Ведомость эксплуатационных документов."

Правила обращения с программой регламентируются стандартами шестой группы :

- ГОСТ 19.601-78 "ЕСПД. Общие правила дублирования, учета и хранения " ;
- ГОСТ 19.602-78 "ЕСПД. Правила дублирования, учета и хранения документов, выполненных печатным способом " ;
- ГОСТ 19.603-78 "ЕСПД. Общие правила внесения изменений " ;
- ГОСТ 19.604-78 "ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом".

Как видно даже из заглавия стандартов, они определяют достаточно общие правила на документацию, но не ограничивают разработчика в выборе стиля изложения, содержания разделов, касающихся специфики задачи (ее постановки, технологии разработки алгоритмов решений и т.д.). В основном стандартизации подлежат административные документы (Спецификация; Ведомость держателей подлинников), документы, необходимые для сопровождения (Текст программы; Описание программы ; Порядок и методика испытаний), а также группа документов, необходимых для эксплуатации программы (Техническое задание ; Пояснительная записка ; Ведомость эксплуатационных документов ; Общее описание ; Руководство системного программиста ; Руководство программиста ; Руководство оператора; Описание языка ; Руководство по техническому обслуживанию).

Эксплуатационные документы нужны для пояснений пользователю возможностей программы и порядка ее использования. От качества этих документов будет зависеть, насколько быстро пользователь сможет освоить программу.

Знакомство с программой обычно начинается с "Описания программы". Оно состоит из четырех разделов : а) назначения программы; б) условий применения ; в) описания задачи; г) описания входных и выходных данных. В разделе "Назначение программы" сообщается назначение, основные характеристики и ограничения на область использования программы. Раздел "Условия применения" содержит информацию о необходимых для работы программы технических средствах, условиях организационного, технического и технологического характера. В разделе "Описание задачи" приводится математическая

формулировка задачи и метод ее решения.

В разделе "Входные и выходные данные" приводятся сведения о количестве, типе входных и выходных данных программы.

Таким образом, пользователь получает общие сведения о назначении и возможностях программы и может сделать вывод о том, удовлетворяет ли данная программа его требованиям. Дальнейшее знакомство с программой позволяет выяснить, как запустить программу на счет. В "Руководстве системного программиста" содержатся сведения о необходимых технических и программных средствах, обеспечивающих выполнение программы. Кроме этого, описывается структура программы, связи между ее составными частями и другими программами, даются рекомендации по выполнению настройки программы, проверки правильности ее работы. Для облегчения освоения программы пользователем приводятся тексты сообщений и контрольные примеры.

Следующим этапом освоения программы является подготовка её к решению конкретной прикладной задачи пользователя. Необходимые сведения и разъяснения даются в "Руководстве пользователю". В этом документе указаны :

- условия применения программы (необходимый объем оперативной памяти для размещения программы, требования к программному обеспечению и т.п.) ;
- порядок кодирования входных данных ;
- необходимые ресурсы (время работы и др.) ;
- структура выдаваемых текстов и представление результатов.

При эксплуатации крупных программных комплексов (ППК) часто возникает необходимость в действиях оператора ЭВМ, например для восстановления программы после прерываний, установки магнитных лент и дисков, на которых хранятся потребовавшиеся модули. Перечень сообщений, выдаваемых в ходе выполнения программы на пульт оператора, а также рекомендуемые действия содержатся в "Руководстве оператору".

Таким образом, эксплуатационные документы должны содержать исчерпывающую информацию для пользователя. Большинство сведений о структуре, характеристиках программы и порядке ее подготовки к работе могут дать только специалисты-разработчики. Поэтому большая и трудоемкая работа по выполнению программной документации возложена на программистов-разработчиков СПО.

Г Л А В А 5

РАЗРАБОТКА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ САПР ДЛЯ РАСЧЕТОВ КОНСТРУКЦИЙ НА ПРОЧНОСТЬ ПРИ ДИНАМИЧЕСКОМ НАГРУЖЕНИИ

5.1. Проблемы расчета конструкций на прочность в условиях динамического нагружения

Современная инженерная практика требует исследования неустановившихся процессов деформирования при воздействии на конструкции и их элементы ударных волн, вызванных взрывом, ударом, импульсным тепловым или лучевым воздействием.

В случаях, когда на элементы конструкций действуют факторы (массовые и поверхностные силы, температура), быстро изменяющиеся во времени, как показывает практика, нельзя ограничиться решением задачи о прочности конструкций в статической или квазистатической постановке. Главной особенностью импульсных нагружений является действие в конструкциях в течение короткого промежутка времени (секунды — доли микросекунд) напряжений, во много раз превышающих уровни напряжений, достигающиеся при статическом деформировании. При нагружении взрывом или высокоскоростным ударом в материалах напряжения могут достигать 100 МПа, а скорости деформации — 10^{+7} с⁻¹.

Импульсное приложение нагрузок вызывает распространение по конструкции ударных волн, которые взаимодействуют со свободными поверхностями и поверхностями раздела материалов. В результате в элементах конструкций вырабатываются нестационарные поля напряжений и деформаций, возникают очаги и области разрушения материала. Поэтому основной проблемой конструкторских расчетов на прочность при динамическом нагружении является определение эволюции напряженно-деформированного состояния во времени и оценка стойкости материала конструкции.

Поскольку напряженно-деформированное состояние в конструкции будет определяться взаимодействием волн напряжений (волн нагружения, волн разгрузки, отраженных волн), решение задачи о прочности будет сводиться к решению системы дифференциальных уравнений, включающей:

- а) уравнение сохранения массы;
- б) уравнение сохранения импульса;
- в) законы термодинамики (уравнение сохранения энергии и уравнение баланса энтропии);
- г) определяющие уравнения для описания свойств материала конструкции (уравнения упругопластической реакции сплошного материала и уравнения модели разрушения).

Условия нагружения и особенности геометрии конструкции учитываются начальными и граничными условиями [15].

Математическая формулировка задачи об импульсном нагружении твердых тел, как известно, за исключением некоторых частных случаев приводит к гиперболическим системам нелинейных дифференциальных уравнений. Для таких систем в настоящее время нет достаточно полной теории решения, нет общих теорем существования и единственности. Поэтому при решении практических задач очень остро стоит проблема развития и применения численных методов [13, 16, 17, 29 - 32].

Долгое время из-за ограниченной мощности ЭВМ численное решение динамических задач было возможно только в одномерной постановке либо в частных случаях, допускающих такие упрощения, которые позволяют провести расчет инженерными методами. В настоящее время появилась возможность решать более сложные задачи с математическим моделированием неустановившихся процессов деформирования в двумерной, а в некоторых случаях и в трехмерной постановке. Решение двумерных задач может быть проведено на ЭВМ ЕС 1060, ЕС 1045, БЭСМ - 6, МБК "Эльбрус".

При разработке пакета прикладных программ расчета динамического нагружения элементов конструкций следует, исходя из мощности, имеющейся в составе САПР ЭВМ, определить наиболее рациональную размерность постановки задачи. При этом необходимо иметь в виду, что для получения количественно достоверных результатов расчет должен проводиться с мелким пространственным и временным шагом. Время решения задач в многомерной постановке возрастает на порядки по сравнению с временем решения одномерной задачи.

Эффективность создаваемого программного комплекса во многом будет зависеть от решения круга вопросов, связанных с выбором математических моделей поведения материалов при высокоскоростном деформировании.

Реакция материала при импульсном нагружении носит упруго-вязкопластический характер, но универсальных математических моделей для ее описания не существует. Механическая реакция материала является макроскопическим проявлением разнообразных процессов, идущих на микроуровне и связанных с изменениями структуры материалов. В зависимости от условий деформирования преобладающими оказываются вклады в макроскопическую пластическую деформацию разных микромеханизмов. Поэтому модели упруговязкопластического поведения материалов имеют определенные границы применимости по скоростям деформации, температуре, материалам и т.д. В условиях импульсных нагружений скорость деформации и температура могут изменяться в широком диапазоне. Например, во фронте ударных волн интенсивностью в 10 ГПа скорость деформации изменяется от 10 до 10^6 с^{-1} [16-19]. Вследствие этого вопрос о выборе адекватной модели поведения материала стоит очень остро, поскольку от его решения зависит достоверность результатов работы всей САПР.

В ряде случаев конструкционные материалы деформируются под действием импульсных нагрузок без нарушения сплошности. В этих случаях с полным основанием можно использовать модели механики сплошных сред. Однако при достижении некоторых критических условий нагружения начинают развиваться процессы разрушения. На первых стадиях в материале возникают микропоры и микротрещины, а на последующих — происходит их рост и слияние в очаги макроразрушений и макротрещины. Специфика динамических нагружений состоит в том, что модели разрушения, в статических условиях обеспечивающие согласие с экспериментом, совершенно не пригодны.

В настоящее время ведутся работы по созданию динамических моделей разрушения, но пока они еще далеки от завершения. Поэтому при решении вопроса о разрушении конструкции разработчику САПР приходится проводить исследование адекватности модели и оценивать достоверность полученной в расчете информации.

Эффективность программного комплекса зависит также от метода численного решения задачи. В каждой прикладной задаче существуют свои особенности и разработчику программного обеспечения

САПР часто приходится сталкиваться с необходимостью либо разработать новый численный метод, либо адаптировать существующий метод к решению прикладной задачи. При этом всегда возникает проблема доказательства достоверности получаемых результатов.

Наиболее распространенным приемом доказательства является применение вычислительной схемы к решению задачи и исследование устойчивости, сходимости схемы в ходе численного эксперимента. Сравнение результатов численного решения с результатами, полученными с помощью других методов, а также с данными физического эксперимента хотя и не является строгим доказательством пригодности метода, но позволяет ее оценить. Следует заметить, что исследование метода осуществляется уже в процессе использования готового программного комплекса. Не исключена вероятность того, что выбранный метод окажется неприемлемым. В этом случае будет необходимо перестроить алгоритм, а возможно и саму структуру программы для использования другого численного метода.

Таким образом, при разработке программного комплекса для расчетов на прочность в условиях динамического нагружения должно быть решено несколько проблем:

- 1) наиболее ответственная и сложная проблема математической формулировки задачи, куда входят вопросы выбора модели поведения материала, и численного метода решения полученной системы уравнений;
- 2) проблема разработки и отладки алгоритма решения задачи;
- 3) проблема вычислительной эффективности программных комплексов, которая становится серьезной из-за больших затрат машинного времени и других ресурсов ЭВМ для решения динамических задач.

5.2. Математическая постановка динамических задач

Основной целью при решении задач динамического нагружения конструкций является определение напряженно-деформированного состояния в любой момент времени $t \in (t_1, t_2)$, пока не закончится процесс деформирования или разрушения. Эта информация может быть получена в ходе численного моделирования процесса динамического нагружения. Моделирование включает выполнение следующих операций:

- а) моделирование геометрии элементов конструкций в началь-

ний момент времени ;

б) моделирование взаимодействия элементов конструкции между собой и с окружающей средой ;

в) моделирование физических процессов деформирования материала конструкции [15, 19, 20] .

В ходе геометрического моделирования сложная конфигурация реальной конструкции путем упрощения сводится, с определенной степенью точности, к правильному геометрическому телу или фигуре. При этом определяется исходная область, занимаемая телом, свободные границы, границы раздела материалов в случае, если конструкция состоит из нескольких материалов с разными физико-механическими характеристиками .

В ходе моделирования условия взаимодействия элементов конструкций упрощаются либо идеализируются.

Результатом этого моделирования является :

а) определение граничных условий (связей с другими элементами конструкций и внешней средой) ;

б) определение в рассматриваемом интервале времени $t \in (t_1, t_2)$ процессов термомеханических воздействий.

Граничные условия могут быть механическими и термическими.

Механические граничные условия могут задаваться несколькими способами [15, 19, 20] :

1) путем задания на поверхности тела или ее части вектора перемещения или вектора скорости (кинематические Г У)

$$\text{либо } u_i = g_i \quad \text{на } S_n \quad (5.1)$$

$$\dot{u}_i = f_i \quad \text{на } S_n ; \quad (5.2)$$

2) путем задания на поверхности тела или ее части силовых факторов либо напряжений (силовые Г У)

$$\sigma_{ij} n_j = P_i \quad \text{на } S_n, \quad (5.3)$$

где n_j - вектор внешней нормали к поверхности S_n ;

σ_{ij} - напряжение ;

P_i - усилия ;

3) путем одновременного задания на поверхности тела или ее части и компонент вектора перемещений и компонент тензора напряжений (смешанные или иначе контактные Г У)

$$L_1(\bar{u}) + L_2(\sigma_{ij}) = \bar{F} \quad \text{на } S_n, \quad (5.4)$$

где L_1, L_2 - некоторые линейные алгебраические операторы ;

$\bar{F}$ - известная функция координат точек границы [20] .

Для моделирования физических процессов при высокоскоростном деформировании элементов конструкций необходимо решить ряд проблем, связанных с упрощением и идеализацией реальных процессов. Поскольку предметом моделирования является механическое поведение конструкции, для математической формулировки задачи используются подходы и методы механики сплошных сред.

Важным моментом, который предшествует записи системы уравнений, описывающей движение среды, является выбор независимых переменных. В основном в механике сплошных сред используются два подхода к описанию движения точек сжимаемых сред — Лагранжа и Эйлера [15-22,3]. В подходе Эйлера рассматривается движение среды относительно неподвижной системы координат с ковариантным базисом

χ_i . В этой системе координаты подвижных материальных точек переменны. Относя эти координаты к неподвижному базису $\bar{z}_i$, соответствующему начальному моменту времени, получим исходную лагранжеву систему координат. В лагранжевом подходе вводятся система материальных координат a^i ($i=1,2,3$).

Какой же способ описания среды удобнее выбрать при решении конкретной задачи? Обычно система координат выбирается таким образом, чтобы координатные поверхности совпадали хотя бы частично с поверхностью рассматриваемого тела. В этом случае упрощается постановка граничных условий (5.1) — (5.3). Базисные векторы начальной системы будем считать фиксированными во все моменты времени $t \in (t_1, t_2)$. Обозначим их через $\bar{z}_i = \partial/\partial a^i$. Если координаты относятся к сопутствующей лагранжевой системе координат, то они будут сохранять свои постоянные значения, а изменения положения точек будут определяться изменением векторов подвижного базиса $\bar{z}_i$ относительно неподвижной системы координат.

Параметры упругопластических течений в лагранжевой системе координат являются функциями времени и начальных координат, которые позволяют различать точки среды. Достоинства лагранжевых материальных координат обусловлены движением координатной системы вместе со средой. Использование лагранжевых координат позволяет:

- а) легко рассчитывать положение границ раздела сред и границ тела;
- б) сохранять точность аппроксимации в расчетах на всех этапах, т.к. количество узлов в рассматриваемой области остается неизменным;

в) получить качественно правильное решение со сравнительно небольшим количеством узлов [16].

Вместе с тем подход Лагранжа не применим, если в результате движения среды аппроксимирующая сетка претерпевает сильные искажения. В этом случае в расчетах приходится значительно уменьшать временной шаг, чтобы обеспечить устойчивость счета. При еще больших искажениях, которые могут приводить к "выворачиванию ячеек", необходима перестройка сетки. Процесс перестройки достаточно сложен и основан на методах интерполяции. В ходе частых перестроек сетки теряется точность расчетов.

При сильных искажениях деформируемых тел предпочтительным является подход Эйлера, который позволяет проводить расчеты без затруднений. Сложности применения подхода Эйлера возникают из-за необходимости для получения точного решения использовать значительно большее количество узлов. Кроме того, значительно сложнее оказывается определение подвижных границ.

Таким образом, при рассмотрении динамических нагружений, которые могут вызвать малые и умеренные (порядка нескольких процентов) деформации, целесообразно использовать подход Лагранжа. При решении задач с большими деформациями, особенно если существует возможность завихренных течений, рекомендуется применять подход Эйлера. В задачах со сравнительно небольшой областью больших деформаций целесообразно применять подход Лагранжа с последующей перестройкой сетки в области развитых деформаций.

Система уравнений, описывающих движение частиц тела в произвольный момент времени $t \in (t_1, t_2)$, будет иметь следующий вид [20, 21].

В переменных Лагранжа она включает:

- 1) соотношения, связывающие компоненты вектора перемещений u^i ($i = 1, 2, 3$) с тензором конечных деформаций Грина

$$\epsilon_{ij}^e = \frac{1}{2} \left(\frac{\partial u^i}{\partial x^j} + \frac{\partial u^j}{\partial x^i} + \frac{\partial u^k}{\partial x^i} \frac{\partial u^k}{\partial x^j} \right); \quad (5.5)$$

- 2) уравнение сохранения массы (уравнение неразрывности)

$$\rho_0 = \rho \det \left\| \delta_{ij}^0 + \frac{\partial u^i}{\partial x^j} \right\|; \quad (5.6)$$

- 3) уравнение движения

$$\frac{\partial \sigma_{ij}^0}{\partial x^j} + \rho_0 F_i = \rho_0 \frac{\partial^2 u^i}{\partial t^2}, \quad (5.7)$$

где σ_{ij}^0 - тензор напряжений Лагранжа.

Система (5.5) - (5.7) содержит неизвестные $u^i, \epsilon_{ij}^e, \rho$, в количестве $3 + 6 + 1 + 6 = 16$ (с учетом симметрии).

рии тензоров ϵ_{ij} и ϵ_{ij}^G).

В переменных Эйлера система уравнений включает :

- 1) соотношения связи между тензором скоростей деформации $\dot{\epsilon}_{ij}$ и компонентами вектора скорости частиц v_i ($i = 1, 2, 3$):

$$\dot{\epsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right); \quad (5.8)$$

- 2) уравнение неразрывности

$$\frac{d\rho}{dt} + \rho \operatorname{div} \vec{v} = 0; \quad (5.9)$$

- 3) уравнение движения

$$\frac{d t_{ij}}{dt} + \rho F_i = \rho \frac{d v_i}{dt}, \quad (5.10)$$

где t_{ij} - тензор напряжений Эйлера.

Неизвестными являются величины v_i , $\dot{\epsilon}_{ij}$, ρ , t_{ij} в количестве $3 + 6 + 1 + 6 = 16$.

Как видно из анализа, системы уравнений, описывающие общие закономерности движения сплошных сред, не замкнуты. Дополнение систем уравнений законами термодинамики (уравнением сохранения энергии и уравнением баланса энтропии) также не позволяет замкнуть систему. Это совершенно естественно, т.к. индивидуальные физико-механические свойства среды нигде не учтены. Эти свойства отражают определяющие уравнения, устанавливающие связь тензора напряжений с тензором деформаций и некоторыми термодинамическими параметрами. Совокупность определяющих уравнений составляет модель среды или иначе - модель поведения материала.

5.3. Выбор модели среды

Модель поведения материала при высокоскоростном деформировании в общем случае включает описание упруговязкопластической реакции материала как сплошной среды и описание процесса разрушения. Как уже говорилось раньше, универсальной модели поведения материалов при деформировании не существует. Поэтому применение тех или иных моделей обосновано лишь для определенного диапазона изменений условий деформирования (температуры, скорости деформации, амплитуды давлений и т.п.). Выбор модели поведения материала для рассматриваемых процессов в конкретных динамических задачах является едва ли не самым ответственным моментом. В настоящее время существует достаточно большое количество моделей, конкретизирующих определенные виды функциональной связи процессов нагружения и деформирования (рис.13). Любая модель

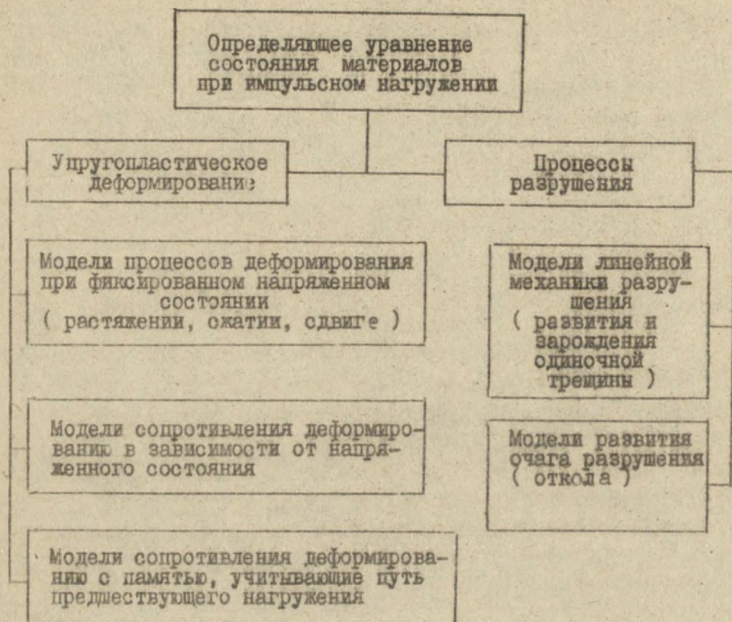


Рис.13. Структура определяющих уравнений, применявшихся для описания реакции конструкционных материалов при импульсном нагружении

предполагает определенную идеализацию реального поведения материала.

Хорошо известно, что уровень действующих в материале сдвиговых напряжений зависит от скорости деформации и температуры. Для динамического нагружения характерны высокие скорости деформации во фронте волн напряжений. Поэтому модели деформационной теории пластичности, строящиеся без учета скоростных факторов, оказываются в условиях высокоскоростного деформирования неадекватными. Их применение в расчетах дает большое расхождение с экспериментом, причем не только количественное, но и качественное [18]. Например, не удается описать затухание упругих предвестников и нестационарность фронтов при распространении волн напряжений интенсивностью в $1 - 5 \text{ ГПа}$ [18, 33, 34, 36].

Применение моделей пластического течения, учитывающих временной фактор, также не всегда обеспечивает правильное описание явлений волновой динамики. Поэтому разработчик программного комплекса обычно сталкивается с проблемой выбора модели, наиболее близко отражающей упруговязкопластическое поведение материала в конкретных условиях решаемой задачи. Для того чтобы решить её, следует четко определить :

- 1) какие физические факторы и процессы являются наиболее важными в конкретной задаче ;
- 2) в какой постановке (одномерной или многомерной) должна решаться задача.

Анализ результатов экспериментальных и теоретических исследований физических процессов и факторов в сходных условиях нагружения помогает оценить ожидаемые интенсивности и температурный режим воздействий, характер волн нагружения, а также сделать предположения относительно диапазона изменения скорости деформаций, необходимости учета релаксационных явлений.

Вопрос о размерности решаемой задачи возникает вторично. Первоначально выбор размерности постановки задачи определялся желанием учесть особенности деформирования конструкций, связанные с пространственными эффектами (например, с затуханием сферических ударных волн, пространственным взаимодействием элементов конструкций и т.п.). При выборе модели среди этот вопрос должен решаться с учетом ожидаемого соответствия достоверности результатов расчета затратам на решение задачи в многомерной постановке.

На основе принятых предположений выбирается совокупность переменных, определяющих состояние среды, и тип модели. Затем путем последовательного анализа существующих моделей данного типа отыскивается наиболее удобный вид определяющих уравнений.

При описании высокоскоростного деформирования моделями пластического течения тензор скорости деформации $\dot{\epsilon}_{ij}$ представляется в виде суммы тензоров скоростей упругих $\dot{\epsilon}_{ij}^e$ и пластических $\dot{\epsilon}_{ij}^p$ деформаций:

$$\dot{\epsilon}_{ij} = \dot{\epsilon}_{ij}^e + \dot{\epsilon}_{ij}^p. \quad (5.11)$$

Постулируется пластическая несжимаемость материала (5.12), откуда следует (5.13):

$$\dot{\epsilon}_{kk} = \dot{\epsilon}_{kk}^e \quad \text{или} \quad \dot{\epsilon}_{kk}^p = 0, \quad (5.12)$$

$$\epsilon_{kk} = \epsilon_{kk}^e \quad \text{или} \quad \epsilon_{kk}^p = 0. \quad (5.13)$$

Обычно при решении задач об импульсном нагружении твердых тел тензор напряжений разбивают на шаровую и девиаторную части:

$$\sigma_{ij} = -p\delta_{ij} + s_{ij}, \quad (5.14)$$

где $-p = \frac{1}{3}(\sigma_{kk})$ - гидростатическое давление;

$s_{ij} = \sigma_{ij} - \frac{1}{3}(\sigma_{kk})\delta_{ij}$ девиатор напряжения.

Аналогично представляются тензоры деформаций и скоростей деформаций:

$$\dot{\epsilon}_{ij} = \theta\delta_{ij} + \dot{e}_{ij} = \frac{1}{3}(\dot{\epsilon}_{kk})\delta_{ij} + \dot{e}_{ij}, \quad (5.15)$$

где θ - объемная деформация, которая с учетом (5.13) будет представлять чисто упругую деформацию;

$\dot{e}_{ij}$ - девиатор деформации;

$$\dot{\epsilon}_{ij} = \frac{1}{3}(\dot{\epsilon}_{kk})\delta_{ij} + \dot{e}_{ij} = \theta\delta_{ij} + \dot{e}_{ij}. \quad (5.16)$$

Напряжения, действующие в изотропном теле, связаны с упругой составляющей деформации законом Гука:

$$\dot{s}_{ij} = 2\mu(\dot{e}_{ij}), \quad (5.17)$$

$$-p = \kappa\theta, \quad (5.18)$$

где μ - модуль сдвига;

κ - модуль объемного сжатия.

Упругая часть тензора скоростей деформации может быть найдена из (5.11) по известным скоростям полной деформации $\dot{\epsilon}_{ij}$ и скорости пластической деформации $\dot{\epsilon}_{ij}^p$. Скорость деформации будет найдена из (5.8), а вот скорость пластической деформации

*
 $\dot{\epsilon}_i$ необходимо определить, например, с помощью соотношений модели пластического течения (Прил. I).

Модели пластического течения предполагают, что напряжения в каждый момент времени вырабатываются в результате двух конкурирующих процессов. За счет роста полной деформации тела под действием внешних нагрузок напряжения увеличиваются, а вследствие развития пластических сдвигов - релаксируют. Понятие "динамическое нагружение" связано с такими режимами, при которых характерные времена нагружения ($\tau_n = 1/\dot{\epsilon}_u$) сравнимы или меньше характерных времен развития пластических сдвигов ($\tau_r = 1/\dot{\epsilon}_u^p$).

Характерные времена нагружения τ_n могут изменяться в широком диапазоне в зависимости от условий приложения нагрузки, а характерные времена релаксации определяются для данного материала скоростью превращений микроструктуры.

Исследования показали, что для большинства конструкционных материалов можно выделить четыре характерные области зависимости напряжения течения от скорости деформации и температуры (рис. I4.) [23-25]. Существование этих областей связывают с преобладанием вкладов в пластическое течение определенных микромеханизмов [23,24].

Область I характеризуется нечувствительностью напряжения течения к скорости деформации и температуре. Пластическое течение в этой области предположительно вызывается движением краевых дислокаций. Предел текучести определяется дальнедействующими полями внутренних напряжений, связанных с наличием скоплений дислокаций, границ зерен, микровключений и других дефектов структуры.

В области II напряжение течения заметно изменяется при смене скорости и температуры. Пластическое течение в основном контролируется термоактивируемым движением дислокаций. Энергия активации зависит от приложенного напряжения и температуры.

Для области III характерна слабая зависимость напряжения течения от скорости и температуры. Механизмы пластического деформирования в ней изучены слабо.

Область IV характеризуется очень высокими скоростями деформаций - от 10^3 до 10^7 с⁻¹. Напряжение течения чрезвычайно чувствительно к изменению скорости деформации. Пластическое течение связано с движением дислокаций.

Таким образом, феноменологические модели течения, построенные путем обобщения данных макроскопического эксперимента, могут

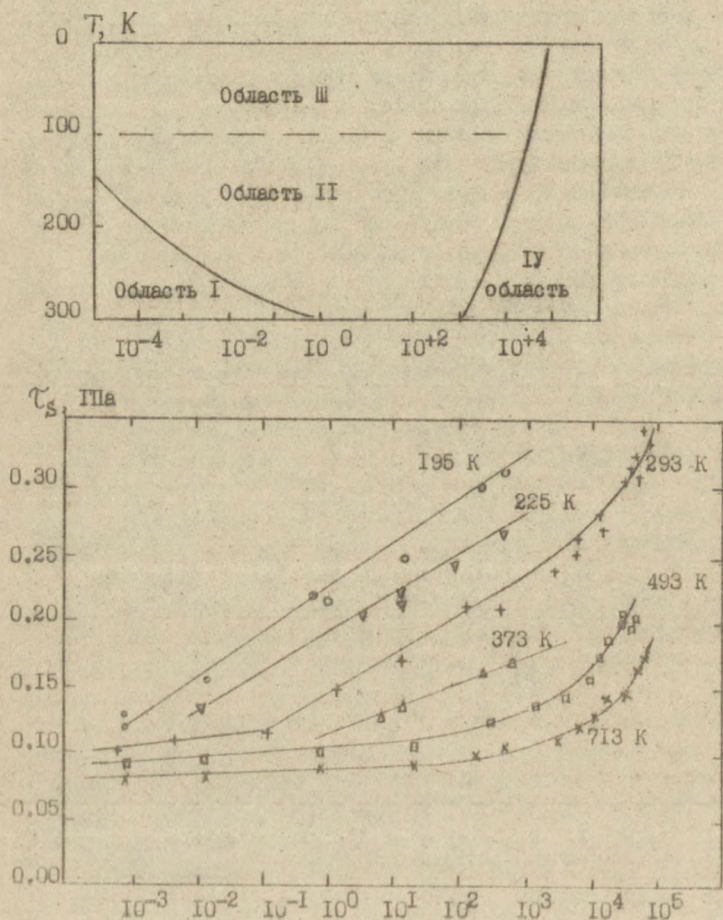


Рис. 14. Зависимость предела текучести мягкой стали от скорости деформации и температуры по данным работы [23]

обоснованно применяться в интервале ($\dot{\epsilon}$ и T°) пока не произойдет смена физических механизмов пластичности.

В условиях импульсного нагружения скорость деформации изменяется в диапазоне, где смена физических механизмов происходит неоднократно. Поэтому в расчетах, выполненных с использованием феноменологических моделей течения, получают большое количественное расхождение с экспериментом.

Попытка учесть особенности превращений микроструктуры в процессе деформирования делается в моделях пластического течения с внутренними параметрами [18,19,21]. Модели этого класса развиваются по двум направлениям.

В микродинамических моделях скорость сдвиговой пластической деформации связывается с изменением усредненных параметров дефектной структуры (плотности дислокаций, средней скорости их движения и т.д.). Сложность использования этих моделей вызвана необходимостью определения достаточно большого количества параметров модели, входящих в кинетические соотношения, которые описывают изменения в микроструктуре. По этой же причине микродинамические модели применяются только в одномерных вариантах.

Использование микродинамических моделей в расчетах показало, что они способны адекватно описывать течение в очень широком диапазоне скоростей деформации — от 10^{-3} до 10^6 с^{-1} [18,19,21].

Реологические модели течения с внутренними параметрами строятся исходя из общих термодинамических принципов и дополнительных постулатов. В подавляющем большинстве моделей этого класса предполагается, что тензор скоростей деформации может быть представлен

$$\dot{\epsilon}_{ij} = \dot{\epsilon}_{ij}^e + \dot{\epsilon}_{ij}^p + \dot{\epsilon}_{ij}^v, \quad (5.19)$$

где $\dot{\epsilon}_{ij}^e$ — скорость упругих деформаций;
 $\dot{\epsilon}_{ij}^p$ — скорость необратимых пластических деформаций;
 $\dot{\epsilon}_{ij}^v$ — скорость вязких деформаций.

Наличие вязких свойств приводит к существенной зависимости механического поведения от времени, а появление пластических деформаций связывается с остаточными изменениями формы и структуры тел. Сложность использования таких моделей связана с необходимостью определения для конкретных материалов $\dot{\epsilon}_{ij}^v$ как функции от внутренних параметров состояния. Подробнее познакомиться с построением реологических моделей с внутренними параметрами можно в [16,21,23].

Наиболее развитыми моделями упруговязкопластического поведения материалов являются одномерные [19]. Переход к моделям для расчетов течений в двух- и трехмерной постановке очень часто осуществляется путем эквивалентных записей определяющих уравнений в интенсивностях напряжений и деформаций [19,21].

Интенсивности деформаций ϵ_u , $\dot{\epsilon}_u$, ϵ_u^e и напряжений σ_u , $\dot{\sigma}_u$, а также их скоростей $\dot{\epsilon}_u$, $\dot{\epsilon}_u^e$, $\dot{\sigma}_u$, $\dot{\sigma}_u^e$ вводятся как скалярные функции вторых инвариантов соответствующих тензоров:

$$\begin{aligned}\epsilon_u &= \frac{\sqrt{2}}{3} \sqrt{3\epsilon_{ij}\epsilon^{ij} - (\epsilon_{kk})^2}, \\ \dot{\epsilon}_u &= \frac{\sqrt{2}}{3} \sqrt{3\dot{\epsilon}_{ij}\dot{\epsilon}^{ij} - (\dot{\epsilon}_{kk})^2}, \\ \sigma_u &= \frac{1}{\sqrt{2}} \sqrt{3\sigma_{ij}\sigma^{ij} - (\sigma_{kk})^2},\end{aligned}\quad (5.20)$$

и т.д.

Модели пластического течения для сложного напряженно-деформированного состояния в настоящее время находятся в стадии разработки.

Для большинства практических задач мощность современных ЭВМ оказывается недостаточной, чтобы решать эти задачи с мелким пространственным разбиением в двумерной и трехмерной постановке.

Применяя грубое разбиение (с интервалами от 1 – 5 мм до 1 – 2 см) не удастся правильно определить истинные скорости деформации во фронте волн напряжений. Таким образом, даже при использовании адекватных релаксационных моделей в расчете будут получены значения сдвиговых напряжений, далекие от реальных. В свою очередь, ошибки в определении сдвиговых напряжений повлекут неадекватное описание эволюции напряженно-деформированного состояния конструкции. Поэтому часто используют усредненное по процессу значение напряжений течения – "динамический предел текучести". Динамический предел текучести определяют по остаточным деформациям цилиндра Тейлора [17] и по амплитуде отрелаксировавшего упругого предвостика σ_{sa} [18]:

$$\sigma_{sd} = \frac{1-2\nu}{1-\nu} \sigma_{sa}, \quad (5.21)$$

где ν – коэффициент Пуассона.

Расчеты в этом случае проводятся в рамках жесткопластической модели. Пока сдвиговые напряжения не достигнут значения динамического предела текучести, реакция тела считается чисто упругой. Связь тензоров напряжений и деформаций выражается

уравнениями (5.17) и (5.18). После достижения сдвиговыми напряжениями значения динамического предела текучести реакция считается упругопластической, причем напряжение течения принимается постоянным. В качестве критерия начала пластического деформирования обычно используется условие Мизеса

$$\sigma_{ij} \dot{\epsilon}_{ij} = 2 \sigma_{sd}^2 \dot{\epsilon} \quad (5.22)$$

Изложенный выше подход часто применяется при решении многомерных задач динамики твердых тел. Подбирая значение динамического предела текучести, удается получать согласие с экспериментом по интегральным характеристикам. Расчетные значения локальных характеристик (напряжений, деформаций, скоростей деформаций) могут быть использованы только для изучения качественной картины распределения напряжений и деформаций.

Многочисленные экспериментальные исследования свидетельствуют о том, что уравнение (5.18), связывающее среднее нормальное напряжение с объемными деформациями, выполняется при напряжениях, не превышающих 1 ГПа [15, 19]. При более высоких напряжениях связь становится нелинейной и уравнение (5.18) не применимо.

При импульсном нагружении твердых тел возникают высокие напряжения (до 100 ГПа), поэтому в расчетах вместо (5.18) используются нелинейные уравнения состояния вида $p = p(\rho, E, T)$.

Экспериментальное построение уравнения состояния – задача очень сложная. В настоящее время универсального широкодиапазонного уравнения состояния не существует [15]. Поэтому при его выборе следует учитывать диапазон применимости (Прил. 2).

Для давлений до 50 ГПа удобно применять баротропные уравнения состояния $p = p(\rho)$ (см. Прил. 2). В этом случае в системы (5.5)–(5.7) или (5.8)–(5.10) не вводятся дополнительные неизвестные и модель пластического течения позволяет замкнуть общую систему уравнений.

Для амплитуд ударных волн, выше 10–20 ГПа, необходимо использовать уравнения состояния вида $p = p(\rho, E, T)$ (см. Прил. 2). Для замыкания общей системы уравнений в этом случае потребуются дополнительно два уравнения. В качестве их могут быть использованы законы сохранения энергии и баланса энтропии (законы термодинамики) [35].

5.4. Выбор модели разрушения

Одной из главных задач при проектировании конструкций является обеспечение их прочности. В конечном итоге проектировщика интересует вопрос: выдержит ли конструкция нагрузки или произойдет ее разрушение? При этом часто важно также знать, где и как произойдет разрушение. Ответы на эти вопросы дает расчет конструкции на прочность. Система уравнений (5.6-5.7) и определяющие уравнения (5.11)-(5.22) составляют механико-математическую модель реакции сплошного материала, в которой возможность появления трещин и повреждений другого тела не предусматривается. Таким образом, для того чтобы ответить на вопрос, разрушится или не разрушится конструкция, решаемая система уравнений должна включать совокупность соотношений, составляющих модель разрушения.

При разработке прикладного программного обеспечения САПР динамических систем вопрос о выборе модели разрушения встает очень остро. Ситуация сходна с той, которая возникала при выборе модели упруговязкопластического поведения материала. В современной научной литературе предлагается большое число разнообразных моделей и критериев разрушения. Универсальных моделей нет, а оценить пригодность предлагаемых вариантов для решения конкретной задачи сложно.

Наиболее развитыми являются модели разрушения, ориентированные на статические режимы нагружения. Эти модели, как показывает практика, в условиях динамического нагружения применяться не могут. Неадекватность моделей вызвана двумя причинами.

Во-первых, при импульсном нагружении поля напряжений и деформаций существенно нестационарны. Растягивающие напряжения, которые вызывают разрушение, возникают в той области материала, где происходит встреча волн разгрузки. Время действия и амплитуда этих напряжений будут связаны с условиями нагружения (формой, амплитудой, длительностью импульса), геометрией тела и физико-механическими свойствами материала. Таким образом, на разрушение будет влиять не только уровень действующих в теле напряжений, но и время их действия.

Во-вторых, макроскопическое разрушение, т.е. разделение ранее сплошного тела на части, является результатом совместного протекания многих микропроцессов (зарождения микроповреждений,

их слияния в макротрещину, распространения магистральной трещины по материалу). Все эти микропроцессы имеют ограниченные скорости, которые зависят от условий нагружения (температуры, скорости деформации, реализующегося напряженно-деформированного состояния). Появление микротрещин и их раскрытие приводит к уменьшению несущей поверхности и изменению напряжений, действующих в теле. Следовательно, временной фактор является существенным для правильного описания разрушения в условиях импульсных нагружений.

Учесть временной характер прочности материалов можно различными способами. Простейший путь заключается в том, что константы статических теорий прочности считаются функциями времени, которые подбираются экспериментально. Практика использования подобных моделей показала, что более или менее достоверные результаты удается получить в очень ограниченном диапазоне условий нагружения, для которого удачно подобраны функции.

Разрушение в условиях импульсного нагружения имеет ряд особенностей, которые не могут реализовываться в статике. Связано их появление с высокими давлениями и высокими градиентами напряжений во фронте ударных волн. Так, например, действие высокого давления приводит к "залечиванию" возникших ранее микрповреждений.

Часто в инженерных расчетах применяется критерий локального разрушения, использующий понятие долговечности. Под долговечностью понимается время до разрушения при фиксированном напряжении и температуре. Зависимость долговечности от температуры и действующего сдвигового напряжения τ может быть задана в виде формулы, предложенной С.Н. Мурковым [17, 21, 22]:

$$t_p = t_0 \cdot \exp\left(\frac{U_0 - \gamma \tau}{kT}\right), \quad (5.23)$$

где t_0 - период тепловых колебаний атомов;
 k - постоянная Больцмана;
 U_0 - энергия, приблизительно равная для металлов энергии сублимации;
 γ - структурный параметр.

В моделях длительной прочности предполагается, что разрушение непрерывно и необратимо. В течение времени Δt при неизменном в этом интервале времени напряжении происходит относительное уменьшение долговечности на величину, равную $\Delta t/t_p(t)$. Полное разрушение произойдет при условии

$$\int_0^t \frac{dt}{t_p(\tau)} = 1, \quad (5.24)$$

где t^* - время разрушения.

Серьезным недостатком критерия долговечности является то, что он не учитывает изменения напряженно-деформированного состояния в теле, вызванные образованием микродефектов структуры (микропор, микротрещин, скоплений дислокаций и т.п.). Многочисленные экспериментальные исследования показывают, что разрушению при импульсном нагружении предшествует появление в материале многочисленных повреждений структуры [17, 21, 22, 25]. Поскольку с ростом количества микроповреждений структуры увеличивается свободная от напряжений поверхность, макроскопические напряжения изменяются.

Релаксация напряжений вызывается двумя разными по своей физической природе процессами. Во-первых, из-за раскрытия образовавшихся в теле трещин происходит упругая релаксация объемных деформаций и соответствующее уменьшение растягивающих напряжений. Во-вторых, из-за уменьшения площади несущей поверхности изменяется напряжение, действующее в сплошном материале.

Для учета этих явлений проводят коррекцию напряжений с помощью поправочного коэффициента ω - параметра поврежденности:

$$\bar{\sigma}_{ij}^0 = \bar{\sigma}_{ij} / (1 - \omega), \quad (5.25)$$

где $\bar{\sigma}_{ij}^0$ - тензор напряжений в материальной матрице;

$\bar{\sigma}_{ij}$ - тензор напряжений в континуальной среде (среднее напряжение);

ω - скалярный параметр поврежденности среды, изменяющийся в диапазоне $0 \leq \omega \leq 1$

($\omega = 0$ - соответствует сплошному материалу,

$\omega = 1$ - полностью разрушенному).

В качестве ω принимают степень уменьшения эффективной площади сечения образца [24], относительный объем пустот [17, 21].

С появлением повреждений (пор, микротрещин) сплошной материал становится неоднородным. Такую среду можно представить как смесь двух фаз, на межфазных границах которых макроскопические характеристики терпят разрыв. Для описания движения таких сред необходимо применять математический аппарат, разработанный для многофазных сред [26].

В качестве первой фазы, занимающей часть общего объема V , принимается сплошной материал конструкции, а под второй фазой,

занимающей объем V_n , понимается суммарный объем микроповреждений.

Введем параметр поврежденности среды в виде

$$\omega = \frac{V_n}{V} \quad (5.26)$$

Средняя плотность поврежденной среды может быть найдена

$$\rho = \rho_0 (1 - \omega) + \rho_n \omega \approx \rho_0 (1 - \omega), \quad (5.27)$$

где ρ_0 - плотность сплошного материала ;

ρ_n - плотность газа, заполняющего полости повреждений.

Осредненные значения напряжений $\bar{\sigma}_{ij}$ в поврежденной среде могут быть представлены :

$$\bar{\sigma}_{ij} = \bar{\sigma}_{ij}^0 (1 - F(\omega)) + \bar{\sigma}_{ij}^n \Phi(\omega), \quad (5.28)$$

где $\bar{\sigma}_{ij}^0$ - напряжения в сплошном материале ;

$\bar{\sigma}_{ij}^n$ - напряжения в материале полостей ;

$F(\omega)$ и $\Phi(\omega)$ - функции формы микроповреждений.

Учитывая, что $\bar{\sigma}_{ij}^n \ll \bar{\sigma}_{ij}^0$, т.к. полости заполнены газом, будем иметь (5.25) как частный случай (5.28).

Уменьшение средних значений модулей упругости в повреждаемой среде учитывается аналогично корректировке напряжений :

$$M = M^0 (1 - F(\omega)), \quad (5.29)$$

где M^0 - модуль сдвига в сплошном материале .

При возникновении микроповреждений и их раскрытии, как уже было сказано, происходит упругая релаксация объемных деформаций. В моделях сред с накоплением повреждений это обстоятельство может быть учтено путем введения скорости неупругих объемных деформаций

$$\frac{d\epsilon_{ij}^H}{dt} = \frac{d\omega}{dt} \delta_{ij}. \quad (5.30)$$

Эти деформации должны учитываться в определяющих уравнениях. Тогда, в отличие от моделей пластичности, в тензоре неупругих деформаций будет присутствовать не только девиаторная часть

$$\epsilon_{ij}^H, \text{ но и шаровая } \frac{1}{3} \frac{d\epsilon_{kk}^H}{dt} \delta_{ij}.$$

В моделях сред с повреждениями деформация, связанная с появлением микрон неоднородностей, не является постоянной, т.к. их количество растёт.

Параметр поврежденности среды ω , входящий в (5.30), изменяется во времени. Его изменение учитывает результат протекания нескольких процессов :

а) зарождения микроповреждений ;

б) роста геометрических размеров дефектов ;

в) слияния и дробления микроповреждений .

Построение уравнений, описывающих изменение ω , ведется по двум направлениям. В первом развиваются микростатистические модели. В основе этих моделей лежат кинетические уравнения, построенные на анализе реальных физических механизмов развития микроповреждений конкретного типа. Интегрирование кинетических уравнений позволяет найти в каждый момент времени суммарный объем повреждений. В большинстве моделей предполагается, что зарождение дефектов начинается при превышении некоторого порогового критерия, после чего возникают дефекты с характерной скоростью и размерами.

Пороговый критерий обычно состоит из двух условий : энергетического и механического. Энергетическое условие требует, чтобы свободная энергия в материальной точке убывала, а механическое – чтобы уровень действующих напряжений превышал прочность связи.

Применение конкретного условия зависит от того, какой характер носят рассматриваемые дефекты. Для микропор вводится энергетический критерий, а для микротрещин – механический. На практике выбрать тип критерия помогают экспериментальные исследования внутренней структуры образцов после ударного нагружения [17,18,21,22,25] .

Общий вид пороговых условий :

$$f(\sigma_u, \epsilon_u^p, T^0, R) \geq 0. \quad (5.31)$$

где σ_u – интенсивность напряжений в континуальной среде;

ϵ_u^p – интенсивность эквивалентной пластической деформации континуальной среды;

T^0 – температура;

R – размер неоднородности в месте зарождения дефекта.

Конкретный вид условия (5.31) зависит от выбранной модели.

При выполнении (5.31) подсчитывается число дефектов, зародившихся в единице объема. Скорость зарождения в общем случае задается выражением

$$N^0 \equiv \left(\frac{\partial N}{\partial t} \right)_x = A(\sigma_u, T) + B(\sigma_u) \dot{\sigma}_u + C(\epsilon_u^p) \dot{\epsilon}_u^p, \quad (5.32)$$

где A, B, C – функции указанных аргументов, которые отражают соответственно процессы термической и вызванной напряжениями диффузии, зарождения микроповреждений вслед-

ствие механического нарушения связей.

Число дефектов вычисляется в лагранжевой точке. Закон роста геометрических размеров зародившихся дефектов записывается в виде

$$\dot{R} = f(\bar{\sigma}_u, R), \quad (5.33)$$

где $f(\bar{\sigma}_u, R)$ — некоторая функция, выбор которой осуществляется на основе предположений модели. Например,

$$f(\bar{\sigma}_u, R) = (\bar{\sigma}_u - \bar{\sigma}_{go}) R / 4\eta \quad \text{при } \bar{\sigma}_u > \bar{\sigma}_{go}$$

(η — параметр, имеющий размерность вязкости) [17, 21].

Совокупность соотношений (5.32, 5.33) кинетики роста и зарождения микроповреждений позволяет определить суммарный объем дефектов каждого рассматриваемого типа.

Для ответа на вопрос о возникновении макроразрушения необходимо знать, как будет происходить процесс образования макротрещины. К сожалению, удовлетворительной модели слияния микроповреждений не существует [21, 25]. Поэтому целесообразно пользоваться интегральными критериями разрушения, хотя определенные трудности возникают и в этом случае.

Многочисленные экспериментальные исследования свидетельствуют, что относительный объем микроповреждений, при котором возникает макроскопическое разрушение, для одного и того же материала не одинаков. Если этот объем выбран в качестве меры поврежденности, то разрушение наступит не при $\omega = 1$, а при значительно меньших значениях ω . Так, в экспериментах фиксировалось разрушение при $\omega = 0.05 + 0.2$ в зависимости от величины и времени действия растягивающих напряжений [17, 21]. Таким образом, выводы о разрушении конструкции, сделанные на основании полученных в расчете значений параметра поврежденности среды, могут оказаться неверными.

Часто в расчетах используется критерий типа

$$\int_0^{t^*} \bar{\sigma}_{ij} \frac{d\varepsilon_{ij}}{dt} \delta_{ij} dt \leq I_* \quad (5.34)$$

или его аналог, полученный с учетом (5.30),

$$\int_0^{t^*} \bar{\sigma}_{kk} \dot{\omega} dt \leq I_*. \quad (5.35)$$

Критерий (5.34) или (5.35) предполагает, что разрушение произойдет, когда удельная работа напряжений на объемных неупругих деформациях достигнет критической величины I_* [17].

Микростатистические модели достаточно полно отражают особенности развития очагов разрушения, но включают большое количество функций и параметров модели, которые для конкретного материала

необходимо отыскивать по экспериментальным данным. Систематические экспериментальные исследования микроструктуры после приложения импульсных нагрузок представляют редкость, поэтому при подборе параметров модели возникают сложности. Вследствие этого в инженерных задачах микростатистические модели пока применяются ограниченно [17,25].

Другой подход к построению моделей повреждаемой среды использует данные экспериментов, в которых изучаются механические свойства материалов при высокоскоростном деформировании. Вид кинетического уравнения, описывающего изменение параметра поврежденности среды $F(\sigma_1, \sigma_0, T, \omega, \dot{\omega}, \chi_k) = 0$, задается из соображений общего характера. Численные значения коэффициентов подбираются таким образом, чтобы характер разрушения в расчете совпадал с экспериментально наблюдаемым.

Например, в [17,25] предлагается кинетическое уравнение вида

$$\dot{\omega} = \frac{(\sigma_1^0 - \sigma^*)}{t_R \sigma^*} (1 - \omega) \quad \text{при } \sigma_1^0 > \sigma^* \text{ и } \omega < \omega^*, \quad (5.36)$$

$\dot{\omega} = 0$, если условия не выполняются,

где $\sigma^* = \sigma_0 (1 - (\dot{\omega}^*)^{1/2})$;

t_R - время релаксации;

σ^* - некоторый пороговый уровень напряжений, зависящий от формы повреждений;

σ_1^0 - растягивающее напряжение.

В работе [21] предлагается использовать уравнение вида

$$\dot{\omega} = A \left(\frac{\sigma_1^0}{1 - \omega} \right)^K \quad \text{при } \sigma_1^0 > \sigma^*, \quad (5.37)$$

где A и K - постоянные материала.

Как правило, подобные модели содержат значительно меньшее, по сравнению с микростатистическими, количество неизвестных коэффициентов.

При реализации моделей повреждаемой среды в программных комплексах необходимо помнить, что параметры поврежденности должны сохраняться для каждой лагранжевой координаты тела, а решение кинетических уравнений роста микроповреждений должно осуществляться для каждого узла на каждом временном слое.

Большинство моделей повреждаемой среды создано для решения одномерных задач. Переход на случай сложного нагружения осуществляется путем введения некоторого эквивалентного напряжения - скалярной функции второго инварианта тензора напряжений, аналогич-

но введению интенсивности тензора напряжений [21,24] .

Кроме моделей длительной прочности и моделей повреждаемых сред, при решении динамических задач в двумерной постановке, применяются модели распространения и остановки трещин [21,27] . Применение моделей линейной механики разрушения предпочтительнее, если конструкция содержит концентраторы в виде надрезов.

Реализация моделей в программных комплексах отличается сложностью, т.к. возникает ряд проблем, связанных с необходимостью разрыва расчетной сетки, обхода особенности в вершине трещины, расчета взаимодействия волн напряжений с краями трещины. Обзор работ, посвященных развитию подхода линейной механики разрушения к динамическим условиям нагружения, можно найти в [27] .

Г Л А В А 6

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ РАСЧЕТОВ РАСПРОСТРАНЕНИЯ ПЛОСКИХ ВОЛН НАПРЯЖЕНИЙ В ТВЕРДЫХ ТЕЛАХ

6.1. Выработка требований к программному комплексу

Рассмотрим последовательность разработки модулей прикладных программ для решения динамических задач на примере создания программного комплекса для расчетов распространения плоских волн напряжений в твердых телах.

Целью создания программного комплекса является определение структуры фронтов волн напряжения и исследование эволюции этой структуры по мере распространения волны.

Подобные вопросы возникают, например, при автоматизации обработки результатов экспериментов с плоскими волнами, а также при решении технологических задач, связанных с ударным либо взрывным упрочнением металлов и сплавов.

Основные функциональные требования пользователей свелись к следующему :

- 1) программный комплекс должен обеспечивать численное моделирование плоского соударения пластин в диапазоне скоростей от 100 до 500 м/с ;
- 2) расчеты должны обеспечивать правильное описание нестационарных явлений волновой динамики (затухания упругих предвестников, нестационарность фронтов слабых ударных волн).

Помимо функциональных, был выдвинут ряд эксплуатационных требований :

- 1) программа должна быть составлена на Фортране и исполняться как на ЭВМ БЭСМ - 6 , так и ЭВМ Единой Серии;
- 2) объем доступной оперативной памяти ограничен 500 кбайтами;

- 3) комплект оборудования для работы с программным комплексом может включать устройства прямого доступа ;
- 4) время счета варианта с распространением волны на 1 см не должно превышать 1 ч на ЭВМ БЭСМ - 6 .

Анализ требований показывает, что на разработчика программы возложено решение ряда вопросов, связанных с математической постановкой задачи, выбором модели, выбором схемы счета, разработкой алгоритма программы.

Целью создания программы является численное моделирование процессов распространения ударных волн при плоском соударении пластин. Схема соударения показана на рис.15. Поскольку толщина пластин значительно меньше двух других размеров, волны боковой разгрузки не успевают в рассматриваемом интервале времени повлиять на структуру фронта плоской ударной волны.

В момент удара на контактной поверхности соударяющихся пластин возникает разрыв, который распадается на две ударные волны, распространяющиеся в глубь материала метаемой и неподвижной пластин. Если толщина метаемой пластины меньше толщины неподвижной, то волна, распространяющаяся в бойке, успеет отразиться от свободной поверхности в виде волны разрежения и начнет догонять фронт ударной волны в неподвижной пластине. Поскольку волна разрежения распространяется по сжатому материалу, скорость ее будет превосходить скорость ударной волны. При достаточно толстой неподвижной пластине волна разрежения успеет догнать фронт ударной волны. В результате взаимодействия этих волн амплитуда волны нагружения будет затухать.

В рассматриваемом диапазоне скоростей фронты ударных волн имеют сложную структуру. Они состоят из упругого предвестника, распространяющегося с продольной скоростью звука, и волны объемного сжатия (пластической волны), скорость которой зависит от амплитуды ударной волны. При амплитудах меньших 10 ГПа для большинства металлов и сплавов скорость пластической волны будет меньше скорости звука. Таким образом, ширина ударного перехода по мере распространения волны будет увеличиваться.

Амплитуда упругого предвестника с увеличением пройденного волной расстояния уменьшается. Релаксация упругого предвестника заканчивается на расстоянии не более 5-10 см .

При распространении плоских ударных волн завихрених течений не возникает, поэтому целесообразно использовать для описания



Рис. 15. Схема моделируемых экспериментов

движения среды лагранжеву систему координат.

Выберем исходную систему таким образом, чтобы направление оси X совпадало с направлением распространения ударной волны, а начало координат поместим в точку на свободной поверхности метаемой пластины. В этом случае значительно упрощается постановка граничных и начальных условий.

Задача о распространении плоских ударных волн может решаться в одномерной постановке, т.к. влияние боковых волн разгрузки в интересующем пользователей интервале времени можно не учитывать.

Для описания одномерных пластических течений при высокоскоростном деформировании разработаны релаксационные микродинамические модели, которые обеспечивают адекватное описание нестационарных явлений волновой динамики (затухание упругих предвестников, эволюцию нестационарных волн напряжений). Можно воспользоваться вариантом, предложенным в [28].

Таким образом, главные физические стороны задачи уяснены и можно переходить к ее математической постановке.

6.2. Математическая постановка задачи

1. Геометрическое моделирование

Рассмотрим случай соударения двух пластин конечной толщины. Будем считать началом рассматриваемого процесса момент соприкосновения соударяющихся пластин. С учетом ранее выбранной системы координат будем иметь:

- координаты точек метаемой пластины толщиной H_2 : $0 \leq x \leq H_2$;
- координаты точек неподвижной пластины толщиной H_1
 $H_2 < x \leq H_1 + H_2$.

2. Постановка граничных условий

Как видно из рис.15, в момент контакта пластин образуются три границы:

- а) свободная поверхность при $x = 0$, граничные условия для которой имеют вид $\sigma_1|_{x=0} = 0$;
 - б) контактная поверхность при $x = H_2$, на которой в момент удара терпит разрыв массовая скорость:
при $0 \leq x \leq H_2$ $u(x) = V_0$ (V_0 - скорость соударения),
при $H_2 < x \leq H_1 + H_2$ $u(x) = 0$;
- кроме того, в общем случае материалы соударяющихся пластин могут быть разными;

в) свободная поверхность при $x = H_\delta + H_0$, на которой $\sigma_x = 0$.

3. Моделирование физических процессов

В плоской волне напряжений сжатие происходит только в направлении распространения волны. Обозначим его 1. Следовательно, в плоской волне будет отлична от нуля только одна составляющая массовой скорости U_1 , нормальная к фронту волны. Отличной от нуля будет только одна полная деформация — ε_1 .

Предположим, что приращение полной деформации элемента тела $\dot{\varepsilon}_{ij}$ складывается из приращения упругой $\dot{\varepsilon}_{ij}^e$ и пластической $\dot{\varepsilon}_{ij}^p$ составляющих:

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}_{ij}^e + \dot{\varepsilon}_{ij}^p \quad (6.1)$$

В главных осях будем иметь

$$\dot{\varepsilon}_1 = \dot{\varepsilon}_1^e + \dot{\varepsilon}_1^p, \quad \dot{\varepsilon}_2 = \dot{\varepsilon}_2^e + \dot{\varepsilon}_2^p, \quad \dot{\varepsilon}_3 = \dot{\varepsilon}_3^e + \dot{\varepsilon}_3^p \quad (6.2)$$

Учитывая, что в плоской волне $\dot{\varepsilon}_2 = 0$ и $\dot{\varepsilon}_3 = 0$, получим

$$\dot{\varepsilon}_2^e = \dot{\varepsilon}_2^p, \quad \dot{\varepsilon}_3^e = \dot{\varepsilon}_3^p \quad (6.3)$$

Вследствие осевой симметрии плоской волны имеем

$$\dot{\varepsilon}_2^e = \dot{\varepsilon}_3^e, \quad \dot{\varepsilon}_2^p = \dot{\varepsilon}_3^p \quad (6.4)$$

Согласно постулату пластической несжимаемости пластическая деформация не изменяет объем:

$$\dot{\varepsilon}_1^p + \dot{\varepsilon}_2^p + \dot{\varepsilon}_3^p = 0 \quad (6.5)$$

Объемная деформация в случае плоской волны полностью определяется деформацией

$$\theta = \frac{1}{3} (\dot{\varepsilon}_1^e + \dot{\varepsilon}_2^e + \dot{\varepsilon}_3^e) = \frac{1}{3} \dot{\varepsilon}_1 \quad (6.6)$$

Введем скорость сдвиговой пластической деформации в виде

$$\dot{\gamma}^p = \frac{1}{2} (\dot{\varepsilon}_1^p - \dot{\varepsilon}_2^p) = \frac{3}{4} \dot{\varepsilon}_1^p \quad (6.7)$$

Напряженное состояние в каждой точке среды характеризуется симметричным тензором напряжения σ_{ij} , который удобно представить в виде суммы шаровой $p\delta_{ij}$ и дивiatorной S_{ij} частей:

$$\sigma_{ij} = -p\delta_{ij} + S_{ij} \quad (6.8)$$

В главных осях будем иметь

$$\sigma_1 = -p + S_1, \quad \sigma_2 = -p + S_2, \quad \sigma_3 = -p + S_3 \quad (6.9)$$

Вследствие симметрии плоской волны

$$\sigma_2 = \sigma_3, \quad s_2 = s_3 \quad (6.10)$$

Из (6.9) и (6.10) следует

$$s_1 + s_2 + s_3 = 0 \quad (6.11)$$

Введем сдвиговое напряжение

$$\tau = 1/2 (\sigma_1 - \sigma_2) = 1/2 (s_1 - s_2) = \frac{3}{4} s_1 \quad (6.12)$$

Учитывая (6.10)–(6.12), будем иметь

$$\tau = \frac{3}{4} s_1 \quad (6.13)$$

Для рассматриваемого диапазона скоростей соударения напряжения в ударных волнах не превышают 10 ГПа, поэтому возможно использование баротропного уравнения состояния. Выберем уравнение состояния в виде

$$p = A \left(\frac{p}{p_0} - 1 \right) + B \left(\frac{p}{p_0} - 1 \right)^2 + C \left(\frac{p}{p_0} - 1 \right)^3, \quad (6.14)$$

где параметры A , B , C являются константами для конкретных материалов (см. Прил. 2).

Для описания пластического течения используем микродинамическую модель с дислокационной кинетикой [28].

Закон Гука в приращениях будет иметь вид

$$\dot{s}_{ij} = 2\mu (\dot{\epsilon}_{ij}^e) \quad (6.15)$$

где $\dot{\epsilon}_{ij}^e = (\dot{\epsilon}_{ij}^e - \dot{\theta} \delta_{ij}^p)$ – девиатор тензора скорости упругих деформаций $\dot{\epsilon}_{ij}^e$; μ – модуль сдвига.

Учитывая, что $\theta = 1/3 \dot{\epsilon}_1$, а также выражения (6.2), (6.5) и (6.6), получим

$$\dot{s}_1 = 2\mu (\dot{\epsilon}_1^e - 1/3 \dot{\epsilon}_1) = 2\mu (\dot{\epsilon}_1 - \dot{\epsilon}_1^p - 1/3 \dot{\epsilon}_1) = \mu (4/3 \dot{\epsilon}_1 - 2 \dot{\epsilon}_1^p) \quad (6.16)$$

Из (6.16), (6.12) и (6.7) получаем уравнение для приращений сдвиговых напряжений

$$\dot{\tau} = \mu (\dot{\epsilon}_1 - 2 \dot{\gamma}^p) \quad (6.17)$$

Скорость сдвиговой пластической деформации в микродинамической модели [28] связывается с помощью уравнения Орована с плотностью подвижных дислокаций N_m , средней скоростью движения дислокаций v и вектором Бюргерса b :

$$\dot{\gamma}^p = g b N_m v, \quad (6.18)$$

где g – ориентационный коэффициент, равный для поликристал-

лических материалов приблизительно 0,5.

Изменения дислокационных параметров определяются кинетическими уравнениями следующего вида :

$$\begin{aligned}
 N_m &= N \cdot f; \\
 N &= N^* + (N_0 - N^*) \exp\left(-\frac{A}{|g|B} \gamma_K^P\right), \quad \gamma_K^P = \int_0^t |\dot{\gamma}^P| dt, \\
 N_0' &= N_0, \\
 N_0' &= N_0 + \frac{|\bar{\epsilon}_i| - \tau^*}{\beta_1} \quad \text{при } \bar{\epsilon}_i > -\tau^*, \\
 \bar{\epsilon}_i &= \min\{\tau_i^n, n=0, \Delta t, \dots, t\}, \\
 f &= f^* + (f_0 - f^*) \exp\left(-\frac{B}{|g|B} (\gamma_K^P - \gamma_{KR}^P)\right), \quad (6.19) \\
 \gamma_{KR}^P &= \gamma_K^P, \text{ если } (\tau_i^{n+1} \cdot \tau_i^n) < 0, (n=0, \Delta t, \dots, t), \\
 v &= v_0 \frac{ST^2}{1 + ST^2} \cdot \text{Sign}(\tau), \\
 ST &= \frac{|\tau| - \tau_0'}{\beta_2} \quad \text{при } |\tau| > \tau_0', \quad \tau_0' = \tau_0 + \alpha \mu \beta \sqrt{N}, \\
 ST &= 0 \quad \text{при } |\tau| \leq \tau_0'.
 \end{aligned}$$

Константами модели являются $N_0, N^*, A, B, \tau^*,$

$f_0, f^*, \beta_1, v_0, \tau_0, \beta_2, \alpha, \mu$.

Начальные условия для кинетических уравнений учитывают то, что материал до момента соударения не деформировался пластически. Следовательно :

$$\gamma_K^P = 0, \quad \gamma_{KR}^P = 0, \quad \bar{\epsilon}_i = 0.$$

Изменения микроструктуры материала, связанные с пластической деформацией при подготовке образцов, учтены в начальных значениях микропараметров N_0, f_0, τ_0, τ^* .

Существенно, что кинетические соотношения модели задают скорость сдвиговой пластической деформации в виде нелинейной функции самой пластической деформации и сдвигового напряжения. Вследствие этого определяющее уравнение для сдвигового напряжения (6.17) становится существенно нелинейным

$$\dot{\epsilon} = \mu (\epsilon_i - 2\gamma^P(\tau, \gamma^P)). \quad (6.20)$$

Система уравнений, описывающая распространение плоской волны напряжений в переменных Лагранжа, включает :

а) уравнение движения

$$\rho \frac{\partial u_i}{\partial t} = \frac{\partial \sigma_i}{\partial x}; \quad (6.21)$$

б) уравнение неразрывности

$$\text{где } V = \rho^0 / \rho : \quad \frac{1}{V} \frac{\partial u}{\partial x} = \frac{\partial V}{\partial t}, \quad (6.22)$$

в) определяющее уравнение, включающее соотношения (6.8), (6.13), (6.14), (6.17)–(6.19).

Начальные условия записываются с учетом различия материалов соударяющихся пластин как по плотности и механическим характеристикам, так и по константам микродинамической модели.

Полученная система уравнений является гиперболической, следовательно, необходимо решать ее численным методом.

6.3. Метод решения

Для решения нелинейных систем гиперболического типа целесообразно выбирать явные схемы. Исходя из особенностей решаемой задачи, желательно, чтобы выбранная схема позволяла проводить сквозной счет, без специального выделения скачков в решении. Так как создаваемый программный комплекс предназначен для исследования структуры ударноволновых фронтов, необходимо, чтобы метод решения вносил наименьшее искажение в структуру скачков. Следовательно, необходимо выбирать схемы с малой аппроксимационной вязкостью. По этой же причине следует выбирать схему не ниже второго порядка точности.

Указанным требованиям удовлетворяют схемы Уилкинса, Лакса-Вендроффа, Рунанова и др. [13, 21]. Выберем для решения схему второго порядка типа крест, предложенную Уилкинсом [29].

При использовании этой схемы тело разбивается на пространственную сетку с равномерным шагом H :

$$K_0 = \frac{H_1 + H_2}{H}, \quad X_{j+1}^0 = X_j^0 + H, \quad (j=1, \dots, K_0+1), \quad X_1^0 = 0. \quad (6.23)$$

Расчет ведется с временного слоя n на слой $n+1$ либо со слоя $n-1/2$ на $n+1/2$.

Уравнение движения в разностном аналоге имеет вид

$$u_j^{n+1/2} = u_j^{n-1/2} + \frac{\Delta t}{\Delta x} [\sigma_{j+1/2}^n - \sigma_{j-1/2}^n], \quad (6.24)$$

$$\text{где } y_j^n = \left[\rho_{j+1/2}^0 \left(\frac{x_{j+1}^n - x_j^n}{V_{j+1/2}^n} \right) + \rho_{j-1/2}^0 \left(\frac{x_j^n - x_{j-1}^n}{V_{j-1/2}^n} \right) \right]$$

$$V_{j+1/2}^n = \rho_{j+1/2}^0 / \rho_{j+1/2}^n,$$

$$x_j^{n+1} = x_j^n + u_j^{n+1/2} \cdot \Delta t^{n+1/2}, \quad x_j^{n+1/2} = (x_j^{n+1} - x_j^n) / 2$$

Уравнение неразрывности (6.22) примет вид

$$V_{j+1/2}^{n+1} = V_{j+1/2}^n + \frac{\Delta t^{n+1/2} (u_{j+1}^{n+1/2} - u_j^{n+1/2})}{(x_{j+1}^0 - x_j^0)} \quad (6.25)$$

Схема Уилкинса обладает малой аппроксимационной вязкостью. Для стабилизации расчетной сетки и "размазывания" фронта ударной волны схема использует искусственную вязкость Рихтмайера [29]. Искусственная вязкость позволяет проводить сквозной счет при описании реакции среды моделями (гидродинамической, идеальной упругопластической), в которых нет механизма вязкости и диссипации. Величина констант, входящих в выражение для искусственной вязкости, подбирается таким образом, чтобы фронт ударной волны "размазывался" не менее чем на 3-5 ячеек сетки. В этом случае ударный переход уже не является разрывом.

Схемы с искусственной вязкостью никогда не претендовали на сколько-нибудь правильное описание течения в самом фронте ударной волны, поскольку искусственная вязкость не имеет никакого отношения к физической вязкости, характеризующей процесс диссипации энергии.

Применение для описания пластического течения микродинамической модели позволяет учесть вязкопластическое поведение материала, и принципиальная необходимость использования механизма искусственно вводимой диссипации отпадает. Поэтому при расчетах распространения плоских волн напряжения с использованием микродинамической модели будем исключать искусственную вязкость из расчетной схемы .

Определяющее уравнение для разностного аналога запишется в виде

$$p_{j+1/2}^n = -p_{j+1/2}^n + \frac{4}{3} \tau_{j+1/2}^n \quad (6.26)$$

Давление будет определено с помощью (6.14):

$$p_{j+1/2}^n = A \xi_{j+1/2}^n + B (\xi_{j+1/2}^n)^2 + C (\xi_{j+1/2}^n)^3, \quad (6.27)$$

где

$$\xi_{j+1/2}^n = \left(\frac{1}{V_{j+1/2}^n} - 1 \right).$$

Скорость деформации может быть найдена с помощью уравнения

$$\dot{\epsilon}_{j+1/2}^{n+1/2} = \frac{(u_{j+1/2}^{n+1/2} - u_j^{n+1/2})}{(x_{j+1}^{n+1/2} - x_j^{n+1/2})} \quad (6.28)$$

Для отыскания значений $\tau_{j+1/2}^n$ необходимо для каждого узла и на каждом временном слое решать нелинейное дифференциальное уравнение (6.20). Сделать это можно несколькими способами. Например, можно организовать итеративный процесс совместного решения определяющего уравнения для сдвигового напряжения в разностном аналоге и кинетического уравнения для скорости сдвиговой пластической деформации (6.18) - (6.19):

$$\begin{aligned} [\tau_{j+1/2}^{n+1}]^M &= \tau_{j+1/2}^n + \mu \Delta t^{n+1/2} (\dot{\epsilon}_{j+1/2}^{n+1/2} - [\dot{\gamma}_{j+1/2}^{P n+1/2}]^{M-1}), \\ [\tau_{j+1/2}^{n+1/2}]^M &= \frac{1}{2} ([\tau_{j+1/2}^{n+1}]^M + \tau_{j+1/2}^n), \\ [\dot{\gamma}_{j+1/2}^{P n+1/2}]^M &= \dot{\gamma}^P ([\tau_{j+1/2}^{n+1/2}]^M, [\dot{\gamma}_{j+1/2}^{P n+1/2}]^{M-1}), \\ [\dot{\gamma}_{j+1/2}^{P n+1}]^M &= \dot{\gamma}_{j+1/2}^{P n} + \Delta t^{n+1/2} [\dot{\gamma}_{j+1/2}^{P n+1/2}]^M, \\ [\dot{\gamma}_{j+1/2}^{P n+1/2}]^M &= \frac{1}{2} (\dot{\gamma}_{j+1/2}^{P n} + [\dot{\gamma}_{j+1/2}^{P n+1}]^M), \\ [\dot{\gamma}_{j+1/2}^{P n+1/2}]^M &= \dot{\gamma}^P ([\tau_{j+1/2}^{n+1/2}]^M, [\dot{\gamma}_{j+1/2}^{P n+1/2}]^M), \end{aligned} \quad (6.29)$$

где M - номер итерации.

В качестве первого приближения выбирается значение скорости пластической деформации на предыдущем временном слое:

$$[\dot{\gamma}_{j+1/2}^{P n+1/2}]^0 = \dot{\gamma}_{j+1/2}^{P n-1/2}, \quad [\dot{\gamma}_{j+1/2}^{P n+1/2}]^0 = \dot{\gamma}_{j+1/2}^{P n-1/2}. \quad (6.30)$$

Для приведенного варианта итерационного процесса выполняется условие сходимости.

Выход из итерационного процесса будем осуществлять при выполнении критерия

$$|([\tau_{j+1/2}^{n+1}]^M - [\tau_{j+1/2}^{n+1}]^{M-1}) / [\tau_{j+1/2}^{n+1}]^M| < \delta. \quad (6.31)$$

Скорость сдвиговой пластической деформации будет вычисляться следующим образом :

$$\begin{aligned} \gamma_{j+1/2}^{P n+1/2} &= g \cdot N_{j+1/2}^{n+1/2} f_{j+1/2}^{n+1/2} v_{j+1/2}^{n+1/2}, \\ \gamma_{K j+1/2}^{P n+1/2} &= \gamma_{K j+1/2}^{P n-1/2} + \left| |\gamma_{j+1/2}^{P n+1/2}| - |\gamma_{j+1/2}^{P n-1/2}| \right|, \\ \bar{\tau} &= \min \{ \tau_{j+1/2}^{n+1/2}, \bar{\tau} \}, \text{ если } (\tau_{j+1/2}^{n+1/2} \cdot \tau_{j+1/2}^{n-1/2}) < 0 \quad \gamma_{Kq}^P = \gamma_{K j+1/2}^{P n+1/2}, \\ N'_0 &= N_0, \text{ если } \bar{\tau} \geq -\tau^*, \quad N'_0 = N_0 + \frac{|\bar{\tau}| - \tau^*}{\beta_1}, \text{ если } \bar{\tau} < -\tau^*, \\ N_{j+1/2}^{n+1/2} &= N^* + (N'_0 - N^*) \exp \left(-\frac{A}{1g\beta} \gamma_{K j+1/2}^{P n+1/2} \right), \\ \tau_{j+1/2}^{n+1/2} &= \tau_0 + \alpha \mu \beta \sqrt{N_{j+1/2}^{n+1/2}}, \\ f_{j+1/2}^{n+1/2} &= f^* + (f_0 - f^*) \exp \left(-\frac{B}{1g\beta} (\gamma_{K j+1/2}^{P n+1/2} - \gamma_{KR j+1/2}^{P n+1/2}) \right), \end{aligned} \quad (6.32)$$

$$\begin{aligned} \text{если } |\tau_{j+1/2}^{n+1/2}| > \tau'_{0 j+1/2} \quad , \quad ST &= \frac{|\tau_{j+1/2}^{n+1/2}| - |\tau'_{0 j+1/2}|}{\beta_2}, \\ \text{если } |\tau_{j+1/2}^{n+1/2}| \leq \tau'_{0 j+1/2} \quad , \quad ST &= 0, \\ v_{j+1/2}^{n+1/2} &= v_0 \frac{ST^2}{1+ST^2} \cdot \text{Sign}(\tau_{j+1/2}^{n+1/2}). \end{aligned}$$

Граничные условия для равностного аналога системы уравнений (6.21)-(6.23) задаются в виде

$$\begin{aligned} \text{для } x(1) &= 0 \quad \sigma_{j-1/2}^n = -\sigma_{j+1/2}^n, \\ \text{для } x(K_0+1) &= H_n + H_\delta \quad \sigma_{K_0+1/2} = -\sigma_{K_0+1/2}. \end{aligned} \quad (6.33)$$

В (6.33) учтено, что материальная граница находится в целых узлах $j = I$ и $j = K_0 + I$. Для того чтобы обеспечить выполнение условий на свободной поверхности, необходимо ввести псевдоузел с нулевой массой, напряжение в котором отыскивается из (6.33). Граничные условия должны выполняться на любом временном слое.

Начальные условия будут иметь вид

$$V_{j+1/2}^0 = 1, \quad \bar{G}_{j+1/2}^0 = 0, \quad \gamma_{K_{j+1/2}}^{P_{0+1/2}} = 0, \quad \gamma_{j+1/2}^{P_{0+1/2}} = 0, \quad \gamma_{KR_{j+1/2}}^{P_{0+1/2}} = 0, \\ \bar{U} = \bar{U}^* \quad \text{для всех } j \in (1, K_0+1), \\ \text{где } K_0 = \frac{H_\delta + H_\Pi}{H}; \quad L_0 = \frac{H_\delta}{H}; \quad (6.34) \\ U_j^{0+1/2} = \bar{U}_0 \quad \text{для } 1 \leq j \leq L_0+1; \\ U_j^{0+1/2} = 0 \quad \text{для } L_0+1 < j \leq K_0+1.$$

Кроме того, необходимо учесть, что для узлов $1 \leq j \leq L_0+1$ все значения параметров, характеризующих среду (плотность, модуль сдвига, параметры микродинамической модели), должны соответствовать материалу метаемой пластины, а для узлов $L_0+1 < j < K_0+1$ — материалу неподвижной пластины.

Для устойчивости решения необходимо выполнение условия Куранта. Схема Уилкинса использует это условие в форме, предложенной фон Нейманом и Рихтмайером [29]:

$$\Delta t^{n+3/2} = \frac{1}{3} \frac{\Delta X^{nn}}{\sqrt{C_L^2 + \beta^2}} \Big|_{\min n_0}, \quad \Delta X^{nn+1} = X_{j+1}^{nn+1} - X_j^{nn+1}. \quad (6.35)$$

Если $\Delta t^{n+3/2} > 1,1 \Delta t^{n+1/2}$, то положим

$$\Delta t^{n+3/2} = 1,1 \Delta t^{n+1/2}, \quad \Delta t^{n+1} = \frac{1}{2} (\Delta t^{n+3/2} + \Delta t^{n+1/2}).$$

Здесь C_L — продольная скорость звука;

$$b = 2 C_0 \Delta X^{nn} \left(\frac{\dot{V}}{V} \right)^{n+1/2}; \quad C_0 \approx 2;$$

$$b = 0, \quad \text{если} \quad \left(\frac{\dot{V}}{V} \right) \geq 0.$$

Таким образом, уравнения в конечных разностях (6.23) — (6.30) с граничными и начальными условиями (6.33), (6.34) позволяют при условии (6.35) получить решение системы уравнений, описывающей распространение плоских волн напряжений.

6.4. Разработка алгоритма и структуры программного комплекса

Последовательная запись (6.23) — (6.35) еще не является алгоритмом решения задачи, хотя и определяет порядок пересчета параметров упругопластического течения с одного временного

слоя на другой.

Решение задачи заключается в нахождении параметров течения в узлах пространственной сетки для каждого момента времени в рассматриваемом диапазоне $t \in (0, t_n)$. Здесь t_n — время протекания исследуемого процесса.

Таким образом, алгоритм решения задачи будет иметь циклическую структуру вида, показанного на рис. 16, а.

Примененная схема относится к схемам типа "крест", которые для нахождения параметров течения в узле сетки j на последующем временном слое ($n+1$ или $n+1/2$) используют не только значения на предыдущем слое (n или $n-1/2$), но и значения в соседних узлах $j+1/2$ и $j-1/2$ на промежуточном временном слое. Поэтому необходимо сохранять все значения параметров на расчетном временном слое. Для этого введем массивы переменных. Для удобства кодирования заменим полуцелые значения узлов на целые значения, как это показано на рис. 15. В этом случае изменится интервал $j' \in (1, K_0+2)$. Так, для $\bar{b}_{j'+1/2}^n$ j' будет изменяться от 1 до K_0+2 , т.к. узлы $j'=1$ и $j'=K_0+2$ соответствуют псевдоузлам.

Блок вычисления параметров течения на временном слое будет представлять собой последовательность циклических структур, каждая из которых предназначена для нахождения $u_j^{n+1/2}, x_j^{n+1}, x_j^{n+1/2}, v_{j+1/2}^{n+1}, \bar{b}_{j+1/2}^n$. Общая структура вычислительного блока показана на рис. 16, б.

При использовании итерационного процесса (6.29) для нахождения значения сдвигового напряжения приходится неоднократно вычислять скорость сдвиговой пластической деформации (6.32), поэтому удобно выделить эту группу операций как предопределенный процесс (подпрограмму).

Началу вычислений должно предшествовать присвоение элементам массивов и переменным их начальных значений. Эту группу операций можно выделить как блок начальных условий. Этот блок выполняется только один раз.

В конце выполнения вычислительного блока (т.е. при завершении вычислений параметров течения на соответствующем временном слое) необходимо проверить выполнение условия устойчивости (6.35). В ударной волне пространственные интервалы сетки уменьшаются и, следовательно, уменьшается величина временного шага.

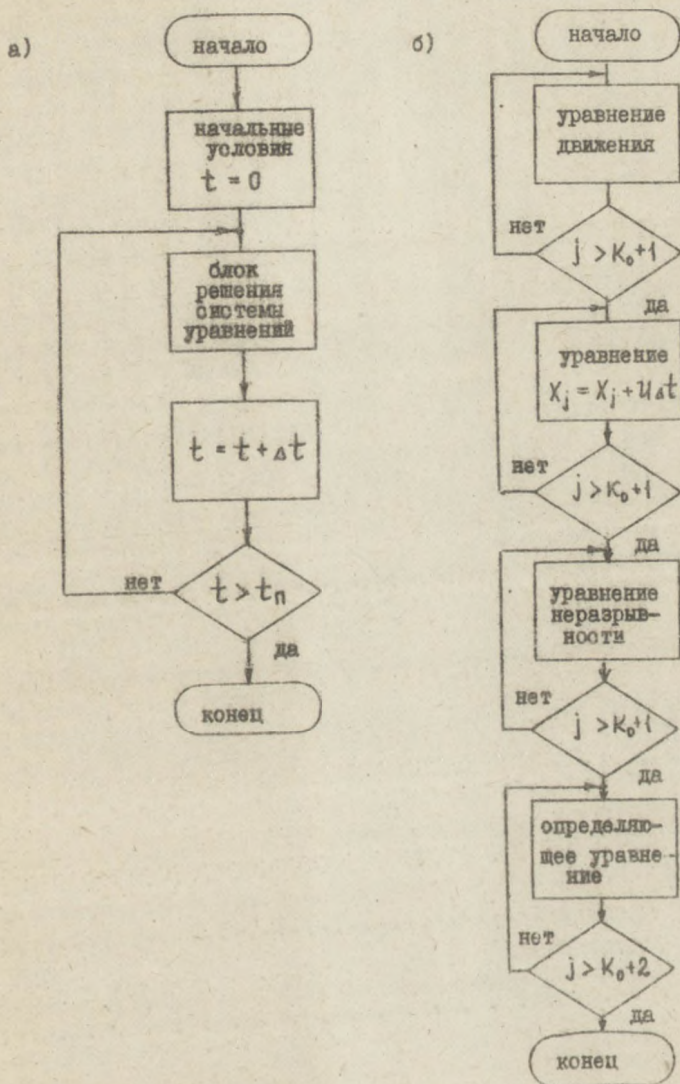


Рис.16. Алгоритм решения :

а - общий алгоритм программы ;

б - алгоритм блока решения системы уравнений

Построенный алгоритм должен обеспечивать вычисление параметров течения во всех узлах сетки и последовательные моменты времени. Значения параметров при переходе к следующему временному слою не сохраняются, поэтому, для того чтобы исследовать эволюцию структуры фронта волны напряжений, нужно организовать сохранение необходимой информации. Можно сделать это несколькими способами. Наиболее простой заключается в выдаче параметров течения на временном слое (во всех узлах сетки или их части) на АЦПУ. Сравнивая выдачи для последовательных моментов времени, можно проследить изменение структуры фронта и эволюцию напряженно-деформированного состояния соударяющихся пластин. Операции, связанные с подготовкой и выдачей результатов расчетов на АЦПУ, объединим в блок выдачи.

На этом этап разработки общей структуры алгоритма можно считать законченным.

В процессе отладки и тестирования, когда будет выяснена работоспособность алгоритма в целом, его необходимо доработать и оптимизировать. Забегая вперед, отметим, что алгоритм в рассмотренном варианте предусматривает большое количество бесполезных операций. Например, пересчитываются значения параметров течения в областях, куда еще не дошло возмущение (т.е. выполняется пересчет нулей в нули). Возмущение распространяется от поверхности соударения вглубь пластин с конечной скоростью, поэтому в начальные моменты времени основная масса ячеек не участвует в упругопластическом течении. Сокращение ненужных операций увеличивает быстродействие программы.

6.5. Отладка и тестирование

Центральным в теории разностных схем является вопрос о существовании и единственности решения. До настоящего времени для гиперболических систем он не решен. Вопросы сходимости и устойчивости для нелинейных гиперболических систем, как и в данном случае, приходится решать практически — путем численного эксперимента. Оценки устойчивости схемы Уилкинса, которая была выбрана за основу численного метода решения, в литературе приводились. Однако переносить их на модифицированную схему, строго говоря, нельзя. Особенность в решении может

возникнуть при решении уравнения (6.20) для сдвигового напряжения . Ошибка в его решении вызовет ошибку в напряжении, входящем в уравнение (6.25) и, в свою очередь , повлияет на величину скорости деформации (6.28). Ошибки могут вызвать появление осцилляций в решении и привести к неустойчивости.

При расчетах приходится сталкиваться с тем, что схема в некоторых случаях приводит к значительным (5 - 10 % от амплитудных значений) осцилляциям расчетных значений параметров течения . Устранение осцилляций основных параметров течения u , v , b , p , τ методами пересчета (методами подавления колебаний пересчетом) недопустимо . В этом случае не удастся получить правильную величину массовой скорости и скорости распространения волн.

Уменьшить величину осцилляций иногда удается соответствующим подбором параметров разностной схемы (шага по пространству, шага по времени). Кроме того, схема Уилкинса использовала для подавления осцилляций искусственную вязкость, которая была исключена из схемы, реализованной в программе .

Таким образом, перед отладкой программы уже известно, что причинами возможных авостов могут являться не только ошибки программирования, но и неустойчивость схемы.

В этом случае рационально поступить следующим образом. На первом этапе исправить синтаксические ошибки. После этого выключить блок вычисления сдвиговых напряжений и ввести в схему искусственную вязкость . Проводя тестовые расчеты, в этом случае удается уменьшить неопределенность в причинах авостов.

Расчет течения при нулевых сдвиговых напряжениях равносильно решению задачи в гидродинамической постановке. В качестве теста можно использовать сравнение расчетного давления во фронте волны с оценками из соотношений Ренкина - Гюгонио [15] .

В ходе отладки и тестирования программы с гидродинамической моделью среды было выявлено, что источником осцилляций может являться контактная поверхность, если материалы имеют существенно разные плотности и уравнения состояния. Уменьшить эти осцилляции можно, применяя аналог формулы пересчета для контактных узлов, предложенной Рихтмайером. Формула пересчета будет иметь вид

$$u_{i,0+1}^{n+1/2} = u_{i,0+1}^{n-1/2} + \Delta t \left[\frac{1}{2} (p_{i,1/2}^n - p_{i,0+1/2}^n) - \frac{1}{2} (p_{i,2}^n - p_{i,0}^n) \right] / (u_{i,0}^n - u_{i,0+2}^n).$$

После того как программа будет отлажена на гидродинамической модели, приступают к отладке блока решения кинетического уравнения, но с присутствием искусственной вязкости в схеме. В ходе такой отладки было выяснено, что величина осцилляций в значительной степени зависит от выбора пространственного шага сетки и использованного в условиях устойчивости числа Куранта.

В соотношении (6.35) было использовано число Куранта, равное $1/3$, однако наименьшие осцилляции и лучшую устойчивость счета при разных амплитудах ударных волн удается получить при числе Куранта равном $1/4$. Применение меньших значений вызывает увеличение времени счета, которое не сравнимо с получаемым выигрышем в величине осцилляций.

Важным оказался вопрос о выборе пространственного шага сетки. При использовании искусственной вязкости фронт ударной волны размывается на 3-5 счетных интервалов, причем практически не зависимо от величины шага. Необходимо только подобрать значения констант, входящих в выражение для искусственной вязкости : для линейной - C_L

$$q_{j+1/2}^{n+1/2} = C_L a s_{j+1/2}^0 \frac{1}{V_{j+1/2}^{n+1/2}} \left| u_{j+1}^{n+1/2} - u_j^{n+1/2} \right|, \quad (6.36)$$

где a - местная скорость звука ;

для квадратичной - C_v

$$q_{j+1/2}^{n+1/2} = C_v^2 s^0 \frac{1}{V_{j+1/2}^{n+1/2}} \left| u_{j+1}^{n+1/2} - u_j^{n+1/2} \right|^2 \quad (6.37)$$

При использовании релаксационных моделей вопрос о выборе шага пространственной сетки связан с вопросом, будет ли схватываться физическое явление. Анализ тестовых расчетов показал, что явления волновой динамики описываются, если фронт пластической волны составляет не менее 10-15 счетных интервалов. Поскольку ширина ударного перехода зависит от интенсивности волны, расчеты ударных волн разной амплитуды должны выполняться с разными пространственными шагами.

При использовании мелкого шага возрастают осцилляции, возникающие при распаде разрыва течения в самые первые моменты времени. Вследствие этого на начальных этапах расчета возникают особенности нефизического характера. Уменьшить осцилляции удается путем задания начального условия на контактной поверхности в виде

$$u_{L_c+1} = V_0/2.$$

При этом учитывалось, что после распада разрыва в глубь пластины пойдет волна с амплитудой массовой скорости (при соударении пластин из одинаковых материалов), равной приблизительно половине скорости удара.

При таком задании начальных условий в два раза уменьшается градиент массовой скорости вблизи контактной поверхности, а следовательно, и $\dot{\epsilon}_1$. Благодаря этому скорость сходимости итерационного процесса решения (6.20) увеличивается.

Эффект от этого приема особенно заметен при высоких скоростях соударения, когда формируются фронты волн интенсивностью в 5 - 10 ГПа и выше.

При моделировании распространения волн с амплитудами более 10 ГПа этих мер оказывается недостаточно и в решении возникают осцилляции порядка 10 - 20 %. Для уменьшения этих осцилляций было предложено первые 10 - 20 шагов по времени проводить расчет с использованием искусственной вязкости, а затем ее исключать из схемы. При этом амплитуда колебаний в массовой скорости и напряжении уменьшается до 2 - 5 %.

На последнем этапе отладки и тестирования программы проводятся расчеты распространения волн разной интенсивности с применением микродинамической модели. Искусственная вязкость из схемы исключается. В ходе этих расчетов осуществляется исследование сходимости итерационного процесса и влияния на структуру фронта, получаемую в расчете, изменений параметров микродинамической модели.

Дополнительные расчеты проводятся с целью проверки сохранения полной энергии системы пластин. Эти расчеты показали, что энергия сохраняется на протяжении всего расчета, в том числе и после взаимодействия ударных волн со свободными поверхностями, с точностью до 1 %.

В качестве тестов используется сравнение скоростей распространения упругой и пластической волн, найденных по результатам расчетов, с экспериментальными значениями. Кроме того, о правильности работы программы можно судить, сравнивая расчеты эволюции фронта волны (структуру фронта, затухание упругого предвестника) с экспериментальными данными.

Текст программы со списком принятых обозначений и идентификаторов приведен в Прил. 3.

ПРИЛОЖЕНИЕ I

Часто применяющиеся модели упруговязкопластических
изотропных сред

1. Модель Бингама [19].

$$\begin{aligned} \sigma_1 &= E \varepsilon_1 & \text{при } \sigma_1 \leq \sigma_T, \\ \sigma_1 &= \sigma_T + \eta \left(\dot{\varepsilon}_1 - \frac{\dot{\sigma}_1}{E} \right) & \text{при } \sigma_1 > \sigma_T, \end{aligned}$$

где σ_T - начальный предел текучести при растяжении;
 E - модуль Юнга;
 η - коэффициент вязкости.

Модель Бингама не учитывает деформационное упрочнение.

2. Модель Шведова [19].

$$\begin{aligned} \dot{\varepsilon}_1 &= \frac{\dot{\sigma}_1}{E} & \text{при } \sigma_1 \leq \sigma_T, \\ \dot{\varepsilon}_1 &= \left(\frac{1}{E_2} + \frac{1}{E} \right) \dot{\sigma}_1 + \frac{(\sigma_1 - \sigma_T)}{\eta} & \text{при } \sigma_1 > \sigma_T, \end{aligned}$$

где E_2 , E , η - константы материала;
 σ_T - предел текучести при растяжении.

Модель Шведова не учитывает деформационное упрочнение.

3. Модель Малверна - Соколовского - Пажины [19].

$$\begin{aligned} \dot{\varepsilon}_1 &= \frac{\dot{\sigma}_1}{E} + \frac{1}{\eta} \langle \Phi(\sigma_1 - \sigma_T) \rangle, \\ \langle \Phi(\sigma_1 - \sigma_T) \rangle &= 0 & \text{при } \sigma_1 \leq \sigma_T, \\ \langle \Phi(\sigma_1 - \sigma_T) \rangle & & \text{при } \sigma_1 > \sigma_T, \end{aligned}$$

где Φ - нелинейная функция от $(\sigma_1 - \sigma_T)$, подлежащая определению из эксперимента для каждого материала. Часто эта функция аппроксимируется степенной, экспоненциальной и др. элементарными функциями. Например:

$$\begin{aligned} \langle \Phi \rangle &= \left(\frac{\sqrt{J_2}}{K} - 1 \right)^p, & \text{где } K &= \sqrt{(\sigma_T^2 + 2H W_p) / 3}, \\ \langle \Phi \rangle &= \left(\frac{\sqrt{J_2}}{K} - 1 \right)^p, & \text{где } K &= (\sigma_T + H \varepsilon_u^p) \sqrt{3}, \\ \langle \Phi \rangle &= \left(\frac{\sqrt{J_2} - K}{\sqrt{3} \sigma_1} \right), & \text{где } K &= (\sigma_T + H \varepsilon_u^p) / \sqrt{3}. \end{aligned}$$

Здесь η - коэффициент вязкости ;
 J_2 - второй инвариант тензора напряжений ;
 H - Параметр, учитывающий деформационное упрочнение ;
 ϵ_p^P - интенсивность пластических деформаций ;
 W_P - работа на пластических деформациях ;
 σ_T - предел текучести при растяжении ;
 E - модуль Юнга .

Модель учитывает деформационное упрочнение, если $H \neq 0$.

4. Модель Гилмана [19].

$$\begin{aligned}\dot{\epsilon}_I &= \dot{\epsilon}_I^e + \dot{\epsilon}_I^P, \\ \dot{\epsilon}_I^e &= \dot{\sigma}_I / E, \\ \dot{\epsilon}_I^P &= A_D (N_0 + C_D (\epsilon_I^P)^\beta) \exp \left(- \frac{D_C + H_C (\epsilon_I^P)}{\sigma_I} \right) \quad \text{при } \sigma_I > \sigma_T(\epsilon_I^P), \\ \dot{\epsilon}_I^P &= 0 \quad \text{при } \sigma_I \leq \sigma_T(\epsilon_I^P),\end{aligned}$$

где A_D , N_0 , C_D , D_C , H_C , β - константы материала, определяемые на основании исследований микроструктуры и механических свойств материалов. Модель учитывает деформационное упрочнение, если $H_C \neq 0$.

5. Модель Кристоеску [19].

$$\begin{aligned}\dot{\epsilon}_I &= \dot{\epsilon}_I^e + \dot{\epsilon}_I^P, \\ \dot{\epsilon}_I^e &= \frac{1}{E} \dot{\sigma}_I, \\ \dot{\epsilon}_I^P &= f_K (\sigma_I, \epsilon_I) \dot{\sigma}_I + g_K (\sigma_I, \epsilon_I^P) \quad \text{при } \sigma_I > \sigma_T(\epsilon_I^P), \\ \dot{\epsilon}_I^P &= 0 \quad \text{при } \sigma_I \leq \sigma_T(\epsilon_I^P).\end{aligned}$$

Функции f_K и g_K определяются экспериментально для каждого материала.

E - модуль Юнга . Деформационное упрочнение может быть учтено через $\sigma_T(\epsilon_I^P)$. Скорость неупругой деформации состоит из мгновенной пластической деформации (первый член) и мгновенной вязкопластической деформации (второй член) .

6. Модель Диллона - Кратохвила [19].

$$\begin{aligned}\dot{\epsilon}_I &= \dot{\epsilon}_I^e + \dot{\epsilon}_I^P, \\ \dot{\epsilon}_I^e &= \frac{\dot{\sigma}_I}{E},\end{aligned}$$

$$\dot{\epsilon}_1^P = A_c \langle \bar{\sigma}_1 - K_c \alpha \rangle ,$$

$$\dot{\alpha} = R_c \dot{\epsilon}_1^P - (\alpha - \hat{\alpha}) N_c ,$$

где $A_c, R_c, N_c, K_c, \hat{\alpha}$ - константы материала ;
 E - модуль Юнга ;

$$\langle \bar{\sigma}_1 - K_c \alpha \rangle = 0 \text{ при } \bar{\sigma}_1 \leq \bar{\sigma}_T = K_c \alpha ;$$

$$\langle \bar{\sigma}_1 - K_c \alpha \rangle = \bar{\sigma}_1 - K_c \alpha \text{ при } \bar{\sigma}_1 > \bar{\sigma}_T = K_c \alpha .$$

Модель предполагает, что скорость деформации складывается из мгновенных скоростей упругой и неупругой деформации. Деформационное упрочнение учитывается через внутренний параметр α , изменение которого описывается уравнением .

7. Модель Шاپиро [21]

$$\dot{\epsilon}_1 = f(\bar{\sigma}_1, \epsilon_1) \bar{\sigma}_1 + H(\bar{\sigma}_1, \epsilon_1) \quad \text{при } \bar{\sigma}_1 > \bar{\sigma}_T(\epsilon_1) \text{ и } \bar{\sigma}_1 > 0 ,$$

$$\dot{\epsilon}_1 = \frac{\bar{\sigma}_1}{E} + H(\bar{\sigma}_1, \epsilon_1) \quad \text{при } \bar{\sigma}_1 \geq \bar{\sigma}_T(\epsilon_1) \text{ и } \bar{\sigma}_1 < 0 ,$$

$$\dot{\epsilon}_1 = \frac{\bar{\sigma}_1}{E} \quad \text{при } \bar{\sigma}_1 \leq \bar{\sigma}_T(\epsilon_1) ,$$

где E - модуль Юнга ;
 $\bar{\sigma}_T$ - предел текучести .

В модели предел текучести $\bar{\sigma}_T(\epsilon_1)$ принимается зависящим от скорости деформации. Вид зависимости определяется экспериментально, точно так же, как и функции $f(\bar{\sigma}_1, \epsilon_1)$ и $H(\bar{\sigma}_1, \epsilon_1)$.

8. Жесткопластическая модель с условием Мизеса [21].

$$S_{ij} = 2 \mu \epsilon_{ij} \quad \text{при } S_{ij} S^{ij} < 2 \bar{\sigma}_{sd}^2 ,$$

$$S_{ij} = \frac{\sqrt{2}}{(\dot{\epsilon}_{ij} \dot{\epsilon}^{ij})^{1/2}} \dot{\epsilon}_{ij} \quad \text{при } S_{ij} S^{ij} = 2 \bar{\sigma}_{sd}^2 .$$

Напряжение течения может определяться в виде

$$\bar{\sigma}_{sd} = \bar{\sigma}_s^0 f_1(p) f_2(\dot{\epsilon}_u T) f_3(\epsilon_u^p) ,$$

где $\bar{\sigma}_s^0$ - статический предел текучести ;

ϵ_{ij} - деформация ;

$f_1(p)$ - функция зависимости предела текучести от давления ;

$f_2(\dot{\epsilon}_u T)$ - функция зависимости предела текучести от скорости деформации и температуры ;

f_3 - функция зависимости предела текучести от деформации, учитывающая деформационное упрочнение.

Функции $f_1(p)$, $f_2(\dot{\epsilon}_u T)$, $f_3(\epsilon_u^0)$ определяются экспериментально.

Часто употребляются функции следующего вида :

$$f_1(p) = (1 + \alpha p) ,$$

где $\alpha \approx 0,6/\sqrt{\sigma_{sd}}$;

$$f_2(\dot{\epsilon}_u T) = 1 - \left(\frac{C_v T}{3 E_0} \right) \ln(\dot{\gamma} / \dot{\epsilon}_u) ,$$

где C_v - теплоемкость материала;

$\dot{\gamma}$ - частота, близкая к частоте колебаний атомов ;

T - температура;

E_0 - энергия сублимации ;

$\dot{\epsilon}_u$ - скорость деформации .

ПРИЛОЖЕНИЕ 2

Наиболее употребительные уравнения состояния для давлений,
не превышающих 50 ГПа

1. Уравнение Л.П. Орленко [15]:

$$P = A \left(\frac{\rho}{\rho_0} \right)^n + B,$$

где A , B , n - константы материала;

ρ_0 - начальная плотность материала;

ρ - плотность сжатого материала.

Таблица 1

Материал	A , ГПа	B , ГПа	n	Диапазон применимости P , ГПа
Сталь	55,3	- 54,3	3	7 - 13
- " -	14,25	- 6,71	6	16,5 - 50
- " -	23,1	- 31,0	5,6	50 - 500
Дюралюминий	23,5	- 24,0	3,8	4 - 70
- " - " -	22,4	- 29,8	4,1	70 - 200
Медь	31,5	- 31,5	4,65	19 - 100
- " -	36,8	- 58,0	4,75	100 - 430
Свинец	10,3	- 10,3	4,8	20 - 50
- " -	29,4	- 51,5	3,5	100 - 425

2. Уравнение Л.В. Альтулера [15]:

$$P = A \left(\frac{\rho}{\rho_0} - 1 \right) + B \left(\frac{\rho}{\rho_0} - 1 \right)^2 + C \left(\frac{\rho}{\rho_0} - 1 \right)^3,$$

где A , B , C - константы материала.

Таблица 2

Материал	A , ГПа	B , ГПа	C , ГПа	Диапазон применимости
Бериллий	118,2	138,2	0	0 - 50 ГПа
Кадмий	47,9	108,7	282,9	- " -
Хром	207,0	223,6	702,9	- " -
Кобальт	195,4	388,9	172,8	- " -
Медь	140,7	287,1	233,5	- " -

Продолжение табл.2

Материал	A, ГПа	B, ГПа	C, ГПа	Диапазон применимости
Титан	99,0	116,8	124,6	0 - 50 ГПа
Цинк	66,2	157,7	124,2	- " -
Алюминий	76,5	165,9	42,8	- " -
Латунь	103,7	217,7	327,5	- " -
Свинец	41,7	115,9	101,0	- " -
Никель	196,3	375,0	0	- " -

3. Уравнение Осборна [30] :

$$p = [1/(e + \gamma_0)] \left\{ \xi (a_1 + a_2 |\xi|) + e [\gamma_0 + \xi (\gamma_1 + \gamma_2 \xi)] + I (c_0 + c_1 \xi) \right\},$$

где $\xi = (p/\gamma_0 - 1)$; $e = \gamma_0 I$;
 γ_0 - плотность материала при нормальных условиях;
 I - внутренняя энергия на 1 ед. массы;
 $\gamma_0, a_1, a_2, \gamma_0, \gamma_1, \gamma_2, c_0, c_1$ - константы материала.

Таблица 3

Материал	$\gamma_0, \text{г/см}^3$	a_1	a_2	γ_0	γ_1
Алюминий	2,7	1,1867	0,7630	3,4448	1,5451
Бериллий	1,845	0,9512	0,3453	0,9269	2,9484
Железо	7,86	7,78	31,18	9,591	15,676
Медь	8,9	4,9578	3,6884	7,4727	11,591
Титан	4,51	1,9428	0,6591	1,8090	2,6115

Таблица 4

Материал	γ_2	c_0	c_1	γ_0
Алюминий	0,9643	0,43382	0,54873	1,5
Бериллий	0,5080	0,5644	0,6204	0,8
Железо	4,634	0,3984	0,5306	9,0

Продолжение табл. 4

Материал	ν_2	C_0	C_1	γ_0
Медь	5,5251	0,39493	0,52883	3,6
Титан	1,7984	0,4003	0,5182	1,8

Примечание.

Все коэффициенты, приведенные в табл.3 и 4, имеют размерность в самосогласующейся системе (г - см - мкс). Полученные давления будут измеряться в мегабарах (100 ГПа).

ПРИЛОЖЕНИЕ 3

Программа DISP позволяет проводить расчеты плоского соударения многослойных пластин (с общим числом слоев, не превышающим 20) в диапазоне скоростей удара от 100 до 500 м/с .

Для проведения расчетов в программу должны быть введены следующие группы данных :

- 1) ориентировочное время работы программы на ЭВМ БЭСМ-6 ТСВ;
- 2) данные об условиях соударения и расчетной сетке (общее количество слоев различных материалов - IHS , количество слоев в метаемой пластине - $IHSB$, шаг пространственной сетки H (см), скорость соударения - V_0 (см/мкс), начальный временной шаг расчета - $T1$ (мкс), продолжительность времени моделируемого процесса - TP (мкс);
- 3) данные для организации вывода результатов расчета ($TC21I$, $TC22I$, $TC23I$, $TC24I$, $TC25I$ - шаг времени, с которым запоминается эволюция параметров течения во времени в лагранжевых точках, отстоящих от поверхности соударения на расстояниях $HP1$, $HP2$, $HP3$, $HP4$, $HP5$ (см) соответственно);
- 4) данные, характеризующие материалы в последовательности слоев ($DH(I)$ - толщина слоя (см), считая от свободной поверхности бойка, $R(I)$ - плотность материала (г/см³), $SL(1, I)$ - продольная скорость звука в материале при нормальных условиях, $SL(2, I)$ - модуль сдвига (Мбар), $SLY(1, I)$, $SLY(2, I)$, $SLY(3, I)$, $SLY(4, I)$ - коэффициенты уравнения состояния (Мбар), параметры микродинамической модели для материала данного слоя $B(I)$, $BN(I)$, $BNP(I)$, $CONA(I)$, $TMO(I)$, $FN(I)$, $BD1(I)$, $FP(I)$, $CONB(I)$, $VSD(I)$, $TAO(I)$, $BD2(I)$, $D(I)$).

Эта группа параметров повторяется столько раз (IHS), сколько слоев различных материалов имеется в соударяющихся пластинках.

Ввод данных осуществляется операторами READ18 и READ182. Представление данных определяется соответствующими форматами 18 FORMAT(...) и 182 FORMAT(...)

СПИСОК ОБОЗНАЧЕНИЙ, ПРИНЯТЫХ В ПРОГРАММЕ

Идентификатор	Параметр	Единица измерения	Комментарий
AM(I)	$m_{j+1/2}^0$	г/см ²	Масса ячейки
B(I)	$b \cdot 10^{11}$	см ⁻²	Нормированный модуль вектора Бургерса
BB1(I,J)	t	мкс	Массивы BB1(I,J) - BB14(I,J)
BB2(I,J)	$u_j^{n+1/2}$	см/мкс	предназначены для накопления расчетных значений параметров течения в пяти независимых лагранжевых точках, отстоящих в начальный момент времени от поверхности соударения на расстояние HP1, HP2, HP3, HP4, HP5 ;
BB3(I,J)	$\sigma_j^{n+1/2}$	Мбар	в программе эти массивы содержат значения соответствующих параметров течения в моменты времени BB1(I,J) для точек с номерами J
BB4(I,J)	$\tau_j^{n+1/2}$	Мбар	
BB5(I,J)	$\gamma_j^{n+1/2}$	мкс ⁻¹	
BB6(I,J)	$\dot{\epsilon}_j^{n+1/2}$	мкс ⁻¹	
BB7(I,J)	$\delta_{\kappa j+1/2}^{p, n+1/2}$	-	
BB8(I,J)	$v_j^{n+1/2}$	-	
BB9(I,J)	$p_j^{n+1/2}$	Мбар	
BB10(I,J)	$\delta_{j+1/2}^{p, n+1/2}$	-	
BB11(I,J)	$v_j^{n+1/2}$	см/мкс	
BB12(I,J)	$N_j^{n+1/2}$	см ⁻²	
BB13(I,J)	$f_j^{n+1/2}$	-	
BB14(I,J)	$\delta_{\kappa j+1/2}^{p, n+1/2}$	-	
BD1(I)	$\beta_1 \cdot 10^{12}$	Мбар·см ²	Нормированный коэффициент зарождения дислокаций
BD2(I)	β_2	Мбар	Коэффициент торможения дислокаций
BN(I)	$N_0 \cdot 10^{-12}$	см ⁻²	Нормированная начальная плотность дислокаций
BN1(I)	$N_1 \cdot 10^{-12}$	см ⁻²	Нормированная плотность зародившихся дислокаций
BNP(I)	$N^* \cdot 10^{-12}$	см ⁻²	Нормированная плотность дислокаций

Иденти- фикатор	Параметр	Единица измерения	Комментарий
CHK	K	-	Число Куранта
CONA(I)	$A/g/b$	-	Параметр модели, определя- ющий скорость размножения дислокаций
CONB(I)	$B/g/b$	-	Параметр модели, определя- ющий скорость закрепления подвижных дислокаций
DEF(J)	$(g/g_0 - 1)$	-	
DPD(J)	$f_j^{n+1/2}$	-	Доля подвижных дислокаций
EX(J)	$\dot{\epsilon}_j^{n+1/2}$	мкс ⁻¹	Скорость деформации
FI(J)	f_j^n	-	
FN(I)	f_0	-	Начальная доля подвижных дислокаций
FP(I)	f^*	-	Предельная доля подвижных дислокаций
G2GM	$\dot{\gamma}_j^{P, n+1/2}$	мкс ⁻¹	Скорость сдвиговой пласти- ческой деформации
Gx1(J)	$\gamma_K^P j^{n+1/2}$	-	Кумулятивная сдвиговая пластическая деформация
Gx2(J)	$\gamma_{KR}^P j^{n+1/2}$	-	Реверсивная сдвиговая пластическая деформация
H	H	см	Пространственный шаг сетки
HP1-HP5		см	Координаты лагранжевых то- чек, для которых предполага- ется запоминание параметров течения во времени
J	$J, j+1/2$	-	Номер узла пространственной сетки, счетчик цикла
JC		-	Счетчик цикла

Идентификатор	Параметр	Единица измерения	Комментарий
ЖС1-ЖС5		-	Счетчики циклов
ЖСВ1-ЖСВ4		-	Счетчики циклов
ЖМ		-	Счетчик итераций
I		-	Параметр, определяющий номер набора характеристик материала для очередного слоя
ИЖ(I)		-	Узлы сетки, соответствующие поверхностям раздела материалов в слоях соударяющихся пластин
ИНС		-	Общее количество слоев материалов в соударяющихся пластинах
ИНСВ		-	Число слоев материалов в метаемой пластине
ИР1-ИР2		-	Счетчики
КС, ККС		-	Номер узла, соответствующего правой границе возмущенной области
КНС		-	Номер узла сетки, соответствующий левой границе возмущенной области
L	L_0+1	-	Узел сетки, соответствующий поверхности контакта соударяющихся пластин
L1-L5		-	Счетчики
L7-L8		-	Счетчики
MN		-	Номера узлов сетки, соответствующие лагранжевым точкам НР1, НР2, НР3, НР4, НР5
N	K_0+1	-	Общее количество узлов сетки

Иденти- фикатор	Параметр	Единица измерения	Комментарий
NS, KNS	-	-	Номер узла сетки, соответствующий левой границе возмущенной области
P1(J)	$p_{j+1/2}^n$	Мбар	Давление
PLD(J)	$N_{j+1/2}^{j+1/2}$	см ⁻²	Текущая плотность дислокаций
PMA	0,95 PMAX	Мбар	Максимальное давление во фронте ударной волны
PMAX		Мбар	
PMI	0,2 PMAX	Мбар	
R(I)	ρ_0	г/см ³	Начальная плотность материала
R3(J)	$\eta_{j+1/2}^n$	Мбар	Квадратичная искусственная вязкость
SDS(J)	$v_{j+1/2}^{j+1/2}$	см/мкс	Текущая скорость дислокаций
STRS(J)	$\sigma_{j+1/2}^n$	Мбар	Напряжение
SL(1,I)	c_L	см/мкс	Продольная скорость звука
SL(2,I)	μ	Мбар	Модуль сдвига
SL4(1,I)- -SL4(4,I)	A, B, C	Мбар	Параметры уравнения состояния
T	t	мкс	Время
T1	Δt^n	мкс	Шаг по времени "целый"
T2	$\Delta t^{n+1/2}$	мкс	Шаг по времени "полуцелый"
TAD(I)	τ_0	Мбар	Статический предел текучести
TAU(J)	$\tau_{j+1/2}^{j+1/2}$	Мбар	Сдвиговое напряжение
TC1	-	мкс	Моменты времени, для которых осуществляется выдача результатов расчета в узлах сетки
T11	-	мкс	Период времени, с которым осуществляется выдача результатов расчета в узлах сетки

Идентификатор	Параметр	Единица измерения	Комментарий
TC21-TC25		мкс	Моменты времени, с которых начинается запоминание значений параметров течения в лагранжевых точках HP1, HP2, HP3, HP4, HP5
TC211-TC251		мкс	Периоды времени, с которыми осуществляется запоминание параметров течения в лагранжевых точках HP1, HP2, HP3, HP4, HP5
TC3		мкс	Момент времени, с которого прекращается выдача результатов расчета в узлах сетки
TM(J)	$\bar{\epsilon}$	Мбар	Максимальное сдвиговое напряжение, действовавшее в точке среды до этого момента времени
TCW		мин	Ориентировочное время работы программы на ЭВМ БЭСМ-6
TMD(I)		Мбар	Критическое напряжение гетерогенного зарождения дислокаций
U1(J)	$u_j^{n+1/2}$	см/мкс	Массовая скорость
V0	v_0	см/мкс	Скорость соударения пластин
V3(J)	Δt_j^n	мкс	Шаг по времени, найденный из условия Куранта для узла
VSD(I)	v_0	см/мкс	Сдвиговая скорость звука
X1(J)	x_j^n	см	Координаты узлов пространственной сетки
X2(J)			Максимальное количество итераций в решении уравнения для $\tau_j^{n+1/2}$

Иденти- фикатор	Параметр	Единица измерения	Комментарий
z	g	—	Модуль ориентационного коэффи- циента

Примечание.

Идентификаторы, не вошедшие в список обозначений, служат для вспомогательных вычислений либо являются формальными параметрами подпрограммы функции.


```

1. WPP 235411002300 3CM-
2. BPE/IR 300-
3. ACHV 7-
4. TPAKTB 20-
5. O3W 032-
6. POBTOP-
7. EEB00000000010000000000A3
8. *NAME DISP
9. *FTN
10. *NO LIST
11. *NO LOAD LIST
12. PROGRAM DISP
13. COMMON T2,T1/BL0/T/BL1/X1(1002),X2(1002)/BL2/F1(
13+ 1002)/BL3/AM(1002)
14. 1/BL4/U1(1002)/BL5/STRS(1002)/BL6/V1(1002)/BL7/R3
14+ (1002)/BL8/TAU(100
15. 22)/BL9/EX(1002)/BL10/R1(1002)/BL11/SDS(1002),PLD
15+ (1002),OPD(1002)
16. 3/BL12/V2(1002)/BL13/P1(1002)/BL14/TAU2(1002),G2G
16+ (1002),GG1(1002)
17. 4/BL15/GX1(1002),GX2(1002)
18. 7/BL16/Z,VSD(20),CONA(20),CONB(20),B(20),D(20),BN
18+ P(20),
19. 7BN(20),FP(20),FN(20)
20. 6/BL17/TMC(20),BD1(20),BD2(20),BN1(20),TAQ(20)
21. 9/BL18/TM(1002),TMM(20)/BL19/SL(2,20)/BL20/SLV(4,
21+ 20)
22. 9/BL21/DH(20),R(20),IJS(20)
23. DIMENSION BB1(51,5),BB2(51,5),BB3(51,5),BD4(51,
23+ 5),BB6(51,5),
24. 1BB7(51,5),BB8(51,5),BB9(51,5),BB10(51,5),BD11(51
24+ 5),
25. 2BB12(51,5),BB13(51,5),BB14(51,5),BB5(51,5)
26. TCW=2.45
27. READ 18,IHS,IHSB,H,T1,TP,T11,TC3,V0,
28. 1HP1,HP2,HP3,HP4,HP5,TC211,TC221,TC231,TC241,TC25
28+ 1
29. READ 182,(DH(1),R(1),SL(1,1),SL(2,1),SLV(1,1),SL
29+ V(2,1),
30. 1SLV(3,1),SLV(4,1),D(1),BN(1),BNP(1),CONA(1),TMO(
30+ 1),
31. 1BD1(1),FN(1),FP(1),CONB(1)
32. 1,VSD(1),TAQ(1),BD2(1),D(1),I=1,IHS)
33. 18 FORMAT (I2,I2/F4.4/F6.4/F6.4/F6.4/F6.4/F6.4/F6.4/
33+ 4/5F6.4)
34. 182 FORMAT (F6.4/F6.4/F6.4/F6.4/F8.4/F8.4/F8.4/F8.4/
34+ 2 F6.1/F8.6/F6.3/F8.4/F7.5/F7.5/F7.5/F8.6/F6.4/F5
35. 3/F7.5/F7.5/
35+ 3F7.5)
36. T=0
37. IJS(1)=(DH(1)/H)+1
38. IHS1=IHS-1
39. IJS(2)=(DH(2)/H)+2
40. 1F(IJS(1),EQ.1) IJC(1)=2
41. DO 122 J=2,IHS
42. IJS(J)=IJS(J-1)+DH(J)/H
43. 122 CONTINUE
44. L=IJS(IHSB)
45. N=IJS(IHS)
46. TC1=0
47.

```

```

48.      JC5=1
49.      JC4=1
50.      JC3=1
51.      JC1=1
52.      JC2=1
53.      JC51=0
54.      JC52=0
55.      JC53=0
56.      JC54=0
57.      X1(1)=0
58.      AM(1)=0
59.      TC21=HP1/SL(1,2)
60.      TC22=HP2/SL(1,2)
61.      TC23=HP3/SL(1,2)
62.      TC24=HP4/SL(1,2)
63.      TC25=HP5/SL(1,2)
64.      Z=0,5
65.      CIK=0,25
66.      K=L
67.      T2=T1
68.      C *****БЛОК ВВЕДЕНИЯ ДАННЫХ*****
69.      19 FORMAT (2X,4HHS=,12,5X,5HHSB=,12,5X,2HHP=,F6,4/
70.      12X,3HT1=,F6,4,5X,3HHP=,F5,2,5X,4HT11=,F5,3,5X,4H
71.      TC3=,F5,2/
72.      22X,3HV0=,F8,6,5X,4HHP1=,F6,4,5X,4HHP2=,F6,4,5X,4
73.      HHP3=,F6,4,
74.      35X,4HHP4=,F6,4,5X,4HHP5=,F6,4/
75.      32X,6HTC211=,F6,4,5X,6HTC221=,F6,4,5X,6HTC231=,F6
76.      ,4,5X,
77.      46HTC241=,F6,4,5X,6HTC251=,F6,4)
78.      PRINT 19, HHS, HSB, H, T1, TC, T11, TC3, V0, HP1, HP2, HP
79.      3, HP4, HP5,
80.      1TC211, TC221, TC231, TC241, TC251
81.      192 FORMAT (2X,3HDI=,F6,4,5X,2HHC=,F6,4/5X,8HSL(1,1)=
82.      ,F6,4,5X,
83.      18HSL(2,1)=,F6,4/2X,9HSLV(1,1)=,F6,4,5X,9HSLV(2,1
84.      )=,F6,4,
85.      25X,9HSLV(3,1)=,F6,4,5X,9HSLV(4,1)=,F6,4/
86.      32X,2HB=,F6,1,5X,3HBI=,F6,6,5X,4HBNP=,F6,3,5X,5HC
87.      ONA=,F6,4,
88.      45X,4HTMO=,F7,5,5X,4HBD1=,F7,5/3HFN=,F7,5,5X,3HFP
89.      =,F6,6,
90.      55X,5HCONB=,F8,4/4HVSD=,F5,3,5X,4HTAO=,F7,5,5X,4H
91.      BD2=,F7,5,
92.      65X,2HDI=,F7,5/)
93.      PRINT 192, (DH(I), R(I), SL(1,1), SL(2,1), SLV(1,1), S
94.      LV(2,1),
95.      1SLV(3,1), SLV(4,1), B(I), BN(I), BNP(I), CONA(I), TMO(
96.      I), DD1(I),
97.      2FN(I), FP(I), CONB(I), VSD(I), TAO(I), BD2(I), D(I), I=
98.      1, IHS)
99.      PRINT 192, (DH(I), R(I), SL(1,1), SL(2,1), SLV(1,1), S
100.     LV(2,1),
101.     1SLV(3,1), SLV(4,1), B(I), BN(I), BNP(I), CONA(I), TMO(
102.     I), DD1(I),
103.     2FN(I), FP(I), CONB(I), VSD(I), TAO(I), BD2(I), D(I), I=
104.     1, IHS)
105.     *****БЛОК ЗАДАНИЯ НАЧАЛЬНЫХ УСЛОВИЙ*****

```



```

91.      M=N+1
92.      I=1
93.      DO 109 J=1,M
94.      IF(J.LT.N.AND.J.GT.1JS(I)) I=I+1
95.      R1(J)=R(I)
96.      QPD(J)=FN(I)
97.      TM(J)=TMO(I)
98.      PLD(J)=BN(I)
99.      BN1(I)=BN(I)
100.     TMH(I)=TMO(I)
101. 109 CONTINUE
102.     DO 6 J=1,M
103.     IF(J-L) 7,8,9
104.     7 U1(J)=V0
105.     5 X1(J+1)=X1(J)+H
106.     AM(J+1)=R1(J)*(X1(J+1)-X1(J))
107.     FI(J)=(AM(J+1)+AM(J))/2.
108.     STRS(J)=0
109.     TAU(J)=0.
110.     R3(J)=0.0
111.     V1(J)=1.0
112.     GX1(J)=0
113.     GX2(J)=0
114.     P1(J)=0
115.     G2GN(J)=0
116.     GG1(J)=0
117.     SDS(J)=0.
118.     GO TO 6
119.     8 U1(J)=V0/2
120.     4 CONTINUE
121.     GO TO 5
122.     9 U1(J)=0.
123.     GO TO 4
124.     6 CONTINUE
125.     AM(N+2)=0
126.     FI(N+1)=AM(N+1)/2
127.     STRS(N+2)=0.0
128.     33 CONTINUE
129. С...БЛОК ОПРЕДЕЛЕНИЯ ГРАНИЦ ВОЗМУЩЕННОЙ ОБЛАСТИ МАТЕРИ
129+ АЛД***
130.     I=1
131.     DO 3 J=L,N
132.     IF(ABS(U1(J)).LT.1.E-06) GO TO 30
133.     IF(ABS(U1(J)).GE.1.E-06) KS=J+2
134.     3 CONTINUE
135.     30 IF(KS.GE.N) KS=N
136.     NS=1
137.     IF(L.LE.2) NS=2
138.     KKS=KS+1
139.     KNS=NS
140.     IF(L.LE.2) KNS=NS+1
141. С.....РЕШЕНИЕ УРАВНЕНИЯ ДВИЖЕНИЯ.....
141+ *****
142.     DO 10 J=KNS,KKS
143.     IF(ABS(STRS(J+1)-STRS(J)).LT.1.E-12) GO TO 32
144.     U1(J)=U1(J)+(STRS(J+1)-STRS(J))*T1/FI(J)
145.     GO TO 32
146. С УТОЧН. ФОРМУЛА ДЛЯ КОНТ. УЗЛОВ ВВОДИТСЯ ОПЕРАТОРОМ С
146+ МЕ КОД 31
147.     31 U1(J)=U1(J)+T1*(3*(STRS(J+1)-STRS(J))-(1/3)*(STR

```

```

147. S(J+2)-STRS(J-1))
148. 1/(2*F1(J))
149. 32 CONTINUE
150. IF(ABS(U1(J)),LT,1.E-06) U1(J)=0.0
151. 10 CONTINUE
152. DO 105 J=NS,KKS
153. X1(J)=X1(J)+U1(J)*T2
154. 105 CONTINUE
155. C*****БЛОК РЕШЕНИЯ УРАВНЕНИЯ НЕРАЗРЫВНОСТИ*****
156. *****
157. I=1
158. DO 11 J=NS,K5
159. IF(J,LT,N.AND,J,GE,IJS(1)) I=I+1
160. UPR=U1(J+1)-U1(J)
161. IF(ABS(UPR),LT,1.E-09) UPR=0
162. S=T2*UPR/H
163. IF(S,GT,0.AND,S,LT,1.E-09) S=0
164. IF(S,LE,0.AND,S,GT,-1.E-09) S=0
165. PRO=2*S/(T2*(3+2*V1(J)))
166. DXN=X1(J+1)-X1(J)
167. IF(PRO) 111,112,112
168. 112 BK1=0.0
169. GO TO 113
170. 111 BK1=8.*DXN*PRO
171. C*****УСЛОВИЕ УСТОЙЧИВОСТИ КУРАНТА*****
172. *****
173. 113 V2(J)=(DXN/SQRT(SL(1,1)**2+BK1**2))*CHK
174. IF(T,LT,0.02) GO TO 88
175. IF(B(1),GT,0.0) GO TO 70
176. IF(UPR,LT,0.0.AND,S,LT,0.0) GO TO 88
177. 70 R3(J)=0
178. GO TO 100
179. 88 R3(J)=2+1.2*1.2*R1(J)+UPR*UPR/(2*V1(J)+S)
180. 100 CONTINUE
181. E1=ALOG(V1(J))
182. V1(J)=V1(J)+S
183. E2=ALOG(V1(J))
184. EX(J)=(E2-E1)/T2
185. 11 CONTINUE
186. I=1
187. DO 106 J=NS,K5
188. AM(J+1)=R1(J)+(X1(J+1)-X1(J))/V1(J)
189. F1(J)=(AM(J+1)+AM(J))/2.
190. 106 CONTINUE
191. F1(N+1)=AM(N+1)/2.
192. I=1
193. DO 12 J=NS,K5
194. IF(ABS(EX(J)),LE,1.E-09) EX(J)=0.
195. IF(J,LT,N.AND,J,GE,IJS(1)) I=I+1
196. TKT=TAU(J)
197. G22=G21(J)
198. GX1M=GX1(J)
199. JH=0.
200. C***РЕШЕНИЕ РЕЛАКСАЦИОННОГО УРАВНЕНИЯ ИТЕРАЦИОННЫМ МЕТ
201. ОДОБ***
202. IF(ABS(TAU(J)),LT,TAO(1)) TAUM=TAU(J)+SL(2,1)*T2
203. *EX(J)
204. IF(ABS(TAU(J)),LT,TAO(1)) GO TO 501
205. 500 TAUM=TAU(J)+SL(2,1)*T2*(EX(J)-2.*G2GM(J))
206. IF(B(1),EQ,0.0) GO TO 501

```



```

203. IF(JM.GT.30) GO TO 501
204. JM=JM+1
205. IF(ABS(TAUM-TPL).LE.ABS(0.01-TAUM)) GO TO 501
206. X2(J)=JM
207. TPL=TAUM
208. TAU2(J)=(TAUM+TAU(J))/2
209. G2GM(J)=RATE(GG2,TAU2(J),GX1M,GX2(J),I,J)
210. GG2=GG1(J)+G2GM(J)*T2/2.
211. GX1M=GX1(J)+ABS(G2GM(J))*T2/2.
212. G2GM(J)=RATE(GG2,TAU2(J),GX1M,GX2(J),I,J)
213. GO TO 500
214. 501 CONTINUE
215. GG1(J)=GG1(J)+G2GM(J)*T2
216. GX1(J)=GX1(J)+ABS(G2GM(J))*T2
217. TM(J)=TMM(I)
218. TAU(J)=TAUM
219. IF(D(I).EQ.0.J.AND.TAU(J).GT.TAO(1)) TAU(J)=TAO
219+ (1)
220. ZK1=SIGN(Z,TAU(J))
221. ZK2=SIGN(Z,TKT)
222. IF(ZK1*ZK2.LT.0) GX2(J)=GX1(J)
223. DEF=(1./V1(J))-1
224. C***УРАВН СОСТ. I=1 СООТВЕТСТВУЕТ БОЯКУ, I=2 -ПРЕГ
224+ РАДЕ***
225. P1(J)=SLV(1,1)*DEF+SLV(2,1)*DEF**2+SLV(3,1)*DEF+
225+ *3,SLV(4,1)*DEF**4
226. STRS(J+1)=-(P1(J)+R3(J))+(4./3.)*TAU(J)
227. IF(T.GT.0.05) SDS(J)=X1(J)/T
228. 12 CONTINUE
229. T=T+T1
230. N3=KS-1
231. DO 37 J=3,N3
232. IF(V2(J)-V2(J+1)) 38,37,37
233. 38 V2(J+1)=V2(J)
234. 37 CONTINUE
235. T3=V2(N3)
236. IF(T3.GT.1.1*T2) T3=1.1*T2
237. T1=(T2+T3)/2.
238. T2=T3
239. GO TO 86
240. C ОРГАНИЗАЦИЯ ВЫВОДА ПРОФИЛЯ ПЛОСКОЙ ВОЛНЫ ОТ ВРЕМЕНИ
240+ ДЛЯ ФИКСИРОВАННОЙ
241. C*****ЛАГРАНЖЕВОЙ ТОЧКИ С КООРДИНАТОМ НР1*****
241+ *
242. 80 I1=1
243. MN=L+NR1/H
244. JC1=JC1+1
245. TC21=TC21+TC211
246. JC=JC1
247. IF(JC1.GE.51) GO TO 86
248. GO TO 85
249. 81 I1=2
250. MN=L+NR2/H
251. TC22=TC22+TC221
252. JC2=JC2+1
253. JC=JC2
254. IF(JC2.GE.51) GO TO 86
255. GO TO 85
256. 82 I1=3
257. MN=L+NR3/H

```

```

258. TC23=TC23+TC231
259. JC3=JC3+1
260. JC=JC3
261. IF(JC3,GE,51) GO TO 86
262. GO TO 85
263. 83 II=4
264. MH=L+HP4/H
265. TC24=TC24+TC241
266. JC4=JC4+1
267. JC=JC4
268. IF(JC4,GE,51) GO TO 86
269. GO TO 85
270. 84 II=5
271. MH=L+HP5/H
272. TC25=TC25+TC251
273. JC5=JC5+1
274. JC=JC5
275. IF(JC5,GE,51) GO TO 86
276. 85 CONTINUE
277. С ПАРАМЕТРЫ, КОТОРЫЕ ВЫВОДЯТСЯ ДЛЯ ФИКСИРОВАННОЙ ПАТРА
277. НЖЕДОЯ ТОЧКИ
278. BB1(JC,II)=1
279. BB2(JC,II)=U1(MN)
280. BB3(JC,II)=STR3(MN)
281. BB4(JC,II)=TAU(MN)
282. BB5(JC,II)=G20R(MN)
283. BB6(JC,II)=EX(MN)
284. BB7(JC,II)=GX1(MN)
285. BB8(JC,II)=V1(MN)
286. BB9(JC,II)=P1(MN)
287. BB10(JC,II)=GG1(MN)
288. BB11(JC,II)=SDS(MN)
289. BB12(JC,II)=PLD(MN)
290. BB13(JC,II)=ORD(MN)
291. BB14(JC,II)=GX2(MN)
292. IF(JC1,EQ,51) GO TO 61
293. IF(JC2,EQ,51) GO TO 62
294. IF(JC3,EQ,51) GO TO 63
295. IF(JC4,EQ,51) GO TO 64
296. IF(JC5,EQ,51) GO TO 65
297. 86 CONTINUE
298. IF(T,GT,TC21) GO TO 80
299. IF(T,GT,TC22) GO TO 81
300. IF(T,GT,TC23) GO TO 82
301. IF(T,GT,TC24) GO TO 83
302. IF(T,GT,TC25) GO TO 84
303. 48 CONTINUE
304. IF(T,GT,TC1.AND,T,LT,TC3) GO TO 56
305. GO TO 48
306. 56 CONTINUE
307. С ОРГАНИЗАЦИЯ ВЪВОДА ПРОФИЛЯ ДЛЯ ФИКСИРОВАННОГО МОМЕНТА
307. ВРЕМЕНИ TC1
308. TC1=TC1+T11
309. PMAX=0
310. PMIN=1
311. DO 400 J=1,N
312. IF(PMAX,LT,ABS(P1(J))) PMAX=P1(J)
313. 400 CONTINUE
314. DO 401 J=1,L
315. IF(PMIN,GT,ABS(P1(J))) PMIN=P1(J)

```



```

316. 401 CONTINUE
317. P1A=0.95*P1MAX
318. PM1=0.2*P1MAX
319. IP1=0
320. IP2=0
321. DO 402 J=1,N
322. IF(IP1.EQ.1) GO TO 403
323. IF(P1(J).GT.PM1) L7=J
324. IF(P1(J).GT.PM1) IP1=1
325. 403 CONTINUE
326. IF(IP2.EQ.1) GO TO 404
327. IF(P1(J).GT.P1MA) L8=J
328. IF(P1(J).GT.P1MA) IP2=1
329. 404 CONTINUE
330. IF(P1(J).GT.P1MA) L1=J-10
331. IF(L1.LE.2) L1=2
332. IF(P1(J).GT.0.00001) L2=J
333. 402 CONTINUE
334. L3=1
335. IF(T.GT.0.3) L3=2
336. IF(T.GT.1.6) L3=3
337. 301 CONTINUE
338. KL=L1-1
339. C.....ВЫВОД ПАРАМЕТРОВ ТЕЧЕНИЯ В ЛАГРАНЖЕВЫХ ТОЧКАХ...
339. *****
340. 23 FORMAT(1H1,8HВРЕМЯ = ,E11.4,2H ШАГИ ПО ВРЕМЕНИ
340. : ,E11.4,13H ПОЛУЧЕНЫ = ,E11.4/10X,7HГ.В. ,E15.6
341. ,E15.6)
342. 55 FORMAT (14,9E10.4)
343. 43 FORMAT (1H1,9X,2HХ1,9X,2HУ1,8X,4HSTRS,8X,
344. 13HTAU,9X,2HEX,8X,4HG2GM,8X,3HSDS,8X,3HPLD,
345. 27X,3HGG1,7X,3HGG1/(2X,1,10E11.2))
346. PRINT 23,T,T1,T2,STRS(1),STRS(N*2)
347. IF(L1.GE. N-10) GO TO 47
348. PRINT 43,(J,X1(J),U1(J),STRS(J+1),TAU(J),EX(J),
349. 1G2GM(J),SDS(J),PLD(J),GG1(J), X2(J),J=L1,L2,L3)
350. IF(L1.LT.N-1) GO TO 46
351. 47 CONTINUE
352. HL7=L7
353. HL8=L8
354. IF(ABS(HL7-HL8).LT.1) GO TO 46
355. PRINT 43,(J,X1(J),U1(J),STRS(J+1),TAU(J),EX(J),
356. 1G2GM(J),SDS(J),PLD(J),GG1(J), X2(J),J=L7,L8,L3)
357. 46 CONTINUE
358. GO TO 53
359. 61 CONTINUE
360. 54 FORMAT (7E14.6)
361. L4=1
362. L5=51
363. IF(JC1.LT.51) L5=JC1
364. IF(JCB1.EQ.1) GO TO 62
365. PRINT 54,(BB1(J,1),BB2(J,1),BB3(J,1),BB4(J,1),B
365. B5(J,1),BB6(J,1),
366. 1BB7(J,1),BB8(J,1),BB9(J,1),BB10(J,1),BB11(J,1),B
366. B12(J,1),
367. 1BB13(J,1),BB14(J,1),J=L4,L5)
368. JCB1=1
369. IF(IPR.NE.1) GO TO 86
370. 62 CONTINUE

```

```

371.      L4=1
372.      L5=51
373.      IF(JC2,LT,51) L5=JC2
374.      IF(JCB2,EQ,1) GO TO 63
375.      PRINT 54,( BB1(J,2),BB2(J,2),BB3(J,2),BB4(J,2),B
375+ B5(J,2),BB6(J,2),
376.      BB7(J,2),BB8(J,2),BB9(J,2),BB10(J,2),BB11(J,2),B
376+ B
377.      I12(J,2),BB13(J,2),BB14(J,2),J=L4,L5)
378.      JCB2=1
379.      IF(IPR,NE,1) GO TO 86
380.      63 CONTINUE
381.      L4=1
382.      L5=51
383.      IF(JC3,LT,51) L5=JC3
384.      IF(JCB3,EQ,1) GO TO 64
385.      PRINT 54,( BB1(J,3),BB2(J,3),BB3(J,3),BB4(J,3),B
385+ B5(J,3),BB6(J,3),
386.      BB7(J,3),BB8(J,3),BB9(J,3),BB10(J,3),BB11(J,3),B
386+ B12(J,3),
387.      BB13(J,3),BB14(J,3),J=L4,L5)
388.      JCB3=1
389.      IF(IPR,NE,1) GO TO 86
390.      64 CONTINUE
391.      L4=1
392.      L5=51
393.      IF(JC4,LT,51) L5=JC4
394.      IF(JCB4,EQ,1) GO TO 65
395.      PRINT 54,( BB1(J,4),BB2(J,4),BB3(J,4),BB4(J,4),
395+ B5(J,4),BB6(J,4),
396.      BB7(J,4),BB8(J,4),BB9(J,4),BB10(J,4),BB11(J,4),
396+ B12(J,4),BB13(J,4)
397.      ),BB14(J,4),J=L4,L5)
398.      JCB4=1
399.      IF(IPR,NE,1) GO TO 86
400.      65 CONTINUE
401.      L4=1
402.      L5=51
403.      IF(JC5,LT,51) L5=JC5
404.      PRINT 54,( BB1(J,5),BB2(J,5),BB3(J,5),BB4(J,5),B
404+ B5(J,5),BB6(J,5),
405.      BB7(J,5),BB8(J,5),BB9(J,5),BB10(J,5),BB11(J,5),
405+ B12(J,5),BB13(J,5)
406.      ),BB14(J,5),J=L4,L5)
407.      GO TO 600
408.      53 CONTINUE
409.      CALL CTIME (RT,CT)
410.      IF(CT,GT,TCW*60.) IPR=1
411.      IF(CT,GT,TCW*60.) GO TO 61
412.      IF(T,LT,TP) GO TO 33
413.      600 CONTINUE
414.      STOP
415.      END
416.      FUNCTION RATE(V1,V2,V3,V4,I,J)
417.      COMMON /BL11/SDS(1002),PLD(1002),DPD(1002)
418.      1/BL16/Z,VSD(20),EONA(20),CONB(20),B(20),D(20),BN
418+ B(20),
419.      1BN(20),FF(20),FN(20)
420.      2/BL17/TMO(20),BD1(20),BD2(20),BH1(20),TAO(20)
421.      3/BL18/TM(1002),TMM(20),BL19/SL(2,20)

```



```

422. IF(ABS(V2).LT.1.E-08) GO TO 503
423. IF(ABS(V2).GT.TM(J)) TMM(I)=ABS(V2)
424. IF(ABS(V2).LT.TM(J)) TMM(I)=TM(J)
425. TAO(I)=TAO(I)
426. IF(TMO(I).GT.TAO(I)) TAOH=TMO(I)
427. IF(ABS(V2).LT.TAOH) GO TO 503
428. TVM=ABS(TMM(I)-TMO(I))
429. IF(TVM.GT.1.E-12) BN1(I)=BN(I)+TVM/BD1(I)
430. IF(TVM.LE.1.E-12) BN1(I)=BN(I)
431. IF((ABS(V3)*CONA(I)).LT.20) GO TO 510
432. 521 PLD(J)=BNP(I)
433. GO TO 512
434. 510 PLD(J)=BNP(I)+(AN1(I)-BNP(I))/EXP(ABS(V3)*CONA(I)
434+ )
435. 512 IF((DPD(J)-FP(I)).LT.1.E-08) GO TO 520
436. IF((ABS(V3-V4)*CONB(I)).LT.20.) GO TO 511
437. 520 DPD(J)=FP(I)
438. GO TO 513
439. 511 DPD(J)=FP(I)+(FN(I)-FP(I))/EXP(ABS(V3-V4)*CONB(I)
439+ )
440. 513 CONTINUE
441. Z1=SIGN(Z,V2)
442. TAO1=TAO(I)+D(I)*SQRT(PLD(J)*10.)
443. TV=ABS(ABS(V2)-TAO1)
444. IF(TV.GT.1.E-08) ST=TV/BD2(I)
445. IF(TV.LE.1.E-08) ST=0.
446. SCSR=VSD(I)*ST*ST/(1+ST*ST)
447. R=SCSR*DPD(J)*PID(J)*Z1*BI1
448. GO TO 504
449. 514 CONTINUE
450. R=B(I)*VSD(I)*0.5*(TMM(I)-TMO(I))*SIGN(Z,V2)/BD1
450+ (I)
451. GO TO 504
452. 503 R=0.
453. 504 RATE=R
454. RETURN
455. END
456. *CALL YESMEMORY
457. *E ECLTE
458. 2 1
459. 1.002
460. 0.6002
461. 3.5
462. 0.1
463. 2.0
464. 0.0475
465. 1.00 1.45 1.5 1.5 1.
466. 0.01 0.02 0.01 0.05 0.
467. 0.
468. 8.95
469. .476
470. 0.548
471. 1.407
472. 2.871
473. 2.335
474. 0.2
475. 2560.0
476. 0.0003
477. 10.0
478. 0.2

```

479. 0.00116
480. 0.1
481. 1.0
482. 1.0
483. 0.0
484. 0.322
485. 0.00004
486. 0.002
487. 0.25
488. 0.5
489. 7.86
490. 0.502
491. 0.86
492. 1.61
493. 3.56
494. 3.68
495. 0.0
496. 2500.
497. 0.00003.
498. 10.
499. 0.05
500. 0.00556
501. 0.1
502. 1.0
503. 1.0
504. 0.0
505. 0.322
506. 0.006
507. 0.02
508. 2.00005
509. *END FILE

ЛИТЕРАТУРА

1. Аверченков В.И., Камаев В.А. Основы построения САПР. - Волгоград : Изд - во ВПИ, 1984. - 120 с.
2. Норенков И.П. Введение в автоматизированное проектирование технических устройств и систем. - М. : Высш. шк., 1986. - 304 с.
3. Системы автоматизированного проектирования : В 9 кн. Кн. 1. Принципы построения и структура / И.П. Норенков. - М. : Высш. шк., 1986. - 127 с.
4. Системы автоматизированного проектирования : В 9 кн. Кн. 2. Технические средства и операционные системы / Д.М. Жук, В.А. Мартынюк, П.А. Сомов; Под ред. И.П. Норенкова. - М. : Высш. шк., 1986. - 159 с.
5. Системы автоматизированного проектирования : В 9 кн. Кн. 3. Информационное и прикладное программное обеспечение / В.Г. Федорук, В.М. Черненький ; Под ред. И.П. Норенкова. - М. : Высш. шк., 1986. - 159 с.
6. Землянский А.А., Персиц М.Г. Основы операционной системы ЕС ЭВМ. - М. : Сов. радио, 1980. - 144 с.
7. Средства отладки программ в ОС ЕС ЭВМ : Справочное пособие / В.И. Ерофеев, Ю.П. Меркушов, В.И. Першиков, А.П. Соколов. - М. : Статистика, 1979. - 245 с.
8. Эшли Р., Фернандес Д. Язык управления заданиями. - М. : Мир, 1981. - 172 с.
9. Козлов Н.И. Организация вычислительных работ. - М. : Наука, 1981. - 240 с.
10. Агафонов В.Н. Языки и средства спецификации программ. - В кн. : Требования и спецификации в разработке программ. М. : Мир, 1984, с. 285 - 344.
11. Вельбицкий И.В., Ходаковский В.Н., Шолмов Л.И. Технологический комплекс производства программ на машинах ЕС ЭВМ и БЭСМ-6. - М. : Статистика, 1980. - 263 с.
12. Салтыков А.И., Семашко Г.Л. Программирование для всех. - М. : Наука, 1986. - 176 с.
13. Роуч П. Вычислительная гидродинамика. - М. : Мир, 1980. - 616 с.
14. Средства отладки больших систем / Под ред. Р. Растина. - М. : Статистика, 1977. - 240 с.

15. Ионов В.Н., Огибалов П.М. Напряжения в телах при импульсивном нагружении . - М.: Высш. шк. , 1975. - 463 с.
16. Кукуджанов В.Н., Кондауров В.И. Численное решение неоднородных задач динамики твердого деформируемого тела.-В кн.: Проблемы динамики упруго-пластических сред . М.: Мир, 1975 , с. 39 - 85 .
17. Динамика удара /Дж. А. Зукас, Т. Николас, Х.Ф. Свифт и др. - М.: Мир, 1985. - 296 с.
18. Степанов Г.В. Упруго-пластическое деформирование материалов под действием импульсных нагрузок . - Киев : Наук. Думка , 1979 . - 266 с.
19. Коларов Д., Балтов А., Бончева Н. Механика пластических сред. - М.: Мир, 1979. - 302 с.
20. Колтунов М.А., Кравчук А.С., Майборода В.П. Прикладная механика деформируемого твердого тела. - М.: Высш. шк., 1983 . - 349 с.
21. Майборода В.П., Кравчук А.С., Холлин Н.Н. Скоростное деформирование конструкционных материалов. - М.: Машиностроение, 1986. - 264 .
22. Никифоровский В.С., Шеминин Е.И. Динамическое разрушение твердых тел. - Новосибирск : Наука , 1979 . - 270 с.
23. Пэжина П., Савчук А. Проблемы термопластичности.-В кн.: Проблемы теории пластичности и ползучести . М.: Мир, 1979 , с.94 - 203 .
24. Работнов Ю.Н. Механика деформируемого твердого тела . - М. : Наука, 1979. - 744 с.
25. Ударные волны и явления высокоскоростной деформации металлов. - М. : Металлургия , 1984. - 512 с.
26. Нигматуллин Р.И. Основы механики гетерогенных сред. - М. : Наука, 1978. - 336 с.
27. Механика разрушения. Быстрое разрушение, остановка трещин . /Дж. Ирвин, Н. Кальтхофф, И. Бейнерт и др. - М.: Мир, 1981. - 254 с.
28. Макаров П.В., Платова Т.М., Скрипняк В.А. О пластическом деформировании и микроструктурных превращениях металлов в ударных волнах. - ФТБ, 1983, № 5, с. 123 - 126 .

29. Уилкинс М.Л. Расчет упруго-пластических течений. - В кн.: Вычислительные методы в гидродинамике. М.: Мир, 1967, с. 212 - 263 .
30. Рани Т. Численное моделирование явлений при высокоскоростном ударе. - В кн.: Высокоскоростные ударные явления. М.: Мир, 1973, с. 164 - 199 .
31. Платова Т.М. Динамические задачи механики деформируемых сред. - Томск : Изд-во ТТУ, 1980. - 106 с.
32. Хорев И.Е., Горельский В.А., Югов Н.Т. Численное исследование физических особенностей трехмерной задачи удара деформируемого тела о препятствие. - М.: ДАН СССР, 1985, т.283, № 3, с. 612 - 615.
33. Глазырин В.П., Макаров П.В., Платова Т.М. Структура ударных волн первичного и вторичного сжатия. - В кн.: Матер. Всесоюз. симпов. "Физико-химические превращения в ударных волнах." - Черногоровка; АН СССР, ОИХФ, 1978, с. 73 - 82.
34. Глазырин В.П., Платова Т.М., Макаров П.В., Скрипник В.А. Структура волновых фронтов и проявления вязких свойств металлов в стационарных и нестационарных ударных волнах. - В кн.: Матер. Первого Всесоюз. симпов. по макроскопической кинетике и химической газодинамике. - Алма-Ата, 1984, с. 124 - 126.
35. Жданов В.А., Жуков А.В. Термодинамические полные уравнения состояния металлов (твердая фаза). - ПМТФ, № 5, 1978, с. 86 - 89.
36. Альтшулер Л.В., Чекин Б.С. Реология волновой деформации металлов. - ФГВ, 1983, № 5, с. 140 - 142 .

ОГЛАВЛЕНИЕ

Введение	3
ГЛАВА 1. Структура и принципы построения систем автоматизированного проектирования	
1.1. Комплекс средств автоматизации проектирования	5
1.2. Классификация САПР	10
1.3. Структура САПР	13
1.4. Принципы построения САПР	15
ГЛАВА 2. Программное обеспечение САПР	
2.1. Структура программного обеспечения САПР	17
2.2. Общее программное обеспечение САПР	17
2.3. Структура операционных систем	20
2.4. Последовательность обработки задания на ЭВМ	22
2.5. Организация ОС ЕС ЭВМ	25
2.6. Язык управления заданиями ОС ЕС ЭВМ	30
ГЛАВА 3. Специальное программное обеспечение САПР	
3.1. Структура специального программного обеспечения САПР	36
3.2. Принципы организации пакетов прикладных программ	39
3.3. Организация рабочих программ в САПР	41
3.4. Язык пакета прикладных программ	45
ГЛАВА 4. Технология разработки специального программного обеспечения САПР	
4.1. Разработка СПО САПР	47
4.2. Технология программирования модулей	49
4.3. Отладка и тестирование программных модулей	53
4.4. Выпуск программной документации	54
ГЛАВА 5. Разработка прикладного программного обеспечения САПР для расчетов конструкций на прочность при динамическом нагружении	
5.1. Проблемы расчета конструкций на прочность в условиях динамического нагружения	58
5.2. Математическая постановка динамических задач	61
5.3. Выбор модели среды	65
5.4. Выбор модели разрушения	74
ГЛАВА 6. Разработка программного комплекса для расчетов распространения плоских волн напряжений в твердых телах	
6.1. Выработка требований к программному комплексу	82

6.2. Математическая постановка задачи	85
6.3. Метод решения	89
6.4. Разработка алгоритма и структуры программного комплекса	93
6.5. Отладка и тестирование	96
ПРИЛОЖЕНИЕ 1.	100
ПРИЛОЖЕНИЕ 2.	104
ПРИЛОЖЕНИЕ 3.	107
ЛИТЕРАТУРА	124

Татьяна Миновна Платова
Владимир Альбертович Скрипняк

Системы автоматизированного проектирования
конструкций с учетом динамической прочности

Учебное пособие

Редактор Г.Г. Иванова

ИБ 1966 Подписано к печати 8.11.87
КЗ 06171 Формат 60 x 84 1/16. Бумага
 типографская №3. Печать офсетная. Печ. л. 8,1.
Усл. печ. л. 7,7. Уч - изд. л. 6,8. Тираж 300 экз.
Заказ 729 Цена 25 к.

Издательство ТТУ, 634029, Томск, ул. Никитина, 4

1-678200

Томский государственный университет



Научная библиотека 00229689



Издательство
Томского
университета