



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ БЕЛАРУСЬ**

**Белорусский национальный
технический университет**

Кафедра «Микро- и нанотехника»

Е. Н. Щербакова

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
НАНО- И МИКРОСИСТЕМНОЙ
ТЕХНИКИ**

Учебно-методическое пособие

Минск
БНТУ
2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Микро- и нанотехника»

Е. Н. Щербакова

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ НАНО- И МИКРОСИСТЕМНОЙ ТЕХНИКИ

Учебно-методическое пособие
для студентов специальности 1-38 01 04
«Микро- и наносистемная техника»

*Рекомендовано учебно-методическим объединением
по образованию в области приборостроения*

Минск
БНТУ
2017

УДК 621.382.049.77:004(075.8)

ББК 32.81я7

Щ61

Рецензенты:

кафедра электронной техники и технологии
Белорусского государственного университета информатики
и радиоэлектроники (зав. кафедрой *А. П. Достанко*);
декан факультета непрерывного и дистанционного обучения
Белорусского государственного университета информатики
и радиоэлектроники, доц., канд. техн. наук *В. М. Бондарик*

Щербакова, Е. Н.

Щ61 Информационные технологии nano- и микросистемной техники : учебно-методическое пособие для студентов специальности 1-38 01 04 «Микро- и наносистемная техника» / Е. Н. Щербакова. – Минск: БНТУ, 2017. – 78 с.

ISBN 978-985-550-608-0.

В учебно-методическом пособии представлены материалы из лекционного и практического курса дисциплины: общие характеристики процессов сбора, передачи, обработки и хранения информации; организация информационных процессов; информационные технологии при выборе решений в области технологии nano- и микросистемной техники, организации, планирования и управления производством, а также автоматизированные системы научных исследований, САПР, МЭМС и НЭМС, технологии компьютерного моделирования nano- и микросистемной техники.

УДК 621.382.049.77:004(075.8)

ББК 32.81я7

ISBN 978-985-550-608-0

© Щербакова Е. Н., 2017

© Белорусский национальный
технический университет, 2017

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ.....	5
ВВЕДЕНИЕ.....	6
1. КРАТКИЙ КУРС ЛЕКЦИЙ.....	7
1.1. Определение, классификация и инструментарий информационных технологий. Информационная технология и информационная система	7
1.2. Общая характеристика процессов сбора, передачи, обработки и хранения информации. Контроль достоверности данных	8
1.3. Понятия «объективность», «полнота», «достоверность», «адекватность», «доступность» и «актуальность» информации. Технология обеспечения безопасности компьютерных систем. Проблемы использования информационных технологий	12
1.4. Информационная технология поддержки принятия решений. Основные компоненты.....	15
1.5. Экспертные системы и их типы. Основные способы формализованного представления знаний в нано- и микросистемной технике. Области применения экспертных систем.....	17
1.6. Автоматизированные системы научных исследований. Типовая структура.....	22
1.7. Общие сведения о системах автоматизированного проектирования. Типовая структура. Определение и основное назначение CAD- / CAE- / CAM-систем.....	24
1.8. Основы автоматизации проектирования микроэлектромеханических и нанозлектромеханических систем. Методология их проектирования.....	26
1.9. Интегрированные пакеты автоматизированного проектирования МЭМС и НЭМС	27
1.10. Общие сведения о компьютерном математическом моделировании. Классификация математических моделей. Имитационное моделирование.....	31
1.11. Этапы, цели и средства компьютерного математического моделирования.....	32
1.12. Особенности имитационного моделирования производственных систем.....	35

1.13. Основные параметры, определяющие течение и состояние технологического процесса производства компонентов нано- и микросистемной техники	36
1.14. Система моделей в технологии нано- и микросистемной техники	38
1.15. Понятие о вычислительном технологическом эксперименте.....	40
1.16. Общая характеристика технологии создания программного обеспечения. Этапы разработки программного обеспечения	43
1.17. Современные методы разработки ПО: метод нисходящего проектирования, модульное проектирование, CASE-технологии	48
2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ	53
Практическое занятие № 1. МЕТОДЫ И ПРОГРАММЫ ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ НАНОСТРУКТУР	53
Практическое занятие № 2. ОСНОВНЫЕ ПРИНЦИПЫ РАБОТЫ ХМД.....	63
Практическое занятие № 3. МОДЕЛИРОВАНИЕ ПЛАВЛЕНИЯ МЕТАЛЛИЧЕСКИХ НАНОЧАСТИЦ И КЛАСТЕРОВ.....	69
Практическое занятие № 4. МОДЕЛИРОВАНИЕ НАНОКРИСТАЛЛИЗАЦИИ АМОРФНЫХ МЕТАЛЛОВ ПРИ ДЕФОРМАЦИИ.....	75
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	78

СПИСОК СОКРАЩЕНИЙ

- CAD (computer aided design) – системы инженерной графики;
CAE (computer aided engineering – системы инженерных расчетов;
CAM (computer aided manufacturing) – системы автоматизации подготовки и управления производством;
EDA (electronic design automation) – автоматизированное проектирование электронных средств;
АСНИ – автоматизированные системы научных исследований;
ВТ – вычислительная техника;
ВТЭ – вычислительный технологический эксперимент;
МЭМС – микроэлектромеханические системы;
НЭМС – наноэлектромеханические системы;
ПО – программное обеспечение;
САПР – система автоматизированного проектирования;
СБИС – сверхбольшая интегральная микросхема;
ТП – технологический процесс;
ЭВМ – электронно-вычислительная машина.

ВВЕДЕНИЕ

Дисциплина «Информационные технологии нано- и микросистемной техники» является частью специальной подготовки инженера-электромеханика. Данное пособие поможет изучить общие характеристики процессов сбора, передачи, обработки и хранения информации, организацию информационных процессов, информационные технологии при выборе решений в области технологии нано- и микросистемной техники, организации, планирования и управления производством, а также автоматизированные системы научных исследований, системы автоматизированного проектирования МЭМС и НЭМС, технологии компьютерного моделирования нано- и микросистемной техники.

Усложнение разрабатываемых устройств микро- и нанотехники приводит к тому, что подготовка специалистов по производству и проектированию нано- и микросистемной техники должна включать такие важные составляющие формирования базового набора компетенций, как знание теоретических основ информационных технологий, методов моделирования и проектирования компонентов микро- и нанотехники, а также наличие практических навыков использования современных средств автоматизированного проектирования МЭМС и НЭМС, моделирования наноструктур и нанотехнологий.

Пособие состоит из двух частей. В первой, теоретической части, приведен основной материал лекционного курса. Во второй части собраны методические указания к выполнению практических работ. При подготовке учебного материала была использована литература, перечисленная в списке использованных источников [1–10], проанализированы разнообразные источники в сети Internet. Полученная информация была систематизирована и обобщена.

Учебно-методическое пособие предназначено для студентов специальности 1-38 01 04 «Микро- и наносистемная техника».

1. КРАТКИЙ КУРС ЛЕКЦИЙ

1.1. Определение, классификация и инструментарий информационных технологий. Информационная технология и информационная система

Информационная технология (ИТ) – совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта).

Цель информационной технологии: производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-либо действия.

Глобальная информационная технология включает модели, методы и средства, формализующие и позволяющие использовать информационные ресурсы общества.

Базовая информационная технология предназначена для определенной области применения (производство, научные исследования, обучение и т. д.).

Конкретные информационные технологии реализуют обработку данных при решении функциональных задач пользователей (например, задачи учета, планирования, анализа).

На данный момент существуют различные подходы к проблеме классификации информационных технологий. Приведем некоторые из них. В одних работах выделяют следующие виды информационных технологий:

- информационная технология обработки данных;
- информационная технология управления;
- автоматизация офиса;
- информационная технология поддержки принятия решений;
- информационная технология экспертных систем.

Другие источники информационные технологии подразделяют на различные виды следующим образом:

- функционально-ориентированные информационные технологии, предназначенные для реализации определенных задач;
- предметно-ориентированные информационные технологии, предназначенные для решения конкретных задач в определенной предметной области;

– проблемно-ориентированные информационные технологии, предназначенные для решения типовых прикладных задач.

Инструментарий информационной технологии – один или несколько взаимосвязанных программных продуктов для определенного типа компьютера, технология работы в котором позволяет достичь поставленную пользователем цель.

Информационная система – человеко-компьютерная система для поддержки принятия решений и производства информационных продуктов, использующая компьютерную информационную технологию.

1.2. Общая характеристика процессов сбора, передачи, обработки и хранения информации. Контроль достоверности данных

Сбор информации. Система сбора информации представляет собой сложный программно-аппаратный комплекс. Как правило, современные системы сбора информации не только обеспечивают кодирование информации и ее ввод в ЭВМ, но и выполняют предварительную (первичную) обработку этой информации.

Сбор информации – это процесс получения информации из внешнего мира и приведение ее к виду, стандартному для данной информационной системы. Обмен информацией между воспринимающей информацией системой и окружающей средой осуществляется посредством сигналов.

Сигнал можно определить как средство перенесения информации в пространстве и времени. В качестве носителя сигнала в технологии компонентов микро- и нанотехники могут выступать электрический ток, свет, магнитное поле и т. п. Технические системы для приема сигналов оснащаются специальными устройствами. Вне зависимости от носителя информации (сигнала) типичный процесс обработки сигнала может быть охарактеризован следующими шагами. На первом шаге исходный (первичный) сигнал с помощью специального устройства (датчика) преобразуется в эквивалентный ему электрический сигнал (электрический ток). На втором шаге вторичный (электрический) сигнал в некоторый выделенный момент времени оцифровывается специальным устройством – аналого-цифровым преобразователем (АЦП). АЦП значению электрического сигнала ставит в соответствие некоторое число из конечного множества

таких чисел. Таким образом, датчик и АЦП, связанные вместе, составляют цифровой измерительный прибор. Если этот прибор оснастить некоторым устройством для хранения измеренной величины – регистром, – то на следующем шаге по команде от ЭВМ можно ввести это число в машину и подвергать затем любой необходимой обработке.

Передача информации. Взаимодействие между территориально удаленными объектами осуществляется за счет обмена данными. Доставка данных производится по заданному адресу с использованием сетей передачи данных. В современных условиях большое распространение получила распределенная обработка информации, при этом сети передачи данных превращаются в информационно-вычислительные.

В качестве простейшего способа повышения достоверности передачи информации может использоваться контроль на четность. Суть этого способа заключается в следующем. На входе в канал связи производится подсчет числа «1» в двоичной кодовой последовательности – входном сообщении. Если число «1» оказывается нечетным, в хвост передаваемого сообщения добавляется «1», а если нет, то «0». На принимающем конце канала связи производят аналогичный подсчет, и если контрольная сумма оказывается нечетной, то делается вывод о том, что при передаче произошло искажение информации, в противном случае принятая информация признается правильной (неискаженной). В описанном способе используется один добавочный контрольный разряд. Это позволяет обнаруживать ошибку передачи в случае искажения одного-единственного разряда в сообщении. Этот очень простой способ применяют при передаче данных на большие расстояния. В тех случаях, когда вероятность искажения информации при передаче велика, требуются более сложные методы. Так помехоустойчивые коды позволяют не только принимать решение о правильности передачи информации, но и в ряде случаев производить ее исправление. При контроле на четность единственный способ получить достоверную информацию – повторная передача сообщения. В случае корректирующих кодов, что очень важно при высокой стоимости передачи, имеется возможность исправлять ошибки на принимающем конце канала связи, избегая, таким образом, повторной передачи информации.

Обработка информации. *Технология электронной обработки информации* – человеко-машинный процесс исполнения взаимосвязанных операций, протекающих в установленной последовательности с целью преобразования исходной (первичной) информации в результатную. Операция представляет собой комплекс совершаемых технологических действий, в результате которых информация преобразуется. Технологические операции разнообразны по сложности, назначению, технике реализации, выполняются на различном оборудовании, многими исполнителями. В условиях электронной обработки данных преобладают операции, выполняемые автоматически на машинах и устройствах, которые считывают данные, выполняют операции по заданной программе в автоматическом режиме при участии человека, или сохраняя за пользователем функции контроля, анализа и регулирования.

Построение технологического процесса производства компонентов микро- и наносистемной техники определяется следующими факторами: особенностями обрабатываемой информации, ее объемом, требованиями срочности и точности обработки, типами, количеством и характеристиками применяемых технических средств. Они ложатся в основу организации технологии, которая включает установление перечня, последовательности и способов выполнения операций, порядка работы специалистов и средств автоматизации, организацию рабочих мест, установление временных регламентов взаимодействия и т. п. Организация технологического процесса должна обеспечить его экономичность, комплексность, надежность функционирования, высокое качество работ. Это достигается использованием системотехнического подхода к проектированию технологии. При этом имеет место комплексное взаимосвязанное рассмотрение всех факторов, путей, методов построения технологии, применение элементов типизации и стандартизации, а также унификации схем технологических процессов.

Различают два основных типа организации технологических процессов:

- предметный;
- пооперационный.

Предметный тип организации технологии предполагает создание параллельно действующих технологических линий, специализирующихся на обработке информации и решении конкретных

комплексов задач и организующих пооперационную обработку данных внутри линии.

Пооперационный (поточный) тип построения технологического процесса предусматривает последовательное преобразование обрабатываемой информации согласно технологии, представленной в виде непрерывной последовательности сменяющих друг друга операций, выполняемых в автоматическом режиме.

Хранение информации вызвано многократным ее использованием, применением постоянной информации, необходимостью комплектации первичных данных до их обработки. Осуществляется на машинных носителях в виде информационных массивов, где данные располагаются по установленному в процессе проектирования группировочному признаку.

Контроль безопасности данных и систем подразделяется на контроль достоверности данных, безопасности данных и компьютерных систем. Контроль достоверности данных выполняется программно во время ввода и обработки. Средства безопасности данных и программ защищают их от копирования, искажения, несанкционированного доступа. Средства безопасности компьютерных систем обеспечивают защиту от кражи, вирусов, неправильной работы пользователей, несанкционированного доступа.

Контроль достоверности данных. Основная задача заключается в определении местоположения искаженных данных и их исправлении. Различают три типа контроля достоверности:

- синтаксический;
- семантический;
- прагматический.

Синтаксический контроль обеспечивает проверку типа полей, наличие запрещенных символов, порядка следования реквизитов. Выполняется на основе описания входных документов.

Семантический контроль проверяет логические взаимосвязи значений реквизитов, непротиворечивость данных и их согласованность.

Прагматический контроль проверяет плотность, своевременность, полноту данных, предоставляемых для принятия решений. Выполняется на основе выходных документов.

1.3. Понятия «объективность», «полнота», «достоверность», «адекватность», «доступность» и «актуальность информации». Технология обеспечения безопасности компьютерных систем. Проблемы использования информационных технологий

Объективность и *субъективность* информации. Понятие объективности является относительным, так как методы являются субъективными. Более объективной принято считать ту информацию, в которую методы вносят меньший субъективный элемент.

Полнота информации. Полнота информации во многом характеризует качество информации и определяет достаточность данных для принятия решений или для создания новых данных на основе имеющихся. Чем полнее данные, тем шире диапазон методов, которые можно использовать.

Достоверность информации. Данные возникают в момент регистрации сигналов, но не все сигналы являются «полезными» – всегда присутствует какой-то уровень посторонних сигналов, в результате чего полезные данные сопровождаются определенным уровнем «информационного шума». Если полезный сигнал зарегистрирован более четко, чем посторонние сигналы, достоверность информации может быть более высокой.

Адекватность информации – это степень соответствия реальному объективному состоянию дела. Неадекватная информация может образовываться при создании новой информации на основе неполных или недостоверных данных. Однако и полные, и достоверные данные могут приводить к созданию неадекватной информации в случае применения к ним неадекватных методов.

Доступность информации – мера возможности получить ту или иную информацию. На степень доступности информации влияют одновременно как доступность данных, так и доступность адекватных методов для их интерпретации. Отсутствие доступа к данным или отсутствие адекватных методов обработки данных приводит к одинаковому результату: информация оказывается недоступной.

Актуальность информации – это степень соответствия информации текущему моменту времени. Поскольку информационные процессы растянуты во времени, то достоверная и адекватная, но устаревшая информация может приводить к ошибочным решениям.

Репрезентативность информации связана с правильностью ее отбора и формирования в целях адекватного отражения свойств объекта. Важное значение здесь имеют:

- правильность концепции, на базе которой сформулировано исходное понятие;
- обоснованность отбора существенных признаков и связей отображаемого явления.

Содержательность информации отражает семантическую емкость, равную отношению количества семантической информации в сообщении к объему обрабатываемых данных, то есть

$$C = \frac{I_c}{V_d}, \quad (1.1)$$

где I_c – количество семантической информации;

V_d – объем данных.

Точность информации определяется степенью близости получаемой информации к реальному состоянию объекта, процесса, явления и т. п.

Устойчивость информации отражает ее способность реагировать на изменения исходных данных без нарушения необходимой точности. Устойчивость информации, как и репрезентативность, обусловлена выбранной методикой ее отбора и формирования.

Технология обеспечения безопасности компьютерных систем

Кража программных средств, взлом их защиты, порча программных продуктов появились одновременно с возникновением компьютерных технологий. Но наибольшее внимание к этим вопросам привлек запуск в октябре 1988 г. Робертом Моррисом вируса в компьютерную сеть Arpanet. В результате был полностью или частично заблокирован ряд общенациональных компьютерных сетей, в частности Internet, C&net, Bitnet. Общий ущерб оценивается специалистами в 100 млн долл. Поэтому вопросами безопасности компьютерных систем с момента их возникновения занимались ведомства по охране государственных и военных тайн, а в настоящее время – законодательные службы и институты.

Во время эксплуатации наибольший вред наносят *вирусы*. На больших ЭВМ команды обработки прерываний были привилегированными, то есть недоступными программисту. Поэтому на больших ЭВМ вирусы не приносили существенных убытков. Прерывания на ПЭВМ обрабатываются непривилегированными командами и могут быть доступными как операционной системе, так и прикладным программам. Прикладные программы могут перехватить прерывание, выполнить любые действия и вернуть управление. Так образуются вирусы, которые по среде размножения делятся:

- на файловые, заражающие файлы с расширением;
- boot-вирусы, заражающие сектор начальной загрузки;
- компилируемые, изменяющие текст исходных модулей;
- сетевые, запускающие вирусы по сети и являющиеся самыми

разрушительными.

Вирусы могут заражать программы, диски, операционные системы. Для борьбы с ними разработаны программы, которые делятся на следующие виды.

1. *Детекторы и ревизоры* – проверяют целостность информации. В основном они запоминают длину программы или файла, подсчитывают и проверяют контрольные суммы программ, файлов. Являются наиболее перспективными для выявления новых типов вирусов.
2. *Программы-фаги* – вырезают повторяющийся текст или вирусы.
3. *Программы-фильтры* – следят за несанкционированными действиями по обработке прерываний.

Проблемы использования информационных технологий

1. Информационный поток превосходит ограниченные возможности человека по восприятию и переработке информации.
2. Возникает большое количество избыточной информации (так называемый «информационный шум»), которая затрудняет восприятие полезной для потребителя информации.
3. Возникают экономические, политические и другие барьеры, которые препятствуют распространению информации (например, по причине секретности).

Частичный выход из информационного кризиса видится в применении новых информационных технологий. Внедрение современных средств и методов хранения, обработки и передачи инфор-

мации многократно снижают барьер доступа к ней и скорость поиска. Разумеется, одни лишь технологии не могут решить проблему, имеющую и экономический характер (информация стоит денег), и юридический (информация имеет собственника), и ряд других. Эта проблема комплексная и решается усилиями как каждой страны, так и мирового сообщества в целом.

1.4. Информационная технология поддержки принятия решений. Основные компоненты

Системы поддержки принятия решений и соответствующая им информационная технология появились в конце 70-х–начале 80-х гг. XX в., чему способствовали широкое распространение персональных компьютеров, стандартных пакетов прикладных программ, а также успехи в создании систем искусственного интеллекта.

Главной особенностью информационной технологии поддержки принятия решений является качественно новый метод организации взаимодействия человека и компьютера. Выработка решения, что является основной целью этой технологии, происходит в результате итерационного процесса, в котором участвуют:

- система поддержки принятия решений (СППР) в роли вычислительного звена и объекта управления;
- лицо, принимающее решение, оценивающее полученный результат вычислений на компьютере.

Окончание итерационного процесса происходит по воле человека. В этом случае можно говорить о способности информационной системы совместно с пользователем создавать новую информацию для принятия решений.

Дополнительно к этой особенности информационной технологии поддержки принятия решений можно указать еще ряд ее отличительных характеристик:

- ориентация на решение плохо структурированных (формализованных) задач;
- сочетание традиционных методов доступа и обработки компьютерных данных с возможностями математических моделей и методами решения задач на их основе;
- направленность на непрофессионального пользователя компьютера;

– высокая адаптивность, обеспечивающая возможность приспосабливаться к особенностям имеющегося технического и программного обеспечения, а также требованиям пользователя.

Информационная технология поддержки принятия решений может использоваться на любом уровне управления. Кроме того, решения, принимаемые на различных уровнях управления, часто должны координироваться. Поэтому важной функцией и систем, и технологий является координация лиц, принимающих решения, как на разных уровнях управления, так и на одном уровне.

Основные компоненты. Рассмотрим структуру системы поддержки принятия решений (рис. 1.1), а также функции составляющих ее блоков, которые определяют основные технологические операции.

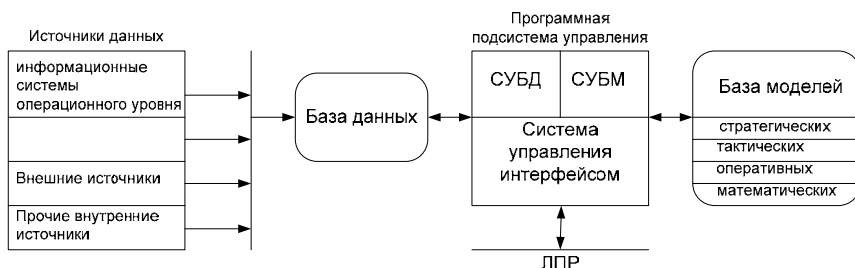


Рис. 1.1. Основные компоненты информационной технологии поддержки принятия решений

В состав системы поддержки принятия решений входят три главных компонента: база данных, база моделей и программная подсистема, которая состоит из системы управления базой данных (СУБД), системы управления базой моделей (СУБМ) и системы управления интерфейсом между пользователем и компьютером.

База данных играет в информационной технологии поддержки принятия решений важную роль. Данные могут использоваться непосредственно пользователем для расчетов при помощи математических моделей. Рассмотрим источники данных и их особенности.

1. Часть данных поступает от информационной системы операционного уровня. Чтобы использовать их эффективно, данные должны быть предварительно обработаны.

2. Помимо данных об операциях предприятия для функционирования системы поддержки принятия решений требуются и другие внутренние данные, например данные о движении персонала, инженерные данные и т. п., которые должны быть своевременно собраны, введены и поддержаны.

3. Важное значение, особенно для поддержки принятия решений на верхних уровнях управления, имеют данные из внешних источников. В отличие от внутренних данных внешние обычно приобретаются у специализирующихся на их сборе организаций.

4. В настоящее время широко исследуется вопрос о включении в базу данных еще одного источника данных – документов, включающих в себя записи, письма, контракты, приказы и т. п. Если содержание этих документов будет записано и затем обработано по некоторым ключевым характеристикам (поставщикам, потребителям, датам, видам услуг и др.), то система получит новый мощный источник информации.

База моделей. Целью создания моделей являются описание и оптимизация некоторого объекта или процесса. Использование моделей обеспечивает проведение анализа в системах поддержки принятия решений. Модели, базируясь на математической интерпретации проблемы, при помощи определенных алгоритмов способствуют нахождению информации, полезной для принятия правильных решений.

Пример. Модель линейного программирования дает возможность определить наиболее выгодную производственную программу выпуска нескольких видов продукции при заданных ограничениях на ресурсы.

1.5. Экспертные системы и их типы. Основные способы формализованного представления знаний в нано- и микросистемной технике. Области применения экспертных систем

Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки экспертных систем (ЭС), основанных на использовании элементов искусственного интеллекта. Экспертные системы дают возможность специалисту получать консультации экспертов по любым проблемам, на основе которых этими системами накоплены знания.

Под *искусственным интеллектом* (ИИ) обычно понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека. Чаще всего здесь имеются в виду способности, связанные с человеческим мышлением. Работы в области искусственного интеллекта не ограничиваются экспертными системами. Они также включают в себя создание роботов, систем, моделирующих нервную систему человека, его слух, зрение, обоняние, способность к обучению.

Решение специальных задач требует специальных знаний. Главная идея использования технологии экспертных систем заключается в том, чтобы получить от эксперта его знания и, загрузив их в память компьютера, использовать всякий раз, когда в этом возникнет необходимость. Являясь одним из основных приложений искусственного интеллекта, экспертные системы представляют собой компьютерные программы, трансформирующие опыт экспертов в какой-либо области знаний в форму эвристических правил. На практике ЭС используются прежде всего как системы-советчики в тех ситуациях, где специалист сомневается в выборе правильного решения. Экспертные знания, хранящиеся в памяти системы, более глубокие и полные, чем соответствующие знания пользователя.

ЭС находят распространение при решении задач с принятием решений в условиях неопределенности (неполноты) для распознавания образов, в прогнозировании, диагностике, планировании, управлении, конструировании и т. д.

Типичная экспертная система состоит из решателя (интерпретатора), БД (базы данных), БЗ (базы знаний), компонентов приобретения знаний, объяснительного и диалогового компонентов.

БД предназначена для хранения исходных и промежуточных данных, используемых для решения задач, фактографических данных.

Решатель, используя исходные данные из БД и БЗ, обеспечивает решение задач для конкретных ситуаций.

Компонент приобретения знаний автоматизирует процесс наполнения БЗ.

Объяснительный компонент объясняет, как система получила решение задачи (или почему не получила) и какие знания она при этом использовала. *Диалоговый компонент* обеспечивает диалог между экспертной системой и пользователем в процессе решения задачи и приобретения знаний.

Экспертные системы создаются для решения разного рода задач профессиональной деятельности человека и в зависимости от этого выполняют разные функции.

Типы экспертных систем.

1. Экспертные системы первого поколения. Предназначены для решения хорошо структурированных задач, требующих небольшого объема эмпирических знаний. Сюда относятся классификационные задачи и задачи выбора из имеющегося набора вариантов.

2. Оболочки ЭС. Имеют механизм ввода-вывода, но БЗ пустая. Требуется настройка на конкретную предметную область. Знания приобретаются в процессе функционирования ЭС, способной к самообучению.

3. Гибридные ЭС. Предназначены для решения различных задач с использованием БЗ. Это задачи с использованием методов системного анализа, исследования операций, математической статистики, обработки информации. Пользователь имеет доступ к объективизированным знаниям, содержащимся в БЗ и пакетах прикладных программ.

4. Сетевые ЭС. Между собой связаны несколько экспертных систем. Результаты решения одной из них являются исходными данными для другой системы. Эффективны при распределенной обработке информации.

При разработке экспертных систем должны участвовать:

– эксперт той предметной области, задачи которой будет решать система;

– инженер по знаниям – специалист по разработкам систем;

– программист – специалист по разработке инструментальных средств.

Эксперт определяет знания, то есть описывает предметную область в виде совокупности данных и правил, обеспечивает полноту и правильность введенных в экспертную систему знаний. Данные определяют объекты, их характеристики и значения. Правила указывают на способы манипулирования данными.

Инженер по знаниям помогает эксперту: выявить и структурировать знания, необходимые для функционирования экспертной системы; осуществить выбор инструментальных средств, которые

наиболее эффективны для решения задач в данной предметной области; указать способы представления знаний.

Программист разрабатывает инструментальную среду, включающую все компоненты экспертной системы, производит ее сопряжение с другими существующими системами.

Основные способы формализованного представления знаний в nano- и микросистемной технике.

1. *Представление знаний продукционными правилами.* Продукционные правила представляют знания в форме ЕСЛИ–ТО. Системы, использующие представления знаний продукционными правилами, называются продукционными. Это самый наглядный и простой способ. В таких системах представления знаний имеются средства, позволяющие использовать в данных и правилах нечеткую информацию с определенной вероятностью, называемой фактором уверенности.

2. *Логика предикатов (раздел математической логики).* Константы и переменные определяют отдельные объекты и обозначаются буквами или набором букв (U, V, W, X, Y). Последовательность из n констант или переменных (n – конечно, $n > 1$) называется функцией. Атомарным предикатом называется последовательность из n сущностей и понятий, описанных константами, переменными или функциями.

Предикат принимает одно из двух значений: истина или ложь. Предикат, в котором все переменные, константы и функции связаны между собой, называется предложением. Предложения используются для представления знаний. Логика предикатов обеспечивает высокий уровень модульности знаний (представляет их как единое целое в определенной предметной области) и позволяет выяснить, имеются или отсутствуют противоречия между новыми и уже существующими знаниями. Но чрезмерный уровень формализации представления знаний, трудность их прочтения снижают эффективность обработки. Кроме этого, в логике предикатов все отношения описываются предикатами, что не позволяет при компьютерной обработке полностью отразить свойства структуры данных.

3. *Семантические сети.* Знания можно рассматривать как отношения между понятиями и сущностями, являющимися конкретными объектами реального мира. Понятия и отношения можно представить в виде семантической сети, состоящей из вершин и дуг. В вершинах располагаются понятия, а направленные связи между

вершинами соответствуют различного рода отношениям между этими понятиями. Семантические сети могут быть выполнены обучаемыми и растущими, что означает возможность автоматического добавления в сеть новых узлов по мере появления в опыте ее использования новых понятий, а также увеличение весовых коэффициентов, соответствующих дугам. В процессе ее обучения между существующими узлами также могут устанавливаться дополнительные связи.

4. *Фреймовые системы.* Фреймы рассматриваются как структура описания отдельной сущности или понятия. Они могут быть в виде их совокупностей, представляемых как отдельное множество знаний, относящихся к одному объекту. Каждый фрейм состоит из множества элементов, называемых слотами, которые в свою очередь представляются определенной структурой данных. Каждый фрейм и слот имеют имя, единственное во всей фреймовой системе. В значении слот содержит конкретную информацию.

Области применения ЭС. ЭС в задачах интерпретации, как правило, используют информацию от датчиков для описания ситуации. В качестве примера приведем интерпретацию показаний измерительных приборов на предприятии для определения состояния технологического процесса. Интерпретирующие системы имеют дело не с четкими символьными представлениями проблемной ситуации, а непосредственно с реальными данными. Они сталкиваются с затруднениями, которых нет у систем других типов, потому что им приходится обрабатывать информацию зашумленную, недостаточную, неполную, ненадежную или ошибочную. Им необходимы специальные методы регистрации характеристик непрерывных потоков данных, сигналов или изображений и методы их символьного представления.

ЭС в задачах прогнозирования определяют вероятные последствия заданных ситуаций. Системы прогнозирования иногда используют имитационное моделирование, то есть программы, которые отражают причинно-следственные взаимосвязи в реальном мире, чтобы сгенерировать ситуации или сценарии, которые могут возникнуть при тех или иных входных данных. Эти возможные ситуации вместе со знаниями о процессах, порождающих эти ситуации, образуют предпосылки для прогноза.

ЭС в задачах диагностики используют описания ситуаций, характеристики поведения или знания о конструкции компонента,

чтобы установить вероятные причины неправильного функционирования диагностируемой системы. Пример: локализация неисправностей в электронных приборах.

ЭС, применяемые в области проектирования, разрабатывают конфигурации объектов с учетом набора ограничений, присущих проблеме. Учитывая то, что проектирование столь тесно связано с планированием, многие проектирующие системы содержат механизмы разработки и уточнения планов для достижения желаемого проекта. Наиболее часто встречающиеся области применения планирующих ЭС: химия, электроника и военное дело.

ЭС, которые используются для решения задач наблюдения, сравнивают действительное поведение с ожидаемым поведением системы. Примером может служить слежение за показаниями измерительных приборов в ядерных реакторах с целью обнаружения аварийных ситуаций. Наблюдающие ЭС по самой своей природе должны работать в режиме реального времени.

ЭС в задачах ремонта аппаратуры следуют плану, который предписывает некоторые рецепты восстановления. Примером является настройка масс-спектрометра, то есть установка ручек регулировки прибора в положение, обеспечивающее достижение оптимальной чувствительности, совместимой с правильным отношением величин пиков и их формы.

ЭС в области обучения подвергают диагностике, «отладке» и исправлению («ремонту») поведение обучаемого. В качестве примера приведем обучение студентов поиску неисправностей в электрических цепях.

1.6. Автоматизированные системы научных исследований. Типовая структура

Автоматизированные системы научных исследований (АСНИ) представляют собой программно-аппаратные комплексы, обрабатывающие данные, поступающие от различного рода экспериментальных установок и измерительных приборов, и на основе их анализа облегчающие обнаружение новых эффектов и закономерностей (рис. 1.2).

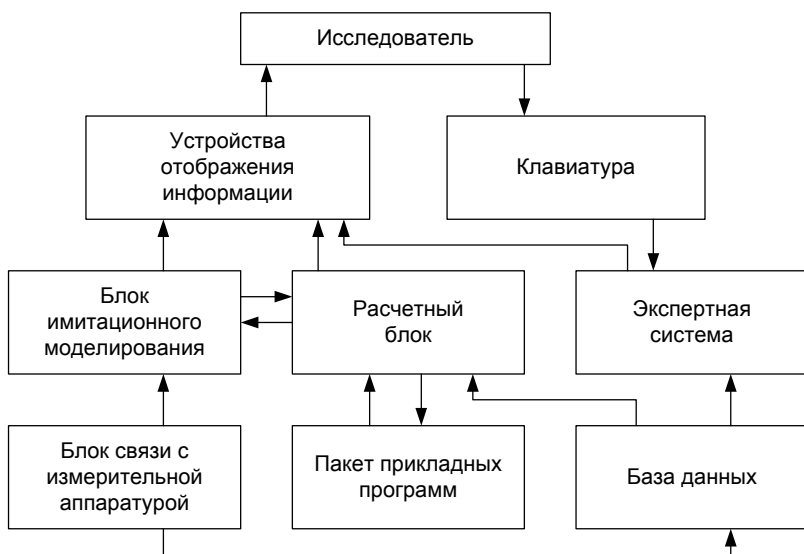


Рис. 1.2. Типовая структура АСНИ

Блок связи с измерительной аппаратурой преобразует к нужному виду информацию, поступающую от измерительной аппаратуры.

В базе данных хранится информация, поступившая из блока связи с измерительной аппаратурой, а также заранее введенная с целью обеспечения работоспособности системы.

Расчетный блок, выполняя программы из пакета прикладных программ, производит все математические расчеты, в которых может возникнуть потребность в ходе научных исследований.

Расчеты могут выполняться как по требованию исследователя, так и блока имитационного моделирования. При этом на основе математических моделей воспроизводится процесс, происходящий во внешней среде.

Экспертная система моделирует рассуждения специалистов данной предметной области. С ее помощью исследователь может классифицировать наблюдаемые явления, диагностировать течение исследуемых процессов.

АСНИ получили широкое распространение в молекулярной химии, минералогии, биохимии, физике элементарных частиц и других науках.

1.7. Общие сведения о системах автоматизированного проектирования. Типовая структура. Определение и основное назначение CAD- / CAE- / CAM-систем

Близкими по своей структуре и функциям к системам автоматизации научных исследований оказываются системы автоматизированного проектирования (САПР).

САПР – комплекс программных и аппаратных средств, предназначенных для автоматизации процесса проектирования человеком технических изделий или продуктов интеллектуальной деятельности.

Проектирование новых изделий – основная задача изобретателей-конструкторов – протекает в несколько этапов, таких как нормирование замысла, поиск физических принципов, обеспечивающих реализацию замыслов и требуемые значения конструкции, поиск конструктивных решений, их расчет и обоснование, создание опытного образца, разработка технологий промышленного изготовления. Если формирование замысла и поиск физических принципов пока остаются чисто творческими, не поддающимися автоматизации этапами, то при конструировании и расчетах с успехом могут быть применены САПР (рис. 1.3).

База данных, блок имитационного моделирования, расчетный блок и экспертная система выполняют функции, аналогичные функциям соответствующих блоков АСНИ. Вместо блока связи с измерительной аппаратурой в САПР имеется блок формирования заданий. Проектировщик вводит в блок техническое задание на проектирование с указанием целей, которые необходимо достичь при проектировании, и все ограничения, которые нельзя нарушить. Блок подготовки технической документации облегчает ее создание для последующего изготовления изделия.

Аппаратное обеспечение САПР составляет ЭВМ с набором устройств, необходимых для ввода и вывода графической информации (графопостроитель, световое перо, графический планшет и др.).

Системы автоматизации проектирования включают в себя *системы инженерной графики (CAD)*, *системы инженерных расчетов (CAE)*, *системы автоматизации подготовки и управления производства (CAM)*. CAD-системы предназначены для решения конструкторских задач и оформления конструкторской документации. В современные CAD-системы входят модули моделирования трехмерной

объемной конструкции и оформления чертежей и текстовой конструкторской документации (спецификаций, ведомостей и т. д.).

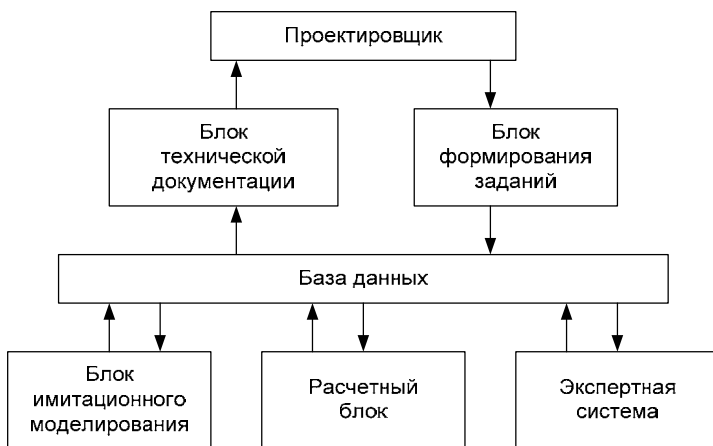


Рис 1.3. Типовая схема САПР

САЕ-системы – это класс систем, каждая из которых позволяет решать определенную расчетную задачу, начиная от расчетов на прочность, анализа и моделирования тепловых процессов до расчетов гидравлических систем и машин, расчетов процессов литья. В САЕ-системах также используется трехмерная модель изделия, созданная в системе CAD.

САМ-системы предназначены для проектирования обработки изделий на станках с числовым программным управлением (ЧПУ) и выдачи программ для этих станков. В настоящее время САМ-системы являются одним из основных способов изготовления сложнопрофильных деталей и сокращения цикла их производства. В них используется трехмерная модель детали, созданная в САЕ-системе.

Традиционно производится условное деление САЕ- / САМ- / САЕ-систем на системы верхнего (например, Unigraphics), среднего (например, Solid Edge) и нижнего уровня (например, AutoCAD).

Основные критерии выбора систем:

- функциональные возможности;
- наличие уникальных функций;
- стоимость;
- простота интерфейса и легкость обучения.

1.8. Основы автоматизации проектирования микроэлектромеханических и наноэлектромеханических систем. Методология их проектирования

Микроэлектромеханические системы (МЭМС) – это интегрированные системы, комбинирующие электрические и механические компоненты, изготовленные по технологиям, совместимым с технологией ИС, и имеющие размеры от 0,1 до сантиметров.

Наноэлектромеханические системы (НЭМС) – это интегрированные системы, комбинирующие электрические и механические компоненты, изготовленные по технологиям, совместимым с технологией ИС, и имеющие размеры от 0,1 до 100 нанометров.

В общем случае они обеспечивают конвертирование величин различной физической природы (электрической, механической и т. д.). В связи с этим обобщенная функциональная блок-схема МЭМС включает в себя следующие компоненты:

- актюаторы и / или сенсоры;
- системы излучения (антенны);
- цепи, используемые для обработки информации от сенсоров, актюаторов;
- цепи управления.

Актюатор (составная часть МЭМС) – это устройство, которое преобразовывает энергию в управляемое движение. Благодаря своей простоте наибольшее распространение в МЭМС получили электростатические преобразователи. Электростатическая активация применяется примерно в одной трети актюаторов.

Сенсор – это устройство, преобразующее входное воздействие различной физической природы в электрический сигнал.

Методы проектирования систем, базирующихся на использовании электрических и механических компонентов, изготовленных по технологиям, совместимым с технологией производства интегральных схем, существенно отличаются от ранее применявшихся методов проектирования электромеханических систем. В основе маршрутов проектирования МЭМС и НЭМС лежит *междисциплинарный подход*, позволяющий учесть разнообразные физические взаимодействия между разными областями системы: электрической, механической, тепловой и др. Важной проблемой в наноинженерии

является выбор моделей, адекватно отражающих поведение микро- и наноразмерных элементов на различных уровнях проектирования наносистем. При выборе проектного решения необходимо также учитывать влияние технологического процесса и различных дестабилизирующих факторов. В связи с этим для реализации сложных маршрутов проектирования МЭМС и НЭМС применяются интегрированные САПР, позволяющие выполнить моделирование и оценку различных проектных решений на различных уровнях проектирования.

Методология проектирования МЭМС базируется:

- на классической механике, основанной на трех законах Ньютона и принципе относительности Галилея;
- теоретической (аналитической) механике (механика Лагранжа);
- классической электродинамики (уравнения Максвелла).

Методология проектирования НЭМС базируется:

- на квантовой механике;
- нанoeлектромеханических концепциях.

Проектирование МЭМС и НЭМС почти на всех своих фазах автоматизировано. В связи с этим рассматриваются методологии, алгоритмы, методы описания и моделирования, используемые при автоматизированном проектировании. Все перечисленное выше объединяется в понятие «CAE – computer aided engineering». Специфические особенности маршрутов проектирования микро- и наносистем по сравнению с традиционными макросистемами вытекают из их размеров.

1.9. Интегрированные пакеты автоматизированного проектирования МЭМС и НЭМС

Для автоматизированного проектирования МЭМС и НЭМС необходимы интегрированные системы, включающие в себя подсистемы:

- проектирование на компонентном уровне;
- проектирование на междисциплинарном уровне;
- проектирование электронных компонентов;
- проектирование с учетом обеспечения надежного функционирования МЭМС и НЭМС.

В настоящее время разработаны различные пакеты автоматизированного проектирования МЭМС и НЭМС, позволяющие выполнить проектирование на всех основных этапах в интегрированном маршруте. Одним из таких пакетов является интегрированный пакет

программ CoventorWare (<http://www.euointech.ru/coventor>). В настоящее время он включает в себя следующие основные модули (рис. 1.4):

- модуль ARCHITECT;
- модуль DESIGNER;
- модуль ANALYZER;
- модуль INTEGRATOR.

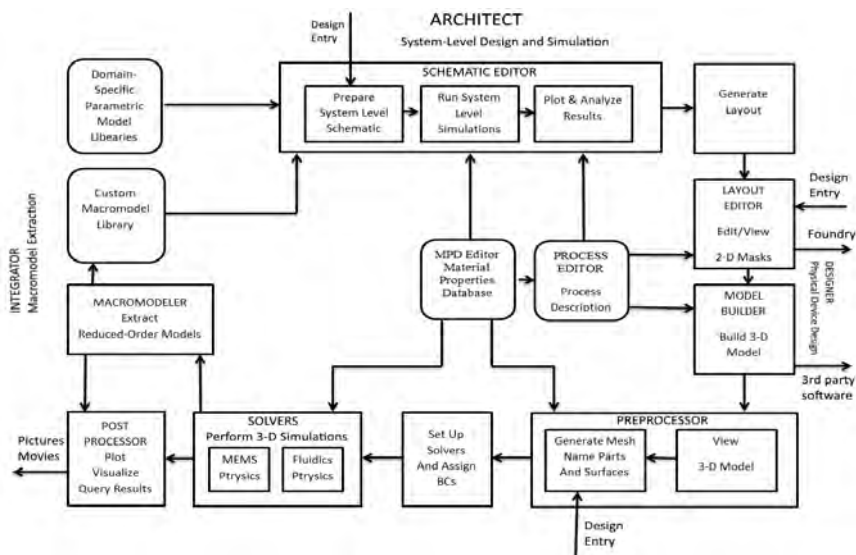


Рис. 1.4. Архитектура интегрированного пакета автоматизированного проектирования CoventorWare

Имеются также дополнительные модули.

- модуль Etch-3D;
- модуль MEMULATOR;

Программа ARCHITECT содержит библиотеки поведенческих моделей электромеханических, оптических, жидкостных и т. д. микроустройств и библиотеки базовых элементов (пружины, массы, демпферы, конденсаторы, индуктивности, операционные усилители, транзисторы и т. д.).

Модуль моделирования позволяет выполнить различные виды анализа:

- расчет рабочих точек по постоянному току;
- анализ переходных процессов;

- частотный анализ в режиме малого сигнала;
- анализ передаточных функций по постоянному току;
- анализ чувствительности.

Программа Architect позволяет значительно облегчить труд проектировщика, предоставляя различные средства визуализации МЭМС и НЭМС. При выборе трехмерного моделирования открывается дополнительное окно CoventorWare – Architect 3D-Visualizer. Окно состоит из трех панелей. На левой панели отражается дерево анализа модели. Из верхней панели конструктор выбирает необходимые инструменты для выбора проекции проектируемого устройства. В центральной панели отражаются результаты работы. Например, при моделировании двойного гироскопа можно:

- увеличить изображение с помощью лупы;
- изучить поведение проектируемого устройства в требуемый промежуток времени;
- получить изображение проектируемого устройства в различных проекциях.

При этом изменение цвета от синего к красному соответствует изменению амплитуды смещения с нулевого до максимального значения. В программе Architect доступна анимация для следующих видов анализа: расчет установившихся и переходных режимов, а также модальный анализ.

Возможности анимации в программе Architect значительно усилились после появления модуля Scene-3D. Продукт Scene-3D является вспомогательной программой для модуля Architect и позволяет отображать схему МЭМС устройства в виде трехмерного прототипа. С введением в состав пакета CoventorWare модуля Scene-3D стало возможным формирование трехмерного вида разрабатываемого устройства, а также визуализация его функционирования непосредственно из модуля Architect, минуя этап формирования послойных масок. Подобная функциональность стала возможной благодаря введению специальной библиотеки трехмерных элементов, связанной с библиотеками поведенческих моделей. Кроме того, помимо МЭМС устройств, 3D-библиотеки могут содержать модели любых других элементов, например транзисторов.

Интегрированный пакет автоматизированного проектирования CoventorWare поддерживает формат gdsii, dxf. Возможен экспорт модели в программный комплекс Ansys.

Другим примером интегрированного пакета автоматизированного проектирования МЭМС и НЭМС является САПР MEMS Pro. Он предназначен для проектирования следующих МЭМС:

- механизмов (радиальные и линейные гребни, зубчатые передачи, мембраны, консоли, микроузлы);
- инерционных блоков (якоря, балки, пластины);
- электромеханических структур (гребневые микродвигатели);
- пьезорезисторов и пьезоэлектрических элементов и т. д.

В настоящее время интегрированный пакет автоматизированного проектирования EMS Pro включает в себя следующие основные модули:

- MEMS Modeler;
- L-Edit;
- Design Rule Checker.

Программа MEMS Modeler позволяет получить поведенческие модели.

Интегрированная среда позволяет выполнить проектирование на всех основных этапах автоматизированного проектирования МЭМС и НЭМС. Интегрированный пакет автоматизированного проектирования MEMS Pro поддерживает формат gdsii, cif. Возможен обмен моделями с программным комплексом Ansys.

Еще одним примером интегрированного пакета автоматизированного проектирования МЭМС и НЭМС является САПР IntelliSuite фирмы IntelliSense Software, США. Поддерживает маршрут проектирования для следующих МЭМС:

- микроустройства (акселерометры, гироскопы, микрофоны, сверхоперативные запоминающие устройства и т. д.);
- микродвигатели (радиальные и линейные гребневые микродвигатели, зубчатые передачи, моторы и т. д.);
- электромагнитные МЭМС (переключатели, перестраиваемые конденсаторы и т. д.);
- СВЧ МЭМС (копланарные волноводы, электрооптические модуляторы, микрополосковые катушки и т. д.);
- анализаторы (тепловые, механические, электростатические, пьезоэлектрические, биоМЭМС и т. д.).

В настоящее время интегрированный пакет автоматизированного проектирования IntelliSuite включает в себя следующие основные модули:

- MEMaterial;
- IntelliFAB;
- AnisE;
- RECIPE;
- IntelliMask.

Интегрированный пакет автоматизированного проектирования IntelliSuite поддерживает формат iges, gdsii, dxf. Возможен обмен конечно-элементными моделями с программным комплексом Ansys.

1.10. Общие сведения о компьютерном математическом моделировании. Классификация математических моделей. Имитационное моделирование

Основой автоматизированного проектирования МЭМС и НЭМС является математическая модель. Она представляет собой множество математических объектов и отношений между ними, адекватно отражающее конкретные свойства МЭМС и НЭМС.

Модель – материальный объект, система математических зависимостей или программа, имитирующая структуру или функционирование исследуемого объекта.

Моделирование – представление различных характеристик поведения физической или абстрактной системы с помощью другой системы.

Математическое моделирование – метод исследования процессов и явлений на их математических моделях.

Изучение компьютерного математического моделирования открывает широкие возможности для осознания связи информатики с математикой и другими науками – естественными и социальными. Компьютерное математическое моделирование в разных своих проявлениях использует практически весь аппарат современной математики.

К *классификации* математических моделей можно подходить по-разному, положив в основу различные принципы:

а) классификация моделей по отраслям наук (математические модели в физике, биологии, социологии и т. д.);

б) классификация моделей по применяемому математическому аппарату (модели, основанные на применении обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, стохастических методов, дискретных алгебраических преобразований и т. д.);

- в) классификация моделей с точки зрения целей моделирования:
- дескриптивные (описательные);
 - оптимизационные;
 - многокритериальные;
 - имитационные.

Пример. Меняя тепловой режим отжига, можем стремиться подобрать такой, чтобы достичь максимальной проводимости пленки, то есть оптимизируем процесс.

Имитационная модель – описание системы и ее поведения, которое может быть реализовано и исследовано в ходе операций на компьютере.

Имитационное моделирование – исследование поведения сложной системы на ее модели.

Можно сказать, что чаще всего имитационное моделирование применяется для того, чтобы описать свойства большой системы при условии, что поведение составляющих ее объектов очень просто и четко сформулировано. Математическое описание тогда сводится к уровню статистической обработки результатов моделирования при нахождении макроскопических характеристик системы. Такой компьютерный эксперимент фактически претендует на воспроизведение натурального.

Имитационное моделирование позволяет осуществить проверку гипотез, исследовать влияние различных факторов и параметров.

1.11. Этапы, цели и средства компьютерного математического моделирования

Рассмотрим процесс компьютерного математического моделирования, включающий численный эксперимент с моделью (рис. 1.5).

Первый этап – определение целей моделирования.

Основные цели:

- 1) модель нужна для того, чтобы понять, как устроен конкретный объект, какова его структура, основные свойства, законы развития и взаимодействия с окружающим миром (понимание);
- 2) модель нужна для того, чтобы научиться управлять объектом (или процессом) и определить наилучшие способы управления при заданных целях и критериях (управление);

3) модель нужна для того, чтобы прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействия на объект (прогнозирование).

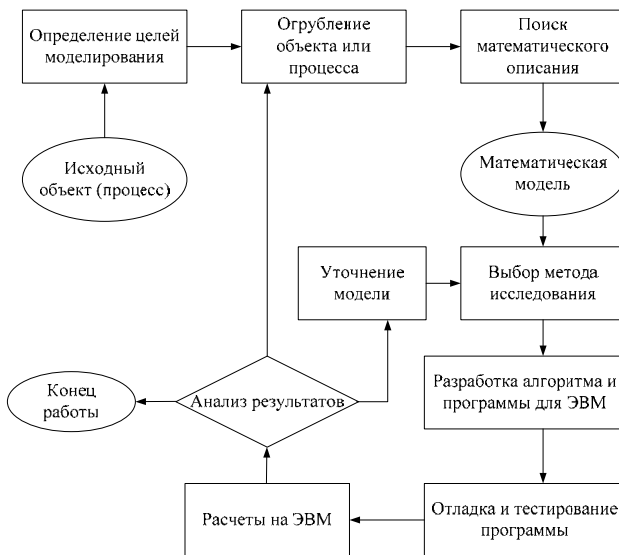


Рис. 1.5. Общая схема процесса компьютерного математического моделирования

Важнейшим этапом моделирования является разделение входных параметров по степени важности влияния их изменений на выходные. Такой процесс называется *ранжированием* (разделением по рангам). От того, насколько умело выделены важнейшие факторы, зависит успех моделирования, быстрота и эффективность достижения цели. Выделить более важные (или, как говорят, значимые) факторы и отсеять менее важные может лишь специалист в той предметной области, к которой относится модель.

Отбрасывание (по крайней мере при первом подходе) менее значимых факторов огрубляет объект моделирования и способствует пониманию его главных свойств и закономерностей. Умело ранжированная модель должна быть адекватна исходному объекту или процессу в отношении целей моделирования. Обычно определить, адекватна ли модель, можно только в процессе экспериментов с ней, анализа результатов.

Следующий этап – поиск математического описания. На этом этапе необходимо перейти от абстрактной формулировки модели к формулировке, имеющей конкретное математическое наполнение. В этот момент модель предстает перед нами в виде уравнения, системы уравнений, системы неравенств, дифференциального уравнения или системы таких уравнений и т. д.

Когда математическая модель сформулирована, выбирается метод ее исследования. Как правило, для решения одной и той же задачи есть несколько конкретных методов, различающихся эффективностью, устойчивостью и т. д. От верного выбора метода часто зависит успех всего процесса.

Разработка алгоритма и составление программы для ЭВМ – это творческий и трудноформализуемый процесс. В настоящее время при компьютерном математическом моделировании часто используются приемы процедурно-ориентированного (структурного) программирования.

При создании имитационной модели можно также воспользоваться возможностями одного из пакетов математической поддержки (Mathematica, MathCad, MathLab и др).

В настоящее время существуют проблемно-ориентированные имитационные языки, в которых объединяются различные альтернативные подходы и которые самой своей структурой определяют возможную схему действий разработчика модели. Характерным примером такого рода является SLAM – *simulating language for alternative modeling* – имитационный язык для альтернативного моделирования.

После составления программы решаем с ее помощью простейшую тестовую задачу (желательно, с заранее известным ответом) с целью устранения грубых ошибок. Это только начало процедуры тестирования, которую трудно описать формально исчерпывающим образом. По существу, тестирование может продолжаться долго и закончиться тогда, когда пользователь по своим профессиональным признакам сочтет программу верной.

Затем следует собственно численный эксперимент, и выясняется, соответствует ли модель реальному объекту (процессу). Модель адекватна реальному процессу, если некоторые характеристики процесса, полученные на ЭВМ, совпадают с экспериментальными с заданной степенью точности. В случае несоответствия модели реальному процессу возвращаемся к одному из предыдущих этапов.

1.12. Особенности имитационного моделирования производственных систем

Для анализа производственных систем, которые очень сложны, разноплановы, не имеют исчерпывающего математического описания, адекватные математические модели, будь то логические или числовые, построить не представляется возможным. Естественным здесь является использование методов имитационного моделирования.

Система может быть однозначно описана набором значений производственных параметров, характерных для каждого конкретного ее состояния. Если их внести в компьютер, то изменения в ходе вычислительного процесса можно интерпретировать как имитацию перехода системы из одного состояния в другое. При таких предположениях имитационное моделирование можно рассматривать как динамическое представление системы путем продвижения ее одного состояния к другому по характерным операционным правилам.

При имитационном моделировании производственных систем изменения их состояния происходят в дискретные моменты времени. Основная концепция имитационного моделирования системы и в этом случае состоит в отображении изменений ее состояния с течением времени. Таким образом, здесь определяющим является выделение и однозначное описание состояний моделируемой системы.

Имитационные модели позволяют без использования каких-либо аналитических или других функциональных зависимостей отображать сложные объекты, состоящие из разнородных элементов, между которыми существуют разнообразные связи. В эти модели также может быть включен и человек.

Без принципиальных усложнений в такие модели могут быть включены как детерминированные, так и стохастические потоки (материальные и информационные). С помощью имитационного моделирования можно отображать взаимосвязи между рабочими местами, потоками материалов и изделий, транспортными средствами и персоналом.

Несмотря на такие очевидные преимущества, прежде всего заключающиеся в широте и универсальности применения, при этом методе из вида упускается существование логических связей, что исключает возможность полной оптимизации получаемых на этой

модели решений. Гарантируется лишь возможность отбора лучшего из рассмотренных вариантов.

Практически же имитационное моделирование во многих реальных случаях – единственно возможный способ исследования. После разработки имитационной модели с ней проводятся компьютерные эксперименты, которые позволяют сделать выводы о поведении производственной системы.

Появление и развитие методов компьютерного имитационного моделирования стало возможным также и в результате развития метода статистических испытаний, позволившего моделировать случайные события и процессы, занимающие большое место в реальных производствах.

При составлении имитационной модели и проведении с ее помощью моделирования исследуемого объекта необходимо решение нескольких связанных между собой задач. К ним относятся:

- анализ моделируемой системы и составление ее формализованного описания, включая выявление информационно-логической структуры системы, идентификацию ее компонентов, выбор параметров, характеризующих состояние этих компонентов, разработку компьютерной модели системы, способной воспроизвести ее поведение, планирование эксперимента по развертыванию событий в компьютерной модели, отображающих события в моделируемой системе;

- разработка методологии компьютерного статистического эксперимента, включая генерацию случайных или псевдослучайных чисел, имитацию различных случайных событий, статистическую обработку данных;

- проведение собственно компьютерного эксперимента на имитационной модели, включая управление параметрами и переменными модели в ходе ее исследования на компьютере.

1.13. Основные параметры, определяющие течение и состояние технологического процесса производства компонентов nano- и микросистемной техники

Современный анализ всего технологического процесса производства материалов электронной техники осуществляют на основе системного подхода, где основным является понятие «большая система» или «большая технологическая система», включающее

совокупность происходящих физико-химических и механических процессов, объектов обработки и средств для их реализации.

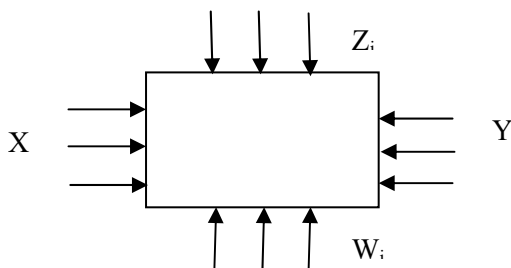


Рис. 1.6. Схема технологического процесса как большой системы

Технологический процесс, как большая система, схематически представлен на рис. 1.6. Здесь:

X_1, X_2, \dots, X_m – входы системы или параметры определения исходных продуктов;

Y_1, Y_2, \dots, Y_n – выходы системы или выходные параметры конечного продукта;

Z_1, Z_2, \dots, Z_i – контролируемые и управляющие факторы, необходимые для поддержания определенного технологического процесса;

W_1, W_2, \dots, W_k – неконтролируемые (возмущающие) факторы, оказывающие случайное возмущающее воздействие на процесс.

Охарактеризуем основную группу параметров, определяющих течение процесса и его состояние в любой момент времени.

1. Входные параметры. К ним относятся величины, которые могут быть определены экспериментально, но возможность воздействия на них в течение процесса отсутствует. Значения этих параметров не зависят от режима процесса. Например, в процессах получения монокристаллов входным параметром может быть состав исходного сырья, загружаемый в печь.

2. Выходные параметры конечного продукта. К ним относят те характеристики системы, значения которых определяются режимом изучаемого процесса в системе, подвергающейся воздействию входных, управляющих и неконтролируемых факторов.

3. Управляющие параметры. Это те параметры процесса, на которые можно оказывать влияние в соответствии с теми или иными требованиями, что позволяет управлять процессом.

4. Неконтролируемые (возмущающие) параметры. Значения их изменяются случайно с течением времени и обычно не могут быть измерены на протяжении процесса.

Формализация системы осуществляется с помощью математической модели, выражающей связь между выходными параметрами системы, параметрами состояния и входными управляющими и возмущающими переменными. Выбор совокупности характеристик процесса определяется главным образом теми целями, для достижения которых и строится математическая модель процесса.

1.14. Система моделей в технологии нано- и микросистемной техники

В зависимости от назначения выделяют нижеприведенную систему моделей.

1. Модели для исследования процессов. Главной целью таких моделей является получение полных знаний об объекте в целом или отдельных его сторонах. Их основу составляет сжатое, наглядное и взаимосвязанное отражение накопленных представлений о физических, физико-химических и других закономерностях протекания процессов. Особенностью моделей этого уровня является достаточно глубокое отражение отдельных сторон и явлений (например, анализ гидродинамики расплава при выращивании кристаллов). Основным назначением таких моделей является получение новых знаний, проникновение во внутренний механизм явлений и процессов. Второе важное направление связано с возможностью использования этих моделей для синтеза алгоритмов управления и выбора эффективных управляющих воздействий.

2. Модели для расчета и оптимизации технологии. Такие модели применяются для расчетов технологических процессов и используются в виде инструкций по ведению процесса в проектируемых и действующих агрегатах или в виде задающих установок автоматическим регуляторам. Для этой цели могут использоваться либо определенным образом модифицированные модели, описанные выше, либо создаваться специальные модели в ориентации на достижение целевых технологических критериев.

3. Модели для прогнозирования оптимальных траекторий процесса во времени. В основу таких моделей могут быть положены

представления, полученные с помощью моделей двух рассмотренных выше уровней. Однако наиболее существенным здесь является отражение особенностей протекания, например, отдельных процессов варки стекла, в зависимости от конкретных начальных условий и состояния с учетом взаимосвязей управляемых и внутренних параметров, что может оказывать значительное влияние на прогнозируемые программные режимы. Эти модели целесообразно использовать для «проигрывания» процессов в ускоренном масштабе времени, например перед предстоящей варкой стекла, когда становятся известными начальные условия (состав стекла, температура и т. д.), для выбора оптимального распределения управляющих воздействий во времени (по ходу варки).

4. Модели для стабилизирующего или следящего регулирования. Модели этого уровня должны отличаться наибольшей простотой и оперативностью отражения динамики процесса. При построении таких моделей широко используется функциональный подход. Они чаще всего описываются зависимостями линейного вида и справедливы для управления по отдельным каналам в относительно узком диапазоне изменения переменных.

5. Модели для автоматизированных систем обучения и повышения квалификации на основе тренажерных комплексов. Тренажеры как средство обучения имеют следующие основные достоинства:

– замена реального объекта учебной информационной моделью, позволяющей выделить наиболее существенные для обучения стороны и элементы, в том числе редко встречающиеся и аварийные ситуации, вести подготовку пооперационно и поэтапно с постепенным переходом от простого к сложному;

– возможность изменения параметров моделируемого объекта и масштаба времени, позволяющая повысить интенсивность обучения, например, за счет многократного повторения необходимых ситуаций;

– резкое сокращение материальных и энергетических затрат на обучение по сравнению с обучением на реальном объекте и др.

К моделям, предназначенным для тренажерных комплексов, по сравнению с моделями для управления предъявляются менее жесткие требования по количественной адекватности, но в них необходима более высокая степень качественной аналогии.

1.15. Понятие о вычислительном технологическом эксперименте

Под *вычислительным технологическим экспериментом* (ВТЭ) понимается такая организация исследований, при которой на основе математической модели проводится изучение технологических устройств и процессов с помощью ЭВМ, моделируется их поведение в различных условиях, находятся оптимальные технологические параметры и режимы действующих или проектируемых конструкций. Идеологической базой ВТЭ является математическое моделирование, методологической – теория вычислительных алгоритмов, технической – современные электронные вычислительные машины.

Необходимость использования ВТЭ вызвана тем, что решение современных научно-технологических задач, отличающихся чрезвычайно сложным математическим описанием, традиционными методами становится затруднительным, а в некоторых случаях вообще невозможным. В настоящее время в программу проведения крупных технологических экспериментов в обязательном порядке включается этап ВТЭ, позволяющий выполнить предварительное «прорасчетывание» для уточнения и коррекции плана проведения эксперимента и прогнозирования ожидаемых результатов. Причем ВТЭ позволяет провести исследования в достаточно широком диапазоне значений параметров процесса без модификации существующих установок или разработки новых. Кроме того, при комплексном взаимодействии многих физико-химических явлений и параметров ВТЭ позволяет рассматривать влияние каждого явления или параметра на технологический процесс в отдельности.

Рассмотрим основные задачи ВТЭ, решаемые, например, при проектировании технологических процессов. Их условно делят на следующие четыре группы:

- анализ технологических процессов;
- синтез технологических процессов;
- диагностика;
- прогноз.

При анализе технологического процесса ВТЭ используется для решения нижеприведенных основных задач.

1. Определение «диапазона устойчивого функционирования» технологического процесса, то есть такой области значений управ-

ляющих параметров, любые изменения которых в этой области не приводили бы к нарушению ограничений, наложенных на выходные параметры процесса.

2. Определение критических условий ведения технологического процесса.

3. Установление степени влияния технологических параметров на режимы ведения процесса и его выходные параметры.

4. Исследование влияния изменений технологических режимов процесса и конструкции технологической установки на показатели качества процесса.

5. Выявление важнейших точек технологической установки, в которых необходимо расположение датчиков для контроля основных параметров процесса и управления.

6. Исследование характера поверхности отклика оптимизируемых при проектировании функций для обоснования выбора метода поиска экстремума.

К группе задач синтеза технологических процессов можно отнести:

1) оптимизацию технологических режимов процесса для действующей технологической установки;

2) оптимизацию конструкционных параметров действующей технологической установки;

3) оптимизацию действующего технологического процесса путем одновременной оптимизации технологических режимов и конструкционных параметров действующей технологической установки;

4) проектирование оптимального технологического процесса.

Из приведенных задач особую важность представляет последняя, так как в этом случае применение ВТЭ позволяет решать задачу значительно более общую, чем задача оптимизации действующего технологического процесса, идущего на уже существующей установке. При проектировании нового процесса или совершенствовании имеющегося без ориентации на существующую базовую технологическую установку ВТЭ позволяет проектировать оптимальный технологический процесс путем одновременного решения следующих задач:

– проектирование оптимальных технологических режимов процесса;

– проектирование оптимальных конструкционных и временных параметров технологической установки;

– проектирование оптимальной системы управления процессом, включая систему датчиков.

К задачам диагностики относятся:

- 1) определение характеристик надежности технологической установки и технологического процесса в целом;
- 2) диагностика правильности функционирования технологического процесса;
- 3) исследование устойчивости функционирования технологического процесса при наличии различного рода случайных воздействий на процесс;
- 4) автоматизированное построение диагностической таблицы.

В группу задач прогноза входят:

- 1) прогнозирование показателей качества технологического процесса;
- 2) оценка влияния предполагаемых технических достижений на показатели качества технологического процесса;
- 3) проектирование и исследование новых схем проведения процесса и новых технологических режимов.

Приведенный перечень задач, естественно, является обобщенным. При проектировании конкретных технологических процессов часть из перечисленных задач может быть исключена или могут быть введены новые задачи.

Для проектирования технологических процессов исключительно важное значение имеют и сопутствующие ВТЭ: по обработке данных натурного эксперимента и по изучению свойств технологической среды. Первый из этих ВТЭ реализуется в ходе проверки адекватности используемых в основном ВТЭ моделей с результатами натуральных экспериментов. Второй сопутствующий ВТЭ направлен на получение базы данных для основного ВТЭ, так как наличие информации о параметрах технологической среды позволяет производить поиск в достаточно широком интервале значений параметров процесса и оценивать влияние изменений, вносимых в технологический процесс, на показатели качества процесса.

1.16. Общая характеристика технологии создания программного обеспечения.

Этапы разработки программного обеспечения

К программно-инструментальным средствам в первую очередь относятся алгоритмические языки и соответствующие им трансляторы, затем системы управления базами данных (СУБД) с языковыми средствами программирования в их среде, электронные таблицы с соответствующими средствами их настройки и т. п.

Рассмотрим этапы разработки программ (рис. 1.7).

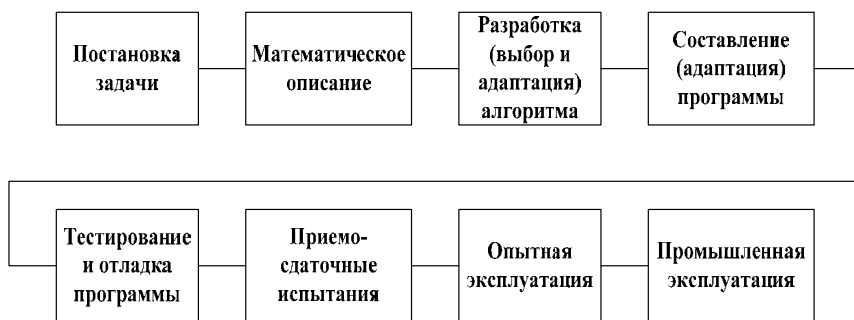


Рис. 1.7. Этапы разработки программного обеспечения

Первый этап представляет собой постановку задачи. На этом этапе раскрывается сущность задачи, то есть формулируется цель ее решения; определяется взаимосвязь с другими задачами; указывается периодичность решения; устанавливаются состав и формы представления входной, промежуточной и результатной информации.

Особое внимание в процессе постановки задачи уделяется детальному описанию входной, выходной (результатной) и промежуточной информации.

Особенность реализации этого этапа технологического процесса заключается в том, что конечный пользователь разрабатываемой программы, хорошо знающий ее проблемную сторону, обычно хуже представляет специфику и возможности использования ЭВМ для решения задачи. В свою очередь, предметная область пользователя (особенно ее отдельные нюансы, способные оказать влияние на решение задачи) зачастую незнакома разработчику программы, хотя

он знает возможности и ограничения на применение ЭВМ. Именно эти противоречия являются основной причиной возникновения ошибок при реализации данного этапа технологического процесса разработки программ, которые затем неизбежно отражаются и на последующих этапах. Отсюда вся важность и ответственность этого этапа, требующего осуществления корректной и полной постановки задачи, а также необходимости однозначного ее понимания как разработчиком программы, так и ее пользователем.

Второй этап в технологии разработки программ – математическое описание задачи и выбор метода ее решения. Наличие этого этапа обуславливается рядом причин, одна из которых вытекает из свойства неоднозначности естественного языка, на котором описывается постановка задачи. В связи с этим на нем выполняется формализованное описание задачи, то есть устанавливаются и формулируются средствами языка математики логико-математические зависимости между исходными и результатными данными. Математическое описание задачи обеспечивает ее однозначное понимание пользователем и разработчиком программы.

Сложность и ответственность этапа математического описания задачи и выбора (разработки) соответствующего метода ее решения часто требуют привлечения квалифицированных специалистов области прикладной математики, обладающих знанием таких дисциплин, как исследование операций, математическая статистика, вычислительная математика и т. п.

Третий этап технологического процесса подготовки решения задач ЭВМ представляет собой алгоритмизацию ее решения, то есть разработку оригинального или адаптацию (уточнение и корректировку) уже известного алгоритма.

Алгоритмизация – это сложный творческий процесс. В его основу положено фундаментальное понятие математики и программирования – алгоритм – конечный набор правил, однозначно раскрывающих содержание и последовательность выполнения операций для систематического решения определенного класса задач за конечное число шагов.

Любой алгоритм обладает следующими важными свойствами: детерминированностью, массовостью, результатностью и дискретностью.

Детерминированность алгоритма (определенность, однозначность) – свойство, определяющее однозначность результата работы алгоритма при одних и тех же исходных данных. Это означает, что набор указаний алгоритма должен быть однозначно и точно понят любым исполнителем.

Массовость алгоритма – свойство, определяющее пригодность использования алгоритма для решения множества задач данного класса. Оно предполагает возможность варьирования исходными данными в определенных пределах. Свойство массовости алгоритма является определяющим фактором, обеспечивающим экономическую эффективность решения задач на ЭВМ, так как для задач, решение которых осуществляется один раз, целесообразность использования ЭВМ, как правило, диктуется внеэкономическими категориями.

Результатность алгоритма – свойство, означающее, что для любых допустимых исходных данных он должен через конечное число шагов (или итераций) завершить работу.

Дискретность алгоритма – свойство, означающее возможность разбиения определенного алгоритмического процесса на отдельные элементарные действия.

Таким образом, алгоритм дает возможность чисто механически решать любую задачу из некоторого класса однотипных задач.

Составление (адаптация) программ (кодирование) является завершающим этапом технологического процесса разработки программных средств. Он предшествует началу непосредственно машинной реализации алгоритма решения задачи. Процесс кодирования заключается в переводе описания алгоритма на один из доступных для ЭВМ языков программирования. В процессе составления программы для ЭВМ конкретизируются тип и структура используемых данных, а последовательность действий, реализующих алгоритм, отражается посредством конкретного языка программирования.

Этапы тестирования и отладки функционально связаны между собой, хотя их цели несколько отличаются друг от друга. Тестирование представляет собой совокупность действий, назначенных для демонстрации правильности работы программы в заданных диапазонах изменения внешних условий и режимов эксплуатации программы. Цель тестирования заключается в демонстрации отсутствия (или выявлении) ошибок в разработанных программах на заранее подготовленном наборе контрольных примеров.

Процессу тестирования сопутствует понятие «отладка», которое подразумевает совокупность действий, направленных на устранение ошибок в программах, начиная с момента обнаружения фактов ошибочной работы программы и завершая устранением причин их возникновения.

По своему характеру (причине возникновения) ошибки в программах делятся на синтаксические и логические.

Синтаксические ошибки в программе представляют собой некорректную запись отдельных языковых конструкций с точки зрения правил их представления для выбранного языка программирования. (Ошибки выявляются автоматически.)

Далее проверяется логика работы программы на исходных данных. При этом возможны следующие формы проявления *логических ошибок*:

– в какой-то момент программа не может продолжать работу (возникает программное прерывание, обычно сопровождающееся указанием места в программе, где оно произошло);

– программа работает, но не выдает всех запланированных результатов и не выходит на остановку (происходит ее «зацикливание»);

– программа выдает результаты и завершает свою работу, но они полностью или частично не совпадают с контрольными.

После выявления логических ошибок и устранения причин их возникновения в программу вносятся соответствующие исправления и отладка продолжается.

Программа считается отлаженной, если она безошибочно выполняется на достаточно представительном наборе тестовых данных, обеспечивающих проверку всех ее участков (ветвей).

Процесс тестирования и отладки программ имеет итерационный характер и считается одним из наиболее трудоемких этапов процесса разработки программ. По оценкам специалистов, он может составлять от 30 до 50 % в общей структуре затрат времени на разработку проектов и зависит от объема и логической сложности разрабатываемых программных комплексов.

Для сокращения затрат на проведение тестирования и отладки в настоящее время широко применяются специальные программные средства тестирования (например, генераторы тестовых данных) и приемы отладки (например, метод трассировки программ, позво-

ляющий выявлять, все ли ветви программы были задействованы при решении задачи с заданными наборами исходных данных).

После завершения процесса тестирования и отладки программные средства вместе с сопроводительной документацией передаются пользователю для эксплуатации.

Основное назначение сопроводительной документации – обеспечить пользователя необходимыми инструктивными материалами по работе с программными средствами. Состав сопроводительной документации обычно оговаривается заказчиком (пользователем) и разработчиком на этапе подготовки технического задания на программное средство. Как правило, это документы, регламентирующие работу пользователя в процессе эксплуатации программы, а также содержащие информацию о программе, необходимую в случае возникновения потребности внесения изменений и дополнений в нее.

В процессе внедрения и эксплуатации прикладных программных средств могут выявляться различного рода ошибки, не обнаруженные разработчиком при тестировании и отладке программных средств. Поэтому при реализации достаточно сложных и ответственных программных комплексов по согласованию пользователя (заказчика) с разработчиком этап эксплуатации программных средств может быть разбит на два подэтапа: экспериментальный (опытный) и промышленный.

Смысл экспериментальной эксплуатации заключается во внедрении разработанных программных средств на объекте заказчика с целью проверки их работоспособности и удобства работы пользователей при решении реальных задач в течение достаточно длительного периода времени (обычно не менее года) Только после завершения периода экспериментальной эксплуатации и устранения выявленных при этом ошибок и учета замечаний программное средство передается в промышленную эксплуатацию.

Для повышения качества работ, оперативности исправления ошибок, выявляемых в процессе эксплуатации программных средств, также выполнения различного рода модификаций, в которых может возникнуть необходимость в ходе эксплуатации, разработчик может по договоренности с пользователем осуществлять их сопровождение.

1.17. Современные методы разработки ПО: метод нисходящего проектирования, модульное проектирование, CASE-технологии

Метод нисходящего проектирования (метод пошаговой детализации, метод иерархического проектирования, top-down-подход.) Суть метода заключается в определении спецификаций компонентов системы путем последовательного выделения в ее составе отдельных составляющих и их постепенной детализации до уровня, обеспечивающего однозначное понимание того, что и как необходимо разрабатывать и реализовывать.

Этот метод является незаменимым при разработке сложных по характеру и больших по объему программ, когда к их разработке необходимо привлекать большое число программистов, работающих параллельно. Он позволяет концентрировать внимание разработчиков на наиболее ответственных частях программы, а также облегчает возможность постоянного контроля за ее работоспособностью по мере разработки, отладки и объединения отдельных составляющих программ за счет организации непрерывности этого процесса в течение всей разработки.

Для ускорения разработки программного комплекса часто вместо некоторых программ нижнего уровня, находящихся в процессе разработки, могут применяться специальные «программы-заглушки». Они требуются только на ранних стадиях разработки для того, чтобы не сдерживать общий ход создания программного комплекса. Суть «программы-заглушки» заключается в том, что при обращении к ней в соответствии с заданным набором исходных тестовых данных она не формирует, а выбирает результат «решения» из заранее подготовленного набора. Благодаря этому обеспечивается возможность имитировать работу на ЭВМ реально создаваемой программы, следовательно, осуществлять проверку работоспособности программ верхнего уровня еще до того, как будут разработаны и отлажены все составляющие программы нижнего уровня.

Модульное проектирование. Реализация метода нисходящего проектирования тесно связана с другим понятием программирования – модульным проектированием, – так как на практике при декомпозиции сложной программы возникает вопрос о разумном пределе ее

дробления на составные части. Вместе с тем понятие модульности нельзя сводить только к представлению сложных программных комплексов в виде набора отдельных функциональных блоков.

Модуль – это последовательность логически взаимосвязанных фрагментов задачи, оформленных как отдельная часть программы. При этом программные модули должны обладать следующими свойствами:

- на модуль можно сослаться (то есть обращаться к нему) по имени, в том числе и из других модулей;
- по завершении работы модуль должен возвращать управление тому модулю, который его вызывал;
- модуль должен иметь один вход и выход;
- модуль должен иметь небольшой размер, обеспечивающий его обзорность.

При разработке сложных программ выделяют головной управляющий модуль, подчиненные ему модули, обеспечивающие реализацию отдельных функций управления, функциональную обработку (то есть непосредственную реализацию основного назначения программного комплекса), а также вспомогательные модули, обеспечивающие сервисное обслуживание пакета (например, сбор и анализ статистики работы программы, обработка различного рода ошибочных ситуаций, обучение и выдача подсказок и т. п.).

Модульный принцип разработки программ обладает следующими преимуществами:

- большую программу могут разрабатывать одновременно несколько исполнителей, и это позволяет сократить сроки ее разработки;
- появляется возможность создавать и многократно использовать в дальнейшем библиотеки наиболее употребимых программ;
- упрощается процедура загрузки больших программ в оперативную память, когда требуется ее сегментация;
- возникает много естественных контрольных точек для наблюдения за осуществлением хода разработки программ, а в последующем – для контроля за ходом исполнения программ;
- обеспечивается более эффективное тестирование программ, проще осуществляются проектирование и последующая отладка.

Преимущества модульного принципа построения программ особенно наглядно проявляются на этапе сопровождения и модификации

программных продуктов, позволяя значительно сократить затраты сил и средств на реализацию этого этапа.

Структурное программирование. Актуальная для начального периода развития и использования ЭВМ проблема разработки программ, занимающих минимум основной памяти и выполняющихся за кратчайшее время, в последующем в связи с резким падением стоимости аппаратной части ЭВМ, значительным возрастанием их быстродействия и объемов памяти сменилась необходимостью разработки и применения принципиально новых методов составления программ. Все это нашло свое воплощение в разработке принципа структурного программирования. Одной из целей структурного программирования было стремление облегчить разработку и отладку программных модулей, а главное – их последующее сопровождение и модификацию.

В настоящее время структурное программирование – это целая дисциплина, объединяющая несколько взаимосвязанных способов создания ясных, легких для понимания программ. Эффективность применения современных универсальных языков программирования во многом определяется удобством написания с их помощью структурных программ.

CASE-технологии. За последнее десятилетие в области средств автоматизации программирования сформировалось новое направление под общим названием CASE-технологии.

CASE-технология представляет собой совокупность средств системного анализа, проектирования, разработки и сопровождения сложных программных систем, поддерживаемых комплексом взаимосвязанных инструментальных средств автоматизации всех этапов разработки программ. Благодаря структурным методам CASE-технология на стадиях анализа и проектирования обеспечивает разработчиков широкими возможностями для различного рода моделирования, а централизованное хранение всей необходимой для проектирования информации и контроль за целостностью данных гарантируют согласованность взаимодействия всех специалистов, занятых в разработке ПО.

С самого начала CASE-технологии развивались с целью преодоления этих ограничений путем автоматизации процессов анализа

и интеграции поддерживающих средств. Они обладают достоинствами и возможностями, перечисленными ниже.

1. Единый графический язык.

CASE-технологии обеспечивают всех участников проекта, включая заказчиков, единым строгим, наглядным и интуитивно понятным графическим языком, позволяющим получать обозримые компоненты с простой и ясной структурой. При этом программы представляются двумерными схемами (которые проще в использовании, чем многостраничные описания), позволяющими заказчику участвовать в процессе разработки, а разработчикам – общаться с экспертами предметной области, разделять деятельность системных аналитиков, проектировщиков и программистов, облегчая им защиту проекта перед руководством, а также обеспечивая легкость сопровождения и внесения изменений в систему.

2. Единая БД проекта.

Основа CASE-технологии – использование базы данных проекта (репозитория) для хранения всей информации о проекте, которая может разделяться между разработчиками в соответствии с их правами доступа. Содержимое репозитория включает не только информационные объекты различных типов, но и отношения между их компонентами, а также правила использования или обработки этих компонентов. Репозиторий может хранить свыше 100 типов объектов: структурные диаграммы, определения экранов и меню, проекты отчетов, описания данных, логика обработки, модели данных, их организации и обработки, исходные коды, элементы данных и т. п.

3. Интеграция средств.

На основе репозитория осуществляется интеграция CASE-средств и разделение системной информации между разработчиками. При этом возможности репозитория обеспечивают несколько уровней интеграции: общий пользовательский интерфейс по всем средствам, передачу данных между средствами, интеграцию этапов разработки через единую систему представления фаз жизненного цикла, передачу данных и средств между различными платформами.

4. Поддержка коллективной разработки и управления проектом.

CASE-технология поддерживает групповую работу над проектом, обеспечивая возможность работы в сети, экспорт-импорт любых фрагментов проекта для их развития и / или модификации, а также планирование, контроль, руководство и взаимодействие, то есть функ-

ции, необходимые в процессе разработки и сопровождения проектов. Функции также реализуются на основе репозитория. В частности, через репозиторий может осуществляться контроль безопасности (ограничения и привилегии доступа), контроль версий и изменений и др.

5. Макетирование.

CASE-технология дает возможность быстро строить макеты (прототипы) будущей системы, что позволяет заказчику на ранних этапах разработки оценить, насколько она приемлема для будущих пользователей и устраивает его.

6. Генерация документации.

Вся документация по проекту генерируется автоматически на базе репозитория (как правило, в соответствии с требованиями действующих стандартов). Несомненное достоинство CASE-технологии заключается в том, что документация всегда отвечает текущему состоянию дел, поскольку любые изменения в проекте автоматически отражаются в репозитории (известно, что при традиционных подходах к разработке ПО документация в лучшем случае запаздывает, а ряд модификаций вообще не находит в ней отражения).

7. Верификация проекта.

CASE-технология обеспечивает автоматическую верификацию и контроль проекта на полноту и состоятельность на ранних этапах разработки, что влияет на успех разработки в целом: по статистическим данным анализа пяти крупных проектов фирмы TRW (США), ошибки проектирования и кодирования составляют соответственно 64 и 32 % от общего числа ошибок, а ошибки проектирования в 100 раз труднее обнаружить на этапе сопровождения ПО, чем на этапе анализа требований.

8. Автоматическая генерация объектного кода.

Генерация программ в машинном коде осуществляется на основе репозитория и позволяет автоматически построить до 85–90 % объектного кода или текстов на языках высокого уровня.

9. Сопровождение и реинжиниринг.

Сопровождение системы в рамках CASE-технологии характеризуется сопровождением проекта, а не программных кодов. Средства реинжиниринга и обратного инжиниринга позволяют создавать модель системы из ее кодов и интегрировать полученные модели в проект, автоматически обновлять документацию при изменении кодов и т. п.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Практическое занятие № 1

МЕТОДЫ И ПРОГРАММЫ ВИЗУАЛИЗАЦИИ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ НАНОСТРУКТУР

Цель работы: научиться визуализировать атомные системы, полученные в результате моделирования, и манипулировать изображениями.

Используемая программа: программа молекулярной графики RasMol.

Теоретическая часть

Предназначение и роль визуализации атомных систем

Большую роль в познании человеком природы играет визуальное восприятие им окружающего мира. Какими бы большими не были возможности компьютеров, человек способен лучше, чем они, распознавать изображения и делать из этого выводы. Например, человек практически мгновенно может сделать заключение о наличии того или иного дефекта в кристаллическом материале по его изображению, тогда как компьютеру необходимо для этого провести сложные расчеты по определенному алгоритму, специально созданному для распознавания именно этого типа дефекта. Поэтому в атомном компьютерном моделировании структур и процессов в твердых телах широко используются методы и программы визуализации.

Основной принцип визуализации очень прост. Если имеется некоторая система (кристалл или сложная молекула), состоящая из одного или нескольких типов атомов, координаты которых известны, то эту систему можно отобразить на экране, изображая атомы в виде определенных геометрических тел (как правило, в виде сфер) в соответствии с их относительным расположением. При этом каждому типу атомов можно присвоить определенный цвет, можно изобразить связи между ними и т. д. Следует сказать, что современная

техника визуализации позволяет получать не только двумерное изображение на экране компьютера, но также трехмерные изображения в специальных камерах.

Первоначально программы визуализации были созданы для изображения сложных молекул. Одной из программ, предназначенных для этой цели, является программа RasMol, разработанная на кафедре биомолекулярных структур Эдинбургского университета (Великобритания). Официальный сайт (<http://www.umass.edu/microbio/rasmol/index2.htm>) содержит информацию о правилах работы с программой, дистрибутивы последних версий. Популярность этой программы и доступность кодов привела к тому, что на ее основе разработаны новые более функциональные средства визуализации структур (OpenRasmol, ProteinExplorer). В настоящее время имеется множество подобных программ, большинство из которых свободно распространяется. Оказалось, что многие из них пригодны также для визуализации кристаллов, поскольку последние являются большими молекулами. По набору выполняемых функций различные программы близки друг к другу, поэтому, зная принципы визуализации на примере одной из них, легко научиться пользоваться остальными.

В данной работе методы визуализации атомных структур изучаются на примере программы RasMol.

Программа визуализации RasMol

Запуск программы и окна. Программа RasMol считывает с определенным образом записанного файла координаты атомов и создает их графическое изображение. Функционирует в режиме двух окон: графического и текстового (рис. 2.1). В графическом окне происходит визуализация структур макромолекул, а в текстовое окно вводится управляющая информация, которая позволяет изменять масштаб рисунка, цвет, представление молекул, выделять группы атомов, остатков, белковых цепей. Управляющие данные вводятся в виде текстовых команд, описание которых приведено на странице помощи. Графическое окно по умолчанию имеет черный фон. В верхней части этого окна располагается панель меню, имеющая следующие раскрывающиеся меню: «Файл», «Правка», «Вид», «Цвет», «Опции», «Установки», «Экспорт» и «Справка». Кроме этого, основное окошко имеет две полосы прокрутки, которые

используются для поворота изображения вокруг вертикальной и горизонтальной осей. Команды могут быть введены с клавиатуры в текстовом окне, даже если активным является графическое. Это позволяет вводить команды и менять изображения без переключения окон. Для чтения координатного файла можно пользоваться командой «Открыть» меню «Файл». Можно также открыть PDB-файл двойным нажатием на него.

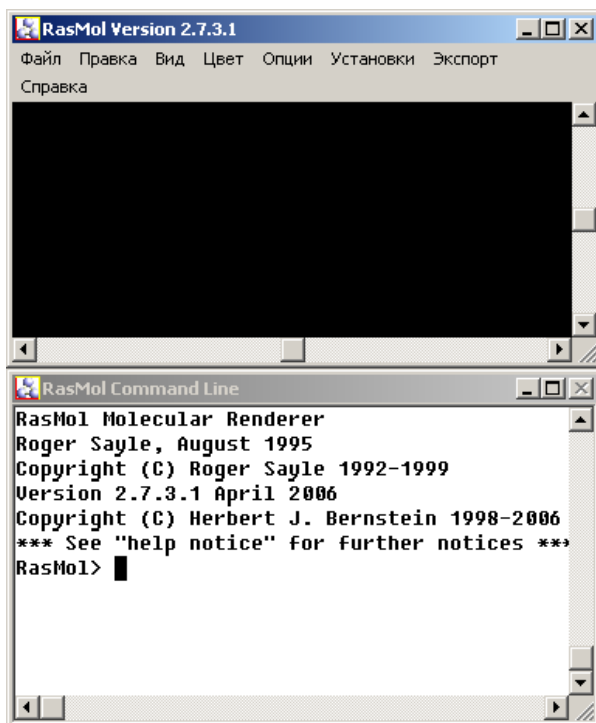


Рис. 2.1. Графическое и текстовое окна программы RasMol

Манипуляции изображением. Вращение молекулы осуществляется передвиганием мыши при нажатой левой кнопке или при помощи полос прокрутки справа и внизу. Вращение в плоскости рисунка осуществляется при нажатой правой кнопке мыши с удержанием клавиши «Shift». Движение мыши при нажатой левой кнопке и удерживаемой клавише «Shift» изменяет масштаб изображения.

Левая кнопка	Поворот X–Y
Правая кнопка	Перемещение X–Y
<i>Shift</i> + левая кнопка	Увеличение
<i>Shift</i> + правая кнопка	Поворот Z

С помощью меню «*Вид*» можно выбрать форму представления изображения. Наиболее используемыми при визуализации кристаллов являются команды «*Ван-дер-Ваальсов радиус*» и «*Атомы и связи*». Первая команда заполняет окно атомами в соответствии с их радиусами. При использовании второй команды атомы представляются в виде сфер небольшого диаметра, с пустым пространством между ними. При этом если расстояние между двумя атомами меньше некоторого значения, между ними рисуется связь в виде прямолинейного отрезка.

Использование команд. С помощью команд RasMol, вводимых в командном окне, можно выполнить значительно большее количество манипуляций изображением, чем с использованием меню графического окна и мыши. Команды вводятся с клавиатуры так же, как команды *dos*. При этом RasMol сохраняет предыдущие команды. Команда «*Ctrl + P*» вызывает из памяти предыдущую команду, а «*Ctrl + N*» – следующую. Эти команды позволяют выбрать формат представления изображения и записать его в файл.

Размеры и расстояния в RasMol могут быть заданы в ангстремах или единицах RasMol. Если значение какой-либо величины содержит десятичную точку, то оно считается заданным в ангстремах. Например, координаты, введенные в PDB-файле с использованием десятичной точки, имеют автоматически размерность ангстрема. Единица RasMol составляет 1/250-ю часть ангстрема. Любая величина, заданная в виде целых чисел, воспринимается в этих единицах. Например, команда «*spacefill 300*» задает сферы радиусом 300 единиц RasMol, то есть 1,2 Å.

Команды программы RasMol: «Backbone Background Bond Cartoon Centre Clipboard Colour Connect CPK CPKnew Define Depth Dots Echo English Exit French HBonds Help Italian Label Load Molecule Monitor Pause Print Quit Refresh Renumber Reset Restrict Ribbons Rotate Save Script Select Set Show Slab Source Spacefill Spanish SSBonds Star Stereo Strands Structure Surface Trace Translate UnBond Wireframe Write Zap Zoom».

Рассмотрим наиболее используемые команды. Подробно о них и об остальных командах можно узнать из руководства к программе.

Background. Формат: `background < colour >` (здесь и далее в угловые скобки заключаются выражения или слова, набираемые по выбору пользователя). Задаёт цвет фона в графическом окне. Можно выбрать цвет, набрав соответствующее английское название (в британской транскрипции). Например, команда «`background white`» устанавливает белый фон. Цвет может быть выбран также заданием интенсивностей трех основных цветов: красного, зеленого и синего. Интенсивность каждого цвета может варьироваться от 0 до 255. Например, команда «`background [255, 0, 0]`» задает красный фон, а команда «`background [100, 100, 100]`» – серый фон (одинаковая интенсивность трех цветов дает различные градации серого цвета). Вместо «`background`» может быть использована также команда «`set background < colour >`», имеющая те же функции.

Clipboard. Позволяет копировать изображение в графическом окне в буфер памяти компьютера, что делает возможным его перенос в другие приложения Windows.

Colour. Формат: `colour < colour >`. Команда позволяет выбирать цвет представления выбранного типа атомов (о команде выбора атомов одного сорта «`select`» см. ниже). Цвета атомов выбираются так же, как для фона.

Print. Позволяет напечатать изображение на принтере, используемом операционной системой по умолчанию.

Quit, Exit. Прекращают выполнение программы и закрывают оба окна.

Rotate. Формат: `rotate < axis > < value >` позволяет поворачивать изображение вокруг выбранной оси на выбранный угол. Параметр `< axis >` может иметь значения «`x`», «`y`» и «`z`». Параметр `< value >` имеет целочисленное значение, равное углу поворота в градусах.

Script. Формат: `script < filename >`. Данная команда последовательно считывает и выполняет команды, записанные в текстовом файле с названием `< filename >`. Пользователь может заранее составить и записать ряд команд, необходимых для приведения изображения к конечному виду, который ему требуется, и выполнить все эти команды заданием одной команды «`script`». Синонимом команды «`script`» является команда «`source`».

Select. Формат: `select < expression >`. Позволяет выбирать определенные атомы в системе. Дальнейшие команды, определяющие, например, цвет, действуют только на выбранные атомы. Выражение `< expression >` может представлять, например, название элемента, атомы которого выбираются (Al, Cu и т. д.). Иногда типы атомов в PDB-файле определяются цифрами; тогда в `< expression >` используется соответствующая цифра. Примеры: «`select Al`», «`select 1`».

Spacefill. Формат: `spacefill < expression >`. Представляет выбранные атомы в виде сплошных сфер с размером, определяемым выражением `< expression >`. Если значение в этом выражении задано целым числом, единицей измерения является единица RasMol, равная 1/250-й части ангстрема. Если число задано с десятичной точкой, размер определяется в ангстремах. Размер атома не должен превышать 500 единиц (2,0 Å); при превышении этого значения программа выдает ошибку «Parameter value too large» («Значение параметра слишком велико») в командной строке.

Wireframe. Формат: `wireframe < value >`. Представляет связи между атомами в виде цилиндров или линий с толщиной, задаваемой значением `< value >` в единицах RasMol (целое число) или ангстремах (с десятичной точкой). Цвет связей задается командой `colour bonds < colour >`. Изображение связей, как правило, является функцией, излишней для визуализации кристаллов, но важной для молекул. Полностью избежать изображения связей можно, только изменив масштабы в координатном файле, умножив координаты на одно и то же число так, чтобы все расстояния между атомами оказались достаточно большими. Командами RasMol можно свести изображение связей к минимуму, задав толщину связей равным 0 (при этом они изображаются тонкими линиями) и цвет, совпадающий с цветом фона.

Write. Формат: write {< format >} < filename >. Записывает изображение в одном из стандартных графических форматов, заданных выражением < format >, в файл с названием < filename >. Поддерживаются многие распространенные графические форматы, как bmp, gif, ps, epsf, vectps (векторный пост-скрипт) и др.

Zoom. Формат: zoom < value >. Изменяет увеличение изображения. Максимальное значение – 1562. Использование команд «zoom on» и «zoom off» позволяет устанавливать и отменять фиксированное увеличение.

Цвет атома или связи может быть задан названием или в виде тройки чисел, задающих интенсивности красного, зеленого и синего (RGB). Предписанные названия цветов и их численные представления даны ниже.

Синий	blue	[0, 0, 256]
Голубой	cyan	[0, 255, 255]
Сине-зеленый	greenblue	[46, 139, 87]
Оранжевый	orange	[255, 165, 0]
Красный	red	[255, 0, 0]
Фиолетовый	violet	[238, 130, 238]
Желтый	yellow	[255, 255, 0]
Черный	black	[0, 0, 0]
Зеленый	green	[0, 255, 0]
Малиновый	magenta	[255, 0, 255]
Пурпурный	purple	[160, 32, 240]
Красно-оранжевый	redorange	[255, 69, 0]
Белый	white	[255, 255, 255]

Цвета CPK. RasMol использует также цветовую схему «CPK», в которой атомы каждого элемента имеют свой цвет, используемый химиками.

Элемент	Element	Colour	Triple
Углерод	Carbon	light grey	[200, 200, 200]
Кислород	Oxygen	red	[240, 0, 0]
Водород	Hydrogen	white	[255, 255, 255]
Азот	Nitrogen	light blue	[143, 143, 255]
Сера	Sulphur	yellow	[255, 200, 50]
Фосфор	Phosphorous	orange	[255, 165, 0]
Хлор	Chlorine	green	[0, 255, 0]
Бром, цинк	Bromine, zinc	brown	[165, 42, 42]
Натрий	Sodium	blue	[0, 0, 255]
Железо	Iron	purple	[160, 32, 240]
Кальций, металл	Calcium, metals	dark grey	[128, 128, 144]
неизвестный	unknown	deep pink	[255, 20, 147]

Формат PDB-файлов

Формат PDB был установлен Брукхавенской национальной лабораторией в США для архивирования структур молекул (The Protein Data Bank). Структура файла содержит полную информацию о молекулах. Однако для визуализации кристаллов может быть использована значительно более простая структура файла. Есть два варианта упрощенной записи.

Вариант 1. Запись с присвоением различным сортам атомов условных названий элементов.

В нескольких первых строках файла записывается количество атомов в системе и другая дополнительная информация, не считываемая программой (эти строки могут отсутствовать). Все последующие строки содержат информацию об отдельных атомах (одна строка на каждый атом). Строка начинается с записи HETATM. Следующие пять позиций отводятся для порядкового номера атома в файле, причем заполнены должны быть правые позиции. Затем, с пропуском двух позиций, печатается название элемента, присвоенное атому (для этого отводятся две позиции, заполняемые слева). Остается 15 пробелов, и печатаются три координаты атомов (на каждую отводятся восемь позиций, три на дробную часть, одна на десятичную точку, эти позиции заполняются справа). Пример заполнения PDB-файла приведен ниже.

3

```
Samplesystem
HETATM 1 H          0.025  72.181  0.880
HETATM 2 Cu        10.025  72.181  20.880
HETATM 3 O          0.025  72.181  4.400
```

Вариант 2. Запись с присвоением номеров различным сортам атомов.

Аналогично первому случаю, в нескольких первых строках могут быть записаны число атомов и другая дополнительная информация. Строки, содержащие информацию об атомах, имеют следующий формат: 1–6 позиции – HETATM; 7–14 – номер атома (занимаются начиная справа); 24-я – тип атома; четыре пробела; три координаты по восемь позиций (три на дробную часть, одна на десятичную точку и четыре на целую часть). Пример такого формата файла приведен ниже.

```
SamplePDBfile
NumberofAtoms = 4
HETATM      1      1  15.693  72.181  0.880
HETATM      3      1  15.693  72.181  7.920
HETATM      4      2  15.693  72.181  11.440
```

Практическая часть

Приведенные ниже упражнения рассчитаны на освоение основных функций программы визуализации на примере просмотра нескольких атомных систем, создаваемых вручную и с помощью программ построения. Необходимые файлы приведены в папке «pract_1».

Упражнение 1. Просмотр изображения готовой атомной системы и манипуляция им.

Запустить программу RasWin2.6 (файл «RasWin2.6.exe»). Открыть файл «Sample_1.pdb». Просмотреть содержимое, открыв в текстовом редакторе «Блокнот» или «WordPad».

Рассмотреть изображение системы в режимах «Spacefill», «Ball&Stick». Рассмотреть его со всех сторон, используя манипуляции мышью. С помощью команд RasMol изменить цвет фона,

представления двух сортов атомов, размеры атомов и другие параметры изображения. Используйте команды «Background», «Colour», «Rotate», «Select», «Spacefill», «Write», «Zoom» и др. В одном из положений объекта визуализации сохранить изображение в графическом формате и использовать для отчета.

Упражнение 2. Создание PDB-файла для визуализации произвольной системы из двух типов атомов

С помощью текстового редактора отредактировать файл Sample_2.pdb, набрав в нем в одном из двух форматов PDB строки для представления системы из атомов двух или трех сортов, расположив атомы так, чтобы они составили какую-либо геометрическую фигуру или изображение Вашего имени. Визуализировать эту систему и привести в отчете распечатку содержания PDB-файла и изображение системы.

Упражнение 3. Визуализация сложной атомной системы.

Открыть PDB-файлы, содержащие вакансию и дислокацию (Sample_vacancy.pdb, Sample_dislocation_1.pdb, Sample_dislocation_2.pdb). Визуализировать дефекты, изучить их и дать описание в отчете. В отчете привести рисунок объектов в наиболее выгодной перспективе.

Требования к содержанию и оформлению отчета

Отчет должен быть оформлен в виде файла Winword и содержать следующие элементы:

- цель работы;
- краткую теорию;
- постановку задач;
- рисунки визуализируемых объектов с пояснениями для каждого упражнения;
- выводы.

Практическое занятие № 2

ОСНОВНЫЕ ПРИНЦИПЫ РАБОТЫ XMD

Цель работы: ознакомиться с методом молекулярной динамики, принципами работы и назначением программы XMD.

Используемая программа: программа XMD.

Теоретическая часть

Для нанотехнологий и наноматериалов наибольший интерес представляют модели, относящиеся к одному из глубочайших уровней описания структуры твердых тел, – атомному. В этих моделях материалы рассматриваются как классические системы взаимодействующих атомов. Основными методами моделирования при этом являются методы молекулярной динамики (МД) – методы моделирования, которые основаны на расчете эволюции системы взаимодействующих частиц (атомов, молекул) путем интегрирования уравнений их движения.

Для того чтобы заниматься моделированием с помощью МД, в настоящее время нет необходимости самому разрабатывать программы, поскольку существует большое множество как коммерческих, так и научных, свободно распространяемых программ. Из коммерческих следует отметить программу Materials Explorer, разработанную компанией Fujitsu. Эта программа, как большинство современных коммерческих программ, отличается хорошим интерфейсом. Кроме того, в пакете имеется весьма богатая коллекция межатомных потенциалов, позволяющая моделировать широкий круг материалов. Однако быстроедействие этой программы недостаточно для моделирования больших систем. Наиболее гибкими являются некоммерческие научные программы с открытыми исходными кодами, в которые можно вводить изменения. Таковыми являются, в частности, программы для однопроцессорных компьютеров Dунано, XMD и для параллельных кластеров Paradyn, LAMMPS, IMD и т. д.

XMD представляет собой одну из программ, реализующих метод МД. Программа разработана Джоном Рифкиным (John Rifkin)

в университете штата Коннектикут (США). В ней могут быть использованы потенциалы метода внедренного атома, потенциал Терсоффа – для кремния и углерода, а также потенциал Стиллингера-Вебера – для кремния. Эта программа распространяется свободно в исходных кодах. На ее веб-странице (<http://xmd.sourceforge.net/about.html>) можно найти также компилированную программу (исполняемый модуль), которая легко запускается в командном окне Windows. Для работы программы требуется входной файл, состоящий из команд XMD. С помощью этого файла можно ввести исходные структуры, задать потенциал, а также режимы моделирования.

В среде Windows программу XMD наиболее удобно использовать с помощью FAR Manager. Работа программы вызывается командой `xmd` с указанием названия командного файла, например, `xmd xmd.in`.

Здесь и далее для определенности входной файл команд XMD будет называться `xmd.in`, хотя для этого может быть использовано любое название и расширение.

Ниже будут приведены основные команды XMD, требуемые для проведения моделирования методом МД. Названия самих команд будут набраны прописными буквами, хотя в командном файле они могут быть набраны и строчными. Названия параметров, требуемых командами, будут набраны строчными буквами.

Ввод основных параметров моделирования

Для задания значения массы атомов используется команда `mass mass`, где `mass` – значение массы атомов в атомных единицах массы (например, при моделировании Ni используется команда `MASS 27`).

Для задания шага по времени используется команда `dtime dtime`, где `dtime` – значение шага по времени в секундах. Например, команда `dtime 5.e-15` устанавливает шаг, равный 5 фс.

Можно выбрать единицу измерения энергии, которая далее будет использована при любых вводах-выводах энергетических величин: `eunit uname`, где `uname` = `erg`, `joule`, `K`, `eV`. Чаще всего единица энергии вводится при чтении межатомного потенциала, который может быть табулирован в любой из этих единиц. Поэтому, как правило, команда `eunit` ставится в первой строке файла с таблицей потенциала.

Для чтения команд из любого другого текстового файла используется команда *read filename*, где *filename* – название файла, содержащего набор команд. Удобно бывает выделять в отдельные файлы команды, вводящие потенциал взаимодействия и исходные координаты атомов. Тогда в командном файле *xmd.in* будут команды чтения из этих файлов, например, *read ni.ptf* и *read ni.in*, где *ni.ptf* – файл, содержащий описание потенциала для никеля, а *ni.in* – файл, содержащий координаты атомов кристаллической решетки никеля. Для многих металлов доступны уже готовые таблицы потенциалов, тогда как исходные структуры чаще всего приходится создавать самим. Для описания исходной конфигурации используются команды *position* или *particle* (они являются синонимами). Формат этой команды следующий. В первой строке задается *position n*, где *n* – число частиц. В последующих *n* строках вводятся типы и координаты всех частиц: первое число обозначает тип атома (если моделируется чистый металл, то цифра 1, если сплав двух элементов – то цифры 1 и 2), три следующие – координаты *x*, *y*, *z* атома в ангстремах. Например, следующие строки вводят исходную конфигурацию системы из трех атомов: *READ coord.in*.

Файл *coord.in* содержит следующую информацию.

```
position 3
1 0.24E-01      0.76E+02      0.88E+00
1 0.52E-01      0.16E+02      0.11E+00
1 0.34E+00      0.76E+01      0.88E+01
```

Особенность программы XMD заключается в том, что все частицы должны находиться в первом октанте системы координат. То есть, левый нижний угол расчетной ячейки всегда совпадает с началом координат. Для задания расчетной ячейки используется команда *box xbox ybox zbox*, где *xbox ybox zbox* определяют координаты правого верхнего угла расчетной ячейки в ангстремах. Как правило, параметры расчетной ячейки рассчитываются при построении исходной структуры системы, поэтому их удобно записывать в одном файле с координатами атомов.

Начальная температура системы задается командой *ITEMP temp*, где *temp* – значение температуры в кельвинах.

Для поддержания температуры в процессе моделирования используется команда *clamp temp [cstep]*. Значение *temp* задает требуемую

температуру. На каждом шаге МД скорости частиц умножаются на величину $(temp/T)*(1/(2*cstep))$, где T – мгновенная температура на данном шаге.

По умолчанию границы расчетной ячейки при моделировании в ХМД фиксированы. Для моделирования при постоянном давлении используется команда *pressure clamp bulkmodulus [cstep]*. При использовании этой моды стенки расчетной ячейки флуктуируют с амплитудой, обратно пропорциональной объемному модулю упругости *bulkmodulus*. Значение последнего задается в мегабарах (1 Мбар = 100 ГПа). Не обязательно его указывать точно, достаточно задать приближенное значение. Параметр *cstep*, который может отсутствовать, определяет, как быстро флуктуирует размер ячейки в ответ на внутреннее давление. Если с помощью этой команды динамические граничные условия включены, то можно задать внешнее давление командой *pressure external pressure X [pressure Y pressure Z]*. Можно задать только одно значение давления (МБар), тогда будет задано изотропное давление. Задав три значения, можно определить независимые давления вдоль трех осей координат.

Команды, контролирующие процесс моделирования

Процесс моделирования контролируется с помощью двух основных команд: *quench* и *cmd*.

Команда *quench nstep [nquench]* (закалка) совершает *nstep* шагов релаксации при абсолютном нуле для нахождения конфигурации с минимумом энергии. Параметр *nquench* задает способ отъема кинетической энергии при релаксации. При *nquench* = 1 (значение по умолчанию) скорости всех атомов полагаются равными нулю, если потенциальная энергия системы превышает значение в предыдущем шаге. При *nquench* = 2 зануляется скорость только той частицы, энергия которой увеличивается.

Команда *cmd nstep* предписывает проведение *nstep* шагов МД. Условия МД описываются приведенными выше командами.

При выполнении этих двух команд часто удобно сохранение промежуточных результатов или выполнение других действий или различное сочетание процессов МД и релаксации. В этих случаях используется цикл команд, который начинается командой *repeat n* и заканчивается командой *end*. Между этими командами может

располагаться любая последовательность команд, которая будет исполнена n раз. Например, цикл

```
repeat 10
  cmd 100
  quench 200
end
```

десять раз выполняет последовательность из 100 шагов МД и 200 шагов релаксации.

Команды вывода данных

XMD предоставляет богатый выбор возможностей вывода различных величин в процессе моделирования. Ниже рассмотрим основные из команд.

Bsave nskip file. При моделировании с командой *cmd* указывает программе сохранять каждые *nskip* шагов в файле с названием *file* размеров расчетной ячейки. В каждой строке записываются четыре числа: номер шага и три размера ячейки по осям x , y , z . Использование этой команды удобно для мониторинга размеров ячейки при моделировании с динамическими границами, то есть при постоянном давлении.

Esave nskip file. Эта команда дает указание каждые *nskip* шагов МД или релаксации записывать в файл с названием *file* значений энергии. Сохраняется средняя энергия, приходящаяся на один атом системы в единице, установленной командой *eunit*. Каждый вывод осуществляется в одну строку, в которой записываются четыре числа: номер шага, полная энергия, потенциальная и кинетическая энергия.

Команда *write* – наиболее универсальная команда для вывода информации при моделировании. Она имеет многочисленные опции, указывающие выводимые величины и названия файлов, а также режимы вывода. Один из наиболее распространенных форматов этой команды приведен ниже.

```
write file [+]filename quantity
```

Здесь *quantity* – название выводимой величины, которая будет записана в файл с названием *filename*. Перед названием файла может быть поставлен знак «+»; тогда запись будет производиться начиная с конца этого файла. В противном случае все содержимое файла будет заменено записываемой информацией.

Рассмотрим некоторые важные значения опции *ouantity*.

Quantity = box. Команда *write file filename box* записывает размеры расчетной ячейки.

Quantity = eatom. Записывает номера и значения энергий всех атомов системы.

Quantity = stress. Произведения компонент тензора внутренних напряжений, усредненных по всем атомам системы, на атомный объем. Напряжение определяется делением этого произведения на атомный объем.

Quantity = particle. Записывает типы и координаты атомов системы. Координаты выводятся в ангстремах. Удобна для сохранения релаксированной конфигурации системы, если используется режим моделирования *quench*.

Quantity = posvel. Сохраняет типы, координаты (в ангстремах) и скорости (в см/с) всех атомов системы. Команда удобна для сохранения промежуточных или конечных состояний системы при ненулевых температурах, моделируемых командой *cmd*.

В XMD предусмотрена также возможность сохранения всей текущей информации, с использованием которой моделирование может быть продолжено с прерванного места. Это делается с помощью команды *write state file*. Ранее сохраненное состояние системы считывается с помощью команды *state file*. В командном окне *dos* выполнение программы XMD может быть прервано нажатием клавиш «*Cntrl + C*». При поступлении сигнала на прерывание прежде всего создается файл состояния системы на момент прерывания, которому дается название *stop.sta*. В дальнейшем моделирование может быть продолжено с этого шага.

Команды *write xmol [+file]* и *write pdb [+file]* позволяют записывать координаты атомов в форматах XMOl и PDB, которые могут быть открыты программами визуализации, в частности, программой RasMol.

Требования к содержанию и оформлению отчета

- Отчет должен быть оформлен в виде файла Winword и содержать:
- теорию (особенности программы XMD, основные команды программы и их назначение);
 - выводы.

Практическое занятие № 3

МОДЕЛИРОВАНИЕ ПЛАВЛЕНИЯ МЕТАЛЛИЧЕСКИХ НАНОЧАСТИЦ И КЛАСТЕРОВ

Цель работы: определить температуру плавления металлических наночастиц и кластеров и сопоставить с температурой плавления объемного кристалла.

Используемые программы:

- программа XMD;
- программа создания расчетной ячейки для моделирования наночастиц и кластеров;
- набор табулированных потенциалов для г. ц. к. металлов;
- программа визуализации атомных структур RasMol;
- программа построения графиков функций.

Теоретическая часть

Наночастицы и кластеры

Под кластерами и наночастицами понимаются частицы твердых тел, содержащие от нескольких до сотен атомов и тысячи и десятки тысяч атомов соответственно. Граница между наночастицами и кластерами условная. В наночастицах (размер их составляет величину порядка 10 нм) число поверхностных атомов отличается от числа объемных лишь на порядок, а в кластерах эти числа сопоставимы, или даже все частицы могут быть отнесены к поверхностным. В дальнейшем наночастицы и кластеры будем объединять под общим названием малых частиц.

Из-за большой доли поверхностных атомов физические свойства малых частиц существенно изменены по сравнению с массивными образцами. Отличие свойств малых частиц от свойств массивных

образцов уже используется в самых разнообразных технических приложениях. Например, порошки из малых частиц работают в качестве катализаторов несравненно лучше, чем массивные образцы из тех же материалов. Введение малых металлических частиц внутрь керамических материалов придает этим материалам (керметам) уникальные механические свойства, из-за чего они используются в авиационной и космической технике.

Сильное влияние поверхности, существенное изменение электронной структуры наночастиц приводят к существенным отличиям термодинамических свойств малых частиц от объемных кристаллов. К таким свойствам относится, например, температура плавления.

Температура плавления объемных кристаллов от их размера не зависит. Однако когда размеры материала уменьшаются до наноразмерного уровня, температура плавления начинает зависеть от размера и уменьшается с ним. Уменьшение температуры плавления может составить несколько сотен градусов.

Для большинства материалов изменение T начинается в области диаметра частиц 50 нм, ниже которого происходит резкое уменьшение температуры плавления (рис. 3.1).

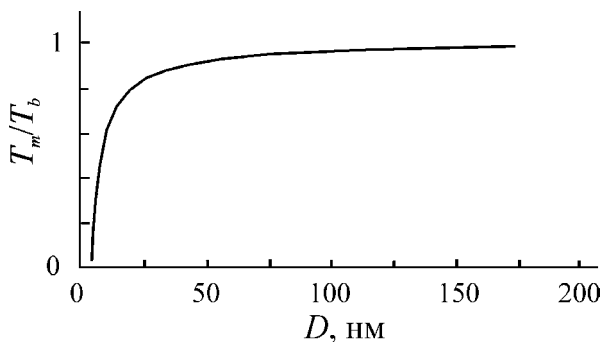


Рис. 3.1. Кривая зависимости нормированной температуры плавления золота от диаметра наночастиц D (T_b – температура плавления объемного кристалла)

Радиальная функция распределения

Важнейшей характеристикой атомной структуры конденсированных сред является радиальная функция распределения (РФР).

Она характеризует корреляцию в расположении частиц газа, жидкости или твердого тела и определяется нижеприведенным образом.

Пусть система из N одинаковых атомов занимает объем V . Выберем в нем два элемента объема dV_1 и dV_2 , фиксируемые радиус-векторами \vec{R}_1 и \vec{R}_2 (рис. 3.2). Если взаимное расположение атомов хаотическое, то есть некоррелированное, то вероятность того, что атом 1 находится в элементе объема dV_1 , а атом 2 в то же время находится в элементе dV_2 , из-за независимости их положений равна

$$dP(\vec{R}_1, \vec{R}_2) = \frac{dV_1}{V} \frac{dV_2}{V}.$$

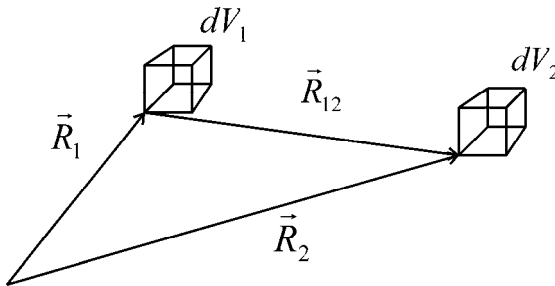


Рис. 3.2. К определению радиальной функции распределения

По результатам компьютерного моделирования РФР можно рассчитать следующим образом. Положения всех атомов рассматриваемой системы записывают в файл. Из атомов составляют всевозможные пары и рассчитывают расстояния между частицами в парах. Промежуток расстояний от 0 до R_{\max} , максимального расстояния, разделяющего пары, делится на малые интервалы (R_i, R_{i+1}) , $i = 1, 2, \dots, K$. Подсчитывается количество пар n_i , расстояние между которыми лежит в каждом из этих интервалов. Если полное число пар равно n , то среднее число пар, приходящихся на каждый интервал, будет равно n/K . Тогда отношение действительного количества пар в интервале n_i к среднему числу пар и будет представлять собой относительную вероятность того, что расстояния между парами атомов будут заключены в этом интервале: $f_i = n_i / p_0$. Можно построить гистограмму этой функции, которая

и будет представлять собой приближение к РФР. Чем больше число атомов в системе и меньше отрезки, на которые разбивается область изменения расстояний, тем ближе будет эта гистограмма к непрерывной функции $g(R)$.

Определение температуры плавления объемных материалов и наночастиц

Ввиду большой роли поверхностных атомов, обладающих повышенной энергией, в наночастицах при нагреве возможно гетерогенное зарождение жидкой фазы вблизи поверхности, особенно у вершин многогранников, которые образуют частицы. Поэтому фазовые переходы плавления и кристаллизации в малых частицах можно моделировать их непосредственным нагреванием или охлаждением. Для того чтобы при нагревании образовался зародыш жидкой фазы, произошло плавление всего кристалла и установилось равновесие в жидкой фазе, необходимо достаточно длительное, но вполне доступное в масштабах моделирования время. Например, для кластеров, содержащих до 1000 атомов, достаточно порядка 4×10^4 шагов, чтобы происходило плавление. На основании результатов моделирования малых частиц при различных температурах строится калорическая кривая. По скачку в калориметрической кривой можно определить температуру плавления $T_m(d)$.

Алгоритм моделирования приведен ниже. Строится исходная модель частицы с заданной формой и размером. Производится релаксация (минимизация энергии) при $T = 0$ К. Затем производится релаксация системы при разных постоянных значениях температуры. При температурах, далеких от температуры плавления, обычно бывает достаточной релаксация в интервале времени порядка 30–50 пс. После этого определяется полная энергия атомной системы, равная сумме потенциальной и кинетической энергий всех атомов, которая представляет собой внутреннюю энергию малой частицы. Прорядав это для ряда значений температуры, строят калорическую кривую, анализ которой и позволяет найти температуру плавления.

Пример такого исследования приведен на рис. 3.3, на котором изображены калорические кривые малых частиц золота, содержащих 219, 477, 879 атомов, и объемного кристалла (ввиду описанного выше

отличия плавления малых частиц и объемного кристалла, для последнего калориметрическая кривая определена другим способом).

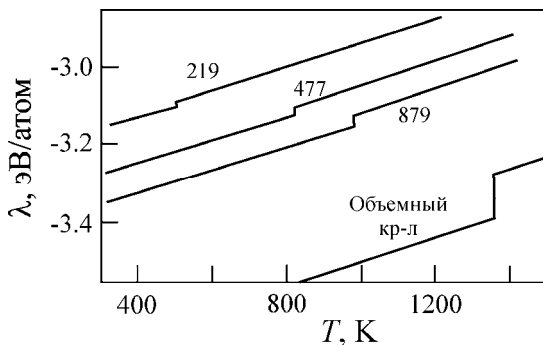


Рис. 3.3. Внутренняя энергия как функция температуры для малых частиц и объемного кристалла золота

Практическая часть

Задания

1. Для того чтобы получить компьютерную модель малой частицы, воспользуйтесь программой построения `nanocluster.exe`. Постройте и сохраните малую частицу никеля с радиусом $R = 1,5$ нм.

2. Напишите список команд для достижения равновесного состояния системы при заданной температуре и определения внутренней энергии при этой температуре. Внутреннюю энергию следует определять усреднением по достаточно большому числу временных шагов после достижения равновесия (оценить среднее значение можно, построив график зависимости от времени полной энергии в процессе моделирования). При температурах, заведомо более низких или более высоких, чем точка плавления кристалла, количество временных шагов может быть маленьким, но в области температуры плавления следует взять как можно большее число шагов МД (не менее 30 тысяч).

3. Используя команду `xmd`

```
write file filename RDF nbin rmin rmax,
```

рассчитайте радиальную функцию распределения (РФР) для двух состояний малой частицы: твердой и жидкой. Эта команда рассчитывает и выводит в файл с названием `filename` число атомных пар

с расстоянием, заключенным в каждом из $nbin$ подинтервалов, на которые делится интересующий нас отрезок расстояний между значениями в англстремах $lmin$ и $lmax$, которые указывает пользователь. В указанный файл выводится следующая информация: на первой строке печатаются слово RDF, значения $nbin$, $lmin$ и $lmax$; на каждой из последующих строк печатаются значение расстояния, $(R_k + R_{k+1})/2$, число пар M_k и кумулятивная сумма чисел пар во всех интервалах от 1 до k . Для того чтобы рассчитать РФР по этим результатам, необходимо воспользоваться формулой

$$g(R_i) = \frac{\Delta M_i}{M} \frac{V}{4\pi R_i^2 \Delta R_i},$$

где $M = N(N-1)/2$ – число пар в системе;

$$V = 4\pi R^3 / 3.$$

4. Для построенной модели малой частицы провести моделирование для значений температуры $T = 400, 600, 800, 1000, 1200, 1350, 1400, 1600, 1800$ и 2000 К.

5. Построить график зависимости внутренней энергии от температуры (калорическую кривую) и по нему определить приближительное значение температуры плавления малой частицы.

6. Сопоставить полученное значение с температурой плавления объемного кристалла никеля, который составляет $T_b = 1740$ К.

Требования к содержанию и оформлению отчета

Отчет должен быть оформлен в виде файла Winword и содержать:

- краткую теорию;
- постановку задач;
- командный файл с комментариями;
- графики изменения полной внутренней энергии частиц при каждой температуре с числом шагов и равновесное значение энергии;
- калорическую кривую и рассчитанное значение температуры плавления;
- выводы.

Практическое занятие № 4

МОДЕЛИРОВАНИЕ НАНОКРИСТАЛЛИЗАЦИИ АМОРФНЫХ МЕТАЛЛОВ ПРИ ДЕФОРМАЦИИ

Цель работы: изучить изменения атомной структуры аморфных металлов при деформации; выработать навыки самостоятельного изучения первоисточников по моделированию процессов нанотехнологии, опубликованных в научных журналах на английском языке.

Используемые программы:

- программа XMD;
- программа создания расчетной ячейки г. ц. к. кристалла;
- набор табулированных потенциалов для г. ц. к. металлов;
- программа визуализации атомных структур RasMol.

Используемый источник: статья Lee B.-J., Lee C.S., Lee J.C. Stress induced crystallization of amorphous materials and mechanical properties of nanocrystalline materials: a molecular dynamics simulation study. – ActaMaterialia, 2003. – V. 51. – P. 62336240.

Теоретическая часть

Кристаллизация аморфных сплавов является одним из важных процессов нанотехнологии, поскольку она позволяет управлять их структурой на наноразмерном уровне. Этот процесс лежит в основе весьма распространенного метода получения нанокристаллов. При определенных условиях отжига, благоприятствующих образованию центров кристаллизации, аморфные сплавы кристаллизуются с образованием нанокристаллов. Контролируя условия отжига, можно изменять размеры зерен от 2–3 до 100 нм и более.

Ряд экспериментальных исследований показал, что нанокристаллизация аморфных сплавов может быть активизирована путем пластической деформации. Имеются также данные молекулярно-динамического моделирования кристаллизации аморфных сплавов при деформации.

Данная практическая работа по сути является повторением одной из таких работ. Для ее выполнения следует самостоятельно изучить статью *используемого источника* в оригинале.

Практическая часть

В указанной работе изучается кристаллизация аморфного никеля при деформации растяжением и механизмы пластической деформации образовавшегося нанокристалла.

В качестве исходной структуры подготавливают образец аморфного никеля в виде пластинки размерами $8 \times 10 \times 2 \text{ нм}^3$ (в направлениях x , y , z). Для этого кристаллический никель выдерживают в течение нескольких десятков пикосекунд при температуре выше температуры плавления, затем система быстро охлаждается до температуры 300 К. Для моделирования деформации растяжением образец упруго удлиняется на 1 % в направлении оси x , после чего проводится МД-релаксация в течение 10 пс при фиксированном новом размере образца в этом направлении. Растяжение моделируется повторением упругого удлинения и МД-релаксации. Скорость деформации, соответствующая этой процедуре, $\dot{\epsilon} = 10^{-2}/10^{-11} \text{ с} = 10^9 \text{ с}^{-1}$, очень велика по сравнению с экспериментально используемыми скоростями, что обусловлено недостатком МД – ограниченным интервалом времени, доступным для моделирования. Все расчеты проводятся при температуре $T = 300 \text{ К}$.

При подготовке аморфного образца моделирование осуществляется при периодических граничных условиях по всем трем направлениям; при моделировании деформации используются периодические граничные условия в направлениях x и z , а в направлении y система имеет свободные поверхности и конечный размер, который может меняться в процессе деформации. Размер ячейки в направлении z фиксирован. Таким образом, моделируется ударная деформация растяжением аморфной тонкой пленки, бесконечно протяженной в направлениях осей x и z и имеющей толщину 10 нм по оси y , в направлении оси x в условиях плоской деформации.

В работе использованы два потенциала, основанные на модифицированном методе погруженного атома, что позволило авторам определить влияние потенциала на процесс. Явление кристаллизации наблюдалось в обоих случаях, но имелись количественные различия, связанные с отличием потенциалов. Поскольку задача данной лабораторной работы – качественное повторение результатов

используемого источника, будет использован один потенциал, построенный на простом методе погруженного атома.

Задания

Основываясь на приведенном выше изложении и изучении оригинальной статьи и используя подготовленные для выполнения работы программы построения компьютерной модели идеального г. ц. к. кристалла, расчета РФР и примерные командные файлы для создания модели аморфного металла и его деформации, самостоятельно проделать описанные исследования.

Для моделирования процесса деформации рекомендуется воспользоваться командой SCALEabc, которая умножает на величины a , b и c соответственно x -, y - и z - координаты всех частиц и размеры расчетной ячейки в соответствующих направлениях. Для того чтобы растянуть аморфный «образец» вдоль оси x , а в других направлениях размеры оставить неизменными, можно положить $a > 1$, $b = c = 1$. Составить цикл команд из 40 шагов, в каждом из которых образец растягивается на небольшую величину, после этого производится МД-релаксация в течение не менее чем 2000 шагов так, чтобы общая деформация составила величину порядка 1,5.

Требования к содержанию и оформлению отчета

Отчет по практической работе должен подробно описывать все этапы исследования.

1. Построение модели аморфного металла, доказательства аморфного состояния полученной модели.
2. Моделирование деформации, полученные структуры при нескольких значениях деформации.
3. Радиальные функции распределения образцов, подвергнутых деформации 40 % и статической выдержке.
4. Выводы из результатов исследования.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Советов, Б. Я. Информационные технологии / Б. Я. Советов. – Москва: Высшая школа, 2003. – 263 с.
2. Шатунова, О. В. Информационные технологии : учебное пособие / О. В. Шатунова. – Елабуга: ЕГПУ, 2007. – 77 с.
3. Гусев, А. И. Нанокристаллические материалы / А. И. Гусев, А. А. Ремпель. – Москва: Физматлит, 2001. – 224 с.
4. Назаров, А. А. Атомистическое моделирование материалов, наноструктур и процессов нанотехнологии : учебное пособие для студентов-физиков / А. А. Назаров, Р. Р. Мулюков. – Уфа: БашГУ, 2010. – 156 с.
5. Основы современных компьютерных технологий : учебное пособие / под ред. А. Д. Хомоненко. – Санкт-Петербург: Корона-Принт, 2002. – 448 с.
6. Орликов, Л. Н. Технология материалов и изделий электронной техники : учебное пособие / Л. Н. Орликов. – Томск: Тусур, 2006. – 364 с.
7. Семененко, М. Г. Введение в математическое моделирование / М. Г. Семененко. – Москва: Со-лон-Р, 2002. – 112 с.
8. Нано- и микросистемная техника. От исследований к разработкам : сборник статей / под ред. П. П. Мальцева. – Москва: Техносфера, 2005. – 582 с.
9. Яшин, К. Д. Системы автоматизированного проектирования МЭМС / К. Д. Яшин, Е. В. Лацапнев, В. С. Осипович // Информационные технологии. – 2007. – № 11. – С. 22–28.
10. Зинченко, Л. А. Конспект лекций по блоку дисциплин «САПР наносистем» // Библиотека Наноинженерии. – Москва: МГТУ им. Н. Э. Баумана, 2008. – 30 с.

Учебное издание

ЩЕРБАКОВА Елена Николаевна

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
НАНО- И МИКРОСИСТЕМНОЙ
ТЕХНИКИ**

Учебно-методическое пособие
для студентов специальности 1-38 01 04
«Микро- и наносистемная техника»

Редактор *Т. В. Грищенкова*
Компьютерная верстка *Е. А. Беспанской*

Подписано в печать 19.10.2017. Формат 60×84 ¹/₈. Бумага офсетная. Ризография.
Усл. печ. л. 4,59. Уч.-изд. л. 3,59. Тираж 100. Заказ 802.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет.
Свидетельство о государственной регистрации издателя, изготовителя, распространителя
печатных изданий № 1/173 от 12.02.2014. Пр. Независимости, 65. 220013, г. Минск.