

Лекция №11

Раздел 2. Этапы построения математической модели.

Реализация математической модели в виде программы для ЭВМ

В настоящее время математические модели, как правило, применяют в качестве основы для создания различных программных комплексов, используемых для решения разнообразных исследовательских, проектно-конструкторских и управленческих задач. Данное обстоятельство приводит к необходимости реализации модели в виде программы для ЭВМ. Процесс разработки надежного и эффективного программного обеспечения является не менее сложным, чем разработка предыдущих этапов создания математической модели. Успешное решение данного вопроса возможно лишь при уверенном владении современными алгоритмическими языками и технологиями программирования, имеющегося программного обеспечения, знании возможностей вычислительной техники, особенностей реализации на ЭВМ методов вычислительной математики, опыта решения подобных задач.

Процесс создания программного обеспечения можно разбить на ряд этапов [15]:

- разработка технического задания на создание программного обеспечения;
- проектирование структуры программного комплекса;
- кодирование алгоритма;
- тестирование и отладка;
- сопровождение и эксплуатация.

Техническое задание на разработку программного обеспечения оформляют в виде *спецификации*. Примерная форма спецификации включает следующие семь разделов.

1) Название задачи

Дается краткое определение решаемой задачи, название программного комплекса, указывается система программирования для его реализации и требования к аппаратному обеспечению (компьютеру, внешним устройствам и т.д.).

2) Описание

Подробно излагается математическая постановка задачи, описывается применяемая математическая модель для задач вычислительного характера, метод обработки входных данных для задач не вычислительного (логического) характера и т.д.

3) Управление режимами работы программы

Формируются основные требования к способу взаимодействия пользователя с программой (интерфейс "пользователь-компьютер").

4) Входные данные

Описываются входные данные, указываются пределы, в которых они могут изменяться, значения, которые они не могут принимать, и т.д.

5) Выходные данные

Описываются выходные данные, указывается, в каком виде они должны быть представлены - в числовом, графическом или текстовом, приводятся сведения о точности и объеме выходных данных, способах их сохранения и т.д.

б) Ошибки

Перечисляются возможные ошибки пользователя при работе с программой, (например, ошибки при вводе входных данных). Указываются способы диагностики (в данном случае под диагностикой понимается выявление, обнаружение ошибок при работе программного комплекса) и защиты от этих ошибок на этапе проектирования, а также возможная реакция пользователя при совершении им ошибочных действий и реакция программного комплекса (компьютера) на эти действия.

7) Тестовые задачи

Приводится один или несколько тестовых примеров, на которых в простейших случаях проводится отладка и тестирование программного комплекса.

Пример: Спецификация задачи о баскетболисте

1) Название задачи

Название программы **Basketball**

Система программирования **Delphi**

Компьютер **IBM PC Pentium**

Операционная система **Windows 10**

2) Описание

Приводится математическая постановка задачи и описание метода ее решения.

3) Управление режимами работы программы

Для управления режимами работы программы необходимо использовать интерфейс **Windows** с использованием меню, диалоговых окон, полей ввода данных, кнопок.

4) Входные данные

Входными данными являются радиус и масса мяча, его начальные координаты и скорость, угол бросания, координаты корзины.

5) Выходные данные

Траектория центра мяча, расчетная величина дальности и точность броска.

б) Ошибки

При вводе исходных данных предусмотреть контроль:

- все вводимые значения должны быть положительны;
- угол бросания лежит в пределах от 5 до 85 градусов;
- начальная скорость мяча лежит в пределах от 0 до 30 м/с;
- горизонтальная координата центра корзины больше начальной горизонтальной координаты мяча.

При диагностировании перечисленных ошибок программа должна выдавать соответствующие сообщения, которые могут сопровождаться звуковым сигналом, и предлагать повторить ввод.

7) Тестовые примеры

При $x_0 = y_0 = y_k = 0$; $x_k = 4,225$; $v_0 = 6,44$; $\alpha = 45^0$;

Получаем: $L = 4,225$; $\Delta = 0$.

На этапе проектирования формируется общая структура программного комплекса. Вся программа разбивается на программные модули. Для каждого программного модуля формулируются требования по реализуемым функциям и разрабатывается алгоритм, выполняющий эти функции. Определяется схема взаимодействия программных модулей, называемая схемой потоков данных программного комплекса. Разрабатывается план и исходные данные для тестирования отдельных модулей и программного комплекса в целом.

Большинство программ, реализующих математические модели, состоят из трех основных частей:

- *препроцессор* (подготовка и проверка исходных данных модели);
- *процессор* (решение задачи, реализация вычислительного эксперимента);
- *постпроцессор* (отображение полученных результатов).

Лишь для относительно простых случаев данные три составные части могут быть оформлены в виде одной программы. При решении современных задач по моделированию поведения жидкостей, газов и твердых тел каждая из частей может включать в себя целый комплекс программ. Например, постпроцессор должен уметь представлять информацию не только в табличном, но и графическом виде (диаграммы, графики зависимости от различных параметров, отображение скалярных, векторных (тензорных) полей и т.п.). Особенно сильное развитие возможности пре- и постпроцессора получают в современных системах автоматизированного проектирования (САПР), где они в значительной степени могут сократить время на получение данных и оценку результатов моделирования.

Как правило, создание современных математических моделей в какой-либо области и доведение их до программных комплексов требует значительных временных затрат (минимум 3-5 лет). Требуется время не только на освоение методик и подходов к моделированию в исследуемой области, но и время на наработку библиотек программ по решению возникающих математических задач, по подготовке исходных данных и отображению получаемых

результатов. Качественные, надежные, обладающие дружественным интерфейсом, легко модифицируемые и хорошо сопровождаемые программные комплексы можно создавать лишь при наличии хорошо продуманной стратегии развития программного обеспечения, обеспечивающей его модульность и совместимость по входным и выходным параметрам.

Большое значение следует придавать освоению современных технологий программирования: структурной, абстрактной, объектно-ориентированной и визуальной. Назначение любой технологии - это, в первую очередь, повышение надежности программного обеспечения и увеличение производительности труда программиста. Причем, чем серьезней и объемней программный проект, тем большее значение приобретают вопросы использования современных технологий программирования. Пренебрежение данными вопросами может привести к значительным временным издержкам и снижению надежности программного комплекса.

Важнейшим фактором, определяющим надежность и малые сроки создания программного комплекса для решения определенного класса задач, является наличие развитой библиотеки совместимых между собой программных модулей. Программа получается более надежной и создается за меньшие сроки при максимальном использовании стандартных программных элементов. Для эффективной разработки программного обеспечения в области математического моделирования в первую очередь следует обратить внимание на создание следующих стандартных библиотек программ:

- приближенные и численные методы;
- средства подготовки исходных данных (препроцессоры);
- средства визуализации и представления результатов (постпроцессоры).

Разработка таких общих библиотек программ возможна лишь при стандартизации потоков передачи данных между препроцессором, процессором и постпроцессором. В простейшем случае речь может идти об унификации форматов передаваемых файлов.

Проверка адекватности модели

Под *адекватностью* математической модели будет пониматься степень соответствия результатов, полученных по разработанной модели, данным эксперимента или тестовой задачи. Прежде чем переходить к проверке адекватности модели, необходимо убедиться в правильном комплексном функционировании всех алгоритмов и программ модели, выполнить независимое тестирование и отладку всех отдельных алгоритмов (например, используемых программных модулей, реализующих используемый численный метод).

Проверка адекватности модели преследует две цели:

1) Убедитесь в справедливости совокупности гипотез, сформулированных на этапах концептуальной и математической постановок. Переходить к проверке гипотез следует лишь после проверки использованных методов решения, комплексной отладки и устранения всех ошибок и конфликтов, связанных с программным обеспечением.

2) Убедитесь, что точность полученных результатов соответствует точности, оговоренной в техническом задании.

Проверка разработанной математической модели выполняется путем сравнения с имеющимися экспериментальными данными о реальном объекте или с результатами других, созданных ранее и хорошо себя зарекомендовавших моделей. В первом случае говорят о проверке путем сравнения с экспериментом, во втором - о сравнении с тестовой задачей.

Решение вопроса о точности моделирования зависит от требований, предъявляемых к модели, и ее назначения. При этом должна учитываться точность получения экспериментальных результатов или особенности постановок тестовых задач. Для моделей, предназначенных для выполнения оценочных и прикидочных расчетов, удовлетворительной считается точность в 10-15%. Для моделей, предназначенных для использования в управляющих и контролирующих системах, требуемая точность может быть 1-2% и даже выше.

Как правило, различают качественное и количественное совпадение результатов сравнения. При качественном сравнении требуется лишь совпадение некоторых характерных особенностей в распределении исследуемых параметров (например, наличие экстремальных точек, положительное или отрицательное значение параметра, его возрастание или убывание и т.д.). Фактически при качественном сравнении оценивается совпадение лишь вида функции распределения параметров (убывающая или возрастающая, с одним экстремумом или с несколькими). Вопрос о количественном сравнении можно ставить лишь после удовлетворительного ответа на вопрос о качественном соответствии результатов. При количественном сравнении большое значение следует придавать точности исходных данных для моделирования и соответствующих им значений сравниваемых параметров.

Неадекватность результатов моделирования возможна, по крайней мере, по трем причинам:

а) Значения задаваемых параметров модели не соответствуют допустимой области этих параметров, определяемой принятой системой гипотез. Например, в задаче о баскетболисте гипотезу об отсутствии сопротивления воздуха можно использовать лишь при относительно малых (<5 м/с) скоростях движения тела. При больших значениях начальной скорости мяча влияние силы сопротивления будет существенным.

б) Принятая система гипотез верна, но константы и параметры в использованных определяющих соотношениях установлены не точно. Например, в случае задачи о

баскетболисте значение ускорения свободного падения g может быть уточнено в зависимости от широты местности, где находится баскетболист.

в) Неверна исходная совокупность гипотез.

Все три случая требуют дополнительного исследования как моделируемого объекта (с целью накопления новой дополнительной информации о его поведении), так и исследования самой модели (с целью уточнения границ ее применимости).

Замечание. В данном случае не анализируется влияние выбранного численного метода на точность получаемого решения, а значит, и на адекватность модели. Вопрос о сходимости алгоритма и устойчивости получаемого выбранным численным методом решения, а также накопление погрешностей, связанных с ошибками округления при использовании ЭВМ, здесь не рассматривается.

При возникновении проблем, связанных с адекватностью модели, ее корректировку следует начинать с последовательного анализа всех возможных причин, приведших к расхождению результатов моделирования с результатами эксперимента. В первую очередь следует исследовать модель и оценить степень ее адекватности при различных значениях варьируемых параметров (начальных и граничных условиях, параметров, характеризующих свойства объектов моделирования). Если модель неадекватна в интересующей исследователя области параметров, то следует попытаться уточнить значения констант и исходных параметров модели. Если же и в этом случае нет положительных результатов, то единственной возможностью улучшения модели остается изменение принятой системы гипотез. Данное решение фактически означает возвращение ко второму этапу процесса разработки модели и может повлечь не только серьезное изменение математической постановки задачи, но и изменение методов ее решения (например, переход от аналитических к численным), полной переработки программного обеспечения и нового цикла проверки модели на адекватность. Поэтому решение об изменении принятой системы гипотез должно быть всесторонне взвешено, и приниматься только в том случае, когда исчерпаны все прочие возможности по улучшению адекватности модели.

Следует предостеречь начинающих исследователей от попыток "перепрыгнуть" рассмотренный этап моделирования, от желания быстрее перейти к решению "настоящей задачи". Как показывает собственный опыт авторов, подобный образ действий приводит к огромным временным издержкам (не говоря уже о психологических). Особенно опасной является ситуация, когда при решении реальной задачи с использованием должным образом не проверенной модели получают правдоподобные результаты. Для других условий модель может дать качественно неверные результаты, но истоки ошибок разработчики будут искать уже не в модели.