

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

И. В. Осиновская, А. Г. Шляпугин, Я. А. Ерисов

**Численные методы решения
алгебраических уравнений и их систем**

Электронное учебное пособие

САМАРА

2012

УДК 629.73 (075.3)

Авторы: **Осиновская Ирина Васильевна,**
Шляпугин Алексей Геннадьевич,
Ерисов Ярослав Александрович.

Рецензенты:

Заместитель канд. техн. наук, доцент кафедры компьютерных технологий и обработки металлов давлением ТГУ Почекуев Е.Н.

Редакторская обработка Т. К. Крестина

Компьютерная верстка И. В. Осиновская

Доверстка И. В. Осиновская

Осиновская, И.В. Численные методы решения алгебраических уравнений и их систем [Электронный ресурс]: электрон. учеб. пособие / И. В. Осиновская, А. Г. Шляпугин, Я. А. Ерисов; Минобрнауки России, Самар. гос. аэрокосм. ун-т им. С. П. Королева (нац. исслед. ун-т). - Электрон. текстовые и граф. дан. (1,41 Мбайт). - Самара, 2012. - 1 эл. опт. диск (CD-ROM).

В учебном пособии рассмотрены этапы решения инженерных задач; источники и классификация погрешностей; статистический и технический подходы к учету погрешностей действий; приведены основные численные методы решения систем алгебраических уравнений, нелинейных уравнений и их систем.

Предназначены для бакалавров инженерно-технологического факультета, обучающихся по направлению 150700.62 «Машиностроение» по дисциплине «Прикладные компьютерные программы» в 4 семестре.

Подготовлены на кафедре обработки металлов давлением.

СОДЕРЖАНИЕ

1.	ЛЕКЦИЯ 1 МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ С ПРИМЕНЕНИЕМ ЭВМ	4
2.	ЛЕКЦИЯ 2 ВВЕДЕНИЕ В ЭЛЕМЕНТАРНУЮ ТЕОРИЮ ПОГРЕШНОСТИ	15
3.	ЛЕКЦИЯ 3 ПРЯМЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ	25
4.	ЛЕКЦИЯ 4 ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ (СЛАУ)	38
5.	ЛЕКЦИЯ 5 МЕТОДЫ ОТЫСКАНИЯ РЕШЕНИЙ НЕЛИНЕЙНЫХ УРАВНЕНИЙ	49
6.	ЛЕКЦИЯ 6 МЕТОДЫ ОТЫСКАНИЯ РЕШЕНИЙ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ	58

ЛЕКЦИЯ 1 МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И РЕШЕНИЕ ИНЖЕНЕРНЫХ ЗАДАЧ С ПРИМЕНЕНИЕМ ЭВМ [1]

В данной лекции дается общее представление о методе математического моделирования, в том числе о процессе создания математических моделей, о последовательности этапов решения инженерной задачи с применением ЭВМ, о вычислительном эксперименте.

Ключевые слова: МОДЕЛЬ, МОДЕЛИРОВАНИЕ, ЭТАПЫ ИНЖЕНЕРНЫХ ЗАДАЧ.

1.1 Математическое моделирование и процесс создания математической модели [1]

Модель — это такой материальный или мысленно представляемый объект, который в процессе исследования замещает объект-оригинал так, что его непосредственное изучение дает новые знания об объекте-оригинале.

Под *моделированием* понимается триединый процесс построения, изучения и применения моделей.

Процесс создания математической модели условно можно разбить на ряд основных этапов: 1) построение математической модели; 2) постановка, исследование и решение соответствующих вычислительных задач; 3) проверка качества модели на практике и модификация модели.

I. Построение математической модели. Выявляются основные «характеристики» явления, которым сопоставляются некоторые величины. Как правило, эти величины принимают числовые значения, т. е., являются переменными, векторами, матрицами, функциями и т.д. Математическая модель неизбежно представляет собой компромисс между бесконечной сложностью изучаемого явления и желаемой простотой, его описания. Модель должна быть достаточно полной, для того чтобы оказаться полезной для изучения свойств исследуемого явления. В то же время она обязана быть

достаточно простой, для того чтобы допускать возможность ее анализа существующими в математике средствами, и ее реализации на ЭВМ. Из огромного числа характеристик явления и действующих на него факторов требуется выделить основные, определяющие, отбросив при этом второстепенные, несущественные.

Нередко в математическую модель закладываются некоторые гипотезы, не подтвержденные на практике. Такую математическую модель часто называют *гипотетической*.

Математические модели часто разделяют на статические и динамические. Статическая модель описывает явление или ситуацию в предположении их завершенности, неизменности (т.е. в статике). Динамическая модель описывает, как протекает явление или изменяется ситуация от одного состояния к другому (т.е. в динамике). При использовании динамических моделей, как правило, задают начальное состояние системы, а затем исследуют изменение этого состояния во времени.

2. Подстановка, исследование и решение вычислительных задач. Для того чтобы найти интересующие исследователя значения величин или выяснять характер их зависимости от других входящих в математическую модель величин, ставят, а затем решают математические задачи.

Выявим основные типы решаемых задач. Для этого все величия, включенные в математическую модель, условно разобьем на три группы: 1) исходные (входные) данные x ; 2) параметры модели a ; 3) искомое решение (выходные данные) y . В динамических моделях искомое решение часто является функцией времени $y = y(t)$; переменная t в таких моделях, как правило, бывает выделенной и играет особую роль.

Наиболее часто решают так называемые *прямые задачи*, постановка которых выглядит следующим образом: по данному значению входного данного x при фиксированных значениях параметров a требуется найти решение y . Большую роль играет решение так называемых *обратных задач*, состоящих в определении входного данного x по данному значению y

(параметры модели a , как и в прямой задаче, фиксированы). Помимо двух рассмотренных типов задач следует упомянуть еще один тип — *задачи идентификации*. В широком смысле задача идентификации модели - это задача выбора среди множества всевозможных моделей той, которая наилучшим образом описывает изучаемое явление. Чаще задачу идентификации понимают в узком смысле, как задачу выбора из заданного параметрического семейства моделей конкретной математической модели (с помощью выбора ее параметров a), с тем, чтобы оптимальным в смысле некоторого критерия образом согласовать следствия из модели с результатами наблюдений.

Как правило, решение вычислительной задачи не удастся выразить через входные данные в виде конечной формулы, однако это совсем не означает, что решение такой задачи не может быть найдено. Существуют специальные методы, которые называют *численными* (или *вычислительными*). Они позволяют свести получение численного значения решения к последовательности арифметических операций над численными значениями входных данных.

3. Проверка качества модели на практике и модификация модели.

На этом этапе выясняют пригодность математической модели для описания исследуемого явления. Теоретические выводы и конкретные результаты, вытекающие из гипотетической математической модели, сопоставляют с экспериментальными, данными. Если они противоречат друг другу, то выбранная модель непригодна и ее следует пересмотреть, вернувшись к первому этапу. Если же результаты совпадают с допустимой для описания данного явления точностью, то модель можно признать пригодной.

1.2 Основные этапы решения инженерной задачи с применением ЭВМ [1]

Решение серьезной инженерной задачи с использованием ЭВМ — довольно длительный и сложный процесс. С определенной *степенью*

условности его можно разбить на ряд последовательных этапов. Выделим следующие этапы: 1) постановка проблемы; 2) выбор или построение математической модели; 3) постановка вычислительной задачи; 4) предварительный (предмашинный) анализ свойств вычислительной задачи; 5) выбор или построение численного метода; 6) алгоритмизация и программирование; 7) отладка программы; 8) счет по программе; 9) обработка и интерпретация результатов; 10) использование результатов и коррекция математической модели.

1. Постановка проблемы. Первоначально прикладная задача бывает сформулирована в самом общем виде: исследовать некоторое явление, спроектировать устройство, обладающее заданными свойствами и т.д. На данной стадии происходит конкретизация постановки задачи, и первостепенное внимание при этом уделяется выяснению цели исследования. От исследователя требуется глубокое понимание существа, задачи и умение сформулировать, ее так, чтобы найденное решение было полезным и в то же время могло быть получено с помощью существующих методов и в реальные сроки.

2. Выбор или построение математической модели. Для последующего анализа исследуемого явления или объекта необходимо дать его формализованное описание на языке математики, т.е. построить математическую модель. Часто имеется возможность выбора модели среди известных и принятых для описания соответствующих процессов, но нередко требуется и существенная модификация известной модели, а иногда возникает необходимость в построении принципиально новой модели.

Важно, чтобы сложность математической модели соответствовала сложности поставленной проблемы. Если поставленных целей можно достичь, используя более простую математическую модель, то ей и следует отдать предпочтение.

Построение математической модели является наиболее сложным и ответственным этапом решения. Если выбранная математическая модель слишком грубо отражает изучаемое явление, то какие бы методы решения вслед за этим ни применялись, найденные значения не будут отвечать условиям реальной задачи и окажутся бесполезными.

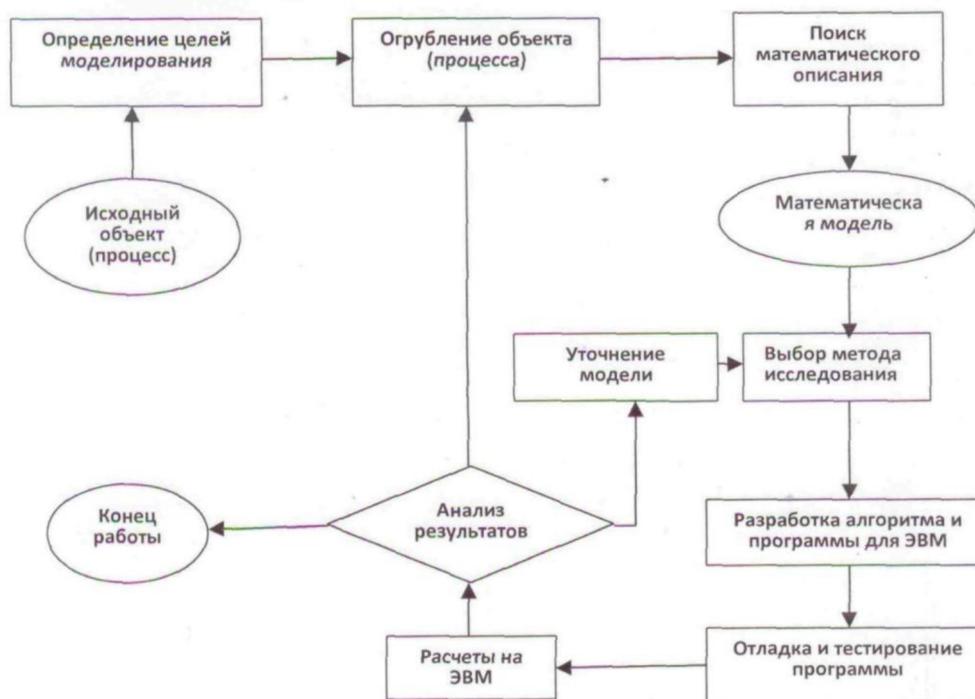


Рис. 1. Общая схема процесса компьютерного математического моделирования [2]

3. Постановка вычислительной задачи. На основе принятой математической модели формулируют вычислительную задачу (или ряд таких задач). Анализируя результаты ее решения, исследователь предполагает получить ответы на интересующие его вопросы.

4. Предварительный анализ свойств вычислительной задачи. На этом этапе проводят предварительное (предмашинное) исследование свойств вычислительной задачи. Большое внимание уделяют анализу корректности ее постановки, т.е. выяснению вопросов существования и

единственности решения, а также исследованию устойчивости решения задачи к погрешностям входных данных.

В прикладных исследованиях существенное значение имеют конкретные (часто — весьма сжатые) сроки получения результата. На этом этапе полезным оказывается изучение упрощенных постановок задачи. Особую ценность имеют различные аналитические решения; они оказываются полезными не только для анализа явления, но и как основа для тестовых испытаний на этапе отладки программы.

5. Выбор или построение численного метода. Для решения вычислительной задачи на ЭВМ требуется использование численных методов.

Часто решение инженерной задачи сводится к последовательному решению стандартных вычислительных задач, для которых разработаны эффективные численные методы. В этой ситуации происходит либо выбор среди известных методов, либо их адаптация к особенностям решаемой задачи. Однако если возникающая вычислительная задача является новой, то не исключено, что для ее решения не существует готовых методов. Построение численного метода для такой задачи может оказаться очень трудной проблемой и потребовать привлечения специалиста по вычислительной математике.

Для решения одной и той же вычислительной задачи обычно может быть использовано несколько методов. Необходимо знать особенности этих методов, критерии, по которым оценивается их качество, чтобы выбрать метод, позволяющий решить проблему наиболее эффективным образом.

6. Алгоритмизация и программирование. Как правило, выбранный на предыдущем этапе численный метод содержит только принципиальную схему решения задачи, не включающую многие детали, без которых невозможна реализация метода на ЭВМ. Необходима подробная детализация всех этапов вычислений, для того чтобы получить реализуемый на ЭВМ алгоритм. Составление программы сводится к переводу этого алгоритма на

выбранный язык программирования. Заметим, что в настоящее время для вычислительных задач наиболее широко используется алгоритмический язык ФОРТРАН.

7. Отладка программы. На этом этапе с помощью ЭВМ выявляют и исправляют ошибки в программе.

Наличие в программах ошибок — вполне нормальное и закономерное явление. Поэтому подготовку к отладке следует начинать уже на этапе алгоритмизации и программирования. Заметим, что эффективность отладки самым существенным образом зависит от общей методики разработки программ.

После устранения ошибок программирования необходимо провести тщательное тестирование программы — проверку правильности ее работы на специально отобранных тестовых задачах, имеющих известные решения.

8. Счет по программе. На этом этапе происходит решение задачи на ЭВМ по составленной программе в автоматическом режиме. Этот процесс, в ходе которого входные данные с помощью ЭВМ преобразуются в результат, называют *вычислительным процессом*. Как правило, счет повторяется многократно с различными входными данными для получения достаточно полной картины зависимости от них решения задачи.

Первые полученные результаты тщательно анализируются, для того чтобы убедиться в правильности работы программы и пригодности выбранного метода решения. Счет по программе продолжается несколько секунд, минут или часов.

9. Обработка и интерпретация результатов. Полученные в результате расчетов на ЭВМ выходные данные, как правило, представляют собой большие массивы чисел.

Зачастую первоочередной интерес представляет лишь небольшая часть полученной информации (например, значения одной из функций в

выделенных точках) или даже некоторая грубая интегральная характеристика (максимальное или минимальное значение, оценка энергии системы и т.д.).

Для того чтобы исследователь мог воспользоваться результатами расчетов, их необходимо представить в виде компактных таблиц, графиков или в иной удобной для восприятия форме. При этом следует максимально использовать возможности ЭВМ для подготовки такой информации и ее представления с помощью печатающих и графических выходных устройств.

Для правильной интерпретации результатов расчетов и оценки их достоверности от исследователя требуется глубокое знание существа решаемой инженерной задачи, ясное представление об используемой математической модели и понимание (хотя бы в общих чертах) особенностей применяемого вычислительного метода.

10. Использование результатов и коррекция математической модели. Завершающий этап состоит в использовании результатов расчетов в практической деятельности, иначе говоря, во внедрении результатов. Не стоит огорчаться, если большинство полученных сначала результатов окажется бесполезным. Действительно полезные для практики результаты являются плодом серьезной целенаправленной работы, в процессе которой цикл решения задачи повторяется неоднократно.

Очень часто анализ результатов, проведенный на этапе их обработки и интерпретации, указывает на несовершенство используемой математической модели и необходимость ее коррекции. В таком случае математическую модель модифицируют (при этом она, как правило, усложняется) и начинают новый цикл решения задачи.

1.3 Вычислительный эксперимент

Создание математических моделей и решение инженерных задач с применением ЭВМ требует выполнения большого объема работ. Нетрудно заметить аналогию с соответствующими работами, проводимыми при организации натуральных экспериментов: составление программы

экспериментов, создание экспериментальной установки, выполнение контрольных экспериментов, проведение серийных опытов, обработка экспериментальных данных и их интерпретация и т.д. Однако вычислительный эксперимент проводится не над реальным объектом, а над его математической моделью, и роль экспериментальной установки играет оснащенная специально разработанной программой ЭВМ. В связи с этим естественно рассматривать проведение больших комплексных расчетов при решении инженерных и научно-технических задач как вычислительный *эксперимент*, а описанную в предыдущем параграфе последовательность этапов решения как один его цикл.

Широкое применение ЭВМ в математическом моделировании, разработанная теория и значительные практические результаты позволяют говорить о вычислительном эксперименте как о новой технологии и методологии научных и прикладных исследований. Серьезное внедрение вычислительного эксперимента в инженерную деятельность лишь начинается, но там где оно происходит реально (в авиационной и космической промышленности) его плоды весьма весомы.

Отметим некоторые достоинства вычислительного эксперимента по сравнению с натурным. Вычислительный эксперимент, как правило, дешевле физического. В этот эксперимент можно легко и безопасно вмешиваться. Его можно повторить еще раз (если в этом есть необходимость) и прервать в любой момент. В ходе этого эксперимента можно смоделировать условия, которые нельзя создать в лаборатории.

Заметим, что в ряде случаев проведение натурального эксперимента затруднено (а иногда и невозможно), так как изучаются быстропротекающие процессы, исследуются труднодоступные или вообще пока недоступные объекты. Часто проведение полномасштабного натурального эксперимента сопряжено с губительными или непредсказуемыми последствиями (ядерная война, поворот сибирских рек) или с опасностью для жизни или здоровья людей. Нередко требуется исследование и

прогнозирование результатов катастрофических явлений (авария ядерного реактора АЭС, глобальное потепление климата, землетрясение). В этих случаях вычислительный эксперимент может стать основным средством исследования. Заметим, что с его помощью оказывается возможным прогнозировать свойства новых, еще не созданных конструкций и материалов на стадии их проектирования.

Существенным недостатком вычислительного эксперимента является то, что применимость его результатов ограничена рамками принятой математической модели.

Для инженерных задач характерно наличие значительного числа параметров (конструктивных, технологических и др.). Создание нового изделия или технологического процесса предполагает выбор среди большого числа альтернативных вариантов, а также оптимизацию по ряду параметров. Поэтому в ходе вычислительного эксперимента расчеты проводятся многократно с разными значениями входных параметров. Для получения нужных результатов с требуемой точностью и в приемлемые сроки необходимо, чтобы на расчет каждого варианта тратилось минимальное время. Именно поэтому при создании программного обеспечения так важно использовать эффективные численные методы.

Контрольные вопросы

1. Что такое модель?
2. Что понимается под процессом моделирования?
3. Из каких основных этапов состоит процесс решения инженерных задач?

Рекомендуемая литература

1. Амосов, А. А. Вычислительные методы для инженеров [Текст]: учеб. пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. - М.: Высшая школа, 1994. – 544с.

2. Лапчик, М. П. Численные методы [Текст]: учеб. пособие для студ. вузов / М. П. Лапчик, М. И. Рагулина, Е. К. Хеннер. – М.: Академия, 2004. – 384 с.

ЛЕКЦИЯ 2 ВВЕДЕНИЕ В ЭЛЕМЕНТАРНУЮ ТЕОРИЮ ПОГРЕШНОСТИ [1]

В данной лекции дается описание источников возникновения погрешностей и приводится их классификация; дается определение понятий абсолютная и относительная погрешность и приводится форма записи данных; рассматриваются статистический и технический подходы к учету погрешностей действий; дается понятие погрешности машинных вычислений и рассматривается вопрос корректности и устойчивости.

Ключевые слова: АБСОЛЮТНАЯ И ОТНОСИТЕЛЬНАЯ ПОГРЕШНОСТЬ, ПОГРЕШНОСТЬ МЕТОДА, ВЫЧИСЛИТЕЛЬНАЯ ПОГРЕШНОСТЬ, КОРРЕКТНОСТЬ И УСТОЙЧИВОСТЬ

2.1 Источники и классификация погрешностей[1]

Возникновение, накопление и распространение ошибок проходит через все стадии решения прикладной задачи, начиная с получения значений исходных данных.

Источниками возникновения погрешности численного решения задачи являются следующие факторы.

- Неточность математического описания, в частности, неточность задания начальных данных.
- Неточность численного метода решения задачи.

Данная причина возникает, например, когда решение математической задачи требует неограниченного или неприемлемо большого числа арифметических операций, что приводит к необходимости ограничения их числа, т. е. использования приближенного решения.

- Конечная точность машинной арифметики.

Виды погрешностей

Все погрешности можно разделить на три вида:

- неустранимая погрешность;
- погрешность метода;
- вычислительная погрешность.

Результирующая погрешность определяется как сумма величин всех перечисленных выше погрешностей.

1. Неустраняемые или исходные погрешности. Исходные, или неустраняемые, погрешности появляются в результате несоответствия математической модели изучаемому физическому явлению. Так, например, могут быть не учтены какие-либо важные черты исследуемого процесса либо нарушены границы применимости данной модели.

К тому же типу погрешностей можно отнести и погрешности исходных данных, которые, в большинстве случаев являясь экспериментальными, уже содержат в себе исходную погрешность. Это могут быть начальные и граничные условия задачи, коэффициенты и правые части (силы) моделируемых уравнений и т. д.

Если обозначить через $y(t)$ координату падающего тела в физической модели, то переход к математической модели, описываемой системой уравнений (ВЗ), может привести к неустраняемым погрешностям вследствие неточного задания начальных значений координаты и скорости тела, а также действующей на него силы. В силу этого значения, координаты в математической и физической моделях будут отличаться. Обозначая через $y(t)$ координату в математической модели, придем к следующему определению неустраняемой погрешности:

$$\varepsilon_1 = y - y \tag{1}$$

Во многих случаях под погрешностью понимают не рассмотренную выше разность между приближениями, а некоторые меры близости между ними. Например, в скалярном случае полагают

$$\rho_1 = |y - y| \tag{2}$$

2. Погрешность численного метода, или остаточная погрешность.

Погрешность численного метода, или остаточная погрешность, есть следствие того, что численным методом решается уже другая, упрощенная задача, которая является приближением исходной. Погрешность численного метода

$$\varepsilon_2 = y_n - y \quad \text{или} \quad \rho_2 = |y_n - y|, \quad (3)$$

как правило, регулируема, так как можно изменить шаг задачи, число итераций, число точек интерполяции, число учитываемых членов ряда и т. д. При моделировании стараются уменьшить величину этой погрешности так, чтобы она оказалась в несколько раз меньше исходной погрешности, так как дальнейшее уменьшение приводит лишь к удорожанию расчета за счет роста затрачиваемого машинного времени, не давая выигрыша в точности результата.

3. Погрешности округления, или вычислительные погрешности.

Погрешности округления, или вычислительные погрешности, связаны с ограниченностью разрядной сетки компьютера. В силу этого при вводе данных в машину, при выполнении арифметических операций и выводе данных производится округление, которое и приводит к дополнительной погрешности. В нашем примере в машине имеется значение координаты тела y_n^* , в которое входит и ошибка округления

$$\varepsilon_3 = y_n^* - y_n, \quad \rho_3 = |y_n^* - y_n|. \quad (4)$$

Полная погрешность, т. е. разность между реально получаемым и точным решением задачи ε_0 удовлетворяет следующему равенству:

$$\varepsilon_0 = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 = y_n^* - y, \quad (5)$$

$$\rho_0 = |y_n^* - y|,$$

$$\rho_0 \leq \rho_1 + \rho_2 + \rho_3.$$

2.2 Абсолютная и относительная погрешности. Формы записи данных [2]

Абсолютной погрешностью приближения i к точному значению величины u называют величину

$$\Delta_u = |u - u|. \quad (6)$$

Относительную погрешность определяют как

$$\delta_u = \frac{\Delta_u}{|u|} \approx \frac{|u - u|}{|u|}. \quad (7)$$

Относительную погрешность часто выражают в процентах.

Абсолютную и относительную погрешность обычно записывают в виде числа, содержащего одну или две значащих цифры.

Информацию о том, что \tilde{u} является приближенным значением числа u с абсолютной погрешностью Δ_u , принято записывать в виде:

$$u = u \pm \Delta_u \quad (8)$$

Соответственно, информацию о том, что \tilde{u} является приближенным значением числа u с относительной погрешностью δ_u , записывают в виде:

$$u = u(1 \pm \delta_u) \quad (9)$$

При сложении или вычитании чисел их абсолютные погрешности складываются, а относительная погрешность суммы заключена между наибольшим и наименьшим значениями относительных погрешностей слагаемых. При умножении или делении чисел друг на друга их относительные погрешности складываются. При возведении в степень приближенного числа его относительная погрешность умножается на показатель степени.

В случае функции многих независимых переменных

$$u = u(x_1, x_2, \dots, x_n) \quad (10)$$

абсолютная и относительная погрешности определяются выражениями:

$$\Delta_u = \sum_{i=1}^n \frac{\partial u}{\partial x_i} \Delta x_i \quad (11)$$

$$\delta_u = \sum_{i=1}^n \left| \frac{1}{u} \frac{\partial u}{\partial x_i} \right| \Delta x_i = \sum_{i=1}^n \left| \frac{\partial}{\partial x_i} \ln u \right| \Delta x_i [0,1]. \quad (12)$$

2.3 Статистический и технический подходы к учету погрешностей действий [3]

Рассмотренный выше *аналитический* (или *классический*) *способ учета погрешностей действий*, предполагающий точное оценивание погрешностей, основанное либо на правилах подсчета погрешностей арифметических действий, либо на параллельной работе с верхними и нижними границами исходных данных, имеет два существенных недостатка. Во-первых, этот способ чрезвычайно громоздок и не может быть рекомендован при массовых вычислениях. Во-вторых, он учитывает крайние, наихудшие случаи взаимодействия погрешностей, которые допустимы, но маловероятны. Ясно, что, например, при суммировании нескольких приближенных чисел (полученных в результате измерений, округлений или каким-либо другим путем) среди них почти, наверное, будут слагаемые как с избытком, так и с недостатком, т.е. произойдет частичная компенсация погрешностей. При больших количествах однотипных вычислений вступают в силу уже *вероятностные* или *статистические законы* формирования погрешностей результатов действий. Например, методами теории вероятностей показывается, что математическое ожидание абсолютной погрешности суммы n слагаемых с одинаковым уровнем абсолютных погрешностей, при достаточно большом n , пропорционально \sqrt{n} . В частности, если $n > 10$ и все слагаемые округлены до m -го десятичного разряда, то для подсчета абсолютной погрешности суммы S применяют *правило Чеботарева*

$$\Delta S \approx \sqrt{3n} \cdot 0,5 \cdot 10^{-m} \quad (13)$$

Различие в результатах классического и статистического подходов к оцениванию погрешности суммы рассмотрим на примере оценки погрешности среднего арифметического нескольких приближенных чисел.

Пусть $x = \frac{1}{n}(x_1 + \dots + x_n)$ - среднее арифметическое n ($n > 10$) приближенных чисел (например, результатов измерений), имеющих одинаковый уровень абсолютных погрешностей $\Delta_{x_i} = 0,5 \cdot 10^m$. Тогда классическая оценка абсолютной погрешности величины x есть

$$\Delta_x = \frac{1}{n}(\Delta_{x_1} + \dots + \Delta_{x_n}) = \frac{1}{n} \cdot n \cdot 0,5 \cdot 10^m = 0,5 \cdot 10^m = \Delta_{x_i} \quad (14)$$

т.е. такая же, как и у исходных данных. В то же время по формуле (13) имеем

$$\Delta_x \approx \frac{1}{n} \sqrt{3n} \cdot 0,5 \cdot 10^m = \sqrt{\frac{3}{n}} \cdot 0,5 \cdot 10^m = \sqrt{\frac{3}{n}} \cdot \Delta_{x_i} \rightarrow_{n \rightarrow \infty} 0$$

Как видим, применение правила Чеботарева приводит к естественному выводу о том, что арифметическое усреднение результатов измерений или наблюдений увеличивает точность, чего нельзя сказать на основе классической теории погрешностей.

Прямое применение вероятностно-статистических оценок погрешностей также является достаточно сложным делом и вряд ли может быть рекомендовано при рядовых массовых вычислениях. Однако именно такие оценки подкрепляют практические правила работы с приближенными числами, составляющие основу так называемого технического подхода. Этот подход связывают с именем известного русского кораблестроителя, математика и механика академика А. Н. Крылова. Согласно принципу А. Н. Крылова, приближенное число должно записываться так, чтобы в нем все значащие цифры, кроме последней, были верными и лишь последняя была бы сомнительна, и притом в среднем не более чем на одну единицу. Напомним, что значащими цифрами числа в его позиционной записи называются все его цифры, начиная с первой ненулевой слева. Значащую цифру приближенного числа называют верной, если абсолютная

погрешность числа не превосходит единицы разряда, в котором стоит эта цифра (или половины единицы; в этом случае иногда применяется термин верная в узком смысле).

Чтобы результаты арифметических действий, совершаемых над приближенными числами, записанными в соответствии с принципом А. Н. Крылова, также соответствовали этому принципу нужно придерживаться следующих простых правил:

1) при сложении и вычитании приближенных чисел в результате следует сохранять столько десятичных знаков, сколько их в приближенном данном с наименьшим количеством десятичных знаков;

2) при умножении и делении в результате следует сохранять столько значащих цифр, сколько их имеет приближенное данное с наименьшим числом значащих цифр;

3) результаты промежуточных вычислений должны иметь один-два запасных знака (которые затем должны быть отброшены).

Таким образом, при техническом подходе к учету погрешностей приближенных вычислений предполагается, что в самой записи приближенного числа содержится информация о его точности. И хотя прямая выгода от применения приведенных правил работы с приближенными числами может быть получена лишь при ручном счете (не нужно оперировать с цифрами, не влияющими на информативную часть приближенного результата), их знание и понимание помогает правильной интерпретации компьютерных расчетов, а иногда и самой организации таковых.

2.4 Понятие погрешности машинных вычислений [1]

Для представления чисел в памяти компьютера применяют два способа:

- с фиксированной запятой;
- с плавающей запятой.

Пусть в основу запоминающего устройства положены однотипные физические устройства, имеющие r устойчивых состояний, при этом каждое из устройств состоит из одинакового количества элементов k , в одном из которых фиксируется знак числа. Упорядоченные элементы образуют разрядную сетку машинного слова: в каждом разряде может быть записано одно из базисных чисел $0, 1, \dots, r-1$ (одна из r "цифр" r -ой системы счисления) и в специальном разряде отображен знак "+" или "-". При записи чисел с фиксированной запятой кроме упомянутых r параметров (основания системы счисления) и k (количество разрядов, отводимых под запись числа) указывается еще общее количество l разрядов, выделяемых под дробную часть числа. Таким образом, положительное вещественное число u , представляющее собой в r -ой системе бесконечную непериодическую дробь, отображающуюся конечной последовательностью

$$u_1 u_2 \dots u_{k-l} u_{k-l+1} \dots u_{k-l} u_k,$$

где $u_i \in \{0, 1, \dots, r-1\}$, т. е.

$$u \approx u_1 r^{k-l-1} + u_2 r^{k-l-2} + \dots + u_{k-l} r^0 + u_{k-l+1} r^{-1} + \dots + u_{k-l} r^{-(l-1)} + u_k r^{-l}$$

Диапазон представляемых таким способом чисел определяется числами с наибольшими цифрами во всех разрядах, т. е. наименьшим числом $-(r-1)(r-1)\dots(r-1)$ и наибольшим $(r-1)(r-1)\dots(r-1)$, а абсолютная точность представления есть оценка величины $|u - u|$, зависящая от способа округления. Абсолютная точность представления вещественных чисел с фиксированной запятой одинакова в любой части диапазона. В то же время относительная точность может значительно различаться в зависимости от того, берется u близким к нулю или к границе диапазона.

2.5 Корректность и устойчивость [2]

Задача $y = Ax$ называется корректно поставленной, если для любых входных данных $x \in X$ (из некоторого класса X) решение $y \in Y$ существует, единственно и устойчиво по входным данным. Первые два требования

естественны, так как, приступая к численному решению задачи нужно быть уверенным, что решение существует, а однозначная последовательность действий в численном алгоритме диктует требование единственности. Кроме того, если исходные данные имеют небольшую погрешность, то и погрешность решения должна быть невелика. Иными словами, задача называется устойчивой по входным данным x , если решение y непрерывно зависит от входных данных.

Отсутствие устойчивости означает, что даже незначительные погрешности в исходных данных могут привести к большим погрешностям решения или вовсе неверному результату. О таких задачах говорят, что они чувствительны к погрешностям исходных данных.

Контрольные вопросы

1. Что такое абсолютная погрешность приближенного значения величины? граница абсолютной погрешности?
2. Что такое относительная погрешность приближенного значения величины?
3. Как можно вычислить абсолютную погрешность приближения u , если известна его относительная погрешность?
4. Из каких частей складывается общая погрешность решения задачи?
5. Что такое статический подход к учету погрешностей действий?
6. Что такое технический подход к учету погрешностей действий?

Рекомендуемая литература

1. Поршневу, С. В. Численные методы на базе Mathcad [Текст]: учеб. пособие для студ.вузов / С. В. Поршневу, И. В. Беленкова. - СПб: БХВ - Петербург, 2005. – 464 с.
2. Ращиков, В. И. Численные методы решения физических задач [Текст]: учеб. пособие / В. И. Ращиков, А. С. Рошаль. - СПб.: Изд-во «Лань», 2005. – 208 с.

3. Вержбицкий, В. М. Основы численных методов [Текст]: учебник для вузов / В. М. Вержбицкий. – М.: Высш. шк., 2002. – 840 с.

ЛЕКЦИЯ 3 ПРЯМЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ [1]

В данной лекции рассматриваются простые, часто применяемые численные методы решения систем линейных алгебраических уравнений с квадратными матрицами коэффициентов, приводятся типы используемых матриц.

Ключевые слова: МАТРИЦА, МЕТОД ГАУССА, МЕТОД ИСКЛЮЧЕНИЯ ГАУССА – ЖОРДАНА, МЕТОД ХОЛЕССКОГО.

В научных исследованиях и инженерном проектировании часто приходится решать алгебраические уравнения. Задачи этого типа, решаемые на компьютере, могут возникать сами по себе или же составлять часть более сложных исследований. Выбор подходящего алгоритма для решения алгебраической системы зависит от класса рассматриваемой задачи. Алгебраические задачи могут быть классифицированы, во-первых, по числу решаемых уравнений, а затем по типу и количеству ожидаемых ответов. На рис. 1 дана схема классификации алгебраических задач. В случае одного уравнения задачу называют *линейной*, *алгебраической* или *трансцендентной*, причем первая имеет одно решение, вторая - n решений, а в третьей число решений заранее не известно. В случае нескольких уравнений задачу называют *линейной* или *нелинейной* в зависимости от математической природы входящих уравнений.

Прямыми в вычислительной математике называются численные методы, в которых [2]:

- требующееся для решения задачи число арифметических операций можно оценить по расчетным формулам заранее, до начала решения;
- решение является точным.



Рис. 1. Схема классификации алгебраических задач [1]

При этом надо иметь в виду, что в вычислительной математике «точное решение» понимается как «решение точностью до погрешностей округления», т. е. как решение, которое можно было бы получить по точным формулам на идеализированном компьютере с бесконечной разрядностью машинного слова. В действительности, погрешности округления, конечно, имеются, так что прямые методы также дают приближенное решение.

3.1 Типы используемых матриц [3]

Эффективность вычислений в линейной алгебре существенно зависит от умения использовать специальную структуру и свойства используемых в расчетах матриц. Напомним некоторые важные типы матриц.

Квадратная матрица A называется *диагональной*, если ее элементы удовлетворяют условию $a_{ij} = 0$ для $i \neq j$ (все отличные от нуля элементы расположены на главной диагонали):

$$A = \begin{bmatrix} a_{11} & 0 & 0 \dots & 0 \\ 0 & a_{22} & 0 \dots & 0 \\ 0 & 0 & a_{33} \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & a_{mm} \end{bmatrix}$$

Диагональную матрицу, у которой все элементы a_{ii} главной диагонали равны единице, называют единичной и обозначают буквой E :

$$E = \begin{bmatrix} 1 & 0 & 0 \dots & 0 \\ 0 & 1 & 0 \dots & 0 \\ 0 & 0 & 1 \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & 1 \end{bmatrix}$$

Важную роль в численном анализе играют треугольные матрицы. Квадратная матрица A называется нижней треугольной, если все ее элементы, расположенные выше главной диагонали, равны нулю ($a_{ij} = 0$ для $i < j$). Если же равны нулю все элементы матрицы, расположенные ниже главной диагонали ($a_{ij} = 0$ для $i > j$), то она называется верхней треугольной.

Нижняя и верхняя треугольная матрицы имеют соответственно следующий вид:

$$\begin{bmatrix} a_{11} & 0 & 0 \dots & 0 \\ a_{21} & a_{22} & 0 \dots & 0 \\ a_{31} & a_{32} & a_{33} \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} \dots & a_{mm} \end{bmatrix}, \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} \dots & a_{1m} \\ 0 & a_{22} & a_{23} \dots & a_{2m} \\ 0 & 0 & a_{33} \dots & a_{3m} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 \dots & a_{mm} \end{bmatrix}.$$

Треугольные матрицы обладают рядом замечательных свойств. Например, для таких матриц определитель легко вычисляется по формуле

$$\det A = a_{11}a_{22}a_{33} \dots a_{mm}$$

Квадратная матрица A называется симметричной, если она совпадает со своей транспонированной матрицей A^T (или, что то же, $a_{ij} = a_{ji}$ для всех i, j).

Будем называть симметричную матрицу A положительно определенной и писать $A > 0$, если для всех векторов $x \neq 0$ квадратичная форма

$$(Ax, x) = \sum_{i,j=1}^m a_{ij} x_i x_j$$

принимает положительные значения.

Обозначим через λ_{\min} и λ_{\max} минимальное и максимальное собственные значения матрицы A . Известно, что для симметричной матрицы

$$\lambda_{\min} \|x\|^2 \leq (Ax, x) \leq \lambda_{\max} \|x\|^2$$

и матрица A положительно определена тогда и только тогда, когда все ее собственные значения положительны.

Одна из трудностей практического решения систем большой размерности связана с ограниченностью оперативной памяти ЭВМ. Хотя объем оперативной памяти вновь создаваемых вычислительных машин растет очень быстро, тем не менее, еще быстрее возрастают потребности практики в решении задач все большей размерности.

В общем случае система линейных уравнений имеет вид

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = c_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = c_2$$

.....

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = c_n$$

В матричной форме эта система принимает вид $Ax = b$, где

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}.$$

Чтобы система имела единственное решение, входящие в нее n уравнений должны быть линейно независимы. Необходимым и достаточным условием этого является условие неравенства нулю определителя данной системы. Алгоритмы решения задач такого типа подразделяются на прямые и итерационные. Возможность систематизировать прямые методы для такой задачи делает их весьма популярными

3.2 Метод Гаусса [2]

Наиболее распространенные прямые методы основаны на приведении системы уравнений к «треугольному» виду. При этом одно из уравнений содержит только одну неизвестную, а в каждом следующем добавляется еще по одной неизвестной. При счете вручную приведение к треугольному виду достигается сложением и вычитанием уравнений после умножения их на соответствующие постоянные множители. Выполняя эту процедуру вручную, нетрудно ошибиться, однако с ее помощью можно построить удобный алгоритм числового решения на ЭВМ. Одним из используемых для этого методов является метод Гаусса. Применяя его, сначала нормируют первое уравнение, для этого его коэффициенты делят на a_{11} . Затем первое уравнение умножают на первые коэффициенты a_{i1} всех других уравнений и последовательно вычитают из остальных уравнений. В результате первая переменная будет исключена из всех уравнений, кроме первого. На следующем этапе решения такая же процедура применяется к остальным $n-1$ уравнениям. В результате из оставшихся $n-2$ уравнений исключается вторая неизвестная. Всю процедуру повторяют до тех пор, пока после n

шагов вся система не будет приведена к треугольному виду. Математически эту процедуру можно описать следующим образом.

На k -м шаге процесса исключения новые нормированные коэффициенты k -го уравнения имеют вид

$$b_{k,j} = a_{k,i} / a_{k,k},$$

а новые коэффициенты в следующих уравнениях принимают вид

$$b_{i,j} = a_{i,j} - a_{i,j} b_{k,j}, \quad i > k.$$

Выполняя эту процедуру, следует помнить, что коэффициенты нижестоящих уравнений $a_{i,j}$ меняются на каждом шаге. Например, коэффициенты $b_{i,j}$ на следующем шаге превращаются в коэффициенты $a_{i,j}$. Проиллюстрируем применение описанной процедуры на следующем простом примере.

Пусть методом исключения Гаусса требуется решить систему уравнений

$$\begin{aligned} x_1 + x_2 + x_3 - x_4 &= 2 \\ x_1 - x_2 - x_3 + x_4 &= 0 \\ 2x_1 + x_2 - x_3 + 2x_4 &= 9 \\ 3x_1 + x_2 + 2x_3 - x_4 &= 7 \end{aligned}$$

Для удобства обозначим уравнения буквами и будем выписывать только коэффициенты при неизвестных и свободные члены уравнений. Тогда

исходная система	примет	вид				
A_1		1	1	1	-1	2
A_2		1	-1	-1	1	0
A_3		2	1	-1	2	9
A_4		3	1	2	-1	7

Исключая члены, содержащие x_1 , получим

$B_1 = A_1 / 1$	1	1	1	-1	2
$B_2 = A_2 - B_1$	0	-2	-2	2	-2
$B_3 = A_3 - 2B_1$	0	-1	-3	4	5

$$B_4 = A_4 - 3B_1 \quad 0 \quad -2 \quad -1 \quad 2 \quad 1$$

После исключения членов с x_2 имеем

$$B_1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 2$$

$$C_2 = B_2 / (-2) \quad 0 \quad 1 \quad 1 \quad -1 \quad 1$$

$$C_3 = B_3 + C_2 \quad 0 \quad 0 \quad -2 \quad 3 \quad 6$$

$$C_4 = B_4 + 2C_2 \quad 0 \quad 0 \quad 1 \quad 0 \quad 3$$

Исключение членов с x_3 дает

$$B_1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 2$$

$$C_2 \quad 0 \quad 1 \quad 1 \quad -1 \quad 1$$

$$D_3 = C_3 / (-2) \quad 0 \quad 0 \quad 1 \quad -3/2 \quad -3$$

$$D_4 = C_4 - D_3 \quad 0 \quad 0 \quad 0 \quad 3/2 \quad 6$$

Дойдя до последнего ряда, получим

$$B_1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 2$$

$$C_2 \quad 0 \quad 1 \quad 1 \quad -1 \quad 1$$

$$D_3 \quad 0 \quad 0 \quad 1 \quad -3/2 \quad -3$$

$$E_4 = 2D_4 / 3 \quad 0 \quad 0 \quad 0 \quad 1 \quad 4$$

Возвращаясь к форме уравнений, получим

$$x_1 + x_2 + x_3 - x_4 = 2,$$

$$x_2 + x_3 - x_4 = 1,$$

$$x_3 - (3/2)x_4 = -3,$$

$$x_4 = 4$$

откуда, подставляя значение x_4 , в 3-е уравнение, x_3 - во 2-е и т.д., находим решение системы уравнений

$$x_1 = 1, \quad x_2 = 2, \quad x_3 = 3, \quad x_4 = 4.$$

На этом примере хорошо видно, что желательно сделать нулевыми все элементы матрицы коэффициентов, стоящие вне главной диагонали. Последнюю процедуру называют приведением к диагональному виду, она

представляет собой усовершенствованную разновидность метода приведения к треугольному виду.

3.3 Метод исключения Гаусса – Жордана [2]

Этот метод дает алгоритм приведения системы линейных уравнений к диагональному виду. Единственным его формальным отличием от предыдущего будет то, что вместе $i > k$ подставляются $i \neq k$. Назовем k -й ряд ведущим. В методе Гаусса преобразования затрагивали только уравнения, стоящие ниже ведущего ряда. В методе же Гаусса – Жордана преобразуются уравнения, стоящие и под ведущим рядом, и над ним.

Чтобы показать, как это делается, решим предыдущую задачу методом Гаусса - Жордана. Исходный массив имеет вид

A_1	1	1	1	-1	2
A_2	1	-1	-1	1	0
A_3	2	1	-1	2	9
A_4	3	1	2	-1	7

Исключая члены, содержащие x_1 , получим

$B_1 = A_1 / 1$	1	1	1	-1	2
$B_2 = A_2 - B_1$	0	-2	-2	2	-2
$B_3 = A_3 - 2B_1$	0	-1	-3	4	5
$B_4 = A_4 - 3B_1$	0	-2	-1	2	1

До сих пор процедура была такой же, как в случае метода Гаусса.

После исключения x_2 получаем

$C_1 = B_1 - C_2$	1	0	0	0	1
$C_2 = B_2 / (-2)$	0	1	1	-1	1
$C_3 = B_3 + C_2$	0	0	-2	3	6
$C_4 = B_4 + 2C_2$	0	0	1	0	3

Теперь хорошо видны отличия от метода Гаусса. Исключив

члены с x_3 , найдем

$$\begin{array}{lcl}
 D_1 = C_1 - (0)D_3 & 1 & 0 \quad 0 \quad 0 \quad 1 \\
 D_2 = C_2 - D_3 & 0 & 1 \quad 0 \quad 1/2 \quad 4 \\
 D_3 = C_3 / (-2) & 0 & 0 \quad 1 \quad -3/2 \quad -3 \\
 D_4 = C_4 - D_3 & 0 & 0 \quad 0 \quad 3/2 \quad 6
 \end{array}$$

Наконец, исключив x_4 из всех уравнений, кроме последнего,

получим

$$\begin{array}{lcl}
 E_1 = D_1 + (0)E_4 & 1 & 0 \quad 0 \quad 0 \quad 1 \\
 E_2 = D_2 - (1/2)E_4 & 0 & 1 \quad 0 \quad 0 \quad 2 \\
 E_3 = D_3 + (3/2)E_4 & 0 & 0 \quad 1 \quad 0 \quad 3 \\
 E_4 = D_4 / (3/2) & 0 & 0 \quad 0 \quad 1 \quad 4
 \end{array}$$

Совершенно очевидно, что этот метод облегчает получение решения. Его недостатком является увеличение объема вычислений. Применение обоих описанных методов усложняется, если какой - либо из элементов ведущего ряда равен нулю. В последнем случае ведущий ряд нельзя нормировать. Однако, изменив порядок, в котором расположены уравнения системы, эту трудность можно обойти. Можно показать, что наибольшая точность достигается тогда, когда ведущий элемент имеет наибольшее значение. Поэтому строку с нулевым или малым ведущим элементом надо заменить на ту из стоящих под ней строк, в которой в том же столбце стоит элемент, имеющий наибольшее значение.

3.4 Отыскание обратной матрицы методом исключения Гаусса – Жордана [2]

Метод исключения Гаусса - Жордана дает систематический способ преобразования матрицы в единичную. Точно такой же процесс нужен для отыскания матрицы, обратной к данной. Для реализации этого процесса пользователь должен дополнить исходную матрицу единичной матрицей.

Тогда в конце процесса исключения, когда исходная матрица преобразуется в единичную, дополняющая часть, которая сначала была единичной, превращается в обратную к исходной матрице.

3.5 Метод Холецкого для систем линейных уравнений (метод квадратных корней) [2]

Метод Холецкого, известный также как метод Кроута, более экономичен по обращению к оперативной памяти и по машинному времени, чем метод Гаусса - Жордана, и поэтому более предпочтителен к использованию на малом компьютере.

В основе метода Холецкого лежит тот факт, что расширенная матрица

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{bmatrix},$$

представляющая систему уравнений, может быть редуцирована к эквивалентной верхней треугольной системе вида

$$U = \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} & u_{15} \\ 0 & 1 & u_{23} & u_{24} & u_{25} \\ 0 & 0 & 1 & u_{34} & u_{35} \\ 0 & 0 & 0 & u_{44} & u_{45} \end{bmatrix}$$

Если этот вид найден, решение системы уравнений находят обратной подстановкой, начиная с последнего уравнения. Первый способ добиться этой верхней треугольной формы - это использовать гауссово исключение. Другой способ - найти такую преобразующую матрицу, которая, будучи умножена на матрицу U , преобразует ее в исходную матрицу A . Можно показать, что преобразующая матрица окажется нижней треугольной матрицей вида

$$L = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}$$

В результате получим $[L][U] = [A]$

или

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} & u_{15} \\ 0 & 1 & u_{23} & u_{24} & u_{25} \\ 0 & 0 & 1 & u_{34} & u_{35} \\ 0 & 0 & 0 & 1 & u_{45} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{bmatrix}$$

Соотношения для отыскания элементов матриц L и U могут быть найдены путем перемножения этих матриц и сравнения членов с обеих сторон равенства. Например,

$$l_{i1} = a_{i1}, \quad i = 1, 2, \dots, n$$

дает первый столбец L . Зная его, первую строку U можно найти так:

$$u_{1j} = a_{1j} / l_{11}, \quad j = 2, 3, \dots, n+1$$

Затем находят второй столбец L и вторую строку U :

$$l_{i2} = a_{i2} - l_{21}u_{12}, \quad i = 2, 3, \dots, n;$$

$$u_{2j} = (a_{2j} - l_{21}u_{1j}) / l_{22}, \quad j = 3, 4, \dots, n+1$$

Потом находят третий столбец L и третью строку U и так далее. Процесс продолжается, пока не найдены все строки и столбцы. В таком случае первый столбец L устанавливается автоматически без дополнительных соглашений. Остальные соотношения, устанавливаемые по соглашению, имеют следующий вид.

1. Для первой строки $a_{1j} = a_{1j} / a_{11}, \quad j = 2, 3, \dots, n+1$

2. Для m -го столбца и m -й строки ($m = 2, 3, \dots, n$)

а) для очередного l_{im} столбца

$$a_{im} = a_{im} - \sum_{k=1}^{m-1} a_{ik}a_{km}, \quad i = m, \dots, n;$$

б) для очередной u_{mj} строки

$$a_{mj} = \frac{a_{mj} - \sum_{k=1}^{m-1} a_{mk} a_{kj}}{a_{mm}}, \quad j = m+1, \dots, n+1.$$

После того как матрица U стала известна, значения решений x_i могут быть найдены обратной подстановкой: $x_n = a_{n,n+1}$,

$$x_i = a_{i,n+1} - \sum_{k=i+1}^n a_{ik} x_k, \quad i = n-1, \dots, 1$$

Поскольку при вычислении столбцов такая процедура содержит деление на a_{mm} важно, чтобы эти элементы не были равны нулю. Наилучшая точность обеспечивается тем, что значения a_{mm} - наибольшие по модулю из возможных.

Если матрица A симметричная, то число вычислительных операций и объем памяти, необходимые для нахождения решения по методу Холесского, могут быть еще в большей степени уменьшены.

В настоящее время разработано много точных методов численного решения систем линейных алгебраических уравнений. Требуемое количество арифметических операций при расчетах с помощью метода Гаусса составляет примерно $(2/3)m^3$. Метод Гаусса не является единственным методом исключения, используемым для решения систем линейных уравнений и приведения матриц к треугольному виду. Существуют два метода исключения, обладающих в отличие от метода Гаусса гарантированной хорошей обусловленностью — метод вращений и метод отражений. Среди выше рассмотренных методов в настоящее время метод отражения рассматривается как один из наиболее устойчивых к вычислительной погрешности, но более трудоемкий по сравнению с методом Гаусса.

Для решения систем линейных алгебраических уравнений с трехдиагональными матрицами используется метод прогонки [3], который является важным частным случаем метода Гаусса..

Контрольные вопросы

1. Какие существуют типы матриц?
2. Назовите прямые методы решения систем линейных алгебраических уравнений.
3. К какой категории методов вычислительной математики относится метод Гаусса?

Рекомендуемая литература

1. Щуп, Т. Прикладные численные методы в физике и технике [Текст] / Т. Е. Щуп // Под ред. С. П. Меркурьева. - М.: Высш.шк., 1990.-255 с.
2. Ращиков, В.И. Численные методы решения физических задач [Текст]: учеб. пособие / В. И. Ращиков, А. С. Рошаль. – СПб.: Лань, 2005.- 208 с.
3. Амосов, А.А. Вычислительные методы для инженеров [Текст]: учеб. пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. - М.: Высшая школа, 1994. – 544 с.

ЛЕКЦИЯ 4 ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ (СЛАУ)

В данной лекции рассматриваются итерационные способы решения систем линейных алгебраических уравнений и обращения матриц, служащие серьезной альтернативой прямым методам решения таких задач, по крайней мере, в случаях, когда их размерность велика. Показывается логика построения нескольких наиболее важных итерационных процессов, таких как методы простых итераций, Якоби, Зейделя, релаксации.

Ключевые слова: МЕТОД ПРОСТЫХ ИТЕРАЦИЙ, МЕТОД ЯКОБИ, МЕТОД ЗЕЙДЕЛЯ, МЕТОД РЕЛАКСАЦИИ...

Итерационными называются численные методы, обладающие следующими особенностями [1]:

- решение находится с помощью последовательных приближений — итераций, начиная с некоторого начального приближения («0-й итерации»), которое должно быть задано (выбрано) заранее;
- решение является приближенным, но с любой заданной погрешностью;
- требуемое для достижения заданной погрешности число итераций, а следовательно, и число арифметических операций определяется в ходе счета и заранее не известно.

Итерационные методы обычно устойчивее прямых к погрешностям округлений, так как эти погрешности постоянно корректируются в ходе итераций наряду с другими типами погрешностей. Различают одношаговые и многошаговые итерационные методы. В одношаговых методах для вычисления очередной итерации достаточно знания одной, предыдущей итерации, тогда как в многошаговых методах используются несколько предыдущих итераций. Одношаговые итерационные методы решения

системы уравнений можно записать в общей, так называемой канонической форме:

$$B^{(k)} \frac{x^{(k)} - x^{(k-1)}}{\tau_k} + Ax^{(k-1)} = b \quad (k=1, 2, \dots)$$

Здесь k — номер итерации, $x^{(0)}$ — начальное приближение, а матрицы $B^{(k)}$ и числа τ_k , называемые итерационным параметром, задают тот или иной итерационный процесс. Очередное приближение $x^{(k)}$ находится из уравнения

4.1 Решение СЛАУ методом простых итераций [1]

Система стандартного вида

$$Ax = b, \quad (1)$$

где $A = (a_{ij})_{i,j=1}^n$ — $n \times n$ -матрица, а $x = (x_1, \dots, x_n)^T$ и $b = (b_1, \dots, b_n)^T$ —

n -мерные векторы-столбцы, может быть преобразована к эквивалентной ей системе вида

$$x = Bx + c, \quad (2)$$

где x — тот же вектор неизвестных, а B и c — некоторые новые матрица и вектор соответственно. Систему (2) можно трактовать как задачу о неподвижной точке линейного отображения B в пространстве R_n и по аналогии со скалярным случаем определить последовательность приближений $x^{(k)}$ к неподвижной точке x^* рекуррентным равенством

$$x^{(k+1)} = Bx^{(k)} + c, \quad k=0, 1, 2, \dots \quad (3)$$

Итерационный процесс (3), начинающийся с некоторого вектора

$$x^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})^T,$$

будем называть методом простых итераций (коротко МПИ).

Изучим комплекс вопросов о сходимости этого процесса. А именно:

1) какие нужно предъявить требования к B , c и $x^{(0)}$, чтобы последовательность $(x^{(k)})$ при $k \rightarrow \infty$ имела пределом x^* — неподвижную точку задачи (2) (и значит, решение эквивалентной (2) исходной системы (1))?

2) с какой скоростью сходится этот процесс, т.е. каков закон убывания абсолютных погрешностей получаемых по формуле (3) приближений?

3) сколько нужно сделать итераций по формуле (3), чтобы при заданном начальном приближении $x^{(0)}$ найти решение задачи (2) с заданной точностью?

Ответы на подобные вопросы теории итерационных методов в R_n часто опираются на следующие два утверждения о сходимости степенных матричных рядов, точнее, «матричной геометрической прогрессии». Во втором из них, а также всюду далее под нормой матрицы понимается мультипликативная норма такая, что $\|E\| = 1$ (E — единичная матрица).

Необходимым и достаточным условием сходимости метода простых итераций (3) при любом начальном векторе к решению x^* системы (2) является требование, чтобы все собственные числа матрицы B были по модулю меньше 1.

Пусть $\|B\| \leq q < 1$. Доказано, что при любом начальном векторе $x^{(0)}$ МПИ (3) сходится к единственному решению x^* задачи (2) и при всех $k \in N$ справедливы оценки погрешности:

$$1) \|x^* - x^{(k)}\| \leq \frac{q}{1-q} \|x^{(k)} - x^{(k-1)}\| \quad (\text{апостериорная-«из опыта»});$$

$$2) \|x^* - x^{(k)}\| \leq \frac{q^k}{1-q} \|x^{(1)} - x^{(0)}\| \quad (\text{априорная-«до опыта»}).$$

Априорная оценка, как правило, грубее апостериорной.

Сравнительная точность апостериорных оценок заметно увеличивается с увеличением номера итерации.

Априорная оценка позволяет подсчитывать заранее число итераций k , достаточное для получения решения x^* с заданной точностью ε (в смысле допустимого уровня абсолютных погрешностей) при выбранном начальном векторе $x^{(0)}$. Для этого нужно найти наименьшее целое решение неравенства

$$\frac{q^k}{1-q} \|x^{(1)} - x^{(0)}\| \leq \varepsilon$$

относительно переменной k . Апостериорной же оценкой удобно пользоваться непосредственно в процессе вычислений и останавливать этот процесс, как только выполнится неравенство

$$\|x^{(k)} - x^{(k-1)}\| \leq \frac{1-q}{q} \varepsilon.$$

Отметим, что неравенство $\|x^{(k)} - x^{(k-1)}\| \leq \varepsilon$ будет гарантией выполнения неравенства $\|x^* - x^{(k)}\| \leq \varepsilon$ только в том случае, когда $q \leq \frac{1}{2}$.

По поводу выбора начального приближения.

Как установлено выше, сходимость МПИ (3) при условии $|\lambda_B| < 1$ гарантируется при любом начальном векторе $x^{(0)}$. Очевидно, итераций потребуется тем меньше, чем ближе $x^{(0)}$ к x^* . Если нет никакой дополнительной информации о решении задачи (2) (например, может быть известным решение близкой задачи или грубое решение данной задачи), за $x^{(0)}$ обычно принимают вектор c свободных членов системы (2). Мотивация этого может быть такой: матрица B «мала», значит вектор Bx «мал», следовательно, и вектор x^* не должен сильно отличаться от вектора c . При выборе $x^{(0)} = c$ фигурирующая в теореме 3.2 априорная оценка принимает вид

$$\|x^* - x^{(k)}\| \leq \frac{\|c\|}{1-q} q^{k+1}, \quad k \in N$$

4.2 Метод Якоби [2]

Вернемся к рассмотрению СЛАУ в виде (1). После выяснения условия, которому должна удовлетворять матрица коэффициентов приведенной системы (2) для сходимости МПИ (3), следует осуществить приведение системы (1) к виду (2) так чтобы это условие выполнялось. Рассмотрим один из способов такого приведения, достаточно эффективный в определенных случаях.

Представим матрицу A системы (1) в виде

$$A = L + D + R,$$

где D — диагональная, а L и R — соответственно левая и правая строго треугольные (т.е. с нулевой диагональю) матрицы. Тогда система (1) может быть записана в виде

$$Lx + Dx + Rx = b \quad (4)$$

и если на диагонали исходной матрицы нет нулей, то эквивалентной (1) задачей вида (2) будет

$$x = -D^{-1}(L + R)x + D^{-1}b \quad (5)$$

т.е. в равенствах (2) и (3) следует положить

$$B = -D^{-1}(L + R), \quad c = D^{-1}b$$

Основанный на таком приведении системы (1) к виду (2) метод простых итераций (3) называют методом Якоби. В векторно-матричных обозначениях он определяется формулой

$$x^{(k+1)} = -D^{-1}(L + R)x^{(k)} + D^{-1}b, \quad k=0, 1, 2, \dots \quad (6)$$

Чтобы записать метод Якоби (6) решения системы (1) в развернутом виде, достаточно заметить, что обратной матрицей к матрице $D = (a_{ii})_{i=1}^n$ служит диагональная матрица D^{-1} с элементами диагонали $d_{ii} = 1 / a_{ii}$. Поэтому представление (5) системы (1), записанной в виде (4), равнозначно выражению «диагональных неизвестных» через остальные:

нулевым шаге заполненный компонентами заданного начального вектора $x^{(0)}$. В таком случае на обычной однопроцессорной вычислительной машине элементы массива x будут постепенно замещаться новыми элементами. Аналогичный взгляд на МПИ (3) показывает, что для компьютерной реализации одного его шага нужно целиком сохранять n -элементный массив x , подставляемый в правую часть, до тех пор, пока не сформируется полностью другой n -элементный массив — результат данного итерационного шага. В связи с такой интерпретацией метод Зейделя называют *методом последовательных смещений*, а метод простых итераций — *одновременных смещений*.

Если в матрице A системы (1) имеет место диагональное преобладание, то метод Зейделя сходится, причем быстрее, чем метод Якоби.

4.4 Понятие о методе релаксации [2]

В случаях, когда применение оценок погрешностей в методах простых итераций и Зейделя невозможно из-за отсутствия констант $q < 1$ или $t < 1$, ограничивающих сверху какие-либо нормы матрицы итерирования соответствующего метода, эти методы неэффективны и, более того, малонадежны ввиду медленной сходимости. Рассмотрим одно обобщение метода Зейделя, позволяющее иногда в несколько раз ускорить сходимость итерационной последовательности.

Пусть $z_i^{(k)}$ — обозначение i -й компоненты k -го приближения к решению системы (1) по методу Зейделя, а обозначение $x_i^{(k)}$ будем использовать для i -й компоненты k -го приближения, получаемого новым методом. Этот метод определим равенством

$$x_i^{(k+1)} = x_i^{(k)} + \omega \left(z_i^{(k+1)} - x_i^{(k)} \right) \quad (12)$$

где $i = 1, 2, \dots, n$; $k = 0, 1, 2, \dots$; $x_i^{(0)}$ — задаваемые начальные значения; ω — числовой параметр, который называют параметром релаксации.

Очевидно, при $\omega = 1$ метод (12), называемый методом релаксации (ослабления), совпадает с методом Зейделя.

Конкретизируем метод релаксации для случая, когда исходная система (1) представляется в виде (4) и, следовательно, метод Зейделя имеет вид (10).

Пользуясь введенными здесь обозначениями, запишем на основании (10) дополнительное к (12) равенство для выражения компонент векторов

$z^{(k)} = \left(z_i^{(k)} \right)_{i=1}^n$ через компоненты векторов $x^{(k)} = \left(x_i^{(k)} \right)_{i=1}^n$:

$$z_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) \quad (13)$$

Таким образом, метод релаксации можно понимать как поочередное применение формул (13) и (12) при каждом $k = 0, 1, 2, \dots$

По теореме Островского - Рейча для нормальной системы $Ax = b$ метод релаксации сходится при любом $x^{(0)}$ и любом $\omega \in (0, 2)$.

При значениях $\omega \in (1, 2)$ метод называют методом последовательной верхней релаксации (сокращенно ПВР). Ввиду низкой эффективности метода при $\omega \in (0, 1)$, называемого в этом случае методом нижней релаксации, название «метод ПВР» в последнее время относят ко всему семейству методов, т.е. для любых $\omega \in (0, 2)$. При этом случай $\omega \in (1, 2)$ называют сверхрелаксацией.

4.5 О других итерационных методах решения СЛАУ[3]

Распространенным методом минимизации функций большого числа переменных является метод градиентного спуска. Последующее приближение получается из предыдущего смещением в направлении, противоположном градиенту функции $F(x)$. Каждое следующее приближение ищется в виде

$$x^{n+1} = x^n - \delta_n \text{grad} F(x^n) \quad (14)$$

Если параметр δ_n определять из условия минимума величины $F(x^n - \delta_n \text{grad} F(x^n))$, то в этом случае рассматриваемый метод называют методом наискорейшего градиентного спуска или просто *методом наискорейшего спуска*.

Для функции $F(x) = (Ax, x) - 2(b, x)$, соответствующей системе линейных уравнений с матрицей $A = A^T > 0$, задача нахождения минимума решается в явном виде. Обозначим $2\delta_n = \Delta_n$ и в окончательном виде получим

$$x^{n+1} = x^n - \Delta_n (Ax^n - b), \quad (15)$$

$$\Delta_n = \frac{(Ax^n - b, Ax^n - b)}{(A(Ax^n - b), Ax^n - b)}. \quad (16)$$

Отметим, что метод наискорейшего спуска является нелинейным итерационным методом; параметры метода на каждом шаге выбираются в зависимости от полученного приближения. У метода наискорейшего спуска (15), (16), однако, есть следующий недостаток по сравнению с простейшим процессом. При нахождении каждого следующего приближения он требует не одной, а двух трудоемких операций умножения матрицы на вектор. В исходном методе наискорейшего спуска погрешность на шаге итерации равносильна возмущению начального приближения и, поскольку процесс сходящийся, ее влияние должно иметь тенденцию к затуханию.

Существуют и другие итерационные методы:

- Метод сопряженных градиентов, предназначенный для решения систем линейных алгебраических уравнений с симметричной положительно определенной матрицей.
- Итерационные методы с использованием спектрально-эквивалентных операторов

Контрольные вопросы

1. Какие численные методы называются итерационными?
2. Какие существуют итерационные методы решения СЛАУ?

Рекомендуемая литература

1. Рашиков, В. И. Численные методы решения физических задач [Текст]: учеб. пособие / В. И. Рашиков, А. С. Рошаль. – СПб.: Лань, 2005.- 208 с.
2. Вержбицкий, В. М. Основы численных методов [Текст]: учеб. для вузов / В. М. Вержбицкий. – М.: Высш. шк., 2002. – 840 с.
3. Бахвалов, Н. С. Численные методы [Текст]: учеб. пособие для физмат. специальностей / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – М.: Физматлит, 2002.- 627 с.

ЛЕКЦИЯ 5 МЕТОДЫ ОТЫСКАНИЯ РЕШЕНИЙ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

В данной лекции дается общее представление о методах отыскания решений нелинейных уравнений

Ключевые слова: МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ, МЕТОД ХОРД, МЕТОД НЬЮТОНА, МЕТОД СЕКУЩИХ, МЕТОД ПРОСТОЙ ИТЕРАЦИИ, МЕТОД ПАРАБОЛ.

Обычно нелинейные уравнения делят на трансцендентные и алгебраические, хотя часто они решаются одними и теми же методами [1].

Трансцендентными называют нелинейные уравнения, содержащие тригонометрические функции или другие специальные функции, например $\lg x$ или e^x .

Методы решения нелинейных уравнений такого типа делятся на прямые и итерационные. Первые позволяют найти решение непосредственно с помощью формул и всегда обеспечивают получение точного решения. Известным примером такого рода является формула корней квадратного уравнения. В итерационных методах процедура решения задается в виде многократного применения некоторого алгоритма. Полученное решение всегда является приближенным, хотя может быть сколько угодно близко к точному. Итерационные методы наиболее удобны для реализации на ЭВМ. В каждом из излагаемых методов считается, что процесс решения задачи состоит в отыскании действительных корней (нулей) уравнения.

Решение задачи отыскания корней нелинейного уравнения осуществляют в два этапа. Первый этап называется *этапом локализации (или отделения) корней*, второй – *этапом итерационного уточнения корней* [2].

5.1 Метод половинного деления [1]

Он состоит из следующих операций. Сначала вычисляют значения функций в точках, расположенных через равные интервалы на оси X . Это

делают до тех пор, пока не будут найдены два последовательных значения функции $f(x_n)$ и $f(x_{n+1})$ имеющие противоположные знаки. (Напомним, если функция непрерывна, изменение знака указывает на существование корня.) Затем по формуле

$$x_{cp} = (x_{n+1} + x_n) / 2 \quad (1)$$

вычисляют середину промежутка $[x_n, x_{n+1}]$ и находят значение функции $f(x_{cp})$. Если знак $f(x_{cp})$ совпадает со знаком $f(x_n)$, то в дальнейшем вместо $f(x_n)$ используют $f(x_{cp})$. Если же $f(x_{cp})$ имеет знак, противоположный знаку $f(x_n)$, т.е. ее знак совпадает со знаком $f(x_{n+1})$, то на $f(x_{cp})$ заменяют это значение функции. В результате интервал, в котором заключено значение корня, сужается. Если значение $f(x_{cp})$ достаточно близко к нулю, процесс заканчивают, в противном случае его продолжают. На рис. 1 эта процедура показана графически. Хотя метод

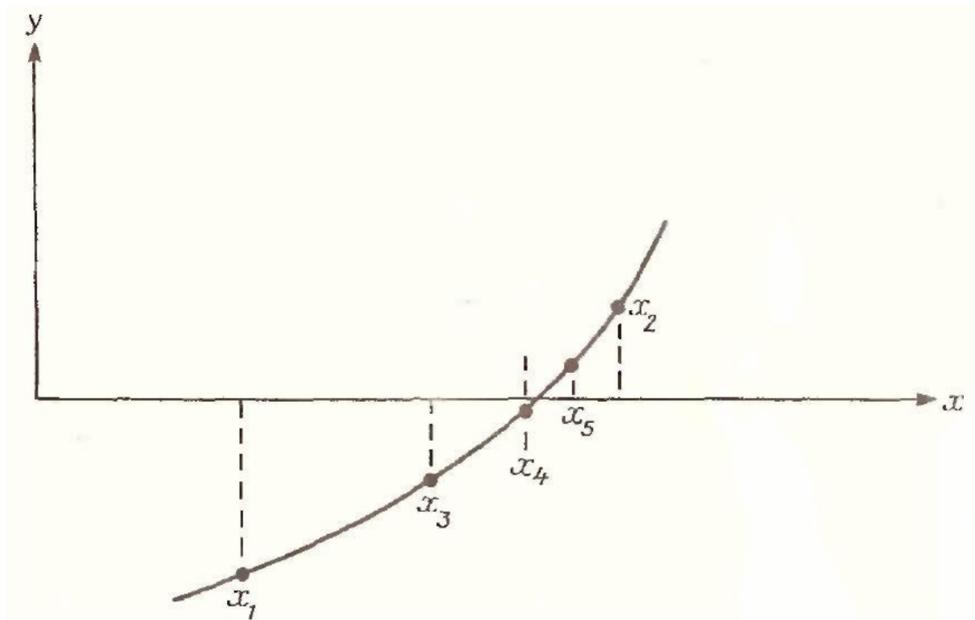


Рис.1

половинного деления не обладает высокой вычислительной эффективностью, при увеличении числа итераций он обеспечивает получение все более

точного приближенного значения корня. После того как найден первый интервал, в котором заключен корень, его ширина после N итераций убывает в $2N$ раза.

5.2 Метод хорд [1]

В основе этого метода лежит линейная интерполяция по двум значениям функции, имеющим противоположные знаки. При отыскании корня этот метод нередко обеспечивает более быструю сходимость, чем предыдущий. Сначала определяют значения функции в точках, расположенных на оси X через равные интервалы. Это делают до тех пор, пока не будет найдена пара последовательных значений функции $f(x_n)$ и $f(x_{n+1})$, имеющих противоположные знаки. Прямая, проведенная через эти две точки, пересекает ось X при значении

$$x^* = x_n - f(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)} \quad (2)$$

Это значение аргумента используют для определения значения функции $f(x^*)$, которое сравнивают со значениями функций $f(x_n)$ и $f(x_{n+1})$ и в дальнейшем применяют вместо того из них, с которым оно совпадает по знаку. Если значение $f(x^*)$ недостаточно близко к нулю, всю процедуру повторяют до тех пор, пока не будет достигнута необходимая степень сходимости. На рис. 2 процесс решения показан графически.

2.3.3 Метод Ньютона [1]

Метод последовательных приближений, разработанный Ньютоном, используют наиболее часто среди всех итерационных алгоритмов. Его популярность обусловлена тем, что в отличие от двух предыдущих методов для определения интервала, в котором заключен корень, не требуется

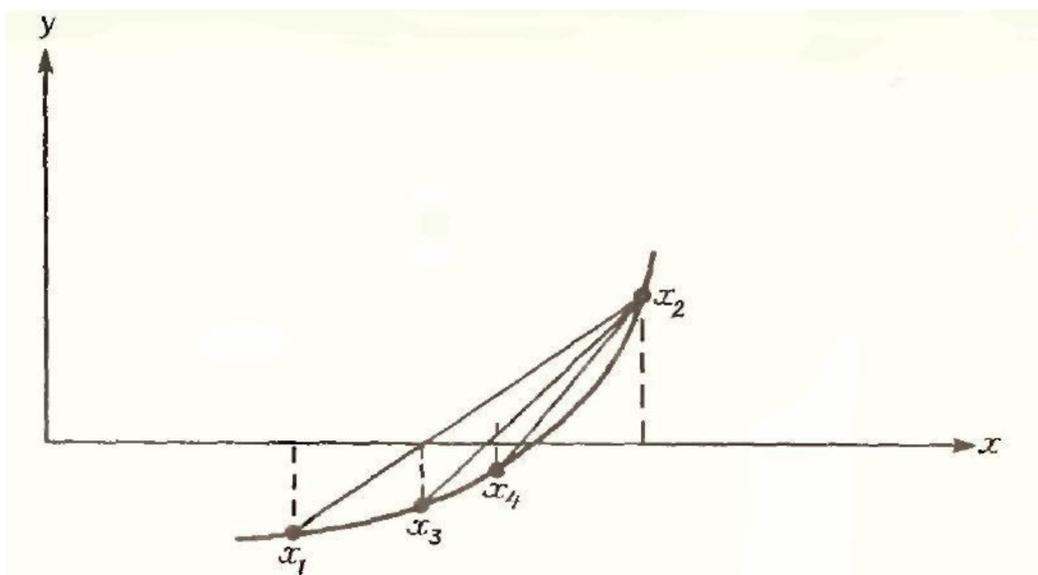


Рис.2

находить значения функции с противоположными знаками. Вместо интерполяции по двум значениям функции экстраполяцию в методе Ньютона осуществляют с помощью касательной к кривой в данной точке. В основе этого метода лежит разложение функции $f(x)$ в ряд Тейлора

$$f(x_n + h) = f(x_n) + hf'(x_n) + (h^2/2)f''(x_n) = \dots \quad (3)$$

Члены, содержащие h во второй и более высоких степенях, отбрасывают; используется соотношение $x_n + h = x_{n+1}$. Предполагается, что переход от x_n к x_{n+1} приближает значение функции к нулю так, что $f(x_n + h) = 0$. Тогда

$$x_{n+1} = x_n - f(x_n) / f'(x_n) \quad (4)$$

Значение x_{n+1} соответствует точке, в которой касательная к кривой в точке x_n пересекает ось X. Так как кривая $f(x)$ отлична от прямой, то значение функции $f(x_{n+1})$ скорее всего не будет в точности равно нулю. Поэтому всю процедуру повторяют, причем вместо x_n используют x_{n+1} . Счет прекращается по достижении достаточно малого значения $f(x_{n+1})$. На рис. 3 процесс решения уравнения методом Ньютона показан графически. Совершенно ясно, что быстрота сходимости в большой мере зависит от удачного выбора исходной точки. Если в процессе итераций тангенс угла

наклонной касательной обращается в нуль, то применение метода усложняется. Можно также показать, что в случае бесконечно большого $f''(x)$ метод также не является достаточно эффективным. Так как условие кратности корней имеет вид $f(x) = f'(x) = 0$, то в этом случае метод Ньютона не обеспечивает сходимость. Отметим, что иногда используют другой способ контроля сходимости, состоящий в сравнении x_n и x_{n+1} .

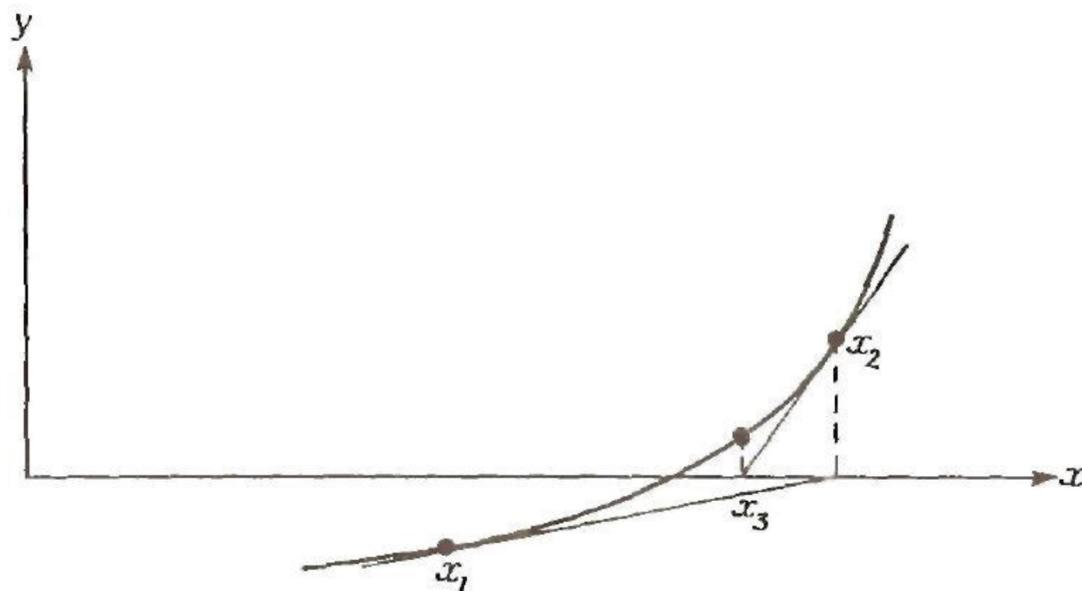


Рис.3

5.4 Метод секущих [1]

Один из недостатков метода Ньютона состоит в том, что, пользуясь им, приходится дифференцировать функцию $f(x)$. Если нахождение производной затруднено, то можно воспользоваться некоторым приближением, которое и составляет основу *метода секущих*. Заменяв производную $f'(x_n)$ используемую в методе Ньютона (4) разностью последовательных значений функции, отнесенной к разности значений аргумента

$$F^*(x_n) = (f(x_n) - f(x_{n-1})) / (x_n - x_{n-1}), \quad (5)$$

получим следующую итерационную формулу:

$$x_{n+1} = x_n - f(x_n)F^*(x_n). \quad (6)$$

Схема алгоритма для этого метода та же, что и для метода Ньютона (несколько иной вид имеет итерационная формула). В сущности, в методе секущих для отыскания корня используют комбинацию интерполяции и экстраполяции. В своей интерполяционной части этот метод эквивалентен методу хорд. Как и в случае метода Ньютона, счет заканчивают, когда последовательные значения x совпадают с некоторой приемлемой точностью или когда значение функции $f(x)$ становится достаточно близким к нулю. В случае кратных корней при использовании метода секущих возникают те же трудности, что и при использовании метода Ньютона.

5.5 Метод простой итерации [3]

Для применения этого метода уравнение

$$f(x) = 0 \quad (7)$$

представляют в следующем виде:

$$x = \varphi(x). \quad (8)$$

т. е. таким, что из $f(x^*) = 0$ следует $x^* = \varphi(x^*)$ и наоборот. Привести уравнение (7) к (8) можно многими способами, например, положив $\varphi(x) = x + \psi(x)f(x)$, где $\psi(x)$ - непрерывная произвольная знакопостоянная функция.

Геометрически уравнение (8) на интервале отделения корня представляется в виде двух пересекающихся линий $y = x$ и $y = \varphi(x)$ (рис. 4). Полагая, что известно начальное приближение $x^{(0)}$ для значения корня x^* построим итерационный процесс

$$x^{(n+1)} = \varphi(x^{(n)}), \quad n = 0, 1, 2, \dots, \quad (9)$$

изображенный на рис. 4 ломаной линией со стрелочками, указывающими направление движения. Видим, что для представленного случая взаимного расположения линий $y = x$ и $y = \varphi(x)$ неограниченное повторение вычислений по соотношению (9) позволяет сколь угодно близко подойти к точному значению корня x^* .

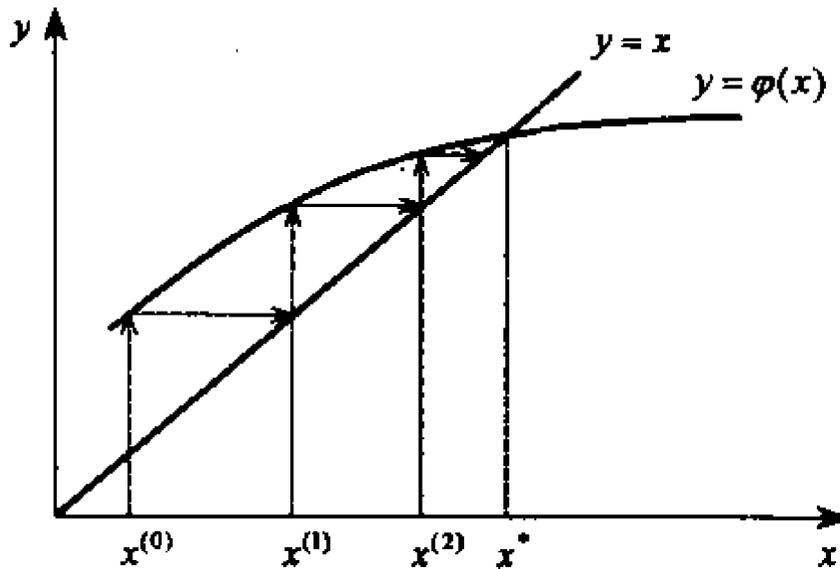


Рис.4

Простота метода такой итерации делает его привлекательным, однако не следует забывать, что и этому методу присущи недостатки, так что он не всегда обеспечивает сходимость. Поэтому для любой программы, в которой используют этот алгоритм, необходимо предусматривать контроль сходимости и прекращать счет, если сходимость не обеспечивается.

5.6 Метод парабол (метод квадратичной интерполяции).

Метод Мюллера для уточнения корней нелинейных уравнений [3]

Рассмотренный выше метод секущих можно интерпретировать как метод, в котором на каждой итерации исходная функция аппроксимируется линейной функцией (секущей), построенной по двум точкам, принадлежащим $f(x)$. Развивая далее идеи аппроксимации, можно для

построения итерационных формул использовать информацию о функции в нескольких точках, предшествующих точке $x^{(k)}$. В методе парабол по трем последовательным приближениям

$$\left[x^{(k-2)}, f(x^{(k-2)}) \right], \quad \left[x^{(k-1)}, f(x^{(k-1)}) \right], \quad \left[x^{(k)}, f(x^{(k)}) \right]$$

строится многочлен второй степени (парабола), приближающий исходную функцию. За новое приближение берется обычно ближайший к $x^{(k)}$ корень соответствующего квадратного уравнения. Геометрическая интерпретация метода парабол дана на рис. 5.

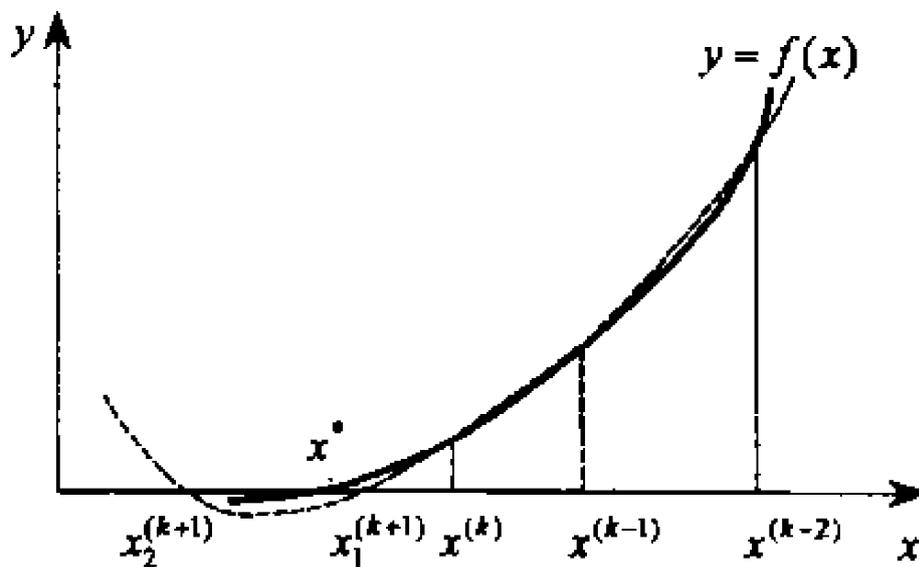


Рис.5

В качестве $x^{(k+1)}$ выбирается тот из корней квадратного уравнения, у которого величина наименьшая. Доказывается, что погрешность метода определяется соотношением

$$\varepsilon^{(k+1)} \approx \varepsilon^{(k)} \varepsilon^{(k-1)} \varepsilon^{(k-2)} \approx \left[\varepsilon^{(k)} \right]^p, \text{ где } p = 1.839.$$

Это означает, что, несмотря на привлечение дополнительной информации о функции, метод парабол имеет порядок сходимости, лишь немного превышающий порядок сходимости метода секущих. Вместе с тем возникают задачи решения квадратного уравнения, выбора одного из двух корней многочлена и, самое важное, определения области гарантированной сходимости метода. Если три приближения для построения многочлена

выбраны далеко от корня и содержат погрешности, то возможно самое неожиданное поведение решения.

Отметим, что метод парабол успешно применяется для отыскания корней многочленов, в том числе комплексных; при этом метод обладает тем замечательным свойством, что начальное приближение может быть действительным. Метод парабол является трехшаговым методом.

Контрольные вопросы

1. В чем состоит основная идея метода половинного деления?
2. Может ли метод половинного деления дать точное значение корня уравнения?
3. По каким причинам методы хорд и касательных предпочтительнее метода простой итерации?
4. Какой из перечисленных методов является трехшаговым методом?

Рекомендуемая литература

1. Щуп, Т. Прикладные численные методы в физике и технике [Текст] / Т. Щуп // Под ред. С. П. Меркурьева. – М.: Высш шк., 1990. – 255 с.
2. Амосов, А. А. Вычислительные методы для инженеров [Текст]: учеб пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. - М.: Высшая школа, 1994. – 544 с.
3. Пирумов, У. Г. Численные методы [Текст]: учеб. пособие / У. Г. Пирумов. – М.: Изд-во МАИ, 1998. – 188 с.

ЛЕКЦИЯ 6 МЕТОДЫ ОТЫСКАНИЯ РЕШЕНИЙ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ [1]

В данной лекции рассматривается задача отыскания решений систем нелинейных уравнений, существенно более сложная по сравнению с задачей отыскания решения одного нелинейного уравнения или системы линейных алгебраических уравнений.

Ключевые слова: МЕТОД ПРОСТОЙ ИТЕРАЦИИ, МЕТОД ИТЕРАЦИЙ ЗЕЙДЕЛЯ, МЕТОД НЬЮТОНА, УСКОРЕНИЕ СХОДИМОСТИ ПО ЭЙТКЕНУ, ИСПОЛЬЗОВАНИЕ ФОРМУЛ ЧИСЛЕННОГО ДИФФЕРЕНЦИРОВАНИЯ, МЕТОД ЛОЖНОГО ПОЛОЖЕНИЯ, МЕТОД СЕКУЩИХ, МЕТОД СТЕФФЕНСЕНА ...

Список обозначений

$\mathbf{x} = (x_1, \dots, x_N)^T$ вектор (жирные строчные буквы)

ε_p требуемая точность вычисления приближённого решения

$\mathbf{x}^* = (x_1^*, \dots, x_N^*)^T$ точное решение системы нелинейных уравнений

$\mathbf{x}^{(k)} = (x_1^{(k)}, \dots, x_N^{(k)})^T$ k -ое приближение к точному решению

$\mathbf{e}^{(k)} = \mathbf{x}^* - \mathbf{x}^{(k)} = (e_1^{(k)}, \dots, e_N^{(k)})^T$ ошибка k -го приближения

Наиболее употребительны для уточнения корней систем нелинейных уравнений - методы итерации (метод простой итерации и метод Зейделя) и метод Ньютона. Как и в случае уточнения корней одного нелинейного уравнения, для систем нелинейных уравнений требуется определение хорошего начального приближения (отделение корня), гарантирующего сходимость метода и высокую скорость сходимости. Для системы двух

$$x_i^{(k+1)} = x_i^* - \varepsilon_i^{(k+1)} = \varphi \left[x_1^* - \varepsilon_1^{(k)}, \dots, x_n^* - \varepsilon_n^{(k)} \right], \quad i=1, \dots, n.$$

Полагая наличие у функций $\varphi_i(x)$ непрерывных частных производных и используя $x_i^* = \varphi_i(x^*)$, можем, используя разложение в ряд, получить

$$\varepsilon_i^{(k+1)} \approx \sum_{i=1}^n \varepsilon_i^{(k)} \left. \frac{\partial \varphi_i(x)}{\partial x_i} \right|_{x=x^*}, \quad i=1, \dots, n. \quad (4)$$

Из (4) следует, что в методе простой итерации вектор погрешности испытывает линейное преобразование, или, иначе, метод имеет первый порядок сходимости.

Систему (4) можно переписать в виде

$$\varepsilon^{(k+1)} = A_\varphi \varepsilon^{(k)}, \quad (5)$$

где $A_\varphi = \begin{bmatrix} \left. \frac{\partial \varphi_1(x)}{\partial x_1} & \frac{\partial \varphi_1(x)}{\partial x_2} & \dots & \frac{\partial \varphi_1(x)}{\partial x_n} \right|_{x=x^*} \\ \dots & \dots & \dots & \dots \\ \left. \frac{\partial \varphi_n(x)}{\partial x_1} & \frac{\partial \varphi_n(x)}{\partial x_2} & \dots & \frac{\partial \varphi_n(x)}{\partial x_n} \right|_{x=x^*} \end{bmatrix}$ -матрица производных

системы функций $\varphi_i(x)$ $i=1, \dots, n$ (матрица Якоби). Достаточное условие сходимости итерационного процесса (3) формулируется следующим образом: если какая-либо норма матрицы A_φ , согласованная с рассматриваемой нормой вектора x , меньше единицы, то метод итераций сходится. Условие сходимости $\|A_\varphi\| < 1$ есть обобщение на случай нелинейной системы условия для одного уравнения.

6.2 Метод итераций Зейделя [1]

Нередко сходимость метода простой итерации можно улучшить, если вновь вычисленные значения компонент вектора неизвестных немедленно включить в расчет. В этом случае итерационный процесс имеет вид

$$x_i^{(k+1)} = \varphi_i \left[x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i^{(k)}, \dots, x_n^{(k)} \right], \quad i=1, \dots, n.$$

Сходимость этого процесса также линейная. Так же, как и при решении систем линейных уравнений, может быть поставлена задача об отыскании оптимальной на каждой итерации последовательности уточнения компонент вектора решения. Удовлетворительных методов построения оптимальной последовательности нет. На практике иногда используется упорядочение неизвестных по убыванию разности назначений на двух последовательных итерациях.

6.3 Метод Ньютона [1]

Основная идея метода Ньютона - решение системы нелинейных уравнений $f(x) = 0$ - сводится к решению последовательности линейных задач, дающих в пределе решение исходной задачи. Линейная задача получается путем выделения из нелинейных уравнений главной линейной части.

Рассмотрим погрешность вычисления корня на k -й итерации $\varepsilon^{(k)} = x^* - x^{(k)}$, где $\varepsilon^{(k)} = [\varepsilon_1^{(k)}, \varepsilon_2^{(k)}, \dots, \varepsilon_n^{(k)}]'$. Полагая, что

функции f_i непрерывны и дифференцируемы в окрестности корня и $\varepsilon_i^{(k)}$ ($i = 1, \dots, n$) - малые величины, разложим $f(x^*) = f(x^{(k)} + \varepsilon^{(k)}) = 0$ в ряд Тейлора, сохранив лишь линейную часть разложения. Получим систему уравнений

$$\sum_{j=1}^n \varepsilon_j^{(k)} \left. \frac{\partial f_i(x)}{\partial x_j} \right|_{x=x^*} = -f_i(x^{(k)}), \quad i=1, \dots, n \quad (6)$$

линейную относительно компонент вектора погрешностей. Если использовать эту систему для отыскания компонент вектора погрешностей, то в силу приближенности системы (6) (оставлена лишь линейная часть) найденное значение вектора погрешности будет лишь приближенным. Тогда при подстановке полученного решения в соотношение $x^* = \varepsilon^{(k)} - x^{(k)}$ будем иметь вместо x^* приближенное уточненное значение корня, которое

обозначим $x^{(k+1)}$. Используя запись системы (6) в виде $A_f \varepsilon^{(k)} = -f^{(k)}$, где A_f - матрица производных системы функций f_1 (матрица Якоби), можем записать итерационный процесс для нахождения вектора x :

$$x^{(k+1)} = x^{(k)} - [A_f^{(k)}]^{-1} f^{(k)}, \quad k = 0, 1, 2, \dots,$$

где $[A_f^{(k)}]^{-1}$ - матрица, обратная матрице Якоби. Уточненное значение вектора может быть вновь использовано для получения следующего приближения к корню, что приводит к итерационному процессу. Заметим, что в большинстве случаев предпочтительным является не вычисление обратной матрицы $[A_f^{(k)}]^{-1}$, а получение каким-либо методом решения $\varepsilon^{(k)}$ линейной системы (6) и вычисление нового приближенного значения по соотношению $x^{(k+1)} = x^{(k)} + \varepsilon^{(k)}$

Итерационный процесс (6) сходится, если определитель матрицы $A_f^{(k)}$ отличен от нуля, т.е. $\det(A_f^{(k)}) \neq 0$. Требуется, однако, хорошее отделение корня, но достаточное условие сходимости метода слишком громоздко, чтобы им можно было воспользоваться на практике.

На каждой итерации метода Ньютона требуется вычислять матрицу производных и решать систему линейных уравнений (6). Можно попытаться уменьшить объем вычислений за счет отказа от вычисления матрицы $A_f^{(k)}$ на каждой итерации и использования на всех итерациях постоянного значения $A_f^{(0)}$, вычисленного по начальному приближению. Напомним, что при этом может быть дополнительно существенно уменьшен объем вычислений, если для решения последовательности линейных систем использовать алгоритм, позволяющий выполнить преобразование матрицы $A_f^{(0)}$ к верхней треугольной только один раз. Следует иметь в виду, однако, что, во-первых, указанная модификация метода Ньютона гарантирует лишь линейную сходимость итераций (против квадратичной в окрестности корня в методе

Ньютона) и, во-вторых, константа в линейной зависимости погрешности при неудачном выборе начального приближения может оказаться весьма большой и сходимость будет медленной. Таким образом, увеличивается число итераций, необходимое для достижения заданной точности, и уменьшение общего объема вычислений не гарантировано.

6.4 Ускорение сходимости по Эйткену.

Предположим, что отношение

$$\frac{x^{(k-1)} - x^*}{x^{(k)} - x^*} = q$$

есть величина постоянная и неизменна в процессе итераций. Тогда

$$\frac{x^{(k-1)} - x^*}{x^{(k)} - x^*} = q = \frac{x^{(k-2)} - x^*}{x^{(k-1)} - x^*}.$$

Полученное таким образом x^* можно принять за следующее приближенное значение $x^{(k+1)}$. Предложенный способ пригоден как для одного нелинейного уравнения, так и для систем нелинейных уравнений. Это предположение означает, что метод применим к процессам с линейной сходимостью (простые итерации), но неприменим к методам Ньютона, секущих.

6.5 Использование формул численного дифференцирования [2]

Довольно часто вычисление производных $\frac{\partial f_1}{\partial x_j}$, являющихся элементами матрицы f' , затруднено или вообще невозможно. В такой ситуации для приближенного вычисления производных можно попытаться использовать формулы численного дифференцирования. Например, можно использовать следующую конечно-разностную аппроксимацию производной:

$$\frac{\partial f_i}{\partial x_j}(x^{(n)}) \approx J_{ij}^{(n)} = \frac{1}{h_j^{(n)}} \left(f_i(x_1^{(n)}, \dots, x_{j-1}^{(n)}, x_j^{(n)} + h_j^{(n)}, x_{j+1}^{(n)}, \dots, x_m^{(n)}) - f_i(x_1^{(n)}, \dots, x_m^{(n)}) \right) \quad (7)$$

Параметры $h_j^{(n)} \neq 0$ ($j = 1, 2, \dots, m$) — это *конечно-разностные шаги*.

Если в расчетных формулах метода Ньютона заменить матрицу $f'(x^{(n)})$ аппроксимирующей ее матрицей $J^{(n)}$ с элементами $J_{ij}^{(n)}$, то получим следующий итерационный метод:

$$J^{(n)} \Delta x^{(n+1)} = -f(x^{(n)}) \quad (8)$$

$$x^{(n+1)} = x^{(n)} + \Delta x^{(n+1)} \quad (9)$$

В простейшем варианте этого метода шаги h_j ($j = 1, 2, \dots, m$) не зависят от n . Отметим, что выбор величины шагов представляет собой не очень простую задачу. С одной стороны, они должны быть достаточно малыми, чтобы матрица $J^{(n)}$ хорошо приближала матрицу $f'(x^{(n)})$ с другой, они не могут быть очень малы, так как в этом случае влияние погрешностей вычисления функций f_j на погрешность формулы (7) численного дифференцирования становится катастрофическим (выполняется вычитание близких приближенных чисел).

Следующие три метода можно рассматривать как варианты метода (8), (9), в которых реализованы специальные подходы к вычислению $h^{(n)} = (h_1^{(n)}, h_2^{(n)}, \dots, h_m^{(n)})^T$. Для того чтобы приведенные ниже рассуждения были формально корректными, в формуле (7) положим

$$J_{ij}^{(n)} = \frac{\partial f_i}{\partial x_j}(x^{(n)}), \text{ если оказалось, что } h_j^{(n)} = 0$$

6.6 Метод ложного положения [2]

Пусть c — фиксированный вектор. Положим $h^{(n)} = c - x^{(n)}$. Тогда формулы (7) — (9) определяют метод ложного положения, обладающий линейной скоростью сходимости в случае, если вектор c и начальное приближение $x^{(0)}$ выбраны достаточно близко к решению.

6.7 Метод секущих [2]

В одном из наиболее популярных своих вариантов метод секущих можно рассматривать как метод (7) — (9), где $h^{(n)} = x^{(n-1)} - x^{(n)}$. Метод секущих является двухшаговым: для вычисления очередного приближения $x^{(n+1)}$ используют два предыдущих приближения $x^{(n)}$ и $x^{(n-1)}$. Для того чтобы начать вычисления, необходимо задать два начальных приближения $x^{(0)}$ и $x^{(1)}$.

При удачном выборе начальных приближений $x^{(0)}$ и $x^{(1)}$ метод секущих сходится со сверхлинейной скоростью с порядком сходимости

$$p = (\sqrt{5} + 1) / 2.$$

6.8 Метод Стеффенсена [2]

Вычисления по методу Стеффенсена производят по формулам (7) — (9), где $h^{(n)} = f(x^{(n)})$.

Замечательно то, что хотя этот метод не требует вычисления производных и в отличие от метода секущих является одношаговым, он, как и метод Ньютона, обладает свойством квадратичной сходимости. Как и в методе Ньютона, его применение затруднено необходимостью выбора хорошего начального приближения.

По-видимому, для решения нелинейных систем вида (1) метод Стеффенсена чаще окажется лучшим выбором, чем метод секущих или метод ложного положения.

Следует отметить, что методы секущих и Стеффенсена теряют устойчивость вблизи решения (фактически это происходит при попадании приближения $x^{(n)}$ в область неопределенности решения \bar{x}). Поэтому при использовании этих методов важно вовремя прекратить выполнение итераций.

6.9 О некоторых подходах к решению задач локализации и отыскания решений систем нелинейных уравнений [2]

Подчеркнем еще раз, что одной из наиболее трудных проблем, возникающих при решении систем нелинейных уравнений, является задача локализации решения. Рассмотрим некоторые подходы к ее решению, широко применяемые в практике вычислений. Указанные методы можно применять и для вычисления решения с заданной точностью ε .

1. Использование методов минимизации. Иногда бывает полезно свести задачу отыскания решения системы нелинейных уравнений к задаче отыскания минимума функции многих переменных. В простейшем варианте это сведение выглядит следующим образом. Введем функцию

$$\Phi(x) = \sum_{i=1}^m (f_i(x))^2$$
. Она неотрицательна и достигает своего минимума (равного нулю) тогда и только тогда, когда $f_i(x) = 0$ для всех $i = 1, 2, \dots, m$, т.е. x является решением системы (1).

Применяя для отыскания минимума функции Φ один из итерационных методов минимизации — метод спуска, можно найти вполне удовлетворительное приближение к решению \bar{x} , которое затем имеет смысл использовать как начальное приближение $x^{(0)}$ в одном из итерационных методов решения нелинейных систем.

Выгода от указанного сведения исходной задачи к задаче минимизации состоит в том, что, как правило, методы спуска имеют более широкую область сходимости. Использование методов спуска можно рассматривать и как один из способов локализации решений системы (1). Применение на

заключительном этапе методов, специально ориентированных на решение нелинейных систем, вызвано тем, что вблизи искомого решения методы спуска сходятся медленнее.

Следует отметить, что функция $\Phi(x)$ может иметь и ненулевые локальные минимумы, и в зависимости от выбора начального приближения методы спуска могут приводить к точкам локального минимума, не являющимися решениями системы (1).

2. *Метод продолжения по параметру.* На практике часто встречаются ситуации, когда система нелинейных уравнений естественным образом зависит от некоторого параметра t , т.е. имеет вид

$$f(x, t) = 0, \quad (10)$$

а решение ее следует найти при некотором фиксированном значении параметра, например при $t = 1$. Предположим, что при каждом $t \in [0, 1]$ система (10) имеет решение $\bar{x} = \bar{x}(t)$, непрерывно зависящее от параметра t , причем при $t = 0$ решение системы $f(x, t) = 0$ известно либо легко вычисляется.

Таким образом, семейство решений $\bar{x}(t)$ описывает в пространстве R^m траекторию, начальная точка $\bar{x}(0)$ которой известна, а конечная $\bar{x}(1)$ подлежит определению (рис1).

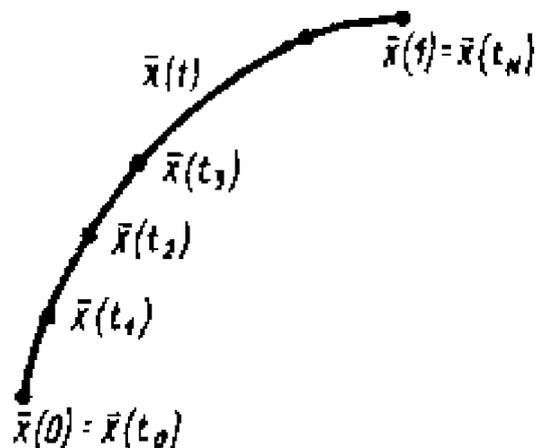


Рис. 1

И в тех случаях, когда система не зависит от параметра, можно ввести параметр t так, чтобы были выполнены указанные выше условия. Например, если x^* — известное приближение к решению системы $f(x) = 0$, то можно рассмотреть систему вида (10), полагая $f(x, t) = f(x) - (1-t)f(x^*)$.

Введем на отрезке $[0, 1]$ набор точек $0 = t_0 < t_1 < \dots < t_N = 1$.

Используя тот или иной итерационный метод, решим последовательно для $k = 0, 1, 2, \dots, N$ системы $f(x, t_k) = 0$. При этом за начальное приближение к $\bar{x}(t_k)$ будем принимать решение $\bar{x}(t_{k-1})$. Если разность $t_k - t_{k-1}$ достаточно мала, то можно ожидать, что $\bar{x}(t_{k-1})$ будет достаточно хорошим начальным приближением к $\bar{x}(t_k)$, обеспечивающим сходимость используемого итерационного метода.

3. Метод дифференцирования по параметру. Иногда метод дифференцирования по параметру называют методом Давиденко. Методы продолжения и дифференцирования по параметру нередко позволяют успешно преодолевать непростую проблему локализации. Однако следует отметить, что эти методы далеко не всегда оказываются эффективными и их практическое применение требует определенной осторожности.

Контрольные вопросы

1. По каким причинам задача решения системы нелинейных уравнений принципиально сложнее задачи решения системы линейных уравнений?
2. В чем состоит метод простой итерации для решения системы нелинейных уравнений?
3. В чем метод Ньютона для решения систем уравнений превосходит метод простой итерации?

Рекомендуемая литература

1. Пирумов, У. Г. Численные методы [Текст]: учеб. пособие / У. Г. Пирумов. – М.: Изд-во МАИ, 1998. – 188 с.
2. Амосов, А. А. Вычислительные методы для инженеров [Текст]: учеб. пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. - М.: Высшая школа, 1994. – 544 с.