

Московский государственный технический университет
имени Н. Э. Баумана

И. К. Марчевский, О. В. Щерица

**ЧИСЛЕННЫЕ МЕТОДЫ
РЕШЕНИЯ ЗАДАЧ
ЛИНЕЙНОЙ АЛГЕБРЫ**

*Методические указания к выполнению
лабораторных работ по курсу
«Методы вычислений»*

Под редакцией *М. П. Галанина*

Москва
Издательство МГТУ им. Н.Э. Баумана
2017

УДК 519.6

ББК 32.97

М30

Издание доступно в электронном виде на портале ...
по адресу: ...

Факультет «Фундаментальные науки»
Кафедра «Прикладная математика»

Рекомендовано

*Редакционно-издательским советом МГТУ им. Н. Э. Баумана
в качестве методических указаний*

Рецензент
канд. техн. наук, доц. ...

Марчевский, И. К.

М30 Численные методы решения задач линейной алгебры :
методические указания к выполнению лабораторных работ по
курсу «Методы вычислений» / И. К. Марчевский, О. В. Щери-
ца; под ред. М. П. Галанина. — Москва : Издательство МГТУ
им. Н. Э. Баумана, 2017. — 59, [1] с.

ISBN

Лабораторный практикум по курсу «Методы вычислений» ориен-
тирован на изучение численных методов решения задач линейной ал-
гебры, систем нелинейных алгебраических уравнений, а также методов
интерполирования функций. Приведены примеры заданий, сформули-
рованы контрольные вопросы и требования, предъявляемые к отчетам
по лабораторным работам.

Для студентов 3-го курса, обучающихся по направлению подготов-
ки бакалавров «Прикладная математика».

УДК 519.6

ББК 32.97

© МГТУ им. Н. Э. Баумана, 2017

© Оформление. Издательство
МГТУ им. Н. Э. Баумана, 2017

ISBN ...

ПРЕДИСЛОВИЕ

Курс «Методы вычислений» занимает одно из центральных мест в цикле дисциплин, определяющих уровень профессиональной подготовки бакалавров, обучающихся по направлению «Прикладная математика». Лабораторный практикум составлен в соответствии с учебной программой дисциплины, рассчитан на один семестр и предполагает выполнение пяти лабораторных работ. Задачами лабораторного практикума являются изучение численных методов и алгоритмов, их использование для решения прикладных задач и получение навыков вычислительного программирования.

Для успешного выполнения заданий необходимо знание следующих дисциплин, включенных в учебный план специальности «Прикладная математика»: математический анализ, линейная алгебра, дифференциальные уравнения, функциональный анализ и интегральные уравнения, информатика, практикум по математическим пакетам.

Выполнение лабораторных работ включает написание программ, реализующих соответствующие численные методы, и подготовку отчета. Содержание отчетов приведено в заданиях к лабораторным работам.

На защите лабораторной работы студент должен продемонстрировать знание теории численных методов в необходимом объеме и умение применять ее на практике. ***Способ проведения защиты определяется преподавателем.*** При подготовке к защите рекомендуется использовать контрольные вопросы, приведенные в конце каждого задания.

Объем лабораторных работ достаточно велик, поэтому рекомендуется выполнять их последовательно в течение всего семестра в соответствии с материалом, изучаемым на лекциях и практических занятиях. Рекомендуемый график защиты лабораторных работ приведен в таблице.

| № п/п | Тема лабораторной работы | Неделя защиты |
|----------|---------------------------------------|------------------|
| 1 | Прямые методы решения СЛАУ | 4 |
| 2 | Итерационные методы решения СЛАУ | 7 |
| 3 | Решение задач интерполирования | 9 |
| 4 | Решение проблемы собственных значений | 12 |
| 5 | Решение нелинейных уравнений | 15 |

Написание программ допускается на любом языке программирования (C/C++, Pascal/Delphi, Fortran и др.). Во всех случаях предполагается, что для хранения в памяти компьютера чисел с плавающей точкой следует использовать тип данных, обеспечивающий **повышенную точность**. В языках программирования C/C++ и Pascal/Delphi это соответствует типу `double`.

Вычислительные процедуры, реализованные в ходе выполнения некоторых лабораторных работ, могут быть использованы при выполнении других лабораторных работ этого курса, а также при освоении последующих дисциплин учебного плана. К таким процедурам относится, к примеру, решение систем линейных алгебраических уравнений и вычисление собственных векторов матрицы. Поэтому рекомендуется их реализовывать максимально гибко, имея в виду их возможное использование в будущем.

Авторы выражают искреннюю благодарность за полезное обсуждение текста пособия и помощь в подготовке заданий для лабораторных работ своим коллегам, а также студентам кафедры «Прикладная математика» МГТУ им. Н.Э. Баумана.

Лабораторная работа № 1

Прямые методы решения систем линейных алгебраических уравнений

Цель работы

Изучение прямых методов решения систем линейных алгебраических уравнений (СЛАУ):

- а) метода Гаусса и его модификаций;
- б) метода QR -разложения.

Исследование устойчивости решения СЛАУ к погрешностям исходных данных, изучение методов оценки числа обусловленности матрицы.

Содержание работы

Рассмотрим систему линейных алгебраических уравнений вида

$$Ax = b, \quad (1.1)$$

где A — матрица системы размера $n \times n$, $\det A \neq 0$, b — вектор правой части.

Требуется выполнить следующие действия.

1. Написать программу решения СЛАУ методом Гаусса (с полным или частичным выбором главного элемента) и методом QR -разложения. Предусмотреть вывод на экран матриц Q и R .
2. В процедурах, реализующих метод Гаусса и QR -алгоритм, предусмотреть проверку существования единственного решения.
3. Подставить полученное решение в исходную систему и вычислить норму вектора невязки $\|b - b_1\|$, где b_1 — вектор правой части, полученный при подстановке решения. Провести расчеты с обычной и повышенной точностью¹ для

¹ См. замечание об используемых типах данных во Введении.

- тестовых задач и двух систем своего варианта. Сравнить результаты, полученные при использовании различных типов данных, и оценить достигнутую точность решения.
4. Написать программу для вычисления числа обусловленности матрицы системы, а также его оценки путем решения системы с возмущенной правой частью.
 5. Внести возмущение в систему, изменив произвольно элементы вектора правых частей на ± 0.01 . Найти решение возмущенной системы и сравнить его с ранее полученным. Сделать вывод об устойчивости решения к погрешностям исходных данных. Расчет провести для обеих систем своего варианта.
 6. При помощи полученных решений систем с возмущенной правой частью дать оценку снизу для числа обусловленности матрицы системы.
 7. Вычислить число обусловленности матрицы. Для вычисления обратной матрицы A^{-1} можно решить матричное уравнение $AX = E$ путем решения n систем $Ax_i = e_i$, где x_i — i -й столбец искомой матрицы $X = A^{-1}$; e_i — i -й столбец единичной матрицы E . Предусмотреть возможность использования в расчетах норм векторов и матриц $\|\cdot\|_1$ и $\|\cdot\|_\infty$.
 8. Провести расчеты с обычной и повышенной точностью для тестовой задачи и двух систем своего варианта. Предусмотреть вывод на экран либо в файл результата умножения матриц $A^{-1}A$.

Методические указания

Общие сведения о прямых методах решения СЛАУ

Прямыми называют методы решения СЛАУ, которые позволяют получить точное решение за конечное число действий при условии, что все арифметические действия выполняются точно (без погрешностей).

В данной лабораторной работе рассмотрим два прямых метода: метод Гаусса и метод QR -разложения.

В координатной форме систему уравнений (1.1) можно записать следующим образом:

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, 2, \dots, n. \quad (1.2)$$

Метод Гаусса², или *метод последовательного исключения неизвестных*, состоит в том, что неизвестные x_j , $j = 1, 2, \dots, n-1$, последовательно исключаются из системы (1.2); в результате она преобразуется к эквивалентной системе с треугольной матрицей:

[illegible]

Коэффициенты $a_{ij}^{(k)}$ и компоненты правой части $b_i^{(k)}$ системы (1.3) вычисляются по формулам

$$\begin{aligned} a_{ij}^{(k)} &= a_{ij}^{(k-1)} - c_{ik} a_{kj}^{(k-1)}, \\ b_i^{(k)} &= b_i^{(k-1)} - c_{ik} b_k^{(k-1)}, \end{aligned} \quad (1.4)$$

где

$$c_{ik} = a_{ik}^{(k-1)} / a_{kk}^{(k-1)},$$

$$k = 1, 2, \dots, (n-1), \quad j = k, k+1, \dots, n, \quad i = k+1, k+2, \dots, n,$$

причем

$$a_{ij}^{(0)} = a_{ij}, \quad b_i^{(0)} = b_i.$$

² Карл Фридрих Гаусс (1777–1855) — выдающийся немецкий математик, астроном и физик, один из величайших математиков в истории.

Вычисления по формулам (1.4) называются **прямым ходом метода Гаусса**. Затем неизвестные x_i последовательно, начиная с x_n , определяются из системы (1.3) по формулам

$$x_i = \left(b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)} x_j \right) / a_{ii}^{(i-1)}, \quad i = n, n-1, \dots, 1. \quad (1.5)$$

Вычисления по этим формулам называются **обратным ходом метода Гаусса**.

Для реализации прямого хода метода Гаусса требуется порядка $O(n^3/3)$ операций умножения и деления чисел с плавающей точкой, для обратного — порядка $O(n^2/2)$ операций.

Выбор главного элемента

Метод Гаусса применим тогда и только тогда, когда все угловые миноры матрицы A ненулевые, что равносильно требованию $a_{ii}^{(i-1)} \neq 0$ для всех значений $i = 1, 2, \dots, n$. Но из условия невырожденности матрицы ($\det A \neq 0$) не следует, что в ходе приведения матрицы A к треугольному виду (1.3) на диагонали не возникнет элементов, равных нулю или малых по абсолютной величине (поскольку это приводит к дополнительным ошибкам округления в вычислениях). В таких случаях метод Гаусса неприменим, поэтому на практике обычно используется вариант алгоритма Гаусса с частичным либо полным выбором главного элемента. Основная идея метода состоит в том, чтобы на очередном шаге исключать не следующее по номеру неизвестное, а то неизвестное, коэффициент при котором является наибольшим по модулю. Этот коэффициент называется **ведущим (главным)** элементом. Выбор ненулевого коэффициента достаточен для исключения неизвестного, а выбор наибольшего элемента диктуется соображениями устойчивости вычислений.

При **частичном выборе** поиск главного элемента ведется только по столбцу (или только по строке). При исключении k -й переменной сначала осуществляется поиск наибольшего по модулю коэффициента в k -м столбце среди элементов $a_{ik}^{(k-1)}$, $i = k, k+1, \dots, n$, лежащих не выше главной диагонали. Пусть

искомый элемент находится в строке i^* , тогда k -я и i^* -я строки меняются местами и далее производится исключение переменной по стандартной процедуре. При переходе к следующему шагу процедура поиска главного элемента осуществляется снова.

Вариант с **полным выбором** главного элемента отличается тем, что на k -м шаге исключения переменной поиск главного элемента осуществляется среди элементов $a_{ij}^{(k-1)}$, $i, j = k, k+1, \dots, n$. Затем происходит перестановка столбцов и/или строк матрицы так, чтобы главный элемент матрицы оказался в позиции (k, k) . При этом требуется учитывать изменение нумерации неизвестных x_j , $j = k, k+1, \dots, n$.

Метод QR -разложения

В основе многих методов решения СЛАУ лежит факторизация матрицы исходной системы уравнений, то есть ее представление в виде произведения матриц, удобных для обращения. Таковыми являются ортогональные, треугольные, диагональные и некоторые другие типы матриц. Заметим, что метод Гаусса эквивалентен LU -разложению матрицы системы, где L — нижнетреугольная, а U — верхнетреугольная матрица.

Рассмотрим метод QR -разложения, основанный на представлении матрицы системы в виде произведения ортогональной матрицы Q и верхней треугольной матрицы R . Один из способов получения такого разложения — **метод вращений**.

На первом этапе этого метода неизвестное x_1 исключается из всех уравнений, кроме первого. Это производится с помощью следующего алгоритма. Для исключения x_1 из второго уравнения вычисляются коэффициенты

$$c_{12} = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}}, \quad s_{12} = \frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}},$$

затем первое уравнение системы заменяется линейной комбинацией первого и второго уравнений с коэффициентами c_{12} и s_{12} , а второе уравнение — линейной комбинацией тех же уравнений, но уже с коэффициентами $(-s_{12})$ и c_{12} . Так как $-s_{12}a_{11} + c_{12}a_{21} = 0$, коэффициент во втором уравнении при x_1 обратится в нуль.

$$T_{13} = \begin{pmatrix} c_{13} & 0 & s_{13} & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ -s_{13} & 0 & c_{13} & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Аналогичным образом неизвестная x_1 исключается из остальных уравнений, затем переменная x_2 — из всех уравнений, кроме первого и второго, при этом используются матрицы $T_{23}, T_{24}, \dots, T_{2n}$ и так далее. Процесс продолжается, пока система не будет приведена к верхней треугольной форме.

В матричном виде все эти операции можно записать так:

$$T = T_{n-1,n} \cdot \dots \cdot T_{24} \cdot T_{23} \cdot T_{1n} \cdot \dots \cdot T_{13} \cdot T_{12},$$

$$R = TA, \quad b^* = Tb,$$

где T_{ij} — матрицы поворота, T — матрица результирующего вращения (она также будет ортогональной как произведение ортогональных матриц), R — получающаяся в итоге верхняя треугольная матрица. В результате получается QR -разложение матрицы A , где $Q = T^{-1} = T^T$.

Если известно QR -разложение матрицы A , то решение системы (1.1) сводится к решению более простых систем уравнений:

- 1) решение системы $Qb^* = b$;
- 2) решение системы $Rx = b^*$.

Следует отметить, что метод вращений требует выполнения значительно большего числа операций, чем метод Гаусса.

Число обусловленности

Будем считать, что система уравнений (1.1) имеет единственное решение. На практике исходные данные — коэффициенты матрицы A и элементы вектора правой части b — обычно заданы лишь приближенно с некоторой точностью. Это чаще всего

связано с тем, что исходные данные получаются на основе результатов измерений, которые невозможно произвести со сколь угодно высокой точностью. Другой источник погрешности связан с представлением вещественных чисел в ЭВМ, которое в общем случае не является точным. Эта погрешность обычно на несколько порядков меньше, чем погрешность исходных данных, однако ее наличие может оказать существенное влияние на получаемое решение системы, поэтому пренебрегать ею нельзя.

Обусловленность вычислительной задачи — это чувствительность ее решения к малым погрешностям входных данных. Задачу называют **хорошо обусловленной**, если малым погрешностям входных данных отвечают малые погрешности решения, и **плохо обусловленной**, если возможны сильные изменения решения при малых возмущениях входных данных.

В данной лабораторной работе исследуем обусловленность задачи решения СЛАУ, считая, что ее матрица задана точно, т. е. оценим влияние погрешностей правой части на ее решение.

Пусть Δb — погрешность, с которой задана правая часть системы (1.1). Тогда в результате решения системы

$$A(x + \Delta x) = b + \Delta b \quad (1.6)$$

решение исходной задачи (1.1) будет найдено с погрешностью Δx , удовлетворяющей условию

$$A\Delta x = \Delta b, \quad \text{или} \quad \Delta x = A^{-1}\Delta b.$$

Отсюда получается оценка абсолютной погрешности решения

$$\|\Delta x\| \leq \|A^{-1}\| \cdot \|\Delta b\|.$$

Однако такая оценка может оказаться недостаточно информативной, поскольку на практике важно знать, как связаны относительные погрешности решения и правой части. Учитывая неравенство

$$\|b\| \leq \|A\| \cdot \|x\|,$$

получаем

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\Delta b\|}{\|b\|}. \quad (1.7)$$

Величину

$$\text{cond} A = \|A^{-1}\| \cdot \|A\| \quad (1.8)$$

называют **числом обусловленности** матрицы A . Заметим, что числа обусловленности матриц A и A^{-1} совпадают.

Матрица называется **плохо обусловленной** когда ее число обусловленности велико, и **хорошо обусловленной**, если ее число обусловленности достаточно мало.

Важно отметить, что умножение матрицы A на произвольную константу $\alpha \neq 0$ не приведет к изменению ее числа обусловленности, т. к. в этом случае обратная матрица окажется умноженной на величину α^{-1} .

Оценка числа обусловленности

Из формулы (1.8) видно, что величина числа обусловленности матрицы зависит от выбора нормы. Тем не менее, можно показать, что во всех случаях

$$\text{cond} A \geq 1.$$

Если известны собственные значения матрицы A , то для числа обусловленности справедлива оценка

$$\text{cond} A \geq \frac{|\lambda_{\max}|}{|\lambda_{\min}|},$$

где λ_{\max} и λ_{\min} — соответственно максимальное и минимальное по модулю собственные значения.

Если матрица $A = A_1 \cdot A_2$, то ее число обусловленности не превышает произведения чисел обусловленности матриц-множителей:

$$\text{cond} A \leq \text{cond} A_1 \cdot \text{cond} A_2.$$

Из соотношения (1.7) следует неравенство

$$\text{cond} A \geq \frac{\delta x}{\delta b},$$

где

$$\delta x = \frac{\|\Delta x\|}{\|x\|}, \quad \delta b = \frac{\|\Delta b\|}{\|b\|}$$

— относительные погрешности решения и правой части.

Для получения нижней оценки числа обусловленности можно несколько раз решить систему (1.6) с разными Δb и выбрать наибольшую величину отношения относительных погрешностей решения и правой части.

Содержание отчета

1. Номер варианта и исходные данные.
2. Краткое описание используемых алгоритмов.
3. Результаты расчетов (численное решение, невязка и погрешность решения, точное значение числа обусловленности, его оценка, номер компоненты вектора правой части, наиболее сильно влияющей на решение).
4. Анализ результатов (сравнение решений, полученных с различной точностью, оценка погрешности решения, оценка влияния числа обусловленности матрицы системы на результаты ее решения численными методами с различной точностью, сравнение полученной оценки с точным значением числа обусловленности, оценка числа обусловленности с использованием норм $\|\cdot\|_1$ и $\|\cdot\|_\infty$).

Контрольные вопросы

1. Каковы условия применимости метода Гаусса без выбора и с выбором ведущего элемента?
2. Докажите, что если $\det A \neq 0$, то при выборе главного элемента в столбце среди элементов, лежащих не выше главной диагонали, всегда найдется хотя бы один элемент, отличный от нуля.
3. В методе Гаусса с полным выбором ведущего элемента приходится не только переставлять уравнения, но и менять нумерацию неизвестных. Предложите алгоритм, позволяющий восстановить первоначальный порядок неизвестных.

4. Оцените количество арифметических операций, требуемых для QR -разложения произвольной матрицы A размера $n \times n$.
5. Что такое число обусловленности и что оно характеризует? Имеется ли связь между обусловленностью и величиной определителя матрицы? Как влияет выбор нормы матрицы на оценку числа обусловленности?
6. Как упрощается оценка числа обусловленности, если матрица является:
 - а) диагональной;
 - б) симметричной;
 - в) ортогональной;
 - г) положительно определенной;
 - д) треугольной?
- 7*. Применимо ли понятие числа обусловленности к вырожденным матрицам?
- 8*. В каких случаях целесообразно использовать метод Гаусса, а в каких — методы, основанные на факторизации матрицы?
- 9*. Как можно объединить в одну процедуру прямой и обратный ход метода Гаусса? В чем достоинства и недостатки такого подхода?
- 10*. Объясните, почему, говоря о векторах, норму $\|\cdot\|_1$ часто называют октаэдрической, норму $\|\cdot\|_2$ — шаровой, а норму $\|\cdot\|_\infty$ — кубической.

Примеры заданий

Системы уравнений для тестирования программы

1. Тестовый пример «Тест 1»:

$$\left\{ \begin{array}{l} x_1 + x_2 + x_3 + x_4 = 4, \\ x_2 + x_3 + x_4 = 3, \\ x_3 + x_4 = 2, \\ x_4 = 1. \end{array} \right. \quad \left| \quad \begin{array}{l} \text{Точное решение} \\ x = (1, 1, 1, 1)^T, \\ \text{cond}_1 A = 8, \\ \text{cond}_\infty A = 8. \end{array} \right.$$

2. Тестовый пример «Тест 2»:

$$\left\{ \begin{array}{l} x_4 = 1, \\ x_3 + x_4 = 2, \\ x_2 + x_3 + x_4 = 3, \\ x_1 + x_2 + x_3 + x_4 = 4. \end{array} \right. \quad \left| \quad \begin{array}{l} \text{Точное решение} \\ x = (1, 1, 1, 1)^T, \\ \text{cond}_1 A = 8, \\ \text{cond}_\infty A = 8. \end{array} \right.$$

3. Тестовый пример «Тест 3»:

$$\left\{ \begin{array}{l} x_1 + x_2 + x_3 + x_4 = 4, \\ 2x_1 + 3x_2 + 3x_3 + 3x_4 = 11, \\ 2x_1 + 4x_2 + 4x_3 + 4x_4 = 15, \\ 4x_1 + 5x_2 + 6x_3 + 7x_4 = 22. \end{array} \right. \quad \left| \quad \begin{array}{l} \text{Матрица системы} \\ \text{вырожденная, система} \\ \text{несовместна, } \text{cond} A = \infty. \end{array} \right.$$

4. Тестовый пример «Тест 4»:

$$\left\{ \begin{array}{l} 10x_1 + 6x_2 + 2x_3 = 25, \\ 5x_1 + x_2 - 2x_3 + 4x_4 = 14, \\ 3x_1 + 5x_2 + x_3 - x_4 = 10, \\ 6x_2 - 2x_3 + 2x_4 = 8. \end{array} \right. \quad \left| \quad \begin{array}{l} \text{Точное решение} \\ x = (2, 1, -0.5, 0.5)^T, \\ \text{cond}_1 A \approx 240.55, \\ \text{cond}_\infty A \approx 269.18. \end{array} \right.$$

5. Тестовый пример «Тест 5»:

$$\left\{ \begin{array}{l} 28.859x_1 - 0.008x_2 + 2.406x_3 + 19.240x_4 = 30.459, \\ 14.436x_1 - 0.001x_2 + 1.203x_3 + 9.624x_4 = 18.248, \\ 120.204x_1 - 0.032x_2 + 10.024x_3 + 80.144x_4 = 128.156, \\ -57.714x_1 + 0.016x_2 - 4.812x_3 - 38.478x_4 = -60.908. \end{array} \right.$$

Точное решение $x = (1, 1000, -20, 3)^T$,
 $\text{cond}_1 A \approx 122414849.9$, $\text{cond}_\infty A \approx 109686235.3$.

Лабораторная работа № 2

Итерационные методы решения систем линейных алгебраических уравнений

Цель работы

Изучение итерационных методов решения систем линейных алгебраических уравнений (СЛАУ):

- а) метода простой итерации и метода Якоби;
- б) метода Зейделя и метода релаксации.

Содержание работы

Рассмотрим систему линейных алгебраических уравнений (СЛАУ) вида

$$Ax = b, \quad (2.1)$$

где A — матрица системы размера $n \times n$, $\det A \neq 0$, b — вектор правой части. Требуется выполнить следующие действия.

1. Написать программу решения СЛАУ методами простой итерации, Якоби, Зейделя и релаксации.
2. Написать программу решения СЛАУ с трехдиагональной матрицей большой размерности методами Зейделя и релаксации.
3. Преобразовать систему (2.1) к виду, удобному для итераций методами простой итерации, Якоби, Зейделя и релаксации

$$x^{k+1} = Cx^k + y,$$

выписав для каждого метода вид матрицы C и вектора y .

4. Для каждого из методов вычислить $\|C\|$ (использовать нормы $\|\cdot\|_1$ и $\|\cdot\|_\infty$). Проверить выполнение условия

$$\|C\| < 1.$$

5. Для методов Зейделя и релаксации представить матрицу C в виде

$$C = C_L + C_D + C_U,$$

где C_L — нижняя треугольная, C_U — верхняя треугольная, C_D — диагональная матрицы. Вычислить нормы матриц C_L и C_U и их сумму; если сумма окажется меньше единицы, оценить скорость сходимости метода.

6. Найти решение системы (2.1) рассматриваемыми методами с погрешностью, не превышающей $\varepsilon = 10^{-4}$. Вычислить нормы векторов невязок. Повторить расчеты для $\varepsilon = 10^{-7}$. Предварительно оценить необходимое число итераций и сравнить с результатами расчетов.
7. Для метода простой итерации экспериментально подобрать значения итерационного параметра τ , при которых итерационный процесс сходится. Исследовать влияние итерационного параметра τ на скорость сходимости алгоритма.
8. Для метода релаксации исследовать влияние параметра релаксации ω на скорость сходимости алгоритма. Провести расчеты для нескольких значений параметра.
9. Экспериментально исследовать скорость сходимости при различных начальных приближениях.

Методические указания

Общие сведения об одношаговых итерационных методах

Канонической формой одношагового итерационного метода решения системы (2.1) называется его запись в виде:

$$B_{k+1} \frac{x^{k+1} - x^k}{\tau_{k+1}} + Ax^k = b, \quad k = 0, 1, 2, \dots \quad (2.2)$$

Здесь k — номер итерации; B_{k+1} — матрицы, задающие тот или иной итерационный алгоритм; τ_{k+1} — итерационный параметр. Предполагается, что задано начальное приближение x^0 и существуют обратные матрицы B_k^{-1} , $k = 1, 2, \dots$

Итерационный метод решения СЛАУ заключается в построении по начальному приближению x^0 последовательности приближений $\{x^k\}_{k=1}^{\infty}$. Из (2.2) следует, что если последовательность $\{x^k\}$ сходится, то ее пределом является вектор x — решение системы (2.1). Для нахождения x^{k+1} по известным B_{k+1} , τ_{k+1} , b и x^k достаточно решить систему уравнений

$$B_{k+1}x^{k+1} = (B_{k+1} - \tau_{k+1}A)x^k + \tau_{k+1}b.$$

Итерационный метод называется **явным**, если $B_k = E$, и **неявным**, если $B_k \neq E$ (здесь E — единичная матрица). Если $B_k = B$ и $\tau_{k+1} = \tau$, т. е. матрица и итерационный параметр не зависят от номера итерации, то итерационный метод (2.2) называется **стационарным**, иначе — **нестационарным**.

Преобразуем систему (2.1) к виду

$$x = Cx + y, \quad (2.3)$$

где C — квадратная матрица размера $n \times n$, y — вектор-столбец. Это можно сделать различными способами. Формула (2.3) подсказывает рекуррентное соотношение вида

$$x^{k+1} = Cx^k + y, \quad k = 0, 1, 2, \dots \quad (2.4)$$

Если последовательность $\{x^k\}$ сходится, то ее пределом является решение системы (2.3), которая эквивалентна исходной системе (2.1). Несложно проверить, что если $\|C\| < 1$, то последовательность $\{x^k\}$ сходится при любом начальном приближении x^0 , при этом справедлива оценка

$$\|x - x^k\| \leq \|C\|^k \|x - x^0\|.$$

В данной лабораторной работе рассмотрим несколько стационарных итерационных методов решения систем линейных алгебраических уравнений: метод простой итерации, метод Якоби, метод Зейделя и метод релаксации.

Метод простой итерации

Методом простой итерации называется явный метод

$$\frac{x^{k+1} - x^k}{\tau} + Ax^k = b, \quad k = 0, 1, 2, \dots \quad (2.5)$$

Здесь τ — специально выбираемый **итерационный параметр**.

Один из способов приведения системы к виду (2.3), удобному для итераций, и построения итерационного процесса в виде (2.4) состоит в следующем. Прибавим слева и справа к исходной системе уравнений (2.1) вектор x :

$$Ax + x = b + x.$$

Это равенство можно переписать в виде

$$x = -(A - E)x + b.$$

Сравнивая это уравнение с (2.3), получаем, что

$$C = -(A - E), \quad y = b.$$

Достаточным условием сходимости итерационного процесса (2.4) является выполнение условия

$$\|C\| < 1.$$

Обычно для улучшения скорости сходимости исходную систему, прежде чем приводить к виду, удобному для итераций, умножают на итерационный параметр τ , который выбирают так, чтобы выполнялась оценка $\|C\| < 1$ и норма матрицы C была как можно меньше. Тогда

$$x + \tau Ax = x + \tau b,$$

отсюда

$$x = -(\tau A - E)x + \tau b$$

и из сравнения с (2.3) получаем, что для метода простой итерации матрица C и вектор y выражаются в виде

$$C = -(\tau A - E), \quad y = \tau b.$$

Метод Якоби

Если представить матрицу A в виде суммы

$$A = L + D + U,$$

где L — нижняя треугольная, U — верхняя треугольная, а $D = \text{diag}\{a_{11}, a_{22}, \dots, a_{nn}\}$ — диагональная матрицы, то **метод Якоби**¹ можно записать в виде

$$D(x^{k+1} - x^k) + Ax^k = b, \quad k = 0, 1, 2, \dots \quad (2.6)$$

Для метода Якоби существует весьма простой способ приведения системы к виду, удобному для итераций, и построения матрицы C .

Из первого уравнения (2.1) выразим переменную x_1 :

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n)/a_{11}.$$

Из второго уравнения — переменную x_2 :

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n)/a_{22}$$

и т.д. В результате получим систему вида

$$x = Cx + y,$$

или в покомпонентной записи

$$\begin{aligned} x_1 &= c_{12}x_2 + c_{13}x_3 + \dots + c_{1n}x_n + y_1, \\ x_2 &= c_{21}x_1 + c_{23}x_3 + \dots + c_{2n}x_n + y_2, \\ &\dots \\ x_n &= c_{n1}x_1 + c_{n2}x_2 + c_{n3}x_3 + \dots + y_n. \end{aligned}$$

¹ Карл Густав Якоб Якоби (1804–1851) — немецкий математик, внес огромный вклад в линейную алгебру, комплексный анализ, динамику и другие разделы математики и механики.

На главной диагонали матрицы C находятся нулевые элементы, остальные коэффициенты матрицы и компоненты вектора y вычисляются по формулам

$$c_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad y_i = \frac{b_i}{a_{ii}}, \quad i, j = 1, 2, \dots, n, \quad i \neq j. \quad (2.7)$$

Такой выбор матрицы C и вектора y соответствует методу Якоби (2.6). Несложно убедиться, что в данном случае

$$C = -D^{-1}(L + U), \quad y = D^{-1}b. \quad (2.8)$$

Следует отметить, что в методе Якоби для построения матрицы C , которая в дальнейшем используется в итерационном процессе (2.4), нет необходимости в явном виде перемножать матрицы, как это следует из (2.8). Коэффициенты этой матрицы намного проще вычислить с использованием формул (2.7). Это же замечание относится к процедуре вычисления вектора y .

Методы Зейделя и релаксации

Каноническая форма **метода Зейделя**² имеет вид

$$(D + L)(x^{k+1} - x^k) + Ax^k = b, \quad k = 0, 1, 2, \dots \quad (2.9)$$

Метод Зейделя является частным случаем **метода релаксации** (или **методом верхней релаксации**), задаваемого в виде

$$(D + \omega L) \frac{x^{k+1} - x^k}{\omega} + Ax^k = b, \quad k = 0, 1, 2, \dots \quad (2.10)$$

Здесь $\omega > 0$ — заданный числовой параметр — **параметр релаксации**. В случае симметричной положительно определенной матрицы A метод релаксации сходится при $0 < \omega < 2$.

Чтобы привести систему (2.10) к виду, удобному для итераций, перепишем ее в виде

$$(E + \omega D^{-1}L)x^{k+1} = ((1 - \omega)E - \omega D^{-1}U)x^k + \omega D^{-1}b.$$

В покомпонентной записи

$$x_i^{k+1} + \omega \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} = (1 - \omega)x_i^k - \omega \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^k + \omega \frac{b_i}{a_{ii}}, \quad i = 1, 2, \dots, n.$$

² Филипп Людвиг Зейдель (1821–1896) — немецкий математик и астроном.

Отсюда можно последовательно найти все x_i^{k+1} , $i = 1, 2, \dots, n$:

$$\begin{aligned}
 x_1^{k+1} &= (1 - \omega)x_1^k - \omega \sum_{j=2}^n \frac{a_{1j}}{a_{11}} x_j^k + \omega \frac{b_1}{a_{11}}, \\
 x_2^{k+1} &= -\omega \frac{a_{21}}{a_{22}} x_1^{k+1} + (1 - \omega)x_2^k - \omega \sum_{j=3}^n \frac{a_{2j}}{a_{22}} x_j^k + \omega \frac{b_2}{a_{22}}, \\
 &\dots \\
 x_i^{k+1} &= -\omega \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} + (1 - \omega)x_i^k - \omega \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^k + \omega \frac{b_i}{a_{ii}}, \\
 &\dots \\
 x_n^{k+1} &= -\omega \sum_{j=1}^{n-1} \frac{a_{nj}}{a_{nn}} x_j^{k+1} + (1 - \omega)x_n^k + \omega \frac{b_n}{a_{nn}}.
 \end{aligned} \tag{2.11}$$

($i = 3, 4, \dots, n - 1$),

Метод Зейделя и метод верхней релаксации, как и метод Якоби, являются неявными, так как в канонической форме записи (2.2) матрица B не является единичной. Однако при определении x^{k+1} матрицу B^{-1} вычислять не нужно. Для проведения расчетов также нет необходимости находить в явном виде матрицу C и вектор y , входящие в формулу (2.4), поскольку каждое последующее приближение получается из предыдущего по формулам (2.11). Однако в лабораторной работе для оценки скорости сходимости в программе необходимо предусмотреть процедуру вычисления матрицы C .

Критерий останова итерационного процесса

Критерием прекращения итераций, обеспечивающим достижение заданной точности ε , является выполнение неравенства

$$\|x^{k+1} - x^k\| \leq \frac{1 - \|C\|}{\|C\|} \varepsilon.$$

Для метода Зейделя существует также другой критерий:

$$\|x^{k+1} - x^k\| \leq \frac{1 - \|C\|}{\|C_U\|} \varepsilon.$$

Замечание. В вариантах заданий для методов Зейделя и релаксации матрица большой размерности является трехдиагональной, поэтому при программной реализации в памяти необходимо хранить не всю систему уравнений, а всего лишь четыре вектора: три диагонали и вектор правой части.

Содержание отчета

1. Номер варианта и исходные данные.
2. Краткое описание используемых алгоритмов с указанием используемых критериев окончания счета.
3. Результаты расчетов, включающие в себя значения норм соответствующих матриц (C , C_L , C_U), оценки скоростей сходимости алгоритмов и последовательность приближений к решению (или выборку из нее).
4. Анализ результатов: сравнение решений, полученных с различной точностью, достигнутая точность (количество верных цифр в ответе), сравнение теоретической оценки и реально выполненного количества итераций. Сравнить априорную и апостериорную погрешность решения.

Контрольные вопросы

1. Почему условие $\|C\| < 1$ гарантирует сходимость итерационных методов?
2. Каким следует выбирать итерационный параметр τ в методе простой итерации для увеличения скорости сходимости? Как выбрать начальное приближение x^0 ?
3. На примере системы из двух уравнений с двумя неизвестными дайте геометрическую интерпретацию метода Якоби, метода Зейделя, метода релаксации.
4. При каких условиях сходятся метод простой итерации, метод Якоби, метод Зейделя и метод релаксации? Какую матрицу называют положительно определенной?
5. Выпишите матрицу C для методов Зейделя и релаксации.
6. Почему в общем случае для остановки итерационного процесса нельзя использовать критерий $\|x^k - x^{k-1}\| < \varepsilon$?
- 7*. Какие еще критерии окончания итерационного процесса Вы можете предложить?

Примеры заданий

Системы линейных уравнений для решения методом простой итерации и методом Якоби

1. Тестовый пример «Тест 1»:

$$\begin{cases} 15x_1 + 2x_2 - 3x_3 + 7x_4 = 53, \\ -5x_1 + 11x_2 + 2x_3 - 3x_4 = -90, \\ \quad \quad \quad -x_2 + 7x_3 + 4x_4 = 107, \\ 12x_1 \quad \quad \quad - 6x_3 + 20x_4 = 68. \end{cases}$$

Точное решение $x = (5, -7, 12, 4)^T$.

2. Тестовый пример «Тест 2»:

$$\begin{cases} 86.00x_1 - 8.93x_2 - 9.59x_3 - 3.91x_4 = 818.58, \\ 4.05x_1 - 100.00x_2 - 9.10x_3 - 8.14x_4 = 898.74, \\ 0.26x_1 + 3.61x_2 - 71.80x_3 - 4.28x_4 = -912.22, \\ -4.03x_1 - 6.88x_2 + 6.57x_3 - 198.60x_4 = -687.06. \end{cases}$$

Точное решение $x = (10, -10, 12, 4)^T$.

Система линейных уравнений большой размерности для решения методом Зейделя и методом релаксации

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & \dots & 0 \\ 0 & a_3 & b_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & a_n & b_n \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \dots \\ d_n \end{pmatrix},$$

где $n = 200 + N$ (номер варианта); $a_i = c_i = 1$; $b_i = 4$; $i = 1, 2, \dots, n$; $d_1 = 6$; $d_i = 10 - 2 \cdot (i \bmod 2)$, $i = 2, 3, \dots, n-1$; $d_n = 9 - 3(n \bmod 2)$.

Точное решение $x_i = 2 - (i \bmod 2)$, $i = 1, 2, \dots, n$.

Лабораторная работа № 3

Решение задач интерполирования

Цель работы

Изучение и сравнение различных методов интерполирования:

- а) многочленами Лагранжа;
- б) кубическими сплайнами.

Содержание работы

Рассмотрим непрерывную на отрезке $[a, b]$ функцию $y(x)$. Требуется выполнить следующие действия.

1. Написать программу, позволяющую проводить интерполирование таблично заданной функции. Предусмотреть использование как равномерной на отрезке $[a, b]$, так и чебышевской сетки с заданным количеством узлов. Для каждой сетки необходимо построить таблицу значений $(x_i, y(x_i))$, по которой будет проводиться интерполирование.
2. Осуществить интерполирование исходной функции с помощью полиномов Лагранжа на равномерной и чебышевской сетках с различным числом узлов. Сделать вывод об устойчивости процесса интерполирования полиномами Лагранжа.
3. Повторить те же действия для интерполирования сплайнами.
4. Построить графики полученных интерполянтов и сравнить их с графиком исходной функции $y(x)$.
5. Определить для интерполяции полиномом Лагранжа необходимое количество узлов чебышевской сетки, которое обеспечивает восстановление исходной функции $y(x)$ с погрешностью, не превышающей $\varepsilon = 0.01$, $\varepsilon = 0.0001$.

Методические указания

Общие сведения об интерполировании функций

Простейшая задача **интерполирования** заключается в следующем. На отрезке $[a, b]$ заданы точки x_0, x_1, \dots, x_n , которые называются **узлами интерполяции**, и значения некоторой функции $y(x)$ в этих точках:

$$y(x_0) = y_0, \quad y(x_1) = y_1, \quad \dots \quad y(x_n) = y_n.$$

Множество $\{y_i\}_{i=0}^n$ называется **сеточной функцией**, заданной на сетке $\{x_i\}_{i=0}^n$. Требуется построить **интерполирующую функцию** $f(x)$, принадлежащую определенному классу и принимающую в узлах интерполяции те же значения, что и $y(x)$, т. е.

$$f(x_i) = y_i, \quad i = 0, 1, \dots, n.$$

Геометрически это означает, что нужно найти кривую, задаваемую функцией $f(x)$ из некоторого класса, проходящую через заданную систему точек (x_i, y_i) , $i = 0, 1, \dots, n$. В такой общей постановке задача может иметь бесконечно много решений или совсем не иметь решений.

В данной лабораторной работе рассматривается интерполяция функций на равномерных и чебышевских¹ сетках при помощи многочленов (полиномов) Лагранжа и кубических сплайнов.

Многочлены Лагранжа

Пусть задана таблица значений интерполируемой функции $y(x)$ в узлах сетки, т. е. каждой точке x_i поставлено в соответствие значение функции в этой точке $y_i = y(x_i)$, $i = 0, 1, \dots, n$.

Многочлен n -ой степени

$$L_n(x) = \sum_{k=0}^n \alpha_k x^k$$

¹ Пафнутий Львович Чебышев (произносится «Чебышёв») (1821–1894) — русский математик и механик; один из основоположников теории приближения функций. Внес большой вклад в развитие теории чисел, теории вероятностей, механики.

называется **интерполяционным многочленом** функции $y(x)$, если он удовлетворяет условиям

$$L_n(x_i) = y_i, \quad i = 0, 1, \dots, n. \quad (3.1)$$

Интерполяционный полином Лагранжа² — это представление многочлена $L_n(x)$ в виде линейной комбинации значений функции $y(x)$ в узлах сетки:

$$L_n(x) = \sum_{k=0}^n c_k(x) y_k.$$

Из условий интерполирования (3.1) можно получить явное выражение для $c_k(x)$:

$$\sum_{k=0}^n c_k(x_i) y_k = y_i, \quad i = 0, 1, \dots, n.$$

Данные соотношения будут выполнены, если $c_k(x)$ удовлетворяют следующим условиям:

$$c_k(x_i) = \begin{cases} 0, & i \neq k, \\ 1, & i = k, \end{cases} \quad i = 0, 1, \dots, n. \quad (3.2)$$

Эти условия означают, что каждая из функций $c_k(x)$, $k = 0, 1, \dots, n$ имеет по крайней мере n нулей на отрезке $[a, b]$. Поскольку $L_n(x)$ — многочлен степени n , функции $c_k(x)$ естественно искать также в виде многочленов. С учетом условий (3.2) получим:

$$c_k(x) = \prod_{\substack{j=0, \\ j \neq k}}^n \left(\frac{x - x_j}{x_k - x_j} \right).$$

² Жозеф Луи Лагранж (1736–1813) — французский математик, астроном и механик. Установил фундаментальный принцип возможных перемещений и завершил математизацию механики. Внес грандиозный вклад в развитие анализа, теории чисел, теорию вероятностей, создал вариационное исчисление.

Для заданного набора точек x_i и значений функции y_i в этих точках ($i = 0, 1, \dots, n$) интерполяционный многочлен степени n существует и единственен.

Если функция $y(x)$ непрерывно дифференцируема $(n+1)$ раз на отрезке $[a, b]$, содержащем все узлы интерполяции, то для любого x из отрезка $[x_0, x_n]$ справедлива следующая оценка погрешности интерполяции:

$$|y(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |\omega_{n+1}(x)|. \quad (3.3)$$

Здесь $M_{n+1} = \max_{x \in [x_0, x_n]} |y^{(n+1)}(x)|$, $\omega_{n+1}(x) = \prod_{k=0}^n (x - x_k)$.

Если узлы сетки упорядочены, т. е. $x_0 < x_1 < \dots < x_n$ и

$$h_i = x_i - x_{i-1}, \quad h_{\max} = \max_i h_i, \quad i = 1, 2, \dots, n,$$

то погрешность интерполяции можно оценить по формуле

$$\max_{x \in [x_0, x_n]} |y(x) - L_n(x)| \leq \frac{M_{n+1}}{n+1} h_{\max}^{n+1}.$$

Экстраполяция

Наряду с интерполированием применяют и **экстраполирование**, т. е. приближенное определение значений функции $y(x)$ в точках x , лежащих вне отрезка $[a, b]$, по ее значениям в точках $x_i \in [a, b]$, $i = 0, 1, \dots, n$.

Для экстраполяции функций часто пользуются интерполяционным полиномом и полагают, что

$$y(x) \approx L_n(x)$$

не только при $x \in [a, b]$, но также и при $x \notin [a, b]$. Однако погрешность экстраполяции обычно оказывается существенно большей, чем погрешность интерполяции.

Если интерполяционный полином $L_n(x)$ построен на равномерной сетке с шагом h , то справедливы оценки:

при $x \in [b, b + h]$

$$|y(x) - L_n(x)| \leq h^{n+1} \max_{\xi \in [a, x]} |y^{(n+1)}(\xi)|;$$

при $x \in [b + h, b + 2h]$

$$|y(x) - L_n(x)| \leq h^{n+1}(n+2) \max_{\xi \in [a, x]} |y^{(n+1)}(\xi)|;$$

при $x \in [b + 2h, b + 3h]$

$$|y(x) - L_n(x)| \leq \frac{1}{2} h^{n+1}(n+2)(n+3) \max_{\xi \in [a, x]} |y^{(n+1)}(\xi)|$$

и так далее. Тем самым ошибка экстраполяции существенно возрастает по мере удаления от отрезка $[a, b]$.

Равномерные и чебышевские сетки

Погрешность интерполяции можно уменьшить за счет выбора узлов сетки. Простейшей сеткой на отрезке $[a, b]$ является **равномерная сетка**. Шаг сетки h в этом случае является постоянным и вычисляется по заданному числу узлов n

$$h = \frac{b - a}{n},$$

а сами узлы имеют координаты

$$x_i = a + h \cdot i = a + \frac{b - a}{n} \cdot i, \quad i = 0, 1, \dots, n.$$

Несмотря на простоту и удобство использования, равномерная сетка часто является неудачной для интерполирования. С ростом числа точек у интерполяционной кривой могут возникнуть осцилляции. Классическим примером, показывающим возникновение осцилляций у интерполяционного многочлена, служит пример Рунге (стр. 35). Вблизи концов отрезка отклонение интерполяционного полинома от исходной функции $y(x)$ возрастает с ростом числа узлов интерполяции. Значения интерполяционного многочлена даже для гладких функций в промежуточных точках могут сильно уклоняться от значений самой функции.

За счет выбора узлов сетки $\{x_i\}_{i=0}^n$ можно минимизировать величину $|\omega_{n+1}(x)|$, входящую в оценку (3.3), и тем самым уменьшить погрешность интерполяции. Решением задачи минимизации $|\omega_{n+1}(x)|$ по норме $\|\cdot\|_C$ на отрезке $[a, b]$ являются корни многочлена Чебышева 1-го рода, т. е. точки

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{(2i+1)\pi}{2(n+1)}, \quad i = 0, 1, \dots, n.$$

Сетку с узлами в указанных точках называют **чебышевской**, для нее справедлива оценка

$$\max_{x \in [a, b]} |\omega_{n+1}(x)| = \frac{(b-a)^{n+1}}{2^{n+1}},$$

и из формулы (3.3) получается, что погрешность интерполяции на чебышевской сетке

$$|y(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \frac{(b-a)^{n+1}}{2^{n+1}}.$$

Сплайн-интерполяция

Как упоминалось выше, интерполирование функции на всем отрезке $[x_0, x_n]$ многочленом Лагранжа при большом числе узлов интерполяции часто приводит к плохому приближению, особенно при использовании равномерных сеток. Другой класс проблем, возникающих при использовании интерполяционных полиномов высокой степени, связан с ростом погрешностей округления. Многих недостатков глобальной полиномиальной интерполяции лишена **кусочно-полиномиальная интерполяция**, иначе называемая **сплайн-интерполяцией**.

Сплайн-функцией, или **сплайном**, называют кусочно-полиномиальную функцию, определенную на всем отрезке $[x_0, x_n]$ и имеющую на этом отрезке некоторое число непрерывных производных. Рассмотрим частный случай, достаточно распространенный на практике, когда сплайн строится с помощью многочленов третьей степени (кубический сплайн).

Кубическим сплайном для функции $y(x)$ на узлах x_i , $i = 0, 1, \dots, n$, называют функцию $S(x)$, которая удовлетворяет следующим условиям:

- 1) на каждом отрезке $[x_{i-1}, x_i]$, $i = 1, 2, \dots, n$, функция $S(x)$ является многочленом третьей степени;
- 2) функция $S(x)$, а также ее первая и вторая производные непрерывны на отрезке $[x_0, x_n]$;
- 3) значения функции $S(x)$ и исходной функции $y(x)$ совпадают в узлах интерполяции:

$$S(x_i) = y_i, \quad i = 0, 1, \dots, n.$$

На каждом из отрезков $[x_{i-1}, x_i]$, $i = 1, 2, \dots, n$ будем искать функцию $S(x) = s_i(x)$ в виде многочлена третьей степени:

$$s_i = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \\ x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n, \quad (3.4)$$

где a_i, b_i, c_i, d_i — коэффициенты, подлежащие определению.

Из условия $S(x_i) = y_i$, $i = 1, 2, \dots, n$, с учетом (3.4) получим:

$$a_i = y_{i-1}, \quad i = 1, 2, \dots, n, \quad (3.5)$$

$$a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, \quad i = 1, 2, \dots, n. \quad (3.6)$$

Здесь $h_i = x_i - x_{i-1}$ — длина отрезка $[x_{i-1}, x_i]$.

Из условия непрерывности первой и второй производной (т. е. производные справа и слева в узлах сетки должны совпадать) для внутренних узлов сетки получаем

$$S'(x_i - 0) = S'(x_i + 0),$$

$$S''(x_i - 0) = S''(x_i + 0), \quad i = 1, 2, \dots, (n - 1),$$

или, с учетом (3.4),

$$b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1}, \quad i = 1, 2, \dots, (n - 1), \quad (3.7)$$

$$2c_i + 6d_i h_i = 2c_{i+1}, \quad i = 1, 2, \dots, (n - 1). \quad (3.8)$$

Таким образом, получаем систему $(4n - 2)$ уравнений с $4n$ неизвестными. Недостающие два уравнения получаются из граничных условий для $S(x)$ (например, на границах может быть задана первая или вторая производная исходной функции). Если таких условий не дано, то можно положить

$$S''(x_0) = S''(x_n) = 0.$$

Далее рассмотрим именно этот вариант граничных условий (т. н. **естественный кубический сплайн**). Тогда два недостающие уравнения имеют вид

$$\begin{aligned} S''(x_0) &= 2c_1 = 0, \\ S''(x_n) &= 2c_n + 6d_n h_n = 2c_{n+1} = 0. \end{aligned} \quad (3.9)$$

Здесь коэффициент c_{n+1} введен для удобства дальнейших выкладок. Исключая из системы уравнений (3.5)–(3.9) переменные a_i , b_i , d_i и вводя вспомогательный параметр

$$g_i = \frac{y_i - y_{i-1}}{h_i},$$

получим систему:

$$\begin{cases} c_1 = 0, \\ h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3(g_i - g_{i-1}), \quad i = 2, 3, \dots, n. \\ c_{n+1} = 0, \end{cases} \quad (3.10)$$

Система (3.10) обладает трехдиагональной матрицей, поэтому для ее решения можно использовать метод прогонки, который в данном случае будет всегда устойчив в силу наличия у матрицы диагонального преобладания. Остальные коэффициенты определяются по следующим формулам:

$$\begin{aligned} b_i &= g_i - \frac{(c_{i+1} + 2c_i)h_i}{3}, \\ d_i &= \frac{c_{i+1} - c_i}{3h_i}, \quad i = 1, 2, \dots, n. \end{aligned}$$

Если $y(x) \in C^4[a, b]$, то для погрешности сплайн-интерполяции имеются оценки

$$\max_{x \in [x_0, x_n]} |y(x) - S(x)| \leq C_1 M_4 h_{\max}^4,$$

$$\max_{x \in [x_0, x_n]} |y'(x) - S'(x)| \leq C_2 M_4 h_{\max}^3,$$

$$\max_{x \in [x_0, x_n]} |y''(x) - S''(x)| \leq C_3 M_4 h_{\max}^2,$$

где $M_4 = \max_{x \in [x_0, x_n]} |y^{IV}(x)|$. Т.е. не только сам сплайн $S(x)$, но и его первая и вторая производные $S'(x)$ и $S''(x)$ также сходятся в равномерной норме к соответствующим производным функции $y(x)$.

Содержание отчета

1. Номер варианта и исходные данные.
2. Краткое описание используемых алгоритмов.
3. Результаты расчетов, включающие в себя графики исходной функции и интерполантов при различном числе узлов для равномерных и чебышевских сеток.
4. Подробный анализ результатов. Погрешность интерполяции на сетках с разным числом узлов. Трудоемкость рассмотренных способов интерполяции.

Контрольные вопросы

1. Определите количество арифметических операций, требуемое для интерполирования функции в некоторой точке многочленом Лагранжа (включая построение самого многочлена) на сетке с числом узлов, равным n .
2. Определите количество арифметических операций, требуемое для интерполирования функции в некоторой точке кубическим сплайном (включая затраты на вычисление коэффициентов сплайна) на сетке с числом узлов, равным n .

3. Функция $f(x) = e^x$ интерполируется многочленом Лагранжа на отрезке $[0, 2]$ на равномерной сетке с шагом $h = 0,2$. Оцените ошибку экстраполяции в точке $x = 2,2$, построив многочлен Лагранжа и подставив в него это значение, а также по формуле для погрешности экстраполяции.
4. Выпишите уравнения для параметров кубического сплайна, если в узлах x_0 и x_n помимо значений функции y_0 и y_n заданы первые производные $y'(x_0)$ и $y'(x_n)$.
- 5*. Каковы достоинства и недостатки сплайн-интерполяции и интерполяции многочленом Лагранжа?
- 6*. Какие свойства полиномов Чебышева и чебышевских сеток Вам известны?

Примеры заданий

1. Тестовый пример «Тест 1»:

$$y = x^2, \quad x \in [-1; 1].$$

2. Тестовый пример «Тест 2» (пример Рунге):

$$y = \frac{1}{1 + x^2}, \quad x \in [-1; 1].$$

3. Тестовый пример «Тест 3»:

$$y = \frac{1}{\operatorname{arctg}(1 + 10x^2)}, \quad x \in [-3; 3].$$

4. Тестовый пример «Тест 4»:

$$y = (4x^3 + 2x^2 - 4x + 2)^{\sqrt{2}} + \arcsin \frac{1}{5 + x - x^2} - 5,$$
$$x \in [-1; 1].$$

Лабораторная работа № 4

Методы решения проблемы собственных значений

Цель работы

Изучение методов определения собственных значений и собственных векторов линейного оператора:

- а) метода QR -разложения для поиска собственных значений (спектра линейного оператора);
- б) метода обратных итераций и его модификации для нахождения собственных векторов.

Содержание работы

Рассмотрим линейный оператор A , заданный своей матрицей в некотором ортонормированном базисе. Требуется выполнить следующие действия.

1. Написать программу для поиска собственных значений линейного оператора методом QR -разложения с предварительным приведением матрицы оператора к форме Хессенберга методом вращений, а также для нахождения собственных векторов методом обратных итераций.
2. С использованием отношения Рэлея реализовать модификацию метода обратных итераций, позволяющую сразу находить и собственные значения, и собственные векторы оператора.

3. Вычислить приближенно собственные значения линейного оператора A методом QR -разложения и выполнить проверку.
4. С использованием полученных приближений к собственным значениям найти собственные векторы оператора A методом обратных итераций и выполнить проверку.
5. Повторить поиск собственных значений и собственных векторов линейного оператора A с использованием комбинации отношения Рэлея и метода обратных итераций.

Методические указания

Общие сведения о собственных значениях и собственных векторах матрицы линейного оператора

Рассмотрим линейный оператор $A: \mathbb{R}^n \rightarrow \mathbb{R}^n$, заданный своей матрицей в некотором ортонормированном базисе. Далее сам оператор и его матрицу будем отождествлять и обозначать через A .

Ненулевой вектор x называют **собственным вектором** линейного оператора A , если для некоторого числа λ выполняется соотношение

$$Ax = \lambda x.$$

При этом число λ называется **собственным значением** (**собственным числом**) линейного оператора A .

В дальнейшем при рассмотрении методов поиска собственных значений и соответствующих им собственных векторов линейного оператора A будем предполагать, что все собственные значения являются действительными и среди них нет кратных.

В данной лабораторной работе рассмотрим метод QR -разложения и метод обратных итераций для отыскания собственных чисел и собственных векторов линейного оператора, а также комбинированный метод, в основе которого лежит использование отношения Рэлея.

Метод ***QR***-разложения для поиска собственных значений

Один из способов нахождения собственных значений квадратной матрицы — это ее приведение к треугольному (для определенности верхнетреугольному) виду ***преобразованием подобия***

$$R = P^{-1}AP,$$

где P — некоторая невырожденная матрица. Такое преобразование сохраняет собственные значения матрицы, а собственными значениями треугольных матриц являются элементы их главных диагоналей.

Для нахождения матрицы подобия P можно использовать ***QR***-алгоритм, основанный на представлении матрицы в виде произведения ортогональной и верхней треугольной матриц (см. Лабораторную работу № 1, стр. 9). Описанный подход к поиску собственных значений называют ***методом QR-разложения***, или ***алгоритмом Френсиса***¹ — ***Кублановской***².

На первой итерации строится ***QR***-разложение матрицы $A^{(0)} = A$:

$$A^{(0)} = Q_1 R_1, \quad \text{отсюда} \quad R_1 = Q_1^{-1} A^{(0)}.$$

Затем вычисляется матрица $A^{(1)} = R_1 Q_1$, или

$$A^{(1)} = Q_1^{-1} A^{(0)} Q_1.$$

Таким образом, матрицы $A^{(0)}$ и $A^{(1)}$ подобны и, следовательно, имеют один и тот же набор собственных значений.

На второй итерации находится ***QR***-разложение матрицы $A^{(1)}$ и вычисляется матрица $A^{(2)}$, подобная матрице $A^{(1)}$ и т. д.

На $(k + 1)$ -й итерации определяется разложение

$$A^{(k)} = Q_{k+1} R_{k+1}$$

и строится матрица

$$A^{(k+1)} = R_{k+1} Q_{k+1} = Q_{k+1}^{-1} A^{(k)} Q_{k+1}.$$

¹ Джон Френсис (р. 1934) — английский ученый, специалист в области вычислительной математики.

² Вера Николаевна Кублановская (1920–2012) — советский и российский ученый, специалист по численным методам линейной алгебры.

В результате получается последовательность матриц $\{A^{(k)}\}$, подобных матрице A . В случае, если собственные значения матрицы A вещественны и различны по модулю, причем

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|,$$

последовательность $\{A^{(k)}\}$ сходится к верхней треугольной матрице R вида

$$R = \begin{pmatrix} \lambda_1 & * & * & \dots & * \\ 0 & \lambda_2 & * & \dots & * \\ 0 & 0 & \lambda_3 & \dots & * \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_n \end{pmatrix}.$$

Звездочками отмечены элементы, в общем случае не равные нулю. При этом элементы $a_{ij}^{(k)}$ матриц $A^{(k)}$, стоящие ниже главной диагонали, сходятся к нулю со скоростью геометрической прогрессии, знаменатель которой равен модулю отношения соответствующей пары собственных значений, то есть

$$|a_{ij}^{(k)}| \leq \left| \frac{\lambda_i}{\lambda_j} \right| \cdot |a_{ij}^{(k-1)}|, \quad i > j, \quad k = 1, 2, \dots$$

Заметим, что если среди собственных чисел матрицы A есть близкие по величине, то есть для некоторых значений i и j ($i > j$)

$$\left| \frac{\lambda_i}{\lambda_j} \right| \approx 1,$$

то сходимость будет очень медленной. Поэтому на практике часто используют алгоритм со сдвигами. В этом случае ищут собственные значения не матрицы A , а матрицы $\bar{A} = A - \sigma E$, которые равны $\bar{\lambda}_i = \lambda_i - \sigma$. То есть определяют значения, «сдвинутые» относительно искомым собственным значениям на величину σ , которая и называется **сдвигом**. При таком подходе скорость сходимости QR -алгоритма определяется величиной

$$\left| \frac{\bar{\lambda}_i}{\bar{\lambda}_j} \right| = \left| \frac{\lambda_i - \sigma}{\lambda_j - \sigma} \right|.$$

Если σ является хорошим приближением для λ_i , то это соотношение будет много меньше единицы и алгоритм будет быстро сходиться. Заметим, что если $\sigma = \lambda_i$, то QR -алгоритм сойдется к собственному значению λ_i за одну итерацию.

Действительно, если $\sigma = \lambda_i$, то матрица $\tilde{A} = A - \sigma E$ вырождена, следовательно в ее QR разложении последняя строка матрицы R будет заполнена нулями. Соответственно, в матрице $\tilde{A}^{(1)} = RQ$ последняя строка также будет нулевой. После обратного сдвига в последней строке матрицы $A^{(1)} = \tilde{A}^{(1)} + \sigma E$ на диагонали будет стоять λ_i . Для поиска остальных собственных значений QR -алгоритм нужно применять к минору размера $(n-1) \times (n-1)$, расположенному в первых $(n-1)$ строке и $(n-1)$ столбце. Таким образом, величину сдвига σ нужно выбирать как можно ближе к собственному значению, тогда алгоритм сойдется быстрее.

В данной лабораторной работе предлагается реализовать следующий вариант QR -алгоритма со сдвигами.

1. Выбираем величину сдвига σ .
2. Выполняем сдвиг: $\tilde{A}^{(k)} = A^{(k)} - \sigma E$.
3. Вычисляем разложение: $\tilde{A}^{(k)} = \tilde{Q}^{(k)} \tilde{R}^{(k)}$.
4. Выполняем обратный сдвиг: $A^{(k+1)} = \tilde{R}^{(k)} \tilde{Q}^{(k)} + \sigma E$.
5. Повторяем итерации до сходимости ($k = 1, 2, 3, \dots$).

В качестве величины сдвига σ можно взять $a_{n,n}^{(k)}$. Когда в последней строке матрицы $A^{(k)}$ внедиагональные элементы станут близки к нулю, будем считать соответствующее собственное значение найденным с достаточной точностью и перейдем к задаче меньшей размерности для поиска остальных собственных значений: будем искать спектр матрицы размерности $(n-1) \times (n-1)$.

QR -разложение матрицы требует весьма значительных затрат вычислительных ресурсов ($O(n^3)$ операций), поэтому используются различные методики ускорения QR -алгоритма. В частности, затраты времени на вычисления можно сократить, если с помощью подобных преобразований матрицу A предварительно привести к форме Хессенберга.

Матрицей Хессенберга³ называется такая матрица H , у которой равны нулю все элементы, лежащие ниже диагонали, примыкающей к главной (т. е. $h_{ij} = 0$ при $i > j + 1$).

³ Герхард Хессенберг (1874–1925) — немецкий математик, крупный специалист в области дифференциальной геометрии и оснований геометрии; занимался также теорией множеств.

Матрицы Хессенберга обладают следующими свойствами:

- 1) матрицы $H^{(k)}$, порождаемые QR -алгоритмом из матрицы $H^{(0)}$, также являются матрицами Хессенберга;
- 2) выполнение одной итерации QR -алгоритма для матрицы Хессенберга требует $O(n^2)$ арифметических операций.

Приведение матрицы к форме Хессенберга

Существует несколько эффективных методов приведения матрицы A к форме Хессенберга. В данной лабораторной работе предлагается использовать **метод вращений**, называемый также *методом Гивенса*.⁴ Этот метод основан на специально подобранном вращении системы координат.

Известно, что любое вращение можно свести к последовательности элементарных вращений — поворотов в двумерных плоскостях, проходящих через k -ю и l -ю оси координат. Для вещественных векторов матрица элементарного вращения имеет вид (элементы α и β находятся в строках и столбцах с номерами k и l):

$$T_{kl} = \begin{pmatrix} 1 & & & & & & 0 \\ & 1 & & & & & \\ & & \dots & & & & \\ & & & \alpha & & \beta & \\ & & & & 1 & & \\ & & & & & \dots & \\ & & & -\beta & & \alpha & \\ 0 & & & & & & \dots & 1 \end{pmatrix}, \quad \begin{aligned} \alpha &= \cos \varphi, \\ \beta &= \sin \varphi, \\ \alpha^2 + \beta^2 &= 1. \end{aligned}$$

Остальные недиагональные элементы матрицы T_{kl} — нули. Заметим, что если из-за погрешностей расчета окажется, что $\alpha^2 + \beta^2 \neq 1$, то нарушается ортогональность матрицы T_{kl} . В частности, если исходная матрица A симметричная, это приведет к потере симметрии при преобразованиях, что в свою очередь сильно ухудшит устойчивость всего метода вращений. Поэтому в расчетах следует определять α и β таким способом, чтобы указанные соотношения выполнялись с высокой точностью.

⁴ Джеймс Уоллес Гивенс (1910–1993) — американский математик, специалист в области численных методов.

Построим такую последовательность элементарных вращений, которая приведет матрицу A к форме Хессенберга. Матрица

$$A^* = T_{kl}AT_{kl}^{-1} = T_{kl}AT_{kl}^T$$

отличается от матрицы A лишь двумя строками и двумя столбцами с номерами k, l , при этом в матрице A^* элемент $a_{l,k-1}^* = 0$, если

$$\alpha = \frac{a_{k,k-1}}{\sqrt{a_{k,k-1}^2 + a_{l,k-1}^2}}, \quad \beta = \frac{a_{l,k-1}}{\sqrt{a_{k,k-1}^2 + a_{l,k-1}^2}}.$$

Будем обнулять все элементы матрицы A , лежащие ниже диагонали, примыкающей к главной. Сначала обнулим a_{31} , получим матрицу

$$A^{(1)} = T_{23}AT_{23}^{-1}.$$

Вращение

$$A^{(2)} = T_{24}A^{(1)}T_{24}^{-1}$$

обнуляет $a_{41}^{(1)}$. При этом в матрице меняются элементы вторых и четвертых строк и столбцов. Поэтому аннулированный элемент, лежащий в третьей строке первого столбца, так и останется равным нулю.

Последовательно обнулив в первом столбце все элементы, лежащие ниже диагонали, примыкающей к главной, начинаем обнулять элементы второго столбца и так далее. Можно убедиться, что однажды обнуленный элемент будет оставаться равным нулю. Поэтому после окончания всех исключений матрица A будет приведена к форме Хессенберга.

Если исходная матрица A симметрична, то для экономии времени при каждом преобразовании достаточно вычислять только изменившиеся элементы нижней половины матрицы (уже обратившиеся в нуль элементы в расчеты включать не нужно). При этом в конечном итоге она будет приведена к трехдиагональной форме. Одновременно с этим произойдет уточнение диагональных элементов и они будут служить лучшими приближениями к собственным значениям в алгоритме со сдвигами, что ускорит сходимость.

Отметим, что QR -алгоритм поиска собственных чисел, предложенный еще в 1961 году, был включен авторитетным журналом *Computing in Science and Engineering* в десятку важнейших алгоритмов вычислительной математики, оказавших наибольшее влияние на развитие науки и техники в XX веке.

Метод обратных итераций

Если известно собственное значение λ_i матрицы A или достаточно точное приближение λ_i^* к нему, то можно рассмотреть задачу нахождения собственного вектора, отвечающего данному собственному значению.

Если исходить непосредственно из определения собственного вектора, то e_i следует искать как нетривиальное решение системы линейных алгебраических уравнений

$$(A - \lambda_i E)e_i = 0$$

с вырожденной матрицей $(A - \lambda_i E)$. Но обычно λ_i известно лишь приближенно, и в действительности приходится решать систему

$$(A - \lambda_i^* E)e_i = 0, \quad (4.1)$$

решение которой может быть только тривиальным, так как матрица $(A - \lambda_i^* E)$ невырождена. Поэтому непосредственное численное решение (4.1) не дает возможности вычислить соответствующий собственный вектор.

Одним из самых эффективных методов нахождения собственных векторов является **метод обратных итераций**, или *метод Виландта*.⁵ Каждая итерация этого метода состоит из двух этапов.

- На первом этапе решается система

$$(A - \lambda_i^* E)y^{(k+1)} = x^{(k)}$$

относительно неизвестного вектора $y^{(k+1)}$.

- На втором этапе производится нормировка решения:

$$x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|}.$$

⁵ Гельмут Виландт (1910–2001) — немецкий математик, основные труды посвящены теории групп.

В качестве $x^{(0)}$ можно взять любой нормированный вектор. При условии, что известное приближение λ_i^* достаточно близко к истинному значению λ_i , последовательность векторов $x^{(k)}$ довольно быстро сходится к собственному вектору e_i , соответствующему собственному значению λ_i .

Отношение Рэля

Одной из проблем, которые могут возникнуть при использовании метода обратной итерации, является получение хорошего приближения λ_i^* для собственного значения λ_i . Известно, что если A — симметричная матрица, то справедлива формула

$$\lambda_{\min} = \min_{x \neq 0} \rho(x),$$

$$\text{где } \rho(x) = \frac{(Ax, x)}{(x, x)} - \text{отношение Рэля.}^6$$

Отношение Рэля может быть использовано для уточнения собственных значений, поскольку если x — хорошее приближение к собственному вектору, то $\rho(x)$ — хорошее приближение к собственному значению.

Комбинация отношения Рэля и метода обратных итераций дает эффективный алгоритм одновременного определения и собственных значений, и собственных векторов матрицы. Этот метод состоит из трех этапов.

- 1) Нахождение приближения к собственному числу с помощью отношения Рэля:

$$\lambda^{(k)} = \rho(x^{(k)}) = (Ax^{(k)}, x^{(k)}) \quad (\text{учтено, что } \|x^{(k)}\| = 1).$$

- 2) Решение системы линейных алгебраических уравнений

$$(A - \lambda^{(k)} E)y^{(k+1)} = x^{(k)}.$$

$$\text{3) Нормировка } x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|}.$$

Комбинированный метод сходится при условии достаточно хорошего приближения начального вектора $x^{(0)}$ к собственному вектору.

⁶ Джон Уильям Стретт, Лорд Рэлей (1842–1919) — британский физик, лауреат Нобелевской премии. Занимался приложениями теории колебаний в разных областях физики — акустике, оптике, электричестве и др.

Содержание отчета

1. Номер варианта и исходные данные.
2. Краткое описание используемых алгоритмов с указанием условия перехода к алгоритму со сдвигами и критериев окончания счета.
3. Результаты расчетов, включающие в себя последовательность матриц $H^{(k)}$ (или выборку из нее), последовательность приближений к собственным векторам и результаты работы комбинированного метода.
4. Анализ результатов: достигнутая точность (количество верных цифр в ответе), оценка эффективности алгоритма со сдвигами.

Контрольные вопросы

1. Почему нельзя находить собственные числа матрицы A , прямо решая уравнение $\det(A - \lambda E) = 0$, а собственные векторы — «по определению», решая систему $(A - \lambda_j E)e_j = 0$?
2. Докажите, что ортогональное преобразование подобия сохраняет симметрию матрицы.
3. Как преобразование подобия меняет собственные векторы матрицы?
4. Почему на практике матрицу A подобными преобразованиями вращения приводят только к форме Хессенберга, но не к треугольному виду?
5. Оцените количество арифметических операций, необходимое для приведения произвольной квадратной матрицы A к форме Хессенберга.
6. Сойдется ли алгоритм обратных итераций, если в качестве начального приближения взять собственный вектор, соответствующий другому собственному значению? Что будет в этой ситуации в методе обратной итерации, использующем отношение Рэлея?

- 7*. Сформулируйте и обоснуйте критерий останова для QR -алгоритма отыскания собственных значений матрицы.
- 8*. Предложите возможные варианты условий перехода к алгоритму со сдвигами. Предложите алгоритм выбора величины сдвига.
- 9*. Для чего нужно на каждой итерации нормировать приближение к собственному вектору?
- 10*. Приведите примеры использования собственных чисел и собственных векторов в численных методах.

Примеры заданий

Матрицы линейных операторов для поиска собственных значений и собственных векторов

1. Тестовый пример:

$$A = \begin{pmatrix} 1.50 & 0.00 & -0.43 & -0.75 \\ 0.00 & 3.00 & 0.87 & -0.50 \\ -0.43 & 0.87 & 2.90 & -0.22 \\ -0.75 & -0.50 & -0.22 & 2.60 \end{pmatrix} \quad \left| \quad \begin{array}{l} \lambda_1 \approx 1.00, \\ \lambda_2 \approx 2.00, \\ \lambda_3 \approx 3.00, \\ \lambda_4 \approx 4.00. \end{array} \right.$$

Собственные векторы:

$$e_1 \approx (-0.87, 0.00, -0.25, -0.43)^T;$$

$$e_2 \approx (0.00, 0.71, -0.61, 0.35)^T;$$

$$e_3 \approx (0.50, 0.00, -0.43, -0.75)^T;$$

$$e_4 \approx (0.00, 0.71, 0.61, -0.35)^T.$$

Лабораторная работа № 5

Методы решения нелинейных уравнений

Цель работы

Изучение численных методов решения нелинейных уравнений и систем нелинейных уравнений:

- а) метода бисекции и метода Ньютона для решения нелинейного уравнения;
- б) метода Ньютона для решения систем нелинейных уравнений.

Содержание работы

Рассмотрим задачу поиска корней нелинейного уравнения вида

$$f(x) = 0, \quad x \in [a, b],$$

и задачу поиска решений системы нелинейных уравнений:

$$\begin{cases} f_1(x_1, x_2) = 0, \\ f_2(x_1, x_2) = 0, \end{cases} \quad |x_1| \leq L_1, \quad |x_2| \leq L_2.$$

Требуется выполнить следующие действия.

1. Написать программу для нахождения на заданном отрезке корней нелинейного уравнения $f(x) = 0$. Программа должна включать в себя процедуру локализации корней и процедуру уточнения корней методами бисекции и Ньютона.
2. Определить количество корней уравнения $f(x) = 0$ на заданном отрезке и локализовать их. Вычислить корни уравнения с точностью $\varepsilon = 10^{-6}$. Сравнить количество итераций в методах бисекции и Ньютона.

3. Предложить и реализовать в программе модификацию алгоритма метода Ньютона так, чтобы исключался выход за пределы отрезка локализации.
4. Написать программу для решения системы нелинейных уравнений методом Ньютона. Алгоритм решения должен быть построен так, чтобы очередное приближение не выходило за границы заданной области.
5. Построить графики уравнений $f_1(x_1, x_2) = 0$ и $f_2(x_1, x_2) = 0$ в области $|x_1| \leq L_1$, $|x_2| \leq L_2$. По графикам определить количество решений системы и области их расположения.
6. Найти решения системы с точностью $\varepsilon = 10^{-6}$. Сравнить результаты, полученные с использованием матриц Якоби, вычисленных аналитически и численно.
7. Построить диаграмму сходимости и провести ее анализ.

Методические указания

Общие сведения о локализации корней нелинейного уравнения

Первым этапом реализации любого численного метода поиска корней нелинейного уравнения является процедура **локализации корней**, т. е. поиска таких отрезков, на которых уравнение

$$f(x) = 0, \quad x \in [a, b], \quad (5.1)$$

имеет единственный корень.

Универсального способа локализации корней нелинейных уравнений не существует. Если имеется график функции $f(x)$, определяющей левую часть уравнения (5.1), то локализовать корни можно визуально. Однако чаще используется локализация корней с помощью таблиц. Она основана на известной теореме Больцано — Коши о свойствах непрерывной функции:

если непрерывная на отрезке функция $f(x)$ принимает на концах этого отрезка значения разных знаков, то этот отрезок содержит по крайней мере один корень уравнения $f(x) = 0$.

В данной лабораторной работе следует реализовать именно такой подход. Для этого нужно построить достаточно подробную таблицу значений функции $f(x)$ в точках, равномерно распределенных по заданному отрезку, а затем выбирать пары соседних точек, в которых функция принимает значения разных знаков. Использование недостаточно подробных таблиц может привести к потере некоторых корней уравнения. В то же время использование слишком подробных таблиц приведет к необоснованным временным затратам, поскольку потребуется слишком много раз вычислять значение функции $f(x)$.

После того, как процедура локализации корней выполнена, можно приступить к их поиску. Для этого существуют различные численные методы, которые по своей сути представляют собой процедуру уточнения корней.

В данной лабораторной работе рассмотрим наиболее распространенные численные методы — метод бисекции и метод Ньютона¹.

Метод бисекции

Метод бисекции, или *метод деления пополам*, основан на том же принципе, что и процедура автоматической локализации корней.

Пусть $[a, b]$ — отрезок локализации одного из корней уравнения $f(x) = 0$, то есть $f(a) \cdot f(b) < 0$. Если в качестве приближенного значения корня уравнения принять середину этого отрезка, то можно утверждать, что он известен с точностью $\varepsilon_0 = (b - a)/2$. Точность можно повысить, если каким-либо образом уменьшить отрезок. Пусть

$$a^{(0)} = a, \quad b^{(0)} = b, \quad x^{(0)} = \frac{a + b}{2}.$$

Тогда в качестве следующего отрезка локализации $[a^{(1)}, b^{(1)}]$ берется тот из отрезков $[a^{(0)}, x^{(0)}]$ и $[x^{(0)}, b^{(0)}]$, для которого

¹ Исаак Ньютон (1642/43–1727) — английский физик, математик и астроном, один из создателей классической физики, разработал дифференциальное и интегральное исчисление и многие другие математические и физические теории.

Нелинейные системы решаются практически только итерационными методами. Скорость сходимости всех итерационных алгоритмов зависит от выбора начального приближения

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

Множество начальных приближений $X^{(0)}$, для которых итерационный процесс сходится к решению системы (5.2), называется **областью сходимости** данного метода. Выбор начального приближения $X^{(0)}$ является нетривиальной задачей и универсальных способов его осуществления не существует.

В данной лабораторной работе рассмотрим метод Ньютона для решения систем нелинейных уравнений.

Метод Ньютона

Полагая в системе (5.2) функции f_1, \dots, f_n непрерывно-дифференцируемыми в некоторой окрестности решения, линеаризуем систему уравнений (5.2), записав для каждой функции $f_i(X)$, $i = 1, 2, \dots, n$, разложение по формуле Тейлора в окрестности известного приближения $X^{(k)}$ к решению X^* :

$$0 = f_i(X^*) = f_i(X^{(k)}) + \sum_{j=1}^n \left. \frac{\partial f_i}{\partial x_j} \right|_{X^{(k)}} (X^* - X^{(k)}) + \dots, \quad i = 1, 2, \dots, n. \quad (5.4)$$

Пренебрегая в (5.4) нелинейными членами и заменяя X^* на $(k+1)$ -е приближение к решению, получим систему, имеющую в матричной форме следующий вид:

$$F(X^{(k)}) + F'(X^{(k)}) (X^{(k+1)} - X^{(k)}) = 0. \quad (5.5)$$

Здесь $F'(X)$ — **матрица Якоби**, имеющая вид

$$F'(X) = \begin{pmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_X & \cdots & \left. \frac{\partial f_1}{\partial x_n} \right|_X \\ \vdots & \ddots & \vdots \\ \left. \frac{\partial f_n}{\partial x_1} \right|_X & \cdots & \left. \frac{\partial f_n}{\partial x_n} \right|_X \end{pmatrix}. \quad (5.6)$$

Выражая из уравнения (5.5) $X^{(k+1)}$, получим итерационную формулу метода Ньютона:

$$X^{(k+1)} = X^{(k)} - (F'(X^{(k)}))^{-1} F(X^{(k)}), \quad k = 0, 1, \dots \quad (5.7)$$

При этом обращать матрицу Якоби в явном виде не требуется, и вместо вычислений по формуле (5.7) на практике решают систему линейных уравнений

$$F'(X^{(k)}) Y = -F(X^{(k)}),$$

а затем с использованием полученного вектора поправки Y находят новое приближение:

$$X^{(k+1)} = X^{(k)} + Y.$$

Поскольку в данной лабораторной работе решается система из двух нелинейных уравнений, рекомендуется все же использовать вариант с прямым обращением матрицы Якоби. Для вычисления компонент матрицы Якоби следует предусмотреть аналитическое и численное вычисление частных производных.

При решении одного нелинейного уравнения $f(x) = 0$ уравнение (5.5) примет вид

$$f(x^{(k)}) + f'(x^{(k)})(x^{(k+1)} - x^{(k)}) = 0,$$

отсюда получается итерационная формула метода Ньютона:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad k = 0, 1, 2, \dots$$

Метод Ньютона называют также **методом касательных**, или **методом линеаризации**. Его геометрическая интерпретация состоит в следующем: к графику функции $f(x)$ в точке $x^{(k)}$ строится касательная, и в качестве следующего приближения к корню $x^{(k+1)}$ берется абсцисса точки пересечения касательной с осью Ox . Этим и объясняется альтернативное название метода.

Если известен отрезок $[a, b]$ локализации корня, то для получения начального приближения $x^{(0)}$ можно использовать **метод хорд**

$$x^{(0)} = \frac{f(a) \cdot b - f(b) \cdot a}{f(a) - f(b)},$$

т.е. $x^{(0)}$ — абсцисса точки пересечения с осью Ox отрезка, соединяющего точки $(a, f(a))$ и $(b, f(b))$.

Если в некоторой окрестности корня x^* выполнены условия

$$|f'(x)| \geq m > 0, \quad |f''(x)| \leq M, \quad \frac{|f(x) \cdot f''(x)|}{(f'(x))^2} < 1,$$

где m, M — константы, то при попадании очередного приближения $x^{(s)}$ в эту окрестность итерационный процесс по методу Ньютона будет сходиться с **квадратичной скоростью**:

$$|x^{(k+1)} - x^*| < C |x^{(k)} - x^*|^2, \quad k = s, s+1, s+2, \dots$$

Это достоинство уравновешивается двумя существенными недостатками. Во-первых, необходимо вычислять значение производной $f'(x)$. Если функция $f(x)$ имеет достаточно простой вид, то производную можно вычислить аналитически, однако на практике приходится заменять $f'(x^{(k)})$ некоторым приближением, например, находить значение производной численно по формуле

$$f'(x^{(k)}) \approx \frac{f(x^{(k)} + \varepsilon) - f(x^{(k)})}{\varepsilon}$$

с достаточно малым ε .

Во-вторых, если начальное приближение выбрано неудачно, то метод будет сходиться медленно, либо не сойдется вообще. В частности, если в очередном приближении производная функции окажется близкой к нулю, то следующее приближение может выйти далеко за пределы отрезка локализации. Чтобы этого избежать, можно использовать модификации алгоритма, например, комбинацию метода Ньютона и метода хорд. В таком варианте если очередное приближение метода Ньютона вышло за пределы отрезка локализации, то вместо него используется приближение, получаемое методом хорд.

Итерации метода Ньютона следует продолжать до тех пор, пока не будет выполнено условие

$$|x^{(k+1)} - x^{(k)}| \leq \varepsilon, \quad (5.8)$$

поскольку можно доказать, что в некоторой малой окрестности корня x^* выполнения этого неравенства достаточно для обеспечения требуемой точности ε , т. е. из выполнения неравенства (5.8) будет следовать, что

$$|x^{(k+1)} - x^*| \leq \varepsilon.$$

Область сходимости

Для построения области сходимости метода Ньютона при решении системы нелинейных уравнений рассмотрим квадратную область

$$\Omega = \{(x_1, x_2): |x_1| \leq L_1, |x_2| \leq L_2\}$$

и введем сетку $\omega = \omega_1 \times \omega_2$, где

$$\omega_1 = \{x_{1,i} = -L_1 + ih_1, \quad i = 0, 1, 2, \dots, N, \quad Nh_1 = 2L_1\},$$

$$\omega_2 = \{x_{2,j} = -L_2 + jh_2, \quad j = 0, 1, 2, \dots, N, \quad Nh_2 = 2L_2\},$$

h_1, h_2 — шаги сетки, N — число ячеек сетки по каждому направлению.

Выбирая в качестве начального приближения в методе Ньютона узлы сетки ω , необходимо определить число итераций N_{ij} , которое требуется выполнить, чтобы алгоритм сошелся (критерий сходимости предложите самостоятельно). В зависимости от N_{ij} выбирается цвет закрашивания области

$$\left[x_{1,i} - \frac{h_1}{2}, x_{1,i} + \frac{h_1}{2} \right] \times \left[x_{2,j} - \frac{h_2}{2}, x_{2,j} + \frac{h_2}{2} \right].$$

Если при начальном приближении $(x_{1,i}, x_{2,j})$ метод Ньютона расходится или число итераций превышает 30, то область

$$\left[x_{1,i} - \frac{h_1}{2}, x_{1,i} + \frac{h_1}{2} \right] \times \left[x_{2,j} - \frac{h_2}{2}, x_{2,j} + \frac{h_2}{2} \right]$$

закрашивается красным цветом.

В качестве основной цветовой палитры целесообразно выбирать градации серого, так чтобы сходимости за одну итерацию соответствовал черный цвет, а сходимости за 30 итераций — белый.

Содержание отчета

1. Номер варианта и исходные данные.
2. Краткое описание используемых алгоритмов с указанием используемых критериев окончания счета, оценка необходимого количества итераций.
3. Описание модификации метода Ньютона, позволяющей избежать выхода очередного приближения за отрезок локализации корня.
4. Результаты расчетов, включающие в себя отрезки локализации корней и все последовательные приближения, полученные методом бисекции и методом Ньютона.
5. Анализ результатов: достигнутая точность, невязка, оценка трудоемкостей метода бисекции и метода Ньютона, сравнение числа итераций при численном и аналитическом вычислении производных в методе Ньютона.
6. Диаграмма сходимости и ее анализ: расположение областей, в которых наблюдается быстрая сходимость к решению системы, а также областей, где сходимости нет или она весьма медленная.

Контрольные вопросы

1. Можно ли использовать методы бисекции и Ньютона для нахождения кратных корней уравнения $f(x) = 0$ (т. е. тех, в которых одна или несколько первых производных функций $f(x)$ равны нулю)? Обоснуйте ответ.
2. При каких условиях можно применять метод Ньютона для поиска корней уравнения (5.1)? При каких ограничениях на функцию $f(x)$ метод Ньютона обладает квадратичной скоростью сходимости? В каких случаях можно применять

- метод Ньютона для решения систем нелинейных уравнений?
3. Каким образом можно найти начальное приближение?
 4. Можно ли использовать метод Ньютона для решения СЛАУ?
 - 5*. Предложите альтернативный критерий окончания итераций в методе бисекции, в котором учитывалась бы возможность попадания очередного приближения в очень малую окрестность корня уравнения.
 - 6*. Предложите различные варианты модификаций метода Ньютона. Укажите их достоинства и недостатки.
 - 7*. Предложите алгоритм для исключения заикливания метода Ньютона и выхода за пределы области поиска решения?

Примеры заданий

1. Тестовый пример «Тест 1»:

$$(x - 0.1)(x - 0.22)(x - 0.55)(x - 0.7)(x - 0.75) = 0, \quad x \in [0, 1].$$

Точные значения корней

$$x_1 = 0.1, \quad x_2 = 0.22, \quad x_3 = 0.55, \quad x_4 = 0.7, \quad x_5 = 0.75.$$

2. Тестовый пример «Тест 2»:

$$\sqrt{x+1} - 1 = 0, \quad x \in [-1, 10].$$

Точное значения корня $x = 0$.

При выборе начального приближения $x^{(0)} = 8$ следующее приближение, получаемое методом Ньютона, выйдет за отрезок локализации корня.

3. Тестовый пример «Тест 3»:

$$35x^3 - 67x^2 - 3x + 3 = 0, \quad x \in [0, 1].$$

Точное значения корня $x = 0.2$.

При выборе начального приближения $x^{(0)} = 0$ в методе Ньютона (с аналитическим вычислением производной) происходит заикливание.

4. Тестовый пример «Тест 4»:

$$\begin{cases} x_1^2 - x_2^2 - 15 = 0, \\ x_1x_2 + 4 = 0. \end{cases}$$

Область поиска решений системы: $|x_1| \leq 10, |x_2| \leq 10$.

Данная система имеет 2 решения:

$$\begin{cases} x_1 = 4, \\ x_2 = -1; \end{cases} \quad \begin{cases} x_1 = -4, \\ x_2 = 1. \end{cases}$$

5. Тестовый пример «Тест 5»:

$$\begin{cases} x_1^2 + x_2^2 + x_1 + x_2 - 8 = 0, \\ x_1^2 + x_2^2 + x_1x_2 - 7 = 0. \end{cases}$$

Область поиска решений системы: $|x_1| \leq 10, |x_2| \leq 10$.

Данная система имеет 4 решения:

$$\begin{cases} x_1 = -3, \\ x_2 = 1; \end{cases} \quad \begin{cases} x_1 = 1, \\ x_2 = -3; \end{cases} \quad \begin{cases} x_1 = 1, \\ x_2 = 2; \end{cases} \quad \begin{cases} x_1 = 2, \\ x_2 = 1. \end{cases}$$

Литература

1. *Галанин М.П., Савенков Е.Б.* Методы численного анализа математических моделей. М.: Изд-во МГТУ им. Н.Э. Баумана, 2010. 592 с.
2. *Амосов А.А., Дубинский Ю.А., Копченова Н.В.* Вычислительные методы для инженеров. М.: Изд-во МЭИ, 2008. 670 с.
3. *Самарский А.А.* Введение в численные методы. СПб.: Лань, 2005. 288 с.
4. *Калиткин Н.Н., Альшина Е.А.* Численный анализ. М.: Академия, 2013. 304 с. (Сер. «Университетский учебник»).
5. *Бахвалов Н.С., Жидков Н.П., Кобельков Г.М.* Численные методы. М.: ФИЗМАТЛИТ, 2000. 622 с.
6. *Самарский А.А., Гулин А.В.* Численные методы. М.: Наука, 1989. 429 с.

Оглавление

| | |
|---|----|
| Предисловие | 3 |
| Лабораторная работа № 1. Прямые методы решения систем линейных алгебраических уравнений | 5 |
| Лабораторная работа № 2. Итерационные методы решения систем линейных алгебраических уравнений | 17 |
| Лабораторная работа № 3. Решение задач интерполирования | 26 |
| Лабораторная работа № 4. Методы решения проблемы собственных значений | 36 |
| Лабораторная работа № 5. Методы решения нелинейных уравнений | 47 |
| Литература | 58 |

Учебное издание

МАРЧЕВСКИЙ Илья Константинович
ЩЕРИЦА Ольга Владимировна

**Численные методы решения задач
линейной алгебры**

Редактор ...

Корректор ...

Художник ...

Компьютерная верстка *И. К. Марчевского*

Оригинал-макет подготовлен
в Издательстве МГТУ им. Н. Э. Баумана

Подписано в печать хх.хх.хххх. Формат 60×90/16.
Усл. печ. л. 3,75. Изд. № ххх-хххх Тираж 100 экз. Заказ №

Издательство МГТУ им. Н. Э. Баумана.
105005, Москва, 2-я Бауманская ул., 5, стр. 1.
press@bmstu.ru
www.baumanpress.ru

Отпечатано в типографии МГТУ им. Н. Э. Баумана.
105005, Москва, 2-я Бауманская ул., 5, стр. 1.