Р. БЕЛЛМАН
Э. ЭНДЖЕЛ

Динамическое программирование и уравнения в частных производных

Р. Беллман,
Э. Энджел

Динамическое программирование и уравнения в частных производных

Р. Беллман,
Э. Энджел

Динамическое программирование и уравнения в частных производных

Перевод с английского
С. П. ЧЕБОТАРЁВА

Под редакцией
А. М. ЛЕТОВА

ИЗДАТЕЛЬСТВО «МИР»
МОСКВА 1974

УДК 517.544+518.61

Книга известных американских математиков Ричарда Беллмана и Эдварда Энджела посвящена одной из важнейших задач современной вычислительной математики — созданию устойчивых численных методов решения уравнений в частных производных. Авторы убедительно показывают, что известные методы динамического программирования и инвариантного погружения приводят к эффективным методам решения уравнений эллиптического и параболического типов в регулярных и, что весьма ценно для практики, нерегулярных областях. Удачно подобранные примеры и упражнения позволяют использвать книгу в качестве учебного пособия.

Изложенные результаты представляют большой интерес для специалистов в области численных методов и открывают заманчивые перспективы для дальнейших исследований. Книга интересна и для широкого круга лиц, работающих в области прикладной математики, которые, кроме четких и ясных методов, найдут в ней программы некоторых алгоритмов на языке ФОРТРАН. Книга вполне доступна аспирантам и студентам старших курсов соответствующих специальностей.

Редакция литературы по математическим наукам

Б $\frac{20204\text{-}021}{041(01)\text{-}74}$ 21-74

Предисловие редактора перевода

Предлагаемая книга посвящена проблеме определения приближенных решений уравнений в частных производных. Авторы излагают собственный опыт в разработке методов определения приближенных решений, приобретенный ими в Университете Южной Калифорнии (Лос-Анджелес) на факультете математики и техники. Методы основаны на идеях динамического программирования, принципа инвариантного погружения и квазилинеаризации. Эти идеи подверглись в последние годы интенсивной разработке коллективом ученых названного научного центра, работающих под руководством Р. Беллмана. В книге демонстрируются эффективность этих идей, многие важные свойства и преимущества, которые они имеют в сравнении с идеями, лежащими в основе других известных методов; оценивается возможность реализации предлагаемых вычислительных алгоритмов на ЭЦВМ.

Важная особенность книги в том, что она снабжена большим количеством упражнений, работа над которыми будет не только содействовать лучшему и более глубокому усвоению читателем материала книги, но может представить также и самостоятельный интерес.

Стиль изложения материала хотя и не безупречен, подкупает своей простотой и доступностью. Многие из читателей, убедившись в практической полезности предложенных схем, пожелают затем самостоятельно заняться поисками их строгих математических обоснований, которые авторы зачастую опускают.

В книге почти нет ссылок на литературные источники многих стран и это является серьезным упущением; достаточно полно представлены лишь американские публикации.

В целом книга безусловно представляет научный и педагогический интерес для широкого круга читателей — научных работников, преподавателей, аспирантов и студентов, занимающихся математической физикой, вычислительной математикой, проблемами управления и переходными процессами в замкнутых системах с распределенными параметрами, а также проблемой идентификации этих систем.

А. М. Летов

Предисловие

Одна из основных задач современного математического анализа состоит в создании вычислительных алгоритмов для численного решения уравнений в частных производных всех типов. Для осуществления этого необходимо обладать целым набором самых разнообразных методов. В этой книге мы хотели прежде всего показать, что динамическое программирование и инвариантное погружение позволяют построить весьма мощные методы решения линейных дифференциальных уравнений параболического и эллиптического типов, заданных в регулярных и нерегулярных областях. Решения нелинейных уравнений можно получить, воспользовавшись методом квазилинеаризации, который можно объединить с указанными выше методами для изучения задач идентификации и обращения. Весьма важным является то, что понимание этих алгоритмов не вызывает затруднений; они легко программируемы и просты в употреблении.

Мы хотим поблагодарить Джона Касти, Дэйва Коллинза, Нестора Дистефано и Арта Лью за весьма ценные советы и замечания, сделанные при подготовке рукописи. Мы особенно признательны Джону Тодду из Калифорнийского Технологического института, прочитавшего рукопись со свойственной ему пунктуальностью. Благодаря своей обширной эрудиции и богатому опыту в этой области он сделал множество ценных указаний, что в значительной степени способствовало улучшению как содержания книги, так и характера изложения. Во время подготовки рукописи Эдвард Энджел работал на факультете электротехники и вычислительной техники Калифорнийского университета в Беркли.

Р. Беллман, Э. Энджел

Глава 1

Введение

Главная цель книги заключается в применении идей и методов теории динамического программирования для анализа и численного решения уравнения Лапласа

$$u_{xx} + u_{yy} = 0 \tag{1}$$

с граничным условием

$$u(x, y) = g(x, y), \qquad (x, y) \in \Gamma, \tag{2}$$

где Γ — граница области R, которая может иметь довольно нерегулярный вид, как, например, на рис. 1.

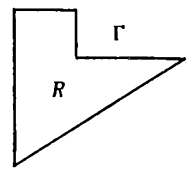


Рис. 1

В настоящее время известно множество различных методов решения задач такого рода, которые хорошо работают при тщательно оговоренных условиях. Однако все они, не исключая, разумеется, и методов, изложенных ниже, обладают различными недостатками. Это обстоятельство и побуждает к постоянному поиску новых подходов. Аргументы за и против этих методов обсуждаются в соответствующих местах книги.

Различные вариации основного подхода и основного уравнения приводят к другим методам и уравнениям, представляющим самостоятельный интерес. Эти вопросы также обсуждаются ниже. Теперь перейдем к изложению основной идеи.

В вариационных задачах, включающих функции одной переменной, скажем задачах минимизации функционала

$$J(u) = \int_0^T g(u, u')\, dt \tag{3}$$

при ограничении $u(0) = c$, фундаментальный результат заключается в том, что минимум $J(u)$ зависит от величин c и T. Будем поэтому считать c и T переменными, удовлетворяющими условиям $T \geqslant 0$ и $-\infty < c < \infty$, и запишем:

$$f(c, T) = \min_u J(u). \tag{4}$$

Исходная вариационная задача таким образом сводится к задаче получения уравнения для $f(c, T)$. В этом заключается подход теории динамического программирования: мы погружаем конкретную задачу с фиксированными значениями c и T в семейство задач, в которых c и T представляют собой области. Затем мы выводим соотношения, связывающие различные элементы этого семейства задач.

Уравнение для $f(c, T)$ можно получить, рассматривая процесс минимизации как многоэтапный процесс решения, в котором выбор функции на сегменте $[0, T]$ разбивается на выбор по подсегменту $[0, s]$ и последующий выбор по подсегменту $[s, T]$, $0 < s < T$. Таким образом,

$$\min_{u \in [0, T]} = \min_{u \in [0, s]} \min_{u \in [s, T]}. \tag{5}$$

С помощью принципа оптимальности легко преобразовать это соотношение в функциональное уравнение для $f(c, T)$. Положив далее $s \to 0$, получим дифференциальное уравнение в частных производных. Подробно это обсуждается в гл. 3.

Аналогичные рассуждения проходят и в многомерном случае. Здесь мы используем тот факт, что в случае двух переменных уравнение в частных производных (1) является уравнением Эйлера для квадратичного функционала

$$J(u) = \int_R (u_x^2 + u_y^2)\, dR, \tag{6}$$

который носит название функционала Дирихле. Как и выше, мы можем рассматривать процесс минимизации $J(u)$ как многошаговый процесс решения, в котором выбор функции $u(x, y)$ на множестве R подразделяется на выбор функции по области R_1 и затем по области $R - R_1$. Это приводит к равенству

$$\min_{u \in R} = \min_{u \in R_1} \min_{u \in (R - R_1)}. \tag{7}$$

Различные способы выбора области R_1 приводят к множеству интересных и полезных результатов. Так, например, если R — прямоугольная область, то мы можем выбрать в качестве R_1 прямоугольную полосу (рис. 2). Это приводит либо к рекуррентному соотношению, либо к дифференциальному уравнению, в зависимости от способа дискретизации и от того, выбрано ли R_1 бесконечно

малым или нет. Если область R имеет форму, показанную на рис. 3, то R_1 можно выбрать, как указано на рисунке, и решение уравнения, заданного на области $R = R_1 + R_2$, можно определить в терминах решений на более простых областях R_1 и R_2.

Все эти вопросы мы будем более подробно рассматривать в последующих главах и приведем численные приложения. Однако

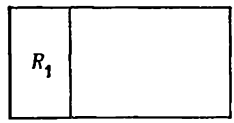


Рис. 2

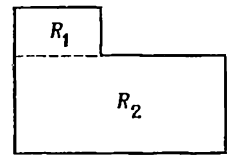


Рис. 3

мы не будем затрагивать соответствующих вопросов, связанных с решением аналогичных уравнений на круговых и сферических областях.

Перейдем теперь к краткому изложению содержания отдельных глав. В гл. 2 мы рассматриваем задачу минимизации квадратичного функционала

$$J(x) = \int_0^T [(x', x') + (x, A(t)x)]\, dt \tag{8}$$

с помощью классического подхода вариационного исчисления. Это приводит к уравнению Эйлера

$$x'' - A(t)x = 0 \tag{9}$$

с различными граничными условиями, зависящими от условий, первоначально наложенных на x. Там же обсуждаются некоторые вопросы, возникающие при численном решении. В конце главы приводится краткое описание методов Рэлея — Ритца и Бубнова — Галеркина. Это мощные методы решения линейных уравнений в частных призводных, возникающих в математической физике и технике. Их недостаток заключается в том, что они сводятся к решению системы линейных алгебраических уравнений большой размерности.

Мы касаемся вопросов такого рода потому, что различные способы дискретизации преобразуют функционал Дирихле в функционал типа (8) и приводят, таким образом, к уравнениям, аналогичным (9).

В гл. 3 излагается применение метода динамического программирования к минимизации функционала $J(x)$ и к минимизации его

дискретного варианта

$$\sum_{n=0}^{N} [(x_{n+1}-x_n,\ x_{n+1}-x_n)+(x_n,\ Ax_n)]. \tag{10}$$

На этом пути мы сталкиваемся с уравнением Риккати и его дискретным аналогом.

Глава 4 посвящена уравнению Лапласа, функциям Грина и различным процедурам дискретизации. Глава 5 содержит более подробное обсуждение применения динамического программирования к дискретному случаю.

В гл. 6 описаны некоторые приложения теории инвариантного погружения к решению линейных уравнений в частных производных. Преимущество такого подхода состоит в том, что он применим к линейным уравнениям более общего вида, которые не обязательно вытекают из вариационного принципа [1]).

В гл. 7 подробно обсуждается применение описанных ранее методов к нерегулярным областям, а гл. 8 посвящена отдельным специальным методам, которые можно применять для регулярных областей.

Теория динамического программирования приводит к некоторым нелинейным уравнениям в частных производных, описывающим непрерывные процессы решения. Эти уравнения легко решаются численно с использованием аналогичных уравнений, описывающих соответствующий дискретный процесс решения. Это предполагает соответствующую процедуру для другого типа уравнений в частных производных. Для иллюстрации этой идеи в гл. 9 рекуррентное соотношение

$$v(x,\ t+\Delta) = v(x+v(x,\ t)\,\Delta,\ t),\quad t = 0,\ \Delta,\ 2\Delta,\ \ldots, \tag{11}$$

используется для изучения уравнения

$$u_t = uu_x \tag{12}$$

и родственных вопросов.

Параболическое уравнение (уравнение теплопроводности)

$$u_t = u_{xx} + u_{yy} \tag{13}$$

изучается в гл. 10 с помощью преобразования Лапласа, сводящего его к уравнению эллиптического типа, которое затем изучается с помощью описанных ранее методов. Значение u далее определяется с помощью численного обращения преобразования Лапласа.

---

[1]) Вариационный принцип заключается в том, что данное дифференциальное уравнение (вместе с условиями на границе) можно рассматривать как необходимое условие минимума некоторого функционала. В тех случаях, когда подобный функционал может быть построен и его минимум существует, решение данного дифференциального уравнения может быть найдено при минимизации этого функционала прямыми методами.— *Прим. ред.*

Наконец, в гл. 11 методы последовательных приближений и квазилинеаризация применяются для решения нелинейных уравнений.

На протяжении всей книги мы касаемся как в высшей степени интересных аналитических вопросов, так и столь же интересной проблемы получения численного ответа на численно сформулированный вопрос с помощью разумно упрощенных регулярных методов и прозрачных вычислительных программ. Мы постоянно подчеркиваем, что эти два типа проблем тесно взаимосвязаны. Аналитический подход следует оценивать исходя из того, является ли он полезным с вычислительной точки зрения или нет. С другой стороны, построение вычислительных методов часто требует привлечения новых аналитических процедур.

Глава 2

Квадратичные вариационные задачи

1. Введение

В этой главе мы хотим рассмотреть некоторые наиболее важные классические методы решения линейных дифференциальных уравнений и квадратичных вариационных задач. Отчетливое понимание тех результатов, которые будут получены, даст возможность применить к тем же самым задачам принципиально иной подход, а именно динамическое программирование. Однако с самого начала следует ясно указать, что все методы сталкиваются с определенными трудностями. Эти трудности различны для различных методов.

Прежде всего рассмотрим задачу минимизации скалярного квадратичного функционала

$$J(u) = \int_0^T (u'^2 + q(t)\, u^2)\, dt \tag{1}$$

при ограничениях $u(0) = c_1$, $u(T) = c_2$ и N-мерный аналог этой задачи, т. е. задачу определения минимума функционала

$$J(x) = \int_0^T [(x', x') + (x, A(t)\, x)]\, dt \tag{2}$$

при ограничениях $x(0) = c$, $x(T) = d$. Нашим основным инструментом при этом будет уравнение Эйлера.

Далее коротко рассмотрим методы Рэлея — Ритца и Бубнова — Галеркина. Все три указанные процедуры в конце концов сводятся к решению систем линейных алгебраических уравнений со всеми вытекающими отсюда осложнениями.

2. Вариационный подход

Предположим, что существует функция u, такая, что u и u' принадлежат классу $L^2(0, T)$, функция u удовлетворяет граничным условиям и доставляет абсолютный минимум функционалу $J(u)$. Наша цель состоит в получении необходимых условий, которым функция u должна удовлетворять. Этим необходимым условием

является уравнение Эйлера, которое в данном случае будет и достаточным.

Поступим следующим образом. Пусть v — другая функция, принадлежащая вместе со своей производной классу $L^2(0, T)$ и такая, что $v(0) = v(T) = 0$. Тогда при любом действительном ε функция $u + \varepsilon v$ удовлетворяет исходным граничным условиям и вместе со своей производной принадлежит $L^2(0, T)$. Рассмотрим функционал

$$J(u+\varepsilon v) = J(u) + \varepsilon^2 J(v) + 2\varepsilon \int_0^T [u'v' + q(t)\, uv]\, dt, \tag{1}$$

который, по предположению, обладает абсолютным минимумом, достигающимся при $\varepsilon = 0$. Этот факт приводит к вариационному условию

$$\int_0^T [u'v' + q(t)\, uv]\, dt = 0 \tag{2}$$

для всех v, обладающих описанными выше свойствами.

Интегрируя (2) по частям, получаем

$$v \int_0^t q(t_1)\, u(t_1)\, dt_1 \Big|_0^T + \int_0^T \left[u'v' - v' \int_0^t q(t_1)\, u(t_1)\, dt_1\right] dt = 0. \tag{3}$$

Проинтегрированный член исчезает, поскольку $v(0) = v(T) = 0$. Следовательно, для любой постоянной c_3

$$\int_0^T v' \left[c_3 + u' - \int_0^t q(t_1)\, u(t_1)\, dt_1\right] dt = 0. \tag{4}$$

Выберем

$$v' = c_3 + u' - \int_0^t q(t_1)\, u(t_1)\, dt_1, \quad v(0) = 0, \tag{5}$$

где c_3 определяется из условия $v(T) = 0$. Тогда (4) приводит к соотношению

$$c_3 + u' - \int_0^t q(t_1)\, u(t_1)\, dt_1 = 0 \tag{6}$$

почти всюду. Слова «почти всюду» можно заменить на «всюду». Таким образом, (6) приводит к уравнению Эйлера

$$u'' - q(t)\, u = 0, \quad u(0) = c_1, \quad u(T) = c_2. \tag{7}$$

3. Положительная определенность, существование и единственность решения

Нетрудно показать, что если функционал $J(u)$ положительно определен, то уравнение (2.7) имеет единственное решение. В самом деле, если имеются два решения, то существует решение v, такое, что $v(0) = v(T) = 0$. Рассмотрим тогда выражение

$$0 = \int_0^T v(v'' - q(t)v)\,dt = vv'\Big|_0^T - \int_0^T [v'^2 + q(t)v^2]\,dt =$$

$$= 0 - \int_0^T [v'^2 + q(t)v^2]\,dt, \qquad (1)$$

что противоречит предположению о положительной определенности J, если v не равно тождественно нулю.

То, что (2.7) имеет решение, следует из изложенного ниже.

4. Вычислительные аспекты

Пусть u_1 и u_2 — фундаментальное решение задачи (2.7), т. е.

$$u_1(0) = 1, \qquad u_2(0) = 0,$$
$$u_1'(0) = 0, \qquad u_2'(0) = 1. \qquad (1)$$

Запишем

$$u = a_1u_1 + a_2u_2, \qquad (2)$$

где a_1 и a_2 определяются из условий

$$c_1 = a_1, \quad c_2 = a_1u_1(T) + a_2u_2(T). \qquad (3)$$

Положительная определенность функционала $J(u)$ гарантирует, что $u_2(T) \neq 0$, как указано в разд. 3, поэтому a_1 и a_2 определены однозначно. Фундаментальное решение находим численным интегрированием уравнения (2.7) с соответствующими начальными условиями.

5. Векторно-матричный случай

Рассуждая аналогичным образом, рассмотрим задачу минимизации

$$J(x) = \int_0^T [(x', x') + (x, A(t)x)]\,dt, \qquad (1)$$

где матрица $A(t)$ положительно определена. Будем предполагать, что $x(0) = c$, $x(T) = d$ и $x' \in L^2(0, T)$. Поступая, как ранее,

получим вариационное уравнение

$$x'' - A(t)\,x = 0, \quad x(0) = c, \quad x(T) = d. \tag{2}$$

Аналогично тому как это было сделано в разд. 3, можно доказать что это уравнение обладает единственным решением.

Однако вычислительные аспекты требуют тщательного анализа. Точно так же, как и в скалярном случае, запишем

$$x = X_1(t)\,a + X_2(t)\,b, \tag{3}$$

где X_1 и X_2 — фундаментальные матричные решения уравнения

$$X'' - A(t)\,X = 0, \tag{4}$$

т. е. решения, для которых соблюдаются условия

$$\begin{aligned} X_1(0) &= I, \qquad X_1'(0) = 0, \\ X_2(0) &= 0, \qquad X_2'(0) = I, \end{aligned} \tag{5}$$

и a и b — постоянные векторы, определяемые из граничных условий, описанных в (2). Эти условия приводят к уравнениям

$$c = a, \quad d = X_1(T)\,a + X_2(T)\,b. \tag{6}$$

Можно показать, что матрица $X_2(T)$ невырожденна, если воспользоваться методом, описанным в разд. 2, который нетрудно распространить и на этот случай. Тогда a и b, а, следовательно, и $x(t)$, однозначно определяются из (2). Однако для определения b требуется решить систему линейных алгебраических уравнений, что является, как правило, довольно сложной задачей. Трудности особенно усугубляются в тех случаях, когда размерность вектора a и число T велики.

Тот факт, что T велико, означает, что $X_2(T)$ близка к вырожденной матрице [1]). Если еще $\dim(X_2)$ велико, то это приводит к тому, что трудно обеспечить достаточную точность численного решения. Подробное обсуждение этих вопросов можно найти в литературе, приведенной в конце главы.

Это является тем самым камнем преткновения для регулярных методов решения систем линейных дифференциальных уравнений большой размерности, который побуждает к постоянному поиску новых методов решения линейных уравнений в частных производных и квадратичных вариационных задач.

---

[1]) Пусть $\Delta(X_2(T))$ — детерминант матрицы $X_2(T)$. Тогда близость этой матрицы к вырожденной означает следующее: сколь бы ни было мало число $\varepsilon > 0$, всегда найдется такое число T^*, что $|\Delta(X_2(T))| < \varepsilon$, коль скоро $T > T^*$.— *Прим. ред.*

6. Метод Рэлея — Ритца

Опишем кратко два из наиболее мощных методов, позволяющих избежать трудностей, указанных в предыдущем разделе. Начнем с метода Рэлея — Ритца. Рассмотрим функционал

$$J(x) = \int_0^T [(x', x') + (x, A(t)x)]\, dt \tag{1}$$

с граничными условиями $x(0) = c$, $x(T) = d$ и используем вспомогательную функцию

$$x = \sum_{k=1}^{M} b_k \varphi_k(t). \tag{2}$$

Мы можем считать φ_k скалярами, а b_k — векторами, и наоборот. В любом случае будем считать φ_k известными функциями, а b_k — неизвестными, удовлетворяющими только граничным условиям.

Тогда

$$J(x) = J(b_1, b_2, \ldots, b_M), \tag{3}$$

и минимизация теперь должна происходить по b_k. Это приводит к системе линейных алгебраических уравнений порядка M, если b_i — скаляры. Если $M \ll N$, то эта задача значительно проще исходной.

Если мы не хотим использовать граничные условия, то можно построить новый функционал

$$J(x, \lambda_1, \lambda_2) = J(x) + \lambda_1 (x(0) - c, x(0) - c) + \\ + \lambda_2 (x(T) - d, x(T) - d), \tag{4}$$

где λ_1 и λ_2 — параметры Куранта и $\lambda_1, \lambda_2 \gg 1$ [1]).

Успех применения данного метода определяется прежде всего удачным выбором последовательности $\{\varphi_k(t)\}$, основанием для которого в каждом конкретном случае служит некая смесь математических рассуждений, физической интуиции и вычислительного опыта.

7. Метод Бубнова — Галеркина

Вместо решения уравнения Эйлера

$$x'' - A(t)x = 0, \quad x(0) = c, \quad x(T) = d \tag{1}$$

рассмотрим задачу минимизации функционала

$$J_1(x) = \int_0^T (x'' - A(t)x, x'' - A(t)x)\, dt \tag{2}$$

---

[1]) В отечественной литературе параметры Куранта обычно называют «коэффициентами штрафа», а соответствующий метод —«методом штрафных функций». Подробнее см., например, Н. Н. Моисеев, «Численные методы в теории оптимальных систем», «Наука», М., 1971.— *Прим. перев.*

по таким $x(t)$, что граничные условия соблюдаются, и интеграл в (2) существует [1]). Теперь для отыскания минимума используем метод Рэлея — Ритца.

Иногда удобно вместо $J(x)$ ввести в рассмотрение смешанное выражение

$$J_2(x) = J(x) + \lambda J_1(x), \tag{3}$$

где λ — параметр Куранта.

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Разделы 1—5. Подробное обсуждение этих задач и ряд дополнительных ссылок можно найти в работе

Беллман (Bellman R.)

Introduction to the mathematical theory of control processes, I: Linear equations and quadratic criteria, Academic Press, New York, 1967.

В качестве введения к методам регуляризации по Тихонову см.

Беллман, Калаба, Локетт (Bellman R., Kalaba R., Lockett J.)

Numerical inversion of the Laplace transform, Amer. Elsevier, New York, 1966.

Разделы 6—7.

Беллман (Bellman R.)

Methods of nonlinear analysis, v. 1, Academic Press, New York, 1970.

---

[1]) Существование интеграла гарантируется, если функция x такова, что $x'' \in L^2(0, T)$.— *Прим. ред.*

Глава 3

Динамическое программирование

1. Введение

В этой главе мы хотим рассмотреть некоторые приложения теории динамического программирования к вариационным процессам, описываемым дифференциальными или разностными уравнениями. Особенно изящные результаты получаются для квадратичных функционалов, сопряженных с линейными уравнениями состояния. Более того, к счастью, именно такие задачи встречаются в множестве различных ситуаций. Использование этих результатов для вычислительных целей обсуждается в следующих главах.

2. Разностные уравнения

Пусть x_n, $n = 0, 1, 2$, есть K-мерный *вектор состояния*, определяемый рекуррентно из разностного уравнения

$$x_{n+1} = g(x_n, y_n), \qquad x_0 = c, \tag{1}$$

где y_n есть L-мерный *вектор управления*. Во многих случаях $L = K$. Будем считать, что множество векторов управления $\{y_n\}$ выбирается таковым, что минимизирует критерий (или функцию «дохода»),

$$J(\{x_n\}, \{y_n\}) = \sum_{n=0}^{N} h(x_n, y_n). \tag{2}$$

Это определяет детерминированный процесс управления дискретного типа. Предполагая, что задача поставлена корректно в том смысле, что минимум достижим, легко показать, что в интересных для нас случаях функция

$$\min_{\{y_n\}} J(\{x_n\}, \{y_n\}) = f_N(c) \tag{3}$$

определена для всех c и $N = 0, 1, 2, \ldots$. Видно, что

$$f_0(c) = \min_y h(c, y), \tag{4}$$

причем во многих случаях эта функция определяется довольно легко.

В этом заключается фундаментальная идея *погружения*, которая будет детально рассмотрена в главе, посвященной инвариант-

ному погружению. При решении конкретной задачи с заданными значениями c и N мы стремимся получить функциональное уравнение для функции $f_N(c)$, которую считаем зависящей от c и N.

3. Функциональное уравнение

Как было отмечено выше, функция $f_0(c)$ определяется выражением

$$f_0(c) = \min_y h(c, y). \tag{1}$$

С учетом этого определения мы намереваемся получить рекуррентное соотношение (т. е. другое разностное уравнение), связывающее $f_{N+1}(c)$ с $f_N(c)$. Теоретически это должно привести к конструктивному способу получения последовательности $f_N(c)$. Практически же, как мы увидим в дальнейшем, здесь остаются некоторые неясные моменты. Аддитивность критерия в (2.2) позволяет легко получить желаемое соотношение.

Имеем

$$\begin{aligned} f_{N+1}(c) &= \min_{\substack{\{y_n\} \\ n=0,\,1,\,2,\,\ldots,\,N+1}} J(\{x_n\}, \{y_n\}) = \\ &= \min_{y_0} \min_{\substack{\{y_n\} \\ n=1,\,2,\,\ldots,\,N+1}} J(\{x_n\}, \{y_n\}) = \\ &= \min_{y_0} \min_{\{y_n\}} \left[h(x_0, y_0) + \sum_{n=1}^{N+1} h(x_n, y_n)\right] = \\ &= \min_{y_0} \left[h(c, y_0) + \min_{\{y_n\}} \sum_{n=1}^{N+1} h(x_n, y_n)\right]. \end{aligned} \tag{2}$$

Если вспомнить определение $f_N(c)$ и учесть, что $x_0 = c$ и $x_1 = g(x_0, y_0) = g(c, y_0)$, то правую часть (2) можно переписать в виде

$$\min_{y_0} [h(c, y_0) + f_N(g(c, y_0))].$$

Поскольку $f_{N+1}(c)$ здесь не зависит от y_0, то мы можем опустить индекс при y_0 и записать полученное уравнение в более простой форме

$$f_{N+1}(c) = \min_y [h(c, y) + f_N(g(c, y))], \quad N = 0, 1, \ldots. \tag{3}$$

Это и есть фундаментальное рекуррентное соотношение, определяющее как последовательность $\{f_N(c)\}$, так и оптимальные значения управляющих переменных.

4. Принцип оптимальности

Функциональное уравнение (3.3) немедленно вытекает из рассмотрения процесса, как многошагового процесса принятия решений и применения принципа оптимальности из теории динамического программирования.

Принцип оптимальности. *Оптимальная политика обладает тем свойством, что, каковы бы ни были начальное состояние и принятое начальное решение, последующие решения должны составлять оптимальную политику относительно состояния, возникшего в результате первоначального решения.*

Часто применимость этого принципа оказывается очевидной, в чем можно убедиться доказательством от противного. Такова, например, ситуация в случае задачи, сформулированной выше. Однако во всех случаях необходимо прежде всего убедиться в его применимости.

5. Нестационарный случай

Во многих важных случаях функции, входящие в описание критерия, зависят от времени, что означает, что J имеет вид

$$J(\{x_n\}, \{y_n\}) = \sum_{n=0}^{N} h_n(x_n, y_n), \tag{1}$$

где каждая функция зависит от состояния.

Для решения задач такого типа с использованием уже описанных понятий мы рассматриваем задачу в обратном времени и вводим функцию

$$\min_{\{y_n\}} \sum_{n=k}^{N} h_n(x_n, y_n) = \varphi_k(c), \tag{2}$$

определенную для $k = 0, 1, 2, \ldots, N$ и $-\infty < c < +\infty$. Здесь N фиксировано, а значение k, соответствующее начальному моменту времени, меняется. Теперь легко определить функцию $\varphi_N(c)$. Она имеет вид

$$\varphi_N(c) = \min_{y} h_N(c, y). \tag{3}$$

При этом рекуррентное соотношение для $\varphi_N(c)$ таково:

$$\varphi_k(c) = \min_{y} [h_k(c, y) + \varphi_{k+1}(g(c, y))],$$
$$k = 0, 1, \ldots, N-1. \tag{4}$$

6. Случай квадратичных функций

Функциональные уравнения существенно упрощаются в случае квадратичного критерия и линейных уравнений состояния. Рассмотрим, например, задачу минимизации квадратичного функ-

ционала

$$J(\{x_n\}, \{y_n\}) = \sum_{n=0}^{N} [(x_n, Ax_n) + (y_n, By_n)], \tag{1}$$

где x и y связаны уравнением

$$x_{n+1} = x_n + Cy_n, \quad x_0 = c \text{ [1]}. \tag{2}$$

Предположим, что A и B положительно определены, так что минимум существует. Как и ранее, запишем

$$f_N(c) = \min_{\{y_n\}} J(\{x_n\}, \{y_n\}), \tag{3}$$

считая, что эта функция определена при $-\infty < c < \infty$, $N = 0, 1, 2, \ldots$. Учитывая (3.3), получаем

$$f_{N+1}(c) = \min_y [[(c, Ac) + (y, By)] + f_N(c + Cy)], \quad N = 0, 1, \ldots, \tag{4}$$

где

$$f_0(c) = \min_y [(c, Ac) + (y, By)] = (c, Ac). \tag{5}$$

Остается упростить выражение (4) и получить, таким образом, некоторые соотношения, более удобные с аналитической и вычислительной точек зрения. Для этой цели мы явно используем тот факт, что $f_N(c)$ является квадратичной функцией от c:

$$f_N(c) = (c, Q_N c), \tag{6}$$

где матрица Q_N не зависит от c.

Квадратичный характер $f_N(c)$ вытекает из линейности обычных вариационных уравнений. Во всяком случае, это легко показать, используя индуктивно соотношение (4), как это сделано ниже. Используя (4) и (6), получаем

$$(c, Q_{N+1}c) = \min_y [(c, Ac) + (y, By) + (c + Cy, Q_N(c + Cy))]. \tag{7}$$

Суть дела заключается в том, что минимум в правой части и соответствующее значение вектора y легко определяются.

---

[1]) Мы пользуемся обычным скалярным произведением для представления квадратичных форм. Так, $(a,b) = \sum_{i=1}^{N} a_i b_i$, где a_i и b_i — i-е компоненты a и b соответственно, поэтому $(x, Ax) = \sum_{i,j=1}^{N} a_{ij} x_i x_j$. Мы используем x_1, x_2, $\ldots$, y_1, y_2, $\ldots$ для обозначения последовательности векторов, чтобы избежать необходимости двойной индексации.

Это значение y является линейной функцией от c. При подстановке этой функции в правую часть (7) получим некоторую другую квадратичную форму от c. Приравнивая соответствующие коэффициенты, получим рекуррентное соотношение, связывающее Q_{N+1} и Q_N. Более подробно это рассматривается в следующей главе в связи с некоторой вариационной задачей, представляющей для нас особый интерес.

Упражнения

1. Рассмотрим задачу минимизации $J(\{u_n\}) = \sum_{n=0}^{N} [(u_{n+1} - u_n)^2 + u_n^2]$, где $u_0 = c$. С помощью замены переменных показать, что $f_N(c) = \min_u J = r_N c^2$.

2. Используя указанную выше процедуру, получить рекуррентное соотношение, связывающее r_N и r_{N+1}.

3. Показать далее, что $\lim_{N \to \infty} r_N$ существует, и определить его величину.

4. Показать непосредственно, что $r_N < r_{N+1}$, что $\{r_N\}$ равномерно ограничена и потому сходится при $N \to \infty$.

5. Рассмотрим функцию $f(c) = \min_u \sum_{n=1}^{\infty} [u_n^2 + (u_n - u_{n-1})^2]$, $u_0 = c$. Показать, что $f(c) = \min_v [c^2 + (v - c)^2 + f(v)]$, и затем определить r_∞ непосредственно.

6. Пусть $f(0) = 0$ и $f(c)$ аналитична по c; определяется ли $f(c)$ из функционального уравнения однозначно?

7. Показать, что $\min_x [(x, Ax) - 2(x, y)] = -(y, A^{-1}y)$.

7. Свертка минимума

Те же идеи можно использовать для получения более общих результатов. Введем функцию двух переменных

$$\varphi_N(c, d) = \min_y J(\{x_n\}, \{y_n\}), \tag{1}$$

где x_n удовлетворяет двум ограничениям

$$x_0 = c, \quad x_N = d \tag{2}$$

и, кроме того, x и y связаны соотношением

$$x_{n+1} = h(x_n, y_n). \tag{3}$$

Запишем

$$\varphi_{M+N}(c,\ d) = \min_{y}\Big[\sum_{n=0}^{M+N}\Big] = \min\Big[\sum_{n=0}^{M}+\sum_{n=M+1}^{N}\Big]. \tag{4}$$

Выберем некоторое значение x_M, скажем $x_M = z$. Тогда из (2) следует

$$\varphi_{M+N}(c,\ d) = \min_{z}\left[\varphi_M(c,\ z) + \varphi_N(z,\ d)\right]. \tag{5}$$

Упражнения

1. Какие необходимы изменения, если функции $g(x,\ y)$ зависят от n?

2. Рассмотреть квадратичный случай и использовать (5) для вывода рекуррентного соотношения, выражающего r_{M+N} через r_M и r_N, где r_N определено так же, как в упражнениях в конце предыдущего раздела.

8. Способ сокращения необходимых вычислений

Во многих случаях требуется определить $f_N(c)$ или $\varphi_N(c)$ только для некоторого конкретного значения N. Если N велико, то может оказаться более эффективным воспользоваться соотношениями из разд. 7 вместо соотношения (3.3). Так, например, из (7.5) вытекает

$$\varphi_{2N+1}(c,\ d) = \min_{z}\left[\varphi_{2N}(c,\ z) + \varphi_{2N}(z,\ d)\right], \tag{1}$$

что позволяет довольно быстро вычислить φ_1, φ_2, φ_4, φ_8, ...

Для вычисления $f_N(c)$ можно использовать соотношение

$$f_{N+M}(c) = \min_{z}\left[\varphi_M(c,\ z) + f_N(z)\right]. \tag{2}$$

Упражнение

Получить соответствующее рекуррентное соотношение, если

$$f_N(c) = \min_{u} J(u).$$

9. Дифференциальные уравнения

Рассмотрим задачу минимизации функционала

$$J(x,\ y) = \int_0^T h(x,\ y)\,dt, \tag{1}$$

где x и y связаны дифференциальным уравнением

$$x' = g(x, y), \quad x(0) = c. \tag{2}$$

Опять-таки предположим, что задача корректно поставлена, и положим

$$\min_y J(x, y) = f(c, T). \tag{3}$$

Запишем

$$\int_0^T = \int_0^S + \int_S^T. \tag{4}$$

Предположив, что S мало, получим приближенное функциональное уравнение

$$f(c, T) = \min_z [h(c, z) S + f(c + Sg(c, z), T - S)] + O(S^2), \tag{5}$$

где $z = y(0)$.

Переходя к пределу при $S \to 0$, получаем нелинейное уравнение в частных производных

$$f_T = \min_z [h(c, z) + (g(c, z), \operatorname{grad} f)] \tag{6}$$

с начальным условием $f(c, 0) = 0$.

10. Квадратичный случай

Результаты значительно упрощаются, когда функция $h(x, y)$ квадратична по x и y, а $g(x, y)$ линейна. Рассмотрим, например, задачу минимизации функционала

$$J(x) = \int_0^T [(x', x') + (x, Ax)]\, dt, \tag{1}$$

где A — положительно определенная матрица и $x(0) = c$. Ясно, что

$$f(c, T) = \min_x J(x) = (c, R(T) c), \tag{2}$$

где $R(T)$ зависит только от T. С другой стороны, из (9.6) следует, что

$$f_T = \min_z [(z, z) + (c, Ac) + (z, \operatorname{grad} f)]. \tag{3}$$

Правую часть (3) легко минимизировать по z. В результате минимизации получаем

$$z = (-\operatorname{grad} f)/2, \tag{4}$$

$$f_T = (c, Ac) - [(\operatorname{grad} f, \operatorname{grad} f)/4]. \tag{5}$$

Подставляя (2) в (5), получаем обыкновенное дифференциальное уравнение

$$R'(T) = A - R^2(T), \quad R(0) = 0. \tag{6}$$

Это уравнение Риккати.

Воспользовавшись соотношениями (2) и (4), получаем

$$x'(0) = z = -R(T)\,c, \tag{7}$$

т. е. требуемое начальное условие.

11. Минимизация с ограничениями

Запишем

$$\min_u \int_S^T h(u, u', t)\,dt = f(a, b; S, T), \tag{1}$$

где u подчинено условиям $u(S) = a$, $u(T) = b$. Тогда, как и выше,

$$f(a, b; S, T) = \min_c [f(a, c; S, R) + f(c, b; R, T)], \quad S \leqslant R \leqslant T. \tag{2}$$

Если функция h квадратична по u и u', то f квадратична по a и b:

$$f(a, b; S, T) = a^2 r_{11}(S, T) + 2abr_{12}(S, T) + b^2 r_{22}(S, T). \tag{3}$$

Используя (2), можно получить функциональные уравнения для $r_{ij}(S, T)$.

Упражнение

Получить эти функциональные уравнения.

12. Тридиагональные матрицы

Рассмотрим задачу решения системы линейных уравнений

$$Ax = c, \tag{1}$$

где A — симметричная тридиагональная матрица[1]), т. е.

$$a_{ij} = 0, \quad |i - j| > 1. \tag{2}$$

---

[1]) То есть матрица вида

$$\begin{vmatrix} a_{11} & a_{12} & 0 & \dots & 0 \\ a_{21} & \ddots & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_{n-1,n} \\ 0 & 0 & \dots & a_{n,n-1} & a_{nn} \end{vmatrix}.$$

—*Прим. ред.*

Если матрица A положительно определена, то решение системы (1) эквивалентно минимизации квадратичной формы

$$Q(x) = (x, Ax) - 2(c, x). \tag{3}$$

Для упрощения записи введем обозначения

$$\alpha_i = a_{ii}, \quad \beta_i = a_{i, i+1} = a_{i, i-1}. \tag{4}$$

Тогда (3) можно переписать в виде

$$Q(x) = \alpha_1 x_1^2 + 2\beta_1 x_1 x_2 + \alpha_2 x_2^2 + 2\beta_2 x_2 x_3 + \ldots + \\ + 2\beta_{n-1} x_{n-1} x_n + \alpha_n x_n^2 - 2c_1 x_1 - 2c_2 x_2 - \ldots - 2c_n x_n. \tag{5}$$

Пусть $Q_k(x, z)$ задано соотношением

$$Q_k(x, z) = \alpha_1 x_1^2 + 2\beta_1 x_1 x_2 + \ldots + 2\beta_{k-1} x_{k-1} x_k + \\ + \alpha_k x_k^2 - 2c_1 x_1 - 2c_2 x_2 - \ldots - 2z x_k, \qquad k = 1, 2, \ldots. \tag{6}$$

Наконец, определим $f_k(z)$ как

$$f_k(z) = \min_{[x_1, x_2, \ldots, x_k]} Q_k(x, z). \tag{7}$$

Ясно, что поскольку

$$Q(x) = Q_n(x, c_n), \tag{8}$$

то $f_n(c_n)$ представляет собой минимум квадратичной формы $Q(x)$. Применяя только что описанные методы, легко проверить, что

$$f_k(z) = \min_{x_k} [\alpha_k x_k^2 - 2z x_k + f_{k-1}(c_{k-1} + \beta_{k-1}, x_k)], \tag{9}$$

где

$$f_1(z) = z^2/\alpha_1. \tag{10}$$

Более того, можно показать, что $f_k(z)$ является квадратичной функцией z

$$f_k(z) = r_k z^2 + 2s_k z + t_k, \tag{11}$$

что позволяет получить рекуррентные уравнения для определения коэффициентов r_k, s_k и t_k. Большая часть материала, изложенного в последующих главах, основана на матричных формулировках этих результатов.

Упражнения

1. Доказать (9).
2. Найти рекуррентные уравнения для коэффициентов r_k, s_k и t_k. Использовать эти результаты для вывода рекуррентного уравнения для x_k.

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Беллман (Bellman R.)

Introduction to the mathematical theory of control processes 1: Linear equations and quadratic criteria, Academic Press, New York, 1967.

Раздел 7. См. статью

Беллман (Bellman R.)

Functional equations in the theory of dynamic programming XVII: Minimum convolutions and Green's functions, *J. Math. Anal. Appl.*, 33 (1971), 497—499.

Раздел 8. Задача вычисления выражения $(2)^N$ с наименьшим числом умножений является нерешенной математической проблемой. Некоторые частные результаты приведены в работах

Като (Kato H.)

On addition chains, Ph. D. Thesis, Univ. of Southern California, 1970.

Кнут (Knuth D. E.)

The art of computer programming II: Semi-Numerical algorithms, Addison — Wesley, Reading, Massachusetts, 1969.

Раздел 12. См. книгу

Беллман Р.

Введение в теорию матриц, «Наука», М., 1969.

Леман (Lehman R. S.)

Dynamic programming and Gaussian elimination, *J. Math. Anal. Appl.*. 5 (1962), 499—501.

Глава 4

Уравнения эллиптического типа

1. Введение

Рассмотрим в этой главе некоторые вопросы, связанные с решением уравнения Лапласа:

$$u_{xx} + u_{yy} = 0, \qquad (x, y) \in R, \tag{1}$$

$$u = g(x, y), \quad (x, y) \in \Gamma, \tag{2}$$

где Γ — граница области R (рис. 1). Мы хотим выявить связь этой

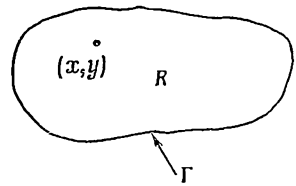


Рис. 1

задачи с задачей минизации квадратичного функционала

$$D(u) = \int_R (u_x^2 + u_y^2)\, dR, \tag{3}$$

обычно называемого функционалом Дирихле, и обсудить ряд других родственных вопросов.

Во многих случаях задача минимизации такого квадратичного функционала может быть эффективно решена с помощью методов типа Рэлея — Ритца, и это представляет собой один из итерационных способов решения уравнения Лапласа [1]). Мы будем пользоваться иным подходом, основанным на дискретизации,— подходом, обладающим в некоторых случаях определенными преимуществами.

Кроме того, коснемся функций Грина, связанных с неоднородным уравнением

$$u_{xx} + u_{yy} = h(x, y) \tag{4}$$

и уравнением более общего типа

$$u_{xx} + u_{yy} + k\ x, y)\ u = h(x, y). \tag{5}$$

---

[1]) См. примечание на стр. 132.— *Прим. ред.*

2. Уравнение Эйлера

Легко можно показать, что уравнение Лапласа является вариационным уравнением, связанным с $D\ u)$. Положим $v = 0$ на Γ, так что функции u и $u + v$ удовлетворяют на Γ одним и тем же граничным условиям [1]). Тогда

$$D(u+v) = D(u) + D(v) + 2\int_R (u_x v_x + u_y v_y)\, dR. \qquad (1)$$

Применив теорему Грина, получаем, что третий член в правой части (1) исчезает, если u удовлетворяет уравнению Лапласа. Следовательно, если u удовлетворяет (1.1), то справедливо неравенство

$$D(u + v) > D(u) \qquad (2)$$

для любого $v \not\equiv 0$. Таким образом, если (1.1) и (1.2) обладают решением, то оно единственно, поскольку любое решение минимизирует $D(u)$ [2]).

С другой стороны, можно показать, что $D\ u)$ достигает минимума на классе функций u, таких, что $u_x,\ u_y \in L^2$, и u удовлетворяет заданному граничному условию. Далее можно показать, что минимизирующая функция определяется уравнениями (1.1) и (1.2).

Таким образом, задачи решения уравнения Лапласа и минимизации функционала Дирихле $D(u)$ эквивалентны. Поэтому мы сконцентрируем внимание на задаче минимизации.

3. Неоднородный и нелинейный случай

Из предыдущего следует, что задача решения неоднородного уравнения

$$u_{xx} + u_{yy} = h(x, y), \quad u(x, y) = 0, \quad (x, y) \in \Gamma, \qquad (1)$$

эквивалентна задаче минимизации функционала

$$D_1(y) = \int_R (u_x^2 + u_y^2 + 2h(x, y)\, u)\, dR \qquad (2)$$

при ограничениях $u = 0$ на Γ и $u_x,\ u_y \in L^2(R)$.

Аналогично, задача решения уравнения

$$u_{xx} + u_{yy} + K(x, y)\, u = 0 \qquad (3)$$

---

[1]) Предполагается, что u доставляет минимум функционалу $D(u)$, а v есть ее вариация, принадлежащая тому же множеству функций, что и функция u.— *Прим. ред.*

[2]) Функционал $D(u)$ достигает минимума в единственной точке, из чего и следует единственность решения.— *Прим. перев.*

эквивалентна задаче минимизации

$$D_2(u) = \int_R (u_x^2 + u_y^2 - K(x, y)\, u^2)\, dR, \tag{4}$$

если $D_2(u)$ — положительно определенный функционал. Достаточным условием для этого является выполнение неравенства $\max_R |K(x, y)| < \lambda_1$, где λ_1 — минимальное характеристическое число задачи

$$u_{xx} + u_{yy} + \lambda u = 0, \quad u = 0, \quad (x, y) \in \Gamma.$$

Наконец, решению нелинейного уравнения

$$u_{xx} + u_{yy} - h(u) = 0 \tag{5}$$

соответствует задача минимизации функционала

$$\int_R (u_x^2 + u_y^2 + 2g(u))\, dR, \tag{6}$$

где $g'(u) = h(u)$. Этот случай будет подробнее рассмотрен в главе, посвященной квазилинеаризации.

4. Функция Грина

Решение неоднородного уравнения

$$u_{xx} + u_{yy} = h(x, y), \quad u = 0, \quad (x, y) \in \Gamma \tag{1}$$

можно представить в виде

$$u = \int_R K(x, y, x_1, y_1)\, h(x_1, y_1)\, dR. \tag{2}$$

Ядро K называется *функцией Грина* для данного уравнения и заданного граничного условия.

Известно, что в предыдущем случае $K \leqslant 0$,— результат, которым мы впоследствии воспользуемся. Однако выведем это фундаментальное свойство непосредственно из соответствующей вариационной задачи.

5. Одномерный случай

Для иллюстрации метода доказательства начнем с простейшего случая одной переменной. Покажем, что решение задачи

$$u'' + g(t)\, u = h(t), \quad u(0) = u(T) = 0, \tag{1}$$

неотрицательно на отрезке $[0, T]$, если $h(t)$ отрицательна и квадратичный функционал

$$J(u) = \int_0^T (u'^2 - g(t)\, u^2)\, dt \tag{2}$$

положительно определен для функций, удовлетворяющих описанным выше граничным условиям.

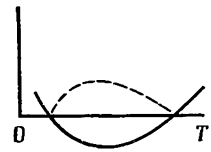
Если это утверждение справедливо, то (1) является уравнением Эйлера, соответствующим минимизации функционала

$$J_1(u) = \int_0^T [u'^2 - g(t)\,u^2 + 2h(t)\,u]\,dt. \tag{3}$$

Решение задачи (1) доставляет абсолютный минимум функционалу (3) на классе функций, удовлетворяющих граничным условиям и таких, что $u' \in L^2(0, T)$.

Предположим, что сформулированное утверждение неверно, т. е. что непрерывная функция u отрицательна на некотором отрезке $[a, b]$ (см. рис. 2). Здесь может быть $a = 0$ или $b = T$.

Рис. 2



Рассмотрим новую функцию v, определенную как

$$\begin{aligned} v &= u, \qquad t \notin [a, b], \\ v &= -u, \qquad t \in [a, b]. \end{aligned} \tag{4}$$

Эта новая функция, возможно, имеет разрывную производную при $t = a$ и $t = b$, но тем не менее она такова, что $v' \in L^2(0, T)$ и v удовлетворяет граничным условиям.

Запишем

$$J_1(u) = \int_0^T = \int_a^b + \int_S, \tag{5}$$

где $S = [0, T] - [a, b]$. Ясно, что интеграл по S не меняется при замене u на v. С другой стороны, поскольку $h(t) \leqslant 0$, мы видим, что интеграл по $[a, b]$ уменьшился. Таким образом,

$$J_1(v) < J_1(u). \tag{6}$$

Это противоречит тому, что u доставляет *абсолютный* минимум функционалу J_1. Поэтому $u \geqslant 0$ на $[0, T]$.

Поскольку

$$u(t) = \int_0^T K(t, t_1)\,h(t_1)\,dt_1 \tag{7}$$

и $h(t)$ — произвольная отрицательная функция, то из неотрицательности u вытекает, что функция Грина $K(t, t_1)$ неположительна при $0 \leqslant t,\ t_1 \leqslant T$, что и требовалось доказать.

Упражнения

1. Доказать, что в действительности $K(t, t_1)$ отрицательна при $0 < t, t_1 < T$.

2. Посредством аналогичных рассуждений показать, что $K(t, t_1)$ обладает интересным вариационным свойством; см. Беллман Р. (Bellman R.) On variation-diminishing properties of Green's functions, *Boll. Un. Mat. Ital.*, **16** (1961), 164—166.

3. Показать, что тех же результатов можно ожидать, если рассмотреть задачу минимизации функционала

$$J_3(u) = \int_0^T [u'^2 + (u - h)^2]\, dt.$$

Почему это можно интерпретировать как «сглаживающую» операцию?

4. Провести предыдущие рассуждения для уравнения

$$u'' - u - 2u^3 = h(t), \quad u(0) = u(T) = 0.$$

5. Что можно сказать о решении уравнения

$$u'' - e^{-u} = h(t), \quad u(0) = u(T) = 0,$$

если $h(t) \geqslant 0$?

6. Показать, что подобными рассуждениями можно получить аналогичный результат для разностного уравнения

$$u_{n+1} - 2u_n + u_{n-1} + q_n u_n = h_n, \quad n = 1, 2, \ldots, N-1,$$
$$u_0 = u_N = 0,$$

соответствующего минимизации функционала

$$\sum_{n=0}^{N-1} [(u_{n+1} - u_n)^2 - q_n u_n^2 + 2h_n u_n].$$

6. Двумерный случай

Тот же способ доказательства можно использовать и для более высокой размерности. Рассмотрим двумерный случай. Пусть

$$D(u) = \int_R [u_x^2 + u_y^2 - q(x, y)\, u^2 + 2h(x, y)\, u]\, dR \tag{1}$$

есть квадратичный функционал, соответствующий уравнению

$$u_{xx} + u_{yy} + q(x, y)\, u = h(x, y), \quad u = 0, \quad (x, y) \in \Gamma. \tag{2}$$

Предположим, что квадратичная часть $D(u)$ положительно определена.

Мы стремимся показать при этом предположении, что из $h \geqslant 0$ следует неравенство $u \leqslant 0$. Предположим противное. Пусть $u \geqslant 0$ в $R_1 \in R$; рассмотрим новую функцию

$$v = -u, \qquad p \in R_1, \tag{3}$$
$$v = u, \qquad p \in R - R_1.$$

Как и ранее, приходим к соотношению $D(v) < D(u)$, т. е. к противоречию. Таким образом, из $h \geqslant 0$ вытекает, что $u \leqslant 0$, следовательно, функция Грина неположительна.

Упражнения

1. Показать, что в действительности функция Грина отрицательна для внутренних точек R.
2. Каково соответствующее вариационное свойство функции Грина?
3. Распространить этот результат на нелинейные уравнения, как в одномерном случае.
4. Каков аналог результата упражнения 6 в конце разд. 5?

7. Дискретизация

Одним из мощных методов изучения свойств дифференциальных уравнений является анализ соответствующих разностных уравнений. Такой подход оказывается особенно эффективным с вычислительной точки зрения, когда мы намереваемся использовать аналоговые или цифровые вычислительные машины. Тем не менее он столь же ценен и в теоретическом плане. Основная идея заключается в том, что с помощью аппроксимации производных

$$u_{xx}(x, y) \simeq \frac{u(x+\Delta, y)+u(x-\Delta, y)-2u(x, y)}{\Delta^2},$$
$$u_{yy}(x, y) \simeq \frac{u(x, y+\Delta)+u(x, y-\Delta)-2u(x, y)}{\Delta^2} \tag{1}$$

уравнение Лапласа сводится к системе линейных алгебраических уравнений. Разумеется, такое преобразование является лишь первым этапом всей проблемы, состоящей в получении полезных аналитических и численных результатов из построенной линейной системы.

8. Прямоугольная область

Посмотрим, что получается для случая прямоугольной области (рис. 3). Выберем целые числа M и N и определим величины Δ и δ как $M\Delta = a$, $N\delta = b$. Положим

$$u_{mn} = u(m\Delta,\ n\delta),\quad m = 0,\ 1,\ \ldots,\ M,\quad n = 0,\ 1,\ \ldots,\ N. \tag{1}$$

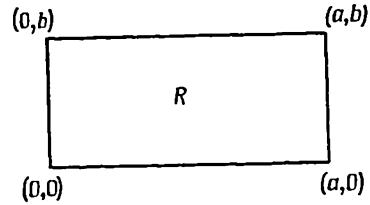


Рис. 3

Исходное граничное условие

$$u(x, y) = g(x, y),\quad (x, y) \in \Gamma \tag{2}$$

приводит к соответствующим граничным условиям

$$\begin{aligned} u_{0n} &= g(0,\ n\delta), \qquad n = 0,\ 1,\ \ldots,\ N,\\ u_{Mn} &= g(a,\ n\delta),\\ u_{m0} &= g(m\Delta,\ 0), \qquad m = 0,\ 1,\ \ldots,\ M,\\ u_{mN} &= g(m\Delta,\ b). \end{aligned} \tag{3}$$

Будем предполагать, что $g(x, y)$ непрерывна, так что $g(0, 0)$ и $g(m\Delta, N\delta)$ однозначно определяются предыдущими соотношениями.

Положив $x = m\Delta$ и $y = n\delta$, запишем уравнение Лапласа в виде системы соотношений

$$\frac{u((m+1)\Delta, n\delta) + u((m-1)\Delta, n\delta) - 2u(m\Delta, n\delta)}{\Delta^2} +$$
$$+ \frac{u(m\Delta,\ (n+1)\delta) + u(m\Delta,\ (n-1)\delta) - 2u(m\Delta,\ n\delta)}{\delta^2} \simeq 0. \tag{4}$$

Таким образом, мы получаем линейные разностные уравнения

$$\frac{v_{m+1,\,n} + v_{m-1,\,n} - 2v_{mn}}{\Delta^2} + \frac{v_{m,\,n+1} + v_{m,\,n-1} - 2v_{mn}}{\delta^2} = 0, \tag{5}$$
$$m = 0,\ 1,\ \ldots,\ M,\qquad n = 0,\ 1,\ \ldots,\ N.$$

Эти соотношения совместно с (3) составляют систему линейных алгебраических уравнений относительно величин v_{mn}.

9. О корректности аппроксимации

Если Δ и δ достаточно малы, то вполне вероятно, что

$$u(m\Delta, n\delta) \simeq v_{mn}. \tag{1}$$

Это можно показать строго; см. ссылки в конце этой главы.

Упражнение

Показать, что если u обладает производными третьего порядка, то предыдущий результат гарантирует устойчивость решения уравнений (8.5).

10. Соответствующая задача минимизации

Необходимо исследовать вопрос о существовании и единственности решения линейной системы (8.5). Ответ на этот вопрос легко получить, если заметить, что эти линейные разностные уравнения являются вариационными уравнениями для квадратичной формы

$$Q_{M,N}(v) = \sum_{m,n} \left[\left(\frac{v_{m+1,n} - v_{m,n}}{\Delta} \right)^2 + \left(\frac{v_{m,n+1} - v_{m,n}}{\delta} \right)^2 \right], \tag{1}$$

где $m = 0, 1, \ldots, M$ и $n = 0, 1, \ldots, N$. Граничные значения определяются, как в (8.3). Поскольку $Q_{M,N}(v)$, очевидно, является положительно определенной, то она обладает единственным минимумом, достигающемся на единственном решении системы (8.5).

11. Аппроксимация сверху

Дискретная задача минимизации (разд. 10) была получена из непрерывной (минимизация $D(u)$) посредством сужения класса допустимых функций u до таких, что u_x и u_y постоянны на прямоугольниках

$$m\Delta \leqslant x \leqslant (m+1)\Delta, \quad n\delta \leqslant y \leqslant (n+1)\delta. \tag{1}$$

Естественно поэтому, что минимум по такому подклассу функций больше или равен минимуму по классу функций, таких, что u_x и u_y принадлежат $L^2(R)$. Следовательно, для любых $\Delta, \delta > 0$, получаем

$$\min_v Q_{M,N}(v) \geqslant \min_u D(u). \tag{2}$$

12. Обсуждение

Если мы хотим получить хорошее приближение для $u(m\Delta, n\delta)$, то величины Δ и δ следует выбрать достаточно малыми. Однако при этом мы придем к системе линейных алгебраических уравне-

ний большой размерности. Задача численного решения таких систем весьма трудоемкая для стандартных методов, и результаты при этом не вполне надежны.

К счастью, специфическая структура таких систем в ряде случаев позволяет воспользоваться мощными и остроумными методами. Эти методы, как и все методы вообще, обладают своими преимуществами и недостатками. Мы же будем использовать здесь совершенно иной подход.

13. Частичная дискретизация

Иногда оказывается удобным проводить дискретизацию только по одной переменной. Так, например, можно построить одномерную сетку только в направлении y. Уравнение Лапласа тогда принимает вид

$$w_{xx}^{(m)} = \frac{w^{(m-1)} + w^{(m+1)} - 2w^{(m)}}{\Delta^2}, \tag{1}$$

где $w^{(m)} = w(x, m\Delta)$.

Для области R_1, где $0 \leqslant x \leqslant a$, $0 \leqslant y \leqslant b$, целое число m пробегает ряд значений от 1 до $M - 1$ с двухточечными граничными условиями

$$w^{(m)}(0) = a_m, \quad w^{(m)}(a) = b_m, \quad w^{(0)}(x) = g(x), \quad w^{(M)}(x) = h(x). \tag{2}$$

$$m = 0, 1, \ldots, M, \quad 0 \leqslant x \leqslant a.$$

Величины a_m, b_m, $g(x)$ и $h(x)$ непосредственно определяются из (8.2).

Решение системы линейных дифференциальных уравнений такой структуры можно получить, решив системы линейных алгебраических уравнений размерности M. Соответствующая вариационная задача состоит в минимизации функционала

$$\int_0^a \left[\sum_{m=1}^{M-1} (w_x^{(m)})^2 + \sum_{m=1}^{M-1} \left(\frac{w^{(m+1)} - w^{(m-1)}}{\Delta} \right)^2 \right] dx. \tag{3}$$

Положительная определенность этого функционала гарантирует существование и единственность решения задачи (1) — (2).

14. Неравномерная сетка

Конечно, вовсе не обязательно использовать равномерную сетку. Пусть y принимает ряд значений

$$y = y_0, y_1, \ldots, y_{n-1}, y_n, \tag{1}$$

где $0 = y_0 < y_1 < y_2 < \ldots < y_{n-1} < y_n = a$. Рассмотрим «квадратурную» формулу

$$u_{yy} \simeq \sum_{j=0}^{n} w_j u(x, y_j) \qquad y = y_0, y_1, \ldots, y_n. \tag{2}$$

Для определения величин w_i и y_i можно, например, потребовать, чтобы эта формула была точной для полиномов степени, меньшей или равной n. С помощью такой аппроксимации уравнение Лапласа приводится к виду

$$(z_i)_{xx} + \sum_j w_j z_j = 0. \tag{3}$$

И здесь мы сталкиваемся с двухточечной граничной задачей. Преимущество такого подхода тем не менее заключается в том, что порядок дифференциального уравнения можно выбрать значительно меньшим, чем порядок уравнения из разд. 13.

Кроме полиномиальной, можно также использовать аппроксимацию сплайнами [1]).

Упражнение

Существует ли для этого случая соответствующая вариационная задача?

15. Решение разностных уравнений

Линейные разностные (или конечно-разностные) уравнения (8.5) можно записать в векторно-матричной форме

$$Av = b, \tag{1}$$

где v — вектор неизвестных $\{v_{ij}\}$, размерности $(M - 1)(N - 1)$, во внутренней точке

$$v = \begin{vmatrix} v_{1,1} \\ \vdots \\ v_{1,N-1} \\ v_{2,1} \\ \vdots \\ v_{M-1,N-1} \end{vmatrix}. \tag{2}$$

Матрица A определяется конечно-разностной аппроксимацией (8.4) и сеткой дискретизации, а вектор b — граничными условиями. Структура матрицы A обладает многими специальными свойствами, которые мы детально рассмотрим в соответствующих главах. Пока лишь отметим, что A не вырождена, что позволяет запи-

---

[1]) О сплайнах см., например, Дж. Алберг, Э. Нильсон, Дж. Уолш, «Теория сплайнов и ее приложения», изд-во «Мир», М., 1972.— *Прим. ред.*

сать v в виде

$$v = A^{-1}b. \tag{3}$$

Сравнивая это равенство с (4.2), видим, что A^{-1} является дискретным аналогом функции Грина. Все элементы A^{-1} положительны [1]), т. е. такая дискретная форма функции Грина обладает свойством, аналогичным неотрицательности $K(t, t_1)$. Доказательство аналогично проведенному для непрерывного случая.

Если желательно получить хорошее приближение к решению, то размерность матрицы A становится весьма большой, поэтому точные методы решения системы (1), такие, как метод Гаусса, практически непригодны. В связи с этим будем существенно использовать блочную тридиагональную структуру A. Возвращаясь к (8.5), мы видим, что любая строка матрицы A содержит не более пяти ненулевых элементов. Благодаря такой слабой заполненности матрицы при численном решении системы (1) итерационные методы могут оказаться весьма эффективными.

16. Метод итераций

Линейный итерационный процесс решения линейной системы $v = Dv + f$ можно записать в виде

$$v^{(k+1)} = Dv^{(k)} + f, \tag{1}$$

где $v^{(0)}$ — некоторое начальное приближение. Если последовательность $\{v^{(k)}\}$ сходится, то она сходится к решению системы линейных уравнений

$$[I - D]\, v = f. \tag{2}$$

Таким образом, если существует невырожденная матрица T, такая, что

$$T\,[I - D] = A, \qquad Tf = b, \tag{3}$$

то, когда последовательность $\{v^k\}$ сходится, мы получим решение (15.1). Поскольку на каждой итерации требуется одно умножение вектора на матрицу D, ограничимся такими матрицами A и T, для которых D слабо заполнена.

Для получения условий сходимости процесса (1) определим погрешность решения как

$$e^{(k)} = v - v^{(k)}. \tag{4}$$

Используя (1) и (2), получаем

$$\begin{aligned} e^{(k+1)} &= v - v^{(k+1)} = \\ &= v - Dv^{(k)} - f = \\ &= v - Dv^{(k)} - [I - D]\, v, \end{aligned} \tag{5}$$

[1]) Имеются в виду ненулевые элементы A^{-1}.— *Прим. перев.*

следовательно,

$$e^{(k+1)} = De^{(k)} \tag{6}$$

и

$$e^{(k)} = D^k e^{(0)}. \tag{7}$$

Таким образом, итерационный процесс (1) сходится при всех $e^{(0)}$ тогда и только тогда, когда все собственные значения матрицы D по модулю меньше единицы. Пусть λ_k есть k-е собственное значение D. Определим спектральный радиус $\rho(D)$ матрицы D как

$$\rho(D) = \max_k |\lambda_k|. \tag{8}$$

Тогда процесс (1) сходится в том и только том случае, когда

$$\rho(D) < 1. \tag{9}$$

Упражнения

1. Пусть A и b определяются из (8.5). Показать, что если $T = I$, то матрица D удовлетворяет (8.3). Так можно ввести метод Якоби.

2. Показать, что для выполнения одной итерации по методу Якоби требуется $O(NM)$ умножений.

17. Возможности итерационного подхода

Необходимо выяснить, каким образом матрица D и вектор f определяются из A и b. Хотя в основном мы будем иметь дело с конечными методами, некоторые из различных итерационных методов мы обсудим в гл. 6. Однако на данном этапе надо ясно представлять себе возможности итерационных методов.

Для простых методов, таких, как метод Якоби или Гаусса — Зейделя, матрица D определяется непосредственно по матрице A. Таким образом, сходимость этих методов зависит лишь от исходного дифференциального уравнения и способа дискретизации. Можно показать, что для широкого класса уравнений, в том числе и для уравнения Лапласа, эти простые методы сходятся при любой величине шага. Однако, к сожалению, сходимость их становится очень медленной при уменьшении шага, что существенно сказывается на точности решения.

В более сложных итерационных методах, таких, как метод последовательной сверхрелаксации или методы чередующихся направлений, требуется определить один или более чередующихся выбираемых для сужения спектрального радиуса матрицы D насколько это возможно. Для уравнений, отличных от уравнения Лапласа на прямоугольной сетке, определение этих параметров является сложной аналитической и вычислительной задачей.

С учетом действительной стоимости решения данной задачи на машине может оказаться неразумным затрачивать определенные усилия для точного определения этих параметров. Однако если мы находимся в окрестности оптимума, отыскание оптимальных значений параметров не столь важно, и можно получить достаточно хорошую их оценку, как побочный результат предыдущих итераций. Более того, методы, подобные неявной схеме чередующихся направлений, могут даже расходиться, если интересующая нас область не прямоугольная, а более общего вида.

Когда мы обратимся к методу квазилинеаризации и проблеме идентификации, то обнаружим некоторые иные причины, побуждающие предпочесть конечные методы итерационным. Однако причина такого предпочтения покоится, вообще говоря, единственно на той основе, что точные методы легки для понимания, просты для программирования и, как правило, безотказны. В эру, когда быстродействие вычислительных машин выражается в микросекундах, а скоростные запоминающие устройства могут хранить сотни тысяч машинных слов, простота использования численных методов даже для научного работника и инженера, а не только для специалиста-вычислителя, приобретает первостепенное значение.

Разные упражнения

1. Пусть $L(u)$ — линейный функционал, непрерывный по u при $u \in L^2(0, 1)$. Рассмотреть задачу минимизации квадратичного функционала $\int_0^1 u^2\,dt + L(u)$ и с ее помощью установить теорему представления Рисса: $L(u) = \int_0^1 uv\,dt$ для некоторой функции $v \in L^2(0, 1)$.

2. Рассмотрим уравнение $u'' = 0$, $u(0) = a$, $u(T) = b$ и соответствующее разностное уравнение $u_{n+1} + u_{n-1} - 2u_n = 0$, $n = 1, 2, \ldots, N-1$, $u_0 = a$, $u_N = b$. Рассмотрим метод последовательных приближений

$$u_n^{(k)} = \frac{u_{n+1}^{(k-1)} + u_{n-1}^{(k-1)}}{2}, \qquad u_0^{(k)} = a, \qquad u_N^{(k)} = b.$$

Сходится ли $\{u_n^{(k)}\}$ при $k \to \infty$? Монотонно?

3. Рассмотреть уравнения (8.4) для $\Delta = \delta$. Показать, что их можно привести к виду

$$v_{m,n} = \frac{1}{4}(v_{m+1,n} + v_{m-1,n} + v_{m,n+1} + v_{m,n-1}).$$

4. Рассмотреть метод последовательных приближений

$$v_{m,n}^{(k)} = \frac{1}{4}\,(v_{m+1,n}^{(k-1)} + \ldots).$$

Сходится ли $\{v_{m,n}^{(k)}\}$ при $k \to \infty$? Монотонно?

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Раздел 1. Задача Дирихле и вариационное исчисление рассматриваются в ряде книг; см., например,

Курант Р.

Принцип Дирихле, конформные отображения и минимальные поверхности, ИЛ. М., 1953.

Курант Р., Гильберт Д.

Методы математической физики, М., 1951.

Ахиезер Н. И.

Лекции по вариационному исчислению, Гостехиздат, М., 1955.

Гельфанд И. М., Фомин С. В.

Вариационное исчисление, Физматгиз, М., 1961.

Относительно других подходов см.

Беллман (Bellman R.)

Methods of nonlinear analysis, Academic Press, New York, 1970.

Варга (Varga R.)

Accurate numerical methods for nonlinear boundary value problems, *SIAM — AMS Proc.*, 2 (1970).

Разделы 4—6. См. книгу

Беллман (Bellman R.)

Introduction to the mathematical theory of control processes I: Linear equations and quadratic criteria, Academic Press, New York, 1967.

Раздел 8. См. статью

Брэмбл, Хаббэрд, Томе (Bramble J. H., Hubbard B. E., Thomée V.)

Math. Comp., 23 (1969), 695—710.
Там же приведена дополнительная литература.

Раздел 9. Доказательства сходимости см. в работе

Айзексон, Келлер (Isaacson E., Keller H. B.)

Analysis of numerical methods, Wiley, New York, 1966.

Дополнительную литературу можно найти в главе, написанной Д. Янгом, из книги

Тодд (Todd J., ed.)

Survey of numerical analysis, McGraw-Hill, New York, 1962.

См. также

Рихтмайер Р. и Мортон К.

Разностные методы решения краевых задач, «Мир», М., 1972.

Коллатц Л.

Численные методы решения дифференциальных уравнений, ИЛ, М., 1953.

Коллатц Л.

Функциональный анализ и вычислительная математика, «Мир», М. 1969.

Раздел 10. Это пример аппроксимации в пространстве политик.

Раздел 13. Метод частичной дискретизации известен также под названием метода прямых. Он очень популярен в литературе на русском языке. Читатель должен иметь в виду, что фактически все методы, описанные в дальнейшем, можно также представить в терминах метода прямых, а не метода конечно-разностной аппроксимации; см.

Канторович Л. В., Крылов В. И.

Приближенные методы высшего анализа, Физматгиз, М.— Л., 1962.

Раздел 14. См. статью

Беллман, Касти (Bellman R., Casti J.)

Differential quadrature and long term integration, *J. Math. Anal. Appl.*, 34 (1971), 235—238.

Раздел 16. Этот подход применил Р. Варга для анализа ряда стандартных методов; см.

Варга (Varga R.)

Matrix iterative analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

Ральстон, Вилф (Ralston A., Wilf H. S.)

Mathematical methods for digital computers, v. 1, 2, Wiley, New York, 1965, 1967.

Раздел 17. Д. Тодд отмечает, что неявные методы чередующихся направлений работают только в тех случаях, когда некоторые матрицы коммутативны и мы находимся в стандартной академической ситуации. На практике, однако, ситуация часто оказывается благоприятной.

Глава 5

Динамическое программирование и эллиптические уравнения

1. Уравнение Лапласа

В этой главе мы изложим один из подходов к численному решению некоторых типов эллиптических уравнений, основанный на теории динамического программирования. Мы начинаем с задачи, поставленной в гл. 4,— решения уравнения Лапласа

$$\nabla^2 u = u_{xx} + u_{yy} = 0 \tag{1}$$

на области R с граничными условиями

$$u(x, y) = f(x, y), \tag{2}$$

заданными на Г, границе области R.

Мы уже видели, что (1) является уравнением Эйлера для задачи минимизации квадратичного функционала

$$D(u) = \int\int_R [u_x^2 + u_y^2]\, dR \tag{3}$$

при ограничениях (2).

Вначале будем считать, что область R представляет собой прямоугольник $0 \leqslant x \leqslant a$, $0 \leqslant y \leqslant b$.

В дальнейшем мы будем трактовать некоторые нерегулярные области как комбинации прямоугольных областей.

2. Дискретизация

Наш метод будет состоять в определении минимума дискретного аналога функционала Дирихле, соответствующего исходному уравнению Лапласа. Для удобства будем считать величины a и b такими, что

$$a = Nh, \qquad b = Mh, \tag{1}$$

где M и N — целые числа. Если это не так, то соответствующие изменения легко сделать, и они не повлияют на дальнейшее. Нас интересуют значения функции в точках

$$u_{ij} = u(ih, jh), \quad i = 0, 1, \ldots, N, \quad j = 0, 1, \ldots, M. \tag{2}$$

Поскольку мы хотим получить решение, определенное только в этих $(N+1)(M+1)$ точках, то удобно заменить частные производные в этих точках выражениями, включающими только значения функции в этих точках. Используя разложение в ряд Тейлора, получаем стандартную конечно-разностную аппроксимацию

$$\frac{\partial u_{ij}}{\partial x} = \frac{u_{ij} - u_{i-1,\,j}}{h} + O(h), \quad \frac{\partial u_{ij}}{\partial y} = \frac{u_{ij} - u_{i,\,j-1}}{h} + O(h). \qquad (3)$$

Таким образом, для получения дискретного аналога функционала (1.3) мы заменяем частные производные выражениями (3) и предполагаем, что эти производные постоянны между точками сетки. Точно так же мы могли бы проинтегрировать (1.3) по прямоугольникам сетки и затем использовать (3). В любом случае двойной интеграл в функционале Дирихле можно аппроксимировать двойной суммой

$$J(u) = \sum_{i=1}^{N} \sum_{j=1}^{M} [(u_{ij} - u_{i,\,j-1})^2 + (u_{ij} - u_{i-1,\,j})^2]. \qquad (4)$$

Легко проверить, что ошибка дискретизации составляет $O(h^2)$. Функционал (4) должен удовлетворять исходным граничным условиям, и поэтому значения u_{0j}, u_{i0}, u_{Nj} и u_{iM} определяются уравнением (1.2).

Мы пришли к конечномерной задаче определения минимума:

$$\min_{\{u_{ij}\}} J(u), \qquad i = 1, 2, \ldots, N-1, \quad j = 1, 2, \ldots, M-1. \qquad (5)$$

Должно быть ясно, что поскольку нас интересуют лишь значения $\{u_{ij}\}$, минимизирующие функционал, а не само минимальное значение $J(u)$, то можно убрать из функционала (4) постоянные члены и при этом множество минимизирующих значений $\{u_{ij}\}$ не изменится. Удобно опустить в (4) все члены вида $(u_{iM} - u_{i-1,\,M})^2$, которые постоянны в силу граничных условий. Поступая таким образом, заменим (4) более удобным функционалом

$$J(u) = \sum_{i=1}^{N} \Big[\sum_{j=1}^{M} (u_{ij} + u_{i,\,j-1})^2 + \sum_{j=1}^{M-1} (u_{ij} - u_{i-1,\,j})^2 \Big]. \qquad (6)$$

3. Векторно-матричная формулировка

Хотя исходная задача теперь полностью дискретна, она все еще не представлена в том виде, в котором может быть легко решена. Прежде всего запишем эту задачу в векторно-матричной форме, а затем представим ее в виде многошагового процесса, к которому можно применить метод динамического программирования.

Поскольку функционал является квадратичной формой, можно легко переписать задачу, используя скалярные произведения. Сначала определим векторы для внутренних точек

$$u_R = \begin{vmatrix} u_{R1} \\ u_{R2} \\ \vdots \\ u_{R,\,M-1} \end{vmatrix}, \qquad R = 1, 2, \ldots, N-1, \tag{1}$$

и заметим, что векторы u_0 и u_N задаются граничными условиями. Теперь определим симметричную матрицу Q размера $M-1$, множество $(M-1)$-мерных векторов r_R и скаляры S_R следующим образом:

$$Q = (q_{ij}), \quad \text{где} \quad q_{ij} \begin{cases} 2, & i = j, \\ -1, & |i-j| = 1, \\ 0 & \text{в остальных случаях}, \end{cases} \tag{2}$$

$$r_R = [r_{Rj}], \quad \text{где} \quad r_{Rj} = \begin{cases} u_{R0}, & j = 1, \\ u_{RM}, & j = M-1, \\ 0 & \text{в остальных случаях}, \end{cases}$$

$$s_R = u_{R0}^2 + u_{RM}^2.$$

Ясно, что Q — постоянная матрица, в то время как r_R и s_R — функции, зависящие лишь от граничных условий. В этих обозначениях $J(u)$ выражается через скалярные произведения следующим образом:

$$J(u) = \sum_{R=1}^{N} [(Qu_R, u_R) - (2r_R, u_R) + s_R + (u_R - u_{R-1}, u_R - u_{R-1})]. \tag{3}$$

Теперь минимизация проводится по векторам u_R $(R = 1, \ldots, N-1)$. При этом исходные граничные условия сводятся к условиям, заданным в двух точках.

4. Динамическое программирование

Вместо исходной задачи рассмотрим последовательность вариационных задач

$$f_R(v) = \min_{[u_R,\, u_{R+1},\, \ldots,\, u_{N-1}]} \sum_{i=R}^{N} [(Qu_i, u_i) - (2r_i, u_i) + s_i + (u_i - u_{i-1}, u_i - u_{i-1})], \qquad R = 1, 2, \ldots, N-1, \tag{1}$$

где v определено как

$$v = u_{R-1}. \tag{2}$$

Сравнивая с (3.3), очевидно, получаем

$$f_1(u_0) = \min_{\{u_R\}} J(u). \tag{3}$$

Каждую из задач в (1) можно рассматривать как задачу решения уравнения Лапласа на усеченном прямоугольнике с левосторонним граничным условием v, как на рис. 1. В этом и состоит фундаментальная идея погружения.

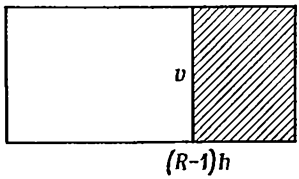


Рис. 1

Найдем теперь связь между функциями $f_R(v)$ и $f_{R+1}(v)$. Замечая, что только первые четыре члена из (1) зависят от u_R, можно записать:

$$f_R(v) = \min_{[u_R,\, u_{R+1},\, \ldots,\, u_{N-1}]} [(Qu_R,\, u_R) - 2(r_R,\, u_R) + s_R + (u_R - v,\, u_R - v) + \sum_{i=R+1}^{N} \{(Qu_i,\, u_i) - 2(r_i,\, u_i) + s_i + (u_i - u_{i-1},\, u_i - u_{i-1})\}]. \tag{4}$$

Первые четыре члена можно вынести из-под знака минимума по $u_{R+1}, u_{R+2}, \ldots, u_{N-1}$, что приводит к выражению

$$f_R(v) = \min_{u_R} \{(Qu_R,\, u_R) - 2(r_R,\, u_R) + s_R + (u_R - v,\, u_R - v) + \min_{[u_{R+1},\, u_{R+2},\, \ldots,\, u_{N-1}]} [\sum_{i=R+1}^{N} (Qu_i,\, u_i) - 2(r_i,\, u_i) + s_i + (u_i - u_{i-1},\, u_i - u_{i-1})]\}. \tag{5}$$

Однако сравнивая члены, стоящие в квадратных скобках, с определением $f_R(v)$ из (1), мы видим, что (5) можно переписать в виде

$$f_R(v) = \min_{u_R} [(Qu_R,\, u_R) - 2(r_R,\, u_R) + s_R + (u_R - v,\, u_R - v) + f_{R+1}(u_R)]. \tag{6}$$

Этот результат можно рассматривать и как следствие принципа оптимальности. Выбор u_R должен сбалансировать цену данного состояния (первые четыре члена из (5)) с ценой продолжения процесса, начинающегося из u_R и проходящего через $N - R$ дополнительных состояний.

Поскольку в силу граничных условий u_N фиксировано, то

$$f_N(v) = (Qu_N,\ u_N) - 2(r_N,\ u_N) + s_N + (u_N - v,\ u_N - v). \tag{7}$$

Теоретически, теперь можно определить минимизирующую последовательность $\{u_R\}$, начиная с (7) и решая (6) в обратном направлении, до тех пор, пока $f_1(u_0)$ не будет определено. Однако мы должны наиболее полно использовать квадратичную структуру вариационной задачи. Если это проделать, то аналитические выражения и вычислительные проблемы станут значительно проще.

5. Рекуррентные уравнения

Легко проверить по индукции, что $f_R(v)$ квадратична по v:

$$f_R(v) = (A_R v,\ v) - 2(b_R,\ v) + c_R, \tag{1}$$

где A_R — симметричная матрица. Кроме того, величины A_R, b_R и c_R не зависят от v. Подставляя эту квадратичную форму вместо $f_{R+1}(u_R)$ в (4.5), получаем

$$f_R(v) = \min_{u_R}[(Qu_R,\ u_R) - 2(r_R,\ u_R) + s_R + (u_R - v,\ u_R - v) + (A_{R+1}u_R,\ u_R) - 2(b_{R+1},\ u_R) + c_R]. \tag{2}$$

Продифференцировав это выражение, легко получить минимизирующее значение u_R. То, что это значение u_R действительно доставляет минимум, непосредственно следует из положительной определенности функционала. Этот момент более подробно будет обсуждаться позднее. Таким образом, искомое значение u_R равно

$$u_R = [Q + A_{R+1} + I]^{-1}(v + b_{R+1} + r_R). \tag{3}$$

Использовав это в (2), получим

$$\begin{aligned} f_R(v) = {} & ([I - [I + Q + A_{R+1}]^{-1}]\,v,\ v) - \\ & - (2[I + Q + A_{R+1}]^{-1}(b_{R+1} + r_R),\ v) + c_{R+1} + s_{R+1} - \\ & - ([I + Q + A_{R+1}]^{-1}(b_{R+1} + r_R),\ b_{R+1} + r_R). \end{aligned} \tag{4}$$

Сравнивая с (1), получаем соотношения:

$$\begin{aligned} A_R &= I - [I + Q + A_{R+1}]^{-1}, \\ b_R &= [I + Q + A_{R+1}]^{-1}(b_{R+1} + r_R) = \\ &= [I - A_R](b_{R+1} + r_R), \\ c_R &= c_{R+1} + s_R - ([I + Q + A_{R+1}]^{-1}(b_{R+1} + r_R),\ b_{R+1} + r_R) = \\ &= c_{R+1} + s_R - (b_R,\ b_{R+1} + r_R). \end{aligned} \tag{5}$$

Из (4.7) получаем начальные условия:

$$A_N = I, \quad b_N = u_N, \quad c_N = ([I+Q]u_N, u_N) - 2(r_N, u_N) + s_N. \tag{6}$$

Поскольку v, по определению, равно u_{R-1}, то (3) принимает вид

$$u_R = [I+Q+A_{R+1}]^{-1}(u_{R-1} + b_{R+1} + r_R), \tag{7}$$

или

$$u_R = [I - A_R]u_{R-1} + b_R. \tag{8}$$

Поскольку нас в основном интересуют значения u_R, то уравнение для c_R можно не выводить.

6. Вычисления

Вычисления проводятся следующим образом. Уравнения (5.5) решаются последовательно, начиная с (5.6), до тех пор, пока не будут вычислены A_1 и b_1. Все промежуточные значения A_R и b_R запоминаются. Соотношение (5.8) является задачей Коши с начальным условием u_0, A_1 и b_1. Это уравнение решается методом итераций, на каждом шаге которого используются последнее вычисленное значение u_R и хранящиеся в памяти значения A_R и b_R. Соответствующая программа для *ЦВМ* приведена в приложении. Двухточечная граничная задача в данном случае свелась к *двум задачам Коши*, что типично для метода динамического программирования. Теперь мы покажем, что изложенный метод всегда приводит к решению и является численно устойчивым.

7. Невырожденность

Докажем, что все матрицы, которые в предыдущем процессе приходится обращать, невырожденны. Доказательство будет основано на том, что матрицы $[I + Q + A_{R+1}]$ положительно определены и их спектральный радиус меньше единицы.

Выражение

$$A > B \tag{1}$$

будет использовано для обозначения того факта, что $A - B$ положительно определена. Прежде всего докажем, что матрица Q положительно определена. Пусть x — произвольный нетривиальный вектор. Используя свойства скалярного произведения, имеем:

$$\begin{aligned}(Qx, x) &= \sum_{i=1}^{N}\sum_{j=1}^{N} q_{ij}x_ix_j = \\ &= 2\sum_{i=1}^{N} x_i^2 - 2\sum_{i=2}^{N} x_ix_{i-1} = \\ &= \sum_{i=1}^{N} x_i^2 + \sum_{i=2}^{N}(x_i - x_{i-1})^2 + x_1^2 + x_N^2 > 0,\end{aligned} \tag{2}$$

следовательно, Q положительно определена.

Далее доказательство проведем по индукции. Имеем

$$A_{N-1} = I - [I + Q]^{-1}. \tag{3}$$

Поскольку

$$Q > 0, \tag{4}$$

то

$$I + Q > I. \tag{5}$$

Таким образом, $[I + Q]$ невырожденна и A_{N-1} существует. Поскольку $[I + Q]$ положительно определена, то

$$[I + Q]^{-1} > 0, \tag{6}$$

и из (5) следует, что

$$[I + Q]^{-1} < I. \tag{7}$$

Таким образом,

$$0 < A_{N-1} < I. \tag{8}$$

Предположим теперь, что

$$0 < A_R < I \tag{9}$$

для некоторого $R < N - 1$. Тогда

$$[I + Q + A_R] > I \tag{10}$$

и, следовательно,

$$0 < [I + Q + A_R]^{-1} < I. \tag{11}$$

Поскольку

$$A_{R-1} = I - [I + Q + A_R]^{-1}, \tag{12}$$

то

$$0 < A_{R-1} < I, \tag{13}$$

что завершает индукцию.

Таким образом, все матрицы, которые мы должны обращать, положительно определены и, следовательно, невырожденны.

Упражнение

Показать, что $(X + Y)^{-1} \simeq X^{-1} - X^{-1}YX^{-1}$, если Y мало.

8. Устойчивость

Мы будем использовать термин *устойчивость* для описания того факта, что ошибка, допущенная на некотором шаге вычислительного процесса, не приводит к возрастанию ошибки при дальнейших вычислениях. Другими словами, локальные ошибки не накапливаются при последующих вычислениях.

Рассмотрим сначала рекуррентное матричное уравнение

$$A_R = I - [I + Q + A_{R+1}]^{-1} \tag{1}$$

и предположим, что уже допущена маленькая погрешность. Это означает, что в действительности мы имеем дело с рекуррентным соотношением

$$\tilde{A}_R = I - [I + Q + \tilde{A}_{R+1}]^{-1}, \tag{2}$$

где

$$\tilde{A}_R = A_R + E_R, \tag{3}$$

т. е. $\tilde{A}_R$ равно искомому решению плюс погрешность E_R. Используя (1) — (3), получаем, что погрешность на следующем шаге определяется из соотношения

$$E_R = [I + Q + A_{R+1}]^{-1} - [I + Q + A_{R+1} + E_{R+1}]^{-1}. \tag{4}$$

После некоторых преобразований это выражение принимает вид

$$E_R = [I + Q + \tilde{A}_{R+1}]^{-1} E_{R+1} [I + Q + A_{R+1}]^{-1}. \tag{5}$$

Если предположить, что начальная ошибка достаточно мала и матрицы не являются плохо обусловленными, то в силу (5.5) получаем

$$E_R \simeq [I - A_{R+1}] E_{R+1} [I - A_{R+1}]. \tag{6}$$

Из (6) естественно следует неравенство для норм:

$$\| E_R \| \leqslant \| I - A_{R+1} \|^2 \| E_{R+1} \|. \tag{7}$$

Поскольку все матрицы симметричны (как следствие теоретических построений и избранных численных процедур), то можно использовать следующую норму:

$$\| M \| = \rho^{\frac{1}{2}} (M^2), \tag{8}$$

где $\rho (M^2)$ обозначает модуль M^2. Тогда (7) принимает вид

$$\| E_R \| \leqslant \rho ([I - A_R]^2) \| E_{R+1} \|. \tag{9}$$

Наконец,

$$\| E_R \| < \| E_{R+1} \|, \tag{10}$$

откуда вытекает, что данное матричное уравнение устойчиво.

Используя тот же подход, можно проанализировать векторное рекуррентное соотношение

$$b_R = [I - A_R] (b_{R+1} + r_R). \tag{11}$$

Опять же фактически мы имеем уравнение

$$\tilde{b}_R = [I - A_R] (\tilde{b}_{R+1} + r_R), \tag{12}$$

где

$$\tilde{b}_R = b_R + e_R. \tag{13}$$

Так же как и выше, будем предполагать, что величина $I - A_R$ из (12) вычисляется точно. Выполняя те же операции, что и выше, находим

$$e_R = [I - A_R]\, e_{R+1} \tag{14}$$

и получаем неравенство для норм

$$\| e_R \| < \| e_{R+1} \|. \tag{15}$$

Таким образом, уравнение (11) также устойчиво.

Наконец, повторим предыдущие рассуждения для уравнения

$$u_R = [I - A_R]\, u_{R-1} + b_R. \tag{16}$$

Если e_R обозначает погрешность вычисления u_R, то видно, что

$$e_R = [I - A_R]\, e_{R+1}, \tag{17}$$

и снова

$$\| e_R \| < \| e_{R+1} \|. \tag{18}$$

Таким образом, мы приходим к выводу, что изложенный метод устойчив.

Упражнения

1. При каком условии решение векторно-матричного дифференциального уравнения

$$x' = Ax + b$$

устойчиво?

2. Рассмотреть вопрос об устойчивости матричного уравнения Риккати

$$R' = Q - R^2, \quad R(0) = 0,$$

если Q положительно определена.

9. Обсуждение

Эта устойчивость не является неожиданной. В действительности при некоторых весьма общих условиях она по существу гарантируется теорией динамического программирования. Это можно пояснить следующими рассуждениями. Решение, которое мы ищем, представляет собой оптимальный путь, соответствующий функционалу, из которого он возникает. Принцип оптимальности указывает, как выбрать оптимальный путь, начинающийся из произвольного состояния. Если из-за погрешности вычислений мы окажемся в некотором состоянии, отличном от вычисленного, то в дальнейшем автоматически пойдем по пути, оптимальному по отношению к данному состоянию. Следовательно, на каждом шаге процесса

мы осуществляем оптимизацию, основываясь на реальном состоянии. Этот многошаговый процесс принятия решений препятствует накоплению ошибок.

Помимо того что устойчивость данного метода, очевидно, позволяет эффективно вычислять решения наших уравнений, она придает ему еще одно преимущество. Поскольку все ошибки являются по существу локальными, то при большом шаге h ошибка будет определяться в основном локальными ошибками округления. Известно, что ошибка округления имеет порядок $O(h^2)$, поэтому можно ожидать, что метод «замедленного стремления к пределу» существенно улучшит результаты. Этот метод обсуждается в разд. 12.

Нетрудно показать, что метод динамического программирования решает ту же самую систему линейных алгебраических уравнений, которая возникает в результате стандартной конечно-разностной аппроксимации уравнения Лапласа. Это станет очевидным в гл. 6.

10. Эффективность

Для предыдущего метода довольно легко подсчитать число необходимых арифметических операций. Мы будем учитывать лишь число требуемых умножений и делений. Матрицы, которые необходимо обращать, не являются слабо заполненными, поэтому для каждого обращения матрицы порядка M потребуется приблизительно $M^3/2$ умножений и делений. Таким образом, для решения рекуррентных матричных уравнений необходимо всего $NM^3/2$ умножений и делений, если пренебречь величинами низшего порядка. Можно видеть, что для решения двух векторных уравнений потребуется всего $2NM^2$ умножений и делений [1]).

На первый взгляд это число операций кажется бо́льшим, чем число операций, необходимое в методе последовательной сверхрелаксации (ПСР) или методе чередующихся направлений (ЧН), где требуется $14N^3 \log N$ и $40N^2 \log^2 N$ операций соответственно (при $M = N$). Однако эти оценки несколько обманчивы. Как ПСР, так и ЧН являются итерационными и требуют для обеспечения сходимости соответствующего выбора некоторых параметров. Если эти параметры не выбраны почти оптимальным образом, то эти методы могут и не сходиться либо может потребоваться

---

[1]) Число арифметических операций можно значительно уменьшить [до $O(NM^2)$], если воспользоваться тем фактом, что матрица, которая приводит Q к диагональной форме, также приводит к диагональной форме и все матрицы A_R. Поэтому все вычисления можно провести, используя собственные значения матрицы Q. Мы не будем обсуждать этот важный сам по себе метод, поскольку область его приложения ограничена уравнениями с постоянными коэффициентами. В гл. 8 мы рассмотрим специальные вычислительные методы такого типа.

много дополнительных итераций. Выбор параметров сам по себе является сложной математической задачей, поскольку эти параметры зависят от вида уравнения, граничных условий и структуры области. В гл. 11, когда будут рассматриваться нелинейные уравнения, мы убедимся в важности этого замечания.

Возможно, более важным является то качество, которое мы называем *эффективностью*. Часто бывает нужно решать одно и то же уравнение с множеством различных граничных условий и, быть может, с измененными параметрами уравнения. Посмотрим, что произойдет, если мы попытаемся второй раз решить уравнение Лапласа с другими граничными условиями. Матричное рекуррентное уравнение, решение которого поглощает большую часть вычислительного времени, *не зависит* от граничных условий. Таким образом, нет необходимости снова вычислять его решение, и вся вычислительная процедура в каждом конкретном случае сводится к решению двух векторных уравнений. Для решения этой части задачи требуется только $2NM^2$ умножений. Эту тему мы будем часто затрагивать и в дальнейшем.

Наконец, остановимся на трудностях, связанных с распределением памяти. Мы вынуждены запоминать значения матриц A_R. Несмотря на их симметричность, для хранения матриц тем не менее требуется очень много памяти — пример «проклятия размерности». Однако поскольку эти матрицы вызываются последовательно и каждая только один раз, то для их хранения могут оказаться очень эффективными запоминающие устройства с малой скоростью выборки. В этой связи заметим, что такими устройствами удобно пользоваться в современных многопроцессорных вычислительных машинах.

11. Пример

В качестве примера использования изложенного выше метода было решено уравнение Лапласа, заданное на единичном квадрате с граничными условиями:

$$u(x, 0) = 1, \qquad u(0, y) = 0, \tag{1}$$
$$u(x, 1) = 0, \qquad u(1, y) = 0.$$

Вычисление обратных матриц проводилось по методу исключений Гаусса — Жордана, программа которого имелась в библиотеке стандартных подпрограмм. Матрицы A_R и векторы b_R хранились на высокоскоростном диске.

Некоторые типичные результаты приведены в таблице I, где ешение, полученное методом динамического программирования, сравнивается с решениями, полученными с помощью последовательной сверхрелаксации (27 итераций) и разложения в ряд

(30 членов). В методах динамического программирования и последовательной сверхрелаксации длина шага была одинаковой и равнялась 1/16. Во всех трех случаях время вычислений было одного порядка. Однако с учетом изложенного в последующем разделе и специальных методов из гл. 8 следует сказать, что было бы нецелесообразно обсуждать здесь вопросы, связанные с временем вычислений.

Таблица I

Точка	Динамическое программирование	Последовательная сверхрелаксация	Разложение в ряды
$(1/4, 1/4)$	0,43178	0,43178	0,43203
$(1/2, 1/4)$	0,53932	0,53933	0,54053
$(1/2, 3/4)$	0,09564	0,09564	0,09541
$(3/4, 1/2)$	0,18251	0,18253	0,18203

12. Замедленное стремление к пределу

В разд. 8 уже отмечалось, что по существу все ошибки в окончательном результате носят локальный характер. Если шаг h выбран не слишком малым (что следует делать для экономии памяти и машинного времени), то погрешность в основном определяется локальными ошибками округления.

Предположим, что решение $u(x, y, h)$ дискретизированной задачи зависит от h^2 аналитическим образом, и посмотрим, как это обстоятельство можно использовать:

$$u(x, y, h) = u(x, y, 0) + u_1(x, y)\, h^2 + u_2(x, y)\, h^4 + \ldots . \quad (1)$$

Нас интересует величина $u(x, y, 0)$. Естественно считать, с погрешностью $O(h^2)$, что

$$u(x, y, 0) \simeq u(x, y, h). \quad (2)$$

Эту оценку можно улучшить, взяв, например, меньшее значение h. Однако поскольку время вычислений пропорционально $1/h^4$, то понятно, что этот способ обладает явными недостатками.

Для получения лучшей оценки, чем (2), используем представление (1). Можно записать

$$u(x, y, h/2) = u(x, y, 0) + u_1(x, y)\,(h^2/4) + u_2(x, y)\,(h^4/16) + \ldots . \quad (3)$$

Тогда

$$u(x, y, 0) = \frac{1}{3}[4u(x, y, h/2) - u(x, y, h)] + O(h^4). \quad (4)$$

Таким образом, мы дважды решаем задачу и вычисляем (4) в общих точках сеток. Некоторые типичные результаты приведены в таблице II.

Таблица II

h	$u(x, y, h)$	$u(x, y, 0)$
$1/8$	0,43105	
$1/16$	0,43178	0,43202
$1/32$	0,43197	0,43202
$1/64$	0,43201 (27)	0,43202
$1/128$	0,43202 (44)	0,4320283

Результат, полученный с помощью разложения в ряд, равен

$$u\left(\frac{1}{4}, \frac{1}{4}\right) = 0{,}4320283. \tag{5}$$

Из таблицы видно, что метод замедленного стремления к пределу приводит к лучшим результатам, чем уменьшение шага, причем это достигается без существенного увеличения времени вычислений и необходимой памяти. Напомним еще раз, что этот прием нелегко использовать в итерационных методах, поскольку при недостаточно большом числе итераций погрешность может состоять не только из локальных ошибок округления.

13. Линейные уравнения общего вида

Пусть задана область R с границей Г, тогда уравнение

$$u_{xx} + u_{yy} - g(x, y)\, u + \varphi(x, y) = 0 \tag{1}$$

с граничными условиями

$$u(x, y) = f(x, y) \text{ на } \Gamma \tag{2}$$

является уравнением Эйлера, соответствующим задаче минимизации функционала

$$J(u) = \int_R \int [u_x^2 + u_y^2 + g(x, y)\, u^2 - 2\varphi(x, y)\, u]\, dx\, dy \tag{3}$$

при условиях (2).

Если записать, как в разд. 2,

$$\begin{aligned} \partial u_{ij}/\partial x &= [(u_{i+1,\,j} - u_{ij})/h] + O(h), \\ \partial u_{ij}/\partial y &= [(u_{i,\,j+1} - u_{ij})/h] + O(h) \end{aligned} \tag{4}$$

и обозначить

$$g_{ij} = g(ih,\ jh)\,h^2, \qquad \varphi_{ij} = \varphi(ih,\ jh)\,h^2, \tag{5}$$

то дискретный вариант функционала (3) примет вид

$$J(u) = \sum_{i=1}^{N} \Big[\sum_{j=1}^{M} (u_{ij} - u_{i,\,j+1})^2 + \sum_{j=1}^{M-1} [(u_{ij} - u_{i+1,\,j})^2 + g_{ij}u_{ij}^2 - 2\varphi_{ij}u_{ij}] \Big]. \tag{6}$$

Определим теперь матрицу Q, векторы r_R и скаляры s_R, как это было сделано ранее [1]), и введем диагональные матрицы G_R и векторы φ_k:

$$G_R = \operatorname{diag}(g_{R1},\ g_{R2},\ \ldots,\ g_{R,\,M-1}), \qquad \varphi_R = [\varphi_{Rj}]. \tag{7}$$

Тогда, используя понятие скалярного произведения, запишем (6) в следующем виде:

$$J(u) = \sum_{R=1}^{N} [(Qu_R,\ u_R) - (2r_R,\ u_R) + s_R + (G_R u_R,\ u_R) - (2\varphi_R,\ u_R) + (u_R - u_{R-1},\ u_R - u_{R-1})]. \tag{8}$$

Теперь можно поступить, как и выше. Определим

$$f_R(v) = \min_{u_R,\ u_{R+1},\ \ldots,\ u_{N-1}} \sum_{i=R}^{N} [([Q + G_i]u_i,\ u_i) - (r_i + \varphi_i,\ 2u_i) + (u_i - v,\ u_i - v) + s_i], \tag{9}$$

где

$$v = u_{R-1}. \tag{10}$$

Используя принцип оптимальности, получаем рекуррентное соотношение

$$f_R(v) = \min_{u_R} [([Q + G_R]\,u_R,\ u_R) - (r_R + \varphi_R,\ 2u_R) + (u_R - v,\ u_R - v) + s_R + f_{R+1}(u_R)], \quad R = 1,\ 2,\ \ldots,\ N-1 \tag{11}$$

при условии

$$f_N(v) = ([Q + G_N]u_N, u_N) - (r_N + \varphi_N, 2u_N) - (u_N - v,\ u_N - v) + s_N. \tag{12}$$

Опять-таки заметим, что $f_R(v)$ можно переписать как

$$f_R(v) = (A_R v,\ v) - (2b_R,\ v) + c_R. \tag{13}$$

---

[1]) См. стр. 45.— *Прим. ред.*

Таким образом, находим, что минимизирующее значение u_R равно

$$u_R = [I + Q + G_R + A_{R+1}]^{-1} (v + b_{R+1} + r_R + \varphi_R), \qquad (14)$$

в то время как A_R и b_R удовлетворяют рекуррентным уравнениям

$$\begin{aligned} A_R &= I - [I + Q + G_R + A_{R+1}]^{-1}, \\ b_R &= [I - A_R] (r_R + \varphi_R + b_{R+1}). \end{aligned} \qquad (15)$$

с начальными условиями

$$A_N = I, \quad b_N = u_N. \qquad (16)$$

Мы опять опустим уравнение для c_R, поскольку нас интересует лишь u_R. Итак, мы решаем уравнения (15) с начальными условиями (16) и запоминаем результат. Далее, (14) можно переписать в виде

$$u_R = [I - A_R] u_{R-1} + b_R. \qquad (17)$$

Это снова задача Коши с начальным условием u_0. Подробности оставляем читателю.

Возвращаясь к вопросу о невырожденности и устойчивости, видим, что достаточным условием невырожденности матриц A_R и устойчивости метода является следующее:

$$g(x, y) \geqslant 0. \qquad (18)$$

Это гарантирует положительную определенность квадратичного функционала.

14. Нерегулярные области

Как уже отмечалось, если имеется решение данного уравнения на некоторой заданной области и при этом известны матрицы A_R, то ценой незначительных усилий можно получить решение того же уравнения при ином множестве граничных условий. Как следует из разд. 11, матрицы A_R не зависят от функции возмущающей силы $\varphi(x, y)$. Матрицы A_R представляют собой функцию Грина дискретной задачи и зависят только от уравнения независимо от функции внешнего воздействия и структуры области.

Более того, ясно, что если известны матрицы A_N, A_{N-1}, ..., A_1, то этого достаточно для решения задач на усеченных областях, как показано на рис. 2. Для этой области потребуется знать A_N, A_{N-1}, . . ., A_{K+1}. Это обстоятельство указывает на то, что во многих приложениях целесообразно хранить массив матриц A_R разной размерности на магнитной ленте.

Теперь мы собираемся рассмотреть метод разбиения для области нерегулярной формы. В гл. 7 мы обсудим прямой метод решения задач с такими областями. Предположим, что задана область, изображенная на рис. 3. Эту область можно разделить на два прямоугольника, I и II. Предположим, что мы хотим решить

уравнение Лапласа, причем уже знаем решения этого уравнения на областях I и II. Предположив далее, что это уравнение решалось

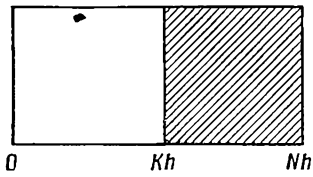


Рис. 2

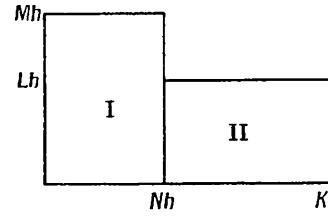


Рис. 3

слева направо на области I и справа налево на II, мы тем самым будем иметь решение уравнения

$$A_R = I - [A_{R-1} + I + Q]^{-1}, \quad R = 1, 2, \ldots, N-1, \quad A_0 = I, \quad (1)$$

где все матрицы порядка $M-1$, и уравнения

$$\overline{A}_R = I - [\overline{A}_{R+1} + I + Q]^{-1}, \; R = N+1, \; N+2, \ldots, K-1, \; \overline{A}_K = I, \quad (2)$$

где матрицы имеют порядок $L-1$. Покажем, что эти матрицы можно использовать для получения решения на составной области, даже в случае, когда граничные условия могут измениться.

Для данного множества граничных условий можно определить r_R и $\overline{r}_R$ и затем решить уравнение

$$b_R = [I - A_R](b_{R-1} + r_R), \quad b_0 = u_0, \quad R = 1, 2, \ldots, N-1 \quad (3)$$

и

$$\overline{b}_R = [I - \overline{A}_R](\overline{b}_{R+1} + \overline{r}_R), \quad \overline{b}_K = u_K, \quad R = N+1, \ldots, K-1, \quad (4)$$

поскольку порядки величин во всех уравнениях согласуются, а u_0, u_K, r_R и $\overline{r}_R$ определяются граничными условиями. Сложность кроется в u_N, где число узлов сетки изменяется. Если, однако, можно определить u_N, то решение на области I можно найти с помощью соотношения

$$u_R = [I - A_R]\, u_{R+1} + b_R, \quad R = 1, 2, \ldots, N-1, \quad (5)$$

а на области II с помощью

$$u_R = [I - \overline{A}_R]\, u_{R-1} + \overline{b}_R, \quad R = N+1, \ldots, K-1. \quad (6)$$

Теперь покажем, как определить u_N. Можно записать u_N в виде

$$u_N = \begin{bmatrix} y \\ w \end{bmatrix}, \tag{7}$$

где

$$\begin{aligned} y &= (u_{N1},\ u_{N2},\ \ldots,\ u_{N,\,L-1}), \\ w &= (u_{N,\,L},\ u_{N,\,L+1},\ \ldots,\ u_{N,\,M-1}). \end{aligned} \tag{8}$$

Такое представление удобно, поскольку y состоит из внутренних точек, а w задано граничными условиями. Таким образом, нам необходимо определить лишь y.

Используя определения из разд. 4, обозначим минимальное значение функционала на области I через $f_{N-1}(u_N)$, а на области II — через $g_{N+1}(y)$. Используя квадратичную структуру функционала, получаем [1])

$$\begin{aligned} f_{N-1}(u_N) &= f_{N-1}\left(\begin{bmatrix} y \\ w \end{bmatrix}\right) = (A_{N-1}u_N,\ u_N) - (2b_{N-1},\ u_N) + c_{N-1}, \\ g_{N+1}(y) &= (\bar{A}_{N+1}y,\ y) - (2\bar{b}_{N+1},\ y) + \bar{c}_{N+1}. \end{aligned} \tag{9}$$

Если $G(u)$ — дискретный функционал, определенный на всей области, то, используя его аддитивность, получаем

$$\min_{\{u_i\}} G(u) = \min_{y} \left\{ f_{N-1}\left(\begin{bmatrix} y \\ w \end{bmatrix}\right) + g_{N+1}(y) \right\}. \tag{10}$$

Это по существу применение идеи свёртывания. Поскольку $f_{N-1}\left(\begin{bmatrix} y \\ w \end{bmatrix}\right)$ и $g_{N+1}(y)$ представлены в удобном виде, y легко определить. Представим A_{N-1} и b_{N-1} в форме

$$A_{N-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad b_{N-1} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{11}$$

где A_{11} и b_1 имеют размерность $L-1$. Теперь можно записать (10) в развернутой форме; тогда

$$\begin{aligned} \min_{\{u_i\}} G(u) = \min_{y} [&([A_{11} + \bar{A}_{N+1}]\,y,\ y) + ([A_{12}w + A_{21}^{T}]\,w,\ y) - (2b_1,\ y) - \\ &- (2\bar{b}_{N+1},\ y)] - (2b_2,\ w) + (A_{22}w,\ w) + \\ &+ c_{N-1} + \bar{c}_{N+1}. \end{aligned} \tag{12}$$

---

[1]) $\bar{c}_R$ и c_R определяются, как в разд. 5. Эти величины не входят в окончательное выражение для u_R, поэтому вычислять их нет необходимости.

Для определения минимизирующего значения y это выражение можно продифференцировать. Учитывая симметрию A_{N-1}, получаем

$$y = [A_{11} + \overline{A}_{N+1}]^{-1} (A_{12} w - b_1 - \overline{b}_{N+1}). \tag{13}$$

Используя эту величину y, заключаем, что (5) и (6) являются задачами Коши с начальными значениями u_N и y соответственно.

Если матрицы A_R и $\overline{A}_R$ известны, то сначала вычислим b_R и $\overline{b}_R$, что является простой задачей. Далее определим значение y из (13) и, наконец, получим искомое значение u_R, решив уравнения (5) и (6). Все эти операции требуют небольшого объема вычислений по сравнению с усилиями, необходимыми для непосредственного решения исходной задачи на нерегулярной области.

15. Уравнения более высокого порядка

Изложенный метод не ограничивается линейными уравнениями второго порядка. Рассмотрим, например, задачу статической деформации упругой пластины под действием поперечной нагрузки, которая описывается бигармоническим уравнением

$$u_{xxxx} + 2u_{xxyy} + u_{yyyy} = p, \tag{1}$$

представляющим собой эллиптическое уравнение четвертого порядка. Граничные условия для уравнения (1) обычно задаются в виде условий на краях пластины. На краях закрепленной пластины выполняется условие

$$u(x, y) = 0 \tag{2}$$

и, кроме того, условие для нормальной производной

$$u_n(x, y) = 0. \tag{3}$$

Хорошо известно, и это легко проверить, что (1) представляет собой уравнение Эйлера для функционала

$$J(u) = \int_R \int [(u_{xx} + u_{yy})^2 - 2pu]\, dx\, dy \tag{4}$$

при ограничениях (2) и (3). Можно попытаться получить приближенное решение задачи (1) с помощью дискретной аппроксимации (4). Действуя, как и ранее, заменим u_{xx}, u_{yy} и u_{xy} в (4) конечно-разностными выражениями. Тогда непосредственно получаем функциональное уравнение

$$f_R(v, w) = \min_{u_R} [G(u_R, v, w) + f_{R+1}(u_R, w)], \tag{5}$$

где

$$v = u_{R-1}, \quad w = u_{R-2}.$$

Используя квадратичную структуру (4), это уравнение можно значительно упростить. Подробные выкладки оставляем читателю.

Более простой прием заключается в следующем. Определим

$$v = u_{xx} + u_{yy}, \tag{6}$$

где u — решение задачи (1). Легко проверить, что функция v должна удовлетворять уравнению

$$v_{xx} + v_{yy} = p. \tag{7}$$

Определив теперь вектор w как

$$w = \begin{bmatrix} u \\ v \end{bmatrix},$$

из (6) и (7) получим, что

$$w_{xx} + w_{yy} = Aw + s, \tag{8}$$

где

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad s = \begin{bmatrix} 0 \\ p \end{bmatrix}.$$

Соответствующее граничное условие для w можно найти из (2) и (3). Поскольку (8) является уравнением Эйлера для функционала

$$J(w) = \int_R \int [(w_x, w_x) + (w_y, w_y) + (Aw, w) - 2(s, w)]\, dx\, dy, \tag{9}$$

то с помощью методов, изложенных в разд. 11, легко минимизировать дискретный аналог этого функционала. Единственное отличие заключается в том, что для уравнений четвертого порядка матрицы и векторы будут иметь размерность 2 ($M - 1$). Подробно эта задача исследована в следующей главе с помощью аппарата инвариантного погружения. Легко показать, что изложенный метод всегда приводит к решению и является численно устойчивым.

Упражнения

1. Вывести рекуррентное матричное уравнение для (8).
2. Показать, что необходимые обратные матрицы всегда существуют.
3. Вывести соответствующее рекуррентное векторное уравнение. Каковы начальные условия для этого уравнения в случае задачи (1) — (3)?
4. Во многих физических ситуациях бывают заданы изгибающие моменты и нормальные силы, действующие на некотором участке границы. Рассмотрим, например, прямоугольную пластину, в которой на ребро $x = 0$ действует изгибающий момент $f(y)$ и нормальная сила $g(y)$, в то время как на остальных ребрах

соблюдаются условия (2) и (3). Таким образом, при $x = 0$ мы имеем граничные условия вида

$$-u_{xx}(0, y) - \nu u_{yy}(0, y) = f(y),$$
$$u_{xxx}(0, y) + (2 - \nu) u_{xyy}(0, y) = g(y),$$

где ν — отношение Пуассона, физическая константа, зависящая от материала. Вместо (4) мы должны теперь рассмотреть функционал более сложной структуры:

$$J(u) = \int_R \int [(u_{xx} + u_{yy})^2 - 2(1 - \nu)(u_{xx}u_{yy} - u_{xy}^2) - 2pu]\, dx\, dy +$$
$$+ 2 \int_{\Gamma} gu\, dy + 2 \int_{\Gamma} fu_x\, dy,$$

поскольку необходимо учесть воздействия сил f и g на ребре. Вывести уравнения динамического программирования, используя в качестве переменных

$$M(x, y) = -u_{xx} - \nu u_{yy},\ V(x, y) = u_{xxx} + (2 - \nu) u_{xyy}.$$

См. статью Н. Дистефано, указанную в конце главы.

16. Управление системой с распределенными параметрами

Задачи управления системами с распределенными параметрами в общем случае приводят к уравнениям в частных производных. Рассмотрим, например, задачу выбора функции управления $v \equiv \equiv v(x, t)$, минимизирующей

$$J(u, v) = \int_0^T \int_0^1 [g(x) u^2 + v^2]\, dx\, dt \tag{1}$$

при ограничениях

$$u_t = u_{xx} + v, \quad u(x, 0) = f(x), \quad u(1, t) = u(0, t) = 0. \tag{2}$$

Будем предполагать, что $g(x) > 0$. Легко проверить, что уравнение Эйлера для задачи (1), (2) имеет вид

$$v_t = -v_{xx} + g(x) u, \quad u_t = u_{xx} + v \tag{3}$$

при ограничениях

$$u(x, 0) = f(x), \qquad u(1, t) = u(0, t) = 0, \tag{4}$$
$$v(x, T) = 0, \qquad v(1, t) = v(0, t) = 0.$$

Для решения задачи (1) — (2) будем использовать дискретизацию и метод динамического программирования.

Непрерывную задачу можно решать и непосредственно, однако для этого потребовалось бы ввести некоторые более сложные понятия. Итак, определим

$$u_{ij} = u(ih, j\delta),$$

где

$$Mh = 1, \quad N\delta = T.$$

Обозначим через u_i и v_i $(M-1)$-мерные векторы

$$u_i = [u_{ij}],$$
$$v_i = [v_{ij}], \quad j = 1, 2, \ldots, M-1.$$

Задачу (1) — (2) можно заменить дискретной задачей отыскания

$$\min_{\{v_i\}} \sum_{i=R}^{N} [(Gu_i, u_i) + (v_i, v_i)] \tag{5}$$

при ограничениях

$$u_{i+1} = [I - \rho Q]\, u_i + hv_i, \tag{6}$$

где Q определяется, как и ранее,

$$G = \operatorname{diag}[g(h), \ldots, g((M-1)h)]$$

и $\rho = h/\delta^2$. Начальное условие u_0 для (6) определяется из (4).

Если мы рассмотрим последовательность вариационных задач

$$f_R(c) = \min_{\{v_i\}} \sum_{i=R}^{N} [(Gu_i, u_i) + (v_i, v_i)], \qquad u_R = c,$$

при ограничениях

$$u_{i+1} = [I - \rho Q]\, u_i + hv_i,$$

то, поступив, как и выше, получим

$$f_R(c) = \min_{v} [(Gc, c) + (v, v) + f_{R+1}([I - GQ]\, c + hv)],$$
$$f_N(c) = (Gc, c). \tag{7}$$

По индукции легко показать, что

$$f_R(c) = (A_R c, c).$$

Теперь можно получить минимизирующее значение v, а именно:

$$v = -h[I - h^2 A_{R+1}]^{-1} A_{R+1}[I - Q], \tag{8}$$

и рекуррентное уравнение для A_R

$$A_R = G + [I - \rho Q][I - [I + h^2 A_{R+1}]^{-1}][I - \rho Q], \tag{9}$$

где

$$A_N = G.$$

Можно показать, что при соответствующих ограничениях на отношение ρ необходимые обратные матрицы всегда существуют, и данный метод численно устойчив. В гл. 10 мы увидим, что при небольшом изменении способа дискретизации можно гарантировать существование и устойчивость для любых положительных значений ρ.

Упражнения

1. Показать, что если дискретизировать уравнение

$$u_t = u_{xx} + v$$

по формуле

$$u_{i+1} = [I + \rho Q]^{-1} u_i + [I + \rho Q]^{-1} v_i,$$

то рекуррентное уравнение для A_R принимает вид

$$A_R = G + P\,[I - [I + h^2 P A_{R+1} P]^{-1}]\,P,$$

где

$$P = [I + \rho Q]^{-1}.$$

2. Показать, что это уравнение имеет решение и является численно устойчивым для любого положительного ρ.

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Раздел 2. Это стандартный способ дискретизации. Давно известно, что переход к вариационной задаче приводит к симметричным разностным схемам; см.

Тодд (Todd J., ed.)

Survey of numerical analysis, McGraw-Hill, New York, 1962.

Раздел 4. Ряд задач дискретного динамического программирования рассмотрен в книге

Беллман Р.

Введение в теорию матриц, М., «Наука», 1969.

Раздел 5. Соответствующие результаты для метода прямых можно найти в работе

Энджел (Angel E.)

Dynamic programming and linear partial differential equations, *J. Math. Anal. Appl.*, 23 (1968), 628—638.

Раздел 7. В большинстве работ для доказательства существования и устойчивости используется теорема Перрона — Фробениуса для неотрицательных матриц. См., например,

Варга (Varga R.)

Matrix iterative analysis, Prentice-Hall, Englewood Cliffs, New Jersey 1962.

Раздел 8. На практике плохая обусловленность не вносит серьезных трудностей для матриц, порядок которых меньше 128.

Раздел 10. См. статью

Дорр (Dorr F. W.)

The direct solution of the discrete Poisson equation on a rectangle, *SIAM Rev.*, 12 (1970), 248—263.

Раздел 12. См. статью

Ричардсон (Richardson L. F.)

The approximate arithmetical solution by finite differences of physical problems involving differential equations, with applications to the stresses in a Masonry dam, *Philos. Trans. Roy. Soc.*, London, Ser. A, 210 (1910), 307—357.

Другие методы ускорения сходимости можно найти в книге

Тодд (Todd J., ed.)

Survey of numerical analysis, McGraw-Hill, New York, 1962.

Раздел 14. См. статью

Энджел (Angel E.)

A building block technique for elliptic boundary-value problems over irregular regions, *J. Math. Anal. Appl.*, 26 (1969), 75—81.

Этот метод аналогичен табличному методу, предложенному в книге

Канторович Л. В., Крылов В. И., Чернин К. И.

Таблицы для численного решения граничных задач в теории гармонических функций, Гостехиздат М., 1956.

Раздел 15. См. статью

Дистефано (Distefano N.)

Dynamic programming and the solution of the biharmonic equation, *Internat. J. Numer. Meth. Engrg.*, 3 (1971), 199—213.

Раздел 16. См. книги

Сейдж (Sage A. P.)

Optimum systems control, Prentice-Hall, Englewood Cliffs, New Jersey, 1968.

Беллман (Bellman R.)

Introduction to the mathematical theory of control processes II: Nonlinear processes, Academic Press, New York, 1971.

Существует много интересных вопросов, связанных с функционалом Дирихле $D(u)$; см.

Беллман и Осборн (Bellman R., Osborn H.)

Dynamic programming and the variation of Green's function, *J. Math. Mech.*, 7 (1958), 81—86, и другие работы Осборна.

6 Глава

Инвариантное погружение

1. Инвариантное погружение

В течение последних десяти лет с помощью теории инвариантного погружения были получены новые аналитические и численные результаты в различных областях знания, таких, например, как физика атмосферы, теория переноса, распространение радиоволн и т. д. С нашей точки зрения, метод инвариантного погружения позволяет заменить линейные двухточечные граничные задачи, вычислительные алгоритмы для которых часто оказываются неустойчивыми, задачами Коши, для решения которых существуют устойчивые численные методы. Метод инвариантного погружения можно применять и в тех случаях, когда соответствующая вариационная задача не существует.

Основные идеи инвариантного погружения можно пояснить на примере следующей задачи.

Рассмотрим процесс переноса нейтронов в стержне длины L (рис. 1). Обозначим через $u(t)$ поток нейтронов вправо в точке t

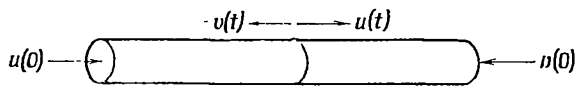


Рис. 1

и через $v(t)$ — поток влево. Начальные потоки $u(0)$ и $v(L)$ считаются заданными. Если рассмотреть движение частиц на интервале между t и $t + \Delta$, где $\Delta \ll L$, и затем положить $\Delta \to 0$, то легко можно получить дифференциальные уравнения, связывающие $u(t)$ и $v(t)$. Во многих случаях, когда частицы в потоке не взаимодействуют между собой, эти уравнения принимают простой вид:

$$\begin{aligned} u' &= Au + Bv, \\ v' &= Cu + Dv, \\ u(0) &= c_1, \quad v(L) = c_2. \end{aligned} \tag{1}$$

Эта двухточечная граничная задача линейна, поэтому можно сделать так, чтобы ее решение зависело от решения системы линейных алгебраических уравнений. Однако есть опасение, что при решении системы (1) обычные методы могут оказаться численно неустойчивыми, если не припять специальных мер.

Рассмотрим теперь эту задачу с иной точки зрения. Поток вправо $u(t)$ в некоторой точке стержня состоит из двух компонент — потока $u(0)$ в прямом направлении и потока $v(t)$, отраженного от правого конца. Если для стержня длины L в точке t обозначить через $T(t, L)$ и $R(t, L)$ коэффициенты пропускания и отражения соответственно, то

$$u(t) = T(t, L)\, u(0) + R(t, L)\, v(L). \tag{2}$$

Аналогичное представление справедливо и для $v(t)$. Видно, что (2) непосредственно вытекает из линейности (1).

Если рассмотреть изменение коэффициентов пропускания и отражения при увеличении длины стержня от L до $L + \Delta L$, то можно получить дифференциальное уравнение для $R(t, L)$ и $T(t, L)$, как для функций от L. Эти уравнения в общем случае являются задачами Коши, устойчивыми по отношению к большинству численных методов, и поэтому легко могут быть решены численно.

Метод инвариантного погружения с успехом применялся для решения многих классов задач, включающих обыкновенные дифференциальные, интегральные и интегро-дифференциальные уравнения. Нас в основном будут интересовать разностные уравнения.

2. Преобразование Риккати

Рассмотрим системы линейных разностных уравнений

$$\begin{aligned} u_{i+1} &= A_i u_i + B_i v_i + e_i, \\ v_{i+1} &= C_i u_i + D_i v_i + f_i, \quad i = 1, 2, \ldots, N-1, \end{aligned} \tag{1}$$

где u_i и v_i суть M-мерные векторы. Кроме того, будем предполагать, что заданы двухточечные граничные условия:

$$u_0 = c, \quad v_N = d. \tag{2}$$

Будем искать решение системы (1) в виде

$$v_i = R_i u_i + s_i, \tag{3}$$

где матрица R_i и вектор s_i не зависят от u_i и v_i. Существование решения такого вида есть следствие линейности (1). К этому вопросу мы еще вернемся в следующей главе.

Выражение (3) можно переписать следующим образом:

$$v_{i+1} = R_{i+1} u_{i+1} + s_{i+1}, \tag{4}$$

и использовать (1) для получения связи R_{i+1} и s_{i+1} с R_i и s_i. Применяя (1) и (4), получаем

$$C_i u_i + D_i v_i + f_i = R_{i+1} [A_i u_i + B_i v_i + e_i] + s_{i+1}, \tag{5}$$

и подставляя вместо v_i выражение (3), находим

$$[C_i + D_iR_i]\, u_i + f_i + D_is_i = \\ = R_{i+1}\,[A_i + B_iR_i]\, u_i + R_{i+1}\,(e_i + B_is_i) + s_{i+1}. \qquad (6)$$

Поскольку это выражение должно выполняться *для всех* u_i, то, приравнивая коэффициенты при u_i, получаем соотношения

$$[C_i + D_iR_i] = R_{i+1}\,[A_i + B_iR_i], \\ f_i + D_is_i = R_{i+1}\,(e_i + B_is_i) + s_{i+1}, \qquad (7)$$

или

$$R_i = [R_{i+1}B_i - D_i]^{-1}\,[C_i - R_{i+1}A_i], \\ s_i = [R_{i+1}B_i - D_i]^{-1}\,[f_i - s_{i+1} - R_{i+1}e_i]. \qquad (8)$$

Поскольку (3) справедливо для всех u_i и v_i, то

$$d = R_Nu_N + s_N. \qquad (9)$$

Это соотношение должно выполняться независимо от второго граничного условия, поэтому мы должны записать

$$R_N = 0, \quad s_N = c. \qquad (10)$$

Итак, теперь мы имеем *задачи Коши* для определения R_i и s_i, при этом начальной точкой является $i = N$, и решение строится в обратном направлении. Для получения задачи Коши для u_i подставим (3) в первое уравнение из (1) и получим

$$u_{i+1} = [A_i + B_iR_i]\, u_i + e_i + B_is_i. \qquad (11)$$

Это и есть задача Коши для u_i с начальным условием

$$u_0 = c, \qquad (12)$$

причем значения R_i и s_i определяются из (8) — (10). Мы отложим вопросы о существовании необходимых обратных матриц и о численной устойчивости до тех пор, пока не получим конкретных значений величин A_i, B_i, C_i и D_i, которые возникают при дискретизации уравнений в частных производных.

Упражнения

1. Исходя из представления

$$u_i = R_iv_i + s_i,$$

сформулировать задачу Коши для определения R_i и s_i.

2. Исходя из представления $u\,(t) = R\,(t)\,v\,(t) + s\,(t)$, сформулировать задачу Коши для решения (1.1).

3. Одношаговые методы

Мы уже установили, что можно заменить системы разностных уравнений с двухточечными граничными условиями двумя системами задач Коши. К сожалению, оказывается, что при этом для решения второй задачи потребуется запоминать результаты решения первой, а именно величины R_i и s_i. Однако при некоторых условиях метод инвариантного погружения приводит к задачам Коши без какого-либо увеличения необходимой памяти. Это обстоятельство можно пояснить на простом примере.

Рассмотрим *скалярные* разностные уравнения

$$\begin{aligned} u_{i+1} &= a_i u_i + b_i v_i, \\ v_{i+1} &= c_i u_i + d_i v_i \end{aligned} \tag{1}$$

с граничными условиями

$$u_0 = 0, \quad v_N = 1. \tag{2}$$

Поскольку система (1) линейна, то легко обобщить эти граничные условия или добавить возмущающее воздействие в правую часть, воспользовавшись принципом суперпозиции. Чтобы подчеркнуть, что процесс (1) — (2) является N-шаговым, перепишем (1) и (2) в виде

$$\begin{aligned} u_{i+1}(N) &= a_i u_i(N) + b_i v_i(N), \quad u_0(N) = 0, \\ v_{i+1}(N) &= c_i u_i(N) + d_i v_i(N), \quad v_N(N) = 1. \end{aligned} \tag{3}$$

Предположим далее, что мы рассматриваем то же самое уравнение с теми же граничными условиями, но с той лишь разницей, что процесс теперь проходит через $N+1$ состояний:

$$\begin{aligned} u_{i+1}(N+1) &= a_i u_i(N+1) + b_i v_i(N+1), \quad u_0(N+1) = 0, \\ v_{i+1}(N+1) &= c_i u_i(N+1) + d_i v_i(N+1), \quad v_{N+1}(N+1) = 1. \end{aligned} \tag{4}$$

Вследствие линейности решения систем (3) и (4) совпадают с точностью до некоторой мультипликативной постоянной k, т. е.

$$\begin{aligned} u_i(N+1) &= k u_i(N), \\ v_i(N+1) &= k v_i(N). \end{aligned} \tag{5}$$

Полагая в (5) $i = N$ и вспоминая граничное условие

$$v_N(N) = 1, \tag{6}$$

видим, что

$$k = v_N(N+1), \tag{7}$$

или

$$\begin{aligned} u_i(N+1) &= v_N(N+1)\, u_i(N), \\ v_i(N+1) &= v_N(N+1)\, v_i(N). \end{aligned} \tag{8}$$

Таким образом, мы получили фундаментальные соотношения инвариантного погружения, которые показывают, как изменяется решение при увеличении длительности процесса.

Определим r_i как

$$r_i = u_i(i). \tag{9}$$

Положив в (4) $i = N$, получим

$$\begin{aligned} r_{N+1} &= a_N u_N(N+1) + b_N v_N(N+1), \\ 1 &= c_N u_N(N+1) + d_N v_N(N+1). \end{aligned} \tag{10}$$

Полагая в (8) $i = N$, получаем

$$u_N(N+1) = v_N(N+1)\, r_N. \tag{11}$$

Решив (10) и (11), можно выразить r_{N+1} через r_N:

$$r_{N+1} = (a_N + b_N r_N)/(c_N + d_N r_N). \tag{12}$$

Поскольку

$$u_0(N) = 0 \tag{13}$$

для всех N, то для (12) получаем начальное условие

$$r_0 = 0. \tag{14}$$

Возвращаясь к предыдущему разделу, видим, что (12) является скалярным аналогом уравнения (7). Однако мы все еще не использовали всех возможностей метода инвариантного погружения.

Пусть мы хотим найти $u_k(N)$ и $v_k(N)$ для некоторых заданных значений k и N. Решая уравнения (10), получаем $v_N(N+1)$

$$v_N(N+1) = \frac{a_N - c_N r_{N+1}}{a_N d_N - b_N c_N}. \tag{15}$$

Подставляя (15) в (8), получаем

$$\begin{aligned} u_i(N+1) &= \frac{a_N - c_N r_{N+1}}{a_N d_N - b_N c_N}\, u_i(N), \\ v_i(N+1) &= \frac{a_N - c_N r_{N+1}}{a_N d_N - b_N c_N}\, v_i(N), \end{aligned} \qquad i \leqslant N. \tag{16}$$

Для определения $u_k(N)$ и $v_k(N)$ поступим следующим образом. Решим (12) для $N = 1, 2, \ldots, k$. Затем добавим (16) с начальным условием

$$u_k(k) = r_k, \quad v_k(k) = 1 \tag{17}$$

и решим (12) и (16) для $N = k, k+1, \ldots, N-1$. Хотя здесь и не требовалось никакой памяти, мы получили лишь значения $u_k(N)$ и $v_k(N)$. Если нас, кроме того, интересуют значения $u_l(N)$

и $v_l(N)$, то мы должны добавить к (12) другую систему уравнений типа (16) и решать эту систему, положив $n = l$ с начальными условиями

$$u_l(l) = r_l, \quad v_l(l) = 1. \tag{18}$$

Хотя эта процедура весьма полезна, мы будем далее предполагать, что имеющейся памяти достаточно для осуществления первоначального варианта процедуры. Однако читатель должен иметь в виду, что можно пользоваться и только что описанным методом, который во многих случаях может оказаться более удобным.

Упражнение

Получить эквивалентные результаты для векторных уравнений из разд. 2. *Указание.* Рассмотрите матричную систему

$$\begin{aligned} U_{i+1}(N) &= A_i U_i(N) + B_i V_i(N), \\ V_{i+1}(N) &= C_i U_i(N) + D_i V_i(N), \\ U_0(N) &= 0, \quad V_N(N) = I, \end{aligned}$$

и покажите, что

$$\begin{aligned} U_i(N+1) &= U_i(N)\, V_N(N+1), \\ V_i(N+1) &= V_i(N)\, V_N(N+1). \end{aligned}$$

Какова физическая интерпретация этих уравнений?

4. Дискретизация

Теперь мы можем вернуться к решению уравнений в частных производных эллиптического типа. В качестве примера опять рассмотрим уравнение Лапласа

$$u_{xx} + u_{yy} = 0 \tag{1}$$

на прямоугольнике

$$0 \leqslant x \leqslant a, \quad 0 \leqslant y \leqslant b \tag{2}$$

с граничными условиями, заданными на всех его сторонах. Покроем эту область сеткой и рассмотрим только точки

$$u_{ij} = u(ih, jh), \quad i = 0, 1, \ldots, N, \quad j = 0, 1, \ldots, M. \tag{3}$$

Здесь мы, как и прежде, предположили, что a и b таковы, что можно использовать фиксированную сетку с размером ячейки h. Заменим частные производные в (1) стандартными конечно-разностными приближениями

$$\begin{aligned} \partial^2 u_{ij}/\partial x^2 &= [(u_{i+1,\,j} - 2u_{ij} + u_{i-1,\,j})/h^2] + O(h^2), \\ \partial^2 u_{ij}/\partial y^2 &= [(u_{i,\,j+1} - 2u_{ij} + u_{i,\,j-1})/h^2] + O(h^2). \end{aligned} \tag{4}$$

Таким образом, мы пришли к задаче решения системы линейных разностных уравнений $(N-1)(M-1)$-го порядка.

$$u_{i+1,\,j}+u_{i,\,j+1}-4u_{ij}+u_{i,\,j-1}+u_{i-1,\,j}=0, \quad i=1, 2, \ldots, N-1,$$
$$j=1, 2, \ldots, M-1. \tag{5}$$

Значения u_{i0}, u_{0j}, u_{Nj} и u_{iM} определяются граничными условиями. Поскольку линейные разностные уравнения в (5) «разреженные» (ни одно уравнение не содержит более пяти неизвестных), то для решения этой системы уравнений выгодно применять итерационные методы. Мы же поступим по-другому.

Определим $(M-1)$-мерные векторы u_R, $(M-1)$-мерную матрицу Q и $(M-1)$-мерные векторы r_R точно так же, как и в гл. 5, а именно:

$$u_R=[u_{Ri}],$$

$$Q=(q_{ij}), \text{ где } q_{ij}=\begin{cases} 2, & i=j, \\ -1, & |i-j|=1, \\ 0 & \text{в противных случаях,} \end{cases}$$

$$r_R=[r_{Ri}], \text{ где } r_{Ri}=\begin{cases} u_{R0}, & i=1, \\ u_{RM}, & i=M-1, \\ 0 & \text{в противных случаях.} \end{cases} \tag{6}$$

Теперь уравнения (5) можно переписать в виде

$$u_{R+1}-2u_R+u_{R-1}-Qu_R+r_R=0, \tag{7}$$

при этом граничными условиями являются два точечных условия, а именно известные значения u_0 и u_N. Уравнение (7) можно, например, рассматривать с той точки зрения, что первые его три члена аппроксимируют производную вектора u_R по x, в то время как последние два члена — производную по y.

5. Рекуррентные соотношения

Будем искать решение (4.7) в виде

$$u_{R+1}=A_Ru_R+b_R, \tag{1}$$

де A_R и b_R не зависят от u_R. Существование такого решения было показано в разд. 2; см. также следующую главу.

Подставляя (1) в равенство

$$u_{R+1}-2u_R+u_{R-1}-Qu_R+r_R=0 \tag{2}$$

и разрешая относительно u_R, получаем

$$u_R=[2I+Q-A_R]^{-1}(u_{R-1}+b_R+r_R). \tag{3}$$

Сравнивая (3) с (1), немедленно получаем

$$A_{R-1} = [2I + Q - A_R]^{-1},$$
$$b_{R-1} = A_{R-1}(b_R + r_R). \tag{4}$$

Подставляя в (1) $R = N - 1$, получаем для (4) начальные условия

$$A_{N-1} = 0, \quad b_{N-1} = u_N. \tag{5}$$

Итак, вычислительная процедура опять сводится к решению задачи Коши (4) и запоминанию результата. Тогда соотношение (1) является задачей Коши с начальным значением u_0.

6. Связь с динамическим программированием

Поскольку изложенная вычислительная схема напоминает как метод динамического программирования, так и метод преобразования Риккати, то было бы интересно точно определить связь между ними.

Пусть A_R и b_R определяются из (3.4) и (3.5). Обозначим

$$B_{R+1} = I - A_R,$$
$$d_{R+1} = b_R. \tag{1}$$

Используя рекуррентные уравнения из разд. 3, найдем уравнения для B_R и d_R:

$$B_R = I - [I + Q - B_{R+1}]^{-1},$$
$$d_R = [I - B_R](d_{R+1} + r_R) \tag{2}$$

и

$$u_R = [I - B_R]\, u_{R-1} + d_R \tag{3}$$

с начальными условиями

$$B_N = I, \quad d_N = u_N. \tag{4}$$

Последние три уравнения точно совпадают с уравнениями, полученными методом динамического программирования.

7. Невырожденность и устойчивость

Поскольку матрицы A_R, фигурирующие в методе преобразования Риккати, связаны с матрицами, полученными методом динамического программирования, соотношением

$$A_R = I - B_{R+1}, \tag{1}$$

то очевидно, что

$$0 < A_R < I, \quad R < N - 1, \tag{2}$$

поскольку

$$0 < B_R < I, \quad R < N, \tag{3}$$

как это известно из разд. 7 гл. 5. Поэтому ясно, что все требуемые матрицы существуют, и нужные обращения можно выполнить.

Для доказательства устойчивости поступим, как и ранее. Предположим, что мы используем рекуррентное уравнение

$$A_{R-1} = [2I + Q - A_R]^{-1}, \tag{4}$$

и допускаем при этом некоторую ошибку. Тогда можно считать, что в действительности мы решаем уравнение

$$\bar{A}_{R-1} = [2I + Q - \bar{A}_R]^{-1}, \tag{5}$$

где

$$\bar{A}_R = A_R + E_R \tag{6}$$

состоит из желаемого решения и ошибки.

Выполняя очевидные алгебраические преобразования, получаем, как и выше, что

$$E_{R-1} \simeq [2I + Q - A_R]^{-1} E_R [2I + Q - A_R]^{-1}, \tag{7}$$

если начальная ошибка достаточно мала. Таким образом,

$$E_{R-1} \simeq A_{R-1} E_R A_{R-1}. \tag{8}$$

Вспоминая, что в силу (2)

$$\| A_{R-1} \| < 1, \tag{9}$$

получаем из (8) неравенство для норм:

$$\| E_{R-1} \| < \| E_R \|. \tag{10}$$

Таким образом, эта процедура численно устойчива [1]). Аналогичное утверждение справедливо для уравнения

$$b_{R-1} = A_{R-1} (b_R + r_R) \tag{11}$$

и для уравнения

$$u_{R+1} = A_R u_R + b_R. \tag{12}$$

Итак, приходим к заключению, что изложенный метод численно устойчив.

8. Связь с методом исключения Гаусса

Стандартные конечно-разностные уравнения (4.5) можно записать как систему линейных алгебраических уравнений

$$Pw = d, \tag{1}$$

где P — матрица размера $(N - 1)(M - 1)$, определяемая способом дискретизации, w — вектор неизвестных во внутренних узлах сетки, а правая часть d определяется из граничных усло-

[1]) Следует помнить, что в методе преобразования Риккати решение строится в обратном направлении.— *Прим. ред.*

вий. Используя обозначения из разд. (4), перепишем (1) в блочной тридиагональной форме:

$$\begin{vmatrix} [2I+Q] & -I & & & 0 \\ -I & [2I+Q] & -I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ 0 & & & -I & [2I+Q] \end{vmatrix} \begin{vmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_{N-2} \\ u_{N-1} \end{vmatrix} = \begin{vmatrix} u_0+r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_{N-1} \\ u_N+r_N \end{vmatrix}. \tag{2}$$

Систему (2) можно решить методом исключения Гаусса, учитывая блоки, целиком состоящие из нулей. Видно, что (5.4) соответствует приведению матрицы к треугольной форме (триангуляризации), а (5.1)— обратной подстановке. Связь между триангуляризацией, динамическим программированием и инвариантным погружением уже отмечалась ранее. Все достоинства излагаемого метода станут очевидными при решении уравнений эллиптического типа на нерегулярных областях.

9. Связь с уравнением Риккати

Выразим A_R из (5.4) через A_{R-1}:

$$A_R = Q + 2I - A_{R-1}^{-1}, \tag{1}$$

или

$$A_R = Q + 2I - [I - (I - A_{R-1})]^{-1}. \tag{2}$$

Поскольку спектральный радиус матрицы $I - A_{R-1}$ в силу (7.2) меньше единицы, то $[I - (I - A_{R-1})]^{-1}$ можно представить в виде ряда

$$[I - (I - A_{R-1})]^{-1} = I + (I - A_{R-1}) + (I - A_{R-1})^2 + \ldots . \tag{3}$$

Теперь можно выразить разность $A_R - A_{R-1}$ следующим образом:

$$A_R - A_{R-1} = Q - (I - A_{R-1})^2 - (I - A_{R-1})^3 - \ldots . \tag{4}$$

Обозначив $I - A_R$ через B_R, получим

$$B_R - B_{R-1} \simeq Q - B_{R-1}^2. \tag{5}$$

Это уравнение похоже на дифференциальное уравнение Риккати

$$dS/dx = Q - S^2. \tag{6}$$

Уравнение Риккати (6) получается точно в непрерывном динамическом программировании или в непрерывном методе инвариантного погружения.

10. Инвариантное погружение

Для решения нашей двухточечной задачи можно также применить метод инвариантного погружения. Мы можем работать либо с системой разностных уравнений, как в разд. 5, либо непосредственно начать с уравнения второго порядка. Пойдем по второму пути.

Замечая, что решение задачи зависит от длины области, запишем [1])

$$u(i+1, N) - [2I + Q]\, u(i, N) + u(i-1, N) + r(i) = 0, \tag{1}$$

где

$$u(0, N) = c, \quad u(N, N) = d. \tag{2}$$

Вследствие линейности задачи решение (1) — (2) для любого $i \leqslant N$ можно представить в виде

$$u(i, N) = U(i, N)\, d + p(i, N), \tag{3}$$

где матрица $U(i, N)$ является решением задачи

$$U(i+1, N) - [2I + Q]\, U(i, N) + U(i-1, N) = 0,$$
$$U(0, N) = 0, \quad U(N, N) = I, \tag{4}$$

а вектор $p(i, N)$ удовлетворяет уравнению

$$p(i+1, N) - [2I + Q]\, p(i, N) + p(i-1, N) + r(i) = 0,$$
$$p(0, N) = c, \quad p(N, N) = 0. \tag{5}$$

Эту задачу можно решить, определив закон изменения решения в зависимости от величины N.

Прежде всего рассмотрим (4) на интервалах длины N и $N+1$, т. е. задачи

$$U(i+1, N) - [2I + Q]\, U(i, N) + U(i-1, N) = 0,$$
$$U(0, N) = 0, \quad U(N, N) = I, \tag{6}$$

и

$$U(i+1, N+1) - [2I + Q]\, U(i, N+1) +$$
$$+ U(i-1, N+1) = 0,$$
$$U(0, N+1) = 0, \quad U(N+1, N+1) = I. \tag{7}$$

Ясно, что благодаря линейности этих уравнений их решения отличаются только мультипликативной постоянной. Используя граничные условия, находим, что

$$U(i, N+1) = U(i, N)\, U(N, N+1). \tag{8}$$

---

[1]) В этом разделе удобнее пользоваться обозначением $u(i, N)$, а не $u_i(N)$.

Это и есть фундаментальное соотношение погружения. Определим $R(N)$ как

$$R(N) = U(N, N+1). \tag{9}$$

Из граничного условия получаем

$$R(0) = 0. \tag{10}$$

Подставляя в (7) $i = N$, находим

$$I - [2I + Q]\, R(N) + U(N-1, N+1) = 0 \tag{11}$$

и, подставив в (8) $i = N-1$, получим

$$U(N-1, N+1) = R(N-1)\, R(N). \tag{12}$$

Теперь можно исключить $U(N-1, N+1)$ из (11) и (12), что приведет к рекуррентному уравнению для $R(N)$

$$R(N) = [2I + Q - R(N-1)]^{-1}. \tag{13}$$

Таким образом, чтобы найти $U(n, N)$, необходимо решить уравнение

$$R(i) = [2I + Q - R(i-1)]^{-1}, \quad R(0) = 0, \tag{14}$$

для $i = 1, 2, \ldots, n$. Затем следует добавить (8) и (9)

$$U(n, i+1) = U(n, i)\, R(i) \tag{15}$$

при условии

$$U(n, n) = R(n). \tag{16}$$

Теперь остается решить (14) и (15) для $i = n, \ldots, N-1$. Для решения задачи (5) поступим аналогичным образом. Записав задачу для отрезков длины N и $N+1$, получаем

$$p(i+1, N) - [2I + Q]\, p(i, N) + p(i-1, N) + r(i) = 0,$$
$$p(0, N) = c, \quad p(N, N) = 0, \tag{17}$$

и

$$p(i+1, N+1) - [2I + Q]\, p(i, N+1) + p(i-1, N+1) + r(i) = 0,$$
$$p(0, N+1) = c, \quad p(N+1, N+1) = 0. \tag{18}$$

Чтобы получить связь между $p(i, N)$ и $p(i, N+1)$, опять используем линейность. Рассмотрев разность $p(i, N+1) - p(i, N)$, заметим, что она обращается в нуль при $i = 0$. Следовательно, эта разность должна входить как постоянный множитель в решение (4). Для определения этой постоянной используем граничные условия и придем к соотношению погружения

$$p(i, N+1) = p(i, N) + U(i, N)\, p(N, N+1). \tag{19}$$

Определим $s(N)$ как

$$s(N) = p(N, N+1), \tag{20}$$

и тогда из (17) получим

$$s(0) = c. \tag{21}$$

Положив в (18) $i = N$, можно записать

$$-[2I + Q]\, s(N) + p(N-1, N+1) + r(N) = 0. \tag{22}$$

Подставив далее $i = N - 1$ в (19), получим

$$p(N-1, N+1) = s(N-1) + R(N-1)\, s(N). \tag{23}$$

Используя (14), (22) и (23), получаем рекуррентное соотношение

$$s(N) = R(N)\,[s(N-1) + r(N)]. \tag{24}$$

Теперь можно полностью описать метод инвариантного погружения для определения $u(n, N)$ — решения уравнений (1) — (2) для некоторого $n \leqslant N$. Процесс начинается с решения уравнений

$$\begin{aligned} R(i) &= [2I + Q - R(i-1)]^{-1}, \\ s(i) &= R(i)\,[s(i-1) + r(i)] \end{aligned} \tag{25}$$

с начальными условиями

$$R(0) = 0, \quad s(0) = c. \tag{26}$$

При $i = n$ добавляются уравнения

$$\begin{aligned} U(n, i+1) &= U(n, i)\, R(i), \\ p(n, i+1) &= p(n, i) + U(n, i)\, s(i) \end{aligned} \tag{27}$$

с начальными условиями

$$\begin{aligned} U(n, n) &= I, \\ p(n, n) &= 0. \end{aligned} \tag{28}$$

Далее опять решаются (25) и (27) до тех пор, пока не будут найдены $U(n, N)$ и $p(n, N)$. Наконец,

$$u(n, N) = U(n, N)\, c + p(n, N). \tag{29}$$

Ясно, что в методе инвариантного погружения при решении одних и тех же фундаментальных уравнений мы уже не должны запоминать $R(i)$ и $s(i)$. В этом состоит основное достоинство метода. Его недостаток заключается в том, что для каждого вектора $u(k, N)$, который мы хотим получить, нужно добавлять другую систему уравнений, аналогичных (27), с начальным условием при $i = k$.

Упражнения

1. Было показано, что система (25) имеет единственное решение и эти уравнения численно устойчивы. Показать, что это справедливо и для системы (27).

2. Применить метод инвариантного погружения для решения уравнения $u''(t) + g(t)\,u(t) = f(t)$, $u(0) = c_1$, $u(\tau) = c_2$.

11. Непрерывное инвариантное погружение

На вопрос о получении численного решения граничной задачи эллиптического типа можно было бы взглянуть с иной точки зрения. Мы могли бы оставить все переменные непрерывными и стараться построить непрерывную задачу Коши. В качестве примера рассмотрим следующую задачу.

Пусть $u(x, y)$ является решением уравнения

$$u_{xx} + u_{yy} = 0 \tag{1}$$

на прямоугольнике $0 \leqslant x \leqslant a$, $0 \leqslant y \leqslant 1$ с граничными условиями

$$u(0, y) = u(x, 0) = u(x, 1) = 0 \tag{2}$$

и еще одним новым граничным условием

$$u_x(a, y) = g(y). \tag{3}$$

Чтобы подчеркнуть зависимость u от длины прямоугольника, запишем

$$u(x, y) = u(x, y, a). \tag{4}$$

Пусть $v(x, y, a, s)$ — решение уравнения

$$v_{xx} + v_{yy} = 0 \tag{5}$$

на том же самом прямоугольнике с граничными условиями

$$v(0, y, a, s) = v(x, 0, a, s) = v(x, 1, a, s) = 0 \tag{6}$$

и

$$v_x(a, y, a, s) = \delta(y - s), \quad 0 < s < 1, \tag{7}$$

где $\delta(y)$ есть дельта-функция Дирака. Таким образом, функция $v(x, y, a, s)$ тесно связана с функцией Грина. Использование дельта-функции можно сделать вполне законным с помощью теории обобщенных функций. Перейдем к формальному изложению.

Сравнивая системы для u и v, с помощью принципа суперпозиции получаем:

$$u(x, y, a) = \int_0^1 v(x, y, a, z)\, g(z)\, dz. \tag{8}$$

Следовательно, если мы сможем определить v, то задача будет решена.

Для определения v начнем с дифференцирования системы (5) — (7) по длине отрезка a. Тогда получим, что функция v_3 удовлетворяет уравнению [1])

$$(v_3)_{11} + (v_3)_{22} = (v_3)_{xx} + (v_3)_{yy} = 0, \tag{9}$$

при этом граничные условия получаются из (6):

$$v_3(0, y, a, s) = v_3(x, 0, a, s) = v_3(x, 1, a, s) = 0. \tag{10}$$

Граничное условие в окончательном виде получается дифференцированием (7) по a, что приводит к уравнению

$$v_{11}(a, y, a, s) + v_{13}(a, y, a, s) = 0. \tag{11}$$

Сравнивая (9) — (11) с (5) — (7) и применяя принцип суперпозиции, получаем фундаментальное уравнение погружения

$$v_3(x, y, a, s) = -\int_0^1 v(x, y, a, z)\, v_{11}(a, z, a, s)\, dz. \tag{12}$$

Поскольку v — решение уравнения Лапласа, то (12) можно переписать в виде

$$v_3(x, y, a, s) = \int_0^1 v(x, y, a, z)\, v_{22}(a, z, a, s)\, dz. \tag{13}$$

Определим $r(a, w, s)$ как

$$r(a, w, s) = v(a, w, a, s). \tag{14}$$

Из (6) получаем начальное и граничные условия

$$r(0, w, s) = 0, \quad r(a, 0, s) = r(a, 1, s) = 0. \tag{15}$$

Продифференцировав (14) по a, получим дифференциальное уравнение для r:

$$r_a(a, w, s) = v_1(a, w, a, s) + v_3(a, w, a, s), \tag{16}$$

которое с учетом (7) принимает вид

$$r_a(a, w, s) = \delta(w - s) + v_3(a, w, a, s). \tag{17}$$

Наконец, подставив вместо v_3 выражение (13) при $x = a$, получим

$$r_a(a, w, s) = \delta(w - s) + \int_0^1 r(a, w, z)\, r_{22}(a, z, s)\, dz. \tag{18}$$

---

[1]) Во избежание недоразумений мы будем пользоваться обозначением v_i, $i = 1, 2, 3, 4$, для записи производной v по i-й переменной. Таким образом, $v_3(x, y, a, s) = v_a(x, y, a, s)$.

Для определения $u(x, y, s)$ решаем (18) при $0 \leqslant a \leqslant x$ с начальными условиями (15). При $a = x$ добавляем уравнение (13), которое с учетом (14) можно записать в виде

$$v_3(x, y, a, s) = \int_0^1 v(x, y, a, z)\, r_{22}(a, z, s)\, dz, \tag{19}$$

что представляет собой задачу Коши с начальным условием

$$v(x, y, x, s) = r(x, y, s). \tag{20}$$

Наконец, при заданном значении l функция $u(x, y, l)$ получается интегрированием уравнения (8).

Из уравнения (18) можно получить ряд стандартных численных методов. Если, например, дискретизировать переменные w и s, т. е.

$$w_i = ih, \quad s_j = jh, \tag{21}$$

то получится метод прямых. Если для вычисления интеграла в (18) воспользоваться формулой трапеций, то опять же получится стандартный метод прямых. Если теперь дискретизировать a, то мы придем к конечно-разностному методу. Преимущество работы с выражением (18) состоит в том, что это уравнение позволяет использовать много новых подходов к решению. Так, например, можно было бы использовать в (18) гауссову квадратурную формулу. Можно ожидать, что в результате получится более точный метод, чем обычные. Однако в этом направлении еще много предстоит сделать.

Упражнения

1. Распространить полученные результаты на случай уравнения Лапласа с граничными условиями

$$u(0, y) = f(y), \quad u_x(a, y) = g(y), \quad u(x, 0) = a(x),$$
$$u(x, 1) = b(x).$$

2. Показать, что способ дискретизации (21) в совокупности с формулой трапеций преобразуют (18) в матричное уравнение Риккати

$$R_a = I - (1/h^2)\, RQR.$$

3. Решить задачу (1) — (2) с дополнительным условием

$$u(a, y) = f(y).$$

Указание. Использовать решение для v и затем решить уравнение Фредгольма при $x = a$.

4. Доказать, что

$$r(x, y, z) = r(x, z, y).$$

12. Обобщенные преобразования Риккати

Преобразование Риккати можно обобщить и для первоначальной непрерывной задачи. Рассмотрим опять уравнение Лапласа

$$u_{xx} + u_{yy} = 0 \tag{1}$$

на прямоугольнике $0 \leqslant x \leqslant a$, $0 \leqslant y \leqslant b$, на всех сторонах которого заданы значения функции u. Будем искать решение в виде

$$u(x, y) = \int_0^b r(x, y, z)\, u_x(x, z)\, dz + s(x, y). \tag{2}$$

Дифференцируя (2) по x, получаем

$$u_x(x, y) = \int_0^b r_x(x, y, z)\, u_x(x, z)\, dz +$$

$$+ \int_0^b r(x, y, z)\, u_{xx}(x, z)\, dz + s_x(x, y) \tag{3}$$

и с учетом (1)

$$u_x(x, y) = \int_0^b r_x(x, y, z)\, u_x(x, z)\, dz -$$

$$- \int_0^b r(x, y, z)\, u_{zz}(x, z)\, dz + s_x(x, y). \tag{4}$$

Дважды дифференцируя (2) по y, получаем

$$u_{yy}(x, y) = \int_0^b r_{yy}(x, y, z)\, u_x(x, z)\, dz + s_{yy}(x, y). \tag{5}$$

Наконец, воспользовавшись этим уравнением и известным свойством дельта-функции, перепишем (4) в виде

$$\int_0^b \delta(y - z)\, u_x(x, z)\, dz = \int_0^b r_x(x, y, z)\, u_x(x, z)\, dz -$$

$$- \int_0^b \int_0^b r(x, y, w)\, r_{ww}(x, w, z)\, dw\, dz -$$

$$- \int_0^b r(x, y, z)\, s_{zz}(x, z)\, dz + s_x(x, y). \tag{6}$$

Приравнивая коэффициенты при u_x, получаем

$$r_x(x, y, z) = \delta(y - z) + \int_0^b r(x, y, w)\, r_{ww}(x, w, z)\, dw,$$
$$s_x(x, y) = \int_0^b r(x, y, z)\, s_{zz}(x, z)\, dz. \tag{7}$$

Потребовав, чтобы (2) выполнялось на границах, придем при $x = a$, $y = 0$ и $y = b$ к равенствам

$$r(x, y, z) = 0, \quad s(x, y) = u(x, y), \tag{8}$$

что дает исходное и дополнительное уравнения для определения r и s. Таким образом, r и s определяются в обратном направлении, т. е. от $x = a$ к $x = 0$. Поскольку $u(0, y)$ известно, то (2) приводит к уравнению Фредгольма для определения недостающего начального условия $u_x(0, y)$ с учетом $r(0, y, z)$ и $s(0, y)$. Наконец, комбинируя (1) и (5), получаем задачу Коши для определения u_x:

$$u_{xx}(x, y) = -\int_0^b r_{yy}(x, y, z)\, u_x(x, z)\, dz + s_{yy}(x, y), \tag{9}$$

при решении которой используются значения r и s, хранящиеся в памяти. Требуемые значения u тогда определяются из уравнения (2).

13. Бигармоническое уравнение

Непрерывный метод можно применить и для решения бигармонического уравнения; при этом получается неожиданный результат. Рассмотрим, например, задачу деформации прямоугольной пластины с тремя закрепленными краями, которая состоит в решении уравнения

$$w_{xxxx} + 2w_{xxyy} + w_{yyyy} = 0 \tag{1}$$

с граничными условиями на закрепленных краях

$$w = w_n = 0 \tag{2}$$

и с условиями

$$w_{xx} = f(y), \quad w_{xxx} + 2w_{xyy} = g(y), \tag{3}$$

заданными на свободном крае, т. е. при $x = a$. Для решения задачи (1) — (3) введем функции $u(x, y, z)$ и $v(x, y, z)$, удовлетворяющие (1) и (2), т. е.

$$u_{xxxx} + 2u_{xxyy} + u_{yyyy} = 0,$$
$$v_{xxxx} + 2v_{xxyy} + v_{yyyy} = 0 \tag{4}$$

с граничными условиями

$$u = u_n = v = v_n = 0 \tag{5}$$

на краях $x = 0$, $y = 0$ и $y = 1$. Потребуем, чтобы на крае $x = a$ выполнялись равенства

$$\begin{aligned} u_{xx} &= \delta(y - z), \\ u_{xxx} + 2u_{xyy} &= 0, \\ v_{xx} &= 0, \\ v_{xxx} + 2v_{xyy} &= -\delta(y - z). \end{aligned} \tag{6}$$

Итак, w определяется с помощью суперпозиции

$$w(x, y) = \int_0^1 u(x, y, z) f(z)\, dz - \int_0^1 v(x, y, z) g(z)\, dz. \tag{7}$$

Будем теперь считать, что u и v являются, кроме того, функциями длины a пластины. Детальное рассмотрение оставляем читателю. Важность изложенного подхода объясняется следующими соображениями. Если мы поступим, как в разд. 11, то получим, что функции p, q, r и s, определяемые как прогибы на крае $x = a$, т. е.

$$\begin{aligned} p(x, y, z) &= u_1(x, y, x, z), \\ q(x, y, z) &= v_1(x, y, x, z), \\ r(x, y, z) &= u(x, y, x, z), \\ s(x, y, z) &= v(x, y, x, z), \end{aligned} \tag{8}$$

удовлетворяют интегро-дифференциальным уравнениям

$$\begin{aligned} p_a(a, x, y) &= \delta(x - y) + 2\int_0^1 p(a, y, z)\, p_{zz}(a, z, y)\, dz - \\ &\quad - \int_0^1 q(a, x, z)\, r_{zzzz}(a, z, y)\, dz, \\ q_a(a, x, y) &= p(a, x, y) + 2\int_0^1 p(a, x, z)\, q_{zz}(a, z, y)\, dz - \\ &\quad - \int_0^1 q(a, x, z)\, s_{zzzz}(a, z, y)\, dz, \\ r_a(a, x, y) &= p(a, x, y) + 2\int_0^1 r(a, x, z)\, p_{zz}(a, z, y)\, dz - \\ &\quad - \int_0^1 s(a, x, z)\, r_{zzzz}(a, z, y)\, dz, \end{aligned} \tag{9}$$

$$s_a(a, x, y) = q(a, x, y) + r(a, x, y) + 2\int_0^1 r(a, x, z)\, q_{zz}(a, z, y)\, dz -$$

$$- \int_0^1 s(a, x, z)\, s_{zzzz}(a, z, y)\, dz.$$

Нетрудно показать, что эти уравнения в совокупности с соответствующими исходными и дополнительными условиями обладают тем свойством, что

$$p(a, x, y) = p(a, y, x), \quad s(a, x, y) = s(a, y, x),$$
$$q(a, x, y) = q(a, y, x). \tag{10}$$

Эти свойства симметрии известны как соотношения взаимности Бетти, играющие важную роль в строительной механике.

Упражнения

1. Пусть u и v определяются соотношениями (4) — (6). Положим

$$u = u(x, y, a, z), \quad v = v(x, y, a, z).$$

Сформулировать задачи Коши для отыскания μ и v как функций от a.

2. Используя результаты упражнения 1, доказать соотношения взаимности (10).

14. Случайное блуждание

Коснемся кратко фундаментальной взаимосвязи между теорией потенциала и теорией вероятностей, что позволит дать простую интерпретацию некоторых результатов, полученных с помощью инвариантного погружения.

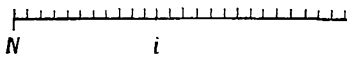


Рис. 2

Рассмотрим для начала задачу об одномерном случайном блуждании (рис. 2). Предполагается, что частица, находящаяся в точке i, с равной вероятностью может двигаться вправо или влево, при этом i принимает значения $N + 1, \ldots, M - 1$. Точки N и M являются поглощающими барьерами в том смысле, что процесс блуждания заканчивается, как только частица достигает одной из этих точек.

Введем функцию

$$v(i) = (\text{вероятность того, что, начиная с } i, \text{ точка достигнет положения } M \text{ прежде, чем положения } N), \quad N + 1 \leqslant i \leqslant M - 1. \tag{1}$$

Область определения этой функции можно расширить с помощью граничных условий

$$v(N) = 0, \quad v(M) = 1. \tag{2}$$

Из вышеизложенного вытекает, что $v(i)$ удовлетворяет разностному уравнению

$$v(i) = \frac{1}{2}[v(i+1) + v(i-1)], \tag{3}$$

или

$$v(i+1) - 2v(i) + v(i-1) = 0. \tag{4}$$

Очевидно, что это уравнение является дискретным аналогом одномерного уравнения Лапласа

$$d^2v/dx^2 = 0, \quad v(a) = 0, \quad v(b) = 1. \tag{5}$$

Рассмотрим теперь процесс случайного блуждания на плоскости, определенный на узлах сетки (m, n), $M_1 \leqslant m \leqslant M_2$, $M_1 \leqslant n \leqslant N_2^*$, причем частица с равными вероятностями может двигаться в любом из четырех возможных направлений (рис. 3).

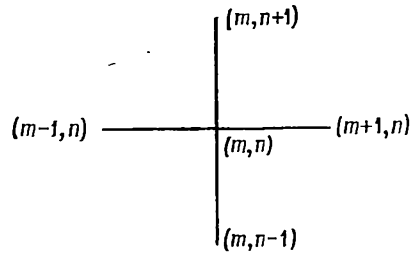


Рис. 3

Введем функцию

$u(m, n)$ = (вероятность того, что, начиная из положения (m, n), частица достигнет ребра (M_2, n) раньше всех других ребер). (6)

Ясно, что $u(m, n)$ удовлетворяет соотношению

$$u(m, n) = \frac{1}{4}[u(m-1, n) + u(m+1, n) + u(m, n+1) + u(m, n-1)], \tag{7}$$

которое точно совпадает с конечно-разностной аппроксимацией двумерного уравнения Лапласа

$$u_{xx} + u_{yy} = 0, \tag{8}$$

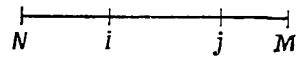
которое мы изучали ранее.

15. Инвариантное погружение и случайное блуждание

Пусть $M + 1 \geqslant j \geqslant i$. Введем функцию

$$u(i, j) = \text{(вероятность того, что, начиная из положения } i\text{, точка достигнет положения } j \text{ прежде, чем положения } N\text{).} \quad (1)$$

Рис. 4

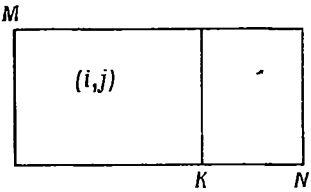


Из рис. 4 ясно, что

$$u(i, j+1) = u(i, j)\, u(j, j+1). \quad (2)$$

Это соотношение является скалярным вариантом равенства (10.8). Аналогично из рис. 5 получим матричное уравнение.

Рис. 5



16. Другой способ погружения

В этом разделе мы приведем другой пример неклассической процедуры погружения. Несмотря на свою простоту, этот вариант поможет проиллюстрировать один из многих способов получения новых численных методов с помощью аппарата инвариантного погружения.

Предположим, что u удовлетворяет уравнению

$$u_{xx} + u_{yy} + \lambda g(x, y)\, u = 0 \quad (1)$$

на области R, причем на границе Γ этой области имеет место равенство

$$u = f(x, y). \quad (2)$$

Во многих случаях параметр λ имеет конкретный физический смысл. Будем предполагать, что

$$\lambda \geqslant 0, \quad (3)$$

и попытаемся выяснить, как изменяется решение исходной задачи в зависимости от величины λ, т. е.

$$u = u(x, y, \lambda). \tag{4}$$

Продифференцировав (1) и (2) по λ, получим

$$(u_\lambda)_{xx} + (u_\lambda)_{yy} + \lambda g(x, y)\, u_\lambda = -g(x, y)\, u \tag{5}$$

с граничным условием на Γ

$$u_\lambda = 0. \tag{6}$$

Пусть $G(x, y, a, b, \lambda)$ — функция Грина для (5) и (6), тогда u_λ определяется выражением

$$u_\lambda(x, y, \lambda) = -\iint_R g(a, b)\, u(a, b)\, G(a, b, x, y, \lambda)\, da\, db. \tag{7}$$

Рассуждая формально, мы можем определить функцию Грина как решение уравнения

$$G_{xx}(x, y, a, b, \lambda) + G_{yy}(x, y, a, b, \lambda) + \\ + \lambda g(x, y)\, G(x, y, a, b, \lambda) = \delta(x-a,\ y-b) \tag{8}$$

с граничным условием на Γ

$$G(x, y, a, b, \lambda) = 0. \tag{9}$$

Дифференцируя (8) и (9) по λ, находим

$$(G_\lambda)_{xx} + (G_\lambda)_{yy} + \lambda g(x, y)\, G_\lambda = -g(x, y)\, G \tag{10}$$

с граничным условием на Γ

$$G_\lambda = u. \tag{11}$$

Таким образом, G_λ определяется формулой

$$G_\lambda(x, y, a, b, \lambda) = \\ = -\iint_R g(a_1, b_1)\, G(a_1, b_1, a, b, \lambda)\, G(x, y, a_1, b_1, \lambda)\, da_1\, db_1. \tag{12}$$

Величину $u(x, y, \lambda)$, как функцию от λ, теперь можно определить следующим образом. Для $\lambda = 0$ решим уравнение

$$u_{xx} + u_{yy} = 0 \tag{13}$$

с граничным условием (2) и уравнение

$$G_{xx} + G_{yy} = \delta(x - a,\ y - b) \tag{14}$$

с граничным условием (9). Теперь можно решить (7) и (12) как задачи Коши по λ с ограничениями (2) и (9).

Упражнение

Рассмотрим задачу

$$u_{xx} + u_{yy} = g(x, y)$$

с граничным условием $u = \lambda f$ на Γ. Определить u как функцию λ.

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Раздел 1. Инвариантное погружение первоначально возникло как обобщение принципов инвариантности, введенных В. А. Амбарцумяном и С. Чандрасекхаром. См. работы

Амбарцумян В. А.

Рассеяние света мутной средой, *ДАН СССР*, 38 (1943), 229—232.

Чандрасекхар (Chandrasekhar S.)

Radiative transfer, Dover, New York, 1960.

Дальнейшие ссылки и приложения можно найти в работах

Винг (Wing G. M.)

Introduction to transport theory, Wiley, New York, 1962.

Беллман (Bellman R.)

Vistas of modern mathematics, Univ. of Kentucky Press, Lexington, Kentucky, 1968.

Беллман и Винг (Bellman R., Wing G. M.)

Invariant imbedding (в печати).

Раздел 2. Преобразование Риккати также было названо пошаговым методом. К сожалению, этот термин повлек за собой многочисленные недоразумения, возникшие из-за путаницы между двухшаговым методом преобразования Риккати и одношаговым методом инвариантного погружения, развитым нами в разд. 3; см. работы

Годунов С. К., Рябенький В. С.

Введение в теорию разностных схем, Физматгиз, М., 1962.

Гельфанд И. М., Фомин С. В.

Вариационное исчисление, Физматгиз, М., 1961.

Рыбицки, Ушер (Rybicki G. B., Usher P. D.)

The generalized Riccati transformations as a simple alternative to invariant imbedding, *Astrophys. J.*, 146 (1966), 871—879.

Мейер (Meyer G. H.)

The invariant imbedding equations for multipoint boundary-value problems, *SIAM J. Appl. Math.*, 18 (1970), 433—453.

Раздел 3. См. статьи

Калаба (Kalaba R.)

A one-sweep method for linear difference equations with two-point boundary conditions, Electronic Sciences Laboratory, Univ. of Southern California, USCEE 23, 1969.

Энджел, Калаба (Angel E., Kalaba R.)

A one-sweep numerical method for vector-matrix difference equations with two-point boundary conditions, *J. Optim. Theory Appl.*, 6 (1970), 345—355.

Беллман, Каживада, Калаба (Bellman R., Kagiwada H., Kalaba R.)

Invariant imbedding and the numerical integration of boundary-value problems for unstable linear systems of ordinary differential equations, *Comm. ACM*, 10 (1967), 100—102.

Раздел 4. Этот способ дискретизации является стандартной конечно-разностной аппроксимацией. Доказательства сходимости см. в любой из работ, указанных к разд. 9, гл. 4.

Раздел 5. То, что решение можно искать в таком виде, известно уже давно; см., например,

Корнок (Cornock A. F.)

The numerical solution of Poisson's and the biharmonic equations by matrices, *Proc. Camb. Phil. Soc.*, Part 4, 50 (1954), 524—535.

Карлквист (Karlquist O.)

Numerical solution of elliptic difference equations by matrix methods, *Tellus*, 4 (1952), 374—384.

Обширная библиография приведена в обзорной статье

Дорр (Dorr F. W.)

The direct solution of the discrete Poisson equation on a rectangle, *SIAM Rev.*, 12 (1970), 248—263.

Раздел 7. Устойчивость других конечных методов обсуждается в работе

Россер (Rosser J. B.)

The directe solution of difference analogs of Poisson's equation, Univ. of Wisconsin, Mathematical Research Center, Tech. Summary Rept., 797, 1967.

Раздел 8. См., например,

Розенберг (von Rosenberg D. V.)

Methods for the numerical solution of partial differential equations, Amer. Elsevier, New York, 1969.

Леман (Lehman R. S.)

Dynamic programming and Gaussian elimination, *J. Math. Anal. Appl.*, 15 (1962), 499—501.

Сойлерс, Хикерсон (Soillers W. R., Hickerson N.)

Optimal elimination for sparse symmetric systems as a graph problem, *Quart. Appl. Math.*, 26, 1968, 425—432.

Раздел 9. См. статью

Энджел (Angel E.)

Discrete invariant imbedding and elliptic boundary-value problems over irregular regions, *J. Math. Anal. Appl.*, 23 (1968), 471—484.

Раздел 10. См. статью

Энджел (Angel E.)

Invariant imbedding and partial differential equations, в трудах летней школы по инвариантному погружению 1970 г. Под редакцией R. Bellman and E. Denman, Springer-Verlag, Berlin and New York, 1971.

Раздел 11. См. статью

Энджел, Джейн, Калаба (Angel E., Jain A., Kalaba R.)

Initial value methods in potential theory, Electronic Sciences Laboratory, Univ. of Southern Calif., USCEE 22, 1970.

Бейли (Bailey P. B.)

A rigorous derivation of some invariant imbedding equations of transport theory, *J. Math. Anal. Appl.*, 8 (1964), 144—169.

См. также указанную выше статью Мейера и работу

Голдберг (Goldberg M.)

A generalized invariant imbedding equation, *J. Math. Anal. Appl.*, 34 (1971), 590—601.

Раздел 12. См. статью

Энджел, Джейн (Angel E., Jain A.)

Initial-value transformation for elliptic boundary value problems *J. Math. Anal. Appl.*, 35 (1971), 3.

Были предложены и другие преобразования; см., например,

Мейнард, Скотт (Maynard C., Scott M.)

Invariant imbedding and linear partial differential equations via generalized Riccati transformations, *J. Math. Anal. Appl.*, 36 (1971), 2.

Раздел 13. См. статью

Энджел, Дистефано, Джейн (Angel E., Distefano N., Jain A.)

Invariant imbedding and the reduction of boundary-value problems of thin plate theory to Cauchy formulations, *Int. J. Eng. Sci.* (в печати).

Раздел 14. См. статью

Беллман, Калаба (Bellman R., Kalaba R.)

Random walk, scattering and invariant imbedding I: one dimensional discrete case, *Proc. Nat. Acad. U. S.*, 43 (1957), 930—933.

Раздел 16. По поводу обыкновенных дифференциальных уравнений см. работу

Хасс, Калаба (Huss R., Kalaba R.)

Invariant imbedding and the numerical determination of Green's functions, *J. Opt. Theory Appl.*, (в печати).

Эти идеи широко используются при исследовании интегральных уравнений Фредгольма. Так, например, в уравнении Беллмана — Крейна для резольвенты уравнения Фредгольма длина отрезка рассматривается как параметр, см.

Беллман (Bellman R.)

Functional equations in the theory of dynamic programming VIII, A partial differential equations for the Fredholm resolvent, *Proc. Amer. Math. Soc.*, 8 (1957), 435—440.

Эти результаты основаны на некоторых абстрактных построениях, которые можно найти в работах

Бергман, Шиффер (Bergman S., Schiffer M.)

Kernal functions and elliptic differential equations in mathematical physics, Academic Press, New York, 1953.

Мак-Набб, Шумицки (McNabb A., Schumitzky A.)

Factorisation of operators I: Algebraic theory and examples (в печати). Factorization and Operators III: Initial-value methods for linear two-point boundary-value problems, *J. Math. Anal. Appl.*, 31 (1970), 391—406.

Приложение к одной задаче строительной механики можно найти в работе

Энджел, Дистефано (Angel E., Distefano N.)

Invariant imbedding and the effect of changes in Poisson's ratio in thin plate theory, *Int. J. Eng. Sci.* (в печати).

Глава 7

Нерегулярные области

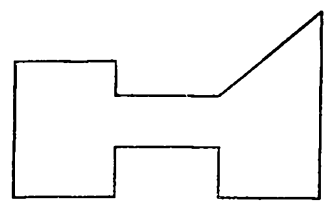
1. Введение

В предыдущих главах мы рассматривали задачи на прямоугольных областях. Можно ожидать, что на таких областях многие методы будут хорошо работать. В этой главе мы хотим рассмотреть некоторые нерегулярные области, представляющие собой серьезную вычислительную проблему для большинства известных методов.

2. Нерегулярные области

Мы будем рассматривать области произвольной формы, удовлетворяющие лишь двум ограничениям. Во-первых, будем предполагать, что узлы сетки, используемой для дискретизации задачи, являются регулярными, т. е. такими, что расстояния между соседними узлами, включая и граничные узлы, равны между собой. Во-вторых, будем считать, что области являются односвязными. Подробнее это означает следующее. Поскольку в нашем методе требуется пересекать область в некотором заданном направлении (в направлении оси x), мы требуем, чтобы данная область была ориентирована таким образом, чтобы любая прямая, проведенная через эту область в другом, отличном от x, направлении (в направлении оси y), целиком лежала внутри области. Таким образом, типичная допустимая область должна иметь форму, аналогичную изображенной на рис. 1.

Рис. 1



Ни одно из этих ограничений не является необходимым, и мы налагаем их, стремясь сделать изложение настолько простым, насколько это возможно. Позднее мы обсудим вопрос о снятии этих ограничений.

Для решения уравнения Лапласа на нерегулярной области мы постараемся выяснить, как следует модифицировать соотноше-

ния (6.3.1) и (6.3.4), когда два соседних вектора неизвестных u_R и u_{R-1} имеют разные размерности. Если размерности векторов u_R и u_{R-1} равны, то уравнения (6.3.1) и (6.3.4) можно применять непосредственно. При этом размерности всех матриц и векторов определяются размерностью векторов u_R и u_{R-1}.

3. Случай I: размерность u_R больше размерности u_{R-1}

Достаточно рассмотреть ситуацию, изображенную на рис. 2, где

$$u_R = (u_{R1}, u_{R2}, \ldots, u_{RM}),$$
$$u_{R-1} = (u_{R-1,1}, u_{R-1,2}, \ldots, u_{R-1,N}), \quad M > N, \tag{1}$$

а

$$w = (w_{N+1}, w_{N+2}, \ldots, w_M) \tag{2}$$

является вектором неизвестных, определяемых из граничных условий. Случай, когда не совпадают верхние границы векторов u_R и u_{R-1}, можно рассмотреть точно таким же образом, как и этот.

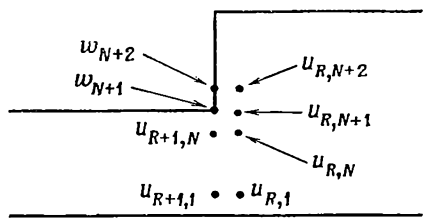


Рис. 2

Обозначим через $\tilde{u}_{R-1}$ вектор

$$\tilde{u}_{R-1} = \begin{bmatrix} u_{R-1} \\ w \end{bmatrix} = (u_{R-1,1}, \ldots, u_{R-1,N}, w_{N+1}, \ldots, w_M). \tag{3}$$

Предположив, что вектор $\tilde{u}_{R-1}$ можно определить каким-либо методом, попытаемся определить вектор u_R с помощью уравнения, аналогичного (6.3.1), иными словами,

$$u_R = \tilde{A}_{R-1} u_{R-1} + \tilde{b}_{R-1}, \tag{4}$$

где, разумеется, размерности $\tilde{A}_{R-1}$ и $\tilde{b}_{R-1}$ равны M. Поскольку размерности A_R и b_R также равны M, то $\tilde{A}_{R-1}$ и $\tilde{b}_{R-1}$, определяемые из (6.3.4), а именно

$$\tilde{A}_{R-1} = [2I + Q - A_R]^{-1},$$
$$\tilde{b}_{R-1} = A_{R-1}(b_R + r_R), \tag{5}$$

являются требуемыми матрицей и вектором. Иными словами, поскольку $\tilde{u}_{R-1}$ полностью определено, то величина u_R не зависит от формы границы слева [1]). Все, что нам остается сделать, это определить, как вычислить u_{R-1} и тем самым $\tilde{u}_{R-1}$.

Мы собираемся вычислить u_{R-1} с помощью уравнения, аналогичного (6.3.1),

$$u_{R-1} = A_{R-2}u_{R-2} + b_{R-2}, \tag{6}$$

где размерности A_{R-2} и b_{R-2} равны N. Предположим, что A_{R-2} и b_{R-2} можно определить по формулам

$$\begin{aligned} A_{R-2} &= [2I + Q - A_{R-1}]^{-1}, \\ b_{R-2} &= A_{R-2}(b_{R-1} + r_{R-1}), \end{aligned} \tag{7}$$

где размерности всех фигурирующих в них величин равны N. Теперь мы должны определить соотношения между A_{R-1} и $\tilde{A}_{R-1}$ и между b_{R-1} и $\tilde{b}_{R-1}$.

Определим $\tilde{u}_R$ как

$$\tilde{u}_R = (u_{R1}, \ldots, u_{RN}) \tag{8}$$

и попытаемся определить $\tilde{u}_R$ из u_{R-1} с помощью соотношения вида

$$\tilde{u}_R = A_{R-1}u_{R-1} + b_{R-1}. \tag{9}$$

Наконец, мы потребуем, чтобы соответствующие компоненты векторов u_R и $\tilde{u}_R$ (т. е., u_{R1}, u_{R2}, ..., u_{RN}) совпадали *при любом выборе* u_{R-1} *и* w (или $\tilde{u}_{R-1}$). Эта процедура устанавливает связь между $\tilde{A}_{R-1}$ и A_{R-1} и между $\tilde{b}_{R-1}$ и b_{R-1}. А именно, представив $\tilde{A}_{R-1}$ и $\tilde{b}_{R-1}$ в виде

$$\tilde{A}_{R-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \tilde{b}_{R-1} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{10}$$

где размерности A_{11} и b_1 равны N, и выполнив операции (4) и (9), получим

$$\begin{aligned} A_{R-1} &= A_{11}, \\ b_{R-1} &= A_{12}w + b_1. \end{aligned} \tag{11}$$

---

[1]) Это замечание есть просто переформулировка принципа причинности для линейных систем.

4. Пример

Решим уравнение Лапласа на области, изображенной на рис. 3, где изменение структуры области происходит при $R = 5$.

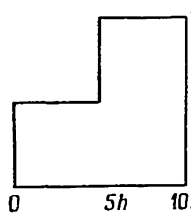


Рис. 3

В этой ситуации вектор

$$w = (u_{55}, u_{56}, u_{57}, u_{58}, u_{59}) \tag{1}$$

состоит из граничных значений. Вычислим выражения

$$\begin{aligned} A_{R-1} &= [2I + Q - A_R]^{-1}, \\ b_{R-1} &= A_{R-1}(b_R + r_R) \end{aligned} \tag{2}$$

с начальными значениями

$$A_9 = 0, \quad b_9 = u_{10}, \tag{3}$$

где размерность всех матриц и векторов равна 9. После четырех итераций по формулам (2) получаем

$$\begin{aligned} \widetilde{A}_5 &= [2I + Q - A_6]^{-1}, \\ \widetilde{b}_5 &= A_5(b_6 + r_6). \end{aligned} \tag{4}$$

Первые четыре компоненты векторов A_5 и b_5 вычисляем с помощью (3.11). Если

$$\begin{aligned} A_5 &= (a_{ij}), & i, j &= 1, \ldots, 4, \\ \widetilde{A}_5 &= (\widetilde{a}_{ij}), & i, j &= 1, \ldots, 9, \\ b_5 &= (b_i), & i &= 1, \ldots, 4, \\ \widetilde{b}_5 &= (\widetilde{b}_i), & i &= 1, \ldots, 9, \end{aligned} \tag{5}$$

то

$$\begin{aligned} a_{ij} &= \widetilde{a}_{ij}, \quad i, j = 1, \ldots, 4, \\ b_i &= \widetilde{b}_i + \sum_{j=1}^{5} a_{i, j+4} w_{j+4}. \end{aligned} \tag{6}$$

Продолжим решение уравнений (2), где размерность всех матриц и векторов равна теперь четырем, до тех пор, пока не определим A_0 и b_0. Далее решим уравнение

$$u_{R+1} = A_R u_R + b_R \tag{7}$$

с начальным условием u_0 и найдем u_5, используя уже найденные значения A_R и b_R, размерность которых пока равна четырем. Сформируем вектор

$$\tilde{u}_5 = \begin{bmatrix} u_5 \\ w \end{bmatrix}$$

и продолжим решение уравнения (7), в котором размерности всех матриц и векторов равны 9.

5. Случай II: размерность u_R меньше размерности u_{R-1}

В этой ситуации тоже достаточно рассмотреть лишь случай, изображенный на рис. 4, где

$$u_{R-1} = (u_{R-1,1},\ u_{R-1,2},\ \ldots,\ u_{R-1,M}), \quad M > N,$$
$$u_R = (u_{R1},\ u_{R2},\ \ldots,\ u_{RN}), \tag{1}$$

а

$$w = (w_{N+1},\ \ldots,\ w_M) \tag{2}$$

определяется граничными условиями. Поступая, как и ранее,

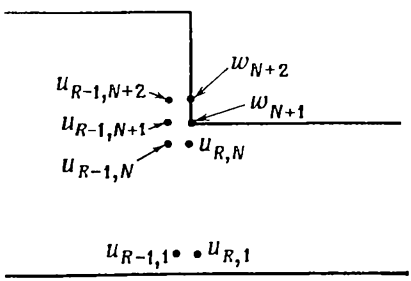


Рис. 4

положим

$$\tilde{u}_R = \begin{bmatrix} u_R \\ w \end{bmatrix} = (u_{R1},\ \ldots,\ u_{RN},\ w_{N+1},\ \ldots,\ w_M). \tag{3}$$

Матрица A_R и вектор b_R раэмерности M таковы, что

$$A_{R-1} = [2I + Q - A_R]^{-1},$$
$$b_{R-1} = A_{R-1}(b_R + r_R). \tag{4}$$

Зная A_{R-1} и b_{R-1}, найдем матрицу $\tilde{A}_{R-1}$ и вектор $\tilde{b}_{R-1}$ размерности N, такие, что $\tilde{u}_R$ можно вычислить как

$$\tilde{u}_R = \tilde{A}_{R-1} u_{R-1} + \tilde{b}_{R-1}. \tag{5}$$

Снова будем считать, что если вектор $\tilde{u}_{R-1}$ имеет вид

$$\tilde{u}_{R-1} = (u_{R-1,1}, \ldots, u_{R-1,M}), \tag{6}$$

то u_R можно определить по формуле

$$u_R = A_{R-1}\tilde{u}_{R-1} + b_{R-1}, \tag{7}$$

и потребуем, чтобы соответствующие координаты (5) и (7) совпадали при любом u_{R-1}. Тогда очевидно, что

$$\tilde{A}_{R-1} = A_{R-1} \oplus 0, \quad \tilde{b}_{R-1} = \begin{bmatrix} b_{R-1} \\ w \end{bmatrix}, \tag{8}$$

где $\oplus$ обозначает прямую сумму, т. е.

$$\tilde{A}_{R-1} = \begin{bmatrix} A_{R-1} & 0 \\ 0 & 0 \end{bmatrix}. \tag{9}$$

Этот результат можно рассматривать как добавление нового множества начальных условий, подобно (6.3.5).

6. Пример

Чтобы найти решения уравнения Лапласа на области, изображенной на рис. 5, поступим следующим образом. Будем решать уравнения

$$\begin{aligned} A_{R-1} &= [2I + Q - A_R]^{-1}, \\ b_{R-1} &= A_{R-1}(b_R + r_R) \end{aligned} \tag{1}$$

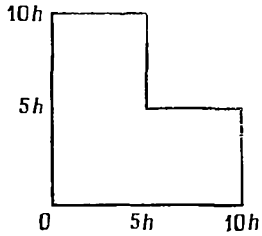


Рис. 5

с начальными условиями

$$A_9 = 0, \quad b_9 = u_{10}, \tag{2}$$

где размерности всех матриц и векторов равны четырем, до тех пор, пока не определим

$$\tilde{A}_5 = [2I + Q - A_6]^{-1}, \quad \tilde{b}_5 = \tilde{A}_5(b_6 + r_6). \tag{3}$$

Вектор

$$w=(u_{55},\ u_{56},\ u_{57},\ u_{58},\ u_{59}) \tag{4}$$

определяется из граничных условий. Матрица A_5 и вектор b_5 размерности 9 формируются по формулам (5.8), или если

$$\begin{aligned}
\widetilde{A}_5&=(\widetilde{a}_{ij}), \quad i,\ j=1,\ \ldots,\ 5,\\
A_5&=(a_{ij}), \quad i,\ j=1,\ \ldots,\ 9,\\
\widetilde{b}_5&=(\widetilde{b}_i), \quad i=1,\ \ldots,\ 4,\\
b_5&=(b_i), \quad i=1,\ \ldots,\ 9,
\end{aligned} \tag{5}$$

то

$$a_{ij}=\begin{cases}\widetilde{a}_{ij}, & i,\ j=1,\ \ldots,\ 4,\\ 0 & \text{в противном случае,}\end{cases}$$

$$b_i=\begin{cases}\widetilde{b}_i, & i=1,\ \ldots,\ 4,\\ w_i, & i=5,\ \ldots,\ 9.\end{cases} \tag{6}$$

Продолжим решение уравнений (1) до тех пор, пока не определим A_0 и b_0, размерность которых равна девяти. Далее из уравнения

$$u_{R+1}=A_Ru_R+b_R \tag{7}$$

с начальным условием u_0 определяется

$$\widetilde{u}_5=\begin{bmatrix}u_5\\ w\end{bmatrix}=A_4u_4+b_4. \tag{8}$$

Наконец, продолжаем решать уравнение (7) с начальным условием u_5, где размерности всех векторов и матриц теперь равны четырем.

7. Невырожденность и устойчивость

В гл. 4 мы показали, что достаточное условие существования необходимых обратных матриц и численной устойчивости данного метода имеет вид

$$0<A_R<1. \tag{1}$$

Покажем, что этот результат остается в силе и в том случае, когда размерности векторов u_R и u_{R-1} (а следовательно, A_R и A_{R-1}) различны.

Для случая, когда размерность вектора u_{R-1} больше размерности u_R, получаем следующий результат. Обращение матрицы $[2I+Q-A_R]$ дает матрицу $\widetilde{A}_{R-1}$, удовлетворяющую неравен-

ствам (1). При построении матрицы $\tilde{A}_{R-1}$ по формулам (5.8) ненулевые собственные значения $\tilde{A}_{R-1}$ не изменяются, а лишь добавляются $N - M$ нулевых собственных значений. Ясно, что матрица $[2I + Q - A_{R-1}]$ положительно определена, и поскольку

$$0 < A_{R-1} < 1, \tag{2}$$

то матрица A_{R-2}, как и все последующие, должна удовлетворять неравенствам (1).

Если размерность вектора u_{R-1} меньше размерности u_R, то при выполнении операций (3.11) можно воспользоваться теоремой отделения Штурма [1]). Тогда мы видим, что (1) имеет место и в этом случае.

8. Снятие ограничений на вид области

Теперь можно обсудить вопрос о том, как можно избавиться от ограничений, наложенных на вид области. При этом мы не будем затрагивать вопрос о корректности дискретной аппроксимации нерегулярной области, поскольку ранее мы уже оговаривали, что эта тема выходит за рамки данной книги.

Ясно, что любую заданную область можно покрыть сеткой с размером ячеек h, такой, что все узлы этой сетки являются регулярными, за исключением узлов, соседних к граничным. В этом случае мы должны заменить конечно-разностную аппроксимацию для узлов, соседних к граничным, более сложными выражениями. Если, например, $u(x - ah, y)$ является граничной точкой, то вторая частная производная в точке (x, y) принимает вид

$$\frac{\partial^2 u}{\partial x^2}(x, y) \simeq \frac{2}{h^2}\left[\frac{u(x+h, y)}{(1+a)} - \frac{u(x, y)}{a} + \frac{u(x-ah, y)}{a(1+a)}\right], \tag{1}$$

где

$$0 < a \leqslant 1. \tag{2}$$

Для нашего метода это приводит к небольшим изменениям в определениях Q и r_R и, возможно, к некоторым изменениям уравнений при $R = 0$ и $R = N$.

Если область неодносвязна, то в этом случае потребуется отдельно рассматривать области, лежащие выше и ниже дыры. Если, например, заданная область имеет вид, изображенный на рис. 6, то при $R_1 < R < R_2$ мы получаем два множества A_R

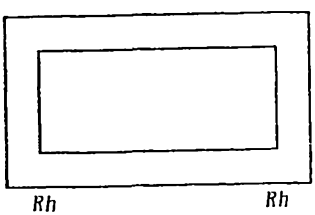


Рис. 6

[1]) См. комментарий в конце главы.— *Прим. перев.*

и два b_R: одно для части области, расположенной выше дыры, и другое — для расположенной ниже. При $R_1 < R < R_2$ мы выполняем две независимые последовательности вычислений на каждом шаге процедуры. Результаты, полученные в разд. 3 и разд. 5, можно обобщить и на этот случай и получить ответ на вопрос, что следует делать при $R = R_1$ и $R = R_2$.

9. Примеры

Уравнение Лапласа было решено для областей, изображенных на рис. 7—9. Граничные условия указаны на рисунках, а в таблице I приведены результаты решения для левого верхнего квадранта области, изображенной на рис. 8.

Рис. 7

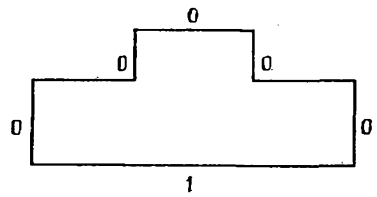


Рис. 8

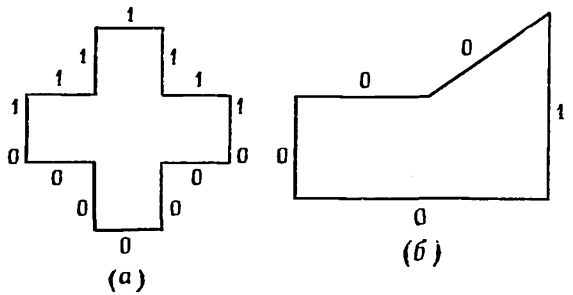
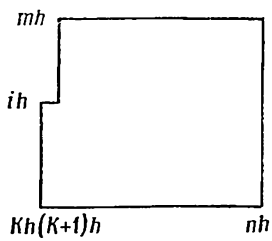


Рис. 9



Следует отметить одно важное обстоятельство. Все три задачи были решены с помощью *одной вычислительной программы*. Единственное необходимое изменение заключалось в простой подпрограмме, описывающей структуру границы области.

Таблица I

				1,0	1,0	1,0	1,0	1,0
				1,0	0,9891	0,9803	0,9746	0,9727
				1,0	0,9662	0,9573	0,9455	0,9416
				1,0	0,9584	0,9272	0,9086	0,9025
				1,0	0,9303	0,8844	0,8592	0,8513
1,0	1,0	1,0	1,0	1,0	0,8786	0,8206	0,7927	0,7842
1,0	0,9104	0,8631	0,8379	0,8143	0,7635	0,7269	0,7067	0,7002
1,0	0,7786	0,7040	0,6744	0,6557	0,6340	0,6169	0,6068	0,6034
0,5	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000	0,5000

10. Линейные уравнения общего вида

Пусть требуется решить уравнение

$$u_{xx} + u_{yy} - g(x, y)\, u + \varphi(x, y) = 0, \tag{1}$$

определенное на прямоугольнике

$$0 \leqslant x \leqslant a, \quad 0 \leqslant y \leqslant b, \tag{2}$$

на всех сторонах которого заданы значения функции $u(x, y)$. Воспользуемся тем же самым способом дискретизации, который использовали до сих пор. Тогда Q и r_R не изменяются, и мы можем определить g_{ij} и φ_{ij}, как в гл. 5:

$$g_{ij} = g(ih, jh), \quad \varphi_{ij} = \varphi(ih, jh). \tag{3}$$

Конечно-разностная аппроксимация уравнения (1) принимает вид

$$u_{i+1,\,j} + u_{i,\,j+1} - 4u_{ij} + u_{i-1,\,j} + u_{i,\,j-1} - h^2 g_{ij} u_{ij} + h^2 \varphi_{ij} = 0. \tag{4}$$

Определив матрицы G_R и векторы φ_R как

$$G_R = h^2 \operatorname{diag}(g_{R1}, g_{R2}, \ldots, g_{R,\,M-1}), \quad \varphi_R = h^2 [\varphi_{Rj}], \tag{5}$$

запишем (4) в векторно-матричном виде

$$u_{R+1} - 2u_R + u_{R-1} - Qu_R + r_R - G_R u_R + \varphi_R = 0, \tag{6}$$

при этом u_0 и u_N определяются из граничных условий.

Будем искать решение (6) в виде

$$u_{R+1} = A_R u_R + b_R. \tag{7}$$

Тогда

$$\begin{aligned} A_{R-1} &= [2I + Q + G_R - A_R]^{-1}, \\ b_{R-1} &= A_{R-1}(b_R + r_R + \varphi_R) \end{aligned} \tag{8}$$

с начальными условиями

$$A_{N-1} = 0, \quad b_{N-1} = u_N \tag{9}$$

Достаточное условие невырожденности и численной устойчивости имеет тот же вид, что и в гл. 3:

$$G_R \geqslant 0. \tag{10}$$

С помощью этого метода легко можно решить уравнение более общего вида:

$$a_1 \frac{\partial^2 u}{\partial x^2} + a_2 \frac{\partial^2 u}{\partial y^2} + a_3 \frac{\partial^2 u}{\partial x\, \partial y} + a_4 \frac{\partial u}{\partial x} + a_5 \frac{\partial u}{\partial y} + a_6 u = \varphi(x, y), \tag{11}$$

где коэффициенты a_i являются функциями от x и y.

11. Другие граничные условия

Покажем, как можно подойти к решению задач, отличных от задачи Дирихле; лучше всего это сделать на примере. Пусть требуется решить уравнение Лапласа

$$u_{xx} + u_{yy} = 0 \tag{1}$$

на прямоугольнике

$$0 \leqslant x \leqslant a, \quad 0 \leqslant y \leqslant b \tag{2}$$

с граничными условиями

$$u(x, 0) = f_1(x), \quad u(x, b) = f_2(x) \tag{3}$$

и

$$\frac{\partial u}{\partial x}(0, y) = \frac{\partial u}{\partial x}(a, y) = 0. \tag{4}$$

Конечно-разностная аппроксимация уравнения (1) имеет тот же вид, что и прежде,

$$u_{R+1} - 2u_R - u_{R-1} - Qu_R + r_R = 0. \tag{5}$$

Заметим, однако, что величины u_0 и u_N теперь неизвестны.

Рассмотрим также вектор u_{N+1}, хотя его компоненты и лежат вне заданной области. Тогда граничное условие

$$\frac{\partial u}{\partial x}(a, y) = 0 \tag{6}$$

с точностью до $O(h^2)$ можно аппроксимировать в силу равенства $Nh = a$ уравнением

$$u_{N+1} - u_{N-1} = 0. \tag{7}$$

Подставляя (7) в (5) при $R = N$, получаем

$$2u_{N-1} - 2u_N - Qu_N + r_N = 0. \tag{8}$$

Однако поскольку мы ищем решение уравнения (5) в виде

$$u_{R+1} = A_R u_R + b_R, \tag{9}$$

то

$$u_N = A_{N-1} u_{N-1} + b_{N-1}. \tag{10}$$

Теперь можно разрешить уравнение (8) относительно u_N:

$$u_N = [2I + Q]^{-1} (2u_{N-1} + r_N). \tag{11}$$

Сравнивая (11) с (10), получаем:

$$\begin{aligned} A_{N-1} &= 2\,[2I + Q]^{-1}, \\ b_{N-1} &= [2I + Q]^{-1} r_N. \end{aligned} \tag{12}$$

Значения A_0 и b_0 определяем, последовательно решая уравнения

$$\begin{aligned} A_{R-1} &= [2I + Q - A_R]^{-1}, \\ b_{R-1} &= A_{R-1} (b_R + r_R). \end{aligned} \tag{13}$$

Наконец, мы должны использовать последнее граничное условие

$$\partial u\,(0,\ y)/\partial x = 0, \tag{14}$$

чтобы найти величины u_0 и u_1, которые необходимы для решения уравнения (9). Предположив существование u_{-1}, построим конечно-разностную аппроксимацию (14):

$$u_{-1} - u_1 = 0. \tag{15}$$

Подставляя (15) в (5) при $R = 0$, получаем

$$2u_1 - 2u_0 - Qu_0 + r_0 = 0 \tag{16}$$

и в силу (9)

$$u_1 = A_0 u_0 + b_0. \tag{17}$$

Эти два уравнения можно решить относительно u_0:

$$u_0 = [2I + Q - 2A_0]^{-1} (2b_0 + r_0), \tag{18}$$

и воспользоваться этим начальным значением для определения всех u_R из уравнения (9).

Если граничные условия типа (4) заданы на верхней и нижней границах, то нетрудно убедиться, что, слегка изменив определения Q и r_R, можно решить и эту задачу. Вообще говоря, этим методом легко решать задачи с граничными условиями вида

$$au\,(x, y) + b\,[\partial u\,(x, y)/\partial x] = c. \tag{19}$$

Упражнение

1. Доказать невырожденность и численную устойчивость для рассмотренного случая.

12. Трехмерные уравнения

Наконец, обсудим некоторые трудности, возникающие при решении трехмерных уравнений эллиптического типа. Для упрощения обозначений рассмотрим уравнение Лапласа

$$u_{xx} + u_{yy} + u_{zz} = 0 \tag{1}$$

на кубе

$$0 \leqslant x \leqslant a, \quad 0 \leqslant y \leqslant a, \quad 0 \leqslant z \leqslant a, \tag{2}$$

где предполагается, что $u(x, y, z)$ задана на всех гранях куба. Положим

$$a = Nh \tag{3}$$

и

$$u_{ijk} = u(ih, jh, kh), \tag{4}$$

тогда конечно-разностная аппроксимация уравнения (1) принимает вид

$$u_{i+1, jk} + u_{i, j+1, k} + u_{ij, k+1} - 6u_{ijk} + u_{i-1, jk} + u_{i, j-1, k} + u_{ij, k-1} = 0. \tag{5}$$

Пусть матрица U_R определяется как

$$U_R = (u_{Rjk}) \tag{6}$$

и описывает расположение узлов сетки при любом фиксированном R. Матрица Q строится, как прежде, а матрица S_R — как

$$S_R = (s_{Rjk}), \quad \text{где} \quad s_{Rjk} = \begin{cases} u_{Rj0}, & k = 1, \\ u_{RjN}, & k = N - 1, \\ u_{R0, k}, & j = 1, \\ u_{RN, k}, & j = N - 1, \\ 0 & \text{в противном случае} \end{cases} \tag{7}$$

и, таким образом, определяются из граничных условий. В этих обозначениях уравнение (5) принимает вид двухточечной граничной задачи

$$U_{R+1} - 2U_R + U_{R-1} - QU_R - U_R Q + S_R = 0, \tag{8}$$

где U_0 и U_N задаются граничными условиями. Это чрезвычайно интересное уравнение, и, по-видимому, оно не может быть решено существующими методами.

Можно было бы предложить следующую процедуру. Пусть u_R и s_R — векторы размерности N^2, построенные как прямые суммы столбцов матриц U_R и S_R соответственно. Теперь уравнение (8) можно записать в виде

$$u_{R+1} - 2u_R + u_{R-1} - [I \otimes Q + Q \otimes I]\, u_R + s_R = 0, \tag{9}$$

где знак $\otimes$ обозначает кронекерово произведение. В гл. 8 мы обсудим некоторые свойства кронекерова произведения более подробно. Решение уравнения (9) можно искать в виде

$$u_{R+1} = A_R u_R + b_R, \tag{10}$$

и можно показать, что этот метод всегда приводит к решению и является численно устойчивым. Подробности оставляем как упражнение для интересующегося читателя.

Трудность здесь состоит в том, что мы должны оперировать с матрицами размерности N^2. При разумных значениях N, скажем $N = 10$, эта задача вполне доступна для современных вычислительных машин, однако для $N = 15$ задача пока слишком велика. Заметим, что размерность N доступных задач растет с каждым годом.

Упражнения

1. Сформулировать задачу Коши для определения A_R и b_R в (10). Доказать невырожденность и устойчивость.

2. Решить трехмерную задачу методом динамического программирования.

13. Бигармоническое уравнение

В качестве примера того, как можно использовать наши методы для решения эллиптических уравнений четвертого порядка, рассмотрим следующую задачу. Статическая деформация однослойной квадратной упругой пластины, закрепленной по краям, под действием поперечной нагрузки описывается бигармоническим уравнением

$$u_{xxxx} + 2u_{xxyy} + u_{yyyy} = p, \tag{1}$$

где через p обозначена нагрузка. Граничные условия имеют вид

$$u = 0, \quad u_{xx} = 0 \tag{2}$$

на краях $x = 0$ и $x = 1$ и

$$u = 0, \quad u_{yy} = 0 \tag{3}$$

на краях $y = 0$ и $y = 1$. Как и в гл. 5, определим новую переменную v:

$$v = u_{xx} + u_{yy}. \tag{4}$$

Тогда уравнение (1) принимает вид

$$v_{xx} + v_{yy} = p, \tag{5}$$

и мы получаем систему из двух уравнений второго порядка.

Уравнения (4) и (5) можно дискретизировать обычным образом, т. е. мы можем воспользоваться стандартной конечно-разностной аппроксимацией вторых производных и, как мы это делали прежде, представить таким образом эту задачу в векторно-матричной форме. Итак, (4) и (5) принимают вид системы двух конечно-разностных уравнений

$$\begin{aligned} v_{i+1} &= [2I + Q]\, v_i - v_{i-1} - p_i, \\ u_{i+1} &= [2I + Q]\, u_i - u_{i-1} + v_i, \end{aligned} \tag{6}$$

где матрица Q определена, как и ранее, а векторы p_i определяются из p. Соответствующие граничные условия для уравнений (6) выводятся с помощью дискретизации (2) и (3).

Определим векторы w_i как

$$w_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix}, \tag{7}$$

так что (6) принимает вид

$$w_{i+1} = \begin{bmatrix} 2I+Q & I \\ 0 & 2I+Q \end{bmatrix} w_i - w_{i-1} - \begin{bmatrix} 0 \\ p_i \end{bmatrix}. \tag{8}$$

Используя изложенные выше процедуры, мы легко можем решить эту задачу, например, с помощью инвариантного погружения или преобразования Риккати.

Упражнения

1. Получить для (8) необходимые граничные условия.
2. Какое уравнение Риккати соответствует уравнению (8). Доказать невырожденность и устойчивость его решения.
3. С помощью непрерывных методов свести (4) и (5) к уравнениям из разд. 13 гл. 6.

14. Инвариантное погружение и разностные уравнения

Математическое описание инвариантного погружения было несколько затуманено тем, что процедуры инвариантного погружения почти всегда основывались на физических рассуждениях. Цель этого раздела состоит в проведении строгого анализа связи инвариантного погружения с линейностью.

Рассмотрим векторно-матричную систему разностных уравнений

$$\begin{bmatrix} u(i+1) \\ v(i+1) \end{bmatrix} = \begin{bmatrix} A(i) & B(i) \\ C(i) & D(i) \end{bmatrix} \begin{bmatrix} u(i) \\ v(i) \end{bmatrix} + \begin{bmatrix} e(i) \\ f(i) \end{bmatrix} \tag{1}$$

с граничными условиями

$$u_M = a \quad v_N = b, \quad N > M. \tag{2}$$

Пусть $u(i)$ есть k-мерный, а $v(i)$ — l-мерный векторы. Тогда $A(i)$, $B(i)$, $C(i)$, $D(i)$, $e(i)$ и $f(i)$ являются $k \times k$-, $l \times k$-, $k \times l$-, $l \times l$-, $k \times l$- и $l \times l$-мерными матрицами соответственно. Запишем $u(i)$ и $v(i)$ в виде

$$u(i) = u(i, M, N, a, b), \quad v(i) = v(i, M, N, a, b) \tag{3}$$

с целью подчеркнуть зависимость $u(i)$ и $v(i)$ от длины отрезка и граничных условий.

В силу линейности можно записать

$$\begin{aligned} u(i, M, N, A, B) &= W(i, M, N)\, a + R(i, M, N)\, b + r(i, M, N), \\ v(i, M, N, a, b) &= S(i, M, N)\, a + T(i, M, N)\, b + s(i, M, N), \end{aligned} \tag{4}$$

или

$$\begin{bmatrix} u(i) \\ v(i) \end{bmatrix} = \begin{bmatrix} W(i, M, N) & R(i, M, N) \\ S(i, M, N) & T(i, M, N) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} r(i, M, N) \\ s(i, m, n) \end{bmatrix}. \tag{5}$$

Функции W, R, T и S зависят только от положения и интервала. Покажем, что можно построить задачи Коши, решениями которых являются эти функции. Применяя граничные условия к (5), получаем

$$\begin{aligned} &W(M, M, N) = I, \quad R(M, M, M) = 0, \quad r(M, M, N) = 0, \\ &S(N, M, N) = 0, \quad T(N, M, N) = I, \quad s(N, M, N) = 0. \end{aligned} \tag{6}$$

Рассмотрим точку

$$\bar{a} = u(M + 1, M, N, a, b). \tag{7}$$

В силу принципа причинности

$$u(i, M, N, a, b) = u(i, M + 1, N, \bar{a}, b), \tag{8}$$

$$v(i, M, N, a, b) = v(i, M + 1, N, \bar{a}, b). \tag{9}$$

Иными словами, значения $u(i)$ и $v(i)$ не зависят от того, начинаем ли мы процесс решения с начальным условием a в момент M или с условием $u(M + 1)$ в момент $M + 1$. Тогда (5) принимает вид

$$\begin{bmatrix} W(i, M, N) & R(i, M, N) \\ S(i, M, N) & T(i, M, N) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} r(i, M, N) \\ s(i, M, N) \end{bmatrix} =$$

$$= \begin{bmatrix} W(i, M+1, N) & R(i, M+1, N) \\ S(i, M+1, N) & T(i, M+1, N) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} r(i, M+1, N) \\ s(i, M+1, N) \end{bmatrix}, \tag{10}$$

$$\cdot \begin{bmatrix} \bar{a} \\ b \end{bmatrix} = \begin{bmatrix} W(M+1,\ M,\ N) & R(M+1,\ M,\ N) \\ 0 & I \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \\ + \begin{bmatrix} r(M+1,\ M,\ N) \\ 0 \end{bmatrix}. \tag{11}$$

Подставляя это выражение в (10) и полагая $i=M+1$, с помощью (6) получаем

$$\begin{bmatrix} W(M+1,\ M,\ N) & R(M+1,\ M,\ N) \\ S(M+1,\ M,\ N) & T(M+1,\ M,\ N) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \\ + \begin{bmatrix} r(M+1,\ M,\ N) \\ s(M+1,\ M,\ N) \end{bmatrix} = \\ = \begin{bmatrix} I & 0 \\ S(M+1,\ M+1,\ N) & T(M+1,\ M+1,\ N) \end{bmatrix} \times \\ \times \left\{ \begin{bmatrix} W(M+1,\ M,\ N) & R(M+1,\ M,\ N) \\ 0 & I \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \right. \\ \left. + \begin{bmatrix} r(M+1,\ M,\ N) \\ 0 \end{bmatrix} \right\} + \begin{bmatrix} 0 \\ s(M+1,\ M+1,\ N) \end{bmatrix}. \tag{12}$$

Используя (5) в первоначальных уравнениях и полагая $i=M$, видим, что

$$\begin{bmatrix} W(M+1,\ M,\ N) & R(M+1,\ M,\ N) \\ S(M+1,\ M,\ N) & T(M+1,\ M,\ N) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \\ + \begin{bmatrix} r(M+1,\ M,\ N) \\ s(M+1,\ M,\ N) \end{bmatrix} = \\ = \begin{bmatrix} A(M) & B(M) \\ C(M) & D(M) \end{bmatrix} \left\{ \begin{bmatrix} I & 0 \\ S(M,\ M,\ N) & T(M,\ M,\ N) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \right. \\ \left. + \begin{bmatrix} 0 \\ s(M,\ M,\ N) \end{bmatrix} \right\} + \begin{bmatrix} e(M) \\ f(M) \end{bmatrix}, \tag{13}$$

и, сравнивая с (12), находим

$$\begin{bmatrix} I & 0 \\ \bar{S}(M+1) & \bar{T}(M+1) \end{bmatrix} \begin{bmatrix} W(M+1,\ M,\ N) & R(M+1,\ M,\ N) \\ 0 & 0 \end{bmatrix} \times \\ \times \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} r(M+1,\ M,\ N) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{s}(M+1) \end{bmatrix} = \\ = \begin{bmatrix} A(M) & B(M) \\ C(M) & D(M) \end{bmatrix} \begin{bmatrix} I & 0 \\ \bar{R}(M) & \bar{T}(M) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} 0 \\ s(M) \end{bmatrix} + \begin{bmatrix} e(M) \\ f(M) \end{bmatrix}, \tag{14}$$

где введены обозначения

$$\begin{aligned}\overline{S}(M) &= S(M,\ M,\ N),\\ \overline{T}(M) &= T(M,\ M,\ N),\\ \overline{s}(M) &= s(M,\ M,\ N).\end{aligned} \tag{15}$$

С помощью равенств (4) первое уравнение из (1) можно переписать в виде

$$\begin{aligned}u(M+1,\ M,\ N,\ a,\ b) &= \\ &= A(M)\,u(M,\ M,\ N,\ a,\ b) + \\ &+ B(M)\,v(M,\ M,\ N,\ a,\ b) + e(M) = \\ &= A(M)\,a + B(M)\,[\overline{S}(M)\,a + \\ &+ \overline{T}(M)\,b + \overline{s}(M)]\,e(M),\end{aligned} \tag{16}$$

Однако, также в силу (4),

$$\begin{aligned}u(M+1,\ M,\ N,\ a,\ b) &= W(M+1,\ M,\ N)\,a + \\ &+ R(M+1,\ M,\ N)\,b + r(M+1,\ M,\ N)\end{aligned} \tag{17}$$

и, таким образом, приравнивая коэффициенты в (16) и (17), находим

$$\begin{aligned}W(M+1,\ M,\ N) &= A(M) + B(M)\,\overline{R}(M),\\ R(M+1,\ M,\ N) &= B(M)\,\overline{T}(M),\\ r(M+1,\ M,\ N) &= B(M)\,\overline{s}(M) + e(M).\end{aligned} \tag{18}$$

Теперь можно подставить эти уравнения в (14) и свести систему к следующей системе уравнений, включающей лишь функции $\overline{T}(M)$, $\overline{S}(M)$ и $\overline{s}(M)$:

$$\begin{aligned}&\begin{bmatrix}A(M) & B(M)\\ C(M) & D(M)\end{bmatrix}\left\{\begin{bmatrix}I & 0\\ \overline{S}(M) & \overline{T}(M)\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix}+\begin{bmatrix}0\\ \overline{s}(M)\end{bmatrix}\right\}+\begin{bmatrix}e(M)\\ f(M)\end{bmatrix}= \\ &\quad=\begin{bmatrix}I & 0\\ \overline{S}(M+1) & \overline{T}(M+1)\end{bmatrix}\times \\ &\quad\times\left\{\begin{bmatrix}A(M)+B(M)\overline{S}(M) & B(M)\overline{T}(M)\\ 0 & I\end{bmatrix}\times\right. \\ &\quad\left.\times\begin{bmatrix}a\\ b\end{bmatrix}+\begin{bmatrix}B(M)\overline{s}(M)+e(M)\\ 0\end{bmatrix}\right\}+\begin{bmatrix}0\\ \overline{s}(M+1)\end{bmatrix}.\end{aligned} \tag{19}$$

Наконец, приравнивая коэффициенты при a и b в этом выражении, находим уравнения для $\overline{S}(M)$, $\overline{T}(M)$ и $\overline{s}(M)$:

$$\overline{S}(M) = (D(M) - \overline{S}(M+1)\,B(M))^{-1}\,(\overline{S}(M+1)\,A(M) - C(M)),$$

$$\overline{T}(M) = (D(M) - \overline{S}(M+1)B(M))^{-1}\overline{T}(M+1),$$
$$\bar{s}(M) = (D(M) - \overline{S}(M+1)B(M))^{-1} \times$$
$$\times (\bar{s}(M+1) + \overline{S}(M+1)e(M) - f(M)). \tag{20}$$

Начальные условия определяются из (6) и имеют вид

$$\overline{S}(N) = 0, \quad \overline{T}(N) = I, \quad \bar{s}(N) = 0. \tag{21}$$

Поступая аналогичным образом, мы можем получить уравнения для функций $R(i, M, N)$, $W(i, M, N)$ и $s(i, M, N)$. Рассмотрим точку

$$\bar{b} = v(N+1, M, N, a, b). \tag{22}$$

Определим $\overline{R}(N)$, $\overline{W}(N)$ и $\bar{s}(N)$ как

$$\overline{R}(N) = R(N, M, N), \quad \overline{W}(N) = W(N, M, N),$$
$$\bar{s}(N) = S(N, M, N), \tag{23}$$

тогда

$$\overline{R}(N+1) = (A(N)\overline{R}(N) + B(N))(C(N)\overline{R}(N) + D(N))^{-1},$$
$$\overline{W}(N+1) = A(N)\overline{W}(N) - \overline{R}(N+1), \tag{24}$$
$$\bar{s}(N+1) = A(N)\bar{s}(N) + e(N) - \overline{R}(N+1)(C(N)\bar{s}(N) + f(N))$$

с начальными условиями

$$R(M) = 0, \; W(M) = I, \; s(M) = 0. \tag{25}$$

Хотя мы теперь в состоянии определить все функции, фигурирующие в (4), для решения исходной задачи это вовсе не обязательно. Все, что нам надо — это функции, определяемые либо соотношениями (20), либо соотношениями (24), поскольку, в силу принципа причинности, мы можем записать

$$\begin{bmatrix} u(i) \\ v(i) \end{bmatrix} = \begin{bmatrix} I & 0 \\ \overline{S}(i) & \overline{T}(i) \end{bmatrix} \begin{bmatrix} u(i) \\ b \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{s}(i) \end{bmatrix} \tag{26}$$

и подставить это выражение в (1). Тогда мы получим

$$\begin{bmatrix} u(i+1) \\ v(i+1) \end{bmatrix} =$$
$$= \begin{bmatrix} A(i) & B(i) \\ C(i) & D(i) \end{bmatrix} \left\{ \begin{bmatrix} I & 0 \\ \overline{S}(i) & \overline{T}(i) \end{bmatrix} \begin{bmatrix} u(i) \\ b \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{s}(i) \end{bmatrix} \right\} + \begin{bmatrix} e(i) \\ f(i) \end{bmatrix}, \tag{27}$$

что представляет собой задачу Коши с начальным значением

$$u(M) = a, \tag{28}$$

поскольку уравнения (20) уже решены. Аналогичным образом мы можем получить задачу Коши вида

$$\begin{bmatrix} u(i+1) \\ v(i+1) \end{bmatrix} =$$

$$= \begin{bmatrix} A(i) & B(i) \\ C(i) & D(i) \end{bmatrix} \left\{ \begin{bmatrix} \overline{W}(i) & \overline{R}(i) \\ 0 & I \end{bmatrix} \begin{bmatrix} a \\ v(i) \end{bmatrix} + \begin{bmatrix} s(i) \\ 0 \end{bmatrix} \right\} + \begin{bmatrix} e(i) \\ f(i) \end{bmatrix}. \qquad (29)$$

Упражнения

1. Что получится, если в (3) зафиксировать i и варьировать N и M?

2. Как изменятся результаты этого раздела для случая пары обыкновенных дифференциальных уравнений?

15. Другой подход

Попробуем теперь использовать способ составных блоков, который позволит нам решать уравнения в частных производных с помощью только лишь скалярных вычислений. Как и прежде, рассмотрим уравнение Лапласа с заданными значениями функции на границах области.

Пусть требуется решить уравнение Лапласа, определенное на области, изображенной на рис. 10. Пусть способ дискретизации таков, как на рис. 11. Обозначим левые граничные значения через $\{v_k\}$, т. е.

$$\begin{aligned} v_j &= u_{jk}, \qquad j = 1, \ldots, l-1, \\ v_j &= u_{j,\,k+1}, \quad j = l+1, \ldots, m-1. \end{aligned} \qquad (1)$$

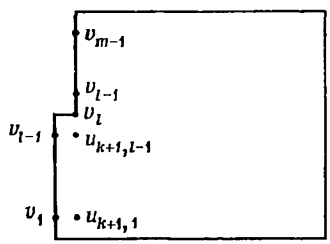


Рис. 10

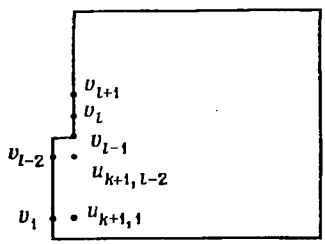


Рис. 11

Обозначим также через $f_{lk}(v_1, \ldots, v_{m-1})$ значение функционала в узле дискретной сетки из рис 11, где, как обычно, левые гранич-

ные значения рассматриваются как параметры. В итоге приходим к той же самой процедуре, что и в гл. 5:

$$f_{lk}(v_1, \ldots, v_{m-1}) = \min_{\{u_{ij}\}} \Big[\sum_{j=1}^{l} (u_{k+1,\,j} - u_{k+1,\,j-1})^2 + \sum_{j=1}^{l-1} (u_{k+1,\,j} - u_{k,\,j})^2 + \\ + \sum_{l=k+2}^{n} \Big[\sum_{j=1}^{m} (u_{ij} - u_{i,\,j-1})^2 + \sum_{j=1}^{m-1} (u_{ij} - u_{i-1,\,j})^2\Big]\Big], \qquad l = 2, 3, \ldots, m, \tag{2}$$

где значения $\{v_i\}$ определены равенствами (1), а минимизация проводится по внутренним точкам. Выражение (2) можно переписать в виде

$$f_{lk}(v_1, \ldots, v_{m-1}) = \min_{u_{k+1,\,l-1}} \big[(v_l - u_{k+1,\,l-1})^2 + (v_{l-1} - u_{k+1,\,l-1}) + \\ + \min_{\{u_{ij}\}} \Big[\sum_{j=1}^{l-1} (u_{k+1,\,j} - u_{k+1,\,j-1})^2 + \sum_{j=1}^{l-2} (u_{k+1,\,j} - u_{kj})^2 + \\ + \sum_{i=k+2}^{n} \Big[\sum_{j=1}^{m} (u_{i,\,j} - u_{ij-1})^2 + \sum_{j=1}^{m-1} (u_{ij} - u_{i-1,\,j})^2\Big]\Big]\Big], \tag{3}$$

или, полагая $\omega = u_{k+1,\,l+1}$,

$$f_{lk}(v_1, \ldots, v_{m-1}) = \min_{\omega} \big[(v_l - \omega)^2 + (v_{l-1} - \omega)^2 + \\ + f_{l-1,\,k}(v_1, \ldots, v_{l-2}, \omega, v_l, \ldots, v_{m-1})\big]. \tag{4}$$

Это важное соотношение является просто другой формулировкой принципа оптимальности. Итак, нам удалось связать задачу на области, изображенной на рис. 11, с задачей на области из рис. 13. Этого результата следовало ожидать, поскольку при переходе от области на рис. 12 к области на рис. 13 мы добавляем два дополнительных члена в дискретный функционал и при минимизации этого функционала вводим в рассмотрение еще одну дополнительную внутреннюю точку.

Численное решение уравнения (4), к чему мы еще вкратце вернемся, позволит нам перейти от области на рис. 13 к области на рис. 14 с помощью последовательности *одномерных* задач минимизации. Однако хотелось бы найти и значения f_{1k} с помощью скалярных операций. С этой целью сначала определим $f_{0,\,k}(v_1, \ldots, v_{m-1})$ как

$$f_{0,\,k}(v_1, \ldots, v_{m-1}) = f_{m,\,k+1}(v_1, \ldots, v_{m-1}), \tag{5}$$

что очевидно из рис. 14.

Рассмотрим теперь области на рис. 15 и 16. В обоих случаях минимизация проводится по одним и тем же внутренним точкам.

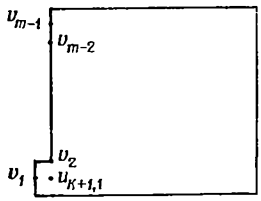


Рис. 12

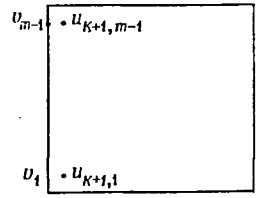


Рис. 13

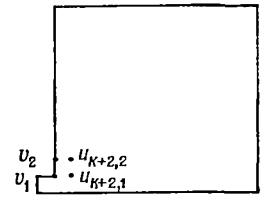


Рис. 14

Обозначив через $f_{1k}(v_1, \ldots, v_{m-1})$ минимум дискретной функции по области на рис. 15 и обозначив, с учетом (5), минимум по обла-

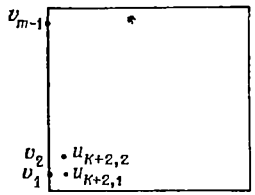


Рис. 15

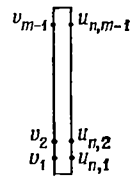


Рис. 16

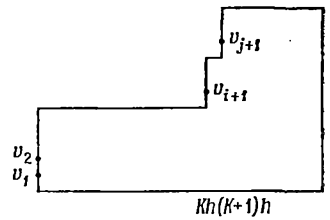


Рис. 17

сти на рис. 16 через $f_{0k}(v_1, \ldots, v_{m-1})$, получим

$$f_{1,k}(v_1, \ldots, v_{m-1}) = (v_1 - v_{k+1,0})^2 + f_{0,k}(v_1, \ldots, v_{m-1}). \quad (6)$$

Для определения $f_{m,n-1}(v_1, \ldots, v_{m-1})$ рассмотрим область на рис. 17. Поскольку здесь нет внутренних точек, то

$$f_{m,n-1}(v_1, \ldots, v_{m-1}) = \sum_{j=1}^{m} (u_{nj} - u_{n,j-1})^2 + \sum_{j=1}^{m-1} (u_{nj} - u_{n-1,j})^2. \quad (7)$$

В это выражение входят только $\{v_i\}$ и граничные значения, поэтому для решения уравнения Лапласа необходимо найти лишь величины $f_{ik}(v_1, \ldots, v_m)$ для соответствующих значений i и k. Это можно сделать, решив последовательность одномерных задач:

$$f_{m,n-1} = f_{0,n-2},\ f_{1,n-2},\ \ldots,\ f_{m-1,n-2},$$

$$f_{m,m-2} = f_{0,n-3},\ f_{1,n-3},\ \ldots,\ f_{i-1,k},\ f_{ik}.$$

16. Векторно-матричные уравнения

Теперь можно обратиться к вычислительной стороне нашего нового метода. Обозначим $(m-1)$-мерный вектор буквой v:

$$v = [v_i], \quad i = 1, \ldots, m-1. \tag{1}$$

Снова по индукции можно показать, что $f_{lk}(v_1, \ldots, v_{m-1}) = f_{lk}(v)$ квадратична по v:

$$f_{lk}(v) = (A^{(l,k)}v, v) - 2(b^{(l,k)}, v) + c^{(l,k)}. \tag{2}$$

Действительно, легко убедиться, что в обозначениях предыдущего раздела

$$f_{m,n-1}(v) = (v, v) - 2(u_n, v) + ([I+Q]u_n, u_n) - (2r_n, u_n) + s_n, \tag{3}$$

поэтому

$$A^{(m,n-1)} = I, \quad b^{(m,n-1)} = u_n. \tag{4}$$

Поскольку, как обычно, мы будем интересоваться лишь значениями $\{u_{ij}\}$, то нам не нужно знать значений $c^{(l,k)}$ и, кроме того, мы можем опустить все скалярные члены. Из (15.5) получаем

$$A^{(0,k)} = A^{(m,k+1)}, \quad b^{(0,k)} = b^{(m,k+1)}. \tag{5}$$

Без потери общности можно считать, что $A^{(l,k)}$ симметрична. Введем обозначения

$$A^{(l,k)} = (a_{ij}^{(l,k)}), \quad b^{(l,k)} = (b_i^{(l,k)}). \tag{6}$$

Тогда в силу (13.6) получаем:

$$a_{ij}^{(i,k)} = \begin{cases} a_{11}^{(0,k)} + 1, & \text{если } i = j = 1, \\ a_{ij}^{(0,k)} & \text{в противном случае,} \end{cases}$$

$$b_i^{(i,k)} = \begin{cases} b_1^{(0,k)} + u_{k+1,0}, & \text{если } i = 1, \\ b_i^{(0,k)} & \text{в противном случае.} \end{cases} \tag{7}$$

С помощью (2) представим (15.4) в виде

$$f_{lk}(v) = \min_{\omega} [(v_l - \omega)^2 + (v_{l-1} - \omega)^2] +$$

$$+ \left(A^{(l-1,k)} \begin{bmatrix} v_1 \\ \vdots \\ \omega \\ v_l \\ \vdots \\ v_m \end{bmatrix}, \begin{bmatrix} v_1 \\ \vdots \\ \omega \\ v_l \\ \vdots \\ v_m \end{bmatrix} \right) - 2 \left(b^{(l-1,k)}, \begin{bmatrix} v_1 \\ \vdots \\ \omega \\ v_l \\ \vdots \\ v_{m-1} \end{bmatrix} \right) + c^{(l-1,k)}. \tag{8}$$

Для определения минимизирующего значения ω это выражение можно продифференцировать:

$$\omega = v_l + \frac{v_{l-1} - b_l^{(l-1,\,k)} - \sum\limits_{\substack{i=1 \\ i \neq l}}^{m-1} a_{li} v_i}{2 + a_{ll}^{(l-1,\,k)}}. \tag{9}$$

Подставим теперь это выражение в (8) для определения $A^{(l,\,k)}$ и $b^{(l,\,k)}$. Для упрощения записи введем обозначения

$$\begin{aligned} \omega - v_l &= (\alpha,\ v) + \beta, \\ \omega - v_{l-1} &= (\alpha',\ v) + \beta, \end{aligned} \tag{10}$$

где в силу (9)

$$\alpha_i = \begin{cases} \dfrac{-(1 + a_{ll}^{(l-1,\,k)})}{2 + a_{ll}^{(l-1,\,k)}}, & i = l, \\ \dfrac{1 - a_{l,\,l-1}}{2 + a_{ll}^{(l-1,\,k)}}, & i = l-1 \\ \dfrac{-a_l}{2 + a_{ll}^{(l-1,\,k)}} & \text{в противном случае,} \end{cases}$$

$$\alpha_i' = \begin{cases} \dfrac{1}{2 + a_{ll}^{(l-1,\,k)}}, & i = l, \\ \dfrac{-(1 + a_{ll}^{(l-1,\,k)} + a_{l,l-1}^{(l-1,\,k)})}{2 + a_{ll}^{(l-1,\,k)}}, & i = l-1 \\ \dfrac{-a_{li}}{2 + a_{ll}^{(l-1,\,k)}} & \text{в противном случае,} \end{cases} \tag{11}$$

$$\beta = \frac{b_l^{(l-1,\,k)}}{2 + a_{ll}^{(l-1,\,k)}}.$$

Тогда

$$f_{lk}(v) = [(\alpha,\ v) + \beta]^2 + [(\alpha',\ v) + \beta]^2 + (A^{(l-i,\,k)}(v + \delta v),\ v + \delta v) - \\ - 2\,(b^{(l-1,\,k)},\ v + \delta v) + c^{(l-1,\,k)}, \tag{12}$$

где в силу (10)

$$\delta v = [\delta v_i],$$

$$\delta_{vi} = \begin{cases} (\alpha,\ v) + \beta, & i = l-1, \\ 0 & \text{в противном случае.} \end{cases} \tag{13}$$

Раскрывая (12) и выписывая все члены, содержащие v, получим

$$(\alpha,\ v)^2 = (Av,\ v), \tag{14}$$

где

$$A = (a_{ij}) = (\alpha_i \alpha_j). \tag{15}$$

Аналогично

$$(\alpha',\ v)^2 = (A'^2 v,\ v) \tag{16}$$

и

$$\sum_{i=1}^{m-1} a_{l-1,\,j} v_i \sum_{j=1}^{m-1} \alpha_j v_j = (A'' v,\ v), \tag{17}$$

где

$$A' = (a'_{ij}) = (\alpha'_i \alpha'_j), \tag{18}$$

$$A'' = (a''_{ij}) = (a_{l-1,\,i}^{(l-1,\,k)} \alpha_j).$$

Итак, пусть

$$\begin{aligned} \bar{A}^{(l,\,k)} &= A^{(l-1,\,k)} + A + A' + A'' \\ &= (a_{ij}^{(l-1,\,k)} + (a_{l-1,\,i}^{(l-1,\,k)} + 2\alpha_i)\,\alpha_j + \alpha'_i \alpha'_j). \end{aligned} \tag{19}$$

Наконец, потребовав, чтобы $A^{(l,\,k)}$ была симметричной, определим ее из $\bar{A}^{(l,\,k)}$, т. е. запишем:

$$A^{(l,\,k)} = (a_{ij}^{(l,\,k)}) = \frac{1}{2}\,(\bar{a}_{ij}^{(l,\,k)} + \bar{a}_{ji}^{(l,\,k)}). \tag{20}$$

Аналогичным образом находим

$$\begin{aligned} b^{(l,\,k)} = [b_i^{(l,\,k)}] = [&-\beta\,(2\alpha_i + \alpha'_i) - a_{l-1,\,j}^{(l-1,\,k)} + \\ &+ b_i^{(l-1,\,k)} + b_{l-1}^{(l-1,\,k)} \alpha_i]. \end{aligned} \tag{21}$$

Таким образом, с помощью этой регулярной процедуры мы можем вычислять и запоминать матрицы $A^{(l,\,k)}$ и векторы $b^{(l,\,k)}$. Далее с помощью (9), где $\omega = u_{k,\,l-1}$, получаем решение задачи.

17. Области общего вида

Ясно, что с помощью методики, изложенной в последних двух разделах, мы можем решать задачи на двумерных областях общего вида путем сведения их к последовательности одномерных задач минимизации. Основная трудность здесь заключается в обозначениях [1]). Например, минимальное значение дискретного функционала на области, изображенной на рис. 18, можно было бы обозначить через $f_{ijk}(v_1, \ldots, v_m)$. Тогда, поступая, как и прежде, мы пришли бы к функциональному уравнению вида

$$\begin{aligned} f_{ijk}(v_1, \ldots, v_{m-1}) = \min_w [&(v_j - w)^2 + (v_{j-1} - w)^2 + \\ &+ f_{i,\,j-1,\,k}(v_1, \ldots, v_{j-2}, w, v_j, \ldots, v_{m-1})] \end{aligned} \tag{1}$$

[1]) При первом чтении в это утверждение трудно поверить.— *Прим. ред.*

для $j > i$ и другим соотношениям, аналогичным уравнениям из последних двух разделов для граничных значений i, j и k.

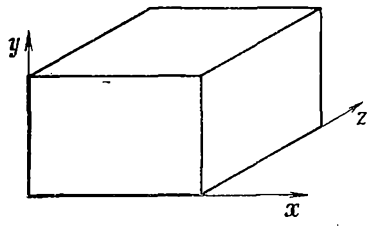


Рис. 18

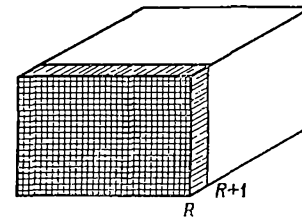


Рис. 19

Однако наиболее интересное и важное приложение этих результатов состоит в решении уравнений в частных производных на областях размерности $\geqslant 3$. Рассмотрим, например, область на рис. 19, где

$$a = Nh, \quad b = Mh, \quad c = Lh. \tag{2}$$

Обозначим через u решение уравнения Лапласа

$$u_{xx} + u_{yy} + u_{zz} = 0 \tag{3}$$

на этой области с граничными условиями, заданными на гранях параллелепипеда. Тогда задачу (3) можно заменить вариационной задачей

$$\min J(u) = \min \int\int\int_R (u_x^2 + u_y^2 + u_z^2)\, dR \tag{4}$$

при тех же граничных условиях, что и в (3). Теперь можно дискретизировать (4) и рассмотреть лишь значения u в узлах сетки. Обозначим эти узлы через $\{u_{ik}\}$, т. е.

$$u_{ijk} = u(ih, jh, kh). \tag{5}$$

Далее применим аппарат динамического программирования к дискретному аналогу функционала (4), который имеет вид

$$\min_{\{u_{ijk}\}} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{L} [(u_{ijk} - u_{i-1,\, jk})^2 + (u_{ijk} - u_{i,\, j-1,\, k})^2 + (u_{ijk} - u_{ij,\, k-1})^2]. \tag{6}$$

Определим теперь $N - 1 > M - 1$ матриц U_l таким же образом, как мы определяли векторы u_R в двумерном случае, т. е.

$$U_l = [u_{ijl}]. \tag{7}$$

Применяя принцип оптимальности, получаем функциональное уравнение в виде

$$f_l(V) = \min_{U_l} [(QU_l,\ U_l) - (2R_l,\ U_R) + S_L + f_{l+1}(U_R)], \qquad (8)$$

где скалярное произведение двух матриц определяется как

$$(A,\ B) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ij} = \mathrm{tr}\,(AB). \qquad (9)$$

Используем теперь еще раз тот факт, что $f_l(V)$ квадратична по V:

$$f_l(V) = (A_l V,\ V) - 2\,(B_l,\ V) + C_l. \qquad (10)$$

Исходя из (10), легко вывести матричные рекуррентные уравнения для матриц A и B. К сожалению, здесь приходится обращать матрицы размерности $(N-1)(M-1)$, поскольку наше погружение определяет, как изменяется решение при удалении целой поверхности, состоящей из узлов сетки; см. рис. 20. Таким образом, даже при сравнительно небольших значениях M и N приходится обращать матрицы, размерности которых слишком велики для большинства вычислительных машин. В последующих главах мы увидим, что это препятствие возникает и в методе инвариантного погружения.

Рассмотрим теперь область, изображенную на рис. 21. Минимальное значение дискретного функционала на этой области обозначим через $f_{lk}(v_1, \ldots, v_m)$, где

$$v_i = u_{l,\,k+1,\,l} \qquad (11)$$

являются значениями функции на границе области. Если мы поступим, как в двумерном случае, то сможем удалить $M-1$ узлов, как это показано на рис. 22, и получим соотношение, связывающее

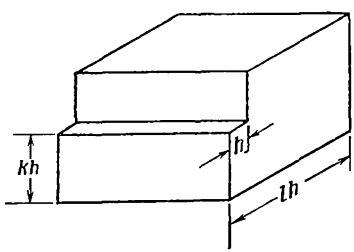


Рис. 20

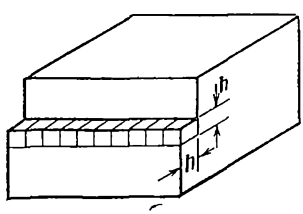


Рис. 21

$f_{l,\,k}(v)$ и $f_{l,\,k+1}(v)$. Поскольку в этом случае за один раз удаляется $M-1$ узлов, то мы должны обращать матрицы размерности $M-1$. Если, наконец, посмотреть на рис. 22, можно заметить, что за один раз удаляется лишь одна ячейка из сетки. Поэтому при

таком подходе мы сможем свести трехмерную задачу к последовательности одномерных задач минимизации. Получение рекуррентных уравнений для всех этих подходов не вызывает принципиальных затруднений, однако является весьма утомительной задачей.

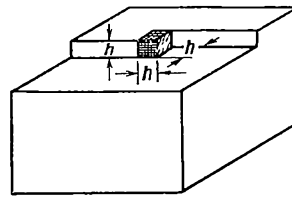


Рис. 22

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Раздел 2. Для решения уравнений Лапласа и Пуассона на нерегулярных областях были предложены различные методы; см., например, статью

Базби, Голуб, Нильсон (Busbee B. L., Golub G. H., Nielson C. W.)
On direct methods for solving Poisson's equations, *SIAM J. Numer. Anal.*, **7** (1970), 627—656.

Раздел 3. См. статью

Энджел (Angel E.)
Discrete invariant imbedding and elliptic boundary value problems over irregular regions, *J. Math. Anal. Appl.*, 23 (1968), 471—484.

Раздел 7. Теорема отделения Штурма сформулирована в книге

Беллман Р.
Введение в теорию матриц, «Наука», М., 1969, стр. 146.

Раздел 8. См. книгу

Михлин С. Г., Смолицкий К. Л.
Приближенные методы решения дифференциальных и интегральных уравнений, изд-во «Наука», М., 1965.

Раздел 9. Дальнейшие численные результаты можно найти в работе

Энджел (Angel E.)
Dynamic programming and partial differential equations, Ph. D. Thesis, Univ. of Southern California, 1968.

Раздел 11. См. список литературы в книге

Варга (Varga R.)
Matrix iterative analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

Раздел 12. См. статью

Энджел (Angel E.)
Invariant imbedding and three dimensional potential problems, Electronic Sciences Laboratory, Univ. of Southern California, USCEE-325, 1969.

Раздел 13. См. отчет

Дистефано, Шуйман (Distefano N., Schujman J.)

Numerical solution of boundary-value problems in structural mechanics by reduction to an initial-value problems, Earthquake Engineering Research Center Rept. EERC 69-4, Univ. of California, Berkeley, 1969.

Раздел 14. Различные соображения о связи уравнений Риккати с линейностью систем можно найти в работах

Денман (Denman E. D.)

Coupled modes in plasmas, elastic media and parameter amplifiers, Amer. Elsevier, New York, 1970.

Редхеффер (Redheffer R.)

Difference equations and functional equations, in «Modern mathematics for the engineer: Transmission line theory», под редакцией Beckenbach E. F., 2nd Ser., McGraw-Hill, New York, 1962.

Рейд (Reid W. T.)

Solution of a Riccati matrix differential equation as a function of initial values, *J. Math. Mech.*, 8 (1959), 221; Riccati differential equations, Academic Press, New York.

Мак-Набб, Шумицки (McNabb A., Schumitzky A.)

Factorization of operators III: Initial value methods for linear two-point boundary-value problems, *J. Math. Anal. Appl.*, 31 (1970), 391—406.

Раздел 15. См. статью

Энджел, Беллман (Angel E., Bellman R.)

Dynamic programming and reduction of dimensionality for the potential equation, *J. Math. Anal. Appl.* (в печати).

Раздел 17. Другой подход к решению задач на нерегулярных областях состоит в том, что нерегулярную область можно рассматривать как погруженную в некоторую регулярную область. Ограничения используются для фиксации граничных условий на границе нерегулярной области, которая теперь находится внутри выбранной регулярной области. Преимущество такого подхода определяется тем, что при этом методы, хорошо работающие на регулярных областях, можно распространить и для задач весьма общего вида. См., например,

Энджел (Angel E.)

Irregular regions and constrained optimization, Electronic Science Laboratory, Univ. of Southern California, USCEE 71—27, 1971.

Базби, Дорр, Джордж, Голуб (Buzbee B. L., Dorr F. W., George J. A., Golub G. H.)

The direct solution of the discrete Poisson equation on irregular regions, Computer Sciences Department, Stanford Univ., STAN-CS-71-195, 1970.

Глава 8

Специальные вычислительные методы

1. Сравнение конечных и итерационных методов

Оба основных подхода, которыми мы до сих пор пользовались,— динамическое программирование и инвариантное погружение — являются конечными (не итерационными) методами. По этой причине нам требовалось последовательно обращать симметрические матрицы произвольной структуры. В результате для задач на квадратной области в этих методах требовалось произвести $O(N^4)$ умножений и делений. Мы уже отмечали, что сравнительно большое число операций компенсируется рядом положительных факторов, таких, как возможность решать аналогичные задачи почти без дополнительных вычислений, простота, с которой эти методы могут быть применены для нерегулярных областей, и свобода выбора критических параметров.

С другой стороны, в итерационных конечно-разностных методах для одной итерации требуется только $O(N^2)$ операций, причем совершенно не требуется вычисления обратных матриц. В этом заключаются преимущества итерационных методов. В настоящей главе мы выведем ряд итерационных схем непосредственно из динамического программирования и инвариантного погружения. Хотя, как мы покажем далее, эти методы эквивалентны стандартным алгоритмам, используемый подход позволит распространить итерационные методы на круг задач с нерегулярными областями.

Мы обсудим также некоторые модификации описанных ранее конечных алгоритмов и применим их для решения уравнений с постоянными коэффициентами. Эти изменения позволят избежать процедуры обращения матриц. В качестве примера снова рассмотрим уравнение Лапласа на квадрате. Этот пример наиболее удобен для демонстрации методики анализа итерационных методов.

Мы будем существенно использовать собственные значения матрицы Q, пользуясь при анализе понятием кронекеровского произведения. Хотя область применения этих методов ограничена классом уравнений с постоянными коэффициентами, их популярность можно объяснить двумя причинами. Во-первых, ясное понимание случая уравнений с постоянными коэффициентами поможет правильно подойти к решению более сложных задач. Во-вторых, эти уравнения сами по себе весьма важны, о чем свидетельствует обширная литература, посвященная уравнениям Лапласа, Пуассона и бигармоническим уравнениям.

2. Собственные значения матрицы Q

Для того чтобы пойти дальше, нам потребуются некоторые предварительные рассуждения. Мы уже видели, что анализ всех методов основывался до сих пор только на положительной определенности матрицы Q[1]). Однако на данном этапе полезно определить собственные значения этой матрицы.

Пусть размерность Q равна N. Представим Q в виде

$$Q = 2I - P, \tag{1}$$

где матрица P определяется как

$$P = \begin{bmatrix} 0 & 1 & & & 0 \\ 1 & 0 & 1 & & \\ & 1 & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & \cdot\ 1 \\ 0 & & & 1 & 0 \end{bmatrix}. \tag{2}$$

Таким образом, если нам удастся определить собственные значения матрицы P, то тем самым мы найдем собственные значения Q [2]). Пусть элементы матрицы F

$$F = (f_{ij}) \tag{3}$$

определяются выражением

$$f_{ij} = \sin\,[ij\pi/(N+1)]. \tag{4}$$

Тогда, если

$$PF = (a_{ij}), \tag{5}$$

то, непосредственно перемножая, получаем:

$$\begin{aligned} a_{ij} &= \sin\,[(i+1)\,j\pi/(N+1)] + \sin\,[(i-1)\,j\pi/(N+1)] = \\ &= 2\sin\,[ij\pi/(N+1)]\cos\,[j\pi/(N+1)]. \end{aligned} \tag{6}$$

Обозначим через D диагональную матрицу

$$D = \begin{bmatrix} \nu_1 & & & \\ & \nu_2 & & \\ & & \cdot & \\ & & & \cdot \\ & & & \nu_N \end{bmatrix}. \tag{7}$$

---

[1]) Матрица Q положительно определена только при четных N; в противном случае одно из собственных значений равно нулю.— *Прим. перев.*

[2]) Поскольку матрицы P и Q, очевидно, коммутируют.— *Прим. перев.*

Тогда, полагая

$$FD = (b_{ij}), \tag{8}$$

находим, что

$$b_{ij} = \sin [ij\pi/(N+1)]\, \nu_j. \tag{9}$$

Итак,

$$PF = FD, \tag{10}$$

если

$$\nu_j = 2 \cos [j\pi/(N+1)]. \tag{11}$$

Следовательно, собственные значения P определяются как [1])

$$\nu_i = 2 \cos [i\pi/(N+1)], \quad i = 1, 2, \ldots, N, \tag{12}$$

и поэтому собственные значения μ_j матрицы Q равны

$$\mu_j = 2\{1 - \cos [j\pi/(N+1)]\}, \quad j = 1, 2, \ldots, N. \tag{13}$$

3. Кронекерово произведение

В предыдущей главе мы кратко коснулись кронекерова произведения. В настоящей главе будем широко использовать это понятие. Пусть A и B суть N-мерная и M-мерная матрицы соответственно. Тогда кронекерово произведение определяется как NM-мерная матрица

$$C = A \otimes B = (a_{ij}B). \tag{1}$$

Легко показать, что определенное таким образом кронекерово произведение обладает рядом удобных алгебраических свойств. Так, например,

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C, \tag{2}$$

$$(A + B) \otimes (C + D) = A \otimes C + A \otimes D + B \otimes C + B \otimes D, \tag{3}$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \tag{4}$$

Упражнение: Доказать (2), (3) и (4).

4. Кронекеровы суммы

Рассмотрим матрицу C, определенную равенством

$$C = I \otimes A + B \otimes I. \tag{1}$$

Если размерность A равна N, а размерность B равна M, то можно использовать в (1) единичные матрицы соответствующих размерностей так, чтобы размерность C была MN. Будем предполагать,

---

[1]) Это следствие того, что F симметрична и ортогональна, поскольку тогда $P = FDF$ в силу (10), откуда и следует (12). Отметим, что доказательство, приведенное в оригинале, некорректно, поскольку построенная авторами матрица неортогональна и, более того, вырождена. Далее изменения в тексте внесены без соответствующих оговорок.—*Прим. перев.*

что A и B — действительные симметрические матрицы, поскольку всюду в дальнейшем нас будет интересовать только этот случай.

Пусть $\{\alpha_i\}$ и $\{\beta_j\}$ — собственные значения матриц A и B соответственно. Представим A и B в виде

$$A = T\bar{A}T', \quad B = S\bar{B}S', \tag{2}$$

где штрих обозначает операцию транспонирования,

$$\bar{A} = \begin{bmatrix} \alpha_1 & & 0 \\ & \ddots & \\ 0 & & \alpha_N \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} \beta_1 & & 0 \\ & \ddots & \\ 0 & & \beta_M \end{bmatrix} \tag{3}$$

и

$$TT' = I, \quad SS' = I. \tag{4}$$

Подставим теперь (2) в (1) и воспользуемся свойствами кронекерова произведения, отмеченными в предыдущем разделе. Тогда

$$\begin{aligned} C &= I \otimes (T\bar{A}T') + (S\bar{B}S') \otimes I = \\ &= (SS') \otimes (T\bar{A}T') + (S\bar{B}S') \otimes (TT') = \\ &= (S \otimes T)(I \otimes \bar{A})(S' \otimes T') + (S \otimes T)(\bar{B} \otimes I)(S' \otimes T') = \\ &= (S \otimes T)(I \otimes \bar{A} + \bar{B} \otimes I)(S \otimes T)'. \end{aligned} \tag{5}$$

Непосредственной проверкой можно убедиться, что

$$(S \otimes T)(S \otimes T)' = I \tag{6}$$

и что матрица $I \otimes \bar{A} + \bar{B} \otimes I$ диагональна. Итак, MN собственных значений матрицы C являются диагональными элементами матрицы $I \otimes A + B \otimes I$ и равны $\alpha_i + \beta_j$. Мы будем широко пользоваться этим результатом в дальнейшем.

Упражнение

Показать, что блочную тридиагональную матрицу

$$D = \begin{bmatrix} A & B & & \\ C & A & \ddots & \\ & \ddots & \ddots & B \\ & & C & A \end{bmatrix}$$

можно представить в виде

$$D = [I \otimes A + L \otimes C + U \otimes B],$$

где L и U определяются как

$$L=\begin{bmatrix} 0 & & & & 0 \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \\ 0 & & & 1 & 0 \end{bmatrix}, \quad U=\begin{bmatrix} 0 & 1 & & & 0 \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & \\ & & & & 1 \\ 0 & & & & 0 \end{bmatrix}.$$

5. Пример

Рассмотрим дискретный аналог уравнения Лапласа

$$u_{i+1} - [4I - P]\, u_i + u_{i-1} - r_i = 0, \tag{1}$$

где, как в разд. 2, матрица Q заменена на $2I - P$. Предположим, что область квадратная. Таким образом, размерность u_i равна N, а u_0 и u_{N+1} известны. В гл. 6 мы видели, что (1) можно записать как

$$\begin{bmatrix} 4I-P & -I & & & 0 \\ -I & 4I-P & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & -I \\ 0 & & & -I & 4I-P \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} u_0+r_1 \\ r_2 \\ \vdots \\ \vdots \\ u_{N+1} \\ u_{N+1}-r_N \end{bmatrix}, \tag{2}$$

т. е. в виде соотношения

$$Au = r, \tag{3}$$

где размерности матрицы и векторов равны N^2.

Подробный анализ показывает, что матрица A имеет следующую структуру:

$$A = -I \otimes P - P \otimes I + 4I \otimes I. \tag{4}$$

Таким образом, решение уравнения (1) можно записать в виде

$$u = [-I \otimes P - P \otimes I + 4I \otimes I]^{-1} r. \tag{5}$$

Представив P как

$$P = FDF, \tag{6}$$

где

$$D=\begin{bmatrix} v_1 & & & \\ & v_2 & & \\ & & \ddots & \\ & & & v_N \end{bmatrix}, \tag{7}$$

получим, что (5) принимает вид

$$u = [F \otimes F]\,[-I \otimes D - D \otimes I + 4I \otimes I]^{-1}\,[F' \otimes F']\,r. \qquad (8)$$

Однако матрица $[-I \otimes D - D \otimes I + 4I \otimes I]^{-1}$ диагональна, и ее элементы равны $1/(4 - \nu_i - \nu_j)$. Поскольку мы уже показали, что

$$\nu_i = 2\cos[i\pi/(N+1)] \qquad (9)$$

и

$$f_{ij} = \sin[ij\pi/(N+1)], \qquad (10)$$

то (8) представляет собой конечный метод решения уравнения Лапласа на прямоугольных областях. В литературе этот метод известен как метод «тензорного произведения» [1]).

Упражнение

Показать, что в методе тензорного произведения требуется $O(N^3)$ операций.

6. Другой конечный метод

Пусть требуется решить уравнение Лапласа или Пуассона на прямоугольной области. В гл. 6 мы видели, что метод инвариантного погружения приводит к следующим уравнениям для определения вектора u_N:

$$\begin{aligned} R_{i+1} &= [2I + Q - R_i]^{-1}, \quad i = 0, 1, \ldots, \quad R_0 = 0 \\ s_{i+1} &= R_{i+1}(s_i + r_i), \qquad i = 0, 1, \ldots, \quad s_0 = c, \end{aligned} \qquad (1)$$

и

$$\begin{aligned} U_{i+1} &= R_{i+1}U_i, \\ &i = n, n+1, \ldots, N, \\ p_{i+1} &= p_i + U_i s_i, \\ p_n &= 0, \quad U_n = R_n. \end{aligned} \qquad (2)$$

Наконец, мы получили требуемое решение в виде

$$u_n = U_N d + p_N. \qquad (3)$$

Представим Q в форме

$$Q = FDF, \qquad (4)$$

где матрица D диагональна, а F — симметрическая ортогональная матрица, введенная в разд. 2. Тогда по индукции можно показать, что

$$R_i = F\overline{R}_i F \qquad (5)$$

---

[1]) Поскольку кронекерово произведение часто называют тензорным. См., например, М. Маркус, Х. Минк, «Обзор по теории матриц и матричных неравенств», изд-во «Наука», М., 1972, стр. 19.— *Прим. перев.*

и

$$U_i = F\bar{U}_i F, \tag{6}$$

где $\bar{R}_i$ и $\bar{U}_i$ — диагональные матрицы. Обозначим векторы $\bar{s}_i$ и $\bar{p}_i$ через

$$\bar{s}_i = Fs_i, \quad \bar{p}_i = Fp_i. \tag{7}$$

Теперь легко убедиться, что введенные величины удовлетворяют уравнениям

$$\begin{aligned} &\bar{R}_{i+1} = [D - \bar{R}_i]^{-1}, \quad \bar{R}_1 = 0, \quad i = 0, 1, \ldots, \\ &\bar{s}_{i+1} = \bar{R}_i(\bar{s}_i + F_i r_i), \quad \bar{s}_1 = Fc, \quad i = 0, 1, \ldots, \end{aligned} \tag{8}$$

и

$$\begin{aligned} &\bar{U}_{i+1} = \bar{R}_i \bar{U}_i, \\ &\bar{p}_{i+1} = \bar{p}_i + \bar{U}_i s_i, \quad i = n, n+1, \ldots, \\ &\bar{U}_n = \bar{R}_n, \quad p_n = 0. \end{aligned} \tag{9}$$

И наконец, по формуле

$$u_n = F\bar{U}_N F d + F\bar{p}_N \tag{10}$$

получаем требуемое решение. Поскольку все матрицы, за исключением F, диагональны, мы, таким образом, избежали обращения матриц. Кроме того, так как D и F известны, все необходимые операции легко выполнить.

Упражнение

Показать, что в этом методе требуется $O(N^2)$ операций для решения уравнения Лапласа и $O(N^3)$ для решения уравнения Пуассона.

7. Диагональная декомпозиция

В гл. 5 мы дискретизировали вариационную задачу, соответствующую уравнению Лапласа, следующим образом:

$$J(u) = \sum_{R=1}^{N} [(Qu_R, u_R) - (2r_R, u_R) + s_R + (u_R - u_{R-1}, u_R - u_{R-1})], \tag{1}$$

и с помощью минимизации функционала (1) методом динамического программирования нашли величины u_R.

Применяя принципы оптимальности, мы получили следующее функциональное уравнение для усеченной задачи минимизации:

$$f_R(v) = \min_{u_R} [(Qu_R, u_R) - (2r_R, u_R) + s_R + (u_R - v, u_R - v) + f_{R+1}(u_R)], \tag{2}$$

где

$$v = u_{R-1}. \tag{3}$$

Функция $f_1(u_0)$ доставляет минимум функционалу (1). Из уравнения (2) видно, что единственная связь между компонентами вектора состояния u_R осуществляется посредством матрицы Q. Возвращаясь опять к гл. 5, мы видим, что именно это обстоятельство вынуждало нас обращать матрицы.

Предположим, что имеется некоторое приближение к решению, т. е. множество векторов $\{u_r^{(0)}\}$. Матрицу Q можно представить в виде

$$Q = 2I - P, \tag{4}$$

где через P обозначена матрица

$$P = (p_{ij}), \quad p_{ij} = \begin{cases} 1, & |i-j| = 1, \\ 0 & \text{в противном случае.} \end{cases} \tag{5}$$

Таким образом, в силу (4)

$$(Qu_R,\ u_R) = (2u_R,\ u_R) - (Pu_R,\ u_R). \tag{6}$$

Если векторы $u_R^{(0)}$ и u_R близки, то

$$(P(u_R - u_R^{(0)}),\ u_R - u_R^{(0)}) \simeq 0, \tag{7}$$

поэтому

$$(Pu_R,\ u_R) \simeq (2Pu_R^{(0)},\ u_R) - (Pu_R^{(0)},\ u_R^{(0)}). \tag{8}$$

Заметим, что соотношение (7) позволяет записать

$$\frac{d}{du_R}(Pu_R,\ u_R) \simeq \frac{d}{du_R}[(2Pu_R^{(0)},\ u_R) - (Pu_R^{(0)},\ u_R^{(0)})], \tag{9}$$

что является очень важным свойством, которого мы бы не имели, если бы воспользовались более грубым приближением

$$(Pu_R,\ u_R) \simeq (Pu_R^{(0)},\ u_R). \tag{10}$$

Подставив теперь (8) в (2), получим приближенную формулу

$$f_R(v) = \min_{u_R}[(2u_R,\ u_R) - (2Pu_R^{(0)},\ u_R) + (Pu_R^{(0)},\ u_R^{(0)}) - (2r_R,\ u_R) + s_R + (u_R - v,\ u_R - v) + f_{R+1}(u_R)], \tag{11}$$

или

$$f_R(v) = \min_{u_R}[(2u_R,\ u_R) - (2\tilde{r}_R,\ u_R) + \tilde{s}_R + (u_R - v,\ u_R - v) + f_{R+1}(u_R)], \tag{12}$$

где

$$\tilde{r}_R = r_R + Pu_R^{(0)}, \quad \tilde{s}_R = s_R + (Pu_R^{(0)},\ u_R^{(0)}). \tag{13}$$

Ясно, что функция $f_R(v)$, определяемая равенством (12), квадратична по v. Итак, полагая

$$f_R(v) = (A_R v, v) - (2b_R, v) + c_R \tag{14}$$

и пользуясь той же процедурой, что и в гл. 5, получаем, что минимизирующее значение u_R равно

$$u_R = [3I + A_{R+1}]^{-1}(v + \tilde{r}_R + b_{R+1}), \tag{15}$$

а функции A_R и b_R удовлетворяют уравнениям

$$\begin{aligned} A_R &= I - [3I + A_{R+1}]^{-1}, \\ b_R &= [I - A_R](b_{R+1} + \tilde{r}_R) \end{aligned} \tag{16}$$

с начальными условиями

$$A_N = I, \quad b_N = u_{N+1}. \tag{17}$$

По индукции ясно, что матрица A_R, определяемая из (16), диагональна. Итак, здесь не требуется обращать матрицы. В действительности для получения A_R из A_{R+1} требуется только N делений, а для решения уравнений (15) — (17) только $2N^2$ умножений и делений. Разумеется, векторы $\{u_R\}$, определенные по этой процедуре, представляют собой лишь следующее приближение к решению, и поэтому мы должны повторять решение (15) и (16). С помощью (13) перепишем наши уравнения в виде

$$A_R = I - [3I + A_{R+1}]^{-1}, \quad A_N = I, \tag{18}$$

$$b_R^{(k+1)} = [I - A_R](b_{R+1}^{(k+1)} + r_R + Pu_R^{(k)}), \quad b_N^{(k+1)} = u_{N+1} \tag{19}$$

и

$$u_{R+1}^{(k+1)} = [I - A_R]\, u_R^{(k+1)} + b_R^{(k+1)}. \tag{20}$$

Поскольку уравнение для A_R не зависит от граничных условий и начального приближения, его достаточно решить только один раз.

Можно показать, что метод, определяемый соотношениями (18) — (20), сходится. Однако мы отложим доказательство сходимости, пока не получим ряд других итерационных методов из инвариантного погружения, поскольку в последнем случае доказательство оказывается проще.

8. Покоординатные итерационные методы

Дискретизация уравнения Лапласа приводит к матрично-векторному разностному уравнению

$$u_{R+1} - 2u_R + u_{R-1} - Qu_R + r_R = 0, \tag{1}$$

где u_0 и u_N известны. Используем тот факт, что Q можно представить в виде

$$Q = 2I - P. \tag{2}$$

Запишем теперь (1) как

$$u_R = \frac{1}{4}[Pu_R + u_{R+1} + u_{R-1} + r_R], \tag{3}$$

откуда немедленно следует простая итерационная схема:

$$u_R^{(k+1)} = \frac{1}{4}[Pu_R^{(k)} + u_{R+1}^{(k)} + u_{R-1}^{(k)} + r_R], \tag{4}$$

где $\{u_R^{(0)}\}$ — вектор начального приближения. Заметим, что хотя (4) по-прежнему записано в матрично-векторной форме, каждая компонента вектора $u_R^{(k+1)}$ вычисляется независимо от всех других компонент. Таким образом, все необходимые вычисления являются операциями над скалярами. Процедура (4) называется «точечным методом Якоби».

Если (4) сходится, то сходится к решению уравнения (3). Определим погрешность $e_R^{(k)}$ как

$$e_R^{(k)} = u_R - u_R^{(k)}. \tag{5}$$

Очевидно, что $e_R^{(k)}$ удовлетворяют разностному уравнению

$$e_R^{(k+1)} = \frac{1}{4}[Pe_R^{(k)} + e_{R+1}^{(k)} + e_{R-1}^{(k)}] \tag{6}$$

с начальными условиями

$$e_0^{(k)} = e_{N+1}^{(k)} = 0 \tag{7}$$

при всех k. Определим N^2-мерный вектор $e^{(k)}$ как

$$e^{(k)} = [e_i^{(k)}]. \tag{8}$$

Воспользовавшись кронекеровым произведением, перепишем (6) в виде

$$e^{(k+1)} = \frac{1}{4}[P \otimes I + I \otimes P]\, e^{(k)}. \tag{9}$$

Обозначим через T матрицу

$$T = \frac{1}{4}[P \otimes I + I \otimes P], \tag{10}$$

тогда

$$e^{(k)} = T^k e^{(0)}, \tag{11}$$

поэтому

$$\lim_{k \to \infty} e^{(k)} = 0 \tag{12}$$

тогда и только тогда, когда

$$\rho(T) < 1. \tag{13}$$

К счастью, мы показали, что собственными значениями матрицы P являются числа

$$\nu_i = 2\cos[i\pi/(N+1)]. \tag{14}$$

Поэтому

$$\rho(T) = \cos[\pi/(N+1)]. \tag{15}$$

Итак, данный метод сходится всегда. Однако при больших значениях N

$$\cos[\pi/(N+1)] \simeq 1 - \frac{1}{2}[\pi/(N+1)]^2. \tag{16}$$

Это означает, что при достаточно мелкой дискретизации сходимость будет чрезвычайно медленной.

В точечном методе Якоби приходится ждать окончания всех вычислений на k-й итерации, прежде чем использовать только что вычисленные новые значения. Если итерации производить в порядке возрастания величин x и y и использовать новые приближения по мере их вычисления, то получится метод Гаусса — Зейделя [1]):

$$u_R^{(k+1)} = \frac{1}{4}[Lu_R^{(k+1)} + Uu_R^{(k)} + u_{R-1}^{(k+1)} + u_{R+1}^{(k)} + r_R], \tag{17}$$

где L и U имеют вид

$$L = \begin{bmatrix} 0 & & & 0 \\ 1 & 0 & & \\ & 1 & \ddots & \\ & & \ddots & \ddots \\ & & \ddots & \ddots \\ 0 & & 1 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 1 & & 0 \\ & 0 & 1 & \\ & & \ddots & \ddots \\ & & \ddots & \ddots \\ & & \ddots & 1 \\ 0 & & & 0 \end{bmatrix}, \tag{18}$$

Очевидно, что L и U являются строго нижней треугольной и строго верхней треугольной матрицами соответственно, так что

$$L + U = P. \tag{19}$$

Упражнения

1. Показать, что матрица T в методе Гаусса — Зейделя определяется выражением

$$T = [I \otimes (4I - L) - L \otimes I]^{-1}[I \otimes U + U \otimes I].$$

2. Показать, что в этом случае $\rho(T) = \{\cos[\pi/(N+1)]\}^2$.

[1]) В этой связи точечный метод Якоби можно назвать «методом простой итерации» по аналогии с известным методом решения систем линейных алгебраических уравнений. См., например, Д. К. Фаддеев. В. Н. Фаддеева «Вычислительные методы линейной алгебры», Физматгиз, М., 1963.— *Прим. перев.*

9. Метод последовательной сверхрелаксации

Мы видели, что как в методе Якоби, так и в методе Гаусса — Зейделя спектральные радиусы матрицы T близки к единице, если размеры ячеек сетки малы. Чтобы обойти эту трудность, введем ускоряющий параметр.

Обозначим $(k+1)$-ю итерацию по методу Гаусса — Зейделя через $\bar{u}_R^{(k+1)}$:

$$\bar{u}_R^{(k+1)} = \frac{1}{4}\left[Lu_R^{(k+1)} + Uu_R^{(k)} + u_{R-1}^{(k+1)} + u_{R+1}^{(k)} + r_R\right]. \tag{1}$$

Построим следующее приближение $u_R^{(k+1)}$ как среднее взвешенное значение этой величины и предыдущей итерации:

$$u_R^{(k+1)} = \omega\bar{u}_R^{(k+1)} + (1-\omega)\, u_R^{(k)}, \tag{2}$$

где ω — ускоряющий параметр. Исключая $\bar{u}_R^{(k+1)}$ из (1) и (2), получаем, что точечный метод релаксации теперь определяется выражением

$$[4I - \omega L]\, u_R^{(k+1)} - \omega u_{R-1}^{(k+1)} = [4(1-\omega)\, I + \omega U]\, u_R^{(k)} + \omega u_{R-1}^{(k)} + \omega r_R. \tag{3}$$

Читатель легко может убедиться, что в этом случае матрица T имеет вид

$$T = [I \otimes (4I - \omega L) - \omega L \otimes I]^{-1}\, [I \otimes (4(1-\omega)\, I + \omega U) + \omega U \otimes I]. \tag{4}$$

Собственные значения матрицы T являются решениями векового уравнения

$$| T - \lambda I | = 0. \tag{5}$$

В упражнениях мы показали, каким образом можно решить это уравнение. Результат заключается в том, что i-е собственное значение λ_i матрицы T должно удовлетворять уравнению

$$\lambda_i + \omega - 1 = \mu_i \omega \lambda_i^{1/2}/2, \tag{6}$$

где μ_i является i-м собственным значением матрицы P. Заметим, что при $\omega = 1$ мы получаем собственные значения для метода Гаусса — Зейделя.

Теперь мы должны определить, при какой величине ω спектральный радиус матрицы T принимает наименьшее значение, т. е. надо найти

$$\rho(T) = \min_{\omega} \max_{i} |\lambda_i|. \tag{7}$$

Анализируя комплексное отображение, определяемое формулой (6), Янг показал, что оптимальное значение ω удовлетворяет уравнению

$$\omega^2 \nu^2 = 4(\omega - 1), \tag{8}$$

где ν — наибольшее собственное значение матрицы P. Определенный таким образом метод был назван «последовательной сверхрелаксацией», поскольку оптимальное значение ω удовлетворяет неравенству

$$1 \leqslant \omega < 2. \tag{9}$$

Наконец, мы получаем, что $\rho(T)$ определяется по формуле

$$\rho^2(T) = \frac{1-[1-(\rho/2)^2]^{1/2}}{1+[1-(\rho/2)^2]^{1/2}}, \tag{10}$$

где ρ — наибольшее собственное значение матрицы P. При больших значениях N будем иметь

$$\rho(T) \simeq 1 - 2\sqrt{2}\,[\pi/(N+1)]. \tag{11}$$

Определив скорость сходимости как

$$r(T) = -\log \rho(T), \tag{12}$$

видим, что при больших N скорость сходимости метода Гаусса — Зейделя вдвое больше, чем у метода Якоби, в то время как скорость сходимости метода последовательной сверхрелаксации в $2\sqrt{\ldots}$ раз больше, чем у метода Гаусса — Зейделя. Таким образом, при больших N метод последовательной сверхрелаксации обладает весьма существенным превосходством над другими итерационными методами, которые мы только что рассмотрели.

Упражнения

1. Показать, что (1) — (2) определяют точечный итерационный метод, т. е. все необходимые вычисления являются операциями над скалярами.

2. Пусть матрицы R и S определяются как

$$R = \begin{bmatrix} I & & & \\ & \lambda^{1/2} I & & 0 \\ & & \ddots & \\ 0 & & & \lambda^{(N-1)/2} I \end{bmatrix}, \quad S = \begin{bmatrix} 1 & & & \\ & \lambda^{1/2} & & 0 \\ & & \ddots & \\ 0 & & & \lambda^{(N-1)/2} \end{bmatrix},$$

так что размерности P и S равны N^2 и N соответственно. Показать, что если T определена, как в (4), то, умножив T слева на $[I \otimes S]\,P$ и затем справа на $[[I \otimes S]\,P]^{-1}$, получим собственные значения T (см. статью Д. Янга в списке литературы к этой главе).

10. Блочные итерационные методы

Все предыдущие итерационные методы обладают тем свойством, что при вычислении некоторой координаты вектора $u_R^{(k+1)}$ требуется выполнять лишь арифметические операции, включающие

имеющиеся приближения в соседних узлах сетки. Идея блочных итерационных методов состоит в том, чтобы вычислять значения функции одновременно в нескольких узлах. Мы рассмотрим только методы итераций по строкам, а итерации по столбцам оставляем в качестве упражнений.

Итак, начнем с исходной системы и опять разложим матрицу Q на составляющие. Тогда получим

$$-u_{R+1} + (4I - P)\, u_R - u_{R-1} + r_R = 0. \tag{1}$$

Это уравнение можно записать в виде следующего итерационного процесса:

$$-u_{R+1}^{(k+1)} + 4u_R^{(k+1)} - u_{R-1}^{(k+1)} = Pu_R^{(k)} + r_R. \tag{2}$$

Эта новая задача по-прежнему является двухточечной граничной задачей с заданными значениями u_0 и u_{N+1}. Будем искать решение в виде

$$u_{R+1}^{(k+1)} = A_R u_R^{(k+1)} + b_R^{(k+1)} \tag{3}$$

и, поступая, как и прежде, получим систему рекуррентных уравнений

$$\begin{aligned} A_{R-1} &= [4I - A_R]^{-1}, \\ b_{R-1}^{(k+1)} &= A_{R-1}\,(b_R^{(k+1)} + r_R + Pu_R^{(k)} \end{aligned} \tag{4}$$

с начальными условиями

$$A_N = 0, \quad b_N^{(k+1)} = u_{N+1}. \tag{5}$$

По индукции можно показать, что матрицы A_i диагональны, не зависят от номера итерации k и что все диагональные элементы матрицы A_i одинаковы. Таким образом, все матрицы A_R^{-1} можно определить с помощью N операций деления. Тогда каждая итерация состоит в отыскании новых значений $b_i^{(k)}$ и $u_i^{(k)}$, для чего требуется $2N^2$ умножений.

Приведенная выше процедура состоит в одновременном вычислении u_R в различных узлах сетки, как в методе Якоби. Иными словами, мы предполагаем, что значения u_R в соседних строках определяются на k-й итерации, в то время как значения в данной строке неявно вычисляются на $(k+1)$-й итерации. Легко также показать, что эта процедура эквивалентна диагональной декомпозиции.

Сходимость этого метода можно исследовать точно так же, как и ранее. Погрешность удовлетворяет уравнению

$$-e_{R+1}^{(k+1)} + 4e_R^{(k+1)} - e_{R-1}^{(k+1)} = Pe^{(k)}. \tag{6}$$

Очевидно, матрица T имеет вид

$$T = [4I \otimes I - I \otimes P]^{-1}\,[P \otimes I], \tag{7}$$

и поэтому

$$\rho(T) = (4 - \nu)/\nu, \tag{8}$$

где ν — наибольшее собственное значение матрицы P. Для больших значений N получаем

$$\rho(T) \simeq 1 - 2[\pi^2/(n+1)]. \tag{9}$$

Таким образом, скорость сходимости блочного метода Якоби вдвое больше скорости сходимости точечного метода Якоби. К сожалению, при блочном методе требуется вдвое больше операций на выполнение одной итерации, поэтому мы мало что выигрываем по сравнению с точечным итерационным процессом. Аналогичные результаты справедливы для блочного метода Гаусса — Зейделя и блочного метода последовательной сверхрелаксации; ни один из них не обладает какими-либо заметными преимуществами перед своими точечными аналогами.

Упражнения

1. Вывести блочные методы Гаусса — Зейделя и последовательной сверхрелаксации. Показать, что матрица T для метода сверхрелаксации определяется выражением

$$T = [(4I - \omega L) \otimes I - I \otimes P]^{-1}[(4(1-\omega)I + \omega U) \otimes I + (1-\omega) I \otimes P].$$

2. Показать, что «метод Якоби по столбцам» описывается уравнением

$$[2I + Q] u_R^{(k+1)} = u_{R+1}^{(k)} + u_{R-1}^{(k)} + r_R.$$

Проанализировать этот метод.

3. Показать, что метод Якоби по строкам эквивалентен диагональной декомпозиции.

11. Неявные схемы чередующихся направлений

Неявная схема Писмена и Ракфорда [1]) аналогична итерации столбцов, сопряженной с последующей итерацией строк. Поскольку этот метод возник из задачи решения параболического уравнения, остановимся на его описании. В наших обозначениях схема Писмена — Ракфорда принимает вид

$$[I + cQ] u_R^{(k+1/2)} = c[u_{R+1}^{(k)} + u_{R-1}^{(k)} + r_R] + (1 - 2c) u_R^{(k)} \tag{1}$$

и

$$cu_{R+1}^{(k+1)} - (2c+1) u_R^{(k+1)} + cu_{R-1}^{(k+1)} = [I - cQ] u_R^{\left(k+\frac{1}{2}\right)} + cr_R, \tag{2}$$

[1]) См. литературу к этой главе.— *Прим. ред.*

где c — скалярный параметр, который может изменяться в зависимости от k. Поскольку (1) является тридиагональной системой, мы можем решить это уравнение точно таким же образом, каким мы решали уравнение Лапласа, см. гл. 3, разд. 12. Все эти уравнения являются скалярными. Мы уже видели, что уравнение (2) также сводится к простым скалярным вычислениям. Легко показать, что для решения (1) и (2) требуется $O(4N^2)$ операций.

Обозначив N^2-мерный вектор через $e^{(k)}$

$$e^{(k)} = [e_R^{(k)}], \tag{3}$$

получим

$$e^{(k+1/2)} = [(I + cQ) \otimes I]^{-1} [(1 - 2c)\, I \otimes I + cP \otimes I]\, e^{(k)} \tag{4}$$

и

$$e^{(k+1)} = [cP \otimes I - (2c + 1)\, I \otimes I]^{-1} [I \otimes (I - cQ)]\, e^{(k+1/2)}. \tag{5}$$

Комбинируя эти равенства, получим, что матрица T определяется выражением

$$\begin{aligned} T = {} & [cP \otimes I - (2c + 1)\, I \otimes I]^{-1} [I \otimes (I - cQ)] \times \\ & \times [(I - cQ) \otimes I]^{-1} [(1 - 2c)\, I \otimes I + \epsilon P \otimes I]. \end{aligned} \tag{6}$$

После некоторых преобразований получим

$$\rho(T) = \left[\frac{1 - (\mu - 2)\, c}{1 + (\mu - 2)\, c} \right]^2, \tag{7}$$

где μ — собственное значение матрицы P. Если допустить, чтобы параметр c мог меняться в зависимости от k, то мы получим задачу отыскания последовательности $\{c_k\}$, минимизирующей абсолютную величину выражения

$$f(M, K) = \prod_{k=1}^{K} \frac{1 - (\mu - 2)\, c_k}{1 + (\mu - 2)\, c_k}. \tag{8}$$

Хотя $\{c_k\}$ можно определить для прямоугольной области, для областей общего вида эта задача очень сложна.

12. Обсуждение

Ясно, что для таких задач, как решение уравнения Лапласа на прямоугольной области, можно предложить ряд очень эффективных конечных и итерационных методов. К сожалению, как только область оказывается нерегулярной, или коэффициенты уравнения становятся переменными, выясняется, что эти методы или неприменимы, или сложны в употреблении. Так, хорошо известно, что неявные разностные схемы не работают на некоторых

нерегулярных областях очень простой структуры. Кроме того, мы видели, что если для ускорения сходимости требуется существенное усложнение итерационных методов, то возникает сложная задача определения оптимальных параметров. По этой причине не будем вдаваться в подробности таких методов. Однако читатель должен понимать, что эти методы тем не менее применимы к задачам, которые мы будем рассматривать впоследствии.

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Раздел 1. Обширную библиографию можно найти в работах

Варга (Varga R.)

Matrix iterative analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

Дорр (Dorr F. W.)

The direct solution of the discrete Poisson equation on a rectangle, *SIAM Rev.*, 12 (1970), 248—263.

Форсайт Г., Вазов В.

Конечно-разностные методы решения уравнений в частных производных, изд-во «Мир» М., 1960.

Карнахан, Лютер, Уилкс (Carnahan B., Luther H. A., Wilkes J. O.)

Applied numerical methods, Wiley, New York, 1969.

Раздел 2. Доказательство следует данному Д. Янгом; см.

Тодд (Todd J.)

Survey of numerical analysis, McGraw-Hill, New York, 1962.

Разделы 3, 4. См. книгу

Беллман Р.

Введение в теорию матриц, изд-во «Наука», М., 1969.

Раздел 5. См. статьи

Линч, Райс, Томас (Lynch R. E., Rice J. R., Thomas D. H.)

Tensor product analysis of partial difference equations, *Bull. Amer. Math. Soc.*, 70, 1964, 378—384.

Direct solution of partial difference equations by tensor product methods, *Numer. Math.*, 6 (1964), 185—199.

Интересный вариант метода тензорного произведения основан на теореме сгущения Дж. Вильямсона (*Bull. Amer. Math. Soc.*, 37 (1931), 585—590).

Если $A = [A_{ij}]$ — блочная матрица размера $M \times M$, каждый блок A_{ij} которой является матрицей размера $N \times N$ и при этом матрицы A_{ij} коммутируют (в частности, если все они являются рациональными функциями r_{ij} матрицы A с собственными значениями $\alpha_1, \alpha_2, \ldots, \alpha_N$), то собственными значениями A являются N множеств собственных значений $M \times M$-блоков $[r_{ij}(\alpha_k)]$, $k = 1, 2, \ldots, N$. Подробности см., например,

Тодд (Todd J.)

Numerical analysis, гл. 7 в книге «Handbook of physics», под редакцией E. U. Condon and H. Odishaw, 2nd ed., Part 1, 1967.

Раздел 6. Известен ряд других методов, зависящих от значения собственных чисел Q. См. статьи

Хокни (Hockney R. W.)

A fast direct solution of Poisson's equation using Fourier analysis, *J. Assoc. Comput. Mach.*, **12** (1965), 95—113.

Базби, Голуб, Нильсон (Buzbee B. L., Golub G. H., Nielson C. W.)

The method of odd-even reduction and factorization with applications to Poisson's equation, Los Alamos Scientific Laboratory Rept. LA-4141, 1969.
On direct methods for solving Poisson's equations, *SIAM J. Numer. Anal.*, **7** (1970), 627—656.

Раздел 7. См. статьи

Коллинз, Энджел (Collins D. C., Angel E.)

The diagonal decomposition technique applied to the dynamic programming solution of elliptic partial differential equations, *J. Math. Anal. Appl.*, 33 (1971), 467—481.

Коллинз, Лью (Collins D. C., Lew A.)

Dimensional approximation in dynamic programming by structural decomposition, *J. Math. Anal. Appl.*, 30 (1970), 375—384.

Ларсон (Larson R. E.)

State increment dynamic programming, American Elsevier, New York, 1968.

Раздел 8. См. статью

Энджел (Angel E.)

Invariant imbedding, difference equations, and elliptic boundary-value problems, *J. Comput. System Sci.*, **4** (1970), 473—491.

Раздел 9. См. статью

Янг (Young D.)

Iterative methods for solving partial difference equations of elliptic type, *Trans. Amer. Math. Soc.*, 76 (1954), 92—111.

Раздел 10. См. любую из ссылок к разд. 1 и 2, а также

Катхилл, Варга (Cuthill E. H., Varga R. S.)

A method of normalized block iteration, *J. Assoc. Comput. Mach.*, 6 (1959), 236—244.

Раздел 11. См. статьи

Писмен, Ракфорд (Peaceman D. W., Rachford H. H., Jr.)

The numerical solution of parabolic and elliptic differential equations, *J. Soc. Indust. Appl. Math.*, 3 (1955), 28—41.

Дуглас (Douglas J., Jr.)

On the numerical integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by implicit methods, *J. Soc. Indust. Appl. Math.*, 3 (1955), 42—65.

Биркгоф, Варга (Birckhoff G., Varga R.)

Implicit alternating directions methods, *Trans. Amer. Math. Soc.*, **92** (1959), 13—24.

Глава 9

Нестандартные разностные методы

1. Введение

В предыдущих главах мы изложили некоторые подходы к численному решению уравнений в частных производных, основанные на использовании аппарата динамического программирования и инвариантного погружения. Алгоритмы, полученные таким образом, довольно сильно отличаются от методов, полученных при использовании обычных конечно-разностных схем. В этой главе мы кратко коснемся некоторых нестандартных конечно-разностных методов, вытекающих из динамического программирования и инвариантного погружения.

2. Инвариантные погружения

Характерным уравнением в частных производных для теории инвариантного погружения является следующее:

$$r_t = g(r)\, r_x + h(r), \quad r(x, 0) = k(x). \tag{1}$$

Здесь функция $r(x, t)$ представляет собой отраженный поток. Разбиение на тонкие слои и последующий «подсчет баланса частиц» во многих физических процессах приводят к приближенному соотношению для этого отраженного потока

$$r(x, t + \Delta) = r(x + g(r)\,\Delta, t) + h(r)\,\Delta + O(\Delta^2), \tag{2}$$

из которого при $\Delta \to 0$ следует (1).

Представляет некоторый интерес рассмотреть этот процесс в обратном порядке и исследовать возможность использования уравнения (2) для получения численных значений $r(x, t)$. Это и есть некоторое конечно-разностное соотношение нестандартного типа.

3. Уравнение $u_t = uu_x$

Для иллюстрации основных идей рассмотрим уравнение

$$u_t = uu_x, \quad u(x, 0) = g(x), \quad -\infty < x < \infty. \tag{1}$$

Это отличный пример, на котором можно проверить различные вычислительные схемы, поскольку для функции u существует про-

стое неявное решение

$$u = g(x + ut). \tag{2}$$

Из (2) получаем

$$\begin{aligned} u_t &= g'[u + tu_t] = g'u/(1 - tg'), \\ u_x &= g'[1 + tu_x] = g'/(1 - tg'), \end{aligned} \tag{3}$$

следовательно, (1) выполняется.

Это решение существует, если только $1 - tg' \neq 0$. Следовательно, можно ожидать, что для некоторого значения x найдется первое значение t, такое, что $1 - tg' = 0$. В этой точке произойдет «скачок». В процессах переноса это соответствует «критической длине».

4. Приближенное конечно-разностное уравнение

Развивая идею разд. 2, рассмотрим соотношение

$$v(x,\ t + \Delta) = v(x + v(x,\ t)\,\Delta,\ t), \quad v(x,\ 0) = g(x), \tag{1}$$

где $-\infty < x < \infty$, а t принимает значения $0, \Delta, 2\Delta, \ldots$. Ради упрощения вычислений будем считать, что $g(x)$ — периодическая функция с периодом единица, поэтому $v(x,\ t)$ также периодична с периодом единица при всех t.

Существует несколько вариантов рекуррентного использования (1) с начальным условием $g(x) = v(x,\ 0)$ для определения функции $v(x,\ t)$. Первый из них состоит в запоминании $v(x,\ t)$ при всех t на множестве узлов сетки $x = k\delta$, $k = 0, 1, \ldots, M$, где $M\delta = 1$. В точках, где $x + v(x,\ t)\,\Delta$ не являются узлами сетки, правая часть уравнения (1) вычисляется с помощью некоторого интерполяционного метода, например посредством линейной интерполяции.

С другой стороны, при каждом t значения функции $u(x,\ t)$ можно запоминать с помощью какого-либо ее аналитического представления, например в виде тригонометрической суммы, отрезка степенного ряда, сплайна [1]), или с помощью дифференциальной аппроксимации и т. д.

Упражнения

1. Показать, что уравнение (1) является конечно-разностной аппроксимацией, сохраняющей положительность и ограниченность решения.

---

[1]) Относительно теории сплайнов см. книгу, упомянутую на стр. 37.— *Прим. ред.*

2. Получить аналогичные соотношения для системы уравнений в частных производных:

$$u_t = uu_x + vu_y, \quad v_t = uv_x + vv_y.$$

5. Сходимость

Возникает естественный вопрос: существует ли предел $v(x, t) \equiv v(x, t, \Delta)$ при $\Delta \to 0$ и, если да, то является ли он решением уравнения (3.1)? Этот вопрос включает несколько различных задач. Первая из них состоит из двух частей: во-первых, отталкиваясь от уравнения (4.1), показать, что существует предел $v(x, t, \Delta)$ при $\Delta \to 0$

$$\lim_{\Delta \to 0} v(x, t, \Delta) = u(x, t), \tag{1}$$

и, во-вторых, показать, что этот предел удовлетворяет уравнению (3.1). Вторая задача состоит в том, чтобы, предположив существование и единственность «хорошего» решения уравнения (3.1), показать, что решение уравнения (4.1) является удовлетворительным приближением к этому решению.

Интересно отметить, что эта вторая задача, наиболее для нас интересная, является задачей теории устойчивости, а именно устойчивости решения уравнения (4.1).

Пусть $u(x, t)$ — решение уравнения (3.1), где $g(x)$ предполагается дифференцируемой достаточное число раз. Тогда

$$\begin{aligned} u(x, t + \Delta) &= u(x, t) + \Delta u_t + (\Delta^2/2)\, u_{tt} + \ldots, \\ u(x + u\Delta, t) &= u(x, t) + \Delta u u_x + (\Delta^2/2)\, u^2 u_{xx} + \ldots. \end{aligned} \tag{2}$$

Следовательно,

$$u(x, t + \Delta) = u(x + u\Delta, t) + k(x, t)\, \Delta^2, \tag{3}$$

где $|k(x, t)| \leqslant k_1$ для $0 \leqslant x \leqslant 1$, $0 \leqslant t \leqslant T$.

Сравнивая это соотношение с (4.1), видим, что $v \simeq u$, если решение уравнения (4.1) устойчиво при $\Delta \ll 1$. Это можно показать для соответствующей области значений (x, t).

6. Повышение точности аппроксимации

Существует несколько способов улучшения точности аппроксимации. Самый простой из них состоит в уменьшении шага Δ, что, однако, немедленно приводит к возрастанию времени вычислений. Это неудобство можно в некоторой степени смягчить, применив идею замедленного устремления к пределу.

С другой стороны, мы можем воспользоваться разностной аппроксимацией более высокого порядка. Рассмотрим, например, соотношение

$$v(x,\ t+\Delta) = v(x + v(x + v(x,\ t)\,\Delta,\ t)\,\Delta,\ t), \qquad (1)$$

где $t = 0,\ \Delta,\ \ldots,$ а $v(x,\ 0) = g(x)$.

Вспоминая исходное уравнение в частных производных, видим, что погрешность этого алгоритма есть $O(\Delta^3)$. Тем не менее уменьшение Δ на практике оказывается более эффективным, чем использование усложненных разностных схем, таких, как (1).

Упражнения

1. Рассмотрим уравнение $u_t = u_{xx}$. Показать, что $u(x,\ t + \Delta) = [u(x + \delta,\ t) + u(x - \delta,\ t)]/2$ является адекватной разностной аппроксимацией, если Δ и δ связаны соответствующим образом. Показать, что этот алгоритм сохраняет положительность и ограниченность решения.

2. Рассмотрим рекуррентное соотношение

$$u(x,\ t+\Delta) = \sum_{i=1}^{N} w_i\,[u(x + a_i\delta,\ t) + u(x - a_i\delta,\ t)].$$

Можно ли выбрать параметры N, w_i, a_i и δ таким образом, чтобы это соотношение давало приближенное решение уравнения $u_t = u_{xx}$ с погрешностью Δ^k при любых $k \geqslant 1$, если, кроме того, $w_i,\ a_i \geqslant 0$?

3. Рассмотрим уравнение Бюргера $u_t = uu_x + u_{xx}$. Показать, что это уравнение можно линеаризовать с помощью преобразования типа преобразования Риккати и, таким образом, решить его как уравнение теплопроводности.

4. Рассмотрим рекуррентное соотношение

$$u(x,\ t+\Delta) = au(x + bu\delta,\ t) + (1-a)\left[\frac{u(x+\delta,\ t) + u(x-\delta,\ t)}{2}\right].$$

Показать, что можно выбрать параметры a, b, δ, удовлетворяющие неравенствам $0 < a < 1$, $b > 0$ и $\delta > 0$ и такие, чтобы это соотношение аппроксимировало уравнение Бюргера с погрешностью Δ. Показать, что при этом сохраняются положительность и ограниченность решения.

5. Рассмотрим систему уравнений:

$$u_t = uu_x + au_{xx},\quad u(x,\ 0) = g(x). \qquad (1)$$

Показать, что имеют место соотношения

$$u(x, t+\Delta) = au\,(u - au(x, t)\,\Delta, t) + \\ + \frac{1-\alpha}{2}\,[u(x + b\Delta^{1/2}, t) + u(x - b\Delta^{1/2}, t)] - \\ - 3\beta\left[h\left(x - \frac{\Delta}{\beta}\,h(x, t), t\right) - h(x, t)\right],$$

$$h(x, t+\Delta) = \gamma h(x - cu(x, t)\,\Delta, t) + \\ + \frac{1-\gamma}{2}\,[h(x + d\Delta^{1/2}, t) + h(x - d\Delta^{1/2}, t)] - \\ - \delta\left[u\left(x - \frac{\Delta}{\delta}\,h(x, t), t\right) - u(x, t)\right], \qquad (2)$$

где Δ — шаг интегрирования, а α, β, γ, δ, a, b, c, d — параметры, подлежащие определению. Получить с точностью до $O(\Delta)$ рекуррентное соотношение, если

$$a = 1/\alpha,\ b = [2/(1-\alpha)\,Re]^{1/2},\ c = 1/\gamma,\ d = [2/(1-\gamma)\,Rm]^{1/2}. \qquad (3)$$

См. статью Йон Да-тена (Dah-Teng Jong, Direct computational approaches to a magnetohydrodynamic model system (в печати)).

7. Дифференциальная квадратурная формула

Наконец, напомним метод дифференциальных квадратур, отмеченный в одной из первых глав [1]). Обозначим через $x_1, x_2, \ldots, x_N$ множество точек отрезка [0,1] и запишем

$$u_x|_{x=x_i} \simeq \sum_{j=1}^{N} a_{ij} u(x_j, t), \qquad (1)$$

где a_{ij} выбраны некоторым удобным образом. Тогда уравнение в частных производных

$$u_t = uu_x, \quad u(x, 0) = g(x) \qquad (2)$$

можно заменить системой обыкновенных дифференциальных уравнений

$$dv_i/dt \simeq v_i \sum_{j=1}^{N} a_{ij} v_j, \quad v_i(0) = g(x), \quad i = 1, \ldots, N, \qquad (3)$$

где $v_i \simeq u(x_i, t)$.

---

[1]) См. гл. 4, разд. 14. — *Прим. перев.*

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Раздел 4. См. статьи

Беллман, Черри, Винг (Bellman R., Cherry I., Wing G. M.)

Quart. Appl. Math., 48 (1962), 1325.

Эйзен, Беллман, Ричардсон (Azen S. P., Bellman R., Richardson J. M.)

Another computational approach to a mathematical model of turbulence, RM-3918-ARPA, The RAND Corporation, 1964.

Раздел 5. См. статью

Беллман, Кук (Bellman R., Cooke K. L.)

Existence and uniqueness theorems, in Invariant imbedding II: Convergence of a new difference algorithm, *J. Math. Anal. Appl.*, 12 (1965), 247—253.

Раздел 6. См. работу

Эйзен (Azen S. P.)

Higher order approximations to the computational solution of partial differential equations, RM-3917-ARPA, The RAND Corporation, 1964.

Раздел 7. См. статью

Беллман, Касти, Кашеф (Bellman R., Casti J., Kashef B.),

Differential quadrature, a rapid method for the computational solution of nonlinear partial differential equations (в печати).

Глава 10

Параболические уравнения

1. Уравнение теплопроводности

Вероятно, наиболее интересным примером уравнения в частных производных параболического типа является уравнение теплопроводности. Распределение температуры в однородном тонком стержне определяется решением уравнения

$$u_t = u_{xx}. \tag{1}$$

для которого известно начальное распределение

$$u(x, 0) = g(x) \tag{2}$$

и, кроме того, заданы условия на обоих концах стержня, т. е.

$$u(0, t) = a(t), \quad u(1, t) = b(t). \tag{3}$$

Мы будем рассматривать более общую задачу — задачу о тепловом потоке через область R с границей Г. Итак, будем искать решение уравнения

$$u_t = k(x, y)[u_{xx} + u_{yy}], \tag{4}$$

где

$$k(x, y) > 0, \quad (x, y) \in R, \tag{5}$$

с начальным и граничным условиями

$$u(x, y, 0) = g(x, y), \tag{6}$$

$$u(x, y, t) = f(x, y), \quad (x, y) \in \Gamma. \tag{7}$$

Мы увидим, что для получения распределения температуры в одномерном стержне можно будет воспользоваться одномерными аналогами методов, построенных нами для задач с двумя пространственными переменными.

И в этом случае основным алгоритмом исследования будет метод конечно-разностной аппроксимации. К сожалению, на этом пути мы немедленно сталкиваемся с двумя новыми трудностями, связанными с размерностью и устойчивостью. Увеличение размерности связано с тем, что теперь мы рассматриваем уравнение с тремя (а не с двумя, как прежде) независимыми переменными. Здесь следует проявлять осторожность, иначе число операций легко может превысить разумные пределы. Мы увидим, что, к сожалению,

простейшим вычислительным схемам присущи некоторые неблагоприятные свойства в отношении их численной устойчивости. Прежде чем приступить к изложению методов, рассмотрим некоторые математические трудности, о которых мы так долго умалчивали.

Упражнения

1. Пусть u удовлетворяет уравнению

$$u_t = u_{xx}, \quad u(x, 0) = g(x), \quad u(0, t) = u(1, t) = 0.$$

Показать, что

$$(d/dt)\left(\int_0^1 u^2\,dx\right) = 2\int_0^1 uu_t\,dx = 2\int_0^1 uu_{xx}\,dx = -2\int_0^1 u_x^2\,dx$$

и что поэтому

$$\int_0^1 u^2\,dx \leqslant \int_0^1 g^2(x)\,dx.$$

2. Аналогичным образом показать, что

$$\int_0^1 u^{2n}\,dx$$

является убывающей функцией t при $t \geqslant 0$.

3. Показать, что

$$\max_x |u| = \lim_{n\to\infty}\left(\int_0^1 u^{2n}\,dx\right)^{1/(2n)}.$$

4. Решив уравнение 3, доказать, что $\max\limits_x |u|$ является убывающей функцией t.

5. Показать, что для некоторого $\lambda > 0$

$$\int_0^1 u^2\,dx \leqslant \left(\int_0^1 g^2(x)\,dx\right) e^{-\lambda t}.$$

6. Получить аналогичные результаты для уравнения

$$u_t = uu_x + u_{xx}, \quad u(0, t) = u(1, t) = 0.$$

7. Получить аналогичные результаты для уравнения

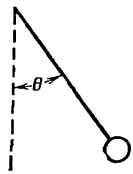
$$u_t = u_{xx} + u_{yy}, \quad u = 0 \quad \text{на } \Gamma, \quad t \geqslant 0.$$

2. Корректно поставленные задачи

В предыдущих главах мы пользовались некоторыми неявными предположениями. Вероятно, главным из них было предположение о том, что конечно-разностные методы приводят к приближенному решению поставленной задачи. В качестве примера ситуации, когда это не выполняется, рассмотрим движение маятника, состоящего из материальной точки, закрепленной на конце жесткого невесомого стержня, как на рис. 1. Движение маятника описывается уравнением

$$d^2\theta/dt^2 + k \sin \theta = 0. \tag{1}$$

Рис. 1



Пусть маятник находится в верхнем положении, т. е.

$$\theta(0) = \pi, \quad \theta'(0) = 0. \tag{2}$$

Тогда задача (1) — (2) имеет решение

$$\theta(t) = \pi, \quad t > 0. \tag{3}$$

Предположим теперь, что мы пытаемся получить решение этой задачи на цифровой вычислительной машине. Поскольку π иррационально, то начальные условия в действительности имеют вид

$$\theta(0) = \pi - \varepsilon, \quad \theta'(0) = 0, \tag{4}$$

где $\varepsilon \ll 1$. К сожалению, задача (1), (4) обладает осциллирующим решением, при котором $\theta(t)$ изменяется от $\pi - \varepsilon$ до $-\pi + \varepsilon$. Это пример плохо или некорректно поставленной задачи. Исследование задач такого типа восходит еще к Адамару.

Мы будем всегда предполагать, что рассматриваемая задача корректно поставлена. Будем называть задачу *корректно поставленной*, если ее решение существует, единственно и непрерывно зависит от начальных данных. Таким образом, приведенный пример не являлся корректно поставленной задачей, поскольку небольшое изменение начальных данных не привело к соответственно небольшому изменению решения.

3. Согласованность и устойчивость

Изложенный численный метод должен был заменить заданную граничную задачу системой разностных уравнений. Мы предполагали, что по мере стремления размера ячейки сетки к нулю реше-

ние разностных уравнений равномерно сходится к решению исходной задачи. Легко можно доказать, что для всех задач, которые мы изучали до сих пор, наши методы сходятся в указанном выше смысле. Однако для параболического уравнения, являющегося задачей Коши по временной переменной, определение сходимости немного сложнее: оно связано с проблемами согласованности и устойчивости.

Для того чтобы заменить параболические уравнения разностными, нам придется рассматривать дискретные значения как временной, так и пространственных переменных. Обозначим шаги дискретизации временной и пространственных переменных через Δ и h соответственно. Назовем систему разностных уравнений *согласованной* с данным уравнением в частных производных, если эти уравнения обращаются в данное уравнение в частных производных, когда Δ и h независимо стремятся к нулю. В литературе можно найти много примеров таких методов, которые являются согласованными, только когда Δ и h стремятся к нулю некоторым специальным образом. Мы будем рассматривать лишь такие способы дискретизации, которые приводят к разностным уравнениям, согласованным с интересующими нас параболическими уравнениями.

Назовем разностную аппроксимацию *устойчивой*, если при стремлении шага дискретизации к нулю никакое изменение начальных условий или появление ошибки вычислений не привносит неограниченного вклада в решение. Можно выяснить физический смысл этого понятия устойчивости. Поскольку нас интересуют параболические уравнения (или уравнения диффузии), то нам известно, что данное уравнение в частных производных не может иметь неограниченного решения. Таким образом, если разностная аппроксимация призвана моделировать интересующее нас уравнение, то мы вправе потребовать, чтобы эта аппроксимация не приводила к неограниченному решению, которое физически просто не может существовать. Такое новое определение устойчивости включает введенное ранее понятие численной устойчивости.

Связь между сходимостью, согласованностью и устойчивостью устанавливается теоремой, полученной П. Д. Лаксом, которую мы приводим без доказательства.

Теорема [1]). *Пусть дана корректно поставленная задача Коши и построена ее конечно-разностная аппроксимация, удовлетворяющая условию согласованности. Тогда устойчивость является необходимым и достаточным условием сходимости.*

---

[1]) Доказательство см., например, в книге Р. Рихтмайер, К. Мортон, Разностные методы решения краевых задач, «Мир», М., 1972, стр. 54.— *Прим. перев.*

Поскольку мы договорились рассматривать только согласованные методы, то в дальнейшем будем исследовать только устойчивость различных аппроксимаций.

Упражнения

1. Обозначим $u(ih, j\Delta)$ через u_{ij}. Показать, что при h и $\Delta \to 0$ разностные уравнения

$$(u_{i,j+1} - u_{ij})/\Delta = (u_{i+1,j} - 2u_{ij} + u_{i-1,j})/h^2 \tag{1}$$

переходят в уравнение теплопроводности

$$u_t = u_{xx}. \tag{2}$$

Таким образом, (1) согласовано с (2).

2. Показать, что схема Дюфорта — Френкеля

$$(u_{i,j+1} - u_{i,j-1})/2\Delta = (u_{i-1,j} - u_{i,j-1} - u_{i,j+1} + u_{i+1,j})/h^2 \tag{3}$$

согласована с (2), если $\Delta \to 0$ и $h \to 0$ таким образом, что Δ/h^2 остается постоянной. Показать, что если $\Delta \to 0$ и $h \to 0$ таким образом, что $\Delta/h = c$, то (3) согласовано с уравнением

$$u_{xx} = u_t + c^2 u_{tt}.$$

4. Явные методы

Теперь мы можем рассмотреть некоторые типичные разностные схемы. Поскольку было бы желательно обсудить свойства устойчивости этих методов более подробно, рассмотрим в качестве примера уравнение теплопроводности

$$u_t = c\,[u_{xx} + u_{yy}], \tag{1}$$

где c — положительная постоянная, и уравнение задано на единичном квадрате. Будем считать, что задано $u(x, y, 0)$ и значения u на границе области при всех t. Воспользуемся следующей дискретизацией:

$$x_i = ih, \quad y_j = jh, \quad t_l = l\Delta, \tag{2}$$

где

$$Nh = 1. \tag{3}$$

Обозначим $u(ih, jh, l\Delta)$ через $u_{ij}(t_l)$. Чтобы получить разностное уравнение, соответствующее уравнению (1), построим конечно-разностные аппроксимации для u_t

$$u_t(x, y, t) = \frac{u(x, y, t+\Delta) - u(x, y, t)}{\Delta} + O(\Delta) \tag{4}$$

и для $u_{xx}+u_{yy}$

$$u_{xx}+u_{yy}=\frac{u(x+h,\ y,\ t)+u(x,\ y+h,\ t)-4u(x,\ y,\ t)}{h^2}+$$
$$+\frac{u(x-h,\ y,\ t)+u(x,\ y-h,\ t)}{h^2}+O(h^2). \qquad (5)$$

Итак, в точке $x=ih$, $y=jh$, $t=l\Delta$ получаем

$$\frac{u_{ij}(t_{l+1})-u_{ij}(t_l)}{\Delta}=\frac{c}{h^2}[u_{i+1,\ j}(t_l)+u_{i,\ j+1}(t_l)-4u_{ij}(t_l)+$$
$$+u_{i-1,\ j}(t_l)+u_{i,\ j-1}(t_l)]+O(\Delta)+O(h^2). \qquad (6)$$

Из этого равенства вытекает

$$u_{ij}(t_{l+1})=u_{ij}(t_l)+(c\Delta/h^2)[u_{i+1,\ j}(t_l)+u_{i,\ j+1}(t_l)-4u_{ij}(t_l)+$$
$$+u_{i-1,\ j}(t_l)+u_{i,\ j-1}(t_l)]. \qquad (7)$$

Поскольку нам заданы $\{u_{ij}(t_0)\}$ и значения $u_{ij}(t_l)$ для i и j, равных 0 и N, то решение уравнения (7) тривиально. Каждое значение $u_{ij}(t_{l+1})$ можно вычислять независимо от остальных; поэтому этот метод назван явным. К сожалению, как мы сейчас увидим, этот метод будет неустойчивым, пока шаг дискретизации h не станет достаточно малым.

Как и прежде, обозначим через Q матрицу

$$Q=(q_{ij}),\quad q_{ij}=\begin{cases}2, & i=j,\\ -1, & |i-j|=1,\\ 0 & \text{в противном случае,}\end{cases} \qquad (8)$$

а через U_l — матрицу

$$U_l=(u_{ij}(t_l)),\quad i,\ j=1,\ \ldots,\ N-1. \qquad (9)$$

В этих обозначениях (7) принимает вид

$$U_{l+1}=U_l-r[QU_l+U_lQ]+S_l, \qquad (10)$$

где матрица S_l определяется граничными условиями и через r обозначено отношение $c\Delta/h^2$. Для исследования устойчивости этой явной разностной схемы выясним, при каких условиях однородное уравнение

$$U_{l+1}=U_l-r[QU_l+U_lQ] \qquad (11)$$

не обладает неограниченными решениями. Анализ однородного уравнения объясняется лишь тем, что если бы мы ввели возмущение в уравнение (10), то получили бы, что ошибка удовлетворяет уравнению (11). Это эквивалентно методике анализа ошибок, изложенной в гл. 5 и 6.

Обозначим $(N-1)^2$-мерный вектор, полученный объединением столбцов матрицы U_l, через u_l. Как и в гл. 8, с помощью кронеке-

рова произведения мы можем представить уравнение (11) в виде

$$u_{l+1} = [I - r(I \otimes Q + Q \otimes I)]\, u_l. \tag{12}$$

Для того чтобы (12) не обладало неограниченным решением, должно выполняться неравенство

$$\rho\,[I - r(I \otimes Q + Q \otimes I)] < 1. \tag{13}$$

Если i-е собственное значение матрицы Q равно μ_i, то собственные значения матрицы $I - r(I \otimes Q + Q \otimes I)$ равны

$$\lambda_k = 1 - r(\mu_i + \mu_j). \tag{14}$$

Поскольку мы уже показали, что

$$0 < \mu_i < 4, \tag{15}$$

то (13) имеет место, если

$$r < \frac{1}{4}. \tag{16}$$

Мы видели, что по мере уменьшения h наибольшее собственное значение Q сколь угодно близко приближается к 4. Таким образом, оценка (16) очень хорошая, и понятно, какое строгое ограничение на наш метод налагает это неравенство. Заметим, что для

$$c = 1, \quad h = 0{,}1 \tag{17}$$

мы должны выбрать шаг Δ, такой, что

$$\Delta < 0{,}0025. \tag{18}$$

Итак, простота этого метода уравновешивается необходимостью проводить столь мелкую дискретизацию временно́й переменной, что число вычислений становится недопустимо большим.

Упражнение

Показать, что при решении одномерного уравнения теплопроводности

$$u_t = cu_{xx}$$

для устойчивости явного метода требуется, чтобы $r < 1$.

5. Неявные методы

Вместо прямой разностной формулы (4.4), которую мы использовали для аппроксимации u_t, запишем теперь обратную формулу

$$u_t(x, y, t) = \frac{u(x, y, t) - u(x, y, t-\Delta)}{\Delta} + O(\Delta) \tag{1}$$

с той же самой погрешностью аппроксимации. Комбинируя (1) и (4.5), получим разностные уравнения

$$u_{ij}(t_l) - u_{ij}(t_{l-1}) = r[u_{i+1,\,j}(t_l) + u_{i,\,j+1}(t_l) - 4u_{ij}(t_l) + u_{i-1,\,j} + u_{i,\,j-1}(t_l)]. \quad (2)$$

Очевидно, что это неявная разностная схема. На каждом этапе мы имеем значения $\{u_{ij}(t_{l-1})\}$ и для определения $\{u_{ij}(t_l)\}$ должны решать систему алгебраических уравнений. В действительности решение уравнений (2) эквивалентно решению эллиптического уравнения того типа, что мы рассматривали ранее. Чтобы пояснить этот момент, вернемся к исходному параболическому уравнению

$$u_t = c(u_{xx} + u_{yy}). \quad (3)$$

Заменив u_t аппроксимацией (1), получим

$$u(x, y, t) - u(x, y, t - \Delta) = c\Delta[u_{xx}(x, y, t) + u_{yy}(x, y, t)], \quad (4)$$

что, очевидно, является эллиптическим уравнением при каждом фиксированном значении t. Если мы попытаемся решать это уравнение методами, разработанными ранее для решения эллиптических уравнений, то немедленно придем к уравнению (2). Итак, мы видим, что неявные методы эквивалентны решению эллиптического уравнения на каждом шаге, и поэтому для их реализации требуется большой объем вычислений.

Устойчивость уравнения (2) можно исследовать непосредственно. Если U_l, Q и S_l определены, как и выше, то (2) можно записать в виде

$$U_l - U_{l-1} = -r[QU_l + U_lQ] + S_l. \quad (5)$$

Рассмотрим соответствующее однородное уравнение

$$u_l - u_{l-1} = -r[Q \otimes I + I \otimes Q]\, u_l, \quad (6)$$

или

$$u_l = [I + r(Q \otimes I + I \otimes Q)]^{-1} u_{l-1}. \quad (7)$$

Если μ_i — собственное значение матрицы Q, то собственные значения матрицы $[I + r(Q \otimes I + I \otimes Q)]^{-1}$ равны

$$\lambda_k = 1/[1 + r(\mu_i + \mu_j)]. \quad (8)$$

Следовательно,

$$0 < \lambda_k < 1, \quad (9)$$

и поэтому (5) не имеет неограниченных решений. Это означает, что данная неявная схема устойчива при любых h и Δ.

6. Метод Кранка — Николсона

Мы показали, что изложенный нами простой неявный метод всегда устойчив, но требует решения эллиптического уравнения на каждом временном шаге процесса. Поскольку ошибка аппроксимации имеет порядок $O(\Delta) + O(h^2)$, то для достижения приемлемой точности мы должны выбрать очень и очень маленький шаг дискретизации по t. Таким образом, этот простой метод практически непригоден, поскольку в нем требуется чрезвычайно большое число вычислений. Эту трудность можно обойти, пользуясь более точными неявными разностными схемами.

Метод Кранка — Николсона вытекает из аппроксимации частных производных в окрестности $u(x, y, t + \Delta/2)$. С помощью разложения в ряд Тейлора легко проверить, что u_t можно записать как

$$u_t(x, y, t+\Delta/2) = \frac{u(x, y, t+\Delta) - u(x, y, t)}{\Delta} + O(h^2). \tag{1}$$

Аналогично $u_{xx} + u_{yy}$ можно представить в виде

$$u_{xx}(x, y, t+\Delta/2) + u_{yy}(x, y, t+\Delta/2) =$$
$$= \frac{1}{2}[u_{xx}(x, y, t+\Delta) + u_{yy}(x, y, t+\Delta) + u_{xx}(x, y, t) +$$
$$+ u_{yy}(x, y, t)] + O(\Delta^2). \tag{2}$$

Выражение $u_{xx} + u_{yy}$ в правой части (2) можно заменить стандартной аппроксимацией (4.5). Тогда комбинируя (1), (2) и (4.5), получим метод Кранка — Николсона:

$$u_{ij}(t_{l+1}) - u_{ij}(t_l) = (r/2)[u_{i+1, j}(t_{l+1}) + u_{i, j+1}(t_{l+1}) - 4u_{ij}(t_{l+1}) +$$
$$+ u_{i-1, j}(t_{l+1}) + u_{i, j-1}(t_{l+1}) + u_{i+1, j}(t_l) + u_{i, j+1}(t_l) -$$
$$- 4u_{ij}(t_l) + u_{i-1, j}(t_l) + u_{i, j-1}(t_l)], \tag{3}$$

погрешность которого составляет $O(\Delta^2) + O(h^2)$. Подставив (1) и (2) в исходное параболическое уравнение, легко убедиться, что уравнение (3) является конечно-разностной аппроксимацией эллиптического уравнения

$$u_{xx}(x, y, t+\Delta) + u_{yy}(x, y, t+\Delta) - (2\Delta/c)\, u(x, y, t+\Delta) =$$
$$= -u_{xx}(x, y, t) - u_{xx}(x, y, t) + (2\Delta/c)\, u(x, y, t). \tag{4}$$

Таким образом, в методе Кранка — Николсона на каждом временном шаге требуется по существу тот же самый объем вычислений, что и в неявном методе из разд. 5. Однако здесь мы можем выбирать больший шаг дискретизации по t, поскольку в данном случае ошибка аппроксимации меньше.

Покажем теперь, что метод Кранка — Николсона устойчив при любых положительных размерах шага. В предыдущих обозна-

чениях уравнение (3) можно записать в виде

$$U_{l+1} - U_l = -r\,[U_{l+1}Q + QU_{l+1} + U_lQ + QU_l] + S_l \qquad (5)$$

и соответствующее однородное уравнение в виде

$$u_{l+1} - u_l = -(r/2)\,[(I \otimes Q + Q \otimes I)\,u_{l+1} + (I \otimes Q + Q \otimes I)\,u_l]. \qquad (6)$$

Решив это уравнение относительно u_{l+1}, получим

$$u_{l+1} = [I + (r/2)\ (I \otimes Q + Q \otimes I)]^{-1} \times \\ \times [I - (r/2)\ (I \otimes Q + Q \otimes I)]\,u_l. \qquad (7)$$

Таким образом, собственные значения матрицы T равны

$$\lambda_k = \frac{1 - (r/2)\,(\mu_i + \mu_j)}{1 + (r/2)\,(\mu_i + \mu_j)}, \qquad (8)$$

где μ_i — собственное значение матрицы Q. Поскольку

$$0 < \mu_i < 4, \qquad (9)$$

то

$$0 < \lambda_k < 1, \qquad (10)$$

поэтому схема (3) устойчива при положительных Δ и h.

7. Неявные методы чередующихся направлений

Рассмотрим при фиксированном значении t следующие уравнения:

$$\frac{v(x, y) - u(x, y, t)}{\Delta/2} = c\,[u_{xx}(x, y, t) + v_{yy}(x, y)] \qquad (1)$$

и

$$\frac{w(x, y) - v(x, y)}{\Delta/2} = c\,[w_{xx}(x, y) + v_{yy}(x, y)]. \qquad (2)$$

Предположив, что функция $u(x, y, t)$ известна, уравнение (1) можно рассмотреть как уравнение для $v(x, y)$, а (2) — как уравнение относительно $w(x, y)$ с известным решением уравнения (1). Используя разложение в ряд Тейлора, нетрудно показать [1]), что если $u(x, y, t)$ является решением уравнения

$$u_t = c\,(u_{xx} + u_{yy}), \qquad (3)$$

то

$$w(x, y) = u(x, y, t + \Delta) + O(\Delta^2). \qquad (4)$$

Таким образом, можно ожидать, что этот метод обладает той же погрешностью, что и метод Кранка — Николсона. Однако он

[1]) Мы предполагаем, что соответствующие граничные условия учитываются при решении (1) и (2).

обладает весьма существенными вычислительными преимуществами. В уравнение (1) входят производные неизвестной функции v только по y, в то время как (2) содержит лишь производные от неизвестной функции v по x; отсюда и происходит название «неявный метод чередующихся направлений». На каждой стадии вычислений мы решаем два уравнения, гораздо более простых, чем эллиптические уравнения типа (5.4) или (6.4). Исследуем теперь разностные уравнения, соответствующие (1) и (2), чтобы убедиться, насколько просты эти вычисления.

Обозначим через $u_{ij}(t_{l+(1/2)})$ конечно-разностную аппроксимацию функции v. Хотя значения $u_{ij}(t_{l+(1/2)})$ непосредственно не связаны со значениями u и в точке $t + \Delta/2$, это обозначение стандартно. Тогда, используя обычную аппроксимацию вторых производных, запишем (1) и (2) в виде

$$u_{ij}(t_{l+1/2}) - u_{ij}(t_l) = (r/2)\,[u_{i+1,\,j}(t_l) - 2u_{ij}(t_l) + u_{i-1,\,j}(t_l) + u_{i,\,j+1}(t_{l+1/2}) - 2u_{ij}(t_{l+1/2}) + u_{i,\,j-1}(t_l)] \quad (5)$$

и

$$u_{ij}(t_{l+1}) - u_{ij}(t_{l-1/2}) = [u_{i+1,\,j}(t_{l+1}) - 2u_{ij}(t_{l+1}) + u_{i+1,\,j}(t_{l+1}) + u_{i,\,j+1}(t_{l+1/2}) - 2u_{ij}(t_{l+1/2}) + u_{i,\,j-1}(t_{l+1/2})]. \quad (6)$$

Из этих уравнений непосредственно вытекает, что ошибка аппроксимации имеет порядок $O(\Delta^2) + O(h^2)$. В матричной форме (5) и (6) принимают вид

$$[I + (r/2)\,Q]\,U_{l+1/2} = U_l\,[I - (r/2)\,Q] + S_l \quad (7)$$

и

$$U_{l+1}\,[I + (r/2)\,Q] = [I - (r/2)\,Q]\,U_{l+1/2} + S_l. \quad (8)$$

Поскольку матрица Q тридиагональная, каждое уравнение можно решить довольно просто. Как мы видели в гл. 8, для решения эллиптических уравнений потребуются скалярные аналоги матрично-векторных вычислений.

Устойчивость этого метода можно проанализировать, рассматривая однородные уравнения, соответствующие (7) и (8). Комбинируя эти два уравнения, получим

$$U_{l+1} = [I - (r/2)\,Q]\;[I + (r/2)\,Q]^{-1}\;U_l\,[I - (r/2)\,Q]\;[I + (r/2)\,Q]^{-1}. \quad (9)$$

Ясно, что матрицы $I - (r/2)\,Q$ и $I + (r/2)\,Q$ коммутируют. Обозначив через T матрицу

$$T = [I - (r/2)\,Q]\;[I + (r/2)\,Q]^{-1}, \quad (10)$$

из (9) получим

$$U_{l+1} = T^l U_0 T^l. \quad (11)$$

Поскольку собственные значения матрицы T равны

$$\lambda_i = \frac{1-(r/2)\,\mu_i}{1+(r/2)\,\mu_i}\,, \tag{12}$$

где μ_i — собственное значение Q, то

$$0 < |\,\lambda_i\,| < 1 \tag{13}$$

и поэтому данный метод устойчив для всех положительных значений Δ и h.

Так как решение уравнения Лапласа можно рассматривать как стационарное решение уравнения

$$u_t = c\,[u_{xx} + u_{yy}], \tag{14}$$

то благодаря высокой эффективности неявного метода чередующихся направлений вместо решения уравнения Лапласа мы можем решать уравнение (14) до тех пор, пока не придем к стационарному решению. В действительности неявный метод чередующихся направлений для решения задач эллиптического типа был предложен именно в такой формулировке. Мы не будем вдаваться в дальнейшее исследование этого метода, поскольку его анализ может оказаться очень сложным и к тому же эти методы до сих пор имеют ограниченную область применения.

8. Преобразование Лапласа

В этом разделе мы опишем совершенно иной подход к решению параболических уравнений, основанный на использовании преобразования Лапласа. Преобразование Лапласа является фундаментальным методом, позволяющим привести к более простому виду большое число уравнений. Мы предполагаем, что читатель имеет некоторое знакомство с преобразованием Лапласа.

Напомним, что преобразование Лапласа $L\,(u)$ функции $u\,(t)$ определяется как

$$L\,(u) = F\,(s) = \int_0^\infty u\,(t)\,e^{-st}\,dt. \tag{1}$$

Будем предполагать, что интеграл сходится абсолютно при $s \geqslant 0$. Легко показать, что

$$L\,(du/dt) = sF\,(s) - u\,(0). \tag{2}$$

Следовательно, преобразование Лапласа можно использовать для сведения обыкновенных дифференциальных уравнений с постоянными коэффициентами к алгебраическим [1]).

---

[1]) Следовало бы сказать «формального сведения».— *Прим. ред.*

Вернемся к нашему исходному уравнению

$$\partial u/\partial t = c\,[(\partial^2 u/\partial x^2) + (\partial^2 u/\partial y^2)], \tag{3}$$

где

$$u\,(x,\ y,\ 0) = g\,(x,\ y) \tag{4}$$

и функция $u\,(x,\ y,\ t)$ задана на сторонах единичного квадрата, и постараемся применить к этой задаче преобразование Лапласа. Применяя к (3) преобразование (1), (2) и обозначая

$$\tilde{u}\,(x,\ y,\ s) = L\,(u\,(x,\ y,\ t)), \tag{5}$$

получаем, что $\tilde{u}$ удовлетворяет уравнению

$$s\tilde{u} - g\,(x,\ y) = K\,(x,\ y)\,[(\partial^2\tilde{u}/\partial x^2) + (\partial^2\tilde{u}/\partial y^2)], \tag{6}$$

что представляет собой эллиптическое уравнение при любом заданном значении s. Граничными условиями для (6) являются условия, полученные после преобразования исходных граничных условий. Если граничные условия, заданные на сторонах квадрата, не зависят от t, то граничные условия для (6) равны просто исходным граничным условиям, деленным на s.

Наш подход состоит в решении задачи (6) при заданных значениях s, таких, как 1, 2, . . ., N, с помощью методов, которые мы до сих пор использовали. Тогда мы получим таблицу значений преобразований Лапласа для искомой функции. Успех в применении этой процедуры определяется нашим умением выполнять численное обращение преобразования Лапласа. Если мы имеем удобный способ вычисления обратного преобразования, то можно надеяться, что нам удастся решить исходное параболическое уравнение, сведя его к сравнительно небольшому числу эллиптических уравнений.

9. Квадратурная формула Гаусса

Численное обращение преобразования Лапласа основано на применении квадратурной формулы для интеграла, определяющего это преобразование. Напомним сначала некоторые основные сведения о квадратурной формуле Гаусса. Пусть требуется аппроксимировать определенный интеграл

$$I = \int_0^1 f(x)\,dx \tag{1}$$

конечной суммой

$$I_s = \sum_{i=1}^{N} w_i f\,(x_i). \tag{2}$$

Здесь $\{x_i\}$ — абсциссы, а $\{w_i\}$ — веса квадратурной формулы. Требуется выбрать абсциссы и веса так, чтобы (1) и (2) совпадали для всех $f(x)$, являющихся полиномами степени не выше $2N-1$. Мы не будем вдаваться в подробности определения абсцисс и весов, поскольку это можно найти в любом учебнике по численным методам анализа. Приведем лишь основные результаты.

Абсциссы $\{x_i\}$ являются корнями многочлена Лежандра $P_N(x)$ степени N, причем все они различны и лежат между нулем и единицей. Определим функции $L_i(x)$ как

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{N} (x - x_j) = P_N(x)/(x - x_i), \tag{3}$$

тогда веса определяются по формулам

$$w_i = [1/L_i(x_i)] \int_0^1 L_i(x)\, dx, \quad i = 1, 2, \ldots, N. \tag{4}$$

Корни многочленов Лежандра и веса можно либо найти в таблицах, либо, если необходимо, вычислить с произвольной степенью точности. Точность квадратурной формулы зависит от того, насколько удачно функция $f(x)$ может быть аппроксимирована полиномом. К этому вопросу мы еще ненадолго вернемся.

Совершив замену переменных

$$x = e^{-t} \tag{5}$$

в определении преобразования Лапласа

$$F(s) = \int_0^\infty u(t)\, e^{-st}\, dt, \tag{6}$$

получим

$$F(s) = \int_0^1 u(-\log x)\, x^{s-1}\, dx. \tag{7}$$

Это равенство послужит для нас отправной точкой. Учитывая (5), можно сказать, что требование того, чтобы квадратурная формула явилась бы хорошей аппроксимацией интеграла (7), эквивалентно тому, чтобы функцию $u(t)$ можно было хорошо аппроксимировать суммой конечного числа экспонент, т. е.

$$u(t) \simeq \sum_{i=1}^{N} a_k e^{-kt}. \tag{8}$$

Будем предполагать, что это всегда так.

10. Обращение преобразования Лапласа

Введем обозначение

$$g(x) = u(-\log x) \tag{1}$$

и перепишем (9.7) в виде

$$F(s) = \int_0^1 x^{s-1} g(x)\, dx. \tag{2}$$

Применяя к (2) квадратурную формулу Гаусса N-го порядка, получим приближенную формулу

$$F(s) = \sum_{i=1}^{N} w_i x_i^{s-1} g(x_i). \tag{3}$$

Если мы можем численно определить $F(s)$ для $s = 1, 2, \ldots, N$, то (3) приводит к системе линейных алгебраических уравнений

$$\sum_{i=1}^{N} w_i x_i^k g(x_i) = F(k+1), \quad k = 0, 1, \ldots, N-1, \tag{4}$$

с неизвестными $g(x_i)$. Чтобы упростить предыдущее выражение, введем обозначения

$$y_i = w_i g(x_i), \quad a_k = F(k+1), \tag{5}$$

тогда (4) примет вид

$$\sum_{i=1}^{N} x_{ik} y_i = a_k. \tag{6}$$

Это эквивалентно матрично-векторному уравнению

$$Xy = a, \tag{7}$$

где

$$X = (x_i^{j-1}). \tag{8}$$

Очевидно, что матрица X является матрицей Вандермонда. Хотя эта матрица и не вырождена (в силу того, что все x_i различны), она все же очень плохо обусловлена. Поэтому, если даже система (6) полностью определена, ее очень трудно решить численно при $N \gg 1$.

Однако небольшое усилие позволяет получить неявную формулу обращения, которая может быть использована с высокой степенью точности.

Умножим сначала k-е уравнение в (6) на неопределенный параметр q_k. Складывая N уравнений, получаем

$$\sum_{i=1}^{N} y_i \Big(\sum_{k=0}^{N-1} q_k x_{ik} \Big) = \sum_{k=0}^{N-1} a_k q_k. \tag{9}$$

Полагая

$$f(x) = \sum_{k=0}^{N-1} q_k x^k, \tag{10}$$

приходим к уравнению

$$\sum_{i=1}^{N} y_i f(x_i) = \sum_{k=0}^{N-1} q_k a_k. \tag{11}$$

Функция $f(x)$, являющаяся многочленом степени $N-1$, по-прежнему остается неопределенной. Повторим эту процедуру N раз и получим N различных многочленов $f(x)$ и, следовательно, N различных наборов коэффициентов q_k. Итак, положим

$$f(x) = f_j(x), \quad q_k = q_{kj}, \quad j = 1, 2, \ldots, N. \tag{12}$$

Поскольку нас интересуют значения y_i, выберем $f_j(x)$ так, чтобы

$$\begin{aligned} f_j(x_i) &= 0, \qquad i \neq j, \\ f_i(x_i) &= 1, \qquad i = j. \end{aligned} \tag{13}$$

Это позволяет переписать (11) в виде

$$y_i = \sum_{k=0}^{N-1} a_k q_{ki}. \tag{14}$$

Если мы сможем определить $f_j(x)$ и получим таким образом коэффициенты q_{ki}, то это и приведет к формуле обращения преобразования Лапласа. Поскольку числа x_i являются корнями многочлена Лежандра $P_N(x)$ степени N, то $P_N(x)$ можно представить в виде

$$P_N(x) = c \prod_{i=1}^{N} (x - x_i). \tag{15}$$

Первое условие в (13) требует, чтобы

$$f_j(x) = P_N(x)/[(x - x_j) P_N(x_j)]. \tag{16}$$

Числа q_{jk} являются коэффициентами этого многочлена. Поскольку коэффициенты многочлена $P_N(x)$ целые и корни его известны, то числа q_{jk} можно вычислить с любой степенью точности.

Подставив теперь (1) и (5) в (14), перепишем формулу обращения в следующем окончательном виде:

$$u(-\log x_i) = (1/w_i) \sum_{k=0}^{N-1} q_{ik} F(k+1). \tag{17}$$

11. Вычислительные аспекты

Коэффициенты q_{ik}/w_k были подсчитаны для значений N вплоть до 15 и приведены в книге, указанной в конце этой главы. Эти коэффициенты можно легко вычислить с помощью простой процедуры, множество входных параметров которой состоит из корней многочлена Лежандра и соответствующих весов. Эту процедуру также можно найти в указанной книге.

Условие

$$u(t) \simeq \sum_{i=1}^{N} a_k e^{-kt} \tag{1}$$

по существу является требованием «гладкости». Таким образом, мы не рассматриваем уравнений, решения которых обладают какими-либо пиками или быстро осциллирующими составляющими. Мы не рассчитываем получить какой-либо универсальный метод, пригодный для всех уравнений, поскольку, как известно, обратное преобразование Лапласа является неограниченным оператором. Однако данный метод можно модифицировать и для случая уравнений с сингулярным поведением решения.

Значения искомой функции были заданы в точках

$$t_i = -\log x_i, \tag{2}$$

где x_i — корни многочлена Лежандра. Поскольку x_i равномерно распределены на отрезке $[0, 1]$, значения вблизи нуля сгущаются. В большинстве задач это не столь важный фактор, поскольку мы можем выбрать подходящую интерполяционную формулу для вычисления $u(t)$ при любом заданном t. Кроме того, мы можем воспользоваться некоторыми свойствами преобразования Лапласа для данных значений $u(t)$ в точках, отличных от (2).

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Разделы 1—3. См., например,

Рихтмайер Р. и Мортон К.

Разностные методы решения краевых задач, «Мир», М., 1972.

Лакс (Lax P.)

Numerical solutions of partial differential equations, *Amer. Math. Monthly*, 72 (1965), 74—84.

Айзексон, Келлер (Isaacson E., Keller H. B.)

Analysis of numerical methods, Wiley, New York, 1966.

Раздел 4. В добавление к приведенным работам см. главу, написанную Х. Келлером, в книге

Ральстон, Вилф (Ralston A., Wilf H. S.)

Mathematical methods for digital computers, Wiley, New York, 1960.

См. также обзорную статью Д. Янга в книге

Тодд (Todd J., ed.)

Survey of numerical analysis, McGraw-Hill, New York, 1962.

Раздел 6. См. статью

Кранк, Николсон (Crank J., Nicolson P.)

A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type, *Proc. Cambridge Philos. Soc.*, 43 (1947), 50—67.

Раздел 7. См. статью

Писмен, Ракфорд (Peaceman D. W., Rachford H. H., Jr.),

The numerical solution of parabolic and elliptic differential equations, *J. Soc. Indust. Appl. Math.*, 3 (1955), 28—41.

Раздел 8. См., например,

Беллман, Калаба, Локетт (Bellman R., Kalaba R., Lockett J.)

Numerical inversion of the Laplace transform, Amer. Elsevier, New York, 1966.

Энджел (Angel E.)

Numerical inversion of the Laplace transform and multidimensiona heat equations, Electronic Sciences Laboratory Rept. 70—5, Univ. of Southern California, 1970.

Ван-дер-Поль, Бреммер (Van der Pol B., Bremmer H.)

Operational calculus based on the two-sided Laplace integral, Cambridge Univ. Press, London and New York, 1955.

Раздел 9. См. книгу

Ланцош К.

Практические методы прикладного анализа, Физматгиз, М., 1961.

Раздел 10. Доказательство невырожденности матрицы Вандермонда приведено в книге

Беллман Р.

Введение в теорию матриц, изд-во «Наука», М., 1969.

Раздел 11. Эти коэффициенты приведены в книге Беллмана, Калаба и Локетта, рекомендованной к разд. 8.

Глава 11

Нелинейные уравнения и квазилинеаризация

1. Введение

В предыдущих главах мы рассматривали исключительно линейные уравнения в частных производных. Требование линейности существенно для метода инвариантного погружения, в то время как квадратичная природа соответствующей вариационной задачи столь же сильно влияет на простоту окончательных результатов, полученных методом динамического программирования. В этой главе мы хотим расширить область применимости наших методов, включив в нее нелинейные уравнения, такие, как, например,

$$u_{xx} + u_{yy} = e^u, \tag{1}$$

уравнение, возникающее в магнитогидродинамике и других областях, и нелинейное уравнение теплопроводности

$$v_t = v_{xx} + v_{yy} + g(v). \tag{2}$$

Самый простой метод получения численного решения состоит в дискретизации данного уравнения. По причинам, которые мы уже отмечали в линейном случае, при этом подходе встречаются серьезные осложнения, которые многократно усиливаются в нелинейном случае. Поэтому здесь мы воспользуемся другим способом.

В конце главы кратко опишем применение этих методов к некоторым задачам идентификации.

2. Метод последовательных приближений

Мы уже показали, что можем построить простую вычислительную программу для численного решения линейного уравнения общего вида

$$u_{xx} + u_{yy} + h(x, y)\, u = f(x, y) \tag{1}$$

на области весьма произвольной структуры. Наша цель состоит в том, чтобы получить решение уравнения типа (1.1) как предел решений последовательности линейных уравнений типа (1). Такой подход включает и метод последовательных приближений — весьма общий и надежный метод анализа.

Простейшим вариантом этого мето а является метод Пикара. Вместо уравнения (1.1) рассмотрим последовательность линейных

уравнений

$$u_{xx}^{(n+1)} + u_{yy}^{(n+1)} = \exp\,(u^{(n)}), \tag{2}$$

где $u^{(0)}$ — заданное начальное приближение. Аналогично вместо (1.2) будем рассматривать последовательность линейных уравнений

$$v_t^{(n+1)} = v_{xx}^{(n+1)} + v_{yy}^{(n+1)} + g\,(v^{(n)}), \tag{3}$$

где $v^{(0)}$ — опять-таки заданная величина.

По-видимому, такой подход решает исходную задачу. Однако здесь имеются два источника трудностей. Во-первых, этот метод сходится медленно, в лучшем случае со скоростью некоторой геометрической прогрессии. Во-вторых, если $u^{(0)}$ и $v^{(0)}$ не выбраны соответствующим образом, то данный метод расходится.

В силу этих причин будем пользоваться другим вариантом метода последовательных приближений, а именно — вытекающим из динамического программирования.

3. Квазилинеаризация

В теории динамического программирования фундаментальную роль играет класс уравнений вида

$$\max_v T\,(u,\ v) = 0, \tag{1}$$

где T есть линейная функция от u. Функция u представляет собой функцию выигрыша. а v — политику управления. При решении уравнения (1) мы можем воспользоваться аппроксимацией в пространстве функций выигрыша, что является классическим подходом, или аппроксимацией в пространстве политик, представляющей собой новый метод, обладающий рядом желательных свойств.

Если мы выбираем политику v_0, то уравнение для соответствующей функции выигрыша u_0 является линейным:

$$T\,(u_0,\ v_0) = 0. \tag{2}$$

Из уравнения (2) вытекает метод последовательных приближений. Пусть новая политика v_1 определяется уравнением

$$T\,(u_0,\ v_1) = \max_v T\,(u_0,\ v), \tag{3}$$

и пусть u_1 — новая функция выигрыша, определяемая решением уравнения

$$T\,(u_1,\ v_1) = 0, \tag{4}$$

и т. д.

Можно показать, что во многих важных случаях сходимость метода монотонная и квадратичная. Ниже приведен соответствующий пример.

4. Пример

Рассмотрим уравнение

$$u^2 - 2 = 0. \tag{1}$$

Воспользуемся тождеством

$$u^2 = (v + (u - v))^2 = v^2 + 2v(u - v) + (u - v)^2. \tag{2}$$

Из (2) следует, что

$$u^2 \geqslant v^2 + 2v(u - v) = 2uv - v^2 \tag{3}$$

при всех v и что

$$u^2 = \max_v [2uv - v^2], \tag{4}$$

причем максимум достигается при $v = u$.

Итак, нелинейное уравнение имеет вид

$$\max_v [2uv - v^2 - 2] = 0. \tag{5}$$

Отсюда следует, что при всех v

$$2uv - v^2 - 2 \leqslant 0. \tag{6}$$

Если нас интересует лишь положительное решение, то

$$u \leqslant (v^2 + 2)/2v, \tag{7}$$

поэтому

$$u = \min_{v>0} [(v^2 + 2)/2v]. \tag{8}$$

Если мы используем (4) как основу для метода последовательных приближений и поступим, как выше, то получим широко известное рекуррентное соотношение

$$u_{n+1} = (u_n^2 + 2)/2u_n. \tag{9}$$

Это выражение совпадает с уравнением, полученным по методу Ньютона — Рафсона, для которого гарантирована квадратичная сходимость.

Упражнения

1. Пусть функция $g(u)$ выпукла. Показать, что $g(u) = \max_v [g(v) + (u - v)\, g'v]$.
2. Какова геометрическая интерпретация этого результата?
3. Каков многомерный аналог результата упражнения (1)?
4. Рассмотрим два соотношения

$$0 = f(u) = f(u_n) + f'(u_n)(u - u_n) + O[(u - u_n)^2],$$
$$0 = f(u_n) + f'(u_n)(u_{n+1} - u_n).$$

Показать, что этот метод сходится к корню уравнения $f(u) = 0$.

5. Уравнение $u_{xx} + u_{yy} = u^2$

Рассмотрим численное решение уравнения

$$u_{xx} + u_{yy} = u^2, \tag{1}$$

определенного в области R, на границе Г которой задано равенство

$$u = f. \tag{2}$$

Мы остановимся сначала на этом простом примере, поскольку он позволит нам наглядно продемонстрировать свойства сходимости метода квазилинеаризации.

Как мы уже отмечали, наиболее общим методом доказательства существования решения (1) и (2) является итерационный метод Пикара. Мы заменяем уравнение (1) последовательностью линейных задач

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} = [u^{(i)}]^2, \tag{3}$$

где

$$u^{(i+1)} = f, \quad (x, y) \in \Gamma. \tag{4}$$

Поскольку $[u^{(i)}]^2$ в уравнении (3) можно трактовать как возмущающую силу, то мы можем исследовать сходимость решения уравнения (3), представив функцию $u^{(i+1)}$ как функцию Грина в следующем виде:

$$u^{(i+1)}(x, y) = \int\int_R k(x, y, a, b)\,[u^{(i)}(a, b)]^2\,da\,db. \tag{5}$$

Не вдаваясь в подробности, отметим, что (3) действительно сходится к (1), если $v(0)$ — достаточно хорошее начальное приближение. Однако сходимость этого метода геометрическая. Это означает, что если мы определим функциональную норму как

$$\| g \| = \max_R | g(x, y) |, \tag{6}$$

то

$$\| u - u^{(i+1)} \| = O(\| u - u^{(i)} \|). \tag{7}$$

Итак, метод Пикара в общем случае не является эффективным численным методом.

Рассмотрим теперь квазилинейную аппроксимацию уравнения (1). Тогда придем к итерационному методу

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} = 2u^{(i+1)}u^{(i)} - [u^{(i)}]^2 \tag{8}$$

с граничным условием

$$u^{(i+1)} = f, \quad (x, y) \in \Gamma. \tag{9}$$

6. Дифференциальное неравенство

Мы хотим получить соотношение между решением дифференциального неравенства

$$u_{xx} + u_{yy} + qu \geqslant 0 \tag{1}$$

с граничным условием

$$u = f \tag{2}$$

и решением дифференциального уравнения

$$v_{xx} + v_{yy} + qv = 0 \tag{3}$$

с граничным условием

$$v = f. \tag{4}$$

Мы считаем q таким, что квадратичный функционал

$$H(u) = \iint_R [u_x^2 + u_y^2 - qu^2]\, dx\, dy \tag{5}$$

положительно определен. Простым достаточным условием этого является неравенство $q \leqslant 0$.

Для любой функции u, удовлетворяющей (1) и (2), можно записать

$$u_{xx} + u_{yy} + qu - p = 0, \tag{6}$$

где

$$p \geqslant 0. \tag{7}$$

Тогда, определив w как

$$w = v - u, \tag{8}$$

получим, что w должно удовлетворять уравнению

$$w_{xx} + w_{yy} + qw + p = 0 \tag{9}$$

с граничным условием

$$w = 0. \tag{10}$$

Поскольку p неотрицательно, а соответствующая функция Грина неположительна (это показано в разд. 6 гл. 1), то получаем, что

$$w \geqslant 0 \tag{11}$$

или, окончательно,

$$v(x, y) \geqslant u(x, y). \tag{12}$$

7. Монотонность

Теперь мы можем показать, что последовательность $\{u^{(i+1)}\}$, определяемая уравнением

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} = 2u^{(i+1)}u^{(i)} - [u^{(i)}]^2 \tag{1}$$

с граничным условием

$$u^{(i+1)} = f, \quad (x, y) \in \Gamma, \tag{2}$$

где $v^{(0)} \equiv f$, при $i \geqslant 1$ монотонно сходится к решению уравнения

$$u_{xx} + u_{yy} = u^2 \tag{3}$$

с граничным условием

$$u = f, \quad (x, y) \in \Gamma, \tag{4}$$

если это решение существует. Прежде всего докажем, что

$$u \geqslant u^{(i)}, \quad i = 1, 2, \ldots. \tag{5}$$

Мы уже отмечали, что в любой точке имеет место неравенство

$$2u^{(i)}u - [u^{(i)}]^2 \leqslant u^2. \tag{6}$$

Следовательно,

$$u_{xx} + u_{yy} \geqslant 2uu^{(i)} - [u^{(i)}]^2. \tag{7}$$

Определим последовательность функций $w^{(i+1)}$ равенством

$$w^{(i+1)} = u - u^{(i+1)}. \tag{8}$$

Тогда в силу (1)—(4) $w^{(i+1)}$ удовлетворяет неравенству

$$w_{xx}^{(i+1)} + w_{yy}^{(i+1)} - 2u^{(i)}w^{(i+1)} \geqslant 0 \tag{9}$$

с граничным условием

$$w^{(i+1)} = 0, \quad (x, y) \in \Gamma. \tag{10}$$

Используя результат из разд. 6, получим

$$w^{(i+1)} \geqslant 0, \tag{11}$$

и поэтому

$$u \geqslant u^{(i+1)}. \tag{12}$$

Выведем теперь аналогичное соотношение между $u^{(i+1)}$ и $u^{(i)}$. Положив в (3) и (6) $u = u^{(i+1)}$, будем иметь

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} \geqslant 2u^{(i+1)}u^{(i)} - [u^{(i+1)}]^2 \tag{13}$$

с граничным условием

$$u^{(i+1)} = f, \quad (x, y) \in \Gamma. \tag{14}$$

Повторяя предыдущие рассуждения, найдем

$$u^{(i)} \leqslant u^{(i+1)}. \tag{15}$$

Комбинируя (12) и (15), получим

$$u \geqslant u^{(i+1)} \geqslant u^{(i)} \geqslant \ldots \geqslant u^{(1)}, \tag{16}$$

т. е. монотонную сходимость.

8. Максимальная область сходимости

По поводу изложенного выше интересно отметить, что если уравнение имеет решение, то сходимость имеет место при любом выборе начального приближения. Таким образом, для данного метода последовательных приближений мы имеем максимально возможную область сходимости. Отметим еще раз, что в данном случае квазилинеаризация приводит к методу Ньютона — Рафсона — Канторовича и при этом сходимость оказывается монотонной.

9. Квадратичная сходимость

Если говорить о вычислительных аспектах, то наиболее важным свойством метода квазилинеаризации является квадратичная сходимость. Мы покажем, что для рассматриваемой задачи

$$\| u - u^{(i+1)} \| \leqslant k \| u - u^{(i)} \|^2, \tag{1}$$

где k — некоторая постоянная. Итерационное уравнение можно записать в виде

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} = [u^{(i)}]^2 + 2\,[u^{(i+1)} - u^{(i)}]\,u^{(i)}, \tag{2}$$

где

$$u^{(i+1)} = f, \quad (x,\ y) \in \Gamma. \tag{3}$$

Поскольку

$$u^2 = [u^{(i)}]^2 + 2u^{(i)}\,[u - u^{(i)}] + [u - u^{(i)}]^2, \tag{4}$$

то исходное уравнение

$$u_{xx} + u_{yy} = u^2 \tag{5}$$

можно представить в виде

$$u_{xx} + u_{yy} = [u^{(i)}]^2 + 2u^{(i)}\,[u - u^{(i)}] + [u - u^{(i)}]^2. \tag{6}$$

Вычитая (2) из (6) и обозначая

$$w^{(i)} = u - u^{(i)}, \tag{7}$$

получаем

$$w_{xx}^{(i+1)} + w_{yy}^{(i+1)} = 2u^{(i)} w^{(i+1)} + [w^{(i)}]^2 \tag{8}$$

с граничным условием

$$w^{(i+1)} = 0, \quad (x,\ y) \in \Gamma. \tag{9}$$

Используя функцию Грина, находим

$$w^{(i+1)}(x,\ y) = \int\int_R k(x,\ y,\ x',\ y')\,[w^{(i)}(x',\ y')]^2\,dx'\,dy'. \tag{10}$$

Теперь легко показать, что

$$w^{(i+1)}(x, y) \leqslant \max_R |w^{(i)}(x, y)| \iint_R |k(x, y, a, b)| \, da \, db. \quad (11)$$

Используя (7), этот результат можно представить в форме

$$\| u - u^{(i+1)} \| \leqslant k_1 \| u - u^{(i)} \|^2, \quad (12)$$

где k_1 — некоторая постоянная.

10. Вычислительные аспекты

В методе квазилинеаризации нелинейное эллиптическое уравнение

$$u_{xx} + u_{yy} = g(u) \quad (1)$$

заменяется последовательностью линейных задач

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} = g(u^{(i)}) + (u^{(i+1)} - u^{(i)}) \, g_u(u^{(i)}), \quad (2)$$

где (2) удовлетворяет тем же граничным условиям, что и (1). Нелинейное параболическое уравнение

$$u_t = u_{xx} + u_{yy} + g(u) \quad (3)$$

решается точно таким же образом, т. е. заменяется последовательностью задач

$$u_t^{(i+1)} = u_{xx}^{(i+1)} + u_{yy}^{(i+1)} + g(u^{(i)}) + (u^{(i+1)} - u^{(i)}) \, g_u(u^{(i)}). \quad (4)$$

Для произвольной функции $g(u)$, не обязательно выпуклой или вогнутой, мы могли бы и не получить столь благоприятных свойств сходимости, которые мы обнаружили в нашей модельной задаче. Однако для достаточно хороших функций $g(u)$ метод квазилинеаризации сходится к решению, если начальное приближение $u^{(0)}$ выбрано достаточно близко к решению. В большинстве физических задач бывает не очень сложно получить довольно хорошее начальное приближение. Обычно возникает проблема получения достаточной степени точности.

Для того чтобы получить приближенное решение $u^{(i+1)}$, мы должны хранить в памяти предыдущее решение $u^{(i)}$. Однако, если шаг сетки не выбран очень малым, объем необходимой памяти незначителен.

11. Пример

В качестве второго численного примера рассмотрим решение уравнения

$$u_{xx} + u_{yy} = e^u \quad (1)$$

на прямоугольнике $0 \leqslant x \leqslant 1/2$, $0 \leqslant y \leqslant 1/4$ с граничным условием

$$u = 0. \quad (2)$$

После квазилинеаризации уравнение (1) принимает вид

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} = \exp(u^{(i)})\,[u^{(i+1)} + (1 - u^{(i)})]. \tag{3}$$

Начальное приближение было выбрано равным

$$u^{(0)} = 0. \tag{4}$$

Было получено решение с точностью до четырех значащих цифр, как это показано в таблице I для двух типичных точек. Затем эта задача была решена еще раз, на этот раз с граничным условием

$$u = 10 \tag{5}$$

и начальным приближением

$$u^{(0)} = 6. \tag{6}$$

В этом случае для получения решения с точностью до четырех значащих цифр оказалось достаточно провести четыре итерации.

Таблица I

Итерация	$u\left(\frac{1}{4}, \frac{1}{8}\right)$	$u\left(\frac{1}{8}, \frac{1}{8}\right)$
0	0,0	0,0
1	−0,00707060	−0,00603040
2	−0,00707072	−0,00603050
3	−0,00707072	−0,00603050

Эта задача сначала была решена с помощью квазилинеаризации и последовательной сверхрелаксации. Было обнаружено, что оптимальное значение параметра релаксации оказывается различным для (3), (4) и (3), (5). Выбор этого параметра не оптимальным образом может удвоить число итераций. Более того, этот параметр был «оптимальным» лишь в том смысле, что он оставался постоянным на всех итерациях. Безусловно, такой выбор не является оптимальным для каждой конкретной итерации, поскольку функция $e^{u(i)}$ постоянно изменяется от итерации к итерации. Ясно таким образом, что итерационные методы решения линейных эллиптических уравнений не очень хорошо приспособлены для совместного исследования с методом квазилинеаризации. Однако конечные методы динамического программирования и инвариантного погружения, по-видимому, идеально приспособлены для целей квазилинеаризации.

12. Задачи идентификации

До сих пор мы рассматривали только одну задачу. Требовалось получить численное решение заданного уравнения в частных производных, удовлетворяющего заданному множеству граничных условий. Предположим, однако, что мы рассматриваем некоторый физический процесс, относительно которого известно, что интересующая нас переменная удовлетворяет уравнению вида

$$u_{xx} + u_{yy} = g(u, a), \tag{1}$$

где a — неизвестный параметр или вектор параметров. Пусть имеется возможность наблюдать данный физический процесс и замерять значения u в различных точках. На основе этих измерений требуется определить параметр a так, чтобы решение уравнения (1) согласовывалось с данными наблюдений.

В других формулировках этой задачи восстановления или идентификации требуется по данным наблюдений восстановить начальные или граничные условия. Задачи восстановления возникают в таких различных областях, как изучение распределения лекарств в человеческом организме, кардиология, предсказание погоды и др.

13. Критерий наименьших квадратов

Рассмотрим следующую задачу. Пусть имеется некоторый физический процесс, который, по предположению, описывается уравнением

$$u_{xx} + u_{yy} + g(u, a) = 0, \tag{1}$$

где

$$a = (a_1, a_2, \ldots, a_k) \tag{2}$$

является неизвестным вектором параметров. Будем считать известным, что

$$u = f \tag{3}$$

на границе интересующей нас области. Предположим, что мы производим ряд наблюдений этого процесса. Каждое наблюдение состоит из значения u_l, измеренного в точке наблюдения (x_l, y_l).

В качестве меры расхождения между наблюдениями и решением задачи (1), (2) для данного вектора a будем использовать критерий наименьших квадратов:

$$S = \sum_{l=1}^{L} [u(x_l, y_l) - u_l]^2. \tag{4}$$

Задача идентификации теперь становится задачей минимизации S по всем возможным выборам вектора a. Хотя существует множество других способов выбора критериев расхождения, удобные аналитические свойства критерия наименьших квадратов делают его популярным.

14. Метод Ньютона — Рафсона — Канторовича

Ясно, что если граничные условия и функция g достаточно гладкие, то S является непрерывной функцией от a. Чтобы подчеркнуть это, перепишем задачу в виде

$$\min_a S(a) = \min_a \sum_{l=1}^{L} [u(x_l, y_l, a) - u_l]^2. \tag{1}$$

Дифференцируя сумму в выражении (1) по a, получим необходимые условия минимума:

$$\frac{\partial S}{\partial a_i} = S_{a_i} = 0, \quad i = 1, 2, \ldots, k. \tag{2}$$

Или, вводя знак градиента, имеем

$$\nabla S(a) = 0. \tag{3}$$

Наш метод состоит в решении (2) [или (3)] методом Ньютона — Рафсона — Канторовича.

Определим векторную функцию $f(a)$ как

$$f(a) = \nabla S(a), \tag{4}$$

где

$$f(a) = [f_i(a)]. \tag{5}$$

Сначала разложим $f(a)$ в ряд Тейлора в окрестности вектора b [1]):

$$\begin{aligned} f_1(a) &= f_1(b) + (a_1 - b_1)[\partial f_1(b)/\partial a_1] + \\ &+ (a_2 - b_2)[\partial f_1(b)/\partial a_2] + \ldots + (a_k - b_k)[\partial f_1(b)/\partial a_k] + \ldots, \\ f_2(a) &= f_2(b) + (a_1 - b_1)[\partial f_2(b)/\partial a_1] + \ldots, \\ f_k(a) &= f_k(b) + (a_1 - b_1)[\partial f_k(b)/\partial a_1] + \ldots. \end{aligned} \tag{6}$$

Сохраняя только линейные члены, получаем приближенное выражение:

$$f(a) \simeq f(b) + J(b)(a - b), \tag{7}$$

где J — матрица Якоби:

$$J(b) = [\partial f_i(b)/\partial a_j]. \tag{8}$$

Поскольку нас интересуют корни уравнения

$$f(a) = 0, \tag{9}$$

то из формулы (7) получаем

$$a = b - J(b)^{-1} f(b). \tag{10}$$

---

[1]) Если скалярную функцию от векторного аргумента трактовать как обычную функцию многих переменных и положить $f_i(a) = f_i(b + a - b)$, то можно получить (6).— *Прим. ред.*

Так как это значение a является лишь приближением действительного значения корня, мы должны применить формулу (10) итеративно:

$$a^{(i+1)} = a^{(i)} - J^{-1}(a^{(i)})\, f(a^{(i)}). \tag{11}$$

Можно ожидать, что сходимость этого процесса будет квадратичной, т. е. что

$$\| a - a^{(i+1)} \| = O(\| a - a^{(i)} \|^2). \tag{12}$$

Поскольку $f(a)$ задана как

$$f(a) = \nabla S(a), \tag{13}$$

то матрица Якоби обращается в матрицу Гессе:

$$J(a) = (S_{a_i a_j}). \tag{14}$$

Итак, на каждой итерации (11) требуется вычислять $S_{a_i a_j}$ и S_{a_i} при $i = 1, \ldots, k$. Обсудим теперь, как вычисляются эти функции.

15. Уравнения чувствительности

Поскольку мы должны вычислять функции $S_{a_i a_j}$, распишем их подробно. Продифференцировав по a_i выражение

$$S = \sum_{l=1}^{L} [u(x_l, y_l) - u_l]^2, \tag{1}$$

получим

$$S_{a_i} = 2 \sum_{l=1}^{L} [u(x_l, y_l) - u_l]\, u_{a_i}(x_l, y_l) \tag{2}$$

и

$$S_{a_i a_j} = 2 \sum_{l=1}^{L} [u_{a_i}(x_l, y_l)\, u_{a_j}(x_l, y_l) - (u(x_l, y_l) - u_l)\, u_{a_i a_j}(x_l, y_l)]. \tag{3}$$

Итак, мы ввели функции чувствительности u_{a_j} и $u_{a_i a_j}$, которые должны вычисляться в каждой точке наблюдения. Покажем теперь, что эти функции удовлетворяют эллиптическим уравнениям специального вида.

Запишем исходное уравнение:

$$u_{xx} + u_{yy} + g(u, a) = 0. \tag{4}$$

Дифференцируя по a_i, получаем

$$(u_{a_i})_{xx} + (u_{a_i})_{yy} + g_u(u, a)\, u_{a_i} + g_{a_i}(u, a_i) = 0. \tag{5}$$

Продифференцировав еще раз по a_j, получим

$$(u_{a_i a_j})_{xx} + (u_{a_i a_j})_{yy} + g_u(u, a)\, u_{a_i a_j} + g_{uu}(u, a)\, u_{a_i} u_{a_j} + g_{u a_j}(u, a)\, u_{a_i} + g_{a_i u}(u, a)\, u_{a_j} + g_{a_i a_j}(u, a) = 0. \tag{6}$$

Поскольку граничные условия, наложенные на u, фиксированы, то на границах области имеют место равенства

$$u_{a_i} = 0 \tag{7}$$

и

$$u_{a_i a_j} = 0. \tag{8}$$

Заметим, что эти $L^2 + L$ уравнений чувствительности *линейны* и все имеют вид

$$v_{xx} + v_{yy} + g_u(u, a)\, v + q = 0, \tag{9}$$

где от уравнения к уравнению изменяется только функция q. Это однообразие мы позже используем.

16. Квазилинеаризация

В общем случае исходное уравнение

$$u_{xx} + u_{yy} + g(u, a) = 0 \tag{1}$$

нелинейно. Поэтому для получения численного решения мы пользуемся квазилинеаризацией. Заменим (1) последовательностью линейных уравнений

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} + g_u(u^{(i)}, a)\,[u^{(i+1)} - u^{(i)}] + g(u^{(i)}, a) = 0. \tag{2}$$

Это уравнение, как и уравнения чувствительности, имеет вид

$$v_{xx} + v_{yy} + g_u(u, v)\, v + p = 0. \tag{3}$$

Таким образом, все уравнения, определяемые равенством (2), и все уравнения чувствительности отличаются лишь правыми частями и граничными условиями. В этом состоит огромное вычислительное преимущество данного метода.

В главах 5 и 6 мы видели, что для численного решения уравнений типа (3) требуется решать матричные и векторные рекуррентные уравнения типа

$$\begin{aligned} A_{R-1} &= [2I + Q + F_R - A_R]^{-1}, \\ b_{R-1} &= A_{R-1}(b_R + r_R + p_R) \end{aligned} \tag{4}$$

с начальными условиями

$$A_N = 0, \quad b_N = v_N. \tag{5}$$

Здесь выражение F_R выводится из g_u, p_R из p и r_R из граничных условий. Таким образом, ясно, что поскольку A_{R-1} не зависит ни от правой части p, ни от граничных условий, то матрицы A_R одинаковы для последней итерации квазилинеаризации и всех $L^2 + L$ уравнений чувствительности. Поэтому решение матрич-

ного рекуррентного уравнения

$$A_{R-1} = [2I + Q + F_R - A_R]^{-1} \tag{6}$$

один раз вычисляется, запоминается и используется для построения решений всех уравнений чувствительности. Поскольку теперь для решения каждого уравнения чувствительности нам достаточно вычислить посредством (4) значение b_R и далее получить решение с помощью равенства

$$v_{i+1} = A_i v_i + b_i, \tag{7}$$

то таким образом мы избавляемся от большей части необходимых вычислений.

17. Пример

Рассмотрим пример только что описанной общей задачи. Пусть дано нелинейное уравнение

$$u_{xx} + u_{yy} + ae^{bu} = 0 \tag{1}$$

с граничным условием

$$u = g. \tag{2}$$

Попытаемся найти такие a и b, чтобы выражение

$$S(a, b) = \sum_{l=1}^{L} [u(x_l, y_l) - u_l]^2 \tag{3}$$

было минимальным. Прежде всего, при любых фиксированных a и b заменим задачу (1) последовательностью линейных задач

$$u_{xx}^{(n+1)} + u_{yy}^{(n+1)} + a \exp(bu^{(n)}) + ab(u^{(n+1)} - u^{(n)}) \exp(bu^{(n)}) = 0 \tag{4}$$

с граничным условием

$$u^{(n+1)} = g. \tag{5}$$

Оптимальные значения a и b определяются решением системы

$$S_a = 0, \qquad S_b = 0. \tag{6}$$

Используя далее метод Ньютона — Рафсона — Канторовича, строим последовательности $a^{(m)}$ и $b^{(m)}$ по формулам

$$\begin{bmatrix} a^{(m+1)} \\ b^{(m+1)} \end{bmatrix} = \begin{bmatrix} a^{(m)} \\ b^{(m)} \end{bmatrix} - J(a^{(m)}, b^{(m)})^{-1} \begin{bmatrix} S_a(a^{(m)}, b^{(m)}) \\ S_b(a^{(m)}, b^{(m)}) \end{bmatrix}, \tag{7}$$

где

$$J(a, b) = \begin{bmatrix} S_{aa}(a, b) & S_{ab}(a, b) \\ S_{ba}(a, b) & S_{bb}(a, b) \end{bmatrix}. \tag{8}$$

Дифференцируя (3), получаем

$$S_a = 2\sum_{l=1}^{L}[u(x_l, y_l) - u_l]\,u_a(x_l, y_l),$$
$$S_b = 2\sum_{l=1}^{L}[u(x_l, y_l) - u_l]\,u_b(x_l, y_l) \tag{9}$$

и

$$S_{aa} = 2\sum_{l=1}^{L}[u(x_l, y_l) - u_l]\,u_{aa}(x_l, y_l) + [u_a(x_l, y_l)]^2,$$
$$S_{bb} = 2\sum_{l=1}^{L}[u(x_l, y_l) - u_l]\,u_{bb}(x_l, y_l) + [u_b(x_l, y_l)]^2, \tag{10}$$
$$S_{ab} = S_{ba} = 2\sum_{l=1}^{L}[u(x_l, y_l) - u_l]\,u_{ab}(x_l, y_l) + u_a(x_l, y_l)\,u_b(x_l, y_l).$$

Дифференцируя (1), находим необходимые уравнения чувствительности:

$$(u_a)_{xx} + (u_a)_{yy} + e^{bu} + abe^{bu}u_a = 0,$$
$$(u_b)_{xx} + (u_b)_{yy} + aue^{bu} + abe^{bu}u_b = 0 \tag{11}$$

и

$$(u_{aa})_{xx} + (u_{aa})_{yy} + 2be^{bu}u_a + abe^{bu}u_{aa} + ab^2e^{bu}u_a^2 = 0,$$
$$(u_{bb})_{xx} + (u_{bb})_{yy} + 2ae^{bu}u_b + 2abu^{bu}eu_b +$$
$$+ au^2e^{bu} + abe^{bu}u_{bb} + ab^2e^{bu}u_b^2 = 0, \tag{12}$$
$$(u_{ab})_{xx} + (u_{ab})_{yy} = ue^{bu} + be^{bu}u_b + ae^{bu}u_a +$$
$$+ abue^{bu}u_a + abe^{bu}u_{ab} + ab^2e^{bu}u_au_b = 0,$$
$$u_{ba} = u_{ab},$$

с граничным условием

$$u_a = u_b = u_{aa} = u_{bb} = u_{ab} = 0. \tag{13}$$

Заметим, что все уравнения в (11) и (12) имеют вид

$$v_{xx} + v_{yy} + abe^{bu}v + p = 0. \tag{14}$$

Процедура их решения состоит в следующем. При данных значениях $a = a^{(m)}$ и $b = b^{(m)}$ решаем квазилинейное уравнение (4). После того как процесс сойдется и будет получено решение $u^{(n+1)}(x, y, a^{(m)}, b^{(m)})$, запомним матрицы A_R по последней итерации уравнения (4). С помощью этих матриц решим уравнения чувствительности (11) и (12) и последнее значение $u^{(n+1)}$ примем в ка-

честве u. Затем вычислим (9) и (10) и, решив (7), получим

$$a^{(m+1)} = a^{(m)} - \frac{S_{bb}(a^{(m)}, b^{(m)})\, S_a(a^{(m)}, b^{(m)}) - S_{ab}(a^{(m)}, b^{(m)})\, S_b(a^{(m)}, b^{(m)})}{S_{aa}(a^{(m)}, b^{(m)})\, S_{bb}(a^{(m)}, b^{(m)}) - [S_{ab}(a^{(m)}, b^{(m)})]^2},$$

$$b^{(m+1)} = b^{(m)} - \frac{S_{ab}(a^{(m)}, b^{(m)})\, S_a(a^{(m)}, b^{(m)}) - S_{bb}(a^{(m)}, b^{(m)})\, S_a b(a^{(m)}, b^{(m)})}{S_{aa}(a^{(m)}, b^{(m)})\, S_{bb}(a^{(m)}, b^{(m)}) - [S_{ab}\, a^{(m)}, b^{(m)})]^2}. \quad (15)$$

Эта процедура повторяется до тех пор, пока последовательности для $a^{(m)}$ и $b^{(m)}$ не сойдутся. При этом на каждой итерации предыдущее решение используется как начальное приближение, т. е.

$$u^{(0)}(x, y, a^{(m+1)}, b^{(m+1)}) = u^{(\bar{n})}(x, y, a^{(m)}, b^{(m)}), \quad (16)$$

где $\bar{n}$ относится к последней итерации предыдущего шага квазилинейной итерации.

Эта задача была численно решена на единичном квадрате, содержащем $(15^2) = 225$ внутренних точек. Результаты решения приведены в таблице II. Все вычисления заняли 8 секунд процессорного времени.

Таблица II

Итерация	a (k)	b (k)
0	1,5000	1,5000
1	1,6819	2,2737
2	1,6805	2,2686
3	1,6806	2,2687
4	1,6806	2,2687

Разные упражнения

1. Введем операцию S отыскания стационарной точки, определенную следующим образом: $S_u(f(u)) = f(v)$, где v есть предполагаемое единственное решение уравнения $f'(u) = 0$. Показать, что если $f'(u)$ нигде не обращается в нуль, то

$$f(u) = S_v[f(v) + (u - v) f'(v)].$$

Когда этот оператор является оператором максимизации и когда минимизации?

2. Показать, что если, кроме того, $f'(u)$ нигде не равна нулю, то можно записать

$$r = S_v[v - f(v)/f'(v)],$$

где r — предполагаемое единственное решение уравнения $f(u) = 0$.

ЛИТЕРАТУРА И КОММЕНТАРИЙ

Раздел 1. См. книгу

Беллман Р., Калаба Р.

Квазилинеаризация и нелинейные граничные задачи, «Мир», М., 1968.

Раздел 11. См. статью

Беллман, Юнкоза, Калаба (Bellman R., Juncosa M., Kalaba R.)

Some numerical experiments using Newton's method for nonlinear parabolic and elliptic boundary-value problems, *Comm. ACM*, 4 (1961), 187—191.

Раздел 6. См. книгу

Беккенбах Е. и Беллман Р.

Неравенства, «Мир», М., 1965.

Раздел 12. См. статьи

Касти, Детчменди, Кажива да, Калаба (Casti J., Detchmendy D., Kagiwada H., Kalaba R.)

Estimating the parameters of an inhomogeneous medium by probing with rays, *Comput. Approach in Appl. Mech., Amer. Soc. of Mech. Eng.*, 4 (1969), 200—209.

Энджел (Angel E.)

Inverse boundary-value problems for elliptic equations, *J. Math. Anal. Appl.*, 30 (1970), 86—98.

Приложение

Программы для ЭЦВМ

В этом приложении приведены программы четырех алгоритмов и результаты решения контрольных примеров. Хотя в качестве примеров выбраны уравнения с постоянными коэффициентами, настоящие программы не содержат никаких упрощений, описанных в гл. 8. Поэтому эти программы легко можно приспособить для решения широкого класса уравнений.

В программах использованы следующие процедуры.

MV (A, N, B, C) — процедура вычисляет произведение квадратной матрицы A на N-мерный вектор B и помещает результат в C.

MM (A, N, B, C) — процедура вычисляет произведение AB N-мерных квадратных матриц. Результат помещается в C.

MINV (A, N, D, L, M) — процедура обращает N-мерную матрицу A методом Гаусса — Жордана. Здесь L и M суть N-мерные рабочие векторы. При выходе из процедуры в D содержится определитель матрицы A, а в A — результат обращения.

Программа 1. Динамическое программирование

Уравнение Лапласа

$$u_{xx} + u_{yy} = 0,$$

определенное в квадратной области, решается методом динамического программирования. Матрицы $I - A_k$ и векторы b_R строятся, как это было описано в гл. 5, и хранятся на диске. Процесс решения описывается следующими уравнениями:

$$A_R = I - [I + Q - A_{R+1}]^{-1}, \quad A_N = I,$$

$$b_R = [I - A_R]\,(b_{R+1} + r_R), \quad b_N = u_N$$

и

$$u_R = [I - A_R]\,u_{R-1} + b_R.$$

```
0001              DIMENSION A(15,15),B(15),LL(15),LM(15),BB(15),U(15),
                 1UL(17),UR(17),UT(15),UB(15)
0002              DEFINE FILE 1(240,360,L,ID)
                C
                C  INITIALIZATION
                C
0003                   ID=1
0004                   READ(5,100) N,M
0005                100 FORMAT(2I2)
0006                   READ(5,101) UL
0007                   READ(5,101) UR
0008                   READ(5,101) UT
0009                   READ(5,101) UB
0010                101 FORMAT(9F6.3,/,8F6.3)
0011                   WRITE(6,103) N,M
0012                103 FORMAT('1 SOLUTION OF POTENTIAL EQUATION BY DYNAMIC PROGRAMMING',/
                    1/,110,' POINTS IN X DIRECTION',/,110,' POINTS IN Y DIRECTION',//)
0013                   DO 1 I=1,M
0014                   DO 1 J=1,M
0015                 1 A(I,J)=0.
0016                   DO 2 I=1,M
0017                   A(I,I)=1.
0018                 2 B(1)=UR(1+1)
0019                   DO 3 K=1,N
0020                   NN=N-K+1
                C
                C  COMPUTE I-A(I-1)
                C
0021                   DO 4 I=1,M
0022                 4 A(1,I)=A(I,I)+3.
0023                   DO 5 I=2,M
0024                   A(I,I-1)=A(I,I-1)-1.
0025                 5 A(I-1,I)=A(I-1,I)-1.
0026                   CALL MINV(A,M,D,LL,LM)
0027                   IF(D.EQ.0.) STOP
                C
                C  COMPUTE B(I-1)
                C
0028                   DO 6 I=1,M
0029                 6 BB(I)=B(I)
0030                   BB(1)=BB(1)+UB(NN)
0031                   BB(M)=BB(M)+UT(NN)
0032                   CALL MV(A,M,BB,B)
0033                   IF(K.EQ.N) GO TO 7
```

```
C
C  STORE I-A(I-1) AND B(I-1) ON DISC
C
0034          DO 8 I=1,M
0035        8 WRITE(1'ID) (A(I,J),J=1,M)
0036          WRITE(1'ID) (B(I),I=1,M)
C
C  COMPUTE A(I-1)
C
0037          DO 9 I=1,M
0038          DO 9 J=1,M
0039        9 A(I,J)=-A(I,J)
0040          DO 3 I=1,M
0041        3 A(I,I)=A(I,I)+1.
0042        7 ID=ID+M+1
0043          WRITE(6,102) UL
0044      102 FORMAT(1X,9F12.5,/,10X,8F12.5)
0045          DO 10 I=1,M
0046       10 BB(I)=UL(I+1)
0047          CALL MV(A,M,BB,U)
0048          DO 11 I=1,M
0049       11 U(I)=U(I)+B(I)
0050          WRITE(6,102) UB(1),U,UT(1)
0051          DO 12 K=2,N
C
C  RECALL I-A(I) AND B(I)
C
0052          ID=ID-2*(M+1)
0053          DO 13 I=1,M
0054       13 READ(1'ID) (A(I,J),J=1,M)
0055          READ(1'ID) (B(J),J=1,M)
C
C  COMPUTE U(I)
C
0056          CALL MV(A,M,U,BB)
0057          DO 14 I=1,M
0058       14 U(I)=BB(I)+B(I)
0059       12 WRITE(6,102) UB(K),U,UT(K)
0060          WRITE(6,110) UK
0061      110 FORMAT(1X,9F12.5,/,10X,8F12.5,/,'1')
0062          STOP
0063          END
```

SOLUTION OF POTENTIAL EQUATION BY DYNAMIC PROGRAMMING
15 POINTS IN X DIRECTION
15 POINTS IN Y DIRECTION

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.00429	0.00876	0.01359	0.01898	0.02517	0.03246	0.04124	0.05205
	0.06570	0.08346	0.10757	0.14231	0.19691	0.29386	0.49570	1.00000
0.0	0.00842	0.01717	0.02662	0.03715	0.04924	0.06343	0.08045	0.10127
	0.12728	0.16057	0.20450	0.26476	0.35149	0.48283	0.68895	1.00000
0.0	0.01220	0.02488	0.03856	0.05378	0.07120	0.09157	0.11586	0.14531
	0.18158	0.22704	0.28509	0.36077	0.46144	0.59701	0.77729	1.00000
0.0	0.01550	0.03160	0.04894	0.06822	0.09020	0.11580	0.14612	0.18251
	0.22670	0.28092	0.34804	0.43177	0.53649	0.66647	0.82321	1.00000
0.0	0.01819	0.03708	0.05740	0.07995	0.10558	0.13531	0.17029	0.21191
	0.26180	0.32188	0.39440	0.48179	0.58629	0.70917	0.84907	1.00000
0.0	0.02018	0.04113	0.06365	0.08858	0.11887	0.14956	0.18784	0.23305
	0.28670	0.35042	0.42590	0.51468	0.61772	0.73486	0.86389	1.00000
0.0	0.02140	0.04361	0.06747	0.09386	0.12376	0.15823	0.19846	0.24576
	0.30152	0.36720	0.44412	0.53329	0.63507	0.74864	0.87165	1.00000
0.0	0.02181	0.04445	0.06876	0.09564	0.12607	0.16114	0.20201	0.24999
	0.30644	0.37272	0.45007	0.53932	0.64061	0.75299	0.87407	1.00000
0.0	0.02140	0.04361	0.06747	0.09386	0.12376	0.15823	0.19846	0.24576
	0.30152	0.36720	0.44412	0.53330	0.63507	0.74864	0.87165	1.00000
0.0	0.02018	0.04113	0.06365	0.08858	0.11687	0.14956	0.18784	0.23305
	0.28670	0.35042	0.42591	0.51468	0.61773	0.73486	0.86389	1.00000
0.0	0.01819	0.03708	0.05740	0.07995	0.10558	0.13531	0.17030	0.21192
	0.26180	0.32189	0.39441	0.48179	0.58630	0.70918	0.84907	1.00000
0.0	0.01550	0.03160	0.04895	0.06822	0.09020	0.11580	0.14612	0.18251
	0.22670	0.28092	0.34805	0.43178	0.53650	0.66647	0.82321	1.00000
0.0	0.01220	0.02488	0.03856	0.05378	0.07120	0.09157	0.11586	0.14531
	0.18158	0.22704	0.28509	0.36077	0.46144	0.59701	0.77729	1.00000
0.0	0.00842	0.01717	0.02662	0.03715	0.04924	0.06343	0.08045	0.10127
	0.12728	0.16057	0.20450	0.26477	0.35149	0.48283	0.68896	1.00000
0.0	0.00429	0.00876	0.01359	0.01898	0.02517	0.03246	0.04124	0.05205
	0.06570	0.08346	0.10757	0.14231	0.19692	0.29386	0.49570	1.00000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Программа 2. Преобразование Риккати

Здесь также решается уравнение Лапласа. В этой программе используется только оперативная память. Поэтому можно ожидать, что программа работает быстрее, чем предыдущая. Процесс описывается уравнениями

$$A_R = [2I + Q - A_{R+1}]^{-1}, \quad A_{N-1} = 0,$$
$$b_R = A_R (b_R + r_R), \quad b_{N-1} = u_N$$

и

$$u_{R+1} = A_R u_R + b_R.$$

```
0001              DIMENSION A(15,15,16),B(15,16),U(15),BB(15),UL(17),UR(17),
                 1UT(15),UB(15),LL(15),LM(15)
                C
                C  INITIALIZATION
                C
0002              READ(5,100) M,N
0003            100 FORMAT(2I2)
0004              READ(5,101) UL
0005              READ(5,101) UR
0006              READ(5,101) UT
0007              READ(5,101) UB
0008            101 FORMAT(9F6.3,/,8F6.3)
0009              WRITE(6,103) N,M
0010            103 FORMAT('1 SOLUTION OF POTENTIAL EQUATION BY THE RICCATI TRANSFORMA
                   1TION',//,I10,' POINTS IN X DIRECTION',/,I10,' POINTS IN Y DIRECTIO
                   2N',//)
0011              DO 1 I=1,M
0012              B(I,N+1)=UR(I+1)
0013              DO 1 J=1,M
0014            1 A(I,J,N+1)=0.
0015              DO 2 KK=1,N
0016              K=N-KK+1
                C
                C  COMPUTE A(I+1) FROM A(I)
                C
0017              DO 3 I=1,M
0018              DO 3 J=1,M
0019            3 A(I,J,K)=-A(I,J,K+1)
0020              DO 4 I=1,M
0021            4 A(I,I,K)=A(I,I,K)+4.
0022              DO 5 I=2,M
0023              A(I,I-1,K)=A(I,I-1,K)-1.
0024            5 A(I-1,I,K)=A(I-1,I,K)-1.
0025              CALL MINV(A(1,1,K),M,D,LL,LM)
0026              IF(D.EQ.0.) STOP
```

```
C
C  COMPUTE B(I+1) FROM A(I+1) AND B(I)
C
0027          DO 6 I=1,M
0028        6 BB(1)=B(I,K+1)
0029          BB(1)=BB(1)+UB(K)
0030          BB(M)=BB(M)+UT(K)
0031        2 CALL MV(A(1,1,K),M,BB,B(1,K))
0032          WRITE(6,102) UL
0033      102 FORMAT(1X,9F12.5,/,10X,8F12.5)
0034          DO 7 I=1,M
0035        7 U(I)=UL(I+1)
0036          DO 8 K=1,N
C
C  COMPUTE U(I+1) FROM A(1),B(1) AND U(I)
C
0037          CALL MV(A(1,1,K),M,U,BB)
0038          DO 9 I=1,M
0039        9 U(I)=BB(I)+B(I,K)
0040        8 WRITE(6,102) UB(K),U,UT(K)
0041          WRITE(6,110) UR
0042      110 FORMAT(1X,9F12.5,/,10X,8F12.5,/,'1')
0043          STOP
0044          END
```

SOLUTION OF POTENTIAL EQUATION BY THE RICCATI TRANSFORMATION
15 POINTS IN X DIRECTION
15 POINTS IN Y DIRECTION

1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.0	0.49570	0.68896	0.77729	0.82321	0.84907	0.86389	0.87165	0.87407
	0.87165	0.86389	0.84907	0.82321	0.77729	0.68896	0.49570	0.0
0.0	0.29386	0.48283	0.59700	0.66647	0.70917	0.73485	0.74864	0.75299
	0.74864	0.73485	0.70917	0.66647	0.59701	0.48283	0.29386	0.0
0.0	0.19691	0.35148	0.46143	0.53649	0.58629	0.61772	0.63506	0.64060
	0.63506	0.61772	0.58629	0.53649	0.46144	0.35149	0.19691	0.0
0.0	0.14231	0.26476	0.36076	0.43177	0.48178	0.51467	0.53329	0.53931
	0.53329	0.51467	0.48178	0.43177	0.36076	0.26476	0.14231	0.0
0.0	0.10757	0.20449	0.28508	0.34804	0.39440	0.42590	0.44411	0.45006
	0.44411	0.42590	0.39440	0.34804	0.28508	0.20450	0.10757	0.0
0.0	0.08346	0.16057	0.22704	0.28092	0.32188	0.35042	0.36719	0.37272
	0.36719	0.35042	0.32188	0.28092	0.22704	0.16057	0.08346	0.0
0.0	0.06570	0.12728	0.18158	0.22670	0.26180	0.28670	0.30152	0.30644
	0.30152	0.28670	0.26180	0.22670	0.18158	0.12728	0.06570	0.0
0.0	0.05205	0.10127	0.14530	0.18251	0.21191	0.23305	0.24575	0.24999
	0.24575	0.23305	0.21191	0.18251	0.14531	0.10127	0.05205	0.0
0.0	0.04124	0.08045	0.11586	0.14612	0.17029	0.18784	0.19846	0.20201
	0.19846	0.18784	0.17030	0.14612	0.11586	0.08045	0.04124	0.0
0.0	0.03246	0.06343	0.09157	0.11580	0.13531	0.14956	0.15823	0.16113
	0.15823	0.14956	0.13531	0.11580	0.09157	0.06343	0.03246	0.0
0.0	0.02517	0.04924	0.07120	0.09020	0.10558	0.11687	0.12376	0.12607
	0.12376	0.11687	0.10558	0.09020	0.07120	0.04924	0.02517	0.0
0.0	0.01898	0.03715	0.05378	0.06822	0.07995	0.08858	0.09386	0.09564
	0.09386	0.08858	0.07995	0.06822	0.05378	0.03715	0.01898	0.0
0.0	0.01359	0.02662	0.03856	0.04894	0.05740	0.06365	0.06747	0.06876
	0.06747	0.06365	0.05740	0.04894	0.03856	0.02662	0.01359	0.0
0.0	0.00876	0.01717	0.02488	0.03160	0.03708	0.04113	0.04361	0.04445
	0.04361	0.04113	0.03708	0.03160	0.02488	0.01717	0.00876	0.0
0.0	0.00429	0.00842	0.01220	0.01550	0.01819	0.02018	0.02140	0.02181
	0.02140	0.02018	0.01819	0.01550	0.01220	0.00842	0.00429	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Программа 3. Инвариантное погружение

Эта программа тоже предназначена для численного решения уравнения Лапласа. В этом примере для получения решения внутри области используется сетка с тем же шагом, что и в двух предыдущих случаях. Заметим, что одношаговая природа вычислительной схемы позволяет ослабить требования к необходимой памяти. Таким образом, сначала решаются уравнения

$$R_i = [2I + Q - R_{i-1}]^{-1}, \quad R_0 = 0,$$
$$s_i = R_i [s_{i-1} + r_i], \quad s_0 = d,$$

и при выбранном k добавляются уравнения

$$U_{i+1} + U_i R_i, \quad U_k = R_k,$$
$$p_{i+1} = p_i + U_i s_i, \quad p_k = 0.$$

На последнем этапе вычисляется

$$u_k = U_N c + p_N.$$

```
0001            DIMENSION R(15,15),U(15,15),V(15,15),S(15),W(15),P(15),
               1UL(15),UR(15),UT(15),UB(15),IW(15),JW(15)
C
C   INITIALIZATION
C
0002            READ(5,100)  M,N,NN
0003          100 FORMAT(3I2)
0004            READ(5,101) UL
0005            READ(5,101) UR
0006            READ(5,101) UT
0007            READ(5,101) UB
0008          101 FORMAT(15F4.2)
0009            WRITE(6,102) N,M
0010          102 FORMAT('1 SOLUTION OF POTENTIAL EQUATION BY INVARIANT IMBEDDING',/
               1,110,' POINTS IN X DIRECTION',/,I10,' POINTS IN Y DIRECTION')
0011            WRITE(6,105) UT,UB,UL,UR
0012          105 FORMAT(/,' BOUNDARY VALUES UT,UB,UL,UR',(/,1X,8F12.5,/,5X,7F12.5))
0013            DO 1 I=1,M
0014            P(I)=0.
0015            S(I)=UL(I)
0016            DO 1 J=1,M
0017            U(I,J)=0.
0018          1 R(I,J)=0.
0019            DO 9 I=1,M
0020          9 U(I,I)=1.
0021            DO 2 K=1,N
C
C   COMPUTE R(I+1) FROM R(I)
C
0022            DO 3 I=1,M
0023            DO 3 J=1,M
0024          3 R(I,J)=-R(I,J)
0025            DO 4 I=1,M
0026          4 R(I,I)=R(I,I)+4.
0027            DO 5 I=2,M
0028            R(I,I-1)=R(1,I-1)-1.
0029          5 R(I-1,I)=R(I-1,I)-1.
0030            CALL MINV(R,M,DET,IW,JW)
0031            IF(DET) 13,6,13
```

```
C
C  COMPUTE S(I+1) FROM R(I+1) AND S(I)
C
0032       13 DO 7 I=1,M
0033        7 UL(I)=S(I)
0034          UL(I)=UL(I)+UB(K)
0035          UL(M)=UL(M)+UT(K)
0036          CALL MV(R,M,UL,S)
0037          IF(K-NN) 2,14,14
C
C  ADJOIN EQUATIONS FOR U(I) AND P(I)
C
0038       14 CALL MV(U,M,S,UL)
0039          DO 11 I=1,M
0040       11 P(I)=P(I)+UL(I)
0041          CALL MM(R,M,U,V)
0042          DO 10 I=1,M
0043          DO 10 J=1,M
0044       10 U(I,J)=V(I,J)
0045        2 CONTINUE
C
C  COMPUTE SOLUTION VECTOR
C
0046          CALL MV(U,M,UR,W)
0047          DO 12 I=1,M
0048       12 W(I)=P(I)+W(I)
0049          WRITE(6,110) NN
0050      110 FORMAT(77,' SOLUTION AT X=',I3,//)
0051          WRITE(6,104) W
0052      104 FORMAT(//,1X,8F12.5,/,5X,7F12.5,/,'1')
0053        6 STOP
0054          END
```

```
 SOLUTION OF PUTENTIAL EQUATION BY INVARIANT IMBEDDING
       15 POINTS IN X DIRECTION
       15 POINTS IN Y DIRECTION

BOUNDARY VALUES UI,UB,UL,UR
     1.00000     1.00000     1.00000     1.00000     1.00000     1.00000     1.00000     1.00000
         1.00000     1.00000     1.00000     1.00000     1.00000     1.00000     1.00000

     0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0         0.0

     0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0         0.0

     0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0         0.0


SOLUTION AT X=  8



     0.02181     0.04445     0.06876     0.09564     0.12607     0.16114     0.20201     0.24999
         0.30644     0.37272     0.45007     0.53932     0.64061     0.75299     0.87407
```

Программа 4. Квазилинеаризация

Здесь для решения нелинейного уравнения

$$u_{xx} + u_{yy} = u^2$$

используется комбинация метода преобразования Риккати и квазилинеаризации. Таким образом, программа решает последовательность линейных уравнений

$$u_{xx}^{(i+1)} + u_{yy}^{(i+1)} - 2u^{(i)}\,[u^{(i+1)}] = -[u^{(i)}]^2$$

с начальным условием

$$u^{(0)} = 0.$$

В распечатке приведены только первые три итерации, поскольку при последующих итерациях первые пять значащих цифр не изменяются.

```
      C
      C  QUASILINEARIZATION AND RICCATI TRANSFORMATION
      C
0001        DIMENSION A(15,15,16),B(15,16),U(15,15),BB(15),UL(17),UR(17),
           1UT(15),UB(15),LL(15),LM(15),UU(15)
0002        G(X)=X*X
0003        GU(X)=2.*X
0004        READ(5,100) M,N,H
0005      100 FORMAT(2I2,F10.5)
0006        READ(5,101) UL
0007        READ(5,101) UR
0008        READ(5,101) UT
0009        READ(5,101) UB
0010      101 FORMAT(9F6.3,/,8F6.3)
0011        DO 1 I=1,N
0012        1 READ(5,101) (U(I,J),J=1,M)
0013        ITER=0
0014        WRITE(6,102) ITER
0015      102 FORMAT(1H1,' ITERATION ',I2,///)
0016        WRITE(6,103) UL
0017      103 FORMAT(1X,9F12.5,/,10X,8F12.5)
0018        DO 2 I=1,N
0019        2 WRITE(6,103) UB(I),(U(I,J),J=1,M),UT(I)
0020        WRITE(6,103) UR
0021        DO 3 I=1,M
0022        B(I,N+1)=UR(I+1)
0023        DO 3 J=1,M
0024        3 A(I,J,N+1)=0.
0025       13 ITER=ITER+1
0026        DO 4 KK=1,N
0027        K=N-KK+1
0028        DO 5 I=1,M
0029        DO 5 J=1,M
0030        5 A(I,J,K)=-A(I,J,K+1)
```

```
0031              DO 6 I=1,M
0032            8 A(1,I,K)=A(1,I,K)+4.+H*H*GU(U(K,I))
0033              DO 7 I=2,M
0034              A(I,I-1,K)=A(I,I-1,K)-1.
0035            7 A(I-1,I,K)=A(I-1,I,K)-1.
0036              CALL MINV(A(1,1,K),M,D,LL,LM)
0037              IF(D.EQ.0.) STOP
0038              DO 8 I=1,M
0039            8 BB(I)=B(I,K+1)+H*H*(U(K,1)*GU(U(K,I))-G(U(K,I)))
0040              BB(1)=BB(1)+UB(K)
0041              BB(M)=BB(M)+UT(K)
0042            4 CALL MV(A(1,1,K),M,BB,B(1,K))
0043              WRITE(6,102) ITER
0044              WRITE(6,103) UL
0045              DO 9 I=1,M
0046            9 BB(I)=UL(I+1)
0047              SUM=0.
0048              DO 10 K=1,N
0049              CALL MV(A(1,1,K),M,BB,UU)
0050              DO 11 I=1,M
0051           11 BB(I)=UU(I)+B(I,K)
0052              DO 10 I=1,M
0053              SUM=SUM+ABS((BB(I)-U(K,I))/BB(I))
0054           10 U(K,I)=BB(I)
0055              DO 12 I=1,N
0056           12 WRITE(6,103) UB(I),(U(I,J),J=1,M),UT(I)
0057              WRITE(6,103) UR
0058              IF(SUM.GT.1.E-5.AND.ITER.LT.6) GO TO 13
0059              STOP
0060              END
```

```
ITERATION  0
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          1.00000
0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
          0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0
```

ITERATION 1

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.00429	0.00876	0.01359	0.01898	0.02517	0.03245	0.04124	0.05205
	0.06570	0.08346	0.10757	0.14231	0.19691	0.29386	0.49570	1.00000
0.0	0.00842	0.01717	0.02662	0.03715	0.04924	0.06343	0.08045	0.10127
	0.12728	0.16057	0.20450	0.26476	0.35149	0.48283	0.68895	1.00000
0.0	0.01220	0.02488	0.03856	0.05378	0.07120	0.09157	0.11586	0.14531
	0.18158	0.22704	0.28509	0.36077	0.46144	0.59701	0.77729	1.00000
0.0	0.01550	0.03160	0.04894	0.06822	0.09020	0.11580	0.14612	0.18251
	0.22670	0.28092	0.34804	0.43177	0.53649	0.66647	0.82321	1.00000
0.0	0.01819	0.03708	0.05740	0.07995	0.10558	0.13531	0.17029	0.21191
	0.26180	0.32188	0.39440	0.48179	0.58629	0.70917	0.84907	1.00000
0.0	0.02018	0.04113	0.06365	0.08858	0.11687	0.14956	0.18784	0.23305
	0.28670	0.35042	0.42590	0.51468	0.61772	0.73486	0.86389	1.00000
0.0	0.02140	0.04361	0.06747	0.09386	0.12376	0.15823	0.19846	0.24576
	0.30152	0.36720	0.44412	0.53329	0.63507	0.74864	0.87165	1.00000
0.0	0.02181	0.04445	0.06876	0.09564	0.12607	0.16114	0.20201	0.24999
	0.30644	0.37272	0.45007	0.53932	0.64061	0.75299	0.87407	1.00000
0.0	0.02140	0.04361	0.06747	0.09386	0.12376	0.15823	0.19846	0.24576
	0.30152	0.36720	0.44412	0.53330	0.63507	0.74864	0.87165	1.00000
0.0	0.02018	0.04113	0.06365	0.08858	0.11687	0.14956	0.18784	0.23305
	0.28670	0.35042	0.42591	0.51468	0.61773	0.73486	0.86389	1.00000
0.0	0.01819	0.03708	0.05740	0.07995	0.10558	0.13531	0.17030	0.21192
	0.26180	0.32189	0.39441	0.48179	0.58630	0.70918	0.84907	1.00000
0.0	0.01550	0.03160	0.04895	0.06822	0.09020	0.11580	0.14612	0.18251
	0.22670	0.28092	0.34805	0.43178	0.53650	0.66647	0.82321	1.00000
0.0	0.01220	0.02488	0.03856	0.05378	0.07120	0.09157	0.11586	0.14531
	0.18158	0.22704	0.28509	0.36077	0.46144	0.59701	0.77729	1.00000
0.0	0.00842	0.01717	0.02662	0.03715	0.04924	0.06343	0.08045	0.10127
	0.12728	0.16057	0.20450	0.26477	0.35149	0.48283	0.68896	1.00000
0.0	0.00429	0.00876	0.01359	0.01898	0.02517	0.03246	0.04124	0.05205
	0.06570	0.08346	0.10757	0.14231	0.19692	0.29386	0.49570	1.00000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ITERATION 2

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.00418	0.00852	0.01322	0.01847	0.02451	0.03164	0.04024	0.05086
	0.06431	0.08187	0.10581	0.14042	0.19498	0.29206	0.49436	1.00000
0.0	0.00818	0.01670	0.02589	0.03615	0.04793	0.06180	0.07846	0.09890
	0.12451	0.15741	0.20097	0.26097	0.34760	0.47921	0.68634	1.00000
0.0	0.01186	0.02419	0.03749	0.05231	0.06929	0.08918	0.11294	0.14181
	0.17750	0.22236	0.27987	0.35515	0.45572	0.59176	0.77361	1.00000
0.0	0.01506	0.03072	0.04758	0.06633	0.08774	0.11272	0.14235	0.17800
	0.22142	0.27486	0.34130	0.42454	0.52918	0.65986	0.81870	1.00000
0.0	0.01768	0.03603	0.05579	0.07771	0.10267	0.13165	0.16581	0.20655
	0.25551	0.31468	0.38638	0.47321	0.57770	0.70150	0.84394	1.00000
0.0	0.01961	0.03996	0.06184	0.08609	0.11361	0.14546	0.18282	0.22703
	0.27964	0.34234	0.41693	0.50512	0.60819	0.72643	0.85833	1.00000
0.0	0.02079	0.04237	0.06555	0.09120	0.12028	0.15386	0.19310	0.23933
	0.29399	0.35857	0.43455	0.52313	0.62497	0.73977	0.86584	1.00000
0.0	0.02119	0.04318	0.06679	0.09292	0.12253	0.15667	0.19654	0.24343
	0.29874	0.36392	0.44030	0.52895	0.63033	0.74397	0.86817	1.00000
0.0	0.02079	0.04237	0.06555	0.09120	0.12029	0.15386	0.19310	0.23933
	0.29399	0.35857	0.43455	0.52313	0.62498	0.73977	0.86584	1.00000
0.0	0.01961	0.03996	0.06184	0.08609	0.11361	0.14546	0.18282	0.22704
	0.27964	0.34234	0.41693	0.50512	0.60820	0.72644	0.85833	1.00000
0.0	0.01768	0.03603	0.05579	0.07771	0.10267	0.13165	0.16582	0.20655
	0.25551	0.31468	0.38639	0.47322	0.57770	0.70150	0.84394	1.00000
0.0	0.01506	0.03072	0.04758	0.06633	0.08775	0.11272	0.14235	0.17800
	0.22142	0.27487	0.34130	0.42454	0.52918	0.65986	0.81870	1.00000
0.0	0.01186	0.02419	0.03749	0.05231	0.06929	0.08918	0.11294	0.14182
	0.17750	0.22236	0.27987	0.35515	0.45572	0.59176	0.77361	1.00000
0.0	0.00818	0.01670	0.02589	0.03615	0.04794	0.06180	0.07846	0.09890
	0.12451	0.15741	0.20097	0.26097	0.34761	0.47922	0.68634	1.00000
0.0	0.00418	0.00852	0.01322	0.01847	0.02451	0.03164	0.04024	0.05086
	0.06431	0.08187	0.10581	0.14042	0.19498	0.29206	0.49436	1.00000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ITERATION 3

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.00418	0.00852	0.01322	0.01847	0.02451	0.03164	0.04024	0.05086
	0.06431	0.08187	0.10581	0.14042	0.19498	0.29206	0.49436	1.00000
0.0	0.00818	0.01670	0.02589	0.03615	0.04793	0.06180	0.07846	0.09890
	0.12451	0.15741	0.20097	0.26097	0.34760	0.47922	0.68634	1.00000
0.0	0.01186	0.02419	0.03749	0.05231	0.06929	0.08918	0.11294	0.14181
	0.17750	0.22236	0.27987	0.35515	0.45572	0.59176	0.77381	1.00000
0.0	0.01506	0.03072	0.04758	0.06633	0.08774	0.11272	0.14235	0.17800
	0.22142	0.27486	0.34130	0.42454	0.52918	0.65986	0.81870	1.00000
0.0	0.01768	0.03603	0.05579	0.07771	0.10267	0.13165	0.16581	0.20855
	0.25551	0.31468	0.38638	0.47321	0.57770	0.70150	0.84394	1.00000
0.0	0.01961	0.03996	0.06184	0.08608	0.11361	0.14546	0.18282	0.22705
	0.27964	0.34234	0.41693	0.50512	0.60819	0.72643	0.85833	1.00000
0.0	0.02079	0.04237	0.06555	0.09120	0.12028	0.15386	0.19310	0.23933
	0.29399	0.35857	0.43455	0.52313	0.62497	0.73977	0.86584	1.00000
0.0	0.02119	0.04318	0.06679	0.09292	0.12252	0.15667	0.19654	0.24343
	0.29874	0.36392	0.44030	0.52895	0.63033	0.74397	0.86817	1.00000
0.0	0.02079	0.04237	0.06555	0.09120	0.12028	0.15386	0.19310	0.23933
	0.29399	0.35857	0.43455	0.52313	0.62497	0.73977	0.86584	1.00000
0.0	0.01961	0.03996	0.06184	0.08609	0.11361	0.14546	0.18282	0.22703
	0.27964	0.34234	0.41693	0.50512	0.60819	0.72644	0.85833	1.00000
0.0	0.01768	0.03603	0.05579	0.07771	0.10267	0.13165	0.16581	0.20655
	0.25551	0.31468	0.38638	0.47322	0.57770	0.70150	0.84394	1.00000
0.0	0.01506	0.03072	0.04758	0.06633	0.08774	0.11272	0.14235	0.17800
	0.22142	0.27487	0.34130	0.42454	0.52918	0.65986	0.81870	1.00000
0.0	0.01186	0.02419	0.03749	0.05231	0.06929	0.08918	0.11294	0.14182
	0.17750	0.22236	0.27987	0.35515	0.45572	0.59176	0.77361	1.00000
0.0	0.00818	0.01670	0.02589	0.03615	0.04793	0.06180	0.07846	0.09890
	0.12451	0.15741	0.20097	0.26097	0.34761	0.47922	0.68634	1.00000
0.0	0.00418	0.00852	0.01322	0.01847	0.02451	0.03164	0.04024	0.05086
	0.06431	0.08187	0.10581	0.14042	0.19498	0.29206	0.49436	1.00000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Именной указатель

Предметный указатель

Оглавление