Министерство науки и высшего образования
Российской Федерации

Томский государственный университет систем
управления и радиоэлектроники

С. П. Куксенко

ЭЛЕКТРОМАГНИТНАЯ СОВМЕСТИМОСТЬ: ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ЭЛЕКТРОСТАТИКИ

Учебное пособие
для студентов технических направлений подготовки
и специальностей

Томск
Издательство ТУСУРа
2020

УДК 621.391.823(075.8)
ББК 32.841.174я73
К898

Рецензенты:
Костарев И. С., канд. техн. наук, нач. отд. АО «НПЦ «Полюс»;
Гизатуллин З. М., д-р техн. наук, проф.
каф. систем автоматизированного проектирования
Казанского национального исследовательского
технического университета им. А. Н. Туполева

Куксенко, Сергей Петрович

К898 Электромагнитная совместимость: численные методы решения задач электростатики: учеб. пособие для студентов техн. направлений подготовки / С. П. Куксенко. – Томск: Изд-во Томск. гос. ун-та систем упр. и радиоэлектроники, 2020. – 268 с.

ISBN 978-5-86889-879-2

Показана актуальность применения математического моделирования при решении проблемы обеспечения электромагнитной совместимости различных технических средств. Обсуждаются общие вопросы, связанные с интегральными и дифференциальными уравнениями. Рассмотрены особенности использования численных методов конечных разностей, моментов и элементов при решении задач электростатики. Особое внимание уделено способам повышения точности вычислений и экономии машинных ресурсов. Изложены методы решения систем линейных алгебраических уравнений и тенденции их развития. Приведены примеры решения тестовых задач, в том числе с использованием пакета GNU Octave. Для закрепления пройденного материала и самопроверки приведены контрольные вопросы и задания.

Для студентов высших учебных заведений, обучающихся по техническим направлениям подготовки и специальностям.

УДК 621.391.823(075.8)
ББК 32.841.174я73

ISBN 978-5-86889-879-2

Предисловие

The world may be utterly crazy
And life may be labour in vain;
But I'd rather be silly than lazy,
And would not quit life for its pain.
James Clerk Maxwell

Электромагнитное взаимодействие не только объясняет все электрические и магнитные явления, но и обеспечивает силы, благодаря которым вещество на атомном и молекулярном уровне существует как целое. Изучение электромагнитных явлений рассматривается в теории электромагнитного поля, описывающей взаимодействие между электрическими зарядами с помощью уравнений Максвелла (в дифференциальной или интегральной форме), связывающих источники (заряды и токи) с создаваемыми ими электромагнитными полями и потоками.

Аналитические решения в замкнутом виде известны только в ограниченном количестве частных случаев, которые крайне редко применимы к решению практических задач. Поэтому для преодоления разрыва между теорией и требованиями практики при решении реальных задач используются различного рода (более или менее грубые) упрощения или приближения, например квазистатический подход.

Появление компьютеров сильно изменило акценты при решении уравнений Максвелла. Так, до появления компьютеров было выгодно прикладывать значительные усилия для предотвращения громоздких вычислений, часто ценой длительных аналитических манипуляций и в итоге значительно уменьшенной применимости. Наоборот, с появлением мощных компьютеров привлекательным стало использование более простых методов, требующих больших вычислений. Такие вычислительные методы применимы при решении различного рода задач без необходимости модификации алгоритмов или компьютерных программ.

Сказанное выше послужило развитию такого направления, как вычислительная электродинамика. Это направление важно для современных инженеров и ученых, которые решают электромаг-

нитные задачи с помощью специализированных инструментальных средств. Такие средства позволяют ускорить и удешевить процесс проектирования, где использование дорогостоящих и трудоемких прототипов (физическое моделирование) сведено к минимуму. Эти инструменты могут дать важную информацию об электромагнитных процессах, протекающих в проектируемом устройстве, получение которой осложнено или даже невозможно посредством экспериментов или аналитических расчетов. Автоматизация вычислений позволяет провести обширные структурные и параметрические исследования, а при необходимости быстрой разработки, анализа и оптимизации проектируемых устройств решающее значение для поддержания их конкурентоспособности может иметь использование инструментария вычислительной электродинамики. Таким образом, средства вычислительной электродинамики являются базовым инструментарием, который необходим современным техническим специалистам в повседневной трудовой деятельности при решении задач электромагнитной совместимости в целом и электростатики в частности.

Автор благодарен аспирантам Иванову А. А. и Квасникову А. А. за помощь в подготовке материалов пособия, а заведующему кафедрой телевидения и управления Газизову Т. Р. за ряд ценных замечаний при обсуждении материалов.

Список сокращений

КЛБФ – кусочно-линейная базисная функция

КПБФ – кусочно-постоянная базисная функция

КСБФ – кусочно-синусоидальная базисная функция

КЭ – конечный элемент

МВН – метод взвешенных невязок

МКР – метод конечных разностей

МКЭ – метод конечных элементов

МоМ – метод моментов

МПЛ – микрополосковая линия

МПЛП – многопроводная линия передачи

РЭС – радиоэлектронное средство

САПР – система автоматизированного проектирования

СВЧ – сверхвысокие частоты

СЛАУ – система линейных алгебраических уравнений

ЭМС – электромагнитная совместимость

ACA – adaptive cross approximation, адаптивная перекрестная аппроксимация

AINV – approximate inverses, приближенный обратный

BiCG – biconjugate gradient method, метод бисопряженных градиентов

BiCGStab – biconjugate gradient stabilized method, метод стабилизированных бисопряженных градиентов

CGS – conjugate gradient squared method, метод сопряженных квадратичных градиентов

CSC – compressed column storage, разреженный столбцовый формат

CSR – compressed row storage, разреженный строчный формат

FDM – finite difference method, метод конечных разностей

FEM – finite element method, метод конечных элементов

GMRES – generalized minimal residual method, метод обобщенной минимальной невязки

ILU – incomplete LU factorization, неполное LU-разложение

ILUT – incomplete LU factorization with treshold, неполное LU-разложение с порогом

MoM – method of moments, метод моментов

SAINV – sparse approximate inverse, разреженный приближенно обратный

SPAI – sparse approximate inverse, разреженный приближенный обратный

TEM – transverse electromagmetic, поперечная электромагнитная

1 ЭЛЕКТРОМАГНИТНАЯ СОВМЕСТИМОСТЬ И ЭЛЕКТРОСТАТИКА: ОБЩИЕ СВЕДЕНИЯ

1.1 Электромагнитная совместимость

Открытие электромагнитных явлений и последующее изобретение электромагнитного телеграфа (П. Л. Шиллинг, 1832) и радио (А. С. Попов, 1895) фактически стало началом создания радиоэлектронных средств (РЭС) и тем самым глобального процесса информатизации, а также борьбы с радиопомехами, сначала с непреднамеренными (атмосферными и индустриальными), а затем и с преднамеренными. Достижения в области радиотехники и электроники, а также в вычислительных, информационных, телекоммуникационных и других технологиях послужили широкому внедрению во все сферы современного общества различных РЭС.

Под РЭС понимают техническое средство, состоящее из одного или нескольких радиоприемных и (или) радиопередающих устройств и вспомогательного оборудования. В более общем смысле под РЭС подразумевается изделие и его составные части, в основу функционирования которых положены принципы радиотехники и электроники. Структура и состав этих средств могут сильно варьировать в зависимости от их функционального назначения.

Конкуренция производителей РЭС требует регулярного и быстрого появления с минимальными затратами все более совершенных их видов. Однако выполнение этого требования с ростом сложности РЭС становится невозможным без применения автоматизированного проектирования, в основе которого лежит компьютерное моделирование. Поэтому наличие эффективной системы автоматизированного проектирования (САПР) с возможностью моделирования особенно важно для плодотворной работы современного специалиста, а ее использование позволяет существенно снизить затраты времени на разработку и повысить качество конечного изделия, сделав его более рентабельным. Помимо прочего, использование САПР позволяет экономить временные и финансовые ресурсы, требуемые для разработки, оценить правильность принятых технических решений, учесть требования

электромагнитной совместимости (ЭМС) и возможные дестабилизирующие факторы, влияющие на работу конечного изделия [1] .

Основными элементами РЭС являются антенны и СВЧ-устройства (прежде всего линии передачи (волноводы) и резонаторы) [2]. Из-за конструктивных особенностей последних для повышения эффективности их проектирования часто используется квазистатический подход. Он применим, когда поперечные размеры рассматриваемой структуры малы по сравнению с длиной распространяющейся электромагнитной волны. Это позволяет свести уравнения Максвелла к телеграфным и тем самым уменьшить вычислительные затраты. Данный подход получил широкое распространение при проектировании на основе анализа многопроводных линий передачи (МПЛП). Особенностью такого проектирования является учет распределенных параметров между всеми проводниками. На основе МПЛП моделируются различные реальные полосковые структуры, которые широко используются для создания элементов РЭС: печатных плат, фильтров, средств снижения уровня перекрестных помех, антенн и др.

Помимо сугубо конструкторских аспектов проектирования, тенденции развития современных РЭС обостряют проблему электромагнитной совместимости, появившуюся со времен первых радиопередатчиков А. С. Попова. Согласно ГОСТ Р 50397-2011 ЭМС технического средства – это его способность функционировать с заданным качеством в заданной электромагнитной обстановке и не создавать недопустимых электромагнитных помех другим техническим средствам. При этом под техническим средством подразумевается электротехническое, электронное и радиоэлектронное изделие, а также любое изделие, содержащее электрические и/или электронные составные части (оно может быть устройством, оборудованием, системой или установкой). Конструктивное усложнение РЭС и ужесточение требований ЭМС, обусловленное ростом верхних частот полезных и помеховых сигналов, плотности монтажа, а также возможностей генераторов преднамеренных электромагнитных воздействий, в совокупности с необходимостью учета межэлементных, межблочных и межсистемных взаимовлияний требует все более тщательного проектиро-

вания РЭС. Например, при обеспечении ЭМС антенн важен контроль коэффициента стоячей волны и диаграммы направленности не только в рабочем диапазоне частот, но и в намного более широком диапазоне частот помеховых сигналов.

Классическими способами обеспечения ЭМС являются фильтрация, экранирование и заземление, тесно связанные между собой. Для помехозащиты традиционно используется установка на входе защищаемого изделия устройств на основе сосредоточенных компонентов (в виде сборок из *RLC*-цепей, варисторов, разрядников, TVS-диодов и др.). Еще одним, сравнительно новым способом защиты является разложение помехового сигнала большой амплитуды на серию импульсов меньшей амплитуды, представляющую значительно меньшую опасность для РЭС по сравнению с исходным помеховым сигналом. Это разложение возможно за счет «полезного» использования взаимных связей в линиях передачи. При этом линиями передачи применительно к РЭС могут выступать межблочные кабели, печатные дорожки и другие монтажные соединения (межсоединения). Эти соединения, помимо электрических характеристик, отличаются по важным для ЭМС показателям: волновому сопротивлению, скорости распространения электромагнитной волны, эффективности экранирования и т. д. Следовательно, при проектировании таких соединений необходимо тщательно учитывать требования ЭМС для получения конечного изделия, удовлетворяющего этим требования на всем протяжении жизненного цикла самого соединения и всего изделия.

Перекрестные наводки в линиях передачи представляют собой электромагнитные помехи, обусловленные близостью расположения проводников линии и других компонентов РЭС. Их необходимо учитывать при проектировании, в том числе за счет контроля взаимовлияний между всеми проводниками. При квазистатическом анализе это реализуется посредством вычисления матриц погонных параметров линии передачи. Эти матрицы интегрально содержат всю необходимую информацию для последующего анализа, в том числе целостности сигналов и питания.

При проектировании элементов РЭС с учетом ЭМС необходимо также учитывать частотную зависимость их параметров, а также потери в проводниках и диэлектриках. Вследствие расширения спектра полезных и помеховых сигналов количество повторных вычислений существенно возрастает. Для обеспечения ЭМС целесообразен выбор рационального расположения проводников, например с целью экранирования одних проводников другими. При этом необходимо контролировать волновое сопротивление для обеспечения основных функций проектируемой линии передачи. Как следует из вышеизложенного, проектирование линий передачи посредством многовариантного анализа или оптимизации является нетривиальной задачей.

Последним, но не менее важным аспектом обеспечения ЭМС при проектировании является выбор соответствующей схемы заземления. Так, наиболее благоприятная организация сплошных полигонов земли, как правило, является экономически невыгодной и технологически невыполнимой. Например, необходимость переходных отверстий на печатных платах нарушает целостность системы заземления, следовательно, необходим поиск оптимальных технических решений ее организации. Кроме того, особенности заземления оказывают существенное влияние на внутреннюю электромагнитную обстановку внутри проектируемого РЭС. Так, способ заземления экранирующего проводника на печатной плате существенно влияет на перекрестные наводки, что сказывается на обеспечении ЭМС РЭС в целом.

Поиск оптимального решения задачи проектирования РЭС приводит к необходимости применения параметрического синтеза, осуществляемого средствами многовариантного анализа в диапазоне параметров или оптимизации. При этом из-за сложности проектируемых РЭС и необходимости учета требований ЭМС анализ и оптимизация невозможны без применения методов вычислительной электродинамики (одним из которых является метод моментов), поскольку имеющиеся аналитические выражения пригодны только для простых структур (например, одиночных и связанных линий передачи). В основе этих методов лежит замена непрерывных функций их дискретными аналогами (построение сетки), что

часто сводит задачу к решению системы линейных алгебраических уравнений (СЛАУ). С учетом требований ЭМС возрастает порядок (определяется сложностью моделируемого объекта и окружающего его пространства) и количество решаемых систем уравнений (определяется верхней частотой помехового сигнала, количеством и диапазоном оптимизируемых параметров и др.). Это резко увеличивает вычислительные затраты, что становится главной преградой для эффективного проектирования РЭС.

1.2 Автоматизированное проектирование

Появление первых САПР датируется концом 60-х гг. XX в., а конец 80-х гг. ознаменовался началом их интенсивного развития, и прежде всего в части электромагнитного анализа. Сегодня эти системы находятся в стадии расцвета. История развития отечественных и зарубежных САПР СВЧ-устройств до 2010 г. достаточно полно отражена в [3]. Следует отметить, что отечественные САПР, к сожалению, до сих пор не так распространены, как зарубежные. При этом их развитие основано в большей степени на энтузиазме разработчиков.

Для численного анализа какой-либо физической задачи необходимо построить ее математическую модель, учитывающую существенные для данной задачи особенности реального объекта (процесса или явления). Математическая модель не идентична исследуемому объекту, а является его приближенным описанием с помощью языка математики и реализуемых на компьютере алгоритмов. В зависимости от универсальности, адекватности, точности и экономичности модели могут различаться по их сложности и требованиям к вычислительным ресурсам.

Процесс построения математической модели для анализа электромагнитных задач формально можно представить в виде нескольких взаимосвязанных этапов [4] (в скобках указаны возможные варианты).

1. Постановка задачи. Определение целей расчета, класса решаемой задачи, необходимой для этого входной и выходной информации, а также требуемой погрешности результатов.

2. Аналитическая обработка. Формулировка уравнений (уравнения Максвелла в частотной или временной области), условий (начальных, граничных), описание формы (геометрические параметры) и свойств (электрофизические параметры) расчетной области, выбор метода решения (аналитический, численный), при необходимости преобразование уравнений к виду, наиболее пригодному для выбранного метода (Пуассона, волновое).

3. Дискретизация модели (построение сетки). Переход от функциональных уравнений к СЛАУ (с плотной или разреженной матрицей) с помощью замены непрерывных функций их дискретными аналогами.

4. Решение СЛАУ. На этом этапе важную роль играет выбор как типа (прямой, итерационный), так и самого метода решения (LU-разложение, Якоби), наиболее подходящего для сформированной СЛАУ и обеспечивающего требуемую точность.

5. Обработка результатов. Вычисление из решения СЛАУ требуемых значений характеристик и параметров (напряженность электрического поля, поверхностный заряд) исследуемого объекта и при необходимости их визуализация. Формирование общего решения задачи (частотный/временной отклик на заданное воздействие, эффективность экранирования).

Подчеркнем, что данная градация условна и только облегчает процесс систематизации информации. Так, например, при использовании явной схемы решения дифференциальных уравнений методом конечных разностей не требуется формирование СЛАУ, а вычисления основаны на итерационном уточнении компонентов решения. Тогда этапы 3 и 4 могут быть реорганизованы в один, на котором происходит явное решение сформированного на этапе 2 уравнения. Для примера на рисунке 1.1 показана последовательность построения математической модели (workflow) и передачи данных в системе Altair FEKO для получения результатов моделирования [5]. Видно, что в данном случае выделено 7 этапов, а не 5, как показано выше. При этом функционально они идентичны.

Очевидно, что указанные этапы взаимосвязаны. Так, выбор метода построения сетки влияет на свойства формируемой СЛАУ, что сказывается на выборе метода ее решения (прямой или итера-

ционный) и тем самым на его времени. Одной из самых важных характеристик используемой математической модели является погрешность требуемых результатов.

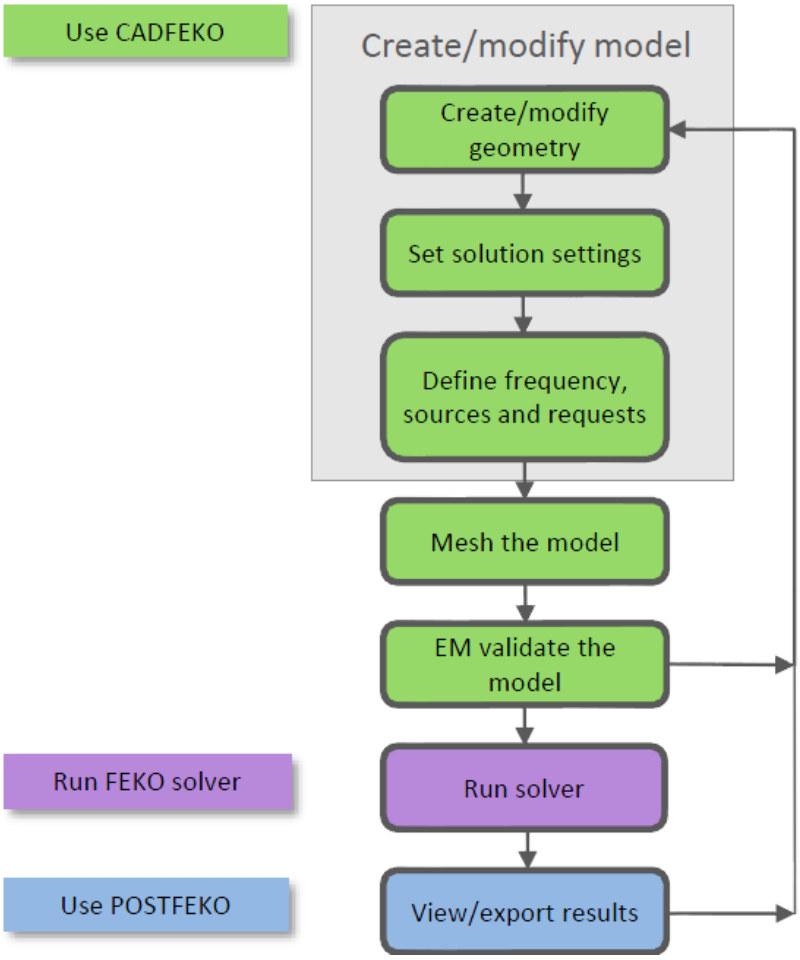


Рисунок 1.1 – Последовательность работ
в системе Altair FEKO

Она складывается из нескольких составляющих, вносимых на каждом этапе: погрешностей задания исходных данных на этапе 1, самой модели (неполное соответствие реальному объекту), дискретизации (грубая сетка, погрешности численного интегрирования) и вычислительной (округление при оперировании с числами с конечной точностью на этапах 4 и 5). При решении СЛАУ основополагающим является наличие погрешности в задании элементов матрицы и правой части.

На этапе постановки задачи с учетом имеющихся данных и требований к точности необходимо определить, какой подход будет использован: схемотехнический, квазистатический или

электродинамический. В их рамках применяются эвристические, аналитические, численные и комбинированные (комплексные) методы [6, 7]. Схемотехнический подход основан на законах Кирхгофа, а моделирование с его помощью выполняется с использованием принципиальных электрических схем (SPICE-моделирование). Электродинамический подход, основанный на решении уравнений Максвелла, является универсальным и позволяет решать задачи с произвольной геометрией, однако при этом требования к вычислительным ресурсам могут оказаться чрезвычайно высокими. Промежуточное положение занимает квазистатический подход, основанный на допущениях, что поперечные размеры рассматриваемой системы много меньше длин волн распространяющихся в ней сигналов, что позволяет рассматривать наличие только основной Т-волны (ТЕМ-волны), не рассматривая волны высшего типа. Распределения полей вычисляются из электро- и магнитостатических задач, требующих решения уравнения Пуассона – Лапласа. Этот подход получил широкое распространение при моделировании линий передачи с помощью телеграфных и производных от них уравнений [8].

1.3 Уравнения Максвелла

Схемотехническое моделирование используют, когда геометрические размеры моделируемого объекта малы по сравнению с длиной волны из рассматриваемого диапазона частот. При этом полагается, что уровни электромагнитного излучения пренебрежимо малы, и можно использовать анализ, основанный на схемах из сосредоточенных элементов. Когда геометрические размеры объекта сопоставимы с длинами волн рассматриваемого диапазона, проводники, кабели, соединители, переходные отверстия и прочие объекты начинают действовать подобно антеннам, излучающим или принимающим электромагнитную энергию. Поэтому схемотехнический подход становится неприемлемым и необходимо применять электродинамический подход (или в частных случаях квазистатический). Данный подход основан на решении уравнений Максвелла – системе из 4 уравнений (современная

трактовка) в дифференциальной или интегральной форме, связывающих электромагнитное поле (**D**, **E**, **B**, **H**) с распределениями тока и заряда (**J**, ρ) и характеристиками заполняющей среды объекта (ε, μ). Уравнения Максвелла приведены в таблице 1.1, где ∇ – оператор набла; **E** – вектор напряженности электрического поля (В/м); **D** – вектор электрической индукции или электрического смещения (Кл/м^2); **H** – вектор напряженности магнитного поля (А/м); **B** – вектор магнитной индукции (Тл); ρ – объемная плотность стороннего электрического заряда (Кл/м^3); **J** – вектор плотности электрического тока (А/м^2); ε – абсолютная диэлектрическая проницаемость (Ф/м); μ – абсолютная магнитная проницаемость (Гн/м).

Таблица 1.1 – Уравнения Максвелла во временной области

Дифференциальная форма	Интегральная форма	Название
$\nabla\times\mathbf{H} = \mathrm{rot}\mathbf{H} = \mathbf{J} + \partial\mathbf{D}/\partial t$	$\oint_L \mathbf{H}dl = \int_S \left(\mathbf{J} + \partial\mathbf{D}/\partial t\right)d\mathbf{S}$	Теорема о циркуляции магнитного поля
$\nabla\times\mathbf{E} = \mathrm{rot}\mathbf{E} = -\partial\mathbf{B}/\partial t$	$\oint_L \mathbf{E}dl = -\int_S \frac{\partial\mathbf{B}}{\partial t}d\mathbf{S}$	Закон индукции Фарадея
$\nabla\cdot\mathbf{D} = \mathrm{div}\mathbf{D} = \rho$	$\oint_S \mathbf{E}d\mathbf{S} = \int_v \rho\, dv$	Закон Гаусса
$\nabla\cdot\mathbf{B} = \mathrm{div}\mathbf{B} = 0$	$\oint_S \mathbf{B}d\mathbf{S} = 0$	Закон Гаусса для магнитного поля

Оператор набла в декартовой системе координат определяется следующим образом:

$$\nabla = \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z},$$

где **i**, **j**, **k** – единичные векторы по осям *x*, *y* и *z* соответственно. В дополнение к приведенным четырем уравнениям используются три материальных уравнения:

$$\mathbf{D} = \varepsilon\mathbf{E};$$

$$\mathbf{B} = \mu\mathbf{H};$$

$$\mathbf{J} = \gamma \mathbf{E},$$

где γ – удельная проводимость среды (См/м).

При изучении большинства физических процессов зачастую свойства объекта исследования описываются функциями не одной, а нескольких переменных величин. Поэтому при поиске количественного описания физического явления обычно решается система дифференциальных уравнений с частными производными. Аргументами неизвестных функций таких систем уравнений являются пространственные переменные и время. Тогда дифференциальные уравнения с частными производными, описывающие реальные физические модели, называются уравнениями математической физики, а изучающая их наука – математической физикой.

Большое число задач, связанных с анализом физических (и не только физических) полей, описывается дифференциальными уравнениями в частных производных. К сожалению, во многих случаях, представляющих практический интерес, найти аналитическое решение таких задач трудно или невозможно. Это обычно обусловлено сложной формой или неоднородностью свойств области, в которой отыскивается решение. Однако результат можно получить численно с помощью компьютера. Подходы к решению дифференциальных уравнений в частных производных определяются их математической формой. Поэтому рассмотрим сначала некоторые полезные для дальнейшего изложения понятия, а затем классификацию уравнений в частных производных и методов их решения.

1.4 Дифференциальные уравнения в частных производных

Во многих случаях для описания физических процессов используют уравнения с частными производными до второго порядка включительно. Так, например, свободные колебания различной природы представляются волновыми уравнениями вида

$$\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)-\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2}=0, \qquad (1.1)$$

где $u(x, y, z, t)$ – функция, описывающая волновой процесс; x, y, z – координаты; c – скорость распространения волны в данной среде; t – время. Выражение $\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)$ носит название оператора Лапласа и обозначается ∇^2.

Процессы распространения тепловой энергии описываются уравнением теплопроводности

$$\rho C\frac{\partial T}{\partial t}-k\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)=Q, \qquad (1.2)$$

где ρ и C – плотность и теплоемкость вещества соответственно; T – температура; k – коэффициент теплопроводности; Q – мощность источников тепла.

Анализ стационарных состояний, например статических тепловых, электрических, магнитных полей или деформаций при статических нагрузках, проводят, используя уравнение Пуассона

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}=-f(x, y, z), \qquad (1.3)$$

где $u(x, y, z)$ – функция, описывающая статическое поле; $f(x, y, z)$ – распределенные источники. Если $f(x, y, z) = 0$, то выражение (1.3) обращается в уравнение Лапласа

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}=0. \qquad (1.4)$$

Известны и другие виды задач и соответствующие им дифференциальные уравнения в частных производных, например уравнение диффузии или уравнение Гельмгольца.

Несмотря на различие процессов, описываемых рассмотренными уравнениями, и форм их записи, все они с математической точки зрения могут быть представлены как частные случаи обобщенной формы дифференциального уравнения второго порядка.

Рассмотрим уравнение второго порядка с двумя независимыми переменными x и y:

$$A\frac{\partial^2 u}{\partial x^2} + 2B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D = 0, \qquad (1.5)$$

где A, B, C и D – некоторые функции, зависящие в общем случае от x, y, u, $\partial u/\partial x$ и $\partial u/\partial y$, причем A, B и C одновременно не обращаются в ноль. Дифференциальные уравнения, описывающие физические поля, могут быть нелинейными. Однако на практике многие задачи рассматриваются в линейном приближении, когда уравнение с частными производными линейно относительно неизвестной функции u и ее частных производных.

На основании того что уравнению (1.5) можно поставить в соответствие квадратичную форму $A\zeta_1^2 + B\zeta_1\zeta_2 + C\zeta_2^2 = 0$, по математической природе различают следующие типы уравнений:

– гиперболические, если $B^2 - 4AC > 0$, их аналогом является волновое уравнение (1.1);

– параболические, если $B^2 - 4AC = 0$, их аналог уравнение теплопроводности (1.2);

– эллиптические, если $B^2 - 4AC < 0$, их аналог уравнение Пуассона (1.3) или Лапласа (1.4).

Приведенная классификация позволяет определить общие подходы к решению дифференциальных уравнений в задачах, различных по физической сути, но сходных с математической точки зрения. В задачах, описываемых дифференциальными уравнениями в частных производных, важной составляющей, помимо самого уравнения, является формулировка дополнительных условий. Для задач с уравнениями гиперболического или параболического типа, содержащих в качестве независимой переменной время t, условия по времени обычно формулируются как начальные, описывающие исходное состояние системы. По координатам x, y и z задают граничные условия. В тепловых задачах они, например, описывают распределение температуры на границе расчетной области. В задачах с уравнениями эллиптического типа, не содержащими переменную t, используют только граничные условия по координатам x, y и z, а саму задачу называют краевой.

Если краевое условие задает распределение функции *u* на границе, то его принято называть условием Дирихле. Условие, определяющее производную на границе расчетной области, называют условием Неймана. Условия, представляющие собой комбинацию двух вышеназванных, называют смешанными.

1.5 Интегральные уравнения

Под интегральными понимают уравнения, в которых неизвестная функция Ф независимого (скалярного или векторного) аргумента встречается под знаком интеграла. Термин «интегральное уравнение» ввел в употребление в 1886 г. немецкий математик Поль Дюбуа-Рейман. Различают линейные и нелинейные интегральные уравнения в соответствии с тем, зависит уравнение от неизвестной функции линейным или нелинейным образом. Построение общей теории линейных интегральных уравнений было начато в конце XIX в. Ее основоположниками считаются Вито Вольтерра, Эрик Эвир Фредгольм, Давид Гильберт и Эрхард Шмидт. Простыми примерами интегральных уравнений являются интегральные преобразования Фурье, Лапласа и Ганкеля. Так, одной из первых была задача обращения интеграла

$$F(u) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} f(x) e^{jux} dx,$$

т. е. по известной функции $F(u)$ требуется найти функцию $f(x)$. Как известно, первым решил эту задачу в 1811 г. Ж. Фурье. Функция $f(x)$ определяется следующим образом:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(u) e^{-jux} du.$$

Последние два выражения называются прямым и обратным преобразованием Фурье соответственно.

Линейные интегральные уравнения, которые наиболее часто подлежат рассмотрению, делятся на два класса, названные в честь Фредгольма и Вольтерра. (Наиболее распространенными представителями нелинейных интегральных уравнений являются

уравнения Урысона и уравнения Гаммерштейна. Данные уравнения далее не рассматриваются.) Первый класс – уравнения Фредгольма первого, второго и третьего рода соответственно:

$$f(x) = \int_a^b K(x, t)\Phi(t)dt; \tag{1.6}$$

$$f(x) = \Phi(x) - \lambda\int_a^b K(x, t)\Phi(t)dt; \tag{1.7}$$

$$f(x) = a(x)\Phi(x) - \lambda\int_a^b K(x, t)\Phi(t)dt, \tag{1.8}$$

где $a \le t \le b$, $a \le x \le b$; λ – в общем случае комплексный параметр (в некоторых задачах равен единице). Функция $K(x, t)$, называемая ядром интегрального уравнения, и функция $f(x)$, а также пределы интегрирования a и b известны, а функция $\Phi(x)$ неизвестна. Область $S = [a, b] \times [a, b]$ изменения переменных x и t называется основным квадратом. Промежуток $[a, b]$, на котором ищется функция $\Phi(x)$, называется областью определения интегрального уравнения.

Второй класс – уравнения Вольтерра первого, второго и третьего рода соответственно:

$$f(x) = \int_a^x K(x, t)\Phi(t)dt; \tag{1.9}$$

$$f(x) = \Phi(x) - \lambda\int_a^x K(x, t)\Phi(t)dt; \tag{1.10}$$

$$f(x) = a(x)\Phi(x) - \lambda\int_a^x K(x, t)\Phi(t)dt \tag{1.11}$$

с переменным верхним пределом интегрирования. При этом $K(x, t) = 0$ при $t > x$.

Если $f(x) = 0$, то уравнения (1.6)–(1.11) являются однородными, в противном случае – неоднородными. Еще раз отметим, что

уравнения (1.6)–(1.11) являются линейными, поскольку неизвестная функция линейна. Так, уравнение вида

$$f(x) = \Phi(x) - \int_a^b K(x, t)\Phi^2(t)dt$$

является нелинейным. Если пределы интегрирования a или b или ядро $K(x, t)$ обращаются в бесконечность, то интегральное уравнение называют сингулярным. Если выполняется условие $K(x, t) = K(t, x)$, то ядро называется симметричным.

Вышеприведенная классификация одномерных интегральных уравнений возникает в теории дифференциальных уравнений естественным образом, демонстрируя тесную связь между интегральной и дифференциальной постановками задачи. Большинство обыкновенных дифференциальных уравнений может быть сведено к интегральным, но обратное неверно. Граничные условия вводятся отдельно от дифференциальных уравнений, а при интегральном уравнении они включены в него. Например, рассмотрим обыкновенное дифференциальное уравнение первого порядка

$$\frac{d\Phi}{dx} = F(x, \Phi), \quad a \le x \le b, \tag{1.12}$$

при условии $\Phi(a) =$ константа. Оно может быть записано в виде уравнения Вольтерра второго рода. Так, после интегрирования уравнения (1.12) получим

$$\Phi(x) = \int_a^x F(x, \Phi(t))dt + c_1,$$

где $c_1 = \Phi(a)$. В итоге вместо уравнения (1.12) получим

$$\Phi(x) = \Phi(a) + \int_a^x F\big(x, \Phi(t)\big)dt. \tag{1.13}$$

Любое решение уравнения (1.13) удовлетворяет исходному уравнению (1.12) и заданным граничным условиям. Таким образом, интегральная формулировка уравнения включает как само дифференциальное уравнение, так и соответствующие граничные условия.

Рассмотрим обыкновенное дифференциальное уравнение второго порядка

$$\frac{d^2\Phi}{dx^2} = F(x, \Phi), \;\; a \le x \le b. \tag{1.14}$$

Его интегрирование дает

$$\frac{d\Phi}{dx} = \int_a^x F(x, \Phi(t))dt + c_1,$$

где $c_1 = \Phi'(a)$. Интегрирование полученного выражения дает

$$\Phi(x) = c_2 + c_1 x + \int_a^x (x-t)F\big(x, \Phi(t)\big)dt,$$

где $c_2 = \Phi(a) - a\Phi'(a)$. Тогда

$$\Phi(x) = \Phi(a) + (x-a)\Phi'(a) + \int_a^x (x-t)F\big(x, \Phi(t)\big)dt. \tag{1.15}$$

Видно, как и ранее, что интегральное уравнение (1.15) обобщает собой как дифференциальное уравнение (1.14), так и граничные условия.

Проиллюстрируем сказанное примерами.

Пример 1.1

Решить интегральное уравнение Вольтерра $\Phi(x) = 1 + \int_a^x \Phi(t)dt$.

Решение

Решить уравнение можно напрямую или косвенно, найдя решение соответствующего дифференциального уравнения. Для прямого решения дифференцируем обе части заданного уравнения. В общем виде заданный интеграл имеет вид $g(x) = \int_{\alpha(x)}^{\beta(x)} f(x,t)dt$ с переменными пределами интегрирования. Дифференцируем его с помощью правила Лейбница (в интегральном исчислении называется правилом дифференцирования функ-

ции под знаком интеграла, зависящего от параметра, пределы которого зависят от переменной дифференцирования):

$$g'(x) = \int_{\alpha(x)}^{\beta(x)} \frac{\partial f(x,t)}{\partial x} dt + f(x,\beta)\beta' - f(x,\alpha)\alpha'. \qquad (1.16)$$

В результате получим

$$\frac{d\Phi}{dx} = \Phi(x) \quad \text{или} \quad \frac{d\Phi}{\Phi} = dx. \qquad (1.17)$$

Проинтегрировав выражение (1.17), получим $\ln \Phi = x + \ln c_0$ или $\Phi = c_0 e^x$, где $\ln c_0$ – постоянная интегрирования. Согласно решаемому интегральному уравнению $\Phi(0) = 1 = c_0$. Тогда $\Phi(x) = e^x$ будет требуемое решение. Это можно проверить путем его подстановки в заданное уравнение.

Косвенный способ решения заключается в сравнении решаемого интегрального уравнения и уравнения (1.13). Так, $a = 0$, $\Phi(a) = \Phi(0) = 1$, а $F(x, \Phi) = \Phi(x)$. Следовательно, соответствующее дифференциальное уравнение первого порядка имеет вид $\frac{d\Phi}{dx} = \Phi$, $\Phi(0) = 1$, т. е. совпадает с уравнением (1.17), и его решение соответственно $\Phi(x) = e^x$.

Пример 1.2

Составить интегральное уравнение, соответствующее дифференциальному уравнению $\frac{d^2\Phi}{dx^2} - \frac{d\Phi}{dx}\sin x + e^x \Phi = x$ и начальным условиям $\Phi(0) = 0$, $\left.\frac{d\Phi}{dx}\right|_{x=0} = 1$.

Решение

Пусть $\frac{d^2\Phi}{dx^2} = \varphi(x)$, тогда $\frac{d\Phi}{dx} = \int_0^x \varphi(t)dt + \left.\frac{\partial \Phi}{\partial x}\right|_{x=0} = \int_0^x \varphi(x)dt + 1$,

а $\Phi = \int_0^x (x-t)\varphi(t)\,dt + x$. Подставив эти выражения в заданное

уравнение, получим интегральное уравнение Вольтерра второго рода $\varphi(x) = \int_0^x \left(\sin x - (x-t)e^x\right)\varphi(t)\,dt + \sin x + \left(1 - e^x\right)x$.

Пример 1.3

Составить интегральное уравнение, которое соответствует дифференциальному уравнению $\Phi''' - 3\Phi'' - 6\Phi' + 8\Phi = 0$ при условии $\Phi''(0) = \Phi'(0) = \Phi(0) = 1$.

Решение

Пусть $\Phi''' = F(\Phi, \Phi, \varphi, x) = 3\Phi'' + 6\Phi' - 8\Phi$. Интегрирование обеих частей уравнения дает $\Phi'' = 3\Phi' + 6\Phi - 8\int_0^x \Phi(t)dt + c_1$, где коэффициент c_1 определен с помощью начальных значений, т. е. $1 = 3 + 6 + c_1$, следовательно, $c_1 = -8$. Проинтегрировав выражение, получим

$$\Phi' = 3\Phi + 6\int_0^x \Phi(t)dt - 8\int_0^x (x-t)\Phi(t)dt - 8x + c_2,$$

где $c_2 = -2$ $(1 = 3 + c_2)$. Окончательно проинтегрировав обе части выражения, найдем

$$\Phi(x) = 3\int_0^x \Phi(t)dt + 6\int_0^x (x-t)\Phi(t)dt - 4\int_0^x (x-t)^2\Phi(t)dt - 4x^2 - 2x + c_3,$$

где $c_3 = 1$. Таким образом, для данного дифференциального уравнения получено эквивалентное интегральное уравнение

$$\Phi(x) = 1 - 2x - 4x^2 + \int_0^x \left[3 + 6(x-t) - 4(x-t)^2\right]\Phi(t)dt.$$

Пример 1.4

Свести дифференциальное уравнение

$$\frac{d^2\Phi}{dx^2} + a_1(x)\frac{d\Phi}{dx} + a_2(x)\Phi = F(x),\ \Phi(0) = c_0,\ \Phi'(0) = c_1$$

к интегральному уравнению Вольтерра 2-го порядка [9].

Решение

Положим $\dfrac{d^2\Phi}{dx^2} = \varphi(x)$. Тогда с учетом начальных условий последовательно найдем

$$\frac{d\Phi}{dx} = \int_0^x \Phi(t)dt + c_1, \quad \Phi = \int_0^x (x-t)\Phi(t)dt + c_1 x + c_0.$$

При этом использована формула

$$\underbrace{\int_{x_0}^{x} dx \int_{x_0}^{x} dx \ldots \int_{x_0}^{x} f(x)dx}_{n} = \frac{1}{(n-1)!}\int_{x_0}^{x} (x-z)^{n-1} f(z)dz\,.$$

Подставив полученные выражения в исходное уравнение, запишем

$$\Phi(x) + \int_0^x a_1\Phi(t)dt + c_1 a_1(x) +$$

$$+\int_0^x a_2(x)(x-t)\Phi(t)dt + c_1 x a_2(x) + c_0 a_2(x) = F(x)$$

или

$$\Phi(x) + \int_0^x \left[a_1(x) + a_2(x)(x-t)\right]\Phi(t)dt =$$

$$= F(x) - c_1 a_1(x) - c_1 x a_2(x) - c_0 a_2(x).$$

Полагая

$$K(x,t) = -\left[a_1(x) + a_2(x)(x-t)\right],$$

$$f(x) = F(x) - c_1 a_1(x) - c_1 x a_2(x) - c_0 a_2(x),$$

найдем

$$\Phi(x) = \int_0^x K(x,t)\Phi(t)dt + f(x),$$

т. е. выражение, эквивалентное уравнению (1.10).

1.6 Уравнения электростатики

Электростатическое поле – это электрическое поле системы неподвижных относительно наблюдателя заряженных тел при отсутствии электрических токов. Если в системе нет намагниченных тел и магнитное поле отсутствует, то для электростатического поля справедливо $\mathbf{J} = 0$, $\mathbf{B} = 0$ и $\mathbf{H} = 0$. Наличие в электрическом поле свободных и распределенных в объеме зарядов привело бы к возникновению электрического тока, а поскольку $\mathbf{J} = 0$, то всюду $\rho = 0$. Поэтому существуют только заряды, распределенные по поверхностям заряженных тел.

С учетом сказанного из системы уравнений Максвелла вытекают следующие формулировки уравнений электростатики:

$$\operatorname{rot}\mathbf{E} = 0;\ \ \mathbf{D} = \varepsilon\mathbf{E};\ \ \operatorname{div}\mathbf{D} = 0. \qquad (1.18)$$

(Аналогично магнитостатика изучает не изменяющиеся во времени магнитные поля. Так, если положить $\partial/\partial t = 0$, из первого и четвертого уравнений Максвелла формулируется система уравнений магнитостатики: $\operatorname{rot}\mathbf{H} = \mathbf{J}$; $\operatorname{div}\mathbf{B} = 0$; $\mathbf{B} = \mu\mathbf{H}$.)

Условие $\operatorname{rot}\mathbf{E} = 0$ свидетельствует, что электростатическое поле имеет безвихревой характер. Тогда согласно теореме Стокса для любого замкнутого контура имеем

$$\oint_l \mathbf{E}d\mathbf{l} = \int_S \operatorname{rot}\mathbf{E}d\mathbf{S} = 0.$$

Отсюда следует, что в электростатическом поле линейный интеграл вектора **E**, взятый от точки A до точки B, не зависит от выбора пути интегрирования и полностью определяется положением этих точек в данном поле. С учетом этого формулируется понятие потенциала электростатического поля. Так, потенциалу электростатического поля в точке A соответствует линейный интеграл вектора **E**, взятый от точки A до точки P, в которой потенциал равен нулю, т. е.

$$\Phi_A = \int_A^P \mathbf{E}d\mathbf{l}.$$

Линейный интеграл вектора **E** вдоль некоторого пути от точки *A* до точки *B* есть разность электрических потенциалов в точках *A* и *B*, т. е.

$$\int_A^B \mathbf{E}d\mathbf{l} = \Phi_A - \Phi_B.$$

Пусть положение точки *A*, в которой рассматриваем потенциал Φ, определяется ее расстоянием *l* от начальной точки *O* вдоль некоторого пути, идущего в точку *P*, где потенциал принят равным нулю (рисунок 1.2) [10].

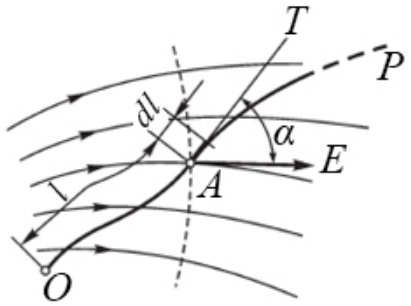


Рисунок 1.2 – К определению связи
между напряженностью и изменением потенциала

Выражение для потенциала при этом можно записать в виде

$$\Phi = \int_l^{l_P} \mathbf{E}d\mathbf{l} = \int_l^{l_P} \mathbf{E}\cos\alpha d\mathbf{l},$$

где l_P – длина всего пути от точки *O* до точки *P*; α – угол между направлением вектора **E** и касательной к пути. Взяв частную производную от обеих частей равенства по нижнему пределу, получим

$$\frac{\partial\Phi}{\partial l} = -E\cos\alpha,$$

откуда следует, что приращение потенциала, рассчитанное на единицу перемещения в каком-либо направлении, численно равно взятой с обратным знаком составляющей напряженности поля в этом направлении. Тогда в декартовых координатах получим

$$\frac{\partial\Phi}{\partial x} = -E_x, \quad \frac{\partial\Phi}{\partial y} = -E_y, \quad \frac{\partial\Phi}{\partial z} = -E_z.$$

Если направление перемещения *dl* составляет прямой угол ($\alpha = \pi/2$) с вектором **E**, то $\cos\alpha = 0$ и $\partial\Phi/\partial l = 0$. Следовательно, мысленно перемещаясь в направлении, нормальном к направлению линий напряженности поля, будем иметь $\Phi = \text{const}$, т. е. будем оставаться на поверхности равного потенциала. Линии напряженности поля нормальны к поверхностям равного потенциала. Таким образом, уравнение $\Phi(x, y, z) = \text{const}$ определяет линии эквипотенциальных поверхностей. Линии равного потенциала пересекаются с линиями напряженности поля всюду под прямым углом.

Совмещая направление перемещения *dl* с направлением вектора **E**, получим $\alpha = 0$, тогда

$$\cos\alpha = 0 \ \text{ и } \ \frac{\partial\Phi}{\partial l} = -E.$$

Это характерное направление совпадает с нормалью к поверхности равного потенциала. Условившись обозначать перемещение *dl* в этом направлении через *dn*, получим

$$\frac{\partial\Phi}{\partial n} = -E.$$

Очевидно, что *dn* является элементом длины линии напряженности поля, причем полагаем координату *n* растущей в направлении вектора **E**.

Производная от потенциала по координате имеет наибольшее значение в направлении, нормальном к поверхности равного потенциала и противоположном направлению вектора **E**. Это наибольшее значение производной может быть изображено вектором, направленным против вектора **E** и носящим название градиента электрического потенциала. Его обозначают символом $\text{grad}\Phi$ или $\nabla\Phi$.

Таким образом, градиент потенциала равен приращению потенциала, отнесенному к единице длины и взятому в направлении, в котором это приращение имеет наибольшее значение:

$$\left|\text{grad}\Phi\right| = \left|\frac{\partial\Phi}{\partial n}\right|.$$

Векторы **E** и gradФ равны между собой по величине и направлены в противоположные стороны, т. е.

$$\operatorname{grad}\Phi = -\mathbf{E}. \qquad (1.19)$$

Знак минус указывает на то, что потенциал убывает в направлении линий напряженности поля. Это является следствием определения потенциала как линейного интеграла напряженности электрического поля, взятого от рассматриваемой точки *A* до заданной точки *P*, в которой Ф = 0. Такое определение целесообразно, так как при этом потенциал положительно заряженного тела оказывается также положительным при условии, что потенциал бесконечно удаленных точек принимается равным нулю.

Все сказанное свидетельствует, что всякое безвихревое поле есть поле потенциальное, т.е. такое, которое может быть охарактеризовано потенциальной функцией Ф(*x*, *y*, *z*). Обратно, всякое потенциальное поле является безвихревым, что вытекает из тождества rotgradФ = 0.

Выражение потенциала точечного заряда для однородной среды позволяет сформулировать общий способ вычисления потенциала при заданном распределении в конечной области пространства электрических зарядов. Так, разделив все распределенные в пространстве заряды на элементарные части *dq*, рассмотрим эти элементы *dq* как точечные заряды (рисунок 1.3) [10].

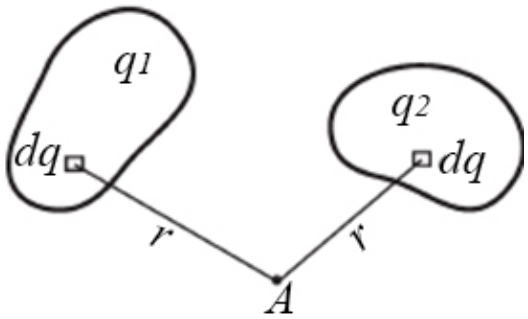


Рисунок 1.3 – К определению потенциала
по заданному распределению зарядов

Потенциал в точке *A*, определяемый каждым таким элементом, будет $d\Phi = \dfrac{dq}{4\pi\varepsilon r}$. Тогда потенциал, определяемый всей

совокупностью зарядов, распределенных в пространстве, находится как

$$\Phi = \int \frac{dq}{4\pi\varepsilon r} = \frac{1}{4\pi\varepsilon}\int \frac{dq}{r}.$$

Если электрический заряд распределен по объему V, причем объемная плотность заряда в некоторой точке пространства есть ρ (Кл/м^3), то следует разбить весь объем на элементы dV. Тогда

$$\Phi = \frac{1}{4\pi\varepsilon}\int_V \frac{\rho dV}{r}.$$

Если заряд распределен лишь в весьма тонких слоях у поверхности заряженных тел, как это имеет место у тел из проводящего материала, то можно считать, что заряд распределен на поверхности тел. Разбивая заряженные поверхности на элементы dS, можно записать $dq = \sigma dS$, где σ – поверхностная плотность заряда (Кл/м^2). Тогда выражение для потенциала принимает вид

$$\Phi = \frac{1}{4\pi\varepsilon}\int_S \frac{\sigma dS}{r},$$

причем интеграл должен быть распространен по всем заряженным поверхностям. То обстоятельство, что в объемах, занятых заряженными телами, находится проводящая среда и, следовательно, среда во всем пространстве неоднородна, в данном случае несущественно, так как внутри проводящих тел поле отсутствует. Мы могли бы мысленно убрать проводящее вещество тел, заменив его диэлектриком с проницаемостью ε и сохранив все поверхностные заряды тел. При этом поле осталось бы без изменения.

В случае, когда заряд распределен на проводах, диаметр сечения которых мал по сравнению с расстояниями от проводов до точек поля, в которых определяется потенциал, можно считать заряд сосредоточенным на осях проводов. Если τ – линейная плотность заряда, то $dq = \tau dl$ и тогда

$$\Phi = \frac{1}{4\pi\varepsilon}\int_l \frac{\tau dl}{r},$$

причем интеграл распространяется вдоль всех заряженных проводов.

Следует отметить два важных аспекта. Приведенными выше формулами можно пользоваться для вычисления потенциала лишь в том случае, если заряды распределены в конечной области пространства. При этом физический смысл имеет только объемное распределение зарядов. Тем не менее, условное представление о поверхностном, линейном или точечном распределении зарядов весьма полезно при решении многих практических задач.

С учетом выражения $\mathbf{E} = -\nabla\Phi$ напряженность электрического поля вычисляется как

$$\mathbf{E} = \frac{1}{4\pi\varepsilon}\int_V \frac{\rho\mathbf{r}}{r^3}dV,$$

где **r** – вектор, направленный от точки расположения заряда к точке определения напряженности поля и равный расстоянию между ними.

Поскольку вектор электрической индукции **D** связан с вектором **E** соотношением $\mathbf{D} = \varepsilon\mathbf{E}$, то

$$\mathbf{D} = -\varepsilon\nabla\Phi. \qquad (1.20)$$

Если индукция создается зарядом с объемной плотностью ρ, то в соответствии с законом Гаусса

$$\nabla\cdot\mathbf{D} = \nabla\cdot(-\varepsilon\nabla\Phi) = \rho,$$

тогда при ε = const (диэлектрик однороден) получим

$$\nabla^2\Phi = -\frac{\rho}{\varepsilon}. \qquad (1.21)$$

Уравнение (1.21), устанавливающее связь между потенциалом, созданным произвольно распределенными зарядами, и объемной плотностью этих зарядов, известно как уравнение Пуассона. В частном случае, когда в рассматриваемом объеме нет зарядов, например вне проводников линии передачи, уравнение упрощается:

$$\nabla^2\Phi = 0. \qquad (1.22)$$

Это уравнение известно как уравнение Лапласа. Задача формулируется следующим образом: найти решение уравнения (1.22) для двух- или трехмерного потенциала, удовлетворяющее заданным граничным условиям. Когда распределение потенциала

найдено, по нему с помощью формулы (1.19) находится структура электрического поля.

1.7 Граничные условия

1.7.1 Граничные условия на поверхности проводников

Из материального уравнения $\mathbf{J} = \gamma\mathbf{E}$ следует, что внутри проводников, так как $\gamma \neq 0$, всюду должно быть $\mathbf{E} = 0$ (так как в электростатическом поле электрические токи отсутствуют). Тогда из выражения $\mathbf{E} = -\nabla\Phi$ следует, что для каждого проводника потенциал всех его точек имеет одно и то же значение, т. е. $\Phi = \text{const}$. При этом поверхности проводников – это поверхности равного электрического потенциала и линии напряженности электрического поля в диэлектрике нормальны к ним. Обозначив через E_n и E_t нормальную и тангенциальную (касательную) к поверхности проводника составляющие вектора $\mathbf{E}$ в диэлектрике около поверхности проводника, получим граничное условие для поля в диэлектрике на поверхности проводника. Так как $\Phi = \text{const}$, то $E_t = 0$ и $\mathbf{E} = E_n = -\partial\Phi/\partial n$, поэтому

$$\mathbf{D} = \varepsilon\mathbf{E} = -\varepsilon\partial\Phi/\partial n = \sigma.$$

1.7.2 Граничные условия на поверхности раздела диэлектриков

Отдельно рассмотрим граничные условия для нормальных и тангенциальных составляющих электрического поля на поверхности раздела диэлектрик-диэлектрик.

Граничные условия для нормальных составляющих электрического поля. В данном случае возможны два варианта.

– Плотность поверхностных электрических зарядов равна нулю, $\sigma = 0$. Нетрудно показать, что при этом $D_{n1} = D_{n2}$, где индексы 1 и 2 обозначают среды с ε_1 и ε_2 соответственно. Тогда $\varepsilon_1 E_{n1} = \varepsilon_2 E_{n2}$. Таким образом, при отсутствии поверхностных электрических зарядов на границе раздела двух сред нормальные составляющие индукции электрического поля будут непрерывны,

а нормальные составляющие векторов напряженности электрического поля будут испытывать скачок.

– На границе раздела равномерно распределен поверхностный электрический заряд, который имеет плотность $\sigma \neq 0$. Используя закон Гаусса, можно показать, что $D_{n1} - D_{n2} = \sigma$. Это означает, что при наличии заряженной границы раздела двух сред нормальные составляющие индукции электрического поля испытывают скачок на величину плотности поверхностного заряда.

Граничные условия для тангенциальных составляющих электрического поля. С помощью закона электромагнитной индукции можно показать, что на границе раздела диэлектрик-диэлектрик тангенциальные составляющие векторов напряженности электрического поля сред непрерывны, а тангенциальные составляющие индукции электрического поля претерпевают разрыв, т. е.

$$E_{t1} = E_{t2}, \quad \varepsilon_2 D_{n1} = \varepsilon_1 D_{n2}.$$

1.8 Основная задача электростатики и теорема единственности

Основной задачей электростатики является определение напряженности во всех точках электростатического поля при заданных зарядах или потенциалах тел, нахоящихся в этом поле. Если полностью задано распределение электрических зарядов в однородной и изотропной среде, то решение может быть получено методом, изложенным в подразделе 1.6. Обратная задача отыскания распределения зарядов по заданному распределению потенциала решается с помощью уравнения Лапласа и граничного условия на поверхности заряженных проводящих тел. Однако большей частью задача оказывается значительно сложнее. Обычно рассматривается система заряженных проводящих тел, окруженных диэлектриком, в котором отсутствуют объемные заряды. Заданы потенциалы всех тел (задача Дирихле) либо полные их заряды (задача Неймана) или их комбинация (задача Дирихле – Неймана). Распределение же зарядов по поверхности каждого тела неизвест-

но и подлежит определению. В этом и заключается основная трудность задачи. Такая ситуация возникает, например, при анализе линий передачи. Также неизвестным является распределение потенциала в пространстве. Особенно усложняется задача для неоднородной или неизотропной среды.

Решение такой задачи аналитическим путем в конечном виде возможно только для частных случаев. Иногда удается найти решение при помощи искусственных приемов. В связи с этим чрезвычайно важно установить необходимые и достаточные требования, при удовлетворении которых поле определяется единственным образом:

– поле в диэлектрике должно удовлетворять уравнениям (1.18). Для однородной среды эти уравнения приводятся к одному уравнению Лапласа;

– поверхности проводящих тел должны быть поверхностями равного потенциала, т. е. для каждой поверхности должно быть соблюдено условие Ф = const;

– потенциалы на поверхности тел должны быть равны заданным значениям Φ_k, если по условиям задачи известны эти потенциалы. Если же заданы полные заряды тел, то для каждого тела должно быть удовлетворено условие

$$q_k = \int\limits_{s_k} \sigma dS = -\int\limits_{s_k} \varepsilon \frac{\partial \Phi}{\partial n} dS.$$

Выполнение данных требований является достаточным для того, чтобы задача имела единственное решение. Это положение известно как теорема единственности.

1.9 Метод зеркальных изображений

Некоторые задачи электростатики можно решить аналитически, если результирующее поле обладает симметрией. В случае ее отсутствия задача существенно усложняется и при ее решении стремятся аппроксимировать рассматриваемую систему ее симметричным аналогом. Одним из методов аппроксимации является метод зеркальных изображений, применимый, когда поле ограни-

чено поверхностями (проводящими или раздела диэлектрик-диэлектрик) правильной геометрической формы. В этом случае расчет поля заряженных зарядов или проводников сводится к расчету поля нескольких зарядов или проводников при отсутствии проводящей среды. Рассмотрим особенности этого метода на конкретном примере.

Предположим, что положительный точечный заряд q находится на расстоянии h над идеальной проводящей и заземленной плоскостью (рисунок 1.4). Это соответствует, например, проводу, подвешенному над поверхностью земли [10].

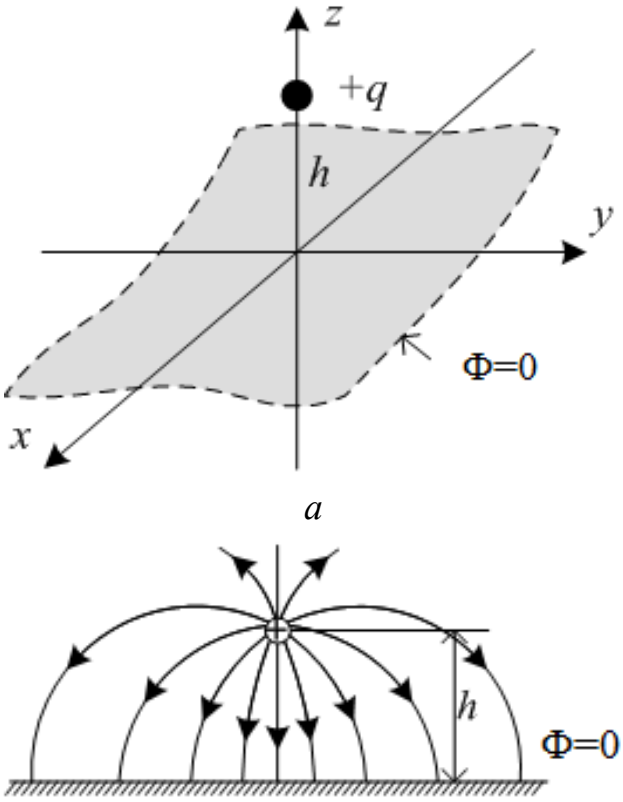


Рисунок 1.4 – Общий вид (*а*) и картина распределения силовых линий (*б*) для точечного заряда над заземленной проводящей поверхностью

Возникает вопрос, а какой потенциал будет над поверхностью земли? Ответ $(1/4\pi\varepsilon)qh$ будет неверным, так как силовые линии напряженности электрического поля, начинающиеся на положительном заряде (или заряженном проводе), заканчиваются у заземленной поверхности, где появляется индуцированный от-

рицательный заряд. Поэтому полный потенциал будет частично обусловлен как непосредственно зарядом q, так и зарядом, индуцированным на заземленной поверхности. В результате возникает другой резонный вопрос, а как рассчитать потенциал, когда неизвестно, какой заряд индуцирован на заземленной поверхности и как он распределен по ней?

С математической точки зрения эта задача заключается в том, что надо решить уравнение Пуассона в области $z > 0$ с одним точечным зарядом q в точке $(0, 0, h)$ при следующих граничных условиях:

– $\Phi = 0$ при $z = 0$ (поскольку поверхность заземлена);

– $\Phi \to 0$ при удалении от точечного заряда (когда $x^2 + y^2 + z^2 >> h^2$).

Согласно теореме единственности возможное распределение заряда по поверхности проводника только одно, т. е. существует только одна функция (решение), которая удовлетворяет этим требованиям. Соответственно если каким-либо образом найти такую функцию, то она и будет являться ответом на поставленный вопрос. Для наглядности сначала рассмотрим другую задачу, особенность решения которой послужит идеей для нахождения требуемой функции.

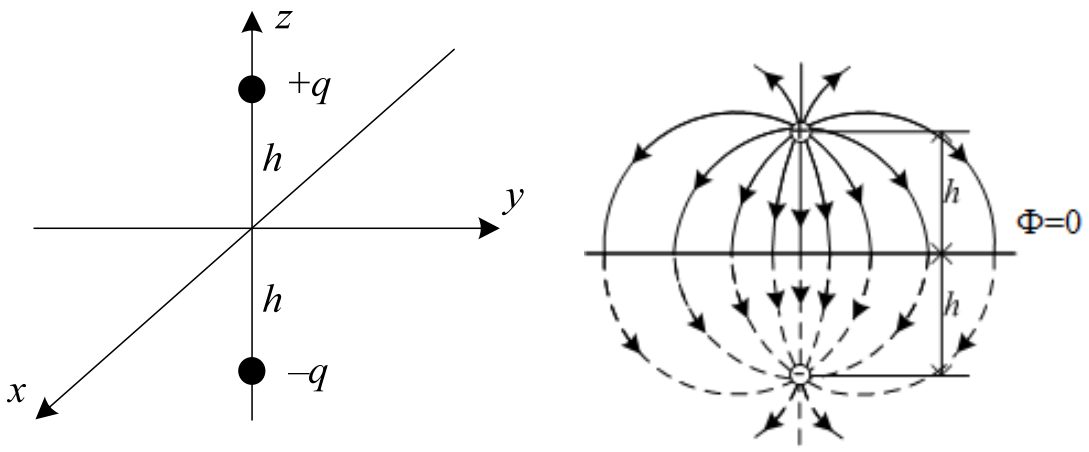
Пусть два точечных заряда $+q$ и минус q расположены в точках $(0, 0, h)$ и $(0, 0, –h)$ соответственно (рисунок 1.5). Тогда потенциал в точке (x, y, z) определяется выражением

$$\Phi(x, y, z) = \frac{1}{4\pi\varepsilon}\left[\frac{q}{\sqrt{x^2+y^2+(z-h)^2}} - \frac{q}{\sqrt{x^2+y^2+(z+h)^2}}\right], \quad (1.23)$$

где знаменатели соответствуют расстояниям от точки (x, y, z) до зарядов $+q$ и минус q соответственно. Отсюда следует, что $\Phi = 0$ при $z = 0$ и $\Phi \to 0$ при $x^2 + y^2 + z^2 >> h^2$, а единственным зарядом в области, где $z > 0$, является точечный заряд $+q$. Это соответствует условиям исходной задачи.

Очевидно, что в «верхней» области ($z \geq 0$) распределение заряда будет одинаковыми как в первом (см. рисунок 1.4), так и во втором (см. рисунок 1.5) случае. При этом «нижняя» область

($z < 0$) будет иметь совершенно другое распределение потенциала, но для решения исходной задачи это не имеет никакого значения. Таким образом, потенциал точечного заряда над бесконечным заземленным проводником ($z \geq 0$) может быть найден по выражению (1.23).



а *б*

Рисунок 1.5 – Общий вид (*а*) и картина распределения (*б*) силовых линий для двух точечных зарядов

Когда потенциал найден, очень просто вычислить поверхностный заряд σ, индуцированный на проводнике. Рассмотрим уравнение

$$\sigma = -\varepsilon \frac{\partial \Phi}{\partial n}.$$

В нашем примере нормальным направлением является направление вдоль оси z, поэтому

$$\sigma = -\varepsilon \left. \frac{\partial \Phi}{\partial z} \right|_{z=0}.$$

Тогда, используя уравнение (1.23), получим

$$\frac{\partial \Phi}{\partial z} = \frac{1}{4\pi\varepsilon} \left[\frac{-q(z-h)}{\left[x^2 + y^2 + (z-h)^2 \right]^{3/2}} + \frac{q(z+h)}{\left[x^2 + y^2 + (z+h)^2 \right]^{3/2}} \right],$$

в результате

$$\sigma(x,y)=\frac{-qh}{(x^2+y^2+h^2)^{3/2}}. \qquad (1.24)$$

Как и ожидалось, индуцированный заряд является отрицательным (при условии, что заряд q положительный) и наибольшим при $x = y = 0$.

Очевидно, что, зная плотность распределения заряда, легко найти полный индуцированный заряд:

$$Q=\int \sigma da.$$

Этот интеграл в плоскости xy может быть найден в декартовых координатах при $da = dxdy$, но проще использовать полярные координаты, т. е. $r^2 = x^2 + y^2$ и $da = rdrd\varphi$. Тогда

$$\sigma(r)=\frac{-qh}{2\pi(r^2+h^2)^{3/2}}$$

и

$$Q=\int_0^{2\pi}\int_0^{\infty}\frac{-qh}{2\pi\left(r^2+h^2\right)^{3/2}}rdrd\varphi=\left.\frac{qh}{\sqrt{r^2+h^2}}\right|_0^{\infty}=-q. \qquad (1.25)$$

Таким образом, общий заряд, индуцированный на заземленной поверхности, равен минус q.

Положительный заряд q притягивается к заземленной поверхности из-за отрицательного индуцированного заряда на ней. Оценим силу притяжения. Поскольку потенциал в окрестности расположения заряда q такой же, как и в конфигурации с двумя разноименными зарядами (см. рисунок 1.5), то одинаковы и поле, и сила притяжения:

$$\mathbf{F}=-\frac{q^2}{4\pi\varepsilon 2h^2}\mathbf{z}.$$

При этом энергия в случае двух разноименных зарядов (см. рисунок 1.5) будет

$$W=-\frac{q^2}{4\pi\varepsilon 2h},$$

а в случае заряда над заземленной поверхностью –

$$W = -\frac{q^2}{4\pi\varepsilon 4h},$$

т. е. в два раза меньше (поскольку для конфигурации из двух зарядов обе области ($z > 0$ и $z < 0$) вносят свой вклад в суммарную энергию, а так как заряды расположены симметрично, эти области вносят одинаковый вклад).

Метод зеркальных изображений не ограничивается случаем единичного точечного заряда. Так, любое стационарное распределение заряда вблизи заземленной поверхности можно трактовать таким же образом, вводя его зеркальное изображение. При этом каждый заряд должен быть зеркально отражен с изменением знака, после чего заземленная поверхность может быть мысленно удалена. Тогда плоскость, расположенная на месте удаленной поверхности, является поверхностью равного потенциала, так как заряды противоположных знаков размещены симметрично относительно нее. Поэтому найденное поле будет соответствовать полю над заземленной поверхностью.

Рассмотренный метод зеркальных изображений может использоваться, когда заземленная поверхность образована двумя плоскостями, сходящимися под углом $\alpha = \pi/n$, где n – целое число. При этом пространство разделяется на одинаковые части плоскостями, пересекающимися под углом α, и, последовательно отражая заряд в этих плоскостях, получают систему из действительного заряда и серии его зеркальных изображений. Кроме того, этот метод применим, когда плоская и заземленная поверхность разделяет две среды с различными диэлектрическими проницаемостями, а также когда источником поля является электрический заряд, распределенный вдоль провода с линейной плотностью τ [10].

1.10 Квазистатический подход и линии передачи

С помощью уравнений Максвелла можно получить решение для системы любой сложности, но это, как правило, сопровождается значительными вычислительными затратами. На практике

часто используют мощные вычислительные кластеры (вычислительные центры, суперкомпьютеры) или прибегают к различного рода упрощениям, одним из которых является использование квазистатического подхода. Данный подход (также называемый Т- или ТЕМ-аппроксимацией) применяют, когда поперечные размеры рассматриваемой системы (структуры) малы по сравнению с длиной распространяющейся электромагнитной волны [11]. Например, в большинстве линий передачи (устройств, ограничивающих область распространения электромагнитных колебаний и направляющих поток сверхвысокочастотной электромагнитной энергии в заданном направлении[1]) распространяются поперечные ТЕМ- или квази-ТЕМ-волны, где электрические и магнитные поля перпендикулярны (или почти перпендикулярны) к направлению распространения. Такой подход позволяет упростить уравнения Максвелла, в частности пренебречь током смещения в законе полного тока Ампера. В результате требуется решение не уравнений Максвелла, а уравнения Пуассона – Лапласа. Квазистатический подход применим в диапазоне от постоянного тока до частот 10 ГГц [12].

Характерное поперечное сечение линии передачи, в данном случае воздушной микрополосковой, показано на рисунке 1.6. Распространение ТЕМ-волны существует на любой частоте. Однако выше определенной частоты также распространяются и высшие типы волн. Линиями передачи с ТЕМ-волной являются коаксиальные, микрополосковые, полосковые, копланарные и др. Линиями передачи, не относящимися к ТЕМ-линиям, являются полый круглый или прямоугольный металлические волноводы.

Квазистатический подход получил широкое распространение при математическом моделировании линий передачи, в общем случае многопроводных, применяемых, например, в системах кабельного и воздушного электроснабжения. Особенностью таких линий является необходимость учета распределенности их параметров, что находит отражение при решении проектных и эксплуатационных задач, например при определении мест короткого замыкания. Частным случаем МПЛП являются различные полос-

[1] ГОСТ 18238-72. Линии передачи сверхвысоких частот. Термины и определения.

ковые структуры, которые широко используются для создания элементов РЭС: печатных плат, фильтров, средств снижения уровня перекрестных помех или их компенсации, устройств защиты, фазовращателей, антенн, линий задержки, высокоскоростных межсоединений и пр.

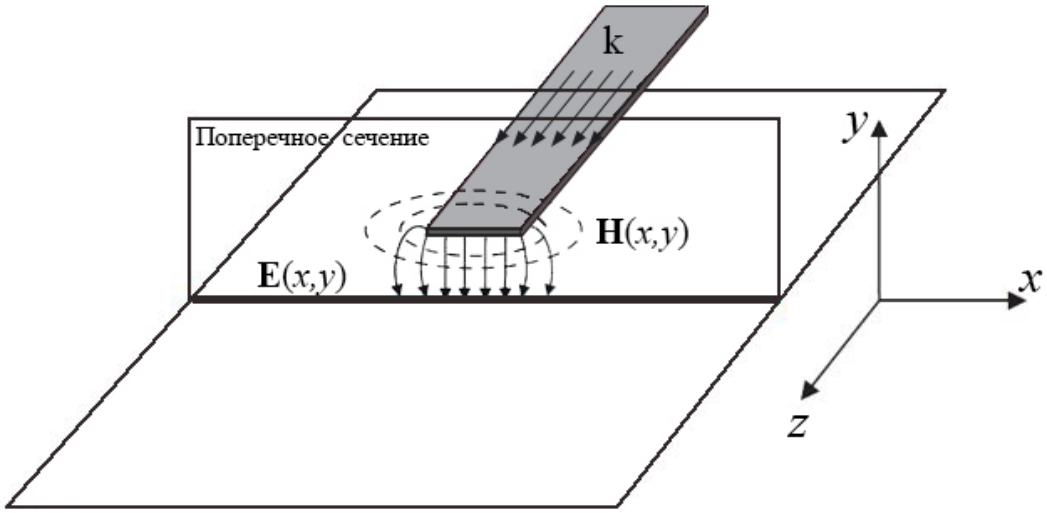


Рисунок 1.6 – Общий вид воздушной микрополосковой линии передачи

При квазистатическом подходе электрические характеристики, меняющиеся вдоль отрезков МПЛП длиной *dx*, описываются матрицами погонных первичных параметров **R** (Ом/м), **L** (Гн/м), **C** (Ф/м) и **G** (См/м), или кратко **RLCG**-параметрами. Отдельно отметим, что вычисление (экстракция) этих матриц (в частности, **R** и **C**) является важной задачей при проектировании с учетом паразитных параметров выводов интегральных схем, а случайные вариации их параметров в технологическом процессе вынуждают многократно вычислять эти матрицы. Вычисленные матрицы затем используются при решении телеграфных уравнений Хевисайда или производных от них для анализа целостности сигналов, получения временного отклика и других параметров.

В случае неучета потерь в проводниках и диэлектриках, из которых состоит линия передачи, вычисляются только матрицы **L** и **C**. Эти потери объясняются неидеальностью материлов, выражающейся частотной зависимостью электрофизических параметров линии (относительных диэлектрической и магнитной

проницаемостей), скин-эффектом (ток высокой частоты протекает преимущественно в тонком поверхностном слое проводника), эффектом близости (притяжение противоположных токов в соседних проводниках) и угловым эффектом (сжатие тока вблизи углов проводника). Таким образом, строгое решение уравнений Максвелла сводится к двум независимым граничным задачам электростатики и магнитостатики, определяющим поведение поперечных электрических и магнитных компонентов поля. При этом решение первой задачи дает матрицы **C** и **G**, а второй – **L** и **R**. Для экономии вычислительных затрат часто прибегают только к решению электростатической задачи, а из вычисленной матрицы **C** находят **L** и затем **R**. При этом вычисленная матрица **L** является частотно-независимой и тем самым лишь приближенной. На практике это приемлемо, например, при проектировании интегральных схем, поскольку частоты сигналов не так высоки и индуктивные эффекты проявляются слабо.

Для пояснения процесса вычислений рассмотрим уравнение Пуассона (дифференциальная форма)

$$\nabla^2 \varphi = -\rho/\varepsilon,$$

где φ – электростатический потенциал; ρ – объемная плотность заряда. При отсутствии в анализируемой области свободных зарядов данное уравнение сводится к уравнению Лапласа. Для нахождения волнового сопротивления и других параметров одиночной линии передачи с неоднородным диэлектрическим заполнением без потерь необходимо определить погонную емкость с диэлектрическим заполнением C и без него (без границ диэлектрик-диэлектрик) C_0. Волновое сопротивление линии передачи без потерь определяется как

$$Z = \sqrt{\frac{L}{C}}, \tag{1.26}$$

где L – погонная индуктивность; C – погонная емкость. Для нахождения индуктивности можно воспользоваться выражением

$$L = \mu_0 \varepsilon_0 C_0^{-1}.$$

При этом фазовая скорость определяется как

$$u = \frac{1}{\sqrt{LC}} \quad \text{или} \quad u = c\sqrt{\frac{C_0}{C}} = \frac{c}{\sqrt{\varepsilon_{r\text{eff}}}}, \tag{1.27}$$

где c – скорость света в свободном пространстве; $\varepsilon_{r\text{eff}} = C/C_0$ – эффективная относительная диэлектрическая проницаемость.

В случае простых конфигураций линии можно воспользоваться известными аналитическими выражениями [13]. Для сложных МПЛП такие выражения отсутствуют или имеют ограниченную точность и необходимо прибегать к помощи численных методов.

В общем случае величины L и C в приведенных выражениях являются матрицами погонных коэффициентов электростатической (**C**) и электромагнитной (**L**) индукции [14]. Также матрицу **C** часто называют емкостной матрицей Максвелла, узловой емкостной матрицей и погонной емкостной матрицей (для краткости далее емкостная матрица). Поясним особенности вычисления емкостной матрицы **C** на примере трехпроводной линии передачи, поперечное сечение которой приведено на рисунке 1.7.

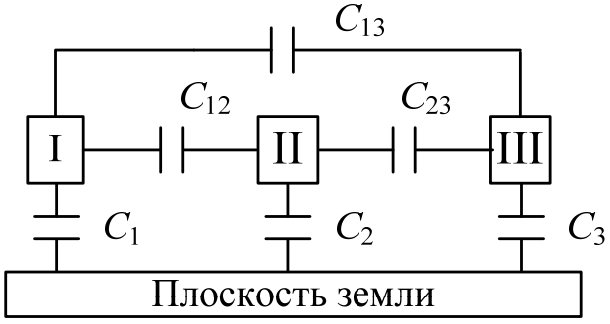


Рисунок 1.7 – Поперечное сечение трехпроводной линии передачи

Рассмотрим задачу нахождения погонных зарядов Q при известных потенциалах на них [10]. На практике часто потенциал опорного проводника (плоскости земли) устанавливается равным нулю. Разности потенциалов между ним и проводниками обозначим Φ_1, Φ_2 и Φ_3, а между проводниками – Φ_{12}, Φ_{13} и Φ_{23}. Тогда

$$Q_{\text{I}} = C_1\Phi_1 + C_{12}\Phi_{12} + C_{13}\Phi_{13},$$

$$Q_{\text{II}} = C_{21}\Phi_{12} + C_2\Phi_2 + C_{23}\Phi_{23},$$

$$Q_{\text{III}} = C_{31}\Phi_{13} + C_{32}\Phi_{23} + C_3\Phi_3.$$

Перепишем эту систему уравнений в виде

$$\begin{aligned}
Q_{\text{I}} &= C_1\Phi_1 + C_{12}(\Phi_1 - \Phi_2) + C_{13}(\Phi_1 - \Phi_3) = \\
&= (C_1 + C_{12} + C_{13})\Phi_1 - C_{12}\Phi_2 - C_{13}\Phi_3, \\
Q_{\text{II}} &= C_{21}(\Phi_2 - \Phi_1) + C_2\Phi_2 + C_{23}(\Phi_2 - \Phi_3) = \\
&= -C_{21}\Phi_1 + (C_2 + C_{21} + C_{23})\Phi_2 - C_{23}\Phi_3, \\
Q_{\text{III}} &= C_{31}(\Phi_3 - \Phi_1) + C_{32}(\Phi_3 - \Phi_2) + C_3\Phi_3 = \\
&= -C_{31}\Phi_1 - C_{32}\Phi_2 + (C_3 + C_{31} + C_{32})\Phi_3,
\end{aligned}$$

или в матричном виде

$$\mathbf{Q} = \underline{\mathbf{C}}\boldsymbol{\Phi},$$

где

$$\underline{\mathbf{C}} = \begin{pmatrix} \underline{C}_{11} & \underline{C}_{12} & \underline{C}_{13} \\ \underline{C}_{21} & \underline{C}_{22} & \underline{C}_{23} \\ \underline{C}_{31} & \underline{C}_{32} & \underline{C}_{33} \end{pmatrix} =$$

$$= \begin{pmatrix} C_1 + C_{12} + C_{13} & -C_{12} & -C_{13} \\ -C_{21} & C_2 + C_{21} + C_{23} & -C_{23} \\ -C_{31} & -C_{32} & C_3 + C_{31} + C_{32} \end{pmatrix}.$$

Коэффициенты $\underline{C}_{ij}$ называются коэффициентами электростатической индукции – собственными при одинаковых индексах и взаимными при разных индексах. Они имеют размерность погонной емкости [15]. Несмотря на отрицательный знак у внедиагональных элементов матрицы $\underline{\mathbf{C}}$, емкость между отдельным проводником и плоскостью земли положительна. Тогда матрица **L** вычисляется как

$$\mathbf{L} = \mu_0\varepsilon_0\underline{\mathbf{C}}_0^{-1}.$$

При учете потерь в проводниках и диэлектриках волновое сопротивление МПЛП описывается комплексной матрицей

$$\mathbf{Z} = \sqrt{\frac{\mathbf{R} + j\omega\mathbf{L}}{\mathbf{G} + j\omega\underline{\mathbf{C}}}}$$

порядка N_{COND} – число проводников МПЛП, не считая опорного. Для вычисления матрицы **G** используется та же модель, что и для

матрицы **C**, с той лишь разницей, что диэлектрическая проницаемость i-го подынтервала заменяется на комплексную с использованием тангенса угла потерь $\underline{\varepsilon}_r = \underline{\varepsilon}_r' - j\underline{\varepsilon}_r'' = \underline{\varepsilon}_r'(1 - j\tan\delta)$, где $\tan\delta = \underline{\varepsilon}_r'' / \underline{\varepsilon}_r'$; σ – удельная проводимость диэлектрика.

Нахождение матрицы **R** представляет собой сложную задачу. Так, для нетиповых структур часто прибегают к ее измерениям или различного рода упрощениям. Для ее вычисления разработано несколько подходов разной степени сложности, пригодных только для некоторых частотных диапазонов. С точки зрения минимизации вычислительных затрат на вычисление матрицы **R** выделяется работа [16], где представлено обобщение на случай МПЛП результатов для одиночных линий передачи, использующих правило дифференциальной индуктивности (incremental inductance rule). Рассмотрим ее основные результаты.

В случае одиночной линии передачи потери (омические) в проводнике определяются как

$$R = \frac{1}{\mu_0}\sum_j R_{sj}\frac{\partial L}{\partial n_j}, \quad R_{sj} = \sqrt{\pi f \mu \rho_{sj}},$$

где $\partial L/\partial n_j$ – производная индуктивности при небольшой вариации (возмущения) j-й границы поверхности проводника к его центру (сужение проводника); n_j – вектор нормали к границе j; R_{sj} – поверхностное сопротивление слоя вариации для этой границы (рисунок 1.8); ρ_{sj} – удельное сопротивление проводника с границей j.

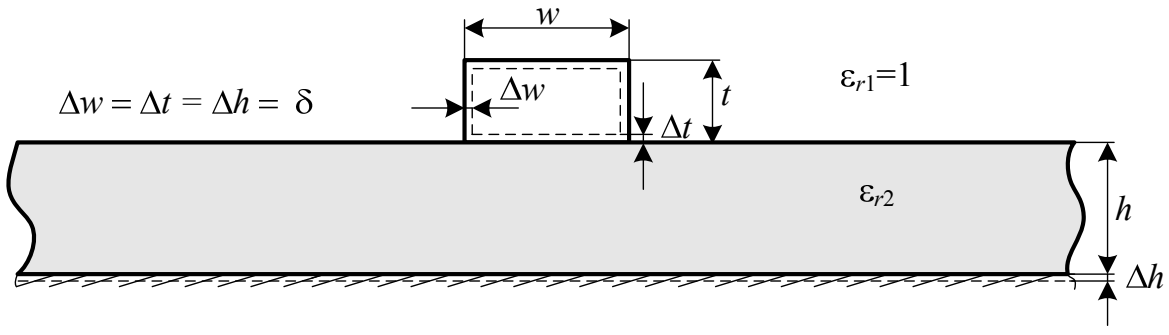


Рисунок 1.8 – Пояснение вариации границ проводников

В работе [16] предложено не сужать границы проводников, а наоборот, расширять, что позволило обобщить данный подход

на случай МПЛП и получить более точные результаты. Элементы матрицы **R** вычисляются как

$$R_{ik} = \begin{cases} \dfrac{1}{\mu_0}\sum_j R_{sj}\dfrac{-\partial L_{ii}}{\partial n_j}, & \text{если } i = k; \\ \dfrac{1}{\mu_0}\sum_j R_{sj}\dfrac{-\partial L_{ik}}{\partial n_j}, & \text{иначе.} \end{cases}$$

При $i = k$ возмущения применяются к i-му и опорному проводникам, а в противном случае – только к опорным.

Таким образом, особый интерес для минимизации затрат времени на получение всех первичных параметров МПЛП представляет вычисление емкостной матрицы, поскольку, как было показано выше, остальные три матрицы являются ее производными. Поэтому в последующих разделах особое внимание уделяется именно вычислению емкостной матрицы.

Контрольные вопросы и задания

1. Опишите процесс построения математической модели для анализа электромагнитных задач.

2. Запишите уравнения электростатики.

3. В чем различие между уравнениями Лапласа и Пуассона?

3. Как классифицируются дифференциальные уравнения в частных производных?

4. Классифицируйте интегральное уравнение

$$\Phi(x) = 2 - \int_0^x \Phi(t)dt.$$

5. Поясните основные положения квазистатического подхода.

6. Почему внедиагональные элементы емкостной матрицы имеют отрицательные значения?

7. Опишите граничные условия на поверхности проводников и диэлектриков.

8. Когда применим метод зеркальных уравнений?

9. Опишите процесс вычисления погонных параметров линии передачи.

2 МЕТОДЫ РЕШЕНИЯ СИСТЕМЫ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ (СЛАУ)

2.1 Общие сведения

2.1.1 Постановка задачи

В вычислительной линейной алгебре выделяют 4 основные задачи: решение СЛАУ; вычисление определителей; нахождение обратных матриц; определение собственных значений и собственных векторов. При этом решение СЛАУ называют первой основной задачей. Эффективность способа решения СЛАУ вида

$$\mathbf{Ax} = \mathbf{b} \tag{2.1}$$

во многом зависит от структуры матрицы **A**: размера, обусловленности, симметричности, заполненности (т.е. соотношения между числом ненулевых и нулевых элементов), специфики расположения ненулевых элементов в матрице и т. д. Будем полагать, что матрица **A** задана и является невырожденной. Известно, что в этом случае решение системы существует, единственно и устойчиво по входным данным, т. е. рассматриваемая задача корректна.

Пусть $\mathbf{x}^* = \left(x_1^*, x_2^*, ..., x_N^*\right)^T$ – приближенное решение СЛАУ (T – символ транспонирования). Будем стремиться к получению решения, для которого погрешность $\mathbf{e} = \mathbf{x} - \mathbf{x}^*$ мала. Заметим, что качество полученного решения далеко не всегда характеризуется тем, насколько мала погрешность $\mathbf{x} - \mathbf{x}^*$. Иногда вполне удовлетворительным является критерий малости невязки $\mathbf{r} = \mathbf{b} - \mathbf{Ax}^*$. Так, вектор **r** показывает, насколько отличается правая часть системы от левой, если подставить в нее приближенное решение. Заметим также, что $\mathbf{r} = \mathbf{Ax} - \mathbf{Ax}^* = \mathbf{A}(\mathbf{x} - \mathbf{x}^*)$, поэтому погрешность и невязка связаны равенством

$$\mathbf{e} = \mathbf{x} - \mathbf{x}^* = \mathbf{A}^{-1}\mathbf{r}. \tag{2.2}$$

2.1.2 Нормы векторов

Решением СЛАУ является вектор $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$, который будем рассматривать как элемент векторного пространства $\boldsymbol{C}^N$. Приближенное решение $\mathbf{x}^* = \left(x_1^*, x_2^*, ..., x_N^*\right)^T$ и погрешность $\mathbf{e} = \mathbf{x} - \mathbf{x}^* = \left(x_1 - x_1^*, x_2 - x_2^*, ..., x_N - x_N^*\right)^T$ также являются элементами пространства $\boldsymbol{C}^N$. Для того чтобы анализировать методы решения СЛАУ, необходимо уметь количественно оценивать «величины» векторов $\mathbf{x}^*$ и $\mathbf{x} - \mathbf{x}^*$, а также векторов $\mathbf{b}$ и $\mathbf{b} - \mathbf{b}^*$, где $\mathbf{b}^* = \left(b_1^*, b_2^*, ..., b_N^*\right)^T$ – вектор приближенно заданных правых частей. Удобной для этой цели количественной характеристикой является широко используемое понятие нормы вектора.

Говорят, что в пространстве $\boldsymbol{C}^N$ задана норма, если каждому вектору $\mathbf{x}$ из $\boldsymbol{C}^N$ сопоставлено вещественное число $||\mathbf{x}||$, называемое нормой вектора $\mathbf{x}$ и обладающее следующими свойствами:

1) $||\mathbf{x}|| \geq 0$, причем $||\mathbf{x}|| = 0$ тогда и только тогда, когда $\mathbf{x} = 0$;

2) $||\alpha\mathbf{x}|| = |\alpha| \, ||\mathbf{x}||$ для любого вектора $\mathbf{x}$ и любого числа α;

3) $||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}||$ для любых векторов $\mathbf{x}$ и $\mathbf{y}$.

Заметим, что такими же свойствами обладает обычная геометрическая длина вектора в трехмерном пространстве. Свойство 3 в этом случае следует из правила сложения векторов и из того известного факта, что сумма длин двух сторон треугольника всегда больше длины третьей стороны.

Существует множество различных способов введения норм. В вычислительных методах наиболее употребительными являются следующие три нормы:

$\left\|\mathbf{x}\right\|_1 = \sum_{i=1}^{N} |\, x_i \,|$ – первая (манхэттенская, октоэдрическая) норма;

$\left\|\mathbf{x}\right\|_2 = \left(\sum_{i=1}^{N} |\, x_i \,|^2 \right)^{1/2}$ – вторая (евклидова норма, сферическая) норма;

$\|\mathbf{x}\|_\infty = \max\limits_{1\le i\le N} |x_i|$ – бесконечная (равномерная, чебышевская, кубическая) норма.

В определенном смысле эти нормы эквивалентны, так как каждая из них оценивается любой из двух других с точностью до множителя.

Пример 2.1

Найти нормы $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$, $\|\mathbf{x}\|_\infty$ для вектора $\mathbf{x} = (0.12, -0.15, 0.16)^T$.

Решение

По приведенным выше формулам имеем

$$\|\mathbf{x}\|_1 = 0.12 + 0.15 + 0.16 = 0.43,$$
$$\|\mathbf{x}\|_2 = (0.12^2 + 0.15^2 + 0.16^2)^{1/2} = 0.25,$$
$$\|\mathbf{x}\|_\infty = \max\{0.12, 0.15, 0.16\} = 0.16.$$

2.1.3 Скалярное произведение векторов

Скалярным произведением векторов $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ и $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$ называется величина $\mathbf{x}^T\mathbf{y}$ или

$$(\mathbf{x}, \mathbf{y}) = x_1 y_1 + \ldots + x_N y_N = \sum_{i=1}^{N} x_i y_i.$$

Нетрудно заметить, что $\|\mathbf{x}\|_2 = (\mathbf{x}, \mathbf{x})^{1/2}$. Когда векторы $\mathbf{x}$ и $\mathbf{y}$ имеют комплексные компоненты, скалярное произведение записывается в виде

$$(\mathbf{x}, \mathbf{y}) = x_1 \overline{y}_1 + \ldots + x_N \overline{y}_N = \sum_{i=1}^{N} x_i \overline{y}_i,$$

где черта означает комплексное сопряжение.

2.1.4 Абсолютная и относительная погрешности векторов

Далее будем всюду считать, что в пространстве N-мерных векторов $\mathbf{C}^N$ введена и фиксирована некоторая норма $\|\mathbf{x}\|$. В этом случае в качестве меры степени близости векторов $\mathbf{x}$ и $\mathbf{x}^*$ естественно использовать величину $\|\mathbf{x} - \mathbf{x}^*\|$, являющуюся аналогом

расстояния между точками $\mathbf{x}$ и $\mathbf{x}^*$. Введем абсолютную и относительную погрешности вектора $\mathbf{x}^*$ с помощью формул

$$\Delta\left(\mathbf{x}^*\right) = \left\| \mathbf{x} - \mathbf{x}^* \right\|,$$

$$\delta\left(\mathbf{x}^*\right) = \left\| \mathbf{x} - \mathbf{x}^* \right\| \Big/ \left\| \mathbf{x} \right\|.$$

Выбор той или иной нормы в практических задачах диктуется тем, какие требования предъявляются к точности решения. Выбор нормы $\|\mathbf{x}\|_1$ фактически отвечает случаю, когда малой должна быть суммарная абсолютная погрешность в компонентах решения; выбор нормы $\|\mathbf{x}\|_2$ соответствует критерию малости среднеквадратичной погрешности, а принятие в качестве нормы $\|\mathbf{x}\|_\infty$ означает, что малой должна быть максимальная из абсолютных погрешностей в компонентах решения.

2.1.5 Сходимость по норме

Пусть $\mathbf{x}^{(m)}$ $(m = 1, 2, \ldots, \infty)$ – последовательность векторов $\mathbf{x}^{(m)} = (x_1^{(m)}, x_2^{(m)}, \ldots, x_N^{(m)})^T$. Говорят, что последовательность векторов $\mathbf{x}^{(m)}$ сходится к вектору $\mathbf{x}$ при $m \to \infty$ ($\mathbf{x}^{(m)} \to \mathbf{x}$ при $m \to \infty$), если $\Delta(\mathbf{x}^{(m)}) = \|\mathbf{x}^{(m)} - \mathbf{x}\| \to 0$ при $m \to \infty$.

Сам факт наличия или отсутствия сходимости $\mathbf{x}^{(m)}$ к $\mathbf{x}$ при $m \to \infty$ в конечномерных пространствах не зависит от выбора нормы. Известно, что из сходимости последовательности по одной из норм следует сходимость этой последовательности по любой другой норме. Более того, $\mathbf{x}^{(m)} \to \mathbf{x}$ при $m \to \infty$ тогда и только тогда, когда для всех $i = 1, 2, \ldots, N$ имеем $\mathbf{x}_i^{(m)} \to \mathbf{x}_i$ при $m \to \infty$, т. е. сходимость по норме в пространстве $\boldsymbol{C}^N$ эквивалентна покомпонентной (покоординатной) сходимости.

2.1.6 Нормы матриц

Величина

$$\left\|\mathbf{A}\right\| = \max_{\mathbf{x}\neq 0} \frac{\left\| \mathbf{A}\mathbf{x} \right\|}{\left\| \mathbf{x} \right\|} \tag{2.3}$$

называется нормой матицы $\mathbf{A}$, подчиненной норме векторов, в $\boldsymbol{C}^N$. Понятие матрицы ввел Джеймс Джозеф Сильвестр в 1850 г.

Заметим, что множество всех квадратных матриц размером $N \times N$ является векторным пространством. Можно показать, что введенная в этом пространстве норма обладает следующими свойствами, аналогичными свойствам нормы вектора:

1) $||\mathbf{A}|| \geq 0$, причем $||\mathbf{A}|| = 0$ тогда и только тогда, когда $\mathbf{A} = 0$;

2) $||\alpha\mathbf{A}|| = |\alpha| \, ||\mathbf{A}||$ для любой матрицы $\mathbf{A}$ и любого числа α;

3) $||\mathbf{A} + \mathbf{B}|| \leq ||\mathbf{A}|| + ||\mathbf{B}||$ для любых матриц $\mathbf{A}$ и $\mathbf{B}$.

Дополнительно к этому верны следующие свойства:

4) $||\mathbf{A} \cdot \mathbf{B}|| \leq ||\mathbf{A}|| \cdot ||\mathbf{B}||$ для любых матриц $\mathbf{A}$ и $\mathbf{B}$;

5) для любой матрицы $\mathbf{A}$ и любого вектора $\mathbf{x}$ справедливо неравенство

$$||\mathbf{Ax}|| \leq ||\mathbf{A}|| \cdot ||\mathbf{x}||. \tag{2.4}$$

Как следует из определения (2.3), каждой из векторных норм $||\mathbf{x}||$ соответствует своя подчиненная норма матрицы $\mathbf{A}$. Известно, в частности, что нормам $||\mathbf{x}||_1$, $||\mathbf{x}||_2$ и $||\mathbf{x}||_\infty$ подчинены нормы $||\mathbf{A}||_1$, $||\mathbf{A}||_2$ и $||\mathbf{A}||_\infty$, вычисляемые по формулам

$$\| \mathbf{A} \|_1 = \max_{1 \leq j \leq N} \sum_{i=1}^{N} | a_{ij} |, \tag{2.5}$$

$$\| \mathbf{A} \|_2 = \max_{1 \leq j \leq N} \sqrt{\lambda_j \left(\mathbf{A}^T \mathbf{A} \right)}, \tag{2.6}$$

где $\lambda_j(\mathbf{A}^T\mathbf{A})$ – собственные числа матрицы $\mathbf{A}^T\mathbf{A}$;

$$\| \mathbf{A} \|_\infty = \max_{1 \leq i \leq N} \sum_{j=1}^{N} | a_{ij} |. \tag{2.7}$$

Нормы $||\mathbf{A}||_1$ и $||\mathbf{A}||_\infty$ вычисляются просто. Для вычисления $||\mathbf{A}||_1$ нужно найти сумму модулей элементов каждого из столбцов матрицы $\mathbf{A}$, а затем выбрать максимальную из этих сумм. Для получения значения $||\mathbf{A}||_\infty$ нужно аналогичным образом поступить со строками матрицы $\mathbf{A}$. Как правило, вычислить значение нормы $||\mathbf{A}||_2$ бывает трудно, так как для этого следует искать собственные числа λ_j. Для оценки величины $||\mathbf{A}||_2$ можно, например, использовать неравенство

$$\| \mathbf{A} \|_2 \leq \| \mathbf{A} \|_E,$$

где $\|\mathbf{A}\|_E = \left(\sum_{i,j=1}^{N} | a_{ij} |^2 \right)^{1/2}$ – величина, называемая евклидовой нормой матрицы **A**. Ее также называют нормой Фробениуса.

Пример 2.2

Найти нормы $||\mathbf{A}||_1$, $||\mathbf{A}||_2$, $||\mathbf{A}||_\infty$ для матрицы $\mathbf{A} = \begin{pmatrix} 0.1 & -0.4 & 0 \\ 0.2 & 1 & -0.3 \\ 0 & 0.1 & 0.3 \end{pmatrix}$.

Решение

По приведенным выше формулам имеем

$$\|\mathbf{A}\|_1 = \max\{0.1 + 0.2 + 0;\ 0.4 + 1 + 0.1;\ 0 + 0.3 + 0.3\} = 1.5;$$

$$\|\mathbf{A}\|_\infty = \max\{0.1 + 0.4 + 0;\ 0.2 + 1 + 0.3;\ 0 + 0.1 + 0.3\} = 1.5;$$

$$\|\mathbf{A}\|_2 \le \|\mathbf{A}\|_E = \left(\sum_{i,j=1}^{3} | a_{ij} |^2 \right)^{1/2} \approx 1.18.$$

2.1.7 Обусловленность задачи решения СЛАУ

Известно, что решения различных СЛАУ обладают разной чувствительностью к погрешностям входных данных. Так, задача вычисления решения **x** уравнений **Ax** = **b** может быть как хорошо, так и плохо обусловленной.

Пусть элементы матрицы **A** из (2.1) считаются заданными точно, а правая часть – приближенно. Тогда для погрешности приближенного решения системы справедлива оценка

$$\Delta(\mathbf{x}^*) \le ||\mathbf{A}^{-1}|| \cdot ||\mathbf{r}||, \qquad (2.8)$$

где $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^*$ – невязка, отвечающая $\mathbf{x}^*$. Для доказательства достаточно взять норму левой и правой частей равенства (2.2) и воспользоваться свойством (2.4).

Пусть $\mathbf{x}^*$ – точное решение системы $\mathbf{A}\mathbf{x}^* = \mathbf{b}^*$, в которой правая часть $\mathbf{b}^*$ является приближением к **b**. Тогда верны следующие оценки абсолютной и относительной погрешностей:

$$\Delta(\mathbf{x}^*) \le \nu_\Delta \Delta(\mathbf{b}^*); \qquad (2.9)$$

$$\delta(\mathbf{x}^*) \le \nu_\delta \delta(\mathbf{b}^*), \qquad (2.10)$$

где $\nu_\Delta = \|\mathbf{A}^{-1}\|$, $\nu_\delta(\mathbf{x}) = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{r}\| / \|\mathbf{x}\| = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{Ax}\| / \|\mathbf{x}\|$.

В рассматриваемом случае $\mathbf{r} = \mathbf{b} - \mathbf{Ax}^* = \mathbf{b} - \mathbf{b}^*$ и оценка (2.8) принимает вид (2.9). Разделив обе части этого неравенства на $\|\mathbf{x}\|$ и записав его в виде

$$\Delta(\mathbf{x}^*) / \|\mathbf{x}\| \le (\|\mathbf{A}^{-1}\| \cdot \|\mathbf{b}\| / \|\mathbf{x}\|) \cdot (\Delta(\mathbf{x}^*) / \|\mathbf{b}\|),$$

приходим к оценке (2.10).

Сделаем несколько замечаний относительно числа обусловленности СЛАУ.

– Величина $\nu_\Delta = \|\mathbf{A}^{-1}\|$ для задачи $\mathbf{Ax} = \mathbf{b}$ играет роль абсолютного числа обусловленности.

– Величина $\nu_\delta(\mathbf{x}) = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{r}\| / \|\mathbf{x}\|$ называется естественным числом обусловленности. Она зависит от конкретного решения $\mathbf{x}$ и характеризует коэффициент возможного возрастания относительной погрешности этого решения, вызванного погрешностью задания правой части. Это означает, что $\nu_\delta(\mathbf{x})$ для задачи решения системы $\mathbf{Ax} = \mathbf{b}$ играет роль относительного числа обусловленности.

– Полученные оценки (2.9) и (2.10) точны в том смысле, что для системы $\mathbf{Ax} = \mathbf{b}$ с произвольной невырожденной правой частью $\mathbf{b} \neq 0$ найдется сколь угодно близкий к $\mathbf{b}$ приближенно заданный вектор $\mathbf{b}^* \neq \mathbf{b}$, для которого эти неравенства превращаются в равенства.

Вычислим максимальное значение естественного числа обусловленности, используя определение нормы матрицы:

$$\max_{\mathbf{x}\neq 0} \nu_\delta(\mathbf{x}) = \frac{\| \mathbf{A}^{-1} \| \cdot \| \mathbf{Ax} \|}{\| \mathbf{x} \|} = \| \mathbf{A}^{-1} \| \cdot \| \mathbf{A} \| . \qquad (2.11)$$

Полученную величину называют числом обусловленности матрицы $\mathbf{A}$ и обозначают $\mathrm{cond}(\mathbf{A})$. Таким образом,

$$\mathrm{cond}(\mathbf{A}) = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{A}\|. \qquad (2.12)$$

Отметим, что впервые этот термин предложен А. Тьюрингом в 1948 г.

Следствием оценок (2.9) и (2.10) является оценка

$$\delta(\mathbf{x}^*) \le \mathrm{cond}(\mathbf{A}) \cdot \delta(\mathbf{b}^*). \qquad (2.13)$$

Оценка (2.13) точна в том смысле, что для системы $\mathbf{Ax} = \mathbf{b}$ с произвольной невырожденной матрицей $\mathbf{A}$ найдутся правая часть $\mathbf{b} \neq 0$ (и отвечающее этой правой части решение $\mathbf{x}$) и сколь угодно близкий к $\mathbf{b}$ приближенно заданный вектор $\mathbf{b}^* \neq \mathbf{b}$ такие, что это неравенство превращается в равенство.

Величина $\mathrm{cond}(\mathbf{A})$ является широко используемой количественной мерой обусловленности системы $\mathbf{Ax} = \mathbf{b}$. В частности, систему и матрицу $\mathbf{A}$ принято называть плохо обусловленными, если $\mathrm{cond}(\mathbf{A}) >> 1$. В силу последнего замечания и оценки (2.13) для такой системы существуют решения, обладающие чрезвычайно высокой чувствительностью к малым погрешностям задания входного данного $\mathbf{b}$. Тем не менее заметим, что для всякого решения $\mathbf{x}$ коэффициент $\nu_\delta(\mathbf{x})$ роста относительной погрешности достигает значений, близких к максимально возможному значению $\mathrm{cond}(\mathbf{A})$.

Приведем часто используемые, свойства числа обусловленности.

– Для единичной матрицы $\mathrm{cond}(\mathbf{E}) = 1$.

– Справедливо неравенство $\mathrm{cond}(\mathbf{A}) \geq 1$. Так, из равенства $\mathbf{E} = \mathbf{A}\mathbf{A}^{-1}$, свойства 4 норм матриц и равенства $||\mathbf{E}|| = 1$ следует, что $1 = ||\mathbf{E}|| \leq ||\mathbf{A}^{-1}|| \cdot ||\mathbf{A}|| = \mathrm{cond}(\mathbf{A})$.

– Число обусловленности матрицы $\mathbf{A}$ не меняется при умножении матрицы на произвольное число $\alpha \neq 0$. Заметим, что $(\alpha\mathbf{A})^{-1} = \alpha^{-1}\mathbf{A}^{-1}$. Поэтому $\mathrm{cond}(\alpha\mathbf{A}) = ||\alpha\mathbf{A}|| \cdot ||(\alpha\mathbf{A})^{-1}|| = |\alpha| \cdot ||\mathbf{A}|| \cdot |\alpha|^{-1} \cdot ||\mathbf{A}^{-1}|| = \mathrm{cond}(\mathbf{A})$.

Величина $\mathrm{cond}(\mathbf{A})$ зависит, вообще говоря, от выбора нормы векторов в пространстве $\boldsymbol{C}^N$. Фактически, это есть зависимость максимального коэффициента роста погрешности от способа измерения величины входных данных и решения.

Пример 2.3

Найти $\mathrm{cond}_\infty(\mathbf{A})$ для матрицы $\mathbf{A} = \begin{pmatrix} 1.03 & 0.991 \\ 0.991 & 0.943 \end{pmatrix}$.

Решение

Сначала найдем обратную матрицу: $\mathbf{A}^{-1} \approx \begin{pmatrix} -87.4 & 91.8 \\ 91.8 & -95.4 \end{pmatrix}$.

Тогда $\mathrm{cond}_\infty(\mathbf{A}) = \|\mathbf{A}\|_\infty \cdot \|\mathbf{A}^{-1}\|_\infty \approx 2.021 \cdot 187.2 \approx 378$. Если входные данные для СЛАУ с матрицей **A** содержат относительную погрешность порядка 0,1–1 %, то систему можно расценивать как плохо обусловленную.

Пример 2.4

Рассмотрим систему уравнений

$$1.03x_1 + 0.991x_2 = 2.51,$$
$$0.991x_1 + 0.943x_2 = 2.41,$$

с матрицей из примера 2.3. Ее решением является $x_1 = 1.981$, $x_2 = 0.4735$. Правая часть системы известна с точностью до 0.005, если считать, что числа 2.51 и 2.41 получены округлением «истинных» значений при вводе в память трехзначного десятичного компьютера. Как влияет погрешность во входных данных такого уровня на погрешность решения?

Решение

Возмутим каждый из компонентов вектора $\mathbf{b} = (2.51, 2.41)^T$ на 0.005, взяв $\mathbf{b}^* = (2.505, 2.415)^T$. Решением системы, отвечающим $\mathbf{b}^*$, является теперь $x_1^* = 2.877$, $x_2^* = -0.4629$. Таким образом, решение оказалось полностью искаженным. Относительная погрешность правой части $\delta(\mathbf{b}^*) = \|\mathbf{b} - \mathbf{b}^*\|_\infty / \|\mathbf{b}\|_\infty = 0.005 / 2.51 \approx 0.2$ % привела к относительной погрешности решения $\delta(\mathbf{x}^*) = \|\mathbf{x} - \mathbf{x}^*\|_\infty / \|\mathbf{x}\|_\infty \approx 0.9364 / 1.981 \approx 47.3$ %. Следовательно, погрешность возросла примерно в 237 раз.

Можно ли внести в правую часть системы такую погрешность, чтобы получить существенно большее, чем 237, значение коэффициента роста погрешности? Вычислим естественное число обусловленности, являющееся максимальным значением рассматриваемого коэффициента, отвечающим решению $x_1 = 1.981$, $x_2 = 0.4735$, и получим

$$\nu_\delta(\mathbf{x}) = \|\mathbf{A}^{-1}\|_\infty \cdot \|\mathbf{b}\|_\infty / \|\mathbf{x}\|_\infty \approx 187.2 \cdot 2.51 / 1.981 \approx 237.$$

Таким образом, на поставленный вопрос следует ответить отрицательно.

2.1.8 Масштабирование

Перед началом решения СЛАУ целесообразно масштабировать систему так, чтобы ее коэффициенты были величинами

порядка единицы. Существуют два естественных способа масштабирования системы (2.1). Первый заключается в умножении каждого из уравнений на некоторый масштабирующий множитель t_i. Второй состоит в умножении на масштабирующий множитель α_j каждого j-го столбца матрицы, что соответствует замене переменных $x'_j = \alpha_j^{-1} x_j$ (фактически это замена единиц измерения). В реальных ситуациях чаще всего масштабирование может быть выполнено без существенных трудностей. Однако подчеркнем, что в общем случае удовлетворительного способа масштабирования пока не найдено. На практике масштабирование обычно производят с помощью деления каждого уравнения на его наибольший по модулю коэффициент. Это вполне удовлетворительный способ для большинства реально встречающихся задач.

2.1.9 Форматы хранения матриц

Плотной (полной) считается матрица размером $N \times M$, содержащая NM ненулевых элементов. На практике если в матрице очень мало нулевых элементов по сравнению с их общим количеством, то она тоже считается плотной. Для хранения плотных матриц используются стандартные двухмерные массивы.

Термин «разреженная матрица» имеет достаточно много определений. Наиболее употребительным сейчас является следующее. Разреженной является матрица, для которой существуют специальные приемы, позволяющие извлечь выгоды из большого числа ее нулевых элементов и их расположения. Упомянутая выгода, как правило, заключается в уменьшении требуемой компьютерной памяти (для хранения матрицы по сравнению со стандартным хранением плотных матриц) и количества математических операций с этой матрицей. Исторически извлечение выгоды из наличия разреженности было направлено на решение СЛАУ. Так, в 1950-х гг. началось широкое использование разреженности при решении СЛАУ.

К настоящему времени разработано достаточно большое количество форматов (схем) хранения общих разреженных матриц, например из двух векторов, Кнута, Рейнболдта и Местеньи, Лар-

кума, Шермана, разреженные строчный (CSR) и столбцовый (CSC), разреженный неравномерный диагональный, разреженный блочный строчный, координатный, модифицированный разреженный строчный и др. Также для хранения ленточных матриц разработан сжатый диагональный формат, а для симметричных – симметричный скайлайн-формат. Они отличаются степенью сжатия матрицы и требуемыми затратами машинного времени при их использовании. Для каждого класса задач целесообразен выбор оптимального формата, обеспечивающего экономию машинной памяти. При этом стоит помнить, что увеличение коэффициента сжатия, как правило, ведет к увеличению затрат времени.

Пожалуй, самыми распространенными на данный момент являются форматы CSR, CSC и координатный. Например, в GNU Octave (далее Octave), используется формат CSC, также известный как формат Harwell-Boeing. Указанные форматы предъявляют минимальные требования к памяти и эффективны для реализации базовых операций с разреженными матрицами при решении СЛАУ как прямыми, так и итерационными методами и т. д. Так, в формате CSR используются 3 массива (**aelem** – ненулевые элементы, **jptr** – индексы столбцов, **iptr** – указатели на ненулевые элементы, с которых начинается очередная строка). Формат CSC реализует хранение элементов матрицы по столбцам. При этом также используются 3 массива. Координатный формат использует также 3 массива: для хранения значений матричных элементов (**AA**), их индексов строк (**JR**) и столбцов (**JC**). Достоинством координатного формата является то, что данные, хранящиеся с его помощью, могут быть легко преобразованы в другие форматы. Поэтому он часто используется как стандартный входной формат в пакетах прикладных программ.

2.1.10 Методы решения СЛАУ

Все методы решения СЛАУ можно разбить на два класса: прямые и итерационные. Прямыми называются методы, которые приводят к решению за конечное число арифметических операций. Если операции реализуются точно, то и решение будет точным (в связи с чем к классу прямых методов применяют еще

название «точные методы»). Итерационными являются методы, в которых точное решение может быть получено лишь в результате бесконечного повторения, как правило, единообразных действий.

Уделим основное внимание задаче вычисления вектора **x**, являющегося решением СЛАУ (2.1). Будем полагать, что матрица **A** задана и является невырожденной ($\det\mathbf{A} \neq 0$). Известно, что в этом случае решение системы существует, единственно и устойчиво по входным данным, т. е. рассматриваемая задача корректна.

На точность и время решения СЛАУ прямым методам сильное влияние оказывает обусловленность матрицы. Чем выше число обусловленности матрицы, тем хуже она обусловлена. С ростом числа обусловленности растет погрешность решения из-за представления чисел с плавающей запятой конечным числом разрядов. Одним из важных следствий этого является невозможность получения корректного решения СЛАУ методом Гаусса при чрезмерном росте $\mathrm{cond}(\mathbf{A})$ матриц больших порядков и малой разрядности представления чисел.

2.2 Прямые методы решения СЛАУ

2.2.1 Метод исключения Гаусса

Задача численного решения СЛАУ вида (2.1) имеет продолжительную историю. Так, до н.э. в Китае были изданы «Девять книг о математическом искусстве», где метод, который теперь принято называть методом исключения Гаусса или просто методом Гаусса, был представлен в «натуральной» форме. Метод имеет долгую и интересную историю, а к его становлению приложило немало усилий большое количество ученых и инженеров. Этот метод известен в различных вариантах, которые алгебраически тождественны, уже более 2000 лет. Варианты отличаются характером хранения матриц, порядком исключения, способами предупреждения больших погрешностей округления и тем, как уточняются вычисленные решения. Имеются также варианты, специально приспособленные для систем с симметричными положительно определенными матрицами, которые хранятся в при-

мерно вдвое меньшем объеме. Вычисления с помощью метода Гаусса включают два основных этапа, называемых прямым и обратным ходом. Прямой ход метода Гаусса состоит в последовательном исключении неизвестных из системы (2.1) для преобразования ее к эквивалентной системе с верхней треугольной матрицей. Вычисление значений неизвестных производится на этапе обратного хода [17].

На практике широкое распространение получила версия метода, основанная на LU-разложении матрица **A**. Рассмотрим ее подробнее.

Пусть $\mathbf{A} = \left(a_{ij}\right)_{i,j=1}^{N}$ – данная $N \times N$-матрица, а $\mathbf{L} = \left(l_{ij}\right)_{i,j=1}^{N}$ и $\mathbf{U} = \left(u_{ij}\right)_{i,j=1}^{N}$ – соответственно нижняя (левая) и верхняя (правая) треугольные матрицы. Справедливо следующее утверждение. Если все главные миноры квадратной матрицы **A** отличны от нуля, то существуют такие нижняя **L** и верхняя **U** треугольные матрицы, что $\mathbf{A} = \mathbf{LU}$ (главными минорами матрицы $\mathbf{A} = \left(a_{ij}\right)_{i,j=1}^{N}$ называются определители подматриц $\mathbf{A}_k = \left(a_{ij}\right)_{i,j=1}^{k}$, где $k = 1, 2, \ldots, N-1$). Если элементы диагонали одной из матриц, **L** или **U**, фиксированы (ненулевые), то такое разложение единственно.

Реализация LU-разложения с фиксированием диагонали верхней треугольной матрицы ($u_{ij} = 1$ при $i = j$) называется методом Краута. Рассмотрим часто используемое на практике разложение матриц при фиксировании диагонали нижней треугольной матрицы ($l_{ij} = 1$ при $i = j$) – метод Дулитла. Находят l_{ij} при $i > j$ ($l_{ij} = 0$ при $i < j$) и u_{ij} при $i \le j$ ($u_{ij} = 0$ при $i > j$) такие, чтобы

$$\begin{pmatrix} 1 & 0 & \ldots & 0 \\ l_{21} & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ l_{N1} & l_{N1} & \ldots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1N} \\ 0 & u_{22} & \ldots & u_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & u_{NN} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ 0 & a_{22} & \ldots & a_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & a_{NN} \end{pmatrix}.$$

Выполнив перемножение матриц, на основе поэлементного приравнивания левых и правых частей приходим к $N \times N$-матрице уравнений

$$\begin{array}{llll} u_{11} = a_{11}, & u_{12} = a_{12}, & \ldots & u_{1N} = a_{1N}, \\ l_{21}u_{11} = a_{21}, & l_{21}u_{12} + u_{22} = a_{22}, & \ldots & l_{21}u_{1N} + u_{2N} = a_{2N}, \\ \ldots & \ldots & \ldots & \ldots \\ l_{N1}u_{11} = a_{N1}, & l_{N1}u_{12} + l_{N2}u_{22} = a_{N2}, & \ldots & l_{N1}u_{1N} + \ldots + u_{NN} = a_{NN}, \end{array}$$

относительно $N \times N$-матрицы неизвестных

$$\begin{array}{llll} u_{11}, & u_{12}, & \ldots & u_{1N}, \\ l_{21}, & lu_{22}, & \ldots & u_{2N}, \\ \ldots & \ldots & \ldots & \ldots \\ l_{N1}, & l_{N2}, & \ldots & u_{NN}. \end{array} \qquad (2.14)$$

Видно, что все отличные от 0 и 1 элементы матриц **L** и **U** могут быть однозначно вычислены с помощью всего двух формул:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}, \text{ где } i \le j, \qquad (2.15)$$

$$l_{ij} = \frac{1}{u_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj} \right), \text{ где } i > j. \qquad (2.16)$$

Из приведенных преобразований следует, что реализовать LU-разложение по данным формулам можно различными методами, например построчным вычислением, т. е. пока не вычислена i-я строка матриц **L** и **U**, алгоритм не модернизирует $(i + 1)$-ю строку.

При практическом выполнении разложения (факторизации) матрицы **A** нужно иметь в виду следующие два обстоятельства.

Во-первых, организация вычислений по формулам (2.15)–(2.16) должна предусматривать переключение счета с одной формулы на другую. Это удобно делать, ориентируясь на матрицу неизвестных (2.14) (ее, кстати, можно интерпретировать как N^2-мерный массив для компактного хранения LU-разложения в памяти компьютера), а именно: первая строка матрицы (2.14) вычисляется по формуле (2.15) при $i = 1$, $j = 1, 2, \ldots, N$; первый столбец

(2.14) (без первого элемента) – по формуле (2.16) при $j = 1$, $i = 2,\ldots, N$, и т. д.

Во-вторых, препятствием для осуществимости описанного процесса LU-разложения матрицы **A** может оказаться равенство нулю диагональных элементов матрицы **U**, поскольку на них выполняется деление в формуле (2.16). Отсюда следует требование, накладываемое на главные миноры. Для определенных классов матриц требования о разложении заведомо выполняются. Это относится, например, к матрицам с преобладанием диагональных элементов.

Далее приведена построчная схема LU-разложения (так называемая *ikj*-версия) без выбора ведущего элемента, которая является, пожалуй, самой предпочтительной для программной реализации. Можно показать, что перестановка трех циклов дает шесть возможных версий разложения.

Алгоритм *ikj*-версии LU-разложения без выбора ведущего элемента (результат хранится на месте исходной матрицы)

Для $i = 2, \ldots, N$
    Для $k = 1, \ldots, i - 1$
        $a_{ik} = a_{ik} / a_{kk}$
        Для $j = k + 1, \ldots, N$
            $a_{ij} = a_{ij} - a_{ik} \cdot a_{kj}$
        Увеличить j
    Увеличить k
Увеличить i

Ниже приведены функции на языке Octave, реализующие разложение по этому алгоритму (листинг 2.1). Общие сведения по разработке программ на языке Octave изложены в приложении А.

```
function A = ikj (A)
 n = size(A,1) ;
for i=1:n
  for k=1:i-1
    A(i,k) = A(i,k)/A(k,k);
    A(i,k+1:n) = A(i,k+1:n) - A(i,k)*A(k,k+1:n);
  end
end
%L = diag(ones(n,1)) + tril(A,-1);
%U = triu(A);
```

Листинг 2.1 – Функция LU-разложения на языке Octave

Этот алгоритм позволяет пересчитать i-ю строку матрицы **A** в i-е строки матриц **L** и **U**. Каждая из строк 1, 2,…, $j-1$ участвует в определении j-х строк матриц **L** и **U**, но сами они больше не модернизируются.

Вообще, существует более десяти вариантов LU-разложения. Иногда данное представление матрицы в виде произведения матриц называют LR-разложением.

Пример 2.5

Найти LU-разложение для матрицы $\mathbf{A} = \begin{pmatrix} 10 & 6 & 2 & 0 \\ 5 & 1 & -2 & 4 \\ 3 & 5 & 1 & 4 \\ 0 & 6 & -2 & 2 \end{pmatrix}$.

Решение

Воспользовавшись формулами (2.15)–(2.16), получим

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0.3 & -1.6 & 1 & 0 \\ 0 & -3 & 2.5 & 1 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} 10 & 6 & 2 & 0 \\ 0 & -2 & -3 & 4 \\ 0 & 0 & -4.4 & 5.4 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}.$$

Если матрица **A** исходной системы **Ax** = **b** разложена в произведение треугольных матриц **L** и **U**, то вместо **Ax** = **b** можно записать

$$\mathbf{LUx} = \mathbf{b}.$$

Введя вектор вспомогательных переменных **y**, последнее выражение можно переписать в виде системы

$$\begin{cases} \mathbf{Ly} = \mathbf{b}, \\ \mathbf{Ux} = \mathbf{y}. \end{cases}$$

Таким образом, решение данной системы с квадратной матрицей коэффициентов свелось к последовательному решению двух систем с треугольными матрицами коэффициентов.

Очевидно, все y_i могут быть найдены из системы **Ly** = **b** при $i = 1, 2, \ldots, N$ по формуле (прямой ход)

$$y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k. \qquad (2.17)$$

Значения неизвестных x_i находятся из системы $\mathbf{Ux} = \mathbf{y}$ в обратном порядке, т.е. при $i = N, N-1,\ldots, 1$, по формуле (обратный ход)

$$x_i = \frac{1}{u_{ii}}\left(y_i - \sum_{k=i+1}^{N} u_{ik} x_k \right). \qquad (2.18)$$

Итак, решение СЛАУ посредством LU-факторизации сводится к организации вычислений по четырем формулам: совокупности формул (2.15), (2.16) для получения матрицы $\mathbf{L} + \mathbf{U} - \mathbf{I}$ (2.14) ненулевых и неединичных элементов матриц для $\mathbf{L}$ и $\mathbf{U}$; формулы (2.17) для получения вектора свободных членов треугольной системы $\mathbf{Ux} = \mathbf{y}$; формулы (2.18), генерирующей решение исходной системы $\mathbf{Ax} = \mathbf{b}$.

Для обращения матрицы $\mathbf{A}$ с помощью LU-разложения можно N-кратно использовать формулы (2.17) и (2.18) для получения столбцов матрицы $\mathbf{A}^{-1}$; при этом в качестве b_i в формуле (2.17) должны фигурировать только 0 и 1: для нахождения первого столбца полагают $b_1 = 1$, $b_2 = 0$, $b_3 = 0$, …, $b_N = 0$; для второго – $b_1 = 0$, $b_2 = 1$, $b_3 = 0$, …, $b_N = 0$ и т. д.

В результате после N шагов на месте исходной матрицы $\mathbf{A}$ находится ее обратная матрица $\mathbf{A}^{-1}$.

2.2.2 Метод прогонки

Метод прогонки предложен в начале 1950-х гг. независимо несколькими авторами, в том числе советскими. Это эффективный алгоритм решения СЛАУ с трехдиагональными матрицами вида

$$\begin{aligned}
b_1x_1 &+ c_1x_2 &&&&&&&= d_1,\\
a_2x_1 &+ b_2x_2 &+ c_2x_3 &&&&&&= d_2,\\
&\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots &&&&&&&\\
&a_ix_{i-1} + b_ix_i + c_ix_{i+1} &&&&&&= d_i,\\
&\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots &&&&&&&\\
&a_{N-1}x_{N-2} + b_{N-1}x_{N-1} + c_{N-1}x_N &&&&&&= d_{N-1},\\
&a_Nx_{N-1} + b_Nx_N &&&&&&= d_N.
\end{aligned} \qquad (2.19)$$

Системы такого вида часто возникают при решении различных задач математической физики.

Преобразуем первое уравнение (2.19) к виду

$$x_1 = \alpha_1 x_2 + \beta_1, \qquad (2.20)$$

где $\alpha_1 = -c_1 / b_1$; $\beta_1 = d_1 / b_1$.

Подставим выражение для x_1 во второе уравнение системы:

$$a_2(\alpha_1 x_2 + \beta_1) + b_2 x_2 + c_2 x_3 = d_2.$$

Преобразуем это уравнение к виду

$$x_2 = \alpha_2 x_3 + \beta_2, \qquad (2.21)$$

где $\alpha_2 = -c_2 / (b_2 + a_2\alpha_1)$; $\beta_2 = (d_2 - a_2\beta_1) / (b_2 + a_2\alpha_1)$. Это выражение подставим в третье уравнение системы и т. д. На i-м шаге ($1 < i < N$) i-е уравнение системы преобразуется к виду

$$x_i = \alpha_i x_i + 1 + \beta i, \qquad (2.22)$$

где $\alpha_i = -c_i / (b_i + a_i\alpha_{i-1})$; $\beta_i = (d_i - a_i\beta_{i-1}) / (b_i + a_i\alpha_{i-1})$.

На N-м шаге подстановка в последнее уравнение выражения $x_{N-1} = \alpha_N x_{N+1} + \beta_{N-1}$ дает

$$a_N(\alpha_{N-1} x_N + \beta_{N-1}) + b_N x_N = d_N,$$

откуда можно определить

$$x_N = \beta_N = \left(d_N - a_N\beta_{N-1}\right)\big/\left(b_N + a_N\alpha_{N-1}\right).$$

Значения остальных неизвестных x_i для $i = N - 1, N - 2, \ldots, 1$ теперь легко вычисляются по формуле (2.22).

Сделанные преобразования позволяют организовать вычисления методом прогонки в два этапа.

Прямой ход (прямая прогонка) состоит в вычислении прогоночных коэффициентов α_i ($1 \le i < N$) и β_i ($1 \le i < N$). При $i = 1$ коэффициенты вычисляют по формулам

$$\alpha_1 = -c_1 / \gamma_1, \quad \beta_1 = d_1 / \gamma_1, \quad \gamma_1 = b_1, \qquad (2.23)$$

а при $i = 2, 3, \ldots, N - 1$ – по рекуррентным формулам

$$\alpha_i = -c_i / \gamma_i, \quad \beta_i = (d_i - a_i\beta_{i-1}) / \gamma_i, \quad \gamma_i = b_i + a_i\alpha_{i-1}. \qquad (2.24)$$

При $i = N$ прямая прогонка завершается вычислениями

$$\beta_N = \left(d_N - a_N\beta_{N-1}\right)\big/\gamma_N, \quad \gamma_N = b_N + a_N\alpha_{N-1}. \qquad (2.25)$$

Обратный ход метода прогонки (обратная прогонка) дает значения неизвестных. Сначала полагают $x_N = \beta_N$. Затем значения остальных неизвестных вычисляют по формуле

$$x_i = \alpha_i x_{i+1} + \beta_i, \quad i = N-1, N-2, \ldots, 1. \tag{2.26}$$

Вычисления выполняют в порядке убывания значений i от $N-1$ до 1.

Пример 2.6

С помощью метода прогонки решить СЛАУ

$$\begin{aligned} 5x_1 - x_2 &= 2.0, \\ 2x_1 + 4.6x_2 - x_3 &= 3.3, \\ 2x_2 + 3.6x_3 - 0.8x_4 &= 2.6, \\ 3x_3 + 4.4x_4 &= 7.2. \end{aligned}$$

Решение

Прямой ход. Согласно формулам (2.23)–(2.25) получим:

$$\gamma_1 = b_1 = 5, \ \alpha_1 = -c_1 / \gamma_1 = 1 / 5 = 0.2, \ \beta_1 = d_1 / \gamma_1 = 2.0 / 5 = 0.4;$$

$$\gamma_2 = b_2 + a_2\alpha_1 = 4.6 + 2 \cdot 0.2 = 5, \ \alpha_2 = -c_2 / \gamma_2 = 1 / 5 = 0.2;$$

$$\beta_2 = (d_2 - a_2\beta_1) / \gamma_2 = (3.3 - 2 \cdot 0.4) / 5 = 0.5;$$

$$\gamma_3 = b_3 + a_3\alpha_2 = 3.6 + 2 \cdot 0.2 = 4, \ \alpha_3 = -c_3 / \gamma_3 = 0.8 / 4 = 0.2;$$

$$\beta_3 = (d_3 - a_3\beta_2) / \gamma_3 = (2.6 - 2 \cdot 0.5) / 4 = 0.4;$$

$$\gamma_4 = b_4 + a_4\alpha_3 = 4.4 + 3 \cdot 0.2 = 5;$$

$$\beta_4 = (d_4 - a_4\beta_3) / \gamma_4 = (7.2 - 3 \cdot 0.4) /5 = 1.2.$$

Обратный ход. Полагаем $x_4 = \beta_4 = 1.2$. Далее находим:

$$x_3 = \alpha_3 x_4 + \beta_3 = 0.2 \cdot 1.2 + 0.4 = 0.64;$$

$$x_2 = \alpha_2 x_3 + \beta_2 = 0.2 \cdot 0.64 + 0.5 = 0.628;$$

$$x_1 = \alpha_1 x_2 + \beta_1 = 0.2 \cdot 0.628 + 0.4 = 0.5256.$$

Итак, найденное решение: $x_1 = 0.5256$, $x_2 = 0.628$, $x_3 = 0.64$, $x_4 = 1.2$.

Непосредственный подсчет показывает, что для реализации вычислений по формулам (2.23)–(2.26) требуется примерно $8N$ арифметических операций, тогда как в методе Гаусса это число составляет примерно $(2/3)N^3$. Важно и то, что трехдиагональная

структура матрицы системы позволяет использовать для ее хранения $3N - 2$ машинных слова.

Таким образом, при одной и той же производительности и оперативной памяти компьютера метод прогонки позволяет решать системы гораздо большей размерности, чем стандартный метод Гаусса для систем уравнений с плотной (заполненной) матрицей.

О других прямых методах можно узнать, например, из [18].

2.2.3 Многократное решение СЛАУ

Часто необходимы многовариантный анализ или оптимизация рассматриваемого объекта (явления) в диапазоне изменения его параметров (изменение частоты воздействующего сигнала, учет частотной зависимости или разброса параметров структуры) с целью получения набора параметров, используемых для дальнейшего моделирования. Нередко такой анализ в САПР выполняется в интерактивном режиме, что соответствует изменению параметров математической модели. В этом случае временные затраты возрастают из-за необходимости решения (многократного) последовательности СЛАУ вида

$$\mathbf{A}_k\mathbf{X}_k = \mathbf{B}_k, \quad k = 1, 2, \ldots, m, \tag{2.27}$$

где $\mathbf{B} = [\mathbf{b}_1| \ \ldots \ |\mathbf{b}_s]$, $s << N$.

При неизменной матрице и нескольких правых частях используется LU-разложение. Если изменения (вариации) в матрицах СЛАУ незначительны и структурированы (меняется только малое количество строк и столбцов), а правая часть постоянна, то применим метод на основе формулы Шермана – Моррисона – Вудбери (метод окаймления [18]), а также схожие с ним, основанные на корректировке обратной матрицы. Вообще, формула Шермана – Моррисона – Вудбери использовалась и используется для исправления элементов обратной матрицы из-за ошибок округления и неточности задания исходных данных. Для СЛАУ с ленточной матрицей также разработаны эффективные алгоритмы. К сожалению, данные методы неприменимы для решения СЛАУ (2.27) при неструктурированных различиях в матрицах.

В практических приложениях при использовании метода моментов обусловленность матрицы СЛАУ ухудшается с приближением частоты воздействия к частотам собственных колебаний (внутренним резонансам) исследуемой структуры. Матрица таких структур на частотах собственных колебаний становится вырожденной, и ее определитель равен или близок к нулю. Поскольку частота воздействия выбирается дискретно, то существует вероятность такого совпадения. Решение на этой частоте может быть вовсе не получено, а на близких частотах оказывается неточным или некорректным, например показывает наличие мнимых резонансов. Для решения данной проблемы разработано несколько подходов различной степени кардинальности [19].

Предположим, что после того как было найдено решение $\mathbf{x}$ системы $\mathbf{Ax} = \mathbf{b}$, возникла необходимость решить систему $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \mathbf{b}$ с матрицей, отличающейся от матрицы $\mathbf{A}$ несколькими элементами. Например, могло выясниться, что заданные ранее элементы содержали грубые ошибки. Возможно также, что решаемая задача такова, что в ней элементы матрицы последовательно меняются, а правая часть остается неизменной.

Разумеется, решение $\tilde{\mathbf{x}}$ можно найти, решив систему снова и проигнорировав полученную на предыдущем этапе информацию. Однако в рассматриваемом случае можно найти поправку $\Delta\mathbf{x} = \tilde{\mathbf{x}} - \mathbf{x}$ к найденному ранее решению, используя всего $O(N^2)$ операций.

Пусть $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{u}\upsilon^T$, где $\mathbf{u}$ и υ – векторы размерностью N. Тогда справедлива формула Шермана – Моррисона – Вудбери [20]

$$\tilde{\mathbf{A}}^{-1} = \mathbf{A}^{-1} - \alpha\left(\mathbf{A}^{-1}\mathbf{u}\right)\left(\upsilon^T\mathbf{A}^{-1}\right),$$

где $\alpha = 1\Big/\left(1 - \upsilon^T\mathbf{A}^{-1}\mathbf{u}\right)$. Для ее применения необходимо выполнить следующую последовательность действий:

– решить СЛАУ $\mathbf{Ay} = \mathbf{u}$ относительно $\mathbf{y}$;

– решить СЛАУ $\mathbf{A}^T\mathbf{z} = \upsilon$ относительно $\mathbf{z}$;

– вычислить $\alpha = 1 / (1 + \upsilon^T\mathbf{y})$, $\beta = \mathbf{z}^T\mathbf{b}$ и $\Delta\mathbf{x} = \alpha\beta\mathbf{y}$;

– вычислить $\tilde{\mathbf{x}} = \mathbf{x} - \Delta\mathbf{x}$.

Суммарное число операций будет действительно составлять $O(N^2)$, если для решения систем $\mathbf{Ay} = \mathbf{u}$ и $\mathbf{A}^T\mathbf{z} = \upsilon$ применить обратную подстановку с использованием LU-разложения матрицы $\mathbf{A}$, найденного ранее на этапе решения $\mathbf{Ax} = \mathbf{b}$.

Когда матрица $\tilde{\mathbf{A}}$ отличается от матрицы $\mathbf{A}$ только одним элементом $\tilde{a}_{ij} = a_{ij} + \Delta a_{ij}$, можно положить $\mathbf{u} = \Delta a_{ij}\mathbf{e}_i$ и $\upsilon = \mathbf{e}_j$. Тогда последовательность действий примет вид:

– решить СЛАУ $\mathbf{Ay} = \Delta a_{ij}\mathbf{e}_i$ относительно $\mathbf{y}$;

– решить СЛАУ $\mathbf{A}^T\mathbf{z} = \mathbf{e}_j$ относительно $\mathbf{z}$;

– вычислить $\alpha = 1 / (1 - y_j)$, $\beta = \mathbf{z}^T\mathbf{b}$ и $\Delta\mathbf{x} = \alpha\beta\mathbf{y}$;

– вычислить $\tilde{\mathbf{x}} = \mathbf{x} + \Delta\mathbf{x}$.

Основываясь на формуле Шермана – Моррисона, можно указать способ пересчета LU-разложения матрицы [21].

Пример 2.7

Дано

$$\mathbf{A}_1\mathbf{x}_1 = \mathbf{b},\ \mathbf{A}_2\mathbf{x}_2 = \mathbf{b} \text{ и } \mathbf{A}_2 = \mathbf{A}_1 + \mathbf{u}\upsilon^T,$$

где $\mathbf{A}_1 = \begin{pmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 5 & 3 & 4 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 3 \\ -1 \\ 6 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \upsilon^T = [2\ 3\ 7]$. Найти $\mathbf{x}_2$, используя формулу Шермана – Моррисона – Вудбери.

Решение

Результатом LU-разложения матрицы $\mathbf{A}_1$ будет

$$\mathbf{L}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0.333 & 1 & 0 \\ 1.667 & -0{,}143 & 1 \end{pmatrix},\ \mathbf{U}_1 = \begin{pmatrix} 3 & 2 & 2 \\ 0 & 2.333 & 0.333 \\ 0 & 0 & 0.714 \end{pmatrix},$$

а решением – $\mathbf{x}_1 = (1, -1, 1)^T$. Далее, решив системы $\mathbf{A}_1\mathbf{y} = \mathbf{u}$ и $\mathbf{A}_1{}^T\mathbf{z} = \upsilon$, где $\upsilon = \mathbf{A}^T = (\mathbf{LU})^T = \mathbf{U}^T\mathbf{L}^T$, получим $\mathbf{y} = (-0.4, 0.4, 0.2)^T$ и $\mathbf{z} = (-12.6, 1.8, 7.6)^T$.

Тогда $\upsilon^T\mathbf{y} = 1.8$, $\alpha = 0.357$ и $\beta = 6$. В результате $\Delta\mathbf{x} = (-0.857, 0.857, 0.429)^T$ и $\mathbf{x}_2 = (1.857, -1.857, 0.571)^T$.

2.3 Итерационные методы решения СЛАУ

2.3.1 Особенности итерационных методов

Развитие вычислительной техники и вызванный этим процессом переход ко все более сложным моделям привели к необходимости решения больших СЛАУ. Точные методы понятны и просты для программной реализации, однако их вычислительные затраты серьезно ограничивают круг рассматриваемых проблем. Поэтому итерационные методы получили широкое распространение при решении различных прикладных и научных задач.

Одной из главных особенностей итерационных методов является то, что получаемая погрешность решения из-за конечного числа разрядов много меньше, чем в методе Гаусса, так как она не накапливается, а определяется только последней итерацией и не зависит от их числа. Поэтому решение с заданной точностью при росте числа обусловленности матрицы достигается просто увеличением числа итераций. Очевидно, что снижение числа обусловленности матрицы позволяет уменьшить время решения с требуемой точностью за счет уменьшения числа итераций. Также можно уменьшить количество разрядов представления чисел с плавающей запятой для снижения вычислительных затрат. Такая опциональность для выбора пользователя реализована, например, в FEKO [5]. Применение итерационных методов является эффективным способом решения СЛАУ с плохо обусловленной матрицей. Кроме того, итерационные методы приемлемы и при хорошо обусловленной матрице. Так, метод моментов дает СЛАУ с плотной матрицей, для решения которой традиционно используются точные методы, например метод Гаусса. Однако их вычислительные затраты $\sim N^3$ (N – порядок матрицы), что существенно ограничивает круг рассматриваемых задач даже при использовании высокоскоростных компьютеров. Практика же диктует необходимость решения сложных задач с постоянно увеличивающимися порядками СЛАУ. Поскольку в основе каждой итерации лежит умножение матрицы на вектор, то затраты итерационных методов (без предобусловливания) $\sim N_{it} \cdot N^2$, где N^2 – вычислительные

затраты на итерацию; N_{it} – число итераций, необходимых для сходимости. Если по грубой эмпирической оценке $N_{it} \approx N^{0,5}$, тогда выигрыш по сравнению с методом Гаусса также $\approx N^{0,5}$. Однако ряд разработанных математических подходов к ускорению процедуры умножения матрицы на вектор может уменьшить вычислительные затраты с $\sim N^2$ до $\sim N \log_2 N$, позволяя получить значительный дополнительный выигрыш.

Итерационные методы имеют долгую и интересную историю развития, как и прямые, и первые из них также связаны с именем К. Ф. Гаусса. Ставшие уже классическими методы Гаусса – Зейделя, Якоби, Ричардсона, релаксации и другие, основанные на расщеплении матрицы СЛАУ, редко применяются на практике из-за их медленной или неустойчивой сходимости и невозможности применения предобусловливания. Исключение составляют их блочные или параллельные версии. В целом эти методы являются отличным началом для освещения общих принципов построения итерационного процесса в учебных дисциплинах по численным методам и формированию предобусловливателей.

Наиболее эффективными и устойчивыми среди итерационных методов являются так называемые проекционные методы, и особенно тот их класс, который связан с проектированием на подпространства Крылова. Эти методы обладают рядом достоинств: они устойчивы, допускают эффективное распараллеливание и работу с предобусловливателями разных типов. В 1906 г. в своих первых «Лекциях о приближенных вычислениях» (неоднократно переиздававшихся) Алексей Николаевич Крылов заложил принципы вычислительной математики и по праву считается ее основателем.

Исторически итерационные методы разрабатывались для решения разреженных СЛАУ высокого порядка. Главный источник таких СЛАУ – сеточные методы (конечных разностей, конечных элементов и др.) решения многомерных краевых задач.

Первые итерационные методы основывались на циклическом покомпонентном изменении вектора решения, осуществляемом таким образом, чтобы обнулить соответствующий коэффициент вектора невязки и тем самым уменьшить его норму. Подобная ме-

тодика уточнения решения получила название «релаксация». Хотя в настоящее время такие методы в их классической формулировке уже практически не применяются, существуют определенные классы задач, для которых разработаны их модификации, хорошо себя зарекомендовавшие. Кроме того, как будет показано далее, эти методы могут быть применены не в качестве самостоятельного средства решения СЛАУ, а для предобусловливания.

2.3.2 Методы Якоби и Гаусса – Зейделя

Пусть матрица **A** системы **Ax** = **b** такова, что ее главная диагональ не содержит нулевых элементов. Представим ее в виде разности

$$\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}, \qquad (2.28)$$

где матрица **D** содержит диагональные элементы матрицы **A**; матрица **E** – только поддиагональные; матрица **F** – только наддиагональные. Тогда система **Ax** = **b** может быть записана в виде

$$\mathbf{Dx} - \mathbf{Ex} - \mathbf{Fx} = \mathbf{b}.$$

Если имеется приближение $\mathbf{x}_k$ к точному решению СЛАУ $\mathbf{x}_*$, то при $\mathbf{x}_k \neq \mathbf{x}_*$ это соотношение не выполняется. Однако, если в выражении

$$\mathbf{Dx}_k - \mathbf{Ex}_k - \mathbf{Fx}_k = \mathbf{b} \qquad (2.29)$$

одно или два из вхождений вектора $\mathbf{x}_k$ заменить на $\mathbf{x}_{k+1}$ и потребовать, чтобы равенство имело место, можно получить некоторую вычислительную схему для уточнения решения.

Наиболее простой с точки зрения объема вычислений вариант получается при замене в уравнении (2.29) $\mathbf{Dx}_k$ на $\mathbf{Dx}_{k+1}$. При этом получается схема

$$\mathbf{x}_{k+1} = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})\mathbf{x}_k + \mathbf{D}^{-1}\mathbf{b}, \qquad (2.30)$$

известная как метод Якоби (еще называемая методом Гаусса – Якоби). Метод известен с 1840-х гг.

Выражение (2.30) в скалярной форме имеет вид

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=k+1}^{N} a_{ij} x_j^{(k)} \right), \; i = 1,\ldots,N, \qquad (2.31)$$

откуда хорошо видна основная идея метода: на (k+1)-й итерации i-й компонент вектора решения изменяется по сравнению с k-й итерацией так, чтобы i-й компонент вектора невязки $\mathbf{r}_{k+1}$ стал нулевым (при условии отсутствия изменений в других компонентах вектора $\mathbf{x}$). Далее приведем алгоритм метода Якоби.

Алгоритм метода Якоби

Выбрать произвольное начальное приближение $\mathbf{x}^{(0)}$
Для $k = 1, 2, \ldots$
    Для $i = 1, \ldots, N$
        $\tilde{x}_i = 0$
        Для $j = 1, \ldots, i-1, i+1, \ldots, N$
            $\tilde{x}_i = \tilde{x}_i + a_{ij} x_j^{(k-1)}$
        Увеличить j
        $\tilde{x}_i = (b_i - \tilde{x}_i) / a_{ii}$
    Увеличить i
    $\mathbf{x}^{(k)} = \tilde{\mathbf{x}}$
    Проверить сходимость и продолжить при необходимости
Увеличить k

Пример 2.8

С помощью метода Якоби решить СЛАУ $\begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 11 \end{pmatrix}$.

Решение

Пусть $\mathbf{x}^{(0)} = (1, 1)^T$, тогда получим следующий результат.

Первая итерация	$x_1^{(1)} = (b_1 - a_{12}x_2^{(0)})/a_{11} =$ $= (4 - 1\cdot 1) / 2 = 1.5$	$x_2^{(1)} = (b_2 - a_{21}x_1^{(0)})/a_{22} =$ $= (11 - 3\cdot 1) / 4 = 2$
Вторая итерация	$x_1^{(2)} = (b_1 - a_{12}x_2^{(1)})/a_{11} =$ $= (4 - 1\cdot 2) / 2 = 1$	$x_2^{(2)} = (b_2 - a_{21}x_1^{(1)})/a_{22} =$ $= (11 - 3\cdot 1.5) / 4 = 1.625$
Третья итерация	$x_1^{(3)} = (b_1 - a_{12}x_2^{(2)})/a_{11} =$ $= (4 - 1\cdot 1.625) / 2 = 1.1875$	$x_2^{(3)} = (b_2 - a_{21}x_1^{(2)})/a_{22} =$ $= (11 - 3\cdot 1) / 4 = 2$
Четвертая итерация	$x_1^{(4)} = (b_1 - a_{12}x_2^{(3)})/a_{11} =$ $= (4 - 1\cdot 2) / 2 = 1$	$x_2^{(4)} = (b_2 - a_{21}x_1^{(3)})/a_{22} =$ $= (11 - 3\cdot 1.1875) / 4 \approx 1.9063$
Пятая итерация	$x_1^{(5)} = (b_1 - a_{12}x_2^{(4)})/a_{11} =$ $= (4 - 1\cdot 1.9063) / 2 \approx 1.0469$	$x_2^{(5)} = (b_2 - a_{21}x_1^{(4)})/a_{22} =$ $= (11 - 3\cdot 1) / 4 = 2$
Шестая итерация	$x_1^{(6)} = (b_1 - a_{12}x_2^{(5)})/a_{11} =$ $= (4 - 1\cdot 2) / 2 = 1$	$x_2^{(6)} = (b_2 - a_{21}x_1^{(5)})/a_{22} =$ $= (11 - 3\cdot 1.0469) / 4 \approx 1.9648$

Седьмая итерация	$x_1^{(7)} = (b_1 - a_{12}x_2^{(6)})/a_{11} =$ $= (4 - 1\cdot 1.9648) / 2 \approx 1.0176$	$x_2^{(7)} = (b_2 - a_{21}x_1^{(6)})/a_{22} =$ $= (11 - 3\cdot 1) / 4 = 2$
Восьмая итерация	$x_1^{(8)} = (b_1 - a_{12}x_2^{(7)})/a_{11} =$ $= (4 - 1\cdot 2) / 2 = 1$	$x_2^{(8)} = (b_2 - a_{21}x_1^{(7)})/a_{22} =$ $= (11 - 3\cdot 1.0176) / 4 \approx 1.9868$
Девятая итерация	$x_1^{(9)} = (b_1 - a_{12}x_2^{(8)})/a_{11} =$ $= (4 - 1\cdot 1.9868) / 2 \approx 1.0061$	$x_2^{(9)} = (b_2 - a_{21}x_1^{(8)})/a_{22} =$ $= (11 - 3\cdot 1)/4 = 2$
Десятая итерация	$x_1^{(10)} = (b_1 - a_{12}x_2^{(9)})/a_{11} =$ $= (4 - 1\cdot 2) / 2 = 1$	$x_2^{(10)} = (b_2 - a_{21}x_1^{(9)})/a_{22} =$ $= (11 - 3\cdot 1.0061) / 4 \approx 1.9954$

Недостатком схемы (2.30)–(2.31) является то, что при нахождении компонента $x_i^{(k+1)}$ никак не используется информация о пересчитанных компонентах $x_1^{(k+1)},\ldots, x_{i-1}^{(k+1)}$. Исправить этот недостаток можно, переписав выражение (2.31) в виде

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=k+1}^{N} a_{ij}x_j^{(k)} \right), \quad i = 1,\ldots,N, \qquad (2.32)$$

что в векторной форме эквивалентно

$$\mathbf{x}_{k+1} = (\mathbf{D} - \mathbf{E})^{-1}\mathbf{F}\mathbf{x}_k + (\mathbf{D} - \mathbf{E})^{-1}\mathbf{b}. \qquad (2.33)$$

Такая схема называется методом Гаусса – Зейделя (иногда его называют просто методом Зейделя), который известен с 1870-х гг. Приведем алгоритм метода Гаусса – Зейделя.

Алгоритм метода Гаусса – Зейделя

Выбрать произвольное начальное приближение $\mathbf{x}^{(0)}$
Для $k = 1, 2, \ldots$
  Для $i = 1, \ldots, N$
    $\sigma = 0$
    Для $j = 1, \ldots, i - 1$
      $\sigma = \sigma + a_{ij}x_j^{(k)}$
    Увеличить j
    Для $j = i + 1, \ldots, N$
      $\sigma = \sigma + a_{ij}x_j^{(k-1)}$
    Увеличить j
    $x_i^{(k)} = (b_i - \sigma) / a_{ii}$
  Увеличить i
  Проверить сходимость и продолжить при необходимости
Увеличить k

Пример 2.9

С помощью метода Якоби решить СЛАУ из примера 2.8.

Решение

Пусть $\mathbf{x}^{(0)} = (1, 1)^T$, тогда получим следующий результат.

Первая итерация	$x_1^{(1)} = (b_1 - a_{12}x_2^{(0)})/a_{11} =$ $= (4 - 1\cdot 1) / 2 = 1.5$	$x_2^{(1)} = (b_2 - a_{21}x_1^{(1)})/a_{22} =$ $= (11 - 3\cdot 3 / 2) / 4 = 1.625$
Вторая итерация	$x_1^{(2)} = (b_1 - a_{12}x_2^{(1)})/a_{11} =$ $= (4 - 1\cdot 1.625) / 2 = 1.1875$	$x_2^{(2)} = (b_2 - a_{21}x_1^{(2)})/a_{22} =$ $= (11 - 3\cdot 1.1875)/4 \approx 1.9063$
Третья итерация	$x_1^{(3)} = (b_1 - a_{12}x_2^{(2)})/a_{11} =$ $= (4 - 1\cdot 1.9063) / 2 \approx 1.0469$	$x_2^{(3)} = (b_2 - a_{21}x_1^{(3)})/a_{22} =$ $= (11 - 3\cdot 1.0469) / 4 \approx 1.9648$
Четвертая итерация	$x_1^{(4)} = (b_1 - a_{12}x_2^{(3)})/a_{11} =$ $= (4 - 1\cdot 1.9648) / 2 \approx 1.0176$	$x_2^{(4)} = (b_2 - a_{21}x_1^{(4)})/a_{22} =$ $= (11 - 3\cdot 1.0176) / 4 \approx 1.9868$
Пятая итерация	$x_1^{(5)} = (b_1 - a_{12}x_2^{(4)})/a_{11} =$ $= (4 - 1\cdot 1.9868) / 2 \approx 1.0061$	$x_2^{(5)} = (b_2 - a_{21}x_1^{(5)})/a_{22} =$ $= (11 - 3\cdot 1.0061) / 4 \approx 1.9954$
Шестая итерация	$x_1^{(6)} = (b_1 - a_{12}x_2^{(5)})/a_{11} =$ $= (4 - 1\cdot 1.9954) / 2 \approx 1.0023$	$x_2^{(6)} = (b_2 - a_{21}x_1^{(6)})/a_{22} =$ $= (11 - 3\cdot 1.0023) / 4 \approx 1.9983$
Седьмая итерация	$x_1^{(7)} = (b_1 - a_{12}x_2^{(6)})/a_{11} =$ $= (4 - 1\cdot 1.9983) / 2 \approx 1.0009$	$x_2^{(7)} = (b_2 - a_{21}x_1^{(7)})/a_{22} =$ $= (11 - 3\cdot 1.0009) / 4 \approx 1.9994$

Выражение (2.32) получается из (2.29) заменой $\mathbf{x}_k$ на $\mathbf{x}_{k+1}$ при матрицах **D** и **E**. Если вместо **D** и **E** взять **D** и **F**, то получится похожая схема

$$\mathbf{x}_{k+1} = (\mathbf{D} - \mathbf{F})^{-1}\mathbf{E}\mathbf{x}_k + (\mathbf{D} - \mathbf{F})^{-1}\mathbf{b}, \qquad (2.34)$$

которая называется обратным методом Гаусса – Зейделя.

Еще одной модификацией является симметричный метод Гаусса – Зейделя, который заключается в циклическом чередовании формул (2.33) и (2.34) на соседних итерациях.

Заметим, что выражения (2.30), (2.33) и (2.34), могут быть записаны в виде

$$\mathbf{K}\mathbf{x}_{k+1} = \mathbf{R}\mathbf{x}_k + \mathbf{b}, \qquad (2.35)$$

где матрицы **K** и **R** связаны соотношением

$$\mathbf{A} = \mathbf{K} - \mathbf{R}. \qquad (2.36)$$

Такое представление матрицы **A** называется расщеплением, а методы вида (2.35) – методами, основанными на расщеплении. Очевидно, что матрица **K** должна быть невырожденной и легко обратимой (т.е. число затрачиваемых арифметических операций должно быть как можно меньше).

2.3.3 Релаксационные методы

Скорость сходимости методов, основанных на расщеплении, непосредственно связана со спектральным радиусом матрицы (максимальное по абсолютной величине собственное число) $\mathbf{K}^{-1}\mathbf{R}$; с другой стороны, выбор **K** ограничен требованием легкой обратимости. Одним из распространенных способов улучшения сходимости является введение дополнительного параметра. Пусть ω – некоторое вещественное число. Рассмотрим вместо системы $\mathbf{Ax} = \mathbf{b}$ масштабированную систему

$$\omega\mathbf{Ax} = \omega\mathbf{b}, \tag{2.37}$$

и вместо выражения (2.28) воспользуемся представлением

$$\omega\mathbf{A} = (\mathbf{D} - \omega\mathbf{E}) - \big(\omega\mathbf{F} + (1-\omega)\mathbf{D}\big), \tag{2.38}$$

где матрицы **D**, **E**, **F** имеют тот же смысл, что и в (2.28).

Тогда на основании соотношений (2.37) и (2.38) можно построить итерационную схему, похожую на метод Гаусса – Зейделя:

$$(\mathbf{D} - \omega\mathbf{E})\mathbf{x}_{k+1} = (\omega\mathbf{F} + (1-\omega)\mathbf{D})\mathbf{x}_k + \omega\mathbf{b}. \tag{2.39}$$

Эта схема называется методом последовательной верхней релаксации (SOR). Для нее

$$\mathbf{K}_{\mathrm{SOR}}(\omega) = \mathbf{D} - \omega\mathbf{E}\ ,$$

$$\mathbf{R}_{\mathrm{SOR}}(\omega) = \omega\mathbf{F} + (1-\omega)\mathbf{D}.$$

Выбор параметра ω, минимизирующего спектральный радиус, является, вообще говоря, достаточно сложной проблемой. Но для многих классов матриц такая задача исследована и оптимальные значения известны. Приведем алгоритм метода последовательной верхней релаксации.

Алгоритм метода последовательной верхней релаксации (SOR)

Выбрать произвольное начальное приближение $\mathbf{x}^{(0)}$
Для $k = 1, 2, \ldots$
    Для $i = 1, \ldots, N$
        $\sigma = 0$
        Для $j = 1, \ldots, i - 1$
            $\sigma = \sigma + a_{ij} x_j^{(k)}$
        Увеличить j
        Для $j = i + 1, \ldots, N$
            $\sigma = \sigma + a_{ij} x_j^{(k-1)}$
        Увеличить j
        $\sigma = (b_i - \sigma) / a_{ii}$
        $x_i^{(k)} = x_i^{(k-1)} + \omega(\sigma - x_i^{(k-1)})$
    Увеличить i
    Проверить сходимость и продолжить при необходимости
Увеличить k

Выражение (2.38) остается тождеством, если в нем поменять местами матрицы **E** и **F**. Такая перестановка дает обратный метод последовательной верхней релаксации:

$$(\mathbf{D} - \omega\mathbf{F})\mathbf{x}_{k+1} = [\omega\mathbf{E} + (1 - \omega)\mathbf{D}]\mathbf{x}_k + \omega\mathbf{b}.$$

Последовательное применение прямого и обратного методов SOR является симметричным методом последовательной верхней релаксации (SSOR):

$$(\mathbf{D} - \omega\mathbf{E})\mathbf{x}_{k+1/2} = [\omega\mathbf{F} + (1 - \omega)\mathbf{D}]\mathbf{x}_k + \omega\mathbf{b};$$

$$(\mathbf{D} - \omega\mathbf{F})\mathbf{x}_{k+1} = [\omega\mathbf{E} + (1 - \omega)\mathbf{D}]\mathbf{x}_{k+1/2} + \omega\mathbf{b}.$$

Приведем алгоритм метода симметричной последовательной верхней релаксации.

Алгоритм метода симметричной последовательной верхней релаксации (SSOR)

Выбрать произвольное начальное приближение $\mathbf{x}^{(0)}$
$\mathbf{x}^{(1/2)} = \mathbf{x}^{(0)}$
Для $k = 1, 2, \ldots$
    Для $i = 1, \ldots, N$
        $\sigma = 0$
        Для $j = 1, \ldots, i - 1$

$\sigma = \sigma + a_{ij} x_j^{(k-1/2)}$

Увеличить j

Для $j = i + 1, \ldots, N$

$\sigma = \sigma + a_{ij} x_j^{(k-1)}$

Увеличить j

$\sigma = (b_i - \sigma) / a_{ii}$

$x_i^{(k-1/2)} = x_i^{(k-1)} + \omega(\sigma - x_i^{(k-1)})$

Увеличить i

Для $i = N, N - 1, \ldots, 1$

$\sigma = 0$

Для $j = 1, \ldots, i - 1$

$\sigma = \sigma + a_{ij} x_j^{(k-1/2)}$

Увеличить j

Для $j = i + 1, \ldots, N$

$\sigma = \sigma + a_{ij} x_j^{(k)}$

Увеличить j

$x_i^{(k)} = x_i^{(k-1/2)} + \omega(\sigma - x_i^{(k-1/2)})$

Проверить сходимость и продолжить при необходимости

Увеличить k

2.3.4 Методы крыловского типа

Рассмотрим систему (2.1) и сформируем для нее следующую задачу. Пусть заданы два подпространства $K \subset R^N$ и $L \subset R^N$. Требуется найти такой вектор $\mathbf{x} \in K$, который обеспечит решение системы (2.1), оптимальное относительно подпространства L, т. е. будет выполняться условие Петрова – Галеркина

$$\forall\, \mathbf{l} \in L\colon (\mathbf{Ax}, \mathbf{l}) = (\mathbf{b}, \mathbf{l})$$

или

$$\forall \mathbf{l} \in L\colon (\mathbf{r}_\mathbf{x}, \mathbf{l}) = 0 \Rightarrow \mathbf{r}_\mathbf{x} = \mathbf{b} - \mathbf{Ax} \perp L. \qquad (2.40)$$

Такая задача называется задачей проектирования $\mathbf{x}$ на подпространство K ортогонально к подпространству L. В более общей постановке задача формулируется следующим образом. Пусть известно некоторое приближение $\mathbf{x}_0$ к точному решению $\mathbf{x}_*$ системы (2.1) и требуется уточнить его поправку $\boldsymbol{\delta}_\mathbf{x} \in K$ таким образом, чтобы $\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \boldsymbol{\delta}_\mathbf{x}) \perp L$. Тогда условие (2.40) примет вид

$$\forall\, \mathbf{l} \in L\colon (\mathbf{r}_{\mathbf{x}_0 + \boldsymbol{\delta}_x}, \mathbf{l}) = ((\mathbf{b} - \mathbf{Ax}_0) - \mathbf{A}\boldsymbol{\delta}_\mathrm{x}, \mathbf{l}) = (\mathbf{r}_0 - \mathbf{A}\boldsymbol{\delta}_\mathbf{x}, \mathbf{l}) = 0.$$

Введем в подпространствах K и L базисы $\{\boldsymbol{\nu}_j\}_{j=1}^{m}$ и $\{\boldsymbol{\omega}_j\}_{j=1}^{m}$, где $\dim K = \dim L = m$. Тогда последнее выражение будет справедливо при

$$\forall j\ (1 \le j \le m)\colon (\mathbf{r}_0 - \mathbf{A}\boldsymbol{\delta}_{\mathbf{x}}, \omega_j) = 0. \qquad (2.41)$$

Введя для базисов матричные обозначения $\mathbf{V} = [\boldsymbol{\nu}_1| \ldots |\boldsymbol{\nu}_m]$ и $\mathbf{W} = [\boldsymbol{\omega}_1| \ldots |\boldsymbol{\omega}_m]$, можно записать $\boldsymbol{\delta}_{\mathbf{x}} = \mathbf{V}\mathbf{y}$, где $\mathbf{y} \subset R^m$ – вектор коэффициентов. Тогда выражение (2.41) преобразуется к виду

$$\mathbf{W}^T(\mathbf{r}_0 - \mathbf{A}\mathbf{V}\mathbf{y}) = 0 \Rightarrow\ \mathbf{y} = (\mathbf{W}^T\mathbf{A}\mathbf{V})^{-1}\mathbf{W}^T\mathbf{r}_0. \qquad (2.42)$$

С учетом этого решение системы (2.1) должно уточняться в соответствии с формулой

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{V}\mathbf{y} = \mathbf{x}_0 + \mathbf{V}(\mathbf{W}^T\mathbf{A}\mathbf{V})^{-1}\mathbf{W}^T\mathbf{r}_0,$$

из которой вытекает важное требование для организации вычислений: произведение $\mathbf{W}^T\mathbf{A}\mathbf{V}$ должно быть или малой размерности, или легко обращаться. Из выражения (2.42) также вытекает соотношение

$$\mathbf{V}\mathbf{y} = \mathbf{A}^{-1}\mathbf{r}_0 = \mathbf{A}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_0) = \mathbf{x}_* - \mathbf{x}_0 = \boldsymbol{\delta}_{\mathbf{x}},$$

где $\mathbf{x}_*$ – точное решение СЛАУ (2.1). Таким образом, $\mathbf{V}\mathbf{y}$ – проекция разности между точным решением и начальным приближением на подпространство K.

В качестве подпространства K часто выбирают подпространства Крылова – линейные пространства размерностью m, порожденные вектором $\boldsymbol{\nu}$ и матрицей $\mathbf{A}$ (линейная оболочка):

$$K_m(\boldsymbol{\nu}, \mathbf{A}) = \operatorname{span}\{\boldsymbol{\nu}, \mathbf{A}\boldsymbol{\nu}, \mathbf{A}^2\boldsymbol{\nu}, \ldots, \mathbf{A}^{m-1}\boldsymbol{\nu}\}.$$

При этом в качестве вектора $\boldsymbol{\nu}$ выбирается невязка начального приближения $\mathbf{r}_0$. Тогда конкретный выбор подпространства L и способа построения базисов подпространств (биортогонализация Ланцоша, A-биортогонализация, ортогонализация Арнольди) полностью определяет вычислительную схему метода.

Пусть пространства K и L связаны соотношением $L = \mathbf{A}K$, причем в качестве K используем подпространство Крылова $K_m(\boldsymbol{\nu}_1, \mathbf{A})$, где $\boldsymbol{\nu}_1 = \mathbf{r}_0 / \beta$, $\beta = \|\mathbf{r}_0\|_2$, а для построения ортонормированного базиса – ортогонализацию Арнольди. Рассмотрим задачу

минимизации функционала $\mathbf{\Phi}(\mathbf{x}) = \|\mathbf{r}_x\|_2^2$ (эквивалентную задаче проектирования (2.42)), так как любой вектор **x** из пространства $(\mathbf{x}_0 + K_m)$ может быть записан в виде

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y},$$

где **y** – вектор размера m. Таким образом, задачу минимизации функционала можно переписать как

$$\mathbf{y}_m = \arg\min_{\mathrm{y}} \|\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{V}_m\mathbf{y})\|_2^2 .$$

Вычислительная схема, построенная с использованием таких предпосылок, называется методом обобщенных минимальных невязок (GMRES). На практике чаще используется версия этого метода с рестартами, сокращенно GMRES(m).

В отличие от ортогонализации Арнольди, биортогонализация Ланцоша использует для построения базиса экономичные трехчленные формулы. Так, выбрав в качестве пространства K – $K_m(\mathbf{r}_0, \mathbf{A})$, а в качестве L – $L_m(\underline{\mathbf{r}}_0, \mathbf{A}^T)$, где вектор $\underline{\mathbf{r}}_0$ выбирается из условия $(\mathbf{r}_0, \underline{\mathbf{r}}_0) \neq 0$, строят метод бисопряженных градиентов (BiCG). Достаточно часто для него характерна неустойчивость решения и осциллирующее поведение нормы невязки. Более того, итерационный процесс может полностью оборваться без возможности его дальнейшего продления. К тому же метод BiCG плохо поддается реализации на многопроцессорных вычислительных системах с распределенной памятью за счет использования операций с транспонированной матрицей. Эти проблемы привели к разработке целого класса методов, в которых операция с транспонированной матрицей не используется.

Алгебраически это достигается за счет изменения специальным образом полинома p_m, которому удовлетворяет последовательность невязок в методах, использующих подпространства Крылова. Из ряда методов, свободных от транспонирования, в настоящее время широко применяется стабилизированный метод бисопряженных градиентов (BiCGStab), использующий соотношение $\mathrm{r}_m = \mathrm{p}_m(\mathbf{A})\mathrm{q}_m(\mathbf{A})\mathbf{r}^{(0)}$, где q_m – специальным образом строящийся полином, такой, что произведение $\mathrm{p}_m\mathrm{q}_m$ не содержит нечетных степеней. Еще один метод этого семейства – квадратичный

метод сопряженных градиентов (CGS). Данный метод строится на основе соотношения $r_m = p_m(\mathbf{A}^T)\underline{\mathbf{r}}_0$.

Если матрица СЛАУ симметричная и положительно определенная, то метод BiCG имеет более простой вид. Метод, который получается за счет упрощений, вносимых симметричностью, называется методом сопряженных градиентов (CG). Следует заметить, что метод бисопряженных градиентов исторически появился как обобщение CG на несимметричный случай. Существуют и другие методы. Одни из них появились самостоятельно, а другие – как модификации уже известных.

В таблице 2.1 приведен перечень итерационных методов крыловского типа (1950–2014 гг.) для решения СЛАУ с одной правой частью [22]. При решении электромагнитных задач широкое распространение нашли методы BiCGStab и GMRES(m). Стоит отметить, что данный перечень неполный, но еще раз подтверждает тот факт, что развитие итерационных методов актуально.

Таблица 2.1 – Итерационные методы крыловского типа

Год	Разработчик(и)	Метод
1950	Lanczos	Lanczos
1951	Arnoldi	Arnoldi
1952	Hestenes, Stiefel	CG, CGNR
1952	Lanczos	Lanczos (CG)
1955	Craig	CGNE
1975	Paige, Saunders	MINRES
1975	Paige, Saunders	SYMMLQ
1975	Fletcher	BiCG
1976	Concus, Golub	CGW
1977	Vinsome	ORTHOMIN
1977	Meijerink, Van der Vorst	ICCG
1978	Widlund	CGW
1980	Jea, Young	ORTHODIR
1980	Wesseling, Sonneveld	IDR
1981	Saad	FOM
1982	Paige, Saunders	LSQR
1983	Eisenstat и др.	GCR
1986	Saad, Schultz	GMRES
1989	Sonneveld	CGS
1990	Van der Vorst, Melissen	COCG

Окончание таблицы 2.1

Год	Разработчик(и)	Метод
1991	Freund and Nachtigal	QMR
1992	Van der Vorst	BiCGStab
1993	Gutknecht	BiCGStab2
1993	Sleijpen, Fokkema	BiCGStab(*l*)
1994	Freund	TFQMR
1994	Weiss	GMERR
1994	Chan и др.	QMR-BiCGStab
1994	Freund, Nachtigal	SQMR
1995	Kasenally, Ebrahim	GMBACK
1996	Fokkema и др.	CGS2
1997	Pocock, Walker	CBICG
1997	Zhang	GPBi-CG
1999	de Sturler	GCROT
1999	Sadok	CMHR
2001	Szyld, Vogel	FQMR
2007	Sogabe, Zhang	COCR
2008	Sonneveld, van Gijzen	IDR(*s*)
2008	Ильин	BiCR (A-BiCG, A-CGS, A-BiCGStab), SCR
2009	Jing и др.	BiCOR, BiCORSTAB
2010	Abe, Sleijpen	BiCR
2010	Tanio, Sugihara	GBi-CGSTAB
2010	Hicken, Zingg	GCROT(*m*, *k*)
2011	Carpentieri и др.	BiCOR, CORS
2013	Zhao и др.	GPBiCOR
2013	Zhao, Huang	BiCORSTAB2
2013	Hajarian	QMRCGSTAB
2014	Sun и др.	QMRCORSTAB
2014	Zhang	GCORS

Для примера приведем алгоритмы методов BiCGStab и CGS с предобусловливанием.

Алгоритм метода BiCGStab

Вычислить предобусловливатель $\mathbf{M}$
Выбрать начальное приближение $\mathbf{x}_0$
$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
Выбрать вектор $\tilde{\mathbf{r}}$, удовлетворяющий условию $(\mathbf{r}_0, \tilde{\mathbf{r}}) \neq 0$ (например, $\tilde{\mathbf{r}} = \mathbf{r}_0$)

Для $i = 1, 2, \ldots$ до сходимости или до $N_{it}^{\max}$

$\rho_{i-1} = (\tilde{\mathbf{r}}, \mathbf{r}_{i-1})$

Если $\rho_{i-1} = 0$

то метод не может решить данную систему

Если $i = 1$

$\mathrm{p}_i = \mathrm{r}_{i-1}$

Иначе

$\beta_{i-1} = (\rho_{i-1} / \rho_{i-2})\ (\alpha_{i-1} / \omega_{i-1})$

$\mathbf{p}_i = \mathbf{r}_{i-1} + \beta_{i-1}(\mathbf{p}_{i-1} - \omega_{i-1}\, \mathbf{v}_{i-1})$

Решить СЛАУ $\mathbf{M}\tilde{\mathbf{p}} = \mathbf{p}_i$ относительно $\tilde{\mathbf{p}}$

$\mathbf{v}_i = \mathbf{A}\tilde{\mathbf{p}}$

$\alpha_i = \rho_{i-1} / (\tilde{\mathbf{r}}, \mathbf{v}_i)$

$\mathbf{s} = \mathbf{r}_{i-1} - \alpha_i\, \mathbf{v}_i$

Если $\|\mathbf{s}\|_2 / \|\mathbf{r}_0\|_2 \le Tol$

то КОНЕЦ ($\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \tilde{\mathbf{p}}$ – полученное решение)

Решить СЛАУ $\mathbf{M}\tilde{\mathbf{s}} = \mathbf{s}_i$ относительно $\tilde{\mathbf{s}}$

$\mathbf{t} = \mathbf{A}\,\tilde{\mathbf{s}}$

$\omega_i = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$

$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \tilde{\mathbf{p}} + \omega_i\, \tilde{\mathbf{s}}$

$\mathbf{r}_i = \mathbf{s} - \omega_i\, \mathbf{t}$

Если $\|\mathbf{r}\|_2 / \|\mathbf{r}_0\|_2 \le Tol$

то КОНЕЦ ($\mathbf{x}^{(i)}$ – полученное решение)

Увеличить i

Алгоритм метода CGS

Вычислить предобусловливатель $\mathbf{M}$

Выбрать начальное приближение $\mathbf{x}_0$

$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

Выбрать вектор $\tilde{\mathbf{r}}$, удовлетворяющий условию $(\mathbf{r}_0, \tilde{\mathbf{r}}) \neq 0$ (например, $\tilde{\mathbf{r}} = \mathbf{r}_0$)

Для $i = 1, 2, \ldots$ до сходимости или до $N_{it}^{\max}$

$\rho_{i-1} = (\tilde{\mathbf{r}}, \mathbf{r}_{i-1})$

Если $\rho_{i-1} = 0$

то метод не может решить данную систему

Если $i = 1$

$\mathbf{u}_1 = \mathbf{r}_0$

$\mathbf{p}_1 = \mathbf{u}_1$

Иначе

$\beta_{i-1} = (\rho_{i-1} / \rho_{i-2})$

$\mathbf{u}_i = \mathbf{r}_{i-1} + \beta_{i-1}\mathbf{q}_{i-1}$

$\mathbf{p}_i = \mathbf{u}_i + \beta_{i-1}(\mathbf{q}_{i-1} + \beta_{i-1}\mathbf{p}_{i-1})$

Решить СЛАУ $\mathbf{M}\tilde{\mathbf{p}} = \mathbf{p}_i$ относительно $\tilde{\mathbf{p}}$

$\tilde{\mathbf{v}} = \mathbf{A}\tilde{\mathbf{p}}$

$\alpha_i = \rho_{i-1} / (\tilde{\mathbf{r}}, \tilde{\mathbf{v}})$

$\mathbf{q}_i = \mathbf{u}_i - \alpha_i \tilde{\mathbf{v}}$

Решить СЛАУ $\mathbf{M}\tilde{\mathbf{u}} = \mathbf{u}_i + \mathbf{q}_i$ относительно $\tilde{\mathbf{u}}$

$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \tilde{\mathbf{u}}$

$\tilde{\mathbf{q}} = \mathbf{A}\tilde{\mathbf{u}}$

$\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i \tilde{\mathbf{q}}$

Если $\|\mathbf{r}\|_2 / \|\mathbf{r}_0\|_2 \le Tol$

то КОНЕЦ ($\mathbf{x}_i$ – полученное решение)

Увеличить i

2.3.5 Предобусловливание

Для ускорения процесса итерационного решения разреженных СЛАУ часто используется предобусловливание, которое позволяет улучшить обусловленность результирующей матрицы и тем самым уменьшить число итераций. Так, с ростом cond(**A**) обусловленность ухудшается и для ряда проблем сходимость может оказаться очень медленной, поэтому итерационный процесс может стагнировать или даже оборваться. Термин «предобусловливание» (preconditioning), по всей видимости, был впервые использован Форсайтом в 1955 г. при рецензировании работы Ланцоша. (В [18] данный подход назван «подготовка».) В современной трактовке этот термин впервые использован в работе [23]. Позже предобусловливание было применено и при решении СЛАУ с плотной матрицей. При этом для решения таких систем, как правило, используют метод исключения Гаусса и его модификации.

Поясним суть данного подхода. Пусть **M** – некоторая невырожденная квадратная матрица порядка *N*.

Тогда, домножив систему (2.1) на матрицу $\mathbf{M}^{-1}$, получим эквивалентную систему

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}, \qquad (2.43)$$

имеющую то же точное решение $\mathbf{x}_*$. Хотя выражения (2.1) и (2.43) алгебраически эквивалентны, спектральные характеристики матрицы $\mathbf{M}^{-1}\mathbf{A}$ отличаются от характеристик исходной матрицы **A**,

что ведет к изменению скорости сходимости методов для системы (2.1) в конечной арифметике. Процесс перехода от (2.1) к (2.43) и называется предобусловливанием, а матрица $\mathbf{M}^{-1}$ – предобусловливателем (предобусловливающая матрица). При этом матрица $\mathbf{M}$ должны быть близка к матрице $\mathbf{A}$ (сформирована из нее) и легко вычислима и обратима. Невязка ($\underline{\mathbf{r}}$) системы (2.43) связана с невязкой ($\mathbf{r}$) исходной системы (2.1) соотношением $\mathbf{M}\underline{\mathbf{r}} = \mathbf{r}$, которое справедливо и для других матрично-векторных произведений, что позволяет вместо явного перехода от (2.1) к (2.43) вводить в схемы методов корректирующие шаги (матрично-векторное умножение).

Описанное выше предобусловливание принято называть левым, так как домножение на предобусловливатель выполняется слева. Другой метод основан на переходе к системе

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{y} = \mathbf{b}, \tag{2.44}$$

решение которой $\mathbf{y}_*$ связано с точным решением $\mathbf{x}_*$ исходной СЛАУ (2.1) соотношением

$$\mathbf{x}_* = \mathbf{M}^{-1}\mathbf{y}_*. \tag{2.45}$$

Предобусловливание (2.44) реализуется путем двойных умножений вида $\mathbf{z} = \mathbf{A}\left(\mathbf{M}^{-1}\mathbf{q}\right)$. Кроме того, при достижении требуемой точности осуществляется пересчет решения в соответствии с выражением (2.45). Такая схема предобусловливания называется правой.

Последний возможный вариант – двухстороннее предобусловливание, являющееся компромиссным относительно левого и правого. В этом случае имеет место представление $\mathbf{M} = \mathbf{M}_1\mathbf{M}_2$, тогда решение вычисляется в виде $\mathbf{M}_1^{-1}\mathbf{A}\mathbf{M}_2^{-1}\mathbf{z} = \mathbf{M}_1^{-1}\mathbf{b}$ и $\mathbf{x} = \mathbf{M}_2^{-1}\mathbf{z}$.

Способы предобусловливания можно разбить на два вида: явные и неявные. Для представления системы (2.1) с явным предобусловливанием удобно воспользоваться записью (2.43). Как указано выше, предобусловливание может быть введено в схему метода без необходимости явного вычисления матричного произведения. Так, явное предобусловливание требует нахождения матрицы $\mathbf{M}^{-1}$ и умножения матрицы предобусловливания на вектор в

каждой итерации. Для неявного необходимо решать СЛАУ с матрицей **M** в каждой итерации. Наиболее простым является предобусловливание Якоби. Оно заключается в том, что диагональные элементы матриц **M** и **A** совпадают, а внедиагональные элементы матрицы **M** полагаются равными нулю. Рассмотрим эти два вида предобусловливания.

Неявно предобусловленную систему (2.1) удобно представить в виде уравнения

$$\mathbf{MAx} = \mathbf{Mb}. \qquad (2.46)$$

Наиболее работоспособные и часто используемые методы неявного предобусловливания основаны на LU-разложении. Если матрица, подвергающаяся разложению, является разреженной, то результирующие матрицы **L** и **U** являются более плотными. Поэтому применяют неполную факторизацию (неполное разложение). Идея создания таких методов принадлежит Булееву Н.И., который разрабатывал их с 1950-х гг. Впоследствии эти методы неоднократно переоткрывались зарубежными учеными. Неявное предобусловливание намного чаще используется при решении СЛАУ с плотной матрицей. Ограничимся кратким рассмотрением этих методов.

С учетом требования близости матриц положим, что **M** = **A**, и представим ее в виде

$$\mathbf{M} = \mathbf{LU} + \mathbf{R},$$

где **L** и **U** – нижне- и верхнетреугольные матрицы соответственно; **R** – матрица ошибки. Тогда приближенное представление $\mathbf{M} \approx \mathbf{LU}$ и есть неполное LU-разложение матрицы **A**, или коротко ILU. Самое простое ILU(0)-разложение заключается в применении LU-разложения к матрице **A**, но если $a_{ij} = 0$, то сразу полагается m_{ij} (l_{ij} или u_{ij}) равным нулю. Если обозначить $NZ(\mathbf{A})$ структуру (портрет) разреженности матрицы **A**, а $NZ(\mathbf{M})$ – матрицы **M**, то очевидно, что данный способ приведет к тому, что их структуры (портреты) совпадут. Чтобы подчеркнуть, что в данной постановке задачи в структуру не вводятся новые ненулевые элементы, такую факторизацию часто называют факторизацией с нулевым заполнением (zero fill-in) или просто ILU(0). Это позволяет

оптимально использовать оперативную память, когда матрица **A** хранится с помощью специальных форматов, учитывающих только ненулевые элементы. Тогда алгоритм (*ikj*-версия) этого разложения можно представить в следующем виде.

Алгоритм ILU(0)-разложения

Для $i = 2, 3, \ldots, N$
  Для $k = 1, 2, \ldots, i-1$
    Если $(i, k) \in NZ(\mathrm{A})$
      $m_{ik} = m_{ik} / m_{kk}$
    Для $j = k+1, 2, \ldots, N$ и
      Если $(i, j) \in NZ(\mathrm{A})$
        $m_{ij} = m_{ij} - m_{ik} \cdot m_{kj}$
    Увеличить j
  Увеличить k
Увеличить i

Поскольку полное разложение приводит к заполнению структуры матрицы без каких-либо ограничений (факторизация с бесконечным заполнением), а ILU(0)-разложение – к нулевому заполнению, очевидно, что еще одним вариантом может служить разложение с неким уровнем заполнения. Таким способом является ILU(p)-разложение. Здесь каждому ненулевому элементу матрицы первоначально присваивается нулевой уровень заполнения, в противном случае уровень принимается равным бесконечности. В ходе вычислений ($m_{ij} = m_{ij} - m_{ik}\, m_{kj}$) после нахождения элемента в позиции (i, j) необходимо провести корректировку его уровня заполнения согласно правилу

$$lev_{ij} = \min\left\{lev_{ij}, lev_{ik} + lev_{kj} + 1\right\}.$$

Отметим, что элементы, имеющие до начала вычислений нулевой уровень, никогда его не меняют. Эти элементы определяют структуру разреженности, совпадающую с исходной. Элементы, получившие во время вычислений уровень $\le p$ (задаваемого значения), расширяют эту структуру. Таким образом, при $p = \infty$ данное разложение вырождается в полное, а если $p = 0$ – в ILU(0).

Другой стратегией расширения структуры разреженности является использование постфильтрации, осуществляемой в два эта-

па и позволяющей контролировать уплотнение структуры (портрета) матрицы. Данное разложение называется неполным LU-разложением с упорождиванием (отбрасыванием), или ILUT.

Алгоритм ILUT-разложения

Для $i = 1, \ldots, N$
  $\mathbf{w} = \boldsymbol{a}_{i*}$
  Для $k = 1,\ldots, i-1$ и $w_k \neq 0$
    $w_k = w_k / u_{kk}$
    Применить правило обнуления к w_k
    Для $j = k+1,\ldots, N$
      $\mathbf{w} = \mathbf{w} - w_k \cdot \boldsymbol{u}_{k*}$
    Увеличить j
  Увеличить k
  Применить правило обнуления к строке **w**
  $l_{ij} = w_j$, для $j = 1, \ldots, i-1$
  $u_{ij} = w_j$, для $j = i, \ldots, N$
Увеличить i

Приведем правила обнуления, основанные на постфильтрации элементов.

1. В строке 5 алгоритма ILUT-разложения элемент w_k обнуляется, если его значение меньше, чем евклидова норма преобразуемой строки, умноженная на задаваемый допуск обнуления τ.

2. В строке 10 первоначально происходит обнуление элементов строки **w** аналогично правилу пункта 1. На втором этапе происходит сохранение по p элементов в **L** и **U** частях преобразуемой строки в сочетании с диагональными элементами, которые никогда не обнуляются.

Таким образом, пункт 2 позволяет контролировать количество элементов в i-й строке. По-существу, параметр p позволяет контролировать затраты машинной памяти, а параметр τ – вычислительные затраты. Созданы различные модификации пункта 2. Одной из них является сохранение $nu(i)+p$ элементов в **U** части преобразуемой строки и $nl(i)+p$ элементов в **L** части, где $nu(i)$ и $nl(i)$ – число ненулевых элементов в **U** и **L** частях i-й строки матрицы **A** соответственно.

Существуют и другие алгоритмы неполного LU-разложения, например разложение ILUC, основанное не на *ikj*-версии LU-разложения, а на так называемой краутовской версии и постфильтрации, описанной в ILUT.

Данный вид предобусловливания исходит из нахождения матрицы $\mathbf{M}^{-1}$. Из него по работоспособности выделяются способы, основанные на приближенном обращении (approximate inverse). Их удобно разделить на три группы.

Способы первой группы появились первыми и базируются на нахождении матрицы **M**, минимизирующей норму Фробениуса матрицы невязки $\mathbf{I} - \mathbf{AM}$. Существует две стратегии вычисления обратной матрицы: первая основана на вычислении обратной матрицы «глобально»; вторая основана на постолбцовом вычислении обратной матрицы. Среди способов этой группы выделяются: SPAI – разреженный приближенный обратный (sparse approximate inverse), использующий постолбцовую стратегию в сочетании с QR-разложением и итерационным уточнением; SPMR – метод минимальных невязок с самопредобусловливанием (Self-preconditioned Minimal Residual), использующий постолбцовую стратегию, реализуемую с помощью итерационного метода минимальных невязок с предобусловливанием.

Вторую группу образуют способы, формирующие матрицу предобусловливания с помощью факторизованных приближенных обратных матриц. Эти способы основаны на неполном обращении треугольных матриц **L** и **U**, т. е. неполном нахождении обратной матрицы. Так, если существует разложение $\mathbf{A} = \mathbf{LDU}$, где **L** и **U** – нижне- и верхнетреугольные матрицы с единичными диагоналями, а **D** – диагональная матрица, то матрица $\mathbf{A}^{-1}$ может быть найдена с помощью разложения $\mathbf{A}^{-1} = \mathbf{U}^{-1}\mathbf{D}^{-1}\mathbf{L}^{-1} = \mathbf{Z}\mathbf{D}^{-1}\mathbf{W}^{T}$, где $\mathbf{Z} = \mathbf{U}^{-1}$ и $\mathbf{W} = \mathbf{L}^{-T}$ – верхнетреугольные матрицы с единичной диагональю. Факторизованные приближенные обратные предобусловливатели могут быть получены вычислением с помощью разреживания матриц $\underline{\mathbf{Z}} \approx \mathbf{Z}$, $\underline{\mathbf{W}} \approx \mathbf{W}$ и $\underline{\mathbf{D}} \approx \mathbf{D}$, тогда $\mathbf{M} = \underline{\mathbf{Z}}\,\underline{\mathbf{D}}^{-1}\underline{\mathbf{W}}^{T} \approx \mathbf{A}^{-1}$. Имеется несколько подходов для вычисления приближенного обращения. Эти методы не требуют нахождения треугольных матриц **L** и **U**, т. е. предобусловливатель вычисляется непосредственно из мат-

рицы **A**. К данному классу относятся методы FSAI, AINV и SAINV.

В третью группу входят методы, вычисляющие обратную матрицу, основываясь на двухшаговом процессе. Первоначально производится неполное LU-разложение (используя стандартные способы, описанные выше), а затем полученные при неполном разложении матрицы приближенно обращаются, например путем решения $2N$ треугольных систем.

2.3.6 Предфильтрация

Как было показано выше, для улучшения сходимости итерационного процесса решения разреженных СЛАУ применяется предобусловливание, основанное на использовании структуры (портрета) разреженности матрицы **A**. Таким образом, для формирования предобусловливателя при решении СЛАУ с плотной матрицей сначала необходимо определить или задать некую структуру разреженности матрицы **A**. Процесс, позволяющий сформировать разреженную матрицу $\mathbf{A}_S$ из плотной матрицы **A**, называют предфильтрацией в качестве альтернативы постфильтрации как, например, в методе ILUT. Далее из матрицы $\mathbf{A}_S$ формируется предобусловливатель **M**.

Одним из простейших способов (правил) является выбор заранее известной структуры, например ленточной или блочно-диагональной, т. е. элементы матрицы $\mathbf{A}_S$ совпадают с ленточной (блочно-диагональной) частью матрицы **A** и равны нулю вне данной структуры. Такая предфильтрация характеризуется шириной ленты (размером блока) и, таким образом, основана на позиции элемента в матрице (структурная предфильтрация), а не на его значении. Результаты ее использования при решении плотных СЛАУ можно найти, например, в [24].

В работе [25] приведено несколько методов, специально приспособленных для построения матрицы $\mathbf{M}^{-1}$. Одним из них является выбор k максимальных элементов в каждом столбце, так что в результате в матрице будет kN элементов, не равных нулю. Данный способ связан с многократным поиском, поэтому он

характеризуется значительными временными затратами на предфильтрацию.

Чаще используется динамическое определение структуры разреженности, например с помощью некоторого порога (алгебраическая предфильтрация). Этот метод заключается в задании (нахождении) определенного порога, с помощью которого происходит обнуление малозначащих элементов путем сравнения модуля каждого элемента с данным (полученным) значением.

Порог обнуления в свою очередь может быть получен поразному. Первый способ основан на малости значений матрицы относительно заданного порога (ε):

$$a_{ij}^{s} = a_{ij}, \text{ если } \left|a_{ij}\right| > \varepsilon, \;\; i, j = 1, 2, \ldots, N \;\text{ или }\; i = j. \qquad (2.47)$$

Он является простейшим с точки зрения вычислений и требует наименьшего времени на формирование разреженной матрицы $\mathbf{A}^{s}$, поскольку порог обнуления задается непосредственно, без его вычисления.

Второй способ обнуления основан на нормировке всей матрицы с помощью максимального элемента:

$$a_{ij}^{s} = a_{ij}, \text{ если } \left|a_{ij} \Big/ a^{\max}\right| > \varepsilon, \;\; i, j = 1, 2, \ldots, N \text{ или } i = j, \quad (2.48)$$

где $a^{\max}$ – максимальный элемент матрицы. Понятно, что при использовании данного способа достаточно много времени требуется на поиск наибольшего элемента в матрице.

Способ был предложен для применения явного предобусловливания при решении СЛАУ с плотной матрицей. Первоначально происходило обнуление малозначащих элементов исходной матрицы, т. е. плотную матрицу приводили к разреженному виду. Далее на основе полученной структуры разреженности матрицы формировался предобусловливатель. Таким образом решение СЛАУ с плотной матрицей сводилось к решению разреженной системы итерационным методом с предобусловливанием. Впоследствии, уйдя от разреживания исходной матрицы, исследователи пришли к идее использовать данный подход для формирования предобусловливателя, т. е. при предфильтрации. Он является, пожалуй, самым распространенным.

Следующий способ обнуляет (игнорирует) элементы с помощью бесконечной нормы матрицы:

$$a_{ij}^{s} = a_{ij}, \text{ если } |a_{ij}| > \varepsilon = \|\mathbf{A}\|_{\infty}\tau/N,\ i, j = 1, 2, \ldots, N \text{ или } i = j, \quad (2.49)$$

где $\| \mathbf{A} \|_{\infty} = \max_i \sum_{j=1}^{N} | a_{ij} |$; τ – допуск обнуления. Очевидно, что значительные временные затраты при использовании данного способа связаны с поиском наибольшей суммы строки.

Способ использовался следующим образом. Первоначально система с плотной матрицей преобразовывалась с помощью вейвлетов, после чего она оставалась достаточно плотной, но основная часть ее элементов имела небольшие абсолютные значения. Обнуление элементов по описанному выше правилу приводило к получению разреженной матрицы СЛАУ. Таким образом, была продемонстрирована возможность приведения СЛАУ с плотной матрицей к разреженному виду и тем самым применения эффективных итерационных алгоритмов, ориентированных на разреженные системы.

В последующем этот подход использовали при решении задачи излучения и он оказался эффективным. Вычисления методом моментов производились на примере излучения цилиндра. Было установлено, что выбор $\tau = 0{,}1–0{,}2$ является оптимальным для минимизации времени решения СЛАУ. Поскольку такое упрощение, дающее ускорение, не всегда обосновано и приемлемо для некоторых классов задач, то исследователи стали использовать это правило обнуления как способ определения структуры разреженности матрицы предобусловливания, т. е. как предфильтрацию.

Еще один способ выбора структуры разреженности основан на нахождении наибольшего элемента в строке

$$a_{ij}^{s} = a_{ij}, \text{ если } | a_{ij} / a_i^{\max} | > \varepsilon,\ i, j = 1, 2, \ldots, N \text{ или } i = j, \quad (2.50)$$

где $a_i^{\max}$ – максимальный элемент в i-й строке. Поскольку способ основан на нахождении максимального элемента в каждой строке, порог обнуления является разным для каждой из строк. После того как максимальный элемент найден, происходит процесс, подобный описанному при использовании подхода (2.48), но вместо

всей матрицы сканируется ее строка. Очевидно, что при использовании данного метода значительные затраты времени связаны с поиском максимальных элементов строк.

Последний способ был предложен для решения задач магнитостатики [26]:

$$a_{ij}^{S} = a_{ij}, \text{ если } a_{ij} > \varepsilon = \tau \min\left(|\, a_{ij}\,|, |\, a_{jj}\,|\right),\ i, j = 1, 2, \ldots, N \text{ или } i = j. \tag{2.51}$$

К сожалению, рекомендаций по выбору допуска обнуления в [26] не приведено. Результаты вычислений представлены только при фиксированном значении τ, при котором плотность матрицы $\mathbf{A}_S$ составляет 5 %.

Как видно из рассмотренных способов, с течением времени исследователи пытались найти предфильтрацию, адаптивную к различным задачам. Существуют и другие способы предфильтрации. Одни позволяют представить исходную матрицу суммой нескольких разреженных матриц посредством специальных подходов и в дальнейшем использовать структуру одной из них в качестве структуры матрицы предобусловливания. Другие строятся на основе геометрических или топологических особенностей исследуемой задачи.

Отметим, что алгебраическая предфильтрация может сочетаться как со структурной, так и с динамической. Приведенные способы алгебраической предфильтрации могут быть использованы и при постфильтрации. Таким образом, данный вид предфильтрации является, пожалуй, самым универсальным. Однако трудно сказать, какой способ предпочтительнее с точки зрения как минимизации времени решения СЛАУ, так и стабильности допуска/порога обнуления.

2.3.7 Многократное решение СЛАУ

Помимо упомянутых ранее достоинств подпространств Крылова, они позволяют строить эффективные итерационные методы для решения СЛАУ с несколькими правыми частями и неизменной матрицей $\mathbf{AX} = \mathbf{B}$, где $\mathbf{X}$ и $\mathbf{B}$ – матрицы размером $N \times m$, $m << N$. Это уравнение является частным случаем СЛАУ (2.27).

Очевидно, что для решения таких СЛАУ самым простым способом (с точки зрения реализации) будет последовательное решение СЛАУ с каждой правой частью по отдельности. Однако данный подход характеризуется наибольшими вычислительными затратами. Поэтому для решения таких систем разработаны разные блочные версии итерационных методов. Так, известны методы, использующие подпространства вида $Km(\mathbf{R}_0, \mathbf{A})$, где $\mathbf{R}_0$ – обобщенная невязка начального приближения. Назовем только некоторые из них [22]: Bl-CG и Bl-BiCG, Bl-GMRES, Bl-QMR, Bl-BiCGStab, Bl-LSQR, Bl-IDR(s), Bl-GCROT(m, k), Bl-CMHR. Именно эти методы считаются наиболее подходящими для решения СЛАУ с плотной матрицей. Их особенностью является то, что все правые части должны быть одновременно доступны до начала вычислений. Другой подход основан на выборе опорной (seed) СЛАУ, построении подпространств Крылова для нее и последующем решении остальных систем путем проектирования их невязок на подпространство Крылова. На его базе разработаны методы Seed-GMRES, Seed-EGCR, Seed-QMR, Seed-MINRES, Seed-MEGCR, Seed-BiCGStab и др. (здесь для единообразного представления использованы отличающиеся от оригинальных аббревиатуры названий методов). Этот подход считается эффективным, если правые части близки между собой. Еще один подход (глобальный) основан на формировании и решении тензоризованных систем, например Gl-GMRES и Gl-FOM, Gl-BiCG и Gl-BiCGStab, Gl-CMRH, Gl-SCD, Gl-CGS, Gl-BiCR. Последний способ, разработанный недавно, использует перестановку внутренних циклов (loop-interchanging) [27], что, по сути, является альтернативой последовательному раздельному вычислению с каждой правой частью и позволяет несколько снизить вычислительную сложность одной итерации. Представителем этих методов является Li-BiCG.

Особым случаем является решение последовательности (2.27), когда изменяется матрица, а правая часть или постоянна, или изменяется несущественно. Подобное многократное решение СЛАУ возникает во многих научных и инженерных приложениях: при восстановлении изображений, рекурсивном вычислении

наименьших квадратов, оптимизации, решении нелинейных уравнений, в прикладной статистике и т. д.

Еще раз отметим, что блочные методы применимы, только если все правые части доступны до вычислений. Однако на практике это не всегда выполнимо. Так, часто «новые» матрица и правая часть формируются с использованием предыдущего решения СЛАУ. Для решения последовательности (2.27) с разреженными матрицами (для не электромагнитных задач) разработано несколько подходов, в основном связанных с построением эффективного предобусловливателя. При этом подавляющее число исследователей рассматривает задачу решения последовательности «сдвинутых» (shifted) СЛАУ, т. е. когда от матрицы к матрице изменяются только диагональные элементы:

$$\mathbf{A}_k = \mathbf{A} + \mathrm{diag}\left(\delta_1^k, ..., \delta_N^k\right),\ \delta_i^k \geq 0,\ \ i = 1, \ldots, N,\ k = 1, 2, \ldots, m.$$

Один из подходов основан на перевычислении предобусловливателя для каждой отдельно взятой СЛАУ. Другой подход использует «замороженный» (frozen) предобусловливатель, вычисленный из матрицы первой СЛАУ последовательности с одной правой частью, для решения остальных систем. В [28] рассмотрено использование такого предобусловливателя в сочетании с фиксированным числом (шагов по времени p) СЛАУ, после решения которых происходит переформирование предобусловливателя. Определение оптимального значения p с точки зрения минимизации временных затрат на моделирование выполнено эмпирически и находится в диапазоне от 5 до 10. При этом число шагов по времени не выбиралось априори, а определялось динамически во время моделирования до достижения стационарности решения. Очевидно, что при $p = 1$ данный подход эквивалентен первому подходу.

Основная мысль следующего подхода достаточно проста. Повторное использование одного и того же предобусловливателя часто приводит к медленной сходимости итерационного процесса, а повторное вычисление предобусловливателя для каждой новой системы является вычислительно затратным. Поэтому очевидно, что для этих крайних случаев существуют промежуточные аль-

тернативы. Так, необходимо найти возможность обновления предобусловливателя с меньшими затратами, чем на его повторное вычисление. Тогда можно ожидать, что полученный предобусловливатель, хоть и будет менее эффективным, чем заново вычисленный, с точки зрения числа итераций, но общая сложность окажется значительно сниженной. В [29] сформулирован общий подход к вычислению «идеального» предобусловливателя для решения последовательности СЛАУ, не ограничивающийся «сдвинутыми» системами. Для ясности изложения рассмотрим последовательность СЛАУ, где $\mathbf{Ax} = \mathbf{b}$ обозначим первую из них, а $\mathbf{A}^{+}\mathbf{x}^{+} = \mathbf{b}^{+}$ – одну из последующих. В качестве предобусловливателя используем $\mathbf{M} = \mathbf{LDU}$. Очевидно, что матрица разницы (вариации) будет $\underline{\mathbf{A}} = \mathbf{A} - \mathbf{A}^{+}$. В [29] предложено вычисление приближения для «идеального» предобусловливателя вида $\mathbf{M}^{+} = \mathbf{M} - \underline{\mathbf{A}}$, которое является очень затратным. Поэтому для практических вычислений вместо $\mathbf{M} - \underline{\mathbf{A}}$ рассмотрены приближения, основанные на малости $\|\mathbf{L} - \mathbf{I}\|$ и $\|\mathbf{U} - \mathbf{I}\|$, что справедливо, если матрица $\mathbf{A}$ строго диагональная. Такие последовательности СЛАУ часто встречаются при решении нелинейных уравнений. Тогда если матрица $\mathbf{M} - \underline{\mathbf{A}}$ неособенная, то она может быть представлена в виде

$$\mathbf{M} - \underline{\mathbf{A}} = \mathbf{L}(\mathbf{DU} - \mathbf{L}^{-1}\underline{\mathbf{A}}) \approx \mathbf{L}(\mathbf{DU} - \underline{\mathbf{A}}) \approx \mathbf{L}(\mathbf{DU} - \mathrm{triu}(\underline{\mathbf{A}}))$$

или

$$\mathbf{M} - \underline{\mathbf{A}} = (\mathbf{LD} - \underline{\mathbf{A}}\mathbf{U}^{-1})\mathbf{U} \approx (\mathbf{LD} - \underline{\mathbf{A}})\mathbf{U} \approx (\mathbf{LD} - \mathrm{tril}(\underline{\mathbf{A}}))\mathbf{U}.$$

Выбор первого или второго варианта обновления осуществляется с помощью простой оценки малости $\|\mathbf{L} - \mathbf{I}\|$ и $\|\mathbf{U} - \mathbf{I}\|$. Так, если значение $\|\mathbf{L} - \mathbf{I}\|$ меньше, чем $\|\mathbf{U} - \mathbf{I}\|$, то более точным является первый вариант, а в противном случае – второй. Такие обновления называются структурированными (structured update). Указанные требования справедливы при решении нелинейных уравнений методами ньютоновского и бройденовского типа. Упрощенным вариантом обновлений является

$$\mathbf{M} - \underline{\mathbf{A}} \approx \mathrm{diag}(\mathbf{DU} - \underline{\mathbf{A}}).$$

При этом подход в целом является обобщением для обновления симметричных матриц. Известен более общий случай за счет использования трансформаций Гаусса – Жордана. Он разработан

специально для задач, когда вариации существенны по всей матрице **A**, а не только по ее треугольным частям. При этом обновления происходят или периодически, или перед началом решения текущей системы.

Еще один подход сочетает в себе идеи предыдущих. Он состоит в том, что задается диапазон СЛАУ, на котором используется вычисленный для первой СЛАУ из диапазона «замороженный» предобусловливатель. Количество требуемых для ее решения итераций фиксируется ($iter_0$). Если для решения последующей СЛАУ требуемое количество итераций превышает значение $iter_0 + j$, где j – заданный порог, то происходит обновление предобусловливателя.

Последний подход основан на адаптивном использовании информации об уже построенных подпространствах Крылова (recycling) при решении предыдущих систем для обновления предобусловливателя [30]. Отметим, что аналогичная идея использована при разработке итерационных методов, ориентированных на решение «сдвинутых» систем [31].

Контрольные вопросы и задания

1. Вычислить $(\mathbf{x}, \mathbf{x})$ и $\|\mathbf{x}\|_2$ для вектора $\mathbf{x} = (3, 4, -5)^T$.

2. Вычислить скалярное произведение векторов $\mathbf{x} = (3, 4, -5)^T$ и $\mathbf{y} = (1, 2, i)^T$.

3. Как оценивается обусловленность задачи решения СЛАУ?

4. Найти $\mathrm{cond}_2(\mathbf{A})$ для матрицы $\mathbf{A} = \begin{pmatrix} 5.001 & 5.1 \\ 7.1 & 7.001 \end{pmatrix}$.

5. Найти LU-разложение матрицы $\mathbf{A} = \begin{pmatrix} 4 & 3 & 4 & 6 \\ 2 & 1 & 2 & 4 \\ 3 & 2 & 1 & -1 \\ 5 & 3 & -2 & 2 \end{pmatrix}$.

6. Чему пропорциональны вычислительные затраты прямых и итерационных методов решения СЛАУ?

7. В чем отличие методов Якоби и Гаусса – Зейделя?

8. Разработать программу на языке Octave для обращения матриц с помощью LU-разложения.

9. Разработать программу на языке Octave для решения СЛАУ методом Гаусса – Зейделя.

10. Разработать программу на языке Octave для решения СЛАУ методом BiCGStab.

11. Разработать программу на языке Octave для решения двух СЛАУ, матрицы которых отличаются только элементами одной строки.

3 МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ

3.1 Конечно-разностная аппроксимация

Метод конечных разностей (МКР, FDM) разработан А. Томом в 1920 г. под названием «Метод квадратов» для решения нелинейных уравнений гидродинамики. С тех пор метод нашел применение для решения задач из различных областей. Методы конечных разностей основаны на аппроксимациях, которые позволяют заменить дифференциальные уравнения уравнениями конечных разностей. Разностные уравнения имеют алгебраический вид. Они ставят в соответствие значению зависимой переменной в точке расчетной области значение в некоторой соседней точке. Решение методом конечных разностей в основном состоит из трех этапов: разделения области решения на сетку узлов; аппроксимации данного дифференциального уравнения разностным эквивалентом; решения разностных уравнений с учетом заданных граничных и начальных условий. Таким образом, при вычислениях МКР осуществляется аппроксимация производных функций одной или нескольких переменных значениями этой функции в дискретном множестве значений аргументов (в узлах). Совокупность узлов образует сетку, покрывающую расчетную область [32].

Пусть функция $f(x)$ дифференцируема на интервале $(x_0 - h_1, x_0 + h_2)$ и имеет производную в точке B $(x = x_0)$, равную тангенсу угла наклона касательной DE в этой точке (рисунок 3.1). Таким образом, существует предел отношения приращения функции к приращению аргумента:

$$f'_{FD}(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} =$$

$$= \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}.$$

Ее можно вычислить в близко расположенной точке, например в точке C. Тогда $\Delta x = h_2$ и

$$f'_{FD}(x_0) \approx \frac{f(x_0 + h_2) - f(x_0)}{h_2}.$$

Это так называемая правая или правосторонняя конечно-разностная формула (в англоязычной литературе ее часто называют forward-difference). Вычислим вторую производную:

$$f''_{FD}(x_0) \approx \frac{f'\left(x_0 + \frac{3}{2}h_2\right) - f'\left(x_0 + \frac{1}{2}h_2\right)}{h_2} =$$

$$= \frac{1}{h_2}\left[\frac{f(x_0 + 2h_2) - f(x_0 + h_2)}{h_2} - \frac{f(x_0 + h_2) - f(x_0)}{h_2}\right] =$$

$$= \frac{f(x_0 + 2h_2) - 2f(x_0 + h_2) + f(x_0)}{{h_2}^2}.$$

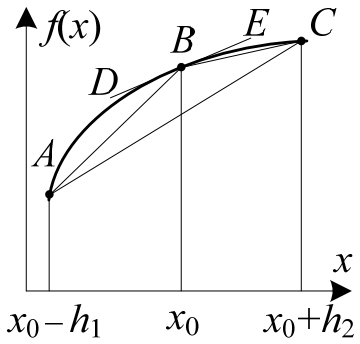


Рисунок 3.1 – Аппроксимация производной конечными разностями

С тем же успехом можно использовать левую или левостороннюю формулу (backward-difference):

$$f'_{BD}(x_0) = \lim_{x \to x_0} \frac{f(x_0) - f(x)}{x_0 - x} =$$

$$= \lim_{\Delta x \to 0} \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} \approx \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x}.$$

Тогда в точке A ($x = x_0 - h_1$) получим

$$f'_{BD}(x_0) \approx \frac{f(x_0) - f(x_0 - h_1)}{h_1}. \qquad (3.1)$$

Расстояние между точками, в которых вычисляются значения функции, называется шагом сетки (поэтому метод конечных разностей также называют методом сеток). В данном случае шаги

отличаются по значению, т. е. используется неравномерная сетка. Вычислим вторую производную:

$$f''_{BD}(x_0) \approx \frac{f'\left(x_0 - \frac{1}{2}h_1\right) - f'\left(x_0 - \frac{3}{2}h_1\right)}{h_1} =$$

$$= \frac{1}{h_1}\left[\frac{f(x_0) - f(x_0 - h_1)}{h_1} - \frac{f(x_0 - h_1) - f(x_0 - 2h_1)}{h_1}\right] =$$

$$= \frac{f(x_0) - 2f(x_0 - h_1) + f(x_0 - 2h_1)}{h_1{}^2}.$$

Еще одним вариантом является использование центральной или двусторонней разностной формулы (central-difference)

$$f'_{CD}(x_0) \approx \frac{f(x_0 + h_2) - f(x_0 - h_1)}{h_1 + h_2}.$$

Последняя формула дает более точное, чем предыдущие, решение, поскольку значение производной равно тангенсу угла наклона хорды AC, что ближе к точному решению.

При $h_2 = h_1 = h$ получим

$$f'_{CD}(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}. \qquad (3.2)$$

Далее, используя центральные разности и полагая, что $h_2 = h_1 = h$, вычислим вторую производную:

$$f''_{CD}(x_0) \approx \frac{f'(x_0 + h/2) - f'(x_0 - h/2)}{h} =$$

$$= \frac{1}{h}\left[\frac{f(x_0 + h) - f(x_0)}{h} - \frac{f(x_0) - f(x_0 - h)}{h}\right],$$

$$f''_{CD}(x_0) \approx \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2}. \qquad (3.3)$$

Эта формула называется второй разностной производной. В ней точка, в которой аппроксимируется производная, центральная среди точек, вовлеченных в аппроксимацию. Зависимость производной (3.3) от значений функции $f(x)$ в точках, используемых для аппроксимации, часто иллюстрируется «шаблоном» или «молекулой», как показано на рисунке 3.2.

Необходимо отметить важность правосторонних и левосторонних разностей.

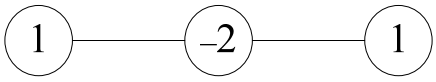


Рисунок 3.2 – Трехточечный шаблон для второй разностной производной (одномерный случай)

Так, на рисунке 3.3 схематично показана ситуация, когда неприменимы ни левосторонние, ни центральные разности, поскольку для их использования необходимо знать значение функции в точке x_{-1} ($x_0 - h$).

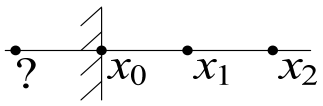


Рисунок 3.3 – Случай применимости только правосторонних разностей

Оценим количественно ошибку конечно-разностных аппроксимаций. Это можно сделать, используя разложение функции в ряд Тейлора:

$$FD: f(x_0 + h_2) =$$

$$= f(x_0) + h_2 f'(x_0) + \frac{1}{2!} h_2^2 f''(x_0) + \frac{1}{3!} h_2^3 f'''(x_0) + \ldots, \qquad (3.4)$$

$$BD: f(x_0 - h_1) =$$

$$= f(x_0) - h_1 f'(x_0) + \frac{1}{2!} h_1^2 f''(x_0) - \frac{1}{3!} h_1^3 f'''(x_0) + \ldots \qquad (3.5)$$

Видно, что оценки (3.4) и (3.5) имеют погрешность порядка $h_{1,2}$ (записывается как $O(h_{1,2})$), возникающую за счет отбрасывания высших членов ряда Тейлора. Вычтем (3.5) из (3.4):

$$f(x_0 + h_2) - f(x_0 - h_1) =$$

$$= (h_2 + h_1) f'(x_0) + \frac{1}{2!}\left(h_2^2 - h_1^2\right) f''(x_0) + O(h^3). \qquad (3.6)$$

Поделив левую и правую части выражения (3.6) на $h_2 + h_1$, получим

$$f'(x_0) = \frac{f(x_0 + h_2) - f(x_0 - h_1)}{h_2 + h_1} + O(h_2 - h_1), \qquad (3.7)$$

т.е. формулу, которая также имеет первый порядок погрешности. Однако при $h_1 = h_2 = h$ в формуле (3.6) исчезает член, содержащий вторую производную, и тогда получим формулу

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} + O\left(h^2\right), \qquad (3.8)$$

имеющую погрешность $O\left(h^2\right)$. Сложим (3.4) и (3.5):

$$f(x_0 + h) + f(x_0 - h) = 2f(x_0) + h^2 f''(x_0) + O\left(h^4\right). \qquad (3.9)$$

Поделив это выражение на h^2, получим производную (3.3), погрешность которой имеет порядок $O\left(h^2\right)$.

Таким образом, погрешность конечно-разностных формул определяется значением шага h. Чем меньше шаг, тем формула точнее. Однако при неограниченном уменьшении шага погрешность вычисления производных начинает увеличиваться, так как при этом разность между значениями функции в соседних узлах сетки уменьшается, что приводит к возрастанию влияния ошибок округления. Типовая зависимость погрешности конечно-разностной формулы от шага сетки показана на рисунке 3.4.

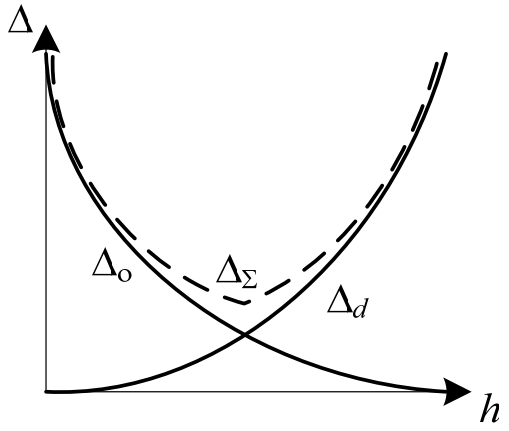


Рисунок 3.4 – Зависимость погрешности конечно-разностной аппроксимации от шага сетки: Δ_d – погрешность, вызванная отбрасыванием высших членов ряда Тейлора; Δ_o – погрешность, вызванная конечной точностью представления чисел; Δ_Σ – суммарная погрешность

3.2 Способы повышения точности вычислений

3.2.1 Разложение в ряд Тейлора

Рассмотрим способы повышения точности. Сначала используем трехточечный шаблон и правосторонние разности. Для этого потребуются три точки: x_0, x_0+h, $x+2h$. Далее воспользуемся разложением в ряд Тейлора функции в этих точках:

$$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{1}{2}h^2f''(x_0)+O\left(h^3\right);$$

$$f(x_0+2h)=f(x_0)+2hf'(x_0)+2h^2f''(x_0)+O\left(h^3\right).$$

Вычитая второе уравнение из первого, умноженного на 4, получим выражение для производной *FD*:

$$f'_{FD}(x_0)=\frac{4f(x_0+h)-f(x_0+2h)-3f(x_0)}{2h}. \qquad (3.10)$$

При использовании трехточечного шаблона и левосторонних разностей запишем

$$f(x_0-h)=f(x_0)-hf'(x_0)+\frac{1}{2}h^2f''(x_0)+O\left(h^3\right),$$

$$f(x_0-2h)=f(x_0)-2hf'(x_0)+2h^2f''(x_0)+O\left(h^3\right).$$

Также вычтем второе уравнение из первого, умноженного на 4:

$$4f(x_0-h)-f(x_0-2h)=3f(x_0)-2hf'(x_0)+O\left(h^3\right).$$

В результате получим выражение для производной *BD*:

$$f'_{BD}(x_0)=\frac{3f(x_0)-4f(x_0-h)+f(x_0-2h)}{2h}.$$

Ее погрешность имеет порядок $O\left(h^2\right)$.

Применим пятиточечный шаблон (центральные разности). Для этого потребуется пять точек: x_0-2h, x_0-h, x, x_0+h, $x+2h$. Далее воспользуемся разложением в ряд Тейлора функции в этих точках:

$$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{1}{2!}h^2f''(x_0)+\frac{1}{3!}h^3f'''(x_0)+O\left(h^4\right);$$

$$f(x_0-h)=f(x_0)-hf'(x_0)+\frac{1}{2!}h^2f''(x_0)-\frac{1}{3!}h^3f'''(x_0)+O\left(h^4\right);$$

$$f(x_0+2h)=f(x_0)+2hf'(x_0)+\frac{4}{2!}h^2f''(x_0)+\frac{8}{3!}h^3f'''(x_0)+O\left(h^4\right);$$

$$f(x_0-2h)=f(x_0)-2hf'(x_0)+\frac{4}{2!}h^2f''(x_0)-\frac{8}{3!}h^3f'''(x_0)+O\left(h^4\right).$$

Вычитая второе уравнение из первого и четвертое из третьего, получим

$$S_h:\ f(x_0+h)-f(x_0-h)=2hf'(x_0)+\frac{1}{3}h^3f'''(x_0)+O\left(h^4\right),$$

$$S_{2h}:\ f(x_0+2h)-f(x_0-2h)=4hf'(x_0)+\frac{8}{3}h^3f'''(x_0)+O\left(h^4\right).$$

Для исключения $f'''(x_0)$ вычислим $8S_h–S_{2h}$:

$$8f(x_0+h)-8f(x_0-h)-f(x_0+2h)+$$
$$+\ f(x_0-2h)=12hf'(x_0)+O\left(h^4\right).$$

В результате получим первую производную

$$f'_{CD}(x_0)=\frac{8f(x_0+h)-8f(x_0-h)-f(x_0+2h)+f(x_0-2h)}{12h},\quad (3.11)$$

погрешность которой имеет порядок $O\left(h^4\right)$.

Для нахождения второй производной запишем:

$$f(x_0+h)=f(x_0)+hf'(x_0)+\frac{h^2}{2}f''(x_0)+\frac{h^3}{8}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)+$$
$$+\frac{h^5}{120}f^{(5)}(x_0)+O\left(h^6\right);$$

$$f(x_0-h)=f(x_0)-hf'(x_0)+\frac{h^2}{2}f''(x_0)-\frac{h^3}{8}f'''(x_0)+\frac{h^4}{24}f^{(4)}(x_0)-$$
$$-\frac{h^5}{120}f^{(5)}(x_0)+O\left(h^6\right);$$

$$f(x_0+2h)=f(x_0)+2hf'(x_0)+2h^2f''(x_0)+\frac{4h^3}{3}f'''(x_0)+$$

$$+\frac{2h^4}{3}f^{(4)}(x_0)+\frac{32h^5}{120}f^{(5)}(x_0)+O\left(h^6\right);$$

$$f(x_0-2h)=f(x_0)-2hf'(x_0)+2h^2f''(x_0)-\frac{4h^3}{3}f'''(x_0)+$$

$$+\frac{2h^4}{3}f^{(4)}(x_0)-\frac{32h^5}{120}f^{(5)}(x_0)+O\left(h^6\right).$$

Сложив первое уравнение со вторым и третье с четвертым, получим

$$S_h:\ f(x_0+h)-f(x_0-h)=2f(x_0)+h^2f''(x_0)+\frac{h^2}{12}f^{(4)}(x_0)+O\left(h^6\right),$$

$$S_{2h}:\ f(x_0+2h)-f(x_0-2h)=f(x_0)+4h^2f''(x_0)+\frac{4}{3}h^2f^{(4)}(x_0)+O\left(h^6\right).$$

Для исключения $f^{(4)}(x_0)$ вычислим S_h–$S_{2h}/16$:

$$f(x_0+h)-f(x_0-h)-f(x_0+2h)+f(x_0-2h)=$$

$$=\frac{15}{8}f(x_0)+\frac{3}{4}h^2f''(x_0)+O\left(h^6\right).$$

В результате получим производную

$$f''_{CD}(x_0)=$$

$$=\frac{-f(x_0-2h)+16f(x_0-h)-30f(x_0)+16f(x_0+h)-f(x_0+2h)}{12h^2},$$

(3.12)

погрешность которой имеет порядок $O\left(h^4\right)$.

3.2.2 Интерполяционные полиномы

Рассмотрим другой способ получения конечно-разностных аппроксимаций, заключающийся в использовании интерполяционных полиномов вида

$$P_{N-1}(x)=\sum_{j=0}^{N-1}a_jx^j.$$

Тогда в окрестности x_0 можно предположить, что $f'(x_0) \approx P'(x_0)$ и $f''(x_0) \approx P''(x_0)$. Здесь N – нечетное количество эквидистантных точек в окрестности x_0, таких, что $f_k = f(x_k)$, $x_k = x_0 + kh$, $k = -(N-1)/2, \ldots, (N+1)/2$. Используя введенные обозначения, получим систему уравнений

$$\{P_{N-1}(x_k) = f_k\}, \; k = -(N-1)/2, \ldots, (N+1)/2.$$

Рассмотрим процесс нахождения $f'(x_0)$ и $f''(x_0)$ при $N = 3$, для простоты полагая $x_0 = 0$. Запишем интерполяционный полином

$$P_{N-1}(x) = a_0 + a_1 x + a_2 x^2.$$

Тогда $f'(0) \approx P'(0) = a_1$, $f''(0) \approx P''(0) = 2a_2$. Таким образом, для нахождения значений производных необходимо найти коэффициенты a_1 и a_2. Для этого запишем систему уравнений

$$\begin{cases} P_2(-h) = f_{-1} \\ P_2(0) = f_0 \\ P_2(h) = f_1 \end{cases} \Rightarrow \begin{cases} a_0 - a_1 h + a_2 h^2 = f_{-1} \\ a_0 = f_0 \\ a_0 + a_1 h + a_2 h^2 = f_1 \end{cases}.$$

Вычитая первое уравнение из третьего, получим

$$f'(0) \approx P'(0) = a_1 = \frac{-f_{-1} + f_1}{2h},$$

а сложив их и используя второе уравнение, получим $2f_0 + 2a_2 h^2 = f_{-1} + f_1$. Тогда

$$f''(0) \approx P''(0) = 2a_2 = \frac{f_{-1} - 2f_0 + f_1}{h^2}.$$

Видно, что данные выражения аналогичны формулам (3.2) и (3.3).

Повышение точности аппроксимации производных возможно за счет увеличения степени интерполяционного полинома и использования большего числа точек, чем это необходимо для вычисления производной данного порядка. Рассмотрим процесс нахождения $f'(x_0) \approx P'(x_0)$ и $f''(x_0) \approx P''(x_0)$ при $N = 5$ (пятиточечный шаблон). В данном случае получим следующую систему уравнений:

$$\begin{cases} P_4(-2h) = f_{-2} \\ P_4(-h) = f_{-1} \\ P_4(0) = f_0 \\ P_4(h) = f_1 \\ P_4(2h) = f_2 \end{cases} \Rightarrow \begin{cases} a_0 - 2a_1h + 4a_2h^2 - a_3h^3 + 16a_4h^4 = f_{-2} \\ a_0 - a_1h + a_2h^2 - a_3h^3 + a_4h^4 = f_{-1} \\ a_0 = f_0 \\ a_0 + a_1h + a_2h^2 + a_3h^3 + a_4h^4 = f_1 \\ a_0 + 2a_1h + 4a_2h^2 + a_3h^3 + 16a_4h^4 = f_2 \end{cases}.$$

Вычитая первое уравнение из пятого и второе из четвертого, получим систему

$$\begin{cases} 4a_1h + 16a_3h^3 = f_{-2} - f_2 \\ 2a_1h + 2a_3h^3 = f_1 - f_{-1} \end{cases},$$

а сложив их, –

$$\begin{cases} 2a_0 + 8a_2h^2 + 32a_4h^4 = f_{-2} + f_2 \\ 2a_0 + 2a_2h^2 + 2a_4h^4 = f_{-1} + f_1 \end{cases}.$$

Решая первую из этих систем относительно a_1, найдем

$$f'(0) \approx P'(0) = a_1 = \frac{f_{-2} + 8f_1 - 8f_{-1} - f_2}{12h}.$$

Данное выражение эквивалентно (3.11), полученному с помощью разложения в ряд Тейлора. Решая вторую систему относительно a_2 (при $a_0 = f_0$), найдем

$$f''(0) \approx P''(0) = 2a_2 = \frac{-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2}{12h^2}.$$

Погрешность имеет порядок $O\left(h^4\right)$. Данное выражение эквивалентно (3.12), полученному с помощью разложения в ряд Тейлора.

Запишем итоговые формулы для производных, полученные с помощью семиточечного шаблона ($N = 7$):

$$f'(0) \approx \frac{-f_{-3} + 9f_{-2} - 45f_{-1} + 45f_1 - 9f_2 + f_3}{60h},$$

$$f''(0) \approx \frac{2f_{-3} - 27f_{-2} + 270f_{-1} - 490f_0 + 270f_1 - 27f_2 + 2f_3}{12h^2}.$$

3.2.3 Многочлены Лагранжа

Рассмотрим способ получения конечно-разностных аппроксимаций за счет использования многочленов Лагранжа. Напомним, что интерполяционный многочлен Лагранжа – это многочлен минимальной степени, принимающий данные значения в данном наборе точек. Для $n + 1$ пар чисел $(x_0, f(x_0))$, $(x_1, f(x_1)),\ldots,$ $(x_n, f(x_n))$, где все x_j различны, существует единственный многочлен $L(x)$ степени не более n, для которого $P(x_j) = f(x_j)$.

В общем виде базисные полиномы $P(x)$ определяются по формуле

$$P_n(x) = f(x_0)\frac{(x-x_1)(x-x_2)...(x-x_k)}{(x_0-x_1)(x_0-x_2)...(x_0-x_k)} +$$
$$+ f(x_1)\frac{(x-x_0)(x-x_2)...(x-x_k)}{(x_1-x_0)(x_1-x_2)...(x_1-x_k)} +$$
$$+ ... + f(x_k)\frac{(x-x_0)(x-x_1)...(x-x_j)}{(x_k-x_0)(x_k-x_2)...(x_k-x_j)}.$$

Для нахождения $f'(x_0)$ достаточно трех точек: x_0, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$. Тогда итоговый полином имеет вид

$$P_3(x) = f(x_0)\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} +$$
$$+ f(x_2)\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.$$

Продифференцировав его и сделав замену x на x_0, x_1 на x_0+h и x_2 на $x_0 + 2h$, получим

$$f'_{FD}(x_0) = P'_3(x_0) = \frac{-3f(x_0) + 4f(x_0+h) - f(x_0+2h)}{2h}.$$

Данное выражение эквивалентно (3.10), а его погрешность будет $O\left(h^2\right)$.

Далее найдем $f''(x_0)$ с погрешностью $O\left(h^2\right)$, используя правосторонние разности. Для этого потребуются четыре точки: x_0, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$ и $x_3 = x_0 + 3h$. Тогда

$$P_4(x) = f(x_0)\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} +$$
$$+f(x_1)\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} +$$
$$+f(x_2)\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} +$$
$$+f(x_3)\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}.$$

Продифференцировав дважды многочлен $P_4(x)$ и заменив x на x_0, x_1 на $x_0 + h$, x_2 на $x_0 + 2h$ и x_3 на $x_0 + 3h$, получим производную

$$f''_{FD}(x_0) = P''_4(x_0) = \frac{2f(x_0) - 5f(x_0+h) + 4f(x_0+2h) - f(x_0+3h)}{h^2},$$

погрешность которой имеет порядок $O\left(h^2\right)$.

В таблице 3.1 приведены часто используемые конечно-разностные аппроксимации с указанием их погрешности.

Таблица 3.1 – Конечно-разностные аппроксимации и их погрешность

Конечно-разностная аппроксимация	Погрешность
$f'_{FD}(x_0) = \frac{f(x_0+h) - f(x_0)}{h}$	$O(h)$
$f'_{BD}(x_0) = \frac{f(x_0) - f(x_0-h)}{h}$	$O(h)$
$f'_{CD}(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h}$	$O(h^2)$
$f'_{FD}(x_0) = \frac{4f(x_0+h) - f(x_0+2h) - 3f(x_0)}{2h}$	$O(h^2)$
$f'_{BD}(x_0) = \frac{3f(x_0) - 4f(x_0-h) + f(x_0-2h)}{2h}$	$O(h^2)$
$f'_{CD}(x_0) = \frac{8f(x_0+h) - 8f(x_0-h) - f(x_0+2h) + f(x_0-2h)}{12h}$	$O(h^4)$
$f''_{FD}(x_0) = \frac{f(x_0+2h) - 2f(x_0+h) + f(x_0)}{h^2}$	$O(h^2)$

Окончание таблицы 3.1

Конечно-разностная аппроксимация	Погрешность
$f''_{BD}(x_0) = \frac{f(x_0) - 2f(x_0 - h) + f(x_0 - 2h)}{h^2}$	$O(h^2)$
$f''_{CD}(x_0) = \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2}$	$O(h^2)$
$f''_{CD}(x_0) = \\ = \frac{-f(x_0-2h)+16f(x_0-h)-30f(x_0)+16f(x_0+h)-f(x_0+2h)}{12h^2}$	$O(h^4)$

3.3 Решение эллиптических уравнений

3.3.1 Двухмерное уравнение Лапласа: однородный диэлектрик

Первая проблема, с которой приходится сталкиваться при реализации конечно-разностного метода, – это вывод конечно-разностных уравнений в исследуемой пространственной области из соответствующего дифференциального уравнения в частных производных. Когда для анализа линии передачи используется уравнение Лапласа, распределение потенциала ищется в ограниченной области, которая разбивается координатными линиями на некоторое число элементарных ячеек. Каждая точка пересечения двух линий, являющихся сторонами ячейки, образует узел. Значения потенциала в узловых точках и являются искомыми величинами.

В двухмерном случае простейшая равномерная сетка соответствует декартовой системе координат и состоит из прямоугольных ячеек. Из каждого узла сетки, двигаясь вдоль сторон ячеек, можно попасть в четыре соседних узла (рисунок 3.5, *а*). Такая сетка называется пятиточечной. Если рассматривается трехмерная задача, то при движении по сетке из каждого узла можно попасть в шесть соседних. В этом случае сетка семиточечная.

Применим конечно-разностный подход, чтобы найти решение функции Ф, зависящей от двух пространственных переменных *x* и *y* (уравнение Лапласа):

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0. \qquad (3.13)$$

Разделим область решения в плоскостях x и y на равные прямоугольники, как показано на рисунке 3.5, *а*. Обозначим координаты узлов

$$x = i\Delta x,\ i = 0, 1, 2, \ldots;\ y = j\Delta y,\ j = 0, 1, 2, \ldots,$$

и значение функции в некоторой точке P

$$\Phi_P = \Phi(i, j) = \Phi(i\Delta x,\ j\Delta y)\ .$$

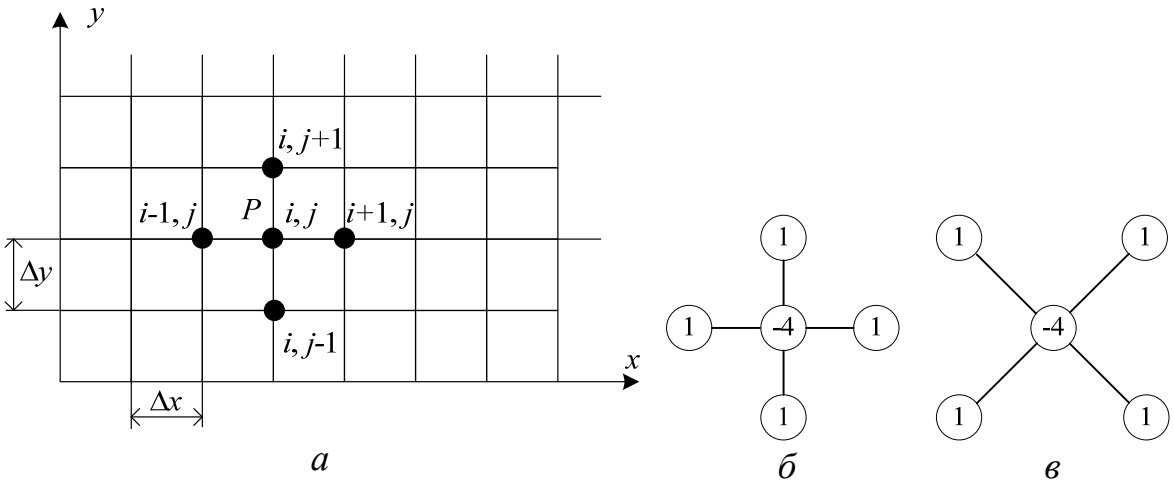
Тогда, используя выражения (3.2) и (3.3), получим производные функции Ф в узле (i, j):

$$\frac{\partial \Phi}{\partial x} = \Phi_x\,|_{i,j} \approx \frac{\Phi(i+1, j) - \Phi(i-1, j)}{2\Delta x};$$

$$\frac{\partial \Phi}{\partial y} = \Phi_y\,|_{i,j} \approx \frac{\Phi(i, j+1) - \Phi(i, j-1)}{2\Delta y};$$

$$\frac{\partial^2 \Phi}{\partial x^2} = \Phi_{xx}\,|_{i,j} \approx \frac{\Phi(i+1, j) - 2\Phi(i, j) + \Phi(i-1, j)}{(\Delta x)^2};$$

$$\frac{\partial^2 \Phi}{\partial y^2} = \Phi_{yy}\,|_{i,j} \approx \frac{\Phi(i, j+1) - 2\Phi(i, j) + \Phi(i, j-1)}{(\Delta y)^2}.$$



а *б* *в*

Рисунок 3.5 – Двухмерная конечно-разностная сетка (*а*), стандартный (*б*) и «косой» (диагональный) (*в*) двухмерные шаблоны

Подставляя эти выражения в уравнение (3.13), запишем его конечно-разностное представление:

$$\frac{\Phi(i+1,j)-2\Phi(i,j)+\Phi(i-1,j)}{(\Delta x)^2}+\frac{\Phi(i,j+1)-2\Phi(i,j)+\Phi(i,j-1)}{(\Delta y)^2}=0.$$

Полагая $\Delta x = \Delta y = h$, получим

$$\Phi(i,j)=\frac{1}{4}\left[\Phi(i+1,j)+\Phi(i-1,j)+\Phi(i,j+1)+\Phi(i,j-1)\right]. \quad (3.14)$$

Данное приближение называется пятиточечным центральным приближением лапласиана. Его шаблон (вычислительная молекула) приведен на рисунке 3.5, *б* (в отечественной литературе – пятиточечный шаблон «крест»).

Другое приближение можно получить с помощью четырех точек ($x \pm \Delta$, $t \pm \Delta$), расположенных на двух диагоналях. Эти точки используются таким же образом, как и для первого приближения, с тем отличием, что изменится шаг сетки (расстояние между узловыми точками равно $\sqrt{2}h$). Соответствующий шаблон приведен на рисунке 3.5, *в* (в отечественной литературе – пятиточечный диагональный шаблон). Конечный вид для решаемого уравнения Лапласа

$$\Phi(i+1,j)+\Phi(i-1,j)+\Phi(i,j+1)+\Phi(i,j-1)=\frac{1}{4}\Phi(i,j).$$

Если комбинировать эти два пятиточечных шаблона, то можно получить девятиточечные шаблоны. Для этого удобно воспользоваться выражением

$$\alpha S_{+}+(1-\alpha S_{\times}), \quad (3.15)$$

где S_{+} – стандартный и $S_{\times}$ – косой шаблоны.

Записав α в виде рационального числа a/b, при использовании пятиточечных шаблонов лапласиан ($\Delta\Phi$) можно представить в виде

$$\Delta\Phi=\frac{1}{2bh^2}\left[2a\Phi(i+1,j)+2a\Phi(i-1,j)+2a\Phi(i,j+1)+2a\Phi(i,j-1)+\right.$$
$$+(b-a)\Phi(i+1,j+1)+(b-a)\Phi(i+1,j-1)+(b-a)\Phi(i-1,j+1)+$$
$$\left.+(b-a)\Phi(i-1,j-1)-4(a+b)\Phi(i,j)\right].$$

Легко заметить, что при $b = a = 1$ будем иметь стандартный пятиточечный шаблон. При $a = 1/2$ и $b = 3/2$, $a = 1$ и $b = 2$, $a = 2$ и

$b = 3$ получим ряд девятиточечных шаблонов с коэффициентами 8, 12 и 20 соответственно (в отечественной литературе их называют шаблонами типа «ящик»):

$$\frac{1}{3h^2}\begin{bmatrix}1 & 1 & 1\\ 1 & -8 & 1\\ 1 & 1 & 1\end{bmatrix};\quad \frac{1}{4h^2}\begin{bmatrix}1 & 2 & 1\\ 2 & -12 & 2\\ 1 & 2 & 1\end{bmatrix};\quad \frac{1}{6h^2}\begin{bmatrix}1 & 4 & 1\\ 4 & -20 & 4\\ 1 & 4 & 1\end{bmatrix}.$$

Аналогично с использованием выражения (3.12) получим стандартный (в отечественной литературе иногда называется «большим крестом») и диагональный девятиточечные шаблоны:

$$\frac{1}{12h^2}\begin{bmatrix} & & -1 & & \\ & & 16 & & \\ -1 & 16 & -60 & 16 & -1\\ & & 16 & & \\ & & -1 & & \end{bmatrix};\quad \frac{1}{24h^2}\begin{bmatrix}-1 & & & & -1\\ & 16 & & 16 & \\ & & -60 & & \\ & 16 & & 16 & \\ -1 & & & & -1\end{bmatrix}.$$

Комбинируя эти два шаблона с помощью выражения (3.15), получим общее выражение для семнадцатиточечных шаблонов:

$$\Delta\Phi = \frac{1}{24bh^2}\big[-2a\Phi(i-2,j)+32a\Phi(i-1,j)+32a\Phi(i+1,j)-2a\Phi(i+2,j)-$$
$$-2a\Phi(i,j-2)+32a\Phi(i,j-1)+32a\Phi(i,j+1)-2a\Phi(i,j+2)-$$
$$-(b-a)\Phi(i-2,j-2)+16(b-a)\Phi(i-1,j-1)+16(b-a)\Phi(i+1,j+1)-$$
$$-(b-a)\Phi(i+2,j+2)-(b-a)\Phi(i-2,j+2)+16(b-a)\Phi(i-1,j+1)+$$
$$+16(b-a)\Phi(i+1,j-1)-(b-a)\Phi(i+2,j-2)-60(a+b)\Phi(i,j)\big].$$

Так, при $a = b = 1$ будем иметь стандартный девятиточечный шаблон, а при $a = 1$ и $b = 2$ получим

$$\frac{1}{48h^2}\begin{bmatrix}-1 & & -2 & & -1\\ & 16 & 32 & 16 & \\ -2 & 32 & -180 & 32 & -2\\ & 16 & 32 & 16 & \\ -1 & & -2 & & -1\end{bmatrix}.$$

Пример 3.1

Решить уравнение Лапласа (3.13) при $0 \le x \le 1,\ 0 \le y \le 1$ и найти распределение потенциала в прямоугольной области, ограниченной идеально проводящими электродами (рисунок 3.6, *а*). Вычислительная модель с указанием используемых значений потенциалов и узлов сетки приведена на рисунке 3.6, *б*.

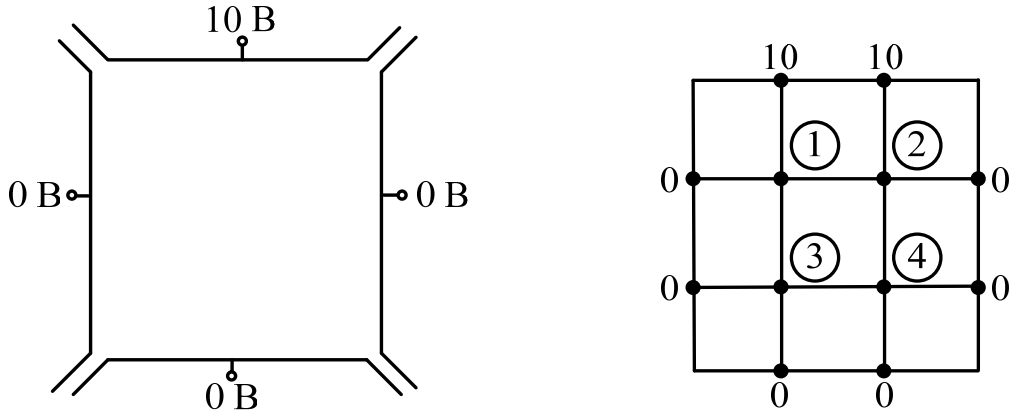


Рисунок 3.6 – К примеру решения конечно-разностных уравнений:
а – физическая модель; *б* – вычислительная модель

Решение

Используем пятиточечный шаблон, описываемый уравнением (3.14), и $h = 1/3$, а также итерационный метод для нахождения значений потенциала в узлах 1–4. При использовании метода Якоби уравнение (3.14) примет вид

$$\Phi(i,j)^{(n+1)} =$$

$$= \frac{1}{4}\left[\Phi(i+1,j)^{(n)} + \Phi(i-1,j)^{(n)} + \Phi(i,j+1)^{(n)} + \Phi(i,j-1)^{(n)}\right],$$

где верхний индекс обозначает номер итерации. При использовании метода Гаусса – Зейделя (при организации вычислений по строкам) уравнению (3.14) соответствует

$$\Phi(i,j)^{(n+1)} =$$

$$= \frac{1}{4}\left[\Phi(i+1,j)^{(n)} + \Phi(i-1,j)^{(n+1)} + \Phi(i,j+1)^{(n+1)} + \Phi(i,j-1)^{(n)}\right],$$

а метода релаксации –

$$\Phi(i,j)^{(n+1)} = \Phi(i,j)^{(n)} + \frac{\omega}{4} \times$$

$$\times \left[\Phi(i+1,j)^{(n)} + \Phi(i-1,j)^{(n+1)} + \Phi(i,j+1)^{(n+1)} + \Phi(i,j-1)^{(n)} - \Phi(i,j)^{(n)} \right],$$

где ω – параметр релаксации. При ω = 1 получим метод Гаусса – Зейделя, при ω > 1 – метод последовательной верхней релаксации, а при ω < 1 – метод последовательной нижней релаксации. Далее используем метод Гаусса – Зейделя.

Положив потенциал во всех внутренних узлах (1–4) равным нулю ($\Phi_{ij} = 0$), с помощью уравнения (3.14) найдем «новые» значения в них за 10 итераций.

Первая итерация	(1/4)(0+0+10+0)=2.5 (1/4)(0+0+2.5+0)=0.625	(1/4)(0+2.5+10+0)=3.125 (1/4)(0+0.625+3.125+0)=0.9375
Вторая итерация	3.4375 1.09375	3.59375 1.171875
Третья итерация	3.671875 1.2109375	3.7109375 1.23046875
Четвертая итерация	3.73046875 1.240234375	3.740234375 1.2451171875
Пятая итерация	3.7451171875 1.24755859375	3.74755859375 1.248779296875
Шестая итерация	3.748779296875 1.2493896484375	3.7493896484375 1.24969482421875
Седьмая итерация	3.74969482421875 1.249847412109375	3.749847412109375 1.2499237060546875
Восьмая итерация	3.7499237060546875 1.24996185302734375	3.74996185302734375 1.249980926513671875
Девятая итерация	3.749980926513671875 1.2499904632568359375	3.7499904632568359375 1.24999523162841796875
Десятая итерация	3.74999523162841796875 1.249997615814208984375	3.749997615814208984375 1.2499988079071044921875

Видно, что с каждой итерацией значения потенциала в каждом узле постепенно сближаются, так как разность между ними уменьшается. Отметим (см. рисунок 3.4), что все расчеты

проводятся с числами, имеющими ограниченное число десятичных разрядов (на рассмотренном небольшом примере это не так критично). В результате этого появляется дополнительная погрешность округления, которая добавляется к погрешности, возникающей при конечно-разностной аппроксимации.

В данном случае значение потенциала в одном из внутренних узлов сразу же используется для отыскания потенциала в соседнем узле. Указанная процедура повторяется для каждого из узлов до тех пор, пока два следующих друг за другом приближения не совпадут с требуемой точностью (TOL). Для этого в конце каждой итерации необходимо выполнять проверку вида

$$\text{error} \le \text{TOL},$$

где $\text{error} = \max\left(\left|\Phi_i^{(\text{it})} - \Phi_i^{(\text{it}-1)}\right|\right) \Big/ \Phi_i^{(\text{it})}$, $\text{TOL} = 10^{-m}$. Если условие выполняется, следует прекратить итерации, в противном случае необходимо продолжить вычисления. Параметр m позволяет контролировать точность вычислений (количество верных знаков после запятой). Поскольку итерационный процесс может расходиться или стагнировать, для предотвращения образования бесконечного цикла при программной реализации необходимо задавать максимальное количество итераций, при достижении которого итерационный процесс должен прерваться.

Рассмотрим другой подход к решению этой же задачи. Воспользовавшись уравнением (3.14), запишем систему уравнений для нахождения потенциалов во внутренних узлах:

$$\begin{gathered}
4\Phi_1 - \Phi_2 - \Phi_3 - 0\Phi_4 = 10; \\
-\Phi_1 + 4\Phi_2 - 0 - \Phi_4 = 10; \\
-\Phi_1 - 0\Phi_2 + 4\Phi_3 - \Phi_4 = 0; \\
-0\Phi_1 - \Phi_2 - \Phi_3 + 4\Phi_4 = 0.
\end{gathered}$$

Тогда в матричном виде получим

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 0 \\ 0 \end{pmatrix} \text{ или } \mathbf{Av} = \mathbf{b}.$$

Очевидно, что данная матрица является ленточной. Решив систему, например, с помощью метода исключения Гаусса, найдем

$$\mathbf{v} = \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{pmatrix} = \begin{pmatrix} 3{,}75 \\ 3{,}75 \\ 1{,}25 \\ 1{,}25 \end{pmatrix}.$$

Видно, что полученные решения близки. Первый вариант решения соответствует явной схеме, а второй – неявной.

3.3.2 Двухмерное уравнение Пуассона

Для общности изложения продемонстрируем на примере последовательность решения с использованием МКР двухмерного уравнения Пуассона.

Пример 3.2

Решить уравнение $\nabla^2\Phi = -\frac{\rho_s}{\varepsilon}$, $0 \le x, y \le 1$, и найти распределение потенциала в узлах сетки, показанной на рисунке 3.7.

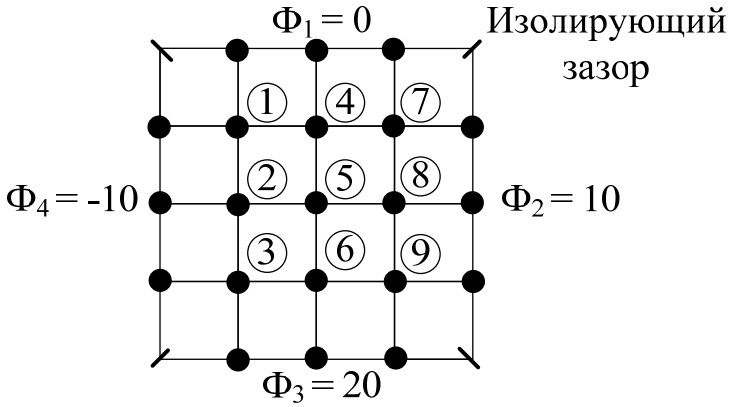


Рисунок 3.7 – Область решения для уравнения Пуассона

Решение

Положим $\rho_s = x(y-1)$ и $\varepsilon_r = 1$, тогда

$$g(x,y) = -\frac{\rho_s}{\varepsilon} = -\frac{x(y-1)10^{-9}}{10^{-9}/36\pi} = -36\pi x(y-1).$$

Используем метод последовательной релаксации.

Получим конечно-разностную аппроксимацию уравнения Пуассона (при $\Delta x = \Delta y = h$) для каждого узла сетки:

$$\Phi(i,j) =$$

$$= \frac{1}{4}\Big[\Phi(i+1,j)+\Phi(i-1,j)+\Phi(i,j+1)+\Phi(i,j-1)-h^2 g(i,j)\Big]. \quad (3.16)$$

Тогда невязка для узла (i,j) по выражению (3.16) вычисляется как

$$R(i,j) = \Phi(i+1,j) + \Phi(i-1,j) +$$

$$+ \Phi(i,j+1) + \Phi(i,j-1) - h^2 g(i,j) - 4\Phi(i,j). \quad (3.17)$$

Обозначим значение невязки на k-й итерации для узла (i,j) через $R^k(i,j)$. Это значение можно рассматривать как поправку к решению $\Phi(i,j)$ для приближения его к точному решению. Тогда по мере приближения $\Phi(i,j)$ к точному решению невязка $R(i,j)$ будет стремиться к нулю. Умножив невязку для узла (i,j) на ω и прибавив результат к $\Phi(i,j)$ на k-й итерации, получим $\Phi(i,j)$ на $k+1$-й итерации:

$$\Phi^{k+1}(i,j) = \Phi^k(i,j) + \frac{\omega}{4} R^k(i,j)$$

или

$$\Phi^{k+1}(i,j) = \Phi^k(i,j) + \frac{\omega}{4}\Big[\Phi^k(i+1,j) + \Phi^k(i-1,j) +$$

$$+ \Phi^k(i,j+1) + \Phi^k(i,j-1) - h^2 g(i,j) - 4\Phi^k(i,j)\Big].$$

Таким образом, для использования уравнения (3.17) необходимо задание начального (в общем случае произвольного) приближения для каждого внутреннего узла. Оно может быть, например, нулевым или среднеарифметическим значением, полученным на основании граничных условий, т. е. $(\Phi_1 + \Phi_2 + \Phi_3 + \Phi_4) / 4$. Оптимальное значение ω для прямоугольной области соответствует наименьшему корню квадратного уравнения

$$t^2\omega^2 - 16\omega + 16 = 0,$$

где $t = \cos(\pi/nx) + \cos(\pi/ny)$; nx и ny – число интервалов, на которые разбита область решения по осям x и y соответственно.

Решение уравнения дает (наименьший корень)

$$\omega = \frac{8 - \sqrt{64 - 16t^2}}{t^2}.$$

Для останова итераций можно использовать среднюю суммарную невязку по всем внутренним узлам (1–9). Итерации продолжаются до достижения требуемой точности, задаваемой параметром TOL, или максимального числа итераций (MaxIter).

В таблице 3.2 приведены результаты вычислений (листинг 3.1) при $nx = ny =$ 4; 12; 20, а также точное решение, полученное по аналитическим формулам из [32]. Из таблицы видно, что при уменьшении шага сетки h конечно-разностная аппроксимация дает более близкое к точному решение. При этом необходимо большее число итераций N_{it} для достижения требуемой точности.

Таблица 3.2 – Решение уравнения Пуассона для примера 3.2

Номер узла	$h = 1/4$ $\omega = 1.171$ $N_{\text{it}} = 10$	$h = 1/12$ $\omega = 1.589$ $N_{\text{it}} = 29$	$h = 1/12$ $\omega = 1.729$ $N_{\text{it}} = 46$	Точное решение [32]
1	–3.247	–3.409	–3.424	–3.429
2	–1.703	–1.982	–2.012	–2.029
3	4.306	4.279	4.280	4.277
4	0.039	–0.096	–0.109	–0.118
5	3.012	2.928	2.921	2.913
6	9.368	9.556	9.578	9.593
7	3.044	2.921	2.909	2.902
8	6.111	6.072	6.069	6.065
9	11.038	11.118	11.126	11.130

```
clear; clc;
weight=1; MaxIter=100; TOL=0.0001;
V1=0;V2=10;V3=20;V4=-10;
nx= 4; ny= nx; h = weight/nx;
V(1,2:nx)=V1;
V(ny+1,2:nx)=V3;
V(2:ny,1)=V4;
V(2:ny,nx+1)=V2;
V(2:nx,2:ny)=(V1 + V2 + V3 + V4)/4.0;
```

```
t = cos(pi/nx) + cos(pi/ny);
omega = ( 8 - sqrt(64 - 16*t^2))/(t^2);
disp(['SOR Factor Omega = ',num2str(omega)])
for Iter=1:MaxIter
  Rsum = 0;
 for num_row =1:ny-1
   y = h*num_row;
   for num_col = 1:nx-1
     x = h*num_col;
     G = -36.0*pi*x*(y - 1.0);
R = omega/4*( V(num_row+2,num_col+1) + V(num_row,num_col+1) +
V(num_row+1,num_col+2) + V(num_row+1,num_col)-
4.0*V(num_row+1,num_col+1) - G*h*h );
     V(num_row+1,num_col+1) = V(num_row+1,num_col+1) + R;
     Rsum = Rsum + abs(R);
   end
  end
  if((Rsum/((nx-1)*(ny-1))) >= TOL)
   if(Iter == MaxIter)
disp(['Solution does not converge in ' , num2str(Iter),' iterations'])
break;
   end
else
disp(['Solution Converges in ',num2str(Iter),' iterations'])
disp(['h = ', num2str(h)])
break;
  end
end
k=nx/4;
V(k+1:k:(3*k+1),k+1:k:(3*k+1))
```

Листинг 3.1 – Программный код для решения примера 3.2

3.4 Математическая модель вычисления емкостной матрицы многопроводной линии передачи

Рассмотренный в п. 3.3.1 пример решения двухмерного уравнения Лапласа характеризовался однородным диэлектрическим заполнением расчетной области. Далее рассмотрим случай слоистого диэлектрика. Подобная ситуация возникает при моделировании линий передачи, содержащих более одной диэлектрической среды. Поэтому необходимо соответствующим образом преобра-

зовать уравнение, описывающее распределение потенциала в пространстве, например уравнение (3.14), при использовании пятиточечного шаблона аппроксимации.

Рассмотрим микрополосковую линию (МПЛ) передачи. Слоистая среда состоит из двух диэлектриков с плоской границей раздела между ними. Простейший вариант такого заполнения, допускающий введение равномерной квадратной сетки, изображен на рисунке 3.8. Сетку построим так, чтобы границы части ячеек сетки совпали с границей раздела между диэлектриками. Шаг сетки равен h и отсчет номеров узлов ведется от узла в центре, которому присвоим индексы i, j.

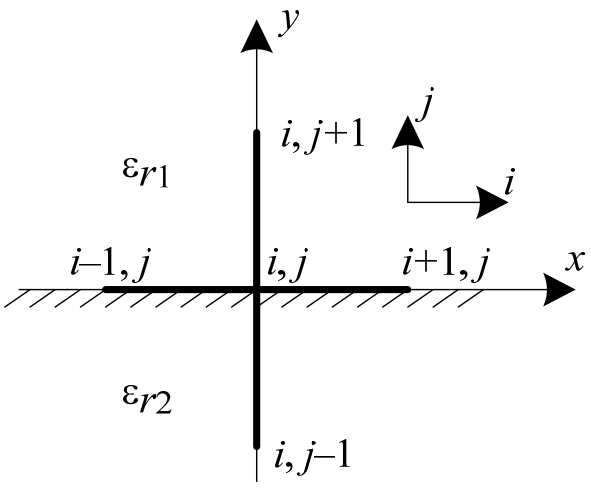


Рисунок 3.8 – Двухслойный диэлектрик

Запишем уравнение Лапласа при однородном заполнении: $\nabla^2\Phi = 0$. Чтобы ввести более общее уравнение, справедливое при неоднородном заполнении, обратимся к закону Гаусса в дифференциальной форме (электрические заряды могут концентрироваться только на поверхности проводника)

$$\nabla \cdot \mathbf{D} = \rho, \quad \mathbf{D} = \varepsilon \mathbf{E}.$$

Так как в пространстве вне проводников нет свободных зарядов, то $\rho = 0$ и $\nabla \cdot \mathbf{D} = 0$, т. е. в двухмерном случае для прямоугольных координат

$$\partial Dx/\partial x + \partial Dy/\partial y = 0.$$

Из этого уравнения вытекают конечно-разностные уравнения, связывающие поля по обе стороны от границ.

Поскольку $\mathbf{D} = -\varepsilon\nabla\Phi$, то в полупространстве над границей раздела, где $\varepsilon = \varepsilon_0\varepsilon_{r1}$, составляющая вектора индукции, параллельная координате *y*, имеет вид

$$\frac{\partial Dy^{\text{над}}}{\partial y} = -\frac{\varepsilon_0\varepsilon_{r1}\left(\Phi(i,j+1) - \Phi(i,j)\right)}{h},$$

а под границей, где $\varepsilon = \varepsilon_0\varepsilon_{r2}$, –

$$\frac{\partial Dy^{\text{под}}}{\partial y} = -\frac{\varepsilon_0\varepsilon_{r2}\left(\Phi(i,j) - \Phi(i,j-1)\right)}{h}.$$

При записи этих выражений использовались правосторонняя и левосторонняя разности. Тогда скорость изменения составляющей электрической индукции, нормальной к границе, при переходе через границу раздела двух сред определяется как

$$\frac{\partial Dy}{\partial y} = \frac{\partial Dy^{\text{под}}}{\partial y} - \frac{\partial Dy^{\text{над}}}{\partial y} =$$

$$= -\frac{\varepsilon_0}{h}\left[\varepsilon_{r1}\left(\Phi(i,j+1) - \Phi(i,j)\right) - \varepsilon_{r2}\left(\Phi(i,j) - \Phi(i,j-1)\right)\right]. \quad (3.18)$$

Аналогичное выражение можно записать и для производной от составляющей *Dx* в направлении координаты *x*, однако пока не ясно, какое значение ε_r в данном случае использовать в узлах $i-1, j$ и $i-1, j$. Сначала положим $\varepsilon_r = \varepsilon_{r3}$, тогда

$$\frac{\partial Dx^{\text{пр}}}{\partial x} = -\frac{\varepsilon_0\varepsilon_{r3}(\Phi(i+1,j) - \Phi(i,j))}{h},$$

$$\frac{\partial Dx^{\text{лев}}}{\partial x} = -\frac{\varepsilon_0\varepsilon_{r3}(\Phi(i,j) - \Phi(i-1,j))}{h}.$$

В результате получим

$$\frac{\partial Dx}{\partial x} = \frac{\partial Dx^{\text{лев}}}{\partial x} - \frac{\partial Dx^{\text{пр}}}{\partial x} =$$

$$= \frac{\varepsilon_0\varepsilon_{r3}}{h}\left[\Phi(i+1,j) + \Phi(i-1,j) - 2\Phi(i,j)\right]. \quad (3.19)$$

Складывая выражения (3.18) и (3.19) и приравнивая результат к нулю, найдем значение потенциала в точках, расположенных на границах раздела:

$$\Phi(i,j)=\frac{\varepsilon_{r1}\Phi(i,j+1)+\varepsilon_{r2}\Phi(i,j-1)+\varepsilon_{r3}(\Phi(i+1,j)+\Phi(i-1,j))}{\varepsilon_{r1}+\varepsilon_{r2}+2\varepsilon_{r3}}. \quad (3.20)$$

Еще раз отметим, что данное уравнение используется только для точек на границе, а выше или ниже ее – уравнение (3.14).

Перейдем к определению величины ε_{r3}, полагая, что вклад в величину $\Phi(i,j)$ узловых потенциалов $\Phi(i–1,j)$ и $\Phi(i+1,j)$ вдоль оси y одинаков, т. е.

$$\varepsilon_{r1}\Phi(i,j+1)+\varepsilon_{r2}\Phi(i,j-1)=\varepsilon_{r3}(\Phi(i+1,j)+\Phi(i-1,j)),$$

тогда

$$\varepsilon_{r3}=\frac{\varepsilon_{r1}\Phi(i,j+1)+\varepsilon_{r2}\Phi(i,j-1)}{\Phi(i+1,j)+\Phi(i-1,j)}. \quad (3.21)$$

Прежде чем при вычислениях использовать уравнение (3.20), необходимо на каждой итерации определять ε_{r3} по формуле (3.21). Выражение (3.21) можно упростить, если предположить, что в соседних узлах сетки значения потенциала отличаются незначительно, т. е. $\Phi(i,j+1)\approx\Phi(i,j–1)$, тогда

$$2\varepsilon_{r3}=\varepsilon_{r1}+\varepsilon_{r2}.$$

Описанный метод основан на использовании закона Гаусса в дифференциальной форме. Рассмотрим подход, основанный на использовании закона Гаусса в интегральной форме. Согласно граничным условиям для нормальных составляющих электрического поля, если плотность поверхностных электрических зарядов равна нулю (что соответствует границе раздела диэлектрик-диэлектрик), то $D_{1n}=D_{2n}$. Таким образом, данное выражение справедливо, если

$$\oint_l \mathbf{D}\cdot d\mathbf{l}=\oint_l \varepsilon\mathbf{E}\cdot d\mathbf{l}=q_\Sigma=0.$$

Подставив в это уравнение $\mathbf{E}=–\nabla\Phi$, получим

$$0=\oint_l \varepsilon\mathbf{E}\cdot d\mathbf{l}=\oint_l \varepsilon\frac{\partial\Phi}{\partial n}dl,$$

где $\partial\Phi/\partial n$ – производная по направлению нормали к контуру l (рисунок 3.9), тогда

$$0 = \varepsilon_{r1}\varepsilon_0 \frac{\Phi(i,j+1)-\Phi(i,j)}{h} h +$$

$$+\varepsilon_{r1}\varepsilon_0 \frac{\Phi(i-1,j)-\Phi(i,j)}{h}\frac{h}{2} + \varepsilon_{r2}\varepsilon_0 \frac{\Phi(i-1,j)-\Phi(i,j)}{h}\frac{h}{2} +$$

$$+\varepsilon_{r2}\varepsilon_0 \frac{\Phi(i,j-1)-\Phi(i,j)}{h} h +$$

$$+\varepsilon_{r2}\varepsilon_0 \frac{\Phi(i+1,j)-\Phi(i,j)}{h}\frac{h}{2} + \varepsilon_{r1}\varepsilon_0 \frac{\Phi(i+1,j)-\Phi(i,j)}{h}\frac{h}{2}.$$

После упрощений получим

$$\Phi(i,j) = \frac{\varepsilon_{r1}}{2(\varepsilon_{r1}+\varepsilon_{r2})}\Phi(i,j+1) +$$

$$+\frac{\varepsilon_{r2}}{2(\varepsilon_{r1}+\varepsilon_{r2})}\Phi(i,j-1) + \frac{1}{4}\Phi(i-1,j) + \frac{1}{4}\Phi(i+1,j). \qquad (3.22)$$

Выражение (3.22) эквивалентно (3.20) при $2\varepsilon_{r3} = \varepsilon_{r1} + \varepsilon_{r2}$, а при отсутствии границы диэлектрик-диэлектрик, т. е. $\varepsilon_{r1} = \varepsilon_{r2}$, оно эквивалентно (3.14).

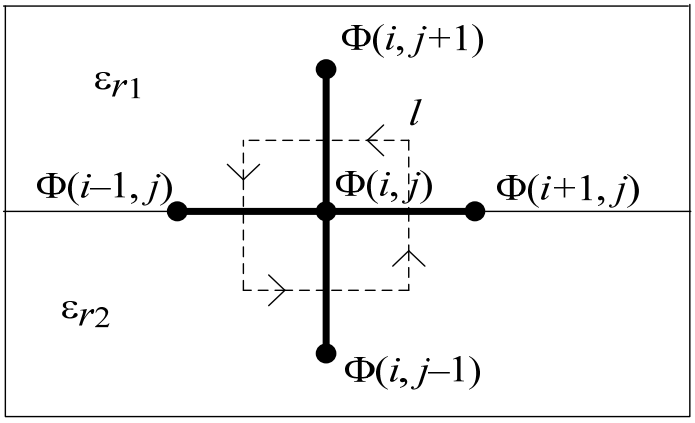


Рисунок 3.9 – Граница раздела диэлектрик-диэлектрик и контур интегрирования

Если в структуре больше границ диэлектрик-диэлектрик, например 4 (рисунок 3.10), используя аналогичный прием, получим

$$\Phi(i,j) = \frac{(\varepsilon_{r1}+\varepsilon_{r2})\Phi(i,j+1) + (\varepsilon_{r1}+\varepsilon_{r4})\Phi(i-1,j)}{2(\varepsilon_{r1}+\varepsilon_{r2}+\varepsilon_{r3}+\varepsilon_{r4})} +$$

$$+\frac{(\varepsilon_{r3}+\varepsilon_{r4})\Phi(i,j-1) + (\varepsilon_{r2}+\varepsilon_{r3})\Phi(i+1,j)}{2(\varepsilon_{r1}+\varepsilon_{r2}+\varepsilon_{r3}+\varepsilon_{r4})}.$$

Таким же образом можно вывести выражение для случая с большим числом диэлектрических границ.

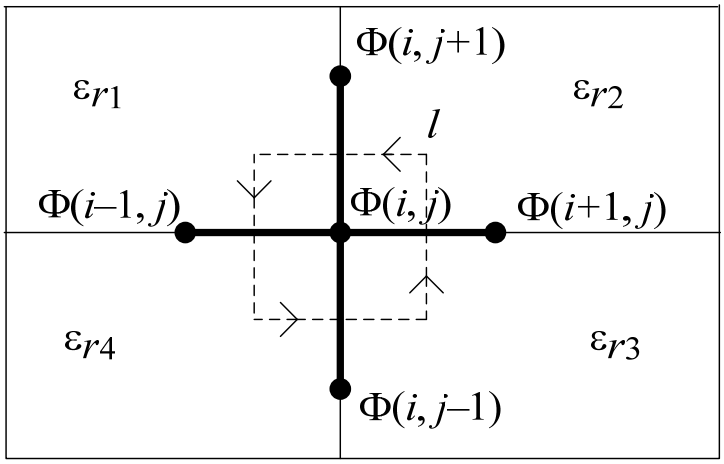
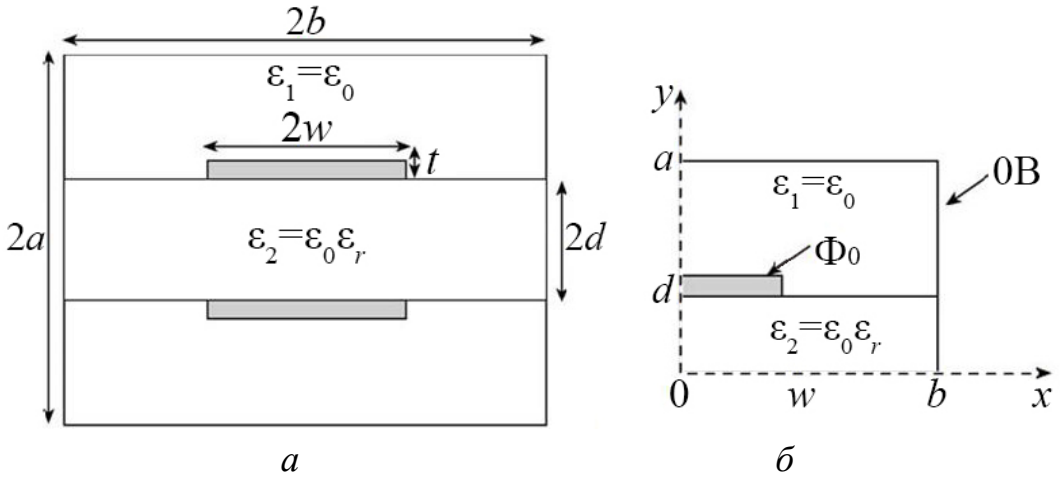


Рисунок 3.10 – Структура со слоистым диэлектриком

Оценим изменения в уравнениях при исследовании симметричных структур. Для примера на рисунке 3.11, *а* приведена экранированная двухполосковая линия передачи (shielded double-strip line with partial dielectric support [32]), а на рисунке 3.11, *б* показана ее часть, полученная с использованием полной симметрии.



а *б*

Рисунок 3.11 – Экранированная двухполосковая линия передачи (*а*) и ее вид при использовании полной симметрии (*б*)

На линии симметрии выполняется условие

$$\frac{\partial \Phi}{\partial n} = 0 .$$

Это означает, что на линии симметрии вдоль оси y ($x = 0$) производная $\dfrac{\partial\Phi}{\partial x} = \dfrac{\Phi(i+1, j) - \Phi(i-1, j)}{2h} = 0$, откуда следует, что $\Phi(i+1, j) = \Phi(i-1, j)$. Тогда выражение (3.14) упрощается:

$$\Phi(i, j) = \frac{1}{4}\left[\Phi(i, j+1) + \Phi(i, j-1) + 2\Phi(i+1, j)\right]. \qquad (3.23)$$

На линии симметрии вдоль оси x ($y = 0$) имеем $\dfrac{\partial\Phi}{\partial y} = \dfrac{\Phi(i, j+1) - \Phi(i, j-1)}{2h} = 0$, тогда $\Phi(i, j+1) = \Phi(i, j-1)$, а выражение (3.14) упрощается:

$$\Phi(i, j) = \frac{1}{4}\left[\Phi(i-1, j) + \Phi(i+1, j) + 2\Phi(i, j+1)\right] \qquad (3.24)$$

Таким образом, для нахождения распределения потенциала в соответствующих узлах используются выражения (3.14), (3.22), (3.23) и (3.24).

Перейдем к определению погонной емкости. В общем случае емкость связана с величиной полного заряда в анализируемой структуре (системе) равенством

$$C = k\frac{q_\Sigma}{\Phi_0}, \qquad (3.25)$$

где Φ_0 – потенциал между внутренним (центральным) и внешним проводниками (см. рисунок 3.11, *б*); k – коэффициент, необходимый для учета симметрии. В рассматриваемом случае он равен 4 (полная симметрия).

Для вычисления полного заряда воспользуемся законом Гаусса для контура l, охватывающего внутренний проводник. Выберем прямоугольный контур между двумя смежными прямоугольными границами, как показано на рисунке 3.12. Тогда

$$q_\Sigma = \oint_l \mathbf{D}\cdot d\mathbf{l} = \oint_l \varepsilon\frac{\partial\Phi}{\partial n}dl = \varepsilon\left(\frac{\Phi_P - \Phi_N}{\Delta x}\right)\Delta y + \varepsilon\left(\frac{\Phi_M - \Phi_L}{\Delta x}\right)\Delta y +$$

$$+ \varepsilon\left(\frac{\Phi_H - \Phi_L}{\Delta y}\right)\Delta x + \varepsilon\left(\frac{\Phi_G - \Phi_K}{\Delta y}\right)\Delta x + \ldots \qquad (3.26)$$

Полагая $\Delta x = \Delta y = h$, получим

$$q_\Sigma = (\varepsilon\Phi_P + \varepsilon\Phi_M + \varepsilon\Phi_H + \varepsilon\Phi_G + \ldots) - (\varepsilon\Phi_N + 2\varepsilon\Phi_L + \varepsilon\Phi_K + \ldots).$$

Если узел расположен на границе диэлектрик-диэлектрик, то $\varepsilon_{ri} = (\varepsilon_{r1} + \varepsilon_{r2})/2$. Если же он расположен на оси симметрии, то Ф = Ф/2, поскольку симметрия учтена в уравнении (3.25).

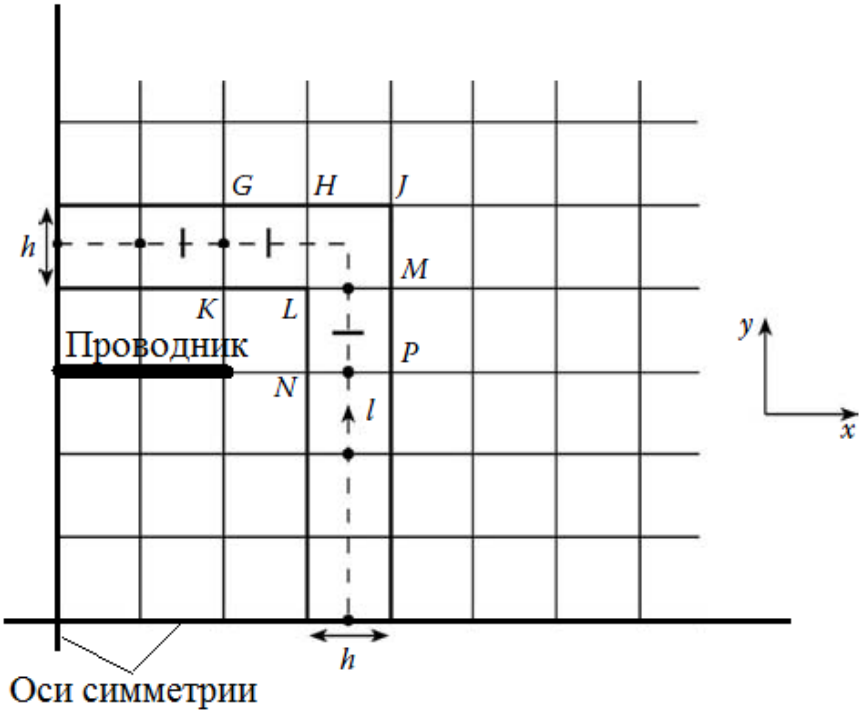


Рисунок 3.12 – Прямоугольный контур *l*, используемый для определения сосредоточенного заряда на поверхности проводника

Далее необходимо аналогично вычислить C_0, удалив диэлектрические границы (положив для всех узлов $\varepsilon_{ri} = 1$). После этого можно вычислить волновое сопротивление по формуле (1.26).

Необходимо отметить, что при вычислениях часто проверяют сходимость не первичного параметра (потенциала) в узлах, а вторичного (емкость), для чего выполняется оценка

$$\text{error} = \max \left| \frac{C_{ij}^{(it)} - C_{ij}^{(it-1)}}{C_{ij}^{(it)}} \right| \le \text{TOL}.$$

Используя равенство $\mathbf{E} = -\nabla\Phi$ для двухмерного случая в виде

$$\mathbf{E} = -(\mathbf{i}(\partial\Phi/\partial x) + \mathbf{j}(\partial\Phi/\partial y)),$$

можно определить напряженность с помощью вычисленных потенциалов. Так, вычислив потенциалы в узлах сетки и используя центральную конечно-разностную аппроксимацию, можно вычислить значения компонентов вектора напряженности, а затем суммарные значения (рисунок 3.13).

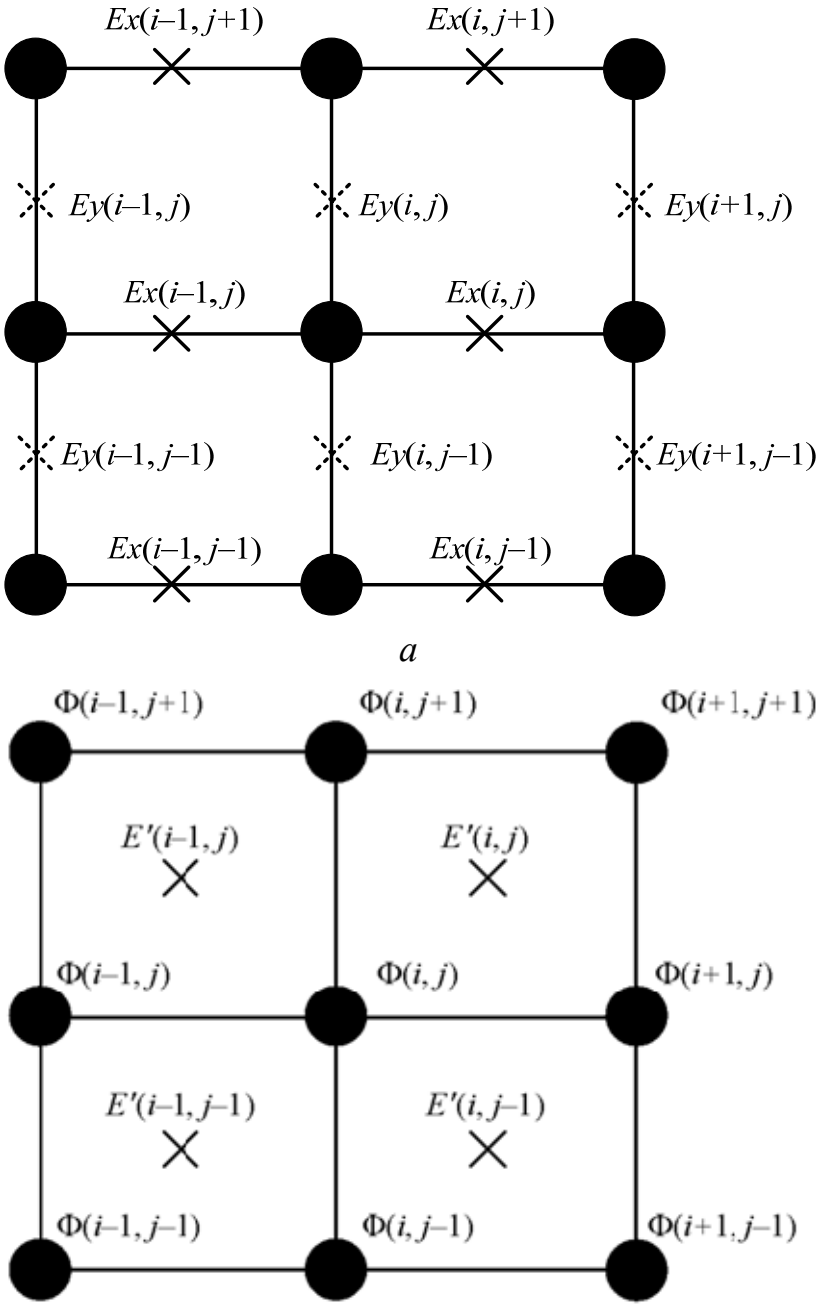


Рисунок 3.13 – Шаблон для вычисления компонентов вектора напряженности электрического поля:
а – центральные разности вдоль осей *x* и *y* для вычисления индивидуальных компонентов вектора; *б* – узлы сетки для вычисления суммарных значений вектора напряженности

Так, компоненты вектора напряженности в соответствии с рисунком 3.13, *а* вычисляются по формулам

$$Ex(i,j) = -\frac{\Phi(i+1,j) - \Phi(i,j)}{h},\ Ey(i,j) = -\frac{\Phi(i,j+1) - \Phi(i,j)}{h}.$$

Далее вычисляются суммарные значения вектора напряженности в узлах новой сетки (количество строк и столбцов этой сетки на единицу меньше количества строк и столбцов сетки для вычисления потенциалов, рисунок 3.13, *б*) как среднее арифметическое значение от индивидуальных значений компонентов вектора:

$$E'x(i,j) = \frac{Ex(i,j+1) + Ex(i,j)}{2},\ E'y(i,j) = \frac{Ey(i+1,j) + Ey(i,j)}{2}.$$

После вычисления значений компонентов вектора напряженности электрического поля можно воспользоваться процедурами Octave (figure и quiver) для наглядного представления полученных результатов.

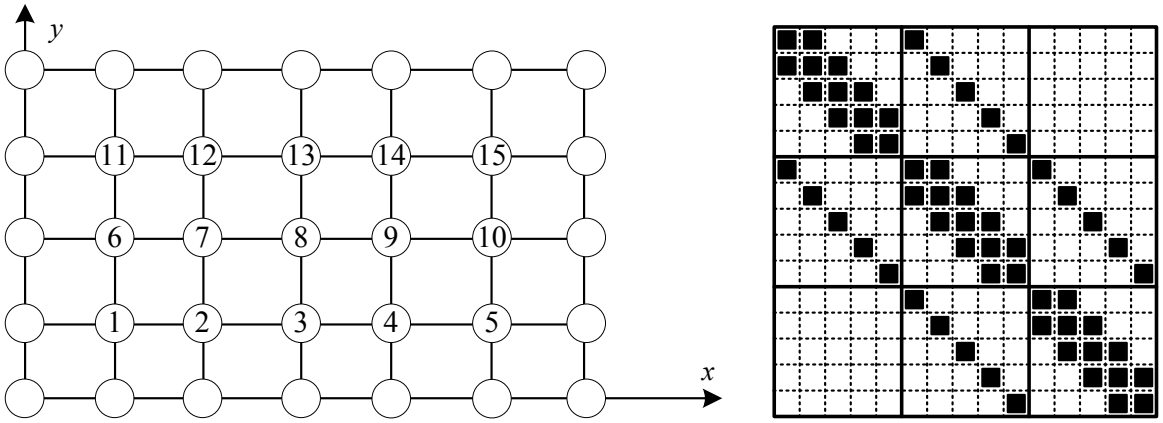
3.5 О нумерации узлов сетки

Еще раз рассмотрим уравнение Лапласа для функции Ф в области Ω с известными на границе Г значениями. Пусть Ω – прямоугольник, а Г – его граница. Пронумеруем только внутренние узлы, поскольку значения функции на границе известны. Точки разметим при движении по горизонтали и снизу вверх. Эта разметка называется естественным упорядочиванием (рисунок 3.14, *а*). Структура результирующей матрицы приведена на рисунке 3.14, *б*.

При такой нумерации узлов возникает зависимость в данных. Так, при вычислении Ф(i, j) требуются значения в соседних узлах как на текущей итерации, так и на предыдущей, что создает проблему для организации параллельных вычислений.

Рассмотрим другой подход к нумерации внутренних узлов, пригодный для параллельной реализации, который называется красно-черным упорядочиванием (рисунок 3.15). В этом случае на каждой итерации производится расчет значений функции сначала

в узлах одного цвета, затем в узлах другого цвета. Причем нумеруются сначала красные узлы, а затем черные.



а *б*

Рисунок 3.14 – Естественное упорядочивание неизвестных для сетки 7×5 (*а*) и соответствующая ей структура матрицы (*б*)

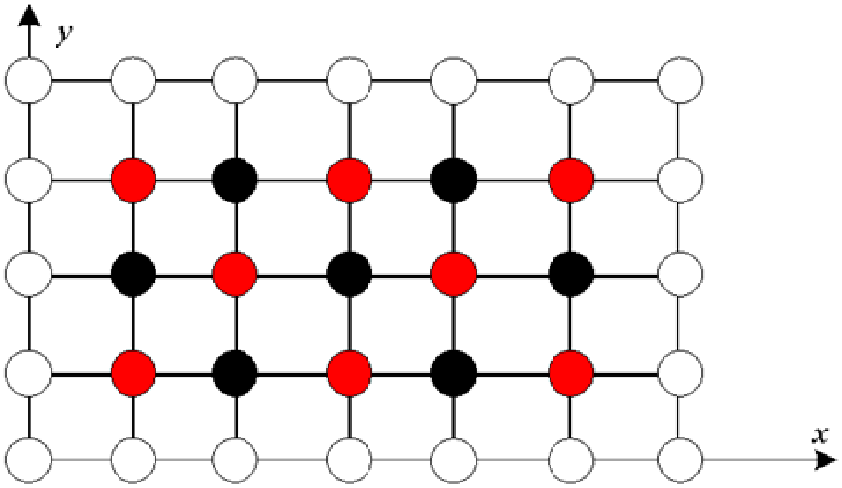


Рисунок 3.15 – Красно-черное упорядочивание неизвестных для сетки 7×5

Заметим, что смежны с каждым красным узлом лишь черные узлы. Поэтому, если вначале перевычисляются компоненты из красных узлов, используются только старые значения из черных узлов. Когда перевычисляются компоненты из черных узлов, которые смежны лишь с красными узлами, используются лишь новые значения из этих красных узлов. Можно сказать, что красно-черное упорядочивание обеспечивает перевод рекуррентных формул метода Зейделя в двухшаговое использование формул Якоби.

Контрольные вопросы и задания

1. Для решения каких уравнений используется метод конечных разностей?

2. Из каких компонентов состоит погрешность решения при использовании конечно-разностной аппроксимации?

3. Как можно повысить точность конечно-разностной аппроксимации?

4. Какой тип конечно-разностной аппроксимации характеризуется меньшей погрешностью?

5. В чем отличие явной и неявной конечно-разностных схем?

6. Для чего используется красно-черное упорядочивание в методе конечных разностей?

7. Разработать программу на языке Octave для решения задачи из примера 3.1.

8. Разработать программу на языке Octave для вычисления волнового сопротивления линии передачи, изображенной на рисунке 3.11.

9. Для линии передачи, изображенной на рисунке 3.11, разработать программу на языке Octave для вычисления значений компонентов вектора напряженности электрического поля и их отображения.

10. Разработать программу на языке Octave для вычисления емкости коаксиальной структуры, показанной на рисунке 3.16 ($a = b = 1$ см, $c = d = 2$ см).

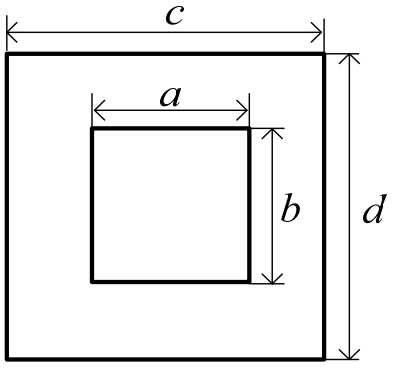


Рисунок 3.16 – Поперечное сечение коаксиальной структуры

4 ВАРИАЦИОННЫЕ МЕТОДЫ

4.1 Операторы в линейных пространствах

При решении задач математической физики, в том числе электростатики, задачу интегрирования дифференциального уравнения часто можно заменить на эквивалентную задачу поиска функции, которая дает минимальное значение некоторого интеграла. Задачи такого типа называются вариационными. Соответственно методы, позволяющие свести проблему интегрирования дифференциального уравнения к эквивалентной вариационной задаче, называются вариационными. Эти методы образуют общую базу для метода моментов и метода конечных элементов, в связи с чем уместно перед изучением последних рассмотреть особенности вариационных методов. Кроме того, решение некоторых дифференциальных и интегральных уравнений относительно легко сформулировать в вариационных терминах. Так, вариационные методы дают точные результаты, не предъявляя чрезмерных требований к вычислительным ресурсам.

Рассмотрим некоторые принципы операторов в линейных пространствах и введем обозначения. Так, скалярное произведение функций u и v определяется выражением

$$\left(u,v\right)=\int_{\Omega} uv^{*}d\Omega, \tag{4.1}$$

где * обозначает комплексное сопряжение, а интегрирование выполняется по области Ω, которая в зависимости от задачи может быть одно-, двух- или трехмерным физическим пространством. В некотором смысле скалярное произведение дает проекцию функции u в направлении v. Если $\mathbf{u}$ и $\mathbf{v}$ – векторные поля, то формулу (4.1) следует модифицировать:

$$\left(\mathbf{u},\mathbf{v}\right)=\int_{\Omega} \mathbf{u}\cdot\mathbf{v}^{*}d\Omega. \tag{4.2}$$

Однако для простоты изложения рассмотрим только случай, когда u и v – комплексные скалярные функции. Тогда для каждой

такой пары функций, принадлежащей линейному пространству, их скалярное произведение обладает следующими свойствами:

$$(u, v) = (v, u)^*; \tag{4.3}$$

$$(u_1 + \beta u_2, v) = \alpha(u_1, v) + \beta(u_2, v); \tag{4.4}$$

$$(u, v) > 0, \text{ если } u \neq 0; \tag{4.5}$$

$$(u, v) = 0, \text{ если } u = 0. \tag{4.6}$$

Если скалярное произведение u и v равно нулю, то говорят, что u и v ортогональны. Следует отметить, что эти свойства имитируют свойства скалярного произведения для трехмерного пространства. Тогда согласно формулам (4.3) и (4.4) получим

$$(u, \alpha v) = \alpha^*(v, u)^* = \alpha^*(u, v), \tag{4.7}$$

где α – комплексное число.

Уравнение (4.1) также называется невзвешенным или стандартным скалярным произведением. Взвешенное скалярное произведение определяется выражением

$$(u, v) = \int_{\Omega} u v^* w d\Omega, \tag{4.8}$$

где w – «подходящая» весовая функция.

Норма функции u определяется как

$$\| u \| = \sqrt{(u, u)}.$$

Норма является мерой величины функции. Что касается поля, то нормой является его среднеквадратичное значение.

Далее определим операторное уравнение

$$L\Phi = g, \tag{4.9}$$

где L – произвольный линейный оператор; Φ – неизвестная функция; g – функция источника (воздействия). Пространство, охватываемое всеми функциями, вытекающими из оператора L, дает

$$(L\Phi, g) = (\Phi, L^* g).$$

Говорят, что оператор L самосопряженный, если $L = L^*$, т. е. $(L\Phi, g) = (\Phi, Lg)$; положительный, если $(L\Phi, \Phi) > 0$ для любого $\Phi \neq 0$ в области определения L; отрицательный, если $(L\Phi, \Phi) < 0$ для любого $\Phi \neq 0$ в области определения L.

Свойства решения уравнения (4.9) сильно зависят от свойств оператора L. Так, если L положительно определенный оператор, то легко показать, что решение единственно. Для этого предположим, что Ф и Ψ являются двумя решениями уравнения (4.9), тогда $L\Phi = g$ и $L\Psi = g$. В силу линейности оператора L разность $f = \Phi - \Psi$ также является решением уравнения. Следовательно, $Lf = 0$. Так как L положительно определенный оператор, то $f = 0$, а значит Ф и Ψ равны между собой, что подтверждает единственность решения.

Пример 4.1

Найти скалярное произведение $u(x) = 1 - x$ и $v(x) = 2x$ на интервале (0, 1).

Решение

Так как u и v – это действительные функции, то

$$(u, v) = (v, u) = \int_0^1 (1 - x)2x dx = 0,33(3).$$

4.2 Вариационное исчисление

Вариационное исчисление является дисциплиной, которая связана прежде всего с отысканием экстремумов функционалов. Для пояснения этого понятия вспомним определение функции. В математике это соответствие между элементами двух множеств, установленное по такому правилу, что каждому элементу одного множества ставится в соответствие некоторый элемент из другого множества. Для зрительной ассоциации приведем следующий пример. Пусть функция задана таблично, тогда каждому значению x соответствует значение y. Таким образом, числу соответствует число.

Теперь вернемся к определению функционала. Рассмотрим небольшой пример [33]. Пусть на плоскости (t, x) заданы две точки (t_0, x_0) и (T, x_T) (рисунок 4.1). Требуется соединить их гладкой кривой, имеющей наименьшую длину. Для этого выделим небольшой участок dt и соответствующий ему участок dx. Длина

кривой на участке dt по теореме Пифагора будет $dl^2 = dx^2 + dt^2$, а общая длина соответственно

$$I\left[x(t)\right] = \int_{t_0}^{T} \sqrt{dx^2 + dt^2} = \int_{t_0}^{T} \sqrt{dt^2\left(\frac{dx^2}{dt^2} + 1\right)} = \int_{t_0}^{T} \sqrt{1 + \left(x'(t)\right)^2}\, dt.$$

Тогда решение задачи сводится к определению такой непрерывной функции $x^*(t)$, которая имеет на отрезке $[t_0, T]$ непрерывную производную и удовлетворяет заданным граничным условиям $x(t_0) = x_0$, $x(T) = x_T$. При этом критерий $I[x(t)]$ имеет минимальное значение.

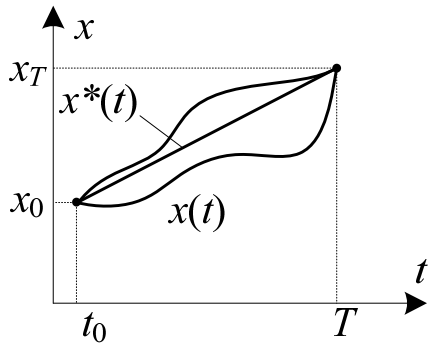


Рисунок 4.1 – К пояснению понятия функционала

Критерий функции $x(t)$ и представляет собой функционал. Очевидно, решением является прямая $x^*(t)$, соединяющая две заданные точки. Таким образом, переменная $I[x(t)]$ называется функционалом, зависящим от функции $x(t)$, если каждой кривой из заданного класса функций соответствует вполне определенное действительное значение I, т. е. функции $x(t)$ соответствует число (функционал – это оператор, множество значений которого состоит из чисел).

Далее будем рассматривать особенности поиска экстремумов интегральных уравнений. В то время как функция производит число в результате представления значений одной или нескольких независимых переменных, функционал в общем виде дает число, которое зависит от всей формы одной или нескольких функций между заданными пределами. В некотором смысле функционал является мерой функции. Простым примером функционала является скалярное произведение. В вариационном исчислении

необходимым является требование, чтобы функционал имел стационарное значение. Это требование обычно имеет вид дифференциального уравнения и соответствующих граничных условий.

Рассмотрим задачу нахождения функции $y(x)$ такой, что функция

$$I(y) = \int_a^b F(x, y, y')dx \tag{4.10}$$

при граничных условиях $y(a) = A$ и $y(b) = B$ оказывается стационарной (кривая $I(y)$ проходит через концы кривой y). Подынтегральная функция $F(x, y, y')$ является заданной функцией, зависящей от x, y и $y' = dy/dx$. В уравнении (4.10) $I(y)$ выражает функциональный или вариационный (стационарный) принцип. Задачей здесь является нахождение экстремальной функции $y(x)$, для которой функционал $I(y)$ имеет экстремум. Прежде чем приступить к решению этой задачи, введем оператор δ, называемый вариационным символом.

Вариацией δy функции $y(x)$ является бесконечно малое изменение y при фиксированном значении независимой переменной x, т. е. $\delta x = 0$. Вариация δy функции y обращается в нуль в точках, где функция y задана (граничные условия), и произвольна в других местах (рисунок 4.2).

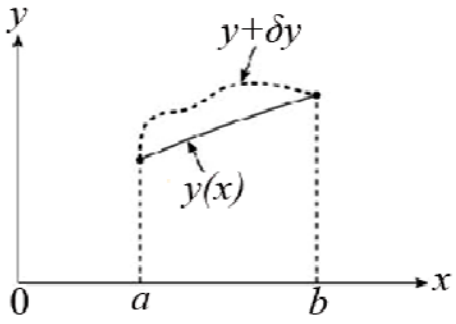


Рисунок 4.2 – Вариация экстремальной функции
на заданном диапазоне

Изменения y (т. е. $y \to y + \delta y$) дают соответствующие изменения F. Первая вариация F определяется как

$$\delta F = \frac{\partial F}{\partial y}\delta y + \frac{\partial F}{\partial y'}\delta y'. \tag{4.11}$$

Выражение (4.11) является аналогом дифференциала

$$dF = \frac{\partial F}{\partial x}dx + \frac{\partial F}{\partial y}dy + \frac{\partial F}{\partial y'}\delta y',$$

где $\delta x = 0$, так как x не изменяется, а y изменяется на величину δy. Таким образом, оператор δ подобен дифференциальному оператору. Тогда, если $F_1 = F_1(y)$ и $F_2 = F_2(y)$, получим ряд свойств:

$$\delta(F_1 \pm F_2) = \delta F_1 \pm \delta F_2 ;$$

$$\delta(F_1 F_2) = F_2 \delta F_1 + F_1 \delta F_2 ;$$

$$\delta\left(\frac{F_1}{F_2}\right) = \frac{F_2 \delta F_1 - F_1 \delta F_2}{F_2^2};$$

$$\delta\left(F_1\right)^n = n\left(F_1\right)^{n-1}\delta\left(F_1\right)^n ;$$

$$\frac{d}{dx}(\delta y) = \delta\left(\frac{dy}{dx}\right);$$

$$\delta\int_a^b y(x)dx = \int_a^b \delta y(x)dx.$$

Чтобы функция $I(y)$ из уравнения (4.10) имела экстремум, ее вариация должна быть равна нулю, т. е.

$$\delta I = 0. \qquad (4.12)$$

Для применения условия (4.12) необходимо найти вариацию I согласно уравнению (4.10). Для этого пусть $h(x)$ является приращением функции $y(x)$. Чтобы удовлетворялись граничные условия из уравнения (4.10) для $y(x) + h(x)$, необходимо выполнить условие

$$h(a) = h(b) = 0.$$

Тогда соответствующее приращение I из уравнения (4.10) вычисляется как

$$\Delta I = I(y+h) - I(y) = \int_a^b \left[F(x, y+h, y'+h') - F(x, y, y')\right]dx .$$

Применив разложение Тейлора (для первого подынтегрального выражения) и воспользовавшись уравнением (4.11), получим

$$\Delta I = \int_a^b \left[\frac{\partial F(x,y,y')}{\partial y} h + \frac{\partial F(x,y,y')}{\partial y'} h' \right] dx + O\left(h^2\right) = \delta I + O\left(h^2\right).$$

Таким образом, δI представляет собой главную линейную часть приращения ΔI, т. е. дифференциал. Интегрируя по частям[1] второе слагаемое, найдем

$$\delta I = \int_a^b \left[\frac{\partial F(x,y,y')}{\partial y} - \frac{d}{dx}\left(\frac{\partial F(x,y,y')}{\partial y'} \right) \right] h dx + \frac{\partial F(x,y,y')}{\partial y'} h \Big|_a^b .$$

Учитывая, что $h(a) = h(b) = 0$, запишем

$$\delta I = \int_a^b \left[\frac{\partial F(x,y,y')}{\partial y} - \frac{d}{dx}\left(\frac{\partial F(x,y,y')}{\partial y'} \right) \right] h dx .$$

С учетом условия (4.12) имеем

$$\frac{\partial F(x,y,y')}{\partial y} - \frac{d}{dx}\left(\frac{\partial F(x,y,y')}{\partial y'} \right) = 0$$

или

$$F_y - \frac{d}{dx} F_{y'} = 0. \qquad (4.13)$$

Полученное выражение называется уравнением Эйлера. Таким образом, для наличия экстремума у функции $y(x)$ необходимо, чтобы она удовлетворяла уравнению Эйлера.

Сказанное выше непосредственно обобщается на случай функционалов, зависящих от нескольких функций. Так, в рассмотренной задаче имелись одна зависимая и одна независимая переменные – y и x соответственно, т. е. $y = y(x)$. Если есть одна зависимая переменная u и две независимые переменные x и y, т.е. $u = u(x, y)$, то

$$I(u) = \int_S F(x, y, u, u_x, u_y) dS, \qquad (4.14)$$

где $u_x = \partial u/\partial x$, $u_y = \partial u/\partial y$, $dS = dxdy$. Функционал (4.14) стационарен, если $\delta I = 0$, и можно показать, что соответствующее уравнение Эйлера имеет вид [34]

---

[1] $\int u dv = uv - \int v du.$

$$\frac{\partial F}{\partial u}-\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial u_x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial F}{\partial u_y}\right)=0. \qquad (4.15)$$

В случае двух независимых переменных x и y и двух зависимых переменных $u(x, y)$ и $v(x, y)$ функционал, подлежащий минимизации, имеет вид

$$I(u,v)=\int_S F(x,y,u,v,u_x,u_y,v_x,v_y)dS, \qquad (4.16)$$

а соответствующее уравнение Эйлера –

$$\begin{aligned}&\frac{\partial F}{\partial u}-\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial u_x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial F}{\partial u_y}\right)=0,\\ &\frac{\partial F}{\partial v}-\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial v_x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial F}{\partial v_y}\right)=0.\end{aligned} \qquad (4.17)$$

Еще одним примером функционала, зависящего от производных второго или более высокого порядка, является

$$I(y)=\int_a^b F\left(x,y,y',y'',...,y^{(n)}\right)dx. \qquad (4.18)$$

В этом случае уравнение Эйлера имеет вид

$$F_y-\frac{d}{dx}F_{y'}+\frac{d^2}{dx^2}F_{y''}-\frac{d^3}{dx^3}F_{y'''}+...+(-1)^n\frac{d^n}{dx^n}F_{y^{(n)}}=0. \qquad (4.19)$$

Все приведенные уравнения Эйлера являются дифференциальными уравнениями.

Пример 4.2

Получить уравнение Эйлера для функционала

$$I(\Phi)=\int_S\left[\frac{1}{2}\left(\Phi_x^2+\Phi_y^2\right)-f(x,y)\Phi\right]dxdy.$$

Решение

Поскольку $F(x,y,\Phi,\Phi_x,\Phi_y)=\frac{1}{2}\left(\Phi_x^2+\Phi_y^2\right)-f(x,y)\Phi$, то имеем одну зависимую переменную Ф и две независимые переменные

x и y. Тогда согласно выражению (4.15) уравнение Эйлера примет вид

$$-f(x,y)-\frac{\partial}{\partial x}\Phi_x-\frac{\partial}{\partial y}\Phi_y=0 \text{ или } \Phi_{xx}+\Phi_{yy}=-f(x,y),$$

т. е. это уравнение Пуассона вида $\nabla^2\Phi=-f(x,y)$. Таким образом, решение уравнения Пуассона соответствует нахождению функции Ф, дающей экстремум заданного функционала $I(\Phi)$.

4.3 Получение функционала из дифференциального уравнения

Выше показано, что уравнение Эйлера – это дифференциальное уравнение, соответствующее функциональному (вариационному) принципу. Рассмотрим обратную процедуру – получение функционала для данного дифференциального уравнения. В общем случае эту процедуру можно представить в виде последовательности действий:

– перемножить операторное уравнение $L\Phi = g$ (уравнение Эйлера) с вариацией $\delta\Phi$ зависимой переменной Ф и проинтегрировать результат по области определения задачи;

– используя формулу Остроградского – Гаусса или интегрирование по частям, «перенести» производные на вариацию $\delta\Phi$;

– выразить пределы интегрирования в терминах граничных условий;

– вынести вариационный оператор δ за знак интеграла.

Для большей ясности проиллюстрируем эту процедуру с помощью следующего примера. Предположим, что необходимо найти вариационный принцип, связанный с уравнением Пуассона

$$\nabla^2\Phi=-f(x,y),$$

что является задачей, обратной рассмотренной в примере 4.2. После выполнения первого пункта алгоритма получим

$$\delta I=\iint\left(-\nabla^2\Phi-f(x,y)\right)\delta\Phi\,dxdy=$$

$$=-\iint\nabla^2\Phi\,\delta\Phi\,dxdy-\iint f\,\delta\Phi\,dxdy=0.$$

Для выполнения второго этапа воспользуемся интегрированием по частям. Для этого положим $u = \delta\Phi, dv = \frac{\partial}{\partial x}\left(\frac{\partial \Phi}{\partial x}\right)dx$, тогда $du = \frac{\partial}{\partial x}(\delta\Phi)dx$, $v = \frac{\partial \Phi}{\partial x}$ и

$$-\int\left[\int \frac{\partial}{\partial x}\left(\frac{\partial \Phi}{\partial x}\right)\delta\Phi\, dx\right]dy = -\int\left[\delta\Phi\frac{\partial \Phi}{\partial x} - \int \frac{\partial \Phi}{\partial x}\frac{\partial}{\partial x}\delta\Phi\, dx\right]dy.$$

После интегрирования получим

$$\delta I = \iint\left(\frac{\partial \Phi}{\partial x}\frac{\partial}{\partial x}\delta\Phi + \frac{\partial \Phi}{\partial y}\frac{\partial}{\partial y}\delta\Phi - \delta f\Phi\right)dxdy - \int \delta\Phi\frac{\partial \Phi}{\partial x}dy - \int \delta\Phi\frac{\partial \Phi}{\partial y}dx =$$

$$= \frac{\delta}{2}\iint\left[\left(\frac{\partial \Phi}{\partial x}\right)^2 + \left(\frac{\partial \Phi}{\partial y}\right)^2 - 2f\Phi\right]dxdy - \delta\int \Phi\frac{\partial \Phi}{\partial x}dy - \delta\int \Phi\frac{\partial \Phi}{\partial y}dx.$$

Последние два слагаемых обнуляются, если на границах заданы однородные условия Дирихле или Неймана. Тогда

$$\delta I = \delta\iint\frac{1}{2}\left(\Phi_x^2 + \Phi_y^2 - 2\Phi f\right)dxdy,$$

и

$$I(\Phi) = \frac{1}{2}\iint\left(\Phi_x^2 + \Phi_y^2 - 2\Phi f\right)dxdy, \qquad (4.20)$$

как и ожидалось.

Рассмотренная процедура нахождения функции $I(\Phi)$ соответствующего операторного уравнения (4.9) имеет альтернативу. Так, если оператор L – вещественный, положительно определенный и самосопряженный, то задача решения уравнения (4.9) эквивалентна задаче минимизации функционала [35]

$$I(\Phi) = (L\Phi, \Phi) - 2(\Phi, g). \qquad (4.21)$$

Таким образом, уравнение (4.20) может быть решено с помощью уравнения (4.21). Данный подход используется и для решения интегральных уравнений.

Пример 4.3

Найти функционал для дифференциального уравнения

$$\frac{d^2y}{dx^2} + y + x = 0,\ 0 < x < 1$$

при условии, что $y(0) = y(1) = 0$.

Решение

Поскольку $\delta I = 0$, тогда

$$\delta I = \int_0^1 \left(\frac{d^2y}{dx^2} + y + x \right) \delta y\, dx = \int_0^1 \frac{d^2y}{dx^2} \delta y\, dx + \int_0^1 y\, \delta y\, dx + \int_0^1 x\, \delta y\, dx = 0.$$

Интегрирование по частям дает

$$\delta I = \delta y \frac{dy}{dx}\Bigg|_{x=0}^{x=1} - \int_0^1 \frac{dy}{dx} \frac{d}{dx} \delta y + \int_0^1 \frac{1}{2} \delta \left(y^2 \right) dx + \delta \int_0^1 x\, y\, dx.$$

Поскольку величина y фиксирована в точках $x = 0$ и $x = 1$, то $\delta y(0) = \delta y(1) = 0$. Тогда

$$\delta I = -\delta \int_0^1 \frac{1}{2} \left(\frac{dy}{dx} \right)^2 dx + \frac{1}{2} \delta \int_0^1 y^2\, dx + \delta \int_0^1 x\, y\, dx = \frac{\delta}{2} \int_0^1 \left(-y'^2 + y^2 + 2xy \right) dx.$$

В результате получим

$$I(y) = \frac{1}{2} \int_0^1 \left(-y'^2 + y^2 + 2xy \right) dx.$$

4.4 Метод Рэлея – Ритца

Метод Рэлея – Ритца является вариационным методом минимизации заданного функционала, дающим решение вариационной задачи без обращения к связанному с ней дифференциальному уравнению. Другими словами, это прямое применение вариационных принципов, обсуждавшихся выше. Метод был впервые предложен Рэлеем в 1877 г. и развит Ритцем в 1909 г. Для упрощения изложения (без потери общности рассуждений) рассмотрим функционал

$$I\left(\Phi\right)=\int_{S}F\left(x,y,\Phi,\Phi_{x},\Phi_{y}\right)dS. \qquad (4.22)$$

Требуется минимизировать этот интеграл. При использовании метода Рэлея – Ритца составляется линейно независимый набор базисных функций φ_n (координатных элементов [35]) и строится приближенное решение уравнения (4.20), удовлетворяющее некоторым заданным граничным условиям, т. е. решение ищется в виде конечной линейной комбинации этих функций

$$\hat{\Phi}\approx\sum_{n=1}^{N}a_n\varphi_n+\varphi_0, \qquad (4.23)$$

где φ_0 удовлетворяет неоднородным граничным условиям; φ_n – однородным граничным условиям[1]; a_n – некоторые константы (коэффициенты разложения), определяемые из условия наилучшего приближения функции $\hat{\Phi}$ к точному решению – функции Ф. Подставив (4.23) в (4.22), конвертируем интеграл $I(\Phi)$ в функцию, зависящую от N коэффициентов:

$$I\left(\Phi\right)=I(a_1,a_2,...,a_N).$$

Минимум этой функции достигается, когда ее частные производные по каждому коэффициенту равны нулю, т. е.

$$\frac{\partial I}{\partial a_n}=0,\ \ n=1,\,2,\,...,N. \qquad (4.24)$$

Таким образом формируется набор из N независимых уравнений и полученная СЛАУ решается для нахождения коэффициентов a_n, которые затем подставляются в выражение (4.23). Если решение $\hat{\Phi}\to\Phi$ при $N\to\infty$, то говорят, что процесс сходится к точному решению.

Рассмотрим одну из альтернативных процедур вычисления коэффициентов a_n [35]. Подставим уравнение (4.23) (игнорируя φ_0, так как оно может быть учтено в правой части СЛАУ)

[1] Условие (начальное или граничное) называется однородным, если сумма любых двух функций u_1 и u_2, удовлетворяющих условию (начальному или граничному), также удовлетворяет этому условию.

в уравнение (4.21). Это преобразует $I(\Phi)$ в функцию из N независимых переменных $a_1, a_2, \ldots, a_N$. В результате получим

$$I = \left(\sum_{m=1}^{N} a_m L\varphi_m, \sum_{n=1}^{N} a_n \varphi_n \right) - 2\left(\sum_{m=1}^{N} a_m \varphi_m, g \right) =$$

$$= \sum_{m=1}^{N} \sum_{n=1}^{N} \left(L\varphi_m, \varphi_n \right) a_n a_m - 2\sum_{m=1}^{N} \left(\varphi_m, g \right) a_m .$$

Так как нас интересует выбор a_m, минимизирующий интеграл I, то полученное уравнение должно удовлетворять условию (4.24). Продифференцировав его по a_m и приравняв результат к нулю, получим систему уравнений

$$\sum_{n=1}^{N} (L\varphi_m, \varphi_n) a_n = (g, \varphi_m), m = 1, 2, ..., N, \qquad (4.25)$$

или

$$\begin{pmatrix} (L\varphi_1, \varphi_1) & (L\varphi_1, \varphi_2) & ... & (L\varphi_1, \varphi_N) \\ (L\varphi_2, \varphi_1) & (L\varphi_2, \varphi_2) & ... & (L\varphi_2, \varphi_N) \\ ... & ... & ... & ... \\ (L\varphi_N, \varphi_1) & (L\varphi_N, \varphi_2) & ... & (L\varphi_N, \varphi_N) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ ... \\ a_N \end{pmatrix} = \begin{pmatrix} (g, \varphi_1) \\ (g, \varphi_2) \\ ... \\ (g, \varphi_N) \end{pmatrix}. \qquad (4.26)$$

Решив СЛАУ (4.25) и подставив вектор решения, состоящий из коэффициентов $a_1, a_2, \ldots, a_N$, в уравнение (4.23), найдем требуемое решение $\hat{\Phi}$. Систему уравнений (4.26) иногда называют системой Рэлея – Ритца.

Таким образом, базисные функции выбираются из условия согласования с граничными условиями. Метод Рэлея – Ритца имеет два ограничения. Первое заключается в том, что вариационная формулировка согласно уравнению (4.22) может не существовать при решении некоторых задач. Второе состоит в том, что сложно, а иногда и невозможно найти функцию φ_0, соответствующую граничным условиям для областей со сложной геометрией. Далее на примерах подробнее рассмотрим особенности выбора базисных функций.

Пример 4.4

Используя метод Рэлея – Ритца, решить обыкновенное дифференциальное уравнение

$$\Phi'' + 4\Phi - x^2 = 0,\ 0 < x < 1,$$

удовлетворяющее граничным условиям $\Phi(0) = \Phi(1) = 0$.

Решение

Точное решение данного уравнения известно как

$$\Phi(x) = \frac{\sin 2(1-x) - \sin 2x}{8\sin 2} + \frac{x^2}{4} - \frac{1}{8}.$$

Для исходного уравнения имеем

$$I(\Phi) = \int_0^1 \left[\left(\Phi'\right)^2 - 4\Phi^2 + 2x^2\Phi \right] dx.$$

Будем искать приближенное решение в виде

$$\widehat{\Phi} = \sum_{n=1}^{N} a_n \varphi_n + \varphi_0$$

при $\varphi_0 = 0$ и $\varphi_n = x^n(1-x)$, удовлетворяющих заданным граничным условиям. Следует отметить, что выбор таких базисных функций не единственен. Могут быть выбраны, например, $\varphi_n = x\left(1-x^n\right)$ или $\varphi_n = \sin n\pi x$, которые также удовлетворяют заданным граничным условиям. Для нахождения коэффициентов разложения a_n можно воспользоваться двумя способами: применить функционал напрямую согласно уравнению (4.24); решить систему (4.26). Сначала используем первый способ.

При $N = 1$ получим $\widehat{\Phi} = a_1\varphi_1 = a_1 x(1-x)$. Подстановка в интеграл $I(\Phi)$ дает

$$I(a_1) = \int_0^1 \left[a_1^2(1-2x)^2 - 4a_1^2\left(x - x^2\right)^2 + 2a_1 x^3(1-x) \right] dx = a_1^2/5 + a_1/10.$$

Функционал $I(a_1)$ минимален, когда $\dfrac{\partial I}{\partial a_1} = 0$, что достигается при $a_1 = 0$ или $a_1 = -0{,}25$. Таким образом, находим приближенное решение

$$\widehat{\Phi} = -\frac{1}{4}x(1-x).$$

При $N = 2$ получим $\widehat{\Phi} = a_1\varphi_1 + a_2\varphi_2 = a_1x(1-x) + a_2x^2(1-x)$ и

$$I(a_1,a_2) = \int_0^1 \left\{ \left[a_1(1-2x) + a_2\left(2x-3x^2\right)\right]^2 - 4\left[a_1\left(x-x^2\right) + a_2\left(x^2-x^3\right)\right]^2 + \right.$$

$$\left. + 2a_1x^2\left(x-x^2\right) + 2a_1x^2\left(x^2-x^3\right)\right\} dx =$$

$$= a_1^2/5 + 2a_2^2/21 + a_1a_2 + a_1/10 + a_2/15.$$

При $\dfrac{\partial I}{\partial a_1} = 0$ имеем $4a_1 + 2a_2 = -1$, а при $\dfrac{\partial I}{\partial a_2} = 0$ имеем $21a_1 + 20a_2 = -7$. Решив систему из этих двух уравнений с двумя неизвестными, найдем $a_1 = -6/38$, $a_2 = -7/38$. Таким образом, приближенное решение в данном случае имеет вид

$$\widehat{\Phi} = \frac{x}{38}(7x^2 - x - 6).$$

Рассмотрим второй способ решения. Сформируем систему вида (4.26). В данном случае

$$L = \frac{d^2}{dx^2} + 4,\ g = x^2.$$

Тогда

$$s_{mn} = (L\varphi_m, \varphi_n) = (\varphi_m, L\varphi_n) =$$

$$= \frac{n(n-1)}{m+n-1} - \frac{2n^2}{m+n} + \frac{n(n+1)+4}{m+n+1} - \frac{8}{m+n+2} + \frac{4}{m+n+3},$$

$$b_n = (g, \varphi_n) = \int_0^1 x2n^n(1-x)dx = \frac{1}{n+3} - \frac{1}{n+4}.$$

При $N = 1$ получим $s_{11} = -1/5$, $b_1 = 1/20$, тогда, как и ранее, $a_1 = -0{,}25$. При $N = 2$ получим $s_{11} = -1/5$, $s_{21} = s_{12} = -1/10$, $s_{22} = -2/21$, $b_1 = 1/20$, $b_2 = 1/30$. В результате, сформировав СЛАУ

$$\begin{pmatrix} -1/5 & -1/10 \\ -1/10 & -2/21 \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1/20 \\ 1/30 \end{pmatrix}$$

и решив ее, находим $a_1 = -6/38$, $a_2 = -7/38$.

В таблице 4.1 приведено сравнение точного решения с решением, полученным методом Рэлея – Ритца.

Таблица 4.1 – Решение дифференциального уравнения методом Рэлея – Ритца

x	Точное решение	Метод Рэлея – Ритца	
		$N = 1$	$N = 2$
0.0	0.0	0.0	0.0
0.2	–0.0301	–0.0400	–0.0312
0.4	–0.0555	–0.0600	–0.0556
0.6	–0.0625	–0.0625	–0.0644
0.8	–0.0489	–0.0400	–0.0488
1.0	0.0	0.0	0.0

Пример 4.5

Используя метод Релея – Ритца, решить уравнение Пуассона

$$\nabla^2\Phi = -\rho_0,\ \rho_0 = \text{const},$$

при $-1 \le x \le 1$, $-1 \le y \le 1$ и $\Phi(x, \pm 1) = \Phi(y, \pm 1) = 0$.

Решение

За счет симметрии задачи используем базисные функции вида

$$\varphi_{mn} = \left(1 - x^2\right)\left(1 - y^2\right)\left(x^{2m}y^{2n} + x^{2n}y^{2m}\right),\ \ m, n = 0, 1, 2, \ldots$$

Тогда

$$\hat{\Phi} = \left(1 - x^2\right)\left(1 - y^2\right)\left(a_1 + a_2\left(x^2 + y^2\right) + a_3x^2y^2 + a_4\left(x^4 + y^4\right) + \ldots\right).$$

При $m = n = 0$ получим первое приближение ($N = 1$) в виде

$$\hat{\Phi} = a_1\varphi_1,$$

где $\varphi_1 = \left(1 - x^2\right)\left(1 - y^2\right)$. Тогда

$$s_{11} = \left(L\varphi_1, \varphi_1\right) = \int_{-1}^{1}\int_{-1}^{1}\left(\frac{\partial^2\varphi_1}{\partial x^2} + \frac{\partial^2\varphi_1}{\partial y^2}\right)\varphi_1 dxdy =$$

$$= -8\int_0^1\int_0^1\left(2 - x^2 - y^2\right)\left(1 - x^2\right)\left(1 - y^2\right)dxdy = -256/45,$$

$$b_1 = (g, \varphi_1) = -\int_{-1}^{1}\int_{-1}^{1}\left(2 - x^2 - y^2\right)\left(1 - x^2\right)\left(1 - y^2\right)dxdy =$$

$$= -8\int_{0}^{1}\int_{0}^{1}\left(1 - x^2\right)\left(1 - y^2\right)\rho_0 dxdy = -\frac{16}{9}\rho_0.$$

В результате имеем

$$-256a_1/45 = -16\rho_0/9 \;\rightarrow\; a_1 = 5\rho_0/16$$

и

$$\hat{\Phi} = \frac{5}{16}\rho_0\left(1 - x^2\right)\left(1 - y^2\right).$$

При $m = n = 1$ получим первое приближение ($N = 2$) в виде

$$\hat{\Phi} = a_1\varphi_1 + a_2\varphi_2,$$

где $\varphi_1 = \left(1 - x^2\right)\left(1 - y^2\right)$ и $\varphi_2 = \left(1 - x^2\right)\left(1 - y^2\right)\left(x^2 + y^2\right)$. Значения s_{11} и b_1 такие же, как при $N = 1$. При этом

$$s_{21} = s_{12} = (L\varphi_1, \varphi_2) = -1024/525,$$

$$s_{22} = (L\varphi_2, \varphi_2) = -1124/4725,$$

$$b_2 = (g, \varphi_2) = -32\rho_0/45.$$

В результате имеем СЛАУ

$$\begin{pmatrix} -252/45 & -1024/525 \\ -1024/525 & -11264/4725 \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} -16\rho_0/9 \\ -32\rho_0/45 \end{pmatrix},$$

решение которой дает $a_1 = 0,2922\rho_0$, $a_2 = 0,0592\rho_0$. Тогда

$$\hat{\Phi} = \left(1 - x^2\right)\left(1 - y^2\right)\left(0,2922 + 0,0592\left(x^2 + y^2\right)\right)\rho_0.$$

Пример 4.6

Вычислить емкость экранированной МПЛ (рисунок 4.3).

Если положить, что в данной структуре распространяется только поперечная электромагнитная волна (квазистатическое приближение), то для решения поставленной задачи требуется решить уравнение Лапласа вида

$$\nabla^2\Phi = 0.$$

Будем использовать полную геометрическую симметрию структуры и полярную систему координат (см. рисунок 4.3, *б*).

При этом добавится граничное условие $\partial\Phi/\partial x = 0$ при $x = -w$. Допустим сингулярность на краю полоски.

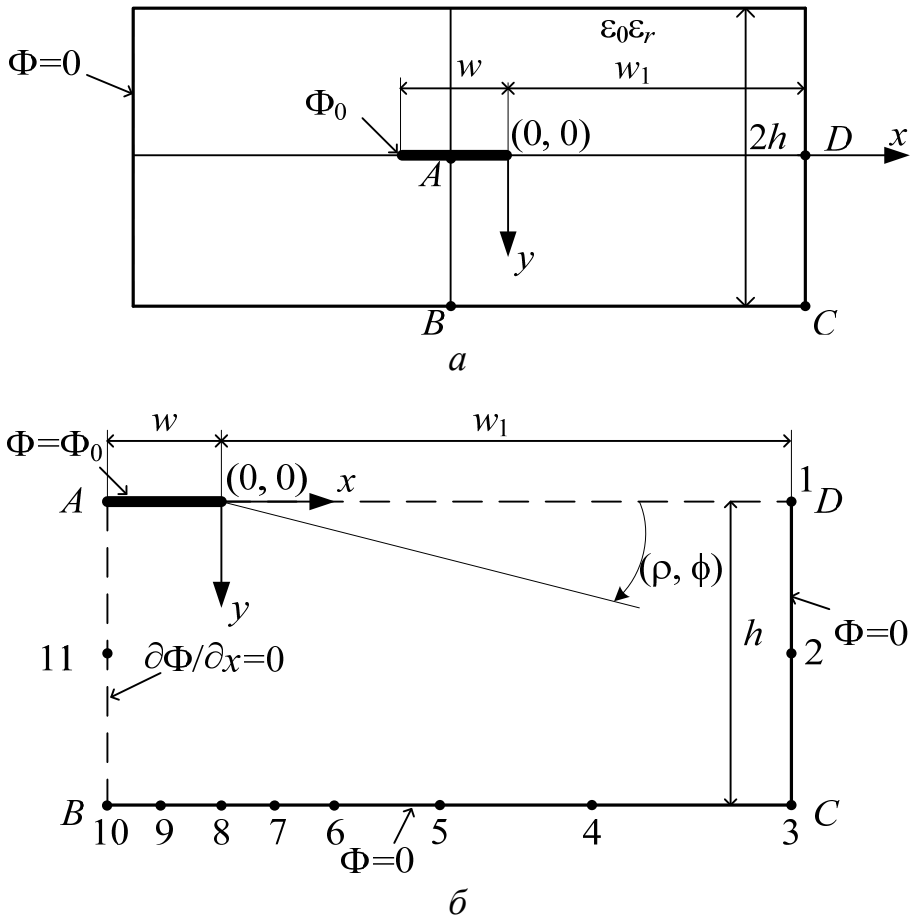


Рисунок 4.3 – Поперечное сечение экранированной МПЛ (*а*) и ее четверть (*б*)

Тогда вариацию потенциала в окрестности этой сингулярности аппроксимируем с помощью тригонометрических базисных функций:

$$\widehat{\Phi} = \Phi_0 + \sum_{k=1,\,3,5}^{\infty} c_k \rho^{k/2} \cos\frac{k\varphi}{2}, \qquad (4.27)$$

где Φ_0 – потенциал полоски. Коэффициенты c_k требуется вычислить.

Если ограничить бесконечный ряд в данном уравнении до N слагаемых, это будет эквивалентно требованию его выполнения в M ($\geq N$) точках на границе. Применив аппроксимацию для каждой из M граничных точек, получим СЛАУ из M уравнений

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_M \end{pmatrix} = \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \dots \\ \Phi_M \end{pmatrix} \text{ или } \mathbf{Ax} = \mathbf{b}.$$

Вектор решения **x** нельзя однозначно определить из данной переопределенной СЛАУ (при $M > N$). Поэтому определим невязку

$$\mathbf{r} = \mathbf{Ax} - \mathbf{b}$$

и воспользуемся методом наименьших квадратов. Далее будем искать **x**, минимизирующий квадрат невязки $\mathbf{r}^2$. Для этого рассмотрим

$$\mathbf{r}^2 = \mathbf{r}^t\mathbf{r} = (\mathbf{Ax} - \mathbf{b})^t(\mathbf{Ax} - \mathbf{b}).$$

Тогда

$$\frac{\partial \mathbf{r}^2}{\partial \mathbf{x}} = 0 \;\rightarrow\; \mathbf{A}^t\mathbf{Ax} - \mathbf{A}^t\mathbf{b} = 0$$

или

$$\mathbf{x} = \left(\mathbf{A}^t\mathbf{A}\right)^{-1}\mathbf{A}^t\mathbf{b},$$

где индекс *t* обозначает операцию транспонирования соответствующей матрицы. Таким образом, вместо переопределенной СЛАУ требуется решение СЛАУ из *N* уравнений с *N* неизвестными. Решив ее, получим аппроксимирующее решение $\hat{\Phi}$. После этого можно приступить к вычислению погонной емкости линии при заданном отношении ширины к высоте структуры. Емкость линии вычисляется с помощью выражения $C = Q/\Phi_0 = Q,\ \Phi_0 = 1$. Для вычисления заряда *Q* разобьем границу *BCD* (см. рисунок 4.3, *б*) на сегменты, тогда

$$Q = \int \sigma_L dl = 4\sum_{BCD} \sigma_L \Delta l = 4\left(\sum_{BC} \sigma_L \Delta l + \sum_{CD} \sigma_L \Delta l\right), \qquad (4.28)$$

где поверхностная плотность заряда $\sigma_L = \mathbf{D}\cdot\mathbf{a}_n = \varepsilon_0\varepsilon_r\mathbf{E}\cdot\mathbf{a}_n$, $\mathbf{E} = -\nabla \mathrm{V}$, а коэффициент 4 введен из-за использования полной геометрической симметрии. В полярных координатах имеем

$$\nabla\Phi = \frac{\partial\Phi}{\partial\rho}\mathbf{a}_\rho + \frac{1}{\rho}\frac{\partial\Phi}{\partial\varphi}\mathbf{a}_\varphi,$$

$$\mathbf{E} = -\sum_{k\text{-нечет}} \frac{k}{2} c_k \rho^{k/2-1}\left(\cos\frac{k\varphi}{2}\mathbf{a}_\rho - \sin\frac{k\varphi}{2}\mathbf{a}_\varphi\right).$$

Поскольку $\mathbf{a}_x = \cos\varphi\mathbf{a}_\rho - \sin\varphi\mathbf{a}_\varphi$ и $\mathbf{a}_y = \sin\varphi\mathbf{a}_\rho + \cos\varphi\mathbf{a}_\varphi$, то

$$\sigma_L\,|_{BC} = \varepsilon\mathbf{E}\cdot\mathbf{a}_y = -\varepsilon\sum_{k\text{-нечет}} \frac{k}{2} c_k \rho^{k/2-1}\left(\cos\frac{k\varphi}{2}\sin\varphi - \sin\frac{k\varphi}{2}\cos\varphi\right),$$

$$\sigma_L\,|_{CD} = \varepsilon\mathbf{E}\cdot\mathbf{a}_x = -\varepsilon\sum_{k\text{-нечет}} \frac{k}{2} c_k \rho^{k/2-1}\left(\cos\frac{k\varphi}{2}\cos\varphi + \sin\frac{k\varphi}{2}\sin\varphi\right).$$

Теперь при известных параметрах структуры достаточно легко можно найти значение заряда и соответственно емкость линии.

Контрольные вопросы и задания

1. Поясните, что такое функционал и вариация функции.

2. Для решения каких уравнений применяют метод Рэлея – Ритца?

3. Опишите процедуру получения функционала для заданного дифференциального уравнения.

4. Назовите обязательное условие наличия экстремума функции.

5. Решить задачу из примера 4.4 при $N = 3$.

6. Разработать программу на языке Octave для решения задачи из примера 4.6.

5 МЕТОД МОМЕНТОВ

5.1 Общие сведения

Метод моментов (МоМ) – численный метод формирования матричных уравнений [36]. Для пояснения идеи метода рассмотрим обобщенную задачу (детерминированное уравнение)

$$Lf = g \text{ в области } \Omega, \tag{5.1}$$

где L – линейный оператор (дифференциальный, интегральный или интегро-дифференциальный); g – известная функция; f – искомая неизвестная функция. Область Ω представляет собой пространственную область, описываемую координатами. Функция f может быть как скалярной, так и векторной. Пусть f представлена линейной комбинацией N базисных функций f_n в области действия L, т. е.

$$f = \sum_{n=1}^{N} a_n f_n, \tag{5.2}$$

где a_n – неизвестные коэффициенты. Очевидно, что для приближенного решения (5.2) является конечной суммой, а для точного – бесконечной. Подставив выражение (5.2) в уравнение (5.1) и используя линейность L, получим

$$\sum_{n=1}^{N} a_n L f_n \approx g. \tag{5.3}$$

Тогда невязка

$$R = g - \sum_{n=1}^{N} a_n L f_n \, .$$

Базисные функции выбираются так, чтобы они моделировали ожидаемое поведение неизвестной функции по всей ее области, и они могут быть скалярными или векторными в зависимости от решаемой задачи. Если базисные функции определены на локальных областях, они называются локальными базисными функциями или базисными функциями подобластей. Если они определены на всей области, то их называют глобальными базисными функциями

или базисными функциями полной области. (Далее рассматриваются только локальные функции.)

В области определения L составим систему тестовых (весовых) функций w_m и зададим соответствующее скалярное произведение [37]. Так, для функций f и w оно задается в виде

$$(f, w) = \int f w dL,$$

где интеграл может быть линейным (как в рассматриваемом случае), поверхностным или объемным в зависимости от используемых функций.

Для точного решения требуется, чтобы невязка была равна нулю, а это соответствует равенству нулю скалярного произведения каждой тестовой функции и невязки, т. е.

$$(w_m, g) - \sum_{n=1}^{N} a_n (w_m, Lf_n) = 0 \text{ или } \sum_{n=1}^{N} a_n (w_m, Lf_n) = (w_m, g).$$

Это уравнение соответствует СЛАУ **Ax** = **b** с матрицей порядка N, где элементы матрицы и правой части вычисляются соответственно как

$$a_{mn} = (w_m, Lf_n) \text{ и } b_m = (w_m, g).$$

При использовании того или иного численного метода важна скорость его сходимости и точность получаемых результатов. Сходимость МоМ напрямую зависит от оператора L, базисных f_n и тестовых w_m функций, а также их числа N. При этом эффективность применения метода для получения результата с заданной точностью определяется вычислительными затратами (времени и памяти используемой рабочей станции). При одновариантном анализе использование МоМ сводится к следующим этапам:

– получение из уравнений Максвелла интегрального уравнения для заданной структуры;

– построение сетки (разбиение границ структуры на N подобластей и аппроксимация искомой функции в каждой из них соответствующей базисной функцией);

– формирование СЛАУ (вычисление элементов матрицы порядка N и элементов правой части);

– решение сформированной СЛАУ;

– вычисление требуемых характеристик структуры из полученного решения СЛАУ.

Таким образом, суть метода заключается в том, что неизвестная величина (например, поле или плотность тока, зависящая от пространственных координат) аппроксимируется конечным рядом известных функций (называемых базисными), умноженных на неизвестные коэффициенты. Это приближение подставляется в линейное операторное уравнение. Левую и правую части полученного уравнения умножают на подходящую функцию (называемую тестовой или весовой функцией) и интегрируют по области, в которой определена тестовая функция. В результате линейное операторное уравнение сводится к линейному алгебраическому уравнению. Повторяя эту процедуру для набора независимых тестовых функций, число которых должно равняться числу базисных функций, получают СЛАУ. Решение СЛАУ дает неизвестные коэффициенты и позволяет найти приближенное решение операторного уравнения.

В версии МоМ, предложенной Харрингтоном, используются кусочно-постоянные (импульсные) функции в виде базисных и функции Дирака в виде тестовых. Она известна также как метод коллокаций. Харрингтон в работе [37] выполнил обобщение, позволяющее считать методы коллокаций, Галёркина и наименьших квадратов частным случаем МоМ. Однако в прикладной математике этот подход принято называть по-другому. Так, в 1956 г. Crandall предложил термин «метод взвешенных невязок» (МВН), обобщив под ним целое семейство методов [38]. Под невязкой уравнения $Lf = g$ подразумевается выражение

$$R = L(\underline{f} - f) = L\underline{f} - g,$$

где $\underline{f}$ – функция, аппроксимирующая функцию f с помощью набора базисных функций. Далее накладывается условие ортогональности (взятие скалярного произведения) невязки всем тестовым функциям (в случае метода Галёркина – базисным функциям).

В таблице 5.1 приведены основные этапы становления МВН. Из таблицы видно, что в его развитие существенный вклад внесли отечественные ученые (рисунок 5.1) [22].

Таблица 5.1 – Основные этапы разработки метода взвешенных невязок

Год	Разработчик(и)	Метод
1915	Галёркин Б. Г.	Метод Галёркина (Бубнова – Галёркина[1])
1921	Karman T. Pohlhausen E.	Интегральный метод (Кармана – Польгаузена)
1923	Biezeno C. B. Koch J. J.	Метод подобластей
1926	Крылов Н. М.	Метод наименьших квадратов
1926–1932	Крылов Н. М. Кравчук М. Ф.	Метод моментов
1933	Канторович Л. В.	Метод приведения к обыкновенным дифференциальным уравнениям
1937	Канторович Л. В. и Frazer R. A. Jones W. P., Skan S. V.	Интерполяционный метод, метод коллокаций
1940	Репман Ю. В.	Обоснование метода Бубнова – Галёркина применительно к интегральным уравнениям типа Фредгольма
1940	Петров Г. И.	Обоснование метода Бубнова – Галёркина применительно к дифференциальным уравнениям
1942	Келдыш М. В.	Доказательство сходимости метода Бубнова – Галёркина для стационарных задач
1947	Yamada H.	Метод моментов
1948	Михлин С. Г.	Общий признак сходимости метода Бубнова – Галёркина
1949	Faedo S.	Доказательство сходимости метода Бубнова – Галёркина для нестационарных задач
1953	Green J. W.	Оценки сходимости и погрешности метода Бубнова – Галёркина для нестационарных задач
1956	Crandall S. H.	Обобщение методов, МВН

[1] Бубнов И.Г. предложил схожий метод (Михлин С.Г. Вариационные методы в математической физике. М.: Гостехиздат, 1957. 476 с.). Поэтому в литературе метод Галёркина часто называют методом Бубнова – Галёркина.

Иван Григорьевич Бубнов

Борис Григорьевич Галёркин

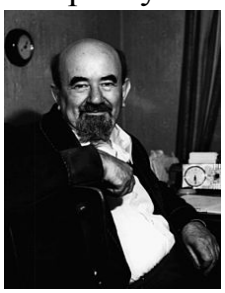
Николай Митрофанович Крылов

Михаил Филиппович Кравчук

Мстислав Всеволодович Келдыш

Леонид Витальевич Канторович

Григорий Иванович Петров



Соломон Григорьевич Михлин

Рисунок 5.1 – Отечественные ученые,
внесшие существенный вклад в развитие МВН

Рассмотрим пример, демонстрирующий универсальность МоМ. Как было сказано, этот метод широко применим при решении интегральных уравнений. Однако он может использоваться и для решения дифференциальных уравнений.

Пример 5.1

Решить линейное дифференциальное уравнение первого рода вида

$$\frac{dy}{dx} - y = 0 \tag{5.4}$$

на интервале [0, 1] с граничным условием $y(x = 0) = y_0$. Точное решение этого уравнение известно: $y(x) = y_0 e^x$.

Решение

Сначала используем метод Галёркина и полиномиальные базисные функции ($\varphi_j = x^j$)

$$y_G = y_0 + \sum_{j=1}^{N} \alpha_j x^j . \tag{5.5}$$

Дифференциальный оператор $L = \frac{d}{dx} - 1$. Будем искать решение согласно формулировке уравнения (5.5)

$$Ly_G = \left(\frac{d}{dx} - 1\right)\left(y_0 + \sum_{j=1}^{N} \alpha_j x^j\right) = \sum_{j=1}^{N}\left(jx^{j-1} - x^j\right)\alpha_j - y_0 = 0. \quad (5.6)$$

Тестовые функции w_k выберем из того же семейства полиномиальных функций, что и базисные функции, но меньшего порядка. Тогда тестовые функции определим в виде $w_k = x^{k-1}$. Сформируем скалярное произведение $(Ly_G, w_k) = 0$. Поскольку $(u,v) = \iiint_D uvdxdydz$ и в нашем случае $(Ly_G, w_k) = \int_0^1 Ly_G w_k dx$, то

$$\left(\sum_{j=1}^{N}\left(jx^{j-1} - x^j\right)\alpha_j - y_0, w_k\right) = \int_0^1\left(\sum_{j=1}^{N}\left(jx^{j-1} - x^j\right)\alpha_j - y_0\right)x^{k-1}dx = 0,$$

следовательно,

$$\sum_{j=1}^{N}\alpha_j\int_0^1\left(jx^{j+k-2} - x^{j+k-1}\right)dx = \int_0^1 y_0 x^{k-1}dx, \;\; k = 1,2,...,N.$$

Полученное равенство запишем в матрично-векторной форме

$$\left[S_{kj}\right]\left[\alpha_j\right] = \left[\beta_k\right], \quad (5.7)$$

где

$$S_{kj} = \int_0^1\left(jx^{j+k-2} - x^{j+k-1}\right)dx = \frac{j}{j+k-1} - \frac{1}{j+k};$$
$$\beta_k = \int_0^1 y_0 x^{k-1}dx = \frac{y_0}{k}. \quad (5.8)$$

Решим уравнение (5.7) при $y_0 = 1$ и $N = 1, 2, 3, 4, 5$. Интервал определения x [0, 1] представим в дискретном виде: $x_d = 0, 0.25, ..., 1$. При $N = 1$ получим $S_{11} = 1/2$, $\beta_1 = 1$, следовательно, $\alpha_1 = 2$. Тогда выражение (5.5) примет вид

$$y_G^{N=1} = y_0 + \sum_{j=1}^{1} \alpha_1 x_d^1 = 1 + 2x_k.$$

Последовательно подставляя значения x_d в данное выражение, получим решение при $N = 1$

$$y_G^{N=1} = \begin{bmatrix} 1.00 \\ 1.50 \\ 2.00 \\ 2.50 \\ 3.00 \end{bmatrix}.$$

Выполняя аналогичные действия при $N = 2, 3, 4, 5$ (листинг 5.1), найдем значения y_G (таблица 5.2). Для сравнения в таблице 5.2 также приведено точное решение.

Далее рассмотрим решение уравнения (5.4) с помощью метода коллокаций (метод моментов с согласованием по точкам). В этом случае в качестве тестовых функций используются дельта-функции Дирака, т. е. $w_k = \delta(x - x_k)$. В виде базисных функций, как и ранее, используем полиномиальные функции $\varphi_j = x^j$. Разделим интервал [0, 1] на N дискретов, что эквивалентно

$$x_k = \frac{k}{N+1}, \quad k = 1, 2, 3, ..., N.$$

Подставив базисные и тестовые функции в скалярное произведение (Ly_C, w_k), где $y_C = y_0 + \sum_{j=1}^{N} \alpha_j x^j$, получим формулы для вычисления элементов матрицы и вектора сводных членов (правой части):

$$\begin{aligned} S_{kj} &= \int_0^1 \left(jx^{j-1} - x^j \right) \delta(x - x_k) dx = \left(\frac{k}{N+1} \right)^{j-1} \left(j - \frac{k}{N+1} \right); \\ \beta_k &= \int_0^1 y_0 \delta(x - x_k) dx = y_0. \end{aligned} \tag{5.9}$$

Таблица 5.2 – Решение уравнения (5.4) методом Галёркина

x	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	Точное решение
0.00	1.00	1.000000	1.000000	1.000000	1.000000	1.000000
0.25	1.50	1.267857	1.284331	1.284052	1.284024	1.284025
0.5	2.00	1.642857	1.647887	1.648716	1.648723	1.648721
0.75	2.50	2.125000	2.117077	2.116969	2.116999	2.117000
1.00	3.00	2.714286	2.718310	2.718282	2.718282	2.718282

```
clc; clear;
xh=0.25;
x=0:xh:1;
xn=1/xh+1;
y0=1; N=5;
for k=1:N
   for j=1:N
      s(k,j)=j/(j+k-1)-1/(j+k);
   end
   b(k)=y0/k;
end
a=s\b'
for n=1:xn
   solution(n)=y0;
   for j=1:N
      solution(n)=solution(n)+a(j)*x(n).^j;
   end
end
disp(solution)
```

Листинг 5.1 – Программный код для решения уравнения (5.4) методом Галёркина

Результаты использования метода коллокаций при $y_0 = 1$ и $N = 1, 2, 3, 4, 5$ сведены в таблицу 5.3 (листинг 5.2).

Таблица 5.3 – Решение уравнения (5.4) методом коллокаций

x	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	Точное решение
0.00	1.00	1.000000	1.000000	1.000000	1.000000	1.000000
0.25	1.50	1.255682	1.286957	1.283803	1.284039	1.284025
0.5	2.00	1.613636	1.652174	1.648444	1.648739	1.648721
0.75	2.50	2.073864	2.121739	2.116651	2.117023	2.117000
1.00	3.00	2.636363	2.721739	2.717642	2.718300	2.718282

```
clc; clear;
xh=0.25;
x=0:xh:1;
xn=1/xh+1;
y0=1; N=5;
for k=1:N
   for j=1:N
      s(k,j)=(k/(N+1))^(j-1)*(j-k/(N+1));
   end
   b(k)=y0;
end
a=s\b'
for n=1:xn
   solution(n)=y0;
   for j=1:N
      solution(n)=solution(n)+a(j)*x(n).^j;
   end
end
disp(solution)
```

Листинг 5.2 – Программный код для решения уравнения (5.4) методом коллокаций

Название «метод моментов», по мнению некоторых авторов, является неудачным, поскольку оно имеет немного другое значение в современной прикладной математике [39]. Однако при решении задач электромагнетизма этот подход, как и прежде, называют МоМ. Харрингтон при выборе названия для использованного им метода позаимствовал его из [40]. Ранее метод с таким названием был предложен в [41, 42], где в качестве тестовых функций для решения интегральных уравнений использовались полиномы (моменты функции). При этом необходимо отметить, что Харрингтон в работе [37] под термином «метод моментов», по сути, подразумевал МВН, хотя и ограничивался лишь линейными электромагнитными задачами. Далее в этой работе использовано историческое название метода применительно к решению электромагнитных задач – МоМ.

Решение методом моментов электродинамических задач предусматривает следующие этапы. Сначала металлические части рассчитываемой структуры заменяются эквивалентными поверхностными электрическими токами, после чего решается задача

возбуждения окружающей среды этими токами. При этом среда может содержать магнито-диэлектрические слои сложной формы. После того как задача возбуждения решена, на полученное решение соответственно металлическим элементам накладываются граничные условия для вычисления эквивалентных токов. Важным аспектом процесса решения является разбиение металлических поверхностей на элементарные площадки и аппроксимация тока в пределах каждой такой площадки. Для аппроксимации криволинейных границ поверхностей произвольной формы принято использовать разбиение на треугольники, а для представления тока в них – базисные функции РВГ (RWG – Rao, Wilton, Glisson). Рассмотрению вычислительных аспектов MoM посвящены, например, публикации [43, 44]. Он является наиболее часто используемым при моделировании задач ЭМС.

В качестве небольших и наглядных примеров, демонстрирующих простоту и элегантность метода моментов, рассмотрим электростатические задачи расчета распределения заряда. Напомним, что электрический потенциал φ в точке наблюдения $\mathbf{r}$, обусловленный объемной плотностью электрического заряда ρ, задается интегралом

$$\varphi(\mathbf{r}) = \int_V \frac{\rho(\mathbf{r}')}{4\pi\varepsilon\,|\,\mathbf{r}-\mathbf{r}'\,|}\,d\mathbf{r}'. \qquad (5.10)$$

Если в точке источника $\mathbf{r}'$ известна плотность электрического заряда $\rho(\mathbf{r}')$, то в любой точке наблюдения можно найти соответствующий потенциал, и наоборот, зная потенциал, можно определить соответствующую плотность заряда.

5.2 Примеры решения электростатических задач

5.2.1 Тонкая проволока

Рассмотрим тонкую проводящую проволоку (thin wire) длиной L и диаметром $2a$, ориентированную вдоль оси x (рисунок 5.2). Если радиус проволоки намного меньше ее длины, выражение (5.10) упрощается:

$$\varphi(\mathbf{r}) = \int_0^L \frac{\tau(x')}{4\pi\varepsilon\,|\,\mathbf{r}-\mathbf{r}'\,|}\,dx', \tag{5.11}$$

где

$$|\,\mathbf{r}-\mathbf{r}'\,| = \sqrt{(x-x')^2+(y-y')^2},$$

τ – линейная плотность заряда.

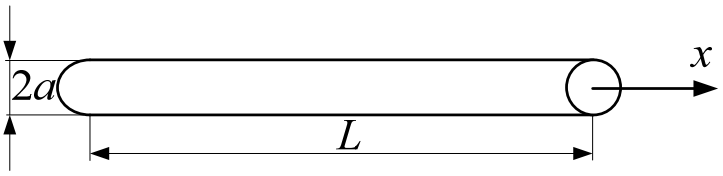


Рисунок 5.2 – Тонкая проволока

При такой постановке задача сводится к решению интегрального уравнения. Перейдем к ее приближенному решению. Для этого сегментируем проволоку на N подынтервалов длиной h каждый (рисунок 5.3). Нам потребуется $N+1$ узлов с координатами

$$x_n = (n-1)h \;\text{ при }\; h = L/N \;\text{ и }\; n = 1, 2, \ldots, N+1.$$

Тогда n-й подынтервал будет расположен между узлами x_n и x_{n+1}.

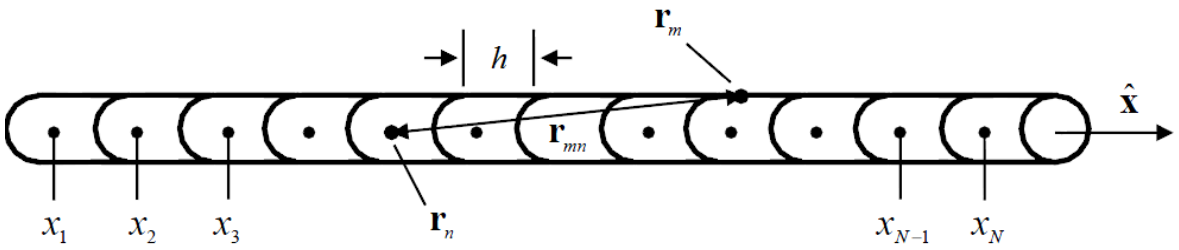


Рисунок 5.3 – Сегментация тонкой проволоки

С помощью кусочно-постоянных базисных функций в каждом подынтервале положим, что плотность заряда имеет постоянное значение, т. е. $\tau(x')$ кусочно-постоянна по длине проволоки. Математически это записывается как

$$\tau(x') = \sum_{n=1}^{N} \alpha_n f_n(x'), \tag{5.12}$$

где α_n – неизвестные коэффициенты (которые надо вычислить); $f_n(x')$ – импульсная функция, имеющая постоянное значение на подынтервале и равная нулю вне его, т. е.

$$f_n(x') = \begin{cases} 0, & x' < (n-1)h \\ 1, & (n-1)h \le x' \le nh \\ 0, & x' > nh \end{cases}. \qquad (5.13)$$

Подставляя выражения (5.13) в уравнение (5.11), получим

$$\varphi(\mathbf{r}) = \int_0^L \sum_{n=1}^{N} \alpha_n f_n(x') \frac{1}{4\pi\varepsilon\,|\,\mathbf{r}-\mathbf{r}'\,|}\,dx'. \qquad (5.14)$$

Используя определение импульсной функции (5.13), перепишем выражение (5.14) в виде

$$4\pi\varepsilon\varphi(\mathbf{r}) = \sum_{n=1}^{N} \alpha_n \int_{(n-1)h}^{nh} \frac{1}{|\,\mathbf{r}-\mathbf{r}'\,|}\,dx' \qquad (5.15)$$

или более наглядно

$$4\pi\varepsilon\varphi(\mathbf{r}) = \alpha_1 \int_0^h \frac{1}{|\,\mathbf{r}-\mathbf{r}'\,|}\,dx' + \alpha_2 \int_h^{2h} \frac{1}{|\,\mathbf{r}-\mathbf{r}'\,|}\,dx' + \ldots + \alpha_N \int_{(N-1)h}^{L} \frac{1}{|\,\mathbf{r}-\mathbf{r}'\,|}\,dx',$$

что дает одно уравнение с N неизвестными, а именно $\alpha_1, \ldots, \alpha_N$. Но чтобы получить единственное решение, требуется N уравнений или допущений (ограничений). Строго говоря, эти уравнения должны быть линейно независимы. Сопоставим граничное условие (потенциал на проводе) в N точках с координатами x_m вдоль провода. Этот процесс называется коллокацией или согласованием по точкам. Ему соответствует процесс аппроксимации с помощью так называемых тестовых или весовых функций. В данном случае это дельта-функции Дирака[1]. Для удобства, но без потери общности выберем в качестве координат N точек коллокации середины подынтервалов:

[1] Обобщенная функция, позволяющая записать точечное воздействие, а также пространственную плотность физических величин (масса, заряд и др.), сосредоточенных или приложенных в одной точке.

$$x_m = (m - 1/2)h, \;\; m = 1, 2, \ldots, N.$$

Тогда уравнение (5.15) для каждой из N точек даст систему из N уравнений. Таким образом, получим сумму интегралов, каждый из которых определен на соответствующем подынтервале. Для исключения сингулярности в подынтегральном выражении, возникающей из-за $|\mathbf{r} - \mathbf{r}'|$ точки наблюдения (поля) расположим на поверхности проволоки. Тогда знаменатель подынтегральной функции преобразуется к виду

$$|\mathbf{r} - \mathbf{r}'| = \sqrt{(x - x')^2 + a^2} \tag{5.16}$$

и уравнение (5.15) может быть переписано:

$$\begin{aligned}
4\pi\varepsilon\varphi(x_1) &= \alpha_1 \int_0^h \frac{1}{\sqrt{(x_1 - x')^2 + a^2}} dx' + \ldots + \alpha_N \int_{(N-1)h}^{Nh} \frac{1}{\sqrt{(x_1 - x')^2 + a^2}} dx', \\
&\vdots \\
4\pi\varepsilon\varphi(x_N) &= \alpha_1 \int_0^h \frac{1}{\sqrt{(x_N - x')^2 + a^2}} dx' + \ldots + \alpha_N \int_{(N-1)h}^{Nh} \frac{1}{\sqrt{(x_N - x')^2 + a^2}} dx',
\end{aligned} \tag{5.17}$$

или в матричной форме

$$\begin{pmatrix} z_{11} & z_{12} & \ldots & z_{1N} \\ z_{21} & z_{22} & \ldots & z_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ z_{N1} & z_{N2} & \ldots & z_{NN} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \ldots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_N \end{pmatrix}. \tag{5.18}$$

Элементы матрицы вычисляются как

$$z_{mn} = \int_{(n-1)h}^{nh} \frac{1}{\sqrt{(x_m - x')^2 + a^2}} dx', \tag{5.19}$$

а элементы правой части –

$$b_m = 4\pi\varepsilon\varphi(x_m). \tag{5.20}$$

Интеграл в выражении (5.19) можно представить в виде конечных комбинаций элементарных функций

$$z_{mn} = \log\left[\frac{(x_b - x_m) + \sqrt{(x_b - x_m)^2 - a^2}}{(x_a - x_m) + \sqrt{(x_a - x_m)^2 - a^2}}\right], \qquad (5.21)$$

где $x_b = nh$; $x_a = (n-1)h$. Линейная геометрия рассматриваемой задачи (при $M = N$) приводит к тому, что матрица **Z** является Теплицевой матрицей, т. е. имеет вид

$$\mathbf{Z} = \begin{pmatrix} z_1 & z_2 & z_3 & \dots & z_N \\ z_2 & z_1 & z_2 & \dots & z_{N-1} \\ z_3 & z_2 & z_1 & \dots & z_{N-2} \\ \dots & \dots & \dots & \dots & \dots \\ z_N & z_{N-1} & z_{N-2} & \dots & z_1 \end{pmatrix}. \qquad (5.22)$$

На рисунке 5.4 приведено искомое распределение заряда по поверхности проволоки (листинг 5.3) при $N = 10$ и 100 (для решения СЛАУ использовалось обращение матрицы). Видно, что увеличение числа базисных функций повышает точность аппроксимации. Как следует из рисунка, заряд накапливается на краях проволоки.

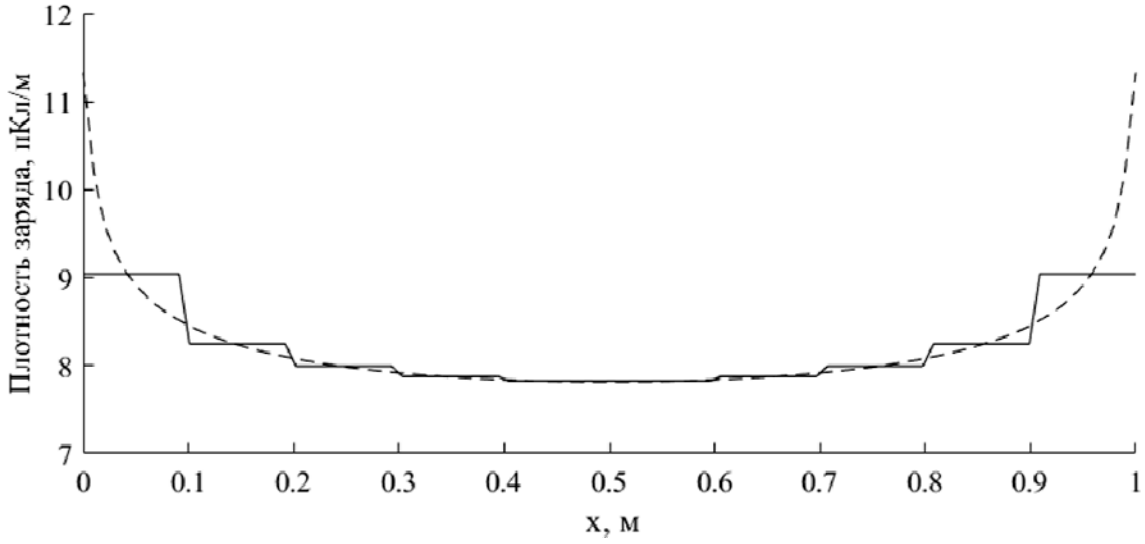


Рисунок 5.4 – Распределение заряда по поверхности тонкой проволоки при $N = 10$ (—) и 100 (– –)

```
clear; clc;
set(0,'DefaultAxesFontSize',14,'DefaultAxesFontName','Times New Roman');
N=100;
```

```
L=1;
a=0.00125;
Nx=100;
bf=10;
delta=L/bf;
x=delta/2:delta:L;
for n=1:bf
   xb=n*delta;
   xa=(n-1)*delta;
   for m=1:bf
       arg1=((xb-x(m)+sqrt((xb-x(m)).^2-a^2)))
       arg2=((xa-x(m)+sqrt((xa-x(m)).^2-a^2)))
       Z(m,n)=log(arg1/arg2);
   end
end
m=1:bf;
b(m)=4*pi*8.85e-12;
sol=Z\b';
m=1:Nx;
h=L/bf;
xh=0:h:L
x=0:L/(Nx-1):L;
yr=0;
for n=1:bf
   basic_func{n}=1.*((x(m)>=xh(n))&(x(m)<=xh(n+1)))+0.*(x(m)>xh(n));
   yr=yr+sol(n)*basic_func{n};
end
ylabel('Плотность заряда (пКл/м)','fontsize',14);
xlabel('Длина, м')
plot(x,yr*1e+12,'k');
```

Листинг 5.3 – Программный код для вычисления распределения заряда по поверхности тонкой проволоки

На основании полученных результатов сделаем несколько выводов.

– Замена поверхностной плотности заряда на линейную позволила существенно уменьшить вычислительную сложность задачи.

– В процессе коллокации граничные условия были точно соблюдены только в N дискретных точках. Между этими точками потенциал отклоняется от указанного (постоянного) значения.

Чтобы снизить влияние данной особенности, необходимо уменьшать длину подынтервалов.

– Выбранные кусочно-постоянные базисные функции оканчиваются на краях подынтервалов. Поскольку аппроксимируется непрерывный заряд, использование этих базисных функций в целом дает нефизичные результаты. Это видно из рисунка 5.4 при $N = 10$. Следует помнить, что полученное решение является приближенным. Для повышения его точности необходимо увеличивать число базисных функций (уменьшать длину подынтервалов).

5.2.2 Тонкая пластина

Рассмотрим бесконечно тонкую ($z = 0$) квадратную проводящую пластину размером $L \times L$ (рисунок 5.5). В этом случае электростатический потенциал связан с поверхностной плотностью заряда соотношением

$$\varphi(\mathbf{r}) = \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} \frac{\sigma(x', y')}{4\pi\varepsilon\,|\,\mathbf{r} - \mathbf{r}'\,|}\, dx' dy'. \qquad (5.23)$$

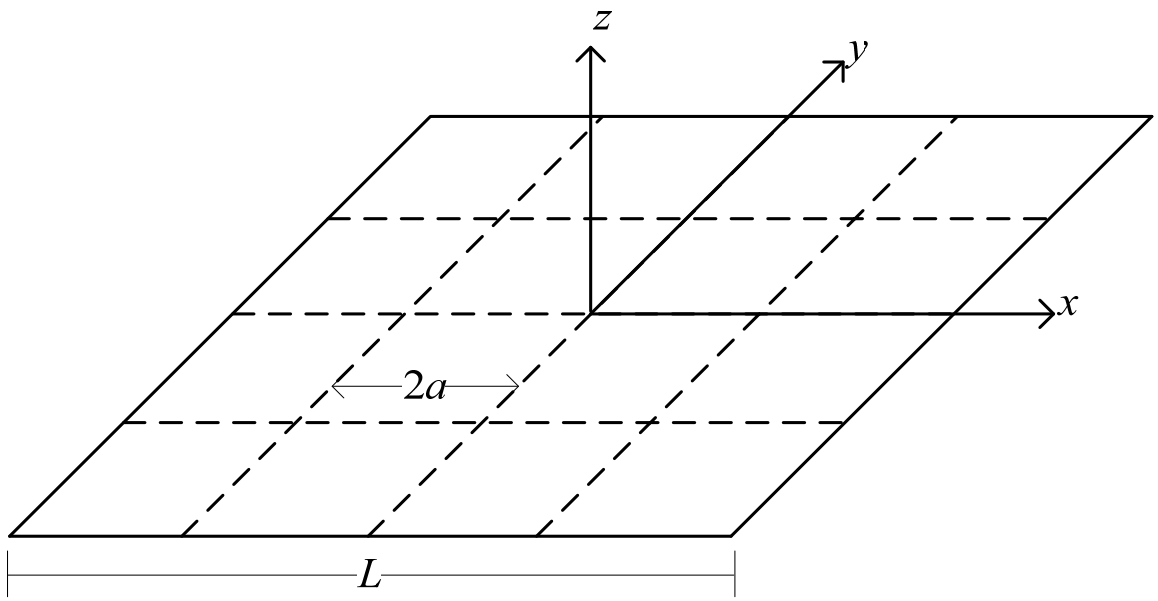


Рисунок 5.5 – Тонкая квадратная пластина

Зафиксируем на пластине потенциал 1 В. Тогда уравнение (5.23) преобразуется к виду

$$1 = \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} \frac{\sigma(x', y')}{4\pi\varepsilon\sqrt{(x-x')^2 + (y-y')^2}} dx'dy'. \qquad (5.24)$$

Разобьем пластину на N квадратных площадок со сторонами $2a$ (площадь $\Delta = 4a^2$) и положим, что на каждой из них заряд имеет постоянное значение. Далее выберем N независимых точек наблюдения, расположенных в центрах (x_m, y_m) площадок. Элементы матрицы СЛАУ при этом вычисляются как

$$z_{mn} = \int_{S_n} \frac{1}{4\pi\varepsilon\sqrt{(x_m - x')^2 + (y_m - y')^2}} dx'dy', \qquad (5.25)$$

где S_n – площадь n-й площадки, а элементы правой части $b_m = 1$.

Когда точки источника и наблюдения поля совпадают ($m = n$), подынтегральная функция становится сингулярной, поэтому интеграл должен вычисляться аналитически. С учетом постоянства заряда на поверхности каждой площадки диагональные элементы матрицы вычисляются как

$$z_{mm} = \int_{-a}^{a} \int_{-a}^{a} \frac{1}{4\pi\varepsilon\sqrt{(x')^2 + (y')^2}} dx'dy'. \qquad (5.26)$$

Раскрыв внутренний интеграл, получим

$$z_{mm} = \frac{1}{4\pi\varepsilon} \int_{-a}^{a} \log\left[\frac{\sqrt{a^2 + (y')^2} + a}{\sqrt{a^2 + (y')^2} - a} dy' \right]. \qquad (5.27)$$

Раскрыв внешний интеграл, получим

$$z_{mm} = \frac{1}{4\pi\varepsilon}\left(2a \log\left[y + \sqrt{a^2 + (y')^2} \right] + \right.$$

$$\left. + y \log\left[\frac{y^2 + 2a(a + \sqrt{a^2 + (y')^2})}{y^2} \right]\Bigg|_{-a}^{a} \right),$$

после упрощений

$$z_{mm} = \frac{2a}{\pi\varepsilon} \log\left(1 + \sqrt{2}\right). \qquad (5.28)$$

Элементы, расположенные вне главной диагонали ($m \neq n$) матрицы, могут быть вычислены с помощью простой аппроксимации по формуле

$$z_{mn} \approx \frac{\Delta}{4\pi\varepsilon \mathbf{r}_{mn}} = \frac{a^2}{\pi\varepsilon\sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}}, \qquad (5.29)$$

где x_n и y_n – координаты центра n-й площадки источника. Эта аппроксимация обладает весьма существенной погрешностью при вычислении элементов матрицы, соответствующих близко расположенным площадкам. Для повышения точности требуется или получение аналитических выражений, или использование методов численного интегрирования. В результате получается СЛАУ вида (5.18).

На рисунке 5.6 показана вычисленная поверхностная плотность заряда (листинг 5.4) на пластине при $N = 225$ (по 15 площадок по осям x и y) и $N = 1225$ (по 35 площадок). Для решения СЛАУ использовалось обращение матрицы. Видно, что, как и в случае с проволокой, заряд накапливается на краях пластины. Поэтому для повышения точности необходимо применять более частую сегментацию.

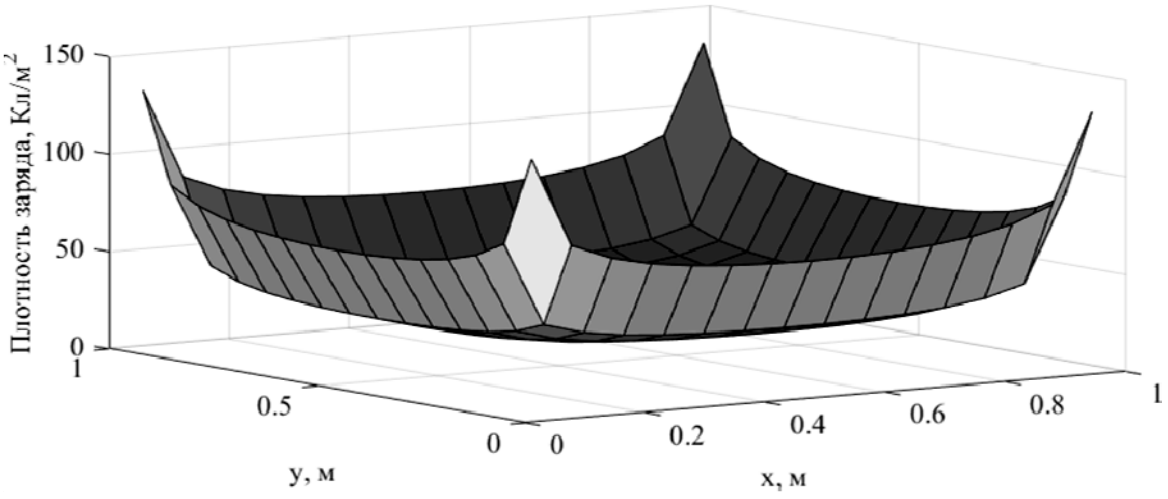
При известной поверхностной плотности заряда на пластине емкость вычисляется по формуле

$$C = \frac{Q}{V} = \frac{1}{V}\int_{-L/2}^{L/2}\int_{-L/2}^{L/2} \sigma(x', y')\, dx' dy' \approx \frac{1}{V}\sum_{n=1}^{N} \alpha_n \Delta. \qquad (5.30)$$

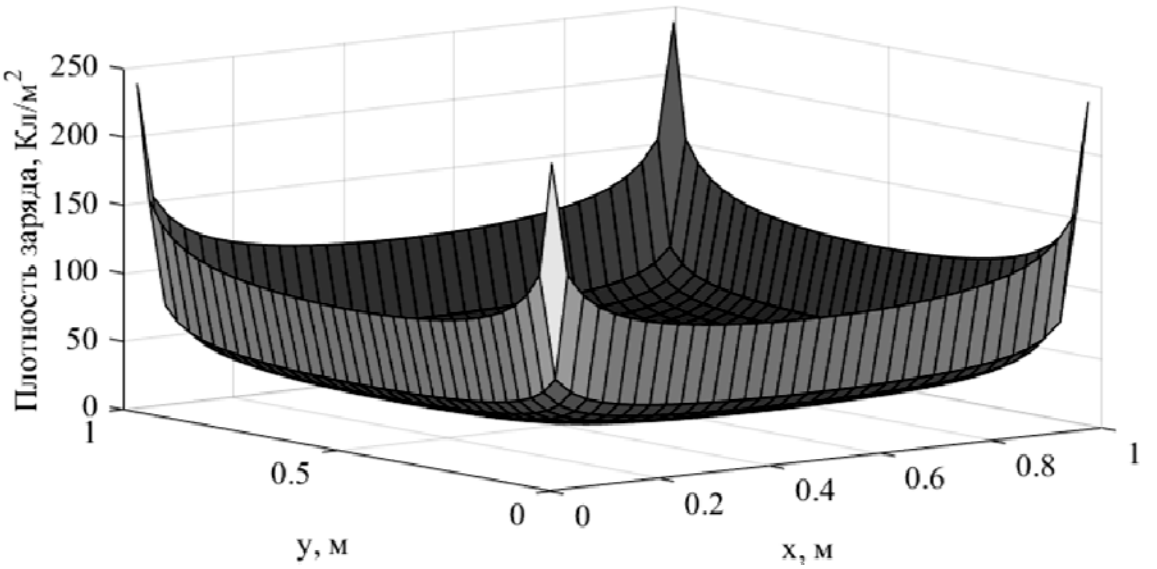
Как следует из таблицы 5.4, при увеличении N значения емкости сходятся к определенной величине.

Таблица 5.4 – Емкость квадратной пластины при изменении N

N	4	9	16	25	36	100	225	625	1225
C, пФ	35.70	37.37	38.24	38.77	39.12	39.83	40.18	40.45	40.57



а



б

Рисунок 5.6 – Распределение заряда на квадратной пластине при $N = 225$ (*а*) и 1225 (*б*)

```
clear;clc;
L=1;V=1;
eps0=8.854e-12;
patch  x=15;
patch_y=patch_x
N=patch_x*patch_y;
la=L/patch_x;
a=la/2;
xc1=la/2:la:L;
yc1=xc1;
xc2=-L/2+la/2:la:L/2;
yc2=xc2;
[X,Y]=meshgrid(xc1,yc1);
c=1;
```

```
for i=0:patch_x-1
   for j=0:patch_x-1
      xc(c)=la/2+j*la;
      yc(c)=la/2+i*la;
      c=c+1;
   end
end
for m=1:N;
    for n=1:N
       if m==n
         Z(m,m)=2*a*log(1+sqrt(2))/(pi*eps0);
       else
         Z(m,n)=(a*a)/(pi*eps0)/sqrt((xc(m)-xc(n))^2+(yc(m)-yc(n))^2);
       end
    end
       b(m)=V;
end
sol=inv(Z)*b';
j=1;
   for i=1:patch_x
      SOL(1:patch_x,i)=sol(j:i*patch_x,1);
      j=j+patch_x;
   end
SOL=SOL/1e-12;
surf(X,fliplr(Y),SOL)
zlabel('Плотность заряда, Кл/м^2','fontsize',16);
xlabel('Длина, м','fontsize',16);
ylabel('Длина, м','fontsize',16);
Q=sum(sol);
Q=Q*4*a*a
```

Листинг 5.4 – Программный код для вычисления распределения заряда по поверхности тонкой пластины и ее емкости

5.2.3 Плоский конденсатор

Рассмотрим плоский конденсатор, состоящий из двух квадратных пластин, расположенных одна над другой на расстоянии H (рисунок 5.7). Пусть разница потенциалов между пластинами равна 1 В (на верхней пластине +0.5 В, а на нижней – минус 0.5 В). Найдем распределение заряда на конденсаторе и его емкость. Для этого разобьем верхнюю и нижнюю пластины на N площадок.

Таким образом, число неизвестных составит $2N$, а матрица будет иметь блочную структуру:

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}^{\text{ВВ}} & \mathbf{Z}^{\text{ВН}} \\ \mathbf{Z}^{\text{НВ}} & \mathbf{Z}^{\text{НН}} \end{pmatrix},$$

где индексы «в» и «н» обозначают верхнюю и нижнюю пластины соответственно. Диагональные блоки $N \times N$ ($\mathbf{Z}^{\text{ВВ}}$ и $\mathbf{Z}^{\text{НН}}$) соответствуют матрицам, вычисленным для одной пластины. Из-за симметрии рассматриваемой геометрии элементы этих блоков равны между собой, т. е. $\mathbf{Z}^{\text{ВВ}} = \mathbf{Z}^{\text{НН}}$, и вычисляются по формулам (5.28) и (5.29). Внедиагональные блоки ($\mathbf{Z}^{\text{ВН}}$ и $\mathbf{Z}^{\text{НВ}}$) также равны между собой и соответствуют площадкам пластина-пластина.

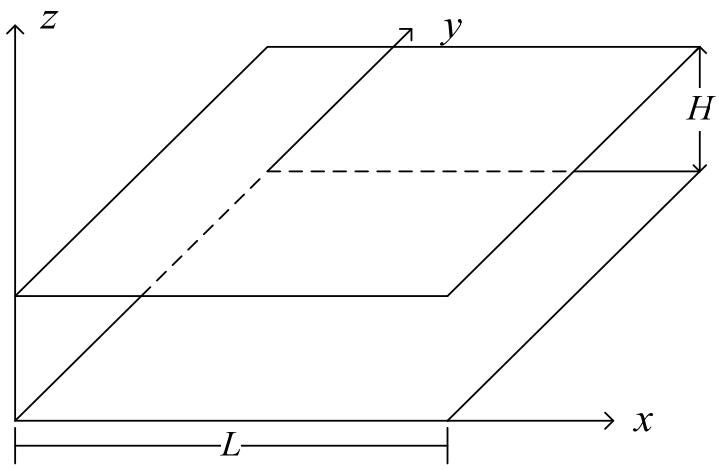


Рисунок 5.7 – Общий вид плоского конденсатора

Диагональные элементы блоков $\mathbf{Z}^{\text{ВН}}$ и $\mathbf{Z}^{\text{НВ}}$ с учетом изменений по координате z вычисляются аналогично внедиагональным элементам $\mathbf{Z}^{\text{ВВ}}$ и $\mathbf{Z}^{\text{НН}}$:

$$z_{mm}^{\text{ВН}} = \frac{a^2}{\pi\varepsilon\sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + H^2}}. \qquad (5.31)$$

Внедиагональные элементы вычисляются как

$$z_{mn}^{\text{ВН}} = \frac{0{,}564a}{\varepsilon}\left[\sqrt{1 + \frac{\pi}{4}\left(\frac{H}{a}\right)^2} - \frac{H}{2a}\sqrt{\pi}\right]. \qquad (5.32)$$

Правая часть СЛАУ также имеет блочную структуру:

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}^{\text{В}} \\ \mathbf{b}^{\text{Н}} \end{bmatrix}.$$

При этом $b_m^{\text{В}} = 0.5$, $b_m^{\text{Н}} = -0.5$ и $b_m^{\text{Н}} = -b_m^{\text{В}}$. Таким образом, результирующая СЛАУ имеет блочный вид:

$$\begin{pmatrix} \mathbf{Z}^{\text{ВВ}} & \mathbf{Z}^{\text{ВН}} \\ \mathbf{Z}^{\text{ВН}} & \mathbf{Z}^{\text{ВВ}} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{\text{В}} \\ \boldsymbol{\alpha}^{\text{Н}} \end{pmatrix} = \begin{pmatrix} \mathbf{b}^{\text{В}} \\ -\mathbf{b}^{\text{В}} \end{pmatrix}. \tag{5.33}$$

Задача может быть еще упрощена за счет использования симметрии структуры. Поскольку $\boldsymbol{\alpha}$ соответствует плотности заряда на обеих пластинах, а их размеры одинаковы и на них одинаковый и противоположный по знаку потенциал, то $\boldsymbol{\alpha}^{\text{В}} = -\boldsymbol{\alpha}^{\text{Н}}$. Тогда уравнение (5.33) можно упростить:

$$\left[\mathbf{Z}^{\text{ВВ}} - \mathbf{Z}^{\text{ВН}}\right]\boldsymbol{\alpha}^{\text{В}} = \mathbf{b}^{\text{В}}. \tag{5.34}$$

Решив данную систему уравнениий, получим плотность заряда. Далее найдем емкость конденсатора. Для этого воспользуемся формулой

$$C = \frac{Q_{\text{В}}}{V} = \frac{1}{V} \sum_{\substack{\text{верхняя} \\ \text{пластина}}} \alpha_n^{\text{В}} \Delta = 4a^2 \sum_{\substack{\text{верхняя} \\ \text{пластина}}} \alpha_n^{\text{В}}. \tag{5.35}$$

В таблице 5.5 приведены результаты вычисления емкости конденсатора (листинг 5.5) при изменении *N*. Видно, что значения емкости сходятся к определенной величине. На рисунке 5.8 показано распределение поверхностной плотности заряда при $N = 512$.

Таблица 5.5 – Емкость плоского конденсатора при учащении сегментации

N	8	32	128	512	2048	8192	32768
C, пФ	25.12	27.27	28.45	29.06	29.35	29.47	29.55

```
clear; clc;
set(0,'DefaultAxesFontSize',14,'DefaultAxesFontName','Times New Roman');
L=1;H=1;V=1;
```

```
eps=8.854e-12;
patch_x=32;
patch_y=patch_x;
N=patch_x*patch_y;
N=N*2; la=L/patch_x;
a=la/2; xc1=la/2:la:L;
yc1=xc1;zc1(size(yc1))=0;
zc2=zc1;zc2(:)=H;
xc_surf=0:L/(patch_x-1):L;
yc_surf=xc_surf;
[X,Y]=meshgrid(xc_surf,yc_surf);
c=1;
for i=0:patch_x-1
   for j=0:patch_x-1
      xc(c)=la/2+j*la;
      yc(c)=la/2+i*la;
      zc(c)=0;
      c=c+1;
   end
end
z=zeros(size(xc));
z(:)=H;
zc=[zc,z];
xc=[xc,xc];
yc=[yc,yc];
for m=1:N/2;
   for n=1:N/2
      if m==n
         Z(m,m)=la*log(1+sqrt(2))/(pi*eps);
      else
         Z(m,n)=(la*la)/(4*pi*eps)/sqrt((xc(m)-xc(n))^2+(yc(m)-yc(n))^2);
      end
   end
   b(m)=0.5;
end
for m=N/2+1:N;
   for n=1:N/2
      if m==n
         Z(m,m)=0.564*la/eps*(sqrt(1+pi*(H/la)^2)/4-H*sqrt(pi)/(2*la));
      else
         Z(m,n)=(la*la)/(4*pi*eps)/sqrt((xc(m)-xc(n))^2+(yc(m)-
yc(n))^2+H^2);
      end
   end
```

```
end
sol=inv(Z(1:N/2,1:N/2)-Z(N/2+1:N,1:N/2))*b';
sol=[sol;-sol];
j=1;
for i=1:patch_x
   SOL(1:patch_x,i)=sol(j:i*patch_x,1);
   j=j+patch_x;
end
j=1;
for i=1:patch_x

SOL2(1:patch_x,i)=sol(patch_x*patch_x+j:patch_x*patch_x+i*patch_x,1);
   j=j+patch_x;
end
SOL=SOL/1e-12;
SOL2=SOL2/1e-12;
surf(X,Y,SOL)
hold on;
surf(X,Y,SOL2)
hold off;
zlabel('Плотность заряда, пКл/м^2','fontsize',14);
xlabel('Длина, м','fontsize',14);
ylabel('Длина, м','fontsize',14);
Q = sum(abs(sol(1:N/2))*la*la);
disp(Q/V)
```

Листинг 5.5 – Программный код для вычисления распределения заряда по поверхности плоского конденстора и его емкости

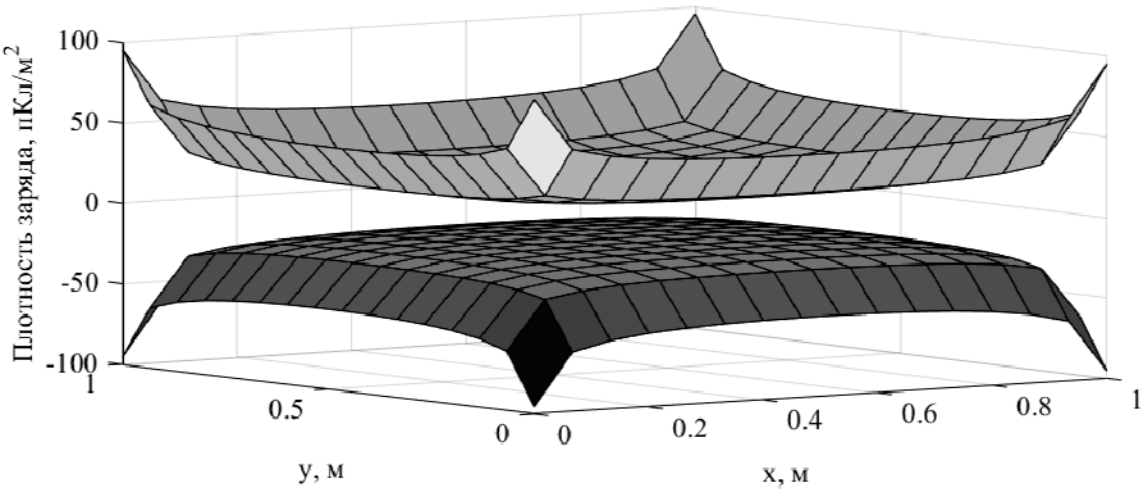


Рисунок 5.8 – Распределение заряда на плоском конденсаторе при $N = 512$

5.3 Базисные и тестовые функции

Эффективность МоМ во многом зависит от выбора ячеек и базисных и тестовых (пробных, весовых) функций, что определяется формой всей проводящей поверхности. Если поверхность (или совокупность поверхностей) имеет форму многоугольника с прямыми углами, целесообразно использовать ячейки прямоугольной формы с размерами $h_x \times h_y$. Поскольку базисные и тестовые функции по сути идентичны, то дальше речь пойдет только о первых. При этом системы базисных функций делятся на два вида: функции подобластей и функции полной области. Ниже на одномерном примере рассмотрим наиболее часто используемые базисные и тестовые функции.

Простейшие базисные функции имеют постоянное значение в ячейке и равны нулю вне ее. Они называются кусочно-постоянными (КПБФ) или импульсными (pulse functions) (рисунок 5.9). При этом интересующий интервал делится на N подынтервалов. На рисунке 5.9 подынтервалы имеют одинаковую длину, но это условие не является обязательным. На рисунке 5.9 приведен пример использования КПБФ, задаваемых как

$$f_m(x) = \begin{cases} 1, & x_m \le x \le x_{m+1},\ m = 1,\ldots,N; \\ 0 & \text{иначе.} \end{cases}$$

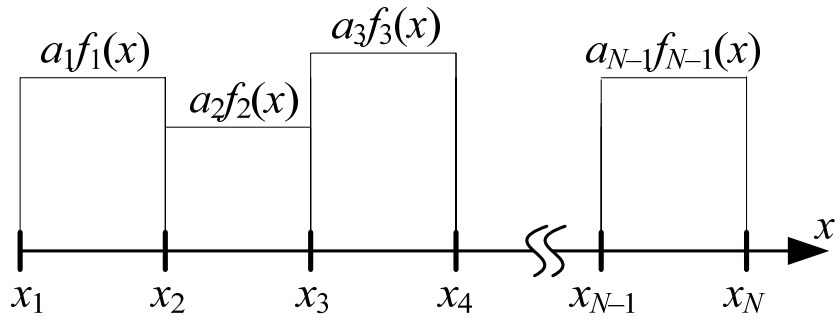


Рисунок 5.9 – Кусочно-постоянные базисные функции

На рисунке 5.10 показан результат использования КПБФ для аппроксимации функции

$$f(x) = \frac{1}{\sqrt{\left(\frac{\pi}{2}\right)^2 - x^2}} \qquad (5.36)$$

на интервале [0, 1.5] (листинг 5.6).

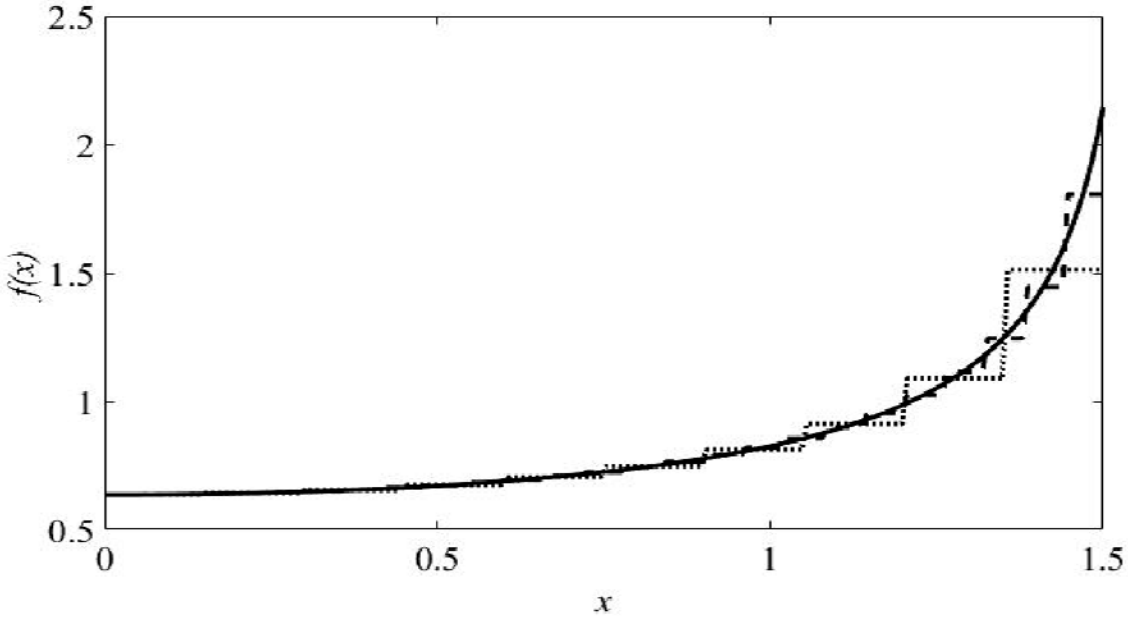


Рисунок 5.10 – Аппроксимация функции (5.36) (—)
при использовании N = 10 (····) и 25 (- -) КПБФ

```
clc; clear;
xstart=0;
xend=1.5;
N=10;
h=(xend-xstart)/N;
Nx=200;
hx=(xend-xstart)/(Nx-1);
x=0:hx:xend;
Y0 = 1./sqrt((pi/2).^2-x.^2);
xh=xstart:h:xend;
xc=h/2:h:xend;
yr=0;
m=1:Nx;
for n=1:N
   bf{n}=1.*((x(m)>=xh(n))&(x(m)<=xh(n+1)))+0.*(x(m)>xh(n));
   koef=1./sqrt((pi/2).^2-xc(n).^2);
   yr=yr+koef*bf{n};
end
```

```
hL=plot(x,yr,'g',x,Y0)
xlabel('x')
ylabel('f(x)')
```

Листинг 5.6 – Программный код для кусочно-постоянной аппроксимации функции (5.36)

Недостатком применения КПБФ является то, что получаемая аппроксимирующая функция разрывна. В электродинамике это в ряде случаев неприемлемо, так как разрывные функции могут порождать сингулярные поля, не отвечающие физической реальности. Тем не менее рассматриваемая система базисных функций нашла достаточно широкое применение.

Более сложные базисные функции предполагают изменение искомой величины вдоль одной или обеих координат в пределах ячейки. Так, если ток течет в направлении оси x и его можно считать постоянным вдоль оси y, то можно использовать так называемые «крышечные», или кусочно-линейные базисные функции (КЛБФ). Семейство таких функций показано на рисунке 5.11.

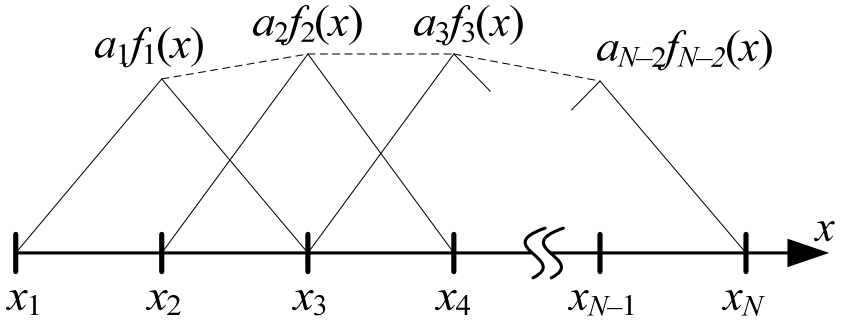


Рисунок 5.11 – Семейство кусочно-линейных базисных функций

Интервал делится на N точек и N–1 подынтервалов, что требует использования N–2 БФ. Длина КЛБФ одинакова, но, как и в случае с импульсными функциями, это не обязательное условие. Поскольку смежные функции перекрывают один сегмент, то использование треугольников позволяет кусочно линеаризовать решение между сегментами. КЛБФ определяются как

$$f_m(x) = \begin{cases} 1 - \dfrac{x_m - x}{h_x}, & x_m - h_x \le x < x_m; \\ 1 + \dfrac{x_m - x}{h_x}, & x_m \le x < x_m + h_x; \\ 0 & \text{иначе.} \end{cases}$$

Пример использования КЛБФ для аппроксимации функции (5.36) приведен на рисунке 5.12 (листинг 5.7).

Из рисунков 5.11 и 5.12 видно, что решение на концах интервала (x_1 и x_N) равно нулю. Таким образом, данная реализация желательна, когда априори известно, что значение решения на концах интервала равно нулю, но не желательна, если решение может быть ненулевым. Если вместо этого добавить первый и последний сегменты полутреугольника, решение больше не будет принудительно сбрасываться в ноль (рисунок 5.13). В этом случае используется уже не N–2, а N КЛБФ.

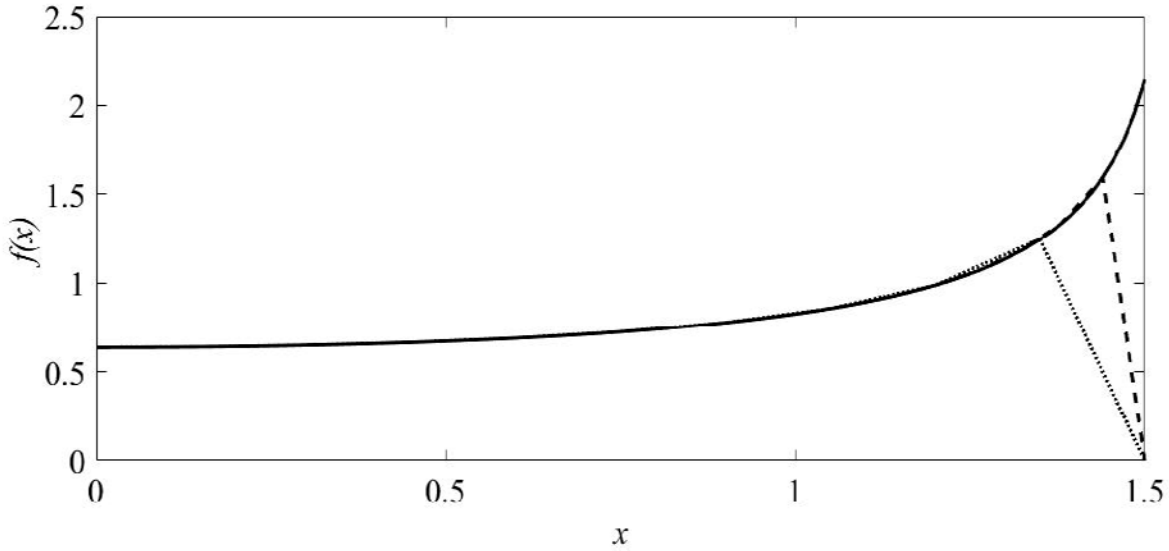


Рисунок 5.12 – Аппроксимация функции (5.36) (—)
при использовании N = 10 (····) и 25 (- -) КЛБФ

```
clc;clear;
xstart=0;
xend=1.5;
N=25;
h=(xend-xstart)/N;
```

```
Nx=200;
hx=(xend-xstart)/(Nx-1);
x=0:hx:xend
Y0 = 1./sqrt((pi/2).^2-x.^2);
xh=xstart:h:xend
xc=0:h:xend;
yr=0;
m=1:Nx;
for n=1:N
   bf{n}=(1 - (xh(n) - x(m)) / h).*(xh(n) - h <= x(m) & x(m) < xh(n))+...
   (1 + (xh(n) - x(m)) / h).*((xh(n) <= x(m) & x(m) < xh(n+1)))+...
      0.*((xh(n)>=x(m)+h/2)&(xh(n)<x(m)-h));
   koef=1./sqrt((pi/2).^2-xc(n).^2);
   yr=yr+koef*bf{n};
end
hL=plot(x,yr,'g',x,Y0)
set(hL, 'LineWidth',2)
xlabel('x')
ylabel('f(x)')
```

Листинг 5.7 – Программный код для кусочно-линейной аппроксимации функции (5.36)

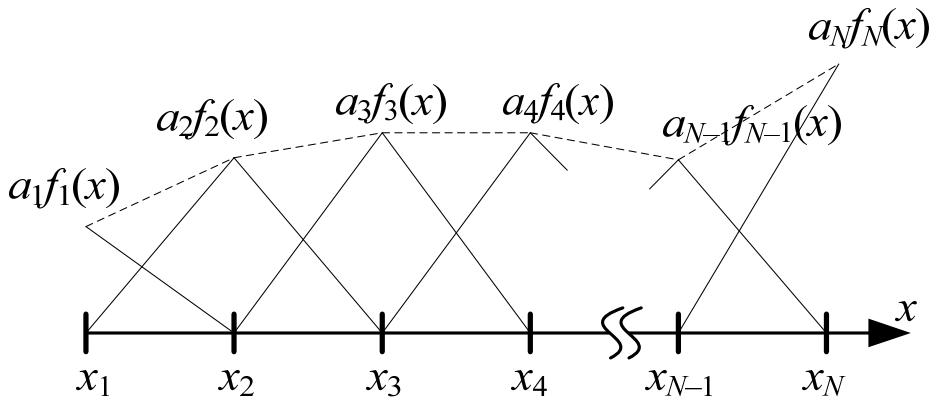


Рисунок 5.13 – Семейство кусочно-линейных базисных функций с дополнительными функциями на концах интервала

Для моделирования быстро изменяющихся в пространстве токов удобнее использовать кусочно-синусоидальные базисные функции (КСБФ), которые подобны КЛБФ (рисунок 5.14). Они часто используются при анализе проводных антенн из-за их способности представлять синусоидальные распределения токов. Такие функции определяются как

$$f_m(x) = \begin{cases} \dfrac{\sin\left[k\left(x-(x_m-h_x)\right)\right]}{\sin(kh_x)}, & x_m - h_x \le x < x_m; \\ \dfrac{\sin\left[k\left((x_m+h_x)-x\right)\right]}{\sin(kh_x)}, & x_m \le x < x_m + h_x; \\ 0 \qquad\qquad \text{иначе}, \end{cases}$$

где $k = 2\pi/\lambda$ – волновое число. При этом требуется, чтобы длина сегмента была значительно меньше периода используемой синусоиды (не менее 10).

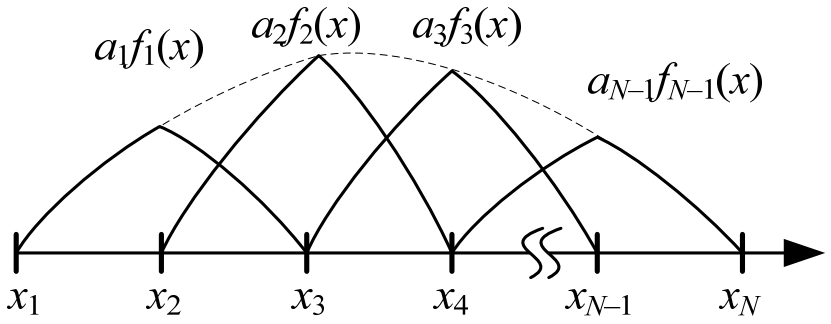


Рисунок 5.14 – Семейство кусочно-синусоидальных базисных функций

Применение КПБФ для аппроксимации функции (5.36) показано на рисунке 5.15.

Представленные БФ подобластей, как следует из названия и приведенных примеров, отличны от нуля только в части рассматриваемого интервала/области. БФ полной области определены на всем интервале. Наиболее известными являются следующие БФ:

Фурье

$$f_m = \cos(2m-1)\pi z/2;$$

Чебышёва

$$f_m = T_{2m-2}(z), \text{ где } T_0(z) = 1,\ T_1(z) = z,\ \ldots,\ T_{m+1}(z) = 2xT_m(z) - T_{m-1}(z);$$

Маклорена

$$f_m = z^{2m-2};$$

Лежандра

$$f_m = P_{2m-2}(z), \text{ где } P_m(z) = \frac{1}{2^m m!}\frac{d^m}{dz^m}(z^2-1)^m,$$

$m = 1, 2, \ldots, N$; $z = 2x/L$; L – длина интервала.

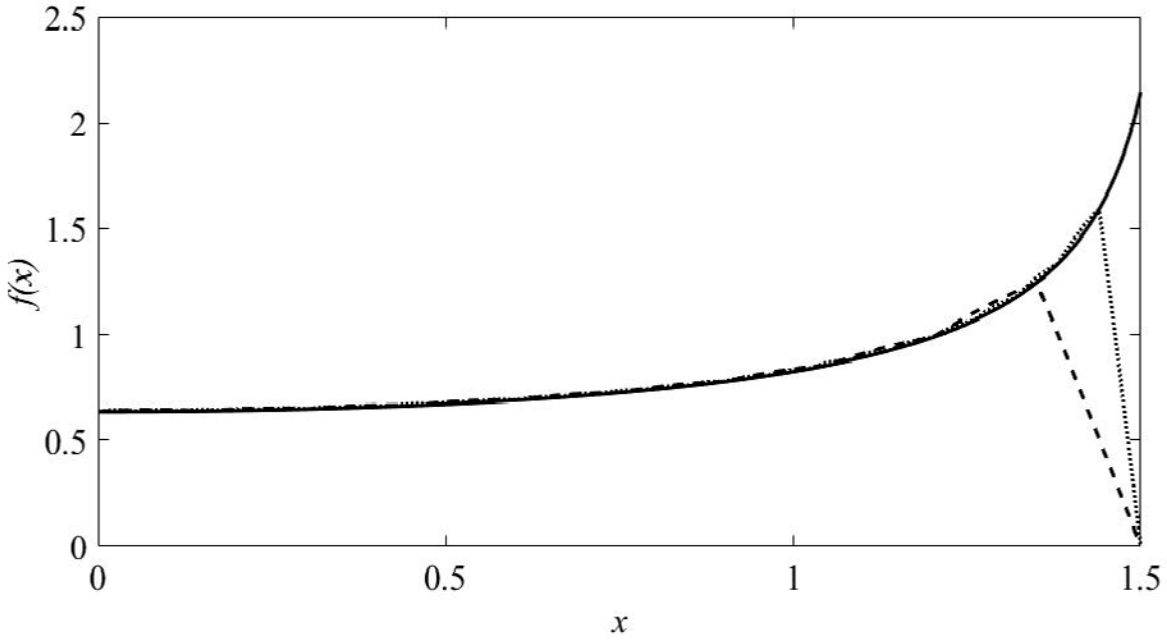


Рисунок 5.15 – Аппроксимация функции (5.36) (—)
при использовании $N = 10$ (····) и 25 (- -) КСБФ при $k = 2\pi/20h$

Все БФ полной области имеют ограниченное применение, поскольку они требуют априорного знания природы аппроксимируемой функции. Поэтому БФ подобластей чаще используются на практике, особенно при разработке универсальных программных кодов.

5.4 Математическая модель вычисления емкостной матрицы многопроводной линии передачи

Системным средством сведения дифференциальных уравнений к интегральным является построение вспомогательной функции – функции Грина[1]. Функция Грина (функция источника или функция влияния) – это функция ядра, полученная из линейной

[1] Джордж Грин (1793–1841) – английский математик, внесший значительный вклад в разные разделы математической физики. Самая известная работа: «Опыт приложения математического анализа к теориям электричества и магнетизма», 1828 г.

краевой задачи, которая образует важную связь между дифференциальными и интегральными формулировками. Чтобы получить поле, вызванное распределенным источником, по методу функции Грина, необходимо определить вклад каждой элементарной части источника и просуммировать все вклады (принцип суперпозиции[1]).

Если $G(\mathbf{r}, \mathbf{r}')$ – поле в точке наблюдения $\mathbf{r}$ (observation point), вызванное единичным точечным источником, расположенным в точке источника $\mathbf{r}'$ (source point или integration point), то это поле относительно распределенного источника $g(\mathbf{r}')$ является интегралом от $g(\mathbf{r})G(\mathbf{r}, \mathbf{r}')$. Функция G и есть функция Грина. Таким образом, функция Грина $G(\mathbf{r}, \mathbf{r}')$ представляет собой потенциал в точке $\mathbf{r}$, возникающий из-за единичного точечного заряда в точке $\mathbf{r}'$. Например, рассмотрим линейное дифференциальное уравнение второго порядка в частных производных

$$L\Psi = g.$$

Найдем функцию Грина, соответствующую дифференциальному оператору L, как решение неоднородного уравнения точечного источника, т. е.

$$LG(\mathbf{r}, \mathbf{r}') = \delta(\mathbf{r}, \mathbf{r}'), \tag{5.37}$$

где $\mathbf{r}$ и $\mathbf{r}'$ – позиции точек наблюдения (x, y, z) и источника (x', y', z') соответственно (рисунок 5.16); $\delta(\mathbf{r}, \mathbf{r}')$ – дельта-функция Дирака, определенная при $\mathbf{r} \neq \mathbf{r}'$ и удовлетворяющая условию

$$\int \delta(\mathbf{r}, \mathbf{r}')g(\mathbf{r}')dv' = g(\mathbf{r}). \tag{5.38}$$

Из уравнения (5.37) видно, что функцию Грина G можно интерпретировать как решение краевой задачи с известной функцией g, замененной на дельта-функцию (единичную импульсную функцию). То есть функцию Грина можно интерпретировать как отклик линейной системы на единичное импульсное воздействие в точке $\mathbf{r} = \mathbf{r}'$.

[1] Принцип суперпозиции в электростатике утверждает, что напряженность электростатического поля, создаваемого в данной точке системой зарядов, есть векторная сумма напряженностей полей отдельных зарядов (следствие того, что уравнения Максвелла в вакууме линейны).

Функция Грина обладает следующими свойствами:

– удовлетворяет уравнению $LG = 0$ за исключением точки источника согласно уравнению (5.37);

– удовлетворяет заданному граничному условию f на границе B, т.е. $G = f$ на границе B;

– симметрична, т. е.

$$G(\mathbf{r}, \mathbf{r}') = G(\mathbf{r}', \mathbf{r}). \tag{5.39}$$

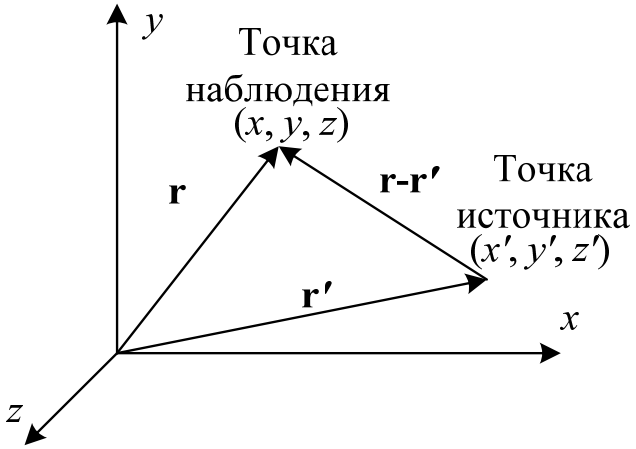


Рисунок 5.16 – Точки источника и наблюдения

В [45] представлен подробный вывод моделей вычисления емкостной матрицы с использованием МоМ в сочетании с аналитическими выражениями для элементов матрицы СЛАУ. Эти модели предназначены для двухмерных и трехмерных структур с границами произвольной сложности, включающих идеально проводящую плоскость и без нее. Для ясности дальнейшего изложения кратко поясним суть данного подхода на примере двухмерной задачи (для трехмерной задачи подход аналогичен). В качестве операторного уравнения выступает уравнение Пуассона в интегральной форме

$$\Phi = \mathrm{L}^{-1}\rho,\ \mathrm{L}^{-1} = \frac{1}{\varepsilon_0}\int G(\mathbf{r}, \mathbf{r}')d\Gamma, \tag{5.40}$$

где $G(\mathbf{r}, \mathbf{r}')$ – функция Грина; $\mathbf{r}$ – точка наблюдения (x, y); $\mathbf{r}'$ – точка источника (x', y'); dΓ – дифференциал по поверхности структуры.

При такой постановке задачи считаются заданными граничные условия по приложенному напряжению Ф, требуется найти плотность заряда ρ. Отметим, что для двухмерного случая функция Грина имеет вид

$$G(\mathbf{r},\mathbf{r}') = \frac{\ln|\mathbf{r}-\mathbf{r}'|}{2\pi}, \tag{5.41}$$

а

$$\nabla G(\mathbf{r},\mathbf{r}') = \frac{\mathbf{r}-\mathbf{r}'}{2\pi|\mathbf{r}-\mathbf{r}'|^2}. \tag{5.42}$$

Рассмотрим математическую модель вычисления емкостной матрицы на примере связанной МПЛ, поперечное сечение которой приведено на рисунке 5.17. Структура содержит два проводника (I и II), расположенных на диэлектрическом основании с относительной диэлектрической проницаемостью ε_{r2} над идеально проводящей (бесконечной) плоскостью.

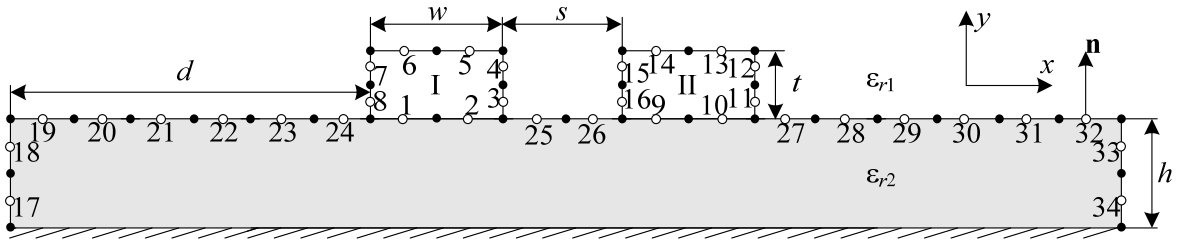


Рисунок 5.17 – Поперечное сечение связанной МПЛ

Для вычисления необходимо выполнить следующие действия [17].

Шаг 1. Дискретизировать границы проводников и диэлектрической подложки (граница раздела двух сред) на небольшие отрезки (подынтервалы) длиной l_n, $n = 1, 2, \ldots, N$. На рисунке 5.17 деление на подынтервалы показано с помощью закрашенных кругов, а центры подынтервалов отмечены с помощью незакрашенных кругов. Причем дискретизируются сначала проводниковые, а затем диэлектрические границы. В данном случае границы дискретизированы на $N = 34$ подынтервала, границам проводник-диэлектрик соответствует $N_C = 16$, а диэлектрик-диэлектрик – $N_D = 18$ ($N = N_C + N_D$).

Шаг 2. Когда в структуре имеются границы соприкосновения диэлектрика с проводником, необходимо работать в терминах полной плотности заряда σ_T, которая представляет собой сумму плотностей свободного и поляризационного зарядов:

$$\sigma_T(\mathbf{r}) = \sigma_S(\mathbf{r}) + \sigma_P(\mathbf{r}).$$

На границе диэлектрик-диэлектрик полная плотность заряда состоит только из плотности поляризационных зарядов.

Шаг 3. Учесть наличие плоскости земли с помощью метода зеркальных изображений. При этом вместо функции (5.41) необходимо использовать

$$G(\mathbf{r},\mathbf{r}') = \frac{\ln|\mathbf{r}-\mathbf{r}'|}{2\pi} - \frac{\ln|\mathbf{r}-\underline{\mathbf{r}}'|}{2\pi}, \qquad (5.43)$$

где $\underline{\mathbf{r}}'$ – точка мнимого источника. При отсутствии плоскости земли используется функция (5.41).

Шаг 4. Задать потенциалы (1 В) на проводниковых подынтервалах и подставить выражения для аппроксимации плотности заряда и функции Грина в уравнение (5.40). Тогда

$$\Phi(\mathbf{r}) = \frac{1}{2\pi\varepsilon_0}\int_L \sigma_T(\mathbf{r}')\left[\ln|\mathbf{r}-\mathbf{r}'| - \ln|\mathbf{r}-\underline{\mathbf{r}}'|\right]dl',\ \mathbf{r}\in L_C, \qquad (5.44)$$

где dl' – элемент контура границ проводник-диэлектрик; L_C – длина этого контура. Полученное уравнение является уравнением Фредгольма 1-го рода.

Шаг 5. Получить аналогичное уравнение для границ диэлектрик-диэлектрик. Следует учесть, что на границе диэлектрик-диэлектрик, т. е. между средами с диэлектрическими проницаемостями ε_1 и ε_2, нормальная составляющая общего вектора электрического смещения $\mathbf{D}^n = \varepsilon\mathbf{E}^n$ не меняется. Тогда

$$\varepsilon_1\mathbf{n}\cdot\mathbf{E}_1^n(\mathbf{r}) = \varepsilon_2\mathbf{n}\cdot\mathbf{E}_2^n(\mathbf{r}),\ \mathbf{r}\in L_D, \qquad (5.45)$$

где $\mathbf{n}$ – единичный вектор внешней нормали (см. рисунок 5.17); $\mathbf{E}_1^n(\mathbf{r})$ и $\mathbf{E}_2^n(\mathbf{r})$ – общее электрическое поле в средах 1 (с ε_1) и 2 (с ε_2) соответственно (в рассматриваемой структуре $\varepsilon_{r1} = 1$ (воздух)) при приближении к линии L_D границы раздела этих сред.

Далее используется связь потенциала с напряженностью поля

$$\mathbf{E}(\mathbf{r}) = -\nabla\Phi(\mathbf{r}).$$

Подставив уравнение (5.44) в последнее, получим

$$\mathbf{E}(\mathbf{r}) = -\frac{1}{2\pi\varepsilon_0}\int\limits_L \sigma_T(\mathbf{r}')\nabla\left[\ln|\mathbf{r}-\mathbf{r}'| - \ln|\mathbf{r}-\underline{\mathbf{r}}'|\right]dl' =$$

$$= \frac{1}{2\pi\varepsilon_0}\int\limits_L \sigma_T(\mathbf{r}')\left[\frac{\mathbf{r}-\mathbf{r}'}{|\mathbf{r}-\mathbf{r}'|^2} - \frac{\mathbf{r}-\underline{\mathbf{r}}'}{|\mathbf{r}-\underline{\mathbf{r}}'|^2}\right]dl',\ \mathbf{r}\in L_C, \qquad (5.46)$$

поскольку

$$\int\limits_L \nabla\left[\ln|\mathbf{r}-\mathbf{r}'| - \ln|\mathbf{r}-\underline{\mathbf{r}}'|\right]dl' = \int\limits_L \left[\frac{\mathbf{r}'-\mathbf{r}}{|\mathbf{r}-\mathbf{r}'|^2} - \frac{\underline{\mathbf{r}}'-\mathbf{r}}{|\mathbf{r}-\underline{\mathbf{r}}'|^2}\right]dl'.$$

Рассматривая предел уравнения (5.46), когда $\mathbf{r}$ приближается к границе раздела двух сред, можно показать, что он будет различным, если $\mathbf{r}$ приближается к границе со стороны среды 1 или со стороны среды 2. С учетом этого запишем

$$\begin{aligned}
\mathbf{E}_1(\mathbf{r}) &= \int\limits_S \frac{\sigma_T(\mathbf{r}')}{4\pi\varepsilon_0}\nabla\frac{1}{|\mathbf{r}-\mathbf{r}'|}dl' + \mathbf{n}\frac{\sigma_T(\mathbf{r})}{2\varepsilon_0}, \\
\mathbf{E}_2(\mathbf{r}) &= \int\limits_S \frac{\sigma_T(\mathbf{r}')}{4\pi\varepsilon_0}\nabla\frac{1}{|\mathbf{r}-\mathbf{r}'|}dl' - \mathbf{n}\frac{\sigma_T(\mathbf{r})}{2\varepsilon_0},\ \mathbf{r}\in L_D.
\end{aligned} \qquad (5.47)$$

В результате, подставив выражения (5.47) в уравнение (5.45), получим

$$0 = \frac{\varepsilon_2+\varepsilon_1}{\varepsilon_2-\varepsilon_1}\frac{\sigma_T(\mathbf{r})}{2\varepsilon_0} + \frac{1}{2\pi\varepsilon_0}\int\limits_L \sigma_T(\mathbf{r}')\left[\frac{\mathbf{r}-\mathbf{r}'}{|\mathbf{r}-\mathbf{r}'|^2} - \frac{\mathbf{r}-\underline{\mathbf{r}}'}{|\mathbf{r}-\underline{\mathbf{r}}'|^2}\right]\cdot\mathbf{n}dl,\ \mathbf{r}\in L_D. \qquad (5.48)$$

Шаг 6. Выразить полную плотность в виде линейной комбинации известных базисных функций ω_n и неизвестных коэффициентов α_n:

$$\sigma_T(\mathbf{r}) = \sum_{n=1}^{N}\alpha_n\omega_n(\mathbf{r}). \qquad (5.49)$$

Часто используемыми являются кусочно-постоянные базисные функции, которые равны единице на подынтервале с номером n и нулю вне его. Коэффициенты α_n соответствуют значению равномерной плотности заряда на подынтервалах с длиной l_n. Так, $q_n = l_n\alpha_n$ – погонный заряд на подынтервале длиной l_n (Кл/м).

Шаг 7. Подставить выражение (5.49) в соотношения (5.44) и (5.48) и, взяв для них скалярные произведения с тестовыми функциями (Дирака), сформировать СЛАУ вида $\mathbf{S\sigma} = \mathbf{v}$, где $\mathbf{S}$ – матрица размером $N \times N$, а $\boldsymbol{\sigma}$ и $\mathbf{v}$ – $N \times 1$. Вектор $\mathbf{v}$ содержит единицы в тех строках, которые соответствуют подынтервалам проводник-проводник.

Описанные шаги справедливы, если в структуре имеется один проводник, не считая опорного. При наличии нескольких проводников, как на рисунке 5.17, эти шаги повторять нет необходимости, а нужно лишь изменить вектор $\mathbf{v}$. Все используемые векторы $\mathbf{v}$ можно заменить на одну матрицу $\mathbf{V}$, состоящую из N_{COND} столбцов, где N_{COND} – число проводников в структуре, не считая опорного. Столбцы этой матрицы соответствуют векторам $\mathbf{v}$, сформированным для каждого из проводников. Тогда задача сводится к СЛАУ вида $\mathbf{S\Sigma} = \mathbf{V}$, где $\mathbf{\Sigma}$ и $\mathbf{V}$ – матрицы размером $N \times N_{\text{COND}}$. Структура полученной СЛАУ приведена на рисунке 5.18.

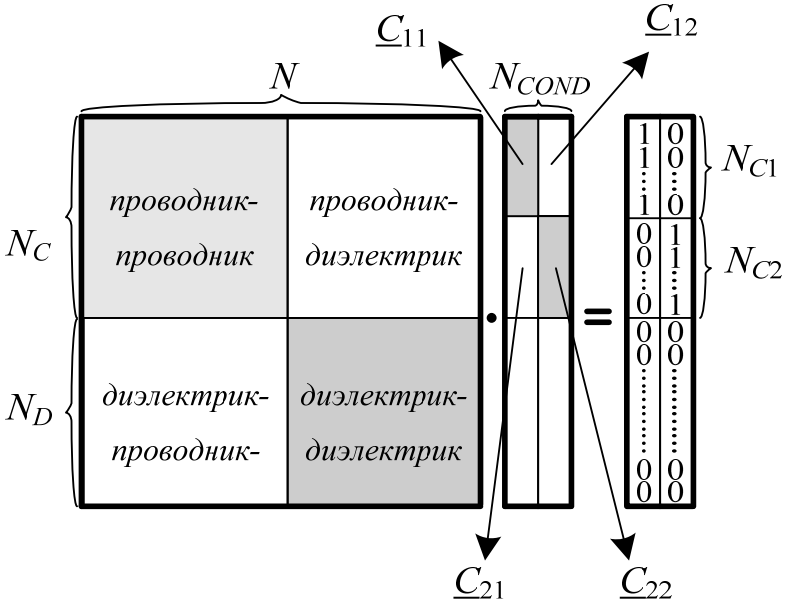


Рисунок 5.18 – Структура матрицы СЛАУ

Элементы матрицы **S** вычисляются по формулам [45]

$$s_{mn} = \frac{1}{2\pi\varepsilon_0} \int\limits_{L_n} \left[\ln |\, \mathbf{r}_m - \mathbf{r}'_n \,| - \ln |\, \mathbf{r}_m - \underline{\mathbf{r}}'_n \,| \right] dl', \tag{5.50}$$

$$m = 1,..., N_C, \;\; n = 1,..., N;$$

$$s_{mn} = \frac{1}{2\pi\varepsilon_0} \int\limits_{L_n} \left[\frac{\mathbf{r}_m - \mathbf{r}'_n}{|\, \mathbf{r}_m - \mathbf{r}'_n \,|^2} - \frac{\mathbf{r}_m - \underline{\mathbf{r}}'_n}{|\, \mathbf{r}_m - \underline{\mathbf{r}}'_n \,|^2} \right] dl', \tag{5.51}$$

$$m = N_C + 1,..., N, \;\; n = 1,..., N, \;\; m \neq n;$$

$$s_{mm} = \frac{\varepsilon_2 + \varepsilon_1}{2\varepsilon_0 (\varepsilon_2 - \varepsilon_1)} + \frac{1}{2\pi\varepsilon_0} \int\limits_{L_n} \left[\frac{\mathbf{r}_m - \mathbf{r}'_n}{|\, \mathbf{r}_m - \mathbf{r}'_n \,|^2} - \frac{\mathbf{r}_m - \underline{\mathbf{r}}'_n}{|\, \mathbf{r}_m - \underline{\mathbf{r}}'_n \,|^2} \right] dl', \tag{5.52}$$

$$m = N_C + 1,... N.$$

Для вычисления интегралов в уравнениях (5.50)–(5.52) используют численное интегрирование или аналитические выражения в замкнутом виде [45]. Полученная матрица **S** является плотной (практически полностью отсутствуют нулевые элементы). Если в рассматриваемой структуре отсутствует плоскость земли, то необходимо соблюсти закон сохранения заряда. Для этого к матрице СЛАУ **S** добавляются дополнительные строка и столбец, элементы которых вычисляются по простым формулам, а соответствующая строка матрицы **V** заполняется нулями [45]. При формировании СЛАУ вместо выражения (5.43) используется (5.41).

Шаг 8. Решить СЛАУ.

Шаг 9. Вычислить элементы емкостной матрицы $\underline{\mathbf{C}}$. При этом учесть, что поверхностная плотность свободных зарядов определяется как

$$\sigma_S(\mathbf{r}) = \varepsilon_r(\mathbf{r})\sigma_T(\mathbf{r}),\ \mathbf{r} \in L_{CD},$$

где $\varepsilon_r(\mathbf{r})$ – относительная диэлектрическая проницаемость диэлектрика, соприкасающегося с поверхностью проводника по контуру L_C. Тогда элементы емкостной матрицы

$$\underline{C}_{ij} = \int\limits_{L_{Ci}} \varepsilon_r(\mathbf{r})\sigma_S^j(\mathbf{r}) dl_i \Big/ V, \;\; V = 1 \text{ В},$$

где индекс i относится к проводнику, по контуру L_{C_i} которого ведется интегрирование, а j – к проводнику, находящемуся под потенциалом 1 В, когда остальные проводники под потенциалом 0 В. В матричном виде получим

$$\underline{C}_{ij} = \sum_{k \in L_{Ci}} \varepsilon_r \Sigma_{kj} l_k / V, \; i, j = 1, ..., N_{\text{COND}}, \; V = 1 \text{ В}. \qquad (5.53)$$

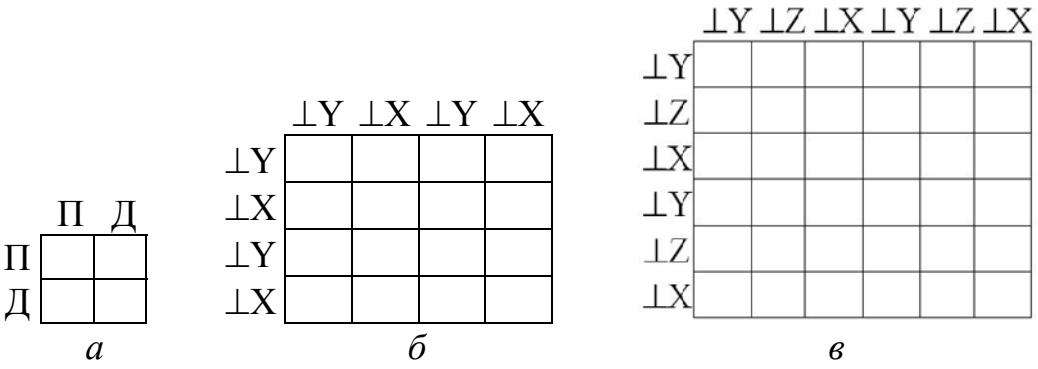
На рисунке 5.18 продемонстрированы блоки матрицы $\mathbf{\Sigma}$, участвующие в вычислении.

В результате задача нахождения емкостной матрицы МПЛП сводится к решению СЛАУ вида $\mathbf{S\Sigma = V}$ с квадратной и плотной матрицей $\mathbf{S}$ размером $N \times N$ ($N = N_C + N_D$), связывающей плотности заряда на подобластях дискретизированных границ проводников и диэлектриков, составляющих матрицу $\mathbf{\Sigma}$, с потенциалами этих подобластей, задаваемыми матрицей $\mathbf{V}$, обе матрицы размером $N \times N_{\text{COND}}$. После вычисления матрицы $\mathbf{\Sigma}$ вычисляются элементы искомой емкостной матрицы.

Еще раз вернемся к структуре матрицы СЛАУ (см. рисунок 5.18). Для общего случая произвольно ориентированных границ проводников и диэлектриков эта матрица имеет структуру, показанную на рисунке 5.19, *а*, где демонстрируется расположение ее элементов, соответствующих проводниковым (П) и диэлектрическим (Д) границам. Для частного случая линейных и ортогональных границ двухмерной структуры [45] каждый блок матрицы на рисунке 5.19, *а* имеет структуру, показанную на рисунке 5.19, *б*, т. е. состоит из 4 субблоков, соответствующих границам, которые ортогональны осям Y ($\perp Y$) и X ($\perp X$). В случае трехмерных структур с ортогональными осям границами организация субблоков показана на рисунке 5.19, *в*.

Таким образом, сначала сегментируются границы проводник-диэлектрик и подынтервалам присваиваются номера с 1 по N_C. В первую очередь сегментируются и последовательно нумеруются подынтервалы, которые ортогональны оси Y (номер последнего N_{CY}), а затем ортогональные оси X (до N_C). Далее сегментируются границы диэлектрик-диэлектрик и полученным подын-

тервалам присваиваются номера с $N_C + 1$ по N. При этом сначала сегментируются и последовательно нумеруются подынтервалы, которые ортогональны оси Y (номер последнего N_{DY}), а затем ортогональные оси X (до N).



а *б* *в*

Рисунок 5.19 – Структуры матрицы СЛАУ для произвольно (*а*) и ортогонально ориентированных границ двухмерных (*б*) и трехмерных (*в*) структур

Каждый подынтервал описывается уникальным набором параметров, в который входят: x_n – координата X центра n-го подынтервала; y_n – координата Y центра n-го подынтервала; d_n – длина n-го подынтервала; ε_n – относительная диэлектрическая проницаемость вблизи n-го подынтервала границы проводник-диэлектрик; ε_n^+ и ε_n^- – относительные диэлектрические проницаемости на положительной (на которую направлен вектор $\boldsymbol{n}_n$) и отрицательной (от которой направлен вектор $\boldsymbol{n}_n$) сторонах n-го подынтервала границы диэлектрик-диэлектрик соответственно; $\boldsymbol{n}_n$ – единичный вектор нормали из центра n-го подынтервала в направлении соответствующей оси. Из этих параметров подынтервалов вычисляются элементы матрицы СЛАУ по приведенным ниже формулам [45].

Для строк с индексами $m = 1, \ldots, N_C$

$$s_{mm} = -\frac{I_{mn}}{2\pi\varepsilon_0},\ m=1, \ldots, N_C,\ n = 1, \ldots, N, \qquad (5.54)$$

где

$$I_{mn} = a_1 \cdot \ln\left(a_1^2 + c_1^2\right) - 2a_1 + 2c_1 \cdot \arctan\left(\frac{a_1}{c_1}\right) -$$

$$- a_2 \cdot \ln\left(a_2^2 + c_2^2\right) + 2a_2 - 2c_1 \cdot \arctan\left(\frac{a_2}{c_1}\right).$$

При этом для $n = 1, \ldots, N_{CY}, (N_C + 1), \ldots, N_{DY}$

$$a_1 = \frac{d_n}{2} - (x_m - x_n);\ \ a_2 = -\frac{d_n}{2} - (x_m - x_n);\ \ c_1 = y_m - y_n, \qquad (5.55)$$

а для $n = (N_{CY} + 1), \ldots, N_C, (N_{DY} + 1), \ldots, N$

$$a_1 = \frac{d_n}{2} - (y_m - y_n);\ a_2 = -\frac{d_n}{2} - (y_m - y_n);\ c_1 = x_m - x_n. \qquad (5.56)$$

Для строк с индексами $m = (N_C + 1), \ldots, N$

$$s_{mm} = \frac{I_{mn}}{2\pi\varepsilon_0}\ ,\ m = (N_C + 1), \ldots, N,\ \ n = 1, \ldots, N,\ \ m \neq n;$$

$$s_{mm} = \frac{I_{mn}}{2\pi\varepsilon_0} + \frac{1}{2\varepsilon_0}\frac{\varepsilon_m^+ + \varepsilon_m^-}{\varepsilon_m^+ - \varepsilon_m^-},\ \ m = (N_C + 1), ..., N,$$

где для строк с индексами $m = (N_C + 1), \ldots, N_{DY}$ при $n = 1, \ldots, N_{CY}, (N_C + 1), \ldots, N_{DY}$

$$I_{mn} = \arctan\left(\frac{a_1}{c_1}\right) - \arctan\left(\frac{a_2}{c_1}\right) \qquad (5.57)$$

и переменные аналогичны (5.55), а при $n = (N_{CY} + 1), \ldots, N_C, (N_{DY} + 1), \ldots, N$

$$I_{mn} = \frac{1}{2}\ln\left(\frac{a_2^2 + c_1^2}{a_1^2 + c_1^2}\right), \qquad (5.58)$$

где переменные аналогичны (5.56).

Для строк с индексами $m = (N_{DY} + 1), \ldots, N$ при $n = 1, \ldots, N_{CY}, (N_C + 1), \ldots, N_{DY}$ I_{mn} вычисляется по формуле (5.58) с переменными (5.55), а при $n = (N_{CY} + 1), \ldots, N_C, (N_{DY} + 1), \ldots, N$ – по формуле (5.57) с переменными (5.56).

Когда плоскость земли отсутствует, землей считается $(N_{\mathrm{COND}}+1)$-й проводник. Тогда добавляются $(N+1)$-я строка и $(N+1)$-й столбец с элементами

$$s_{nN+1}=\frac{d_n}{2s_{nn}},\quad s_{N+1n}=d_n\varepsilon_n,\quad n=1,\ldots,N_C.$$

5.5 Адаптивная перекрестная аппроксимация

При использовании метода моментов задача сводится к решению СЛАУ с плотной матрицей. Затраты машинной памяти пропорциональны $O(N^2)$, а затраты на ее решение, например, методом Гаусса – $O(N^3)$, что затрудняет применение этого метода при решении задач со сложной геометрией исследуемой структуры. Одним из способов экономии вычислительных ресурсов при использовании метода моментов является адаптивная перекрестная аппроксимация (ACA).

Суть ACA состоит в представлении исходной матрицы **S** в виде произведения матриц **U** и **V** меньшего ранга. Преимущества данного алгоритма заключаются в его алгебраическом характере. За счет методов линейной алгебры, таких как QR-разложение, сингулярное разложение, LU-разложение и др., достигается ускорение вычислений. Алгоритм может быть модульно реализован и легко интегрирован в различные программы на основе метода моментов. Однако матрица СЛАУ обладает сингулярностью на резонансных частотах, поэтому применить ACA к исходной матрице **S** не представляется возможным. В то же время вследствие специфики функции Грина данная матрица состоит из блоков, соответствующих взаимодействию хорошо сепарабельных базисных функций, благодаря чему они могут быть представлены в виде иерархических матриц. Таким образом, в общем случае алгоритм ACA представляет собой многоуровневую систему сжатия матриц, которая обеспечивает ускорение вычислений. Следует отметить, что алгоритм аппроксимирует исходную матрицу, требуя лишь частичную информацию о ней. Поясним суть ACA.

Пусть $\mathbf{S} = \mathbf{R} + \tilde{\mathbf{S}}$, где $\mathbf{R}$ – матрица ошибки аппроксимации; $\tilde{\mathbf{S}}$ – матрица малого ранга (ранг $\tilde{\mathbf{S}} \le r$, $r << N$). Требуется найти матрицу аппроксимации $\tilde{\mathbf{S}}$ для плотной матрицы $\mathbf{S}$ (размером $N \times M$) в виде произведения матриц $\mathbf{U}$ и $\mathbf{V}$, т. е. $\tilde{\mathbf{S}} = \mathbf{UV} = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i$, где $\mathbf{U}$ – матрица размером $M \times r$, а $\mathbf{V}$ – матрица размером $r \times N$, с точностью, удовлетворяющей условию минимизации матрицы ошибки

$$\|\mathbf{R}\|_F = \|\mathbf{S} - \tilde{\mathbf{S}}\|_F \le \mathrm{TOL}\|\mathbf{S}\|_F,$$

где TOL – требуемая точность.

Для ясности дальнейшего изложения введем обозначения (с использованием синтаксиса Octave). Пусть имеем матрицы $\mathbf{I} = [I_1 \ldots I_r]$ и $\mathbf{J} = [J_1 \ \ldots \ J_r]$, содержащие выбранные строчные и столбцовые индексы матрицы $\mathbf{S}$. Для обозначения строки I_k, например, матрицы $\mathbf{R}$ используется запись $\mathbf{R}(I_k, :)$, а для столбца J_k – запись $\mathbf{R}(:, J_k)$. Тогда АСА можно представить в виде следующего алгоритма.

Алгоритм адаптивной перекрестной аппроксимации

Задать ε. Положить $\tilde{\mathbf{S}} = \mathbf{0}$ и $I_1 = 0$.
$\mathbf{R}(I_1, :) = \mathbf{S}(I_1, :)$
Найти индекс J_1, удовлетворяющий: $|\mathbf{R}(I_1, J_1)| = \max_j(|\mathbf{R}(I_1, j)|)$
$\mathbf{v}_1 = \mathbf{R}(I_1, :) / \mathbf{R}(I_1, J_1)$
$\mathbf{R}(:, J_1) = \mathbf{S}(:, J_1)$
$\mathbf{u}_1 = \mathbf{R}(:, J_1)$
$\| \tilde{\mathbf{S}}^{(1)} \|_F^2 = \| \tilde{\mathbf{S}}^{(0)} \|_F^2 + \| \mathbf{u}_1 \|_F^2 \| \mathbf{v}_1 \|_F^2$
Найти индекс I_2, удовлетворяющий: $|\mathbf{R}(I_2, J_1)| = \max_i(|\mathbf{R}(i, J_1)|)$, $i \ne I_1$
Для $k = 2, 3, \ldots,$ вычислить

$$\mathbf{R}(I_k,:) = \mathbf{S}(I_k,:) - \sum_{l=1}^{k-1} (\mathbf{u}_l)_{I_k} \mathbf{v}_l$$

Найти J_k, удовлетворяющий: $|\mathbf{R}(I_k, J_k)| = \max_j(|\mathbf{R}(I_k, j)|)$, $j \ne J_1, \ldots, J_{k-1}$
Если
$\mathbf{R}(I_k, J_k) = 0$, то конец, аппроксимация не получена
Иначе
$\mathbf{v}_k = \mathbf{R}(I_k, :) / \mathbf{R}(I_k, J_k)$

$\mathbf{R}(:, J_k) = \mathbf{S}(:, J_k) - \sum_{l=1}^{k-1} (\mathbf{v}_l)_{J_k} \mathbf{u}_l$

$\mathbf{u}_k = \mathbf{R}(:, J_k)$

$\| \widetilde{\mathbf{S}}^{(k)} \|_F^2 = \| \widetilde{\mathbf{S}}^{(k-1)} \|_F^2 + 2 \sum_{j=1}^{k-1} | \mathbf{u}_j^T \mathbf{u}_k | | \mathbf{v}_j^T \mathbf{v}_k | + \| \mathbf{u}_k \|_F^2 \| \mathbf{v}_k \|_F^2$

Если

$\|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 \leq \varepsilon \| \widetilde{\mathbf{S}}^{(k)} \|_2$, то **конец** итерационного процесса, $r = k$

Иначе

Найти индекс I_{k+1}, удовлетворяющий равенству $|\mathbf{R}(I_{k+1}, J_k)| = \max_i(|\mathbf{R}(i, J_k)|)$, $i \neq I_1, \ldots, I_k$

Увеличить k

Из алгоритма видно, что для построения аппроксимации не нужно заранее знать все элементы исходной матрицы **S**. Для выполнения вычислений с помощью данного алгоритма требуется не более r итераций, сложность каждой из которых пропорциональна $O(r(M + N))$. Таким образом, общие вычислительные затраты алгоритма пропорциональны $O(r^2 (M + N))$. Далее приведен программный код Octave, предназначенный для считывания матрицы и ее разложения.

```
function ACA_main
clc;
clear;
A=dlmread('matrix_forACA_.txt');
b=A;
[n,m]=size(b);
eps_tol=1.e-3;
[U,V] = ACA(b,eps_tol,n,m);
size(U)
size(V)
norm(b-U*V)
function [U,V] = ACA(A,ACA_tol,M,N)
  J = zeros(N,1);
  I = zeros(M,1);
  i = (2:M);
  j = (1:N);
  I(1) = 1;
  Rik=A(I(1),:);
  col = find( abs(Rik(j)) == max(abs(Rik(j))) );
  J(1) = j(col(1));
```

```
 j(j==J(1))=[];
 V = Rik/Rik(J(1));
 Rjk=A(:,J(1));
 U = Rjk;
 normZ = norm(U)^2 * norm(V)^2;
 row = find( abs(Rjk(i)) == max(abs(Rjk(i))) );
 I(2) = i(row(1));
 i(i==I(2))=[];
 for k=2:min(M,N)
    Rik=A(I(k),:)- U(I(k),:)*V;
    col = find(abs(Rik(j)) == max(abs(Rik(j))) );
    J(k) = j(col(1));
    j(j==J(k))=[];
    if(Rik(J(k)) == 0)
       break;
    end
    Vk = Rik/Rik(J(k));
    Rjk=A(:,J(k)) - U*V(:,J(k));
    Uk = Rjk;
    normZ = normZ + 2*sum(real((U'*Uk).*(Vk*V').')) +...
    norm(Uk)^2*norm(Vk)^2;
    U = [U Uk]; V = [V; Vk];
    if norm(Uk)*norm(Vk) <= ACA_tol*sqrt(normZ)
       break
    end
    if k==min(M,N)
       break;
    end
    row = find( abs(Rjk(i)) == max(abs(Rjk(i))) );
    I(k+1) = i(row(1));
    i(i==I(k+1))=[];
end
```

Контрольные вопросы и задания

1. Для решения каких уравнений применяется метод моментов?

2. К СЛАУ с какой матрицей сводит задачу метод моментов?

3. На каких два вида делятся системы базисных функций, используемых в методе моментов?

4. Какой тип тестовых функций используется в методе коллокаций?

5. Назовите тип тестовых функций в методе Галёркина.

6. Назовите последовательность действий при использовании метода моментов для решения электростатических задач.

7. Для чего применяется адаптивная перекрестная аппроксимация?

8. Разработать программу на языке Octave для аппроксимации функции (5.36) с использованием кусочно-синусоидальных базисных функций.

9. Модифицировать программный код из листинга 5.4 для учета симметрии структуры и тем самым уменьшения вычислительных затрат.

10. Разработать программу на языке Octave для вычисления распределения заряда на поверхности структуры, изображенной на рисунке 3.16 (при $a = b = 1$ см, $c = d = 2$ см), и ее емкости.

6 МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ

6.1 Конечные элементы

Математическая трактовка метода конечных элементов (МКЭ, FEM) была предложена Р. Курантом в 1943 г. Первоначально метод применялся при решении задач строительной механики. Для решения электромагнитных задач он начал использоваться с 1968 г. при анализе волноводов, электрических машин, полупроводниковых приборов, микрополосковых линий, электромагнитного излучения биологическими объектами и т. д. По сравнению с МКР и МоМ МКЭ является более мощным и универсальным численным методом, подходящим для решения задач, связанных со сложной геометрией и неоднородными средами. В то же время МКЭ считается сложным с точки зрения его концепции и реализации в программном коде. Методическая общность метода позволяет строить на его основе универсальные компьютерные программы для решения широкого круга задач. При этом программы, разработанные для решения задач из одной предметной области, могут успешно применяться в других областях с незначительными модификациями или без таковых.

Использование МКЭ в общем виде состоит из следующих этапов.

– Дискретизация области решения на конечное число подобластей (конечных элементов) и выбор базисных функций. В рассматриваемой области решения фиксируется конечное число точек, называемых узлами или узловыми точками. Область определения непрерывной величины разбивается на конечное число элементов. Эти элементы имеют общие узловые точки и в совокупности аппроксимируют форму области решения.

– Формирование уравнений для каждого конечного элемента. На данном этапе формируются локальные матрицы, непрерывную величину аппроксимируют на каждом элементе полиномом (функцией элемента), который определяется с помощью узловых значений этой величины. Для каждого элемента подбирают свой

полином таким образом, чтобы сохранить непрерывность величины вдоль границ элемента.

– Сбор всех элементов в области решения в конечно-элементную сетку (ансамблирование). Результаты аппроксимации подставляются в уравнения Максвелла или производные от них с учетом граничных условий. В результате формируется общая СЛАУ.

– Решение общей СЛАУ.

– Вычисление интересующих величин из полученного вектора-решения СЛАУ.

Дискретизация области решения заключается в ее разбиении на подобласти, называемые конечными элементами (КЭ). На рисунке 6.1 показаны некоторые типовые конечные элементы для одномерных, двухмерных и трехмерных задач.

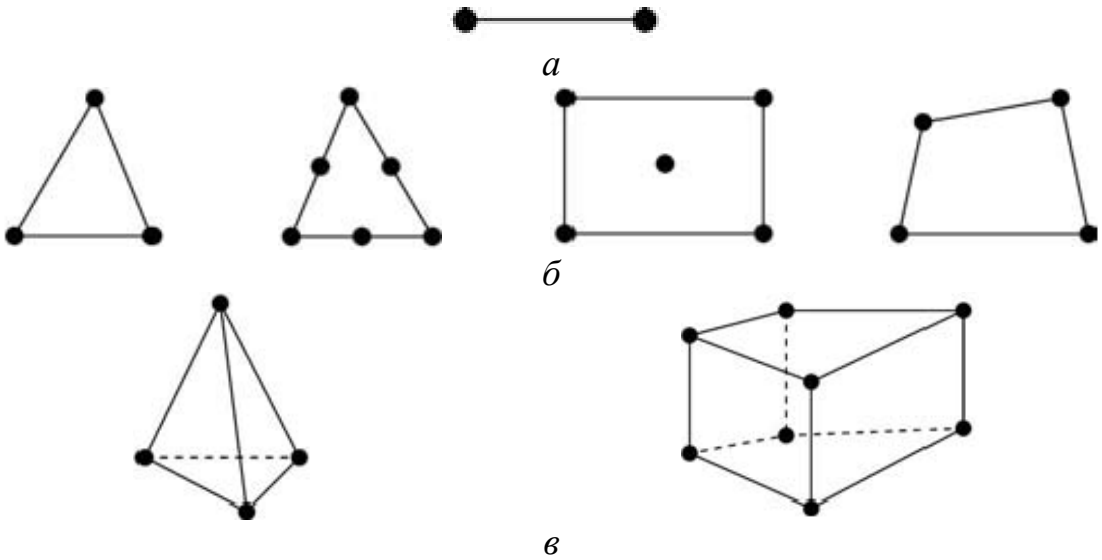


Рисунок 6.1 – Типовые конечные элементы для одномерных (*а*), двухмерных (*б*) и трехмерных (*в*) задач

Самыми распространенными функциями, которые используются для аппроксимации непрерывной функции дискретной моделью, являются полиномы. Они строятся на множестве кусочно-непрерывных функций, определенных на конечном числе подобластей. Порядок полинома в каждом узле элемента зависит от числа используемых данных непрерывной функции. Классификация КЭ может быть выполнена в соответствии с порядком полиномиальных функций этих элементов. Рассматриваются три группы элементов: симплекс-, комплекс- и мультиплекс-элементы.

Первой группе соответствуют полиномы, содержащие константу и линейные члены, в которых число коэффициентов на единицу больше размерности координатного пространства.

Комплекс-элементы являются расширением класса симплекс-элементов. В них могут присутствовать не только линейные слагаемые, но и члены более высоких порядков. Формы комплекс-элементов такие же, как и у симплекс-элементов, но имеют дополнительные граничные узлы и могут иметь внутренние узлы. Главное различие между симплекс- и комплекс-элементами состоит в том, что число узлов у последних больше размерности координатного пространства, увеличенной на единицу.

Мультиплекс-элементы также содержат члены высокого порядка, но границы элементов должны быть параллельны координатным осям, что необходимо для достижения непрерывности при переходе от одного элемента к другому. Это требование достаточно строгое, поэтому мультиплекс-элементы используют в редких случаях.

Рассмотрим одномерный симплекс-элемент, представляющий собой отрезок с двумя узлами (рисунок 6.2).

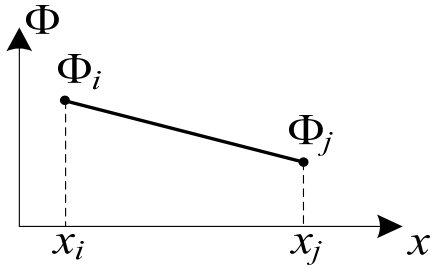


Рисунок 6.2 – Одномерный конечный элемент
с узлами (двухузловой симплекс-элемент)

Обозначим эти узлы индексами i и j, а узловые значения Φ_i и Φ_j соответственно. Начало координат расположено вне рассматриваемого КЭ. Полиномиальная функция имеет вид

$$\Phi = a + bx . \qquad (6.1)$$

Используя условия в узловых точках, можно найти коэффициенты a и b ($x_j - x_i = L$) из системы уравнений

$$\Phi_i = a + bx_i,$$
$$\Phi_j = a + bx_j.$$

Тогда

$$a = \frac{\Phi_i x_j - \Phi_j x_i}{L}, \quad b = \frac{\Phi_j - \Phi_i}{L}. \tag{6.2}$$

Подставив коэффициенты (6.2) в уравнение (6.1), получим

$$\Phi = \frac{x_j - x}{L}\Phi_i + \frac{x - x_i}{L}\Phi_j. \tag{6.3}$$

Линейные функции в равенстве (6.3) называются функциями формы или интерполяционными функциями. Каждая функция формы должна быть дополнена нижним индексом для обозначения узла, к которому она относится. Обозначим $\alpha_i = (x_j - x)/L$ и $\alpha_j = (x_i - x)/L$. Тогда выражение (6.3) примет вид

$$\Phi = \alpha_i\Phi_i + \alpha_j\Phi_j = \begin{pmatrix}\alpha_i & \alpha_j\end{pmatrix}\begin{pmatrix}\Phi_i \\ \Phi_j\end{pmatrix}. \tag{6.4}$$

Очевидно, что функция α_i равна единице в *i*-м узле и равна нулю в *j*-м, а функция α_j наоборот.

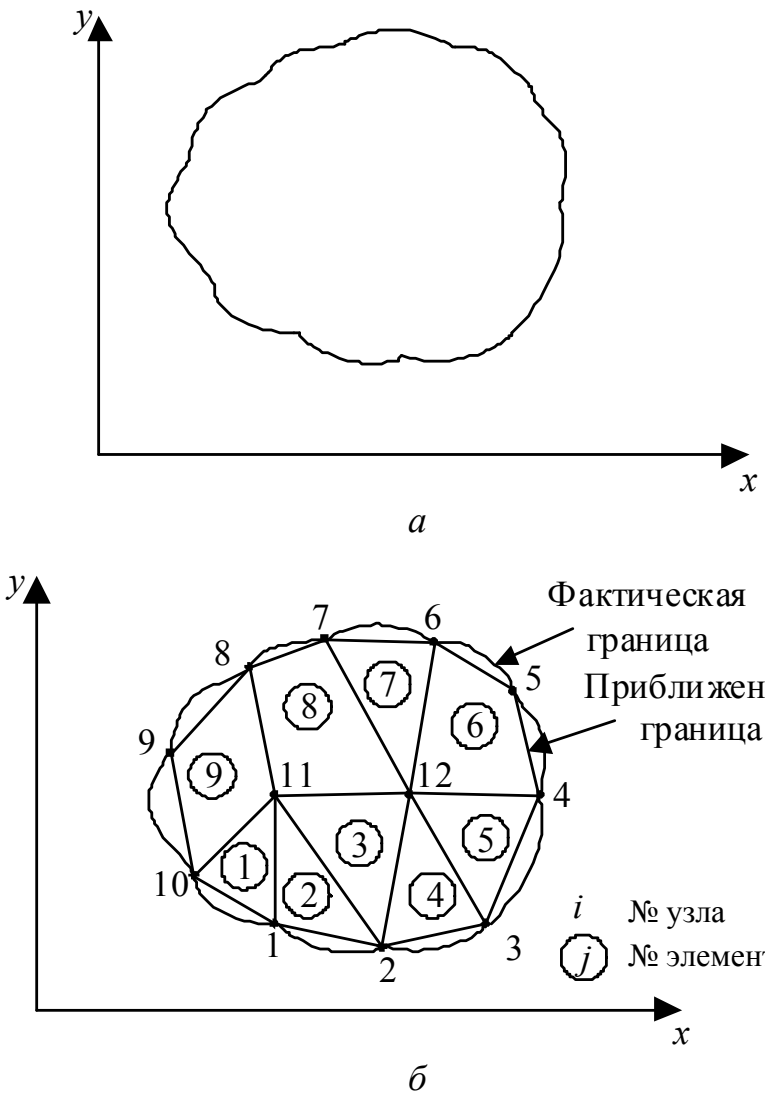
6.2 Решение двухмерного уравнения Лапласа

6.2.1 Дискретизация области

Найдем распределение потенциала $\Phi(x, y)$, соответствующее уравнению Лапласа $\nabla^2\Phi = 0$, в двухмерной области (рисунок 6.3, *а*).

Разделим область решения на несколько КЭ (рисунок 6.3, *б*). В данном случае это девять неперекрывающихся элементов: элементы 6, 8 и 9 – четырехузловые четырехугольники; остальные элементы – трехузловые треугольники. Для удобства вычислений предпочтительно разбивать всю область на элементы одного и того же типа. Так, если разбить четырехугольники на два треугольника каждый, то получим 12 треугольных элементов. Разбиение

областей несложной формы может выполняться вручную, а сложной – с помощью автоматической генерации сетки.



а

б

Рисунок 6.3 – Область решения (*а*)
и ее дискретизация конечными элементами (*б*)

Найдем приближение для потенциала Φ_e внутри элемента e, а затем свяжем распределение потенциала во всех элементах с его непрерывностью на межэлементных границах. Приближенное решение для всей области определяется как

$$\Phi(x, y) = \sum_{e=1}^{N} \Phi_e(x, y), \qquad (6.5)$$

где N – число КЭ, на которое разбита расчетная область.

Для приближенного вычисления потенциала Φ_e внутри элемента используем полиномиальные базисные функции. Для треугольного КЭ

$$\Phi_e(x, y) = a + bx + cy, \tag{6.6}$$

а для четырехугольного КЭ

$$\Phi_e(x, y) = a + bx + cy + dxy. \tag{6.7}$$

Тогда расчетная область, в которой ищется решение, является пространством кусочно-полиномиальных функций.

Таким образом, для приближенного решения должны быть определены коэффициенты a, b, c и d. Потенциал Φ_e отличен от нуля в пределах элемента e и равен нулю вне его. Поскольку треугольные элементы по сравнению с четырехугольными лучше подходят для описания изогнутых границ, то далее используем только их. Линейное изменение потенциала внутри элемента предполагает равномерность электрического поля внутри него, т. е.

$$\mathbf{E}_e = -\nabla\Phi_e = -(b\mathbf{i} + c\mathbf{j}). \tag{6.8}$$

6.2.2 Формирование уравнений отдельного конечного элемента

Рассмотрим типовой треугольный КЭ e (рисунок 6.4). Потенциалы Φ_{e1}, Φ_{e2} и Φ_{e3} в узлах 1, 2 и 3 соответственно получаются с помощью уравнения (6.6):

$$\begin{pmatrix} \Phi_{e1} \\ \Phi_{e2} \\ \Phi_{e3} \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}. \tag{6.9}$$

Коэффициенты a, b и c определяются из СЛАУ (6.9), например, с помощью обратной матрицы:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}^{-1} \begin{pmatrix} \Phi_{e1} \\ \Phi_{e2} \\ \Phi_{e3} \end{pmatrix}. \tag{6.10}$$

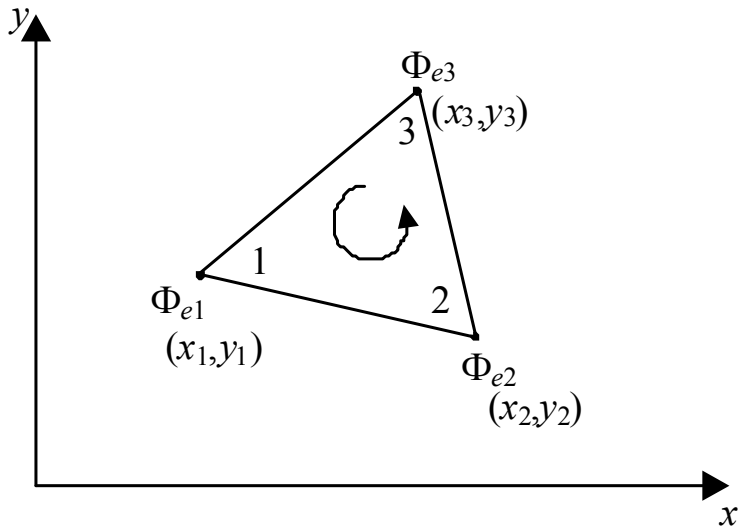


Рисунок 6.4 – Типовой треугольный конечный элемент
(нумерация локальных узлов против часовой стрелки)

Подставив уравнение (6.10) в (6.6) и воспользовавшись обращением матрицы на основе союзной матрицы, получим

$$\Phi_e = \begin{pmatrix} 1 & x & y \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & x & y \end{pmatrix} \frac{1}{2S} \begin{pmatrix} x_2 y_3 - x_3 y_2 & x_3 y_1 - x_1 y_3 & x_1 y_2 - x_2 y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{pmatrix} \begin{pmatrix} \Phi_{e1} \\ \Phi_{e2} \\ \Phi_{e3} \end{pmatrix}$$

или

$$\Phi_e = \sum_{i=1}^{3} \alpha_i (x, y) \Phi_e^i, \qquad (6.11)$$

где Φ_e^i – потенциал в i-м узле;

$$\alpha_1 = \frac{1}{2S} \left[\left(x_2 y_3 - x_3 y_2 \right) + \left(y_2 - y_3 \right) x + \left(x_3 - x_2 \right) y \right]; \qquad (6.12)$$

$$\alpha_2 = \frac{1}{2S} \left[\left(x_3 y_1 - x_1 y_3 \right) + \left(y_3 - y_1 \right) x + \left(x_1 - x_3 \right) y \right]; \qquad (6.13)$$

$$\alpha_3 = \frac{1}{2S} \left[\left(x_1 y_2 - x_2 y_1 \right) + \left(y_1 - y_2 \right) x + \left(x_2 - x_1 \right) y \right]; \qquad (6.14)$$

S – площадь элемента e, определяемая из уравнения [46]

$$2S = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} = \left[(x_2 y_3 - x_3 y_2) + (x_3 y_1 - x_1 y_3) + (x_1 y_2 - x_2 y_1) \right],$$

или

$$S = \frac{1}{2}\left[(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \right]. \qquad (6.15)$$

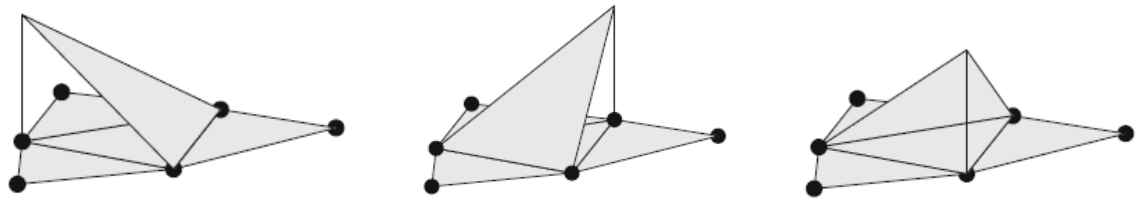
Значение величины S положительно, если узлы пронумерованы против часовой стрелки (начиная с любого узла), как показано стрелкой на рисунке 6.4. Заметим, что выражение (6.11) дает потенциал в любой точке внутри элемента с координатами (x, y) при условии, что потенциалы в его вершинах известны. Это является одним из отличий от конечно-разностного подхода, где потенциал известен только в узлах сетки.

Как и ранее, α_i – функции формы, которые обладают следующими свойствами:

$$\alpha_i = \begin{cases} 1, & i = j, \\ 0, & i \neq j; \end{cases} \qquad (6.16)$$

$$\sum_{i=1}^{3} \alpha_i(x, y) = 1. \qquad (6.17)$$

Функции формы α_1, α_2 и α_3 показаны на рисунке 6.5.



а *б* *в*

Рисунок 6.5 – Функции формы треугольного элемента (также показан смежный элемент): *а* – α_1; *б* – α_2; *в* – α_3

В соответствии с принципом минимума потенциальной энергии распределение потенциала в рассматриваемой области должно быть таким, чтобы минимизировать запасенную энергию

$$W_e = \frac{1}{2}\int \varepsilon |\nabla \Phi_e|^2 d\Omega, \tag{6.18}$$

где интегрирование ведется по всей двухмерной области решения Ω. Физически функционал W_e – это погонная энергия, запасенная в элементе e.

Принцип минимума энергии математически эквивалентен уравнению Лапласа $\nabla^2 \Phi = 0$. Распределение потенциала, удовлетворяющее уравнению Лапласа, соответствует минимальной запасенной энергии, а значение потенциала Φ_e, минимизирующее функционал (6.18), удовлетворяет уравнению Лапласа. Поэтому возможны два варианта решения краевых задач теории поля.

Из уравнения (6.11) получим

$$\nabla \Phi_e = \sum_{i=1}^{3} \Phi_{ei} \nabla \alpha_i. \tag{6.19}$$

Подстановка выражения (6.19) в (6.18) дает

$$W_e = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} \varepsilon \Phi_{ei} \left[\int \nabla \alpha_i \cdot \nabla \alpha_j d\Omega\right] \Phi_{ei}. \tag{6.20}$$

Если обозначить выражение в скобках через

$$c_{ij}^{(e)} = \int \nabla \alpha_i \cdot \nabla \alpha_j d\Omega, \tag{6.21}$$

то уравнение (6.20) в матричной форме примет вид

$$W_e = \frac{1}{2} \varepsilon \mathbf{\Phi}_e^t \mathbf{C}^{(e)} \mathbf{\Phi}_e, \tag{6.22}$$

где верхний индекс t обозначает транспонирование матрицы;

$$\mathbf{\Phi}_e = \begin{bmatrix} \Phi_{e1} \\ \Phi_{e2} \\ \Phi_{e3} \end{bmatrix}; \tag{6.23}$$

$$\mathbf{C}^{(e)} = \begin{bmatrix} c_{11}^{(e)} & c_{12}^{(e)} & c_{13}^{(e)} \\ c_{21}^{(e)} & c_{22}^{(e)} & c_{23}^{(e)} \\ c_{31}^{(e)} & c_{32}^{(e)} & c_{33}^{(e)} \end{bmatrix}. \tag{6.24}$$

Матрица $\mathbf{C}^{(e)}$ называется матрицей коэффициентов. Элемент $c_{ij}^{(e)}$ этой матрицы соответствует связи между узлами *i* и *j*, а его значение вычисляется по формуле (6.21) с помощью выражений (6.12)–(6.14). Например,

$$c_{12}^{(e)} = \int \nabla\alpha_1 \cdot \nabla\alpha_2 d\Omega =$$

$$= \int \left[\frac{1}{2S}\left(\left(y_2 - y_3\right) + \left(x_3 - x_2\right)\right) \cdot \frac{1}{2S}\left(\left(y_3 - y_1\right) + \left(x_1 - x_3\right)\right) \right] d\Omega =$$

$$= \frac{1}{4S^2}\left[\left(y_2 - y_3\right)\left(y_3 - y_1\right) + \left(x_3 - x_2\right)\left(x_1 - x_3\right)\right] \int d\Omega =$$

$$= \frac{1}{4S}\left[\left(y_2 - y_3\right)\left(y_3 - y_1\right) + \left(x_3 - x_2\right)\left(x_1 - x_3\right)\right]. \qquad (6.25)$$

Аналогично получим

$$c_{13}^{(e)} = \frac{1}{4S}\left[\left(y_2 - y_3\right)\left(y_1 - y_2\right) + \left(x_3 - x_2\right)\left(x_2 - x_1\right)\right]; \qquad (6.26)$$

$$c_{23}^{(e)} = \frac{1}{4S}\left[\left(y_3 - y_1\right)\left(y_1 - y_2\right) + \left(x_1 - x_3\right)\left(x_2 - x_1\right)\right]; \qquad (6.27)$$

$$c_{11}^{(e)} = \frac{1}{4S}\left[\left(y_2 - y_3\right)^2 + \left(x_3 - x_2\right)^2\right]; \qquad (6.28)$$

$$c_{22}^{(e)} = \frac{1}{4S}\left[\left(y_3 - y_1\right)^2 + \left(x_1 - x_3\right)^2\right]; \qquad (6.29)$$

$$c_{33}^{(e)} = \frac{1}{4S}\left[\left(y_1 - y_2\right)^2 + \left(x_2 - x_1\right)^2\right]. \qquad (6.30)$$

При этом

$$c_{21}^{(e)} = c_{12}^{(e)}, c_{31}^{(e)} = c_{13}^{(e)}, c_{32}^{(e)} = c_{23}^{(e)}. \qquad (6.31)$$

Для примера рассмотрим один конечный элемент (рисунок 6.6). На языке Octave этот элемент описывается массивом координат своих узлов [–0.5 0.0 0.6; 0.5 –0.2 0.4].

По выражению (6.15) получим площадь треугольника

$$S = \frac{1}{2}\left[(0.0 + 0.5)(0.4 - 0.5) - (0.6 + 0.5)(-0.2 - 0.5)\right] = 0.36.$$

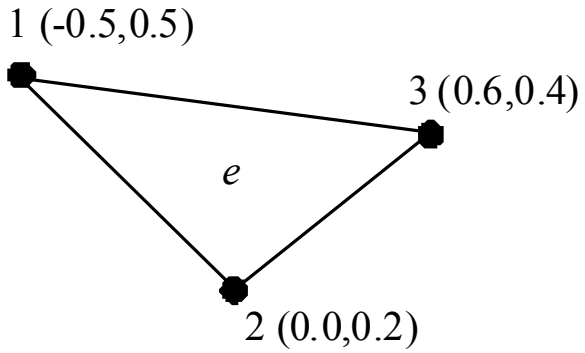


Рисунок 6.6 – Конечный элемент e
с локальной нумерацией узлов

Далее вычислим элементы матрицы коэффициентов $\mathbf{C}^{(e)}$:

$$c_{11}^{(e)} = \frac{1}{4 \cdot 0.36}\left[\left(-0.2 - 0.4\right)^2 + \left(0.6 - 0.0\right)^2 \right] = 0.5;$$

$$c_{12}^{(e)} = \frac{1}{4 \cdot 0.36}\left[\left(-0.2 - 0.4\right)\left(0.4 - 0.5\right) + \left(0.6 - 0.0\right)\left(-0.5 - 0.6\right) \right] = -0.4167;$$

$$c_{13}^{(e)} = \frac{1}{4 \cdot 0.36}\left[\left(-0.2 - 0.4\right)\left(0.5 + 0.2\right) + \left(0.6 - 0.0\right)\left(0.0 + 0.5\right) \right] = -0.0833;$$

$$c_{22}^{(e)} = \frac{1}{4S}\left[\left(y_3 - y_1\right)^2 + \left(x_1 - x_3\right)^2 \right] =$$

$$= \frac{1}{4 \cdot 0.36}\left[\left(0.4 - 0.5\right)^2 + \left(-0.5 - 0.6\right)^2 \right] = 0.8472;$$

$$c_{33}^{(e)} = \frac{1}{4S}\left[\left(y_1 - y_2\right)^2 + \left(x_2 - x_1\right)^2 \right].$$

6.2.3 Ансамблирование

После рассмотрения типовых элементов следующим шагом является объединение отдельных элементов в конечно-элементную сетку в области решения. Данный процесс называется ансамблированием или просто сборкой. С математической точки зрения ансамблирование состоит в объединении матриц коэффициентов отдельных элементов в одну глобальную матрицу. Полная энергия совокупности многих элементов в общем случае равна сумме энергий отдельных элементов:

$$W = \sum_{e=1}^{N} W_e = \frac{1}{2}\varepsilon \mathbf{\Phi}^t \mathbf{C} \mathbf{\Phi}, \tag{6.32}$$

где

$$\mathbf{\Phi} = \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \dots \\ \Phi_n \end{pmatrix}; \tag{6.33}$$

n – общее число узлов; N – общее число КЭ. Матрица **C** является глобальной и состоит из матриц коэффициентов отдельных элементов.

При получении уравнения (6.32) предполагалось, что вся расчетная область Ω однородна, т. е. величина ε постоянна. Дискретизация неоднородной области должна выполняться так, чтобы каждый из конечных элементов был однородным (рисунок 6.7). При этом в уравнении (6.22) от элемента к элементу меняется значение $\varepsilon = \varepsilon_r \varepsilon_0$.

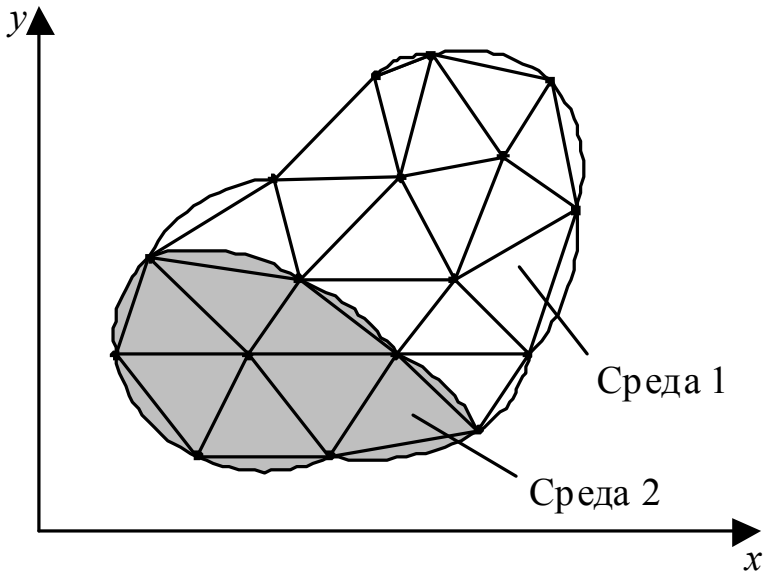


Рисунок 6.7 – Дискретизация неоднородной области решения

Проиллюстрируем процесс ансамблирования. Рассмотрим сетку, состоящую из трех КЭ (рисунок 6.8). Особого внимания заслуживает нумерация узлов сетки. Так, нумерация узлов 1–5

называется глобальной, тогда как нумерация 1-2-3 внутри треугольников называется локальной. Она соответствует нумерации узлов элемента на рисунке 6.4. Например, для элемента 3 на рисунке 6.8 глобальная нумерация 3-5-4 соответствует локальной нумерации 1-2-3, поскольку локальная нумерация должна выполняться последовательно против часовой стрелки, начиная с любого узла элемента. Для этого элемента можно выбрать и нумерацию 4-3-5 вместо 3-5-4, что также соответствует нумерации 1-2-3 для элемента на рисунке 6.4.

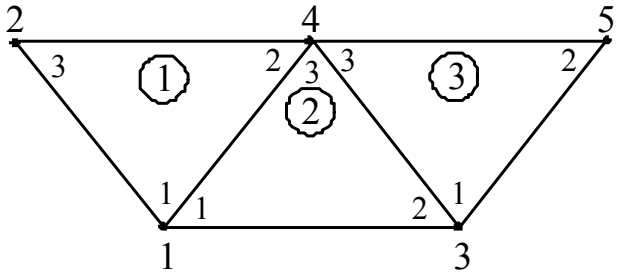


Рисунок 6.8 – Ансамблирование трех элементов

Нумерация на рисунке 6.8 не является уникальной, но важно то, что какая бы нумерация не использовалась, глобальная матрица коэффициентов остается неизменной. Так, задавшись определенной нумерацией, получим матрицу

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} \\ c_{31} & c_{32} & c_{33} & c_{34} & c_{35} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} \\ c_{51} & c_{52} & c_{53} & c_{54} & c_{55} \end{bmatrix}, \qquad (6.34)$$

которая имеет размер 5×5 по числу глобальных узлов ($n = 5$). Значения элементов c_{ij} вычисляются исходя из того, что распределение потенциала должно быть непрерывным на смежных границах КЭ. Вклад в значение элемента c_{ij} вносят все элементы, содержащие узлы i и j. Например, элементы 1 и 2 содержат общий глобальный узел 1, поэтому

$$c_{11} = c_{11}^{(1)} + c_{11}^{(2)}. \qquad (6.35)$$

Узел 2 соответствует только элементу 1, тогда

$$c_{22} = c_{33}^{(1)}, \tag{6.36}$$

а узел 4 – элементам 1, 2 и 3, а значит

$$c_{44} = c_{22}^{(1)} + c_{33}^{(2)} + c_{33}^{(3)}. \tag{6.37}$$

Узлы 1 и 4 принадлежат элементам 1 и 2, поэтому

$$c_{14} = c_{41} = c_{12}^{(1)} + c_{13}^{(2)}, \tag{6.38}$$

а поскольку между узлами 2 и 3 нет прямой связи, то

$$c_{23} = c_{32} = 0. \tag{6.39}$$

Аналогично вычисляются остальные элементы глобальной матрицы и в результате она принимает вид

$$\mathbf{C} = \begin{bmatrix} c_{11}^{(1)} + c_{11}^{(2)} & c_{13}^{(1)} & c_{12}^{(2)} & c_{12}^{(1)} + c_{13}^{(2)} & 0 \\ c_{31}^{(1)} & c_{33}^{(1)} & 0 & c_{32}^{(1)} & 0 \\ c_{21}^{(2)} & 0 & c_{22}^{(2)} + c_{11}^{(3)} & c_{23}^{(2)} + c_{13}^{(3)} & c_{12}^{(3)} \\ c_{21}^{(1)} + c_{31}^{(2)} & c_{23}^{(1)} & c_{32}^{(2)} + c_{31}^{(3)} & c_{22}^{(1)} + c_{33}^{(2)} + c_{33}^{(3)} & c_{32}^{(3)} \\ 0 & 0 & c_{21}^{(3)} & c_{23}^{(3)} & c_{22}^{(3)} \end{bmatrix}. \tag{6.40}$$

Таким образом, значения матрицы коэффициентов суммируются в узлах, являющихся общими для разных элементов (в рассматриваемом случае 1, 4 и 3, 4). Видно, что глобальная матрица содержит 27 ненулевых элементов (по 9 от каждого из трех элементов). Также она обладает следующими свойствами:

– она симметрична, как и матрицы коэффициентов элементов;

– поскольку есть элементы $c_{ij} = 0$ (между узлами i и j нет прямой связи), то при их большом числе она становится разреженной. Элементы матрицы также группируются (она становится ленточной или блочно-диагональной), если узлы надлежащим образом пронумерованы. Это можно показать, используя уравнения (6.25)–(6.30) и тот факт, что

$$\sum_{i=1}^{3} c_{ij}^{(e)} = \sum_{j=1}^{3} c_{ij}^{(e)} = 0;$$

– она сингулярная. Это можно показать, используя выражение (6.24).

6.2.4 Решение результирующего матричного уравнения

Используя понятия вариационного исчисления, можно показать, что уравнение Лапласа выполняется, когда полная энергия в области решения Ω минимальна. Таким образом, требуется, чтобы частные производные W по отношению к величине потенциала в каждом узле были равными нулю:

$$\frac{\partial W}{\partial \Phi_1} = \frac{\partial W}{\partial \Phi_2} = ... = \frac{\partial W}{\partial \Phi_n} = 0$$

или

$$\frac{\partial W}{\partial \Phi_k} = 0,\ k = 1,\ 2,\ ...,\ n. \qquad (6.41)$$

Например, получить $\partial W/\partial \Phi_1 = 0$ для сетки конечных элементов на рисунке 6.8 можно, подставив матрицу (6.34) в уравнение (6.32) и взяв частную производную W по Φ_1. Тогда

$$2\Phi_1 c_{11} + \Phi_2 c_{12} + \Phi_3 c_{13} + \Phi_4 c_{14} +$$

$$+ \Phi_5 c_{15} + \Phi_2 c_{21} + \Phi_3 c_{31} + \Phi_4 c_{41} + \Phi_5 c_{51} = 0,$$

что с учетом симметрии матрицы **C** дает

$$0 = \Phi_1 c_{11} + \Phi_2 c_{12} + \Phi_3 c_{13} + \Phi_4 c_{14} + \Phi_5 c_{15}. \qquad (6.42)$$

В общем виде при $\partial W/\partial \Phi_k = 0$ получим

$$0 = \sum_{i=1}^{n} \Phi_k c_{ik}, \qquad (6.43)$$

где n – число узлов сетки. Записав уравнение (6.43) для всех узлов $k = 1, 2, \ldots, n$, получим СЛАУ, из которой найдем решение $\mathbf{\Phi}^T = [\Phi_1, \Phi_2, \ldots, \Phi_n]$. Для этого можно воспользоваться теми же методами, которые применяются при решении конечно-разностных уравнений.

Сначала используем итерационный метод. Например, предположим, что узел 1 на рисунке 6.8 является свободным (значение потенциала в нем неизвестно). Из уравнения (6.42) получим

$$\Phi_1 = -\frac{1}{c_{11}} \sum_{i=2}^{5} \Phi_i c_{1i}. \qquad (6.44)$$

Таким образом, в общем случае для сетки, состоящей из n узлов, в k-м узле имеем

$$\Phi_k = -\frac{1}{c_{kk}} \sum_{i=1, i\neq k}^{n} \Phi_i c_{ki} \qquad (6.45)$$

где узел k является свободным. Если между узлами k и i нет прямой связи, то $c_{ki} = 0$. Следовательно, только узлы, которые имеют прямую связь с узлом k, вносят вклад в Φ_k согласно выражению (6.45). Уравнение (6.45) можно применить итерационно ко всем свободным узлам. Итерационный процесс начинается с задания значений потенциалов в граничных узлах. Хорошим начальным приближением для свободных узлов является их равенство нулю или среднему значению потенциала:

$$\Phi_{\text{ср}} = 0.5(\Phi_{\min} + \Phi_{\max}), \qquad (6.46)$$

где $\Phi_{\min}$ и $\Phi_{\max}$ – минимальное и максимальное значения потенциала в граничных узлах. При этих начальных значениях потенциалы в свободных узлах вычисляются с использованием выражения (6.45). В конце первой итерации, когда для всех свободных узлов вычислены новые значения, они становятся старыми значениями для второй итерации. Процедура повторяется до тех пор, пока разница между значениями, полученными на текущей и предыдущей итерациях, не будет достаточно малой (с требуемой точностью).

Теперь рассмотрим метод решения ленточных систем. Если все свободные узлы пронумерованы первыми, а граничные узлы – последними, то уравнение (6.22) может быть записано в виде

$$W = \frac{1}{2}\varepsilon \begin{bmatrix} \boldsymbol{\Phi}_f & \boldsymbol{\Phi}_p \end{bmatrix} \begin{bmatrix} \mathbf{C}_{ff} & \mathbf{C}_{fp} \\ \mathbf{C}_{pf} & \mathbf{C}_{pp} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_f \\ \boldsymbol{\Phi}_p \end{bmatrix}, \qquad (6.47)$$

где индексы f и p соответствуют узлам со свободными и граничными значениями потенциала. Поскольку матрица $\mathbf{\Phi}_p$ постоянна, то дифференцировать уравнение (6.47) согласно (6.41) нужно только по $\mathbf{\Phi}_f$. В результате получим

$$\frac{\partial W}{\mathbf{\Phi}_f} = \begin{bmatrix} \mathbf{C}_{ff} & \mathbf{C}_{fp} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_f \\ \mathbf{\Phi}_p \end{bmatrix} = 0$$

или

$$\mathbf{C}_{ff}\mathbf{\Phi}_f = -\mathbf{C}_{fp}\mathbf{\Phi}_p. \tag{6.48}$$

СЛАУ (6.48) запишем в матричном виде:

$$\mathbf{A\Phi} = \mathbf{B}, \tag{6.49}$$

где $\mathbf{\Phi} = \mathbf{\Phi}_f$; $\mathbf{A} = \mathbf{C}_{ff}$; $\mathbf{B} = -\mathbf{C}_{fp}\mathbf{\Phi}_p$. Так как матрица $\mathbf{A}$ является невырожденной, то потенциал в свободных узлах можно найти, используя выражение (6.49), с помощью подходящего метода решения СЛАУ.

Ранее при описании МКР было показано, что на границах области решения или на линиях симметрии (в случае их использования) иногда необходимо накладывать условие Неймана ($\partial\Phi/\partial n = 0$). Предположим, что область решения симметрична вдоль оси y (рисунок 6.9).

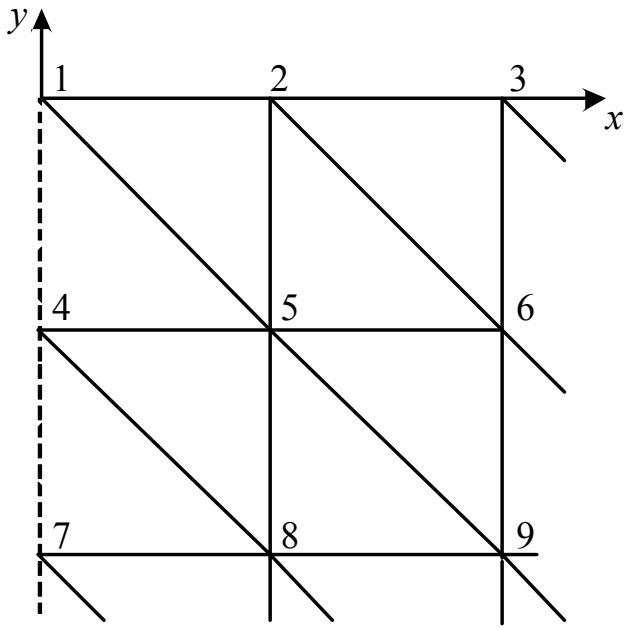


Рисунок 6.9 – Область решения, симметричная вдоль оси y

Тогда для выполнения условия Неймана необходимо положить

$$\Phi_1 = \Phi_2,\ \Phi_4 = \Phi_5,\ \Phi_7 = \Phi_8. \qquad (6.50)$$

Отметим, что согласно уравнению (6.18) решение ищется в двухмерной области и соответствует уравнению Лапласа $\nabla^2\Phi = 0$. Рассмотренные выше основные понятия распространяются также на конечно-элементный анализ задач, связанных с решением уравнения Пуассона и волнового уравнения.

Пример 6.1

Найти потенциалы внутри заданной сетки (рисунок 6.10), используя МКЭ.

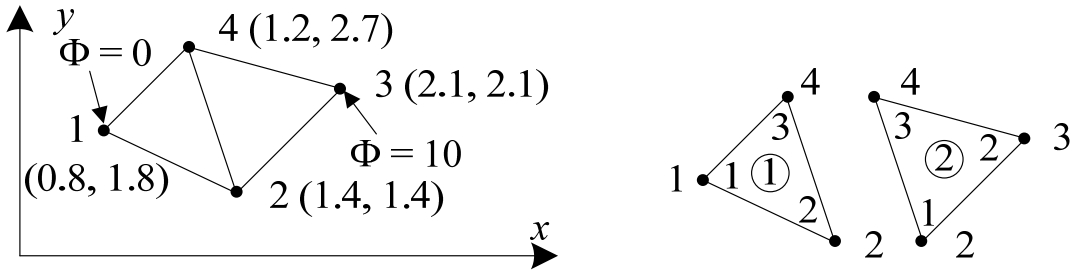


Рисунок 6.10 – Конечно-элементная сетка (*а*), локальная и глобальная нумерация ее узлов (*б*)

Решение

Матрицы коэффициентов для элементов можно вычислить с использованием уравнений (6.25)–(6.31). Однако вычисления можно упростить, если положить

$$\begin{aligned} P_1 &= (y_2 - y_3),\ P_2 = (y_3 - y_1),\ P_3 = (y_1 - y_2), \\ Q_1 &= (x_3 - x_2),\ Q_2 = (x_1 - x_3),\ Q_3 = (x_2 - x_1), \end{aligned} \qquad (6.51)$$

где P_i и Q_i $(i = 1, 2, 3)$ – номера локальных узлов. Тогда каждый элемент матрицы вычисляется с помощью выражения

$$c_{ij}^{(e)} = \frac{1}{4S}\left(P_i P_j + Q_i Q_j\right), \qquad (6.52)$$

где $S = 0.5(P_2 Q_3 - P_3 Q_2)$. Формула (6.52) более удобна для использования по сравнению с уравнениями (6.25)–(6.31). Применив

выражение (6.52) для элемента 1, содержащего глобальные узлы 1-2-4, соответствующие локальным узлам 1-2-3 (рисунок 6.10, *б*), получим

$$P_1 = (1.4 - 2.7) = -1.3;\ P_2 = (2.7 - 1.8) = 0.9;\ P_3 = (1.8 - 1.4) = 0.4;$$

$$Q_1 = (1.2 - 1.4) = -0.2;\ Q_2 = (0.8 - 1.2) = -0.4;$$

$$Q_3 = (1.4 - 0.8) = 0.6;\ S = 0.5(0.54 + 0.16) = 0.35.$$

Подстановка вычисленных значений в формулу (6.52) дает

$$\mathbf{C}^{(1)} = \begin{bmatrix} 1.2357 & -0.7786 & -0.4571 \\ -0.7786 & 0.6929 & 0.0857 \\ -0.4571 & 0.0857 & 0.3714 \end{bmatrix}. \qquad (6.53)$$

Аналогично для элемента 2, содержащего глобальные узлы 2-3-4, соответствующие локальным узлам 1-2-3 (см. рисунок 6.10, *б*), получим

$$P_1 = -0.6;\ P_2 = 1.3;\ P_3 = -0.7;$$

$$Q_1 = -0.9;\ Q_2 = 0{,}2;\ Q_3 = 0.7;$$

$$S = 0.5(0.91 + 0.14) = 0.525.$$

Тогда

$$\mathbf{C}^{(2)} = \begin{bmatrix} 0.5571 & -0.4571 & -0.1 \\ -0.4571 & 0.8238 & -0.3667 \\ -0.1 & -0.3667 & 0.4667 \end{bmatrix}. \qquad (6.54)$$

Важно, что при вычислении элементов матрицы отдельного КЭ необходимо придерживаться локальной нумерации, а глобальную нумерацию следует использовать только при формировании глобальной матрицы. В результате вычислим элементы глобальной матрицы:

$$\mathbf{C}_{22} = \mathbf{C}_{22}^{(1)} + \mathbf{C}_{11}^{(2)} = 0.6929 + 0.5571 = 1.25;$$

$$\mathbf{C}_{24} = \mathbf{C}_{23}^{(1)} + \mathbf{C}_{13}^{(2)} = 0.0857 - 0.1 = -0.0143;$$

$$\mathbf{C}_{44} = \mathbf{C}_{33}^{(1)} + \mathbf{C}_{33}^{(2)} = 0.3714 + 0.4667 = 0.8381;$$

$$\mathbf{C}_{21} = \mathbf{C}_{21}^{(1)} = -0.7786;$$

$$\mathbf{C}_{23} = \mathbf{C}_{12}^{(2)} = -0.4571;$$

$$\mathbf{C}_{41} = \mathbf{C}_{31}^{(1)} = -0.4571$$

$$\mathbf{C}_{43} = \mathbf{C}_{32}^{(2)} = -0.3667.$$

Теперь найдем глобальную матрицу:

$$\mathbf{C} = \begin{pmatrix} c_{11}^{(1)} & c_{12}^{(1)} & 0 & c_{13}^{(1)} \\ c_{21}^{(1)} & c_{22}^{(1)} + c_{11}^{(2)} & c_{12}^{(2)} & c_{23}^{(1)} + c_{12}^{(2)} \\ 0 & c_{21}^{(2)} & c_{22}^{(2)} & c_{23}^{(2)} \\ c_{31}^{(1)} & c_{32}^{(1)} + c_{31}^{(2)} & c_{32}^{(2)} & c_{33}^{(1)} + c_{33}^{(2)} \end{pmatrix} =$$

$$= \begin{pmatrix} 1.2357 & -0.7786 & 0 & -0.4571 \\ -0.7786 & 1.25 & -0.4571 & -0.0143 \\ 0 & -0.4571 & 0.8238 & -0.3667 \\ -0.4571 & -0.0143 & -0.3667 & 0.8381 \end{pmatrix}. \qquad (6.55)$$

Видно, что $\sum_{i=1}^{4} c_{ij} = \sum_{j=1}^{4} c_{ij} = 0$, это говорит о правильности вычисления элементов матрицы. Применив уравнение (6.45) к свободным узлам 2 и 4, получим

$$\Phi_2 = -\frac{1}{c_{22}}\left(\Phi_1 c_{12} + \Phi_3 c_{32} + \Phi_4 c_{42}\right) = -\frac{1}{1,25}\left(-4,571 - 0,0143\Phi_4\right),$$

$$\Phi_4 = -\frac{1}{c_{44}}\left(\Phi_1 c_{14} + \Phi_2 c_{24} + \Phi_3 c_{34}\right) = -\frac{1}{0,8381}\left(-0,143\Phi_2 - 3,667\right). \qquad (6.56)$$

Используя нулевое начальное приближение и выражения (6.56), найдем после первой итерации $\Phi_2 = 3.6568$, $\Phi_4 = 4.4378$, а после второй – $\Phi_2 = 3.7075$, $\Phi_4 = 4.4386$.

Итерационная схема вычисления обычно имеет быструю сходимость и предпочтительна при большом числе узлов. Как только значения потенциалов в узлах становятся известны, потенциал в любой точке сетки можно определить с помощью уравнения (6.11).

Пример 6.2

Решить уравнение Лапласа для двухмерной области, изображенной на рисунке 6.11, *а*.

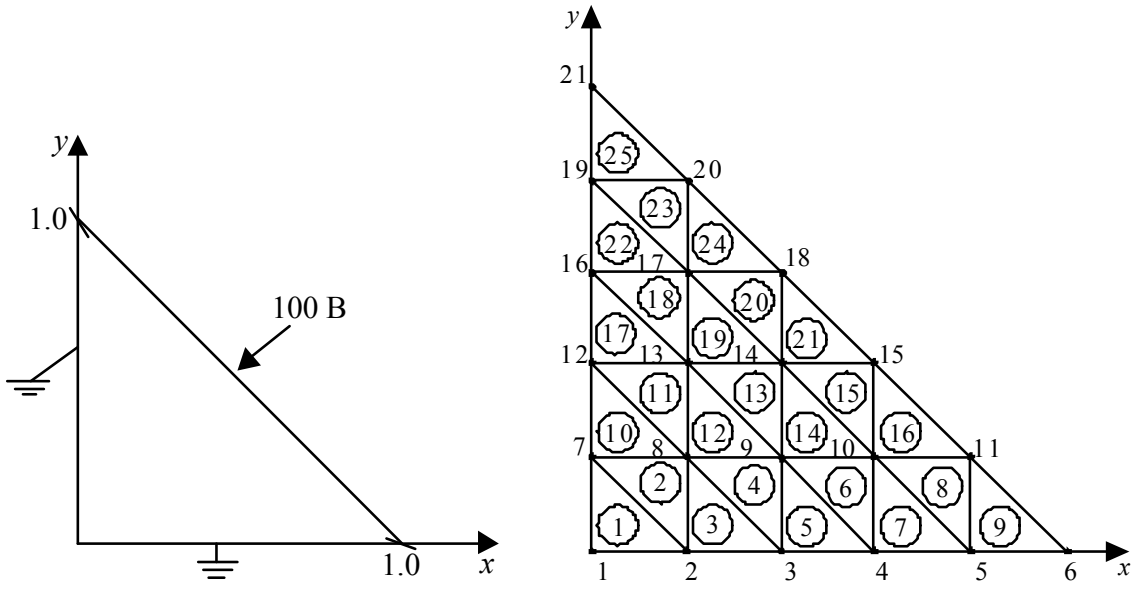


Рисунок 6.11 – Двухмерная область решения (*а*)
и конечно-элементная сетка (*б*)

Решение

На этапе 1 необходимо определить начальные данные для выполнения вычислений. Отметим, что это единственный этап, который зависит от геометрии задачи. Здесь определяется число конечных элементов, узлов, граничных узлов, значения потенциала в свободных узлах, координаты x и y всех узлов и список, идентифицирующий узлы, принадлежащие каждому элементу в порядке локальной нумерации 1-2-3. Для решения задачи разделим расчетную область на 25 треугольных КЭ с числом глобальных узлов 21 (рисунок 6.11, *б*). В результате получим три набора данных: координаты, отношения элемент-узел и заданные потенциалы на граничных узлах (таблицы 6.1–6.3).

На этапе 2 в соответствии с п. 6.2.2 и п. 6.2.3 выполняется вычисление значений элементов матриц $\mathbf{C}^{(e)}$ для каждого КЭ, а также их ансамблирование в глобальную матрицу $\mathbf{C}$.

На этапе 3 формируется список свободных узлов с использованием списка значений потенциалов в граничных узлах. Далее ко

всем свободным узлам итерационно применяется уравнение (6.45). В данном примере для сходимости требуется порядка 50 итераций, поскольку задействовано только 6 узлов.

На этапе 4 выполняется вывод результатов вычислений (таблица 6.4).

Таблица 6.1 – Координаты узлов (рисунок 6.11)

Узел	*x*	*y*	Узел	*x*	*y*
1	0	0	12	0	0.4
2	0.2	0	13	0.2	0.4
3	0.4	0	14	0.4	0.4
4	0.6	0	15	0.6	0.4
5	0.8	0	16	0	0.6
6	1	0	17	0.2	0.6
7	0	0.2	18	0.4	0.6
8	0.2	0.2	19	0	0.8
9	0.4	0.2	20	0.2	0.8
10	0.6	0.2	21	0	1
11	0.8	0.2			

Таблица 6.2 – Соответствие между элементами и узлами (рисунок 6.11)

Элемент	Локальный	Узел	Номер	Элемент	Локальный	Узел	Номер
	1	2	3		1	2	3
1	1	2	7	14	9	10	14
2	2	8	7	15	10	15	14
3	2	3	8	16	10	11	15
4	3	9	8	17	12	13	16
5	3	4	9	18	13	17	16
6	4	10	9	19	13	14	17
7	4	5	10	20	14	18	17
8	5	11	10	21	14	15	18
9	5	6	11	22	16	17	19
10	7	8	12	23	17	20	19
11	8	13	12	24	17	18	20
12	8	9	13	25	19	20	21
13	9	14	13				

Таблица 6.3 – Значения потенциала в граничных узлах (рисунок 6.11)

Узел	Значение потенциала	Узел	Значение потенциала
1	0	18	100
2	0	20	100
3	0	21	50
4	0	19	0
5	0	16	0
6	50	12	0
11	100	7	0
15	100		

Таблица 6.4 – Выходные данные для задачи из примера 6.2 (число узлов 21, число КЭ 25, граничных узлов 15)

Узел	*X*	*Y*	Потенциал
1	0	0	0
2	0.2	0	0
3	0.4	0	0
4	0.6	0	0
5	0.8	0	0
6	1	0	50
7	0	0.2	0
8	0.2	0.2	18.182
9	0.4	0.2	36.364
10	0.6	0.2	59.091
11	0.8	0.2	100
12	0	0.4	0
13	0.2	0.4	36.364
14	0.4	0.4	68.182
15	0.6	0.4	100
16	0	0.6	0
17	0.2	0.6	59.091
18	0.4	0.6	100
19	0	0.8	0
20	0.2	0.8	100
21	0	1	50

При использовании МКР получены следующие значения: $\Phi_8 = 15.41$; $\Phi_9 = 26.74$; $\Phi_{10} = 56.69$; $\Phi_{13} = 34.88$; $\Phi_{14} = 65.41$;

$\Phi_{17} = 58.72$ [32]. Несмотря на то что эти значения достаточно точные, в данной задаче можно добиться повышения точности методом конечных элементов путем деления области решения на большее число КЭ или за счет использования элементов более высокого порядка. Как упоминалось ранее, МКЭ имеет два основных преимущества по сравнению с МКР. Во-первых, при использовании МКР значения потенциала могут быть получены только в узлах сетки области решения, а при использовании МКЭ – в любой точке области решения. Во-вторых, МКЭ лучше подходит для решения задач со сложной геометрией.

6.3 Решение уравнения Пуассона

Рассмотрим особенности решения двухмерного уравнения Пуассона

$$\nabla^2 \Phi = -\frac{\rho}{\varepsilon}. \tag{6.57}$$

При его решении выполняются те же этапы, что и при решении уравнения Лапласа. Главное отличие заключается только в необходимости учета члена в правой части уравнения. Поэтому отметим лишь основные различия в этапах при решении этих уравнений.

После того как область решения разбита на треугольные КЭ, необходимо аппроксимировать распределение потенциала $\Phi_e(x, y)$ и поверхностной плотности заряда ρ_e линейными комбинациями интерполяционного многочлена α_i на каждом КЭ, т. е.

$$\Phi_e = \sum_{i=1}^{3} \Phi_{ei} \alpha_i(x, y), \tag{6.58}$$

$$\rho_e = \sum_{i=1}^{3} \rho_{ei} \alpha_i(x, y). \tag{6.59}$$

Коэффициенты Φ_{ei} и ρ_{ei} соответствуют значениям в i-м узле e-го элемента. Значения ρ_{ei} известны, поскольку $\rho(x, y)$ известно, а значения Φ_{ei} должны быть вычислены.

Функционал, описывающийся уравнением (6.57), ассоциируется с уравнением Эйлера

$$F(\Phi_e) = \frac{1}{2}\int_{\Omega}\left(\varepsilon \mid \nabla\Phi_e \mid^2 -2\rho_e\Phi_e\right)d\Omega, \tag{6.60}$$

где $F(\Phi_e)$ представляет собой погонную энергию, запасенную внутри e-го элемента. Первый член под знаком интеграла $\frac{1}{2}\mathbf{D}\cdot\mathbf{E} = \frac{1}{2}\varepsilon|\nabla\Phi_e|^2$ – это плотность энергии в электростатической системе, а второй член $\rho_e\Phi_e d\Omega$ – работа, совершаемая при перемещении заряда $\rho_e d\Omega$ в место, где находится потенциал Φ_e. После подстановки выражений (6.58) и (6.59) в уравнение (6.60) получим

$$F(\Phi_e) = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\varepsilon\Phi_{ei}\left[\int\nabla\alpha_i\cdot\nabla\alpha_j d\Omega\right]\Phi_{ej} - \sum_{i=1}^{3}\sum_{j=1}^{3}\Phi_{ei}\left[\int\alpha_i\alpha_j d\Omega\right]\rho_{ej}$$

или в матричной форме

$$F(\Phi_e) = \frac{1}{2}\varepsilon\mathbf{\Phi}_e^T\mathbf{C}^{(e)}\mathbf{\Phi}_e - \mathbf{\Phi}_e^T\mathbf{T}^{(e)}\boldsymbol{\rho}_e, \tag{6.61}$$

где

$$c_{ij}^{(e)} = \int\nabla\alpha_i\cdot\nabla\alpha_j d\Omega, \tag{6.62}$$

что аналогично уравнению (6.25), а

$$t_{ij}^{(e)} = \int\alpha_i\alpha_j d\Omega. \tag{6.63}$$

Можно показать, что

$$t_{ij}^{(e)} = \begin{cases} S/12, & \text{если} \quad i \neq j, \\ S/6, & \text{если} \quad i = j, \end{cases} \tag{6.64}$$

где S – площадь треугольного элемента.

Уравнение (6.61) может быть применено к каждому элементу в области решения. Тогда получим дискретный аналог функционала для всей области решения (с N элементами и n узлами) в виде суммы функционалов для отдельных элементов:

$$F(\Phi) = \sum_{e=1}^{N}F(\Phi_e) = \frac{1}{2}\varepsilon\mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi} - \mathbf{\Phi}^T\mathbf{T}\boldsymbol{\rho}. \tag{6.65}$$

В данном уравнении матрица-строка **Φ** состоит из значений Φ_{ei}, а матрица-строка **ρ** – из n значений исходной функции ρ в узлах сетки.

Результирующие уравнения могут быть решены итерационно или с помощью метода для ленточных матриц, как показано в п. 6.2.4. Далее используем итерационный подход. Рассмотрим область решений на рисунке 6.8, имеющую пять узлов (n = 5). С помощью уравнения (6.65) получим

$$F(\Phi)=\frac{1}{2}\varepsilon\begin{pmatrix}\Phi_1 & \Phi_2 & \dots & \Phi_5\end{pmatrix}\begin{pmatrix}c_{11} & c_{12} & \dots & c_{15}\\ c_{21} & c_{22} & \dots & c_{25}\\ \dots & \dots & \dots & \dots\\ c_{51} & c_{52} & \dots & c_{55}\end{pmatrix}\begin{pmatrix}\Phi_1\\ \Phi_2\\ \dots\\ \Phi_5\end{pmatrix}-$$

$$-\begin{pmatrix}\Phi_1 & \Phi_2 & \dots & \Phi_5\end{pmatrix}\begin{pmatrix}t_{11} & t_{12} & \dots & t_{15}\\ t_{21} & t_{22} & \dots & t_{25}\\ \dots & \dots & \dots & \dots\\ t_{51} & t_{52} & \dots & t_{55}\end{pmatrix}\begin{pmatrix}\rho_1\\ \rho_2\\ \dots\\ \rho_5\end{pmatrix}. \qquad (6.66)$$

Минимизируем энергию:

$$\frac{\partial F}{\partial \Phi_k}=0,\quad k=1,2,...,n. \qquad (6.67)$$

Для $\frac{\partial F}{\partial \Phi_1}=0$ из уравнения (6.66) найдем

$$\frac{\partial F}{\partial \Phi_1}=\varepsilon\left[\Phi_1 c_{11}+\Phi_2 c_{21}+...+\Phi_5 c_{51}\right]-\left[t_{11}\rho_1+t_{12}\rho_2+...+t_{51}\rho_5\right]=0,$$

$$\Phi_1=-\frac{1}{c_{11}}\sum_{i=2}^{5}\Phi_i c_{i1}+\frac{1}{\varepsilon c_{11}}\sum_{i=1}^{5}t_{i1}\rho_i. \qquad (6.68)$$

Таким образом, в общем случае для сетки, содержащей n узлов, имеем

$$\Phi_k=-\frac{1}{c_{kk}}\sum_{i=1,i\neq k}^{n}\Phi_i c_{ik}+\frac{1}{\varepsilon c_{kk}}\sum_{i=1}^{n}t_{ik}\rho_i, \qquad (6.69)$$

где k считается свободным узлом.

Для получения решения, как и при решении уравнения Лапласа, необходимо итерационно применить уравнение (6.69) ко всем свободным узлам до достижения сходимости, зафиксировав потенциалы в граничных узлах и первоначально установив потенциалы в свободных узлах (например, равными нулю).

Теперь рассмотрим процесс решения на основе системы с ленточной матрицей. Для этого необходимо пронумеровать сначала свободные узлы, а затем граничные. Тогда уравнение (6.65) можно записать в виде

$$F(\Phi)=\frac{1}{2}\varepsilon\begin{bmatrix}\boldsymbol{\Phi}_f & \boldsymbol{\Phi}_p\end{bmatrix}\begin{bmatrix}\mathbf{C}_{ff} & \mathbf{C}_{fp}\\ \mathbf{C}_{pf} & \mathbf{C}_{pp}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_f\\ \boldsymbol{\Phi}_p\end{bmatrix}-$$

$$-\begin{bmatrix}\boldsymbol{\Phi}_f & \boldsymbol{\Phi}_p\end{bmatrix}\begin{bmatrix}\mathbf{T}_{ff} & \mathbf{T}_{fp}\\ \mathbf{T}_{pf} & \mathbf{T}_{pp}\end{bmatrix}\begin{bmatrix}\boldsymbol{\rho}_f\\ \boldsymbol{\rho}_p\end{bmatrix}. \qquad (6.70)$$

Минимизируя $F(\Phi)$ по Φ_f, получим

$$\frac{\partial F(\Phi)}{\partial \boldsymbol{\Phi}_f}=0=\varepsilon\left(\mathbf{C}_{ff}\boldsymbol{\Phi}_f+\mathbf{C}_{fp}\boldsymbol{\Phi}_p\right)-\left(\mathbf{T}_{ff}\boldsymbol{\rho}_f+\mathbf{T}_{fp}\boldsymbol{\rho}_p\right)$$

или

$$\mathbf{C}_{ff}\boldsymbol{\Phi}_f=-\mathbf{C}_{fp}\boldsymbol{\Phi}_p+\frac{1}{\varepsilon}\mathbf{T}_{ff}\boldsymbol{\rho}_f+\frac{1}{\varepsilon}\mathbf{T}_{fp}\boldsymbol{\rho}_p. \qquad (6.71)$$

Данное выражение в матричной форме имеет вид

$$\mathbf{A\Phi}=\mathbf{B},$$

где $\mathbf{A}=\mathbf{C}_{ff}$, $\boldsymbol{\Phi}=\boldsymbol{\Phi}_f$; $\mathbf{B}$ – правая часть выражения (6.71). Полученную СЛАУ можно решить относительно $\boldsymbol{\Phi}$ любым подходящим методом. Следует отметить, что различия между уравнениями (6.45), (6.48) и (6.69), (6.71) незначительны.

6.4 Решение уравнения Гельмгольца

Для демонстрации универсальности МКЭ рассмотрим решение одномерного уравнения Гельмгольца

$$-\frac{d}{dx}\left(\alpha\frac{df}{dx}\right)+\beta f=s,\ \ a<x<b, \qquad (6.72)$$

$$f(a) = f_a, \tag{6.73}$$

$$f(b) = f_b, \tag{6.74}$$

где функция $f(x)$ – искомое решение; $\alpha = \alpha(x)$ и $\beta = \beta(x)$ – функции, определяющие свойства материала; $s = s(x)$ – параметры источника.

Найдем функцию $f(x)$ на интервале $a < x < b$. Для ясности положим $a = -2$, $b = 5$ и разделим ось x на 7 одинаковых подынтервалов. Пронумеровав узлы подынтервала, получим их координаты $x_i = i - 3$, $i = 1, 2, \ldots, 8$. Далее, введем узловые кусочно-линейные базисные функции $\varphi_i(x)$, которые линейны на каждом подынтервале. Они равны единицам в i-х узлах и нулям в остальных узлах (рисунок 6.12).

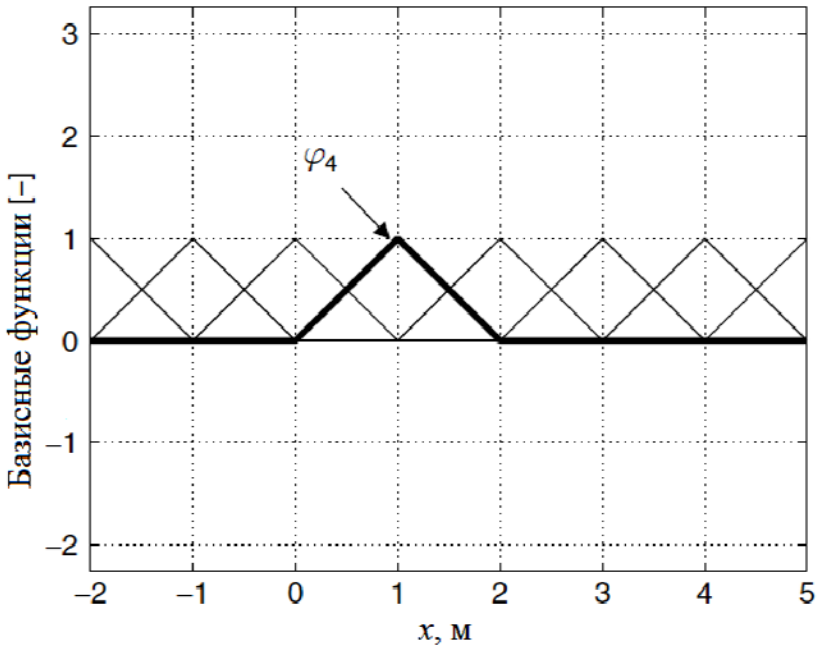


Рисунок 6.12 – Одномерные кусочно-линейные базисные функции

Будем искать приближенное решение с помощью разложения по базисным функциям:

$$f(x) = \sum_{i=1}^{8} f_i \varphi_i(x). \tag{6.75}$$

Заметим, что $f(x_i) = f_i$. Так как $f(a) = f_a$ и $f(b) = f_b$ известны, то положим $f_1 = f_a$ и $f_8 = f_b$.

Следуя методу Галёркина, выберем тестовые функции $w_i(x) = \varphi_i(x)$, где $i = 2, 3, \ldots, 7$ (конечные точки исключены, поскольку соответствующие значения функции известны). Умножим невязку уравнения (6.72) на тестовые функции $w_i(x)$ и результат проинтегрируем от $x = a$ до $x = b$. Будем интегрировать по частям, чтобы освободиться от одной из производных от f по w_i. В результате получим слабую форму (weak form) записи исходной задачи, которая представляет собой взвешенное значение невязки:

$$\int_a^b \left(\alpha w_i' f' + \beta w_i f - \omega_i s\right) ds = 0. \qquad (6.76)$$

Тогда слагаемое $\alpha w_i' f'$ обнуляется, поскольку $w_i(a) = w_i(b) = 0$.

Подставив выражение (6.75) в (6.5) и выбрав $w_2(x) = \varphi_2(x)$, получим уравнение, содержащее 6 неизвестных коэффициентов f_j для внутренних узлов x_j, $j = 2, 3, \ldots, 7$. Аналогично, выбрав $w_3(x) = \varphi_3(x)$, получим второе уравнение с шестью неизвестными и т. д. В результате получим СЛАУ вида **Az** = **b**, где

$$a_{ij} = \int_a^b \left(\alpha \varphi_i \varphi_j' + \beta \varphi_i \varphi_j\right) dx, \qquad (6.77)$$

$$z_j = f_j, \qquad (6.78)$$

$$b_i = \int_a^b \varphi_i s dx. \qquad (6.79)$$

Здесь $i = 2, 3, \ldots, 7$ (по числу уравнений); $j = 1, 2, \ldots, 8$ (по числу коэффициентов). Матрица **A** состоит из 6 строк и 8 столбцов, а векторы **z** и **b** – из 8 и 6 строк соответственно. Значения коэффициентов f_1 и f_8 известны благодаря граничным условиям, поэтому слагаемые, содержащие их, могут быть перенесены в правую часть СЛАУ. Тогда

$$\begin{pmatrix} a_{22} & a_{23} & \ldots & a_{27} \\ a_{32} & a_{33} & \cdots & a_{37} \\ \vdots & \vdots & \ddots & \vdots \\ a_{72} & a_{73} & \cdots & a_{77} \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ \vdots \\ f_7 \end{pmatrix} = \begin{pmatrix} b_2 \\ b_3 \\ \vdots \\ b_7 \end{pmatrix} - \begin{pmatrix} a_{21} f_1 + a_{28} f_8 \\ a_{31} f_1 + a_{38} f_8 \\ \vdots \\ a_{71} f_1 + a_{78} f_8 \end{pmatrix}.$$

Поскольку значения функции в конечных точках известны, можно не использовать соответствующие весовые функции (что и было сделано). При этом матрица СЛАУ является квадратной (число уравнений совпадает с числом неизвестных), разреженной (поскольку использованные базисные функции описывают только связь ближайших элементов) и симметричной (поскольку оператор Гельмгольца является самосопряженным и использован метод Галёркина).

Граничные условия (6.73) и (6.74) задают значение функции $f(x)$ на границах. При других типах граничных условий может быть задана производная функции или линейная комбинация из функции и ее производной. На любой границе, например слева $x = a$, могут быть применены условия стандартных типов

$$f(a) = p \tag{6.80}$$

или

$$f'(a) + \gamma f(a) = q. \tag{6.81}$$

Уравнение (6.80) соответствует граничному условию Дирихле. При $\gamma = 0$ уравнение (6.81) – это граничное условие Неймана, а при $\gamma \neq 0$ – смешанное. В случае использования условий Неймана или смешанного функция $f(a)$ является дополнительной неизвестной. При этом формируется еще одно уравнение с помощью тестовой функции $w_1(x) = \varphi_1(x)$.

Теперь расширим модельную задачу (6.72) до двух измерений. Функция f, как и ранее, является скалярной:

$$-\nabla \cdot (\alpha \nabla f) + \beta f = s \text{ в области } \Omega, \tag{6.82}$$

$$f = p \text{ на границе } L_1, \tag{6.83}$$

$$\overline{\mathbf{n}} \cdot (\alpha \nabla f) + \gamma f = q \text{ на границе } L_2. \tag{6.84}$$

Граница области решения Ω имеет две части – L_1 и L_2, с разными заданными на них граничными условиями.

В качестве примера рассмотрим задачу вычисления сопротивления между левым и нижним краями проводящей пластины (рисунок 6.13). Тогда функция f – электростатический потенциал, α – проводимость, $\beta = 0$, $s = 0$. Потенциал вдоль границы, показанной на рисунке жирной линией, установим равным 10 В, т. е.

используется граничное условие Дирихле $f = 10$. Вдоль границы, показанной на рисунке жирной пунктирной линией, установим потенциал 0 В. Оставшуюся часть границы будем считать изоляцией. На ней используем граничное условие Неймана $\overline{\mathbf{n}} \cdot \nabla f = 0$, что означает равенство нулю величины потока через границу.

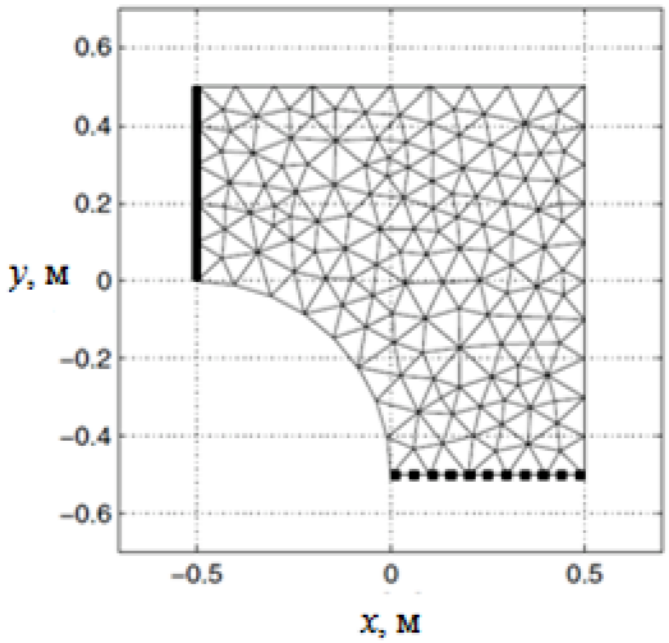


Рисунок 6.13 – Двухмерная проводящая пластина

Умножим уравнение (6.82) на тестовые функции w_i и выполним интегрирование по области Ω:

$$\int_{\Omega} w_i \left[-\nabla \cdot (\alpha \nabla f) + \beta f \right] d\Omega = \int_{\Omega} w_i s d\Omega.$$

Далее интегрируем по частям:

$$\nabla \cdot [w_i (\alpha \nabla f)] = \alpha \nabla w_i \cdot \nabla f + w_i \nabla \cdot (\alpha \nabla f). \qquad (6.85)$$

По теореме Гаусса

$$\int_{\Omega} \nabla \cdot F d\Omega = \int_{L_1 + L_2} \overline{\mathbf{n}} \cdot F dl,$$

где $F = w_i \alpha \nabla f$. В результате получим слабую форму записи для (6.82)

$$\int_{\Omega} \left(\alpha \nabla w_i \cdot \nabla f + \beta w_i f \right) d\Omega - \int_{L_2} w_i (q - \gamma f)\, dl = \int_{\Omega} w_i s d\Omega, \qquad (6.86)$$

где использовано граничное условие (6.84). Интеграл по той части границы, где решение известно (L_1), не учитывается, поскольку тестовые функции на ней обращаются в нуль. Отметим, что дифференциальное уравнение (6.86) включает в себя сведения о граничных условиях.

Узлы сетки пометим целыми числами i. Они расположены в точках r_i, $i = 1, 2, \ldots, N_n$. Конечные элементы представляют собой треугольники. Выберем КЛБФ $\varphi_i(r)$, которые линейны внутри каждого треугольника, т. е. $\varphi_i(r_i) = 1$, $\varphi_i(r_j) = 0$ при $i \neq j$. Существует по одной такой функции, связанной с каждым узлом. В качестве примера на рисунке 6.14 показаны две таких функции.

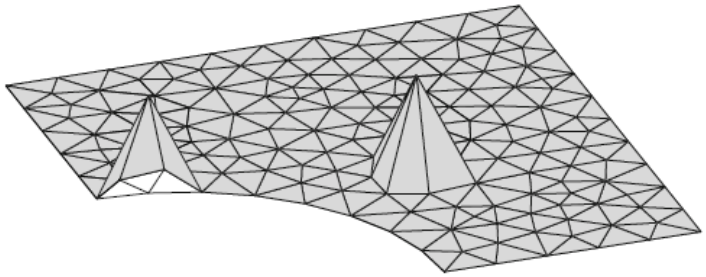


Рисунок 6.14 – Иллюстрация двух узловых базисных функций (одна на границе и одна внутри области решения)

Найдем приближенное решение $f(r)$ с помощью разложения по базисным функциям:

$$f(r) = \sum_{j=1}^{N_n} f_j \varphi_j(r). \tag{6.87}$$

Подставив выражение (6.87) в уравнение слабой формы (6.86) и используя метод Галёркина, т. е. $w_i(r) = \varphi_i(r)$, найдем неизвестные значения функции f в узлах. Это дает СЛАУ вида $\mathbf{Az} = \mathbf{b}$, элементы которой вычисляются как

$$a_{ij} = \int_{\Omega} \left(\alpha \nabla \varphi_i \cdot \nabla \varphi_j + \beta \varphi_i \varphi_j \right) d\Omega + \int_{L_2} \gamma \varphi_i \varphi_j dl, \tag{6.88}$$

$$z_j = f_j, \tag{6.89}$$

$$b_i = \int_{\Omega} \varphi_i s d\Omega + \int_{L_2} \varphi_i q dl. \tag{6.90}$$

Здесь индекс j нужен для обхода всех узлов, а i – только тех, в которых значение f неизвестно (кроме границы L_1, где задано условие Дирихле). Переупорядочим переменные, чтобы известным значениям f соответствовал вектор $\mathbf{z}_e$, а неизвестным – $\mathbf{z}_n$. Аналогично разбивается матрица $\mathbf{A}$. В результате СЛАУ преобразуется к виду

$$\begin{pmatrix}\mathbf{A}_e & \mathbf{A}_n\end{pmatrix}\begin{pmatrix}\mathbf{z}_e \\ \mathbf{z}_n\end{pmatrix} = \mathbf{A}_e\mathbf{z}_e + \mathbf{A}_n\mathbf{z}_n = \mathbf{b},$$

где значения $\mathbf{A}_n$ и $\mathbf{b} - \mathbf{A}_e\mathbf{z}_e$ полностью известны. Поэтому СЛАУ можно переписать в виде

$$\mathbf{A}_n\mathbf{z}_n = \mathbf{b} - \mathbf{A}_e\mathbf{z}_e.$$

Процедура нахождения неизвестных подробно описана в подразделе 6.3. В итоге процесс решения сведен к работе с матрицами и векторами, что более предпочтительно для двухмерных и трехмерных задач.

Теперь вернемся к примеру по вычислению сопротивления металлической пластины. Толщину пластины обозначим h. Результирующее распределение электростатического потенциала показано на рисунке 6.15. Зная это распределение, сопротивление пластины можно вычислить двумя способами.

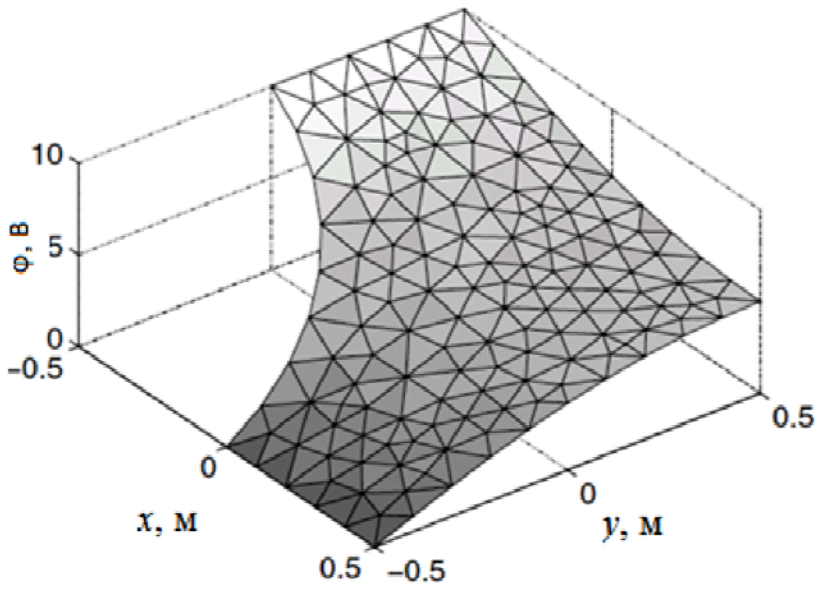


Рисунок 6.15 – Распределение потенциала
на проводящей пластине

Первый способ – интегрирование нормальной составляющей плотности тока $\mathbf{J} = -\sigma\nabla\Phi$ по поперечному сечению пластины для получения величины полного тока, протекающего через пластину:

$$I = \int_{z=0}^{h} \int_{x=0}^{0,5} \sigma \left. \frac{\partial \Phi}{\partial y} \right|_{y=-0.5} dx. \qquad (6.91)$$

После этого сопротивление вычисляется с помощью закона Ома $R = U/I$, где $U = \Delta\Phi = 10$ В.

Второй способ – вычисление общей рассеиваемой пластиной мощности (аналогично вычислению емкости)

$$P = \int_V \mathbf{J} \cdot \mathbf{E} dV = \int_V \sigma |\nabla\Phi|^2 dV = h\mathbf{z}^t \mathbf{A}\mathbf{z} = h\mathbf{z}^t \mathbf{b}, \qquad (6.92)$$

а затем сопротивления $R = U^2 / P$.

6.5 Особенности построения сетки

Этап подготовки данных для решения задач электростатики является одной из основных трудностей, возникающих при конечно-элементном анализе. Поэтому вычислительно эффективные программы должны иметь универсальные средства генерации сетки. Это не только сокращает время, затрачиваемое на подготовку данных, но и устраняет возможные ошибки, возникающие при ручной обработке. Комбинирование автоматической генерации сетки с компьютерной графикой особенно важно, так как результирующую сетку можно оценить визуально. Поскольку электромагнитные задачи часто основаны на простой прямоугольной геометрии области решения, то рассмотрим особенности создания сетки для прямоугольных областей.

Пусть прямоугольная область решения имеет размер $a \times b$ (рисунок 6.16). Задачей является разделение области на прямоугольные элементы, каждый из которых затем делится на два треугольных элемента. Пусть n_x и n_y – число делений по осям x и y соответственно. Тогда общее число элементов и узлов определяется как

$$n_e = 2n_x n_y, \; n_d = (n_x + 1)(n_y + 1). \qquad (6.93)$$

Для хранения глобальных координат (x, y) каждого узла требуется массив, содержащий значения Δx_i, $i = 1, 2, \ldots, n_x$ и Δy_j, $j = 1, 2, \ldots, n_y$, которые соответствуют расстояниям между узлами по осям x и y. Если выбрать порядок нумерации узлов слева направо и снизу вверх, то первый узел является началом координат (0, 0), следующий узел – $x + \Delta x_1$, $y = 0$, затем $x + \Delta x_2$, $y = 0$ и т. д., пока все Δx_i не будут пройдены. Переходя на вторую горизонтальную строку, необходимо повторить процесс, т. е. сначала $x = 0$ и $y + \Delta y_1$, затем инкрементировать x до тех пор, пока все Δx_i не будут пройдены. Процесс продолжается до тех пор, пока не будут получены координаты последнего узла $(n_x + 1)(n_y + 1)$.

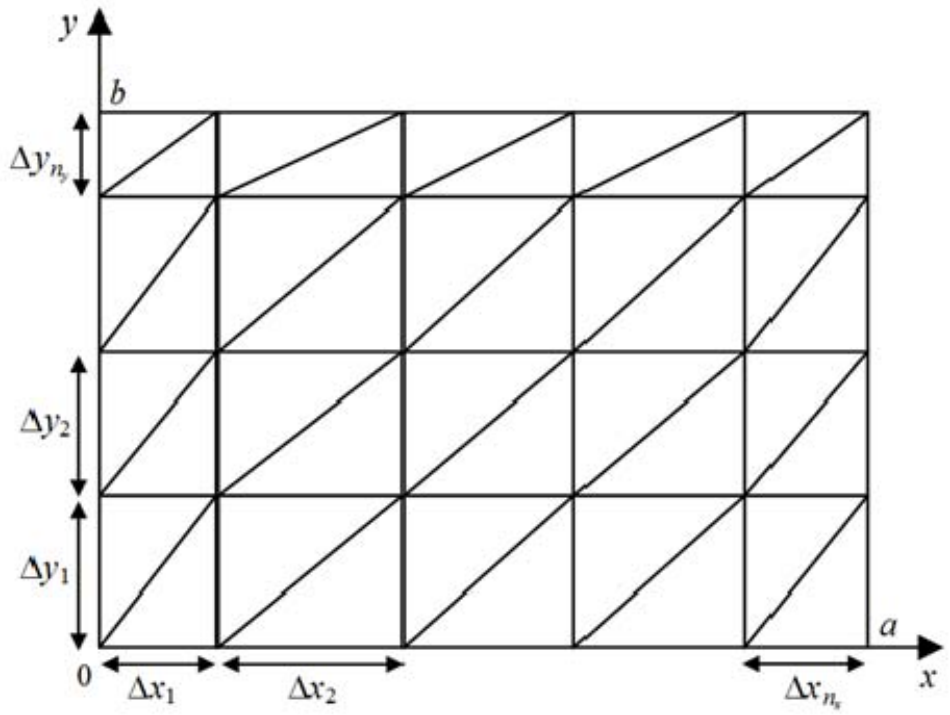


Рисунок 6.16 – Дискретизация прямоугольной области неравномерной сеткой

Представленная процедура позволяет генерировать равномерные и неравномерные сетки. Так, сетка считается равномерной, если все Δx_i равны между собой, а все Δy_j – между собой. В противном случае сетка считается неравномерной. Использование неравномерной сетки предпочтительней, если заранее известно, что интересующий параметр быстро меняется в некоторых частях об-

ласти решения. Это позволяет концентрировать малые элементы в областях, где данный параметр изменяется быстро, поскольку именно такие области часто представляют наибольший интерес для решения. При отсутствии указанных данных может быть использована равномерная сетка, для которой

$$\Delta x_1 = \Delta x_2 = \ldots = h_x,\ \Delta y_1 = \Delta y_2 = \ldots = h_y, \qquad (6.94)$$

где $h_x = a/n_x$; $h_x = b/n_y$.

Предположим, что на всей границе значение потенциала известно, тогда число граничных узлов оценивается как

$$n_p = 2(n_x + n_y). \qquad (6.95)$$

Простым способом формирования списка узлов является их перечисление последовательно на нижней, правой, верхней и левой (против часовой стрелки) границах прямоугольной области решения.

6.6 Математическая модель вычисления емкостной матрицы многопроводной линии передачи

Поскольку для вычисления погонных параметров линии передачи требуется решение уравнения Лапласа, то при вычислении емкостной матрицы линии передачи справедлива описанная в подразделе 6.2 последовательность действий. Так, область решения дискретизируется на трехузловые треугольники. Искомыми величинами являются потенциалы в свободных узлах. Данные узлы используются для вычисления полной энергии структуры и затем ее погонной емкости. Для одиночной линии емкость определяется по формуле

$$C = 2\frac{W_e}{\Phi^2}.$$

Для МПЛП полная энергия вычисляется столько раз, сколько имеется проводников N_{COND} в линии. В результате формируется емкостная матрица. Ее диагональные элементы вычисляются как

$$c_{ii} = 2W_{ii},\ \ i = 1, 2, \ldots, N_{\mathrm{COND}}, \qquad (6.96)$$

где W_{ii} – энергия, вычисленная при установлении потенциала 1 В на i-м проводнике и 0 В на остальных. Внедиагональные элементы емкостной матрицы вычисляются как

$$c_{ij} = 2W_{ii} - 0.5(c_{ii} + c_{jj}), \tag{6.97}$$

где W_{ii} – энергия, вычисленная при задании потенциала 1 В на i-м и j-м проводниках.

Контрольные вопросы и задания

1. Назовите этапы решения задач при использовании МКЭ.
2. В чем различие между комплекс-элементом и симплекс-элементом?
3. К матрице СЛАУ какого вида приводит использование МКЭ?
4. Опишите особенности вычисления емкостной матрицы МПЛП методом конечных элементов.
5. Для чего применяется ансамблирование в МКЭ?
6. Назовите особенности нумерации узлов конечно-элементной сетки.
7. Разработать программу на языке Octave для решения задачи из примера 6.2.
8. Разработать программу на языке Octave для вычисления методом конечных элементов емкости коаксиальной структуры, изображенной на рисунке 3.16, при $a = b = 1$ см, $c = d = 2$ см.

ЗАКЛЮЧЕНИЕ

Вследствие массового распространения различных технических средств в различных областях человеческой деятельности выполнение требований ЭМС становится все более сложным, а число аспектов, принимаемых во внимание, стремительно увеличивается. Поэтому проблема обеспечения ЭМС тесно связана с тем, что составляет обширную область радиотехники, электроники и электротехники. Для минимизации как финансовых затрат, так и затрат времени на проектирование технических средств с учетом требований ЭМС целесообразно использовать математическое моделирование, в основе которого лежат численные методы.

В данном пособии показана актуальность применения математического моделирования при решении задач ЭМС в части электростатики. Обсуждаются общие вопросы, связанные с интегральными и дифференциальными уравнениями, особенности использования численных методов, а также способы повышения точности вычислений и экономии машинных ресурсов. Приведены примеры решения тестовых задач. Помимо этого, подробно рассмотрены математические модели для анализа линий передачи (без потерь), широко используемые на практике при решении задач ЭМС. В первую очередь пособие предназначено для будущих специалистов в области радиоэлектроники и информационных технологий.

Автор не ставил перед собой цель всесторонне осветить все численные методы, применяемые при решении задач электростатики, а заострил внимание только на наиболее широко используемых: методах конечных разностей, моментов и конечных элементов. Приведенные примеры и программы на языке GNU Octave, а также задания нацелены на изучение особенностей этих методов. Тем не менее, остались нерассмотренными представляющие практический интерес методы матрицы линий передачи, эквивалентной схемы из частичных элементов, метод прямых, Монте-Карло и др. Однако, по мнению автора, математическая основа численных методов, рассмотренная в пособии, позволит заинтересованному читателю самостоятельно изучить особенности этих

методов. Следует заметить, что до сих пор не существует универсального численного метода, поэтому разработка новых и, по всей видимости, гибридных методов станет одним из направлений дальнейших исследований.

В основе упомянутых численных методов лежит замена непрерывных функций их дискретными аналогами, что часто сводит задачу к решению линейных алгебраических систем, поэтому значительная часть пособия посвящена рассмотрению методов их решения. Эта часть вычислительной линейной алгебры, несмотря на давнюю историю, остается бурно развивающейся и заслуживает пристального рассмотрения в виде отдельного и объемного пособия. Однако автор предпринял попытку кратко, но емко осветить все ее аспекты, чтобы читателю было легче в дальнейшем изучении этой удивительной и увлекательной проблемы – решение СЛАУ, которой он сам увлекся, еще обучаясь в университете. Необходимо отметить, что не существует универсального метода решения СЛАУ, обеспечивающего требуемую точность при минимальных затратах времени и машинной памяти. Поэтому поиски продолжаются. Так, одним из передовых направлений является аппроксимация матрицы СЛАУ матрицами меньшего ранга для сокращения объема требуемой машинной памяти, например с помощью адаптивной перекрестной аппроксимации.

Таким образом, актуальна разработка новых численных методов и читателю пособия найдется место для приложения своих усилий в этом направлении исследований.

Литература

1. Уилльямс Т. ЭМС для разработчиков / Т. Уилльямс. – М.: Технологии, 2003. – 540 с.
2. Неганов В. А. Электродинамические методы проектирования устройств СВЧ и антенн: учеб. пособие для вузов / В. А. Неганов, Е. И. Нефедов, Г. П. Яровой ; под ред. В. А. Неганова. – М.: Радио и связь, 2002. – 415 с.
3. Банков С. Е. История САПР СВЧ (1950–2010) / С. Е. Банков, А. А. Курушин. – LAP LAMBERT Academic Publishing, 2016. – 100 с.
4. Григорьев А. Д. Методы вычислительной электродинамики / А. Д. Григорьев. – М.: Физматлит, 2013. – 430 с.
5. Altair FEKO [Электронный ресурс]. – Режим доступа: www.altair.com.
6. Нефедов Е. И. Полосковые линии передачи / Е. И. Нефедов, А. Т. Фиалковский. – М.: Наука, 1980. – 312 с.
7. Garg R. Analytical and computational methods in electromagnetic / R. Garg. – Norood: Artech House, 2008. – 528 p.
8. Paul C. R. Transmission lines in digital systems for EMC practitioners / C. R. Paul. – Hoboken, New Jersey: John Wiley & Sons, 2012. – 270 p.
9. Краснов М. И. Интегральные уравнения. Задачи и примеры с подробными решениями / М. И. Краснов, А. И. Киселев, Г. И. Макаренко. –3-е изд., испр. – М.: Едиториал УРСС, 2003. – 192 с.
10. Теоретические основы электротехники: учеб. для вузов. В 3 т. Т. 3 / К. С. Демирчян [и др.]. – 4-е изд. – СПб.: Питер, 2003. – 364 с.
11. Jackson J. D. Classical electrodynamics / J. D. Jackson. – NY: John Wiley & Sons, 1962. – 641 p.
12. Ramo S. Fields and waves in communication electronics / S. Ramo, J. R. Whinnery, T. van Duzer. – 3nd ed. – Hoboken, New Jersey: John Wiley & Sons, 1994. – 844 p.

13. Вольман В. И. Справочник по расчету и конструированию СВЧ полосковых устройств / В. И. Вольман. – М.: Радио и связь, 1982. – 328 с.

14. Ховратович В. С. Параметры многопроводных передающих линий / В. С. Ховратович // Радиотехника и электроника. – 1975. – № 3. – С. 469–473.

15. Djordjevic A. R. Time-domain response of multiconductor transmission lines / A. R. Djordjevic, T. K. Sarkar, R. F. Harrington // Proceedings of the IEEE. – 1987. – Vol. 75, no 6. – P. 743–764.

16. Matthaei G. L. Approximate calculation of the high-frequency resistance matrix for multiple coupled lines / G. L. Matthaei, G. C. Chinn // IEEE Microwave Symposium Digest. – 1992. – P. 1353–1354.

17. Куксенко С. П. Итерационные методы решения системы линейных алгебраических уравнений с плотной матрицей / С. П. Куксенко, Т. Р. Газизов. – Томск: Томский государственный университет, 2007. – 208 с.

18. Фаддеев Д. К. Вычислительные методы линейной алгебры / Д. К. Фаддеев, В. Н. Фаддеева. – М.: Физматгиз, 1963. – 734 с.

19. Olyslager F. Numerical and experimental study of the shielding effectiveness of a metallic enclosure / F. Olyslager, E. Lermans, D. de Zutter // IEEE Transactions on electromagnetic compatibility. – 1999. – Vol. 41, no 3. – P. 202–213.

20. Голуб Д. Матричные вычисления / Д. Голуб, Ч. Ван Лоун. – М.: Мир, 1999. – 548 с.

21. Канахер Д. Численные методы и программное обеспечение / Д. Канахер, К. Моулер , С. Нэш. – М.: Мир, 1999. – 576 с.

22. Куксенко С. П. Методы оптимального проектирования линейных антенн и полосковых структур с учетом электромагнитной совместимости : дис. … д-ра техн. наук / С. П. Куксенко. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2019. – 435 с.

23. Evans D. J. The use of pre-conditioning in iterative methods for solving linear equations with symmetric positive definite matrices /

D. J. Evans // Journal of the institute of mathematics and its applications. – 1968. – Vol. 4. – P. 295–314.

24. Preconditioners for adaptive integral method implementation / W.-B. Ewe [et al.] // IEEE Transactions on antennas and propagation. – 2005. – Vol. 53, no 7. – P. 2346–2350.

25. Alleon G. Sparse approximate inverse preconditioning for dense linear systems arising in computation of electromagnetic / G. Alleon, M. Benzi, L. Giraud // Numerical algorithms. – 1997. – Vol. 16. – P. 1–15.

26. Solution of dense systems of linear equations arising from integral equation formulations / K. Forsman [et al.] // IEEE Transactions antennas and propagation. – 1995. – Vol. 37, no 6. – P. 96–100.

27. On short recurrence Krylov type methods for linear systems with many right-hand sides / S. Rashedi [et al.] // Journal of computational and applied mathematics. – 2016. – Vol. 300. – P. 18–29.

28. Knoll D. A. Newton-Krylov methods for low-Mach-number compresible combustion / D. A. Knoll, P. R. McHugh, D. E. Keyes // AIAA Journal. – 1996. – Vol. 34, no 5. – P. 961–967.

29. Tebbens J. D. Efficient preconditioning of sequences of nonsymmetric linear systems / J. D. Tebbens, M. Tuma // SIAM Journal on scientific computing. – 2007. – Vol. 29, no 5. – P. 1918–1941.

30. Jolivet P. Block iterative methods and recycling for improved scalability of linear solvers / P. Jolivet, P. H. Tournier // Proceedings of the International conference for high performance computing, networking, storage and analysis. – 2016. – P. 1–15.

31. Multipreconditioned GMRES for shifted systems / T. Bakhos [et al.] // SIAM Journal on scientific computing. – 2017. – Vol. 39, no 5. – P. 222–247.

32. Sadiku M. N. O. Numerical techniques in electromagnetic / M. N. O. Sadiku. – 3-rd edition. – CRC Press LLC, 2009. – 710 p.

33. Пантелеев А. В. Вариационное исчисление в примерах и задачах: учеб. пособие / А. В. Пантелеев. – М.: Изд-во МАИ, 2000. – 228 с.

34. Гельфанд И. М. Вариационное исчисление / И. М. Гельфанд, С. В. Фомин. – М.: Гос. изд-во физ.-мат. лит., 1961. – 227 с.

35. Михлин С. Г. Вариационные методы в математической физике / С. Г. Михлин. – М.: Наука, 1970. – 512 с.

36. Харрингтон Р. Ф. Применение матричных методов к задачам теории поля / Р. Ф. Харрингтон // Труды института инженеров по электронике и радиотехнике. – 1967. – № 2. – С. 5–19.

37. Harrington R. F. Field computation by moment methods / R. F. Harrington. – USA, NY: Macmillan, 1968. – 240 p.

38. Crandall S. H. Engineering analysis / S. H. Crandall. – New York: McGraw-Hill, 1956. – 151 p.

39. Davidson D. B. Computational electromagnetics for RF and microwave engineering / D. B. Davidson. – Cambridge: University Press, 2011. – 505 p.

40. Канторович А. В. Функциональный анализ в нормированных пространствах / А. В. Канторович, Г. П. Акилов. – М.: Физматлит, 1959. – 684 с.

41. Kryloff N. M. Les méthodes de résolution approchee des problèmes de la physique methematique / N. M. Kryloff . – Paris: Gauthier-Villars, 1931. – 71 p.

42. Кравчук М. Ф. О методе Крылова в теории приближенного интегрирования дифференциальных уравнений / М. Ф. Кравчук // Труды физ.-мат. отдел. ВУАН. – 1926. – Vol. 5, no 2. – P. 12–33. (на украинском).

43. Gibson W. C. The method of moments in electromagnetic / W. C. Gibson. – Boca Raton: Chapman &Hall/CRC, 2008. – 272 p.

44. Makarov S. N. Antenna and EM modeling with MATLAB / S. N. Makarov. – New York: John Wiley & Sons, 2002. – 288 p.

45. Газизов Т. Р. Уменьшение искажений электрических сигналов в межсоединениях / Т. Р. Газизов ; под ред. Н. Д. Малютина. – Томск: НТЛ, 2003. – 212 с.

46. Привалов И. И. Аналитическая геометрия / И. И. Привалов. – М.: Наука, 1966. – 272 с.

Приложение А
(справочное)
Программирование в GNU Octave

GNU Octave – это свободно распространяемый язык программирования высокого уровня, ориентированный на проведение численных расчетов и обладающий богатым инструментарием для решения различных задач.

А.1 Основы синтаксиса

После запуска Octave пользователю доступно окно интерпретатора (рисунок А.1), которое содержит:

1) главное меню, предназначенное для получения доступа ко всем возможным командам и интерфейсам;

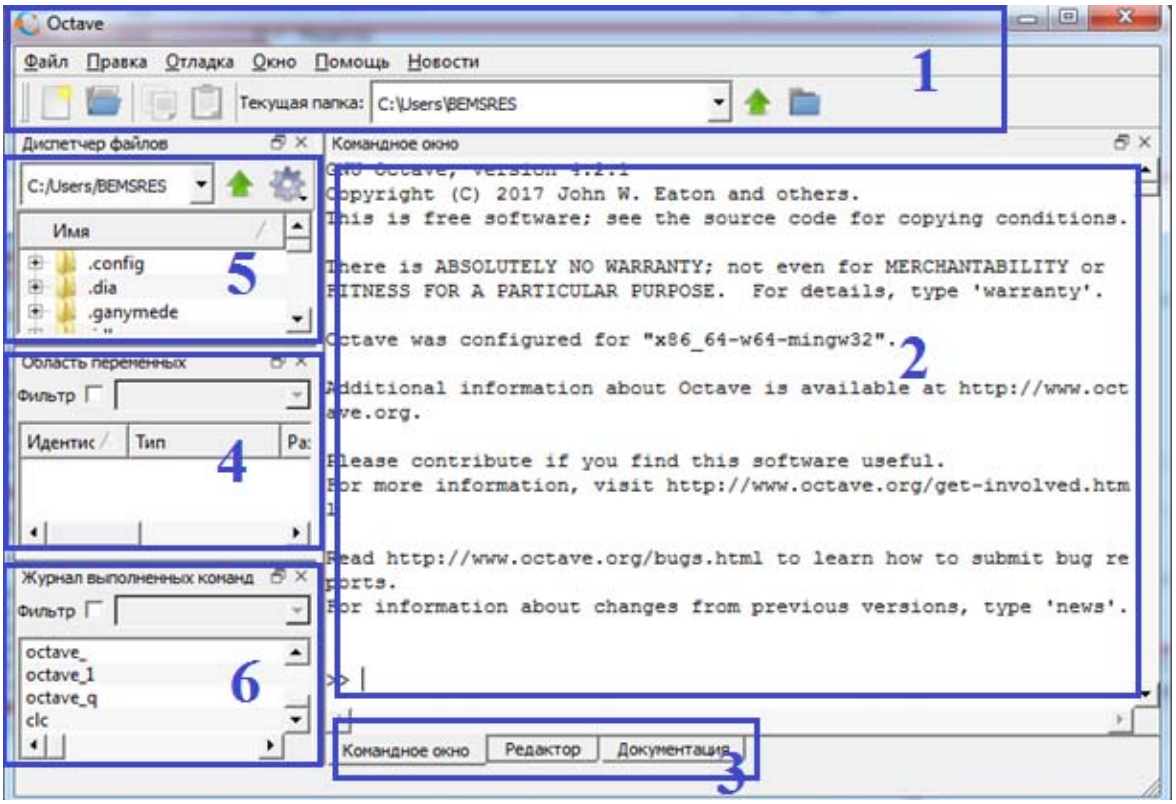


Рисунок А.1 – Графическое окно Octave

2) окно ввода и вывода информации (командное окно/ редактор) для внесения необходимых изменений в программы, созданные или импортированные пользователем. Здесь под программой

понимается текстовый файл, содержащий команды для среды Octave с расширением .m;

3) командное окно/редактор, где пользователь может вводить как отдельные команды языка Octave, так и группы команд. Группы команд удобно объединять в программы (окно Редактор). Содержащиеся в файлах команды последовательно передаются на исполнение, а результат выводится во вкладке *Командное окно*;

4) область переменных, предназначенную для контроля значений и типов данных переменных;

5) файловый менеджер;

6) журнал выполненных команд.

Возможны два варианта решения любой задачи в Octave:

– терминальный режим через вкладку *Командное окно*. В этом режиме в окно интерпретатора последовательно вводятся отдельные команды;

– программный режим через вкладку *Редактор*. В этом режиме создается текстовый файл с расширением .m, в котором хранятся последовательно выполняемые команды Octave. Затем программа запускается на выполнение.

На рисунке А.2, *а* показан пример создания и перемножения двух матриц в терминальном режиме. Для подтверждения ввода и запуска вычислений используется нажатие на клавишу *Enter*. В переменной *ans* хранится результат последней операции. Причем полученное значение можно использовать в последующих вычислениях.

При работе в программном режиме нужно перейти во вкладку *Редактор* и последовательно ввести команды (здесь и далее для облегчения восприятия примеры команд будут показаны курсивом):

```
a=[2 3; 4 5];
b=[4 5; 9 8]
b*a
```

После этого следует сохранить файл, например, с именем example1.m в нужной директории рабочей станции и нажать F5. Перейдя назад во вкладку *Командное окно*, можно увидеть результат вычисления (рисунок А.2, *б*). Если строка заканчивается символом «;», результаты на экран не выводятся. Текст в строке,

следующий за символом «%», является комментарием и интерпретатором не обрабатывается.

```
>> a=[2 3; 4 5]%create matrix
a =

   2   3
   4   5

>> [4 5; 9 8]%create matrix
ans =

   4   5
   9   8

>> ans*a
ans =

   28   37
   50   67
```

а

```
>> example1

b =

   4   5
   9   8

ans =

   28   37
   50   67
```

б

Рисунок А.2 – Перемножение двух матриц:
а – терминальный режим; *б* – программный режим

Рассмотрим основные моменты синтаксиса языка Octave, используя терминальный режим. Основной рабочей структурой данных в Octave является матрица. Поэтому необходимо помнить, что любому числу соответствуют матрицы размером 1×1. Продемонстрируем это с помощью операции присваивания «=», которая передает значение выражения в левой части равенства переменной, стоящей справа. Так, если написать в командном окне строку

>>a=5

то переменной *a* будет присвоено значение 5. Чтобы узнать размер этой переменной, воспользуемся командой

>> size(a)
ans =
1 1

Видно, что размер матрицы *a* в данном случае 1×1.

Имя переменной не должно совпадать с именами встроенных системных процедур, функций и переменных. Octave различает прописные и строчные буквы в именах переменных, т. е. ABC, abc, Abc, aBc и т. д. – это имена разных переменных.

Кроме переменной *ans*, в Octave существуют и другие системные переменные, которые можно использовать в математических выражениях:

– i, j – мнимая единица;

– pi – число π;

– *e* –экспонента, 2,71828183;

– realmin – наименьшее число с плавающей точкой (2.2251e-308);

– realmax – наибольшее число с плавающей точкой (1.7977e+308);

– inf – машинный символ бесконечности;

– NaN – неопределенный результат.

Как было показано выше, возможно выполнение операции присвоения сразу ко всем элементам матрицы, например

```
>> B=[1,2,3]
B =
   1 2 3
```

Эта команда генерирует матрицу размерности 1×3 (вектор-строка). При введении данных через «;» будет сформирован вектор-столбец

```
>> C=[4;5;6]
C =
   4
   5
   6
```

Аналогичным образом генерируются матрицы любой размерности.

Если значения некоторой величины меняются с равным шагом, то оператор присваивания может использоваться совместно с оператором «:», определяющим пределы изменения величины. Значение шага задается пользователем. По умолчанию шаг равен единице. Например:

```
>> E=1:4:17 %начальное значение : шаг : конечное значение
E =
   1 5 9 13 17.
```

Оператор «:» имеет еще одно применение – он может определять совокупность строк или столбцов в матрице:

```
>> D=zeros(4,4)
```

```
D =
     0   0   0   0
     0   0   0   0
     0   0   0   0
     0   0   0   0
>> D(1,:)=1
D =
     1   1   1   1
     0   0   0   0
     0   0   0   0
     0   0   0   0
```

Здесь с помощью оператора «:» произошло присвоение значения 1 всем элементам первой строки матрицы *D*.

Для очистки командного окна можно воспользоваться командой *clc*, а для удаления из памяти всех переменных или конкретной переменной – командами *clear* и *clear имя переменной* соответственно. Простейшие арифметические операции в Octave выполняют с помощью следующих операторов:

– сложение «+»;

– вычитание «–»;

– умножение «*»;

– деление слева направо «/»;

– деление справа налево «\»;

– возведение в степень «^».

Если вычисляемое выражение слишком длинное, то перед нажатием клавиши ENTER следует ввести троеточие. Это будет означать продолжение текущей (командной) строки.

```
>> 1+2+...
4+5
ans =  12
```

Допускается переназначение не только значений переменных, но и их типов. Например, вещественной переменной можно присвоить символьное значение, при этом автоматически произойдет смена ее типа. Но интерпретатор команд отслеживает только согласование размерности переменных в выражении. Так, умножение символьной строки на матрицу или сложение двух матриц разной размерности вызовет ошибку.

```
>> A=[3 4; 7 5];
>> A='line';
```

```
>> B=[1 2; 3 4];
>> A*B
error: operator *: nonconformant arguments (op1 is 1x4, op2 is 2x2)
```

Для удобства пользователя большинство арифметических операций можно выполнить поэлементно. Для этого используется знак «.» в дополнение к требуемому оператору.

```
>> A=[1,2;3,4];
>> B=A*A
B =
    7 10
    15 22
>> C=A.*A
C =
    1 4
    9 16
```

В первом случае матрица **A** была умножена сама на себя, а во втором произошло поэлементное возведение в квадрат каждого ее элемента.

Числовые значения результатов могут быть представлены с плавающей или с фиксированной точкой. Для представления чисел с плавающей точкой используется экспоненциальная форма записи $mE \pm p$, где m – мантисса (целое или дробное число с десятичной точкой), p – порядок (целое число).

```
>> 5.49e-2
ans =  0.054900
>> 5.49E-2
ans =  0.054900
>> 5.49.*1E-2
ans =  0.054900
>> 5.49.*10^-2
ans =  0.054900
>> 5.49*10^-2
ans =  0.054900
```

Рассмотрим ввод числа:

```
>> 0.987654321
ans =  0.98765
```

Видно, что количество знаков в дробной части числа при вводе больше, чем при выводе. Это связано с тем, что вывод результата вычислений определяется предварительно заданным форматом представления чисел. В Octave предусмотрено несколько форматов чисел. Приведем некоторые из них:

– Short – краткая запись, применяется по умолчанию;

– Long – длинная запись;

– Short E (Short e) – краткая запись в формате с плавающей точкой;

– Long E (Long e) – длинная запись в формате с плавающей точкой.

Продемонстрируем их использование на примере числа π.

```
>> format short
>> pi
ans =  3.1416
>> format long
>> pi
ans =  3.14159265358979
>> format short E
>> pi
ans =   3.1416E+000
>> format long E
>> pi
ans =   3.14159265358979E+000
```

Имеются также тригонометрические функции, экспонента и натуральный логарифм.

```
>> x=pi/2
>> x =  1.5708
>> sin(x)
ans =  1
>> tan(x)
ans =   1.6331e+016
>> exp(x)
ans =  4.8105
>> log(x)
ans =  0.45158
```

Кроме того, имеется ряд других полезных функций:

– fix(x) – округление числа *x* до ближайшего целого в сторону нуля;

– floor(x) – округление числа *x* до ближайшего целого в сторону отрицательной бесконечности;

– ceil(x) – округление числа *x* до ближайшего целого в сторону положительной бесконечности;

– round(x) – округление числа *x* до ближайшего целого;

– rem(x, y) – вычисление остатка от деления *x* на *y*;

– sign(x) – выдает 0, если x=0, –1 при $x < 0$ и 1 при $x > 0$;

– sqrt(x) – корень квадратный из числа x;

– abs(x) – модуль числа x;

– log10(x) – десятичный логарифм от числа x;

– log2(x) – логарифм по основанию два от числа x;

– pow2(x) – возведение двойки в степень x;

– gcd(x, y) – наибольший общий делитель чисел x и y;

– lcm(x, y) – наименьшее общее кратное чисел x и y;

– rats(x) – представление числа x в виде рациональной дроби.

Для обозначения мнимой единицы используются i и/или j. Ввод комплексного числа производится в формате *действительная часть* + *i*мнимая часть*. К комплексным числам применимы элементарные арифметические операции +, –, *, \, /, ^, а также специальные функции:

– real(z) – выдает действительную часть комплексного аргумента z;

– imag(z) – выдает мнимую часть комплексного аргумента z;

– angle(z) – вычисляет значение аргумента комплексного числа z в радианах от $-\pi$ до π;

– conj(z) – выдает число комплексно сопряженное z.

Рассмотрим операции отношения, предназначенные для выполнения сравнения двух операндов и определения истинности выражения. Результатом операции отношения является логическое значение (1 или «истина» и 0 или «ложь»). Предусмотрены следующие операции отношения:

– меньше «<»;

– больше «>»;

– равно «==»;

– не равно «~=»;

– меньше или равно «<=»;

– больше или равно «>=».

В Octave существует возможность представления логических выражений в виде логических операторов и логических операций (таблица А.1).

Таблица А.1 – Виды логических выражений

Тип выражения	Выражение	Логический оператор	Логическая операция
Логическое «и»	A and B	and(A, B)	A & B
Логическое «или»	A or B	or(A, B)	A | B
Исключающее «или»	A xor B	xor(A,B)	
Отрицание	not A	not (A)	~ A

Для визуализации входных данных и полученных результатов Octave располагает обширными библиотеками графических построений. Возможно построение как двухмерных, так и трехмерных графиков. Основной функцией для построения двухмерных графиков служит функция plot, у которой несколько вариантов вызова:

– plot(X,Y) – построение зависимости $y(x)$, где значения y и x берутся из матриц **Y** и **X**, как правило, одномерных;

– plot(X1,Y1,..., Xn,Yn) – одновременное построение нескольких функциональных зависимостей;

– plot(X1,Y1,LineSpec1,..., Xn,Yn,LineSpecn) – наиболее полный вариант вызова функции, где LineSpec – это шаблон, с помощью которого определяется цвет линии, ее толщина, вид маркеров и другие параметры (таблица А.2).

```
>> x=0:0.1:2*pi;
>> plot(x,sin(x),'Color','m','LineStyle','-.','LineWidth',4);
```

Таблица А.2 – Параметры функции plot для задания шаблона линии

Параметр	Возможные значения
Color' – цвет линии	'y' – желтый; 'm' – магента (пурпурный); 'c' – циан (зелено-голубой); 'r' – красный; 'g' – зеленый; 'b' – голубой; 'k' – черный; [r g b] – цвет в формате *RGB* (параметры *r*, *g*, *b* лежат в пределах от 0 до 1)
LineStyle' – вид линии	'-' – сплошная; '- -' – двойная сплошная; ':' – пунктирная; '-.' – штрихпунктирная; 'none' – без линии (только маркеры)
'LineWidth' – ширина линии	Положительные значения с шагом 0.5
Marker' – вид маркера	'o' – круговой; '+' – знак «+»; 's' – квадрат; 'p' – пятиугольник; '^' – ориентированный вверх треугольник

Если последовательно использовать еще одну команду plot, то после ее выполнения график новой функции будет нарисован поверх предыдущего. Чтобы это не происходило, надо настроить графическое окно (таблица А.3).

Таблица А.3 – Функции для задания параметров работы графического окна

Функция	Варианты использования
figure – функция запроса графического окна. После применения все изображения будут перенаправляться в запрошенное окно	figure() – вызывает графическое окно, номер которому будет присвоен автоматически (возможен вызов в виде N=figure(), в этом случае переменной *N* будет присвоен номер графического окна) figure(N) – будет создано новое окно с номером *N*, если такого окна не существовало, в противном случаев в него будет перенаправлен графический вывод
hold – функция, включающая и отключающая сохранение в графическом окне предыдущих графиков	hold on – новые графики будут изображаться совместно с предыдущими hold off – новые графики будут изображаться поверх предыдущих, затирая их
grid – функция, включающая и отключающая отображение линий сетки	grid on – включает отображение линий сетки grid off – выключает отображение линий сетки
xlabel, ylabel – функции подписывания осей	xlabel('НАЗВАНИЕ ОСИ X') ylabel('НАЗВАНИЕ ОСИ Y')
legend – функция нанесения на график легенды (сопровождающих и поясняющих надписей)	legend(' СТРОКА 1', ' СТРОКА 2', ...)

Следующий пример демонстрирует построение зависимостей sin(*x*) и cos(*x*) на одном графике.

```
>> x=0:0.1:2*pi;
>> figure(1);
>> hold on;
>> plot(x,sin(x),'Color','m','LineStyle','-.','LineWidth',4)
>> plot(x,cos(x),'Color','g','LineStyle','--','LineWidth',3)
>> legend('sin(x)','cos(x)')
>> grid on;
```

```
>> xlabel('x');
>> ylabel('sin(x)');
```

В Octave есть возможность строить несколько координатных осей в графическом окне и выводить на каждую из них разные графики. Для этого используется функция *subplot(row, col, cur)*. Параметры *row* и *col* определяют требуемое число графиков по вертикали и горизонтали соответственно, а *cur* – номер текущего графика. Повторное обращение к этой функции с теми же значениями *row* и *col* позволяет изменять номер текущего графика и переключаться между графиками.

Для построения графиков поверхностей используется несколько функций. Сначала формируется прямоугольная сетка, содержащая координаты узловых точек, с помощью функции meshgrid():

```
>> [x y]=meshgrid(-2:2,-1:1)
x =
    -2  -1   0   1   2
    -2  -1   0   1   2
    -2  -1   0   1   2
y =
    -1  -1  -1  -1  -1
     0   0   0   0   0
     1   1   1   1   1
```

Далее используется функция mesh() для построения «каркасного» графика. Рассмотрим это на примере функции $z(x, y) = 4x^2 - 2\sin^2 y$.

```
>> z=4*x.^2-2*sin(y).^2
z =
    14.58385   2.58385  -1.41615   2.58385  14.58385
    16.00000   4.00000   0.00000   4.00000  16.00000
    14.58385   2.58385  -1.41615   2.58385  14.58385
>> mesh(x, y, z)
```

Еще одной функцией для построения трехмерных графиков является surf(**x**, **y**, **z**) или surf(**x**, **y**, **z**, *C*), где **x** и **y** – векторы-строки, определяющие значения координат узлов; **z** – матрица с размерностью, равной произведению размерностей матриц **x** и **y**, задающая значения координат узлов по оси *z* для соответствующих пар *x* и *y*. Параметр *C* определяет способ отображения результата (цвет, режим отображения кромок и т. д.).

Перечисленные способы построения как двухмерных, так и трехмерных графиков не являются единственно возможными. О других способах можно узнать из документации Octave.

А.2 Работа с матрицами

Элементы вектора-строки отделяют пробелами или запятыми, а всю конструкцию заключают в квадратные скобки. Вектор-столбец можно задать, если элементы отделять друг от друга точкой с запятой. Для обращения к элементу вектора надо указать его порядковый номер. Нумерация элементов начинается с единицы.

```
>> a=[2 -3 5 6 -1 0 7 -9]
a =
     2 -3 5 6 -1 0 7 -9
>> b=[-1,0,1]
b =
     -1 0 1
>> c=[-pi;-pi/2;0;pi/2;pi]
c =
     -3.14159
     -1.57080
     0.00000
     1.57080
     3.14159
>> b(3)
ans = 1
```

Ввод элементов матрицы, как и в случае с векторами, осуществляется в квадратных скобках. При этом элементы строки отделяются друг от друга пробелом или запятой, а строки разделяются между собой точкой с запятой. Обратиться к элементу матрицы можно, указав номера строки и столбца, на пересечении которых расположен интересующий элемент.

```
>> Matr=[0 1 2 3;4 5 6 7]
Matr =
     0 1 2 3
     4 5 6 7
>> Matr(2,3)
ans = 6
>> Matr(1,1)=pi; Matr(2,4)=-pi;
>> Matr
Matr =
```

```
    3.1416 1.0000 2.0000 3.0000
    4.0000 5.0000 6.0000 -3.1416
```

Матрицы и векторы можно формировать, составляя их из ранее заданных матриц и векторов, используя конкатенацию.

```
>> a=[-3 0 2];b=[3 2 -1];c=[5 -2 0];
>> M=[a b c]
M = -3 0 2 3 2 -1 5 -2 0
>> N=[a;b;c]
N =
    -3 0 2
    3 2 -1
    5 -2 0
>> Matrica=[N N N]
Matrica =
    -3 0 2 -3 0 2 -3 0 2
    3 2 -1 3 2 -1 3 2 -1
    5 -2 0 5 -2 0 5 -2 0
```

Важную роль при работе с матрицами играют знак двоеточия «:» (задание диапазона индексов) и оператор [] (удаление строк и столбцов из матрицы).

```
>>Tabl=[-1.2 3.4 0.8; 0.9 -0.1 1.1; 7.6 -4.5 5.6; 9.0 1.3 -8.5];
>> Tabl(:,3)
ans =
    0.80000
    1.10000
    5.60000
    -8.50000
>> Tabl(1,:)
ans = -1.20000 3.40000 0.80000
>> Matr=Tabl(2:3,1:2)
Matr =
     0.90000 -0.10000
     7.60000 -4.50000
>> Tabl(3:4,2:3)=Matr
Tabl =
    -1.20000 3.40000 0.80000
    0.90000 -0.10000 1.10000
    7.60000 0.90000 -0.10000
    9.00000 7.60000 -4.50000
>> Tabl(:,2)=[]
Tabl =
    -1.20000 0.80000
    0.90000 1.10000
```

```
    7.60000 -0.10000
    9.00000 -4.50000
>> Vector=Matr(:)
Vector =
    0.90000
    7.60000
    -0.10000
    -4.50000
>> V=Vector(1:3)
V =
    0.90000
    7.60000
    -0.10000
>> V(2)=[]
V =
    0.90000
    -0.10000
```

Рассмотрим действия над векторами, предусмотренные в Octave. Сложение и вычитание (знаки «+» и «–») возможно только для векторов одного типа, то есть суммировать/вычитать можно либо векторы-столбцы, либо векторы-строки одинаковой длины. Знак апострофа «'» применяется для транспонирования. Умножение вектора на число осуществляется с помощью знака «*». Знак деления «/» применяют для того, чтобы разделить элементы вектора на число.

```
>> a=[2 4 6];b=[1 3 5];
>> c=a+b
c =
    3 7 11
>> a'
ans =
    2
    4
    6
>>> z=2*a+a/4
z =
    4.5000 9.0000 13.5000
```

Умножение вектора на вектор выполняется также с помощью знака «*». Эта операция применима только к векторам одинаковой длины, причем один из них должен быть вектором-столбцом, а второй – вектором-строкой.

```
>> a=[2 4 6]; b=[1 3 5];
>> a*b'
ans = 44
>> a'*b
ans =
     2 6 10
     4 12 20
     6 18 30
>> a*b
error: operator *: nonconformant arguments (op1 is 1x3, op2 is 1x3)
```

Кроме перечисленных действий с векторами, возможно их поэлементное преобразование. Например, если к некоторому вектору применить математическую функцию, то в результате получим новый вектор того же размера и структуры, но с преобразованными в соответствии с заданной функцией элементами.

```
>> x=[-pi/2,-pi/3,-pi/4,0,pi/4,pi/3,pi/2]
x =
     -1.5708 -1.0472 -0.7854 0.0000 0.7854 1.0472 1.5708
>> y=sin(2*x)+cos(2*x)
y =
     -1.0000 -1.36603 -1.0000 1.0000 1.0000 0.36603 -1.0000
```

Поэлементное умножение векторов выполняется при помощи оператора «.*». В результате генерируется вектор, каждый элемент которого равен произведению соответствующих элементов исходных векторов. Поэлементное деление осуществляется оператором «./». В результате получается вектор, каждый элемент которого является частным от деления соответствующих элементов первого и второго векторов. Для обратного поэлементного деления используется совокупность знаков «.\». Поэлементное возведение в степень элементов вектора выполняет оператор «.^».

```
>> a=[2 4 6];b=[1 3 5];
>> a.*b
ans = 2 12 30
>> a=[2 4 6];b=[1 3 5];
>> a./b
ans =
     2.0000 1.3333 1.2000
>> a.\b
ans =
     0.50000 0.75000 0.83333
```

Далее рассмотрим действия над матрицами, которые по сути совпадают с операциями над векторами, но есть принципиальные отличия. При сложении «+» и вычитании «–» важно помнить, что матрицы должны быть одной размерности.

При умножении матриц «*» число столбцов первой матрицы должно совпадать с числом строк второй матрицы. При умножении матрицы на число результатом будет матрица, каждый элемент которой помножен на заданное число. Операция транспонирования «'» меняет местами строки и столбцы заданной матрицы.

Возведение матрицы в степень «^» эквивалентно ее умножению на себя указанное число раз. При этом показатель степени может быть как положительным, так и отрицательным. Матрица в степени минус один называется обратной к данной. Возведение в отрицательную и целочисленную степень соответствует умножению обратной матрицы на себя указанное число раз. Для поэлементного преобразования матриц могут применяться операции, описанные для векторов.

```
>> A=[1 2 3; 4 5 6; 7 8 9]; B=[-1 -2 -3; -4 -5 -6; -7 -8 -9];
>> (2*A+1/4*B')^2-A*B^(-1)
ans =
    83.875   93.125  104.375
   219.250  244.062  272.875
   356.625  395.000  457.375
```

Операторы «/» и «\» используются для деления матриц слева направо и справа налево соответственно. Так, деление матриц *B/A* соответствует выражению $\mathbf{BA}^{-1}$ и, как правило, используется при решении матричных уравнений вида $\mathbf{XA} = \mathbf{B}$. Соответственно деление *A\B* эквивалентно $\mathbf{A}^{-1}\mathbf{B}$ и применяется для решения СЛАУ вида $\mathbf{AX} = \mathbf{B}$.

```
>> A=[1 2;1 1];
>> b=[7;6];
>> x=A\b
x =
   5
   1
```

В Octave имеются специальные функции, которые можно разделить на 3 группы: функции для работы с векторами; функции

для работы с матрицами; функции, реализующие алгоритмы решения задач линейной алгебры.

Приведем наиболее часто используемые функции для векторов:

– length(x) – определение длины вектора **x**;

– prod(x) – вычисление произведения элементов вектора **x**;

– sum(x) – вычисление суммы элементов вектора **x**;

– diff(x) – генерация вектора длиной на единицу меньше, чем у вектора **x**, каждый элемент которого есть разность между двумя соседними элементами вектора **x**;

– diag(x, k) – генерация квадратной матрицы с элементами вектора **x** на главной (если *k* не задано) или *k*-й диагонали;

– min(x) – поиск минимального элемента вектора **x**;

– max(x) – поиск максимального элемента вектора **x**;

– mean(x) – вычисление среднего арифметического элементов вектора;

– dot(x1,x2) – вычисление скалярного произведения двух векторов;

– cross(x1,x2) – вычисление векторного произведения векторов;

– sort(x) – сортировка массива **x** по возрастанию (сортировка по убыванию – sort(-x)).

Наиболее часто используемые функции для матриц:

– eye(n , m) – генерация единичной матрицы размерности $n \times m$;

– ones(n , m) – генерация матрицы, состоящей из единиц, размерности $n \times m$;

– zeros(n, m) – генерация матрицы, состоящей из нулей, размерности $n \times m$;

– diag(A, k) – генерация вектора-столбца из элементов, расположенных на главной (если *k* не задано) или *k*-й диагонали матрицы **A**;

– rand(n, m) – генерация матрицы размерности $n \times m$, элементы которой случайные числа, распределенные по равномерному закону;

– randn(n, m) – генерация матрицы размерности $n \times m$, элементы которой случайные числа, распределенные по нормальному закону;

– linspace(a, b, n) – генерация массива из n элементов, равномерно распределенных между значениями a и b;

– repmat(A, n, m) – генерация матрицы, состоящей из $n \times m$ копий матрицы **A**;

– reshape(A, m, n) – генерация матрицы размерности $m \times n$ из матрицы **A** путем последовательной выборки по столбцам;

– cat(n, A, B) – конкатенация матриц **A** и **B** (при $n = 1$ конкатенация по столбцам, а при $n = 2$ – по строкам);

– rot90(A, k) – поворот матрицы **A** на $90k$ градусов, где k – целое число;

– tril(A, k) – генерация нижнетреугольной матрицы из матрицы **A** начиная с главной (если k не задано) или k-й диагонали;

– triu(A , k) – генерация верхнетреугольной матрицы из матрицы **A** начиная с главной (если k не задано) или k-й диагонали;

– size(A) – определение числа строк и столбцов матрицы **A**;

– prod(A, k) – генерация вектора-столбца ($k = 1$) или вектора-строки ($k = 2$), каждый элемент которого является произведением элементов соответствующего столбца или строки матрицы **A**;

– sum(A, k) – генерация вектора-столбца ($k = 1$) или вектора-строки ($k = 2$), каждый элемент которого является суммой элементов соответствующего столбца или строки матрицы **A**;

– diff(A) – генерация матрицы размерности $(n - 1) \times m$, элементы которой являются разностями между элементами соседних строк матрицы **A** размерности $n \times m$;

– min(A) – генерация вектора-строки, элементы которого являются наименьшими элементами из соответствующих столбцов матрицы **A**;

– max(A) – генерация вектора-строки, элементы которого являются наибольшими элементами из соответствующих столбцов матрицы **A**;

– mean(A, k) – генерация вектора-столбца ($k = 1$) или вектора-строки ($k = 2$), каждый элемент которого является средним ариф-

метическим значением элементов соответствующего столбца или строки матрицы **A**;

– sort(A) – генерация из матрицы **A** матрицы той же размерности, каждый столбец которой упорядочен по возрастанию.

Рассмотрим наиболее часто используемые функции, реализующие численные алгоритмы решения задач линейной алгебры:

det(M) – вычисление определителя квадратной матрицы **M**;

trace(M) – вычисление суммы элементов на главной диагонали матрицы **M**;

– norm(A, p) – вычисление нормы матрицы **A**, где p = 1, 2, 'inf', 'fro';

– cond(A, p) – вычисление числа обусловленности матрицы **A** по норме p, где p = 1, 2, 'inf', 'fro';

– rcond(A) – вычисление величины, обратной значению числа обусловленности матрицы **A** относительно первой нормы (если полученная величина близка к единице, то матрица считается хорошо обусловленной, а если к нулю, то плохо обусловленной);

– inv(A) – генерация матрицы, обратной к **A**;

– eig(A) – вычисление собственных значений столбцов матрицы **A**;

– chol(A, str) – генерация верхнетреугольной (если str = 'upper') или нижнетреугольной (str = 'lower') матрицы **L**, полученной разложением по Холецкому для положительно определенной и симметричной матрицы **A**;

– lu(A) – LU-разложение матрицы **A**, результат является объединением матриц **L** и **U**;

– qr(A) – QR–разложение матрицы **A**, результат является объединением матриц **Q** и **R**;

– svd(A) – генерация вектора-столбца, состоящего из сингулярных чисел матрицы **A**.

A.3 Основы программирования

Рассмотренные выше группы команд представляют собой простейшие программы Octave. Если такая программа хранится в файле с расширением .m, то для ее выполнения достаточно в командной строке Octave ввести имя этого файла (без расширения).

Очевидно, что часто требуется разработка более сложных программ, содержащих функции как составные элементы, поэтому возникает необходимость в использования циклов, условий и прочих атрибутов программирования. Далее рассмотрим основные операторы языка Octave и примеры их использования.

Даже при разработке простейших программ возникает необходимость ввода исходных данных и вывода результатов. Если для вывода результатов на экран можно просто не ставить «;» в конце нужной строки, то для ввода данных при разработке программы, реализующей диалоговый режим, следует использовать функцию input ('подсказка'). Если в тексте программы встречается эта функция, то ее выполнение приостанавливается и на экран выводится текст подсказки, а Octave переходит в режим ожидания ввода. После того как пользователь введет с клавиатуры требуемое значение и нажмет клавишу *Enter*, это значение будет присвоено переменной, указанной слева от знака присваивания.

```
>> d=input('Enter value ')
Enter value 5
d =  5
```

В Octave, как и в большинстве других языков программирования, одним из основных операторов, предназначенных для ветвления, является условный оператор *if-else*. Рассмотрим его простую и расширенную версии. Простой условный оператор работает следующим образом. Если некое условие истинно, то выполняется *блок кода 1*, содержащий один или несколько операторов, а если условие ложно, то выполняется *блок кода 2*.

```
if условие
    блок кода 1
else
    блок кода 2
end
```

Расширенная версия условного оператора используется для организации сложных проверок. Например, если *условие 1* истинно, то выполняется *блок кода 1*, иначе проверяется *условие 2*, если оно истинно, то выполняется *блок кода 2* и т. д. Если ни одно из условий по веткам *elseif* не является истинным, то выполняются операторы по ветке *else*.

if условие 1
    блок кода 1
elseif условие 2
    блок кода 2
elseif условие 3
    блок кода 3
…
elseif условие n
    блок кода n
else
    блок кода m
end

Еще одним способом организации сложных ветвлений является оператор *switch*. Так, если значение контролируемого *параметра* равно *значению 1*, то выполняется *блок кода 1*, иначе, если *параметр* равен *значению 2*, то выполняется *блок кода 2*, и т. д. Если значение *параметра* не совпадает ни с одним из значений в группах *case*, то выполняется *блок кода*, следующий за ключевым словом *otherwise*.

swith параметр
case значение 1
    блок кода 1
case значение 2
    блок кода 2
case значение 3
    блок кода 3
…
otherwise
    блок кода m
end

Для организации повторяющихся действий предусмотрены циклы *while* и *for*. Цикл с предусловием *while* работает следующим образом. Если *выражение* (переменная или логическое выражение) истинно, то выполняется *блок кода*, находящийся после слова *while*, иначе цикл игнорируется и управление передается оператору, следующему за телом цикла. Перед каждой итерацией цикла происходит проверка *выражения*.

while выражение
    блок кода
end

Цикл *for* в общем случае является циклом с заранее известным числом итераций. Выполнение цикла начинается с присвоения контролируемому параметру цикла начального значения, после чего следует проверка, не превышает ли оно конечное значение. Если результат утвердительный, то выполняется *блок кода* в теле цикла, иначе цикл прерывается и управление передается следующему за телом цикла оператору. По окончании итерации значение параметра изменяется на значение шага и следует его повторная проверка.

```
for параметр = начальное значение:шаг:конечное значение
    блок кода
end
```

Если значение шага цикла равно 1, то можно использовать сокращенную запись.

```
for параметр = начальное значение:конечное значение
    операторы
end
```

Иногда при выполнении цикла надо прервать итерации, например при выполнении какого-то условия. Поэтому нужны операторы, которые принудительно изменяют порядок выполнения команд за счет передачи управления. Для этого внутри циклов используются операторы *break* и *continue*. Оператор *break* осуществляет немедленный выход из цикла и управление передается следующему за циклом оператору. Оператор *continue* начинает новую итерацию цикла, даже если предыдущая не была завершена.

В Octave файлы с расширением .m могут быть оформлены как отдельные функции. В этом случае имя функции должно совпадать с именем файла, в котором она хранится. Например, функция с именем *example* должна храниться в файле example.m. Функции имеют определенную структуру. Так, первая строка функции – это заголовок вида

$$function\ [y_1,y_2,...y_n]=name\ function(x_1,x_2,...,x_m),$$

где *name function* – имя функции; x_1, x_2, …, x_m – список ее входных параметров; y_1, y_2, …, y_n – список выходных параметров. После заголовка следуют требуемые операторы. Для обозначения конца функции используется ключевое слово *end*.

```
function [y1,y2,...yn]=name_function(x1,x2,...,xm)
```

```
    оператор1;
    оператор2;
    ...
    оператор n;
end
```

В файле с расширением .m, кроме основной функции, могут находиться так называемые подфункции, которые сами является функциями, но доступны они только внутри этого файла.

```
%Начало основной функции из  m-файла, имя которой должно совпадать
% с именем файла, в котором она хранится
function [y1,y2,...yn]=name function(x1,x2,...,xm)
% Среди операторов основной функции могут быть операторы
% вызова подфункций f1, f2, f3, ...,fl
    оператор1;
    оператор2;
    ...
    оператор n;
end; % здесь заканчивается основная функция

function [y1,y2,...yn]=f1(x1,x2,...,xm) % начало первой подфункции
    операторы
end % конец первой подфункции
...
function [y1,y2,...yn]=fn(x1,x2,...,xm) % начало n-й подфункции
    операторы
end % конец n-й подфункции
```

В Octave есть возможность передавать как входной параметр имя функции, что существенно расширяет рамки программирования. Для этого используется функция *feval*, аргументами которой являются строка с именем вызываемой функции (встроенной или определенной пользователем) и параметры вызываемой функции, разделенные запятой.

```
>> a=[2;4];
>> feval('sum',a)
ans =
   6
```

Встроенные функции *tic* и *toc* используются для измерения времени работы любого блока кода, расположенного между ними.

```
tic
    блок кода
toc
```

Иногда необходимо просмотреть информацию по всем переменным или по какой-либо определенной переменной. Это можно сделать через область переменных (см. рисунок А.1) или воспользоваться командами who (выводит список переменных) и whos (выводит список переменных с указанием их размера и объема занимаемой памяти).

```
>> A=[1 2; 3 4];
>> b='line';
>> c=5;
>> d=1+i;
>> who
Variables in the current scope:
A  b  c  d
>> whos
Variables in the current scope:
  Attr      Name       Size                Bytes      Class
  ====      ====       ====                =====      =====
            A          2x2                    32      double
            b          1x4                     4      char
            c          1x1                     8      double
  c         d          1x1                    16      double
Total is 10 elements using 60 bytes
>> whos d
Variables in the current scope:
  Attr      Name       Size                Bytes      Class
  ====      ====       ====                =====      =====
  c         d          1x1                    16      double
Total is 1 element using 16 bytes
```

Оглавление

Учебное издание

Куксенко Сергей Петрович

ЭЛЕКТРОМАГНИТНАЯ СОВМЕСТИМОСТЬ: ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ЭЛЕКТРОСТАТИКИ

Учебное пособие

Подписано в печать 18.11.20. Формат 60x84/16.
Усл. печ. л. 15,58. Тираж 100 экз. Заказ № 259.

Томский государственный университет
систем управления и радиоэлектроники.
634050, г. Томск, пр. Ленина, 40.
Тел. (3822) 533018.