Федеральное государственное бюджетное
образовательное учреждение высшего образования
Московский государственный университет имени М.В.Ломоносова

Физический факультет, Кафедра математики

На правах рукописи

Белов Александр Александрович

Экономичные методы расчета жестких задач в моделях кинетики, теплопроводности, диффузии

Диссертация на соискание ученой степени
кандидата физико-математических наук

по специальности 05.13.18 — математическое моделирование, численные методы и комплексы программ

Научный руководитель
член-корреспондент РАН,
доктор физико-математических наук, профессор
Калиткин Николай Николаевич

Москва
2017

Содержание

1. Введение

1.1 Понятие жесткости

Впервые математики столкнулись с жесткостью около 1950-го года при расчетах горения ракетного топлива. Поэтому изначально понятие жесткости было введено применительно к системам обыкновенных дифференциальных уравнений $d\mathbf{u}/dt = \mathbf{f}(\mathbf{u}, t)$. Несмотря на большую прикладную значимость таких задач, безупречного определения жесткости пока не предложено.

Существует ряд формальных критериев. Например, большое различие модулей спектральных чисел матрицы Якоби $\mathbf{f_u}$ или большое различие величины правых частей для разных компонент. Однако существуют примеры, когда эти критерии не работают. Например, жесткой может быть не только система уравнений, но и одно скалярное уравнение. Типичным примером является задача с пограничным слоем для одного уравнения [1].

В пограничном слое производная очень велика, но на небольшом промежутке времени; в регулярной части производная невелика, но соответствующий отрезок времени немал. Поэтому разумным является качественное определение, предложенное Ю. В. Ракитским: ***Задача называется жесткой, если в ней имеется большая разномастабность процессов (то есть имеются сильно разномасштабные участки решения)***. Общепринятое определение жесткости систем через отношение границ спектра матрицы Якоби является лишь частным случаем разномасштабности. Определение жесткости по Ракитскому применимо не только к начальным задачам для ОДУ, но и к краевым сингулярно возмущенным задачам для уравнений в частных производных.

В данной работе рассматриваются ***1°*** задачи кинетики реакций, ***2°*** задачи с разрушающимися решениями, ***3°*** счет на установление для эллиптических уравнений и ***4°*** сингулярно возмущенное уравнение Гельмгольца. Все указанные задачи относятся к жестким и предъявляют высокие требования к надежности методов расчета. Для этих задач актуальна разработка экономичных численных методов, позволяющих проводить расчеты с высокой гарантированной точностью. Построению последних посвящена данная работа. С использованием построенных методов проведено моделирование кинетики 4-х важных термоядерных реакций. В связи с этой задачей решается вспомогательная, но имеющая большую практическую значимость проблема ***5°*** о нахождении зависимости скоростей этих реакций от температуры.

Ниже формулируются постановки перечисленных задач, описываются трудности, возникающие при их решении, и приводится обзор современного состояния исследований. Дается общая характеристика работы и приводится ее краткое содержание.

1.2 Нестационарные жесткие задачи

1.2.1. Задачи кинетики.

Жесткость. Пусть в реакциях участвуют J различных частиц. Их концентрации обозначим через n_j, $1 \leqslant j \leqslant J$. Реакция происходит при одновременном столкновении нескольких частиц, причем в термоядерных реакциях столкновения являются парными. Число актов реакций пропорционально произведению концентраций сталкивающихся частиц. Поэтому в случае двухчастичных реакций уравнения для концентраций принимают следующий вид:

$$\frac{dn_j}{dt} = \sum (\pm) K_{jil}(T) n_i n_l, \quad 1 \leqslant j \leqslant J. \tag{1.1}$$

Здесь t – время. Знак "+" ставят для реакций с образованием j-го вещества, а "−" – для реакций со сгоранием. Суммируют по всем концентрациям в правой части. Коэффициенты пропорциональности $K(T)$ называются скоростями реакции. Они зависят от температуры T.

Система ОДУ (1.1) может одновременно содержать как жесткие (быстро затухающие), так и плохо обусловленные (быстро нарастающие) компоненты. Скорости разных реакций могут отличаться друг от друга на много порядков и быстро возрастают с увеличением температуры. Поэтому задачи кинетики являются жесткими.

Численные методы. Общие явные схемы в принципе не позволяют считать подобные задачи: счет может развалиться на первых же шагах из-за переполнения. На необходимость использования неявных методов впервые указал Далквист в 1952 г.

Исторически первыми методами для жестких задач были программы Гира для схем Гиршфельдера-Кертиса [1], [2], использующие дифференцирование назад. Они применяются до сих пор [3], но иногда дают сбои. Все схемы являются неявными, поэтому решение на новом слое находится каким-либо итерационным процессом, который при достаточно сильной жесткости перестает сходиться. Встроенный автоматический выбор шага ненадежен, так как фактическая точность может отличаться от заданной на несколько порядков. То же относится и к подавляющему большинству других алгоритмов с автоматическим выбором шага.

Наиболее употребительными схемами для жестких задач в целом и расчетов горения в частности являются явно-неявные схемы. Наилучшей одностадийной схемой из этого класса является комплексная схема Розенброка CROS [4]. Она имеет точность $O(\tau^2)$ и L_2-устойчивость и обеспечивает хорошие качественные свойства решения.

Однако у неявных схем есть существенный недостаток: на каждом шаге необходимо решать систему нелинейных алгебраических уравнений. Метод простых итераций хорошо сходится лишь для задач невысокой жесткости. Уже при умеренной

жесткости он становится недостаточно надежным, а при высокой жесткости и вовсе отказывает. Решение по методу Ньютона требует вычисления матрицы Якоби. На больших системах уравнений ее явное нахождение трудно реализуемо, а разностное неоправданно увеличивает трудоемкость и обычно требует 128-битовых вычислений.

Кроме того в методе Ньютона стоит вопрос о выборе нулевого приближения. Решение с предыдущего шага не гарантирует сходимости, а заметно более хороших способов выбора не найдено. В итоге неявные схемы оказываются сложными и трудоемкими. Вдобавок они менее надежны; например, иногда начальные концентрации нельзя задавать нулями, а нужно вводить малые числа и т.п. Отметим, что эти схемы являются общими и не используют специфику задачи.

Н. Н. Калиткин и В. Я. Гольдин предложили [5] специализированную явную схему, основанную на специфическом виде задачи (1.1). Эта схема обладала хорошими качественными свойствами (например, обеспечивала неотрицательность решения), но имела лишь первый порядок точности. Предлагались методы повышения порядка точности, но они оказались неудачными. В них решения имели очень большую немонотонность, из-за которой порядок точности фактически не повышался. Кроме того, они требовали, чтобы в начальный момент времени все концентрации были ненулевыми, что не соответствует физической постановке задачи.

1.2.2. Диагностика режимов с обострением.

Приложения. В нестационарных уравнениях с существенной нелинейностью возможно разрушение решения, то есть решение или его производные обращаются в бесконечность за конечное время. Разумеется, в точке сингулярности сама модель теряет применимость. Однако для того, чтобы убедиться в этом обстоятельстве, нужно уметь решить дифференциальное уравнение и установить наличие в нем сингулярности. Такие режимы, называемые ***режимами с обострением***, имеют важное практическое значение.

Например, в лазерном термоядерном синтезе в газовых мишенях при помощи амплитудно-модулированного импульса создается последовательность сходящихся сферических ударных волн, где амплитуда следующей волны больше, чем предыдущей. Параметры импульса подбираются так, чтобы волны догоняли друг друга в центре мишени. Это позволяет создавать сверхвысокие давление и температуру, что приводит к режиму с обострением [6].

Некоторые модели плазменных неустойчивостей, приводящих к пробою, также описываются уравнениями с разрушающимися решениями (см., например, [7]). Разрушение имеет место также в некоторых моделях нелинейного горения. При определенном виде коэффициента теплопроводности и правой части (источников) тепло выделяется быстрее, чем его отводит тепловая волна. Это приводит к локализации и неограниченному росту решения в каждой точке пространства, такой режим горения называется S-режимом [8]. Хотя в реальном процессе бесконечной температуры не возникает из-за быстрого выгорания, S-режим описывает его достаточно хорошо.

Методы диагностики. При аналитическом рассмотрении нелинейные уравнения исследуют на разрешимость, и если последняя оказывается локальной, то строят двусторонние оценки времени разрушения [7]. При этом находят не конкретный момент разрушения, а только диапазон, в котором он лежит. Получить сам момент разрушения этим методом намного труднее. Построить решение в явном виде удается только для простейших моделей, в более сложном случае их все равно приходится реализовывать численно.

В зарубежной литературе, начиная с 1990-х годов, предлагались расчетные методы диагностики, основанные на априорном теоретическом анализе (то есть наполовину численные, наполовину аналитические). Однако эти методы чрезвычайно сложны и каждый из них применим лишь к одной конкретной особенности.

Поэтому актуален вопрос – как диагностировать разрушение численно; то есть, не проводя теоретического анализа, определить тип особенности и вычислить ее параметры (момент времени t_0 и порядок q) с гарантированной точностью. Уравнение в частных производных сводится к небольшой системе ОДУ (несколько уравнений), если есть автомодельная замена; либо методом прямых к системе ОДУ большого порядка (несколько сотен уравнений). Таким образом, вопрос о численной диагностике разрушения сводится к диагностике особенности типа полюс у системы ОДУ. Впервые такой вопрос был поставлен на семинаре академика Г. И. Марчука в Институте вычислительной математики РАН в 2003 г.

Первая универсальная процедура численной диагностики сингулярностей была предложена в [9], [10]. Она основывалась на анализе характера сходимости при сгущении сеток по времени. Эта процедура позволяла диагностировать особенности типа полюс и логарифмический полюс, а также определять разрывы старших производных, сохраняющие непрерывность решения. Однако она обладала рядом недостатков.

1° Эта методика была разработана только для комплексной схемы Розенброка (CROS) в переменной t. Это неудобно потому, что расчет по явно-неявным схемам может выйти за момент t_0, где точное решение не существует, а численное может выйти за пределы представимых чисел. Гораздо надежнее аргумент l (длина дуги), на что также указал Г. И. Марчук в 2006 году. Кроме того, для других схем эта методика оказалась непригодной. ***2°*** Не было предложено аккуратных оценок погрешности для найденных t_0 и q. ***3°*** Процедура была сложной и громоздкой.

Жесткость. При численном решении сингулярность можно рассматривать как участок резкого изменения решения. Поэтому задачи с разрушениями также следует относить к жестким.

1.3 Жесткие краевые задачи

1.3.1. Уравнение Пуассона.

Разностные методы. Решение многомерных эллиптических уравнений разностными методами приводит к системам линейных алгебраических уравнений $Au =$

b огромной размерности [11]. Решение таких систем является нетривиальной проблемой, которой посвящена обширная литература (см., например, [12]).

Нахождение решения с гарантированной точностью имеет два основных аспекта. ***1°*** Для построения апостериорных оценок погрешности нужно применять процедуры сгущения сеток по Ричардсону. Для получения хороших точностей нужно сгущать сетки многократно, что приводит к системам большого порядка. ***2°*** На каждой сетке сеточное решение должно быть найдено с точностью, достаточной для применения метода Ричардсона. Поэтому от итерационных методов требуется очень высокая точность при умеренном числе итераций. Само решение требуется находить с точностью вплоть до ошибок компьютерного округления.

На произвольных сетках получаются линейные системы достаточно общего вида. Для них работоспособны только итерационные методы сопряженных направлений [13], сходящиеся довольно медленно. Для получения 12 верных знаков требуется число итераций $S \approx 10\sqrt{\lambda_{\max}/\lambda_{\min}} \approx 10N$, где $\lambda_{\min}$, $\lambda_{\max}$ – границы спектра, N – среднее число узлов по каждой координате. Для хорошей аппроксимации требуются большие $N \sim 300 \div 1000$, что приводит к неприемлемо большим S. Поэтому нужно искать ограничения, при которых возможно построение более быстрых, но достаточно общих методов.

Постановка задачи. Ограничимся эллиптическими задачами без смешанных производных

$$Lu \equiv \sum_{\alpha} \frac{\partial}{\partial x_\alpha} \left(k_\alpha(\mathbf{r}) \frac{\partial u}{\partial x_\alpha} \right) = -f(\mathbf{r}). \tag{1.2}$$

В частности, это может быть уравнение теплопроводности или уравнение электростатики. В первом случае коэффициенты k_α имеют смысл компонент тензора теплопроводности, во втором – диэлектрической проницаемости. При этом k_α будем считать переменными, а сетки – прямоугольными и неравномерными. Такая постановка достаточно содержательна.

Если предполагать обобщения на слоистые среды (т.е. разрывные коэффициенты), то надо пользоваться консервативными схемами. Для гладких коэффициентов консервативные схемы не ухудшают результата, и их считают предпочтительными. Будем предполагать, что коэффициенты непрерывны вместе со своими вторыми производными кроме, быть может, отдельных узлов сетки (границ слоев). Тогда классическая консервативная схема имеет следующий вид:

$$(\Lambda_x + \Lambda_y + \Lambda_z)\, u = -f. \tag{1.3}$$

Здесь трехточечный оператор

$$(\Lambda_x u)_n = \frac{2}{h_{x,n+1/2} + h_{x,n-1/2}} \left[\frac{k_{x,n+1/2}}{h_{x,n+1/2}} (u_{n+1} - u_n) - \frac{k_{x,n-1/2}}{h_{x,n-1/2}} (u_n - u_{n-1}) \right], \tag{1.4}$$
$$h_{x,n+1/2} = x_{n+1} - x_n;$$

в (1.4) оставлен только индекс по координате x. Выражения для Λ_y и Λ_z аналогичны.

Приведем некоторые известные прямые и итерационные методы решения разностных уравнений и опишем границы их применимости [12], [14], [15].

Быстрое преобразование Фурье. Этот метод является самым быстрым из известных прямых методов. Он применим к задаче Дирихле в прямоугольнике (прямоугольном параллелепипеде), причем только на равномерных сетках и при постоянных коэффициентах k_α. Метод экономичен, если число узлов N_α является произведением малых целых чисел. Он особенно эффективен при $N_\alpha = 2^{r_\alpha}$. Условия применимости этого метода к задачам математической физики являются слишком стеснительными, но в задачах обработки изображений он применяется широко.

Заметим также, что для всех прямых методов не возникает вопроса о точности сходимости итераций. Имеются только ошибки округления, которые невелики, поскольку в случае эллиптических уравнений матрица A хорошо обусловлена.

Нечетно-четная редукция. Этот метод есть модификация метода исключения Гаусса, в котором исключение неизвестных происходит в специальном порядке. Сначала исключают неизвестные с нечетными номерами n, затем из остальных уравнений – с номерами n, равными произведению 2 на нечетное число, затем – 4 на нечетное число и т.д.

Применительно к разностным эллиптическим задачам метод нечетно-четной редукции имеет такую же трудоемкость, как быстрое преобразование Фурье. Он также применим лишь при постоянных k_α и равномерных сетках с числом узлов 2^{r_α}.

Метод сопряженных градиентов. Этот метод заключается в построении полного ортогонального базиса, минимизирующего квадратичную форму $\Phi(u) = (Au - 2b, u)$ в пространстве $u \in R_M$. На практике M настолько велико, что вычисления не успевают дойти до исчерпывания. Однако получить требуемую точность ε можно уже за разумное число итераций. Теоретическая оценка сходимости имеет вид

$$S \approx \frac{N}{\pi} \ln \frac{1}{\varepsilon} = O(N). \tag{1.5}$$

Такая скорость считается лучшей для известных общих методов, но при больших $N > 1000$ алгоритм становится слишком трудоемким.

Контролировать сходимость этого метода можно косвенно по невязке. Но оценка погрешности по невязке имеет мажорантный характер, причем константа в этой мажорантной оценке неизвестна и велика. Поэтому аккуратно оценить погрешность практически невозможно.

Метод сопряженных градиентов применим к эрмитовым знакоопределенным матрицам, то есть его можно применять к широкому классу задач: уравнение со смешанными производными, криволинейная граница, непрямоугольные сетки и т.д. Каждый шаг метода устойчив, и он не требует задания границ спектра и числа итераций S. Кроме того, метод имеет простую одношаговую реккурентную форму записи, исключающую накопление ошибок округления, и легко распараллеливается.

Счет на установление. Эллиптические уравнения обычно рассматривают как стационарный предел для соответствующего параболического уравнения $u_t = Lu + f$. Такой прием называется ***счетом на установление***. Характерные времена затухания низшей и высшей пространственных гармоник различаются в $\sim N^2$ раз, где

$N \gg 1$ – число интервалов пространственной сетки по одной переменной. Поэтому данная задача относится к жестким. Свойства такого итерационного процесса зависят от записи разностной схемы и от используемого набора шагов по времени $\{\tau_s\}$. При использовании неявных схем возникает также проблема факторизации.

Чебышевский набор шагов. Счет на установление можно проводить по явной схеме. Она не требует факторизации и единообразно пишется при любом числе измерений. Область может иметь сложную границу, сетки могут быть неструктурированными, а уравнение может содержать смешанные производные. Каждая итерация нетрудоемка, а метод легко распараллеливается.

Число шагов, нужное для достижения точности ε, равно

$$S = \frac{\ln(1/\varepsilon)}{2}\sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} = O(N). \tag{1.6}$$

Величины $1/\tau_s$ являются корнями многочлена Чебышева 1-го рода степени S, построенного на отрезке $[\lambda_{\min}, \lambda_{\max}]$. Для произвольной области получить оценки $\lambda_{\min}$ и $\lambda_{\max}$ крайне трудно, а требуемое число шагов S весьма чувствительно к точности этих оценок. При этом S нужно задавать еще до начала расчета, что неудобно.

Логарифмический набор шагов. Для системы (1.3), (1.4) при $k_{\alpha,n} \neq \text{const}$, $h_n \neq \text{const}$ наиболее быстро сходящимся процессом является счет на установление по эволюционно факторизованной схеме [16], [17] с набором шагов, выбранным в логарифмической шкале [18]. Хорошие результаты дает логарифмически равномерный набор $\ln \tau_s = \text{const}$. Для него получена априорная оценка сходимости $S \approx 10 \ln(\lambda_{\max}/\lambda_{\min}) \approx 20 \ln N$. Это число остается умеренным даже для $N \sim 1000$. Однако способов оценки фактической точности не предлагалось. Такая скорость сходимости является лучшей среди итерационных методов и эквивалентна трудоемкости быстрого преобразования Фурье.

1.3.2. Диффузия в пограничных слоях.

Приложения. Существует ряд важных прикладных задач, в которых основную роль играет малая диффузия из одной области в другую. Примерами являются ***1°*** насыщение поверхностного слоя стали азотом, что приводит к упрочнению; ***2°*** диффузия магнитного поля в сжимающую оболочку в магнитокумулятивных генераторах сверхсильных полей; ***3°*** поверхностный индукционный нагрев при закалке стальных деталей; ***4°*** поверхностное легирование полупроводников донорами и акцепторами. К такому же типу задач можно отнести ***5°*** скалярную задачу дифракции высокочастотного электромагнитного поля на металлических поверхностях.

Постановка задачи. Перечисленные задачи описываются уравнением диффузии с малым параметром

$$\begin{cases} \mu^2 \mathrm{div}\,(k(\mathbf{r})\,\mathrm{grad} u) - \varkappa(\mathbf{r}) u = -f(\mathbf{r}); \quad k(\mathbf{r}) > 0,\ \varkappa(\mathbf{r}) > 0; \quad \mathbf{r} \in G; \\ u(\mathbf{r}) = \varphi(\mathbf{r}), \quad \mathbf{r} \in \Gamma. \end{cases} \tag{1.7}$$

Внешняя среда заменяется постановкой граничных условий на границе Γ области G, а $\mu \ll 1$. В общем случае граница области может быть криволинейной; это требует введения неструктурированных треугольных сеток.

Среда может быть неоднородной и даже анизотропной. В последнем случае $k(\mathrm{r})$ – это тензор, а в уравнении (1.7) появляются смешанные производные. Если вещество изотропно, либо главные оси анизотропного кристалла параллельны осям координат, то смешанные производные отсутствуют. Тогда уравнение (1.7) записывается в более простом виде

$$\mu^2 \sum_\alpha \frac{\partial}{\partial x_\alpha}\left(k_\alpha(\mathbf{r})\frac{\partial u}{\partial x_\alpha}\right) - \varkappa(\mathbf{r})u = -f(\mathbf{r}). \tag{1.8}$$

Это уравнение можно рассматривать как некоторое обобщение уравнения Гельмгольца на неоднородную среду; оно переходит в уравнение Гельмгольца при $k_\alpha(\mathbf{r}) \equiv 1$, $\varkappa(\mathbf{r}) = \mathrm{const}$.

Структура решения. В решении задачи (1.7) – (1.8) обычно выделяют регулярную внутреннюю часть и узкий пограничный слой шириной $\sim \mu$, в пределах которого решение резко меняется. Указанные участки решения имеют большую разномасштабность, поэтому данная задача относится к жестким. Численный расчет таких задач труден. Еще труднее получить гарантированную оценку погрешности.

Выбор сеток. Очевидно, для хорошей точности расчета разностная схема должна содержать достаточно много узлов в пограничном слое, то есть иметь очень малый шаг вблизи границы. Поэтому использование равномерных сеток привело бы к неприемлемой трудоемкости расчетов даже в одномерном случае, не говоря уже о трехмерном. Таким образом, важнейшим становится вопрос о хорошем подборе неравномерной сетки.

Существуют алгоритмы построения адаптивных треугольных сеток [19], но они сложны и не очень надежны. Проблема сгущения подобных сеток фактически не разработана.

Ситуация упрощается, если ограничиться рассмотрением прямоугольных сеток. В этом случае Н. С. Бахваловым была предложена идея использования произведения одномерных квазиравномерных сеток [20]. Он привел пример сетки, дававшей точность $O(N^{-2})$ [21], но она была далека от оптимальной. Г. И. Шишкин предложил [22] кусочно-равномерные сетки, адаптированные к пограничному слою и регулярной части решения. Для них доказана [23] сходимость $O(N^{-2}\ln^4 N)$, что практически неотличимо от $O(N^{-2})$.

При этом все теоретические оценки мажорантны и могут сильно превышать фактическую погрешность. Использование этих оценок в практических расчетах заставляет завышать N, что в многомерном случае существенно увеличивает трудоемкость расчетов.

1.4 Кинетика термоядерных реакций

Важным прикладным примером задач кинетики является протекание термоядерных реакций в лазерных мишенях и токамаках. Здесь возникает еще одна вспомогательная проблема: для расчетов требуются достоверные данные о скоростях реакций. Скорость реакции равна свертке сечения реакции $\sigma(E)$ с функцией распределения по

энергиям. Строго говоря, процессы горения являются неравновесными, но их функцию распределения можно разумно приблизить распределением Максвелла (в предположении, что в среде имеется локальное термодинамическое равновесие). Сечения реакций измеряются экспериментально, поэтому нахождение скоростей реакций сводится к обработке экспериментальных данных для $\sigma(E)$.

1.4.1. Обработка экспериментальных данных. Нередко важные физические эксперименты проводятся по существу за пределами возможностей экспериментальной техники. Систематические и случайные ошибки оказываются настолько большими, что совокупность экспериментальных точек различных авторов выглядит как облако, размытое вокруг некоторой кривой.

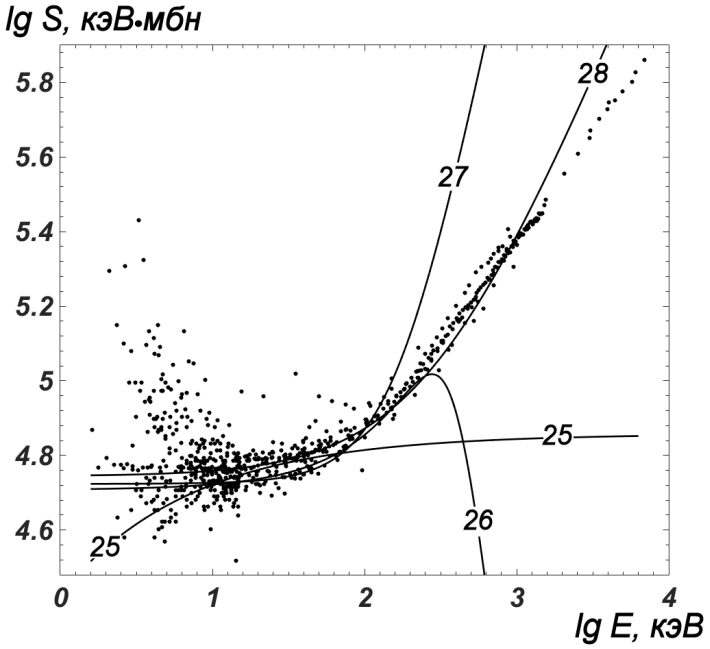

Рис. 1.1. S-фактор для реакции $\mathrm{D} + \mathrm{D} \to \mathrm{p} + \mathrm{T}$; точки – экспериментальные значения [24], линии – различные аппроксимации, цифры около линий соответствуют номеру ссылки по списку литературы: [25] – Арнольд и др. (1954), [26] – Козлов (1957), [27] – Краусс и др. (1973), [28] – Браун и др. (1990).

Характерным примером может служить рис. 1.1, где показана зависимость сечения термоядерной реакции $\mathrm{D} + \mathrm{D} \to \mathrm{p} + \mathrm{T}$ от энергии в специфических координатах; величина $S(E)$ называется S-фактором (см. п. 5.3.2). Видно, что данные отдельных авторов различаются до 6 раз! В то же время для уверенных расчетов кинетики реакций необходимо знать их скорости с точностью в несколько процентов. Указанная реакция является одной из важнейших в проблеме управляемого термоядерного синтеза (УТС), так что обрабатывать подобные кривые необходимо.

Обычно физики используют 2 приема. Во-первых, из физического смысла задачи они подбирают специфические переменные, в которых кривая выглядела бы наиболее простым образом.

Во-вторых, полученную кривую аппроксимируют некоторой аналитической зависимостью, содержащей не слишком много подгоночных параметров. Значения этих

параметров определяют методом наименьших квадратов. Успех такого подбора зависит от того, насколько удачно угадан вид аппроксимирующей формулы. Примеры аппроксимаций приведены на рис 1.1. Видно, что при $E > 300 \div 500$ кэВ они достаточно сильно расходятся и между собой, и с экспериментами. Более подробный анализ этого будет дан ниже.

Если число подгоночных параметров много меньше числа экспериментальных точек, и при этом достигнута приемлемая точность аппроксимации, то вид аппроксимирующей формулы можно считать удачным. Особенно ценны формулы, воспроизводящие априорно известные физические закономерности. Для них можно надеяться на аппроксимацию небольшим числом параметров. Однако такие формулы нечасто удается предложить.

Если же для аппроксимации требуется число параметров, сравнимое с числом экспериментальных точек, то вид формулы неудачен, а сама аппроксимация вряд ли будет надежной. При этом экстраполяция полученной аппроксимации за пределы экспериментального аргумента может привести к большим ошибкам.

1.4.2. Сечения термоядерных реакций. При низких энергиях их часто аппроксимируют (см., например, [25], [29], [30]) формулой Гамова [31]

$$\sigma(E) \approx \frac{A}{E} \exp\left\{-\frac{B}{\sqrt{E}}\right\}, \quad A, B = \text{const}. \tag{1.9}$$

Это формула следует из квазиклассического выражения для проницаемости кулоновского барьера. Для реакций, у которых сечение имеет максимум, применяют [32], [33], [34], [35], [36], [37], [38] формулу Брейта-Вигнера [39]

$$\sigma(E) \approx \frac{\Gamma_1^2}{(E - E_0)^2 + \Gamma_2^2}, \quad E_0, \Gamma_{1,2} = \text{const}. \tag{1.10}$$

В. А. Давиденко предложил произведение формул (1.9) и (1.10) для того, чтобы разумно описать поведение $\sigma(E)$ при низких энергиях и вблизи максимума [40].

Однако область применимости формулы Гамова ограничивается диапазоном $E < 30 \div 50$ кэВ, а формула Брейта-Вигнера выводилась для описания узких резонансов, и ее справедливость для сечений с широким максимумом является спорной. Более надежной теоретической формулы, которая была бы справедлива в широком диапазоне энергий, пока не предложено.

Поэтому нередко экспериментаторы приближают $S(E)$ полиномиальными зависимостями (например, линейными [28] или квадратичными [27]). Б. Н. Козловым были предложены [26] более сложные формулы с несколько бóльшим числом подгоночных параметров (до 6), которые достаточно широко используются в расчетах мишеней УТС (см. [41] и библиографию там).

1.4.3. Регуляризация. Задачу обработки эксперимента можно рассматривать как некорректную. Для этого к задаче о наилучшем приближении экспериментальных данных добавляют некоторый регуляризатор.

Традиционно полученная задача на минимум сводится к решению дифференциального уравнения для аппроксимирующей функции. Правая часть этого уравнения задана с большими экспериментальными ошибками, так что уравнение будет стохастическим, причем роль стохастичности весьма велика. Порядок этого уравнения вдвое выше, чем порядок максимальной производной, входящей в регуляризатор.

Такое уравнение требует соответствующего числа дополнительных условий. Формальная постановка таких краевых условий обычно приводит к заметному отличию регуляризованного решения от истинного вблизи границ интервала, а решение краевой задачи для уравнения высокого порядка само по себе представляет проблему.

1.5 Общая характеристика работы

1.5.1. Актуальность темы исследования. В настоящее время чрезвычайно актуальной является проблема поиска новых источников энергии и повышения эффективности имеющихся. Путями ее решения являются оптимизация процессов горения с точки зрения энерговыхода и осуществление управляемого термоядерного синтеза (УТС). Для решения этих проблем требуются надежные численные методы, позволяющие проводить расчеты с высокой гарантированной точностью (не хуже 0.1%). Для моделирования процессов в мишенях УТС требуются также достоверные данные о сечениях и скоростях реакций.

Другой важной проблемой является расчет электродинамических конструкций, в которых поле лишь незначительно проникает внутрь проводника. Примерами таких процессов являются диффузия магнитного поля в сжимающую оболочку магнитокумулятивных генераторов сверхмощных магнитных полей и сверхсильных токов, поверхностный индукционный нагрев при закалке стальных деталей, поверхностное легирование полупроводников донорами и акцепторами и многие другие.

Все указанные задачи являются актуальными для современной науки и техники. Их необходимо рассчитывать экономично и с высокой точностью.

1.5.2. Степень разработанности темы исследования. Текущее состояние исследуемых вопросов и обзор основных публикаций дан в п. 1.2 – 1.3.

Из них видно, что трудной проблемой является расчет кинетики реакций. Известные методы ненадежны либо чрезмерно трудоемки. В данной работе построен специализированный численный метод, обладающий очень малой трудоемкостью, высокой точностью и надежностью и обеспечивающий физически правильное качественное поведение решения. Таким образом, эту проблему можно считать закрытой.

Разработанный метод применяется к моделированию реальной кинетики 4 термоядерных реакций. В рамках данной задачи возникает вспомогательная проблема: определение зависимости скоростей этих реакций от температуры. Последнее сводится к обработке экспериментальных данных по сечениям этих реакций, измеренных со значительными погрешностями. Существующие методы хорошо работают в тех условиях, когда теоретическая физика уверенно предсказывает качественный характер

искомой зависимости. Если же он неизвестен, то методы, обеспечивающие хорошую точность, отсутствуют. В данной работе предложен метод, надежно работающий в указанной ситуации. Получены аппроксимации сечений и скоростей термоядерных реакций, наиболее важных для УТС. Точность этих аппроксимаций составляет от 1 до 4%. Поэтому данную проблему можно считать решенной.

В расчетах горения термоядерных мишеней и при исследовании плазменных неустойчивостей, приводящих к пробою, важную роль играет анализ сингулярностей решений нелинейных уравнений в частных производных. Построен метод, который позволяет проводить численную диагностику сингулярности с апостериорной асимптотически точной оценкой погрешности. Рассмотрены основные типы сингулярностей, возникающих на практике, поэтому этот вопрос также можно считать практически решенным.

Для решения эллиптических уравнений без смешанных производных в прямоугольных областях ранее был предложен сверхбыстрый логарифмический счет на установление. Однако оставался открытым вопрос о выборе наилучшего набора шагов и об оценках погрешности итераций. В данной работе построен практически неулучшаемый логарифмический набор и предложена процедура упорядочивания его шагов, дающая апостериорную асимптотически точную оценку сходимости итерационного процесса. Таким образом, вопрос об экономичном решении уравнений указанного типа в прямоугольных областях оказывается решенным.

Для сингулярно возмущенных краевых задач в прямоугольных областях ранее предлагались сетки, адаптированные к пограничным слоям. Однако они оказывались далекими от оптимальных. Для разностных схем на этих сетках строились априорные оценки сходимости. В данной работе предложена квазиравномерная сетка, детально передающая все характерные области решения. Это решило вопрос о построении адаптивной квазиравномерной сетки.

Показано, что для таких задач можно получить апостериорную асимптотически точную оценку погрешности по методу Ричардсона и, опираясь на нее, подтвердить фактический порядок точности разностной схемы. Это позволило закрыть вопрос об исследовании сходимости в сингулярно возмущенных задачах для прямоугольных областей.

1.5.3. Цели и задачи. Целями данной работы являются

1. Разработка технологий моделирования кинетики реакций с гарантированной точностью $0.1 \div 0.01\%$. Проведение моделирования кинетики термоядерных реакций.
2. Нахождение скоростей термоядерных реакций с точностью несколько процентов.
3. Разработка надежных методов диагностики сингулярностей с гарантированной точностью для систем ОДУ.
4. Разработка экономичных методов решения эллиптических уравнений (в том числе сингулярно возмущенных) для достаточно широкого класса многомерных

задач (переменные коэффициенты, неравномерные прямоугольные сетки).

1.5.4. Научная новизна. В диссертации впервые предложены и обоснованы следующие новые результаты.

С использованием предложенных технологий обработки экспериментов для 4 термоядерных реакций, наиболее важных для управляемого синтеза в газовых мишенях, найдены аппроксимации сечений и скоростей реакций. Погрешность аппроксимаций для сечений не превышает 1%, а для скоростей реакций составляет $1 \div 4\%$, что в 5 раз точнее использовавшихся ранее. Это существенно для моделирования процессов в термоядерных мишенях.

Проведено моделирование кинетики этих реакций, и оценены условия, необходимые для возникновения самоподдерживающегося горения. Показано, что критерий Лоусона должен быть на 3 порядка больше, чем считалось ранее.

Разработана специализированная экономичная технология моделирования кинетики реакций. Одновременно с решением она предоставляет гарантированную оценку его математической погрешности. В ней используется явная численная схема, трудоемкость которой очень мала. Эта схема имеет более высокий порядок точности (второй) и одновременно является более надежной, чем ранее известные схемы.

Разработана технология обработки экспериментальных данных с нахождением дисперсии аппроксимирующей кривой. Задача рассматривается как некорректная. Решение представляется методом двойного периода, для регуляризации используется стабилизатор А. Н. Тихонова с квадратом второй производной. Это позволяет подавить нефизичные осцилляции и получать высокую точность, хорошо передавая форму экспериментальной кривой. Такой подход позволяет единообразно решать широкие классы задач.

Предложен простой и надежный метод диагностики сингулярностей (полюс, логарифмический полюс, смешанная особенность) для систем обыкновенных дифференциальных уравнений (ОДУ). Он позволяет вычислять параметры этих особенностей с апостериорной асимптотически точной оценкой погрешности. Метод работает при аргументе длина дуги, который оптимален при моделировании таких задач. Метод позволяет исследовать модели, описываемые нелинейными уравнениями в частных производных, поскольку они сводятся методом прямых к системам ОДУ огромного порядка. С использованием этого метода исследована модель S-режима нелинейного горения.

Для моделирования процессов, описываемых эллиптическими уравнениями, предложен новый линейно-тригонометрический набор шагов для счета на установление. Коэффициенты уравнения могут быть переменными, а прямоугольные сетки – неравномерными. Набор строится в логарифмической шкале и дает экспоненциальную скорость сходимости, что является теоретическим пределом. Предложенный набор уменьшает число итераций в 1.5 раза по сравнению с логарифмически равномерным. Он более прост и надежен, чем известные ранее наборы. Разработана процедура упорядочивания шагов логарифмического набора, аналогичная методу Ричардсона

и позволяющая найти апостериорные асимптотически точное значение погрешности итераций. Ранее существовали только мажорантные оценки точности по невязке, отличающиеся от точных на несколько порядков.

Для моделирования процессов диффузии в пограничных слоях, описываемых сингулярно возмущенным уравнением Гельмгольца в прямоугольной области, предложена адаптивная квазиравномерная сетка, обеспечивающая высокую точность даже при очень тонких пограничных слоях ($\sim 10^{-7}$ от размеров области) уже на скромных сетках с небольшим числом узлов (до 500 по каждому направлению). Она позволяет находить апостериорную асимптотически точную оценку погрешности по методу Ричардсона и устанавливать порядок фактической точности. Это существенно экономичнее, чем использование мажорантных априорных оценок.

На основе предложенных методов впервые разработаны 3 пакета программ на языке Matlab (Kinetic для расчета кинетики реакций, SiDiaG для диагностики сингулярностей у систем ОДУ, SuFaReC для решения задачи Дирихле для обобщенного уравнения Гельмгольца). Все расчеты проводятся одновременно с нахождением апостериорного асимптотически точного значения погрешности. Эффективность пакетов подтверждена большим количеством численных экспериментов, которые позволили верифицировать работу соответствующих вычислительных технологий (численных методов и их программных реализаций). Пакет SiDiaG является первым математическим обеспечением, позволяющим численно диагностировать разрушение решения систем ОДУ. Ранее программ с такой функциональностью не предлагалось.

Таким образом, перечисленные задачи рассмотрены на всех уровнях: проведено моделирование реальных задач и получены физически значимые результаты, построены качественно новые численные методы, разработаны комплексы актуальных прикладных программ.

1.5.5. Теоретическая и практическая значимость работы. Полученные в работе аппроксимации для сечений и скоростей термоядерных реакций значительно точнее известных ранее. Это существенно для моделирования процессов в мишенях управляемого синтеза. Предложенные математические методы качественно превосходят по точности, надежности и эффективности ранее известные алгоритмы и представляют интерес для широкого круга исследователей при решении прикладных задач. Разработанные пакеты программ должны найти широкое применение для исследовательских расчетов, а также как прототипы программных комплексов для производственных расчетов. Ниже перечислены организации, для которых будут полезны результаты, полученные в данной работе.

Новые численные методы для задач кинетики должны найти широкое применение как часть больших газодинамических пакетов программ для расчетов химической кинетики, проводимых в Институте проблем механики им. А. Ю. Ишлинского РАН, на Физическом и Химическом факультетах МГУ им. М. В. Ломоносова, в Институте прикладной математики им. М. В. Келдыша РАН и в других организациях.

Вместе с новыми выражениями для скоростей термоядерных реакций они будут

исключительно полезны в расчетах мишеней для УТС, проводимых в федеральных ядерных центрах (Саров и Снежинск), ИПМ им. М. В. Келдыша РАН, Физическом институте академии наук им. П. Н. Лебедева, Объединенном институте ядерных исследований, Национальном исследовательском центре "Курчатовский институт" и других.

Методы диагностики полюсов и других особенностей должны стать надежным инструментом для исследования сингулярностей, проводимых на кафедре математики Физического факультета МГУ им. М. В. Ломоносова, в Математическом институте академии наук им. В. А. Стеклова, в Московском институте электронной техники и в других организациях.

Построенные методы расчета и апостериорного теоретического анализа для сингулярно возмущенных краевых задач следует рассматривать как стандартные вычислительные технологии, которые должны стать обязательными для прикладных расчетов, проводимых в широком круге организаций: соответствующие факультеты МГУ (Физический, Факультет вычислительной математики и кибернетики) и других университетов, Институте математического моделирования Уральского отделения РАН, Вычислительном центре РАН, ИПМ им. М. В. Келдыша РАН и др.

1.5.6. Методология и методы исследования. При разработке математических алгоритмов использовались традиционные методы вычислительной математики. Применялись как формальный, так и эвристический подходы. Последний позволил создать эффективные алгоритмы в тех областях, где формальное исследование проблематично.

Большое внимание уделялось обоснованию сходимости методом сгущения сеток и построению фактических оценок погрешности. Работа всех алгоритмов проверялась на представительных тестовых задачах, поэтому построение тестов также было важным аспектом.

При разработке прикладных пакетов был использован язык высокого уровня Matlab, совместимый со свободно распространяемой средой для математических вычислений GNU Octave. Она позволяет легко визуализировать результаты расчетов.

1.5.7. Положения, выносимые на защиту. На защиту выносятся следующие положения:

1. Разработаны и реализованы экономичные численные алгоритмы решения задач кинетики, диффузии и эффективный метод численного обнаружения и диагностики сингулярностей в ОДУ, работающий в автоматическом режиме. Разработан и успешно применен метод обработки экспериментальных данных с нахождением дисперсии аппроксимирующей кривой.
2. Разработан простой итерационный метод решения многомерных эллиптических уравнений с логарифмической сходимостью, что является теоретическим пределом. Одновременно с решением метод вычисляет асимптотически точную оценку погрешности. Метод позволяет эффективно решать сингулярно возмущен-

ные уравнения.

3. Создано три пакета прикладных программ для решения указанных выше задач. Эффективность пакетов подтверждена численными экспериментами.
4. Разработаны новые математические методы моделирования основных ядерных реакций синтеза изотопов водорода, получены наиболее точные на настоящий момент аппроксимации сечений и скоростей реакций.

1.5.8. Степень достоверности и апробация результатов. Достоверность и надежность разработанных математических методов гарантируется следующим. ***1°*** Все методы проверялись на представительных тестовых задачах с известным точным решением. ***2°*** Все расчеты проводились на сгущающихся сетках с апостериорной оценкой погрешности по методу Ричардсона и контролем фактического порядка точности. В ходе таких расчетов проверяется сходимость сеточного решения к некоторой предельной функции. Согласно известным фундаментальным теоремам Рябенького-Филлипова, эта предельная функция является точным решением. Это обеспечивает математическую точность на уровне ошибок округления компьютера.

Надежность обработки экспериментальных данных следует из большого объема анализируемого материала, а также из физически осмысленного поведения аппроксимирующей кривой. Вычисленные оценки точности аппроксимаций для сечений контролируются по соответствию известным физическим закономерностям (например, формула Гамова). Они подтверждены наиболее надежными экспериментальными данными, измеренными с наибольшей точностью.

1.5.9. Структура и объем работы. Диссертация состоит из введения, пяти глав и заключения. Общий объем диссертации: страниц 159, рисунков 58, таблиц 7. Список литературы включает 57 наименований.

1.5.10. Личный вклад автора. Все результаты диссертации получены автором лично при научном руководстве Н. Н. Калиткина. Автор самостоятельно предложил и разработал метод обнаружения и диагностики сингулярностей у систем ОДУ и провел исследование квадратурных формул Эйлера-Маклорена произвольных порядков точности. В остальных задачах автору принадлежит разработка деталей алгоритмов, программная реализация и проведение тестовых и прикладных расчетов.

1.5.11. Апробация результатов. Результаты работы докладывались на международной конференции “XVII Харитоновские тематические научные чтения” (Саров, 23-27 марта 2015), научно-координационной сессии “Исследования неидеальной плазмы” (Москва, 27-28 ноября 2015), международной конференции “Современные проблемы математической физики и вычислительной математики” к 110-летию со дня рождения академика А. Н. Тихонова (Москва, 31 октября – 3 ноября 2016), на международной конференции “Современные проблемы вычислительной математики и математической физики” памяти академика А.А. Самарского к 95-летию со дня рождения

(Москва, 16-17 июня 2014), на международной конференции “13th Annual Workshop on Numerical Methods for Problems with Layer Phenomena” (Москва, 6-9 апреля 2016), на международной конференции “Days on Diffraction 2015” (Санкт-Петербург, 25-29 мая 2015), на VII и VIII международной конференции “Акустооптические и радиолокационные методы измерений и обработки информации” (Суздаль, 14-17 сентября 2014 и 20-23 сентября 2015), на международном научном семинаре “Актуальные проблемы математической физики” (Москва, 28-29 ноября 2014), на XV Всероссийской школе-семинаре “Физика и применение микроволн” имени профессора А.П. Сухорукова (Москва, Можайск, 1-6 июня 2015), на научной конференции “Ломоносовские чтения” (Москва, 18-27 апреля 2016), на научной конференции “Тихоновские чтения” (Москва, 27-31 октября 2014), на семинарах Кафедры математики Физического факультета МГУ им. М. В. Ломоносова (21 декабря 2016, 2 марта 2016, 2 октября 2013), на семинаре Кафедры вычислительной математики Факультета ВМК (11 декабря 2013) и на конференции Совета молодых ученых ИПМ им. М. В. Келдыша РАН (6 ноября 2015).

1.5.12. Публикации. По теме диссертации всего опубликовано 13 работ в журналах, входящих в перечень ВАК: Доклады академии наук – 3, Математическое моделирование – 6, Журнал вычислительной математики и математической физики – 1, Известия РАН. Серия физическая – 1, Препринты ИПМ им. М.В. Келдыша – 2, а также 14 работ в сборниках трудов международных и Всероссийских конференций.

1. *А.А. Белов.* Численное обнаружение и исследование сингулярностей решения дифференциальных уравнений // ДАН. **468**:1 (2016), 21–25.
2. *А.А. Белов, Н.Н. Калиткин.* Обработка экспериментальных кривых регуляризованным методом двойного периода // ДАН. **470**:3 (2016), 266–270.
3. *Н.Н. Калиткин, А.А. Белов.* Аналог метода Ричардсона для логарифмически сходящегося счета на установление. // ДАН. **452**:3 (2013), 261–265.
4. *А.А. Белов.* Численная диагностика разрушения решений дифференциальных уравнений // ЖВМиМФ. **57**:1 (2017), 91–102.
5. *А.А. Белов.* О коэффициентах квадратурных формул Эйлера-Маклорена // Матем. моделирование. **25**:6 (2013), 72–79.
6. *А.А. Белов, Н.Н. Калиткин.* Эволюционная факторизация и сверхбыстрый счет на установление. // Матем. моделирование. **26**:9 (2014), 47–64.
7. *А.А. Белов, Н.Н. Калиткин, Л.В. Кузьмина.* Моделирование химической кинетики в газах // Матем. моделирование, **28**:8 (2016), 46–64.
8. *А.А. Белов, Н.Н. Калиткин.* Численное моделирование задач с пограничным слоем // Матем. моделирование. **27**:11 (2015), 47–55.
9. *А.А. Белов, Н.Н. Калиткин, Л.В. Кузьмина.* Сравнение высокоустойчивых форм итерационных методов сопряженных направлений // Матем. моделирование. **27**:9 (2015), 110–136.
10. *А.А. Белов, Н.Н. Калиткин.* Сверхбыстрый метод с гарантированной точностью для эллиптических уравнений в прямоугольной области // Матем. моде-

лирование. **27**:7 (2015), 37–43.

11. *А.А. Белов, Н.Н. Калиткин.* Сеточные методы решения задач с пограничным слоем // Известия РАН. Серия физическая. **79**:12 (2015), 1655–1659.
12. *А.А. Белов.* Программы SuFaReC для сверхбыстрого расчета эллиптических уравнений в прямоугольной области // Препринты ИПМ им. М.В. Келдыша. 44 (2015), 1–12.
http://library.keldysh.ru/preprint.asp?id=2015-44
13. *А.А. Белов, Н.Н. Калиткин.* Эволюционная факторизация и сверхбыстрый счет на установление // Препринты ИПМ им. М.В. Келдыша. 69 (2013), 1–32.
http://library.keldysh.ru/preprint.asp?id=2013-69
14. *А.А. Белов, Н.Н. Калиткин.* Численное решение задач Коши с контрастными структурами // Тезисы докладов международной конференции “Современные проблемы математической физики и вычислительной математики” к 110-летию со дня рождения академика А.Н. Тихонова, 2016, с. 87.
15. *А.А. Белов, Н.Н. Калиткин.* Регуляризованный метод двойного периода для обработки экспериментов // Тезисы докладов международной конференции “Современные проблемы математической физики и вычислительной математики” к 110-летию со дня рождения академика А.Н. Тихонова, 2016, с. 136.
16. *А.А. Белов.* Численное исследование разрушения решений дифференциальных уравнений // Сборник тезисов научной конференции “Ломоносовские чтения” (секция физики), 2016, с. 105–108.
17. *А.А Belov, N.N. Kalitkin.* Numerical solving of Cauchy problems with contrast structures // Abstracts of Annual Workshop “Numerical methods for problems with layer phenomena”, 2016, p. 10–11.
18. *А.А. Belov, N.N. Kalitkin, L.V. Kuzmina.* Temperature dependence of rate constants and numerical methods for kinetics problems // Abstracts of Scientific-coordination Workshop on Non-ideal Plasma Physics, 2015, p. 31–32.
19. *А.А. Белов, Н.Н. Калиткин.* Решение задач с пограничным слоем сеточными методами // Сборник докладов VIII Международной конференции “Акустооптические и радиолокационные методы измерений и обработки информации”, 2015, с. 64–66.
20. *А.А. Белов, Н.Н. Калиткин.* Сеточные методы решения задач с пограничным слоем // Тезисы докладов XV Всероссийской школы-семинара “Физика и применение микроволн” имени профессора А. П. Сухорукова, 2015, С. 11-13.
http://waves.phys.msu.ru/files/docs/2015/thesis/Section11.pdf
21. *А.А. Belov, N.N. Kalitkin.* Grid methods for boundary layer problems // Abstracts of International conference “Days on Diffraction 2015”, 2015, p. 27–28.
22. *А.А. Белов, Н.Н. Калиткин, Л.В. Кузьмина.* Зависимость скоростей реакций от температуры и численные методы для задач кинетики // Сборник тезисов докладов Международная конференция “XVII Харитоновские тематические научные чтения”, 2015, с. 11–12.
23. *А.А. Белов, Н.Н. Калиткин.* Апостериорная оценка погрешности для уравне-

ния Гельмгольца с пограничным слоем // Сборник тезисов докладов Международного научного семинара “Актуальные проблемы математической физики”, 2014, с. 134–136.

24. *А.А. Белов, Н.Н. Калиткин.* Сверхбыстрый метод с апостериорной оценкой сходимости для эллиптических уравнений // Тезисы докладов научной конференции “Тихоновские чтения”, 2014, с. 55–56.
25. *А.А. Белов, Н.Н. Калиткин.* Сверхбыстрый метод с гарантированной точностью для эллиптических уравнений в прямоугольной области // Тезисы докладов Международной молодежной конференции-школы “Современные проблемы прикладной математики и информатики”, 2014, с. 48–51.
26. *А.А. Белов, Н.Н. Калиткин.* Сверхбыстрый метод с гарантированной точностью для эллиптических уравнений // Сборник докладов VII Международной конференции “Акустооптические и радиолокационные методы измерений и обработки информации”, 2014, с. 42–45.
27. *А.А. Белов, Н.Н. Калиткин, И.П. Пошивайло.* Численные методы решения сверхжестких задач Коши // Тезисы докладов Международной конференции “Современные проблемы вычислительной математики и математической физики” памяти академика А.А. Самарского к 95-летию со дня рождения, 2014, с. 27–29.

1.6 Краткое содержание работы

Работа состоит из введения, пяти глав и заключения. ***Введение*** содержит определение понятия жесткости, формулировки основных проблем, решаемых в данной работе, и современное состояние исследований в соответствующих областях.

Первая группа проблем касается нестационарных жестких задач. Описана специфика задачи кинетики реакций, и приведены наилучшие методы ее решения. Прикладным примером этой задачи является кинетика термоядерных реакций. Для решения последней задачи сформулирована вспомогательная проблема нахождения скоростей реакций в зависимости от температуры. Описана специфика обработки экспериментальных данных по сечениям реакций, и перечислены основные способы аппроксимации сечений. Сформулирована задача численной диагностики сингулярностей в решении дифференциальных уравнений, описаны известные подходы к этой проблеме.

Вторая группа проблем относится к жестким краевым задачам. Дан обзор разностных методов решения эллиптических уравнений и указаны границы их применимости. Описаны подходы к решению и анализу сходимости для сингулярно возмущенных задач.

Обоснована актуальность решаемых задач, сформулированы цели и задачи работы, указана научная новизна полученных результатов, их теоретическая и практическая значимость. Описаны методы и методология исследования, методы обоснования достоверности результатов и их апробации. Сформулированы положения, выносимые

на защиту.

В главе ***Нестационарные жесткие задачи*** разработан новый специализированный численный метод для задач кинетики. Он прост, надежен, имеет хорошую точность, а его трудоемкость гораздо менее, чем в существующих методах. Одновременно с решением вычисляется апостериорная асимптотически точная оценка его погрешности, чего ранее в задачах кинетики реакций не делалось. Работа метода продемонстрирована на представительной тестовой задаче с точным решением, имитирующей кинетику трехчастичных реакций. Проведено сравнение с наилучшими известными методами.

Разработанный метод применен к моделированию реальной кинетики 4 термоядерных реакций, наиболее актуальных для УТС. Используется простейшая постановка (без учета газодинамического разлета и потерь энергии на излучение). Исследованы условия, при которых происходит вспышка горения, необходимая для возникновения самоподдерживающейся реакции.

Разработан новый численный метод диагностики сингулярностей в решениях дифференциальных уравнений. Он позволяет надежно обнаруживать наиболее важные типы особенностей и находить их характеристики с апостериорной асимптотически точной оценкой погрешности. Этот метод применен к исследованию S-режима нелинейного горения.

В главе ***Уравнение Пуассона*** завершена разработка сверхбыстрого итерационного метода для эллиптических уравнений, применимого для достаточно широкого класса задач: уравнение без смешанных производных (коэффициенты могут быть переменными) и прямоугольные неравномерные сетки. Метод обеспечивает логарифмическую скорость сходимости, что значительно быстрее, чем для общих итерационных методов. Предложен практически неулучшаемый логарифмический набор шагов для счета на установление, построены априорные оценки сходимости и апостериорные асимптотически точные оценки фактической погрешности.

Глава ***Диффузия в пограничных слоях*** посвящена исследованию сингулярно возмущенных краевых задач. В решении уравнения Гельмгольца предложено выделять не только традиционные пограничный слой и регулярную часть, но и переходную зону между ними. В ней решение имеет большую кривизну, что представляет специфические трудности при расчете. Для задач в прямоугольной области предложена адаптивная квазиравномерная сетка. Она детально разрешает все характерные участки решения и позволяет получать высокие точности при небольшом числе узлов. Показано, что в таких задачах можно использовать метод Ричардсона. Он позволяет находить апостериорную асимптотически точную оценку погрешности и исследовать порядок фактической точности.

В главе ***Кинетика термоядерных реакций*** разработан метод аппроксимации экспериментальных результатов, измеренных со значительными погрешностями. В отличие от общепринятых подходов он не использует априорных гипотез о качественном виде искомой зависимости. Метод заключается в построении аппроксимации по методу двойного периода со специальным регуляризатором. С использованием этого

метода получены аппроксимации для сечений 4 термоядерных реакций, и получены оценки достоверности этих аппроксимаций. По ним рассчитаны скорости этих реакций, более точные, чем предлагалось ранее. Исследован частный вопрос прецизионного вычисления квадратур по формулам Эйлера-Маклорена.

Глава ***Пакеты программ*** содержит исходные коды, описания и контрольные тесты для программных пакетов, которые разработаны на основе методов, предложенных в данной работе.

Заключение содержит основные результаты, выносимые на защиту.

2. Нестационарные жесткие задачи

В данной главе предложена специальная явная схема второго порядка точности для задачи кинетики реакций. Проведены расчеты представительного тестового примера, имитирующего уравнение кинетики трехчастичной реакции.

Предложен метод численной диагностики разрушений для систем ОДУ, позволяющий определить момент сингулярности и ее порядок с гарантированной точностью. Он применим и к уравнениям в частных производных, так как при численном решении они сводятся методом прямых к системе ОДУ.

2.1 Кинетика реакций

2.1.1. Вид схем. Специфический вид задачи (1.1) позволяет построить специализированный метод. В самом деле, эта задача может быть записана в следующем виде:

$$\frac{dn_j}{dt} = f_j\left(\mathbf{n}\right), \quad f_j(\mathbf{n}) = -n_j\varphi_j(\mathbf{n}) + \psi_j(\mathbf{n}), \quad \mathbf{n} = \{n_j,\ 1 \leqslant j \leqslant J\} \tag{2.1}$$

причем $n_j \geqslant 0$, $\varphi_j(\mathbf{n}) \geqslant 0$, $\psi_j(\mathbf{n}) \geqslant 0$. Существенно то, что концентрация n_j входит в j-ю отрицательную правую часть как сомножитель. Эти особенности позволяют построить явную схему, обладающую малой трудоемкостью.

Введем следующие обозначения: t – исходный момент времени, $\hat{t} = t + \tau$ – новый момент времени, n_j и $\hat{n}_j$ – решения в эти моменты. В [5] была построена схема точности $O\left(\tau\right)$. Решение в новый момент времени по этой схеме обозначим через $\tilde{n}_j$; оно равно

$$\tilde{n}_j = \frac{n_j + \tau\psi_j\left(\mathbf{n}\right)}{1 + \tau\varphi_j\left(\mathbf{n}\right)}. \tag{2.2}$$

Эта схема разумна: увеличение $\psi_j(\mathbf{n})$ как в точном, так и в численном решении приводит к увеличению образования вещества, а увеличение $\varphi_j(\mathbf{n})$ действует противоположно. При этом n_j остается неотрицательным.

Недостатком схемы (2.2) является невысокая точность. В [5] предлагались методы повышения порядка точности до второго. Однако они оказались неудачными: в численном решении возникала сильная немонотонность, из-за чего порядок точности фактически не повышался. Кроме того, схема теряла надежность; например, начальные концентрации нельзя было задавать нулями, нужно было вводить малые числа.

В данной работе построена явная схема второго порядка точности, одновременно имеющая более высокую надежность. Приведем ее. Напишем следующую неявную схему:

$$\hat{n}_j = \frac{n_j + \tau\psi_j(\overline{\mathbf{n}})(1 + \tau\varphi_j(\overline{\mathbf{n}})/2)}{1 + \tau\varphi_j(\overline{\mathbf{n}}) + (\tau\varphi_j(\overline{\mathbf{n}}))^2/2}, \quad \overline{\mathbf{n}} = (\mathbf{n} + \hat{\mathbf{n}})/2. \tag{2.3}$$

Будем находить решение алгебраической системы простыми итерациями:

$$\hat{n}_j^{s+1} = \frac{n_j + \tau\psi_j(\overline{\mathbf{n}}^s)(1 + \tau\varphi_j(\overline{\mathbf{n}}^s)/2)}{1 + \tau\varphi_j(\overline{\mathbf{n}}^s) + (\tau\varphi_j(\overline{\mathbf{n}}^s))^2/2}, \quad \overline{\mathbf{n}}^s = (\mathbf{n} + \hat{\mathbf{n}}^s)/2, \quad \hat{\mathbf{n}}^0 = \mathbf{n}. \tag{2.4}$$

При этом выполним только две итерации, то есть по существу получим явную схему. Здесь также увеличение ψ_j приводит к увеличению $\hat{n}_j$, а увеличение φ_j – к уменьшению $\hat{u}_j$. Схемы (2.2) и (2.4) будем называть ***химическими схемами*** (одностадийной и двухстадийной соответственно).

2.1.2. Свойства.

Аппроксимация и устойчивость. Разложением в ряды можно доказать, что двухстадийная схема (2.4) имеет аппроксимацию $O(\tau^2)$. Третью и последующие итерации выполнять не следует, так как это не повышает порядок точности, но может ухудшить надежность схемы. Также нетрудно непосредственно убедиться, что на линейном тесте Далквиста

$$\frac{du}{dt} = -\lambda u \tag{2.5}$$

схема (2.4) является L_2-устойчивой.

Трудоемкость. Схема (2.4) явная, поэтому расчеты по ней имеют малую трудоемкость. В самом деле, расчеты на каждой итерации требуют однократного вычисления J правых частей. Таким образом, трудоемкость двухстадийной химической схемы такова же, как у двухстадийной явной схемы Рунге-Кутты, что гораздо меньше трудоемкости неявных схем. Последние требуют нахождения матрицы Якоби, что соответствует вычислению J^2 правых частей.

Таким образом, схема (2.4) в $\sim J$ раз менее трудоемка, чем явно-неявные схемы и схемы Розенброка и Розенброка-Ваннера. Выигрыш по сравнению с чисто неявными итерационными схемами еще больше. Это преимущество особенно существенно для систем большого порядка, когда в реакциях участвует большое число компонент.

Неконсервативность. Схемы (2.2) и (2.4) имеют недостаток: они неконсервативны. В точном решении суммарное число атомов каждого элемента всегда остается постоянным. Такой баланс есть линейный первый интеграл системы (2.1). Но в методе (2.2) и (2.4) для численного решения эти интегралы передаются не точно, а лишь приближенно с точностью $O(\tau)$ для схемы (2.2) и $O(\tau^2)$ для схемы (2.4).

Этот недостаток не столь существенен. В задаче (1.1) решение является классическим, обобщенных решений нет. Поэтому достаточно провести расчеты со сгущением сеток. Численные решения, полученные на разных сетках, будут стремиться к предельной функции при $\tau \to 0$. Поскольку решение классическое, явление ложной сходимости отсутствует, и предельная функция будет искомым точным решением.

Поэтому нарушение баланса будет стремиться к нулю при $\tau \to 0$. При этом оно имеет приблизительно тот же порядок величины, что и погрешность решения. Поэтому дисбаланс может служить неплохим ддополнительным средством контроля точности.

Знакопостоянность. Поскольку $n_j \geqslant 0$, $\varphi_j(\mathbf{n}) \geqslant 0$, $\psi_j(\mathbf{n}) \geqslant 0$, то, очевидно, $\hat{n}_j \geqslant 0$ и $\tilde{n}_j \geqslant 0$. Поэтому численное решение по схемам (2.2) и (2.4) является знакопостоянным. Это соответствует физическому смыслу задачи (1.1) и является достоинством этих схем. Далее будет показано, что неявные схемы знакопостоянности не гарантируют.

Немонотонность. Схемы (2.2) и (2.4) являются немонотонными, то есть при монотонном точном решении численное решение может иметь участки немонотонности или осциллировать. Однако на нелинейных жестких задачах неявные схемы также иногда оказываются немонотонными. Для исследования этого свойства мы проводили расчеты специальной тестовой задачи, представленной ниже.

2.1.3. Тестовая задача.

Постановка. Рассмотрим задачу для одного уравнения с кубической правой частью, имитирующую химические реакции со столкновением 3 одинаковых частиц:

$$\frac{dn}{dt} = -\lambda n \left(n^2 - a^2\right); \quad a > 0, \quad \lambda \gg 1; \quad n(0) = n^0. \tag{2.6}$$

Эта задача удобна тем, что для нее легко построить точное решение

$$n(t) = \frac{an^0}{\sqrt{(n^0)^2 + \left[a^2 - (n^0)^2\right] \exp\{-2\lambda a^2 t\}}}. \tag{2.7}$$

Поле интегральных кривых задачи (2.6) приведено на рис. 2.1. Точное решение имеет три стационара $n(t) = a, 0, -a$; из них первый и третий устойчивые, а второй неустойчивый. Точное решение имеет пограничный слой шириной $t \sim 1/\lambda a^2$ и быстро выходит на 1-й стационар при $n^0 > 0$ и на 3-й при $n^0 < 0$. Если n имеет смысл концентрации, то осмысленным является только положительное решение $n(t) > 0$, а отрицательные решения физического смысла не имеют.

Расчет по химическим схемам. Задаче (2.6) соответствуют $\varphi = \lambda n^2$, $\psi = \lambda n a^2$. Проанализируем поведение решения (2.2) вблизи стационара. Легко проверить, что эту схему можно преобразовать к следующему виду:

$$\hat{n} - a = (n - a)\frac{1 - \tau\lambda a n}{1 + \tau\lambda n^2} \tag{2.8}$$

Если $\tau\lambda an > 1$ (то есть сетки достаточно грубые), то дробь отрицательна. Тогда $\hat{n} - a$ и $n - a$ имеют разные знаки, и выход на стационар оказывается немонотонным. Решение начинает осциллировать вокруг стационарного значения, причем амплитуда осцилляций убывает со временем. Это не препятствует сходимости (так как амплитуда осцилляций уменьшается как $O(\tau)$ при $\tau \to 0$), но делает неправильным качественное поведение решения (см. рис. 2.2).

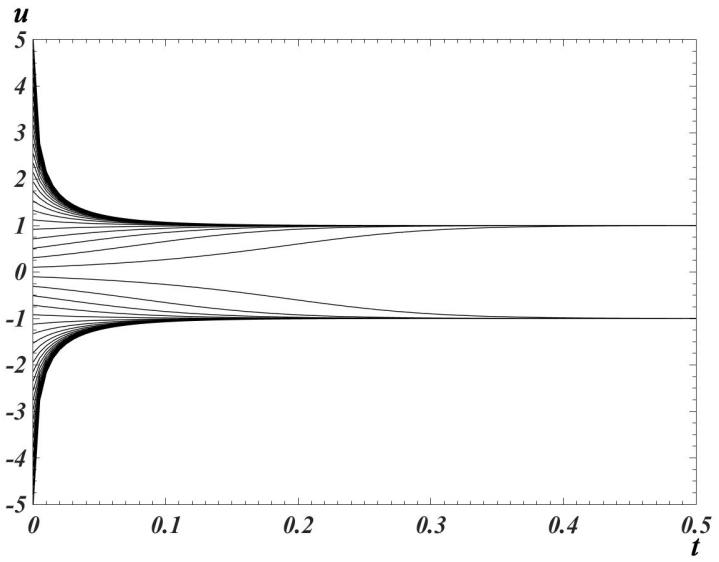

Рис. 2.1. Поле интегральных кривых для теста (2.6) при $a = 1$.

Аналогично проанализируем первую стадию (2.4); для нее $\overline{n} = n$, и

$$n^{(1)} - a = (n - a)\frac{1 - \tau\lambda an - \tau^2\lambda^2 an^3/2}{1 + \tau\lambda n^2 + \tau^2\lambda^2 n^4/2}. \tag{2.9}$$

Знаменатель в правой части всегда положителен, а числитель при достаточно большом τ становится отрицательным. Это значит, что $u^{(1)} - a$ и $u - a$ имеют разные знаки, и на первой стадии решение "перепрыгивает" через стационар. Поэтому если вести расчет по схеме (2.4) только с одной стадией, то численное решение будет иметь пилообразный вид. Результаты расчета с двумя стадиями представлен на рис. 2.3.

Решение на самой грубой сетке с $N = 10$ шагами имеет качественно правильный вид. В этом случае шаг заметно больше ширины пограничного слоя, то есть все узлы сеток лежат в области регулярного решения. На следующей сетке с $N = 40$ первый шаг "перепрыгивает" на другую сторону стационара $n = 1$, и далее решение стремится к этому стационару с неправильной стороны. Здесь шаг близок к ширине пограничного слоя; эти условия наиболее трудны для схемы. Решение на более подробных сетках имеют правильное качественное поведение, поскольку они уже разумно разрешают пограничный слой.

Отметим, что ни на одном из решений мы не видели осцилляций в отличие от одностадийной схемы (2.2). Это означает, что двухстадийная химическая схема (2.4) не только обладает лучшей точностью, но и одновременно является более надежной, чем известная ранее схема (2.2).

Сравнение с неявными схемами. Чисто неявная и комплексная схемы Розенброка, а также неявная схема Эйлера известны как монотонные, так как они дают монотонное решение на линейном тесте Далквиста (2.5). Однако на нелинейном тесте (2.6) картина оказывается иной.

Расчеты теста (2.6) по этим схемам показали, что все эти схемы могут становиться немонотонными (см. рис. 2.4). На грубых сетках решение может и вовсе притягиваться к отрицательному или нулевому стационарам, что противоречит физическому

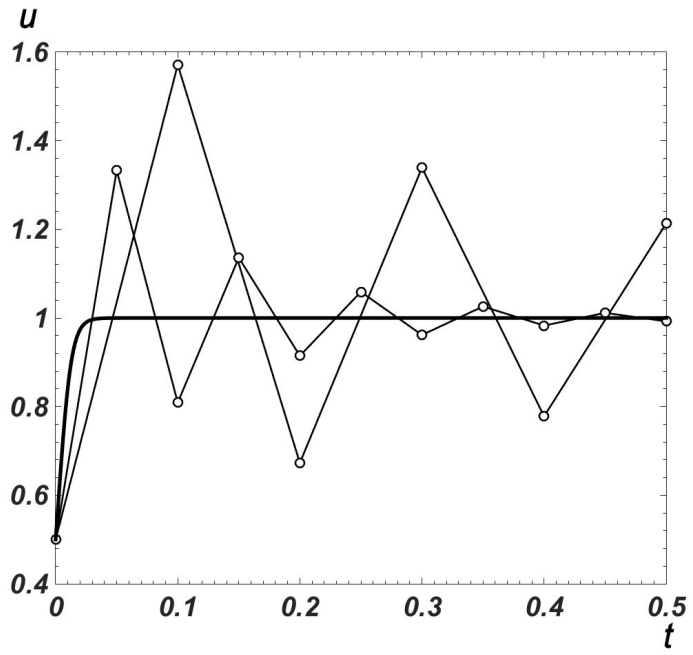

Рис. 2.2. Решение теста (2.6) по одностадийной химической схеме (2.2); ∘ – расчетные точки, жирная линия – точное решение.

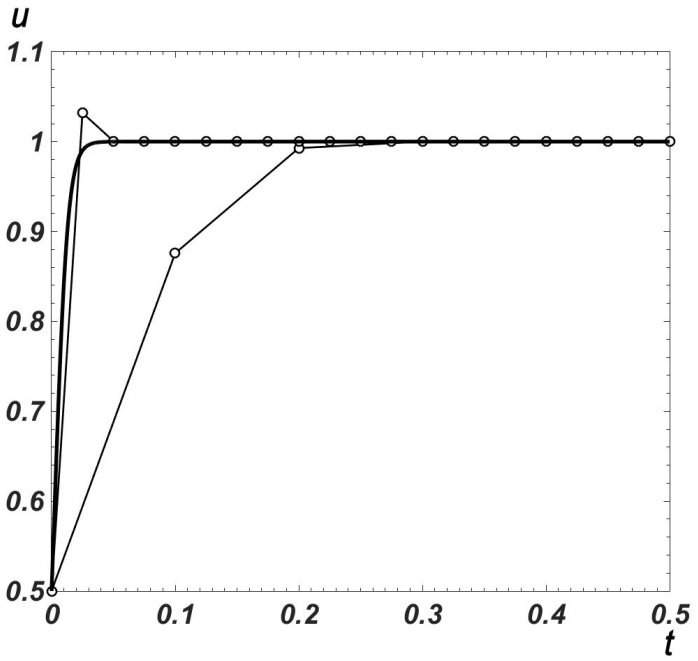

Рис. 2.3. Решение теста (2.6) по двухстадийной химической схеме (2.4); обозначения соответствуют рис. 2.2.

смыслу задачи.

Причина этого заключается в следующем. Несмотря на то, что решение исходной дифференциальной задачи (2.6) единственно, нелинейное алгебраическое уравнение относительно $\hat{u}$ может несколько корней. Если задача жесткая, а шаг грубый, то нулевое приближение, выбранное с предыдущего шага, может оказаться неудачным. В результате итерационный процесс может сойтись к неправильному корню.

Таким образом, на задаче кинетики (1.1) неявные схемы не имеют преимуществ

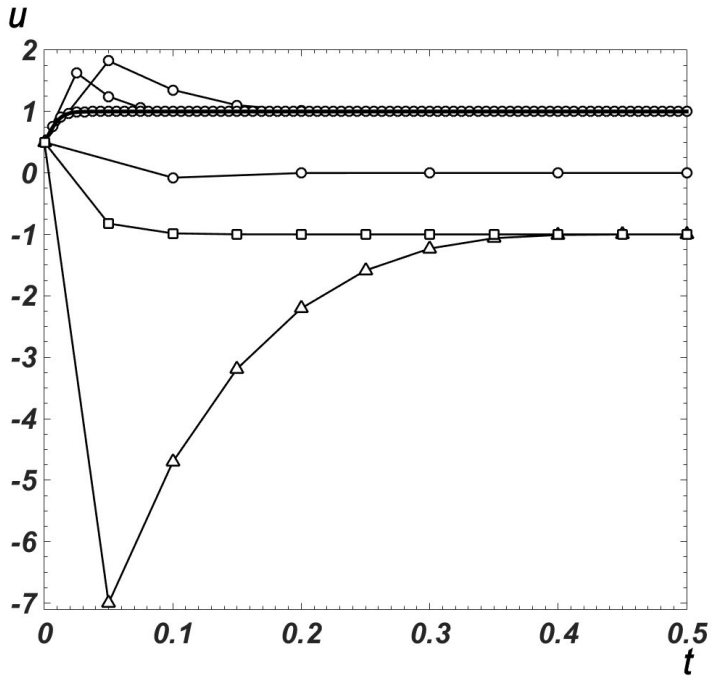

Рис. 2.4. Решение теста (2.6) по неявным схемам: △ – чисто неявная схема Розенброка, ∘ – CROS, □ – неявная схема Эйлера; жирная линия – точное решение.

перед химическими схемами и при этом являются более трудоемкими.

Выводы. Задача (1.1) является жесткой, а двухитерационная химическая схема относится к явным. Может показаться, что это противоречит теореме Далквиста. Однако здесь нет никакого противоречия. В определении устойчивости по Далквисту ошибка не должна нарастать при любом шаге. В химических схемах такая устойчивость имеет место в линейном тесте (2.5), но в нелинейных задачах ошибка не растает лишь при достаточно малом шаге. Именно это условие обеспечивает нужное поведение вблизи пограничного слоя (детальное исследование этого вопроса выходит за рамки данной работы).

Здесь можно провести аналогию с понятием устойчивости по Рябенькому-Филлипову [42]. В нем также требуется, чтобы в линейных задачах ошибка не нарастала при любом шаге, а в нелинейных – при достаточно малом. Заметим также, в реальных задачах кинетики химических и ядерных реакций жесткость редко оказывается сверхвысокой, а химические схемы не требуют неоправданно малого шага и позволяют ограничиться приемлемым объемом вычислений.

Таким образом, двухитерационная химическая схема (2.4) очень перспективна для задач кинетики. При расчетах в длине дуги (см. п. 2.1.4) она является лучшей из известных. Однако она применима только к задачам, сводящимся к (2.1). Кроме того, на сверхжестких задачах к ней (как и к другим схемам) следует относиться с осторожностью и не пренебрегать визуальным контролем по профилям решения и по графикам погрешности (см. п. 2.1.5).

2.1.4. Длина дуги. В задаче (2.1) можно ввести параметризацию через длину дуги интегральной кривой в пространстве переменных $\{t, n_1, \ldots, n_J\}$. Свойства этой и

других параметризаций детально исследованы Е. Б. Кузнецовым (см. [43] и другие работы этого автора). В частности, доказано, что параметризация через длину дуги является наилучшей с точки зрения обусловленности системы.

Чтобы осуществить переход к длине дуги, формально добавим к компонентам вектора $\mathbf{n}$ нулевую компоненту $n_0 \equiv t$. Время и концентрации имеют разный физический смысл и разную размерность. Поэтому целесообразно ввести нормирующие множители (обезразмеривание), соответствующие характерным масштабам: ν_0 порядка полного времени расчета и ν порядка величины основных концентраций. Последнюю величину можно брать из начальных условий. Тогда примем

$$dl^2 = \frac{dn_0^2}{\nu_0^2} + \sum_{j=1}^{J} \frac{dn_j^2}{\nu^2}, \quad dn_0 \equiv dt. \tag{2.10}$$

Отсюда

$$dl = dt\sqrt{\nu_0^{-2} + \nu^{-2}\sum_{j=1}^{J} f_j^2(\mathbf{n})} \equiv S(\mathbf{n})dt. \tag{2.11}$$

В результате получим систему уравнений, порядок которой на 1 больше, чем у исходной:

$$\frac{dn_j}{dl} = F_j(\mathbf{n}), \quad F_j(\mathbf{n}) = -n_j\Phi_j(\mathbf{n}) + \Psi_j(\mathbf{n}) \quad 0 \leqslant j \leqslant J, \tag{2.12}$$

$$\begin{aligned} &\Phi_0(\mathbf{n}) = 0, && \Psi_0(\mathbf{n}) = \frac{1}{S(\mathbf{n})}, \\ &\Phi_j(\mathbf{n}) = \frac{\nu_0\varphi_j(\mathbf{n})}{S(\mathbf{n})}, && \Psi_j(\mathbf{n}) = \frac{\nu_0\psi_j(\mathbf{n})}{S(\mathbf{n})}. \end{aligned} \tag{2.13}$$

Теперь вектор правых частей является единичным, это существенно упрощает численное решение задачи. Очевидно, для этой системы также применимы схемы (2.2) и (2.4).

2.1.5. Расчет со сгущением сеток.

Оценка погрешности. При расчете задачи (1.1) по схеме (2.4) нам известен априорный порядок точности. Тогда погрешность численного решения можно апостериорно оценить по методу Ричардсона, причем эта оценка является асимптотически точной [44], [45]. Этот метод имеет хорошее теоретическое обоснование.

Сравним профили некоторой j-й концентрации на двух соседних сетках. Узлы грубой сетки совпадают с четными узлами подробной сетки. Беря разности σ_j значений j-й концентрации в совпадающих узлах, найдем асимптотически точную оценку погрешности решения в этом узле подробной сетки

$$\Delta_j = \frac{\sigma_j}{(2^p - 1)}, \tag{2.14}$$

где p – порядок точности схемы. Это позволяет вычислить для каждой концентрации ошибку ε_j в норме C по всему профилю. Для общей характеристики относительной

погрешности целесообразно взять некоторое усреднение этой ошибки по всем концентрациям

$$\varepsilon = \frac{1}{\nu}\sqrt{\sum_{j=1}^{J}\frac{\varepsilon_j^2}{J}} \tag{2.15}$$

Контроль точности удобно проводить по графику зависимости ε от N, выполнен-

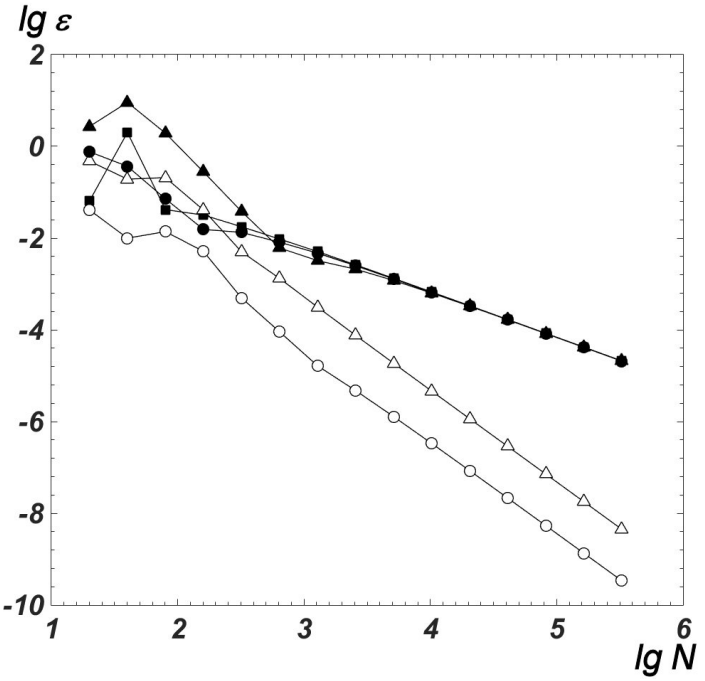

Рис. 2.5. Оценки погрешности по методу Ричардсона в тесте (2.6); ▲ – чисто неявная схема Розенброка, △ – комплексная схема Розенброка, ■ –неявная схема Эйлера, • – одностадийная химическая схема, ∘ – двухстадийная химическая схема.

ному в двойном логарифмическом масштабе. Тогда степенному характеру сходимости соответствует прямая линия, причем наклон этой прямой равен фактическому порядку точности. Если график асимптотически выходит на прямую линию, то полученные оценки погрешности являются достоверными. Сгущение сеток проводится до тех пор, пока не будет достигнута требуемая точность.

В качестве примера применения метода Ричардсона приведем график погрешности в задаче (2.6). Он представлен на рис. 2.5. На первых нескольких сетках погрешность ведет себя нерегулярно, что говорит о неправильном качественном поведении решений. Однако начиная с достаточно подробных сеток, линии выходят на прямые с наклоном, равным теоретическому порядку точности соответствующей схемы. Видно также, что двухстадийная химическая схема позволяет получать гораздо более высокие точности, чем все схемы первого порядка и несколько более высокие, чем гораздо более трудоемкая схема CROS.

Расчеты в длине дуги. Метод Ричардсона можно применять как в аргументе время, так и в аргументе длина дуги [46], [47]. В этом случае оценка погрешности по методу Ричардсона дает погрешность $\Delta_j(l)$ (включая $j = 0$, соответствующее $n_0 \equiv t$) как функцию длины дуги, что не всегда удобно на практике. При химических и тер-

моядерных экспериментах выход продукта регистрируется в определенные моменты времени. К этим моментам и должны относиться оценки погрешности сеточного решения.

Опираясь на ричардсоновские оценки погрешности, при расчете по длине дуги можно построить погрешности $\delta_j(t)$ как функции времени:

$$\delta_j(t) = \Delta_j(l) - f_j(\mathbf{n})\Delta_0(l), \quad 1 \leqslant j \leqslant J. \tag{2.16}$$

Оценку (2.16) назовем ***приведенной***. Из свойств оценок по методу Ричардсона следует, что она асимптотически точна.

2.2 Расчет термоядерного горения

2.2.1. Постановка задачи.

Модели горения. Расчетами мишеней для лазерного термоядерного синтеза в разные годы занимались О. Н. Крохин, В. Б. Розанов, А. А. Самарский, Н. В. Змитренко, М. М. Баско, Г. В. Долголёва и другие. Современные расчеты основаны на достаточно сложных моделях, включающих большое число факторов. Например, в газовых мишенях и в мишенях для тяжелоионного синтеза учитываются газодинамика в двухтемпературном приближении, перенос тепла ионами и электронами, перенос излучения и его взаимодействие с веществом, а также кинетика термоядерных реакций (см. [41] и библиографию там).

Мы рассмотрим упрощенную постановку. Теплопроводность и газодинамический разлет при нагреве оболочки, а также предварительный нагрев мишени за счет электронов, выбиваемых излучением из оболочки, являются конкурирующими процессами одного порядка. Поэтому их следует учесть или отбросить одновременно. Ограничимся только кинетикой термоядерных реакций с учетом повышения температуры за счет их энерговыхода. Рассмотрим следующие реакции:

$$\mathrm{D} + \mathrm{D} \to p + \mathrm{T}, \tag{2.17}$$

$$\mathrm{D} + \mathrm{D} \to n + {}^3\mathrm{He}, \tag{2.18}$$

$$\mathrm{D} + \mathrm{T} \to n + {}^4\mathrm{He}, \tag{2.19}$$

$$\mathrm{D} + {}^3\mathrm{He} \to p + {}^4\mathrm{He}. \tag{2.20}$$

Они являются важнейшими в расчетах УТС. Будем считать, что энергия, связанная с заряженными продуктами реакций, остается внутри плазмы и мгновенно перераспределяется между всеми частицами, включая электроны, а часть энергии, приходящаяся на нейтроны, покидает систему.

Эта постановка простая, но очень грубая и подходит только для расчета начала вспышки термоядерных реакций. Для дальнейших стадий нужно учитывать газодинамический разлет и выход излучения из мишени. Двухтемпературность нужно учитывать только в момент максимальной интенсивности термоядерных реакций. До него ионы успевают обменяться энергией с электронами, и их температуры можно

считать одинаковыми. После этого момента скорости реакций снижаются, поскольку мишень выгорает, и электроны снова успевают прийти в равновесие с тяжелыми частицами.

Система уравнений. Введем следующие обозначения для концентраций: n_1 – нейтроны n, n_2 – протоны p, n_3 – дейтерий D, n_4 – тритий T, n_5 – ^{3}He, n_6 – ^{4}He. Они удовлетворяют уравнениям

$$\frac{dn_1}{dt} = K_2(T)n_3^2 + K_3(T)n_3n_4, \tag{2.21}$$

$$\frac{dn_2}{dt} = K_1(T)n_3^2 + K_4(T)n_3n_5, \tag{2.22}$$

$$\frac{dn_3}{dt} = -2[K_1(T) + K_2(T)]n_3^2 - K_3(T)n_3n_4 - K_4(T)n_3n_5, \tag{2.23}$$

$$\frac{dn_4}{dt} = K_1(T)n_3^2 - K_3(T)n_3n_4, \tag{2.24}$$

$$\frac{dn_5}{dt} = K_2(T)n_3^2 - K_4(T)n_3n_5, \tag{2.25}$$

$$\frac{dn_6}{dt} = K_3(T)n_3n_4 + K_4(T)n_3n_5. \tag{2.26}$$

Здесь K_1, K_2, K_3, K_4 – это скорости реакций (2.17), (2.18), (2.19) и (2.20) соответственно. Вопрос о нахождении этих величин подробно рассмотрен в главе 5 (см. (5.15) и табл. 5.4). Уравнение для температуры имеет вид

$$\frac{dT}{dt} = \frac{2}{3}\frac{dq}{dt} \Big/ \left(n_e + \sum_{k=2}^{6} n_k\right). \tag{2.27}$$

Здесь n_e – концентрация электронов (она не меняется со временем), dq/dt – баланс энергии в единице объема в единицу времени. Она увеличивается за счет энерговыхода реакций и уменьшается на величину энергии, уносимой быстрыми нейтронами:

$$\frac{dq}{dt} = \left(\varepsilon_1 K_1(T) + \frac{1}{4}\varepsilon_2 K_2(T)\right) n_3^2 + \\ + \frac{1}{5}\varepsilon_3 K_3(T)n_3n_4 + \varepsilon_4 K_4(T)n_3n_5 - \\ - \frac{3}{2}T\left(K_2(T)n_3^2 + K_4(T)n_3n_4\right); \tag{2.28}$$

$\varepsilon_1 = 4.033$ МэВ, $\varepsilon_2 = 3.270$ МэВ, $\varepsilon_3 = 17.590$ МэВ, $\varepsilon_4 = 18.354$ МэВ – энергии реакций (2.17), (2.18), (2.19) и (2.20) соответственно.

Скорости реакций $K(T)$ меняются на много порядков с ростом температуры. Поэтому задача (2.21) – (2.27) – это серьезный тест для химической схемы, предложенной выше.

2.2.2. Результаты расчетов.

Концентрации. Расчет проводился для дейтериево-тритиевой мишени (плотность $\rho = 25$ г/см^3, начальные концентрации дейтерия и трития $n_3 = n_4 = 3 \cdot 10^{24}$ см$^{-3}$) и для чисто дейтериевой мишени (плотность $\rho = 20$ г/см^3, начальная концентрация $n_3 = 6 \cdot 10^{24}$ см$^{-3}$). Заданные условия соответствуют 100-кратному сжатию

замороженного газа. Расчеты велись по двухстадийной химической схеме в аргументе t, шаг по времени τ брался постоянным и достаточно малым, чтобы обеспечить хорошую точность расчета. Начальная температура равнялась $T_0 = 1$ кэВ.

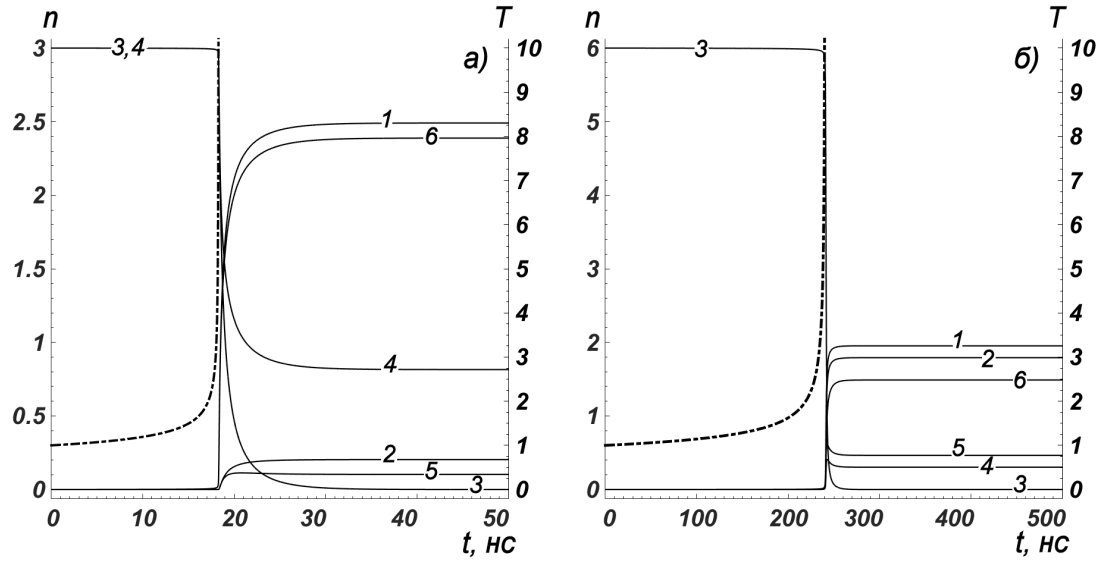

Рис. 2.6. Расчет задачи (2.21) – (2.27); а) DT-мишень, б) DD-мишень; сплошные линии – концентрации в единицах 10^{24} см$^{-3}$, 1 – n, 2 – p, 3 – D, 4 – T, 5 – ^{3}He, 6 – ^{4}He; штрих-пунктирная линия – температура, кэВ.

На рис. 2.6 приведены профили концентраций и температуры в зависимости от времени. В DT-мишени концентрации практически не меняются до момента $t \approx 18.2$ нс. Далее смесь загорается: концентрации D и T монотонно убывают, концентрации p, n и ^{4}He монотонно нарастают, а концентрация ^{3}He сначала нарастает, затем начинает очень медленно убывать. К моменту $t \sim 34$ нс все концентрации выходят на постоянные значения (в частности, D – на нулевое); это означает, что горение завершилось.

В DD-мишени поведение концентраций во многом аналогично. Исходная компонента D расходуется; n, p и ^{4}He нарабатываются, а промежуточные компоненты T и ^{3}He сначала нарабатываются, потом расходуются. Смесь загорается при $t \sim 240$ нс, то есть значительно позже, чем в DT-мишени.

Строго говоря, результат расчета зависит от начальной температуры. При уменьшении T_0 удлиняется начальный участок кривых n и T. Однако на этом участке реакции идут очень медленно. Например, при $T_0 = 0.1$ кэВ в DT-мишени начальный участок удлиняется примерно в 100 раз, однако за это время выделяется всего 0.1% от полной энергии реакции. Поэтому к значению температуры $T_0 = 1$ кэВ мишень приходит практически с тем же начальным составом. Таким образом, профили n и T, начатые с более низких T_0, практически повторят кривые, представленные на рис. 2.6 (но это будет соответствовать более поздним моментам времени). Это значит, что проведенный расчет является представительным.

На рис. 2.7 показаны относительная погрешность по методу Ричардсона и относительный дисбаланс для обоих расчетов в зависимости от числа узлов сетки N.

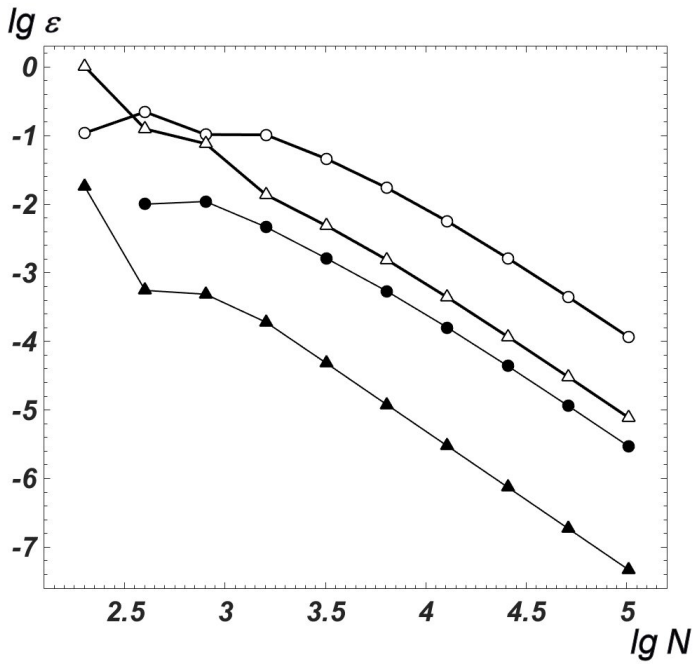

Рис. 2.7. Сходимость в задаче (2.21) – (2.27); ∘ – относительная погрешность, △ – относительный дисбаланс; темные маркеры – DT-мишень, светлые – DD-мишень.

Масштаб графика двойной логарифмический. Хорошо видно, что все кривые выходят на прямые линии с наклоном 2, что соответствует теоретическому порядку сходимости $p = 2$. Это означает, что результаты расчетов и оценки их погрешности являются достоверными. Таким образом, химическая схема позволяет надежно рассчитывать кинетику указанных реакций.

Вспышка. Пусть плазма достаточное время удерживается в состоянии с постоянной плотностью. Тогда сначала реакции идут медленно, а когда температура поднимется до достаточной величины, скорости реакций резко увеличиваются. Это резкое ускорение реакций будем называть вспышкой. Если вспышки удалось достичь, то даже при наличии разлета существенный процент вещества успеет выгореть. Поэтому важно определить, при каких условиях наступает вспышка. Это позволит определить время удержания плазмы, необходимое для получения вспышки и хорошего выгорания.

Оценим время вспышки по поведению правых частей уравнений относительно начальных компонент: (2.23), (2.24) для DT-мишени и (2.23) для DD-мишени. Графики этих правых частей в зависимости от времени показаны на рис. 2.8. На них четко виден участок крутого нарастания. Момент $t_{\text{в}}$, с которого он начинается, и есть начало вспышки. Достижение этого момента необходимо для того, чтобы управляемый термоядерный синтез мог бы иметь параметры, пригодные для промышленного использования, а не для чисто научных демонстраций.

Максимум скорости этих реакций (пик) достигается в момент $t_{\text{п}}$. После пика скорости реакций убывают, причем с течением времени закон убывания становится близок к экспоненциальному (показано для DT-мишени). Значения $t_{\text{в}}$ и $t_{\text{п}}$, а также соответствующие температуры $T_{\text{в}}$ и $T_{\text{п}}$ и доли выгоревшего топлива $d_{\text{в}}$ и $d_{\text{п}}$ приведены

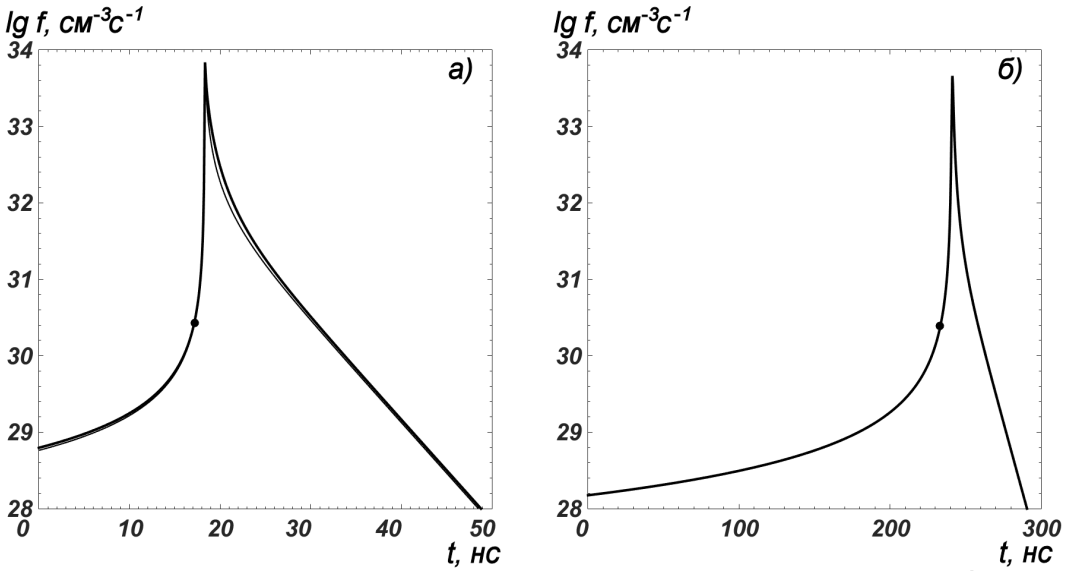

Рис. 2.8. Правые части уравнений: жирная линия – (2.23), тонкая – (2.24); мишени: а) – DT, б) – DD; точка – начало вспышки.

Таблица 2.1. Вспышки при горении в DT- и DD-мишенях.

	DT	DD		DT	DD
$t_в$, нс	17.2	234	$t_п$, нс	18.3	241
$T_в$, кэВ	2	3	$T_п$, кэВ	16	~ 70
$d_в$, %	0.3	0.5	$d_п$, %	3	15

в табл. 2.1 для начальной температуры $T_0 = 1$ кэВ и 100-кратного сжатия. Заметим, что на сегодняшний день получение таких условий весьма проблематично, но даже при них требуются весьма значительные времена удержания плазмы. Видно также, что даже к моменту пика вспышки выгорание топлива невелико (для DT – всего 3%).

Температурные кривые. На рис. 2.9 представлены температурные кривые в DT- и DD-мишенях при плотностях $\rho = 2.5,\ 25,\ 250$ г/см^3 и $\rho = 2,\ 20,\ 200$ г/см^3 соответственно (то есть 10-, 100- и 1000-кратное сжатие замороженного газа). Масштаб по времени логарифмический.

Видно, что для всех трех плотностей кривые являются подобными. Вспышка происходит при одной и той же температуре, а сам момент вспышки $t_в$ обратно пропорционален плотности (при увеличении сжатия в 10 раз вспышка происходит в 10 раз раньше). Это объясняется тем, что все реакции (2.17) – (2.20) являются двухчастичными.

Заметим, что для токамаков, у которых характерная плотность плазмы составляет $\sim 10^{15}$ см$^{-3}$, время время достижения вспышки окажется в 10^{10} раз больше, то есть порядка десятков секунд (и это для начальной температуры $T_0 = 1$ кэВ!). Это значит, что проекты удержания плазмы в токамаках очень далеки от перспектив

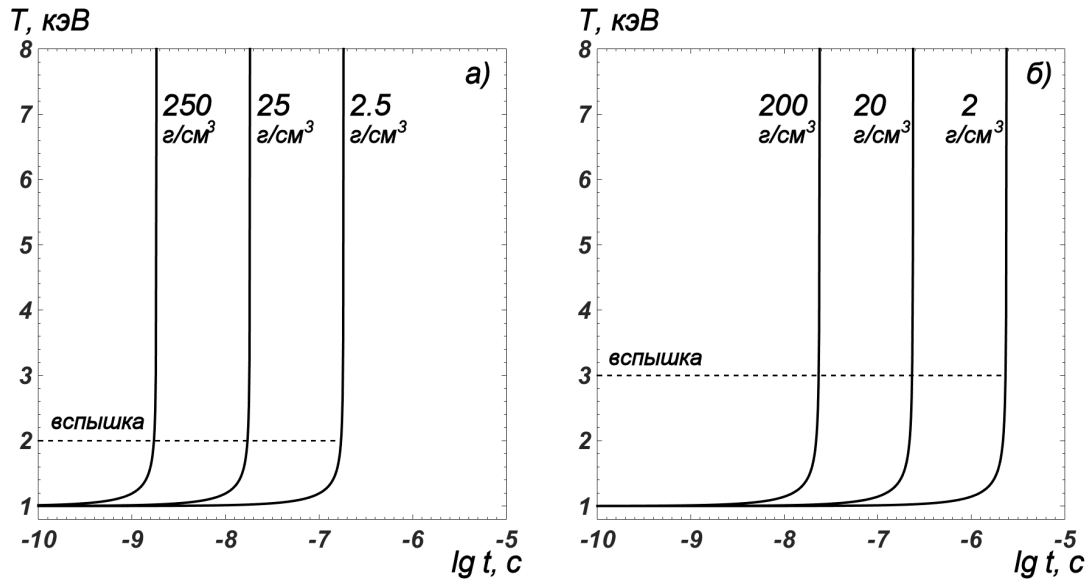

Рис. 2.9. Профили температуры в задаче (2.21) – (2.27); а) DT-мишень, б) DD-мишень; плотности указаны около кривых.

промышленного использования.

Также в серьезной корректировке нуждается критерий Лоусона, который является условием возникновения самоподдерживающейся термоядерной реакции [48]. Так, для того, чтобы энерговыход DT-реактора был не меньше его энерговхода, необходимо выполнение условия $n_{\text{пл}}\tau_{\text{у}} \geqslant 10^{14}$ см$^{-3}$с. Здесь $n_{\text{пл}}$ – плотность плазмы, $\tau_{\text{у}}$ – время ее удержания. Для DD-реактора $n_{\text{пл}}\tau_{\text{у}} \geqslant 10^{15}$ см$^{-3}$ с.

Однако такой критерий слишком мягок. Ему соответствует выгорание всего $\sim 10^{-5}$ топлива. Согласно табл. 2.1, промышленный выход энергии от DT-реактора возможен при $n_{\text{пл}}\tau_{\text{у}} \geqslant 10^{17}$ см$^{-3}$с, а от DD-реактора – при $n_{\text{пл}}\tau_{\text{у}} \geqslant 10^{18}$ см$^{-3}$с; то есть значение критерия Лоусона следует увеличить на 3 порядка!

2.3 Диагностика разрушений

2.3.1. Полюс.

Скалярное уравнение. Поясним основную идею на примере одного уравнения

$$\frac{du}{dt} = f(u), \quad f(u) = u^{\nu}, \quad \nu > 1, \quad u(0) = u_0. \tag{2.29}$$

Для него нетрудно построить точное решение

$$u(t) = \frac{u_0}{(1 - t/t_0)^{1/(\nu-1)}}. \tag{2.30}$$

Оно имеет полюс порядка $q = (\nu-1)^{-1}$ в точке $t_0 = u_0^{1-\nu}/(\nu-1)$. Формально решение по любой явной либо явно-неявной схеме не имеет вертикальной асимптоты, однако фактически оно очень быстро нарастает вблизи особенности. На рис. 2.10 представлено решение задачи (2.29) по схеме CROS для $q = 1/2$ ($\nu = 3$) на разных сетках по l.

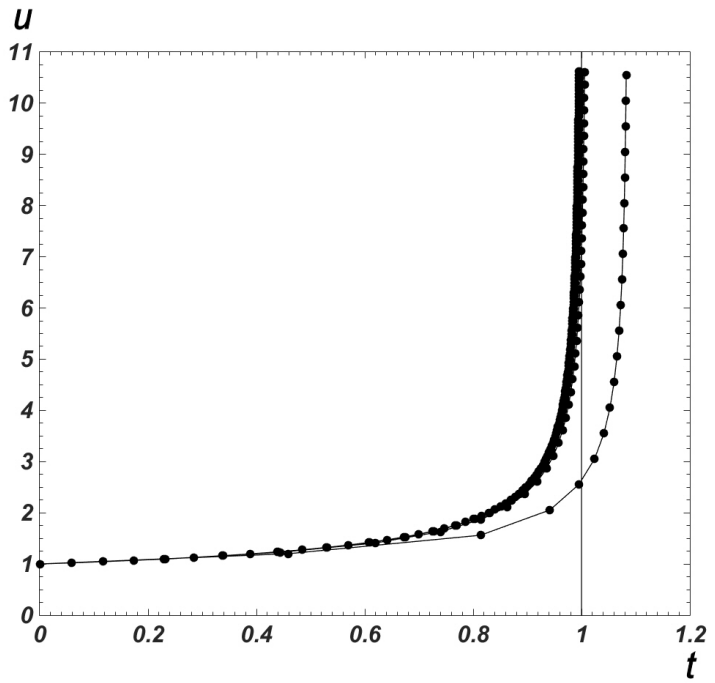

Рис. 2.10. Решения задачи (2.29) для $q = 1/2$, $t_0 = 1$ на сгущающихся сетках. Маркеры – расчетные точки, вертикальная линия – асимптота точного решения (2.30).

После прохождения точки t_0 решение круто уходит вверх. Чем мельче шаг сетки, тем ближе оно ложится к вертикальной асимптоте точного решения.

В окрестности точки t_0 точное решение с полюсом представимо в виде

$$u(t) = \frac{\varphi(t)}{(t_0 - t)^q} \approx \frac{C}{(t_0 - t)^q}, \quad C = \text{const}. \tag{2.31}$$

Дифференцируя его, получим

$$f = \frac{qu}{t_0 - t}. \tag{2.32}$$

Это соотношение справедливо при любых аргументах, и в частности, в расчетные моменты времени t_n. Записав его в узлах n и $n+1$, получим систему алгебраических уравнений относительно q и t_0, которая решается явно

$$q = \frac{t_{n+1} - t_n}{u_n/f_n - u_{n+1}/f_{n+1}}, \quad t_0 = q\frac{u_n}{f_n} + t_n. \tag{2.33}$$

Выражения (2.33) не зависят от параметризации интегральных кривых, поэтому они применимы как при арументе t, так и при аргументе l.

Будем вычислять значения q и t_0 во всех узлах сетки, на рис. 2.11 показаны профили этих величин на сгущающихся сетках. Можно визуально наблюдать, как они сходятся к теоретическим значениям при увеличении n. В прикладных вычислениях рекомендуется строить аналогичные графики.

В аргументе t расчеты по явным и явно-неявным схемам можно формально вести до $t = +\infty$ (поскольку на каждом n-м шаге можно вычислить u_{n+1}). Однако довольно быстро численное решение выходит за пределы представимых чисел, и происходит переполнение. Создается впечатление, что реальный полюс еще не достигнут, хотя

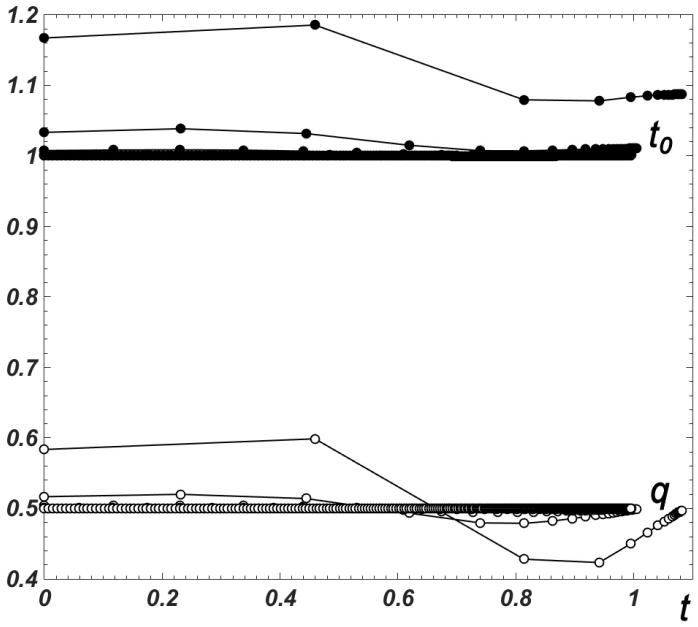

Рис. 2.11. Профили q и t_0 на сгущающихся сетках в задаче (2.29).

на самом деле может быть уже давно пройден. При расчетах по чисто неявным схемам численное решение за полюсом может не существовать (так как нелинейное уравнение относительно u_{n+1} может не иметь решений).

Все это делает расчеты в аргументе t неудобными и ненадежными. Намного удобнее использовать длину дуги, которая неограниченно растет вдоль интегральной кривой. На преимущества длины дуги в этой проблеме указал Г. И. Марчук в 2006 году.

Если на достаточно подробных сетках с увеличением текущей длины дуги значения q и t_0, определяемые из (2.33), выходят на константы, то поведение решения определяется множителем $(t_0 - t)^{-q}$, и можно надежно диагностировать полюс.

В выражения (2.33) входят только сеточные значения u (и f, которые выражаются через них). Поэтому погрешности вычисления q и t_0 определяются только погрешностью u. Таким образом, к q и t_0 можно применить стандартные оценки погрешности по методу Ричардсона. Например, для q

$$\Delta q = \frac{q_N - q_{rN}}{r^p - 1}, \tag{2.34}$$

где N – число узлов более грубой сетки, r – кратность сгущения сетки, p – порядок точности используемой схемы. Оценка для t_0 аналогична. Эти оценки являются асимптотически точными. Поскольку при применении этой процедуры сеточные функции обычно сравнивают поточечно (то есть в совпадающих узлах сеток N и rN), то практически всегда берут $r = 2$.

На рис. 2.12 представлены кривые сходимости u, q, t_0 при сгущении сеток по длине дуги для комплексной схемы Розенброка CROS и явной схемы Рунге-Кутты первого порядка (ERK-1) в зависимости от N. График дан в двойном логарифмическом масштабе, так что степенному характеру зависимости погрешности от числа N шагов сетки соответствует прямая линия. Наклон этой прямой равен порядку точности разностной схемы.

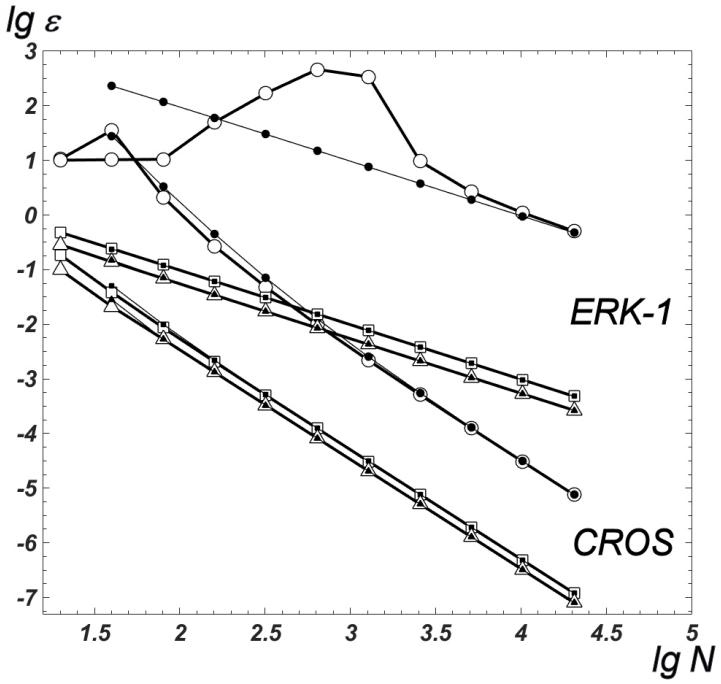

Рис. 2.12. Сходимость в задаче (2.29); $\circ$ – u, $\triangle$ – q, $\square$ – t_0; светлые маркеры – погрешность по точному решению; темные маркеры – оценки точности по методу Ричардсона. Названия схем указаны у кривых.

Нетрудно видеть, что расчетные значения q и t_0 сходятся к теоретическим, причем порядок сходимости равен порядку точности схемы (второй для схемы CROS и первый для ERK-1). Оценки погрешности q и t_0 по методу Ричардсона отлично совпадают с погрешностями на точных значениях этих величин (даже для схемы ERK-1, у которой точность самого решения очень низкая). Поэтому данная методика исключительно надежна.

Практические рекомендации. Для проведения диагностики от схемы требуется одновременно хорошая точность и высокая надежность. Однако явные схемы Рунге-Кутты высокого порядка точности требуют слишком малого шага. Явная схема первого порядка – так как ее точность невелика, а схемы более высокого порядка – из-за невысокой надежности. Поэтому расчеты по явным схемам трудоемки.

Явно-неявная схема CROS4 также оказалась недостаточно надежной несмотря на свои формально высокие теоретические показатели (точность $O(h^4)$ и L_4-устойчивость). Скорее всего, это связано с тем, что каждая из стадий этой схемы не является даже A-устойчивой.

Только схема CROS сочетала неплохую точность $O(h^2)$, L_2-устойчивость и очень высокую надежность. Эта совокупность свойств позволяет рекомендовать именно данную схему для задач диагностики.

Система уравнений. В случае системы ОДУ значения q и t_0 следует вычислять для каждой компоненты, при этом из всех моментов времени $t_0^{(j)}$ следует выбрать наименьший. Например, рассчитывалась система

$$\frac{du}{dt} = \frac{a}{v}, \quad \frac{dv}{dt} = -\frac{b}{u}, \quad u(0) = u_0, \quad v(0) = v_0. \tag{2.35}$$

В ней v имеет полюс порядка $q^{(v)} = b/(a-b)$, а u – нуль порядка $q^{(u)} = -a/(a-b)$ в одной и той же точке $t_0 = -u_0 v_0/(a-b)$. Выберем $a_0 = -1.5$, $b_0 = -0.5$, $u_0 = v_0 = 1$; тогда $t_0 = 1$, $q^{(v)} = 0.5$, $q^{(u)} = 1.5$.

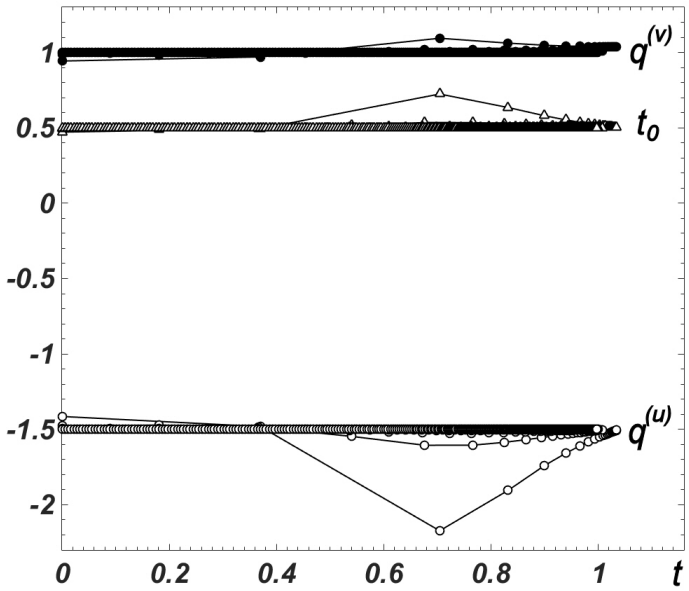

Рис. 2.13. Профили $q^{(u)}$, $q^{(v)}$, t_0 на сгущающихся сетках в задаче (2.35).

Система (2.35) рассчитывалась по схеме CROS в длине дуги. На рис. 2.13 показаны профили t_0, $q^{(u)}$, $q^{(v)}$ на сгущающихся сетках, вычисленные по формулам (2.33). Видно, что поведение профилей t_0 и $q^{(v)}$ аналогично описанному в предыдущем примере, а профили $q^{(u)}$ сходятся к постоянному значению, равному -1.5. Это объясняется тем, что нуль можно рассматривать как полюс отрицательного порядка. Таким образом, поведение численного решения для u и v соответствует теоретическому.

Исследовалось поведение погрешностей в норме C величин t_0, $q^{(u)}$, $q^{(v)}$ на точном решении и их оценок по методу Ричардсона в зависимости от числа узлов N. Оно аналогично расчету по схеме CROS в предыдущем примере. При сгущении сетки по l имела место сходимость с порядком точности $p = 2$; оценки по Ричардсону отлично согласовались с точными значениями погрешности как для u, так и для t_0, $q^{(u)}$, $q^{(v)}$.

Таким образом, предлагаемая процедура позволяет диагностировать особенность типа полюс и для систем уравнений и вычислять его характеристики с гарантированной точностью. Она переносится и на другие типы особенностей, описанные ниже.

2.3.2. Логарифмический полюс.

Метод диагностики. При логарифмической особенности вблизи полюса t_0 точное решение имеет вид

$$u \approx C[\ln(t_0 - t)]^q, \quad C = \text{const}. \tag{2.36}$$

Дифференцируя (2.36), нетрудно получить выражение, связывающее f и u:

$$f = -\frac{qu}{(t_0 - t)\ln(t_0 - t)}. \tag{2.37}$$

Записывая (2.37) в узлах n и $n+1$, получим систему алгебраических уравнений относительно t_0 и q. Она преобразуется к виду

$$\frac{f_n}{u_n}(t_0 - t_n)\ln(t_0 - t_n) = \frac{f_{n+1}}{u_{n+1}}(t_0 - t_{n+1})\ln(t_0 - t_{n+1}), \tag{2.38}$$

$$q = -\frac{f_n}{u_n}(t_0 - t_n)\ln(t_0 - t_n). \tag{2.39}$$

Лемма 1. *Уравнение (2.38) имеет два вещественных корня относительно* t_0. •

Доказательство. Приведем (2.38) к более удобному виду

$$0 = \frac{1}{a}x\ln x - (x+\tau)\ln(x+\tau) \equiv \varphi_1(x) - \varphi_2(x). \tag{2.40}$$

Здесь $a = (f_n u_{n+1})/(f_{n+1}u_n)$, $x = t_0 - t_n > 0$, а $\tau = t_{n+1} - t_n$ – шаг по времени. Вблизи особенности $|f|$ нарастает быстрее, чем $|u|$. Кроме того, все f и u имеют один и тот же знак, а так же $|f_{n+1}| > |f_n|$, $|u_{n+1}| > |u_n|$. Поэтому $0 < a < 1$.

Рассмотрим область малых $x \to 0$, то есть текущая точка t_n находится очень близко от t_0. Тогда $\varphi_1(x) \to 0 - 0$. При этом $\varphi_2(x) \to \tau\ln\tau < 0$ при фиксированном $\tau < 1$. Поэтому $\varphi_1(x) > \varphi_2(x)$ в некоторой окрестности $x = 0$.

Теперь рассмотрим область "средних" x, лежащих вне упомянутой окрестности $x = 0$, но по-прежнему много меньших τ. Легко видеть, что при фиксированных a и τ существует область x, в которой $x\ln x < a\tau\ln\tau$. При таких x $\varphi_1(x) < \varphi_2(x)$. Поэтому на границе областей "малых" и "средних" x имеется корень $x_м < \tau$.

Наконец, рассмотрим область "больших" $x \gg \tau$. Здесь $\varphi_2(x) \approx x\ln x$. Поэтому с учетом того, что $a < 1$, немедленно получаем $\varphi_1(x) > \varphi_2(x)$. Это означает, что имеется еще один корень $x_б > \tau$. ■

Таким образом, доказано существование двух корней уравнения (2.40). Из них следует выбирать тот, который обеспечивает выход q и t_0 на стационары по мере приближения к особенности. На практике проще всего находить их методом Ньютона. Но тогда встает вопрос о выборе начального приближения. В начальный момент времени значение $x = t_0 - t_n$ велико (порядка полного времени расчета T либо полной длины дуги L). Поэтому целесообразно выбрать эти величины в качестве начального приближения, и тогда метод Ньютона сойдется к большему корню $x_б$.

В последующие моменты времени хорошим начальным приближением будет значение x, полученное в предыдущий момент времени. Оно попадает в τ-окрестность искомого корня, и на достаточно подробных сетках метод Ньютона демонстрирует быструю сходимость.

Пример. Рассмотрим задачу

$$\frac{du}{dt} = qu^{1-1/q}\exp\{u^{1/q}\}. \tag{2.41}$$

Точное решение имеет вид $u = [-\ln(t_0 - t)]^q$. При $t < t_0 < 1$ логарифм оказывается отрицательным, и в степень возводится положительное число. Поэтому все вычисления оказываются чисто вещественными. Момент особенности t_0 определяется начальными условиями. Мы брали $q = 1.5$, $t_0 = 0.5$ и подгоняли начальные условия под эти значения.

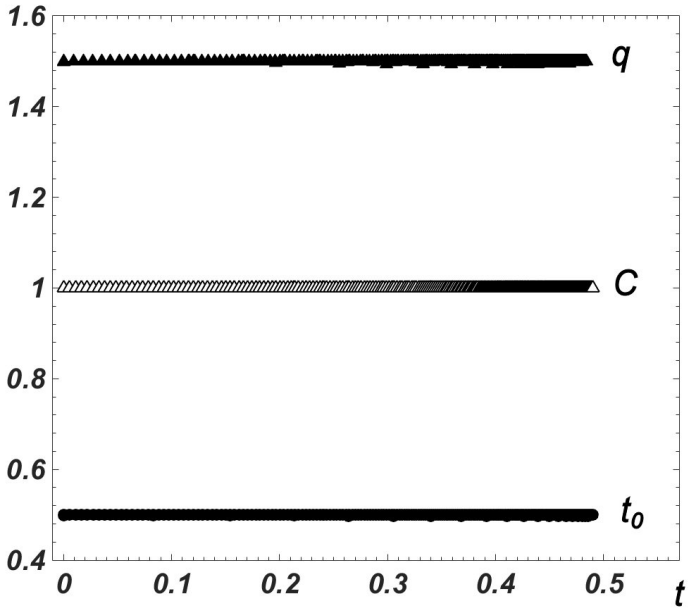

Рис. 2.14. Профили q, t_0 и C на сгущающихся сетках в задачах (2.41) и (2.42).

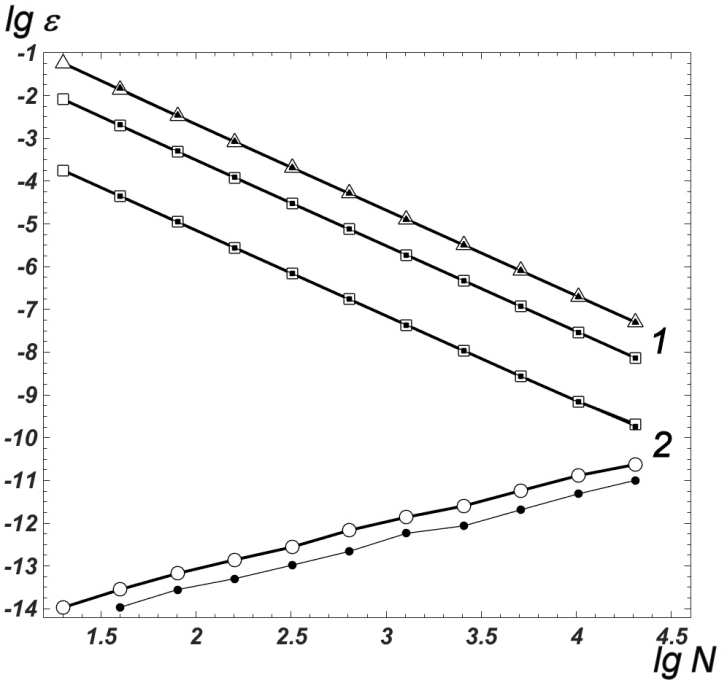

Рис. 2.15. Сходимость: 1 – в задаче (2.41), △ – q, □ – t_0; 2 – в задаче (2.42), ∘ – C, □ – t_0; светлые маркеры – погрешность по точному решению, темные – оценки по точному решению.

Помимо указанного решения имеется также тривиальное $u = \mathrm{const}$, пересекающееся с ним в начальный момент времени. При расчетах по не слишком надежным схемам численное решение может “садиться” на это тривиальное решение. Мы пользовались высоконадежной схемой CROS в длине дуги, в которой таких трудностей не возникало.

Результаты расчетов представлены на рис. 2.14, 2.15. Расчетные профили t_0 и q на сгущающихся сетках выходят на постоянные, равные соответствующим теоре-

тическим значениям. Это доказывает, что характер особенности логарифмический. Погрешности в норме C найденных значений t_0 и q в зависимости от числа узлов N в двойном логарифмическом масштабе выходят на прямые линии с наклоном 2. Поэтому порядок сходимости равнялся $p = 2$, что соответствует теоретическому порядку сходимости для схемы CROS. Кроме того, погрешности по методу Ричардсона хорошо совпадают с погрешностями, определенными непосредственным сравнением с известным точным решением.

Частный случай. При $q = 1$ имеем $u \approx C \ln(t_0 - t)$. Тогда соотношение (2.37) существенно упрощается

$$f = -C \exp\{-u/C\}. \tag{2.42}$$

Записывая (2.42) в узлах n и $n+1$ и деля одно выражение на другое, получим простое уравнение относительно C

$$\frac{f_n}{f_{n+1}} = \exp\{(u_{n+1} - u_n)/C\}, \tag{2.43}$$

откуда

$$C = \frac{u_{n+1} - u_n}{\ln f_n - \ln f_{n+1}}. \tag{2.44}$$

Далее, пользуясь видом $u(t)$ и полученным значением C, находим t_0

$$t_0 = \frac{t_n - t_{n+1} f_{n+1}/f_n}{1 - f_{n+1}/f_n}. \tag{2.45}$$

Данный частный случай интересен постольку, поскольку для C и t_0 удается построить явные выражения, проверка которых удобна в практических расчетах.

Если профили C и t_0, вычисленные по формулам (2.44) – (2.45), выходят на постоянные значения, то можно утверждать, что в момент времени t_0 имеет место логарифмический полюс первого порядка. Сама же величина t_0 определяется начальными условиями. В демонстрационном расчете мы брали $C = 1$, $t_0 = 0.5$.

Как и в предыдущих расчетах, здесь использовалась схема CROS в длине дуги. Профили C и t_0 на сгущающихся сетках показаны на рис. 2.14. Видно, что они отлично ложатся на постоянные значения, равные 1 и 0.5 соответственно. При этом значение t_0 сходится к теоретическому со вторым порядком точности, так как кривая погрешности в двойном логарифмическом масштабе представляет из себя прямую линию с наклоном 2 (см. рис. 2.15). Погрешности величины C принимают очень малые значения (порядка $10^{-14} \div 10^{-10.5}$) и нарастают при увеличении числа узлов N. Причина этого в том, что величина C совпадает со своим теоретическим значением с точностью до ошибок округления, которые нарастают при сгущении сеток.

2.3.3. Смешанная особенность.

Метод диагностики. Данная особенность представима в виде

$$u \approx C \frac{\ln(t_0 - t)}{(t - t_0)^q}, \quad C = \text{const}. \tag{2.46}$$

В этом случае имеем

$$f = -\frac{u}{(t_0 - t)\ln(t_0 - t)} + \frac{qu}{t_0 - t}. \tag{2.47}$$

Записывая (2.47) в узлах n и $n+1$, получаем алгебраические уравнения относительно t_0 и q

$$\begin{gathered}[(t_0 - t_{n+1})\ln(t_0 - t_{n+1})f_{n+1}/u_{n+1} + 1]\ln(t_0 - t_n) = \\ = [(t_0 - t_n)\ln(t_0 - t_n)f_n/u_n + 1]\ln(t_0 - t_{n+1}),\end{gathered} \tag{2.48}$$

$$q = (t_0 - t_n)\frac{f_n}{u_n} + \frac{1}{\ln(t_0 - t_n)}. \tag{2.49}$$

Лемма 2. *Уравнение (2.48) имеет два вещественных корня относительно* t_0. •

Доказательство. Перепишем (2.48) в более удобном виде

$$0 = \left[\frac{f_{n+1}}{u_{n+1}} x\ln x + 1\right]\ln(x+\tau) - \left[\frac{f_n}{u_n}(x+\tau)\ln(x+\tau) + 1\right]\ln x \equiv \varphi_1(x) - \varphi_2(x). \tag{2.50}$$

Здесь по-прежнему $x = t_0 - t_n > 0$, а $\tau = t_{n+1} - t_n$.

Пусть $x \to 0$. Тогда $\varphi_1(x) \to \ln\tau < 0$ – фиксированная величина. При этом $\varphi_2(x) \sim A\ln x$, где $A = (f_n/u_n)\,\tau\ln\tau + 1 = \text{const}$. Если сетка достаточно подробная (то есть $\tau \sim 1/N$ достаточно мало), то $A > 0$. Это значит, что $\varphi_2(x) \to -\infty$. Таким образом, в некоторой малой окрестности $x = 0$ имеем $\varphi_1(x) > \varphi_2(x)$.

Далее, положим $x = \tau$ и исследуем знак выражения (2.50). Оно преобразуется к виду

$$\varphi_1(x) - \varphi_2(x) = \left(\frac{f_{n+1}}{u_{n+1}} - 2\frac{f_n}{u_n}\right) x\ln x\ln 2x + \ln 2. \tag{2.51}$$

Если сетка достаточно подробная, то с хорошей точностью справедливы равенства $f_{n+1} = f(t_n + \tau) \approx f_n + \tau f_n'$, $u_{n+1} = u(t_n + \tau) \approx u_n + \tau f_n$. Подставляя эти выражения в (2.51), нетрудно убедиться, что выражение в круглой скобке отрицательно и равно по порядку величины $\sim -f_n/u_n + O(\tau)$. Поэтому достаточно близко от особенности первое слагаемое в (2.51) оказывается больше по модулю, чем $\ln 2 \approx 0.7$. Таким образом, в некоторой окрестности $x = \tau$ имеем $\varphi_1(x) < \varphi_2(x)$. Следовательно, существует первый (меньший) корень $x_м < \tau$.

Пусть теперь $x \gg \tau$ достаточно велико. Тогда $\varphi_1(x) \approx [(f_{n+1}/u_{n+1})x\ln x + 1]\ln x$, $\varphi_2(x) \approx [(f_n/u_n)x\ln x + 1]\ln x$. Выше при исследовании логарифмической особенности было замечено, что $f_{n+1}/u_{n+1} > f_n/u_n$. Поэтому $\varphi_1(x) > \varphi_2(x)$ при достаточно больших x. Отсюда следует, что существует второй (больший) корень $x_б > \tau$. ■

На практике корни уравнения (2.50) удобно находить методом Ньютона. Начальное приближение можно выбирать так же, как в задаче с логарифмическим полюсом (см. п. 2.3.2).

Пример. Построить автономную задачу с рассматриваемой особенностью не удалось, поэтому рассмотрим следующую неавтономную задачу:

$$\frac{du}{dt} = \left[-\frac{u}{\ln(t_0 - t)}\right]^{1+1/q} + q\frac{u}{t_0 - t}. \tag{2.52}$$

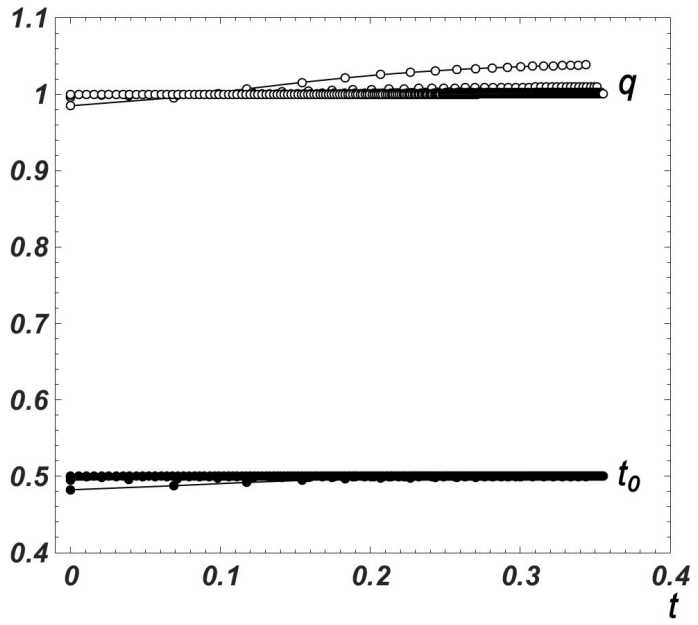

Рис. 2.16. Профили q и t_0 и C на сгущающихся сетках в задаче (2.52).

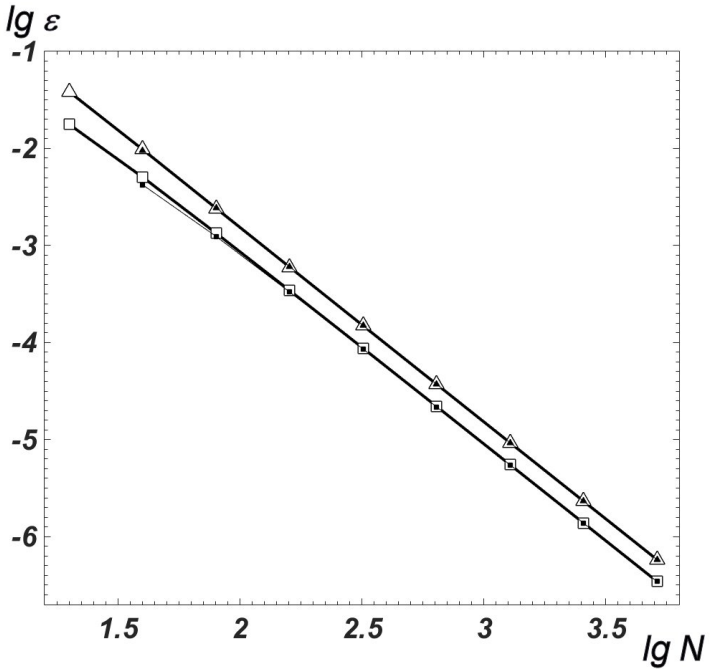

Рис. 2.17. Сходимость в задаче (2.52); обозначения соответствуют рис. 2.12.

Точное решение имеет вид $u = -\ln(t_0 - t)/(t_0 - t)^q$. При $t < t_0 < 1$ логарифм отрицателен, решение $u > 0$ положительно, поэтому все вычисления чисто вещественные. Положение особенности t_0 определяется начальными условиями. В демонстрационном расчете мы брали $q = 1$, $t_0 = 0.5$ и соответственно подгоняли начальные условия.

Как и ранее, мы использовали схему CROS в длине дуги. Результаты расчетов аналогичны предыдущим случаям (см. рис. 2.16, 2.17). Профили расчетных t_0 и q на сгущающихся сетках стремятся к постоянным значениям при увеличении n. С графической точностью эти значения совпадают с теоретическими на всех сетках, начиная со второй. Анализ дальнейших знаков показывает, что имеет место схо-

димость этих величин к теоретическим при сгущении сеток, причем порядок этой сходимости равен $p = 2$.

2.3.4. Неизвестная особенность. В предыдущих разделах были подробно разобраны важнейшие виды особенностей. Однако при расчете реальных задач тип особенности заранее неизвестен. Кроме того особенность может оказаться более сложной, чем рассмотренные.

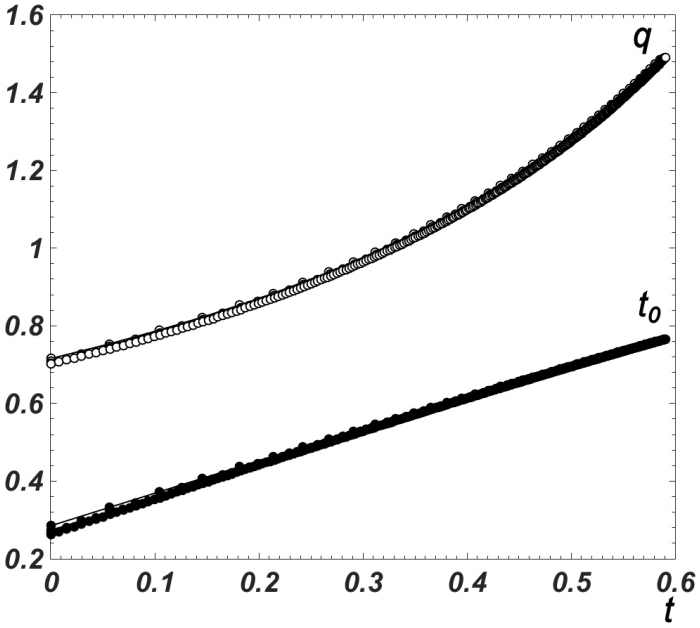

Рис. 2.18. Диагностика задачи (2.29) при $q = 2$, $t_0 = 1$ по формулам (2.38) – (2.39).

На практике рекомендуется проверять все 3 типа особенностей: вычислять значения q и t_0 по формулам (2.33), (2.38) – (2.39), (2.48) – (2.49) и строить профили этих величин. Если при выбранной гипотезе о типе особенности профили выходят на постоянные значения, то гипотеза о типе особенности подтверждается. Если же мгновенное значение q и t_0 меняется по мере приближения к особенности, то это значит, что выбранная гипотеза о ее типе неверна.

Пример такой "ошибочной" диагностики представлен на рис. 2.18. Мы взяли тестовую задачу (2.29) со степенным полюсом порядка $q = 2$ в момент $t_0 = 1$ и применили к ней формулы (2.38) – (2.39) для логарифмического полюса. В результате в расчетном промежутке времени мгновенное значение t_0 меняется в 3 раза, а мгновенное значение q – более, чем в 2 раза, причем по мере приближения к особенности оно меняется все более круто.

Если профили величин q и t_0 не выходят на строго постоянные, но их вариация вблизи особенности не слишком велика (например, не превышает 10-15%), то тип особенности близок к одному из указанных, но сама особенность отличается от него дополнительными меньшими членами. Тогда можно говорить об эффективных значениях q и t_0. В качестве них разумно брать значения в последнем узле $(q)_N$, $(t_0)_N$, который наиболее близок к особенности.

2.4 S-режим нелинейного горения

Нелинейное горение. Предлагаемая методика применялась к исследованию S-режима нелинейного горения, который описывается нелинейным параболическим уравнением

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(u^2 \frac{\partial u}{\partial x}\right) + u^3. \tag{2.53}$$

Это уравнение имеет точное решение

$$u(x,t) = \frac{\sqrt{3}}{2}\frac{1}{\sqrt{t_0 - t}}\cos\left(\frac{x}{\sqrt{3}}\right). \tag{2.54}$$

Температурные профили этого решения в фиксированные моменты времени показаны на рис. 2.19. Поскольку переменные разделяются, то особенность типа полюс

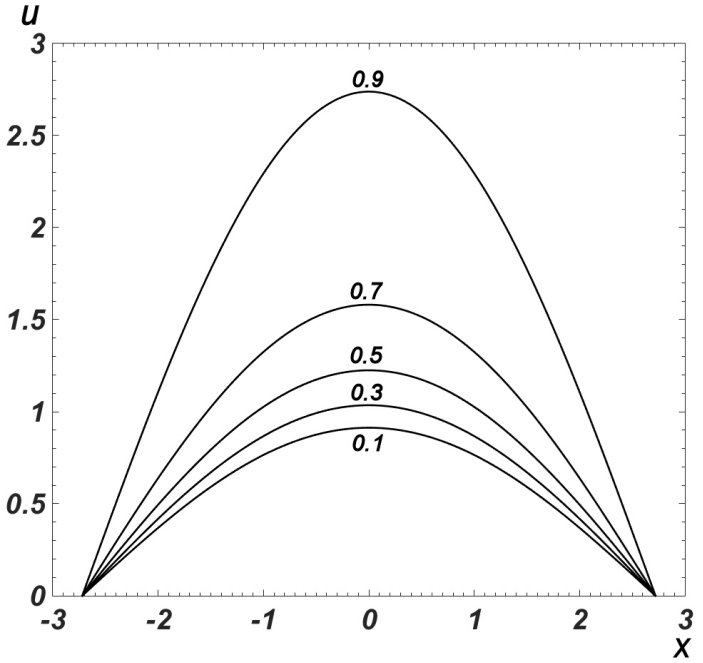

Рис. 2.19. Профили решения (2.54) в фиксированные моменты времени (указаны у кривых).

порядка $q = 1/2$ имеет место в каждой точке пространства, причем на всем отрезке решение разрушается одновременно. Иными словами, при каждом x зависимость температуры от времени имеет вид, показанный на рис. 2.10.

Методом прямых уравнение (2.53) сводится к системе ОДУ

$$\frac{du_j}{dt} = \frac{1}{2h_x^2}\left[(u_{j+1}^2 + u_j^2)(u_{j+1} - u_j) - (u_j^2 + u_{j-1}^2)(u_j - u_{j-1})\right] + u_j^3, \tag{2.55}$$

где j и h_x – номер узла и шаг по пространству соответственно. Нетрудно убедиться, что пространственный оператор в (2.53) аппроксимируется с точностью $O(h_x^2)$. Система (2.55) содержит несколько сотен компонент (для получения хорошей точности по пространству), а ее правые части весьма сложны. Поэтому такой тест достаточно представителен.

Сгущение сеток по l. Вычисления проводились по схеме CROS, имеющей второй порядок точности и обладающей L_2-устойчивостью и чрезвычайно высокой надежностью. Сетка по x содержала $J = 200$ узлов, так что система (2.55) имела огромный порядок.

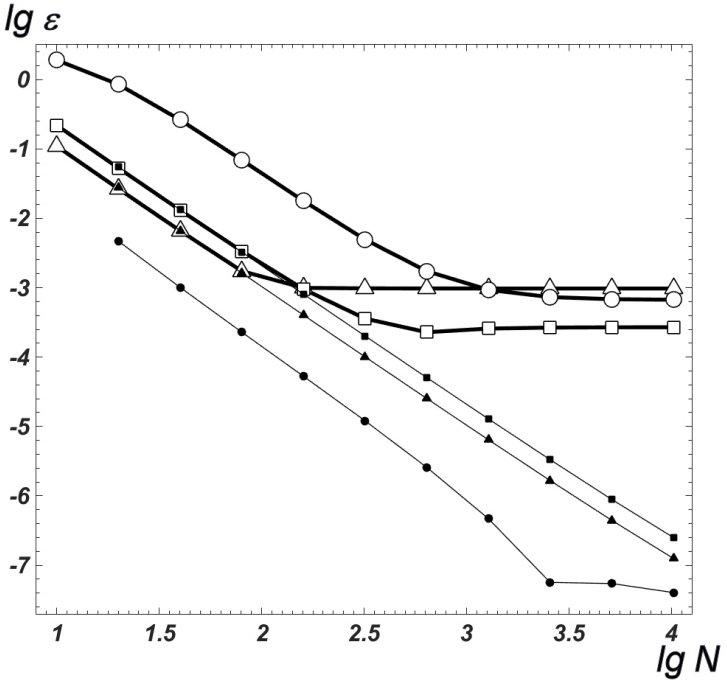

Рис. 2.20. Сходимость в задаче (2.55), расчет по схеме CROS. Обозначения соответствуют рис. 2.12. В качестве точного решения выбрано (2.54).

На рис. 2.20 показаны кривые сходимости u, t_0, q при сгущении сеток по длине дуги. Четко видно, что все 3 величины сходятся со вторым порядком точности. Однако, начиная с некоторой достаточно подробной сетки, кривые погрешности, вычисленные сравнением с точным решением (2.54), выходят на горизонтальный участок на уровне $\sim 10^{-3}$. При этом оценки точности по Ричардсону, относящиеся к системе ОДУ (2.55), продолжают сходиться со вторым порядком вплоть до фона ошибок округления (показан для u).

Это связано с тем, что при введении дискретизации по пространству мы вносим некоторую погрешность, и поэтому полюс точного решения (2.54) отличается от полюсов системы (2.55), либо она их может вовсе не иметь. В последнем случае профили q и t_0 – некоторые кривые не обязательно выходящие на постоянные. Чтобы уменьшить это отличие, нужно вводить более подробные сетки по x. По этой же причине погрешность на точном решении для u не сопадает с погрешностью, полученной методом Ричардсона применительно к системе (2.55). Погрешность, вносимую дискретизацией по x, также можно оценить методом сгущения сеток.

Сгущение сеток по x. При сгущении сеток по пространству увеличивается число компонент в системе (2.55). Если вести расчет этой системы до фиксированной длины дуги, то при увеличении числа компонент длина каждой интегральной кривой уменьшится, и расчетный интервал времени сократится.

Поэтому целесообразно вести расчет до достижения заданного расчетного времени. Тогда добавление новых компонент приводит к увеличению суммарной длины всех интегральных кривых. В результате узлы сеток по l, относящиеся к разным сеткам по x, не будут совпадать. Это не позволяет проводить поточечные сравнения сеточных функций, однако мы будем сравнивать только значения $(q)_N$ и $(t_0)_N$ в последнем N-м узле.

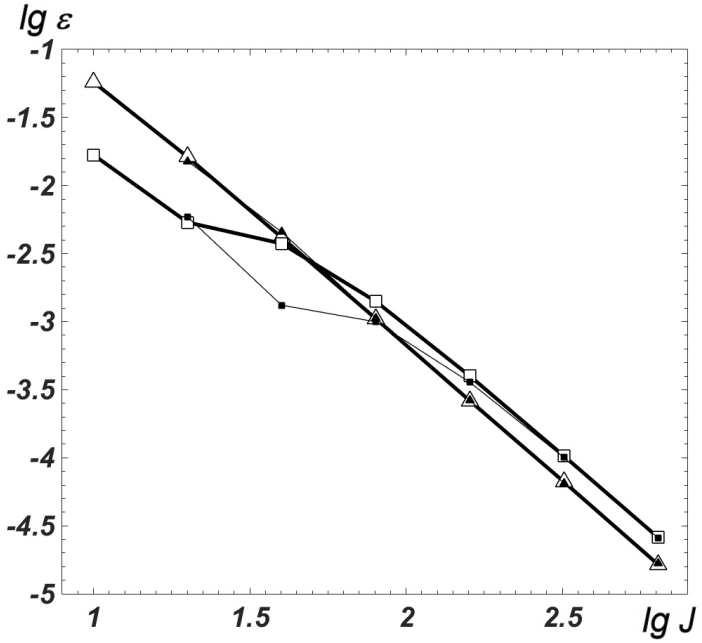

Рис. 2.21. Сходимость в задаче (2.53) при сгущении сеток по x. Обозначения соответствуют рис. 2.12.

Подчеркнем, что для исследования порядка точности по пространству сетки по l должны быть настолько подробными, чтобы погрешности q и t_0 на точном решении достигали горизонтального участка (см. рис. 2.20). Аппроксимация оператора по пространству есть $O(h_x^2)$, поэтому при сгущении сетки по x вдвое высота этого горизонтального участка должна уменьшиться в 4 раза.

На рис. 2.21 приведены кривые сходимости q и t_0 при сгущении сеток по x. Представлены как погрешности, вычисленные сравнением с точным решением, так и их оценки по методу Ричардсона. Видно, что на грубых сетках сходимость является нерегулярной, однако начиная с $J \approx 120$ кривые выходят на прямолинейные участки с наклоном $p = 2$. Это соответствует регулярной сходимости. При этом на регулярном участке оценки по методу Ричардсона отлично совпадают с фактической погрешностью, вычисленной сравнением с точным решением.

Таким образом, предложенная методика является более простой, надежной и удобной для практического применения, чем известные ранее методы.

2.5 Основные результаты главы

1. Предложен новый специализированный численный метод для задач кинетики. Этот метод явный, и его трудоемкость очень мала. Он имеет более высокий порядок точности (второй) и одновременно является более надежным, чем ранее известные методы. На представительном примере показано, что для данного типа задач этот метод является высокоэффективным. Метод позволяет проводить вычисления одновременно с нахождением гарантированной оценки математической погрешности и пригоден к включению в газодинамические пакеты программ.
2. Предложены новые простые и надежные способы диагностики особенностей типа полюс, логарифмический полюс и смешанной особенности для систем обыкновенных дифференциальных уравнений. Они позволяют вычислять характеристики этих особенностей с апостериорной асимптотически точной оценкой погрешности. Методика применима при произвольной параметризации интегральной кривой, в том числе через длину дуги, которая оптимальна при решении жестких и плохо обусловленных задач. Предлагаемый подход применим и к нелинейным уравнениям в частных производных, поскольку они сводятся методом прямых к системам ОДУ огромного порядка.

3. Уравнение Пуассона

В данной главе оптимизирован сверхбыстрый итерационный метод для счета на установление по эволюционно-факторизованной схеме с логарифмическим набором шагов. Предложена практически неулучшаемая производящая функция этого набора и апостериорные асимптотически точные оценки итерационного процесса. Метод позволяет экономично решать эллиптические уравнения на многократно сгущающихся сетках по пространству и получать высокую точность.

3.1 Эволюционная факторизация

3.1.1. Схема. Рассмотрим параболическое уравнение $u_t = Lu + f$ для пространственного оператора $L = \sum L_\alpha$, расщепляющегося на одномерные операторы. Для решения этой задачи можно написать схему "с полусуммой":

$$\frac{\hat{u} - u}{\tau} = \sum_\alpha \Lambda_\alpha \frac{\hat{u} + u}{2} + f, \tag{3.1}$$

где $\hat{u}$ есть решение на новом временном слое. Эта схема безусловно устойчива и имеет аппроксимацию $O\left(\tau^2 + \sum h_\alpha^2\right)$. Однако схема (3.1) приводит к решению системы с ленточной матрицей $\sim N^3$, у которой ширина ленты $\sim N$ в двумерном и $\sim N^2$ в трехмерном случаях, что чрезмерно трудоемко. Поэтому она неэкономична.

Двумерные задачи успешно решаются известной схемой переменных направлений Писмена-Рэкфорда. Однако эта схема не обобщается на трехмерный случай. Кроме того, для получения второго порядка точности нужна нетривиальная форма аппроксимации граничных условий на промежуточном шаге.

Для трехмерных задач созданы различные методы приближенной факторизации [49] и локально-одномерные методы [50]. В них также трудно получить второй порядок точности как из-за проблемы написания граничных условий на промежуточных шагах, так и из-за необходимости симметризовать порядок выполнения промежуточных шагов.

Эти недостатки удается преодолеть в методе эволюционной факторизации [16], [17]. Заменим в уравнении (3.1) полусумму $0.5\left(\hat{u} + u\right) = u + 0.5\tau\left(\hat{u} - u\right)/\tau$ и перенесем слагаемые, содержащие производную по времени, в левую часть:

$$\left(E - \frac{\tau}{2}\sum_\alpha \Lambda_\alpha\right)\frac{\hat{u} - u}{\tau} = \sum_\alpha \Lambda_\alpha u + f. \tag{3.2}$$

Схема (3.2) эквивалентна схеме (3.1) и является неэкономичной.

Приближенно факторизуя оператор в левой части (3.2), получим новую схему, которую назовем ***эволюционно-факторизованной***:

$$\prod_{\alpha}\left(E-\frac{\tau}{2}\Lambda_{\alpha}\right)\frac{\hat{u}-u}{\tau}=\sum_{\alpha}\Lambda_{\alpha}u+f. \tag{3.3}$$

Исследуем свойства этой схемы.

3.1.2. Алгоритм. Рассмотрим наиболее сложный трехмерный случай. Введем вспомогательные сеточные функции v и w и перепишем (3.3) в виде трех уравнений:

$$\left(E-\tau\Lambda_x/2\right)w=\left(\Lambda_x+\Lambda_y+\Lambda_z\right)u+f, \tag{3.4}$$

$$\left(E-\tau\Lambda_y/2\right)v=w, \tag{3.5}$$

$$\left(E-\tau\Lambda_z/2\right)\left(\hat{u}-u\right)/\tau=v. \tag{3.6}$$

Правая часть (3.4) известна, поскольку вычисляется на предыдущем слое. Оператор в левой части этого уравнения является трехдиагональным по направлению x, а потому обращается одномерной прогонкой при каждых фиксированных сеточных y и z. Аналогично трехдиагональным в направлении y является оператор в левой части (3.5). Поэтому его можно обратить одномерной прогонкой при каждых фиксированных сеточных x и z и по вычисленному w найти v. Подставляя это значение v в правую часть (3.6), одномерной прогонкой по z при каждых фиксированных сеточных x и y можно найти $\left(\hat{u}-u\right)/\tau$. Таким образом, нахождение $\hat{u}$ сводится к последовательности трех одномерных прогонок. Это означает, что схема экономична.

Переход к двумерному случаю осуществляется вычеркиванием (3.6) и заменой v на $\left(\hat{u}-u\right)/\tau$ в (3.5). Заметим, что если в схеме Писмена-Рэкфорда исключить промежуточный слой, то она в точности совпадет с двумерной эволюционно-факторизованной схемой.

3.1.3. Аппроксимация. Вычтем схему (3.3) из схемы (3.2). Главный член разности есть $\tau^2/4\sum\Lambda_{\alpha}\Lambda_{\beta}\left(\hat{u}-u\right)/\tau\approx\tau^2/4\sum\Lambda_{\alpha}\Lambda_{\beta}u_t=O\left(\tau^2\right)$. Но схема (3.1) имеет аппроксимацию $O\left(\tau^2+\sum h_{\alpha}^2\right)$. Следовательно, схема (3.3) также аппроксимирует дифференциальное уравнение с порядком $O\left(\tau^2+\sum h_{\alpha}^2\right)$. Заметим, что для этого решение должно иметь пятые непрерывные производные u_{xxxxt}, u_{yyyyt}, u_{zzzzt}; в то время как для схемы с полусуммой достаточно четвертых непрерывных производных u_{xxxx}, u_{yyyy}, u_{zzzz}.

3.1.4. Граничные условия. Для эволюционно-факторизованной схемы они приведены в [51]. Условия первого рода для прогонки по направлению z задаются тривиально, поскольку значения u на границе полагаются известными во все моменты времени. Для прогонки по направлению y граничные условия выразим из (3.6):

$$\begin{gathered}\left[v\right]_{\text{гран.}}=\left[\left(E-\tau\Lambda_z/2\right)\left(\hat{u}-u\right)/\tau\right]_{\text{гран.}}\approx\\ \approx\left[\left(E-\tau/2\cdot\partial/\partial z\left(k_z\partial/\partial z\right)\right)\left(\hat{u}-u\right)/\tau\right]_{\text{гран.}}.\end{gathered} \tag{3.7}$$

В последнем переходе разностное дифференцирование заменено второй производной с точностью $O\left(h_z^2\right)$. Дифференцирование граничного условия нужно выполнять точно. Условие (3.7) вносит дополнительную погрешность $O\left(\tau h_z^2\right)$ третьего порядка малости, которой можно пренебречь.

Граничные условия для w получаются из (3.5) с учетом (3.6):

$$\begin{gathered}[w]_{\text{гран.}} = \left[\left(E-\tau\Lambda_y/2\right)\left(E-\tau\Lambda_z/2\right)\left(\hat{u}-u\right)/\tau\right]_{\text{гран.}} \approx \\ \approx \left[\left(E-\tau\Lambda_y/2-\tau\Lambda_z/2\right)\left(\hat{u}-u\right)/\tau\right]_{\text{гран.}} \approx \\ \approx \left[\left(E-\tau/2\cdot\partial/\partial y\left(k_y\partial/\partial y\right)-\tau/2\cdot\partial/\partial z\left(k_z\partial/\partial z\right)\right)\left(\hat{u}-u\right)/\tau\right]_{\text{гран.}}.\end{gathered} \tag{3.8}$$

Здесь отброшен член $\tau^2\Lambda_y\Lambda_z/4$ порядка $O\left(\tau^2\right)$, разностное дифференцирование заменено точным. Очевидно, это не ухудшает порядок аппроксимации всей схемы. Таким образом, эволюционно-факторизованная схема с описанными граничными условиями обеспечивает аппроксимацию $O\left(\tau^2+\sum h_\alpha^2\right)$.

Для двумерного случая граничные условия выписываются аналогично. Это показывает, что граничные условия для схемы Писмена-Рэкфорда лучше писать без использования промежуточного слоя.

3.1.5. Устойчивость. Одномерные операторы $\Lambda_\alpha < 0$, но в качестве их собственных значений нам удобно выбрать положительные величины λ_α. Тогда множитель роста трехмерной гармоники в (3.3) имеет вид

$$\left(1+\tau\lambda_x/2\right)\left(1+\tau\lambda_y/2\right)\left(1+\tau\lambda_z/2\right)\left(\rho-1\right) = -\tau\left(\lambda_x+\lambda_y+\lambda_z\right). \tag{3.9}$$

Из (3.9) получаем двумерный случай, полагая $\lambda_z = 0$. В одномерном случае надо также полагать $\lambda_y = 0$. Отсюда одномерные, двумерные и трехмерные множители роста имеют вид соответственно

$$\rho = \frac{1-\tau\lambda_x/2}{1+\tau\lambda_x/2}, \tag{3.10}$$

$$\rho = \frac{\left(1-\tau\lambda_x/2\right)}{\left(1+\tau\lambda_x/2\right)}\frac{\left(1-\tau\lambda_y/2\right)}{\left(1+\tau\lambda_y/2\right)}, \tag{3.11}$$

$$\rho = 1-\frac{\tau\left(\lambda_x+\lambda_y+\lambda_z\right)}{\left(1+\tau\lambda_x/2\right)\left(1+\tau\lambda_y/2\right)\left(1+\tau\lambda_z/2\right)}. \tag{3.12}$$

Во всех случаях $|\rho| < 1$. Для (3.10) и (3.11) это очевидно, для (3.12) легко проверяется. Это обеспечивает безусловную устойчивость схемы (3.3) при любом числе измерений. Однако ее асимптотическая устойчивость является лишь условной. Это непосредственно видно для одномерного и двумерного случая по структуре множителей роста, которые таковы же, как для схемы “с полусуммой”. Для трехмерного случая это нетрудно доказать.

3.1.6. Двойная факторизация. Отметим еще один способ факторизации, называемый ***двойной факторизацией*** [16]. Умножим обе части (3.1) на τ. Перенесем все

слагаемые, содержащие $\hat{u}$, в левую часть, а все слагаемые, содержащие u, – в правую. Приближенно факторизуем операторы в обеих частях получившегося уравнения

$$\prod_{\alpha} (E - \tau\Lambda_\alpha/2)\,\hat{u} = \prod_{\alpha} (E + \tau\Lambda_\alpha/2)\,u + \tau f. \tag{3.13}$$

Доказательство экономичности, аппроксимации с порядком $O\left(\tau^2 + \sum h_\alpha^2\right)$ и построение граничных условий для этой схемы аналогичны схеме эволюционной факторизации.

Нетрудно убедиться, что для произвольного числа измерений множитель роста многомерной гармоники в точности равен произведению одномерных множителей

$$\rho = \prod_{\alpha} \frac{(E - \tau\lambda_\alpha/2)}{(E + \tau\lambda_\alpha/2)}, \tag{3.14}$$

что представляется заманчивым. В частности, из (3.14) следует безусловная устойчивость и условная асимптотическая устойчивость.

Нетрудно заметить, что в двумерном случае эволюционная и двойная факторизации точно совпадают. В трехмерном случае они различаются. В частности, схема двойной факторизации не подходит для счета на установление. В этом случае в пределе $\hat{u} = u$, и схема принимает вид

$$\left(\Lambda_x + \Lambda_y + \Lambda_z + \tau^2\Lambda_x\Lambda_y\Lambda_z/4\right) u + f = 0, \tag{3.15}$$

что означает лишь условную аппроксимацию исходного эллиптического уравнения. Перспективных приложений для этой схемы пока не найдено.

3.2 Счет на установление

3.2.1. Стационарное решение. Эллиптическое уравнение $Lu = -f$ можно рассматривать как стационарный предел при $t \to \infty$ для параболического уравнения $u_t = Lu + f$ со стационарными граничными условиями и правой частью. Для разностного уравнения (1.3) это означает решение эволюционной задачи по схеме (3.1) при выполнении условия асимптотической устойчивости. Стационарный предел решения задачи (3.1) есть решение системы (1.3). Такой прием решения эллиптических уравнений называется ***счетом на установление***.

При счете на установление выбирают произвольные начальные данные. Для расчета берут какой-либо факторизованный вариант схемы (3.1) и считают до тех пор, пока левая часть факторизованной схемы не станет достаточно малой. Число необходимых для этого итераций существенно зависит от того, насколько удачно выбран набор шагов по времени. Поэтому актуальной является задача о построении набора, который обеспечивал бы наиболее быструю сходимость.

3.2.2. Оптимальный набор шагов. Для явных схем оптимальным является чебышевский набор шагов, но построенный не для τ, а для величины $1/\tau$. Эта схема лишь условно устойчива, поэтому не любые перестановки шагов допустимы. Для

неявной продольно-поперечной схемы был найден постоянный оптимальный шаг $\tau_0 \approx$ $\approx \sqrt{\tau_{\min}\tau_{\max}}$ [14]. Обобщение этой идеи дано в [18], где для набора шагов употребляется преобразование $\ln\tau$. Такой выбор обеспечивает логарифмическую скорость сходимости, которая, по-видимому, является наибольшей из достижимых скоростей.

Ранее логарифмической сходимостью обладали только узкоспециализированные методы вроде быстрого преобразования Фурье, но они применимы только для тепличных условий: постоянные k, h и специфические числа узлов 2^r. Область применимости логарифмического набора, описанная во введении, значительно шире.

Вопрос о построении оптимального логарифмического набора состоит из двух частей: ***1°*** построение границ этого набора для разного числа измерений и ***2°*** выбор порождающей функции.

3.3 Логарифмические наборы

3.3.1. Границы логарифмического набора. Оценим граничные шаги $\tau_{\min}$, $\tau_{\max}$ для разного числа измерений. Естественно выбрать их так, чтобы они максимально гасили множители роста, соответствующие границам спектрального интервала.

Одномерный и двумерный случаи. Из формул множителей роста видно, что в одномерном случае

$$\tau_{\min} = \frac{2}{\lambda_{x,\max}}, \quad \tau_{\max} = \frac{2}{\lambda_{x,\min}}, \tag{3.16}$$

а в двумерном

$$\tau_{\min} = \frac{2}{\max\{\lambda_{x,\max}, \lambda_{y,\max}\}}, \quad \tau_{\max} = \frac{2}{\min\{\lambda_{x,\min}, \lambda_{y,\min}\}}. \tag{3.17}$$

При этом граничные множители роста точно равны нулю.

В трехмерном случае легко проверяется следующее. При $\tau > 0$ зависимость $\rho(\tau)$ имеет единственный минимум, удовлетворяющий приведенному кубическому уравнению

$$(2/\tau)^3 - b\,(2/\tau) - 2c = 0, \tag{3.18}$$

где $b = \lambda_x\lambda_y + \lambda_x\lambda_z + \lambda_y\lambda_z$, $c = \lambda_x\lambda_y\lambda_z$. Искомый (вещественный положительный) корень вычисляется явно по тригонометрическому варианту формулы Кардано:

$$\frac{2}{\tau} = -2\sqrt{\frac{b}{3}}\cos\left(\varphi + \frac{2\pi}{3}\right), \quad \varphi = \frac{1}{3}\arccos\left(-c\,(b/3)^{-3/2}\right). \tag{3.19}$$

Подставим в (3.18) $\lambda_{\alpha,\max}$. Если для полученного корня τ_* имеет место $\rho(\tau_*) \geqslant 0$, то положим $\tau_{\min} = \tau_*$ (см. рис. 3.1, а). Если $\rho(\tau_*) < 0$ (см. рис. 3.1, б), то множитель роста имеет два вещественных положительных корня $0 < \tau_- < \tau_+$ (третий корень отрицателен). Эти корни удовлетворяют кубическому уравнению

$$(2/\tau)^3 - a\,(2/\tau)^2 + b\,(2/\tau) + c = 0, \tag{3.20}$$

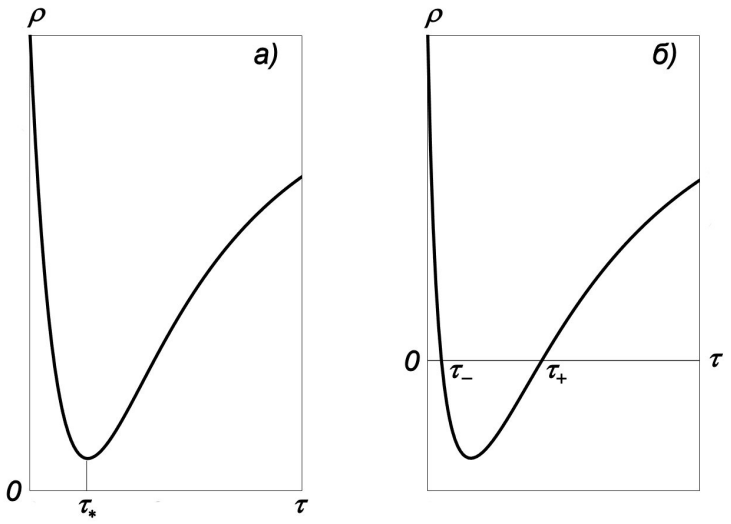

Рис. 3.1. Множитель роста трехмерной гармоники.

где $a = \lambda_x + \lambda_y + \lambda_z$. Они находятся чисто вещественными вычислениями по тригонометрическому варианту формулы Кардано

$$\frac{2}{\tau_\pm} = \frac{a}{3} - 2\sqrt{\frac{f}{3}}\cos\left(\theta \mp \frac{2\pi}{3}\right), \quad \theta = \frac{1}{3}\arccos\left(g/2\ (f/3)^{-3/2}\right). \tag{3.21}$$

Здесь введены обозначения

$$f = 1/2\,(\lambda_x - \lambda_y)^2 + 1/2\,(\lambda_x - \lambda_z)^2 + 1/2\,(\lambda_y - \lambda_z)^2, \tag{3.22}$$

$$\begin{gathered} g = \lambda_x^2(-2\lambda_x + 3\lambda_y + 3\lambda_z)/27 + \lambda_y^2(-2\lambda_y + 3\lambda_z + 3\lambda_x)/27 + \\ +\lambda_z^2(-2\lambda_z + 3\lambda_x + 3\lambda_y)/27 + 14/9\,\lambda_x\lambda_y\lambda_z. \end{gathered} \tag{3.23}$$

В этом случае положим $\tau_{\min} = \tau_-$. Аналогично выглядит процедура вычисления $\tau_{\max}$ по значениям $\lambda_{\alpha,\min}$. Если $\rho(\tau_*) \geqslant 0$, то аналогия полная. Если $\rho(\tau_*) < 0$, то следует положить $\tau_{\max} = \tau_+$.

Частные случаи. Отметим частные случаи трехмерной задачи. Если $\lambda_x = \lambda_y = \lambda_z$, получим $\tau_{\min} = 1/\lambda_{\max}$, $\tau_{\max} = 1/\lambda_{\min}$. В этом случае для крайних гармоник $\rho = 1/9$, что обеспечивает хорошее убывание погрешности за один шаг. Если k_α/h_α^2 по разным направлениям многократно отличаются, то $\rho(\tau_*) < 0$ как для $\tau_{\min}$, так и для $\tau_{\max}$. Заметим также, что если $\rho(\tau_*) < 0$, то соответствующие гармоники гасятся лучше, чем при $\rho(\tau_*) > 0$, и следует ожидать более быстрой сходимости счета на установление.

3.3.2. Оценки границ спектра.

Оценка сверху. Для построения границ логарифмического набора нужны оценки границ спектрального интервала. В одномерном случае имеет место

Лемма 3. *Для произвольных k, h*

$$\lambda_{\max} \leqslant \xi = 4 \max \frac{1}{h_{n+1/2} + h_{n-1/2}} \left(\frac{k_{n+1/2}}{h_{n+1/2}} + \frac{k_{n-1/2}}{h_{n-1/2}}\right). \bullet \tag{3.24}$$

Доказательство. При $k, h = \text{const}$ эта оценка очевидна. В общем случае в силу неравенства треугольника из (1.4) следует

$$\|\lambda u\| = |\lambda| \cdot \|u\| \leqslant$$
$$\leqslant 2\|u\| \max \frac{2}{h_{n+1/2} + h_{n-1/2}} \left(\frac{k_{n+1/2}}{h_{n+1/2}} - \frac{k_{n-1/2}}{h_{n-1/2}} \right). \tag{3.25}$$

После сокращения на $\|u\|$ получим утверждение леммы. ■

Оценка (3.24) без труда обобщается на многомерный случай, причем k_α может зависеть не только от x_α, но и от других переменных. Справедлива

Лемма 4. *Для произвольных k_α, h_α*

$$\lambda_{\max} \leqslant 4 \sum_\alpha \max_{x,y,z} \frac{2}{h_{\alpha,n+1/2} + h_{\alpha,n-1/2}} \left(\frac{k_{\alpha,n+1/2}}{h_{\alpha,n+1/2}} + \frac{k_{\alpha,n-1/2}}{h_{\alpha,n-1/2}} \right). \bullet \tag{3.26}$$

Доказательство дословно повторяет доказательство леммы 3. ■

Асимптотическое свойство. На квазиравномерных сетках $\xi/\lambda_{\max} \to 1$ при $N \to \infty$. Для иллюстрации возьмем пример с сильно пульсирующим коэффициентом теплопроводности и сильно неравномерной сеткой (см. рис. 3.2):

$$k(x) = 1 - 0.9 \sin^2 2\pi x, \quad x \in [0, 1]; \tag{3.27}$$
$$h(x) = (25 + 20 \cos 20x) \,/\, (N\psi_1), \quad \psi_1 = 25 + \sin 20 \approx 25.31 \tag{3.28}$$

В этом случае поведение оценки (3.24) приведено на рис. 3.3. Видно, что наиболь-

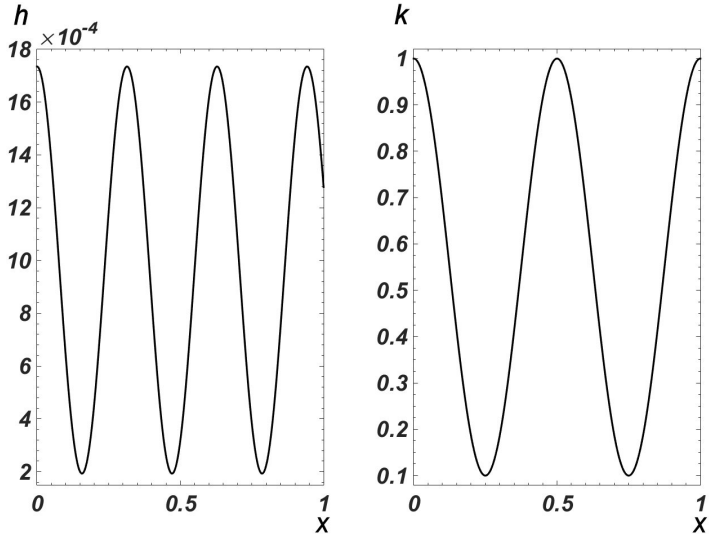

Рис. 3.2. Сильно неоднородная среда (3.27) – (3.28).

шее отличие ξ от $\lambda_{\max}$ составляет 14%, что является приемлемым. Далее отношение $\xi/\lambda_{\max}$ быстро стремится к единице.

Оценка снизу. Величину $\lambda_{\min}$ можно оценить снизу. Справедливо утверждение:

Лемма 5. *Для произвольного непрерывного k*

$$\lambda_{\min} \geqslant \frac{\pi^2}{l^2} \min k, \tag{3.29}$$

где l – длина отрезка. •

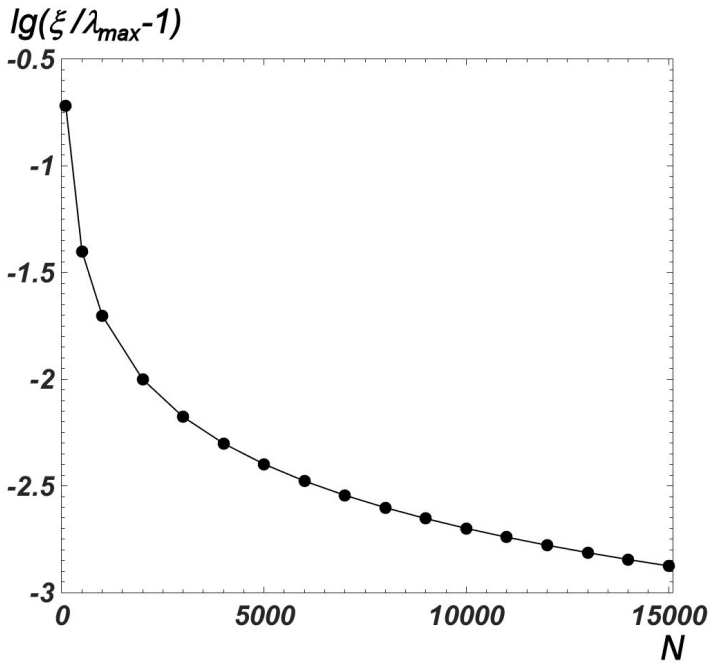

Рис. 3.3. Сильно неоднородная среда (3.27) – (3.28). Оценка (3.24) для $\lambda_{\max}$.

Доказательство проведем, исходя из дифференциального представления:

$$\lambda u = -\frac{d}{dx}\left(k\frac{du}{dx}\right). \tag{3.30}$$

Домножим (3.30) скалярно на u и проинтегрируем по частям:

$$\lambda(u,u) = \left(k\frac{du}{dx}, \frac{du}{dx}\right). \tag{3.31}$$

Применяя теорему о среднем, вынесем $k(x^*)$, $x^* \in [0,l]$ за знак интеграла, разделим обе части (3.31) на (u,u) и перейдем к минимуму по u:

$$\lambda \geqslant k(x^*)\min_u \frac{\int (du/dx)^2\,dx}{\int u^2 dx} = k(x^*)\tilde{\lambda}, \tag{3.32}$$

где $\tilde{\lambda} = \pi^2/l^2$ – наименьшее собственное значение простейшей задачи. Поскольку k предполагается непрерывным, то $k(x^*) \geqslant \min k$, что завершает доказательство леммы. ■

На практике оценка (3.29) может оказаться не очень точной. Однако ее можно взять в качестве нулевого приближения в методе обратных итераций с переменным сдвигом. Тогда уже первая итерация дает хорошую точность (поскольку нижние собственные значения хорошо разнесены). Так, в примере (3.27) – (3.28) отличие первой итерации от $\lambda_{\min}$ не превышало 5%.

Сходимость метода обратных итераций с переменным сдвигом проиллюстрирована на примере простейшей задачи ($k = \text{const}$, $h = \text{const}$) при $N = 1000$. На рис. 3.4 показана зависимость величины $\lg(1 - \lambda_j/\lambda_{\min})$ от номера итерации j. Видно, что уже третья итерация дает точность, сравнимую с фоном ошибок округления.

Лемма 5 допускает обобщение на многомерный случай. Именно, справедливо следующее утверждение:

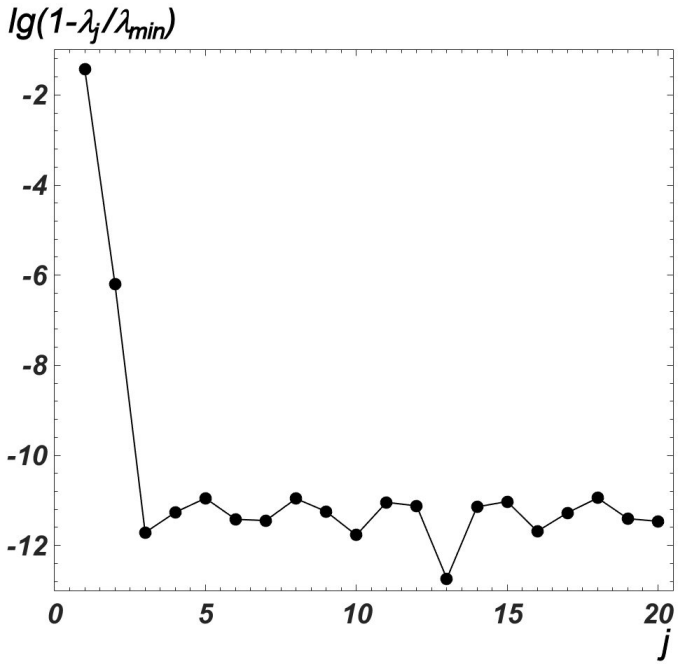

Рис. 3.4. Сходимость метода обратных итераций с переменным сдвигом.

Лемма 6. *Для произвольных непрерывных* k_α

$$\lambda_{\min} \geqslant \sum_\alpha \frac{\pi^2}{l_\alpha^2} \min_{x,y,z} k_\alpha, \tag{3.33}$$

где l_α *– длина отрезка по направлению* α. •

Доказательство проводится по той же схеме, что и для леммы 5. ■

Практические рекомендации. На практике целесообразна следующая процедура. Вычислим первые шаги метода обратных итераций с переменным сдвигом для оператора Λ_x при каждых фиксированных сеточных y, z. В качестве начального приближения выберем $\lambda_x^{(0)}(y_n, z_m) = \pi^2/l_x^2 \min_x k_x(x, y_n, z_m)$. После этого возьмем наименьший из полученных результатов $\tilde{\xi}_x$. Проведем аналогичную процедуру с операторами Λ_y при всех фиксированных сеточных x, z (наименьший результат $\tilde{\xi}_y$) и Λ_z при всех фиксированных сеточных x, y (наименьший результат $\tilde{\xi}_z$). Оценкой для наименьшего собственного значения будет величина $\tilde{\xi}_x + \tilde{\xi}_y + \tilde{\xi}_z$.

3.3.3. Порождающая функция.

Известные наборы. В работе [18] рассмотрено несколько вариантов логарифмической сетки. Общий вид такого набора можно записать как

$$\ln \tau_s = 1/2 \ln (\tau_{\max}\tau_{\min}) + 1/2 \ln (\tau_{\max}/\tau_{\min}) f(s), \quad 0 \leqslant s \leqslant S. \tag{3.34}$$

Здесь порождающая функция $f(s) \in [-1, 1]$ и является монотонной и нечетной. В [18] были рассмотрены следующие наборы: равномерный

$$f_{\text{р}}(s) = 2s/S - 1, \tag{3.35}$$

чебышевский

$$f_{\text{ч}}(s) = -\cos(\pi s/S), \tag{3.36}$$

и интерполяционный

$$f_{\text{и}}(s) = \theta_s\left(1 + (1-\theta_s)/2r\right)^r,\ \theta_s = 2s/S - 1,\ r = \left(1 + 1/8\ln^2(\lambda_{\max}/\lambda_{\min})\right)^{-1}. \tag{3.37}$$

Равномерный набор плохо подавлял граничные гармоники, а чебышевский – центральные. Интерполяционный примерно одинаково подавлял все гармоники, что обеспечивало лучшую точность. Однако он имеет громоздкий и непрозрачный вид.

Линейно-тригонометрический набор. Чтобы скомпенсировать недостатки равномерного и чебышевского наборов, возьмем их линейную комбинацию с некоторым весом C:

$$f^C_{\text{ЛТ}}(s) = C f_{\text{р}}(s) + (1-C) f_{\text{ч}}(s). \tag{3.38}$$

Выражение (3.38) представляет собой однопараметрическое семейство наборов, где параметр C определяет соотношение шагов в центре и на краях интервала. Для практического применения удобнее универсальный и простой набор, поэтому нужно выбрать некоторое фиксированное значение C.

Приведем теоретические соображения для выбора константы C. Гармоники, находящиеся внутри спектрального интервала, гасятся шагами, попадающими с двух сторон от нее, а шаги на границе – шагами только с одной стороны. Чтобы граничные гармоники гасились так же, как центральные, шаги на границах должны быть в 2 раза короче, чем шаги в центре. Тогда число гармоник, эффективно участвующих в гашении будет одно и то же. Применяя это условие к набору (3.38), нетрудно получить следующее значение веса:

$$C = \pi/(\pi + 2). \tag{3.39}$$

Набор (3.38)–(3.39) будем называть ***линейно-тригонометрическим (ЛТ)***.

Было проведено большое количество численных расчетов для набора (3.38) с различными значениями константы C. На график выводились огибающие функции

$$R(\xi) = \lg\left|\prod_{s=1}^{S}\rho(\xi)\right|, \quad \xi = \frac{1}{2}\ln\lambda \tag{3.40}$$

для разных N и разных S. Примеры таких графиков для значения (3.39) приведены на рис. 3.5. Значения S подбирались так, чтобы во всех трех случаях $N = 100, 1000, 10000$ получать примерно одинаковые точности.

Видно, что линейно-тригонометрический набор (3.38) – (3.39) обеспечивает примерно одинаковое подавление всех гармоник и дает лучшую сходимость. Соответствующие точности почти не уступают интерполяционному набору (3.37) и значительно превосходят равномерный (3.35) и чебышевский (3.36) (см. также табл. 3.1). Это убедительно подтверждает изложенный выше эвристический выбор константы C. При этом формулы являются простыми и прозрачными.

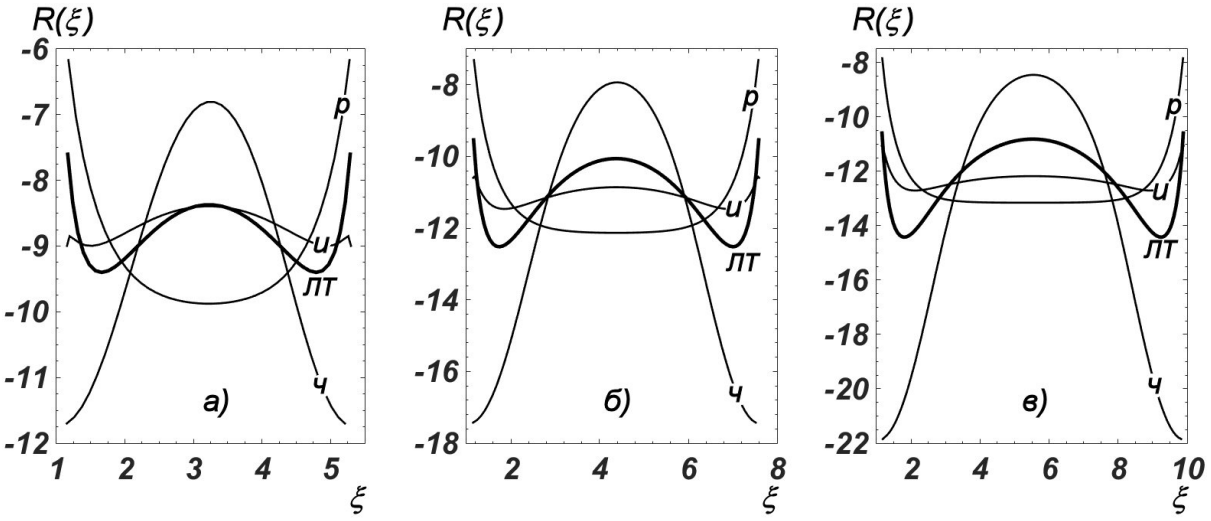

Рис. 3.5. Огибающие функции (3.40): а) $N = 100$, $S = 40$, б) $N = 1000$, $S = 75$, в) $N = 10000$, $S = 110$; **р** – равномерный набор, **ч** – чебышевский набор, **и** – интерполяционный набор, **лт** – линейно-тригонометрический набор.

Заметим также, что качественное поведение интерполяционного набора (3.37) зависит от N: на рис. 3.5 *в*) края расположены выше, чем центр, а на рис. 3.5 *а*) – наоборот. Это значит, что набор может быть непредсказуемым, и его применение нежелательно. Качественное поведение линейно-тригонометрического набора (3.38)–(3.39) практически не меняется: края расположены примерно на том же уровне, что и центр.

3.3.4. Априорные оценки точности. Построим теоретические оценки сходимости, задавая набор $\{\tau_s\}$ в логарифмической шкале. Получим мажорантные оценки, которые являются почти строгими.

Средние гармоники. Пусть сетка равномерна по $\ln\tau$. Ее шаг $\delta = \ln\tau_s - \ln\tau_{s-1} = \text{const}$. Пусть число шагов достаточно велико. Пусть гармоника λ_k удалена от обоих краев спектрального интервала. Это означает, что с обеих сторон от нее лежит много шагов.

Если для одного шага случайно $\tau_s = 2/\lambda_k$, то $\rho = 0$ и гашение наилучшее. Гашение будет наихудшим, если величина $\ln(2/\lambda_k)$ равноудалена от двух соседних шагов $\ln\tau_s$. В этом случае

$$\ln\frac{\tau_s\lambda_k}{2} = \pm\frac{\delta}{2}, \pm\frac{3\delta}{2}, \pm\frac{5\delta}{2}. \tag{3.41}$$

Получим приближенную величину шага δ в этом наихудшем случае. Разобьем набор $\{\tau_s\}$ на пары шагов τ_s, $\tau_{s'}$, расположенных левее и правее $2/\lambda_k$. Потребуем, чтобы гармоника λ_k после s-го шага гасилась в ε раз, т.е.

$$\prod_s |\rho(\tau_s\lambda_k)| = \varepsilon. \tag{3.42}$$

Множителей много, и они убывают достаточно быстро от центрального. Поэтому

Таблица 3.1. Сходимость итераций. Числа в клетках: первое – S, остальные –логарифмы максимальной погрешности гармоник для наборов: равномерного (3.35), чебышевского (3.36), интерполяционного (3.37) и линейно-тригонометрического (3.38) – (3.39).

N	удовлетворительная точность	хорошая точность	отличная точность
100	30	40	50
	-4.78	-6.16	-7.53
	-5.08	-6.81	-8.54
	-6.22	-8.40	-10.57
	-5.87	-7.60	-9.31
1000	55	75	95
	-5.54	-7.31	-9.05
	-5.77	-7.93	-10.10
	-7.88	-10.55	-13.19
	-7.20	-9.53	-11.84
10000	80	110	140
	-5.90	-7.84	-9.76
	-6.08	-8.45	-10.82
	-8.19	-11.06	-13.92
	-7.78	-10.59	-13.23

можно взять бесконечные пределы произведения. Тогда

$$\ln \varepsilon = \sum \ln |\rho(\tau\lambda)| = 2 \sum_{\tau\lambda/2>1} \ln |\rho(\tau\lambda)| . \tag{3.43}$$

Подставляя разложение

$$\ln |\rho| = \ln(1-\zeta) - \ln(1+\zeta) =$$
$$= -2\left(\zeta + 1/3\zeta^3 + 1/5\zeta^5 \ldots\right), \quad \zeta = 2/\tau_s\lambda_k < 1$$

в (3.43) и группируя члены, получим

$$\ln 1/\varepsilon = 4\left(e^{-\delta} + e^{-3\delta} + e^{-5\delta} + \ldots\right) + 4/3\left(e^{-3\delta} + e^{-9\delta} + e^{-15\delta} + \ldots\right) + \ldots =$$
$$= 2\left(\sinh(\delta/2)\right)^{-1} + 2/3\left(\sinh(3\delta/2)\right)^{-1} + \ldots \approx 4/\delta \sum_{n=0}^{\infty} (2n+1)^{-2} = \pi^2/2\delta. \tag{3.44}$$

Отсюда нетрудно выразить δ через ε.

Крайние гармоники. Гармоники, расположенные по краям, гасятся шагами только с одной стороны, т.е. в 2 раза слабее, чем средние гармоники. Этим и объясняется указанный ранее недостаток равномерного набора. Значит, для получения гарантированной точности нужно подставлять в (3.44) не δ, а $\delta/2$. Поскольку для равномерной сетки $\delta = 1/S \ln \lambda_{\max}/\lambda_{\min}$, то

$$S = \frac{4}{\pi^2} \ln \frac{\lambda_{\max}}{\lambda_{\min}} \ln \frac{1}{\varepsilon} \approx 0.4 \ln \frac{\lambda_{\max}}{\lambda_{\min}} \ln \frac{1}{\varepsilon}. \tag{3.45}$$

Поскольку наши оценки были мажорантными, число итераций (3.45) заведомо достаточно для получения точности ε.

Неравномерная сетка. Пусть задан некоторый набор шагов, неравномерный в логарифмической сетке, и пусть число шагов S велико. Тогда каждый шаг эффективно гасит близкие гармоники и слабо – удаленные. Поэтому для фиксированной гармоники можно использовать приближение локальной равномерности набора шагов и оценку (3.45), где δ есть локальная величина шага. Для совокупности всех гармоник нужно подставить в (3.45) величину $\max\delta$.

Подчеркнем, это приближение хорошо работает лишь при больших S, т.е. для расчетов с высокой точностью. Для начальных S возникают нерегулярности.

Оценка для ЛТ-набора. Из (3.44) нетрудно получить оценку гашения центральных гармоник ЛТ-набором, умножив это выражение на отношение шагов равномерного и ЛТ-наборов в центре интервала:

$$S = \frac{4}{(\pi^2+2\pi)} \ln\frac{\lambda_{\max}}{\lambda_{\min}} \ln\frac{1}{\varepsilon} \approx 0.25 \ln\frac{\lambda_{\max}}{\lambda_{\min}} \ln\frac{1}{\varepsilon}. \tag{3.46}$$

Для граничных гармоник в силу вдвое большей плотности шагов получается такая же оценка. Для промежутков между краями и центром доказать эту оценку не удается, но она хорошо подтверждается практическими расчетами.

Многомерный случай. Полученные результаты непосредственно переносятся на двумерный случай, поскольку двумерный множитель роста есть произведение одномерных. В трехмерном случае применимость этих результатов неочевидна, но проведенные вычисления показывают, что ЛТ-набор работает хорошо.

3.4 Примеры расчетов

3.4.1. Графики точности. Было проведено сравнение разных наборов для одномерного случая. Выбиралась задача с известным точным сеточным решением (при $k = \text{const}$, $h = \text{const}$, $N = 1000$). Видно, что сетка весьма густая. Точное решение и начальное приближение полагались гладкими: $u = x^2$, $u^{(0)} = 0$. Заметим, что даже выбор заведомо плохого нулевого приближения – набора псевдослучайных чисел – практически не ухудшает сходимости.

Сравнение наборов. На рис. 3.6 представлены зависимости погрешности ε от S для различных наборов шагов. Это монотонно убывающие линии. При достаточно больших S они переходят в постоянный фон, обусловленный ошибками округления (расчеты производились с 64-разрядными числами). Видно, что чебышевский и равномерный наборы проигрывают по точности. Интерполяционный и ЛТ- наборы имеют почти одинаковую точность. Однако кривая интерполяционного набора похожа на прямую с небольшими случайными осцилляциями, а кривая ЛТ-набора практически прямолинейна. Далее мы увидим, что прямолинейность позволяет получить апостериорные оценки точности. Поэтому все дальнейшие расчеты проводились с ЛТ-набором.

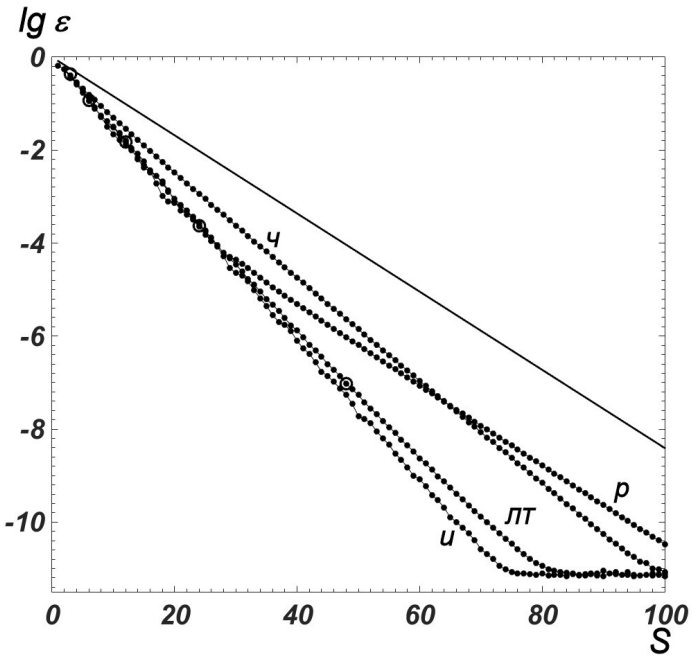

Рис. 3.6. Погрешность в одномерном случае; $N = 1000$, $k = 1$, $h = 1/N$; прямая – оценка (3.45); • – численные расчеты, обозначения наборов – см. рис. 3.5; ∘ – оценка (3.53).

Участки с $S < 25$ следует назвать нерегулярными. На них еще не все кривые выходят на асимптотические режимы. При этом для ЛТ-набора наклон в регулярной области почти не отличается от приблизительного наклона в нерегулярной, что дополнительно свидетельствует о преимуществах этого набора.

Сходимость ЛТ-набора. На рис. 3.7 приведена зависимость погрешностей ε от S для ЛТ-набора при различных N. Видно, что линии очень быстро выходят на регу-

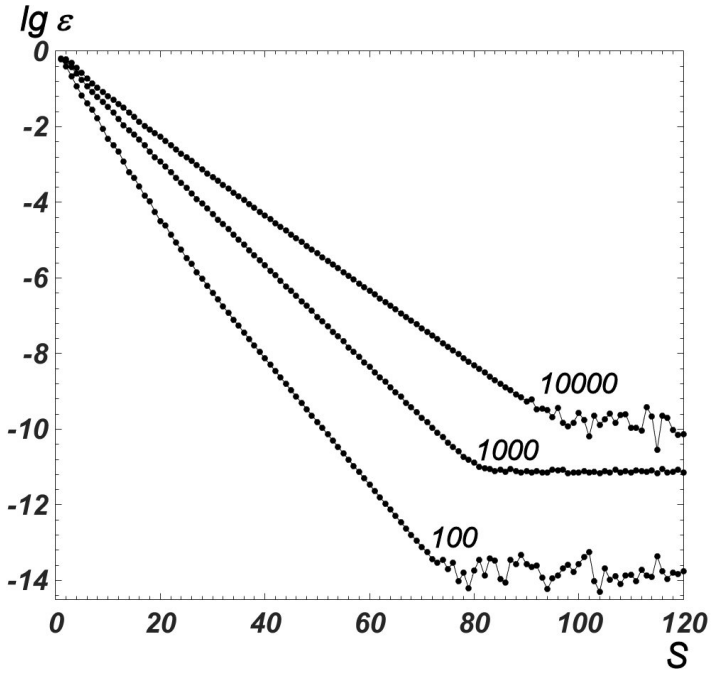

Рис. 3.7. Погрешность в одномерном случае; k, h – см. рис. 3.6; • – численные расчеты по ЛТ-набору; цифры около линий – значения N.

лярный прямолинейный участок, который затем резко (почти изломом) переходит в горизонтальный фон. Поскольку ошибки округления носят стохастический характер, то фон ошибок округления содержит "дрожания". Этот фон тем выше, чем больше N, что объясняется ухудшением обусловленности линейной системы. В [52], [53] показано, что число обусловленности для этих задач зависит от N приблизительно по закону $\varkappa \approx ((N+1)/2)^{3/2}$. Наклоны регулярных участков убывают с увеличением N примерно в соответствии с (3.46), хотя эта оценка является нестрогой.

3.4.2. Теоретические оценки. Было проведено сравнение погрешностей равномерного и ЛТ-наборов с соответствующими теоретическими оценками. Согласие оказалось хорошими. В приведенном примере коэффициенты наклона регулярных участков равномерного и ЛТ-наборов отличаются от оценок (3.45) и (3.46) не более, чем на 1%.

Отметим, что эти оценки определяют скорость сходимости в регулярной области и не учитывают нерегулярную область. Теоретическая оценка является прямой, исходящей из начала координат. Расчетная кривая может содержать нерегулярный участок. Для равномерного набора этот участок с крутым наклоном довольно велик. Поэтому для него оценка (3.45) лежит на два порядка выше регулярного участка. Для ЛТ-набора нерегулярный участок мал, и для него оценка (3.46) лежит выше регулярного участка всего на половину порядка.

3.4.3. Трудные примеры. Приведем несколько представительных примеров: две задачи с сильно переменными средами и задачу в неограниченной области.

"Пульсирующая" среда. Рассмотрим задачу с сильно пульсирующим коэффициентом теплопроводности (3.27) и сильно неравномерной сеткой (3.28). Даже на достаточно подробной сетке ($N = 1000$) у кривых сходимостей равномерного и интерполяционного наборов вообще не наблюдается регулярных участков (см. рис. 3.8). В то же время линейно-тригонометрический набор дает выраженный прямолинейный участок, что соответствует регулярной сходимости.

"Слоистая" среда. Рассмотрим следующую задачу с сильно неоднородными $k(x)$ и $h(x)$:

$$k(x) = 0.1 + \pi/2 + \operatorname{arctg} 50(x - 1/2), \quad x \in [0, 1]; \tag{3.47}$$

$$h(x) = 3e^{3x}/\left(N\psi_2\right), \quad \psi_2 = e^3 - 1 \approx 19.09. \tag{3.48}$$

Она сложна тем, что $k(x)$ меняется очень круто (практически скачком) (см. рис. 3.9), т.е. имитирует слоистую среду.

Здесь равномерный, интерполяционный и ЛТ-наборы обеспечивают примерно одинаковую скорость сходимости, но, также как в предыдущем примере, кривая равномерного набора не имеет регулярного участка (см. рис. 3.10).

Неограниченная область. Рассмотрим задачу в однородной неограниченной области

$$k(x) = 1, \quad h(x) = (1 - x^2)^{-1.5}, \quad x \in [0, 1]; \tag{3.49}$$

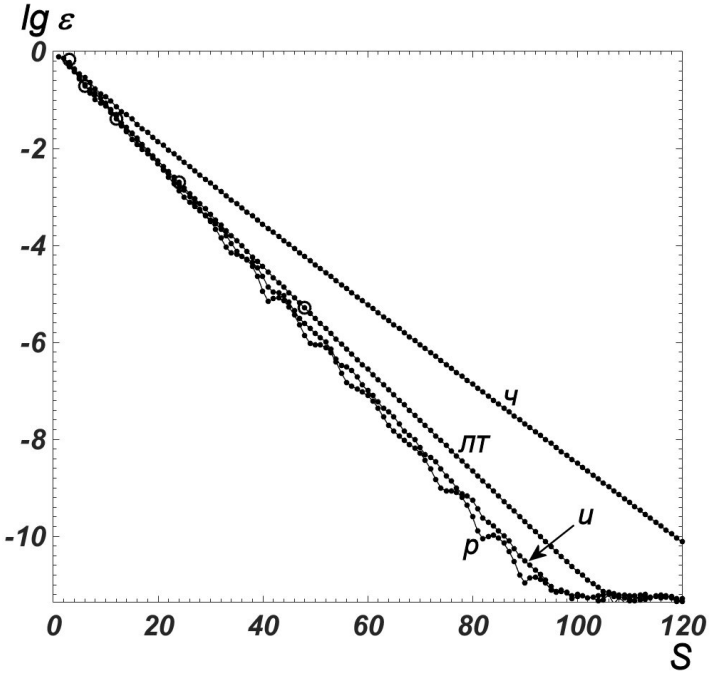

Рис. 3.8. Сильно неоднородная среда (3.27)–(3.28); $N = 1000$; обозначения соответствуют рис. 3.6.

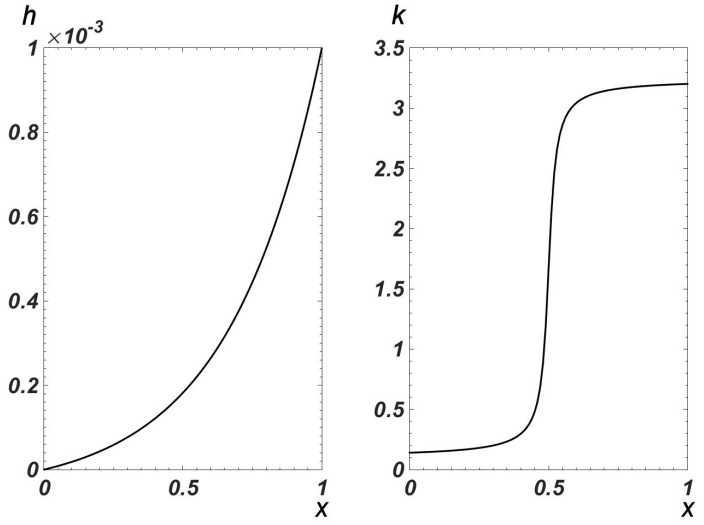

Рис. 3.9. Среда, изменяющаяся скачком (3.47)–(3.48).

Счет на установление в неограниченных областях всегда считался очень трудным, поскольку при использовании квазиравномерных сеток $\lambda_{\max}/\lambda_{\min}$ хуже, чем $O(N^2)$. Нередко это отношение составляет $O(N^4)$ [45]. В этом случае метод с постоянным оптимальным шагом или явным набором параметров давали бы $S = O(N^2)$, т.е. огромное число итераций. Однако для ЛТ-набора число итераций увеличивается всего в 1.5 раза по сравнению с простейшей задачей. Кривые сходимости разных наборов даны на рис. 3.11. Отметим, что во всех этих примерах кривая ЛТ-набора имеет хороший регулярный участок. Это дополнительный довод в пользу ЛТ-набора.

Сходимость в неограниченной области. Рассмотрим подробнее вопрос о сходимости различных методов в последнем примере. Здесь использовались гра-

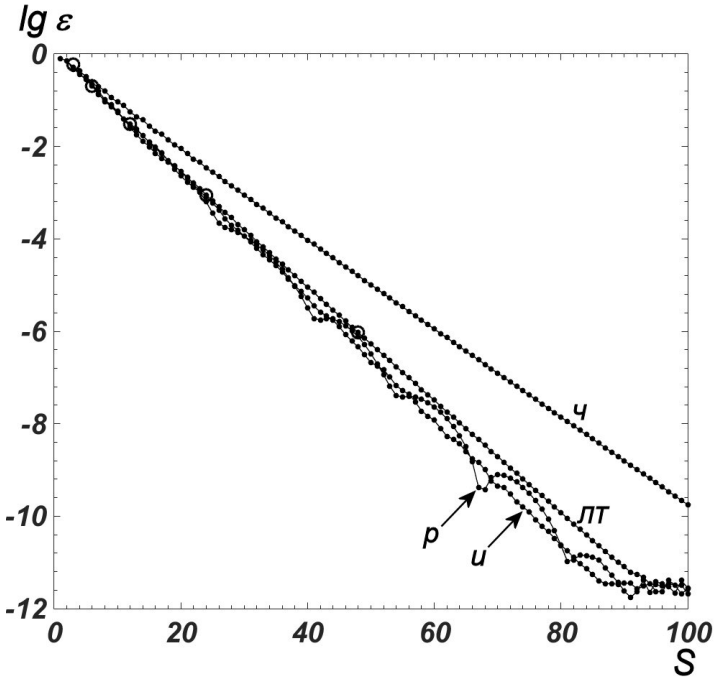

Рис. 3.10. Среда, изменяющаяся скачком (3.47)–(3.48); $N = 1000$; обозначения соответствуют рис. 3.6.

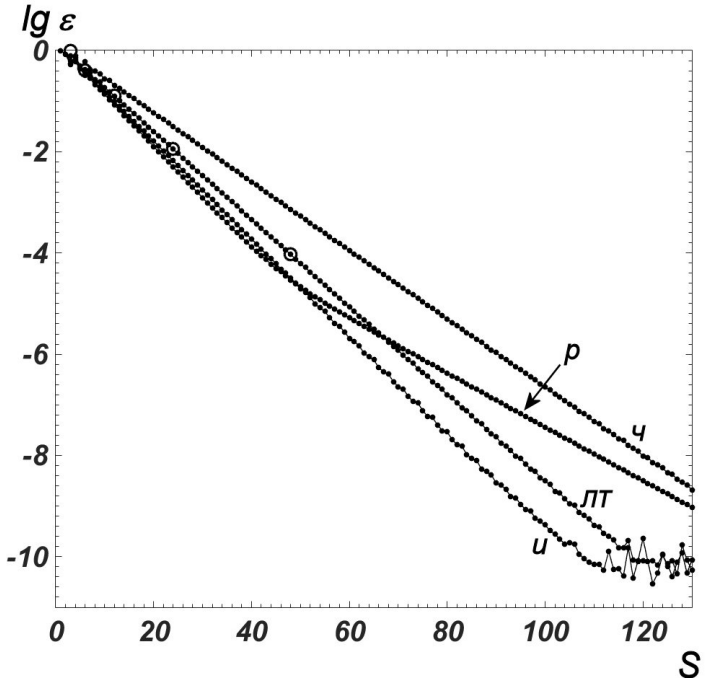

Рис. 3.11. Задача в неограниченной области (3.49); $N = 1000$; обозначения соответствуют рис. 3.6.

ницы спектра, вычисленные методом обратных итераций с переменным сдвигом: $\lambda_{\min} = 3.2380 \cdot 10^{-3}$ и $\lambda_{\max} = = 3.9976 \cdot 10^{6}$. Итерации обрывались при достижении относительной точности 10^{-6}, поэтому эти значения можно считать точными. Отношение границ спектра $\lambda_{\max}/\lambda_{\min} \sim 10^{9}$, что означает плохую обусловленность матрицы линейной системы. Это отношение становится очень большим из-за того, что $\lambda_{\min}$ мало. Это можно понять по характеру точного решения: в бесконечной обла-

сти оно раскладывается не в ряд, а в интеграл Фурье, причем значения λ начинаются от нуля.

Нетрудно убедиться, что при расчете по явной схеме с чебышевским набором параметров для достижения точности $\varepsilon = 10^{-10}$ нужно сделать $S \approx 3.3 \cdot 10^5$ итераций. Такая же трудоемкость будет при расчете по неявной схеме с постоянным оптимальным шагом. Наконец, метод сопряженных градиентов даст такую точность за $S \approx 7.3 \cdot 10^3$ шагов. По рис. 3.11 видно, что при расчетах с логарифмическим набором и ЛТ-сеткой требуется значительно меньше – 115 итераций. Это говорит о больших преимуществах логарифмического набора при решении плохо обусловленных задач.

3.4.4. Двумерные расчеты. На рис. 3.12 представлены двумерные расчеты для разных N. Рассматривались две ситуации. а) По разным направлениям брались одни и те же сетки $N_x = N_y$, $h_x = h_y$, но полагалось $k_y = 10k_x$. Поэтому для гармоник с одинаковыми номерами $\lambda_y = 10\lambda_x$. Таким образом, границы спектров по обоим направлениям сильно отличались. б) Сетки и коэффициенты по обоим направлениям были одинаковы. Поэтому и спектры $\lambda_x = \lambda_y$ совпадали.

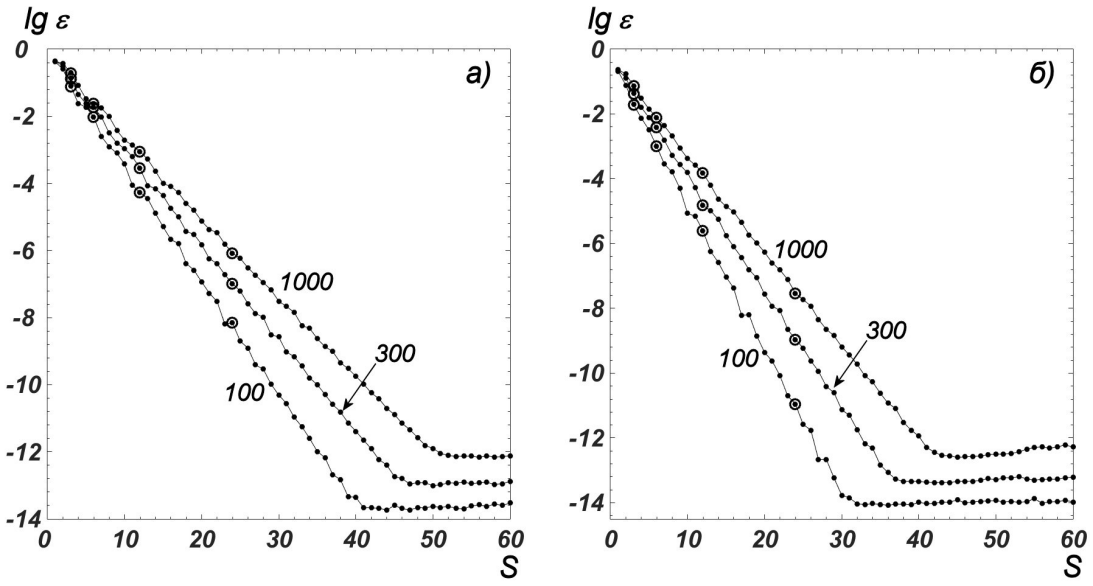

Рис. 3.12. Двумерный случай; цифры около линий – значения N, • – вычисления по ЛТ-набору, ∘ – оценка (3.53); а) сдвинутые спектры, $k_y = 10k_x$, $h_x = h_y = 1/N$; б) совпадающие спектры; $k_x = k_y$, $h_x = h_y = 1/N$.

Для двумерного случая справедливы те же выводы, что и для одномерного. Кривые сходимости ЛТ-набора имеют выраженные прямолинейные участки, наклоны которых убывают с увеличением N. Однако в двумерном случае скорость сходимости выше, чем в одномерном.

Это объясняется тем, что каждую двумерную гармонику гасят сразу два множителя (3.11). Этот эффект особенно велик, если спектры по обоим направлениям одинаковы. Тогда сходимость ускоряется вдвое по сравнению с одномерным случаем (ср. рис. 3.7 и рис. 3.12, *б*). В этом случае число итераций, необходимое для достижения фона ошибок округления, составляет всего $S = 30$. Это является очень малым

для такой высокой точности сходимости и убедительно свидетельствует о преимуществах предложенного метода.

3.4.5. Трехмерные расчеты. На рис. 3.13, *а* показаны трехмерные расчеты в случае сдвинутых спектров. Сетки по разным направлениям также одинаковы, но

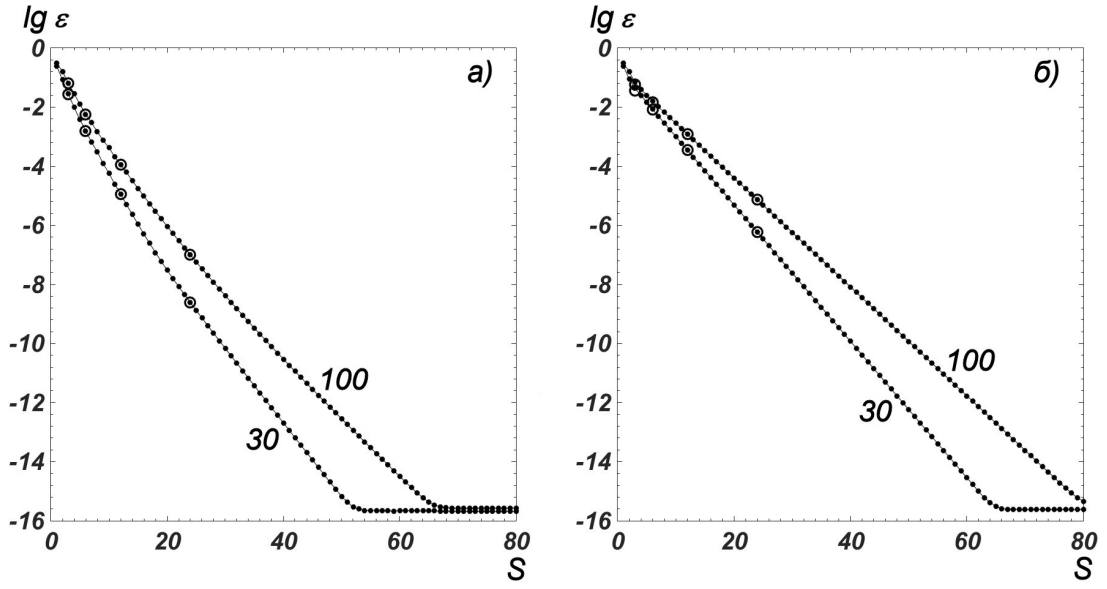

Рис. 3.13. Трехмерный случай, обозначения соответствуют рис. 3.12; а) сдвинутые спектры, $k_y = 3k_x$, $k_z = 10k_x$, $h_x = h_y = h_z = 1/N$; б) совпадающие спектры; $k_x = k_y = k_z$, $h_x = h_y = h_z = 1/N$.

$k_y = 3k_x$, $k_z = 10k_x$. Поэтому спектры и их границы по разным направлениям сильно различаются. Результаты расчетов аналогичны двумерному случаю: видно, что кривые погрешности четко выходят на прямые. Их наклон несколько больше, чем предсказывает теоретическая оценка (3.46).

Проводились также вычисления для совпадающих спектров, когда $k_x = k_y = k_z$. Результаты показаны на рис. 3.13, *б*. В этом случае сходимость оказывается медленнее, чем в расчете со сдвинутыми спектрами (но по-прежнему чуть быстрее, чем оценка (3.46)). Это можно объяснить тем, что для крайних гармоник множитель роста имеет положительный минимум $\rho = 1/9$, и эти гармоники гасятся не полностью. В примере со сдвинутыми спектрами минимум множителя роста оказывается отрицательным, и набор шагов строится по его нулям $\tau_\pm$. Это обеспечивает лучшее гашение.

3.4.6. Влияние неточной оценки границ спектра.

Линейно-тригонометрический набор. Проводились вычисления с ЛТ-набором, построенным для спектра с измененными границами. Вместо точных границ брались $\lambda_{\min}/t$, $\lambda_{\max}t$. При $t < 1$ это эквивалентно сужению расчетных границ набора, при $t > 1$ – расширению.

Графики точности в зависимости от $\lg t$ при фиксированных N, S представлены на рис. 3.14. Видно, что сужение спектра дает сильное ухудшение точности. Это объ-

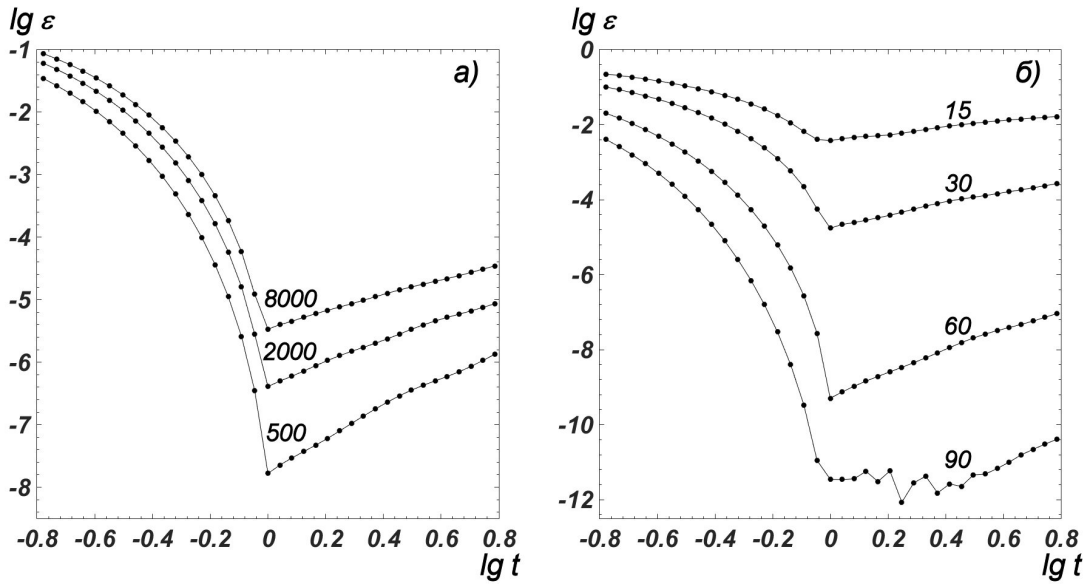

Рис. 3.14. Влияние границ расчетного спектра, $k = 1$, $h = 1/N$; а) $S = 50$, цифры около линий – значения N; б) $N = 500$, цифры около линий – значения S.

ясняется тем, что расчетные границы набора оказываются внутри истинных границ. Поэтому крайние гармоники гасятся плохо. Расширение спектра также ухудшает точность, но не столь заметно. Это происходит из-за увеличения длины каждого шага.

Отметим, что левая ($\lg t < 0$) и правая ($\lg t > 0$) ветви графика не образуют плавного перехода, а пересекаются под углом. Это означает, что ошибка в определении границ спектра приводит к заметному ухудшению точности. Поэтому оценки границ спектра должны быть насколько возможно аккуратными, причем соответствующими расширению.

Приближенные выражения. Видно, что правая ветвь графиков на рис. 3.14 хорошо аппроксимируется прямой, для которой можно вычислить коэффициент наклона. Иными словами, точность при расширении можно представить в виде

$$\lg \varepsilon(t) = \lg \varepsilon_0 + \lg \varepsilon_1 \lg t. \tag{3.50}$$

Проведенные вычисления показывают, что $\lg \varepsilon_1$ зависит от N и от S, причем с неплохой точностью по N зависимость логарифмическая, а по S – линейная:

$$\lg \varepsilon_1 = (\eta_0 + \eta_1 \lg N)S, \tag{3.51}$$

где $\eta_0 \approx 0.12$, $\eta_1 \approx 0.02$. Эта эвристическая закономерность хорошо работает при умеренных расширениях (до $\lg t \approx 0.4$, т.е. $t \approx 2.5$).

По рис. 3.14 видно, что большим значениям S соответствует более резкое ухудшение точности при расширении. Так, для $S = 60$ излом более резкий, чем для $S = 30$, а для $S = 15$ график и вовсе имеет гладкий минимум. Значение $S = 90$ соответствует фону для задачи с истинным спектром. Однако, начиная с некоторого t, кривая точности возвращается на регулярный, дофоновый участок.

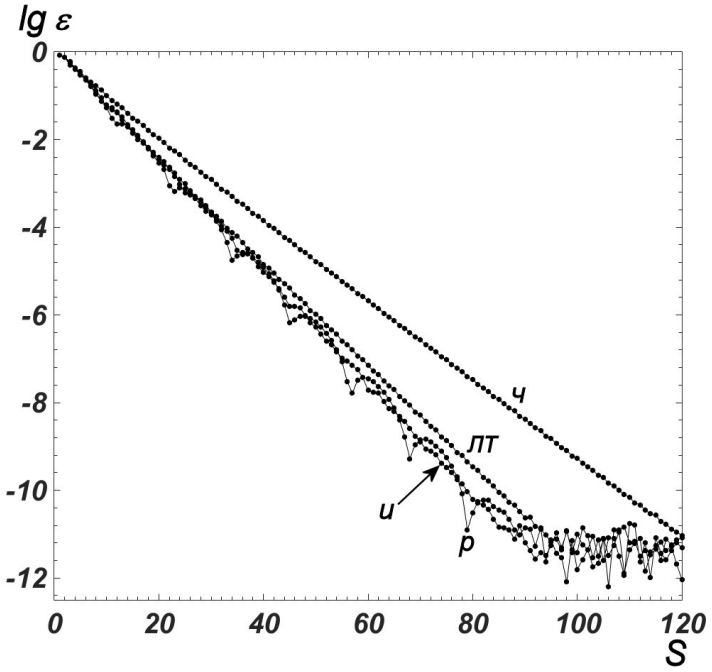

Рис. 3.15. Расчеты с расширенным спектром; $t = 4$; $S = 50$, $k = 1$, $h = 1/N$; обозначения наборов соответствуют рис. 3.5.

Другие наборы. Расширение спектра может менять характер сходимости некоторых наборов (см. рис. 3.15). Так, чебышевский набор значительно теряет в скорости, а на кривой равномерного набора даже появляются участки немонотонности.

Это замечание особенно актуально, поскольку на практике границы спектра неизвестны, и их нужно оценивать с некоторым запасом. Заметим, что ЛТ-набор не имеет этих недостатков: для него кривая сходимости практически прямолинейна даже для заметного расширения ($t = 4$), а потеря скорости не так велика. Это дополнительный довод в пользу ЛТ-набора.

Многомерный случай. Сделанные выводы непосредственно переносятся на двумерный случай. В трехмерном случае вычисления также показывают нежелательность сужения или расширения границ спектра, поскольку это дает ухудшение точности (см. рис. 3.16).

В частных случаях (например, при совпадающих спектрах) расширение может давать преимущества (см. рис. 3.16, *б*). Но, во-первых, достигаемое таким образом улучшение точности не очень велико, и во-вторых, такие простые задачи не представляют практического интереса.

3.4.7. Расчеты с высокой точностью. В литературе при иллюстрации сходимости итерационных методов обычно ограничиваются умеренной точностью $\varepsilon \sim 10^{-4} \div 10^{-6}$ и умеренными $N \sim 100$ (редко $N \sim 300$). Причина этого в медленной сходимости общеизвестных методов. Даже при таких скромных N и ε они требуют сотен итераций. Из рис. 3.6 – 3.13 видно, что эволюционно факторизованный счет на установление с ЛТ-набором обеспечивает такую точность за 15-35 итераций. Это на 2-3 порядка быстрее, чем для методов сопряженных направлений. Предложенный здесь

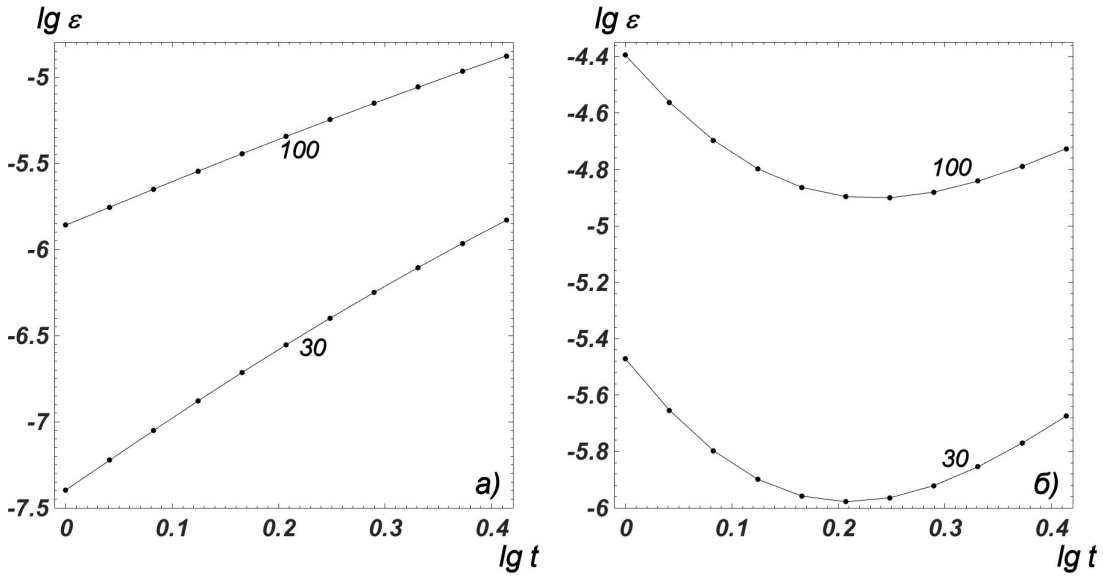

Рис. 3.16. Влияние границ расчетного спектра в трехмерном случае, $S = 20$, цифры около линий – значения N; а) несовпадающие спектры, k_α, h_α соответствуют рис. 3.13, *а*; б) совпадающие спектры, k_α, h_α соответствуют рис. 3.13, *б*.

метод позволяет достичь предельно возможной точности – фоновой ($10^{-11} \div 10^{-13}$), причем за небольшое число итераций ($S \approx 40 \div 80$).

Решение сеточных уравнений с такой высокой точностью позволяет решать эллиптические уравнения на многократно сгущающихся сетках с применением уточнения по Ричардсону. Это дает возможность решать дифференциальные эллиптические уравнения с недостижимой ранее точностью.

3.5 Апостериорные оценки точности

3.5.1. Сгущение сеток.

Невязка. При итерационном решении сеточных уравнений непосредственно вычисляется невязка $R^{(S)}$. Погрешность решения в принципе можно оценить по невязке, поскольку

$$\left\|u^{(S)} - u\right\| \leqslant \lambda_{\max}/\lambda_{\min} \left\|R^{(S)}\right\|. \tag{3.52}$$

Оценка (3.52) мажорантная, а множитель $\lambda_{\max}/\lambda_{\min} = O(N^2) \gg 1$. Поэтому оценка (3.52) малопригодна.

Допустимые наборы. Логарифмический счет на установление использует наборы с границами, определенными по границам спектра. Это порождает следующую неудобную особенность. Нужно заранее задать полное число итераций S и выполнять все шаги этого набора. Если выполнить только первую половину шагов, то будет эффективно погашена только половина гармоник, а остальные гармоники лишь слабо затухнут, и никакой сходимости не будет. Иными словами, нужно строить наборы шагов, плотно накрывающие весь спектральный интервал. Такие классы наборов будем называть ***допустимыми***.

Кроме того, для алгоритмов описанного типа оценка S по ε (из (3.45) или (3.46)) является мажорантной и может давать большее значение S, чем реально нужно. При этом вопрос о фактической точности $u^{(S)}$ остается открытым.

Процедура сгущения. Воспользуемся двумя обстоятельствами. ***1°*** Регулярные участки линий на рис. 3.6 – 3.7, 3.12, 3.13 практически прямолинейны. ***2°*** Если набор шагов $\{\tau_s\}$ выбран, то сами шаги можно выполнять в произвольном порядке благодаря линейности процесса и устойчивости. Это позволяет строить последовательности двукратно сгущающихся сеток по τ, внешне напоминающие метод Ричардсона.

Для этого выберем некоторое небольшое значение S_0 ($1 \leqslant S_0 \leqslant 5$) и построим для него сетку $\{\tau_s^0\}$, $0 \leqslant s \leqslant S_0$ с помощью функции (3.38)–(3.39). Очевидно, эта сетка принадлежит классу допустимых сеток. Выполним на ней итерационный процесс и обозначим его результат через U_0.

Затем возьмем $S_1 = 2S_0$ и с помощью той же порождающей функции построим сетку $\{\tau_s^1\}$. Она также будет допустимой. Четные шаги $\{\tau_s^1\}$ совпадают с шагами сетки $\{\tau_s^0\}$, поэтому их можно не повторять, а взять уже вычисленный результат U_0 в качестве исходного для дальнейших итераций. Выполним счет на установление с нечетными шагами сетки $\{\tau_s^1\}$. Полученный результат обозначим U_1.

Потом возьмем сетку $\{\tau_s^2\}$ с $S_2 = 2S_1$ шагами, повторим описанную процедуру и так далее. В итоге получим последовательность решений U_q, $q = 1, 2, 3, ...$, соответствующих сгущающимся сеткам. При этом вычисления экономичны: суммарное число итераций во всех расчетах равно числу итераций последней сетки. Сама процедура сгущения эквивалентна перестановке шагов этой сетки.

Апостериорные оценки. Из экспоненциального характера сходимости логарифмического счета на установление следуют апостериорные асимптотически точные оценки норм погрешности:

$$\|U_q - u\| \approx \|U_{q+1} - U_q\|\,, \tag{3.53}$$

$$\|U_{q+1} - u\| \approx \|U_{q+1} - U_q\|^3 \,/\, \|U_q - U_{q-1}\|^2\,. \tag{3.54}$$

Оценка (3.54) означает экстраполяцию погрешности на U_{q+1}. Она учитывает, что регулярный участок кривой погрешности не проходит через начало координат. Этими эвристическими закономерностями можно пользоваться, пока расчеты не выйдут на ошибки округления. При этом (3.54) теряет применимость раньше, чем (3.53).

Формально для каждой сетки (кроме последней) можно пользоваться обеими оценками. Эти оценки для $S_0 = 3$ показаны на рис. 3.6, 3.8, 3.10 – 3.13 светлыми кружками. Видно, что они хорошо совпадают с действительными значениями погрешностей, вычисленными непосредственно по точному решению. Чем гуще сетка $\{\tau_s^q\}$, тем лучше совпадение, как и должно быть для асимптотически точной оценки.

Таким образом, предложенный метод дает ***апостериорную асимптотически точную*** оценку погрешности итерационного процесса счета на установление по эволюционно-факторизованной схеме со сгущающейся ЛТ-сеткой по τ. Такой характер сходимости напоминает метод Ричардсона. Ранее оценок точности подобного типа не предлагалось.

Заметим, что аналогия с методом Ричардсона является неполной. Оценки (3.53) – (3.54) применимы лишь к нормам погрешности. Подобную поточечную оценку погрешности при сгущении сетки по τ построить невозможно.

3.5.2. Алгоритм расчета. В расчетах необходимо задавать S заранее. Поэтому вопрос о фактической оценке точности состоит из двух частей: ***1°*** априорная оценка необходимого S, не влекущая за собой непроизводительных расчетов; ***2°*** апостериорное подтверждение точности.

Построение сетки. Для априорной оценки S необходимо задать требуемую точность ε. Здесь возможны две ситуации. 1) Точность $\varepsilon_{\text{п}}$, не превосходящая фоновую $\varepsilon_{\text{ф}}$, может быть задана пользователем. 2) Требуется рассчитать как можно точнее. В последнем случае необходимо оценить фон ошибок округления. Для этого существуют мажорантные оценки. Фон получается умножением ошибки единичного округления на число обусловленности матрицы. Минимально возможной оценкой является угловое число обусловленности [52], [53], но его трудно вычислить. В наших расчетах естественно вычисляется несколько завышенное спектральное число обусловленности $\varkappa_\lambda = \sum \lambda_{\alpha,\max} / \sum \lambda_{\alpha,\min}$. Тогда $\varepsilon_{\text{ф}} = 10^{-16.2}\varkappa_\lambda$ (для 64-разрядного программного обеспечения).

Положим $\varepsilon = \max\{\varepsilon_{\text{п}}, \varepsilon_{\text{ф}}\}$. Исходя из этой точности и оценок границ спектра, вычислим требуемое S. Будем делить это число рекуррентно на 2 до тех пор, пока не получится число, немного меньшее небольшого целого S_0 ($1 \leqslant S_0 \leqslant 5$). Полученное S_0 следует взять в качестве начального и применять процедуру сгущения, описанную в п. 3.5.1. Сгущение ведется до достижения требуемого S.

Апостериорные оценки. Все погрешности, кроме последней, вычисляются по оценке (3.53). Эта оценка надежна, поскольку применяется вдали от фона. Для последнего вычисления применим экстраполяцию (3.54) и сравним полученную оценку $\varepsilon_{\text{э}}$ с фоновой $\varepsilon_{\text{ф}}$. В качестве погрешности последнего вычисления выберем величину $\max\{\varepsilon_{\text{э}}, \varepsilon_{\text{ф}}\}$.

Для визуального контроля удобно выводить оценки погрешностей на график. В реальных расчетах максимальное $S \leqslant 80$, поэтому при $S_0 > 5$ на этом графике будет слишком мало точек.

3.6 Основные результаты главы

1. Для решения эллиптических уравнений счетом на установление предложен новый линейно-тригонометрический набор шагов. Коэффициенты уравнения могут быть переменными, а прямоугольные сетки – неравномерными. Набор строится в логарифмической шкале, что позволяет получать экспоненциальную скорость сходимости. Это значительно быстрее, чем для общих итерационных методов, и по трудоемкости эквивалентно быстрому преобразованию Фурье (но область применимости значительно шире). Предложенный набор уменьшает

число итераций в 1.5 раза по сравнению с логарифмически равномерным. Он более прост и надежен, чем известные ранее наборы.

2. Улучшены имеющиеся теоретические оценки скорости сходимости логарифмического счета на установление. Получена априорная оценка сходимости для линейно-тригонометрического набора. Она используется для построения логарифмической сетки.
3. Построена процедура упорядочивания шагов логарифмического набора, напоминающая метод Ричардсона для разностных сеток. Это позволяет получить апостериорные асимптотически точные оценки сходимости итерационного процесса. Ранее подобные оценки не были известны. Существовали только мажорантные оценки точности по невязке, которые могли быть хуже по точности на несколько порядков.
4. Найдены оптимальные значения границ логарифмического набора для двумерных и трехмерных задач и построены конструктивные оценки границ спектра, необходимые для их вычисления.

4. Диффузия в пограничных слоях

Для сингулярно возмущенных эллиптических уравнений в прямоугольных областях предложена адаптивная квазиравномерная сетка, детально передающая все характерные участки решения. Она позволяет решать сингулярно возмущенные эллиптические уравнения с очень узкими пограничными слоями и ограничиваться сетками с небольшим числом узлов.

4.1 Метод решения

4.1.1. Дифференциальное уравнение. Для простоты записи ограничимся двумерным случаем. Выберем квазиравномерную сетку, адаптированную к ширине пограничного слоя. Заметим, что хорошая сетка будет существенно неравномерной. Аппроксимируем задачу (1.8) консервативной разностной схемой

$$\left(\mu^2\Lambda_x + \mu^2\Lambda_y - \varkappa\right) u = -f. \tag{4.1}$$

Напомним, что одномерный трехточечный оператор по координате x имеет вид

$$\left(\Lambda_x u\right)_n = \frac{2}{h_{x,n+1/2} + h_{x,n-1/2}} \left[\frac{k_{x,n+1/2}}{h_{x,n+1/2}}(u_{n+1} - u_n) - \frac{k_{x,n-1/2}}{h_{x,n-1/2}}\left(u_n - u_{n-1}\right)\right], \tag{4.2}$$
$$h_{x,n+1/2} = x_{n+1} - x_n;$$

Выражение для Λ_y записывается аналогично. Обобщение на трехмерный случай строится очевидным образом.

Схема (4.1), (4.2) имеет точность $O\left(\sum h_\alpha^2\right)$, если коэффициенты k_α дважды непрерывно дифференцируемы. Это справедливо как для равномерных сеток по пространству, так и для неравномерных. Второй порядок сохраняется на неравномерных сетках и для слоистых сред, если поставить узлы сетки в точки разрыва коэффициентов и их производных.

Проведем расчеты на последовательности сгущающихся сеток N, $2N$, $4N$..., начиная с небольшого N. Поточечно сравнивая решения на этих сетках и применяя метод Ричардсона, получим апостериорную асимптотически точную оценку погрешности. При этом можно остановиться на той сетке, которая обеспечивает требуемую точность. Одновременно по скорости убывания погрешности подтверждаем порядок фактической точности. Это позволяет обойтись без использования априорных оценок сходимости или двусторонних оценок (которые зачастую сложно построить).

Еще более эффективным является рекуррентный метод Ричардсона [45], который на основе расчетов на нескольких сетках позволяет построить разностное решение с повышенным порядком точности. Однако он требует более высокой гладкости решения, которая возможна лишь при повышенной гладкости всех входных данных, включая коэффициенты уравнения. Поскольку схема (4.1), (4.2) симметрична, ошибка разлагается по четным степеням $1/N$. Поэтому каждое уточнение требует подключения одной дополнительной сетки и повышения гладкости всех входных данных на 2 единицы; порядок точности при этом повышается на 2.

4.1.2. Сеточные уравнения.

Счет на установление. На каждой сетке по пространству задачу (4.1) будем решать методами, описанными в главе 3. Применим счет на установление по эволюционно-факторизованной схеме, которая в данном случае имеет вид

$$\prod_\alpha \Big(E - \frac{\tau}{2}\Lambda_\alpha - \frac{\tau}{2}\varkappa_\alpha\Big)\,\frac{\hat u - u}{\tau} = \sum_\alpha \Lambda_\alpha u - \varkappa u + f. \tag{4.3}$$

Здесь $\varkappa_x + \varkappa_y = \varkappa$. Формально можно разбить $\varkappa$ на $\varkappa_\alpha$ произвольным образом, поскольку это не влияет на порядок аппроксимации. Однако это может повлиять на величину остаточного члена и на фактическую точность расчета. Вопрос об оптимальном разбиении будет рассмотрен ниже.

В качестве набора шагов выберем линейно-тригонометрический набор (3.38)–(3.39), построенный в логарифмической шкале. Напомним, что данный метод является наиболее быстрым для задачи (4.1) и обеспечивает экспоненциальную скорость сходимости итераций. В частности, решение с точностью ошибок округления получается не более, чем за ~ 100 итераций. Такая малая трудоемкость позволяет проводить многократное сгущение сеток по пространству. Напомним также, что этот итерационный метод позволяет вычислять апостериорную асимптотически точную оценку погрешности итераций.

Квазиравномерная сетка по пространству должна подстраиваться под ширину пограничного слоя (шаг на границе пропорционален μ), поэтому надо выбирать $h_{\max}/h_{\min} \sim 1/\mu$, что должно приводить к увеличению $\lambda_{\max}/\lambda_{\min}$ в $\sim 1/\mu^2$ раз. Однако множитель μ^2 перед лапласианом компенсирует это увеличение. Кроме того, наличие слагаемого $-\varkappa$ уменьшает отношение границ спектра и повышает устойчивость. Поэтому в сингулярно возмущенных задачах счет на установление с логарифмическим набором шагов будет сходиться заведомо не хуже, а в большинстве случаев заметно лучше, чем при $\varkappa = 0$.

Разбиение $\varkappa$. Рассмотрим вопрос о наилучшем разбиении $\varkappa_x + \varkappa_y = \varkappa$. Целесообразно выбрать его так, чтобы оптимизировать сходимость счета на установление. Для этого отношение границ спектра по направлениям должно быть минимальным, то есть

$$\min \frac{\lambda_{x,\max}}{\lambda_{x,\min}} \frac{\lambda_{y,\max}}{\lambda_{y,\min}}, \tag{4.4}$$

где минимум ищется по всем разбиениям $\varkappa_x+\varkappa_y=\varkappa$. В простейшем случае $\varkappa=\text{const}$ это дает

$$\min \frac{\lambda^0_{x,\max}+\varkappa_x}{\lambda^0_{x,\min}+\varkappa_x}\frac{\lambda^0_{y,\max}+(\varkappa-\varkappa_x)}{\lambda^0_{y,\min}+(\varkappa-\varkappa_x)}, \tag{4.5}$$

где λ^0 – собственные значения оператора Лапласа по направлениям. Решением (4.5) является $\varkappa_x=\varkappa_y=\varkappa/2$.

4.1.3. Пример. Проиллюстрируем описанный подход на содержательном примере и покажем трудности, возникающие в важных прикладных расчетах. В качестве области G возьмем квадрат $[-1,1]\times[-1,1]$. Положим также

$$\begin{gathered} k_\alpha(\mathbf{r})\equiv 1,\ \varkappa(\mathbf{r})\equiv 1,\ f(\mathbf{r})=\cos\left(\frac{\pi}{4}(x+y)^2\right)\cos\left(\frac{3\pi}{4}(y-x)\right),\quad \mathbf{r}\in G;\\ u(\mathbf{r})=2.5(x+y),\quad \mathbf{r}\in\Gamma. \end{gathered} \tag{4.6}$$

Общий вид решения этой задачи представлен на рис. 4.1, *а* в виде изолиний с фиксированным шагом. Видно, что при $\mu\ll 1$ решение имеет пограничный слой шириной $\sim\mu$, а на расстоянии нескольких μ от границы становится регулярным $u(\mathbf{r})\approx f(\mathbf{r})$.

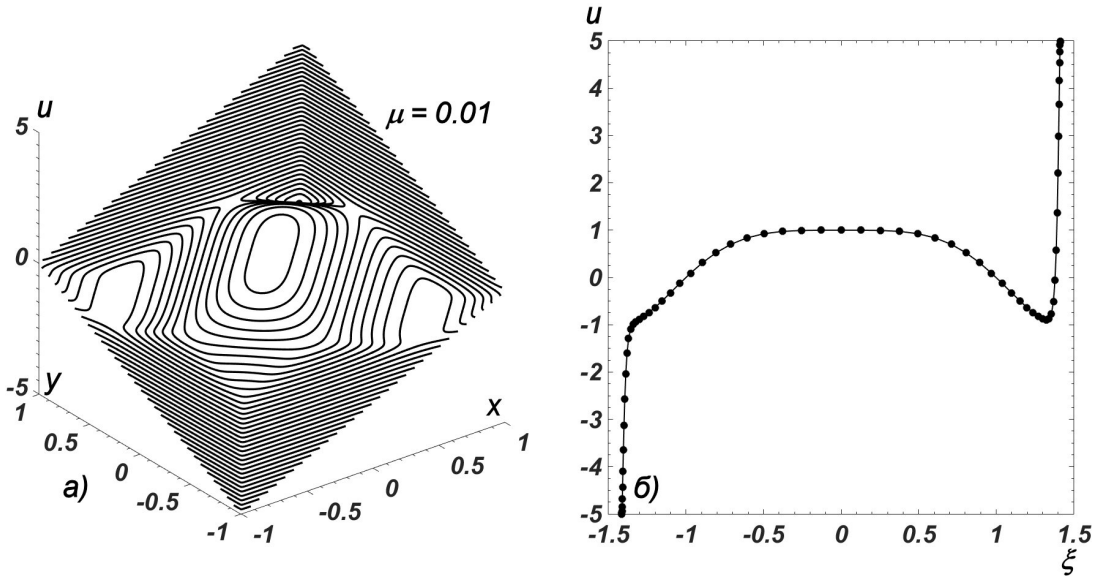

Рис. 4.1. Решение задачи (1.8), (4.6) для $\mu=10^{-2}$: а) общий вид, б) сечение плоскостью $x=y$.

4.2 Сетки по пространству

4.2.1. Квазиравномерная сетка. При решении важную роль играют не только пограничный слой и регулярная часть решения. Мы предлагаем выделять еще один участок решения – ***переходную зону***, расположенную между пограничным слоем и регулярным участком. Переходная зона характеризуется большой кривизной решения, что также трудно для расчета. Поэтому нужно строить такие сетки, в которых каждый из этих участков содержит примерно одинаковое количество узлов.

Эта рекомендация является общей, но на произвольных неструктурированных сетках ее реализовать трудно. На прямоугольных сетках ситуация заметно упрощается, поскольку многомерная сетка строится как декартово произведение одномерных квазиравномерных сеток.

Было опробовано много разных вариантов одномерных сеток. В результате мы остановились на сетке вида

$$x(\zeta) = A \operatorname{th}\left[C\zeta\left(1+\zeta^2/3\right)\right], \quad \zeta \in [-1,1]. \tag{4.7}$$

Параметры A и C подбираются так, чтобы 1) $x \in [-1,1]$ и 2) $x'(1) = \mu/(\mu+\varkappa)$. Второе условие задает шаг на границе. Эти условия сводятся к одному трансцендентному уравнению, которое легко решается методом Ньютона.

В качестве окончательной сетки для задачи (1.8), (4.6) выбирается декартово произведение сеток вида (4.7). Такая сетка подстраивается под особенности данного типа задач и хорошо работает при любом соотношении μ и $\varkappa$. На рис. 4.1, *б* представлено сечение решения задачи (1.8), (4.6) плоскостью $x = y$. Это сечение содержит два угловых пограничных слоя в правом верхнем и левом нижнем углах области G. В других углах пограничных слоев не возникает в силу выбора граничных условий. Видно, что сетка (4.7) дает примерно одинаковое число узлов на каждом из интересующих нас участков решения.

4.2.2. Шаг сетки. Шаг сетки можно определить двумя способами. Первый способ – через разность двух соседних узлов $h_{n+1/2} = x_{n+1} - x_n$. При нем выполняется баланс: сумма всех шагов равна длине отрезка. Поэтому он выглядит естественным.

При втором способе шаг определяется через производную производящей функции в полуцелом узле $h_{n+1/2} = x'(\zeta_{n+1/2})/N$. Этот способ выглядит искусственным, и при нем баланс не выполняется. Однако, во-первых, он эквивалентен некоторой замене переменных.

Во-вторых, в неограниченной области естественный способ неприменим (последний шаг оказывается бесконечно большим). Поэтому второй способ единообразно применим как в конечных, так и в бесконечных областях.

В-третьих, полный дисбаланс составляет величину $O(N^{-2})$ и достаточно быстро стремится к нулю при $N \to \infty$.

Таким образом, в конечной области можно пользоваться обоими способами. Встает вопрос, какой из них является более точным. Этот вопрос обсуждался в [45], однако никаких определенных выводов не было сделано.

Здесь мы исследовали этот вопрос. Для выбора лучшего способа проводилось сравнение погрешности при двух указанных определениях шага. Типичная картина представлена на рис. 4.2, *а*. Видно, что шаг, заданный через производную, дает несколько лучшую точность. В рассматриваемом примере точности отличаются в ~ 2 раза, и это различие практически не меняется при сгущении сеток. Таким образом, целесообразно всегда пользоваться определением шага через производную производящей функции.

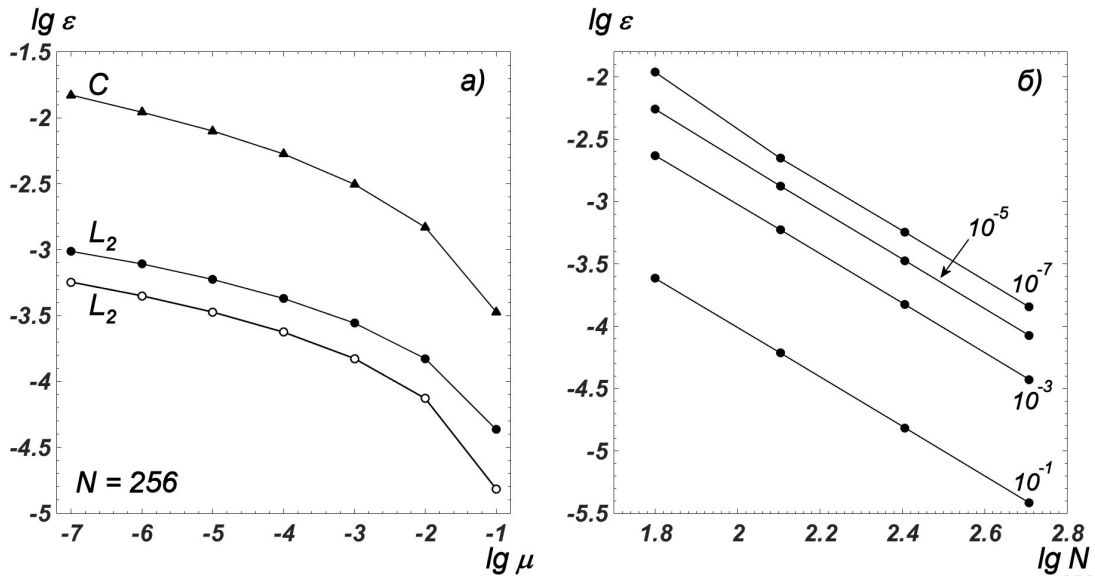

Рис. 4.2. Погрешности в задаче (1.8), (4.6); а) при разных μ и фиксированном $N = 256$, черные маркеры – шаг через разность x_n, светлые маркеры – шаг через производную производящей функции, около кривых указаны нормы, в которых вычислены погрешности; б) при разных N и фиксированных μ (указаны около кривых).

Сравнивались также равномерная (C) и среднеквадратичная (L_2) нормы погрешности. На рис. 4.2, *а* такое сравнение приведено для шага, определенного через разность соседних узлов. Для шага через производную картина аналогична, и во избежание загромождения графика кривая в норме C не приводилась. Видно, что погрешность в норме L_2 оказывается заметно меньше (в среднем в 10 раз).

В самом деле, равномерная и среднеквадратичная нормы точно совпадают в том случае, если во всех пространственных точках значения погрешности одинаковы. Чем больше разброс этих погрешностей, тем больше отличие норм C и L_2. Не слишком большая разница между этими нормами говорит о том, что локальные погрешности на разных участках не слишком сильно (хотя и не так уж мало) отличаются друг от друга. Это свидетельствует о том, что квазиравномерная сетка построена неплохо, хотя ее можно еще улучшить, соответственно несколько повысив точность решения.

4.2.3. Установление сходимости. На рис. 4.2, *б* представлены кривые сходимости при сгущении сеток по пространству, масштаб графика двойной логарифмический. Каждая кривая соответствует своему значению μ: от 10^{-1} до 10^{-7}. Этот диапазон достаточно представительный. Обычно в литературе ограничиваются значениями $\mu \leqslant 10^{-3} \div 10^{-4}$.

Все эти кривые выходят на прямые линии. Это доказывает степенной характер сходимости по пространству. Наклон этих прямых равен 2, что подтверждает второй порядок сходимости схемы (4.1), (4.2). Таким образом, даже для сингулярно возмущенных задач метод Ричардсона с визуальным контролем порядка точности оказывается хорошо применимым.

4.3 Обобщения

1° Описанный алгоритм очевидным образом переносится на случай более общего эллиптического оператора

$$Lu = \sum_{\alpha} L_\alpha u, L_\alpha u = \frac{\partial}{\partial x_\alpha}\left(k_\alpha(\mathbf{r})\frac{\partial u}{\partial x_\alpha}\right) \tag{4.8}$$

с произвольными $k_\alpha(\mathbf{r}) \neq \text{const}$. Это соответствует процессам в неоднородной среде. Напомним, что условиями применимости эволюционной факторизации являются отсутствие смешанных производных и возможность сведения области к прямоугольной.

2° Поскольку эволюционно-факторизованная схема (4.3) единообразно записывается для произвольного числа измерений, то данный метод позволяет решать трехмерные задачи. Вопрос о наилучшем разбиении $\varkappa = \varkappa_x + \varkappa_y + \varkappa_z$ решается аналогично.

3° Данный метод позволяет эффективно решать задачи в неограниченных областях (например, задачи дифракции на непрозрачном теле). В этом случае пространственная сетка должна хорошо передавать как резкий пограничный слой, так и решение на бесконечности. Поэтому ее следует строить в виде гладкого сшивания сгущающейся части вида (4.7) при $\zeta \in [-1, 0]$ и "разбегающейся" части вида

$$x(\zeta) = \frac{A\zeta}{(1-\zeta^2)^a} \quad \zeta \in [0, 1], \tag{4.9}$$

где A и a – некоторые постоянные. В практических расчетах рекомендуется выбирать $a = 1/2$.

В этом случае отношение границ спектра $\lambda_{\max}/\lambda_{\min} \sim N^{2a+2}$, то есть гораздо хуже, чем для ограниченных областей. Однако при применении логарифмического набора трудоемкость увеличивается всего в $a+1$ раз по сравнению с задачей в ограниченной области. Такое увеличение трудоемкости незначительно, так что предложенный метод остается высоко конкурентоспособным по отношению к известным в литературе методам.

4.4 Основные результаты главы

1. В структуре решения сингулярно возмущенных задач предложено выделять не только пограничный слой и регулярную область, но и переходную зону между ними. Она характеризуется большой кривизной решения и также представляет трудности для расчета.
2. Для задач в прямоугольной области предложена адаптивная квазиравномерная сетка, детально разрешающая все участки решения (пограничный слой, переходную зону, регулярную область). Она обеспечивает высокую точность даже при очень тонких пограничных слоях (например $\mu \sim 10^{-7}$) уже на скромных сетках с числом узлов $N \sim 250 \div 500$.

3. Показано, что метод Ричардсона можно использовать не только в регулярных, но и в сингулярно возмущенных задачах. Он позволяет находить апостериорную асимптотически точную оценку погрешности и устанавливать порядок фактической точности. При этом он хорошо передает степенные зависимости погрешности от числа узлов, в том числе с нецелыми показателями, которые возникают при ограниченной гладкости решения. Последнее важно для применения метода к широкому кругу практических задач. Метод позволяет обнаруживать более сложные зависимости (например, $N^\alpha \ln N$) по более медленному выходу на асимптотический режим. В этом случае конкретный тип зависимости можно искать на основе дополнительных гипотез.

5. Скорости термоядерных реакций

Для моделирования процессов в термоядерных мишенях требуются достоверные данные о скоростях реакций. Скорость реакции равна свертке сечения реакции $\sigma(E)$ с функцией распределения по энергиям. Строго говоря, процессы горения являются неравновесными, но их функцию распределения можно приблизить распределением Максвелла. Это приближение разумно при достаточно высокой плотности вещества, когда возможно возникновение локального термодинамического равовесия. Сечения реакций измеряются экспериментально. Поэтому нахождение скоростей реакций сводится к обработке экспериментальных данных для $\sigma(E)$.

В данной главе разработан новый метод обработки экспериментальных данных, измеренных со значительной погрешностью. Он заключается в построении аппроксимации по методу двойного периода со специальным регуляризатором. С использованием этого метода получены аппроксимации для сечений 4 термоядерных реакций, наиболее актуальных для УТС. Вычислены скорости этих реакций, и проведены расчеты их кинетики.

5.1 Метод двойного периода

Кратко напомним суть метода двойного периода. Пусть задан большой массив экспериментальных точек: аргументов x_i и функций $u_i \pm \delta_i$, $1 \leqslant i \leqslant I$ ($I \gg 1$), где δ_i – абсолютные ошибки измерений. Из смысла задачи известно, что $u(x)$ есть гладкая функция с не слишком большими старшими производными, однако экспериментальные ошибки δ_i могут быть значительными.

Для аппроксимации гладких непериодических функций рядами Фурье удобен метод двойного периода [54, 55]. Для простоты записи линейно преобразуем аргумент x так, чтобы $\min x_i = -\pi/2$, $\max x_i = \pi/2$. Затем возьмем следующие гармоники Фурье для отрезка $[-\pi/2, \pi/2]$: нулевую гармонику и N пар синус-косинус. Дополним их M гармониками удвоенного периода $[-\pi, \pi]$, не являющимися гармониками основного периода. Эти гармоники подключаются в расчет не парами синус-косинус, а по одиночке, так что их число M может быть любой четности.

Аппроксимируем $u(x)$ суммой гармоник основного и двойного периода. Для единообразия выберем следующую форму записи:

$$u(x) \approx \sum_{n=0}^{2N+M} a_n \varphi_n(x). \tag{5.1}$$

Коэффициенты a_n подбирают из условия наилучшей аппроксимации $u(x)$ в норме L_2.

Здесь гармоники основного периода имеют индекс $0 \leqslant n \leqslant 2N$ и записываются следующим образом:

$$\begin{array}{llll} \varphi_n(x) = \cos 2mx, & \varphi_n'(x) = -2m\varphi_{n-1}(x), & \varphi_n''(x) = -4m^2\varphi_n(x), & \text{если } n = 2m; \\ \varphi_n(x) = \sin 2mx, & \varphi_n'(x) = 2m\varphi_{n+1}(x), & \varphi_n''(x) = -4m^2\varphi_n(x), & \text{если } n = 2m-1. \end{array} \tag{5.2}$$

В (5.2) приведены производные этих гармоник, которые потребуются нам в дальнейшем. Гармоники двойного периода имеют индекс $2N+1 \leqslant n \leqslant 2N+M$, то есть записываются после гармоник основного периода. Они выглядят следующим образом:

$$\begin{array}{lll} \varphi_n(x) = \cos(2m-1)x, & \varphi_n'(x) = -(2m-1)\varphi_{n-1}(x), & \\ & \varphi_n''(x) = -(2m-1)^2\varphi_n(x), & \text{если } n = 2N+2m; \\ \varphi_n(x) = \sin(2m-1)x, & \varphi_n'(x) = (2m-1)\varphi_{n+1}(x), & \\ & \varphi_n''(x) = -(2m-1)^2\varphi_n(x), & \text{если } n = 2N+2m-1. \end{array} \tag{5.3}$$

Формально уже гармоник основного периода достаточно для аппроксимации при $N \to \infty$. Поэтому присутствие членов с $n > 2N$ является в этом случае избыточным. При конечном N сумма (5.1) избыточной не является. В [54] показано, что подключение в расчет гармоник двойного периода эквивалентно повышению гладкости периодического продолжения $u(x)$ на период $[-\pi, \pi]$. Каждая лишняя гармоника двойного периода повышает гладкость на единицу.

Чем больше M, тем быстрее сходятся разложения по гармоникам основного периода. При $N \to \infty$ и фиксированном M приближение (5.1) аппроксимирует $u(x)$ и ее q-е производные в норме C с точностью $O(N^{q-M})$. При численных расчетах следует брать лишь небольшие значения M, так как увеличение M катастрофически ухудшает обусловленность расчетов. Однако значения N можно брать большими для получения хорошей точности аппроксимации. Разумеется, для полного числа параметров должно выполняться $2N + M + 1 < I$.

5.2 Регуляризация метода двойного периода

Описанный метод был разработан для функций непрерывного аргумента или заданных на равномерной сетке, когда значения $u(x)$ вычисляются с высокой точностью. В экспериментах сетка x_i резко неравномерная и может содержать пробелы, а погрешности δ_i нередко велики. Под пробелами мы понимаем участки аргумента значительной длины, на которых отсутствуют экспериментальные точки. При аппроксимации таких данных надо использовать регуляризацию. Естественно взять стабилизатор А. Н. Тихонова [56], содержащий интегралы от квадратов различных производных $u(x)$. Обсудим, какие производные целесообразно использовать.

5.2.1. Выбор регуляризатора.

Общий случай. Очевидно, мы должны с хорошей точностью аппроксимировать $u(x)$ и $u'(x)$. Поэтому включать их в стабилизатор нельзя: это приведет к занижению их величин и, соответственно, ухудшению аппроксимации. Однако, если не говорить о радиотехнических задачах, то физические кривые обычно являются достаточно плавными и не содержат высокочастотных осцилляций, так что их кривизна невелика. Поэтому величину $u''(x)$ целесообразно ограничивать, включая ее в стабилизатор. Этого обычно достаточно для хорошего подавления нефизичных осцилляций.

При обработке кривых с узкими пиками (например, нейтронные резонансы) вторая производная в регуляризаторе может привести к "заглаживанию" верхушки пика. Для таких задач вместо второй производной в регуляризатор следует включать четвертую.

Включение четных производных имеет еще одно практическое преимущество: при двукратном дифференцировании синус переходит в синус, а косинус – в косинус. Использование нечетных производных нецелесообразно, так как при этом портится структура матрицы.

Специальные слагаемые. Учтем еще два обстоятельства, относящихся к нашей частной задаче аппроксимации сечений для ядерных реакций. Во-первых, важна не абсолютная точность аппроксимации сечений, а относительная, поскольку сечения меняются в очень широких пределах. Во-вторых, специфические переменные при этом выбираются так, чтобы $u(x) \approx \text{const}$ при наименьших x_i. Поэтому в стабилизатор надо добавить значения $(u')^2$ и $(u'')^2$ на левой границе с большими весовыми множителями, чтобы аппроксимирующая кривая выходила на горизонтальную прямую. Это соответствует теоретическому поведению сечения по формуле Гамова.

С учетом этих соображений метод наименьших квадратов приводит к следующей формулировке:

$$\sum_{i=1}^{I} \left(\frac{u(x_i) - u_i}{\delta_i} \right)^2 + \alpha \int_{-\pi/2}^{\pi/2} [u''(x)]^2 dx + \beta [u'(-\pi/2)]^2 + \gamma [u''(-\pi/2)]^2 = \min; \qquad (5.4)$$

здесь $\alpha,\ \beta,\ \gamma > 0$. Подставляя сюда (5.1), получим задачу на минимум для коэффициентов a_n.

Интеграл в (5.4) целесообразно брать в смысле функций непрерывного аргумента, не увязывая его с дискретной сеткой x_i. При его вычислении надо учитывать, что каждая из подсистем (5.2) и (5.3) сама по себе ортогональна. Однако эти подсистемы не ортогональны друг другу, хотя синусы одной подсистемы ортогональны косинусам другой.

Замечание 1. Традиционный подход к регуляризации заключается в том, что задача на минимум (5.4) сводится к решению дифференциального уравнения для $u(x)$. Порядок этого уравнения вдвое выше, чем порядок максимальной производной, входящей в регуляризатор. Такое уравнение требует соответствующего числа дополнительных условий, равного порядку дифференциального уравнения. Формальная постановка таких краевых условий обычно приводит к заметному отличию регуляризованного решения от истинного вблизи границ интервала.

Вдобавок, решение краевой задачи для дифференциального уравнения высокого порядка само по себе представляет определенную математическую трудность. Поэтому на практике регуляризаторы высокого порядка используют нечасто.

Метод двойного периода естественно преодолевает эту трудность. В нем легко использовать регуляризатор высокого порядка, не содержащий низших производных. При этом гораздо проще получить близость регуляризованного решения к точному, а алгоритм решения не усложняется.

Замечание 2. Интегральный стабилизатор А. Н. Тихонова обеспечивает хорошее поведение всей регуляризованной кривой в целом. Поэтому его целесообразно применять для широкого класса прикладных задач. Однако дополнительные члены, связанные с граничными условиями, передают специфику именно данной прикладной задачи обработки данных термоядерного эксперимента. Для многих других задач они не нужны. Тогда можно пользоваться всеми приведенными ниже формулами, положив $\beta = \gamma = 0$.

5.2.2. Линейная система. Выполним минимизацию (5.4) по a_n. Это приведет к системе линейных уравнений для определения этих коэффициентов. Для ее записи введем некоторые вспомогательные обозначения. Запишем следующие скалярные произведения, связанные с экспериментальным материалом:

$$\begin{aligned} \langle \varphi_n, \varphi_k \rangle = \langle \varphi_k, \varphi_n \rangle &= \sum_{i=1}^{I} \frac{\varphi_n(x_i)\varphi_k(x_i)}{\delta_i^2}, \ 0 \leqslant n, k \leqslant 2N + M; \\ \langle \varphi_n, u \rangle &= \sum_{i=1}^{I} \frac{\varphi_n(x_i) u_i}{\delta_i^2}, \qquad 0 \leqslant n \leqslant 2N + M. \end{aligned} \tag{5.5}$$

При произвольных узлах и весах (что соответствует реальному экспериментальному материалу) все эти матричные элементы отличны от нуля, то есть матрица линейной системы будет плотно заполненной.

Введем также скалярные произведения базисных функций, связанные с регуляризатором. Матрица этих скалярных произведений имеет блочный характер. Она содержит две квадратных диагональных клетки

$$(\varphi_0, \varphi_0) = \pi, \quad (\varphi_n, \varphi_k) = \frac{\pi}{2}\,\delta_{n,k} \text{ при } 0 < n, k \leqslant 2N \text{ или } 2N + 1 \leqslant n, k \leqslant 2N + M. \tag{5.6}$$

В нее также входят две прямоугольные недиагональные клетки, в которых элементы, соответствующие скалярным произведениям синуса одной подсистемы на косинус другой, равны нулю. Эти нулевые элементы расположены в шахматном порядке. Поэтому для $n \leqslant 2N$, $k \geqslant 2N + 1$ можно записать

$$(\varphi_n, \varphi_k) = \begin{cases} 0, & \text{если } n + k \text{ нечетное;} \\ \dfrac{(-1)^{(m-n)/2}}{n - m + 1} - \dfrac{(-1)^{(m+n)/2}}{n + m + 1}, & m = k - 2N, \quad \text{если } n \text{ и } k \text{ нечетные;} \\ \dfrac{(-1)^{(m-n)/2}}{n - m + 1} - \dfrac{(-1)^{(m+n)/2}}{n + m - 1}, & m = k - 2N, \quad \text{если } n \text{ и } k \text{ четные.} \end{cases} \tag{5.7}$$

Вторая недиагональная клетка $k \leqslant 2N$, $n \geqslant 2N+1$ симметрична данной. Кроме этого введем обозначения для частот гармоник

$$\begin{aligned} \omega_n &= n + \frac{1-(-1)^n}{2} = 2\left[\frac{n+1}{2}\right], \quad 0 \leqslant n \leqslant 2N; \\ \omega_n &= m - \frac{1+(-1)^m}{2} = 2\left[\frac{m+1}{2}\right], \quad m = n-2N,\ 2N+1 \leqslant n \leqslant 2N+M. \end{aligned} \tag{5.8}$$

Здесь квадратная скобка обозначает целую часть числа. В этих обозначениях линейная система принимает следующий вид:

$$\sum_{k=0}^{2N+M} A_{nk} a_k = \langle u, \varphi_n \rangle, \quad 0 \leqslant n \leqslant 2N+M. \tag{5.9}$$

Здесь

$$A_{n,k} = \langle \varphi_n, \varphi_k \rangle + \alpha \omega_n^2 \omega_k^2 (\varphi_n, \varphi_k) + \beta \omega_n \omega_k B_{nk} + \gamma \omega_n^2 \omega_k^2 C_{n,k}; \tag{5.10}$$

$$B_{nk} = \begin{cases} (-1)^{(n+k)/2+1}, & \text{если } n \text{ и } k \text{ нечетные и } 0 \leqslant n,k \leqslant 2N, \\ (-1)^{(n+k)/2}, & \text{если } n \text{ и } k \text{ четные и } 2N+1 \leqslant n,k \leqslant 2N+M, \\ 0, & \text{во всех остальных случаях;} \end{cases} \tag{5.11}$$

$$C_{nk} = \begin{cases} (-1)^{(n+k)/2}, & \text{если } n \text{ и } k \text{ четные и } 0 \leqslant n,k \leqslant 2N, \\ (-1)^{(n+k)/2+1}, & \text{если } n \text{ и } k \text{ нечетные и } 2N+1 \leqslant n,k \leqslant 2N+M, \\ 0, & \text{во всех остальных случаях.} \end{cases} \tag{5.12}$$

При использовании четвертой производной общий вид линейной системы будет тем же, но во втором слагаемом в (5.10) будет произведение $\omega_n^4 \omega_k^4$.

5.2.3. Решение линейной системы. При аппроксимации функций, заданных на равномерной сетке с одинаковой точностью, скалярные произведения $\langle \varphi_n, \varphi_k \rangle$ имеют структуру, напоминающую структуру матрицы (φ_n, φ_k). Она также содержит две диагональных клетки и две недиагональных. При этом диагональные клетки сами оказываются диагональными подматрицами. Это улучшает обусловленность и позволяет взять даже $M \sim 5 \div 6$. Однако при неравномерной сетке x_i и неодинаковых δ_i диагональные клетки плотно заполнены, и обусловленность существенно ухудшается. В этом случае в расчетах следует ограничиться $M = 3$, что эквивалентно гладкости периодического продолжения $u''(x)$. Меньшее число M не обеспечивает нужной гладкости.

По свойствам метода наименьших квадратов матрица A является симметричной и положительно определенной. Ее целесообразно решать методом Гаусса. При этом выбирать главный элемент не следует, так как он автоматически оказывается на главной диагонали. Такой алгоритм обеспечивает хорошую надежность даже для матриц A с довольно большим числом обусловленности $\varkappa$ (до $\varkappa \sim 10^8$ при 64-битовых вычислениях).

5.3 Сечения термоядерных реакций

5.3.1. Эксперименты. Рассмотрим реакции (2.17) – (2.20). Основной массив экспериментальных данных по их сечениям приведен в базе [24]. Эта база использует около 90 оригинальных работ и содержит 2000 экспериментальных точек. Данные разных авторов не всегда хорошо согласованы. Иногда расхождения достигают 6 раз! Авторским оценкам погрешности далеко не всегда можно доверять, а результат аппроксимации достаточно сильно зависит от значений δ_i.

Поэтому мы провели свой критический анализ экспериментов и дали собственные оценки δ_i. В частности, мы приписывали большие погрешности работам, где мишени содержали тяжелые металлы, насыщенные дейтерием. В тяжелых металлах налетающий пучок теряет много энергии, и аккуратно оценить фактическую энергию в момент столкновения частиц становится невозможно. Лучшими мишенями считались либо чисто дейтериевые, либо содержащие соединения дейтерия с наиболее легкими элементами (LiD, CD_2, CD_4, D_2O). Учитывался также международный авторитет лабораторий и год выполнения работы (например, данные Лос-Аламоса даже 1950-х гг. заслуживают большего доверия, чем новейшие измерения некоторых европейских лабораторий). Подробный разбор каждой работы мы здесь не приводим, так как это представляет интерес только для физиков, специализирующихся в данной области.

5.3.2. Переменные. Опишем выбор переменных. Мы будем строить широкодиапазонную аппроксимацию. Поэтому в качестве аргумента удобно использовать $x = \lg E$, где E – энергия сталкивающихся частиц в системе центра масс. Однако сечение реакций $\sigma(E)$ или его логарифм неудобен в качестве функции. Гораздо удобнее взять величину, называемую S-фактором. Она получается умножением сечения на фактор Гамова, учитывающий проницаемость кулоновского барьера,

$$S(E) = \sigma(E) E \exp\left\{\pi z_1 z_2 \sqrt{\frac{2m}{E}}\right\}, \tag{5.13}$$

где m – приведенная масса, z_1 и z_2 – заряды частиц (система единиц атомная). За функцию берется $u = \lg S$. Эта величина изменяется всего на $1.5 \div 2$ единицы.

Кривые S-факторов всех реакций после обработки приведены на рис. 5.2. Видно, что для обоих каналов реакции D + D они близки и выглядят как переход с горизонтальной прямой на наклонную, но для остальных двух реакций они имеют вид кривой с максимумом. Такие отличия связаны с существенно разными физическими механизмами реакций. При традиционном физическом подходе для каждой реакции нужно подбирать свой вид аппроксимирующей формулы, что достаточно трудно. А метод двойного периода позволяет единообразно обработать все эти реакции.

5.3.3. Результаты расчетов.

Параметры регуляризации. Во всех расчетах использовалось $M = 3$. Это дает достаточную гладкость периодического продолжения, но еще не слишком сильно

портит обусловленность. Значения N и параметров регуляризации следует выбирать для каждой реакции отдельно. Ограничимся только параметрами α и β, а γ будем считать равным нулю; этого вполне достаточно для получения хороших результатов.

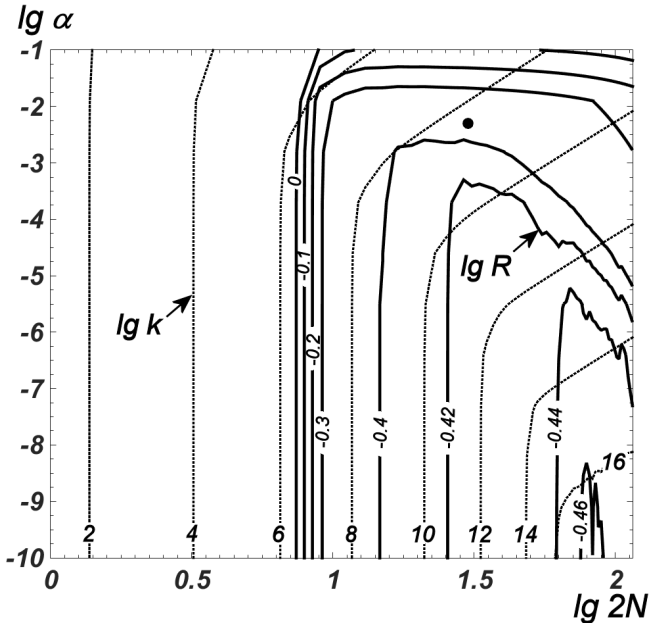

Рис. 5.1. Выбор параметров регуляризации для реакции $\mathrm{D} + \mathrm{T} \to n + {}^4\mathrm{He}$; пунктир – изолинии числа обусловленности $\lg k$, жирные линии – изолинии невязки $\lg R$, • – выбранные N и α.

Процедуру проведем в 2 этапа. На первом этапе положим $\beta = 0$ и будем выбирать N и α совместно по невязке R (первое слагаемое в (5.4)) и числу обусловленности k. Для этого проведем расчеты при разных α и N и построим на одних осях изолинии $\lg R$ и $\lg k$. Пример такого графика для реакции (2.19) приведен на рис 5.1.

Невязка представляет собой желоб с искривленным дном. Будем двигаться по дну этого желоба в сторону увеличения N и остановимся на той паре N и α, при которых невязка перестает заметно убывать, а число обусловленности еще не слишком велико. Величина $\lg k$ равна числу десятичных знаков, теряемых в коэффициентах Фурье на ошибки округления. На 64-разрядном программном обеспечении приемлемы $\lg k \leqslant 10$. Убедимся, что при выбранных N и α на аппроксимирующей кривой отсутствуют крупно- и мелкомасштабные осцилляции. Наличие первых говорит о том, что N недостаточно велико (при увеличении α эти осцилляции сохраняются). Наличие вторых означает, что выбрано слишком маленькое α.

На втором этапе будем увеличивать β и анализировать поведение аппроксимирующей кривой. При этом у нее будет изменяться (выравниваться) только участок вблизи левого края. При включении β это изменение будет ощутимым по сравнению со случаем $\beta = 0$, но по мере увеличения β оно будет становиться все менее заметным. То β, при котором изменение аппроксимирующей кривой перестает быть визуально различимым, и следует взять как окончательное. При этом целесообразно контролировать значения u' и u'' на левой границе и число обусловленности. Параметры, подобранные для различных реакций, и соответствующие числа обусловленности $\lg k$

приведены в табл. 5.1.

Таблица 5.1. Параметры регуляризации в задаче (5.4) для реакций (2.17) – (2.20).

	DD $\to$ pT	DD $\to$ n^{3}He	DT $\to$ n^{4}He	D^{3}He $\to$ p^{4}He
I	717	364	370	544
N	50	40	15	25
α	0.3	0.3	0.005	0.05
β	1000	1000	100	100
k	10^8	$7 \cdot 10^7$	10^6	$4 \cdot 10^7$
Δ, %	0.1	0.2	0.3	0.2

Видно, что общее число свободных параметров от 24 до 14 раз меньше, чем I для соответствующей реакции. Таким образом, для аппроксимации использовались весьма высокие гармоники, что обеспечивало хорошую точность. Одновременно регуляризация хорошо сглаживала высокие гармоники, так что аппроксимирующие кривые получались плавными. При этом объем вычислений ничтожен, все расчеты легко выполняются даже на 64-битовом ноутбуке.

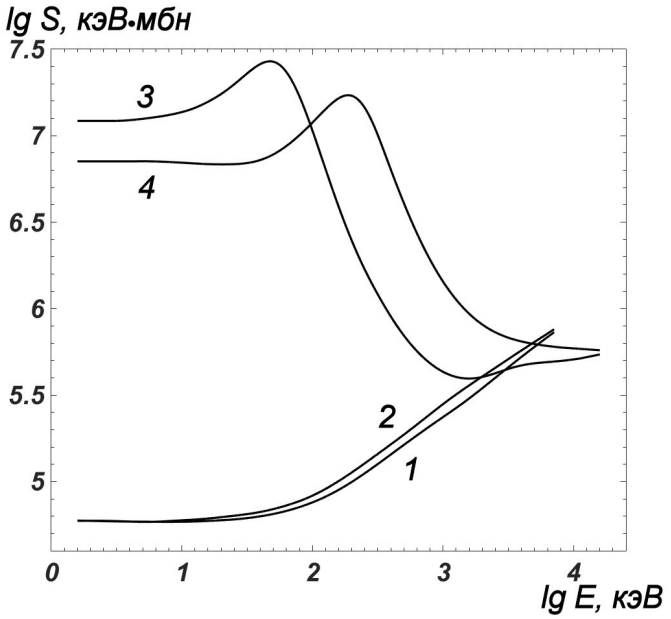

Рис. 5.2. S-факторы из табл. 5.2; 1 – D + D $\to p$ + T, 2 – D + D $\to$ n + ^{3}He, 3 – D + T $\to$ n + ^{4}He, 4 – D +^{3} He $\to$ p + ^{4}He.

Результаты. Приведем окончательные результаты. В табл. 5.2 и на рис. 5.2 представлена зависимость $\lg S$ [кэВ·мбн] от $\lg E$ [кэВ] для всех 4 реакций. В сторону меньших энергий каждая колонка табл. 5.2 продолжается как константа. Из рис. 5.2 хорошо видно, что аппроксимирующая кривая для каждой реакции имеет естественный вид и не содержит нефизичных осцилляций несмотря на использование высоких гармоник. Отметим, что для реакции (2.20) в диапазоне $3 \leqslant \lg E \leqslant 3.2$ отсутствовали экспериментальные данные. Аппроксимирующая кривая хорошо, без осцилляций

Таблица 5.2. S-факторы реакций $\lg S$, кэВ·мбн.

$\lg E$, кэВ	DD → pT	DD → n^{3}He	DT → n^{4}He	D^{3}He → p^{4}He	$\lg E$, кэВ	DD → pT	DD → n^{3}He	DT → n^{4}He	D^{3}He → p^{4}He
0.2	4.774	4.773	7.086	6.852	2.3	4.998	5.053	6.395	7.231
0.3	4.773	4.773	7.086	6.852	2.4	5.048	5.106	6.226	7.161
0.4	4.771	4.773	7.086	6.852	2.5	5.102	5.160	6.082	7.004
0.5	4.770	4.772	7.086	6.852	2.6	5.157	5.215	5.954	6.803
0.6	4.769	4.769	7.090	6.852	2.7	5.212	5.272	5.844	6.610
0.7	4.768	4.767	7.098	6.852	2.8	5.267	5.330	5.754	6.435
0.8	4.768	4.768	7.108	6.851	2.9	5.321	5.389	5.684	6.283
0.9	4.767	4.772	7.120	6.848	3.0	5.373	5.448	5.635	6.156
1.0	4.768	4.776	7.137	6.844	3.1	5.426	5.504	5.605	6.052
1.1	4.770	4.781	7.161	6.840	3.2	5.481	5.557	5.596	5.970
1.2	4.773	4.786	7.196	6.836	3.3	5.540	5.607	5.605	5.908
1.3	4.776	4.794	7.241	6.835	3.4	5.600	5.658	5.627	5.863
1.4	4.781	4.803	7.297	6.836	3.5	5.661	5.708	5.652	5.832
1.5	4.789	4.812	7.360	6.842	3.6	5.721	5.758	5.672	5.810
1.6	4.799	4.824	7.414	6.859	3.7	5.779	5.808	5.684	5.795
1.7	4.813	4.841	7.428	6.892	3.8	5.836	5.856	5.691	5.784
1.8	4.830	4.862	7.368	6.940	3.9	5.896	5.906	5.698	5.776
1.9	4.853	4.887	7.227	7.003	4.0	5.958	5.955	5.707	5.770
2.0	4.880	4.919	7.029	7.076	4.1	6.020	6.005	5.720	5.765
2.1	4.913	4.958	6.807	7.153	4.2	6.082	6.054	5.735	5.760
2.2	4.952	5.003	6.590	7.216	4.3	6.144	6.104	5.751	5.754

прошла этот участок. Это свидетельствует о хорошем качестве предложенной регуляризации. Видно, что данный метод позволяет надежно проводить кривую через "пробелы" экспериментальных данных.

Оценка погрешности. Будем считать, что экспериментальные точки являются случайными; причем их распределение гауссово, средним является построенная аппроксимирующая кривая, а дисперсией – экспериментальный вес δ_j. Для получения доверительного интервала полученной аппроксимирующей кривой проварьируем одновременно все экспериментальные точки по распределению Гаусса с указанными средним и дисперсией и построим для них новую аппроксимирующую кривую.

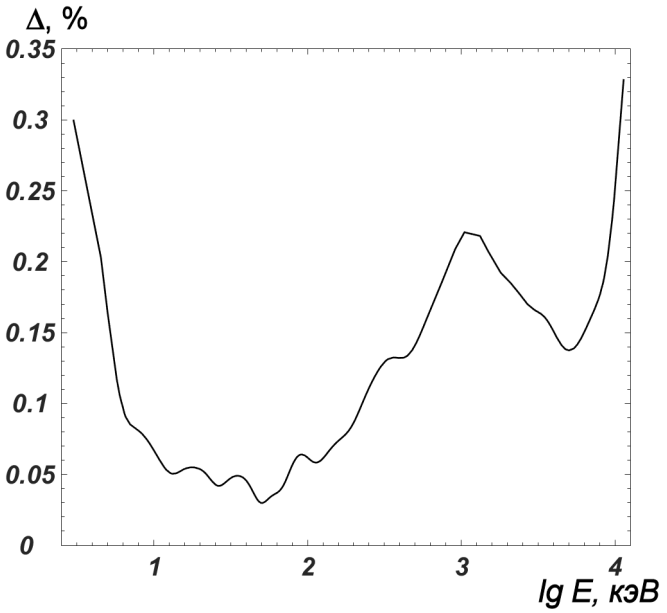

Рис. 5.3. Дисперсия кривой $S(E)$ для реакции $\mathrm{D} + \mathrm{T} \to n + {}^3\mathrm{He}$.

Повторяя эту процедуру достаточное количество раз, получим набор кривых, разброс которых дает оценки коридора дисперсии для полученной кривой. Для реакции (2.19) график относительной дисперсии в зависимости от энергии дан на рис. 5.3 . Нормы C относительных дисперсий Δ для всех реакций составляют от 0.1 до 0.3 % и приведены в табл. 5.1.

Сравнение с другими аппроксимациями. В практических расчетах термоядерных мишеней нередко используются известные формулы Б. Н. Козлова [26]. Сравнение с ними, а также с аппроксимациями Брауна [28] и Давиденко [40], представлено на рис. 5.4 – 5.6.

Видно, что при $E \leqslant 300$ кэВ формулы Козлова и Давиденко хорошо согласуются с экспериментами и нашей кривой, но при больших энергиях быстро теряют точность. Аппроксимации Брауна хорошо работают при $E \leqslant 1$ МэВ для реакции (2.17) и $E \leqslant 500$ кэВ для реакции (2.18). Заметим, что в этой работе измерения проводились в диапазоне $10 \leqslant E \leqslant 58$ кэВ. Успех столь далекой экстраполяции говорит об исключительно хорошем качестве эксперимента. Однако при энергиях выше $0.5 \div 1$ МэВ точность этих формул падает.

Таким образом, табл. 5.2 можно рекомендовать как надежные справочные дан-

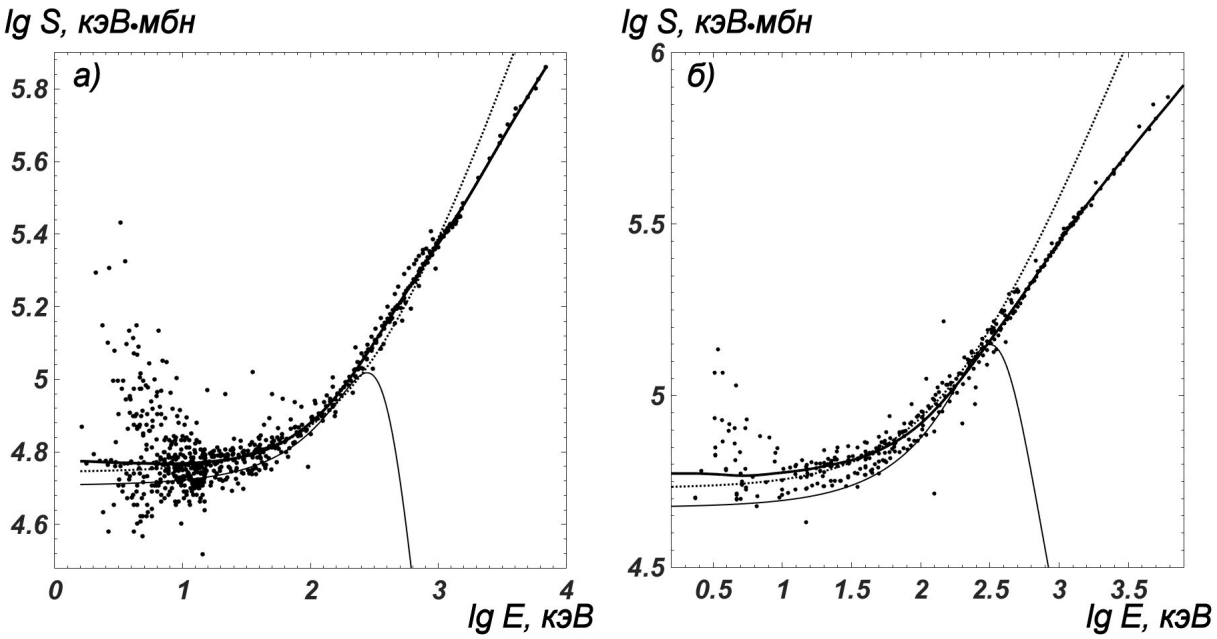

Рис. 5.4. S-фактор для реакций а) $D + D \to p + T$, б) $D + D \to n + {}^3He$; точки – экспериментальные значения, жирная линия – табл. 5.2, тонкая сплошная линия – формула Козлова, пунктир – аппроксимация Брауна.

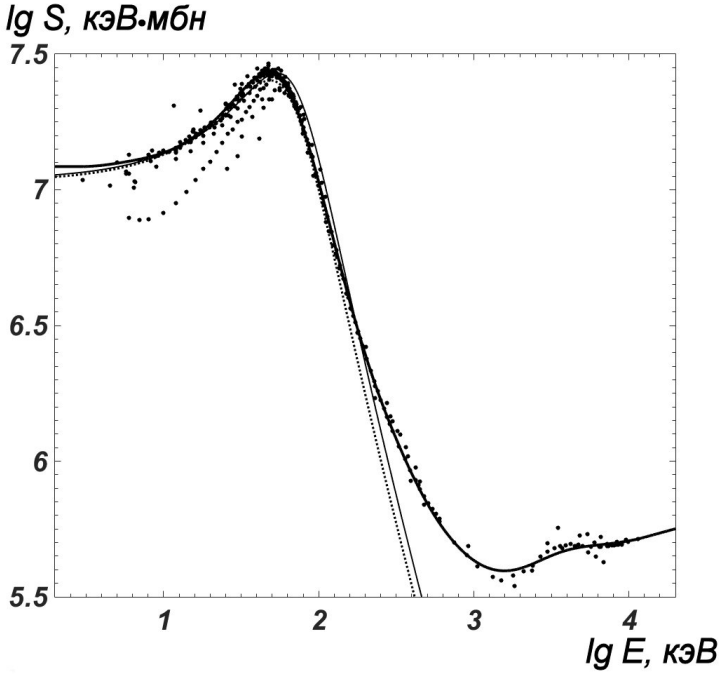

Рис. 5.5. S-фактор для реакции $D + T \to n + {}^4He$; точки – экспериментальные значения, жирная линия – табл. 5.2, тонкая сплошная линия – формула Козлова, пунктир – аппроксимация Давиденко.

ные. Они точнее предлагавшихся ранее (особенно велик выигрыш в точности при высоких энергиях).

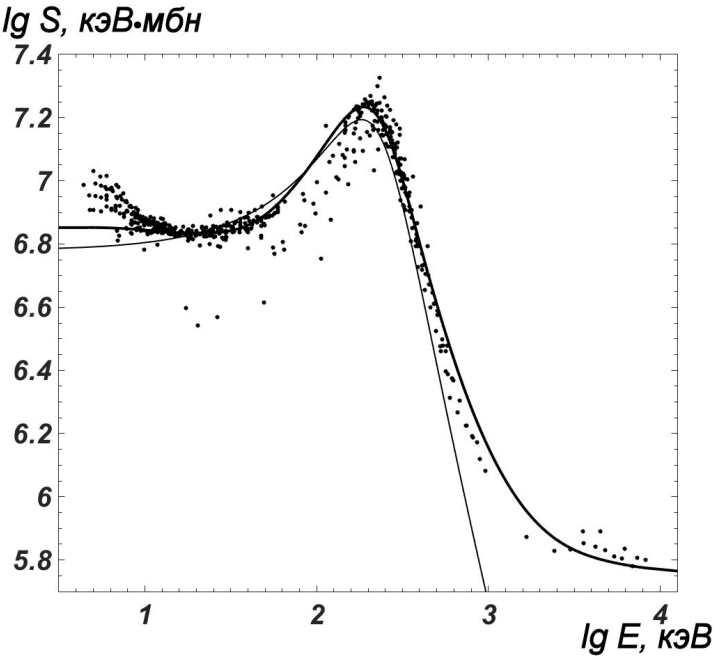

Рис. 5.6. S-фактор для реакции D $+{}^3$He $\to$ p $+{}^4$He; точки – экспериментальные значения, жирная линия – табл. 5.2, тонкая линия – формула Козлова.

5.4 Скорости термоядерных реакций

5.4.1. Таблицы скоростей. Пусть в веществе имеется локальное термодинамическое равновесие с температурой T. Тогда скорость термоядерной реакции определяется сверткой $\sigma(E)\sqrt{2E/m}$ с максвелловским распределением

$$K(T) = \frac{\pi}{\sqrt{m}} \left(\frac{2}{\pi T}\right)^{3/2} \int\limits_0^\infty \sigma(E) E \exp\left\{-\frac{E}{T}\right\} dE \qquad (5.14)$$

Интеграл брался численно по квадратурам высокой точности с подстановкой фурье-аппроксимации (5.1).

Результаты расчетов представлены на рис. 5.7, *а* и в табл. 5.3. В них дана зависимость $\lg K(T)$ [см^{3}с$^{-1}$] от $\lg T$ [кэВ].

В [26] также содержались аппроксимации для $K(T)$. Отклонение R этих величин от наших данных в процентах приведено на рис. 5.7, *б*. Видно, что при малых температурах для реакций (2.17) и (2.20) оно не превышает 11%, для (2.19) составляет всего 5÷7%, а для (2.18) доходит до 20%. Такое уточнение существенно для физических приложений. Разумеется, при $T > 100$ кэВ формулы Козлова быстро теряют точность.

Таким образом, полученные данные позволяют повысить точность расчетов термоядерных мишеней, в которых нужны в первую очередь невысокие температуры. Поэтому табл. 5.3 также рекомендуется использовать в качестве справочных данных. Для определения погрешности $K(T)$ была применена процедура, аналогичная описанной в п. 5.3.3. Нормы C относительных физических погрешностей Δ_Φ составляют от 1 до 4.5 % и приведены в табл. 5.4.

Таблица 5.3. Скорости реакций $\lg K(T)$, $см^3с^{-1}$.

$\lg T$, кэВ	DD → pT	DD → n^{3}He	DT → n^{4}He	D^{3}He → p^{4}He	$\lg T$, кэВ	DD → pT	DD → n^{3}He	DT → n^{4}He	D^{3}He → p^{4}He
-2.0	-50.402	-50.402	-50.492	-74.292	0.7	-19.041	-19.023	-16.866	-20.175
-1.9	-47.667	-47.668	-47.582	-69.633	0.8	-18.749	-18.729	-16.525	-19.629
-1.8	-45.139	-45.140	-44.891	-65.324	0.9	-18.482	-18.460	-16.217	-19.120
-1.7	-42.803	-42.804	-42.405	-61.338	1.0	-18.237	-18.213	-15.947	-18.644
-1.6	-40.645	-40.645	-40.107	-57.651	1.1	-18.013	-17.987	-15.716	-18.199
-1.5	-38.650	-38.651	-37.984	-54.241	1.2	-17.807	-17.779	-15.523	-17.784
-1.4	-36.808	-36.809	-36.022	-51.089	1.3	-17.618	-17.587	-15.369	-17.401
-1.3	-35.107	-35.108	-34.210	-48.174	1.4	-17.444	-17.410	-15.249	-17.050
-1.2	-33.537	-33.538	-32.537	-45.479	1.5	-17.283	-17.246	-15.162	-16.736
-1.1	-32.087	-32.088	-30.993	-42.989	1.6	-17.134	-17.093	-15.103	-16.462
-1.0	-30.750	-30.750	-29.567	-40.687	1.7	-16.996	-16.952	-15.069	-16.228
-0.9	-29.517	-29.517	-28.252	-38.560	1.8	-16.867	-16.820	-15.057	-16.036
-0.8	-28.379	-28.379	-27.039	-36.596	1.9	-16.747	-16.697	-15.062	-15.881
-0.7	-27.331	-27.330	-25.920	-34.781	2.0	-16.635	-16.582	-15.082	-15.762
-0.6	-26.366	-26.365	-24.888	-33.105	2.1	-16.531	-16.476	-15.114	-15.674
-0.5	-25.476	-25.475	-23.936	-31.559	2.2	-16.435	-16.377	-15.156	-15.613
-0.4	-24.657	-24.657	-23.058	-30.131	2.3	-16.345	-16.286	-15.206	-15.575
-0.3	-23.904	-23.903	-22.248	-28.815	2.4	-16.263	-16.201	-15.261	-15.556
-0.2	-23.211	-23.210	-21.501	-27.602	2.5	-16.188	-16.124	-15.318	-15.552
-0.1	-22.573	-22.572	-20.812	-26.484	2.6	-16.118	-16.052	-15.378	-15.561
0.0	-21.988	-21.985	-20.177	-25.454	2.7	-16.054	-15.987	-15.436	-15.579
0.1	-21.450	-21.445	-19.590	-24.505	2.8	-15.995	-15.928	-15.492	-15.603
0.2	-20.957	-20.950	-19.047	-23.630	2.9	-15.940	-15.875	-15.543	-15.632
0.3	-20.504	-20.495	-18.544	-22.824	3.0	-15.888	-15.826	-15.589	-15.662
0.4	-20.089	-20.077	-18.077	-22.081	3.1	-15.841	-15.783	-15.629	-15.694
0.5	-19.708	-19.695	-17.643	-21.396	3.2	-15.796	-15.744	-15.663	-15.724
0.6	-19.360	-19.344	-17.240	-20.762	3.3	-15.755	-15.709	-15.692	-15.754

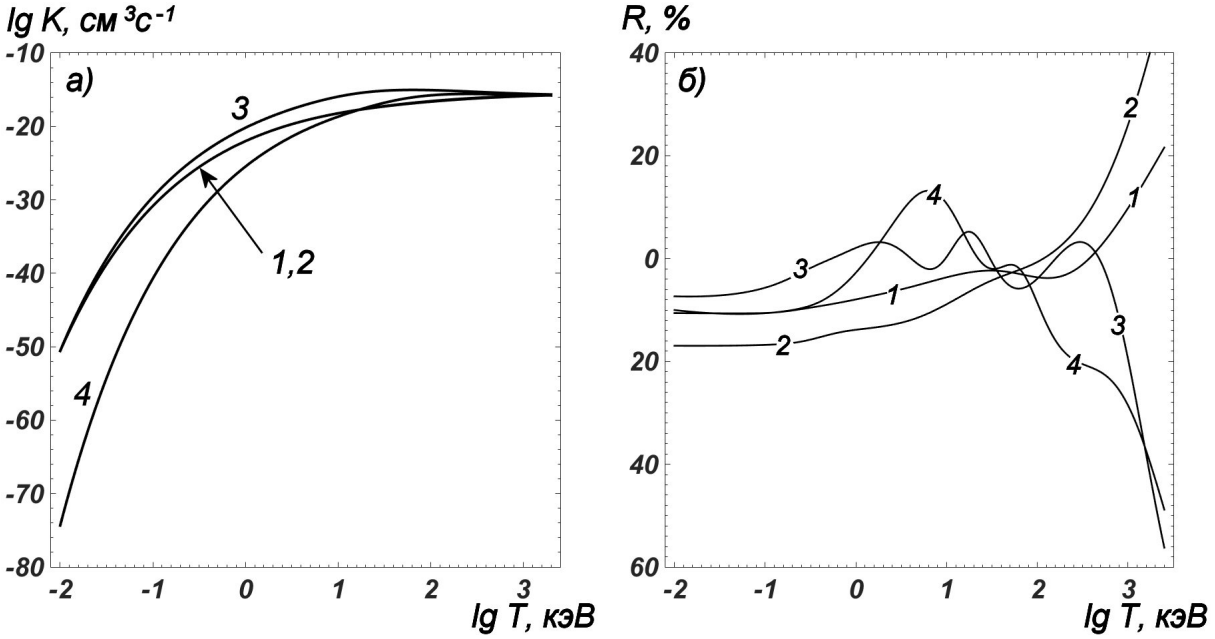

Рис. 5.7. а) Скорости реакций из табл. 5.3, б) отношение формул Козлова для $K(T)$ к данным табл. 5.3; на обоих рисунках обозначения соответствуют рис. 5.2.

5.4.2. Газодинамические приложения. Таблицы скоростей реакций неудобно использовать в газодинамических кодах. Желательно заменить их аппроксимирующими формулами с небольшим числом параметров. Мы построили такие формулы также методом двойного периода, но уже без регуляризации, поскольку теперь аппроксимируются не экспериментальные точки с большим разбросом, а гладкая математическая кривая. Здесь точность, равную точности самой табл. 5.3, удалось получить при $M = 5$ и $N = 3$.

Число гармоник основного периода оказалось почти таким же, как число гармоник двойного периода. Поэтому запись (5.1), где функции двойного периода ставятся после функций основного периода, не слишком удобна. Гораздо удобнее записать результат в виде отрезка ряда Фурье на $[-\pi, \pi]$. Тогда функции двойного периода становятся нечетными гармониками, а функции основного периода становятся четными гармониками. Окончательная аппроксимация в физических единицах без приведения аргумента к стандартному отрезку $[-\pi/2, \pi/2]$ имеет следующий вид:

$$\lg K(T) = \sum_{k=0}^{6} \left(\xi_k \cos kt + \eta_k \sin kt\right), \quad t = \frac{\pi}{5.30}\left(\lg T - 3.65\right), \quad 1.0 \leqslant \lg T \leqslant 6.3. \tag{5.15}$$

Здесь температура берется в единицах эВ. Для всех четырех реакций $\eta_0 = 0$, $\xi_5 = 0$; остальные коэффициенты приведены в табл. 5.4. Там же указаны относительные точности аппроксимаций Δ_{a}, % в норме C. Видно, что математическая точность аппроксимации примерно соответствует оценочной физической точности, так что числа M и N выбраны разумно. Числа обусловленности линейной системы для всех реакций равнялись $\varkappa = 4 \cdot 10^6$. Они довольно велики, но вполне приемлемы для 64-битовых вычислений.

Таблица 5.4. Коэффициенты аппроксимации $\lg K(T)$ рядом Фурье.

	DD → pT	DD → n^{3}He	DT → n^{4}He	D^{3}He → p^{4}He
ξ_0	-41.898	-42.145	-42.559	-60.504
ξ_1	30.017	30.484	33.312	53.039
η_1	24.313	24.459	24.394	43.244
ξ_2	-9.150	-9.451	-9.876	-16.233
η_2	-16.995	-17.170	-16.909	-31.497
ξ_3	2.140	2.265	2.501	3.980
η_3	7.572	7.744	7.406	15.707
ξ_4	-0.316	-0.346	-0.414	-0.749
η_4	-2.540	-2.655	-2.148	-6.055
η_5	0.580	0.630	0.411	1.734
ξ_6	0.012	0.015	0	0.004
η_6	-0.068	-0.079	0	-0.319
$\Delta_{\text{ф}}$, %	1	2.5	4.5	1.6
$\Delta_{\text{м}}$, %	1	0.5	2.5	0.9

Данные из табл. 5.4 можно рекомендовать для использования в прикладных расчетах. Поскольку метод двойного периода позволяет разумно экстраполировать за пределы основного периода, то формулой (5.15) можно пользоваться даже в пределах $0.5 \leqslant \lg T \leqslant 6.8$ (в единицах эВ).

5.5 Прецизионное вычисление квадратур

Для расчета таблиц $K(T)$ требовалось вычислять квадратуру (5.14) с высокой точностью. Поэтому вопросу прецизионного вычисления квадратур и исследованию их свойств было уделено особое внимание.

5.5.1. Квадратурные формулы. Пусть $u(x)$ – гладкая функция. Предполагается, что она имеет достаточное число ограниченных непрерывных производных. Рассмотрим равномерную сетку $\omega = \{x_0 + nh,\ n = 0, \ldots, N\}$. В простейших случаях для численного интегрирования применяется квадратурная формула трапеций $I_{\text{трап}}$, формула средних $I_{\text{ср}}$ и формула Симпсона $I_{\text{Симп}} = 2/3 I_{\text{ср}} + 1/3 I_{\text{трап}}$.

Если функция $u(x)$ дважды непрерывно дифференцируема, то формулы средних и трапеций имеют точность $O(h^2)$. Формула Симпсона имеет четвертый порядок точности при условии, что подынтегральная функция имеет четвертую непрерывную производную.

Известно, что порядок точности этих формул можно существенно повысить с помощью формул Эйлера-Маклорена (см., например, [15], [14]). Уточненная формула

трапеций с K членами имеет вид

$$\int_{x_0}^{x_N} u(x)dx \approx h\left(\frac{u_0}{2} + \sum_{n=1}^{N-1} u_n + \frac{u_N}{2}\right) + \sum_{k=1}^{K}(-1)^k a_k h^{2k}\left(u_N^{(2k-1)} - u_0^{(2k-1)}\right). \qquad (5.16)$$

Аналогично записывается уточненная формула средних

$$\int_{x_0}^{x_N} u(x)dx \approx h\sum_{n=1}^{N} u_{n-1/2} + \sum_{k=1}^{K}(-1)^{k+1} b_k h^{2k}\left(u_N^{(2k-1)} - u_0^{(2k-1)}\right). \qquad (5.17)$$

Наконец, можно определить коэффициенты Эйлера-Маклорена для формулы Симпсона

$$c_k = \frac{1}{3}a_k - \frac{2}{3}b_k, \qquad (5.18)$$

причем $c_1 = 0$. Подчеркнем, что формулы (5.16) – (5.18) можно строить только на равномерной сетке. Учет уже первого уточняющего члена в (5.16), (5.17) увеличивает точность квадратурных формул с $O(h^2)$ до $O(h^4)$ при условии непрерывности $u^{(4)}$.

Вопрос о вычислении поправок Эйлера-Маклорена получил подробное развитие в [57], где были найдены первые 6 коэффициентов формул (5.16), (5.17). В данной работе построены рекуррентные формулы для коэффициентов Эйлера-Маклорена формул (5.16) – (5.18) и вычислены в виде рациональных дробей первые 24 коэффициента. Из них первые 12 приведены ниже. Доказана положительность этих коэффициентов и получена строгая оценка скорости их убывания. Эвристически найдены асимптотики этих коэффициентов. Это позволило установить условия сходимости формул (5.16) – (5.18) при $K \to \infty$.

5.5.2. Рекуррентные формулы для коэффициентов. Для простоты будем рассматривать один интервал $[a, b]$, $b - a = h$. Обобщение на случай равномерной сетки является очевидным. Пусть функция $u(x)$ непрерывно дифференцируема $2K$ раз. Обозначим I_K – значение интеграла от $u(x)$ по $[a, b]$, вычисленное с учетом K поправочных членов. Выведем поправки Эйлера-Маклорена для формулы трапеций.

Разложим $u(x)$ в ряд Тейлора с центром в точке $\overline{x} = (a + b)/2$, учитывая $2K$ слагаемых, и подставим в (5.16). Определим поправку $R_K = I_K - I_{K-1}$, вносимую учетом $2K$-го слагаемого. Вычислим точно интегралы I_K^* и I_{K-1}^* от разложений $u(x)$, учитывающих $2K$ и $2K - 2$ членов соответственно, и определим поправку $R_K^* = I_K^* - I_{K-1}^*$. Сравнивая выражения R_K и R_K^*, выразим a_K через a_k, $(k = 1, \ldots, K-1)$:

$$a_K = \frac{(-1)^{K+1}2K}{2^{2K}(2K+1)!} + \sum_{k=1}^{K-1}\frac{(-1)^{K+k+1}}{(2K-2k+1)!}\frac{a_k}{2^{2K-2k}}. \qquad (5.19)$$

Аналогично можно получить поправки для формулы средних:

$$b_K = \frac{(-1)^{K+1}}{2^{2K}(2K+1)!} + \sum_{k=1}^{K-1}\frac{(-1)^{K+k+1}}{(2K-2k+1)!}\frac{b_k}{2^{2K-2k}}. \qquad (5.20)$$

Суммируя соотношения (5.19) и (5.20) с весами $1/3$ и $-2/3$, получим поправки к формуле Симпсона:

$$c_K = \frac{2}{3}\frac{(-1)^{K+1}(1-K)}{2^{2K}(2K+1)!} + \sum_{k=1}^{K-1}\frac{(-1)^{K+k+1}}{(2K-2k+1)!\,2^{2K-2k}}\frac{c_k}{}. \tag{5.21}$$

5.6 Свойства коэффициентов Эйлера-Маклорена

5.6.1. Положительность. Коэффициенты a_K, b_K, c_K оказываются положительными. Справедливы следующие теоремы

Теорема 1. *Все коэффициенты* $b_K > 0$. •

Доказательство проводится по индукции. Как известно, $b_1 = 1/24 > 0$. Пусть, далее, $b_k > 0$, $k = 1, \ldots, K-1$. Для упрощения выкладок введем величину $B_0 = 1$, $B_k = b_k$. Тогда

$$b_K = \sum_{k=1}^{K}\frac{(-1)^{k+1}}{2^{2k}(2k+1)!}B_{K-k} \geqslant \sum_{k=1}^{K}\frac{(-1)^{k+1}}{2^{2k}(2k+1)!}\min_{1\leqslant j\leqslant K}B_{K-j}. \tag{5.22}$$

Рассмотрим последнюю сумму по k. Очевидно, при $K = 1$ ее значение положительно. Пусть $K = 2$. Слагаемое, соответствующее $k = 2$, не превосходит слагаемого с $k = 1$, поэтому значение этой суммы снова положительно. Пусть значение этой суммы больше нуля при некотором K. Если это K четно, то при суммировании до $K+1$ добавляется положительное слагаемое, значит сумма будет положительна. Если K нечетно, то, поскольку $(K+1)$-й член по модулю не превосходит K-й, все эти члены можно сгруппировать по парам (1-й и 2-й, 3-й и 4-й и т.д.), причем сумма элементов в каждой паре больше нуля. Поэтому и сама сумма в этом случае будет также больше нуля. Итак, сумма по k всегда положительна. Поэтому $b_K > 0$. ∎

Теорема 2. *Все коэффициенты* $a_K > 0$. •

Она доказывается аналогично теореме 1.

Доказать аналогичную теорему для c_K не удалось. Поскольку $a_1 = 1/12$, $b_1 = 1/24$, то $c_1 = 0$ и не удовлетворяет такой теореме. Однако вычисление всех последующих коэффициентов показало, что $c_K > 0$, $K = 2, \ldots, 24$. Кроме того, из эвристической асимптотики коэффициентов, полученной далее, следует $c_K > 0$ при любых $K > 24$.

5.6.2. Скорость убывания коэффициентов. Скорость убывания коэффициентов в зависимости от номера можно оценить. Справедлива

Теорема 3. *Скорость убывания коэффициентов* b_K *подчинена закону*

$$\varepsilon_1 < \frac{b_K}{b_{K-1}} < \varepsilon_2, \tag{5.23}$$

где $\varepsilon_1 = -1/6 + 1/4\ln 2 \approx 0.0066$, $\varepsilon_2 = 1/24 + 1/4\ln\left(2/\sqrt{3}\right) \approx 0.0776$. •

Доказательство. Можно показать, что выражение для b_K/b_{K-1} приводится к виду

$$\frac{b_K}{b_{K-1}} = \frac{1}{2^2 3!} + \frac{1}{4}\left(\sum_{k=2}^{K} \alpha_k \frac{(-1)}{2k(2k+1)}\right) \Big/ \left(\sum_{k=2}^{K} \alpha_k\right), \tag{5.24}$$

где использовано обозначение $\alpha_k = (-1)^k B_{K-k}/(2^{2k-2}(2k-1)!)$, $B_0 = 1$, $B_k = b_k$. Далее, как нетрудно убедиться,

$$\frac{1}{2k(2k+1)} > \sum_{j=1}^{K} \frac{(-1)^j}{2j(2j+1)} \tag{5.25}$$

для любых k и K, так как значение суммы в правой части отрицательно (см. доказательство теоремы 1). Поэтому

$$\frac{b_K}{b_{K-1}} < \frac{1}{2^2 3!} + \frac{1}{4}\sum_{j=1}^{K} \frac{(-1)^{j+1}}{2j(2j+1)} < \frac{1}{2^2 3!} + \frac{1}{4}\sum_{j=1}^{\infty} \frac{2}{2j(2j+1)(2j+2)}. \tag{5.26}$$

Будем рассматривать последний ряд как квадратурную формулу средних для несобственного интеграла. Поскольку подынтегральная функция вогнута, то точное значение интеграла должно быть больше формулы средних. Этот интеграл берется точно: его значение равно $\ln\left(2/\sqrt{3}\right)$. Поэтому справедлива оценка сверху

$$\frac{b_K}{b_{K-1}} \leqslant \frac{1}{24} + \frac{1}{4}\ln\left(\frac{2}{\sqrt{3}}\right) \approx 0.0776. \tag{5.27}$$

Теперь оценим рассматриваемое соотношение снизу. В силу неравенства треугольника с учетом определения величин α_k и положительности b_k имеем

$$\begin{aligned}
\frac{b_K}{b_{K-1}} &> \frac{1}{2^2 3!} + \frac{1}{4}\left(\sum_{k=2}^{K} |\alpha_k| \frac{(-1)^{k+1}}{2k(2k+1)}\right) \Big/ \left(\sum_{k=2}^{K} |\alpha_k|\right) > \\
&> \frac{1}{2^2 3!} - \frac{1}{4}\sum_{k=2}^{K} |\alpha_k| \left(\sum_{j=2}^{K} \frac{1}{2j(2j+1)}\right) \Big/ \left(\sum_{k=2}^{K} |\alpha_k|\right) > \\
&> \frac{1}{2^2 3!} - \frac{1}{4}\sum_{j=2}^{\infty} \frac{1}{2j(2j+1)} = -\frac{1}{6} + \frac{1}{4}\ln 2 \approx 0.0066.
\end{aligned} \tag{5.28}$$

Здесь во втором неравенстве учтено (5.25). Итак, установлена оценка снизу, и утверждение теоремы доказано. ■

Теорема 4. *Скорость убывания коэффициентов a_K подчинена закону*

$$\varepsilon_1 < \frac{a_K}{a_{K-1}} < \varepsilon_2, \tag{5.29}$$

где $\varepsilon_1 = -1/6 + 1/4\ln 2 \approx 0.0066$, $\varepsilon_2 = 1/24 + 1/4\ln\left(2/\sqrt{3}\right) \approx 0.0776$. •

Доказательство аналогично случаю теоремы 3.

5.6.3. Вычисление коэффициентов. По рекуррентным соотношениям (5.19) – (5.21) в среде Maple 14 были вычислены 24 коэффициента Эйлера-Маклорена как отношения целых чисел. Старшие коэффициенты не вычислялись, поскольку целые числа переставали укладываться в сетку 64-разрядного компьютера. Первые 12 коэффициентов приведены в табл. 5.5. Остальные вычисленные коэффициенты не приводятся по двум причинам. Во-первых, они слишком громоздки, а во-вторых, для практических вычислений достаточно младших коэффициентов.

5.6.4. Эвристические соотношения. В [57] было предложено эвристическое соотношение между a_m и b_m:

$$\frac{b_m}{a_m} = 1 - 2^{1-2m}. \tag{5.30}$$

На всех 24 коэффициентах, вычисленных в данной работе, это соотношение точно выполняется (при вычислении с целыми числами). Поэтому его можно считать надежно установленным. Из (5.18) нетрудно получить соотношение между коэффициентами a_k и c_k:

$$c_k = \frac{1 - 2^{2-2k}}{3} a_k. \tag{5.31}$$

5.6.5. Асимптотические соотношения. Исследовались отношения коэффициентов с соседними номерами при увеличении номера K. Вычисления производились на 64-разрядном программном обеспечении, поэтому ошибка единичного округления составляет 10^{-16}. Было установлено, что каждое из них стремится к некоторой константе, которая с точностью 10^{-14} совпадает с $4\pi^2$. Такое совпадение не может быть случайным, поэтому можно утверждать, что $4\pi^2$ и есть предельное значение отношения соседних коэффициентов.

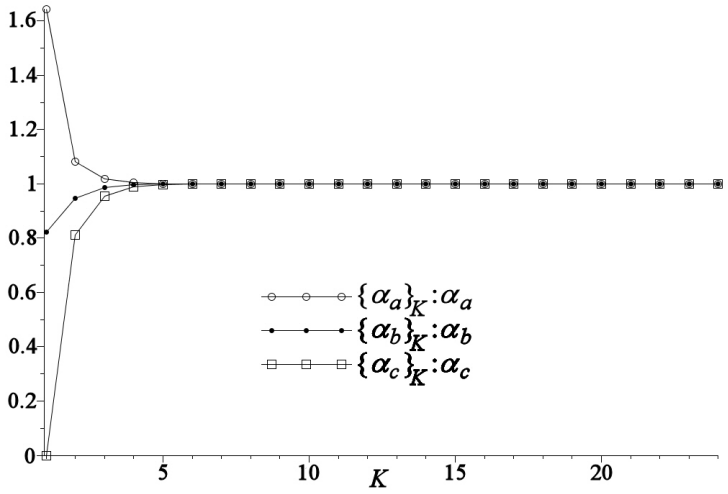

Рис. 5.8. Сходимость коэффициентов Эйлера-Маклорена к асимптотике.

Отсюда следует, что асимптотическое поведение этих отношений при больших номерах описывается степенным законом:

$$x_K (2\pi)^{2K} = \alpha_x, \;\; x = a, b, c, \tag{5.32}$$

Таблица 5.5. Значения коэффициентов Эйлера-Маклорена.

K	a_K	b_K	c_K
1	$\frac{1}{12}$	$\frac{1}{24}$	0
2	$\frac{1}{720}$	$\frac{7}{5760}$	$\frac{1}{2880}$
3	$\frac{1}{30240}$	$\frac{31}{967680}$	$\frac{1}{96768}$
4	$\frac{1}{1209600}$	$\frac{127}{154828800}$	$\frac{1}{3686400}$
5	$\frac{1}{47900160}$	$\frac{8179}{245248819200}$	$\frac{17}{2452488192}$
6	$\frac{691}{1307674368000}$	$\frac{351359}{334764638208000}$	$\frac{21421}{121732595712000}$
7	$\frac{1}{74724249600}$	$\frac{8191}{612141052723200}$	$\frac{1}{224227491840}$
8	$\frac{3617}{1067062284288 \times 10^4}$	$\frac{16931177}{4995070990221312 \times 10^4}$	$\frac{19752437}{17482748465774592 \times 10^4}$
9	$\frac{43867}{5109094217170944 \times 10^5}$	$\frac{5749691557}{669659197233029971968 \times 10^3}$	$\frac{19752437}{174827484657745920 \times 10^3}$
10	$\frac{174611}{8028576626982912 \times 10^5}$	$\frac{91546277357}{4209286382607616966656 \times 10^5}$	$\frac{12746603}{1758264988557901824 \times 10^5}$
11	$\frac{77683}{14101100039391805440000}$	$\frac{3324754717}{603513268363481705349120000}$	$\frac{98735093}{53767545726928370112921600}$
12	$\frac{236364091}{16938241367317436694528 \times 10^5}$	$\frac{1982765468311237}{14208826703980998799521113702 4 \times 10^5}$	$\frac{14367863999617}{30888753704306519129393725 44 \times 10^5}$

где α_x — некоторая постоянная, вообще говоря, своя для каждого x. Введем величину $\{\alpha_x\}_K = x_K(2\pi)^{2K}$ при каждом K. Полученные последовательности $\{\alpha_x\}_K$ сходятся с точностью 10^{-15}. При этом $\{\alpha_{a,b}\}_K \to \alpha_{a,b} = 2$ и $\{\alpha_c\}_K \to \alpha_c = 2/3$. Отложим величины $\{\alpha_x\}_K/\alpha_x,\ x = a, b, c.$ в зависимости от номера K на одном графике (см. рис. 5.8).

Видно, что значения $\{\alpha_x\}_K/\alpha_x$ быстро и монотонно стремятся к пределу и с графической точностью близки к предельному значению 1 уже при $K = 5$. Кроме того, последовательность $\{\alpha_b\}_K/\alpha_b$ быстрее остальных выходит на асимптотику. Закон стремления к пределу не исследовался, так как это не имеет практического значения.

Полученная асимптотика представляет теоретический интерес сама по себе, однако ее нельзя использовать для практических вычислений, поскольку для получения гарантированной точности квадратурной формулы необходимо использовать точные значения коэффициентов, а не приближения.

5.7 Повышение точности квадратурных формул

Найденная асимптотика означает ограничение на класс функций, к которым применим данный квадратурный метод при повышении точности квадратурных формул. Повышать точность квадратурных формул Эйлера-Маклорена и рассматривать сходимость можно в двух случаях: ***1°*** увеличивать число членов K при фиксированном шаге h и ***2°*** устремлять h к нулю при фиксированном K.

Сходимость по K***.*** Стремление $K \to \infty$ имеет смысл только в том случае, если $u(x)$ принадлежит классу функций, имеющих бесконечное количество непрерывных производных. Будем предполагать это условие выполненным. В этом случае легко сформулировать следующие необходимое и достаточное условия сходимости (они не совпадают между собой). Для сходимости необходимо, чтобы

$$\left(\frac{h}{2\pi}\right)^{2K} \left\|u^{(2K)}\right\|_C \to 0,\ K \to \infty. \tag{5.33}$$

Достаточным условием сходимости является убывание членов ряда быстрее, чем $1/K$, т.е.

$$\left(\frac{h}{2\pi}\right)^{2K} \left\|u^{(2K)}\right\|_C < \frac{const}{K^{1+\varepsilon}},\ \varepsilon > 0. \tag{5.34}$$

Доказательство этих утверждений тривиально. Коэффициенты a, b, c заменяются асимптотикой, поскольку речь идет о далеких членах.

Сходимость по h***.*** Пусть существует непрерывная ограниченная производная $u^{(2K+2)}$ (старшие производные разрывны или отсутствуют). Тогда при $h \to 0$ квадратурная формула сходится с точностью $O\left(h^{2K}\right)$.

Замечание. В заключение отметим, что для практического применения предпочтительнее квадратуры Гаусса-Кристоффеля. Последние имеют более высокую точность, поскольку их остаточные члены быстрее стремятся к нулю, чем остаточные

члены в формулах Эйлера-Маклорена. Например, для формулы Гаусса-Кристоффеля с единичным весом имеет место следующее выражение для верхней границы погрешности (см., например, [15], [14]):

$$\max|R| \approx \frac{b-a}{2.5\sqrt{n}} \left(\frac{b-a}{3n}\right)^{2n} \left\|u^{(2n)}\right\|_C, \tag{5.35}$$

где n — число узлов, в то время, как характер убывания остаточного члена в формулах Эйлера-Маклорена лишь степенной.

5.8 Основные результаты главы

1. Предложен новый метод обработки экспериментальных данных, измеренных со значительными погрешностями. Он заключается в построении аппроксимации по методу двойного периода, регуляризованному стабилизатором А. Н. Тихонова второго порядка (с квадратом второй производной). Это дает возможность подавить нефизичные осцилляции, используя высокие гармоники, и без труда обрабатывать кривые с участками, в которых экспериментальных данных мало или они вовсе отсутствуют. Такой подход позволяет единообразно решать широкий круг прикладных задач.
2. С использованием предложенного метода получены аппроксимации сечений для 4 реакций, наиболее важных для управляемого термоядерного синтеза. По ним рассчитаны скорости этих реакций K(T), которые превосходят по точности известные ранее формулы на 7÷20%. Это существенно для физических приложений.
3. С использованием полученных аппроксимаций для $K(T)$ и численного метода, предложенного в главе 2, проведены расчеты кинетики термоядерных реакций в газовых мишенях и токамаках. Приведены характерные оценки времени горения.
4. Исследован вопрос прецизионного вычисления квадратур. Получены рекуррентные формулы для коэффициентов Эйлера-Маклорена на основе квадратурных формул трапеций, средних и Симпсона. Исследованы свойства этих коэффициентов, и в частности, найдены их асимптотики при стремлении номера к бесконечности. Приведены выражения первых 12 коэффициентов в виде отношения целых чисел. Этого вполне достаточно для практических вычислений.

6. Пакеты программ

Все построенные численные методы реализованы в виде пакетов программ в среде Matlab. Это не самый эффективный язык с точки зрения быстродействия, и для решения прикладных задач данные пакеты невыгодны. Однако в них решены и отлажены все принципиальные алгоритмические вопросы. Кроме того, коды на Matlab легко переносятся на более эффективные языки программирования (в первую очередь Fortran и несколько сложнее C/C++). Поэтому предлагаемые пакеты могут служить хорошими прототипами для высокопроизводительных производственных программ (создание последних выходит за рамки данной работы).

6.1 Пакет Kinetic для расчета кинетики реакций

6.1.1. Описание программ.

Алгоритм. Пакет представляет собой реализацию химических схем, описанных в п. 2.1. В него входит 4 подпрограммы, соответствующие разным решателям. Из них пользователь может выбрать нужную. В пакет также входит подпрограмма, содержащая единую методику сгущения сеток и получения решения вместе с апостериорной оценкой погрешности.

Входные параметры. Для запуска пакета нужно вызвать функцию `Kinetic`. Ее входными аргументами являются вектор начальных условий `initial_cond`, промежуток интегрирования `Time` (соответствует переменной t), требуемая относительная точность `epsilon_user` и строка-название решателя. Все эти переменные объявлены глобальными.

В пакет включены одно- и двухстадийные химические схемы по времени (решатели `time_1` и `time_2` соответственно) и по длине дуги (`arc_length_1` и `arc_length_2`). Если последний аргумент не задан, то по умолчанию выбирается двухстадийная схема по времени. Правые части исходной системы ОДУ вводятся в функции `right_hand` аналогично тому, как это делается в стандартных решателях для ОДУ в Matlab.

Результаты расчетов. Функция `Kinetic` возвращает массив `u`, содержащий решение на последней сетке, и фактическую относительную погрешность `epsilon` этого решения, то есть оценку точности по Ричардсону, нормированную на сумму начальных условий.

После прекращения расчета вызывается процедура построения графиков. Программа выводит график решений $u(t)$ на последней сетке и график относительной

погрешности $\lg \varepsilon$ в зависимости от числа шагов $\lg N$ в двойном логарифмическом масштабе. В пакет не включено вычисление относительного дисбаланса, поскольку оно зависит от конкретной задачи.

Сгущения сеток. На первой сетке шаг выбирается равным `initial_step` (по умолчанию `Time`/10). Расчет ведется до тех пор, пока расчетное время не достигнет заданное `Time`. Это актуально для расчетов в аргументе l, поскольку полная длина дуги заранее неизвестна. Фиксируется номер последнего узла N_0. Далее шаг уменьшается вдвое, число шагов полагается равным $2N_0$, и расчет повторяется.

Такие сгущения производятся до тех пор, пока не будет достигнута заданная пользователем точность либо не будет превышено максимальное число сгущений, заданное переменной `max_grid_num` (по умолчанию 15). В первом случае расчет прерывается, во втором – переходит в интерактивный режим. Программа выводит в командном окне соответствующее предупреждение, текущее значение погрешности и предлагает альтернативу: продолжить расчет либо прервать.

В случае продолжения программа выполняет еще одно сгущение, выводит такое же предупреждение и т.д. При сгущениях программа записывает решения только на двух соседних сетках (текущей и предыдущей).

Настройка может понадобиться только при вычислениях в длине дуги. Единственным настроечным параметром является шаг на первой сетке `initial_step`. Если задача имеет очень высокую жесткость, а этот шаг оказался слишком грубым, то решение (и в частности, расчетное время) имеет большую погрешность.

Поэтому число шагов N_0, определяемое из условия $t_{(N_0)} <$ `Time`, может оказаться неправильным. При сгущении сеток полное расчетное время может “уползти” и оказаться как больше, так и меньше `Time`. Данную проблему можно решить, уменьшив величину `initial_step`. Эта величина определяется в функции `Kinetic`.

Вспомогательные процедуры. Опишем другие функции, используемые в этом пакете. В процедуре `iteration` реализована одна стадия химической схемы. Она принимает на вход текущие значения решения `u`, знакопостоянных членов правой части `psi` и `phi` и шага `step`. На выход выдается значение решения на следующем шаге `u_hat`.

В процедуре `chemical_scheme` реализован один шаг одно- и двухстадийных химических схем в аргументах t и l. Входными аргументами являются текущие значения решения `u` и времени `t` (напомним, что при расчетах в длине дуги время тоже является неизвестной функцией), а также шаг `step`. Выходными аргументами являются значения решения `u_hat` и времени `t_hat` на следующем шаге.

Процедура `chemical_solver` осуществляет решение системы ОДУ на сетке с шагом `step` по выбранному решателю. Вторым аргументом является число шагов `N`, которые необходимо выполнить. Если он опущен, то вычисления ведутся до достижения заданного времени `Time`. Выходными аргументами являются сеточные значения `u` и `t` на данной сетке.

Поточечная оценка погрешности по методу Ричардсона вычисляется в процедуре `richardson`. Она принимает решения `u1` (на грубой сетке) и `u2` (на подробной), а

также порядок точности `p`. Выходным аргументом является массив погрешностей `R0`, вычисленных в четных узлах подробной сетки. Наконец, построение графиков осуществляется процедурой `illustrations`.

6.1.2. Контрольный тест. С использованием данного пакета выполнены все расчеты в п. 2.1 и 2.2. Например, условия тестовой задачи (2.6) реализуются следующим образом. Необходимо создать файл с именем `right_hand.m` и ввести в нем правые части `phi` и `psi` по образцу:

Листинг 6.1. Тестовая задача (2.6), правая часть (`right_hand.m`)

```
function [psi,phi] = right_hand ( u )
a = pi;
lambda = 10;
psi = lambda*u*a^2;
phi = lambda*u^2;
end
```

Далее в командном окне или в отдельном файле необходимо задать входные аргументы и вызвать основную функцию:

Листинг 6.2. Тестовая задача (2.6), входные параметры и вызов функции

```
u0 = 0.5; % initial condition
int_span = 1; % integration interval
epsilon_user = 1e-6; % required accuracy
solver_name = 'arc_length_2'; % solver
[u, epsilon] = Kinetic(u0, int_span, epsilon_user, solver_name);
```

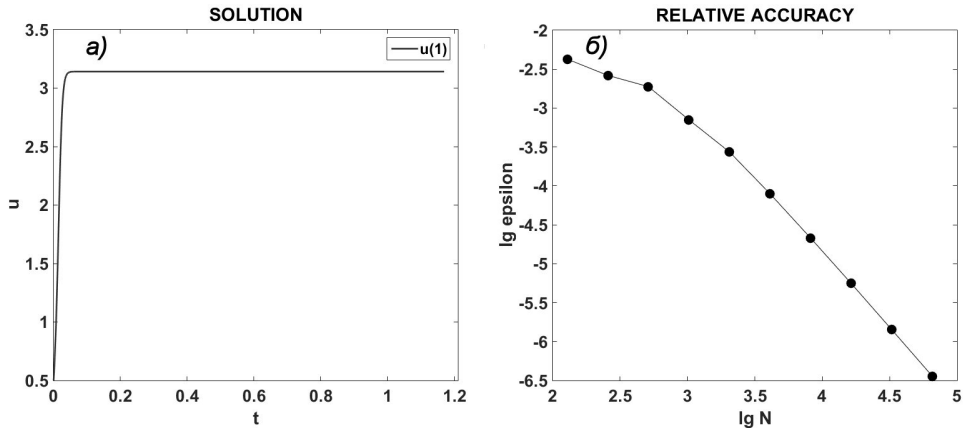

Рис. 6.1. Расчет задачи (2.6); а) решение, б) погрешность.

В этом расчете достигается точность `epsilon = 3.595e-07`. На рис. 6.1, *а* представлен график решения, а на рис. 6.1, *б* – график погрешности в зависимости от числа узлов в двойном логарифмическом масштабе.

6.1.3. Листинги. Ниже приводятся листинги функций, входящих в пакет `Kinetic`. Этот пакет распространяется по свободной лицензии BSD-3-clause и расположен по ссылке `https://bitbucket.org/alexander_belov/kinetic`.

Листинг 6.3. Основная функция `Kinetic.m`

```
function[u, epsilon] = Kinetic(initial_cond, Time, epsilon_user,
    solver_name)
% Solves kinetics problem with guaranteed accuracy. The user sets
% initial conditions, integration interval, required relative accuracy
% and the solver name (default: 2nd order scheme in "time" argument).
% After that, the calculation on a sequence of nested is carried out until
% relative accuracy determined by Richardson method is less than
% user-required accuracy. The adjusting parameter is the step of the
% most coarse mesh "initial_step".
global u0; global T; global solver;
u0 = initial_cond; T = Time; solver = solver_name; J = length(u0);
initial_step = T/10; % adjusting parameter
max_grid_num = 15; precision = zeros(1,max_grid_num);
if( nargin < 4 ); solver_name = 'time_2'; end
if(strcmp(solver,'time_1') == 1||strcmp(solver,'arc_length_1') == 1)
    p = 1;
elseif(strcmp(solver,'time_2') == 1||strcmp(solver,'arc_length_2') == 1)
    p = 2;
end
condition_accuracy = 1; m = 1;
while ( condition_accuracy )
    if ( m == 1)
        step = initial_step;
        [u,t] = chemical_solver( step );
        u1 = u;  t1 = t;
        dim = size(u); N = dim(2) - 1; N0 = N;
    else
        u1 = u2; t1 = t2;
    end
    m = m+1; N = 2*N; step = step/2;
    [u,t] = chemical_solver( step, N );
    u2 = u; t2 = t;
    precision_u = richardson( u1, u2, p );
    if( strcmp(solver_name,'arc_length_1') == 1 ||...
            strcmp(solver_name,'arc_length_2') == 1 )
        precision_t = richardson( t1, t2, p );
        for j = 1:J
            for n = 1:N/2+1
                [psi,phi] = right_hand( u(:,2*n-1) );
                precision_u(j,n) = precision_u(j,n) - ...
                    precision_t(n)*(psi(j) - u(j,2*n-1)*phi(j));
            end
        end
    end
    ic_sum = sum(u0); precision(m-1) = max(max(abs(precision_u)))/ic_sum;
    condition_mesh_num = m < max_grid_num;
    condition_accuracy = precision(m-1) > epsilon_user;
    if( condition_mesh_num == 0 )
        str_warn = 'Warning! Reached maximim mesh number.';
```

```
49          str_acc = strcat(' Current accuracy is: epsilon=' ,...
50              mat2str(precision(m-1)) );
51          disp( strcat(str_warn,str_acc) );
52          choice = input('Do you wish to proceed? 1 -> yes, 0 -> no: ');
53          if( choice == 0 ); break; end;
54      end
55  end
56  actual_grid_num = m; precision_output = zeros(1,actual_grid_num - 1);
57  for m = 1:actual_grid_num - 1
58      precision_output(m) = precision(m);
59  end
60  epsilon = precision_output(actual_grid_num-1);
61  illustrations(t, u, N0, precision_output);
62  end
```

Листинг 6.4. Решатель `chemical_solver.m`

```
1  function [u,t] = chemical_solver( step, N )
2  % Solves the ODE system with given step. If the number of steps is unknown
3  % (on the first mesh for example) the calculation is carried out until the
4  % given time is reached.
5  global u0; global T;
6  J = length(u0);
7  if( nargin < 2 )
8      u = zeros(J,1e+6); t = zeros(1,1e+6);
9      u(:,1) = u0;
10     n = 1;
11     while(t(n) < T)
12         [u(:,n+1),t(n+1)] = chemical_scheme( u(:,n), t(n), step );
13         n = n+1;
14     end
15     N1 = n;
16     u_truncated = zeros(J,N1); t_truncated = zeros(1,N1);
17     for n = 1:N1
18         u_truncated(:,n) = u(:,n); t_truncated( n ) = t(n);
19     end
20     u = u_truncated; t = t_truncated;
21 else
22     u = zeros(J,N+1); t = zeros(1,N+1);
23     u(:,1) = u0;
24     for n = 1:N
25         [u(:,n+1), t(n+1)] = chemical_scheme( u(:,n), t(n), step );
26     end
27 end
28 end
```

Листинг 6.5. Схема `chemical_scheme.m`

```
1  function [u_hat, t_hat] = chemical_scheme( u, t, step )
2  % One step of the chemical solver. Included are the chemical schemes with
3  % 1 and 2 iterations in "time" and "arc length" arguments.
```

```
4  global T; global u0; global solver;
5  U0 = max(u0); J = length(u); u_hat = zeros(J,1);
6  if ( strcmp(solver,'time_1') == 1 )
7      [psi,phi] = right_hand(u);
8      u_hat = iteration(u, psi, phi, step);
9      t_hat = t + step;
10 elseif ( strcmp(solver,'time_2') == 1 )
11     [psi,phi] = right_hand(u);
12     u_hat = iteration(u, psi, phi, step);
13     u_half = (u + u_hat)/2;
14     [psi,phi] = right_hand(u_half);
15     u_hat = iteration(u, psi, phi, step);
16     t_hat = t + step;
17 elseif ( strcmp(solver,'arc_length_1') == 1 )
18     [psi,phi] = right_hand(u);
19     f = (psi - u.*phi).^2;
20     S = sqrt(1/T^2 + sum(f)/U0^2);
21     psi = psi*T/S; phi = phi*T/S;
22     u_hat = iteration(u, psi, phi, step);
23     phi_t = 1/S; psi_t = 0;
24     t_hat = iteration( t, phi_t, psi_t, step );
25 elseif ( strcmp(solver,'arc_length_2') == 1 )
26     [psi,phi] = right_hand(u);
27     f = (psi - u.*phi).^2;
28     S = sqrt(1/T^2 + sum(f)/U0^2);
29     psi = psi*T/S; phi = phi*T/S;
30     u_hat = iteration( u, psi, phi, step );
31     u_half = (u + u_hat)/2;
32     [psi,phi] = right_hand( u_half );
33     f = (psi - u.*phi).^2;
34     S = sqrt( 1/T^2 + sum(f)/U0^2 );
35     psi = psi*T/S; phi = phi*T/S;
36     u_hat = iteration( u, psi, phi, step );
37     phi_t = 1/S; psi_t = 0;
38     t_hat = iteration( t, phi_t, psi_t, step );
39 end
40 end
```

Листинг 6.6. Стадия схемы `iteration.m`

```
1  function u_hat = iteration( u, psi, phi, step )
2  % Performs one chemical scheme iteration for one mesh step.
3  J = length(u); u_hat = zeros(J,1);
4  for j = 1:J
5      u_hat(j) = ( u(j) + step*psi(j)*(1 + step*phi(j)/2) )/...
6          (1 + step*phi(j) + step^2*phi(j)^2/2 );
7  end
8  end
```

Листинг 6.7. Поточечная оценка погрешности `richardson.m`

```
function R0 = richardson( u1, u2, p )
% Calculates point-wise estimation according to Richardson method.
dim = size(u1); R0 = zeros(dim(1),dim(2));
for j = 1:dim(1)
    for n = 1:dim(2)
        R0(j,n) = ( u2( j,2*n-1 ) - u1( j,n ) )/( 2^p-1 );
    end
end
end
```

Листинг 6.8. Иллюстрации `illustrations.m`

```
function illustrations( t, u, N0, precision_output )
% Auxiliary graphics: solution obtained on the thickest mesh and
% accuracy curve
dim = size(u); J = dim(1); N = dim(2);
figure; xlabel('t'); ylabel('u'); title('SOLUTION'); hold on;
for j = 1:J
    c = j/5 - floor(j/5);
    plot(t,u(j,:),'Color',[c,c,c],'LineWidth',2);
end
str = 'u(1)';
for j = 1:J-1
    str0 = strcat('u(',mat2str(j+1),')'); str = char(str,str0);
end
if( J<=10 ); legend(str); end
figure; xlabel('lg N'); ylabel('lg epsilon'); title('RELATIVE ACCURACY');
hold on
NN = zeros(1,length(precision_output));
for m = 1:length(precision_output)
    NN(m) = N0*2^(m);
end
plot(log10(NN),log10(precision_output),...
    '-ok','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',11)
end
```

6.2 Пакет SiDiaG для диагностики сингулярностей систем ОДУ

6.2.1. Описание программ.

Алгоритм. В пакете `SiDiaG` (SIngularity DIAGnostics) реализована автоматическая диагностика сингулярностей решений дифференциальных уравнений на основе метода, описанного в п. 2.3. Программа находит численное решение и параметры его сингулярности с апостериорной асимптотически точной оценки погрешности. В пакет входят две подпрограммы для диагностики степенного и логарифмического полюсов, а также подпрограмма оценка погрешности как решения, так и характеристик сингулярности. Ранее подобное математическое обеспечение не предлагалось.

Входные параметры. Для запуска пакета необходимо вызвать основную функцию `SiDiaG`. Ее аргументами являются следующие величины: вектор начальных условий `initial_cond`, ожидаемый промежуток интегрирования `Time` (соответствует переменной t) и требуемая точность `epsilon_user`. Правые части исходной системы ОДУ записываются в функции `right_hand`.

Результаты расчета. По окончании расчета функция `SiDiaG` возвращает следующие переменные:

1. массив `u`, содержащий решение для всех компонент на последней сетке;
2. вектор `type`, в который записываются обозначения для типа сингулярности каждой компоненты: 1 – степенной полюс, 2 – логарифмический полюс и 0 – если данная компонента не имеет особенности;
3. вектора `q` и `t0`, содержащие порядок полюса и момент разрушения для каждой компоненты (если данная компонента не разрушается, то `q = 0`, `t0 = 0`);
4. вектор `epsilons` фактических погрешностей решения u и параметров сингулярности q, t_0 (берется максимальная из погрешностей всех компонент, погрешность u понимается в смысле нормы C).

Кроме того, программа выводит график всех компонент решения на последней сетке, график профилей q и t_0 в зависимости от текущей длины дуги и график сходимости величин u, q и t_0 при сгущении сетки по длине дуги.

Сгущение сеток. Расчеты ведутся на сгущающихся сетках в длине дуги. Система ОДУ решается по схеме CROS, которая относится к чрезвычайно надежным. Рассматриваются сингулярности типа степенной полюс и логарифмический полюс, поскольку на практике они встречаются наиболее часто.

На каждом шаге первой сетки для всех компонент вычисляются значения q и t_0 по каждой из гипотез. По паре соседних шагов вычисляются наклоны профилей $q(l)$ и $t(l)$ (точнее, разностные производные этих величин). Если для некоторой гипотезы оба эти наклона не превышают заданную величину `slope`, то тип сингулярности считается установленным и в вектор `type` записывается его номер. По умолчанию значение `slope` равно 10^{-3}; это близко к визуальному критерию, но чуть жестче.

Расчет на первой сетке прекращается, если ***1°*** достигнуто заданное время `Time` (и разрушения не обнаружены), либо ***2°*** расчет подошел вплотную к первой из сингулярностей, то есть достигнут момент времени $t = \alpha \min_j t_0^{(j)}$, $\alpha < 1$ (по умолчанию 0.85).

На последующих сетках расчет ведется до той же максимальной длины дуги, что и на первой сетке. Для каждой компоненты вычисляются q и t_0 только по той гипотезе, которая установлена на первой сетке. Сетки сгущаются до тех пор, пока погрешности решения, всех q и t_0 не станут меньше заданного ε_{user}. В качестве ответа для q и t_0 берутся их последние значения на данной сетке (а не весь профиль). Погрешности этих величин вычисляются по обычным формулам метода Ричардсона, а для решения u строится приведенная оценка погрешности.

Трудоемкость. Применение диагностических формул (2.33) и (2.38) – (2.39) одновременно с вычислением решения не сильно увеличивает общую трудоемкость рас-

чета. Для степенного полюса значения q и t_0 вычисляются по явным формулам, что соответствует трудоемкости явных схем.

В случае логарифмического полюса для каждой компоненты нужно решать одно скалярное нелинейное уравнение и применять одну явную формулу. Нелинейное уравнение решается методом Ньютона. В качестве начального приближения выбирается значение корня с предыдущего шага либо текущий шаг по длине дуги. Его целесообразно подстраивать под полное ожидаемое время расчета, по умолчанию на первой сетке он равен $h = \texttt{Time}/10$. При таком выборе метод Ньютона сходится за 4-6 итераций. Таким образом, итоговая трудоемкость лишь в 2-3 раза больше трудоемкости явных схем и по-прежнему остается меньше трудоемкости явно-неявной схемы CROS.

Автономизация. Записи схем для автономной и неавтономной задач различаются. Например, неавтономный вариант схемы CROS, обеспечивающий второй порядок точности, имеет вид

$$(E - 0.5(1+i)\tau f_u)\, w = f(u, t + 0.5\tau), \quad \hat{u} = u + \tau \operatorname{Re} w. \tag{6.1}$$

Иными словами, правая часть берется в полуцелый момент времени, иначе порядок точности ухудшается до первого. Аналогичная ситуация имеет место для диагностических формул (2.33) и (2.38) – (2.39). В таком виде они справедливы для автономных систем. Можно было бы вывести "неавтономные" формулы нужного порядка точности, но они были бы применимы только для той схемы, для которой они получены.

Для того, чтобы единообразно рассчитывать автономные и неавтономные задачи с произвольным решателем, в пакете реализован следующий подход. Сначала проводится тривиальная автономизация, то есть время t объявляется новой $J+1$-й неизвестной функцией, которой соответствует $f_{J+1} \equiv 1$. Далее полученная система рассчитывается в длине дуги с добавлением еще одной, $J+2$-ой компоненты и к ней применяются диагностические формулы (2.33) и (2.38) – (2.39). Поскольку переход к длине дуги также является способом автономизации, описанный подход назовем ***двойной автономизацией***.

На первый взгляд может показаться, что его трудоемкость избыточна, поскольку возникают две компоненты времени. Однако, во-первых, увеличение трудоемкости несущественно. Если система имела небольшой порядок, то и сама трудоемкость невелика. Если порядок исходной системы большой, то добавление двух дополнительных компонент влияет слабо. Во-вторых, этот подход работает надежно и универсально. Представительные демонстрационные примеры, приведенные ниже, рассчитаны с одним и теми же значениями настроечных параметров.

Настройка. Программа настроена, то есть для всех вспомогательных величин указаны рекомендованные значения по умолчанию. Однако в ней предусмотрена возможность "тонкой" настройки для того, чтобы оптимизировать расчет конкретной задачи. Все настроечные параметры определяются в основной функции `SiDiaG`. Перечислим их и дадим практические рекомендации к их подбору.

1. Критерий `slope` установления типа сингулярности. Чем меньше это значение,

тем ближе нужно подходить к точке сингулярности, чтобы установить ее тип. Однако при этом заметно возрастает жесткость задачи. Уменьшение этого параметра может быть актуально для сильно "зашумленных" сингулярностей, когда полюс или логарифм умножаются на сильно меняющуюся функцию. Напомним, что по умолчанию `slope` $= 10^{-3}$.

2. Шаг h_0 самой грубой сетки по длине дуги. Первая сетка должна быть достаточно подробной, так как именно на ней определяются типы сингулярностей. В то же время шаг не следует выбирать слишком мелким: если сингулярность расположена далеко от начального момента, то до ее достижения придется делать слишком много шагов. По умолчанию $h_0 =$ `Time`$/10$.
3. Проверяемые типы сингулярностей, перечисленные в массиве `hypothesis_list`. По умолчанию проверяются оба типа полюсов – степенной и логарифмический, то есть `hypothesis_list = [1,2]`. Однако если есть теоретические соображения в пользу одного из них, то целесообразно проверять только его. Например, если требуется проверить только степенной полюс, то следует задать `hypothesis_list = 1`.
4. Момент прекращения расчета при достижении первой из сингулярностей. Расчет на первой сетке обрывается, если достигнут момент времени $t = \alpha \min_j t_0^{(j)}$. Значение $0 < \alpha < 1$, определенное в переменной `stop`, является настроечным параметром. Чем оно больше, тем ближе мы подходим к первой сингулярности. Жесткость задачи при этом ощутимо возрастает.

   Если сингулярность сильно "зашумленная", то увеличение этого параметра позволяет точнее определить ее характеристики. Если мы подходим к точке разрушения недостаточно близко, то, начиная с некоторого момента, наклон кривых погрешности q и t_0 может уменьшаться вплоть до горизонтального. Это значит, что при текущем значении `stop` вычислить q и t_0 точнее не удается. Напомним, по умолчанию значение `stop` $= 0.85$.

Вспомогательные процедуры. Опишем работу других функций, входящих в данный пакет. Функция `Solver` реализует решение системы ОДУ и анализ разрушений на фиксированной сетке. Обязательным аргументом этой функции является шаг сетки `step`. Она также может принимать два необязательных аргумента – число шагов `N` и вектор `B`, содержащий типы сингулярностей всех компонент.

Если необязательные аргументы не заданы, то это соответствует расчету на первой сетке, а если известны, то это вторая и последующие сетки. На выходе эта функция возвращает решение `u`, вектора параметров сингулярностей `q` и `t0` и вектор типов сингулярности `type`.

Процедура `diagnostics` осуществляет перебор гипотез о типе сингулярности и вычисляет ее параметры для всех компонент. Входными аргументами являются

1. текущий шаг сетки `step`,
2. вектор типов сингулярности `type`, которые необходимо проверить,
3. решение `u` на двух соседних шагах
4. вектор `initial_approx`, в который записываются начальные приближения для

решения нелинейного уравнения относительно t_0 для каждой компоненты. Возвращаются векторы параметров `q`, `t0` и текущий корень `current_root` уравнения относительно t_0.

Тривиальная автономизация делается функцией `right_hand_autonomization`, а переход к длине дуги – функцией `right_hand_arc_length`. Функция `CROS` осуществляет один шаг комплексной схемы Розенброка. Она принимает на вход величину шага `step` и текущее значение решения `u`, а возвращает решение `u_hat` на новом шаге. Матрица Якоби вычисляется в процедуре `Jacobi_matrix`. Метод Ричардсона реализован в функции `richardson`, которая полностью аналогична соответствующей процедуре из пакета `Kinetic`. Наконец, построение графиков осуществляется функцией `illustrations`.

6.2.2. Контрольные тесты.

Неавтономная задача. Рассмотрим пример неавтономной задачи с "зашумленной" сингулярностью. Простейший вариант задачи со степенным полюсом имеет вид

$$\frac{du}{dt} = u\frac{\varphi'(t)}{\varphi(t)} + \frac{qu}{(t_0 - t)},\ u(0) = \frac{\varphi(0)}{t_0^q} \quad \Rightarrow \quad u = \frac{\varphi(t)}{(t_0 - t)^q}. \tag{6.2}$$

Аналогично для логарифмического полюса

$$\frac{du}{dt} = u\frac{\varphi'(t)}{\varphi(t)} + q\varphi u^{1-1/q}\exp\{u^{1/q}\},\ u(0) = \varphi(0)[-\ln t_0]^q \quad \Rightarrow \quad u = \varphi(t)[-\ln(t_0 - t)]^q. \tag{6.3}$$

Рассмотрим систему (6.2), (6.3). Положим $q = 2.5$ и $t_0 = 0.3$. В качестве "зашумляющей" функции возьмем $\varphi = 2 + \cos 6\,t$. Видно, что на отрезке существования решения она меняется в 3 раза.

Для того, чтобы задать в программе эти правые части, необходимо создать файл `right_hand.m` и набрать в нем следующий код:

Листинг 6.9. Неавтономная задача, правые части (`right_hand.m`)

```
function f = right_hand ( u,t )
q = 2.5; t0 = 0.3;
f(1) = ( -sin(6*pi*t)/( 2 + cos(6*pi*t) ) + q/(t0 - t) )*u(1);
f(2) = -sin(6*pi*t)/(2 + cos(6*pi*t))*u(2) ...
    + q*( u(2)^(1-1/q) )*exp( u(2)^(1/q) );
f = f';
end
```

Далее в командном окне или в отдельном файле необходимо задать входные параметры и вызвать основную функцию:

Листинг 6.10. Неавтономная задача, входные параметры

```
initial_cond = [1;1]; % initial conditions
Time = 1; % expected time interval
epsilon_user = 1e-5; % required accuracy
[u, type, q, t0, epsilons] = SiDiaG( initial_cond, Time, epsilon_user );
```

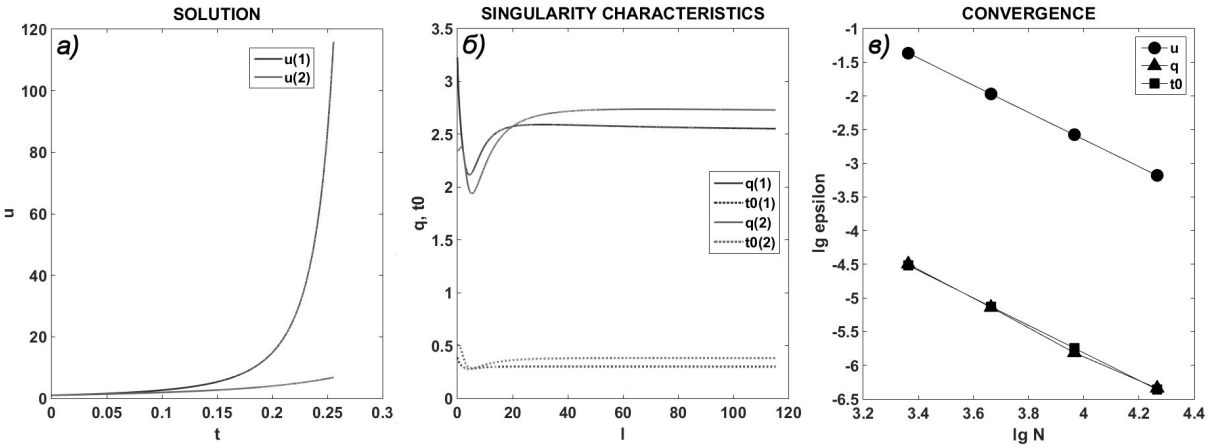

Рис. 6.2. Расчет системы (6.2), (6.3); а) решения, б) профили q и t_0, в) погрешности u, q, t_0.

Завершив расчет, программа выводит вектора `type = [1,2]`, `q = [2.553,2.729]`, `t0 = [0.301,0.381]`. Значения погрешностей равны `epsilons = [6.557e-4,4.531e-7, 4.368e-7]`. На рис. 6.2, *а* показан график решения этой задачи. На рис. 6.2, *б* представлены профили q и t_0 для каждой компоненты в зависимости от длины дуги. Наконец, на рис. 6.2, *в* показаны погрешности u, q и t_0 в зависимости от числа шагов N в двойном логарифмическом масштабе.

Смешанная особенность. В п. 2.3.3 был приведен пример задачи со смешанной особенностью. Напомним его:

$$\frac{du}{dt} = \left[-\frac{u}{\ln(t_0 - t)}\right]^{1+1/q} + \frac{qu}{t_0 - t}. \tag{6.4}$$

Положим $t_0 = 0.5$ и $q = 2$. Реализация этой правой части в файле `right_hand.m` выглядит следующим образом:

Листинг 6.11. Смешанная особенность, правая часть (`right_hand.m`)

```
function f = right_hand ( u,t )
t0 = 0.5;   q = 2;
f = ( -u/log(t0 - t) )^( (q+1)/q ) + q*u/(t0-t);
end
```

Далее в командном окне или отдельном файле зададим входные параметры и вызовем основную функцию:

Листинг 6.12. Смешанная особенность, входные параметры

```
initial_cond = 1; % initial condition
Time = 1; % expected time interval
epsilon_user = 1e-2; % required accuracy
[u, type, q, t0, epsilons] = SiDiaG( initial_cond, Time, epsilon_user );
```

Программа выдает значения `type = 1`, `q = 2.250`, `t0 = 0.503`. Достигнутые погрешности равны `epsilons = [2.505e-4,8.986e-7,2.245e-8]`. Общий вид решения показан на рис. 6.3, *а*. На рис. 6.3, *б* приведены профили параметров сингулярности. Погрешности u, q и t_0 даны на рис. 6.3, *в*.

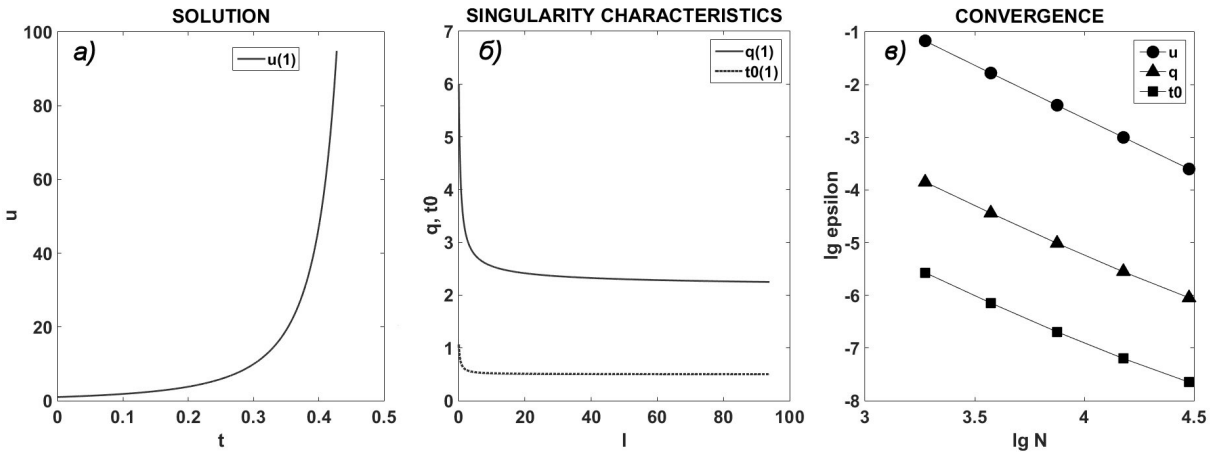

Рис. 6.3. Расчет задачи (6.4); а) решения, б) профили q и t_0, в) погрешности u, q, t_0.

Уравнение в частных производных. В качестве примера рассмотрим S-режим нелинейного горения, описанный в п. 2.4. Напомним, что методом прямых уравнение в частных производных сводится к системе ОДУ с кубическими правыми частями. Пусть сетка по пространству имеет 100 шагов, то есть система ОДУ содержит $J = 101$ уравнение. Ее реализация в файле `right_hand.m` имеет следующий вид:

Листинг 6.13. S-режим, правая часть (`right_hand.m`)

```
function f = right_hand ( u )
J = 100+1; hx = pi*sqrt(3)/(J-1);
f = zeros(J,1); f(1) = u(1)^3; f(J) = u(J)^3;
for j = 2:J-1
    f(j) = 0.5*( u(j+1)^2 + u(j)^2 )*( u(j+1) - u(j) )/hx^2 ...
         - 0.5*( u(j)^2 + u(j-1)^2 )*( u(j) - u(j-1) )/hx^2 + u(j)^3;
end
end
```

Пусть начальные условия u^0 соответствуют точному решению (2.54) исходного уравнения. Зададим их и другие входные параметры и вызовем основную функцию, исполняя в командном окне или отдельном файле следующий код:

Листинг 6.14. S-режим, входные параметры

```
% initial conditions
t0 = 1; J = 100+1; LS = pi*sqrt(3); X = zeros(J,1);
for j = 1:J
    X(j) = -LS/2 + LS*(j-1)/(J-1);
end
initial_cond = zeros(J,1);
for j = 1:J
    initial_cond(j) = 0.5*sqrt(3)/sqrt(t0)*cos( pi*X(j)/LS );
end
Time = 1; % expected time interval
epsilon_user = 1e-5; % required accuracy
[u, type, q, t0, epsilons] = SiDiaG( initial_cond, Time, epsilon_user );
```

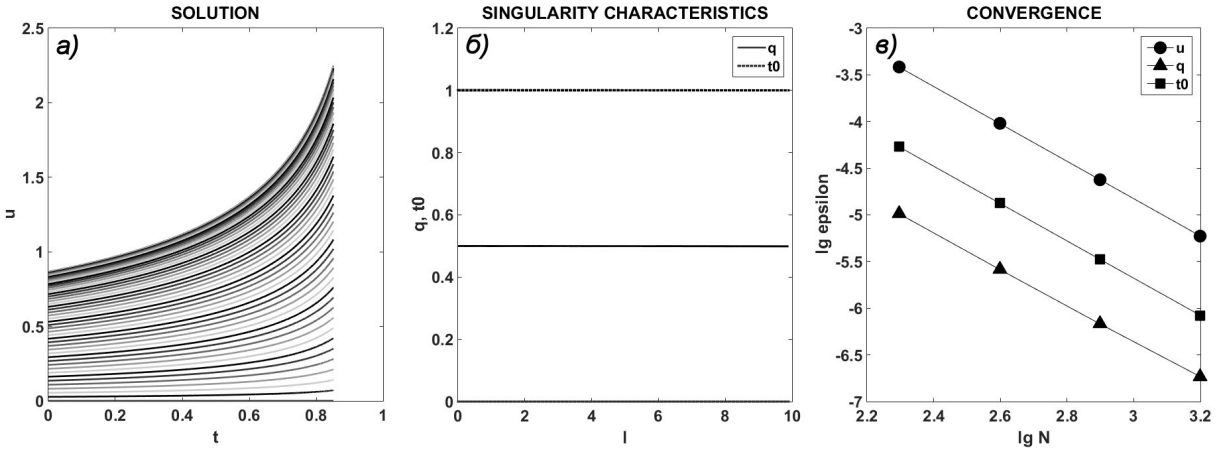

Рис. 6.4. Расчет S-режима горения; а) решения, б) профили q и t_0, в) погрешности u, q, t_0.

В результате расчета программа выдает вектора `type`, `q` и `t0`, длина которых равна 101 элементу. Первый и последний элементы этих векторов нулевые, а остальные отличны от нуля: `type = [0,1,1,...1,0]`, `q = [0,0.499,...,0.5,...,0.499,0]`, `t0 = = [0,1,...,0.999,...1,0]`. Достигнутые погрешности равны `epsilons = [5.940e-6, 1.848e-7,8.387e-7]`. Профили решений показаны на рис. 6.4, *а*. На рис. 6.4, *б* изображены профили q и t_0. Наконец, на рис. 6.4, *в* показаны погрешности u, q, t_0.

6.2.3. Листинги. Ниже приводятся листинги функций, входящих в пакет `SiDiaG`. Этот пакет распространяется по свободной лицензии BSD-3-clause и расположен по ссылке `https://bitbucket.org/alexander_belov/sidiag`.

Листинг 6.15. Основная функция `SiDiaG.m`

```
function [u, type, q_output, t0_output, epsilon_output] =...
    SiDiaG( u0, Time, epsilon_user )
% Solves given ODE system and analyses its singularities. Solution and
% singularity characteristics are calculated with guaranteed accuracy. Mesh
% is thickened until obtained accuracy is less than user-defined error.
global T; global eps; global U0;
global slope; global hypothesis_list; global stop;
T = Time; eps = epsilon_user; U0 = u0;
slope = 1e-3; %adjusting parameters
step  = T/10;
hypothesis_list = [1,2];
stop = 0.85;
max_grid_num = 10;
precision_u  = zeros(1,max_grid_num);
precision_q  = zeros(1,max_grid_num);
precision_t0 = zeros(1,max_grid_num);
check = 1; m = 1;
while ( check )
    if (m > max_grid_num); break; end
    if ( m == 1)
        [u, q, t0, type] = Solver( step );
```

```
22          dim = size(u); N0 = dim(2) - 1; N = N0;
23          u1 = u; q1 = q(:,N); t01 = t0(:,N);
24      else
25          u1 = u2; q1 = q2; t01 = t02;
26      end
27      m = m+1; step = step/2; N = N*2;
28      [u, q, t0] = Solver( step, N, type );
29      u2 = u; q2 = q(:,N); t02 = t0(:,N);
30      p = 2;
31      precision_pointwise_u  = richardson( u1,  u2,  p );
32      precision_pointwise_q  = richardson( q1,  q2,  p );
33      precision_pointwise_t0 = richardson( t01, t02, p );
34      J = length(u0); u_temp = zeros(J,N+1);
35      for j = 1:J
36          u_temp(j,:) = u(j,:);
37      end
38      t = u(J+1,:);
39      for n = 1:N/2+1
40          f = right_hand( u_temp(:,2*n-1), t(2*n-1) );
41          for j = 1:J
42              precision_pointwise_u(j,n) = precision_pointwise_u(j,n)...
43                                         - precision_pointwise_u(J+1,n)*f(j);
44          end
45      end
46      precision_u(m-1)  = max(max(abs(precision_pointwise_u)));
47      precision_q(m-1)  = max(max(abs(precision_pointwise_q)));
48      precision_t0(m-1) = max(max(abs(precision_pointwise_t0)));
49      check = precision_u(m-1)  > epsilon_user...
50            || precision_q(m-1)  > epsilon_user...
51            || precision_t0(m-1) > epsilon_user;
52  end
53  M = m-1; precision_output = zeros(M,3);
54  for m = 1:M
55      precision_output(m,1) = precision_u(m);
56      precision_output(m,2) = precision_q(m);
57      precision_output(m,3) = precision_t0(m);
58  end
59  epsilon_output = precision_output(M,:);
60  q_output  =  q(:,end); t0_output = t0(:,end);
61  illustrations( step, u, q, t0, precision_output );
62  end
```

Листинг 6.16. Решатель `Solver.m`

```
1  function [u, q, t0, type] = Solver( step, N, B )
2  % Solves ODE system and analyses its singularities on a mesh with given
3  % step
4  global T; global U0;
5  global slope; global hypothesis_list; global stop;
6  if( nargin < 2 ); N = 1e+6; end
7  J = length(U0)+1; u = zeros(J+1,N+1);
```

```
for j = 1:J-1
    u(j,1) = U0(j);
end
u_temp = zeros(J+1,2);
H = length(hypothesis_list);
q_temp  = zeros(2,J-1,N); q  = zeros(J-1,N);
t0_temp = zeros(2,J-1,N); t0 = zeros(J-1,N);
type = zeros(J-1,1); r = step*ones(J-1,1);
if( nargin == 3 )
    for n = 1:N
        u(:,n+1) = CROS(step, u(:,n));
        u_temp(:,1) = u(:,n); u_temp(:,2) = u(:,n+1);
        [q(:,n), t0(:,n), newton_root] = diagnostics(step, B, u_temp, r);
        r = newton_root;
    end
else
    n = 1; condition = 1;
    while( condition )
        u(:,n+1) = CROS( step, u(:,n) );
        u_temp(:,1) = u(:,n); u_temp(:,2) = u(:,n+1);
        for h = 1:H
            k = hypothesis_list(h); test_type = k*ones(J-1,1);
            [q_temp(k,:,n), t0_temp(k,:,n), newton_root] = ...
                diagnostics(step, test_type, u_temp, r);
            r = newton_root;
            if( n > 1 )
                for j = 1:J-1
                    q_slope  = abs( q_temp(k,j,n) -  q_temp(k,j,n-1))/step;
                    t0_slope = abs(t0_temp(k,j,n) - t0_temp(k,j,n-1))/step;
                    check = q_slope < slope && t0_slope < slope && ...
                            abs(t0_temp(k,j,n)) > u(J,2) - u(J,1);
                    if( check == 1 ); type(j) = k; end
                end
            end
        end
        for j = 1:J-1
            if( type(j) ~= 0 )
                q(j,n)=q_temp(type(j),j,n); t0(j,n)=t0_temp(type(j),j,n);
            end
        end
        t_min = T/stop;
        for j = 1:J-1
            if( abs(t0(j,n)) ~= 0 && abs(t0(j,n)) < t_min )
                t_min = abs(t0(j,n));
            end
        end
        condition = u(J,n+1) < stop*t_min;
        n = n+1;
    end
    N = n; u_truncated  = zeros(J+1,N);
```

```
58      for n = 1:N
59          u_truncated(:,n)  = u(:,n);
60      end
61      u = u_truncated;
62      q_truncated = zeros(J-1,N-1); t0_truncated = zeros(J-1,N-1);
63      for n = 1:N-1
64          q_truncated(:,n) = q(:,n); t0_truncated(:,n) = t0(:,n);
65      end
66      q = q_truncated; t0 = t0_truncated;
67  end
68  end
```

Листинг 6.17. Диагностический блок `diagnostics.m`

```
1  function [q, t0, current_root] = diagnostics(step, type, u, initial_approx)
2  % Calculates singularity characteristics
3  dim = size(u); J = dim(1) - 1;
4  q  = zeros(J-1,1); t0 = zeros(J-1,1); u_temp = zeros(J,2);
5  for j = 1:J
6      u_temp(j,:) = u(j,:);
7  end
8  t = u(J+1,:); tau = t(2) - t(1);
9  f(:,1) = right_hand_autonomization( u(:,1) );
10 f(:,2) = right_hand_autonomization( u(:,2) );
11 current_root = step*ones(J-1,1);
12 for j = 1:J-1
13     if( type(j) == 1 )
14         q(j)  = tau/( u(j,1)/f(j,1) - u(j,2)/f(j,2) );
15         t0(j) = q(j)*u(j,1)/f(j,1) + t(1);
16     elseif( type(j) == 2 )
17         a1 = f(j,1)*u(j,2); a2 = f(j,2)*u(j,1);
18         f0 = @(x)( a1*(x+tau)*log(x+tau) - a2*x*log(x) );
19         f1 = @(x)( a1*( log(x+tau) + 1 ) - a2*( log(x) + 1 ) );
20         current_root(j) = newton( f0, f1, initial_approx(j) );
21         t0(j) = t(2) + current_root(j);
22         q(j) = -f(j,1)/u(j,1)*(tau + current_root(j)) ...
23             *log(tau + current_root(j));
24     end
25 end
26 end
```

Листинг 6.18. Тривиальная автономизация `right_hand_autonomization.m`

```
1  function f = right_hand_autonomization( u )
2  % Performs trivial autonomization: t becomes a new unknown function
3  J = length(u) - 2; u_temp = zeros(J,1);
4  for j = 1:J
5      u_temp(j) = u(j);
6  end
7  t = u(J+1);
8  f = ones(J+1,1); f0 = right_hand( u_temp, t );
```

```
for j = 1:J
    f(j) = f0(j);
end
f(J+1) = 1;
end
```

Листинг 6.19. Переход к длине дуги `right_hand_arc_length.m`

```
function f = right_hand_arc_length( u )
% Introduces arc length argument after trivial autonomization
J = length(u) - 1;
f = right_hand_autonomization( u ); f(J+1) = 1;
S = sqrt( sum(f.^2) );
f = f/S;
end
```

Листинг 6.20. Одностадийная комплексная схема Розенброка `CROS.m`

```
function u_hat = CROS( step, u )
% Performs one step of complex one-stage Rosenbrock scheme
J = length(u)-1;
Jacobi = Jacobi_matrix( u ); E = eye(J+1);
A = E - 0.5*(1+1i)*step*Jacobi; b = right_hand_arc_length(u);
x = A\b;
u_hat = u + step*real(x);
end
```

Листинг 6.21. Вычисление матрицы Якоби `Jacobi_matrix.m`

```
function Jacobi = Jacobi_matrix( u )
% Calculates Jacobi matrix
J = length(u)-1; delta = zeros(J+1);
hu = 1e-5;
for i = 1:J+1
    up = u; um = u;
    up(i) = u(i) + hu; um(i) = u(i) - hu;
    f1 = right_hand_arc_length( up );
    f2 = right_hand_arc_length( um );
    for j = 1:J+1
        delta(j,i) = f1(j) - f2(j);
    end
end
Jacobi = delta/2/hu;
end
```

Листинг 6.22. Поточечная оценка погрешности `richardson.m`

```
function R0 = richardson( u1, u2, p )
% Calculates point-wise estimation according to Richardson method.
dim = size(u1); R0 = zeros(dim(1),dim(2));
for j = 1:dim(1)
    for n = 1:dim(2)
```

```
6           R0(j,n) = ( u2( j,2*n-1 ) - u1( j,n ) )/( 2^p-1 );
7       end
8   end
9   end
```

Листинг 6.23. Иллюстрации `illustrations.m`

```
1  function illustrations( step, u, q, t0, precision_output )
2  % Auxiliary graphics: solution and singularuty characteristics on the
3  % thickest mesh and accuracy curve
4  dim1 = size(precision_output); mesh_num = dim1(1);
5  dim2 = size(u); J = dim2(1)-1; N_last = dim2(2);
6  N0 = (N_last-1)/2^mesh_num; N = zeros(1,mesh_num);
7  for m = 1:mesh_num
8      N(m) = N0*2^(m);
9  end
10 u_temp = zeros(J-1,N_last);
11 for j = 1:J-1
12     u_temp(j,:) = u(j,:);
13 end
14 t = u(J,:);
15 figure; hold on; xlabel('t'); ylabel('u'); title('SOLUTION');
16 for j = 1:J-1
17     c = j/5 - floor(j/5);
18     plot(t,u_temp(j,:),'Color',[c,c,c],'LineWidth',2);
19 end
20 str = 'u(1)';
21 for j = 1:J-2
22     str0 = strcat('u(',mat2str(j+1),')'); str = char(str,str0);
23 end
24 if( J<=10 ); legend(str); end
25 figure; hold on; xlabel('l'); ylabel('q, t0');
26 title('SINGULARITY CHARACTERISTICS');
27 l = 0:step:(N_last-2)*step;
28 for j = 1:J-1
29     c = j/5 - floor(j/5);
30     plot(l, q(j,:),'-','Color',[c,c,c],'LineWidth',2);
31     plot(l,t0(j,:),':','Color',[c,c,c],'LineWidth',2.5);
32 end
33 str = char( 'q(1)', 't0(1)' );
34 for j = 1:J-2
35     str_q  = strcat('q(', mat2str(j+1),')');
36     str_t0 = strcat('t0(',mat2str(j+1),')');
37     str = char( str, str_q ); str = char( str, str_t0 );
38 end
39 if( J<=10 ); legend(str); else legend('q','t0'); end
40 figure; hold on; xlabel('lg N'); ylabel('lg epsilon'); title('CONVERGENCE')
41 plot(log10(N),log10(precision_output(:,1)),...
42     '-ok','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',14)
43 plot(log10(N),log10(precision_output(:,2)),...
44     '-^k','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',14)
```

```
45 plot(log10(N),log10(precision_output(:,3)),...
46      '-sk','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',14)
47 legend('u','q','t0');
48 end
```

6.3 Пакет SuFaReC для расчета диффузии в пограничных слоях

6.3.1. Описание программ.

Алгоритм. В пакете `SuFaReC` (SUperFAst RElaxation Count) реализовано решение задачи Дирихле для двумерного и трехмерного уравнения Гельмгольца (1.8) на сетках, сгущающихся по пространству, с вычислением оценки точности по методу Ричардсона. На каждой пространственной сетке решение ищется счетом на установление по эволюционно-факторизованной схеме с линейно-тригонометрическим набором шагов. Для итерационного процесса вычисляется апостериорная асимптотически точная оценка погрешности решения сеточной системы (то есть оценка"недоитерированности"). Пакет включает две подпрограммы для решения эллиптических уравнений при разном числе пространственных измерений, а также подпрограммы для построения оценок погрешности.

Входные параметры. Для запуска пакета нужно вызвать основную функцию `SuFaReC` (от SUper FAst RElaxation Count). Ее обязательными аргументами являются

1. значение малого параметра `Mu`,
2. границы отрезков по x, y и z (в трехмерном случае), записанные в массив `Boundaries = [ax,bx,ay,by,az,bz]`.

Размерность задачи определяется по длине массива `Boundaries`: если он содержит 4 элемента, то задача двумерная, а если 6 – то трехмерная. Если хотя бы один из этих аргументов не задан, то программа выводит сообщение об ошибке и прекращает расчет. Следующие аргументы являются необязательными (пользователь может не задавать их, тогда они задаются по умолчанию):

3. точность `Epsilon_pde`, с которой требуется решить исходное уравнение в частных производных (по умолчанию 10^{-3}),
4. флаг `Grid_choice`, определяющий выбор сеток (0 – адаптивная, 1 – заданная пользователем; по умолчанию 0),
5. точность решения сеточных систем `Epsilon_grid_sys` на каждой сетке по пространству (по умолчанию 10^{-5}) и
6. флаг `Norm_choice`, определяющий выбор нормы, в которой будут вычисляться все погрешности (0 – C, 1 – L_2, 2 – среднеквадратичная; по умолчанию 0).

Пользователь должен задавать либо все необязательные аргументы, либо не задавать ни одного из них. В противном случае программа выведет сообщение об ошибке и предложит использовать значения по умолчанию. В случае отказа расчет будет прекращен.

Правая часть исходного уравнения, его коэффициенты и граничные условия вводятся в функциях `right_hand`, `coefficients` и `boundary_conditions` соответственно. При желании пользователь может задать собственные сетки по пространству. Для этого надо ввести производящие функции `x_user`, `y_user`, `z_user` и их производные `x_user_deriv`, `y_user_deriv`, `z_user_deriv` в процедуру `user_mesh`. По умолчанию эти сетки считаются равномерными.

Результаты расчетов. Функция `SuFaReC` возвращает массив `u`, содержащий решение на последней сетке, погрешность `epsilon_final` этого решения по методу Ричардсона и массив `iteration_precision`, в котором записаны точности, достигнутые в итерационном процессе при решении сеточных систем.

Также по окончании расчетов программа выводит графики сходимости счета на установление на каждой сетке, график сходимости при сгущении пространственных сеток, общий вид решения в виде изолиний с фиксированным шагом (в трехмерном случае – при фиксированном $z = (a_z + b_z)/2$) и сечение этого решения плоскостью $x = y$.

Настройка в данном пакете не требуется. Единственными свободными параметрами являются числа шагов самых грубых сеток по пространству `N0`, `K0` и `L0` (в трехмерном случае). Их следует задавать не слишком большими, чтобы можно было сделать достаточное число сгущений; по умолчанию положим $N_0 = K_0 = L_0 = 10$.

Кроме того, во избежание зацикливания введено ограничение на максимальное число сеток по пространству. Оно определяется переменной `max_grid_num`, заданной в функции `SuFaReC`, и по умолчанию равно 6. При использовании адаптивных сеток такого числа сгущений более чем достаточно.

Процедуры. Опишем работу функций, входящих в данный пакет. Процедура `SuFaReC` определяет размерность задачи и осуществляет сгущение сеток по пространству. Сгущения проводятся до тех пор, пока не будет достигнута требуемая точность либо не будет превышено максимальное число сеток.

В функции `SFRC_solver` реализован счет на установление на фиксированной сетке по пространству. На вход она принимает числа внутренних узлов `N`, `K`, `L` этой сетки. Если заданы только два аргумента, то реализуется двумерный случай; если все три, то трехмерный. Выходными аргументами являются решение сеточной системы `u` и его фактическая погрешность `iteration_precision`.

Функция `spectrum_estimates` реализует оценки границ спектра матрицы сеточной системы. Входные параметры – числа внутренних узлов `N`, `K` (и `L` в трехмерном случае). На выход выдаются массивы `lambda_max` и `lambda_min`, содержащие оценки спектров по направлениям x, y (и z в трехмерном случае), а также спектральное число обусловленности `conditioning`.

Функция `number_of_steps` принимает на вход массивы `lambda_max`, `lambda_min` и точность `epsilon`, с которой надо решить сеточную систему. Она строит априорную оценку сходимости линейно-тригонометрического набора и вычисляет числа шагов сгущающихся логарифмических подсеток. Эти числа записываются в массив `S_all`, который является выходным аргументом данной функции. В функции

`logarithmic_set` строится линейно-тригонометрический набор шагов. На вход она принимает массивы `lambda_max`, `lambda_min` и число шагов набора `S`.

Функция `iteration_precision_estimates` реализует апостериорные оценки сходимости счета на установление. Результатом ее работы являются интерполяционная оценка `precision_i` и экстраполяционная оценка `precision_e`. На вход функция принимает массив `u_all`, в котором записаны решения на всех логарифмических сетках, и оценку фона ошибок округления `epsilon_background`. График сходимости итерационного процесса на каждой сетке по пространству строится процедурой `iter_conv_illustration`.

Функция `evolutional_factorization` представляет собой реализацию эволюционно-факторизованной схемы для параболической задачи в двумерном и трехмерном случаях. Входными аргументами являются массив начальных условий `ic` (по которому подпрограмма определяет размерность задачи) и массив `tau`, содержащий набор шагов по времени. Выходным аргументом является массив `u` – решение параболической задачи.

Для работы этой функции требуется процедура `tridiagonal_matrix_algorithm`, в которой реализована одномерная прогонка с нулевым граничным условием Дирихле. Ее входными аргументами являются диагонали матрицы `A`, `B`, `C`, правая часть `F` и число внутренних узлов `N`, фактически равное числу неизвестных. Функция возвращает решение `v` трехдиагональной системы.

Метод Ричардсона при сгущении пространственных сеток реализован в функции `richardson_multidim`. На вход она принимает решения `u1` и `u2` на грубой и на подробной сетках соответственно и порядок точности схемы. Эта функция возвращает поточечную оценку погрешности `R0`, относящуюся к четным узлам подробной сетки.

Нормы всех погрешностей вычисляются процедурой `norm_calc`. Ее входной аргумент – сеточная функция, норму которой надо вычислить; возвращаемое значение – величина нормы. Графики решения и его сечения, а также сходимости при сгущении сеток по пространству строятся в процедуре `pde_solve_illustration`.

Адаптивная сетка генерируется функцией `adaptive_mesh`. Входными аргументами являются числа внутренних узлов `N`, `K` (и `L` в трехмерном случае), а на выходе возвращаются массивы узлов `x`, `y` (и `z`) и шагов `hx`, `hy` (и `hz`). Для построения этой сетки нужно решить нелинейное алгебраическое уравнение относительно ее параметров. В данном пакете для этого предусмотрена процедура `newton`, реализующая метод Ньютона (входные параметры – уравнение, его производная и начальное приближение; а возвращаемые – корень уравнения, его погрешность и число выполненных итераций).

Примечание. Заметим, что процедура `evolutional_factorization` представляет самостоятельный интерес и может использоваться вне предлагаемого пакета. Однако в случае нестационарных граничных условий или другого типа задачи (например, задача Неймана) она требует доработки (необходимо поменять граничные условия в прогонках).

6.3.2. Контрольные тесты.

Двумерная задача. Рассмотрим следующую задачу

$$\begin{gathered}\mu = 10^{-2}, \quad k_x = 2 + y^2, \quad k_y = 2 + x, \quad \varkappa = 1; \\ u(x, y, z) = 2.5(x + y + z), \ (x, y, z) \in \partial G; \\ f(x, y) = \cos\left[\pi(x + y)^2/4\right] \cos\left[3\pi(y - x)/4\right].\end{gathered} \tag{6.5}$$

В качестве области G выберем прямоугольник $[-1, 2] \times [-1, 1]$. Создадим файлы `coefficients.m`, `boundary_condition.m` и `right_hand.m` и введем в них следующий код:

Листинг 6.24. Двумерная задача (6.5), коэффициенты уравнения (`coefficients.m`)

```
function [kx,ky,kz,kappa] = coefficients(x,y)
kx = 2+y^2; ky = 2+x; kappa = 1;
kz = 0; % formal declaration
end
```

Листинг 6.25. Двумерная задача (6.5), граничные условия (`boundary_condition.m`)

```
function bc = boundary_condition(x,y)
bc = 2.5*(x+y);
end
```

Листинг 6.26. Двумерная задача (6.5), правая часть (`right_hand.m`)

```
function f = right_hand(x,y)
f = cos(0.25*pi*(x+y)^2)*cos(0.75*pi*(y-x));
end
```

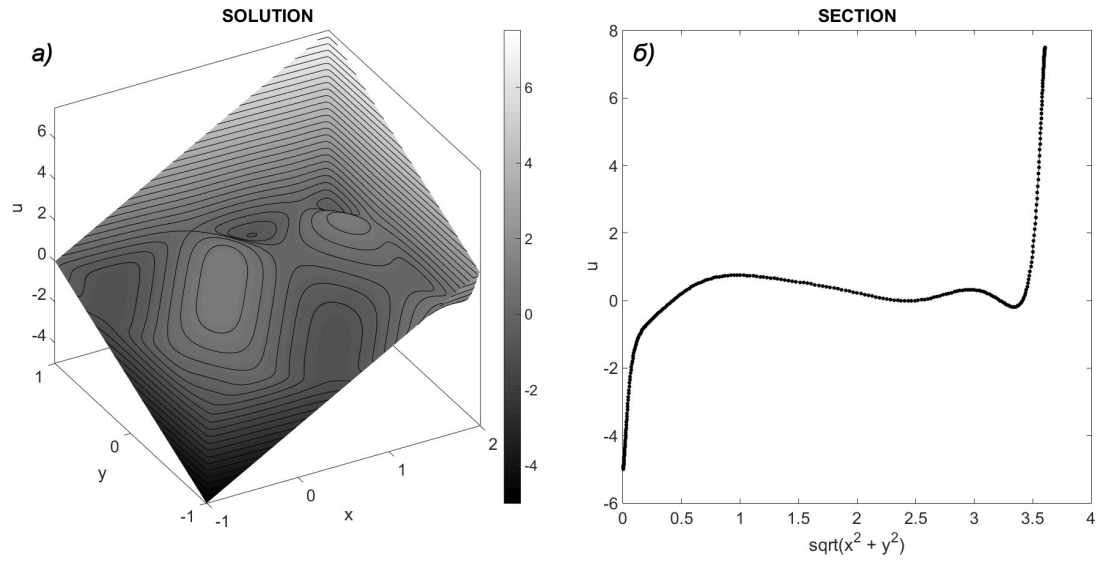

Рис. 6.5. Решение задачи (6.5); а) общий вид, б) сечение плоскостью $x = y$.

В функции `SuFaReC` зададим только обязательные аргументы, а необязательные опустим. Для ее вызова в командном окне или отдельном файле необходимо набрать и выполнить код

Листинг 6.27. Двумерная задача (6.5), входные параметры и вызов функции

```
Mu = 1e-2; % small parameter
ax = -1; bx = 2; ay = -1; by = 1; % domain boundaries
Boundaries = [ax,bx,ay,by];
[u, epsilon_final, iteration_precision] = SuFaReC(Mu, Boundaries);
```

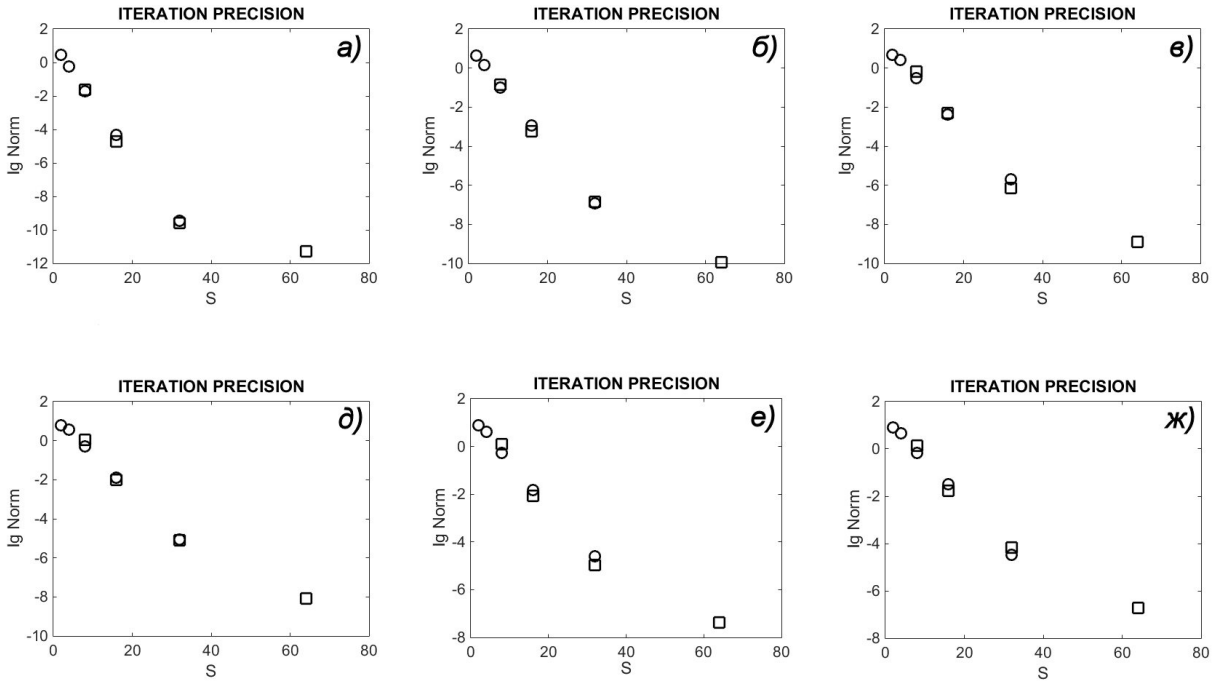

Рис. 6.6. Сходимость итераций в задаче (6.5). а) $N = K = 10$, б) $N = K = 20$, в) $N = K = 40$, г) $N = K = 80$, д) $N = K = 160$, е) $N = K = 320$; круглые маркеры – интерполяционная оценка, квадратные – экстраполяционная оценка.

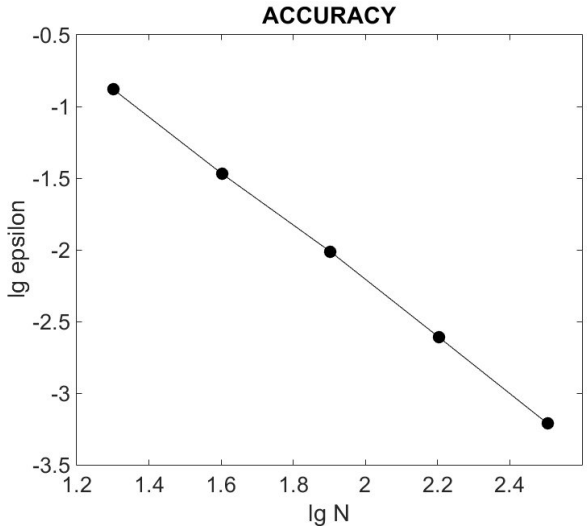

Рис. 6.7. Сходимость по пространству в задаче (6.5).

Точность, получаемая в этом расчете, равна `epsilon_final = 6.119e-4`. На рис. 6.5, *а* и 6.5, *б* показаны общий вид решения и его сечение плоскостью $x = y$ соответственно. Графики сходимости итераций в полулогарифмическом масштабе, относящиеся к разным пространственным сеткам приведены на рис. 6.6, *а* – 6.6, *е*. График

сходимости при сгущении сеток по пространству представлен на рис. 6.7, масштаб двойной логарифмический.

Трехмерная задача. Зададим параметры задачи

$$\begin{gathered}\mu = 10^{-2}; \quad k_x = 3 - z, \quad k_y = 2 + x, \quad k_z = 1 + y^2, \quad \varkappa = 1; \\ f(x,y,z) = 1.5\cos^2\left[\pi(z+1)(x^2+y)\right]\cos\left[\pi(x+z)\right]; \\ G = [-1,1]\times[-1,1]\times[-1,1]; \quad u(x,y,z) = 2.5(x+y+z),\ (x,y,z)\in\partial G.\end{gathered} \tag{6.6}$$

Условия этой задачи реализуются в файлах `coefficients.m`, `boundary_condition.m` и `right_hand.m` следующим образом:

Листинг 6.28. Трехмерная задача (6.6), коэффициенты уравнения (`coefficients.m`)

```
function [kx,ky,kz,kappa] = coefficients(x,y,z)
kx = 3-z; ky = 2+x; kz = 1+y^2; kappa = 1;
end
```

Листинг 6.29. Трехмерная задача (6.6), граничные условия (`boundary_condition.m`)

```
function bc = boundary_condition(x,y,z)
bc = 2.5*(x+y+z);
end
```

Листинг 6.30. Трехмерная задача (6.6), правая часть (`right_hand.m`)

```
function f = right_hand(x,y,z)
f = 1.5*cos( pi*(z+1)*(x^2+y) )^2*cos( pi*(x+z) );
end
```

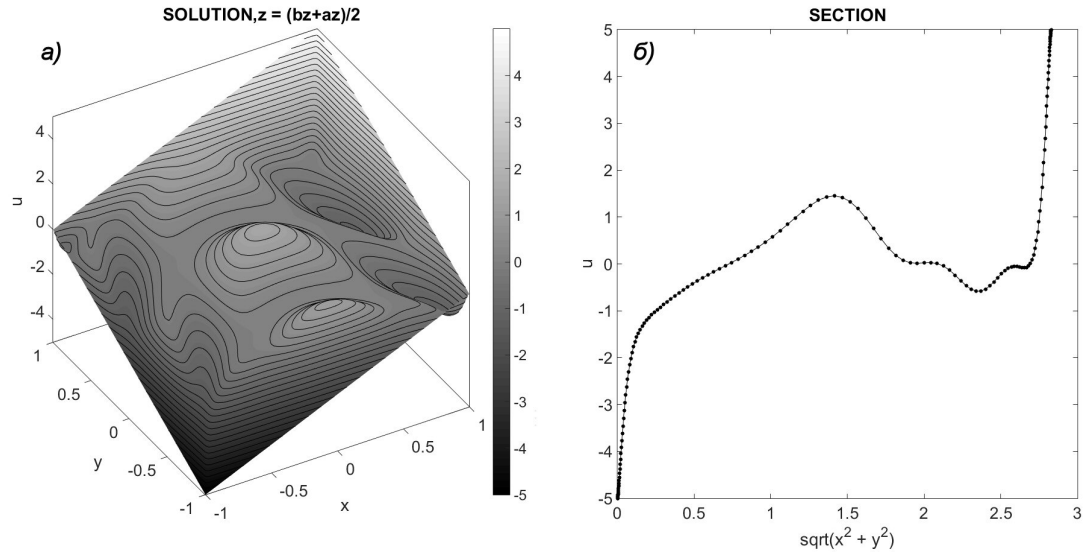

Рис. 6.8. Решение задачи (6.6) при фиксированном $z = (a_z + b_z)/2$; а) общий вид, б) сечение плоскостью $x = y$.

В этом примере зададим как обязательные, так и необязательные аргументы. Напишем в командном окне или в отдельном файле следующий код и выполним его:

Листинг 6.31. Трехмерная задача (6.6), входные параметры и вызов функции

```
Mu = 1e-2; % small parameter
Grid_choice = 0; % grid type
epsilon_grid_sys = 1e-4; % accuracy of the grid system solving
epsilon_pde  = 1e-2; % accuracy of the PDE solving
Norm_choice = 0; % mesh norm type
ax = -1; bx = 1; ay = -1; by = 1; az = -1; bz = 1; % domain boundaries
Boundaries = [ax,bx,ay,by,az,bz];
[u, epsilon_final, iteration_precision] = SuFaReC ...
    (Mu, Boundaries,epsilon_pde,Grid_choice,epsilon_grid_sys, ...
    Norm_choice);
```

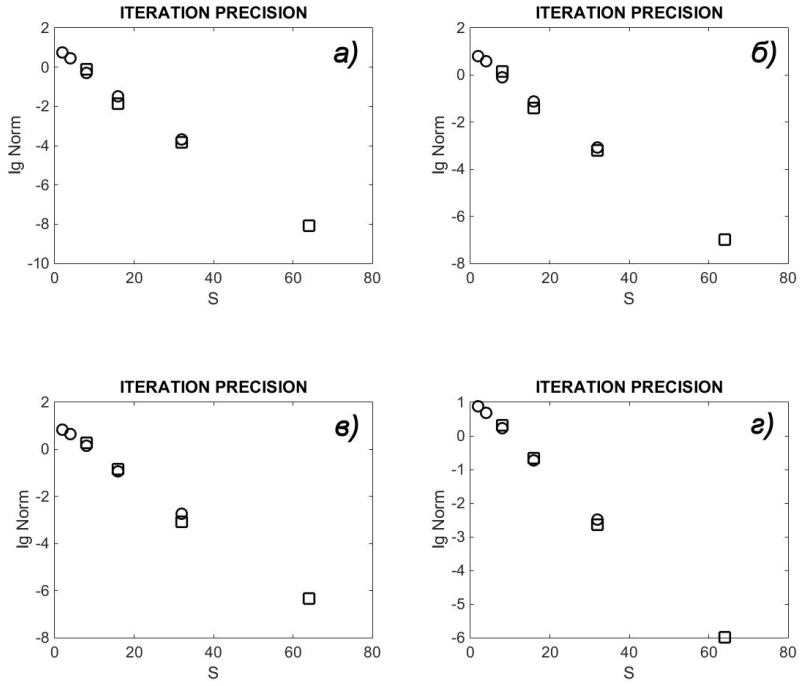

Рис. 6.9. Сходимость итераций в задаче (6.6); а) $N = K = 20$, б) $N = K = 40$, в) $N = K = 80$, г) $N = K = 160$; обозначения соответствуют рис. 6.6.

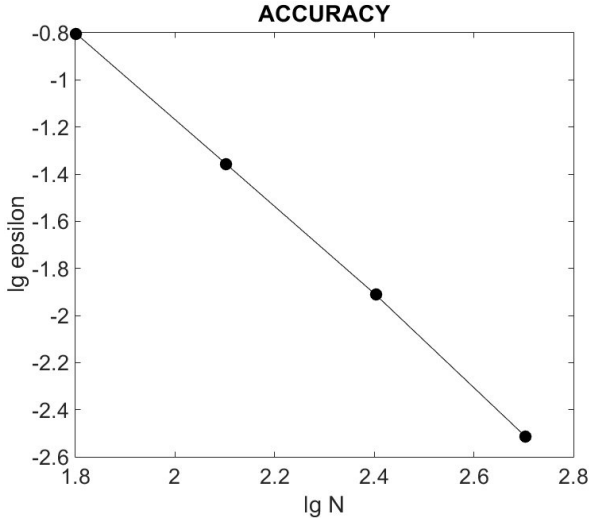

Рис. 6.10. Сходимость по пространству в задаче (6.6).

Точность этого расчета составляет `epsilon_final = 3.055e-3`. Общий вид решения при фиксированном $z = (a_z + b_z)/2$ показан на рис. 6.8, *а* в виде изолиний с фиксированным шагом. Сечение этого решения плоскостью $x = y$ представлено на рис. 6.8, *б*. На рис. 6.9, *а*–6.9, *г* даны графики сходимости итераций в полулогарифмическом масштабе. Наконец график сходимости при сгущении пространственных сеток в двойном логарифмическом масштабе изображен на рис. 6.10.

6.3.3. Листинги. Ниже приводятся листинги функций, входящих в пакет `SuFaReC`. Этот пакет распространяется по свободной лицензии BSD-3-clause и расположен по ссылке `https://bitbucket.org/alexander_belov/sufarec`.

Листинг 6.32. Основная функция `SuFaReC.m`

```
function [u,epsilon_u,iteration_precision] = SuFaReC(Mu, Boundaries,...
    Epsilon_pde, Grid_choice, Epsilon_grid_sys, Norm_choice)
% Solves Helmholz equation with guaranteed accuracy. This function carried
% out calculation on a sequence of nested until accuracy determined by
% Richardson method is less than user-required accuracy or 1e-3 by default.
if(nargin == 6); default_parameters = 0; end
if( nargin < 6 && nargin > 2 )
    disp('Error! Given auxiliary parameters are incomplete!');
    default_parameters = ...
        input('Do you wish to select default ones? 1 -> yes, 0 -> no: ');
    if( default_parameters == 0 )
        disp('Calculation aborted.');
        u = 0; epsilon_u = 0; iteration_precision = 0;
        return;
    end
end
if(nargin == 2); default_parameters = 1; end
if(nargin < 2)
    disp('Error! Problem statement is incomplete!');
    u = 0; epsilon_u = 0; iteration_precision = 0;
    return;
end
if( default_parameters == 1 )
    Epsilon_pde = 1e-3; Epsilon_grid_sys = 1e-5;
    Grid_choice = 0;     Norm_choice = 0;
end
global mu; global boundaries; global grid_choice;
global eps_grid_sys; global norm_choice;
mu = Mu; boundaries = Boundaries; eps_grid_sys = Epsilon_grid_sys;
norm_choice = Norm_choice; grid_choice = Grid_choice;
N0 = 10; K0 = 10; L0 = 10; max_grid_num = 6;
grid_precision = zeros(1,max_grid_num);
iteration_precision = zeros(1,max_grid_num);
condition = 1; grid_num = 1;
while( condition )
    if( grid_num > max_grid_num ); break; end;
    if( grid_num == 1 )
```

```
38          if( length(boundaries) == 4 )
39              N = N0-1; K = K0-1;
40              [u1,iteration_precision(grid_num)] = SFRC_solver( N,K );
41          end
42          if( length(boundaries) == 6 )
43              N = N0-1; K = K0-1; L = L0-1;
44              [u1,iteration_precision(grid_num)] = SFRC_solver( N,K,L );
45          end
46      else u1 = u2;
47      end
48      grid_num = grid_num + 1;
49      if( length(boundaries) == 4 )
50          N = (N+1)*2-1; K = (K+1)*2-1;
51          [u2,iteration_precision(grid_num)] = SFRC_solver( N,K );
52      end
53      if( length(boundaries) == 6 )
54          N = (N+1)*2-1; K = (K+1)*2-1; L = (L+1)*2-1;
55          [u2,iteration_precision(grid_num)] = SFRC_solver( N,K,L );
56      end
57      p = 2;
58      precision_u = richardson_multidim( u1, u2, p );
59      grid_precision(grid_num-1) = norm_calc(precision_u);
60      condition = grid_precision(grid_num-1) > Epsilon_pde;
61  end
62  u = u2; actual_grid_num = grid_num;
63  precision_output = zeros(1,actual_grid_num - 1);
64  for m = 1:actual_grid_num - 1
65      precision_output(m) = grid_precision(m);
66  end
67  epsilon_u = precision_output(actual_grid_num-1);
68  pde_solve_illustration(u, N0, K0, L0, precision_output);
69  end
```

Листинг 6.33. Решатель счетом на установление `SFRC_solver.m`

```
1  function [solution, iteration_precision] = SFRC_solver(N, K, L)
2  % Solves grid system on a given grid via superfast relaxation count.
3  global grid_choice; global eps_grid_sys;
4  if( nargin == 2 )
5      if (grid_choice == 0)
6          [x,~,y,~] = adaptive_mesh( N,K );
7      else
8          [x,~,y,~] = user_mesh( N,K );
9      end
10     [lambda_max, lambda_min, conditioning] = spectrum_estimates( N,K );
11     epsilon_background = 10^(-16.2)*conditioning;
12     epsilon = max(epsilon_background, eps_grid_sys);
13     S_all = number_of_steps(lambda_max, lambda_min, epsilon);
14     I = length(S_all); u_all = zeros(N+2,K+2,I); ic = zeros(N+2,K+2);
15     for log_grid_num = 1:I
16         S = S_all(log_grid_num);
```

```
17          tau_lt = logarithmic_set(lambda_max, lambda_min, S);
18          if( log_grid_num == 1 )
19              tau = tau_lt;
20              for k = 1:K+2
21                  ic( 1 ,k) = boundary_condition(x( 1 ), y(k));
22                  ic(N+2,k) = boundary_condition(x(N+2), y(k));
23              end
24              for n = 1:N+2
25                  ic(n, 1 ) = boundary_condition(x(n), y( 1 ));
26                  ic(n,K+2) = boundary_condition(x(n), y(K+2));
27              end
28          else
29              tau = zeros(1,S/2);
30              for s = 1:S/2
31                  tau(s) = tau_lt(2*s);
32              end
33              ic = u_all(:,:,log_grid_num-1);
34          end
35          u_all(:,:,log_grid_num) = evolutional_factorization(ic, tau);
36      end
37      solution = u_all(:,:,I);
38  end
39  if( nargin == 3 )
40      if (grid_choice == 0)
41          [x,~,y,~,z,~] = adaptive_mesh( N,K,L );
42      else
43          [x,~,y,~,z,~] = user_mesh( N,K,L );
44      end
45      [lambda_max, lambda_min, conditioning] = spectrum_estimates( N,K,L );
46      epsilon_background = 10^(-16.2)*conditioning;
47      epsilon = max(epsilon_background, eps_grid_sys);
48      S_all = number_of_steps(lambda_max, lambda_min, epsilon);
49      I = length(S_all);
50      u_all = zeros(N+2,K+2,L+2,I); ic = zeros(N+2,K+2,L+2);
51      for log_grid_num = 1:I
52          S = S_all(log_grid_num);
53          tau_lt = logarithmic_set(lambda_max, lambda_min, S);
54          if( log_grid_num == 1 )
55              tau = tau_lt;
56              for k = 1:K+2
57                  for l = 1:L+2
58                      ic( 1 ,k,l) = boundary_condition(x( 1 ), y(k), z(l));
59                      ic(N+2,k,l) = boundary_condition(x(N+2), y(k), z(l));
60                  end
61              end
62              for n = 1:N+2
63                  for l = 1:L+2
64                      ic(n, 1 ,l) = boundary_condition(x(n), y( 1 ), z(l));
65                      ic(n,K+2,l) = boundary_condition(x(n), y(K+2), z(l));
66                  end
```

```
67              end
68              for n = 1:N+2
69                  for k = 1:K+2
70                      ic(n,k, 1 ) = boundary_condition(x(n), y(k), z( 1 ));
71                      ic(n,k,L+2) = boundary_condition(x(n), y(k), z(L+2));
72                  end
73              end
74          else
75              tau = zeros(1,S/2);
76              for s = 1:S/2
77                  tau(s) = tau_lt(2*s);
78              end
79              ic = u_all(:,:,:,log_grid_num-1);
80          end
81          u_all(:,:,:,log_grid_num) = evolutional_factorization(ic, tau);
82      end
83      solution = u_all(:,:,:,I);
84  end
85  [precision_i, precision_e] = ...
86      iteration_precision_estimates(u_all, epsilon_background);
87  iter_conv_illustration...
88      (S_all, precision_i, precision_e, epsilon_background);
89  if (I==1 || I==2); iteration_precision = epsilon; end
90  if (I>=3); iteration_precision = precision_e(I-2); end
91  end
```

Листинг 6.34. Оценки границ спектра `spectrum_estimates.m`

```
1  function [lambda_max, lambda_min, conditioning] = spectrum_estimates( N,K,L
       )
2  % Constructs estimates for spectrum boundaries of the grid system.
3  global grid_choice; global mu; global boundaries;
4  if( nargin == 2 )
5      est_x = zeros(N+1,K+1); est_y = est_x;
6      kx = est_x; ky = est_x; kappa1 = est_x; kappa2 = est_x;
7      if (grid_choice == 0)
8          [x,hx,y,hy] = adaptive_mesh( N,K );
9      else
10         [x,hx,y,hy] = user_mesh( N,K );
11     end
12     for n = 1:N+1
13         for k = 1:K+1
14             x_s = 0.5*(x(n+1)+x(n)); y_s = 0.5*(y(k+1)+y(k));
15             [kx(n,k),~,~,kappa1(n,k)] = coefficients(x_s,y(k));
16             [~,ky(n,k),~,kappa2(n,k)] = coefficients(x(n),y_s);
17         end
18     end
19     for n = 2:N+1
20         for k = 2:K+1
21             k1 = kx(n-1,k); k2 = kx(n,k);
22             est_x(n,k) = mu^2*4*(k1/hx(n-1) + k2/hx(n))/(hx(n)+hx(n-1)) ...
```

```
23                  + 0.5*kappa1(n,k);
24              k3 = ky(n,k-1); k4 = ky(n,k);
25              est_y(n,k) = mu^2*4*(k3/hy(k-1) + k4/hy(k))/(hy(k)+hy(k-1)) ...
26                  + 0.5*kappa2(n,k);
27          end
28      end
29      lambda_x_max_est = max(max(est_x)); lambda_y_max_est = max(max(est_y));
30      lambda_max = [lambda_x_max_est, lambda_y_max_est];
31      lx = boundaries(2) - boundaries(1); ly = boundaries(4) - boundaries(3);
32      lambda_x_min_est = mu^2*(pi/lx)^2*min(min(kx + 0.5*kappa1));
33      lambda_y_min_est = mu^2*(pi/ly)^2*min(min(ky + 0.5*kappa2));
34      lambda_min = [lambda_x_min_est, lambda_y_min_est];
35  end
36  if( nargin == 3 )
37      est_x = zeros(N+1,K+1,L+1); est_y = est_x; est_z = est_x;
38      kx = est_x; ky = est_x; kz = est_x;
39      kappa1 = est_x; kappa2 = est_x; kappa3 = est_x;
40      if (grid_choice == 0)
41          [x,hx,y,hy,z,hz] = adaptive_mesh( N,K,L );
42      else
43          [x,hx,y,hy,z,hz] = user_mesh( N,K,L );
44      end
45      for n = 1:N+1
46          for k = 1:K+1
47              for l = 1:L+1
48                  x_s = 0.5*(x(n+1)+x(n));
49                  y_s = 0.5*(y(k+1)+y(k));
50                  z_s = 0.5*(z(l+1)+z(l));
51                  [kx(n,k,l),~,~,kappa1(n,k,l)]= coefficients(x_s,y(k),z(l));
52                  [~,ky(n,k,l),~,kappa2(n,k,l)]= coefficients(x(n),y_s,z(l));
53                  [~,~,kz(n,k,l),kappa3(n,k,l)]= coefficients(x(n),y(k),z_s);
54              end
55          end
56      end
57      for n = 2:N+1
58          for k = 2:K+1
59              for l = 2:L+1
60                  k1 = kx(n-1,k,l); k2 = kx(n,k,l);
61                  est_x(n,k,l) = mu^2*4*(k1/hx(n-1)...
62                      + k2/hx(n))/(hx(n)+hx(n-1)) + kappa1(n,k,l)/3;
63                  k3 = ky(n,k-1,l); k4 = ky(n,k,l);
64                  est_y(n,k,l) = mu^2*4*(k3/hy(k-1)...
65                      + k4/hy(k))/(hy(k)+hy(k-1)) + kappa2(n,k,l)/3;
66                  k5 = ky(n,k,l-1); k6 = ky(n,k,l);
67                  est_z(n,k,l) = mu^2*4*(k5/hz(l-1)...
68                      + k6/hz(l))/(hz(l)+hz(l-1)) + kappa3(n,k,l)/3;
69              end
70          end
71      end
72      lambda_x_max_est = max(max(max(est_x)));
```

```
73      lambda_y_max_est = max(max(max(est_y)));
74      lambda_z_max_est = max(max(max(est_z)));
75      lambda_max = [lambda_x_max_est, lambda_y_max_est, lambda_z_max_est];
76      lx = boundaries(2) - boundaries(1);
77      ly = boundaries(4) - boundaries(3);
78      lz = boundaries(6) - boundaries(5);
79      lambda_x_min_est = mu^2*(pi/lx)^2*min(min(min( kx + kappa1/3 )));
80      lambda_y_min_est = mu^2*(pi/ly)^2*min(min(min( ky + kappa2/3 )));
81      lambda_z_min_est = mu^2*(pi/lz)^2*min(min(min( kz + kappa3/3 )));
82      lambda_min = [lambda_x_min_est, lambda_y_min_est, lambda_z_min_est];
83  end
84  conditioning = sum(lambda_max)/sum(lambda_min);
85  end
```

Листинг 6.35. Числа шагов сгущающихся логарифмических сеток `number_of_steps.m`

```
1  function S_all = number_of_steps( lambda_max, lambda_min, epsilon )
2  % Calculates the numbers of steps in nested logarithmic grids via a priori
3  % convergence estimate.
4  relation = sum(lambda_max)/sum(lambda_min);
5  S = ceil(-( 4/(pi^2 + 2*pi))*log( relation )*log(epsilon));
6  S_temp = zeros(1,10); S_temp(1) = S; I = 1;
7  while (S_temp(I) >= 2)
8      S_temp(I+1) = S_temp(I)/2;
9      I = I+1; if (I == 10); break; end
10 end
11 S_all = zeros(1,I);
12 for m = 1:I
13     S_temp(m) = ceil( S_temp(I) )*2^(m-1);
14     S_all(m) = S_temp(m);
15 end
16 end
```

Листинг 6.36. Линейно-тригонометрический набор шагов `logarithmic_set.m`

```
1  function tau = logarithmic_set( lambda_max, lambda_min, S )
2  % Constructs linear-trigonometric set in logarithmic scale with given
3  % number of steps.
4  if( length(lambda_max) == 2 )
5      tau_min = 2/sum(lambda_max); tau_max = 2/sum(lambda_min);
6  end
7  if( length(lambda_max) == 3 )
8      l_x = lambda_min(1); l_y = lambda_min(2); l_z = lambda_min(3);
9      a = l_x + l_y + l_z; b = l_x*l_y + l_x*l_z + l_y*l_z; c = l_x*l_y*l_z;
10     phi = real( (1/3)*acos( - c * (b/3)^(-1.5 ) ) );
11     tau_star = - sqrt(3)/( sqrt(b) * cos(phi+2*pi/3) );
12     fr = 2/tau_star;
13     Num = fr^3 - a*fr^2 + b*fr + c; Den = fr^3 + a*fr^2 + b*fr + c;
14     rho = Num/Den;
15     if( rho > 0 ); tau_max = tau_star; end
16     if( rho < 0 )
```

```
17          kard1 = (a^2)/3 - b; kard2 = -2*(a^3)/27 + a*b/3 + c;
18          phi = real( (1/3)*acos( (kard2/2)*(kard1/3)^(-1.5 ) ) );
19          fr = -2*sqrt(kard1 /3)*cos( phi - 2*pi/3 );
20          tau_max = 2/( fr + a/3 );
21      end
22      l_x = lambda_max(1); l_y = lambda_max(2); l_z = lambda_max(3);
23      a = l_x + l_y + l_z; b = l_x*l_y + l_x*l_z + l_y*l_z; c = l_x*l_y*l_z;
24      phi = real( (1/3)*acos( - c*(b/3)^(-1.5 ) ) );
25      tau_star = - sqrt(3)/( sqrt(b)*cos(phi+2*pi/3) );
26      fr = 2/tau_star;
27      Num = fr^3 - a*fr^2 + b*fr + c; Den = fr^3 + a*fr^2 + b*fr + c;
28      rho = Num/Den;
29      if( rho > 0 ); tau_min = tau_star; end
30      if( rho < 0 )
31          kard1 = (a^2)/3 - b; kard2 = -2*(a^3)/27 + a*b/3 + c;
32          phi = real( (1/3)*acos( (kard2/2)*(kard1/3)^(-1.5 ) ) );
33          fr = -2*sqrt(kard1 /3)*cos( phi + 2*pi/3 );
34          tau_min = 2/( fr + a/3 );
35      end
36  end
37  center = 0.5*log(tau_min*tau_max); width = 0.5*log(tau_max/tau_min);
38  tau = zeros(1,S+1); theta = tau; lt = tau;
39  for s = 1:S+1
40      theta(s) = (s-1)/(S);
41      lt(s) = (2/(pi+2))*cos(theta(s)*pi-pi) + (pi/(pi+2))*(2*theta(s)-1);
42      tau(s) = exp( center + width*( lt(s) ) );
43  end
44  end
```

Листинг 6.37. Апостериорные оценки погрешности для счета на установление `iteration_precision_estimates.m`

```
1  function [precision_i,precision_e] = iteration_precision_estimates...
2      ( u_all, epsilon_background )
3  % Calculates relaxation count accuracy estimates (interpolational and
4  % extrapolational ones) and compares them with round-off background
5  % estimates.
6  dim = size(u_all); I = dim(end);
7  precision_i = zeros(1,I-1); precision_e = zeros(1,I-2);
8  if( length(dim) == 3 )
9      for log_grid_num = 1:I-1
10         diff1 = u_all(:,:,log_grid_num+1) - u_all(:,:,log_grid_num);
11         precision_i(log_grid_num) = norm_calc( diff1 );
12     end
13     for log_grid_num = 1:I-2
14         diff2 = u_all(:,:,log_grid_num+1) - u_all(:,:, log_grid_num );
15         diff3 = u_all(:,:,log_grid_num+2) - u_all(:,:,log_grid_num+1);
16         Den = norm_calc( diff2 ); Num = norm_calc( diff3 );
17         precision_e(log_grid_num) = Num^3/Den^2;
18     end
19 end
```

```
20 if( length(dim) == 4 )
21     for log_grid_num = 1:I-1
22         diff1 = u_all(:,:,:,log_grid_num+1) - u_all(:,:,:,log_grid_num);
23         precision_i(log_grid_num) = norm_calc( diff1 );
24     end
25     for log_grid_num = 1:I-2
26         diff2 = u_all(:,:,:,log_grid_num+1) - u_all(:,:,:, log_grid_num );
27         diff3 = u_all(:,:,:,log_grid_num+2) - u_all(:,:,:,log_grid_num+1);
28         Den = norm_calc( diff2 ); Num = norm_calc( diff3 );
29         precision_e(log_grid_num) = Num^3/Den^2;
30     end
31 end
32 for j = 1:I-1 % Сравнение с оценкой фона ошибок округления
33     precision_i(j) = max( epsilon_background, precision_i(j) );
34 end
35 for j = 1:I-2
36     precision_e(j) = max( epsilon_background, precision_e(j) );
37 end
38 end
```

Листинг 6.38. Графики сходимости счета на установление `iter_conv_illustration.m`

```
1  function iter_conv_illustration...
2      (S_all, precision_i, precision_e, epsilon_background)
3  % Performs relaxation count convergence plots in semilogarithmic scale.
4  global eps_grid_sys; epsilon = max(epsilon_background, eps_grid_sys);
5  I = length(S_all); S_intr = zeros(1,I-1); S_extr = zeros(1,I-2);
6  for log_grid_num = 1:I-1
7      S_intr(log_grid_num) = S_all(log_grid_num);
8  end
9  for log_grid_num = 1:I-2
10     S_extr(log_grid_num) = S_all(log_grid_num+2);
11 end
12 figure; xlabel('S'); ylabel('lg Norm'); title('ITERATION PRECISION');
13 if (I == 1); hold on; plot(S_all(I),log10(epsilon),...
14         'rs','LineWidth',2,'MarkerEdgeColor','k','MarkerSize',11)
15 end
16 if (I >= 2); hold on;
17     plot(S_intr,log10(precision_i),...
18         'o' ,'LineWidth',2,'MarkerEdgeColor','k','MarkerSize',11)
19 end
20 if (I == 2); hold on; plot(S_all(I), log10(epsilon),...
21         'rs','LineWidth',2,'MarkerEdgeColor','k','MarkerSize',11)
22 end
23 if (I >= 3); hold on; plot(S_extr,log10(precision_e),...
24         'rs','LineWidth',2,'MarkerEdgeColor','k','MarkerSize',14)
25 end
26 end
```

Листинг 6.39. Эволюционная факторизация `evolutional_factorization.m`

```
function u = evolutional_factorization( ic, tau )
% Solves parabolic problem via evolutional factorized scheme with a given
% set of time steps.
global mu; global grid_choice;
u1 = ic; dim = size(ic);
if( length(dim) == 2 )
    N = dim(1) - 2; K = dim(2) - 2; NKL = max(N,K);
    A = zeros(1,NKL+1); B = A; C = A; F = A;
    v = zeros(N+2,K+2); step = v;
    if (grid_choice == 0)
        [x,hx,y,hy] = adaptive_mesh( N,K );
    else
        [x,hx,y,hy] = user_mesh( N,K );
    end
    kx = zeros(N+1,K+1); ky = kx; kappa = kx;
    for n = 1:N+1
        for k = 1:K+1
            x_temp = 0.5*(x(n+1)+x(n)); y_temp = 0.5*(y(k+1)+y(k));
            [kx(n,k),~] = coefficients(x_temp,y(k));
            [~,ky(n,k)] = coefficients(x(n),y_temp);
            [~,~,~,kappa(n,k)] = coefficients(x(n),y(k));
        end
    end
    for m = 1:length(tau)
        for k = 2:K+1
            for n = 2:N+1
                Ax   = 2*(mu^2)*kx(n-1,k)/hx(n-1)/(hx(n)+hx(n-1));
                Bx   = 2*(mu^2)*kx( n ,k)/hx( n )/(hx(n)+hx(n-1));
                Cx   = - (Ax + Bx);
                A(n) = - 0.5*tau(m)*Ax; B(n) = - 0.5*tau(m)*Bx;
                C(n) = - 1 + 0.5*tau(m)*Cx - (tau(m)/4)*kappa(n,k);

                Ay   = 2*(mu^2)*ky(n,k-1)/hy(k-1)/(hy(k)+hy(k-1));
                By   = 2*(mu^2)*ky( n ,k)/hy( k )/(hy(k)+hy(k-1));
                Cy   = - (Ay + By);
                F1   = - Ax*u1(n-1,k) - Cx*u1(n,k) - Bx*u1(n+1,k);
                F2   = - Ay*u1(n,k-1) - Cy*u1(n,k) - By*u1(n,k+1);
                F(n) = - right_hand(x(n),y(k)) + kappa(n,k)*u1(n,k)+F1+F2;
            end
            v(:,k) = tridiagonal_matrix_algorithm(A,B,C,F,N);
        end
        for n = 2:N+1
            for k=2:K+1
                Ay  = 2*(mu^2)*ky(n,k-1)/hy(k-1)/(hy(k)+hy(k-1));
                By  = 2*(mu^2)*ky(n, k )/hy( k )/(hy(k)+hy(k-1));
                Cy  = - (Ay + By);
                A(k) = - 0.5*tau(m)*Ay;  B(k) = - 0.5*tau(m)*By;
                C(k) = - 1 + 0.5*tau(m)*Cy - (tau(m)/4)*kappa(n,k);
                F(k) = - v(n,k);
            end
```

```
51              step(n,:) = tridiagonal_matrix_algorithm(A,B,C,F,K);
52          end
53          u = u1 + tau(m)*step;
54          u1 = u;
55      end
56  end
57  if( length(dim) == 3 )
58      N = dim(1) - 2; K = dim(2) - 2; L = dim(3) - 2; NKL = max( [N,K,L] );
59      A = zeros(1,NKL+1); B = A; C = A; F = A;
60      w = zeros(N+2,K+2,L+2); v = w; step = w;
61      if (grid_choice == 0)
62          [x,hx,y,hy,z,hz] = adaptive_mesh( N,K,L );
63      else
64          [x,hx,y,hy,z,hz] = user_mesh( N,K,L );
65      end
66      kx = zeros(N+1,K+1,L+1); ky = kx; kz = kx; kappa = kx;
67      for n = 1:N+1
68          for k = 1:K+1
69              for l = 1:L+1
70                  x_temp = 0.5*(x(n+1)+x(n));
71                  y_temp = 0.5*(y(k+1)+y(k));
72                  z_temp = 0.5*(z(l+1)+z(l));
73                  [kx(n,k,l),~,~] = coefficients(x_temp,y(k),z(l));
74                  [~,ky(n,k,l),~] = coefficients(x(n),y_temp,z(l));
75                  [~,~,kz(n,k,l)] = coefficients(x(n),y(k),z_temp);
76                  [~,~,~,kappa(n,k,l)] = coefficients(x(n),y(k),z(l));
77              end
78          end
79      end
80      for m = 1:length(tau)
81          for n = 2:N+1
82              for k = 2:K+1
83                  for l = 2:L+1
84                      Az = 2*(mu^2)*kz(n,k,l-1)/hz(l-1)/(hz(l)+hz(l-1));
85                      Bz = 2*(mu^2)*kz(n,k, l )/hz( l )/(hz(l)+hz(l-1));
86                      Cz = - (Az + Bz);
87                      A(l) = - 0.5*tau(m)*Az;  B(l) = - 0.5*tau(m)*Bz;
88                      C(l) = - 1 + 0.5*tau(m)*Cz - (tau(m)/6)*kappa(n,k,l);
89
90                      Ax = 2*(mu^2)*kx(n-1,k,l)/hx(n-1)/(hx(n)+hx(n-1));
91                      Bx = 2*(mu^2)*kx(n,k, l )/hx( n )/(hx(n)+hx(n-1));
92                      Cx = - (Ax + Bx);
93                      Ay = 2*(mu^2)*ky(n,k-1,l)/hy(k-1)/(hy(k)+hy(k-1));
94                      By = 2*(mu^2)*ky(n, k ,l)/hy( k )/(hy(k)+hy(k-1));
95                      Cy = - (Ay + By);
96                      F1 = - Ax*u1(n-1,k,l) - Cx*u1(n,k,l) - Bx*u1(n+1,k,l);
97                      F2 = - Ay*u1(n,k-1,l) - Cy*u1(n,k,l) - By*u1(n,k+1,l);
98                      F3 = - Az*u1(n,k,l-1) - Cz*u1(n,k,l) - Bz*u1(n,k,l+1);
99                      F(l) = - right_hand(x(n),y(k),z(l))...
100                         + kappa(n,k,l)*u1(n,k,l) + F1 + F2 + F3;
```

```
                end
                w(n,k,:) = tridiagonal_matrix_algorithm(A,B,C,F,L);
            end
        end
        for n = 2:N+1
            for l = 2:L+1
                for k=2:K+1
                    Ay  = 2*(mu^2)*ky(n,k-1,l)/hy(k-1)/(hy(k)+hy(k-1));
                    By  = 2*(mu^2)*ky(n, k ,l)/hy( k )/(hy(k)+hy(k-1));
                    Cy  = - (Ay + By);
                    A(k) = - 0.5*tau(m)*Ay;  B(k) = - 0.5*tau(m)*By;
                    C(k) = - 1 + 0.5*tau(m)*Cy - (tau(m)/6)*kappa(n,k,l);
                    F(k) = - w(n,k,l);
                end
                v(n,:,l) = tridiagonal_matrix_algorithm(A,B,C,F,K);
            end
        end
        for k=2:K+1
            for l = 2:L+1
                for n = 2:N+1
                    Ax = 2*(mu^2)*kx(n-1,k,l)/hx(n-1)/(hx(n)+hx(n-1));
                    Bx = 2*(mu^2)*kx( n ,k,l)/hx( n )/(hx(n)+hx(n-1));
                    Cx = - (Ax + Bx);
                    A(n) = - 0.5*tau(m)*Ax;  B(n) = - 0.5*tau(m)*Bx;
                    C(n) = - 1 + 0.5*tau(m)*Cx - (tau(m)/6)*kappa(n,k,l);
                    F(n)  = - v(n,k,l);
                end
                step(:,k,l) = tridiagonal_matrix_algorithm(A,B,C,F,N);
            end
        end
        u = u1 + tau(m)*step;
        u1 = u;
    end
end
```

Листинг 6.40. Прогонка `tridiagonal_matrix_algorithm.m`

```
function v = tridiagonal_matrix_algorithm(A,B,C,F,N)
% Performs three-diagonal matrix algorithm (aka run) for system with zero
% Dirichlet boundary conditions, given diagonals and right hand.
v = zeros(1,N+2); Alpha = v; Beta = v;
for n = 2:N+1
    Alpha(n+1) = B(n)/(C(n) - Alpha(n)*A(n));
    Beta(n+1)  = (F(n) + A(n)*Beta(n))/(C(n) - Alpha(n)*A(n));
end
v(N+2) = 0;
for r = 0:N
    v(N+1-r) = Alpha(N+2-r)*v(N+2-r) + Beta(N+2-r);
end
end
```

Листинг 6.41. Метод Ричардсона `richardson_multidim.m`

```
function R0 = richardson_multidim( u1, u2, p )
% Calculates point-wise estimation according to Richardson method.
dim = size(u1);
if( length(dim) == 2 )
    N = dim(1)-1; K = dim(2)-1; R0 = zeros(N+1,K+1);
    for n = 1:N+1
        for k = 1:K+1
            R0(n,k) = ( u2( 2*n-1,2*k-1 ) - u1( n,k ) )/( 2^p-1 );
        end
    end
end
if( length(dim) == 3 )
    N = dim(1)-1; K = dim(2)-1; L = dim(3)-1; R0 = zeros(N+1,K+1,L+1);
    for n = 1:N+1
        for k = 1:K+1
            for l = 1:L+1
                R0(n,k,l) = (u2(2*n-1,2*k-1,2*l-1) - u1(n,k,l))/( 2^p-1 );
            end
        end
    end
end
end
```

Листинг 6.42. Вычисление нормы `norm_calc.m`

```
function Norm = norm_calc(u)
% Calculates norm of a grid function.
global norm_choice; global grid_choice;
dim = size(u);
if( length(dim) == 2 )
    N = dim(1) - 2; K = dim(2) - 2;
    if (grid_choice == 0)
        [~,hx,~,hy] = adaptive_mesh( N,K );
    else
        [~,hx,~,hy] = user_mesh( N,K );
    end
    if (norm_choice == 0); Norm = max( max( abs(u) ) ); end
    if (norm_choice == 1)
        intx = zeros(1,N+2); Int1 = 0;
        for n = 1:N+2
            for k = 1:K+1
                intx(n) = intx(n) + 0.5*hy(k)*(u(n,k)^2 + u(n,k+1)^2);
            end
        end
        for n = 1:N+1
            Int1 = Int1 + 0.5*hx(n)*(intx(n) + intx(n+1));
        end
        Norm = sqrt(Int1);
    end
    if (norm_choice == 2)
```

```
26          average = 0;
27          for n = 1:N+2
28              for k = 1:K+2
29                  average = average + u(n,k)^2;
30              end
31          end
32          Norm = sqrt(average/(K+2)/(N+2));
33      end
34  end
35  if( length(dim) == 3 )
36      N = dim(1) - 2; K = dim(2) - 2; L = dim(3) - 2;
37      if (grid_choice == 0)
38          [~,hx,~,hy,~,hz] = adaptive_mesh( N,K,L );
39      else
40          [~,hx,~,hy,~,hz] = user_mesh( N,K,L );
41      end
42      if (norm_choice == 0); Norm = max(max(max(abs(u)))); end
43      if (norm_choice == 1)
44          intxy = zeros(N+2,K+2); intx = zeros(1,N+2); Int1 = 0;
45          for n = 1:N+2
46              for k = 1:K+2
47                  for l = 1:L+1
48                      intxy(n,k) = intxy(n,k)...
49                          + 0.5*hz(l)*(u(n,k,l)^2 + u(n,k,l+1)^2);
50                  end
51              end
52          end
53          for n = 1:N+2
54              for k = 1:K+1
55                  intx(n) = intx(n)...
56                      +0.5*hy(k)*(intxy(n,k)+intxy(n,k+1));
57              end
58          end
59          for n = 1:N+1
60              Int1 = Int1 + 0.5*hx(n)*(intx(n) + intx(n+1));
61          end
62          Norm = sqrt(Int1);
63      end
64      if (norm_choice == 2)
65          average = 0;
66          for n = 1:N+2
67              for k = 1:K+2
68                  for l = 1:L+2
69                      average = average + u(n,k,l)^2;
70                  end
71              end
72          end
73          Norm = sqrt(average/(K+2)/(N+2)/(L+2));
74      end
75  end
```

```
76 end
```

Листинг 6.43. Адаптивная сетка `adaptive_mesh.m`

```
1  function [x,hx,y,hy,z,hz] = adaptive_mesh( N,K,L )
2  % Generates semi-uniform mesh adapted to boundary layer and tranzition
3  % zone. The mesh has N, K (and L in 3D) inner points.
4  global boundaries; global mu;
5  ax = boundaries(1); bx = boundaries(2);
6  ay = boundaries(3); by = boundaries(4);
7  if( nargin == 2 )
8      kappa_av = zeros(nargin);
9      x_un = ax:(bx-ax)/N:bx; y_un = ay:(by-ay)/K:by;
10     for n = 1:length(x_un)
11         for k = 1:length(y_un)
12             [~,~,~,kappa_av(n,k)] = coefficients(x_un(n),y_un(k));
13         end
14     end
15     kappa_av = sum(sum(kappa_av))/length(x_un)/length(y_un);
16 end
17 if( nargin == 3 )
18     az = boundaries(5); bz = boundaries(6);
19     kappa_av = zeros(nargin);
20     x_un = ax:(bx-ax)/N:bx; y_un = ay:(by-ay)/K:by; z_un = az:(bz-az)/L:bz;
21     for n = 1:length(x_un)
22         for k = 1:length(y_un)
23             for l = 1:length(z_un)
24                 [~,~,~,kappa_av(n,k,l)] = ...
25                     coefficients(x_un(n),y_un(k),z_un(l));
26             end
27         end
28     end
29     kappa_av = sum(sum(sum(kappa_av))) ...
30         /length(x_un)/length(y_un)/length(z_un);
31 end
32 f0 = @(x)( x/sinh(x) - (2/3)*mu/(mu+kappa_av) );
33 f1 = @(x)( ( sinh(x)-x*cosh(x) )/(sinh(x))^2 );
34 root = newton( f0, f1, 1 );
35 C = (3/8)*root; A = ( tanh(0.5*root) )^(-1);
36 x_auto = @(x)(A*tanh(C*x.*(1+x.^2/3)));
37 x_auto_diff = @(x)(A*C*cosh(C*x.*(1+x.^2/3)).^(-2).*(1+x.^2));
38 y_auto = @(x)(x_auto(x)); y_auto_diff = @(x)(x_auto_diff(x));
39 xi1 = -1:2/(N+1):1; xi1_semi = -N/(N+1):2/(N+1):N/(N+1);
40 xi2 = -1:2/(K+1):1; xi2_semi = -K/(K+1):2/(K+1):K/(K+1);
41 x = 0.5*(bx+ax)*ones(1,N+2) + 0.5*(bx-ax)*x_auto(xi1);
42 y = 0.5*(by+ay)*ones(1,K+2) + 0.5*(by-ay)*y_auto(xi2);
43 hx = 0.5*(bx-ax)*x_auto_diff( xi1_semi )/(N+1);
44 hy = 0.5*(by-ay)*y_auto_diff( xi2_semi )/(K+1);
45 if( nargin == 3 )
46     xi3 = -1:2/(L+1):1; xi3_semi = -L/(L+1):2/(L+1):L/(L+1);
47     z_auto = @(x)(x_auto(x)); z_auto_diff = @(x)(x_auto_diff(x));
```

```
48      z = 0.5*(bz+az)*ones(1,L+2) + 0.5*(bz-az)*z_auto(xi3);
49      hz = 0.5*(bz-az)*z_auto_diff( xi3_semi )/(L+1);
50  end
51  end
```

Листинг 6.44. Метод Ньютона `newton.m`

```
1  function [root, delta, l] = newton( f0, f1, initial_approx )
2  % Newton method for solving scalar non-linear equation.
3  global eps_grid_sys;
4  iter_num = 20; r = zeros(1,iter_num);
5  delta = 1; l = 1; r(1) = initial_approx;
6  while ( delta > min(1e-8,eps_grid_sys) )
7      r(l+1) = r(l) - f0( r(l) )/f1( r(l) );
8      delta = abs(r(l+1) - r(l));
9      if (l > iter_num); break; end
10     l = l+1;
11 end
12 root = r(l);
13 end
```

Листинг 6.45. Графики решения и график сходимости `pde_solve_illustration.m`

```
1  function pde_solve_illustration(u, N0, K0, L0, precision_output)
2  % Performs plots of the solution, its section and convergence for spatial
3  % meshes thickening.
4  global grid_choice;
5  dim = size(u);
6  if( length(dim) == 2 )
7      N = dim(1) - 2; K = dim(2) - 2;
8      if (grid_choice == 0)
9          [x,~,y,~] = adaptive_mesh( N,K );
10     else
11         [x,~,y,~] = user_mesh( N,K );
12     end
13     figure; hold on; grid off; colormap(gray); colorbar;
14     xlabel('x'); ylabel('y'); zlabel('u'); title('SOLUTION');
15     [y1,x1] = meshgrid(y,x);
16     surf(x1,y1,u); contour3(x1,y1,u,40,'k'); shading interp;
17     if (N0 == K0)
18         section = zeros(1,K+2); arg = section;
19         for k = 1:K+2
20             section(k) = u(k,k);
21             arg(k) = sqrt( (x(k)-min(x))^2+(y(k)-min(y))^2 );
22         end
23         figure; xlabel('sqrt(x^2 + y^2)'); ylabel('u'); title('SECTION');
24         hold on; plot(arg,section,'-ok','MarkerEdgeColor','k',...
25             'MarkerFaceColor','k','MarkerSize',4)
26     end
27     figure; xlabel('lg N'); ylabel('lg epsilon'); title('ACCURACY');
28     temp = (N0*K0)^(1/2); NN = zeros(1,length(precision_output));
```

```
29      for m = 1:length(precision_output)
30          NN(m) = temp*2^(m);
31      end
32      hold on; plot(log10(NN),log10(precision_output),...
33          '-ok','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',14)
34  end
35  if( length(dim) == 3 )
36      N = dim(1) - 2; K = dim(2) - 2; L = dim(3) - 2;
37      if (grid_choice == 0)
38          [x,~,y,~] = adaptive_mesh( N,K );
39      else
40          [x,~,y,~] = user_mesh( N,K );
41      end
42      figure; hold on; grid off; colormap(gray); colorbar;
43      xlabel('x'); ylabel('y'); zlabel('u'); title('SOLUTION, z = (bz+az)/2');
44      [y1,x1] = meshgrid(y,x); fixed_z = u(:,:,ceil(L/2+1));
45      surf(x1,y1,fixed_z); contour3(x1,y1,fixed_z,40,'k'); shading interp;
46      if (N0 == K0)
47          section = zeros(1,K+2); arg = section;
48          for k = 1:K+2
49              section(k) = fixed_z(k,k);
50              arg(k) = sqrt( (x(k)-min(x))^2+(y(k)-min(y))^2 );
51          end
52          figure; xlabel('sqrt(x^2 + y^2)'); ylabel('u'); title('SECTION');
53          hold on; plot(arg,section,'-ok','MarkerEdgeColor','k',...
54              'MarkerFaceColor','k','MarkerSize',4)
55      end
56      figure; xlabel('lg N'); ylabel('lg epsilon'); title('ACCURACY');
57      temp = (N0*K0*L0)^(1/2); NN = zeros(1,length(precision_output));
58      for m = 1:length(precision_output)
59          NN(m) = temp*2^(m);
60      end
61      hold on; plot(log10(NN),log10(precision_output),...
62          '-ok','MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',14)
63  end
64  end
```

7. Заключение

1. Разработаны и реализованы экономичные численные алгоритмы решения задач кинетики, диффузии и эффективный метод численного обнаружения и диагностики сингулярностей в ОДУ, работающий в автоматическом режиме. Разработан и успешно применен метод обработки экспериментальных данных с нахождением дисперсии аппроксимирующей кривой.
2. Разработан простой итерационный метод решения многомерных эллиптических уравнений с логарифмической сходимостью, что является теоретическим пределом. Одновременно с решением метод вычисляет асимптотически точную оценку погрешности. Метод позволяет эффективно решать сингулярно возмущенные уравнения.
3. Создано три пакета прикладных программ для решения указанных выше задач. Эффективность пакетов подтверждена численными экспериментами.
4. Разработаны новые математические методы моделирования основных ядерных реакций синтеза изотопов водорода, получены наиболее точные на настоящий момент аппроксимации сечений и скоростей реакций.

Список иллюстраций

Список таблиц

Список литературы

[1] Ракитский Ю. В., Устинов С. М., Черноруцкий И. Г. *Численные методы решения жестких систем.* М.: Наука, 1979.

[2] Хайрер Э., Ваннер Г. *Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи.* М.: Мир, 1999.

[3] Shampine L. F., Reichelt M. W. The Matlab ODE suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, 1997.

[4] Rosenbrock H. H. Some general implicit processes for the numerical solution of differential equations. *The Computer Journal*, 5(4):329–330, 1963.

[5] Гольдин В. Я., Калиткин Н. Н. Нахождение знакопостоянных решений обыкновенных дифференциальных уравнений. *Ж. вычисл. матем. и матем. физ.*, 6(1):162–163, 1966.

[6] Бракнер К., Джорна С. *Управляемый лазерный синтез.* М.: Атомиздат, 1977.

[7] Свешников А. Г., Альшин А. Б., Корпусов М. О., Плетнер Ю. Д. *Линейные и нелинейные уравнения соболевского типа.* М.: Физматлит, 2007.

[8] Самарский А. А., Галактионов В. А., Курдюмов С. П., Михайлов А. П. *Режимы с обострением в задачах для квазилинейных параболических уравнений.* М.: Наука, 1987.

[9] Калиткин Н. Н., Пошивайло И. П. Диагностика особенностей точного решения методом сгущения сеток. *ДАН. Информатика*, 404(3):295–299, 2005.

[10] Альшина Е. А., Калиткин Н. Н., Корякин П. В. Диагностика особенностей точного решения при расчетах с контролем точности. *Ж. вычисл. матем. и матем. физ.*, 45(10):1837–1847, 2005.

[11] Самарский А. А., Андреев В. Б. *Разностные методы для эллиптических уравнений.* М.: Наука, 1976.

[12] Самарский А. А., Николаев Е. С. *Методы решения сеточных уравнений.* М.: Наука, 1978.

[13] Фадеев Д. К., Фадеева В. Н. *Вычислительные методы линейной алгебры.* М.: Физматгиз, 1963.

[14] Калиткин Н. Н. *Численные методы.* М.: Наука, 1978.

[15] Бахвалов Н. С. *Численные методы. Т.1.* М.: Наука, 1973.

[16] Калиткин Н. Н. Улучшенная факторизация параболических схем. *ДАН. Информатика*, 402(4):467–471, 2005.

[17] Марчук Г. И. *Методы расщепления.* М.: Наука, 1988.

[18] Болтнев А. А., Калиткин Н. Н., Качер О. А. Логарифмически сходящийся счет на установление. *ДАН. Информатика*, 404(2):177–180, 2005.

[19] Toth C. D., O'Rourke R., Goodman J. E. *Handbook of discrete and computational geometry, 2nd edition.* London: Chapman and Hall/CRC, 2004.

[20] Бахвалов Н. С. К оптимизации методов решения краевых задач при наличии пограничного слоя. *Ж. вычисл. матем. и матем. физ.*, 9(4):841–859, 1969.

[21] Ершова Т. Я. О решении задачи Дирихле для сингулярно возмущенного уравнения реакции-диффузии в квадрате на сетке Бахвалова. *Вестн. Моск. Ун-та, Серия 15. Вычисл. матем. и киберн.*, (4):7–14, 2009.

[22] Шишкин Г. И. Аппроксимация решений сингулярно возмущенных краевых задач с угловым пограничным слоем. *Ж. вычисл. матем. и матем. физ.*, 27(9):1360–1372, 1987.

[23] Андреев В. Б. О точности сеточных аппроксимаций негладких решений сингулярно возмущенного уравнения реакции-диффузии в квадрате. *Дифференц. уравн.*, 42(7):895–906, 2009.

[24] NEA Data Bank – Nuclear Data Services. `http://www.oecd-nea.org/dbdata/`, 2011–2016.

[25] Arnold W. R., Phillips J. A., Sawyer G. A., Stovall E. J. (Jr.), Tuck J. L. Cross sections for the reactions D(d,p)T, D(d,n)^{3}He, T(d,n)^{4}He and ^{3}He(d,p)^{4}He below 120 kev. *Phys. Rev.*, 93(3):483–497, 1954.

[26] Козлов Б. Н. Скорости термоядерных реакций. *Атомная энергия*, 12(3):238–240, 1962.

[27] Krauss A., Becker H. W., Trautvetter H. P., Rolfs C., Brand K. Low energy fusion cross section of D+D and D+^{3}He reaction. *Nuclear Physics A*, 465(1):150–172, 1987.

[28] Brown R. E., Jarmie N. Differential cross sections at low energies for ^{2}H(d, p)^{3}H and ^{2}H(d, n)^{3}He. *Phys. Rev. C*, 41(4):1391–1400, 1990.

[29] Bretscher E., French A. P. Low energy cross section of the D – T reaction and angular distribution of the alpha-particles emitted. *Phys. Rev.*, 75(8):1154–1160, 1949.

[30] Wenzel W., Whaling W. A. Cross sections for the reactions D(d,p)T, D(d,n)^{3}He, t(d,n)^{4}He and ^{3}He(d,p)^{4}He below 120 kev. *Phys. Rev.*, 88(5):1149–1154, 1952.

[31] Гамов Г. А. Очерк развития учения о строении атомного ядра. Теория радиоактивного распада. *УФН*, 10(4):531–544, 1930.

[32] Argo H. V., Taschek R. F., Agnew H. M., Hemmendinder A., Leland W. T. Cross sections of the D(t,n)He^4 reaction for 80- to 1200-kev tritons. *Phys. Rev.*, 87(4):612–618, 1952.

[33] Conner J. P., Bonner T. W., Smith J. R. A study of the H^3(d, n)He^4 reaction. *Phys. Rev.*, 88(3):468–473, 1952.

[34] Bonner T. W., Conner J. P., Lillie A. B. Cross section and angular distribution of the He^3(d, n)He^4 nuclear reaction. *Phys. Rev.*, 88(3):473–476, 1952.

[35] Blair J. M., Hintz N. M., Van Patter D. M. Radiative capture of deuterons by He^3. *Phys. Rev.*, 96(4):1023–1029, 1954.

[36] Kunz W. E. Deuterium He^3 reaction. *Phys. Rev.*, 97(2):456–462, 1955.

[37] Jarmie N., Brown R. E., Hardekopf R. A. Fusion-energy reaction ^{2}H(t, α)n from E_t=12.5 to 117 kev. *Phys. Rev. C*, 29(6):2031–2046, 1984.

[38] Brown R. E., Jarmie N., Hale G. M. Fusion-energy reaction ^{3}H(d, α)n at low energies. *Phys. Rev. C*, 35(6):1999–2004, 1987.

[39] Breit G., Wigner E. Capture of slow neutrons. *Phys. Rev.*, 49(7):519–531, 1936.

[40] Давиденко В. А., Погребов И. С., Сауков А. И. Определение формы кривой возбуждения реакции T(d, n)^{4}He. *Атомная энергия*, 2(4):386–388, 1957.

[41] Долголева Г. В., Забродина Е. А. Сравнение двух моделей расчета термоядерной кинетики. *Препринты ИПМ им. М. В. Келдыша*, 1(68):1–14, 2014.

[42] Рябенький В. С., Филлипов А. Ф. *Об устойчивости разностных уравнений.* М.: Государственное изд-во технико-теоретической литературы, 1956.

[43] Шалашилин В. И., Кузнецов Е. Б. *Метод продолжения решения по параметру и наилучшая параметризация.* М.: Эдиториал УРСС, 1999.

[44] Richardson L. F., Gaunt J. A. The deferred approach to the limit. *Phil. Trans. A*, 226:299–349, 1927.

[45] Калиткин Н. Н., Альшин А. Б., Альшина Е. А., Рогов Б. В. *Вычисления на квазиравномерных сетках.* М.: Физматлит, 2005.

[46] Калиткин Н. Н., Пошивайло И. П. Гарантированная точность при решении задачи Коши методом длины дуги. *ДАН. Информатика*, 452(5):499–502, 2013.

[47] Калиткин Н. Н., Пошивайло И. П. Решение задачи Коши для жестких систем с гарантированной точностью методом длины дуги. *Матем. Моделирование*, 26(7):3–18, 2014.

[48] Lawson J. D. Some criteria for a power producing thermonuclear reactor. *Proceedings of the Physical Society. Section B*, 70(1):6, 1957.

[49] Яненко Н. Н. *Метод дробных шагов решения многомерных задач математической физики.* Новосибирск: Наука - Сибирское отделение, 1967.

[50] Самарский А. А. *Теория разностных схем.* М.: Наука, 1989.

[51] Калиткин Н. Н., Корякин П. В. *Численные методы. Т.2. Методы математической физики.* М.: Академия, 2013.

[52] Калиткин Н. Н., Юхно Л. Ф., Кузьмина Л. В. Количественный критерий обусловленности систем линейных алгебраических уравнений. *ДАН. Информатика*, 434(4):464–467, 2010.

[53] Калиткин Н. Н., Юхно Л. Ф., Кузьмина Л. В. Критерий обусловленности систем линейных алгебраических уравнений. *Матем. Моделирование*, 23(2):3–26, 2011.

[54] Калиткин Н. Н., Кузьмина Л. В. Аппроксимация и экстраполяция табулированных функций. *ДАН. Информатика*, 374(4):464–468, 2000.

[55] Калиткин Н. Н., Луцкий К. И. Оптимальные параметры метода двойного периода. *Матем. моделирование*, 19(1):57–68, 2007.

[56] Тихонов А. Н., Гончарский А. В., Степанов В. В., Ягола А. Г. *Численные методы решения некорректных задач.* М.: Наука, 1990.

[57] Калиткин Н. Н. Квадратуры Эйлера-Маклорена высоких порядков. *Матем. моделирование*, 16(10):64–66, 2004.