

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ И. Н. УЛЬЯНОВА»

На правах рукописи



КУВШИНОВА Анастасия Николаевна

**Параметрическая идентификация моделей  
конвективно-диффузионного переноса на основе  
рекуррентных алгоритмов дискретной фильтрации**

05.13.18 – Математическое моделирование, численные методы  
и комплексы программ

ДИССЕРТАЦИЯ

на соискание учёной степени  
кандидата физико-математических наук

Научный руководитель  
кандидат физико-математических наук,  
доцент Цыганов Андрей Владимирович

Ульяновск – 2022

## Оглавление

<b>Введение</b> . . . . .	4
<b>Глава 1. Дискретные модели конвективно-диффузионного переноса</b> .	13
1.1. Постановка задачи . . . . .	13
1.2. Случай граничных условий первого рода . . . . .	15
1.3. Случай смешанных граничных условий . . . . .	22
1.4. Анализ свойств дискретных моделей . . . . .	25
1.5. Заключение и выводы к главе 1 . . . . .	30
<b>Глава 2. Идентификация коэффициентов конвекции и диффузии</b> . . .	32
2.1. Постановка задачи . . . . .	32
2.2. Идентификация с использованием стандартного фильтра Калмана и безградиентных методов оптимизации . . . . .	36
2.3. Идентификация с использованием стандартного фильтра Калмана и градиентных методов оптимизации . . . . .	44
2.4. Идентификация на основе SVD-модификации фильтра Калмана .	47
2.5. Заключение и выводы к главе 2 . . . . .	62
<b>Глава 3. Идентификация граничных условий</b> . . . . .	64
3.1. Постановка задачи . . . . .	64
3.2. Идентификация на основе алгоритма одновременного оценивания состояния и входных воздействий . . . . .	65
3.3. Идентификация на основе квадратно-корневой модификации алгоритма одновременного оценивания состояния и входных воздействий . . . . .	83
3.4. Заключение и выводы к главе 3 . . . . .	89
<b>Глава 4. Программный комплекс для моделирования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса</b> . . . . .	90

4.1. Описание программного комплекса . . . . .	90
4.2. Примеры решения задач идентификации . . . . .	105
4.3. Заключение и выводы к главе 4 . . . . .	127
<b>Заключение . . . . .</b>	<b>128</b>
<b>Список литературы . . . . .</b>	<b>130</b>
<b>Список иллюстративного материала . . . . .</b>	<b>141</b>
<b>Список таблиц . . . . .</b>	<b>143</b>
<b>Список алгоритмов . . . . .</b>	<b>144</b>
<b>Приложение А. Скрипты Maple для автоматической проверки свойства полной наблюдаемости . . . . .</b>	<b>145</b>
<b>Приложение Б. Скрипты Maple для автоматической проверки свойства полной управляемости . . . . .</b>	<b>147</b>
<b>Приложение В. Метаэвристические алгоритмы оптимизации . . . . .</b>	<b>150</b>
<b>Приложение Г. Свидетельство о регистрации программы для ЭВМ . . . . .</b>	<b>152</b>

## Введение

### Актуальность работы

Математические модели тепломассопереноса широко используются для описания различных природных и техногенных процессов [1, 2, 3]. На протяжении последних десятилетий в силу важности приложений для данных моделей актуальными являются задачи идентификации их параметров по результатам измерений значений искомой функции в рассматриваемой области. Такого рода задачи называются обратными и в общем случае они относятся к некорректно поставленным задачам математической физики [4].

Традиционные методы, используемые для решения обратных задач, описаны в монографиях А. А. Самарского, О. В. Алифанова, А. Н. Тихонова и других авторов (см., например, [5, 6, 7, 8, 9]). Наряду с данными методами в работах отечественных и зарубежных авторов (Д. Ф. Симбирский, Ю. М. Мацевитый, J. K. Hodge и др.) для решения обратных задач теплопроводности было предложено использовать методы параметрической идентификации, основанные на применении рекуррентных алгоритмов дискретной фильтрации.

Как показывает анализ литературы, данные методы чаще всего применяются для определения параметров моделей теплопереноса по экспериментальным данным и преимущественно основаны на использовании расширенного фильтра Калмана или его модификаций, а их практическое применение подразумевает широкое использование ЭВМ и численных методов. Результаты практического использования методов рекуррентной параметрической идентификации показывают их адекватность и устойчивость к наличию шумов в наблюдаемых данных [10]. Кроме того, использование рекуррентных алгоритмов идентификации в измерительных системах реального времени зачастую предпочтительнее по причине их более высокого быстродействия по сравнению с традиционными методами, требующими большого объема вычислений [11].

Однако, как показывает практика решения различных прикладных задач, результаты использования программных реализаций алгоритмов калмановской фильтрации могут значительно отличаться от теоретически предсказанных. В ра-

ботах ряда исследователей (например, [12, 13, 14]) отмечается, что это во многом обусловлено численной неустойчивостью классических алгоритмов калмановской фильтрации по отношению к ошибкам машинного округления. В связи с чем чрезвычайно важной является разработка новых робастных (то есть численно устойчивых к ошибкам машинного округления) методов дискретной фильтрации.

Среди других факторов, влияющих на качество оценок, получаемых фильтром Калмана, можно отметить [14]:

- неопределенность в значениях матричных параметров, поскольку наличие ошибок в модели не учитывается при выводе фильтра Калмана;
- большой диапазон значений матричных параметров, измерений или переменных состояния, что может привести к трудностям с выбором масштаба или единиц измерения;
- плохо обусловленные теоретические решения матричного уравнения Риккати, что при наличии ошибок численного решения может привести к ошибкам оценивания;
- большие размеры матриц, поскольку число арифметических операций возрастает пропорционально квадрату или кубу размерностей матриц и при каждой операции могут возникать ошибки округления;
- низкая машинная точность, увеличивающая относительные ошибки округления.

В связи с этим применение рекуррентных методов параметрической идентификации требует проведения исследований в части устойчивости, сходимости и оценки погрешностей получаемых результатов, но теоретическое сравнение численной устойчивости различных методов может быть затруднительным. Хотя одни методы считаются более робастными, чем другие, на практике поведение методов зачастую зависит от свойств решаемой задачи. Поэтому важной является задача разработки программного обеспечения для проведения комплексных исследований свойств разрабатываемых методов.

Все вышесказанное обосновывает *актуальность* темы диссертационного исследования.

### **Степень разработанности темы исследования**

Первые применения рекуррентных методов параметрической идентификации для решения обратных задач теплопереноса содержатся в работах Д. Ф. Симбирского, Ю. М. Мацевитого, А. В. Мултановского, А. В. Олейника, J. K. Hodge, A. J. Chen, J. R. Hayes, N. Daouas, M. S. Radhouani, A. A. Карпова, Т. А. Тихоновой и ряда других авторов. Данные методы применялись при решении задач температурной диагностики двигателей и паровых турбин [15, 16, 17, 18, 19], а в последствии при решении различных обратных задач теплопроводности [10, 20, 21, 22, 23] и нестационарной теплотометрии [24, 25, 11].

В работах [26, 27, 28] приводятся примеры применения различных модификаций фильтра Калмана и фильтра частиц к решению обратных задач теплопереноса. В [29, 30] фильтр Калмана и фильтр частиц применяются к решению обратных задач теплопереноса в трубопроводах. В [31] рассматривается применение трех модификаций фильтра частиц для решения обратных задач теплопереноса, возникающих при лечении онкологических заболеваний методом гипертермии. В [32] рассматривается задача идентификации параметров математической модели теплопереноса с помощью сигма-точечного фильтра Калмана.

Рекуррентные методы параметрической идентификации применимы к широкому классу моделей динамических систем, для которых характерны процессы конвекции (адвекции) и диффузии [33]. Например, в работах [34, 35] описывается применение расширенного фильтра Калмана в сочетании с МНК для идентификации коэффициентов конвекции и диффузии одномерной модели конвективно-диффузионного переноса.

### **Объект и предмет исследования**

*Объектом* исследования являются математические модели конвективно-диффузионного переноса, представленные дискретными линейными стохастическими системами в пространстве состояний. *Предмет* исследования — методы идентификации неизвестных параметров рассматриваемых моделей на основе

рекуррентных алгоритмов дискретной фильтрации.

### **Цель и задачи диссертационной работы**

*Целью* диссертационной работы является построение и исследование математических моделей конвективно-диффузионного переноса, описываемых дискретными линейными стохастическими системами в пространстве состояний, и разработка новых методов параметрической идентификации моделей конвективно-диффузионного переноса по данным зашумленных измерений на основе рекуррентных алгоритмов дискретной фильтрации.

Для достижения поставленных целей решались следующие *задачи*:

1. Построение математических моделей конвективно-диффузионного переноса, описываемых дискретными линейными стохастическими системами в пространстве состояний.
2. Анализ свойств построенных моделей.
3. Разработка алгоритмов идентификации коэффициентов конвекции и диффузии уравнения конвективно-диффузионного переноса.
4. Разработка алгоритмов численной идентификации граничных условий моделей конвективно-диффузионного переноса.
5. Реализация разработанных методов в виде комплекса программ для исследования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса.

### **Научная новизна**

Основные результаты, полученные в работе, являются новыми. В частности, построены и исследованы новые модели конвективно-диффузионного переноса, описываемые дискретными линейными стохастическими системами в пространстве состояний. Для построенных моделей разработаны новые методы идентификации коэффициентов конвекции и диффузии на основе рекуррентных алгоритмов дискретной фильтрации. Получены выражения для вычисления градиента критерия идентификации для моделей конвективно-диффузионного переноса на основе стандартного фильтра Калмана. Сформулирован критерий идентификации на основе SVD-модификации фильтра Калмана, обладающий улуч-

шенными вычислительными свойствами по сравнению с критерием на основе стандартного фильтра Калмана. Разработаны методы идентификации граничных условий моделей конвективно-диффузионного переноса на основе алгоритмов одновременного оценивания векторов состояния и входных воздействий Гиллейнса – Де-Мора. Сформулирована квадратно-корневая модификация алгоритма Гиллейнса – Де-Мора в ковариационной форме. Разработан комплекс программ для исследования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса.

### **Теоретическая и практическая значимость**

Теоретическая значимость работы состоит в разработке новых методов и средств параметрической идентификации математических моделей процессов конвективно-диффузионного переноса. Практическая значимость работы заключается в возможности использования результатов диссертационного исследования для решения прикладных задач тепломассопереноса.

**Достоверность и обоснованность результатов** обеспечена строгими математическими доказательствами теоретических результатов, корректным применением математического аппарата и численных методов, использованием современных методов разработки алгоритмов, сравнением результатов с результатами, полученными альтернативными методами, а также использованием при разработке программного комплекса апробированного лицензионного и свободно распространяемого программного обеспечения. Все теоретические положения подтверждены результатами вычислительных экспериментов.

### **Внедрение**

Результаты диссертационного исследования используются в учебном процессе ФГБОУ ВО «Ульяновский государственный педагогический университет им. И. Н. Ульянова» при изучении дисциплин «Компьютерное моделирование в научных исследованиях» и «Математическое моделирование в естественно-научных и социально-гуманитарных исследованиях», читаемых студентам магистратуры факультета физико-математического и технологического образования, обучающимся по профилям «Методология математического образования»

и «Компьютерное моделирование и дизайн информационной образовательной среды».

### **Методы исследования**

Теоретические результаты диссертации получены с применением методов вычислительной линейной алгебры, теории линейных дискретных динамических систем, теории адаптивной и оптимальной дискретной фильтрации, численных методов оптимизации. Для получения практических результатов использованы методы математического моделирования, математической статистики и современные технологии научного программирования.

### **Положения, выносимые на защиту:**

1. Математические модели конвективно-диффузионного переноса, описываемые дискретными линейными стохастическими системами в пространстве состояний.
2. Методы идентификации коэффициентов конвекции и диффузии с использованием безградиентных и градиентных алгоритмов минимизации критериев идентификации. Градиент критерия идентификации для моделей конвективно-диффузионного переноса на основе стандартного фильтра Калмана. Критерий идентификации на основе SVD-модификации фильтра Калмана.
3. Методы идентификации граничных условий моделей конвективно-диффузионного переноса на основе алгоритмов одновременного оценивания векторов состояния и входных воздействий Гиллейнса – Де-Мора. Квадратно-корневая модификация алгоритма Гиллейнса – Де-Мора в ковариационной форме.
4. Комплекс программ для исследования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса с реализацией разработанных методов на языке MATLAB.

### **Соответствие диссертации паспорту научной специальности**

Содержание диссертации соответствует п. 3 «Разработка, обоснование и тестирование эффективных вычислительных методов с применением современ-

ных компьютерных технологий», п. 4 «Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента» и п. 5 «Комплексные исследования научных и технических проблем с применением современной технологии математического моделирования и вычислительного эксперимента» паспорта специальности научных работников 05.13.18 — «Математическое моделирование, численные методы и комплексы программ» по физико-математическим наукам.

### **Апробация результатов**

Основные результаты диссертации докладывались и обсуждались на следующих конференциях, семинарах и научных мероприятиях:

- Международная конференция «Дифференциальные уравнения, математическое моделирование и вычислительные алгоритмы», Белгород, 25–29 октября 2021 г.
- Научно-техническая конференция «Интегрированные системы управления», Ульяновск, 18–19 мая 2021 г.
- XXVIII Международная конференция студентов, аспирантов и молодых учёных «Ломоносов», Ульяновск, 12–23 апреля 2021 г. (доклад «Адаптивное оценивание дискретной модели конвективно-диффузионного переноса на основе численно эффективных алгоритмов Калмановской фильтрации» удостоен диплома II степени).
- Всероссийская научно-практическая конференция с международным участием «Функциональный анализ и математическое образование» (ФАМО-2020), Ульяновск, 8–9 октября 2020 г.
- VI международная конференция и молодежная школа «Информационные технологии и нанотехнологии» (ИТНТ-2020), Самара, 26–29 мая 2020 г.
- V международная конференция и молодежная школа «Информационные технологии и нанотехнологии» (ИТНТ-2019), Самара, 21–24 мая 2019 г. (доклад «Dynamic identification of boundary conditions for convective-diffusion transport model in the case of noisy measurements» удостоен диплома III

степени в конкурсе «Best Paper Award»).

- II Всероссийская научная конференция с международным участием «Информационные технологии в моделировании и управлении: подходы, методы, решения», Тольятти, 22–24 апреля 2019 г.
- Вторая международная научно-практическая конференция «Нечеткие системы и мягкие вычисления. Промышленные применения» (FTI-2018), Ульяновск, 23–25 октября 2018 г.

Исследования по теме диссертации частично поддержаны грантом Российского фонда фундаментальных исследований (проект: р\_а № 19-41-730009) и внутривузовским грантом УлГПУ им. И. Н. Ульянова (приказ № 133 от 31 мая 2019 г.).

#### **Публикации по теме исследования**

По теме исследования опубликованы 16 печатных работ, в том числе 3 статьи в изданиях, входящих в перечень ВАК РФ, 4 статьи в изданиях, индексируемых в международных системах цитирования Scopus и Web of Science, 3 свидетельства о регистрации программы для ЭВМ.

#### **Личный вклад автора**

Все основные теоретические результаты диссертационного исследования получены соискателем самостоятельно. Разработка программного комплекса осуществлялась совместно с научным руководителем. Постановка и проведение вычислительных экспериментов выполнены соискателем также самостоятельно. Анализ результатов осуществлялся совместно с научным руководителем.

#### **Структура и объём диссертации**

Диссертация состоит из введения, четырех глав, заключения, списка литературы, списка иллюстративного материала, списка таблиц, списка алгоритмов и четырех приложений. Общий объём диссертации 152 страницы, включая 62 рисунка и 16 таблиц. Основной текст диссертации составляет 140 страниц. Список литературы состоит из 90 наименований.

Во **введении** приводится обоснование актуальности темы исследования, формулируются цель, задачи, объект и предмет исследования, научная новизна,

положения, выносимые на защиту; описываются теоретическая и практическая значимость работы, внедрение результатов, методология и методы исследования, степень достоверности полученных результатов; приводятся сведения об апробации работы и личном вкладе автора.

В **первой** главе выполняется построение дискретных линейных стохастических моделей конвективно-диффузионного переноса в пространстве состояний, анализируются свойства полной наблюдаемости и управляемости полученных моделей.

Во **второй** главе рассматривается задача идентификации неизвестных коэффициентов уравнения конвекции-диффузии по данным зашумленных измерений: формулируются алгоритмы вычисления критериев идентификации на основе стандартного фильтра Калмана и его SVD-модификации, получаются формулы для вычисления градиента критерия идентификации на основе стандартного фильтра Калмана, приводятся примеры идентификации с использованием различных безградиентных и градиентных алгоритмов минимизации.

В **третьей** главе рассматриваются методы численной идентификации граничных условий моделей конвективно-диффузионного переноса на основе алгоритмов одновременного оценивания векторов состояния и входных воздействий Гиллейнса – Де-Мора, формулируется квадратно-корневая модификация алгоритма Гиллейнса – Де-Мора в ковариационной форме и доказывается ее эквивалентность исходному алгоритму.

В **четвертой** главе приводится описание комплекса программ для исследования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса.

## Глава 1

# Дискретные модели конвективно-диффузионного переноса

## 1.1. Постановка задачи

Пусть дана одномерная математическая модель конвективно-диффузионного переноса, описываемая уравнением

$$\frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} = \alpha \frac{\partial^2 c}{\partial x^2}, \quad (1.1)$$

с начальным условием

$$c(x, 0) = \varphi(x) \quad (1.2)$$

и граничными условиями

$$c(a, t) = f(t), \quad c(b, t) = g(t) \quad (1.3)$$

или

$$c(a, t) = f(t), \quad \frac{\partial c(b, t)}{\partial x} = -\lambda[c(b, t) - g(t)], \quad (1.4)$$

где  $x \in [a; b]$  — пространственная координата;  $t \in [0; T]$  — время;  $c(x, t)$  — искомая функция, например, концентрация или температура некоторого вещества в точке с координатой  $x$  в момент времени  $t$ ;  $v$  — скорость конвекции;  $\alpha$  — коэффициент диффузии (единицы измерения величин зависят от типа решаемой задачи). Таким образом, в данной работе будут рассматриваться модели либо с двумя граничными условиями первого рода (1.3), либо со смешанными граничными условиями первого и третьего рода (1.4).

В работе будут рассмотрены следующие задачи параметрической идентификации: 1) определение коэффициентов  $v$  и  $\alpha$  уравнения (1.1), 2) определение функций  $f(t)$  и  $g(t)$ , входящих в граничные условия (1.3) или (1.4), по данным зашумленных измерений значений функции  $c(x, t)$  в дискретные моменты времени. Более подробно соответствующие задачи будут сформулированы в главах 2 и 3.

Для решения данных задач в работе будут использованы методы теории оптимального оценивания [36] для дискретных линейных стохастических систем в пространстве состояний следующего вида:

$$\begin{cases} c_k = F_{k-1}c_{k-1} + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}, & (1.5) \\ z_k = H_k c_k + \xi_k, \quad k = 1, 2, \dots, & (1.6) \end{cases}$$

где (1.5) — уравнение (модель) объекта/процесса,  $c_k \in \mathbb{R}^n$  — вектор состояния системы,  $u_k \in \mathbb{R}^r$  — вектор входных воздействий (управления),  $w_k \in \mathbb{R}^q$  — шум в объекте,  $F_k \in \mathbb{R}^{n \times n}$  — переходная матрица состояния,  $B_k \in \mathbb{R}^{n \times r}$  — матрица управления,  $G_k \in \mathbb{R}^{n \times q}$  — передаточная матрица шума; (1.6) — уравнение (модель) зашумленных измерений,  $z_k \in \mathbb{R}^m$  — вектор измерений,  $\xi_k \in \mathbb{R}^m$  — шум в измерителе,  $H_k \in \mathbb{R}^{m \times n}$  — матрица измерений;  $c_0 \sim \mathcal{N}(\bar{c}_0, \Pi_0)$ ; шумы  $w_k$  и  $\xi_k$  образуют независимые нормально распределенные последовательности с нулевым математическим ожиданием и ковариационными матрицами  $Q_k \geq 0$  и  $R_k > 0$ .

Уравнение объекта в общем случае представляет собой дискретную линейную динамическую систему, записанную в матричном виде. В уравнении измерений матрица  $H_k$  отвечает за выбор измеряемых компонент вектора состояния. В зависимости от решаемой задачи параметрической идентификации вектор  $u_k$  будет являться известным или неизвестным. Наряду с (1.5) будем также рассматривать уравнение объекта вида

$$c_k = F_{k-1}c_{k-1} + B_{k-1}^1 u_{k-1}^1 + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}, \quad (1.7)$$

в котором вектор входных воздействий  $u_{k-1}^1$  будет считаться известным, а вектор  $u_{k-1}$  — неизвестным.

При формулировке алгоритмов рекуррентной параметрической идентификации стандартным приемом является дискретизация исходной непрерывной модели, а окончательный вид уравнения объекта зависит как от вида уравнения (1.1), так и от типа рассматриваемых граничных условий. В разделах 1.2 и 1.3 будет рассмотрен переход от непрерывных моделей конвективно-диффузионного переноса к моделям, описываемым дискретными линейными стохастическими системами в пространстве состояний. Основной задачей будет являться

получение уравнений объекта вида (1.5) или (1.7) в зависимости от типа граничных условий. Описание процесса построения моделей содержится в работах [37, 38, 39, 40].

Важными свойствами моделей (1.5), (1.6) являются свойства полной наблюдаемости и полной управляемости [41]. В разделе 1.4 будет выполнен анализ свойств полной наблюдаемости и полной управляемости полученных дискретных стохастических систем. Анализ построенных моделей выполнен в работах [42, 43].

## 1.2. Случай граничных условий первого рода

Зададим в рассматриваемой пространственно-временной области конечно-разностную сетку  $\{(x_i, t_k) | i = 0, 1, \dots, N, k = 0, 1, \dots, K\}$ , где

$$x_i = a + i\Delta x, t_k = k\Delta t, \quad \Delta x = \frac{b-a}{N}, \Delta t = \frac{T}{K}.$$

Обозначим:  $c_i^k = c(x_i, t_k)$ ,  $\varphi_i = \varphi(x_i)$ ,  $f^k = f(t_k)$ ,  $g^k = g(t_k)$ . Заменяя частные производные в уравнении (1.1) их конечно-разностными аппроксимациями, в случае граничных условий (1.3) получим следующую систему уравнений

$$\frac{c_i^k - c_i^{k-1}}{\Delta t} + v \frac{c_{i+1}^{k-1} - c_{i-1}^{k-1}}{2\Delta x} = \alpha \frac{c_{i+1}^{k-1} - 2c_i^{k-1} + c_{i-1}^{k-1}}{\Delta x^2}, \quad (1.8)$$

$$i = 1, 2, \dots, N-1, k = 1, 2, \dots, K,$$

$$c_i^0 = \varphi_i, \quad i = 0, 1, \dots, N,$$

$$c_0^k = f^k, \quad c_N^k = g^k, \quad k = 0, 1, \dots, K.$$

Из (1.8) следует, что значение функции  $c(x, t)$  в узловой точке  $k$ -го временного ряда может быть выражено через ее значения в трех точках  $(k-1)$ -го временного ряда:

$$c_i^k = (r_1 + r_2)c_{i-1}^{k-1} + (1 - 2r_2)c_i^{k-1} + (r_2 - r_1)c_{i+1}^{k-1}, \quad (1.9)$$

где  $r_1 = \frac{v\Delta t}{2\Delta x}$ ,  $r_2 = \frac{\alpha\Delta t}{\Delta x^2}$ .

Перепишем (1.9) в виде

$$c_i^k = a_1 c_{i-1}^{k-1} + a_2 c_i^{k-1} + a_3 c_{i+1}^{k-1}, \quad i = 1, 2, \dots, N-1, k = 1, 2, \dots, K,$$

где  $a_1 = r_1 + r_2$ ,  $a_2 = 1 - 2r_2$ ,  $a_3 = r_2 - r_1$ . Тогда исконая дискретная линейная система может быть записана в следующем виде:

$$\underbrace{\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ \vdots \\ c_{n-2}^k \\ c_{n-1}^k \\ c_n^k \end{bmatrix}}_{c_k} = \underbrace{\begin{bmatrix} a_2 & a_3 & 0 & \dots & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & \dots & 0 & 0 & 0 \\ 0 & a_1 & a_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \dots & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & \dots & 0 & a_1 & a_2 \end{bmatrix}}_{F_{k-1}} \underbrace{\begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ \vdots \\ c_{n-2}^{k-1} \\ c_{n-1}^{k-1} \\ c_n^{k-1} \end{bmatrix}}_{c_{k-1}} + \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & a_3 \end{bmatrix}}_{B_{k-1}} \underbrace{\begin{bmatrix} f^{k-1} \\ g^{k-1} \end{bmatrix}}_{u_{k-1}}, \quad (1.10)$$

$k = 1, 2, \dots, K.$

В полученной системе компоненты вектора состояния  $c_k$  соответствуют всем внутренним узлам пространственной сетки ( $n = N - 1$ ), а вектор входных воздействий  $u_k$  является двумерным ( $r = 2$ ).

**Замечание 1.1.** Матрицы  $F_k$  и  $B_k$  являются постоянными, поскольку коэффициенты  $r_1$ ,  $r_2$ ,  $a$ , следовательно,  $a_1$ ,  $a_2$  и  $a_3$  не зависят от  $k$ .

Если левое граничное условие известно, а правое неизвестно, то система (1.10) может быть записана следующим образом:

$$\underbrace{\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ \vdots \\ c_{n-2}^k \\ c_{n-1}^k \\ c_n^k \end{bmatrix}}_{c_k} = \underbrace{\begin{bmatrix} a_2 & a_3 & 0 & \dots & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & \dots & 0 & 0 & 0 \\ 0 & a_1 & a_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \dots & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & \dots & 0 & a_1 & a_2 \end{bmatrix}}_{F_{k-1}} \underbrace{\begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ \vdots \\ c_{n-2}^{k-1} \\ c_{n-1}^{k-1} \\ c_n^{k-1} \end{bmatrix}}_{c_{k-1}} + \underbrace{\begin{bmatrix} a_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_{k-1}^1} \underbrace{f^{k-1}}_{u_{k-1}^1} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ a_3 \end{bmatrix}}_{B_{k-1}} \underbrace{g^{k-1}}_{u_{k-1}}, \quad (1.11)$$

$k = 1, 2, \dots, K.$

Если правое граничное условие известно, а левое неизвестно, то система

(1.10) может быть записана аналогично (1.11), но с другими обозначениями:

$$\underbrace{\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ \vdots \\ c_{n-2}^k \\ c_{n-1}^k \\ c_n^k \end{bmatrix}}_{c_k} = \underbrace{\begin{bmatrix} a_2 & a_3 & 0 & \dots & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & \dots & 0 & 0 & 0 \\ 0 & a_1 & a_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \dots & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & \dots & 0 & a_1 & a_2 \end{bmatrix}}_{F_{k-1}} \underbrace{\begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ \vdots \\ c_{n-2}^{k-1} \\ c_{n-1}^{k-1} \\ c_n^{k-1} \end{bmatrix}}_{c_{k-1}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ a_3 \end{bmatrix}}_{B_{k-1}^1} + \underbrace{\begin{bmatrix} a_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_{k-1}} \underbrace{\left( \underbrace{g^{k-1}}_{u_{k-1}^1} + \underbrace{f^{k-1}}_{u_{k-1}} \right)}_{(1.12)}$$

$$k = 1, 2, \dots, K.$$

К полученным системам добавим уравнение зашумленных измерений вида (1.6):

$$z_k = H_k c_k + \xi_k, \quad k = 1, 2, \dots, K, \quad (1.13)$$

где вид матрицы измерений  $H_k$  и характеристики шума  $\xi_k$  зависят от используемых сенсоров. В результате получим искомую дискретную линейную стохастическую систему.

**Замечание 1.2.** В работе будут рассматриваться только постоянные матрицы измерений  $H_k$ . Таким образом, построенные нами системы будут являться дискретными инвариантными во времени линейными системами.

Рассмотрим переход к дискретной линейной стохастической системе на примере следующей модели.

**Пример 1.1.**

$$\frac{\partial c}{\partial t} + 2 \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}, \quad (1.14)$$

$$c(x, 0) = 0, \quad (1.15)$$

$$c(0, t) = 4t |\sin 10t|, \quad c(1, t) = 0, \quad (1.16)$$

$$x \in [0; 1], \quad t \in [0; 1].$$

Зададим в пространственно-временной области  $[0; 1] \times [0; 1]$  конечно-разностную сетку с 6 узлами по оси  $Ox$  и 101 узлом по оси  $Ot$  (то есть  $N = 5$  и  $K = 100$ ), тогда  $\Delta x = 0.2$ ,  $\Delta t = 0.01$ , а вектор состояния будет состоять из 4 внутренних узлов пространственной сетки ( $n = 4$ ).

На рисунке 1.1 приведен график решения прямой задачи (1.14)–(1.16) методом конечных разностей на построенной сетке.

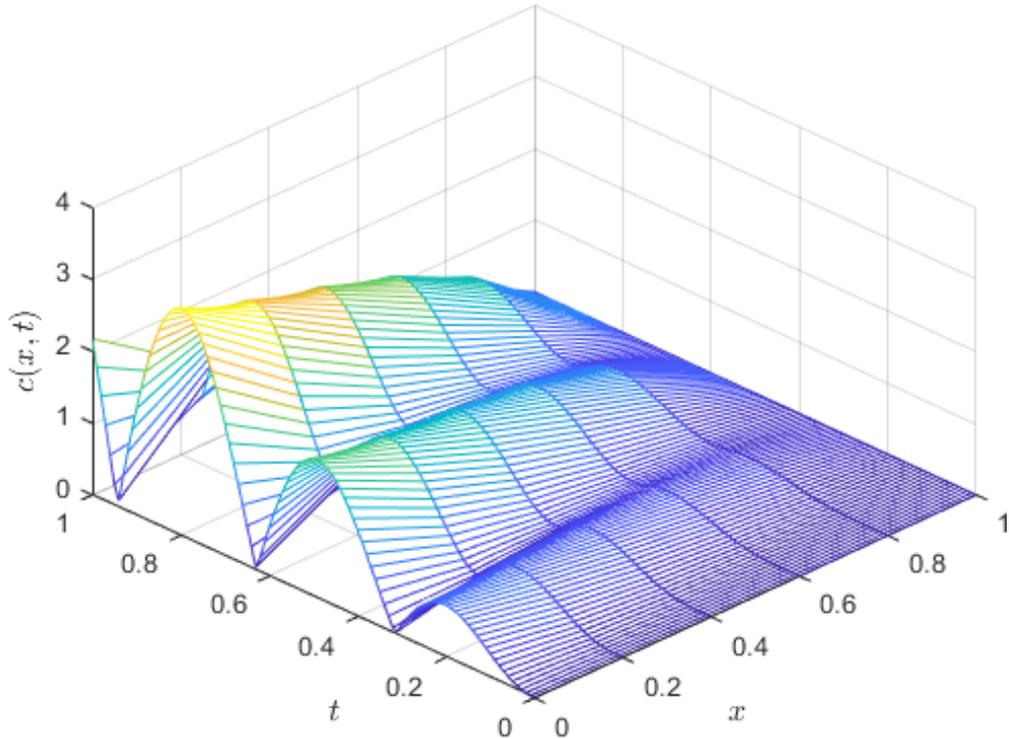


Рис. 1.1. Решение задачи

Система (1.10) в этом случае имеет вид:

$$\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \end{bmatrix} = \begin{bmatrix} 0.5 & 0.2 & 0 & 0 \\ 0.3 & 0.5 & 0.2 & 0 \\ 0 & 0.3 & 0.5 & 0.2 \\ 0 & 0 & 0.3 & 0.5 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \end{bmatrix} + \begin{bmatrix} 0.3 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.2 \end{bmatrix} \begin{bmatrix} f^{k-1} \\ g^{k-1} \end{bmatrix}, \quad (1.17)$$

$$k = 1, 2, \dots, 100,$$

а системы (1.11) и (1.12) записываются следующим образом:

$$\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \end{bmatrix} = \begin{bmatrix} 0.5 & 0.2 & 0 & 0 \\ 0.3 & 0.5 & 0.2 & 0 \\ 0 & 0.3 & 0.5 & 0.2 \\ 0 & 0 & 0.3 & 0.5 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0 \\ 0 \\ 0 \end{bmatrix} f^{k-1} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.2 \end{bmatrix} g^{k-1}, \quad (1.18)$$

$$k = 1, 2, \dots, 100,$$

где  $f^{k-1} = 4(k-1)\Delta t |\sin 10(k-1)\Delta t|$ ,  $g^{k-1} = 0$ .

К уравнениям объекта (1.17) или (1.18) могут быть добавлены различные модели зашумленных измерений. Рассмотрим три таких модели:

*Модель 1.*

$$\begin{bmatrix} z_1^k \\ z_2^k \\ z_3^k \\ z_4^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \end{bmatrix} + \begin{bmatrix} \xi_1^k \\ \xi_2^k \\ \xi_3^k \\ \xi_4^k \end{bmatrix}, \quad k = 1, 2, \dots, 100.$$

В данной модели измеряются значения функции в узлах пространственной сетки, соответствующих всем компонентам вектора состояния ( $x = 0.2, 0.4, 0.6, 0.8$ ). На рисунке 1.2 приведен график зашумленных измерений с ковариационной матрицей шума  $R_k = 0.05^2 I_4$ .

*Модель 2.*

$$\begin{bmatrix} z_1^k \\ z_2^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \end{bmatrix} + \begin{bmatrix} \xi_1^k \\ \xi_2^k \end{bmatrix}, \quad k = 1, 2, \dots, 100.$$

В данной модели измеряются значения функции в узлах пространственной сетки, соответствующих первой и последней компонентам вектора состояния ( $x = 0.2, 0.8$ ). На рисунке 1.3 приведен график зашумленных измерений с ковариационной матрицей шума  $R_k = 0.05^2 I_2$ .

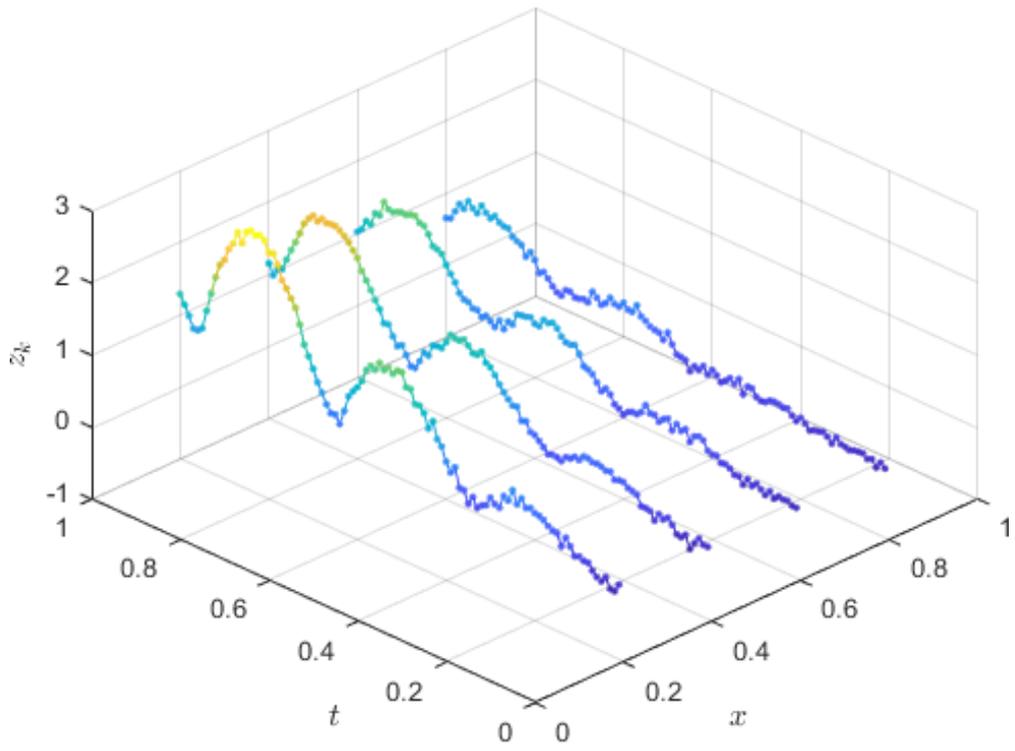


Рис. 1.2. Зашумленные измерения (модель 1)

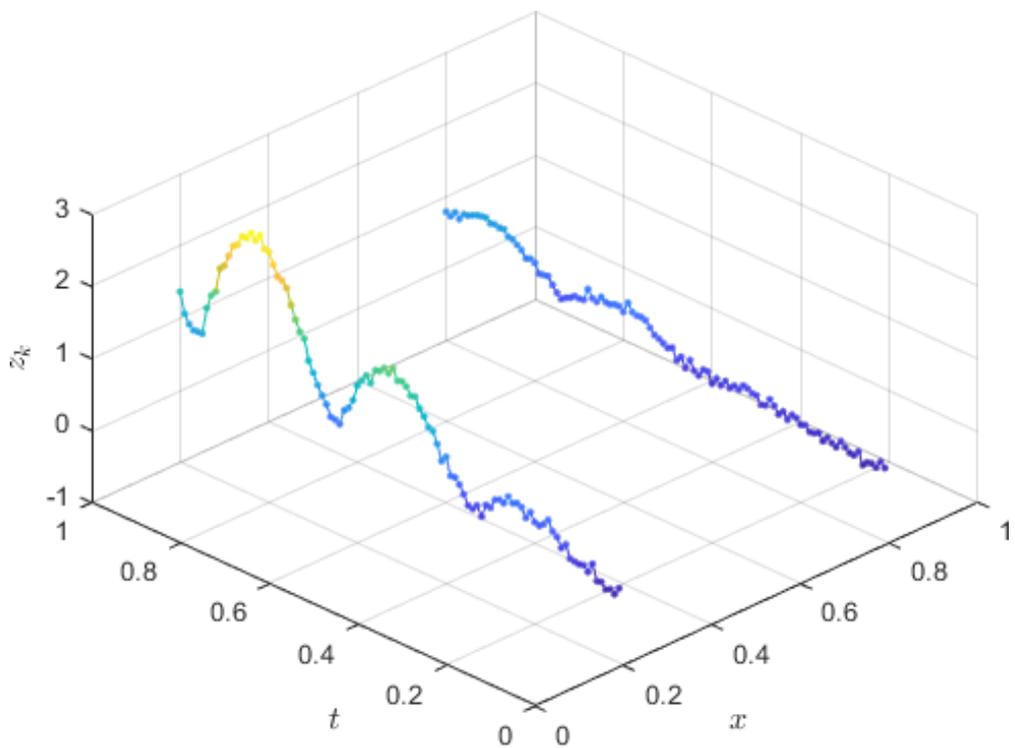


Рис. 1.3. Зашумленные измерения (модель 2)

Модель 3.

$$z_1^k = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \end{bmatrix} + \xi_1^k, \quad k = 1, 2, \dots, 100.$$

В данной модели измеряются значения функции в узле пространственной сетки, соответствующем первой компоненте вектора состояния ( $x = 0.2$ ). На рисунке 1.4 приведен график зашумленных измерений с ковариацией шума  $R_k = 0.05^2$ .

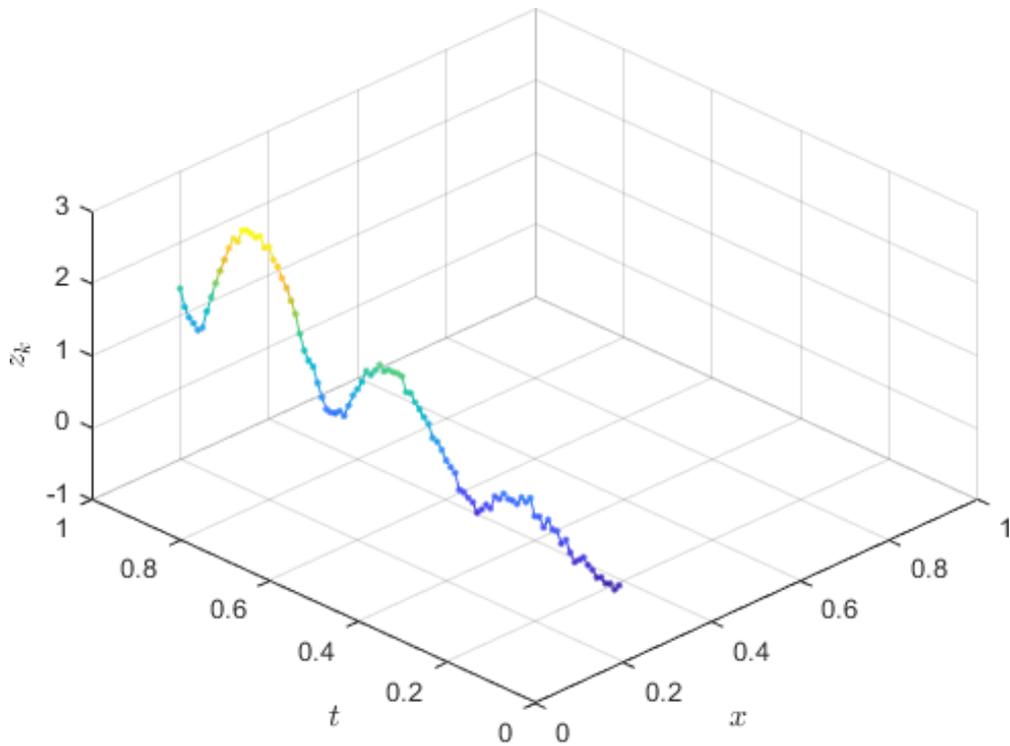


Рис. 1.4. Зашумленные измерения (модель 3)

### 1.3. Случай смешанных граничных условий

Рассуждая как в предыдущем разделе, в случае граничных условий (1.4) получаем систему уравнений

$$\begin{aligned} \frac{c_i^k - c_i^{k-1}}{\Delta t} + v \frac{c_{i+1}^{k-1} - c_{i-1}^{k-1}}{2\Delta x} &= \alpha \frac{c_{i+1}^{k-1} - 2c_i^{k-1} + c_{i-1}^{k-1}}{\Delta x^2}, \\ i &= 1, 2, \dots, N-1, \quad k = 1, 2, \dots, K, \\ c_i^0 &= \varphi_i, \quad i = 0, 1, \dots, N, \\ c_0^k &= f^k, \quad \frac{c_N^k - c_{N-1}^k}{\Delta x} = -\lambda[c_N^k - g^k], \quad k = 0, 1, \dots, K, \end{aligned} \quad (1.19)$$

откуда

$$c_i^k = a_1 c_{i-1}^{k-1} + a_2 c_i^{k-1} + a_3 c_{i+1}^{k-1}, \quad i = 1, 2, \dots, N-1, \quad k = 1, 2, \dots, K.$$

где  $a_1 = r_1 + r_2$ ,  $a_2 = 1 - 2r_2$ ,  $a_3 = r_2 - r_1$ ,  $r_1 = \frac{v\Delta t}{2\Delta x}$ ,  $r_2 = \frac{\alpha\Delta t}{\Delta x^2}$ .

Положим  $a_4 = \frac{1}{1+\lambda\Delta x}$ ,  $a_5 = \frac{\lambda\Delta x}{1+\lambda\Delta x}$ , тогда из (1.19)

$$\begin{aligned} c_N^k &= \frac{c_{N-1}^k + \lambda\Delta x g^k}{1 + \lambda\Delta x} = a_4 c_{N-1}^k + a_5 g^k = a_4 (a_1 c_{N-2}^{k-1} + a_2 c_{N-1}^{k-1} + a_3 c_N^{k-1}) + a_5 g^k = \\ &= a_4 a_1 c_{N-2}^{k-1} + a_4 a_2 c_{N-1}^{k-1} + a_4 a_3 c_N^{k-1} + a_5 g^k, \quad k = 1, 2, \dots, K \end{aligned}$$

и искомая дискретная линейная система может быть записана в следующем виде:

$$\underbrace{\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ \vdots \\ c_{n-2}^k \\ c_{n-1}^k \\ c_n^k \end{bmatrix}}_{c_k} = \underbrace{\begin{bmatrix} a_2 & a_3 & 0 & \cdots & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & \cdots & 0 & 0 & 0 \\ 0 & a_1 & a_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \cdots & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & \cdots & a_4 a_1 & a_4 a_2 & a_4 a_3 \end{bmatrix}}_{F_{k-1}} \underbrace{\begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ \vdots \\ c_{n-2}^{k-1} \\ c_{n-1}^{k-1} \\ c_n^{k-1} \end{bmatrix}}_{c_{k-1}} + \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & a_5 \end{bmatrix}}_{B_{k-1}} \underbrace{\begin{bmatrix} f^{k-1} \\ g^k \end{bmatrix}}_{u_{k-1}}, \quad (1.20)$$

$$k = 1, 2, \dots, K.$$

В полученной системе компоненты вектора состояния  $c_k$  соответствуют всем внутренним узлам пространственной сетки и правой границе ( $n = N$ ),

вектор входных воздействий  $u_k$  является двумерным ( $r = 2$ ), матрицы  $F_k$  и  $B_k$  являются постоянными.

Если левое граничное условие известно, а правое неизвестно, то система (1.20) может быть записана следующим образом:

$$\underbrace{\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ \vdots \\ c_{n-2}^k \\ c_{n-1}^k \\ c_n^k \end{bmatrix}}_{c_k} = \underbrace{\begin{bmatrix} a_2 & a_3 & 0 & \cdots & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & \cdots & 0 & 0 & 0 \\ 0 & a_1 & a_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \cdots & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & \cdots & a_4 a_1 & a_4 a_2 & a_4 a_3 \end{bmatrix}}_{F_{k-1}} \underbrace{\begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ \vdots \\ c_{n-2}^{k-1} \\ c_{n-1}^{k-1} \\ c_n^{k-1} \end{bmatrix}}_{c_{k-1}} + \underbrace{\begin{bmatrix} a_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_{k-1}^1} \underbrace{\underbrace{f^{k-1}}_{u_{k-1}^1}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ a_5 \end{bmatrix}}_{B_{k-1}} \underbrace{\underbrace{g^k}_{u_{k-1}}},$$

$k = 1, 2, \dots, K.$

(1.21)

Если правое граничное условие известно, а левое неизвестно, то система (1.20) может быть записана аналогично (1.21), но с другими обозначениями:

$$\underbrace{\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ \vdots \\ c_{n-2}^k \\ c_{n-1}^k \\ c_n^k \end{bmatrix}}_{c_k} = \underbrace{\begin{bmatrix} a_2 & a_3 & 0 & \cdots & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & \cdots & 0 & 0 & 0 \\ 0 & a_1 & a_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \cdots & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & \cdots & a_4 a_1 & a_4 a_2 & a_4 a_3 \end{bmatrix}}_{F_{k-1}} \underbrace{\begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ \vdots \\ c_{n-2}^{k-1} \\ c_{n-1}^{k-1} \\ c_n^{k-1} \end{bmatrix}}_{c_{k-1}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ a_5 \end{bmatrix}}_{B_{k-1}^1} \underbrace{\underbrace{g^k}_{u_{k-1}^1}} + \underbrace{\begin{bmatrix} a_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_{k-1}} \underbrace{\underbrace{f^{k-1}}_{u_{k-1}}},$$

$k = 1, 2, \dots, K.$

(1.22)

Как и в предыдущем разделе к полученным системам добавим уравнение зашумленных измерений вида (1.6):

$$z_k = H_k c_k + \xi_k, \quad k = 1, 2, \dots, K. \quad (1.23)$$

Рассмотрим переход к дискретной линейной стохастической системе на примере следующей модели.

**Пример 1.2.**

$$\frac{\partial c}{\partial t} + \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}, \quad (1.24)$$

$$c(x, 0) = 10x(x - 1), \quad (1.25)$$

$$c(0, t) = 5t^2, \quad \frac{\partial c(1, t)}{\partial x} = -[c(1, t) - t], \quad (1.26)$$

$$x \in [0; 1], \quad t \in [0; 1].$$

Зададим ту же самую пространственно-временную сетку, что и в примере 1.1. Вектор состояния в данном случае будет состоять из 4 внутренних узлов пространственной сетки и правой границы отрезка ( $n = 5$ ).

На рисунке 1.5 приведен график решения прямой задачи (1.24)–(1.26) методом конечных разностей на построенной сетке.

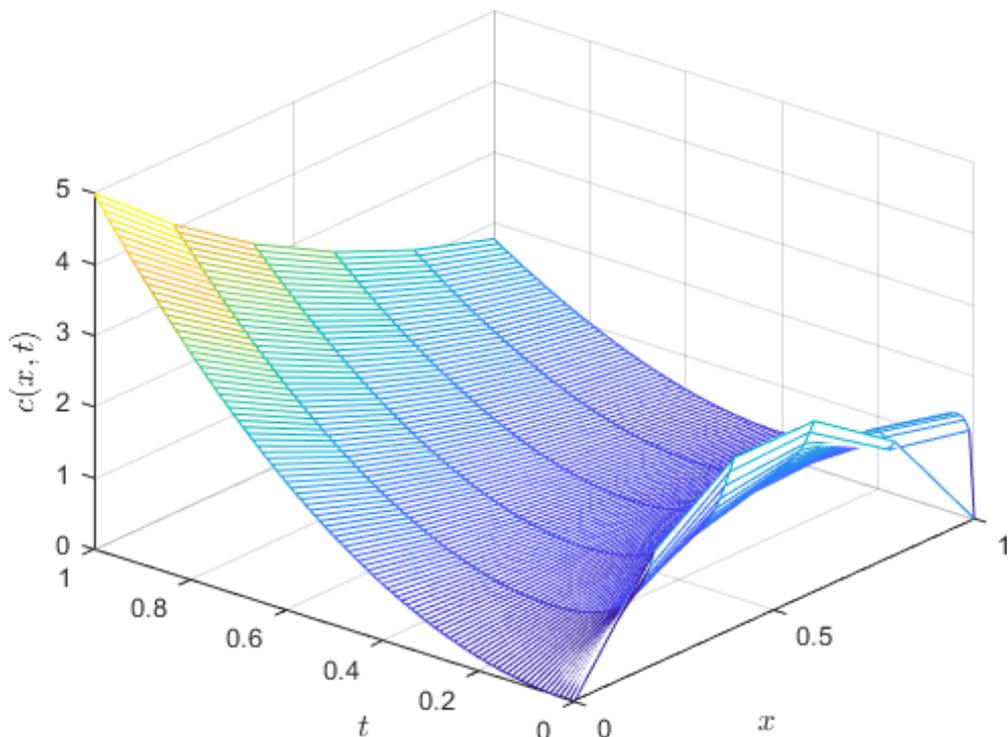


Рис. 1.5. Решение задачи

Система (1.20) в этом случае имеет вид:

$$\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \\ c_5^k \end{bmatrix} = \begin{bmatrix} 0.500 & 0.225 & 0 & 0 & 0 \\ 0.275 & 0.500 & 0.225 & 0 & 0 \\ 0 & 0.275 & 0.500 & 0.225 & 0 \\ 0 & 0 & 0.275 & 0.500 & 0.225 \\ 0 & 0 & 0.229 & 0.417 & 0.188 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \\ c_5^{k-1} \end{bmatrix} + \begin{bmatrix} 0.275 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.167 \end{bmatrix} \begin{bmatrix} f^{k-1} \\ g^k \end{bmatrix},$$

$$k = 1, 2, \dots, 100,$$

а системы (1.21) и (1.22) записываются следующим образом:

$$\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \\ c_5^k \end{bmatrix} = \begin{bmatrix} 0.500 & 0.225 & 0 & 0 & 0 \\ 0.275 & 0.500 & 0.225 & 0 & 0 \\ 0 & 0.275 & 0.500 & 0.225 & 0 \\ 0 & 0 & 0.275 & 0.500 & 0.225 \\ 0 & 0 & 0.229 & 0.417 & 0.188 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \\ c_5^{k-1} \end{bmatrix} + \begin{bmatrix} 0.275 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} f^{k-1} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.167 \end{bmatrix} g^k,$$

$$k = 1, 2, \dots, 100,$$

где  $f^{k-1} = 5((k-1)\Delta t)^2$ ,  $g^k = k\Delta t$ .

Как и в предыдущем разделе к полученным уравнениям объекта могут быть добавлены разные модели зашумленных измерений. В качестве примера рассмотрим модель с измерениями на правой границе отрезка:

$$z_1^k = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \\ c_5^{k-1} \end{bmatrix} + \xi_1^k, \quad k = 1, 2, \dots, 100.$$

На рисунке 1.6 приведен график зашумленных измерений с ковариацией шума  $R_k = 0.05^2$ .

## 1.4. Анализ свойств дискретных моделей

Проверим свойства полной наблюдаемости и полной управляемости полученных моделей. Для этого воспользуемся критериями полной управляемости и

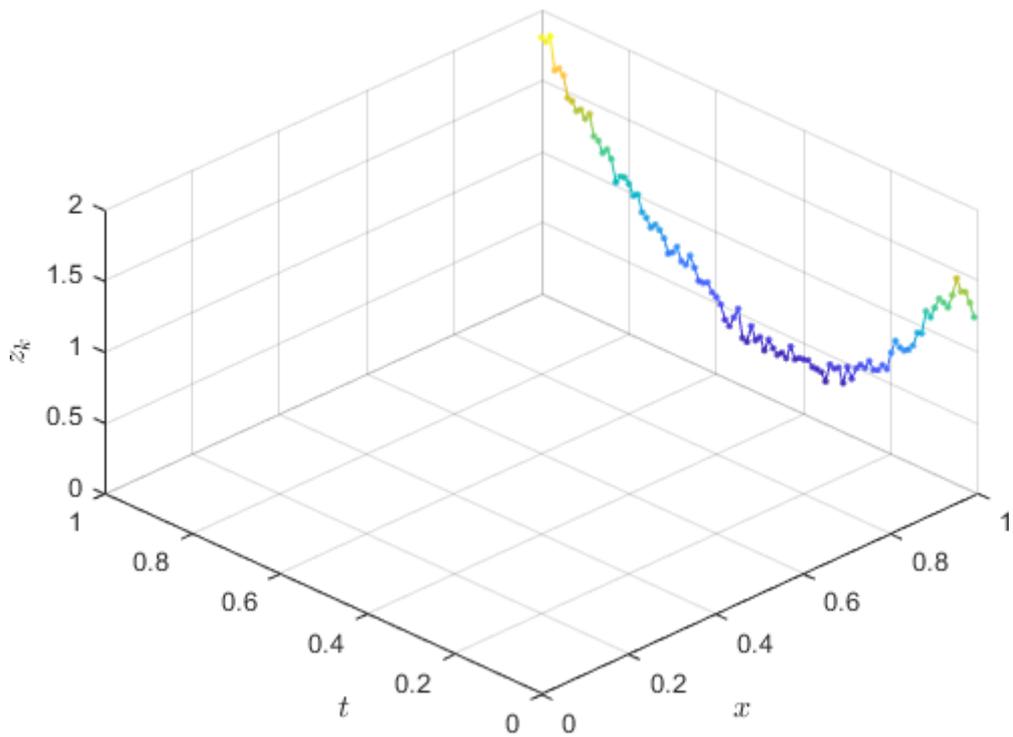


Рис. 1.6. Зашумленные измерения

наблюдаемости дискретной линейной инвариантной во времени системы, приведенными в [44] (индекс  $k$  в обозначении матриц  $F_k$ ,  $B_k$  и  $H_k$  будем далее опускать). Пусть  $n$  — размер вектора состояния системы.

**Определение 1.1** (Критерий полной наблюдаемости). *Дискретная линейная инвариантная во времени система является полностью наблюдаемой тогда и только тогда, когда  $\text{rank } M_{DTI} = n$ , где*

$$M_{DTI} = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \\ HF^{n-1} \end{bmatrix} \quad (1.27)$$

— матрица наблюдаемости.

**Определение 1.2** (Критерий полной управляемости). *Дискретная линейная инвариантная во времени система является полностью управляемой тогда и толь-*

ко тогда, когда  $\text{rank } W_{DTI} = n$ , где

$$W_{DTI} = \left[ B \mid FB \mid F^2B \mid \dots \mid F^{n-1}B \right] \quad (1.28)$$

— матрица управляемости.

### Наблюдаемость дискретных моделей

Будем рассматривать матрицы измерений  $H$  вида

$$H = \begin{bmatrix} e_{1,j_1} \\ e_{2,j_2} \\ \vdots \\ e_{m,j_m} \end{bmatrix}. \quad (1.29)$$

где  $e_{i,j_i} = [0, \dots, 0, 1, 0, \dots, 0]$  —  $i$ -я строка матрицы с единицей на  $j_i$ -м месте). Очевидным выбором матрицы  $H$ , обеспечивающим полную наблюдаемость, является  $H = I_n$ , поскольку в этом случае

$$M_{DTI} = \begin{bmatrix} I \\ F \\ F^2 \\ \vdots \\ F^{n-1} \end{bmatrix}. \quad (1.30)$$

На практике такое решение может оказаться избыточным, поскольку единичная матрица означает наличие сенсоров в узлах пространственной сетки, соответствующих каждой компоненте вектора состояния. Отдельной задачей является поиск минимального размера матрицы  $H$ , при котором система остается наблюдаемой (поиск минимального набора сенсоров).

В общем случае (для произвольного  $n$ ) проверка критерия полной наблюдаемости является затруднительной, но для конкретных значений  $n$  она может быть выполнена с использованием систем компьютерной алгебры. В данной работе для этих целей использовалась система символьных вычислений Maple [45]. В приложении А приведены скрипты для автоматической проверки

свойства полной наблюдаемости. Скрипт А.1 позволяет по заданному  $n$  и списку строк с указанием положения единиц в матрице  $H$  автоматически сформировать матрицы  $H$ ,  $F$ ,  $M_{DTI}$  и проверить выполнение критерия полной наблюдаемости для случая граничных условий первого рода, а скрипт А.2 — для смешанных граничных условий.

В качестве примера рассмотрим пространственную сетку с 6 узлами ( $N = 5$ ) и модель с граничными условиями первого рода. В данном случае  $n = 4$ , а система (1.10) имеет вид

$$\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \end{bmatrix} = \begin{bmatrix} a_2 & a_3 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 \\ 0 & a_1 & a_2 & a_3 \\ 0 & 0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \end{bmatrix} + \begin{bmatrix} a_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & a_3 \end{bmatrix} \begin{bmatrix} f^{k-1} \\ g^{k-1} \end{bmatrix}. \quad (1.31)$$

Задавая в скрипте А.1 различные матрицы  $H$  можно доказать, что имеет место следующее утверждение.

**Утверждение 1.1.** Пусть  $a_1 \neq 0$  и  $a_3 \neq 0$ . Тогда система (1.31) будет полностью наблюдаемой при любой матрице измерений  $H$  вида (1.29).

В частности, при соответствующих условиях на коэффициенты  $a_1$  и  $a_3$  система (1.31) будет полностью наблюдаема для матриц  $H$  вида (1.29) с одной строкой. Например, для  $H = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$  матрица наблюдаемости имеет вид

$$M_{DTI} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_2 & a_3 & 0 & 0 \\ a_3 a_1 + a_2^2 & 2a_2 a_3 & a_3^2 & 0 \\ 3a_2 a_3 a_1 + a_2^3 & 2a_3^2 a_1 + 3a_2^2 a_3 & 3a_2 a_3^2 & a_3^3 \end{bmatrix}.$$

Пусть  $a_3 \neq 0$ , тогда  $\text{rank } M_{DTI} = 4$ .

Таким образом минимально необходимое количество сенсоров для рассматриваемой системы равно 1.

Для той же пространственной сетки, но смешанных граничных условий,

система (1.20) имеет вид

$$\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \\ c_5^k \end{bmatrix} = \begin{bmatrix} a_2 & a_3 & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 & 0 \\ 0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 \\ 0 & 0 & a_4 a_1 & a_4 a_2 & a_4 a_3 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \\ c_5^{k-1} \end{bmatrix} + \begin{bmatrix} a_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & a_5 \end{bmatrix} \begin{bmatrix} f^{k-1} \\ g^k \end{bmatrix}. \quad (1.32)$$

В этом случае сформулировать общее утверждение аналогичное утверждению 1.1 сложнее, но как и в предыдущем случае можно показать, что для  $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$  система (1.32) будет полностью наблюдаемой при  $a_3 \neq 0$ . Таким образом минимально необходимое количество сенсоров для этой системы также равно 1.

### Управляемость дискретных моделей

В приложении Б приведены скрипты для автоматической проверки свойства полной управляемости. Скрипт Б.1 позволяет по заданному  $n$  автоматически сформировать матрицы  $B$ ,  $F$ ,  $W_{DTI}$  и проверить выполнение критерия полной управляемости для случая граничных условий первого рода, а скрипт Б.2 — для смешанных граничных условий.

Используя данные скрипты, можно доказать следующие утверждения.

**Утверждение 1.2.** Пусть  $a_1 \neq 0$  и  $a_3 \neq 0$ . Тогда система (1.31) будет полностью управляема.

**Утверждение 1.3.** Пусть  $a_1 \neq 0$ ,  $a_3 \neq 0$  и  $a_5 \neq 0$ . Тогда система (1.32) будет полностью управляема.

## 1.5. Заключение и выводы к главе 1

В данной главе были получены дискретные линейные стохастические модели, которые будут использоваться для решения задач параметрической идентификации моделей конвективно-диффузионного переноса (1.1), (1.2), (1.3) и (1.1), (1.2), (1.4) по результатам зашумленных измерений, а также проанализированы свойства наблюдаемости и управляемости полученных моделей.

### Выводы.

1. Полученные модели построены на основе явной трехточечной конечно-разностной схемы и состоят из уравнения объекта и уравнения зашумленных измерений. Вид уравнений объекта зависит от типа граничных условий.

2. Модели являются полностью наблюдаемыми и управляемыми при выполнении дополнительных условий на вид матрицы измерений  $H_k$  и коэффициенты конечно-разностной схемы (утверждения 1.1, 1.2, 1.3).

3. Если оба граничных условия (1.3) или (1.4) известны, нами будет рассматриваться задача идентификации коэффициентов  $v$  и  $\alpha$  исходного уравнения по результатам зашумленных измерений. Решение данной задачи будет рассмотрено в главе 2.

4. Если хотя бы одно из граничных условий (1.3) или (1.4) неизвестно, то вектор входных воздействий  $u_k$  является неизвестным. В данном случае для численной идентификации неизвестных граничных условий по результатам зашумленных измерений может быть использован алгоритм одновременного оценивания состояния и входных воздействий Гиллейнса – Де-Мора. Решение данной задачи будет рассмотрено в главе 3.

5. Полученные в данной главе дискретные линейные динамические системы являются детерминированными — шум  $w_k$  в объекте отсутствует ( $Q_k = 0$ ). Для моделирования случайных погрешностей в эти системы может быть добавлено слагаемое  $G_{k-1}w_{k-1}$ . Например, система (1.10) в этом случае будет иметь

ВИД

$$\begin{aligned}
\underbrace{\begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ \vdots \\ c_{n-2}^k \\ c_{n-1}^k \\ c_n^k \end{bmatrix}}_{c_k} &= \underbrace{\begin{bmatrix} a_2 & a_3 & 0 & \dots & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & \dots & 0 & 0 & 0 \\ 0 & a_1 & a_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \dots & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & \dots & 0 & a_1 & a_2 \end{bmatrix}}_{F_{k-1}} \underbrace{\begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ \vdots \\ c_{n-2}^{k-1} \\ c_{n-1}^{k-1} \\ c_n^{k-1} \end{bmatrix}}_{c_{k-1}} + \underbrace{\begin{bmatrix} a_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & a_3 \end{bmatrix}}_{B_{k-1}} \underbrace{\begin{bmatrix} f^{k-1} \\ g^{k-1} \end{bmatrix}}_{u_{k-1}} + G_{k-1} w_{k-1}, \\
k &= 1, 2, \dots, K.
\end{aligned}
\tag{1.33}$$

## Глава 2

# Идентификация коэффициентов конвекции и диффузии

## 2.1. Постановка задачи

Рассмотрим следующую задачу параметрической идентификации.

**Задача 2.1.** *Определить коэффициенты  $v$  и  $\alpha$  в уравнении (1.1) по результатам зашумленных измерений значений функции  $c(x, t)$  в отдельных точках рассматриваемого отрезка в последовательные моменты времени (функции  $\varphi(x)$ ,  $f(t)$ ,  $g(t)$  и коэффициент  $\lambda$ , входящие в начальное и граничные условия предполагаются известными).*

Обозначим через  $\theta = [v, \alpha]^T$  неизвестный векторный параметр, подлежащий идентификации. Перейдем от исходной непрерывной модели (1.1), (1.2), (1.3) или (1.1), (1.2), (1.4) к дискретной линейной стохастической системе в пространстве состояний (1.5), (1.6). Поскольку неизвестные коэффициенты уравнения  $v$  и  $\alpha$  входят в коэффициенты  $a_1, a_2, a_3$  матриц  $F_k$  и  $B_k$  дискретной системы, она будет иметь вид

$$\begin{cases} c_k = F_{k-1}(\theta)c_{k-1} + B_{k-1}(\theta)u_{k-1}, & (2.1) \end{cases}$$

$$\begin{cases} z_k = H_k c_k + \xi_k, & k = 1, 2, \dots, K. \end{cases} \quad (2.2)$$

Задача параметрической идентификации состоит в нахождении неизвестного параметра  $\theta$  по известным входным воздействиям  $U_0^{K-1} = \{u_0, u_1, \dots, u_{K-1}\}$  и данным наблюдений  $Z_1^K = \{z_1, \dots, z_K\}$  в соответствии с выбранным критерием качества идентификации  $J(\theta; Z_1^K, U_0^{K-1})$  [46]. В этом случае задача оценки неизвестного параметра требует решения задачи нелинейного программирования

$$\hat{\theta}_{min} = \underset{\theta}{\operatorname{argmin}} J(\theta; Z_1^K, U_0^{K-1}), \quad (2.3)$$

где  $\theta \in D(\theta)$  (область определения  $\theta$ ).

Поскольку (2.1), (2.2) является дискретной линейной стохастической моделью с гауссовым шумом для решения данной задачи целесообразно использовать *метод максимального правдоподобия*, который часто применяется на практике для решения задач параметрической идентификации дискретных линейных стохастических систем [14], [47]. В данном методе в качестве критерия идентификации используется отрицательная логарифмическая функция правдоподобия [48]

$$J_{CKF}(\theta; Z_1^K, U_0^{K-1}) = \frac{Km}{2} \ln(2\pi) + \frac{1}{2} \sum_{k=1}^K \left\{ \ln[\det(\Sigma_{\nu,k})] + \nu_k^T \Sigma_{\nu,k}^{-1} \nu_k \right\}, \quad (2.4)$$

где вектор невязки  $\nu_k$  и его ковариационная матрица  $\Sigma_{\nu,k}^{-1}$  для заданного значения параметра  $\theta$  вычисляются из известных уравнений дискретного фильтра Калмана [14].

Введем обозначения

$$\begin{aligned} \nu_k &= z_k - H_k \hat{c}_{k|k-1}, \\ \hat{c}_{k|k-1} &= \mathbb{E} \{ c_k | z_1, z_2, \dots, z_{k-1} \}, \\ \hat{c}_k &= \mathbb{E} \{ c_k | z_1, z_2, \dots, z_k \}, \\ P_{k|k-1} &= \mathbb{E} \{ (c_k - \hat{c}_{k|k-1})(c_k - \hat{c}_{k|k-1})^T | z_1, z_2, \dots, z_{k-1} \}, \\ P_k &= \mathbb{E} \{ (c_k - \hat{c}_k)(c_k - \hat{c}_k)^T | z_1, z_2, \dots, z_k \}. \end{aligned}$$

Алгоритм 2.1 классического фильтра Калмана в ковариационной форме для модели (1.5), (1.6) состоит из двух этапов: этапа обновления по времени (экстраполяции, предсказания) и этапа обновления по измерениям (фильтрации, коррекции) [14]. Блок-схема алгоритма 2.1 представлена на рисунке 2.1.

Для модели (2.1), (2.2) строки 3 и 4 алгоритма будут иметь вид:

$$\begin{aligned} \hat{c}_{k|k-1} &= F_{k-1}(\theta) \hat{c}_{k-1} + B_{k-1}(\theta) u_{k-1}, \\ P_{k|k-1} &= F_{k-1}(\theta) P_{k-1} F_{k-1}^T(\theta). \end{aligned}$$

---

**Алгоритм 2.1.** Фильтр Калмана в ковариационной форме (СКФ)
 

---

**Вход:**  $\bar{c}_0, \Pi_0$ 

- 1  $\hat{c}_0 = \bar{c}_0, P_0 = \Pi_0$  // Инициализация
- 2 **for**  $k = 1, 2, \dots, K$  **do**
  - // Обновление по времени (экстраполяция)
  - 3  $\hat{c}_{k|k-1} = F_{k-1}\hat{c}_{k-1} + B_{k-1}u_{k-1}$
  - 4  $P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T$
  - // Обновление по измерениям (фильтрация)
  - 5  $K_k = P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1} = P_{k|k-1}H_k^T(\Sigma_{\nu,k})^{-1}$
  - 6  $\hat{c}_k = \hat{c}_{k|k-1} + K_k(z_k - H_k\hat{x}_{k|k-1}) = \hat{c}_{k|k-1} + K_k\nu_k$
  - 7  $P_k = (I - K_kH_k)P_{k|k-1}$
- 8 **end for**

**Выход:**  $\hat{c}_k, P_k, k = 1, 2, \dots, K$ 


---

Критерий идентификации (2.4), вычисляемый на основе величин, получаемых фильтром Калмана или его модификацией, может служить целевой функцией для алгоритмов минимизации различных типов.

В разделах 2.2 и 2.3 будут рассмотрены соответственно безградиентные и градиентные методы решения задачи (2.3) с критерием идентификации (2.4) на основе стандартного фильтра Калмана. В разделе 2.4 будет рассмотрен метод решения на основе численно-устойчивой модификации фильтра Калмана. Основные результаты данной главы описаны в работах [37, 49, 50, 51, 52].

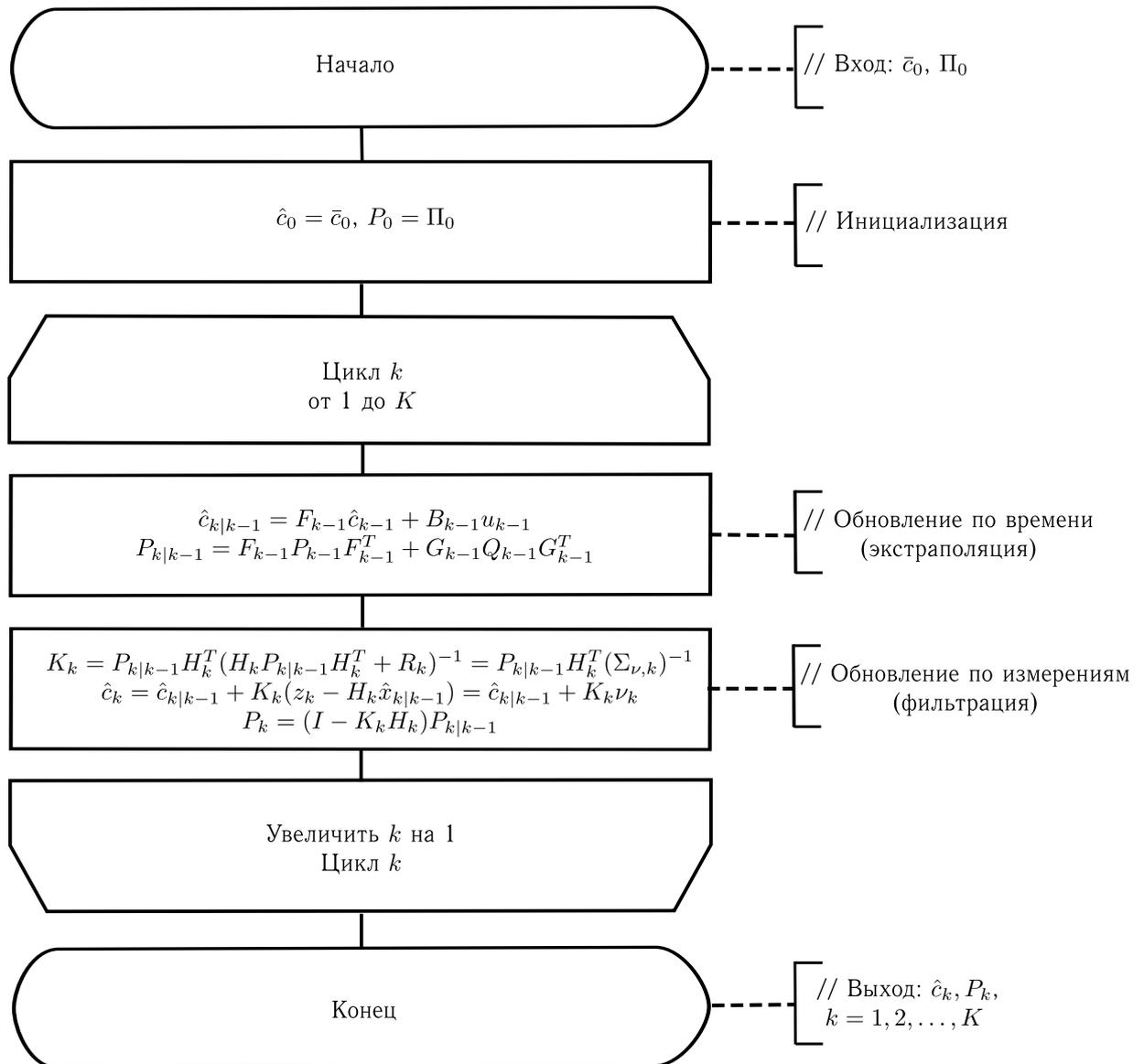


Рис. 2.1. Блок-схема алгоритма 2.1

## 2.2. Идентификация с использованием стандартного фильтра Калмана и безградиентных методов оптимизации

Сформулируем алгоритм вычисления критерия идентификации (2.4) на основе алгоритма 2.1 для дискретных линейных стохастических систем с уравнениями объекта (1.10) и (1.20). Блок-схема алгоритма 2.2 представлена на рисунке 2.2.

---

**Алгоритм 2.2.** Вычисление критерия идентификации на основе стандартного фильтра Калмана

---

**Вход:**  $\theta, Z_1^K, U_0^{K-1}, \bar{c}_0, \Pi_0$

- 1  $\hat{c}_0 = \bar{c}_0, P_0 = \Pi_0, J_{CKF} = Km \ln(2\pi)$
- 2 **for**  $k = 1, 2, \dots, K$  **do**
- 3      $\hat{c}_{k|k-1} = F_{k-1}(\theta)\hat{c}_{k-1} + B_{k-1}(\theta)u_{k-1}$
- 4      $P_{k|k-1} = F_{k-1}(\theta)P_{k-1}F_{k-1}^T(\theta)$
- 5      $\Sigma_{\nu,k} = H_k P_{k|k-1} H_k^T + R_k$
- 6      $\nu_k = z_k - H_k \hat{x}_{k|k-1}$
- 7      $K_k = P_{k|k-1} H_k^T (\Sigma_{\nu,k})^{-1}$
- 8      $\hat{c}_k = \hat{c}_{k|k-1} + K_k \nu_k$
- 9      $P_k = (I - K_k H_k) P_{k|k-1}$
- 10      $J_{CKF} = J_{CKF} + \ln(\det(\Sigma_{\nu,k})) + \nu_k^T \Sigma_{\nu,k}^{-1} \nu_k$
- 11 **end for**
- 12  $J_{CKF} = \frac{1}{2} J_{CKF}$

**Выход:**  $J_{CKF}(\theta; Z_1^K, U_0^{K-1})$

---

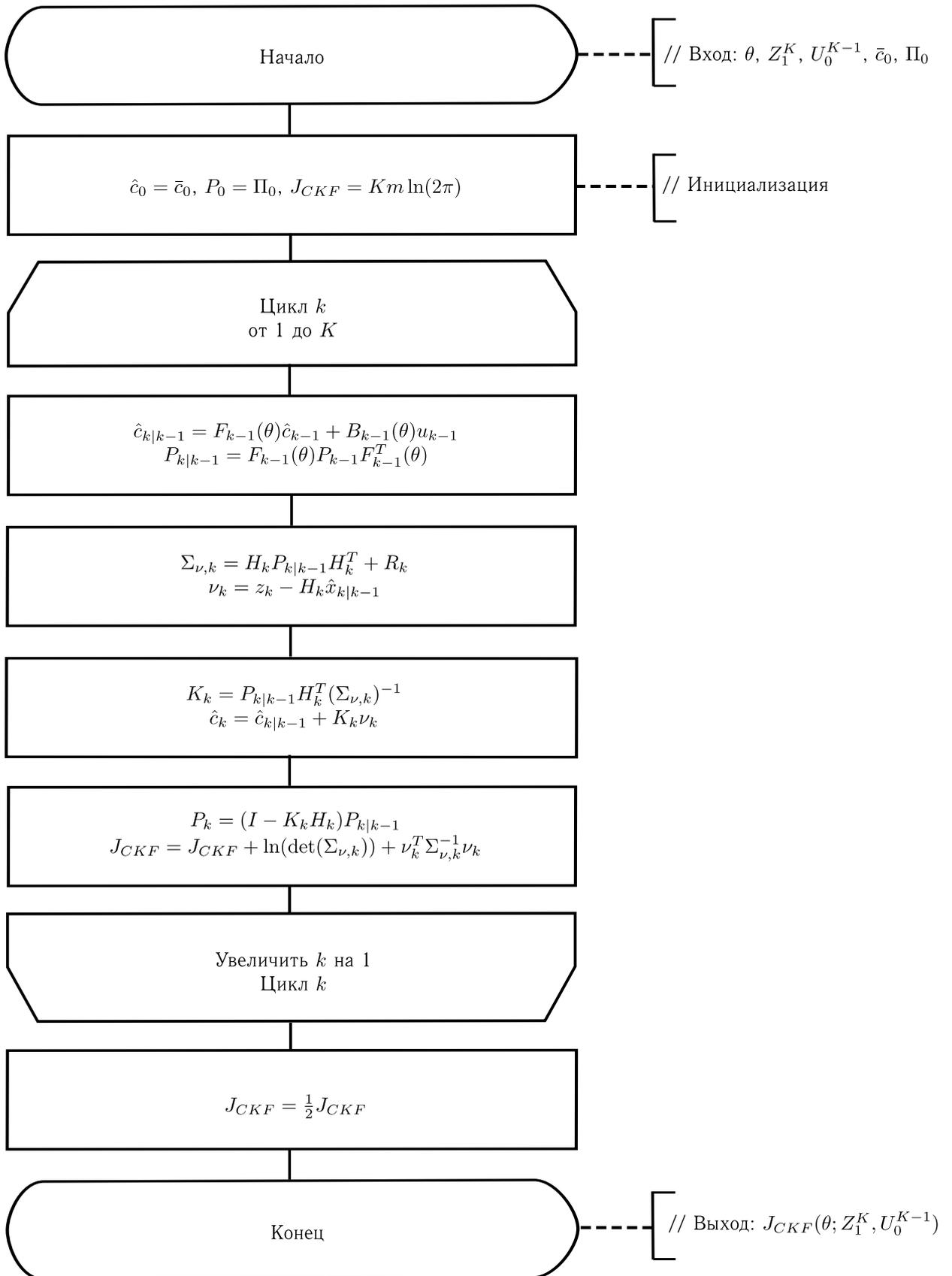


Рис. 2.2. Блок-схема алгоритма 2.2

Начальное значение вектора состояния может быть вычислено на основе начального условия (1.2) следующим образом:

$$\bar{c}_0 = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \dots \\ \varphi_n \end{bmatrix}. \quad (2.5)$$

Последовательность управляющих воздействий  $U_0^{K-1}$  в случае граничных условий первого рода (1.3) имеет вид

$$U_0^{K-1} = \left\{ \begin{bmatrix} f^0 \\ g^0 \end{bmatrix}, \begin{bmatrix} f^1 \\ g^1 \end{bmatrix}, \dots, \begin{bmatrix} f^{K-1} \\ g^{K-1} \end{bmatrix} \right\}, \quad (2.6)$$

а в случае смешанных граничных условий (1.4)

$$U_0^{K-1} = \left\{ \begin{bmatrix} f^0 \\ g^1 \end{bmatrix}, \begin{bmatrix} f^1 \\ g^2 \end{bmatrix}, \dots, \begin{bmatrix} f^{K-1} \\ g^K \end{bmatrix} \right\}. \quad (2.7)$$

Таким образом справедливо следующее утверждение.

**Утверждение 2.1.** *Критерий идентификации  $J_{СКФ}$  для дискретной линейной стохастической системы (2.1), (2.2) по результатам измерений  $Z_1^K$  может быть вычислен при помощи алгоритма 2.2, при этом последовательность управляющих воздействий  $U_0^{K-1}$  для уравнения объекта (1.10) имеет вид (2.6), а для уравнения объекта (1.20) — вид (2.7).*

Одним из мощных классов безградиентных методов решения оптимизационных задач являются метаэвристические алгоритмы, которые широко применяются при решении задач параметрической идентификации [53, 54, 55]. *Метаэвристика* — это высокоуровневая стратегия поиска решений, применимая к широкому спектру задач оптимизации. Особенностью метаэвристических алгоритмов является то, что практически все они недетерминированы.

В зависимости от способа получения решения большинство метаэвристических алгоритмов оптимизации можно разделить на две большие группы: траекторные (с одним состоянием) и популяционные. В траекторных алгоритмах

процесс поиска решения можно рассматривать как движение между отдельными решениями задачи, в то время как в популяционных алгоритмах в процессе поиска оптимального решения изменяется группа решений.

Одним из наиболее популярных траекторных алгоритмов, используемых при решении задач глобальной оптимизации, является *метод имитации отжига* (англ. Simulated Annealing — SA). Ключевой особенностью метода является использование управляющего параметра — температуры, что позволяет управлять недетерминированным процессом поиска решения. Как правило, температура снижается в процессе работы алгоритма по определенному закону, начиная с некоторого начального значения. На каждой итерации алгоритма случайно сгенерированное новое решение из окрестности текущего решения принимается с вероятностью 1, если оно лучше, и с вероятностью менее 1, если хуже текущего, причем вероятность принятия худшего решения уменьшается с понижением температуры. Качество решений оценивается с помощью *функции стоимости* (целое или действительное число) [56].

*Генетический алгоритм* (англ. Genetic Algorithm — GA) является популярной версией алгоритмов эволюционной оптимизации, основанных на моделировании процессов естественного отбора. В эволюционных алгоритмах качество решений оценивается с помощью *функции приспособленности*, а основная идея алгоритмов заключается в том, что решения с наилучшими значениями данной функции «выживают» в ходе эволюции. В генетическом алгоритме на каждой итерации эволюционного процесса новая популяция получается из текущей популяции путем последовательного использования одного или нескольких генетических операторов. Наиболее распространенными генетическими операторами являются кроссовер (скрещивание) — получение из родительских решений решений-потомков и мутация — случайное изменение решения [56].

Псевдокод алгоритмов SA и GA приведен в приложении В. Более подробное описание и обсуждение метаэвристических алгоритмов оптимизации можно найти, например, в [57, 58].

Рассмотренные алгоритмы реализованы в различных библиотеках и системах компьютерной математики. Например, в наборе инструментов глобальной

оптимизации MATLAB [59] имеются функции `simulannealbnd` и `ga`. Для их использования, пользователю необходимо реализовать целевую функцию для решаемой задачи. Рассмотрим применение метаэвристических алгоритмов на примере следующей модели.

### Пример 2.1.

$$\frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} = \alpha \frac{\partial^2 c}{\partial x^2}, \quad (2.8)$$

$$c(x, 0) = \exp\left(\frac{vx}{2\alpha}\right) \sin x, \quad (2.9)$$

$$c(0, t) = 0, \quad c(\pi, t) = 0, \quad (2.10)$$

$$x \in [0; \pi], \quad t \in [0; 4],$$

где  $c(x, t)$  — концентрация вещества в одномерном потоке, (2.9) — начальная концентрация вещества, а граничные условия (2.10) соответствуют случаю поглощающих стенок.

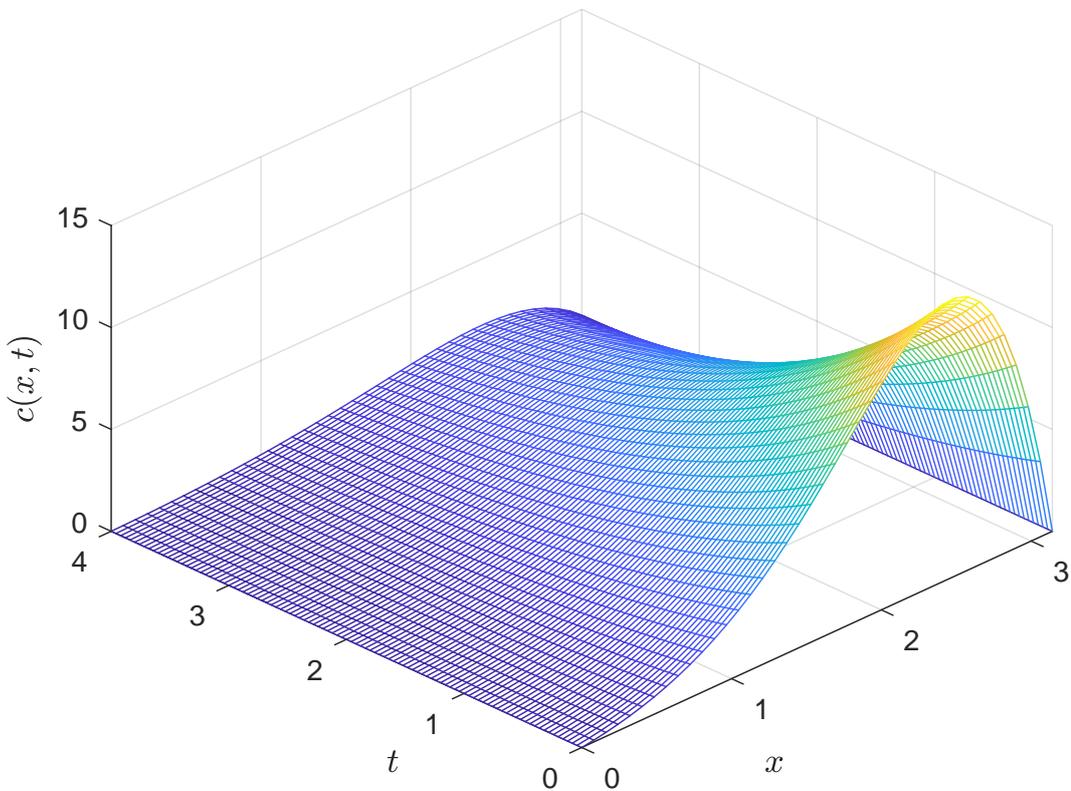


Рис. 2.3. Точное решение

Точное решение прямой задачи (2.8)–(2.10) может быть найдено аналитически и имеет вид

$$c(x, t) = \exp \left[ \frac{v}{2\alpha} \left( x - \frac{vt}{2} \right) \right] \sin x \exp(-\alpha t).$$

Пусть  $v = 0.5$ ,  $\alpha = 0.2$ . График точного решения для этого случая показан на рисунке 2.3.

Предположим, что коэффициенты  $v$  и  $\alpha$  неизвестны. Рассмотрим две дискретные модели вида (1.10), (1.13) со следующими параметрами:

$$N = 9, K = 100 (n = 8, \Delta x \approx 0.349, \Delta t = 0.04), H_k = I_8, R_k = \sigma^2 I_8, \quad (2.11)$$

$$N = 14, K = 100 (n = 13, \Delta x \approx 0.224, \Delta t = 0.04), H_k = I_{13}, R_k = \sigma^2 I_{13}. \quad (2.12)$$

Для идентификации  $v$  и  $\alpha$  воспользуемся критерием идентификации (2.4) и алгоритмами SA и GA системы MATLAB. Реализуем алгоритм вычисления критерия идентификации 2.2 в виде функции MATLAB.

Поиск решений осуществлялся в области  $D(\theta) = [0; 1] \times [0; 1]$ . Основные настройки алгоритмов приведены в таблице 2.1. В качестве критерия остановки для обоих алгоритмов использовалось ограничение по времени 5 с (процессор: Intel Core i3-3110M CPU @ 2.40GHz).

Таблица 2.1. Настройки алгоритмов

SA		GA	
TimeLimit	5	TimeLimit	5
MaxIter	Inf	Generations	Inf
MaxFunEvals	Inf	StallGenLimit	Inf
StallIterLimit	Inf	PopulationSize	10
ReannealInterval	100	PopInitRange	[0 0; 1 1]

На рисунках 2.4, 2.5 показаны усредненные графики критерия (2.4) для моделей (2.11) и (2.12), полученные по результатам 100 экспериментов при  $\sigma^2 = 0.01^2$ .

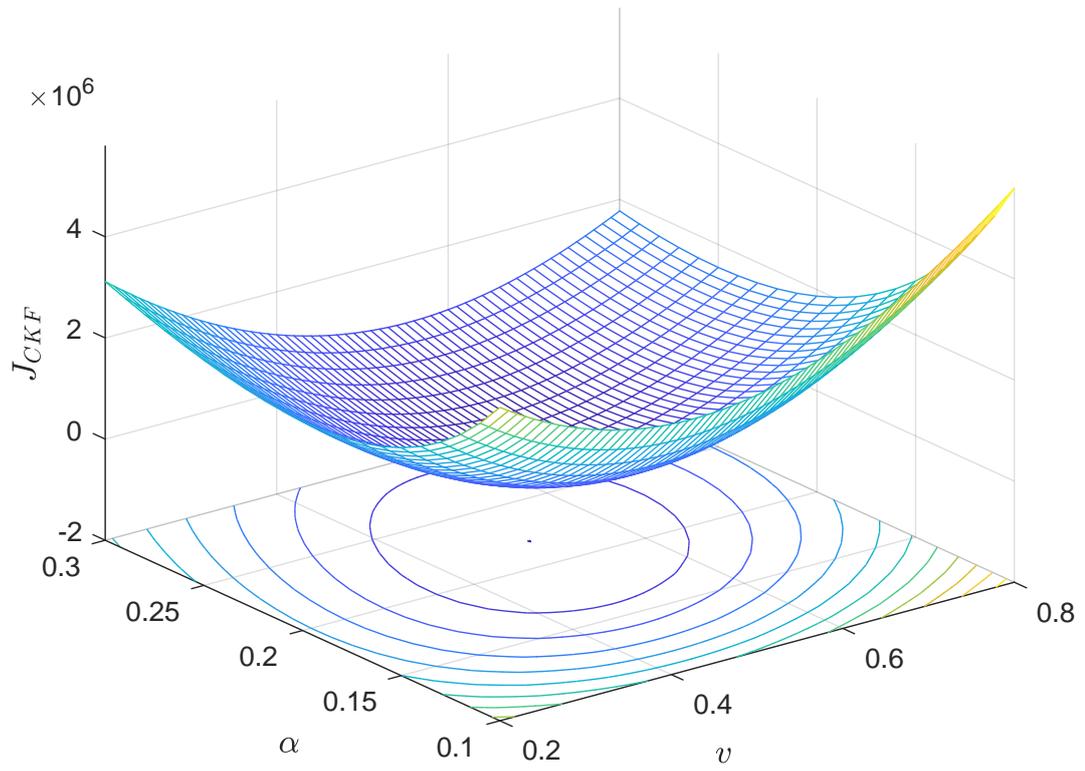


Рис. 2.4. Критерий идентификации для дискретной модели (2.11)

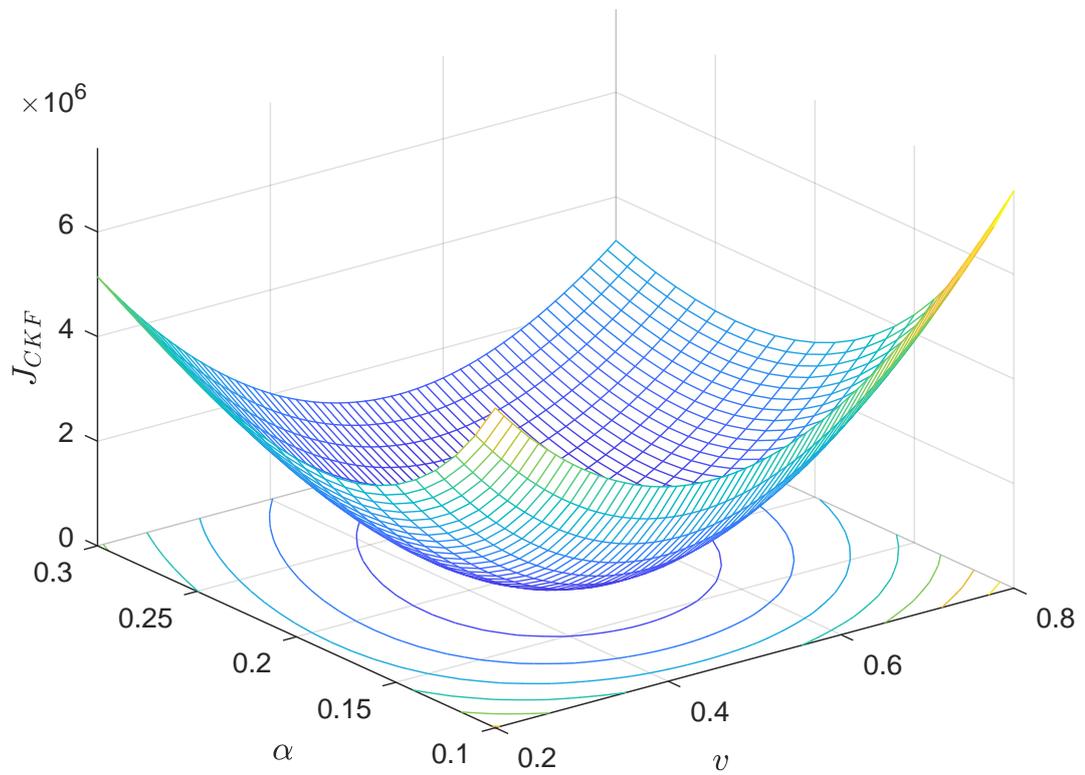


Рис. 2.5. Критерий идентификации для дискретной модели (2.12)

В таблицах 2.2 и 2.3 приведены результаты вычислительных экспериментов для моделей (2.11) и (2.12) при различных значениях дисперсии шума  $\sigma^2$ . Для каждого значения  $\sigma^2$  была проведена серия из 100 экспериментов, и для каждой серии были рассчитаны: средние значения идентифицированных параметров  $v$  и  $\alpha$ , среднеквадратические ошибки (RMSE) и средние абсолютные ошибки в процентах (MAPE).

Таблица 2.2. Результаты экспериментов для модели (2.11)

$\sigma^2$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$5^2$	SA	0.4977	0.2146	0.0445	0.0280	7.3639	10.4879
	GA	0.5064	0.2170	0.0547	0.0297	8.4364	11.1271
$1^2$	SA	0.5097	0.2171	0.0175	0.0181	2.8178	8.5639
	GA	0.5163	0.2196	0.0282	0.0243	4.0656	9.8051
$0.1^2$	SA	0.5080	0.2159	0.0082	0.0160	1.5962	7.9725
	GA	0.5183	0.2201	0.0352	0.0251	3.7041	10.0690
$0.01^2$	SA	0.5078	0.2159	0.0078	0.0159	1.5647	7.9338
	GA	0.5152	0.2187	0.0298	0.0217	3.2452	9.3442

Таблица 2.3. Результаты экспериментов для модели (2.12)

$\sigma^2$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$5^2$	SA	0.5059	0.2078	0.0386	0.0209	6.0407	8.2662
	GA	0.5097	0.2107	0.0438	0.0235	6.6677	9.2259
$1^2$	SA	0.4982	0.2054	0.0130	0.0072	2.0925	2.9151
	GA	0.5045	0.2078	0.0231	0.0108	3.1352	4.0494
$0.1^2$	SA	0.5002	0.2052	0.0014	0.0053	0.2271	2.6197
	GA	0.5073	0.2097	0.0308	0.0253	2.4083	4.9396
$0.01^2$	SA	0.5002	0.2053	0.0004	0.0053	0.0537	2.6543
	GA	0.5113	0.2086	0.0302	0.0114	2.7046	4.3246

Результаты экспериментов показывают, что оба алгоритма позволяют идентифицировать неизвестные коэффициенты  $\nu$  и  $\alpha$ , при этом на качество идентификации влияют как параметры дискретизации исходной модели (в частности, число узлов пространственной сетки), так и уровень шума. При выбранных настройках алгоритмов ошибки RMSE и MAPE для алгоритма SA получаются меньше, чем для GA.

### 2.3. Идентификация с использованием стандартного фильтра Калмана и градиентных методов оптимизации

В случае, когда может быть вычислен градиент целевой функции, для ее минимизации могут быть использованы градиентные методы численной оптимизации. Основным понятием, используемым во всех градиентных методах, является понятие градиента функции, как направления ее наискорейшего изменения. Преимущество градиентных методов обусловлено тем, что их работа гарантируется в условиях теорем сходимости [60].

Найдем выражения для вычисления градиента критерия идентификации. Пусть  $\theta = [\theta_1, \theta_2]^T = [\nu, \alpha]^T$ . Вычислим производную критерия (2.4) по параметру  $\theta_i$  ( $i = 1, 2$ ):

$$\frac{\partial J_{CKF}}{\partial \theta_i} = \frac{1}{2} \sum_{k=1}^K \left\{ \text{tr} \left[ \Sigma_{\nu,k}^{-1} \frac{\partial \Sigma_{\nu,k}}{\partial \theta_i} \right] + \frac{\partial \nu_k^T}{\partial \theta_i} \Sigma_{\nu,k}^{-1} \nu_k - \nu_k^T \Sigma_{\nu,k}^{-1} \frac{\partial \Sigma_{\nu,k}}{\partial \theta_i} \Sigma_{\nu,k}^{-1} \nu_k + \nu_k^T \Sigma_{\nu,k}^{-1} \frac{\partial \nu_k}{\partial \theta_i} \right\}. \quad (2.13)$$

Таким образом, для вычисления градиента критерия идентификации необходимо вычислить производные вектора невязки  $\nu_k$  и его ковариационной матрицы  $\Sigma_{\nu,k}$ . Для их нахождения запишем уравнения чувствительности стандартного фильтра Калмана по параметру  $\theta$ :

$$\frac{\partial \hat{c}_{k|k-1}}{\partial \theta_i} = \frac{\partial F_{k-1}(\theta)}{\partial \theta_i} \hat{c}_{k-1} + F_{k-1}(\theta) \frac{\partial \hat{c}_{k-1}}{\partial \theta_i} + \frac{\partial B_{k-1}(\theta)}{\partial \theta_i} u_{k-1}. \quad (2.14)$$

$$\frac{\partial P_{k|k-1}}{\partial \theta_i} = \frac{\partial F_{k-1}(\theta)}{\partial \theta_i} P_{k-1} F_{k-1}^T(\theta) + F_{k-1}(\theta) \frac{\partial P_{k-1}(\theta)}{\partial \theta_i} F_{k-1}^T(\theta) + F_{k-1}(\theta) P_{k-1} \left( \frac{\partial F_{k-1}(\theta)}{\partial \theta_i} \right)^T, \quad (2.15)$$

$$\frac{\partial K_k}{\partial \theta_i} = \frac{\partial P_{k|k-1}}{\partial \theta_i} H^T (\Sigma_{\nu,k})^{-1} - P_{k|k-1} H^T (\Sigma_{\nu,k})^{-1} \frac{\partial \Sigma_{\nu,k}}{\partial \theta_i} (\Sigma_{\nu,k})^{-1}, \quad (2.16)$$

$$\frac{\partial \Sigma_{\nu,k}}{\partial \theta_i} = H_k \frac{\partial P_{k|k-1}}{\partial \theta_i} H_k^T, \quad (2.17)$$

$$\frac{\partial \hat{c}_k}{\partial \theta_i} = \frac{\partial \hat{c}_{k|k-1}}{\partial \theta_i} + \frac{\partial K_k}{\partial \theta_i} \nu_k + K_k \frac{\partial \nu_k}{\partial \theta_i}, \quad (2.18)$$

$$\frac{\partial \nu_k}{\partial \theta_i} = -H_k \frac{\partial \hat{c}_{k|k-1}}{\partial \theta_i}, \quad (2.19)$$

$$\frac{\partial P_k}{\partial \theta_i} = -\frac{\partial K_k}{\partial \theta_i} H_k P_{k|k-1} + (I - K_k H_k) \frac{\partial P_{k|k-1}}{\partial \theta_i}. \quad (2.20)$$

Выражения (2.14)–(2.15) не зависят от рассматриваемой задачи, а выражения  $\frac{\partial F(\theta)}{\partial \theta_i}$  и  $\frac{\partial B(\theta)}{\partial \theta_i}$  для систем (1.10) и (1.20) будут отличаться.

Для системы (1.10):

$$\frac{\partial F(\theta)}{\partial \theta_i} = \begin{bmatrix} \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} & 0 & \cdots & 0 & 0 & 0 \\ \frac{\partial a_1}{\partial \theta_i} & \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} & \cdots & 0 & 0 & 0 \\ 0 & \frac{\partial a_1}{\partial \theta_i} & \frac{\partial a_2}{\partial \theta_i} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} & 0 \\ 0 & 0 & 0 & \cdots & \frac{\partial a_1}{\partial \theta_i} & \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} \\ 0 & 0 & 0 & \cdots & 0 & \frac{\partial a_1}{\partial \theta_i} & \frac{\partial a_2}{\partial \theta_i} \end{bmatrix}, \quad \frac{\partial B(\theta)}{\partial \theta_i} = \begin{bmatrix} \frac{\partial a_1}{\partial \theta_i} & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{\partial a_3}{\partial \theta_i} \end{bmatrix}, \quad (2.21)$$

где

$$\begin{aligned} \frac{\partial a_1}{\partial \theta_1} &= \frac{\Delta t}{2\Delta x}, & \frac{\partial a_2}{\partial \theta_1} &= 0, & \frac{\partial a_3}{\partial \theta_1} &= -\frac{\Delta t}{2\Delta x}, \\ \frac{\partial a_1}{\partial \theta_2} &= \frac{\Delta t}{\Delta x^2}, & \frac{\partial a_2}{\partial \theta_2} &= -\frac{2\Delta t}{\Delta x^2}, & \frac{\partial a_3}{\partial \theta_2} &= \frac{\Delta t}{\Delta x^2}. \end{aligned} \quad (2.22)$$

Для системы (1.20):

$$\frac{\partial F(\theta)}{\partial \theta_i} = \begin{bmatrix} \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} & 0 & \dots & 0 & 0 & 0 \\ \frac{\partial a_1}{\partial \theta_i} & \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} & \dots & 0 & 0 & 0 \\ 0 & \frac{\partial a_1}{\partial \theta_i} & \frac{\partial a_2}{\partial \theta_i} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} & 0 \\ 0 & 0 & 0 & \dots & \frac{\partial a_1}{\partial \theta_i} & \frac{\partial a_2}{\partial \theta_i} & \frac{\partial a_3}{\partial \theta_i} \\ 0 & 0 & 0 & \dots & \frac{\partial(a_4 a_1)}{\partial \theta_i} & \frac{\partial(a_4 a_2)}{\partial \theta_i} & \frac{\partial(a_4 a_3)}{\partial \theta_i} \end{bmatrix}, \quad \frac{\partial B(\theta)}{\partial \theta_i} = \begin{bmatrix} \frac{\partial a_1}{\partial \theta_i} & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{\partial a_5}{\partial \theta_i} \end{bmatrix}, \quad (2.23)$$

где

$$\frac{\partial(a_4 a_1)}{\partial \theta_i} = \frac{\partial a_4}{\partial \theta_i} a_1 + a_4 \frac{\partial a_1}{\partial \theta_i}, \quad \frac{\partial(a_4 a_2)}{\partial \theta_i} = \frac{\partial a_4}{\partial \theta_i} a_2 + a_4 \frac{\partial a_2}{\partial \theta_i}, \quad \frac{\partial(a_4 a_3)}{\partial \theta_i} = \frac{\partial a_4}{\partial \theta_i} a_3 + a_4 \frac{\partial a_3}{\partial \theta_i}, \quad (2.24)$$

а производные коэффициентов матриц вычисляются по формулам

$$\begin{aligned} \frac{\partial a_1}{\partial \theta_1} &= \frac{\Delta t}{2\Delta x}, & \frac{\partial a_2}{\partial \theta_1} &= 0, & \frac{\partial a_3}{\partial \theta_1} &= -\frac{\Delta t}{2\Delta x}, & \frac{\partial a_4}{\partial \theta_1} &= 0, & \frac{\partial a_5}{\partial \theta_1} &= 0, \\ \frac{\partial a_1}{\partial \theta_2} &= \frac{\Delta t}{\Delta x^2}, & \frac{\partial a_2}{\partial \theta_2} &= -\frac{2\Delta t}{\Delta x^2}, & \frac{\partial a_3}{\partial \theta_2} &= \frac{\Delta t}{\Delta x^2}, & \frac{\partial a_4}{\partial \theta_2} &= 0, & \frac{\partial a_5}{\partial \theta_2} &= 0. \end{aligned} \quad (2.25)$$

Таким образом, может быть сформулировано следующее утверждение.

**Предложение 2.1.** *Градиент критерия идентификации (2.4)*

$$\text{grad } J_{CKF}(\theta; Z_1^K, U_0^{K-1}) = \begin{bmatrix} \frac{\partial J_{CKF}}{\partial \theta_1} \\ \frac{\partial J_{CKF}}{\partial \theta_2} \end{bmatrix} \quad (2.26)$$

для уравнения объекта (1.10) вычисляется по формулам (2.13)–(2.22), а для уравнения объекта (1.20) — по формулам (2.13)–(2.20), (2.23)–(2.25).

**Замечание 2.1.** *Вычисление градиента критерия идентификации (2.4) может быть встроено в алгоритм 2.2.*

Снова рассмотрим пример 2.1. Используем дискретную модель (2.11). Проведем идентификацию неизвестного векторного параметра  $\theta = [v, \alpha]^T$  в системе MATLAB градиентным методом при помощи функции минимизации `fminunc`

с параметрами `MaxFunctionEvaluations = 500` и `SpecifyObjectiveGradient = true`. В таблице 2.4 представлены результаты идентификации для серии из 100 экспериментов.

Таблица 2.4. Результаты экспериментов для модели (2.11)

$\sigma^2$	Среднее		RMSE		MAPE	
	$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$5^2$	0.5078	0.2216	0.0461	0.0344	7.2777	13.4619
$1^2$	0.5084	0.2167	0.0173	0.0180	2.7497	8.3890
$0.1^2$	0.5079	0.2159	0.0080	0.0159	1.5757	7.9548
$0.01^2$	0.5078	0.2159	0.0078	0.0159	1.5658	7.9320

Из таблицы видно, что качество идентификации примерно соответствует результатам, полученным алгоритмом SA, но при этом среднее время идентификации на той же аппаратной платформе составило 2.5 с.

## 2.4. Идентификация на основе SVD-модификации фильтра Калмана

При решении практических задач с использованием компьютера предпочтительнее использовать реализации фильтра Калмана с квадратным корнем и UD-реализации, которые численно устойчивы к ошибкам при машинном округлении, вместо стандартной формы алгоритма Калмана [61, 62, 63, 64, 65].

В данном разделе будет рассмотрен новый подход к идентификации параметров дискретных моделей конвективно-диффузионного переноса на основе численно устойчивой SVD-модификации фильтра Калмана.

Рассмотрим SVD-факторизацию [66, теорема 1.1.6]. Любую матрицу  $A \in \mathbb{C}^{m \times n}$  ранга  $r$  можно представить в виде

$$A = W \Sigma V^*, \quad \Sigma = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{m \times n}, \quad S = \text{diag}\{\sigma_1, \dots, \sigma_r\},$$

где  $W \in \mathbb{C}^{m \times m}$ ,  $V \in \mathbb{C}^{n \times n}$  — унитарные матрицы,  $V^*$  — означает сопряженную и транспонированную к  $V$ ,  $S \in \mathbb{R}^{r \times r}$  — вещественная неотрицательная диагональная матрица. Величины  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$  являются сингулярными значениями матрицы  $A$ . Заметим, что если  $r = n$  и/или  $r = m$ , то некоторые из нулевых подматриц в  $\Sigma$  отсутствуют.

Метод сингулярного разложения (или SVD-факторизация) известен как наиболее точный метод факторизации матриц, особенно для матриц, близких к вырожденным. Кроме того, сингулярное разложение существует для любой матрицы, чего нельзя сказать, например, о разложении Холецкого. Поэтому модификации фильтра Калмана на основе SVD-факторизации обладают такой же улучшенной численной устойчивостью к ошибкам машинного округления, что и все известные квадратно-корневые модификации [14], [67]. Кроме того, как указано в [68], SVD-фильтры имеют следующие дополнительные преимущества:

- 1) все собственные значения матриц ковариации ошибок автоматически вычисляются на каждом шаге работы алгоритма фильтрации и могут использоваться для автоматического анализа и/или сокращения исходной модели;
- 2) информационные матрицы (обратные к ковариационным) легко вычисляются путем инверсии диагональных факторов в SVD-разложении, что создает элегантный способ построения алгоритмов информационного типа и фильтров смешанного типа с автоматическим переключением с ковариационного режима фильтрации на информационный.

Впервые идею построения численно устойчивой модификации фильтра Калмана с применением сингулярного разложения предложили Ошман и Бар-Ицхак [69]. Авторы назвали свой вариант SVD-фильтра как  $V$ - $\Lambda$ -фильтр. Для реализации алгоритма необходимо выполнить как SVD-разложение, так и разложение Холецкого, а также минимум три операции матричного обращения. Затем Ошманом была предложена информационная форма  $V$ - $\Lambda$ -фильтра [70]. В [71]  $V$ - $\Lambda$ -фильтр был применен для решения задачи параметрической идентификации линейной дискретной стохастической системы.

Позднее, другие авторы предложили свои варианты SVD-фильтра, соответ-

ствующие стандартному фильтру Калмана [72] и расширенному фильтру Калмана [73]. Указанные модификации во многом схожи с  $V$ - $\Lambda$ -фильтром. Ограничением их применения является требование положительной определенности матриц ковариации шумов в объекте и измерителе на каждой итерации работы алгоритма, поскольку в нем необходимо применять разложение Холецкого для вычисления квадратного корня ковариационной матрицы. Также необходимо выполнить минимум три матричных обращения.

С целью устранения указанных недостатков предыдущих версий SVD-фильтра в [74] был предложен новый, улучшенный вариант SVD-фильтра Калмана. Его отличие от других вариантов состоит в том, что данная модификация фильтра Калмана свободна от выполнения условий  $Q_k \geq 0$  и  $R_k > 0$ , требуемых как в стандартном фильтре Калмана, так и во всех его квадратно-корневых модификациях [67]. Другим значимым преимуществом указанной модификации является наличие всего лишь одного обращения диагональной матрицы в уравнениях фильтра. Как показано в [74], по результатам сравнительного анализа данный вариант SVD-фильтра показал наилучшие результаты в плане численной устойчивости к ошибкам машинного округления на примере решения плохообусловленных задач.

Следует также отметить, что SVD-фильтр подтвердил свою эффективность при решении задач параметрической идентификации [68], [75], задачи оценки состояния и параметров полета летательного аппарата [73], задачи калмановской фильтрации показаний инерциального измерительного модуля (IMU) [76] и др.

Учитывая сказанное выше, построения процедуры параметрической идентификации будет использоваться улучшенный вариант SVD-фильтра [74].

Рассмотрим представление ковариационных матриц ошибок оценивания в виде  $P_k = \Theta_{P_k} D_{P_k} \Theta_{P_k}^T$ , где  $\Theta_{P_k}$  — ортогональная матрица и  $D_{P_k}$  — диагональная матрица, содержащая сингулярные значения матрицы  $P_k$ . Уравнения SVD-фильтра позволяют рекуррентно обновлять SVD-факторы  $\{\Theta_{P_k}, D_{P_k}\}$  матрицы  $P_k$  с помощью сингулярного разложения (процедуры SVD-факторизации). Запишем уравнения SVD-фильтра (алгоритм 2.3) с учетом модели (2.1), (2.2). Блок-схема алгоритма 2.3 представлена на рисунке 2.6.

---

**Алгоритм 2.3.** SVD-модификация фильтра Калмана
 

---

**Вход:**  $\bar{c}_0, \Pi_0$ 

- 1  $\hat{c}_0 = \bar{c}_0, [\Theta_{P_0}, D_{P_0}] = \text{svd}(\Pi_0)$  // Инициализация
- 2 **for**  $k = 1, 2, \dots, K$  **do**
  - // Обновление по времени (экстраполяция)
  - 3  $\begin{bmatrix} D_{P_{k-1}}^{1/2} \Theta_{P_{k-1}}^T F_{k-1}^T \end{bmatrix} = \mathcal{W}_{TU} \begin{bmatrix} D_{P_{k|k-1}}^{1/2} \end{bmatrix} \Theta_{P_{k|k-1}}^T$
  - 4  $\hat{c}_{k|k-1} = F_{k-1} \hat{c}_{k-1} + B_{k-1} u_{k-1}$
  - // Обновление по измерениям (фильтрация)
  - 5  $[\Theta_{R_k}, D_{R_k}] = \text{svd}(R_k)$
  - 6  $\begin{bmatrix} D_{R_k}^{1/2} \Theta_{R_k}^T \\ D_{P_{k|k-1}}^{1/2} \Theta_{P_{k|k-1}}^T H_k^T \end{bmatrix} = \mathcal{W}_{MU}^{(1)} \begin{bmatrix} D_{\Sigma_{\nu,k}}^{1/2} \\ 0 \end{bmatrix} \Theta_{\Sigma_{\nu,k}}^T$
  - 7  $\bar{K}_k = P_{k|k-1} H_k^T \Theta_{\Sigma_{\nu,k}}, K_k = \bar{K}_k D_{\Sigma_{\nu,k}}^{-1} \Theta_{\Sigma_{\nu,k}}^T$
  - 8  $\begin{bmatrix} D_{P_{k|k-1}}^{1/2} \Theta_{P_{k|k-1}}^T (I - K_k H_k)^T \\ D_{R_k}^{1/2} \Theta_{R_k}^T K_k^T \end{bmatrix} = \mathcal{W}_{MU}^{(2)} \begin{bmatrix} D_{P_k}^{1/2} \\ 0 \end{bmatrix} \Theta_{P_k}^T$
  - 9  $\bar{\nu}_k = \Theta_{\Sigma_{\nu,k}}^T (z_k - H_k \hat{c}_{k|k-1}), \hat{c}_k = \hat{c}_{k|k-1} + \bar{K}_k D_{\Sigma_{\nu,k}}^{-1} \bar{\nu}_k$
- 10 **end for**

**Выход:**  $\hat{c}_k, D_{P_k}, \Theta_{P_k}, k = 1, 2, \dots, K$ 


---

Теперь для построения процедуры параметрической идентификации необходимо переписать выражение для вычисления логарифмической функции правдоподобия (2.4) в терминах SVD-фильтра.

Учитывая, что  $\det(\Sigma_{\nu,k}) = \det(D_{\Sigma_{\nu,k}})$  и  $\nu_k^T \Sigma_{\nu,k}^{-1} \nu_k = \bar{\nu}_k^T D_{\Sigma_{\nu,k}}^{-1} \bar{\nu}_k$ , запишем:

$$J_{SVD}(\theta; Z_1^K, U_0^{K-1}) = \frac{Km}{2} \ln(2\pi) + \frac{1}{2} \sum_{k=1}^K \left\{ \ln[\det(D_{\Sigma_{\nu,k}})] + \bar{\nu}_k^T D_{\Sigma_{\nu,k}}^{-1} \bar{\nu}_k \right\}, \quad (2.27)$$

где диагональная матрица  $D_{\Sigma_{\nu,k}}$  и вектор  $\bar{\nu}_k$  доступны на каждом шаге работы алгоритма 2.3.

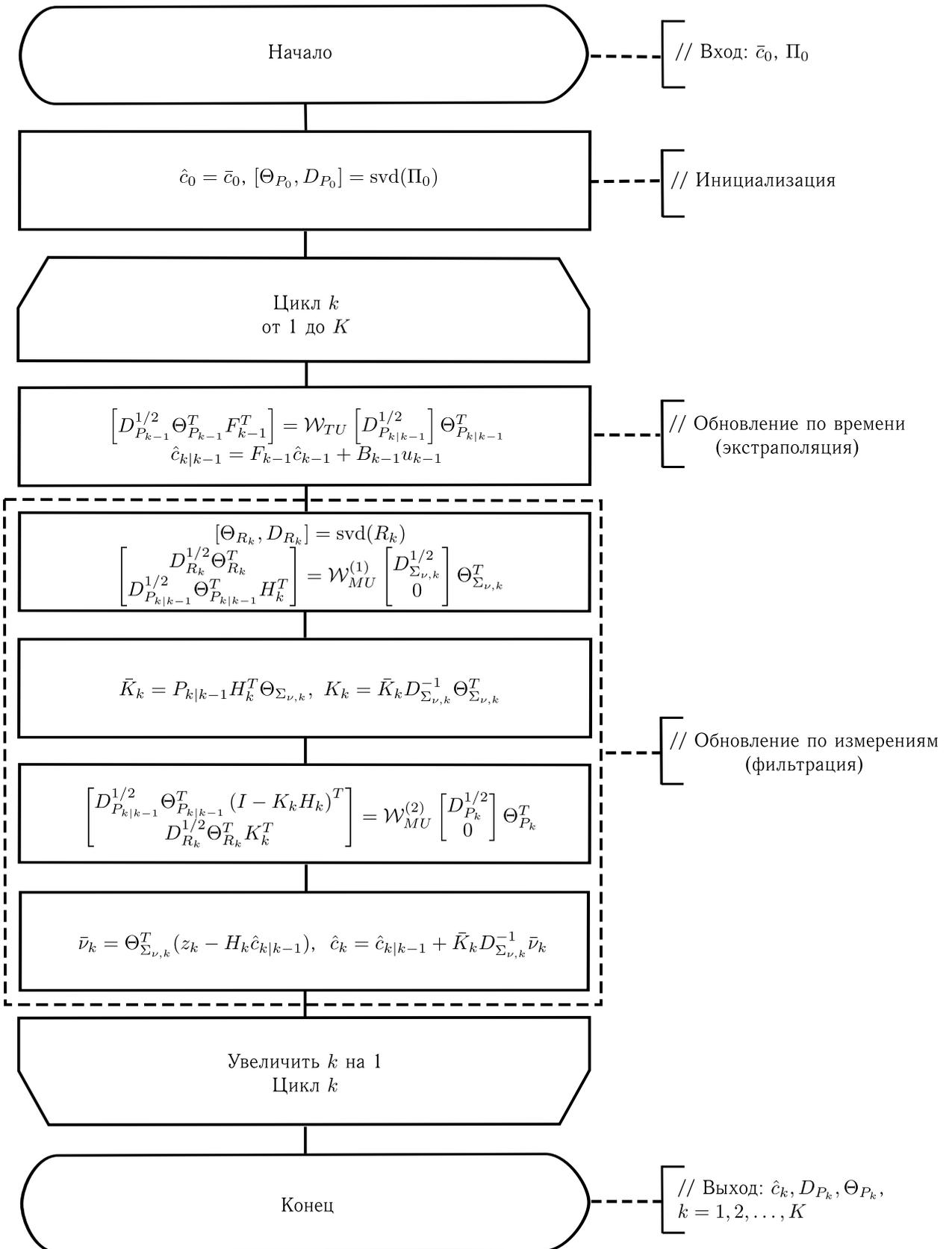


Рис. 2.6. Блок-схема алгоритма 2.3

Вычисление критерия идентификации может быть выполнено на основании следующего утверждения.

**Утверждение 2.2.** Критерий идентификации  $J_{SDV}$  для дискретной линейной стохастической системы (2.1), (2.2) по результатам измерений  $Z_1^K$  может быть вычислен при помощи алгоритма 2.4, при этом последовательность управляющих воздействий  $U_0^{K-1}$  для уравнения объекта (1.10) имеет вид (2.6), а для уравнения объекта (1.20) — вид (2.7).

---

**Алгоритм 2.4.** Вычисление критерия идентификации на основе SVD-модификации фильтра Калмана

---

**Вход:**  $\theta, Z_1^K, U_0^{K-1}, \bar{c}_0, \Pi_0$

$$1 \quad \hat{c}_0 = \bar{c}_0, [\Theta_{P_0}, D_{P_0}] = \text{svd}(\Pi_0), J_{SVD} = Km \ln(2\pi)$$

2 **for**  $k = 1, 2, \dots, K$  **do**

$$3 \quad \begin{bmatrix} D_{P_{k|k-1}}^{1/2} \Theta_{P_{k|k-1}}^T F_{k-1}^T \end{bmatrix} = \mathcal{W}_{TU} \begin{bmatrix} D_{P_{k|k-1}}^{1/2} \end{bmatrix} \Theta_{P_{k|k-1}}^T$$

$$4 \quad \hat{c}_{k|k-1} = F_{k-1} \hat{c}_{k-1} + B_{k-1} u_{k-1}$$

$$5 \quad [\Theta_{R_k}, D_{R_k}] = \text{svd}(R_k)$$

$$6 \quad \begin{bmatrix} D_{R_k}^{1/2} \Theta_{R_k}^T \\ D_{P_{k|k-1}}^{1/2} \Theta_{P_{k|k-1}}^T H_k^T \end{bmatrix} = \mathcal{W}_{MU}^{(1)} \begin{bmatrix} D_{\Sigma_{\nu,k}}^{1/2} \\ 0 \end{bmatrix} \Theta_{\Sigma_{\nu,k}}^T$$

$$7 \quad \bar{K}_k = P_{k|k-1} H_k^T \Theta_{\Sigma_{\nu,k}}^T, K_k = \bar{K}_k D_{\Sigma_{\nu,k}}^{-1} \Theta_{\Sigma_{\nu,k}}^T$$

$$8 \quad \begin{bmatrix} D_{P_{k|k-1}}^{1/2} \Theta_{P_{k|k-1}}^T (I - K_k H_k)^T \\ D_{R_k}^{1/2} \Theta_{R_k}^T K_k^T \end{bmatrix} = \mathcal{W}_{MU}^{(2)} \begin{bmatrix} D_{P_k}^{1/2} \\ 0 \end{bmatrix} \Theta_{P_k}^T$$

$$9 \quad \bar{\nu}_k = \Theta_{\Sigma_{\nu,k}}^T (z_k - H_k \hat{c}_{k|k-1}), \hat{c}_k = \hat{c}_{k|k-1} + \bar{K}_k D_{\Sigma_{\nu,k}}^{-1} \bar{\nu}_k$$

$$10 \quad J_{SVD} = J_{SVD} + \ln[\det(D_{\Sigma_{\nu,k}})] + \bar{\nu}_k^T D_{\Sigma_{\nu,k}}^{-1} \bar{\nu}_k$$

11 **end for**

$$12 \quad J_{SVD} = \frac{1}{2} J_{SVD}$$

**Выход:**  $J_{SVD}(\theta; Z_1^K, U_0^{K-1})$

---

Блок-схема алгоритма 2.4 представлена на рисунке 2.7.

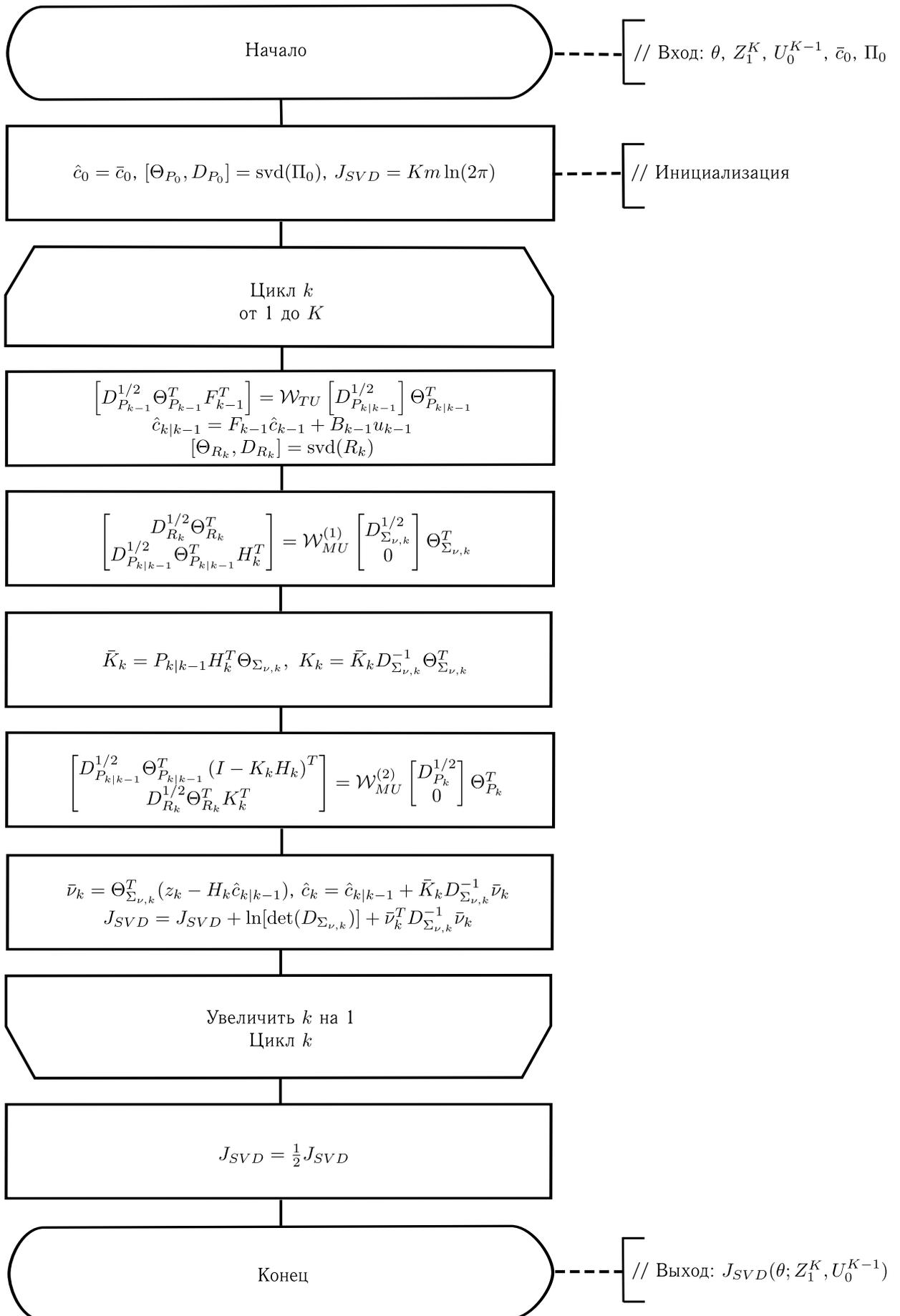


Рис. 2.7. Блок-схема алгоритма 2.4

Рассмотрим на примере работоспособность и преимущества предложенного подхода. Пусть требуется идентифицировать параметры  $v$  и  $\alpha$  следующей модели.

**Пример 2.2.**

$$\frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} = \alpha \frac{\partial^2 c}{\partial x^2}, \quad (2.28)$$

$$c(x, 0) = 0, \quad (2.29)$$

$$c(0, t) = 4t |\sin \pi t|, \quad c(1, t) = t \quad (2.30)$$

или

$$c(0, t) = 4t |\sin \pi t|, \quad \frac{\partial c(1, t)}{\partial x} = -[c(1, t) - t], \quad (2.31)$$

где  $c(x, t)$  — концентрация вещества в одномерном потоке;  $x \in [0; 1]$ ;  $t \in [0; 2]$ ; начальное условие (2.29) соответствует концентрации вещества в начальный момент времени; первое граничное условие (2.30) и (2.31) — периодическому изменению концентрации вещества с возрастающей амплитудой на левом конце отрезка; второе граничное условие (2.30) — линейному возрастанию концентрации вещества на правом конце отрезка; второе граничное условие (2.31) — линейному возрастанию концентрации вещества в окружающей среде на правом конце отрезка.

Пусть  $v = 2$ ,  $\alpha = 1$ . Процесс идентификации будем моделировать в системе MATLAB при помощи специализированного программного комплекса авторской разработки [77]. Зададим в области  $[0; 1] \times [0; 2]$  плоскости  $Oxt$  пространственно-временную сетку с 6 узлами по оси  $Ox$  и 101 узлом по оси  $Ot$  ( $\Delta x = 0.2$ ,  $\Delta t = 0.02$ ) и получим решения рассматриваемой задачи методом конечных разностей для различных комбинаций граничных условий. На рисунке 2.8 приведены графики соответствующих решений.

Рассмотрим модель зашумленных измерений в узлах пространственной сетки, соответствующих первой и последней компонентам вектора состояния. В этом случае для задачи с граничными условиями (2.30) измеритель будет иметь

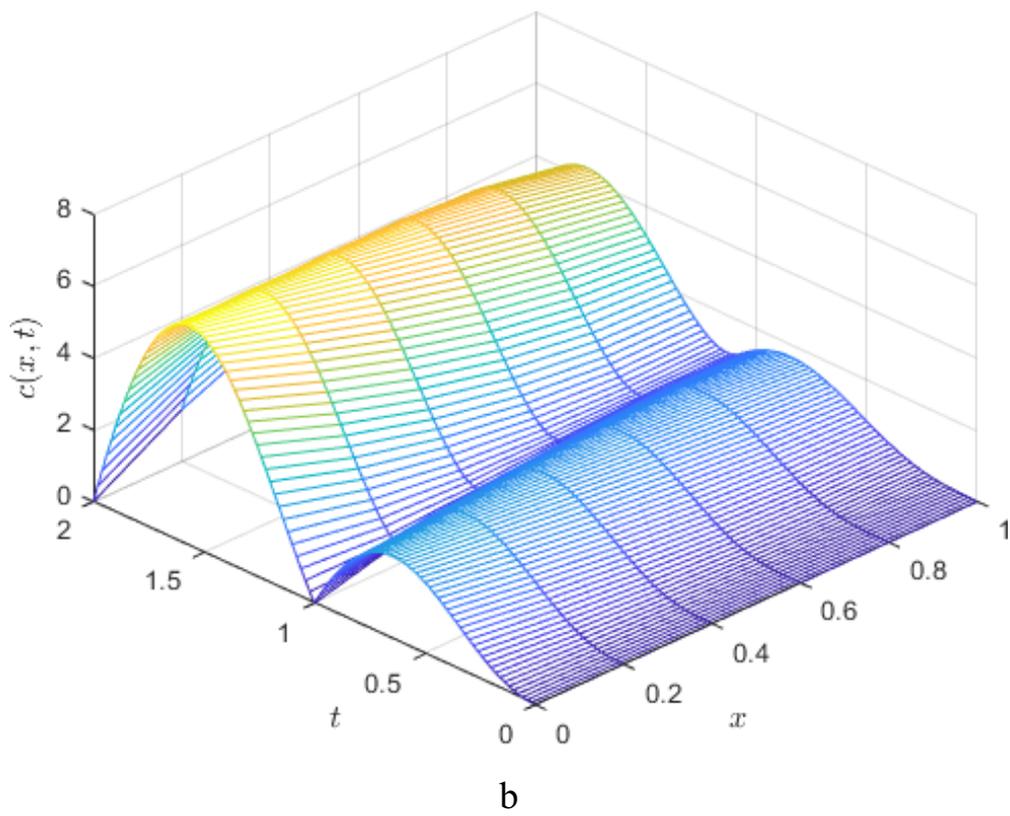
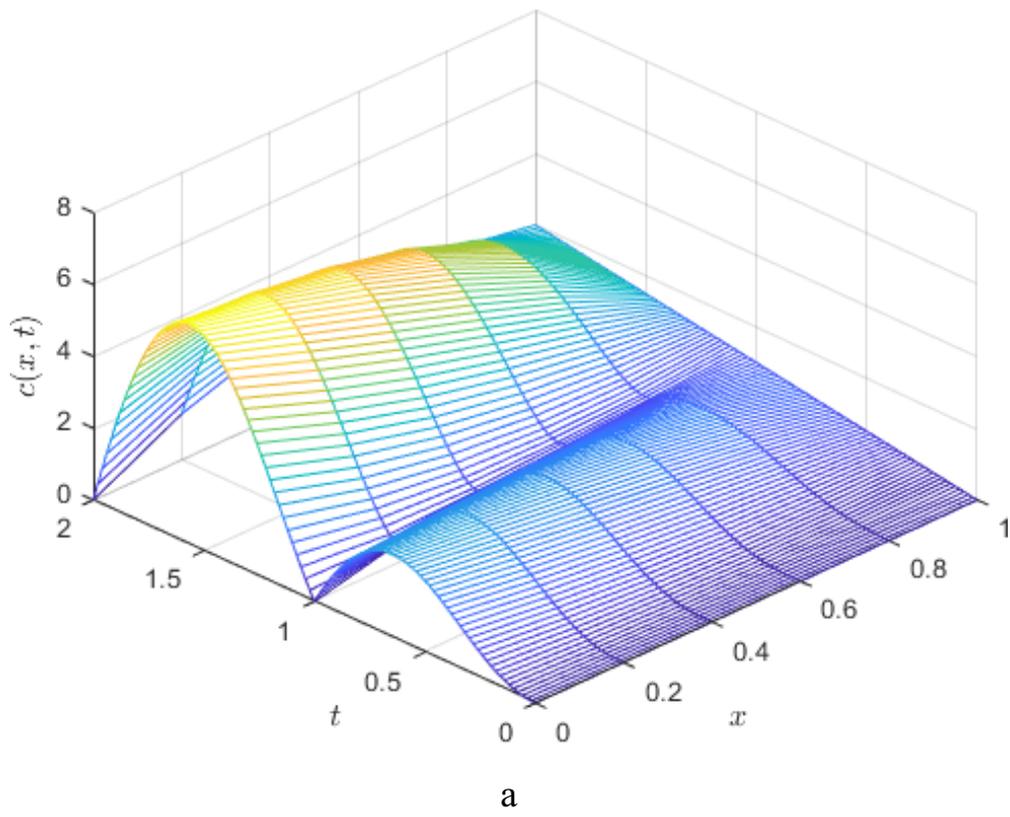


Рис. 2.8. Графики решения задачи с граничными условиями (2.30) (a) и (2.31) (b)

ВИД

$$\begin{bmatrix} z_1^k \\ z_2^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \end{bmatrix} + \begin{bmatrix} \xi_1^k \\ \xi_2^k \end{bmatrix}, \quad k = 1, 2, \dots, K, \quad (2.32)$$

а для задачи с граничными условиями (2.31):

$$\begin{bmatrix} z_1^k \\ z_2^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \\ c_5^k \end{bmatrix} + \begin{bmatrix} \xi_1^k \\ \xi_2^k \end{bmatrix}, \quad k = 1, 2, \dots, K. \quad (2.33)$$

Рассмотрим следующие варианты матрицы ковариации шума  $R_k$  в измерителе:

$$R_0 = \begin{bmatrix} 10^{-2} & 0 \\ 0 & 10^{-2} \end{bmatrix}, R_1 = \begin{bmatrix} \delta & 0 \\ 0 & \delta \end{bmatrix}, R_2 = \begin{bmatrix} \delta & 0 \\ 0 & 10^{-4} \end{bmatrix}, R_3 = \begin{bmatrix} 10^{-4} & 0 \\ 0 & \delta \end{bmatrix}, \quad (2.34)$$

где  $\delta = 10^{-10}, 10^{-11}, \dots, 10^{-16}$ . При уменьшении  $\delta$  погрешность измерений соответствующего сенсора уменьшается.

На рисунке 2.9 приведены примеры графиков смоделированных измерений с матрицей ковариации шума  $R_k = R_0$  для рассматриваемой задачи с различными комбинациями граничных условий.

В качестве критериев идентификации возьмем логарифмические функции правдоподобия (2.4) и (2.27).

В таблицах (2.5)–(2.12) приведены результаты численных экспериментов по идентификации коэффициентов  $v$  и  $\alpha$  для различных значений матрицы  $R_k$  и граничных условий. Минимизация критериев идентификации выполнялась в области  $D(\theta) = \{\theta = [v, \alpha]^T | v \in [0; 5], \alpha \in [0; 5]\}$  при помощи функции `fmincon` системы MATLAB. В качестве начальной точки для функции `fmincon` выбирался центр области  $D$ . Для каждого варианта матрицы  $R$  и каждого значения  $\delta$  выполнялось усреднение найденных параметров и вычисление ошибок RMSE и MAPE по результатам 200 запусков.

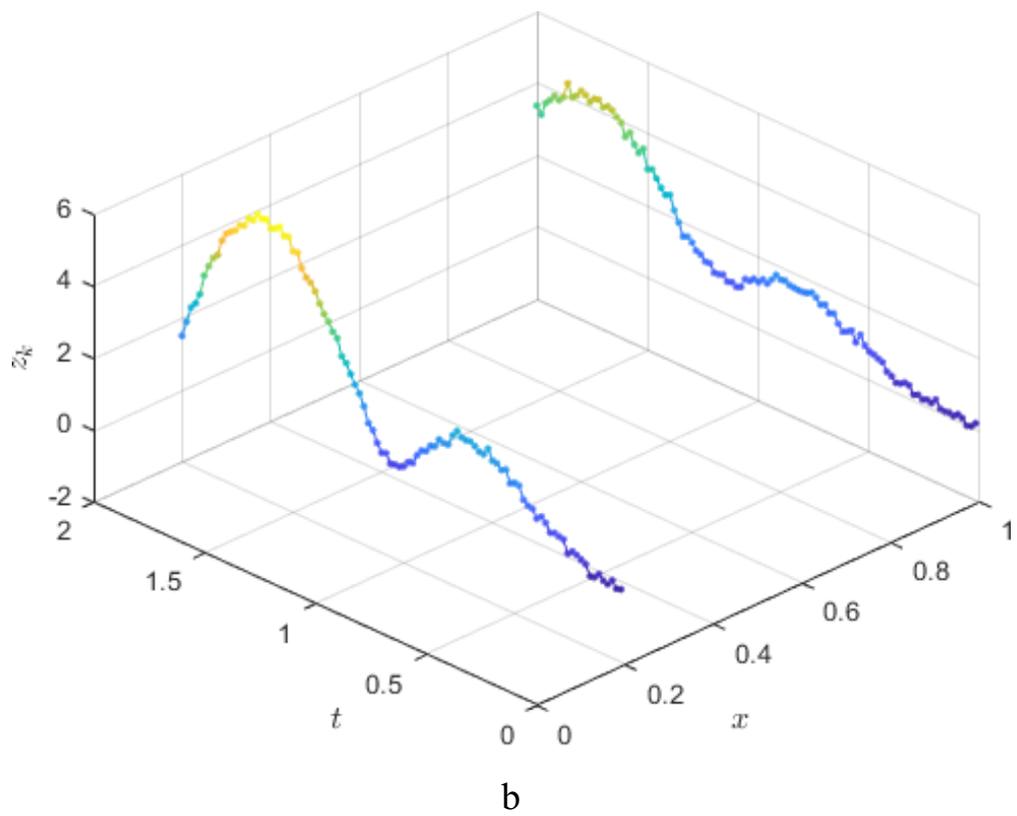
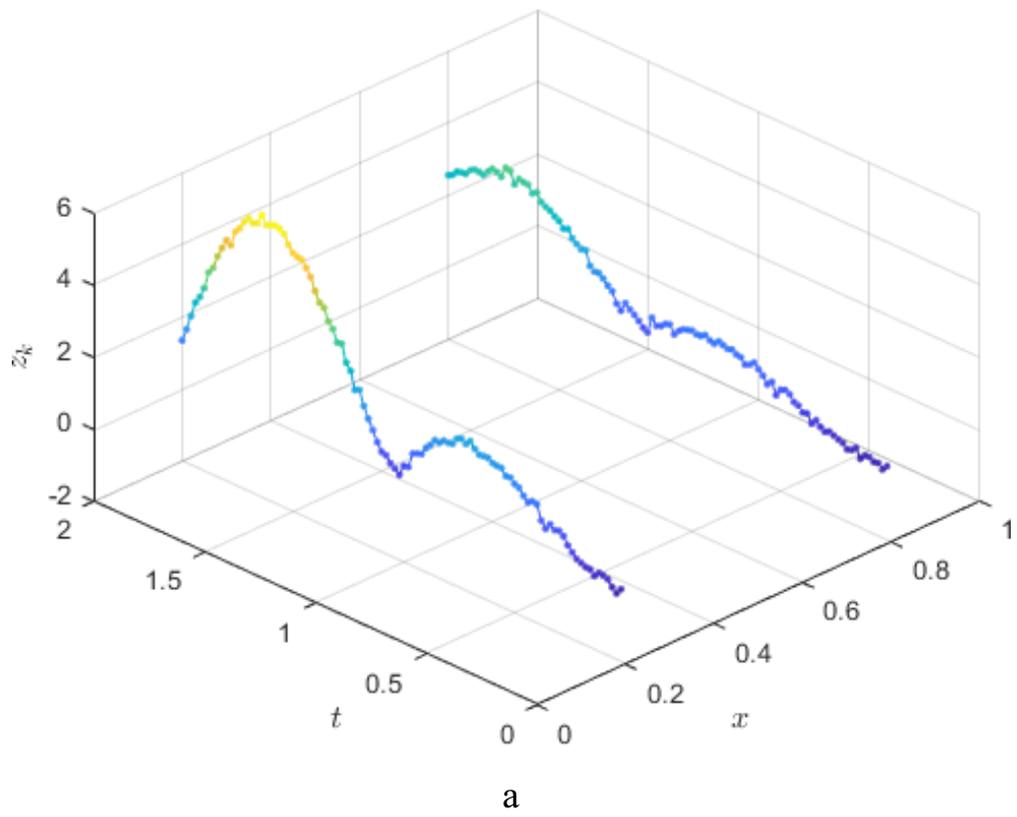


Рис. 2.9. Графики измерений для задачи с граничными условиями (2.30) (a) и (2.31) (b)

Полученные данные численных экспериментов показывают, что сначала для обоих алгоритмов с увеличением точности измерений средние значения идентифицированных параметров  $v$  и  $\alpha$  стремятся к истинным (2 и 1 соответственно), а ошибки RMSE и MAPE уменьшаются. Однако, начиная со значения  $\delta = 10^{-14}$ , для граничных условий (2.30) и  $\delta = 10^{-13}$ , для граничных условий (2.31) ошибки процедуры идентификации на основе стандартного фильтра Калмана начинают быстро нарастать вплоть до аварийного завершения работы функции минимизации `fmincon` (прочерки в таблицах). Данный факт обусловлен расходимостью стандартного фильтра Калмана и, как следствие, некорректным вычислением значений критерия идентификации (2.4). В то же время процедура идентификации на основе SVD-фильтра выполняется корректно для всех значений  $\delta$ .

Заметим, что все вычисления в системе MATLAB по умолчанию проводятся с двойной точностью, однако на практике встречаются ситуации, когда вычисления необходимо проводить с одинарной точностью, например, во встраиваемых системах [76]. В этом случае использование SVD-фильтра будет более предпочтительным.

Таблица 2.5. Средние значения параметров и ошибки идентификации для  $R_k = R_0$  и граничных условий (2.30)

Алгоритм	Среднее		RMSE		MAPE	
	$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
CKF	1.9951	0.9962	$4.63 \cdot 10^{-2}$	$2.23 \cdot 10^{-2}$	1.8242	1.8325
SVD	1.9951	0.9962	$4.63 \cdot 10^{-2}$	$2.23 \cdot 10^{-2}$	1.8242	1.8325

Таблица 2.6. Средние значения параметров и ошибки идентификации для  $R_k = R_0$  и граничных условий (2.31)

Алгоритм	Среднее		RMSE		MAPE	
	$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
CKF	1.9996	0.9984	$4.99 \cdot 10^{-2}$	$2.31 \cdot 10^{-2}$	1.9539	1.8660
SVD	1.9996	0.9984	$4.99 \cdot 10^{-2}$	$2.31 \cdot 10^{-2}$	1.9539	1.8660

Таблица 2.7. Средние значения параметров и ошибки идентификации для  $R_k = R_1$  и граничных условий (2.30)

$\delta$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$10^{-10}$	CKF	2.0000	1.0000	$4.55 \cdot 10^{-6}$	$2.34 \cdot 10^{-6}$	$1.80 \cdot 10^{-4}$	$1.80 \cdot 10^{-4}$
	SVD	2.0000	1.0000	$4.55 \cdot 10^{-6}$	$2.34 \cdot 10^{-6}$	$1.80 \cdot 10^{-4}$	$1.85 \cdot 10^{-4}$
$10^{-11}$	CKF	2.0000	1.0000	$1.47 \cdot 10^{-6}$	$7.09 \cdot 10^{-7}$	$5.91 \cdot 10^{-5}$	$5.78 \cdot 10^{-5}$
	SVD	2.0000	1.0000	$1.47 \cdot 10^{-6}$	$7.10 \cdot 10^{-7}$	$5.91 \cdot 10^{-5}$	$5.78 \cdot 10^{-5}$
$10^{-12}$	CKF	2.0000	1.0000	$5.09 \cdot 10^{-7}$	$1.94 \cdot 10^{-7}$	$1.99 \cdot 10^{-5}$	$1.56 \cdot 10^{-5}$
	SVD	2.0000	1.0000	$5.09 \cdot 10^{-7}$	$1.94 \cdot 10^{-7}$	$1.99 \cdot 10^{-5}$	$1.55 \cdot 10^{-5}$
$10^{-13}$	CKF	2.0000	1.0000	$1.65 \cdot 10^{-7}$	$6.73 \cdot 10^{-8}$	$6.60 \cdot 10^{-6}$	$5.40 \cdot 10^{-6}$
	SVD	2.0000	1.0000	$1.65 \cdot 10^{-7}$	$6.74 \cdot 10^{-8}$	$6.60 \cdot 10^{-6}$	$5.40 \cdot 10^{-6}$
$10^{-14}$	CKF	2.0909	1.1295	$3.72 \cdot 10^{-1}$	$2.45 \cdot 10^{-1}$	13.3856	18.2959
	SVD	2.0000	1.0000	$4.95 \cdot 10^{-8}$	$2.44 \cdot 10^{-8}$	$2.00 \cdot 10^{-6}$	$1.97 \cdot 10^{-6}$
$10^{-15}$	CKF	2.4525	2.4007	$4.92 \cdot 10^{-1}$	1.4422	24.2181	141.0602
	SVD	2.0000	1.0000	$2.84 \cdot 10^{-8}$	$1.38 \cdot 10^{-8}$	$1.24 \cdot 10^{-6}$	$1.21 \cdot 10^{-6}$
$10^{-16}$	CKF	–	–	–	–	–	–
	SVD	2.0000	1.0000	$2.77 \cdot 10^{-8}$	$1.29 \cdot 10^{-8}$	$1.36 \cdot 10^{-6}$	$1.26 \cdot 10^{-6}$

Таблица 2.8. Средние значения параметров и ошибки идентификации для  $R_k = R_2$  и граничных условий (2.30)

$\delta$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$10^{-10}$	CKF	1.9998	1.0002	$8.53 \cdot 10^{-3}$	$3.97 \cdot 10^{-3}$	$3.45 \cdot 10^{-1}$	$3.09 \cdot 10^{-1}$
	SVD	1.9998	1.0002	$8.53 \cdot 10^{-3}$	$3.97 \cdot 10^{-3}$	$3.45 \cdot 10^{-1}$	$3.09 \cdot 10^{-1}$
$10^{-11}$	CKF	1.9996	1.0003	$8.72 \cdot 10^{-3}$	$3.81 \cdot 10^{-3}$	$3.44 \cdot 10^{-1}$	$2.98 \cdot 10^{-1}$
	SVD	1.9996	1.0003	$8.72 \cdot 10^{-3}$	$3.81 \cdot 10^{-3}$	$3.44 \cdot 10^{-1}$	$2.98 \cdot 10^{-1}$
$10^{-12}$	CKF	1.9998	1.0003	$9.17 \cdot 10^{-3}$	$3.58 \cdot 10^{-3}$	$3.70 \cdot 10^{-1}$	$2.91 \cdot 10^{-1}$
	SVD	1.9999	1.0003	$9.17 \cdot 10^{-3}$	$3.58 \cdot 10^{-3}$	$3.70 \cdot 10^{-1}$	$2.91 \cdot 10^{-1}$
$10^{-13}$	CKF	2.0008	0.9999	$9.03 \cdot 10^{-3}$	$3.80 \cdot 10^{-3}$	$3.58 \cdot 10^{-1}$	$3.06 \cdot 10^{-1}$
	SVD	2.0008	0.9999	$9.04 \cdot 10^{-3}$	$3.80 \cdot 10^{-3}$	$3.58 \cdot 10^{-1}$	$3.07 \cdot 10^{-1}$
$10^{-14}$	CKF	2.0011	1.1824	$3.80 \cdot 10^{-1}$	$3.53 \cdot 10^{-1}$	8.5127	18.6786
	SVD	1.9998	1.0000	$8.23 \cdot 10^{-3}$	$3.70 \cdot 10^{-3}$	$3.36 \cdot 10^{-1}$	$2.96 \cdot 10^{-1}$
$10^{-15}$	CKF	2.0556	1.2045	$6.08 \cdot 10^{-1}$	$3.75 \cdot 10^{-1}$	21.0730	23.0515
	SVD	1.9998	1.0000	$8.67 \cdot 10^{-3}$	$3.25 \cdot 10^{-3}$	$3.30 \cdot 10^{-1}$	$2.64 \cdot 10^{-1}$
$10^{-16}$	CKF	2.1827	1.2980	$7.79 \cdot 10^{-1}$	$6.69 \cdot 10^{-1}$	29.2086	41.3944
	SVD	2.0002	1.0001	$9.15 \cdot 10^{-3}$	$3.57 \cdot 10^{-3}$	$3.59 \cdot 10^{-1}$	$2.79 \cdot 10^{-1}$

Таблица 2.9. Средние значения параметров и ошибки идентификации для  $R_k = R_3$  и граничных условий (2.30)

$\delta$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$10^{-10}$	CKF	2.0000	1.0000	$5.88 \cdot 10^{-6}$	$2.97 \cdot 10^{-6}$	$2.40 \cdot 10^{-4}$	$2.40 \cdot 10^{-4}$
	SVD	2.0000	1.0000	$4.79 \cdot 10^{-6}$	$2.97 \cdot 10^{-6}$	$2.40 \cdot 10^{-4}$	$2.40 \cdot 10^{-4}$
$10^{-11}$	CKF	2.0000	1.0000	$1.67 \cdot 10^{-6}$	$9.22 \cdot 10^{-7}$	$6.49 \cdot 10^{-5}$	$7.50 \cdot 10^{-5}$
	SVD	2.0000	1.0000	$1.67 \cdot 10^{-6}$	$9.21 \cdot 10^{-7}$	$6.48 \cdot 10^{-5}$	$7.49 \cdot 10^{-5}$
$10^{-12}$	CKF	2.0000	1.0000	$5.63 \cdot 10^{-7}$	$2.94 \cdot 10^{-7}$	$2.25 \cdot 10^{-5}$	$2.41 \cdot 10^{-5}$
	SVD	2.0000	1.0000	$5.63 \cdot 10^{-7}$	$2.94 \cdot 10^{-7}$	$2.25 \cdot 10^{-5}$	$2.42 \cdot 10^{-5}$
$10^{-13}$	CKF	2.0000	1.0000	$2.04 \cdot 10^{-7}$	$9.85 \cdot 10^{-8}$	$8.06 \cdot 10^{-6}$	$7.78 \cdot 10^{-6}$
	SVD	2.0000	1.0000	$2.05 \cdot 10^{-7}$	$9.85 \cdot 10^{-8}$	$8.08 \cdot 10^{-6}$	$7.78 \cdot 10^{-6}$
$10^{-14}$	CKF	2.4218	2.3225	$5.04 \cdot 10^{-1}$	1.3914	23.2788	132.5812
	SVD	2.0000	1.0000	$6.59 \cdot 10^{-8}$	$3.38 \cdot 10^{-8}$	$2.60 \cdot 10^{-6}$	$2.56 \cdot 10^{-6}$
$10^{-15}$	CKF	2.0530	1.2871	$6.02 \cdot 10^{-1}$	$3.99 \cdot 10^{-1}$	24.6737	30.9475
	SVD	2.0000	1.0000	$4.32 \cdot 10^{-8}$	$2.13 \cdot 10^{-8}$	$1.94 \cdot 10^{-6}$	$1.93 \cdot 10^{-6}$
$10^{-16}$	CKF	3.0785	1.6264	1.1877	0.6924	55.5319	63.5329
	SVD	2.0000	1.0000	$3.54 \cdot 10^{-8}$	$1.79 \cdot 10^{-8}$	$1.74 \cdot 10^{-6}$	$1.76 \cdot 10^{-6}$

Таблица 2.10. Средние значения параметров и ошибки идентификации для  $R_k = R_1$  и граничных условий (2.31)

$\delta$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$10^{-10}$	CKF	2.0000	1.0000	$5.06 \cdot 10^{-6}$	$2.23 \cdot 10^{-6}$	$2.03 \cdot 10^{-4}$	$1.81 \cdot 10^{-4}$
	SVD	2.0000	1.0000	$5.06 \cdot 10^{-6}$	$2.23 \cdot 10^{-6}$	$2.03 \cdot 10^{-4}$	$1.81 \cdot 10^{-4}$
$10^{-11}$	CKF	2.0000	1.0000	$1.47 \cdot 10^{-6}$	$7.24 \cdot 10^{-7}$	$5.80 \cdot 10^{-5}$	$5.89 \cdot 10^{-5}$
	SVD	2.0000	1.0000	$1.47 \cdot 10^{-6}$	$7.25 \cdot 10^{-7}$	$5.80 \cdot 10^{-5}$	$5.90 \cdot 10^{-5}$
$10^{-12}$	CKF	2.2732	1.8860	$5.42 \cdot 10^{-1}$	1.1367	19.9823	88.8022
	SVD	2.0000	1.0000	$4.81 \cdot 10^{-7}$	$2.51 \cdot 10^{-7}$	$1.80 \cdot 10^{-5}$	$1.97 \cdot 10^{-5}$
$10^{-13}$	CKF	2.0295	1.5567	$4.69 \cdot 10^{-1}$	$8.75 \cdot 10^{-1}$	15.5471	56.4061
	SVD	2.0000	1.0000	$1.54 \cdot 10^{-7}$	$7.30 \cdot 10^{-8}$	$6.09 \cdot 10^{-6}$	$5.92 \cdot 10^{-6}$
$10^{-14}$	CKF	1.7208	1.8216	$9.00 \cdot 10^{-1}$	1.06342	36.9710	85.6284
	SVD	2.0000	1.0000	$5.17 \cdot 10^{-8}$	$2.34 \cdot 10^{-8}$	$2.06 \cdot 10^{-6}$	$1.85 \cdot 10^{-6}$
$10^{-15}$	CKF	–	–	–	–	–	–
	SVD	2.0000	1.0000	$1.86 \cdot 10^{-8}$	$8.75 \cdot 10^{-9}$	$7.55 \cdot 10^{-7}$	$7.05 \cdot 10^{-7}$
$10^{-16}$	CKF	–	–	–	–	–	–
	SVD	2.0000	1.0000	$1.02 \cdot 10^{-8}$	$6.07 \cdot 10^{-9}$	$4.54 \cdot 10^{-7}$	$5.55 \cdot 10^{-7}$

Таблица 2.11. Средние значения параметров и ошибки идентификации для  $R_k = R_2$  и граничных условий (2.31)

$\delta$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$10^{-10}$	CKF	1.9998	1.0000	$1.12 \cdot 10^{-2}$	$6.20 \cdot 10^{-3}$	$4.48 \cdot 10^{-1}$	$5.00 \cdot 10^{-1}$
	SVD	1.9998	1.0000	$1.12 \cdot 10^{-2}$	$6.20 \cdot 10^{-3}$	$4.48 \cdot 10^{-1}$	$5.00 \cdot 10^{-1}$
$10^{-11}$	CKF	2.0011	1.0004	$1.20 \cdot 10^{-2}$	$6.30 \cdot 10^{-3}$	$4.82 \cdot 10^{-1}$	$5.02 \cdot 10^{-1}$
	SVD	2.0011	1.0004	$1.20 \cdot 10^{-2}$	$6.30 \cdot 10^{-3}$	$4.82 \cdot 10^{-1}$	$5.02 \cdot 10^{-1}$
$10^{-12}$	CKF	2.0867	1.1835	$4.46 \cdot 10^{-1}$	$4.78 \cdot 10^{-1}$	8.9741	19.0136
	SVD	2.0008	1.0000	$1.18 \cdot 10^{-2}$	$5.00 \cdot 10^{-3}$	$4.63 \cdot 10^{-1}$	$4.60 \cdot 10^{-1}$
$10^{-13}$	CKF	1.7084	1.4499	$8.48 \cdot 10^{-1}$	$6.67 \cdot 10^{-1}$	28.9005	45.2698
	SVD	2.0000	1.0000	$1.14 \cdot 10^{-2}$	$6.37 \cdot 10^{-3}$	$4.53 \cdot 10^{-1}$	$5.07 \cdot 10^{-1}$
$10^{-14}$	CKF	1.3955	1.3397	$8.62 \cdot 10^{-1}$	$5.20 \cdot 10^{-1}$	32.7939	34.9237
	SVD	1.9994	1.0004	$1.19 \cdot 10^{-2}$	$6.03 \cdot 10^{-3}$	$4.54 \cdot 10^{-1}$	$4.66 \cdot 10^{-1}$
$10^{-15}$	CKF	1.5160	1.2420	$8.89 \cdot 10^{-1}$	$4.28 \cdot 10^{-1}$	37.9516	30.0707
	SVD	1.9988	1.0006	$1.12 \cdot 10^{-2}$	$6.11 \cdot 10^{-3}$	$4.40 \cdot 10^{-1}$	$4.85 \cdot 10^{-1}$
$10^{-16}$	CKF	–	–	–	–	–	–
	SVD	2.0004	1.0001	$1.04 \cdot 10^{-2}$	$6.32 \cdot 10^{-3}$	$4.18 \cdot 10^{-1}$	$5.09 \cdot 10^{-1}$

Таблица 2.12. Средние значения параметров и ошибки идентификации для  $R_k = R_3$  и граничных условий (2.31)

$\delta$	Алгоритм	Среднее		RMSE		MAPE	
		$v$	$\alpha$	$v$	$\alpha$	$v$	$\alpha$
$10^{-10}$	CKF	2.0000	1.0000	$5.90 \cdot 10^{-6}$	$2.82 \cdot 10^{-6}$	$2.38 \cdot 10^{-4}$	$2.24 \cdot 10^{-4}$
	SVD	2.0000	1.0000	$5.90 \cdot 10^{-6}$	$2.82 \cdot 10^{-6}$	$2.38 \cdot 10^{-4}$	$2.24 \cdot 10^{-4}$
$10^{-11}$	CKF	2.0000	1.0000	$1.76 \cdot 10^{-6}$	$8.62 \cdot 10^{-7}$	$7.07 \cdot 10^{-5}$	$6.86 \cdot 10^{-5}$
	SVD	2.0000	1.0000	$1.76 \cdot 10^{-6}$	$8.62 \cdot 10^{-7}$	$7.07 \cdot 10^{-5}$	$6.86 \cdot 10^{-5}$
$10^{-12}$	CKF	2.2020	1.7162	$4.04 \cdot 10^{-1}$	1.0238	13.9583	71.8292
	SVD	2.0000	1.0000	$6.42 \cdot 10^{-7}$	$2.59 \cdot 10^{-7}$	$2.65 \cdot 10^{-5}$	$2.05 \cdot 10^{-5}$
$10^{-13}$	CKF	2.4736	2.4251	$4.86 \cdot 10^{-1}$	1.4618	23.6781	142.5113
	SVD	2.0000	1.0000	$1.64 \cdot 10^{-7}$	$9.01 \cdot 10^{-8}$	$6.53 \cdot 10^{-6}$	$7.16 \cdot 10^{-6}$
$10^{-14}$	CKF	1.9462	2.2192	$7.26 \cdot 10^{-1}$	1.3157	29.1861	121.9198
	SVD	2.0000	1.0000	$5.46 \cdot 10^{-8}$	$2.38 \cdot 10^{-8}$	$2.21 \cdot 10^{-6}$	$1.91 \cdot 10^{-6}$
$10^{-15}$	CKF	2.3016	2.2533	$6.05 \cdot 10^{-1}$	1.3677	27.1425	128.7952
	SVD	2.0000	1.0000	$1.88 \cdot 10^{-8}$	$9.76 \cdot 10^{-9}$	$7.56 \cdot 10^{-7}$	$7.76 \cdot 10^{-7}$
$10^{-16}$	CKF	–	–	–	–	–	–
	SVD	2.0000	1.0000	$1.12 \cdot 10^{-8}$	$5.98 \cdot 10^{-9}$	$4.86 \cdot 10^{-7}$	$5.43 \cdot 10^{-7}$

## 2.5. Заключение и выводы к главе 2

В данной главе была рассмотрена задача идентификации неизвестных коэффициентов уравнения (1.1). Предлагаемый подход заключается в переходе от исходных непрерывных моделей (1.1)–(1.3) и (1.1)–(1.4) к моделям, описываемым дискретными линейными стохастическими системами вида (2.1)–(2.2), в которых вектор  $\theta = [v, \alpha]^T$  является неизвестным параметром.

Для идентификации неизвестного векторного параметра  $\theta$  построенных дискретных моделей предлагается использовать метод максимального правдоподобия, основанный на минимизации критерия идентификации в виде отрицательной логарифмической функции правдоподобия, вычисляемый на основе величин, получаемых стандартным фильтром Калмана или его модификацией.

В разделе 2.2 для рассматриваемых моделей сформулировано утверждение 2.1 и алгоритм 2.2 для вычисления критерия идентификации  $J_{CKF}$  (2.4) на основе стандартного фильтра Калмана и рассмотрен процесс решения задач идентификации коэффициентов  $v$  и  $\alpha$  с использованием безградиентных метэвристических методов оптимизации модуля Optimization Toolbox в системе MATLAB.

В разделе 2.3 получены выражения и сформулировано предложение 2.1 для вычисления градиента критерия идентификации  $J_{CKF}$ , а также рассмотрен процесс решения задач идентификации коэффициентов  $v$  и  $\alpha$  с использованием градиентного метода оптимизации в системе MATLAB.

В разделе 2.4 рассмотрен алгоритм 2.3 численно устойчивой модификации фильтра Калмана (SVD-фильтр) и формулирован алгоритм 2.4 вычисления критерия идентификации  $J_{SVD}$  (2.27) на ее основе. Проведен сравнительный анализ решения задач идентификации коэффициентов  $v$  и  $\alpha$  с использованием критериев идентификации  $J_{CKF}$  и  $J_{SVD}$ .

Результаты численных экспериментов показывают работоспособность предложенных подходов. Все рассмотренные алгоритмы позволяют идентифицировать неизвестные коэффициенты конвекции и диффузии, при этом на качество идентификации влияют как параметры дискретизации исходной модели, так и уровень шума в уравнении измерений. В ряде случаев, например, при прове-

дении вычислений с одинарной точностью, более предпочтительным является использование критерия идентификации на основе SVD-модификации фильтра Калмана, обладающей улучшенной численной устойчивостью к ошибкам машинного округления по сравнению со стандартным фильтром.

## Глава 3

# Идентификация граничных условий

### 3.1. Постановка задачи

Рассмотрим следующую задачу параметрической идентификации.

**Задача 3.1.** *Определить функции  $f(t)$  и  $g(t)$ , входящие в граничные условия (1.3) или (1.4) по результатам зашумленных измерений значений функции  $c(x, t)$  в отдельных точках рассматриваемого отрезка в последовательные моменты времени (коэффициенты  $\nu$ ,  $\alpha$ ,  $\lambda$  предполагаются известными).*

Заметим, что в отличие от классических постановок нас интересуют не аналитические выражения для функций  $f(t)$  и  $g(t)$ , а оценки их численных значений в последовательные дискретные моменты времени.

Одной из наиболее известных и актуальных задач для систем (1.5), (1.6) является задача оценивания (фильтрации) вектора состояния по результатам зашумленных измерений. Однако, при решении многих прикладных задач, вектор входных воздействий также является неизвестным, что приводит к необходимости его оценивания. Для решения данной задачи могут быть использованы различные подходы.

Первый подход к оценке вектора входных воздействий в дискретных динамических системах заключается во включении неизвестных входных воздействий в вектор состояния, что предполагает наличие априорной информации о динамике входных воздействий. В этом случае для решения задачи может использоваться расширенный фильтр Калмана. Чтобы уменьшить его вычислительные затраты, в [78] был предложен двухступенчатый фильтр Калмана, в котором оценка состояния и неизвестного входа разделены.

В случае, когда нет априорной информации об эволюции неизвестного входа, используется другой подход. В [79] С. Гиллейнсом и Б. Де-Мором был предложен алгоритм для совместной оценки векторов состояния и входных воздействий. Данная работа явилась развитием результатов, полученных в [80] и

[81], но отличается тем, что в ней предложен рекуррентный фильтр с отдельной оценкой векторов состояния и входных воздействий. В [82] предложенный авторами алгоритм применялся для идентификации граничного условия двумерной задачи теплопроводности.

В разделе 3.2 будет рассмотрено применение алгоритма Гиллейнса – Де-Мора для идентификации неизвестных граничных условий (1.3) и (1.4), а в разделе 3.3 будет получена его квадратно-корневая модификация. Основные результаты данной главы содержатся в работах [38, 39, 40, 83, 84, 85].

### **3.2. Идентификация на основе алгоритма одновременного оценивания состояния и входных воздействий**

Алгоритм Гиллейнса – Де-Мора состоит из трех последовательных этапов, повторяемых в цикле:

- 1) обновление оценки вектора состояния по времени;
- 2) оценка вектора неизвестного входного воздействия;
- 3) обновление оценки вектора состояния по измерениям.

В работе [79] рассмотрены два варианта алгоритма, в которых этапы 1 и 2 совпадают, а этапы 3 различаются. В первом варианте на этапе 2 получается MVU-оценка (MVU — minimum-variance unbiased) вектора входных воздействий  $\hat{u}_{k-1}$ , а на этапе 3 — несмещенная оценка вектора состояния  $\hat{c}_k$ . Во втором варианте алгоритма на этапе 3 за счет более сложных вычислений получается MVU-оценка вектора состояния.

Пусть для любого  $k$  выполняется условие

$$\text{rank } H_k B_{k-1} = \text{rank } B_{k-1} = r. \quad (3.1)$$

Обозначим через  $I_n$ ,  $I_m$ ,  $I_p$  единичные матрицы порядка  $n$ ,  $m$  и  $p$  соответственно ( $p = m - r$ ). Тогда алгоритм Гиллейнса – Де-Мора для системы (1.5), (1.6) может быть сформулирован следующим образом.

---

**Алгоритм 3.1.** Алгоритм Гиллейнса – Де-Мора (вариант 1)
 

---

**Вход:**  $\bar{c}_0, \Pi_0$ 

 1  $\hat{c}_0 = \bar{c}_0, P_0 = \Pi_0$  // Инициализация

 2 **for**  $k = 1, 2, \dots, K$  **do**

// Прогноз оценки вектора состояния

 3  $\hat{c}_{k|k-1} = F_{k-1}\hat{c}_{k-1}$ 

 4  $P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$ 

// Оценка вектора входных воздействий

 5  $\tilde{R}_k = H_k P_{k|k-1} H_k^T + R_k$ 

 6  $D_{k-1} = (B_{k-1}^T H_k^T \tilde{R}_k^{-1} H_k B_{k-1})^{-1}$ 

 7  $M_k = D_{k-1} B_{k-1}^T H_k^T \tilde{R}_k^{-1}$ 

 8  $\hat{u}_{k-1} = M_k(z_k - H_k \hat{c}_{k|k-1})$ 

// Коррекция оценки вектора состояния

 9  $K_k = P_{k|k-1} H_k^T \tilde{R}_k^{-1}$ 

 10  $\hat{c}_k^* = \hat{c}_{k|k-1} + B_{k-1} \hat{u}_{k-1}$ 

 11  $P_k^* = (I_n - K_k H_k) P_{k|k-1}$ 

 12  $\hat{c}_k = \hat{c}_k^* + K_k(z_k - H_k \hat{c}_k^*)$ 

 13  $P_k = P_k^* + (I_n - K_k H_k) B_{k-1} D_{k-1} B_{k-1}^T (I_n - K_k H_k)^T$ 

 14 **end for**
**Выход:**  $\hat{c}_k, P_k, \hat{u}_{k-1}, D_{k-1}, k = 1, 2, \dots, K$ 


---

Блок-схема алгоритма 3.1 приведена на рисунке 3.1.

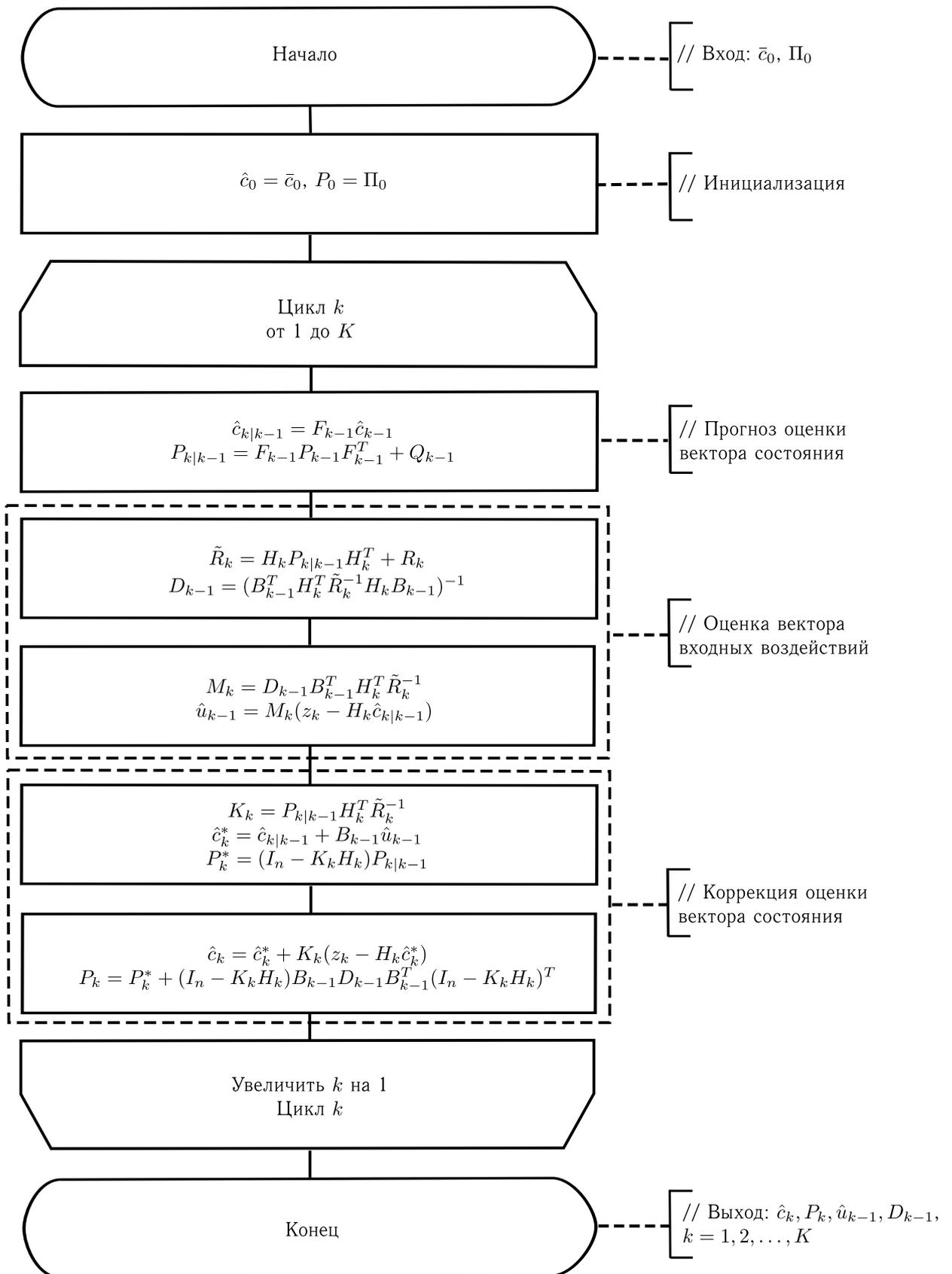


Рис. 3.1. Блок-схема алгоритма 3.1

---

**Алгоритм 3.2.** Алгоритм Гиллейнса – Де-Мора (вариант 2)
 

---

**Вход:**  $\bar{c}_0, \Pi_0$ 

- 1  $\hat{c}_0 = \bar{c}_0, P_0 = \Pi_0$  // Инициализация
- 2 **for**  $k = 1, 2, \dots, K$  **do**
  - // Прогноз оценки вектора состояния
  - 3  $\hat{c}_{k|k-1} = F_{k-1}\hat{c}_{k-1}$
  - 4  $P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$
  - // Оценка вектора входных воздействий
  - 5  $\tilde{R}_k = H_k P_{k|k-1} H_k^T + R_k$
  - 6  $D_{k-1} = (B_{k-1}^T H_k^T \tilde{R}_k^{-1} H_k B_{k-1})^{-1}$
  - 7  $M_k = D_{k-1} B_{k-1}^T H_k^T \tilde{R}_k^{-1}$
  - 8  $\hat{u}_{k-1} = M_k(z_k - H_k \hat{c}_{k|k-1})$
  - // Коррекция оценки вектора состояния
  - 9  $\hat{c}_k^* = \hat{c}_{k|k-1} + B_{k-1} \hat{u}_{k-1}$
  - 10  $P_k^* = (I_n - B_{k-1} M_k H_k) P_{k|k-1} (I_n - B_{k-1} M_k H_k)^T + B_{k-1} M_k R_k M_k^T B_{k-1}^T$
  - 11  $\tilde{R}_k^* = (I_m - H_k B_{k-1} M_k) \tilde{R}_k (I_m - H_k B_{k-1} M_k)^T$
  - 12  $S_k^* = -B_{k-1} M_k R_k$
  - 13  $\alpha_k = [0 \ I_p] U_k^T \tilde{S}_k^{-1}$
  - 14  $K_k = (P_k^* H_k^T + S_k^*) \alpha_k^T (\alpha_k \tilde{R}_k^* \alpha_k^T)^{-1} \alpha_k$
  - 15  $\hat{c}_k = \hat{c}_k^* + K_k(z_k - H_k \hat{c}_k^*)$
  - 16  $P_k = P_k^* - K_k (P_k^* H_k^T + S_k^*)^T$
- 17 **end for**

**Выход:**  $\hat{c}_k, P_k, \hat{u}_{k-1}, D_{k-1}, k = 1, 2, \dots, K$ 


---

В строке (13) матрица  $\tilde{S}_k$  такова, что  $\tilde{S}_k \tilde{S}_k^T = \tilde{R}_k$ , а  $U_k$  — ортогональная матрица, составленная из левых сингулярных векторов матрицы  $\tilde{S}_k^{-1} H_k B_{k-1}$ .

Блок-схема алгоритма 3.2 приведена на рисунке 3.2.

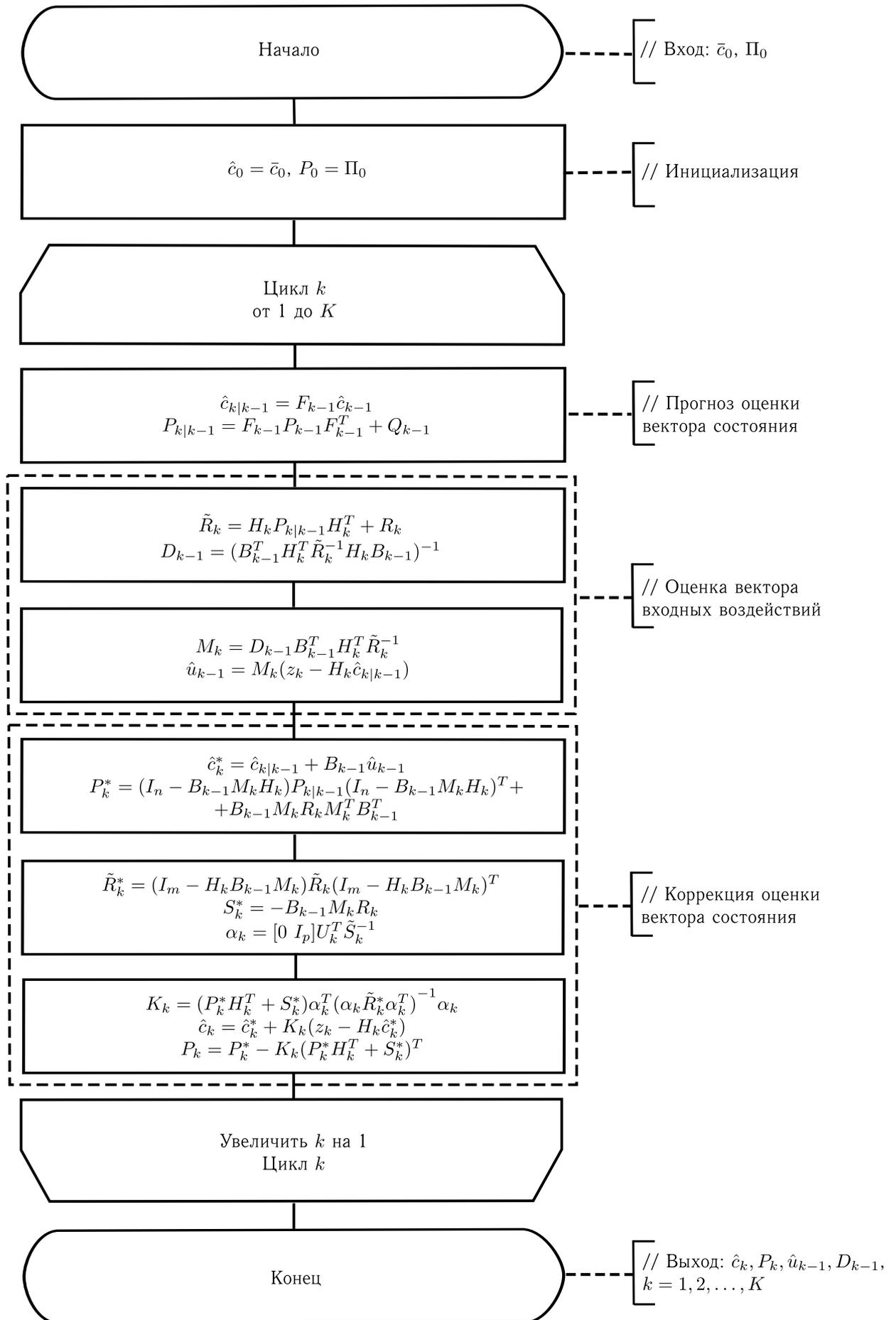


Рис. 3.2. Блок-схема алгоритма 3.2

Рассмотрим применение алгоритма Гиллейнса–Де-Мора на следующем примере.

**Пример 3.1.**

$$\frac{\partial c}{\partial t} + 2 \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}, \quad (3.2)$$

$$c(x, 0) = 0, \quad (3.3)$$

$$c(0, t) = 2 |\sin(10t)|, \quad c(1, t) = 0, \quad (3.4)$$

$$x \in [0; 1], \quad t \in [0; 1],$$

где  $c(x, t)$  — концентрация вещества в одномерном потоке, текущем слева направо. Условие (3.3) означает, что начальная концентрация вещества равна нулю на всем отрезке, первое граничное условие (3.4) соответствует периодическому повышению концентрации на левом конце рассматриваемого отрезка, а второе — поглощающей стенке на правом конце.

Зададим пространственную сетку с шагом  $\Delta x = 0.2$ , состоящую из четырех внутренних ( $x_1 = 0.2, x_2 = 0.4, x_3 = 0.6, x_4 = 0.8$ ) и двух граничных ( $x_0 = 0, x_5 = 1$ ) узлов. Шаг  $\Delta t$  с учетом условия устойчивости конечно-разностной схемы выберем равным  $\frac{\Delta x^2}{4\alpha} = 0.01$ .

Рассмотрим модель измерений с двумя сенсорами ( $m = 2$ ) в первом и последнем внутренних узлах ( $x = 0.2$  и  $x = 0.8$ ). Заметим, что при такой модели измерений условие (3.1) выполняется. Система (1.5), (1.6) в этом случае будет иметь следующий вид:

$$\left\{ \begin{array}{l} \begin{array}{l} \begin{bmatrix} c_k^1 \\ c_k^2 \\ c_k^3 \\ c_k^4 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.2 & 0 & 0 \\ 0.3 & 0.5 & 0.2 & 0 \\ 0 & 0.3 & 0.5 & 0.2 \\ 0 & 0 & 0.3 & 0.5 \end{bmatrix} \begin{bmatrix} c_{k-1}^1 \\ c_{k-1}^2 \\ c_{k-1}^3 \\ c_{k-1}^4 \end{bmatrix} + \begin{bmatrix} 0.3 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.2 \end{bmatrix} \begin{bmatrix} f^{k-1} \\ g^{k-1} \end{bmatrix}, \\ \\ \begin{bmatrix} z_k^1 \\ z_k^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_k^1 \\ c_k^2 \\ c_k^3 \\ c_k^4 \end{bmatrix} + \begin{bmatrix} \xi_k^1 \\ \xi_k^2 \end{bmatrix}, \quad k = 1, 2, \dots, 100. \end{array} \right.$$

Воспользуемся алгоритмом 3.1. На рис. 3.3 приведен график решения прямой задачи (3.2)–(3.4) методом конечных разностей для  $t \in [0; 1]$  ( $k = 1, 2, \dots, 100$ ). На рисунке 3.4 приведены графики зашумленных измерений в первом и последнем внутренних узлах, на рисунках 3.5 и 3.6 — результаты оценивания граничных условий, а на рисунке 3.7 — график оценки решения (во внутренних узлах и на границе), полученные алгоритмом 3.2 с ковариацией шума  $R_k = \text{diag}(0.03^2, 0.03^2)$ .

Заметим, что для рассматриваемой модели на каждой итерации алгоритма в строке 8 вычисляется оценка вектора входных воздействий для предыдущего момента времени, а в строке 15 получается оценка вектора состояния для текущего момента времени. Таким образом, оценка граничных значений функции  $c(x, t)$  (входных воздействий в модели) производится с запаздыванием на один шаг и на последней итерации алгоритма ее значение оказывается неопределенным.

В таблице 3.1 приведены среднеквадратические ошибки оценивания (RMSE) значений функции  $c(x, t)$  в граничных и внутренних узлах сетки и нормализованная среднеквадратическая ошибка (nRMSE) по результатам 100 экспериментов при различных ковариациях шума:  $R_1 = \text{diag}(0.3^2, 0.3^2)$ ,  $R_2 = \text{diag}(0.03^2, 0.03^2)$ ,  $R_3 = \text{diag}(0.003^2, 0.003^2)$ .

Таблица 3.1. Ошибки оценивания

	RMSE <sub>0</sub>	RMSE <sub>1</sub>	RMSE <sub>2</sub>	RMSE <sub>3</sub>	RMSE <sub>4</sub>	RMSE <sub>5</sub>	nRMSE
$R_1$	1.1124	0.2969	0.1079	0.0846	0.2950	1.6624	2.0481
$R_2$	0.1126	0.0300	0.0109	0.0086	0.0296	0.1666	0.2059
$R_3$	0.0110	0.0030	0.0011	0.0009	0.0030	0.0169	0.0207

Значения RMSE<sub>*i*</sub> вычислялись по формуле

$$\text{RMSE}_i = \sqrt{\frac{1}{MK} \sum_{j=1}^M \sum_{k=1}^K (c_{i,j}^k - \hat{c}_{i,j}^k)^2},$$

где  $M$  — число экспериментов,  $K$  — число измерений.

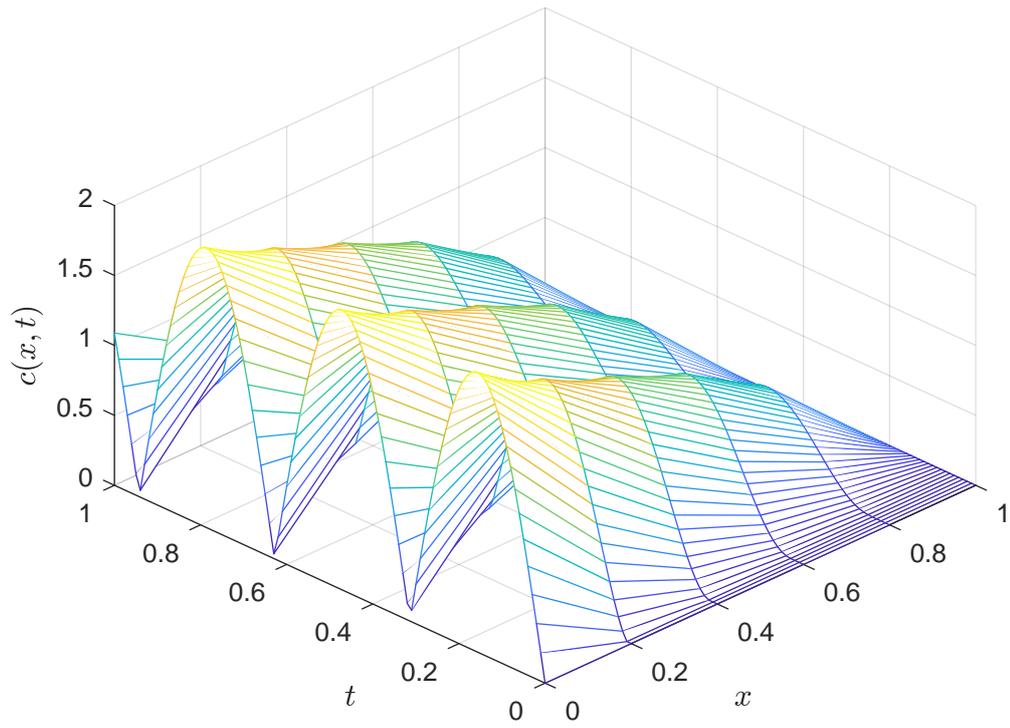


Рис. 3.3. Решение задачи

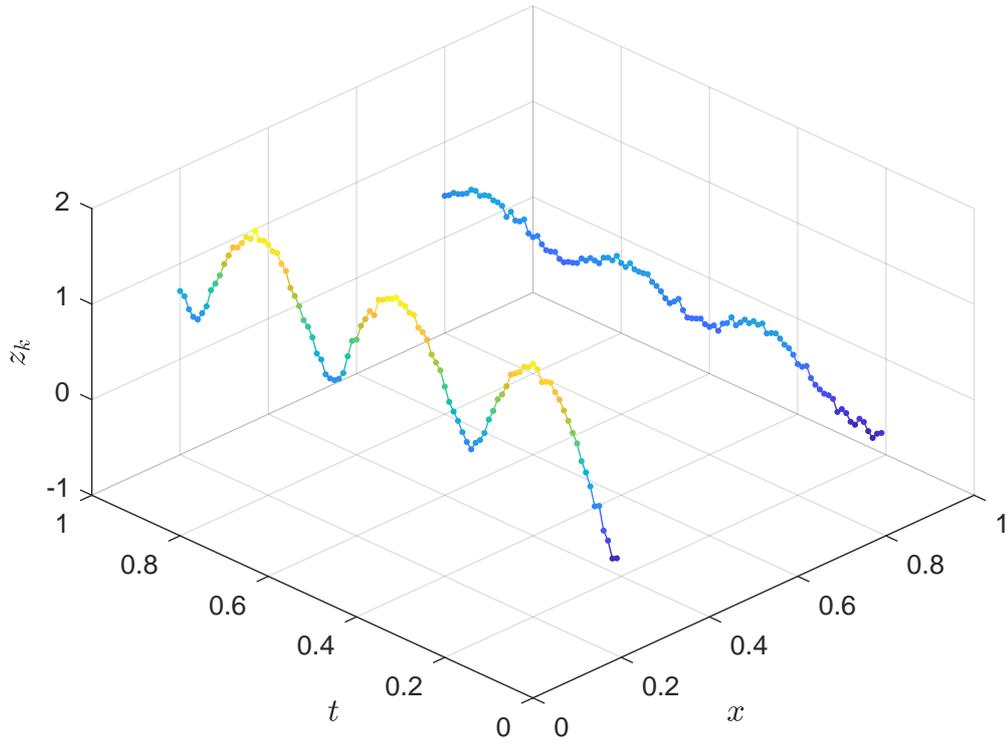
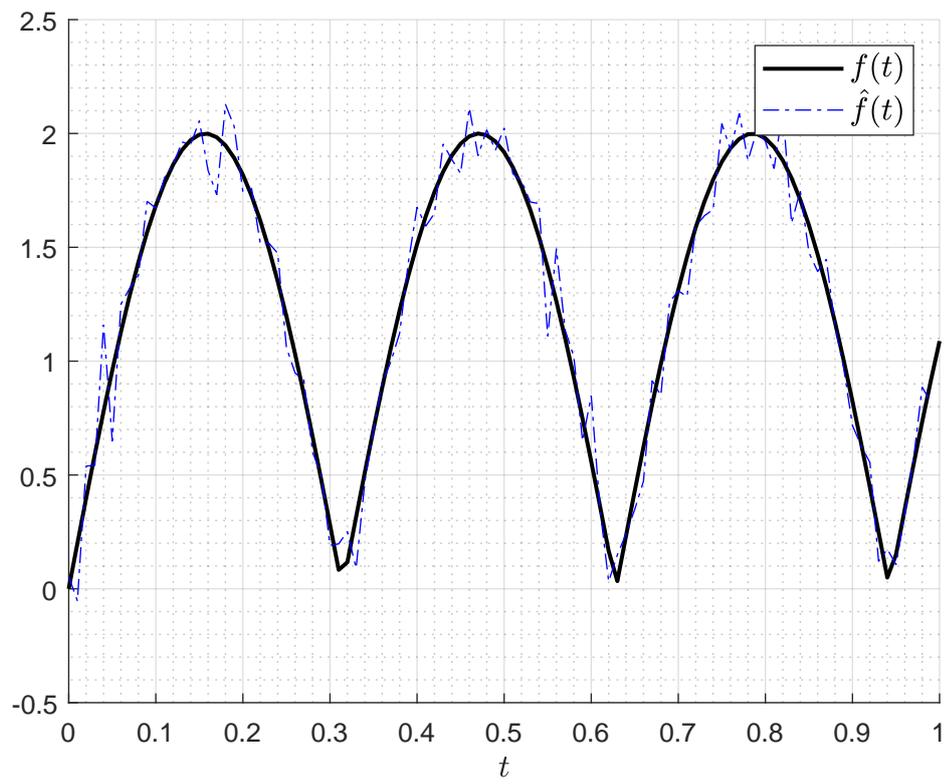
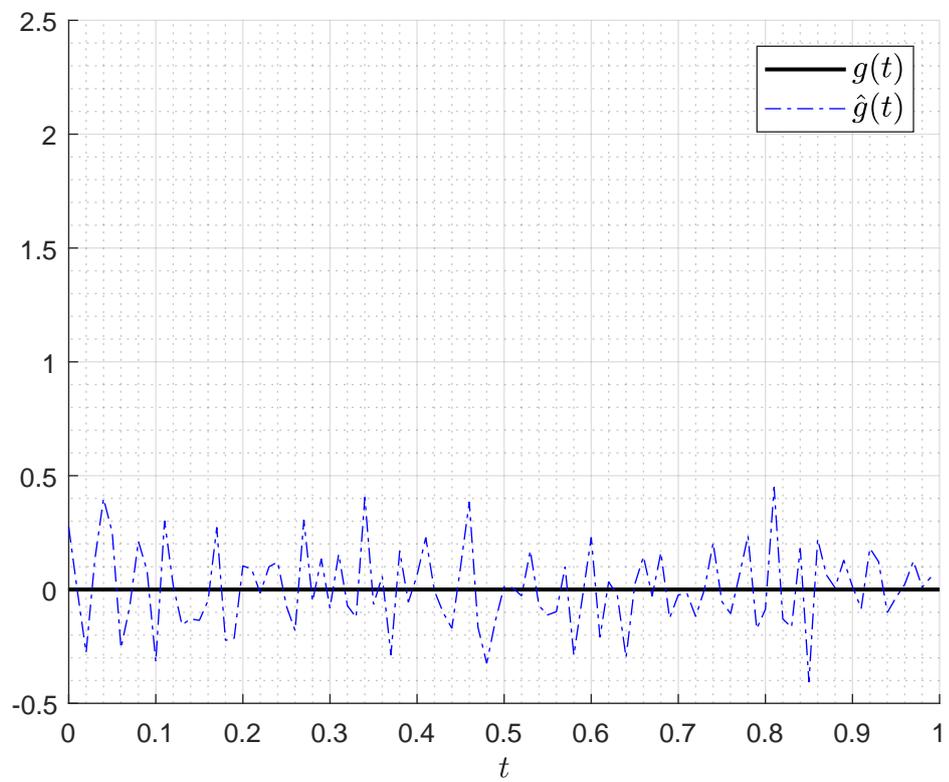


Рис. 3.4. Зашумленные измерения

Рис. 3.5. Оценка  $f(t)$ Рис. 3.6. Оценка  $g(t)$

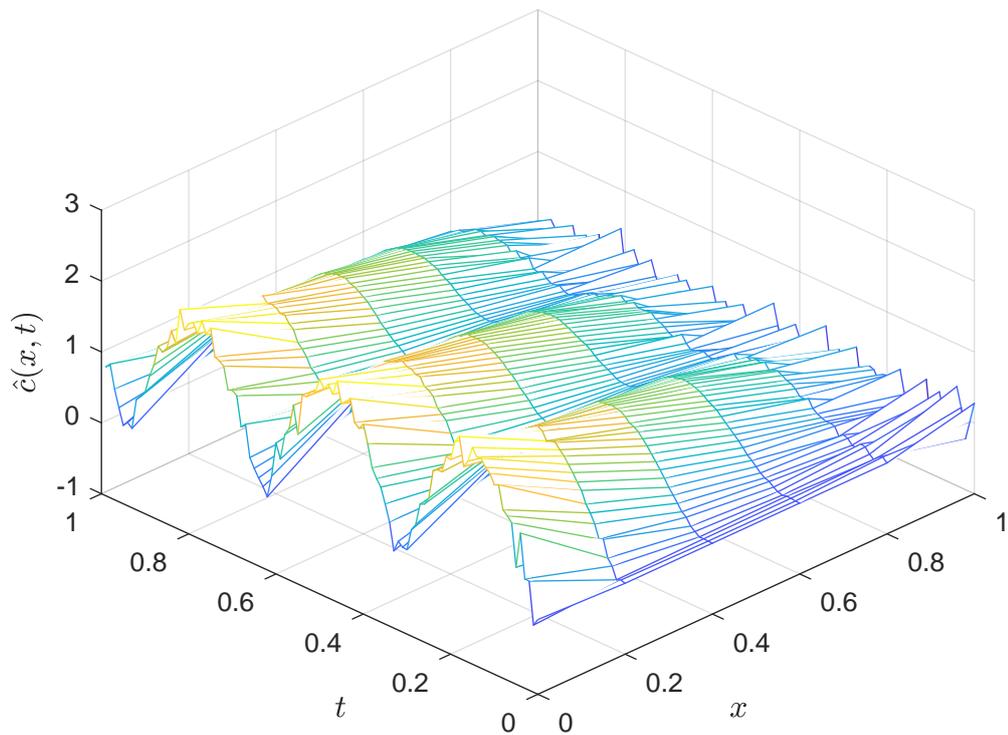


Рис. 3.7. Оценка решения

**Пример 3.2.**

$$\frac{\partial c}{\partial t} + 2 \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}, \quad (3.5)$$

$$c(x, 0) = 0, \quad (3.6)$$

$$c(0, t) = f(t), \quad \frac{\partial c(1, t)}{\partial x} = -[c(1, t) - g(t)], \quad (3.7)$$

$$x \in [0; 1], \quad t \in [0; 1],$$

где  $c(x, t)$  — концентрация вещества в одномерном потоке, текущем слева направо,

$$f(t) = 4 |2t - \lfloor 2t + 0.5 \rfloor|, \quad g(t) = \begin{cases} 0, & t \leq 0.25, \\ 2t - 0.5, & 0.25 < t < 0.75, \\ 1, & t \geq 0.75. \end{cases}$$

Условие (3.6) означает, что начальная концентрация вещества равна нулю на всем отрезке, первое граничное условие (3.7) соответствует периодическому изменению концентрации на левом конце отрезка по закону треугольной волны с

периодом 0.5, а второе граничное условие (3.7) — кусочно-линейному изменению концентрации в окружающей среде на правом конце.

Зададим пространственную сетку с шагом  $\Delta x = 0.2$ , состоящую из четырех внутренних ( $x_1 = 0.2, x_2 = 0.4, x_3 = 0.6, x_4 = 0.8$ ) и двух граничных ( $x_0 = 0, x_5 = 1$ ) узлов. Шаг  $\Delta t$  с учетом условия сходимости конечно-разностной схемы выберем равным  $\frac{\Delta x^2}{4\alpha} = 0.01$ .

Рассмотрим модель измерений с двумя сенсорами ( $m = 2$ ) в узлах  $x_1$  и  $x_5$ . Заметим, что при такой модели измерений условие (3.1) выполняется. Дискретная линейная стохастическая система в этом случае будет иметь следующий вид:

$$\begin{cases} \begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \\ c_5^k \end{bmatrix} = \begin{bmatrix} 0.5 & 0.2 & 0 & 0 & 0 \\ 0.3 & 0.5 & 0.2 & 0 & 0 \\ 0 & 0.3 & 0.5 & 0.2 & 0 \\ 0 & 0 & 0.3 & 0.5 & 0.2 \\ 0 & 0 & 0.25 & 0.4167 & 0.1667 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ c_3^{k-1} \\ c_4^{k-1} \\ c_5^{k-1} \end{bmatrix} + \begin{bmatrix} 0.3 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0.1667 \end{bmatrix} \begin{bmatrix} f^{k-1} \\ g^k \end{bmatrix}, \\ \begin{bmatrix} z_1^k \\ z_2^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1^k \\ c_2^k \\ c_3^k \\ c_4^k \\ c_5^k \end{bmatrix} + \begin{bmatrix} \xi_1^k \\ \xi_2^k \end{bmatrix}, \quad k = 1, 2, \dots, 100. \end{cases}$$

На рисунке 3.8 приведен график решения прямой задачи (3.5)–(3.7) методом конечных разностей. На рисунке 3.9 приведены графики зашумленных измерений, на рисунках 3.10 и 3.11 — результаты оценивания значений функций  $f(t)$  и  $g(t)$ , а на рисунке 3.12 — график оценки решения, полученные алгоритмом 3.2 с ковариацией шума  $R_k = \text{diag}(0.02^2, 0.02^2)$ .

Заметим, что для рассматриваемой нами модели на каждой итерации алгоритма в строке 8 вычисляется оценка вектора входных воздействий, состоящего из значения функции  $f(t)$  в предыдущий момент времени и значения функции  $g(t)$  в текущий момент времени, а в строке 15 получается оценка вектора состояния для текущего момента времени. Таким образом, оценка значений функции

$f(t)$  производится с запаздыванием на один шаг и на последней итерации алгоритма ее значение оказывается неопределенным.

Таблица 3.2. Ошибки оценивания

	RMSE <sub>0</sub>	RMSE <sub>1</sub>	RMSE <sub>2</sub>	RMSE <sub>3</sub>	RMSE <sub>4</sub>	RMSE <sub>5</sub>	nRMSE <sub>c</sub>	RMSE <sub>g</sub>
$R_1$	0.8187	0.2165	0.0826	0.0481	0.0557	0.2178	0.8814	1.3284
$R_2$	0.0819	0.0217	0.0083	0.0048	0.0056	0.0218	0.0881	0.1328
$R_3$	0.0082	0.0022	0.0008	0.0005	0.0006	0.0022	0.0088	0.0133

В таблице 3.2 приведены среднеквадратические ошибки оценивания значений функции  $c(x, t)$  в узлах сетки (RMSE<sub>*i*</sub>) и нормализованная среднеквадратическая ошибка (nRMSE<sub>*c*</sub>), а также среднеквадратические ошибки оценивания функции  $g(t)$  (RMSE<sub>*g*</sub>) по результатам 100 экспериментов при различных ковариациях шума:  $R_1 = \text{diag}(0.2^2, 0.2^2)$ ,  $R_2 = \text{diag}(0.02^2, 0.02^2)$ ,  $R_3 = \text{diag}(0.002^2, 0.002^2)$ .

Из приведенной таблицы видно, что с уменьшением шума качество идентификации граничных условий возрастает.

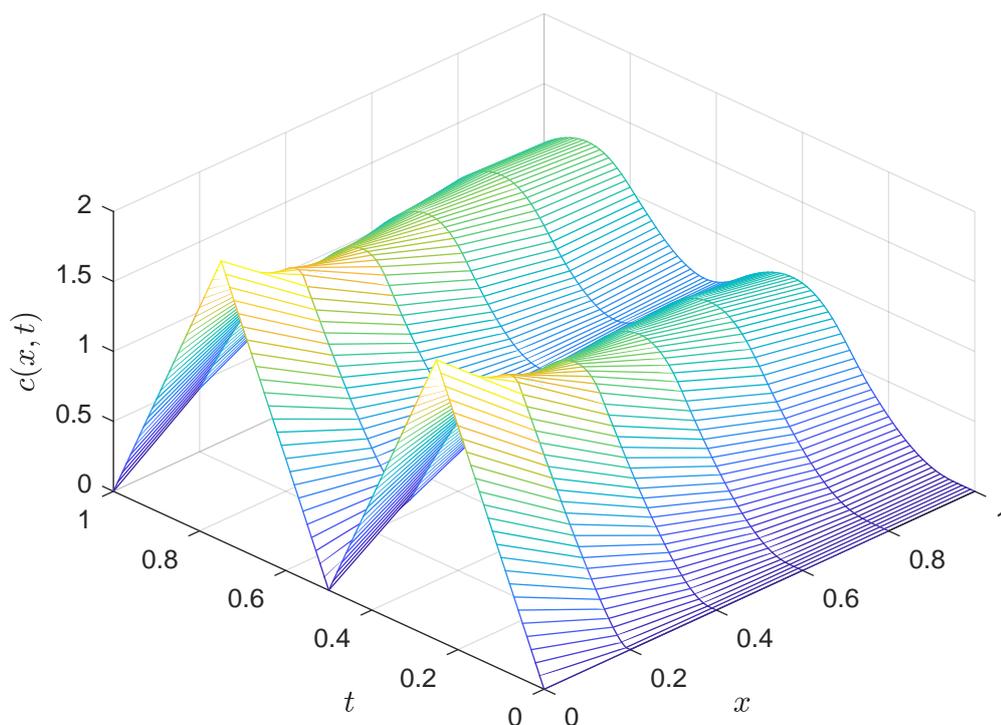


Рис. 3.8. Решение задачи

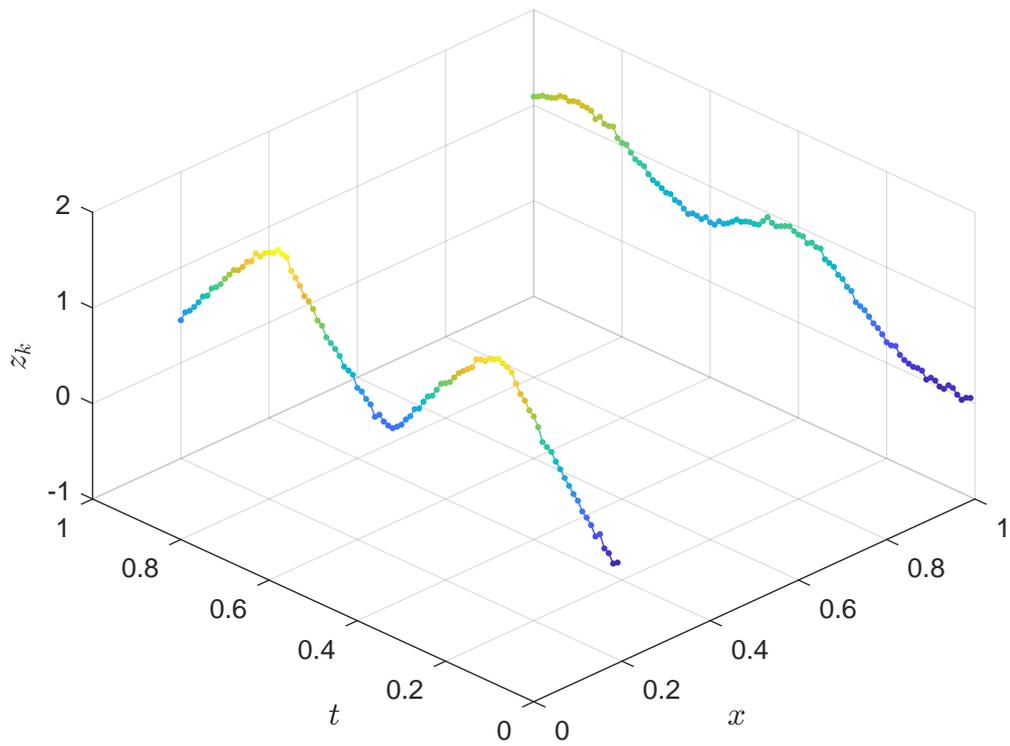
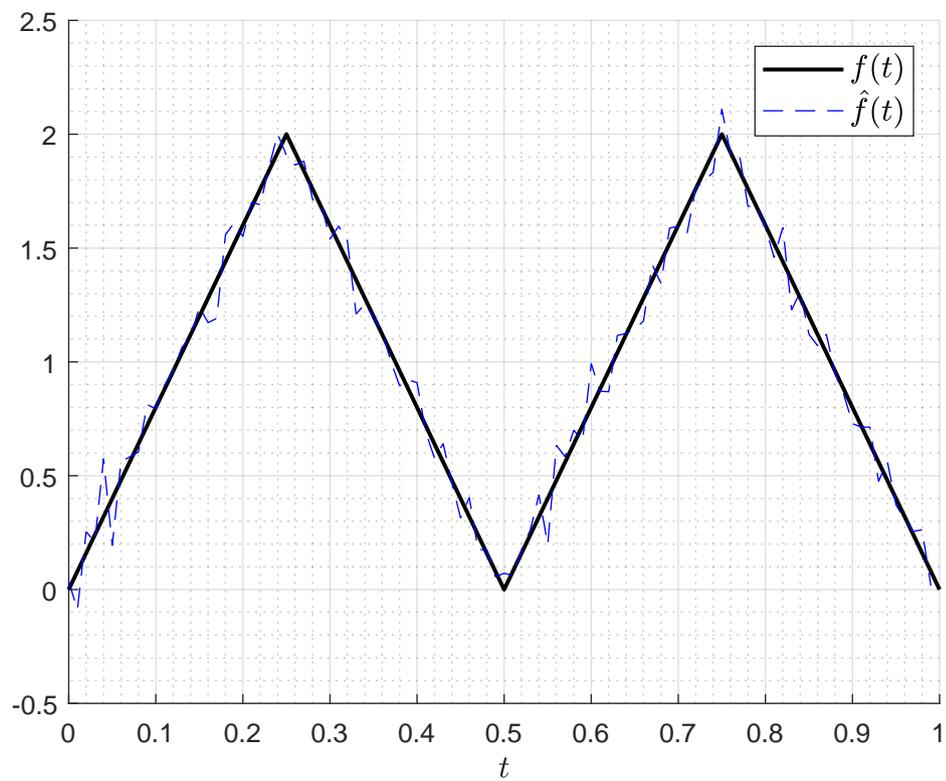


Рис. 3.9. Зашумленные измерения

Рис. 3.10. Оценка  $f(t)$

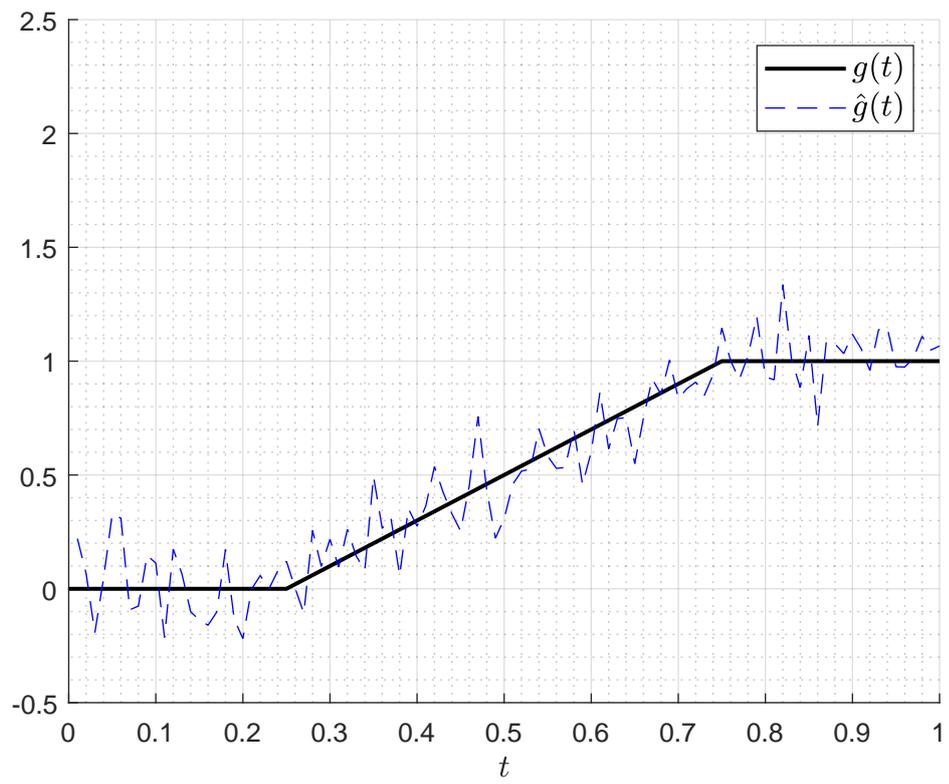
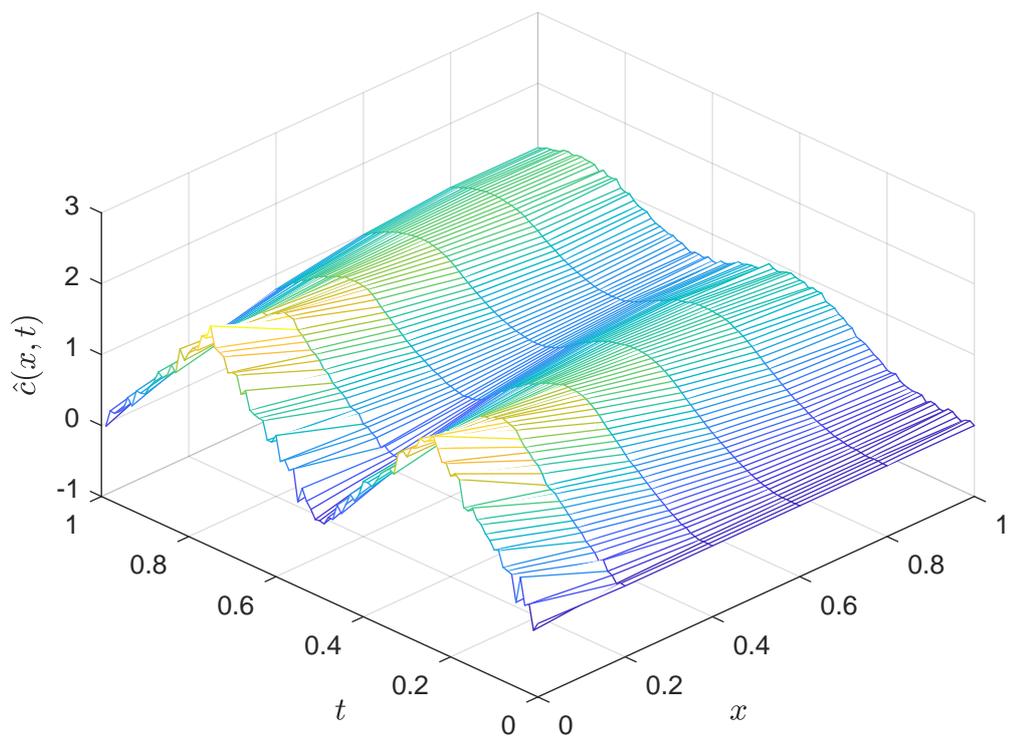
Рис. 3.11. Оценка  $g(t)$ 

Рис. 3.12. Оценка решения

### Случай известного граничного условия

Для случая, когда одно из условий известно, этап прогноза оценки вектора состояния в алгоритмах 3.1 и 3.2 необходимо модифицировать следующим образом:

$$\hat{c}_{k|k-1} = F_{k-1}\hat{c}_{k-1|k-1} + B_1 u_{k-1}^1. \quad (3.8)$$

#### Пример 3.3.

$$\frac{\partial c}{\partial t} + 2 \frac{\partial c}{\partial x} = \frac{1}{2} \frac{\partial^2 c}{\partial x^2}, \quad (3.9)$$

$$c(x, 0) = 10x(1 - x), \quad (3.10)$$

$$c(0, t) = t, \quad c(1, t) = 0, \quad (3.11)$$

$$x \in [0; 1], \quad t \in [0; 1],$$

где  $c(x, t)$  — концентрация вещества в одномерном потоке, текущем слева направо, первое граничное условие (3.11) соответствует линейному увеличению концентрации вещества на левом конце отрезка, второе — поглощающей стенке на правом конце.

Пусть граничное условие на правом конце отрезка известно. Зададим на отрезке  $[0; 1]$  пространственную сетку с 9 узлами ( $N = 8$ ), что соответствует  $\Delta x = 0.125$ . Вектор состояния дискретной модели в данном случае будет состоять из  $n = 7$  узлов. Шаг  $\Delta t$  выберем с учетом условия устойчивости конечно-разностной схемы:  $\Delta t = 0.01 \leq \frac{\Delta x^2}{2\alpha}$  ( $K = 100$ ).

Дискретная линейная система (1.12) в этом случае будет иметь вид:

$$\begin{bmatrix} c_1^k \\ c_2^k \\ \vdots \\ c_6^k \\ c_7^k \end{bmatrix} = \begin{bmatrix} 0.36 & 0.24 & \dots & 0 & 0 \\ 0.4 & 0.36 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0.36 & 0.24 \\ 0 & 0 & \dots & 0.4 & 0.36 \end{bmatrix} \begin{bmatrix} c_1^{k-1} \\ c_2^{k-1} \\ \vdots \\ c_6^{k-1} \\ c_7^{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0.24 \end{bmatrix} g^{k-1} + \begin{bmatrix} 0.4 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} f^{k-1},$$

$$k = 1, 2, \dots, K,$$

где  $f^{k-1} = (k-1)\Delta t$ ,  $g^{k-1} = 0$ .

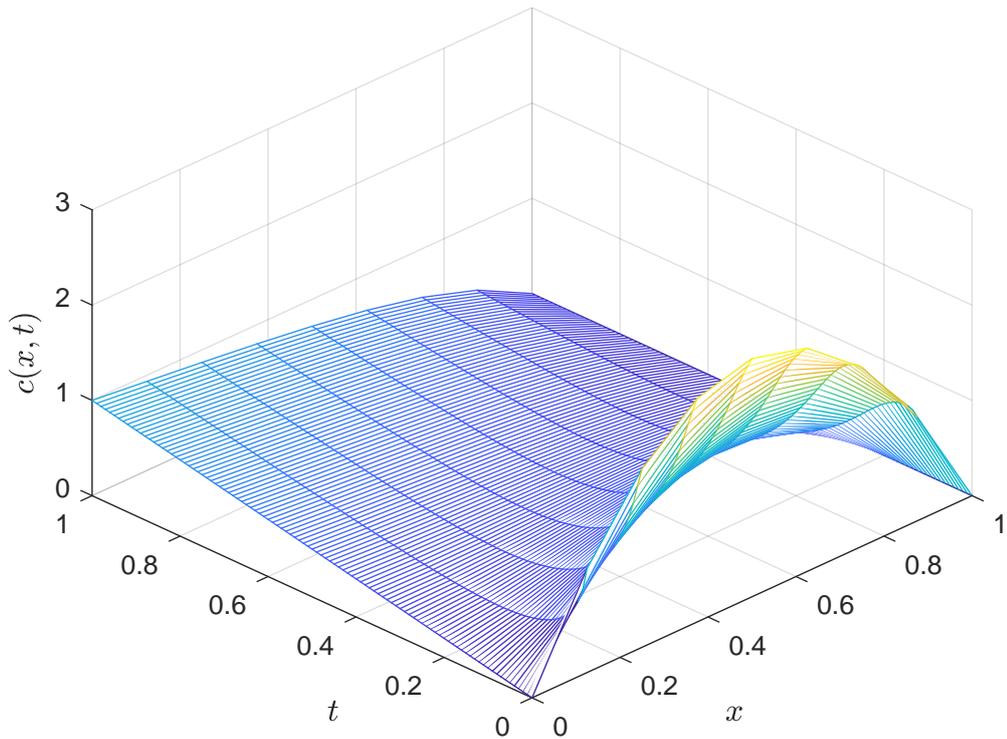


Рис. 3.13. Решение задачи

На рисунке 3.13 приведен график решения прямой задачи, полученного методом конечных разностей на рассматриваемой сетке.

Измерения значений функции  $c(x, t)$  будем производить в первом внутреннем узле пространственной сетки ( $x = 0.125$ ). Для этого зададим матрицу измерений в виде

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

Уравнение зашумленных измерений будет иметь вид

$$z_1^k = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} c_1^k \\ c_2^k \\ \vdots \\ c_6^k \\ c_7^k \end{bmatrix} + \xi_1^k, \quad k = 1, 2, \dots, K.$$

На рисунке 3.14 приведен график смоделированных зашумленных измерений с дисперсией шума  $R_k = 0.001$ , на рисунках 3.15 и 3.16 показаны результаты оценивания левого граничного условия и решения модифицированным алгорит-

мом 3.2, а на рисунке 3.17 — график дисперсии ошибки оценивания граничного условия.

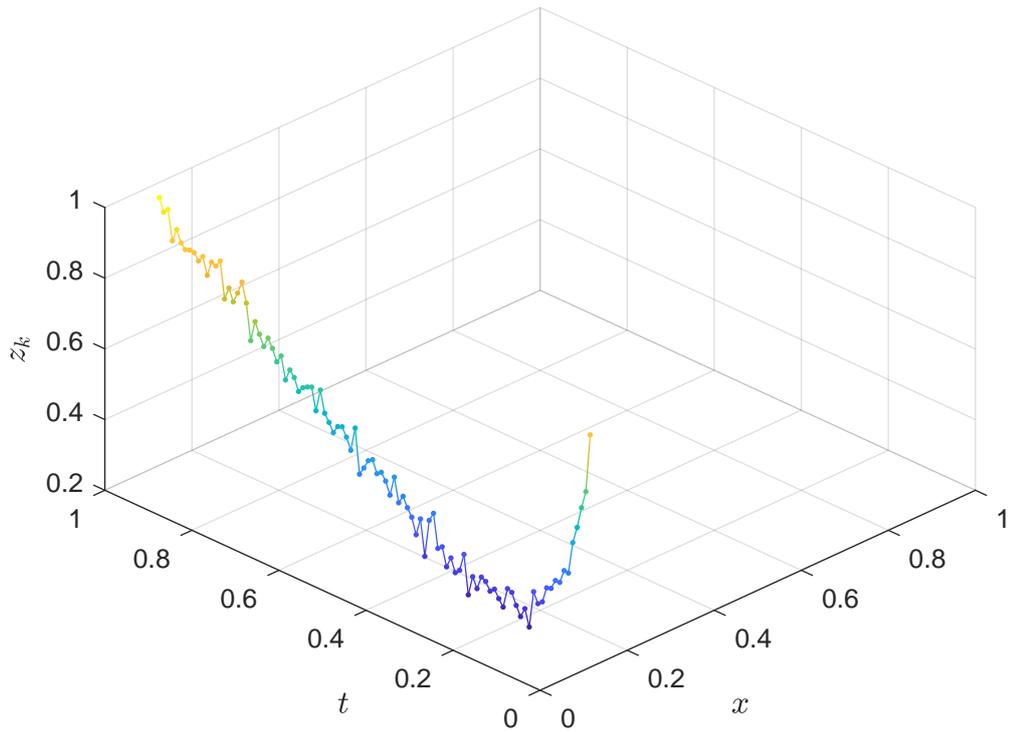


Рис. 3.14. Зашумленные измерения

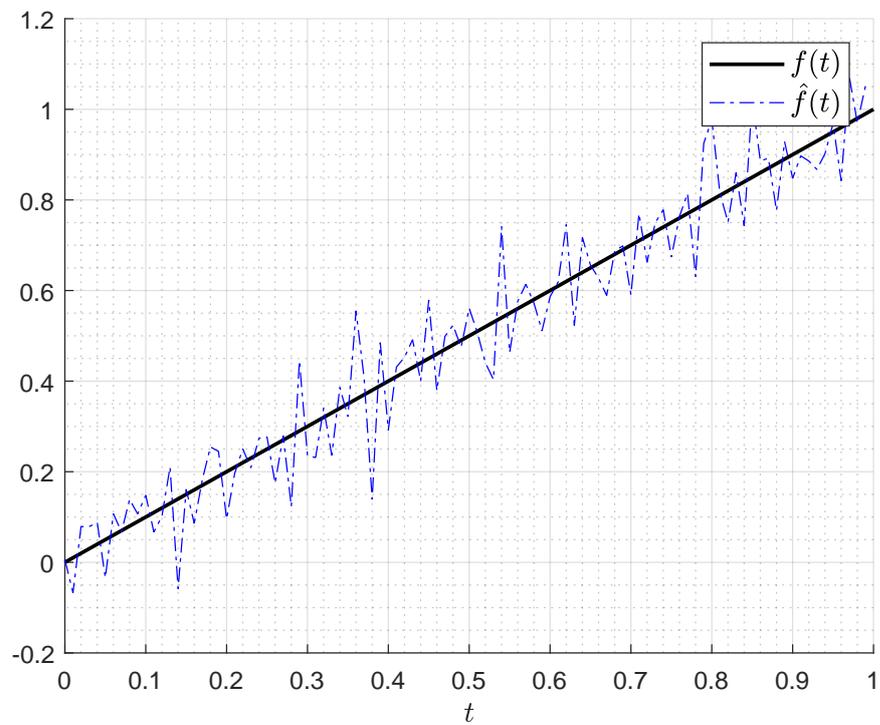


Рис. 3.15. Оценка  $f(t)$

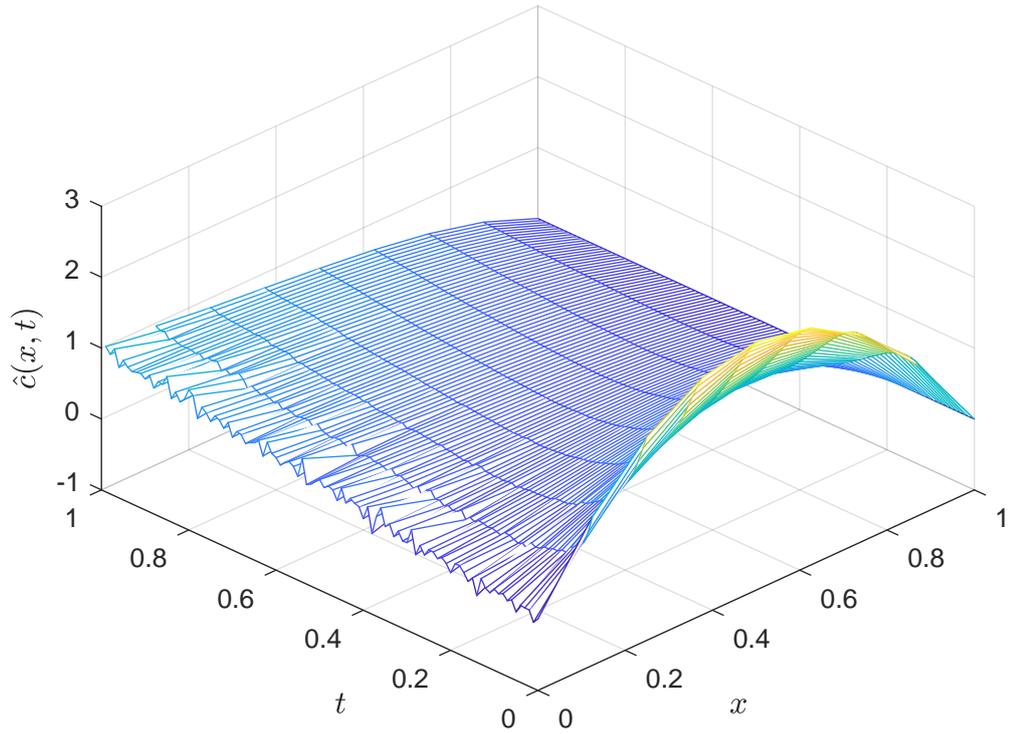


Рис. 3.16. Оценка решения

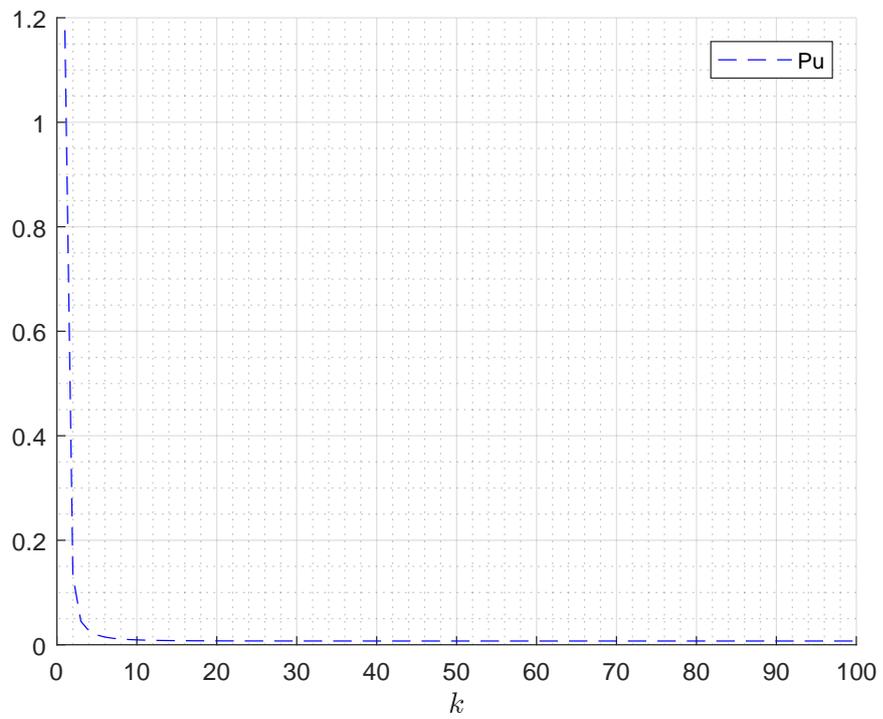


Рис. 3.17. Дисперсия ошибки оценивания граничного условия

### 3.3. Идентификация на основе квадратно-корневой модификации алгоритма одновременного оценивания состояния и входных воздействий

Основная идея квадратно-корневых модификаций алгоритмов дискретной фильтрации типа заключается в представлении положительно полуопределенных матриц, в частности, ковариационной матрицы ошибок оценивания, в виде  $P_k = S_k S_k^T$ , где  $S_k$  — “квадратный корень” матрицы  $P_k$ , являющийся нижней треугольной матрицей [67]. Такое представление может быть получено, например, с помощью разложения Холецкого [14]. На каждом этапе квадратно-корневого алгоритма обновление основных величин фильтра выполняется при помощи ортогональных преобразований, применяемых к матричным массивам, содержащим все нужные для расчетов величины, а результаты вычислений также получаются в матричных массивах.

К преимуществам квадратно-корневых алгоритмов фильтрации традиционно относят [86]:

- 1) сохранение симметричности и положительной определенности ковариационных матриц ошибок оценивания векторов состояния и входных воздействий;
- 2) возможность экономии памяти за счет работы с треугольными факторами;
- 3) удобная форма для реализации параллельной схемы вычислений.

В работе [87] получена информационная форма алгоритма 3.1 и на ее основе построен квадратно-корневой алгоритм одновременного оценивания состояния и входных воздействий, причем полученный алгоритм сформулирован в так называемой “расширенной” форме, то есть оценки для векторов состояния и входных воздействий получаются в матричных массивах одновременно с ковариационными матрицами ошибок оценивания. Кроме того, в указанной работе показано, что построение аналогичного расширенного квадратно-корневого алгоритма на основе ковариационной формы алгоритма 3.1 невозможно.

Тем не менее, можно построить нерасширенную квадратно-корневую вер-

сию алгоритма 3.1, в которой оценки векторов состояния и входных воздействий, а также некоторые вспомогательные вычисления, выполняются отдельно. Данный алгоритм впервые сформулирован в работе [85].

---

**Алгоритм 3.3.** Квадратно-корневой ковариационный алгоритм оценивания состояния и входных воздействий

---

**Вход:**  $\bar{c}_0, \Pi_0$

```

1   $\hat{c}_0 = \bar{c}_0, S_{P_0} = \text{chol}(\Pi_0)$  // Инициализация
2  for  $k = 1, 2, \dots, K$  do
    // Прогноз оценки вектора состояния
3   $\hat{c}_{k|k-1} = F_{k-1}\hat{c}_{k-1}$ 
4   $S_{Q_{k-1}} = \text{chol}(Q_{k-1})$ 
5   $\begin{bmatrix} S_{P_{k|k-1}}^T \\ 0 \end{bmatrix} = \mathcal{Q}_1 \begin{bmatrix} S_{P_{k-1}}^T F_{k-1}^T \\ S_{Q_{k-1}}^T \end{bmatrix}$ 
    // Оценка вектора входных воздействий
6   $S_{R_k} = \text{chol}(R_k)$ 
7   $\begin{bmatrix} S_{\tilde{R}_k}^T & \bar{K}_k^T \\ 0 & S_{P_k^*}^T \end{bmatrix} = \mathcal{Q}_2 \begin{bmatrix} S_{R_k}^T & 0 \\ S_{P_{k|k-1}}^T H_k^T & S_{P_{k|k-1}}^T \end{bmatrix}$ 
8   $\begin{bmatrix} S_{D_{k-1}}^{-T} \end{bmatrix} = \mathcal{Q}_3 \begin{bmatrix} S_{\tilde{R}_k}^{-T} H_k B_{k-1} \end{bmatrix}$ 
9   $M_k = S_{D_{k-1}} S_{D_{k-1}}^T B_{k-1}^T H_k^T S_{\tilde{R}_k}^{-T} S_{\tilde{R}_k}^{-1}$ 
10  $\hat{u}_{k-1} = M_k(z_k - H_k \hat{c}_{k|k-1})$ 
    // Коррекция оценки вектора состояния
11  $K_k = \bar{K}_k S_{\tilde{R}_k}^{-1}$ 
12  $\hat{c}_k^* = \hat{c}_{k|k-1} + B_{k-1} \hat{u}_{k-1}$ 
13  $\hat{c}_k = \hat{c}_k^* + K_k(z_k - H_k \hat{c}_k^*)$ 
14  $\begin{bmatrix} S_{P_k}^T \\ 0 \end{bmatrix} = \mathcal{Q}_4 \begin{bmatrix} S_{P_k^*}^T \\ S_{D_{k-1}}^T B_k^T (I_n - K_k H_k)^T \end{bmatrix}$ 
15 end for
```

**Выход:**  $\hat{c}_k, S_{P_k}, \hat{u}_{k-1}, S_{D_{k-1}}, k = 1, 2, \dots, K$

---

На этапе инициализации и в основном цикле алгоритма разложение матриц  $\Pi_0, Q_{k-1}$  и  $R_k$  осуществляется с помощью алгоритма Холецкого.

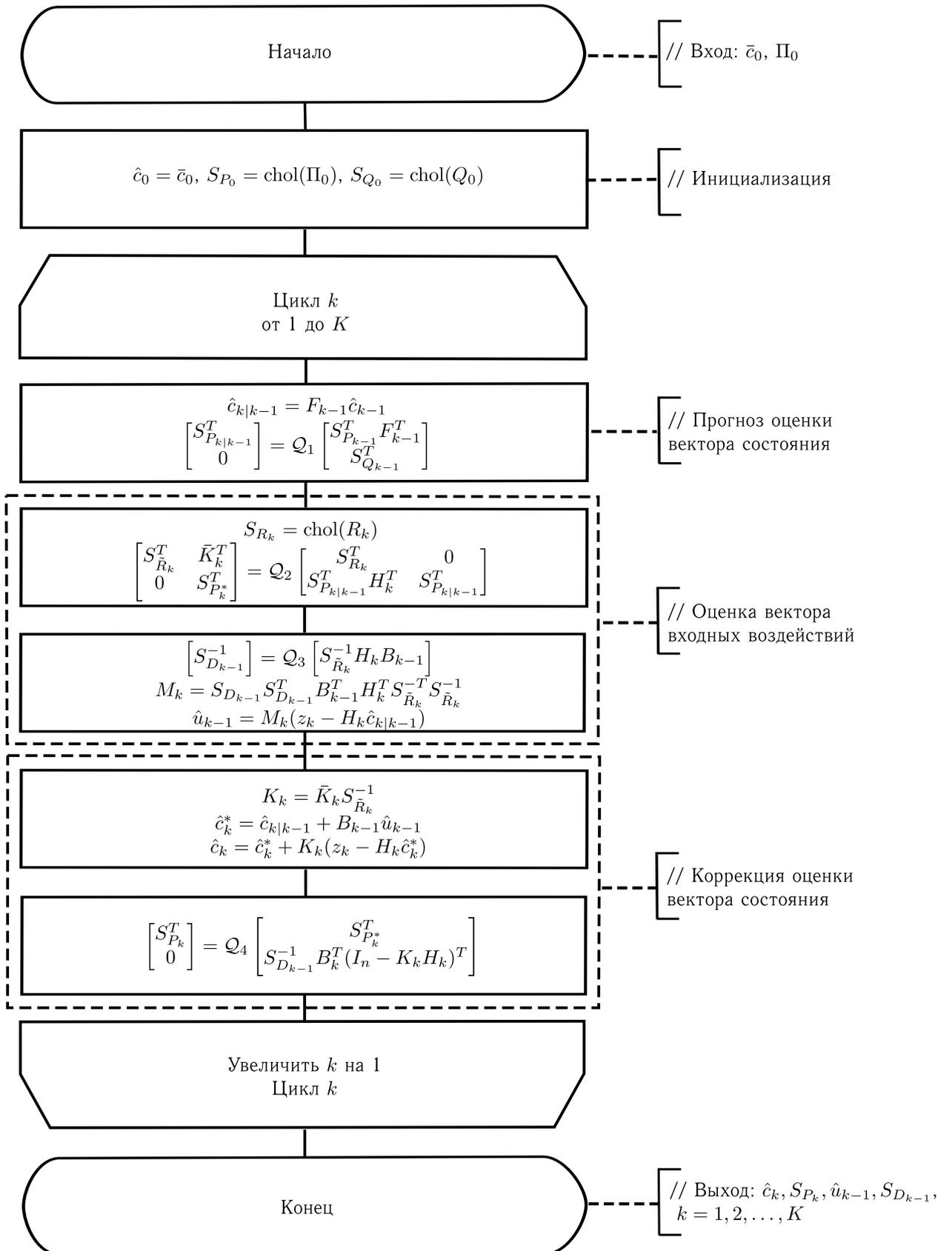


Рис. 3.18. Блок-схема алгоритма 3.3

**Теорема 3.1.** Алгоритм 3.3 алгебраически эквивалентен алгоритму 3.1.

*Доказательство.* Покажем, что в алгоритме 3.3 обновление величин, выполняемое при помощи ортогональных преобразований  $\mathcal{Q}_1$ ,  $\mathcal{Q}_2$ ,  $\mathcal{Q}_3$  и  $\mathcal{Q}_4$ , производится по формулам, которые эквивалентны соответствующим формулам в алгоритме 3.1.

Рассмотрим матричное ортогональное преобразование  $\mathcal{Q}A = R$ . Доказательство основано на проверке матричного равенства

$$A^T \mathcal{Q}^T \mathcal{Q}A = A^T I A = A^T A = R^T R,$$

где  $I$  — единичная матрица.

$$\text{Пусть } \mathcal{Q} = \mathcal{Q}_1, A = \begin{bmatrix} F_{k-1} S_{P_{k-1}} & S_{Q_{k-1}} \end{bmatrix}^T, R = \begin{bmatrix} S_{P_{k-1}} & 0 \end{bmatrix}^T.$$

Вычислим

$$\begin{aligned} A^T A &= \begin{bmatrix} F_{k-1} S_{P_{k-1}} & S_{Q_{k-1}} \end{bmatrix} \begin{bmatrix} S_{P_{k-1}}^T F_{k-1}^T \\ S_{Q_{k-1}}^T \end{bmatrix} = F_{k-1} S_{P_{k-1}} S_{P_{k-1}}^T F_{k-1}^T + S_{Q_{k-1}} S_{Q_{k-1}}^T \\ &= F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1}, \end{aligned}$$

$$R^T R = \begin{bmatrix} S_{P_{k|k-1}} & 0 \end{bmatrix} \begin{bmatrix} S_{P_{k|k-1}}^T \\ 0 \end{bmatrix} = S_{P_{k|k-1}} S_{P_{k|k-1}}^T = P_{k|k-1}.$$

Получим

$$P_{k|k-1} = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1},$$

что соответствует строке 4 алгоритма 3.1.

$$\text{Пусть } \mathcal{Q} = \mathcal{Q}_2, A = \begin{bmatrix} S_{R_k}^T & 0 \\ S_{P_{k|k-1}}^T & H_k^T \\ & S_{P_{k|k-1}}^T \end{bmatrix}, R = \begin{bmatrix} S_{\tilde{R}_k}^T & \bar{K}_k^T \\ 0 & S_{P_k^*}^T \end{bmatrix}.$$

Вычислим

$$A^T A = \begin{bmatrix} S_{R_k} & H_k S_{P_{k|k-1}} \\ 0 & S_{P_{k|k-1}} \end{bmatrix} \begin{bmatrix} S_{R_k}^T & 0 \\ S_{P_{k|k-1}}^T & H_k^T \\ & S_{P_{k|k-1}}^T \end{bmatrix} = \begin{bmatrix} R_k + H_k P_{k|k-1} H_k^T & H_k P_{k|k-1} \\ P_{k|k-1} H_k^T & P_{k|k-1} \end{bmatrix},$$

$$R^T R = \begin{bmatrix} S_{\tilde{R}_k} & 0 \\ \bar{K}_k & S_{P_k^*} \end{bmatrix} \begin{bmatrix} S_{\tilde{R}_k}^T & \bar{K}_k^T \\ 0 & S_{P_k^*}^T \end{bmatrix} = \begin{bmatrix} \tilde{R}_k & S_{\tilde{R}_k} \bar{K}_k^T \\ \bar{K}_k S_{\tilde{R}_k}^T & \bar{K}_k \bar{K}_k^T + P_k^* \end{bmatrix}.$$

Получим

$$\tilde{R}_k = R_k + H_k P_{k|k-1} H_k^T, \quad (3.12)$$

$$\begin{aligned} \bar{K}_k S_{\tilde{R}_k}^T &= P_{k|k-1} H_k^T \Rightarrow \bar{K}_k = P_{k|k-1} H_k^T S_{\tilde{R}_k}^{-T} \\ &= P_{k|k-1} H_k^T \tilde{R}_k^{-1} \tilde{R}_k S_{\tilde{R}_k}^{-T} = \underbrace{P_{k|k-1} H_k^T \tilde{R}_k^{-1}}_{K_k} S_{\tilde{R}_k}^T \underbrace{S_{\tilde{R}_k}^{-T}}_I = K_k S_{\tilde{R}_k}^T, \end{aligned} \quad (3.13)$$

$$\begin{aligned} P_{k|k-1} &= \bar{K}_k \bar{K}_k^T + P_k^* \Rightarrow P_k^* = P_{k|k-1} - \bar{K}_k \bar{K}_k^T \\ &= P_{k|k-1} - K_k S_{\tilde{R}_k}^T S_{\tilde{R}_k}^T K_k^T = P_{k|k-1} - K_k \tilde{R}_k K_k^T = (I - K_k H_k) P_{k|k-1}. \end{aligned} \quad (3.14)$$

Формула (3.12) соответствует строке 5, а (3.14) — строке 11 алгоритма 3.1.

Из (3.13) следует, что коэффициент  $K_k$  может быть вычислен по формуле

$$K_k = \bar{K}_k S_{\tilde{R}_k}^{-1}.$$

$$\text{Пусть } \mathcal{Q} = \mathcal{Q}_3, A = \begin{bmatrix} S_{\tilde{R}_k}^{-T} H_k B_{k-1} \end{bmatrix}, R = \begin{bmatrix} S_{D_{k-1}}^{-T} \end{bmatrix}.$$

Вычислим

$$A^T A = B_{k-1}^T H_k^T S_{\tilde{R}_k}^{-1} S_{\tilde{R}_k}^{-T} H_k B_{k-1}, \quad R^T R = S_{D_{k-1}}^{-1} S_{D_{k-1}}^{-T}.$$

Получим

$$\underbrace{S_{D_{k-1}}^{-1} S_{D_{k-1}}^{-T}}_{D_{k-1}^{-T}} = B_{k-1}^T H_k^T S_{\tilde{R}_k}^{-1} S_{\tilde{R}_k}^{-T} H_k B_{k-1} \Rightarrow$$

$$D_{k-1} = (B_{k-1}^T H_k^T \underbrace{S_{\tilde{R}_k}^{-T} S_{\tilde{R}_k}^{-1}}_{\tilde{R}_k^{-1} = \tilde{R}_k^{-T}} H_k B_{k-1})^{-1},$$

что соответствует строке 6 алгоритма 3.1.

С учетом равенств  $D_{k-1} = S_{D_{k-1}} S_{D_{k-1}}^T$ ,  $\tilde{R}_k^{-1} = S_{\tilde{R}_k}^{-T} S_{\tilde{R}_k}^{-1}$  получаем, что коэффициент  $M_k$  может быть вычислен по формуле

$$M_k = S_{D_{k-1}} S_{D_{k-1}}^T B_{k-1}^T H_k^T S_{\tilde{R}_k}^{-T} S_{\tilde{R}_k}^{-1}.$$

Для ортогонального преобразования  $\mathcal{Q}_4$ , рассуждая аналогично, получаем формулу

$$P_k = P_k^* + (I_n - K_k H_k) B_{k-1} D_{k-1} B_{k-1}^T (I_n - K_k H_k)^T,$$

соответствующую строке 13 алгоритма 3.1.

Все остальные величины:  $\hat{c}_{k|k-1}$ ,  $\hat{u}_{k-1}$ ,  $\hat{c}_k^*$ ,  $\hat{c}_k$  вычисляются в обоих алгоритмах по одинаковым формулам.  $\square$

Продемонстрируем эквивалентность алгоритмов 3.1 и 3.3 на численном примере. Снова рассмотрим пример 3.3. В таблице 3.3 приведены среднеквадратические ошибки оценивания в узлах пространственной сетки по результатам 100 экспериментов при различных дисперсиях шума  $R_1 = 0.1^2$ ,  $R_2 = 0.01^2$  и  $R_3 = 0.001^2$  (узел с номером 8 соответствует известному правому граничному условию). Из приведенных результатов видно, что для рассмотренного примера результаты работы двух алгоритмов одинаковы.

Таблица 3.3. Ошибки оценивания

	Алгоритм 3.1			Алгоритм 3.3		
	$R_1$	$R_2$	$R_3$	$R_1$	$R_2$	$R_3$
RMSE <sub>0</sub>	0.2620	0.0262	0.0027	0.2620	0.0262	0.0027
RMSE <sub>1</sub>	0.0984	0.0098	0.0010	0.0984	0.0098	0.0010
RMSE <sub>2</sub>	0.0438	0.0044	0.0004	0.0438	0.0044	0.0004
RMSE <sub>3</sub>	0.0252	0.0025	0.0003	0.0252	0.0025	0.0003
RMSE <sub>4</sub>	0.0178	0.0018	0.0002	0.0178	0.0018	0.0002
RMSE <sub>5</sub>	0.0140	0.0014	0.0001	0.0140	0.0014	0.0001
RMSE <sub>6</sub>	0.0107	0.0011	0.0001	0.0107	0.0011	0.0001
RMSE <sub>7</sub>	0.0066	0.0007	0.0001	0.0066	0.0007	0.0001
RMSE <sub>8</sub>	0	0	0	0	0	0
nRMSE	0.2856	0.0286	0.0029	0.2856	0.0286	0.0029

### 3.4. Заключение и выводы к главе 3

В данной главе была рассмотрена задача идентификации неизвестных граничных условий моделей конвективно-диффузионного переноса. Предлагаемый подход заключается в переходе от исходных непрерывных моделей (1.1), (1.2), (1.3) и (1.1), (1.2), (1.4) к моделям, описываемым дискретными линейными стохастическими системами вида (1.5), (1.6) или (1.7), (1.6), в которых функции  $f(t)$  и  $g(t)$  входят в неизвестный вектор входных воздействий.

Для численной идентификации граничных условий предложено использовать алгоритмы Гиллейнса–Де-Мора 3.1 и 3.2 [79], предназначенные для одновременного оценивания векторов состояния и входных воздействий. В разделе 3.2 приведены примеры применения данного алгоритма для случаев, когда оба граничных условия неизвестны и когда одно из граничных условий известно. Результаты численных экспериментов подтверждают работоспособность и эффективность предложенных подходов.

В разделе 3.3 сформулирована квадратно-корневая модификация алгоритма Гиллейнса–Де-Мора в ковариационной форме (алгоритм 3.3) и доказана ее алгебраическая эквивалентность исходному алгоритму (теорема 3.1), что подтверждается численными экспериментами. К преимуществам разработанного алгоритма можно отнести:

- 1) сохранение симметричности и положительной определенности ковариационных матриц ошибок оценивания векторов состояния и входных воздействий;
- 2) возможность экономии памяти за счет работы с треугольными факторами;
- 3) удобная форма для реализации параллельной схемы вычислений.

## Глава 4

**Программный комплекс для моделирования  
процессов параметрической идентификации  
математических моделей  
конвективно-диффузионного переноса**

### 4.1. Описание программного комплекса

Для практической проверки теоретических результатов, полученных в рамках диссертационного исследования, был разработан комплекс программ для компьютерного моделирования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса [88, 77, 89, 90].

Программный комплекс реализован на языке MATLAB и представляет собой набор компонентов, каждый из которых решает свой набор задач. На рисунке 4.1 приведена структурная схема программного комплекса. Перечень компонентов приведен в таблице 4.1.

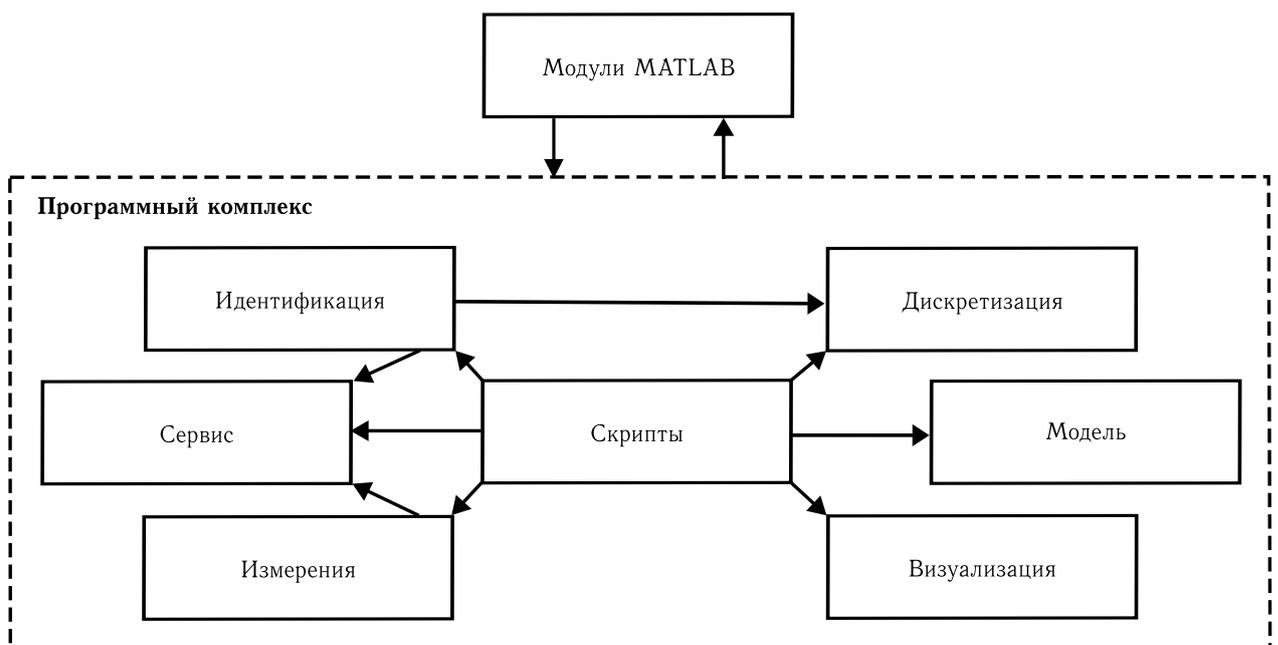


Рис. 4.1. Структура программного комплекса

Таблица 4.1. Компоненты программного комплекса

Компонент	Описание
Модель	<p>Назначение: описание исходной модели</p> <p>Список m-файлов: bc_f1, bc_f2, ic_f, model</p>
Дискретизация	<p>Назначение: построение конечно-разностной сетки, дискретизация исходной модели</p> <p>Список m-файлов: compute_grid, findiff_coef, findiff_dcoef, pde2lds, pde2lds, solve_findiff</p>
Измерения	<p>Назначение: моделирование зашумленных измерений</p> <p>Список m-файлов: measure, sensor</p>
Идентификация	<p>Назначение: идентификация коэффициентов конвекции и диффузии, идентификация граничных условий</p> <p>Список m-файлов: bc_id_gm_1, bc_id_gm_2, bc_id_gm_srcf, coef_id_fmincon, coef_id_fmincon_grad, coef_id_ga, coef_id_sa, kf_lr, kf_lr_grad, kf_lr_s, svd_tm_lr, svd_tm_lr_s</p>
Визуализация	<p>Назначение: визуализация и вывод результатов</p> <p>Список m-файлов: plot_conditions, plot_estimates, plot_measurements, plot_p, plot_solution, print_grid, print_lds, print_sensor</p>
Сервис	<p>Назначение: вспомогательные функции</p> <p>Список m-файлов: sr, ga_out, sa_out</p>
Скрипты	<p>Назначение: проведение численных экспериментов</p> <p>Список m-файлов: run_bc_id_gm, run_coef_id_fmincon, run_coef_id_fmincon_grad, run_coef_id_ga, run_coef_id_sa, run_problem_demo</p>

Ниже приведено краткое описание основных компонентов комплекса.

### Компонент «Модель»

Для представления исходной модели используются следующие структуры:

- `pde` — коэффициенты уравнения,
- `domain` — область изменения переменных,
- `ic` — начальное условие,
- `bc` — граничные условия,
- `bct` — типы граничных условий,

задаваемые внутри функции

```
[pde, domain, ic, bc, bct] = model().
```

Начальное и граничные условия описываются соответственно в функциях

```
c = ic_f(x), c = bc_f1(t), c = bc_f2(t).
```

### Компонент «Дискретизация»

Для дискретизации исходной модели используются структуры:

- `grid` — пространственно-временная сетка,
- `lds` — линейная динамическая система.

В структуре `grid` хранится информация о количестве узлов, узлах и шагах сетки, а в структуре `lds` — матрицы  $F$  и  $B$ .

Построение сетки выполняется с помощью функции

```
grid = compute_grid(pde, domain, nX, nT),
```

где  $nX$ ,  $nT$  — число узлов по соответствующим переменным (в случае отсутствия аргумента  $nT$ , число узлов по переменной  $t$  вычисляется автоматически из условия сходимости конечно-разностной схемы).

Дискретизация исходной модели выполняется при помощи функции

```
lds = pde2lds(pde, bc, bct, grid).
```

Решение прямой задачи методом конечных разностей реализовано в функции

```
solution = solve_findiff(pde, ic, bc, bct, grid),
```

возвращающей численное решение `solution` в виде матрицы размера  $nT \times nX$ .

Также в компоненте реализованы функции вычисления коэффициентов конечно-разностной схемы, их производных и др.

### Компонент «Измерения»

Для моделирования процесса измерений используется структура `sens`, хранящая информацию о матрицах  $H$  и  $R$ . Измеритель задается с помощью функции

```
sens = sensor(bct, nX, variance, problemtype),
```

где `variance` – дисперсия шума в измерителе, `problemtype` – строковый параметр, определяющий вид матрицы  $H$ .

Моделирование процесса измерений осуществляется путем последовательного решения исходного уравнения конечно-разностным методом, и зашумления значений решения в выбранных точках. Для этой цели используется функция

```
Z = measure(solution, bct, sens),
```

возвращающей матрицу результатов измерений  $Z \in \mathbb{R}^{m \times nT}$ , в которой по столбцам хранятся векторы измерений  $z_k$ ,  $k = 1, 2, \dots, nT$ .

### Компонент «Визуализация»

Для визуализации и вывода результатов работы компонентов комплекса в нем реализованы следующие группы функций: функции с именами, начинающимися с префикса `plot`, используются для построения графиков; функции с именами, начинающимися с префикса `print`, используются для вывода результатов в командное окно.

Рассмотрим процесс дискретизации исходной модели и моделирования процесса измерений на примере следующей модели.

**Пример 4.1.**

$$\frac{\partial c}{\partial t} + 2 \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}, \quad (4.1)$$

$$c(x, 0) = 0, \quad (4.2)$$

$$c(0, t) = |\sin 3\pi t|, \quad \frac{\partial c}{\partial x}(1, t) = -[c(1, t) - t], \quad (4.3)$$

$$x \in [0; 1], \quad t \in [0; 1].$$

В данном случае рассматривается модель со смешанными граничными условиями вида (1.1), (1.2), (1.4), где  $v = 2$ ,  $\alpha = 1$ ,  $\varphi(x) = 0$ ,  $f(t) = |\sin 3\pi t|$ ,  $g(t) = t$ ,  $\lambda = 1$ .

Сначала нужно задать основные параметры модели в функции `model`.

Листинг 4.1. Файл `model.m`

```

1 function [pde, domain, ic, bc, bct] = model()
2 pde.v = 2;           % convection speed
3 pde.alpha = 1;      % diffusion coefficient
4
5 domain.a = 0;       % left x bound
6 domain.b = 1;       % right x bound
7 domain.Tmin = 0;    % start time
8 domain.Tmax = 1;    % stop time
9
10 ic.f = @ic_f;       % initial condition function
11
12 bc.f1 = @bc_f1;     % left boundary condition function
13 bc.c1 = 1.0;        % left boundary condition constant
14 bc.f2 = @bc_f2;     % right boundary condition function
15 bc.c2 = 1.0;        % right boundary condition constant
16
17 bct.t1 = 1;         % left boundary condition type {1|3}
18 bct.t2 = 3;         % right boundary condition type {1|3}
19 bct.known = 0;      % known boundary condition {0|1|2}
20 end

```

Также должны быть реализованы функции `ic_f`, `bc_f1` и `bc_f2` с описанием начального и граничных условий. Например, функция, описывающая правое граничное условие, реализована в файле `bc_f1.m` следующим образом:

Листинг 4.2. Файл bc\_f1.m

```

1 function c = bc_f1(t)
2 c = abs(sin(3*pi*t))
3 end

```

На листинге 4.3 приведено содержимое скрипта `run_problem_demo` с описанием модели (4.1)–(4.3). В скрипте выполняется дискретизация модели на сетке с 6 пространственными узлами (количество узлов по переменной  $t$  вычисляется автоматически и равно 101), а затем моделируется процесс зашумленных измерений решения с дисперсией шумов  $0.02^2$ .

Листинг 4.3. Файл run\_problem\_demo.m

```

1 [pde, domain, ic, bc, bct] = model();
2
3 rng('default');
4
5 nX = 6;           % number of x-grid points
6 variance = 0.02^2;% noise variance
7
8 grid = compute_grid(pde, domain, nX);
9 lds = pde2lds(pde, bc, bct, grid);
10 solution = solve_findiff(pde, ic, bc, bct, grid);
11 sens = sensor(bct, nX, variance);
12 Z = measure(solution, bct, sens);
13
14 print_grid(grid);
15 print_lds(lds);
16 print_sensor(sens);
17
18 plot_solution(solution, grid);
19 plot_measurements(bct, grid, sens, Z);

```

Ниже приведены построенные в результате работы скрипта конечно-разностная сетка, линейная динамическая система и измеритель.

Конечно-разностная сетка:

```
nX = 6
nT = 101
xGrid = 0:0.2:1
tGrid = 0:0.01:1
```

Линейная динамическая система:

F(5x5):

```
0.5000    0.2000         0         0         0
0.3000    0.5000    0.2000         0         0
0    0.3000    0.5000    0.2000         0
0         0    0.3000    0.5000    0.2000
0         0    0.2500    0.4167    0.1667
```

B(5x2):

```
0.3000         0
0         0
0         0
0         0
0    0.1667
```

Измеритель:

H(2x5):

```
1    0    0    0    0
0    0    0    0    1
```

R(2x2):

```
1.0e-03 *
```

```
0.4000         0
0    0.4000
```

Графики решения, получаемого методом конечных разностей и смоделированных измерений приведены на рисунках 4.2 и 4.3 соответственно.

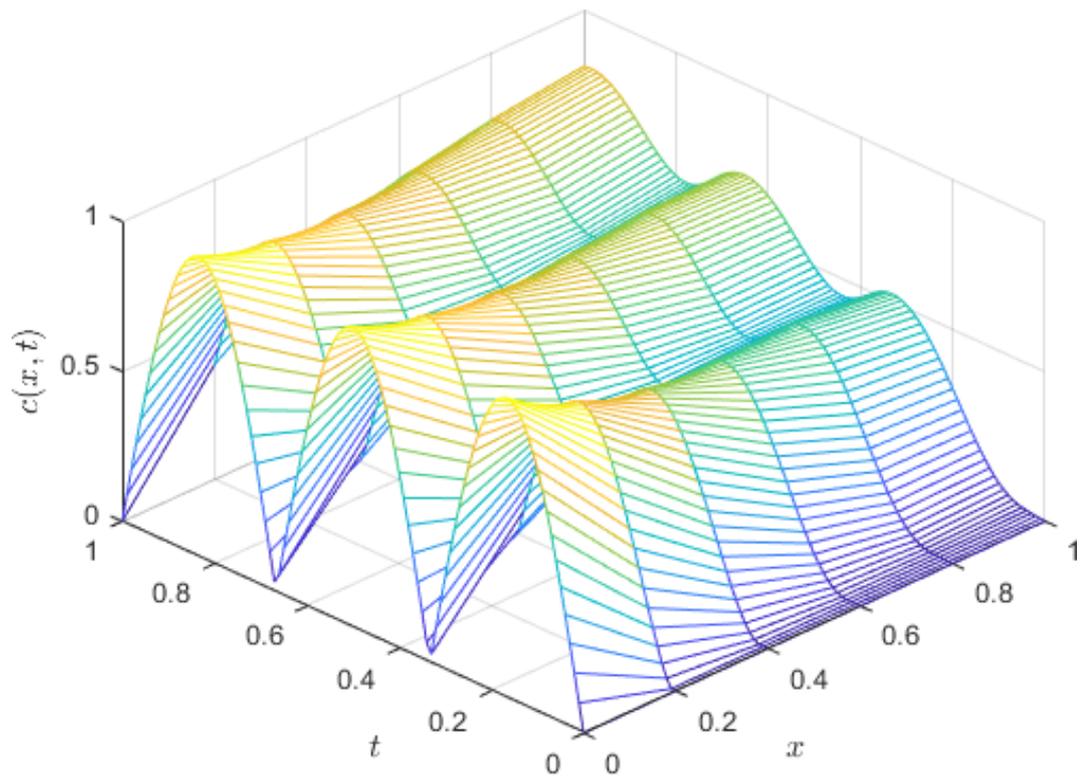


Рис. 4.2. Решение задачи (solution)

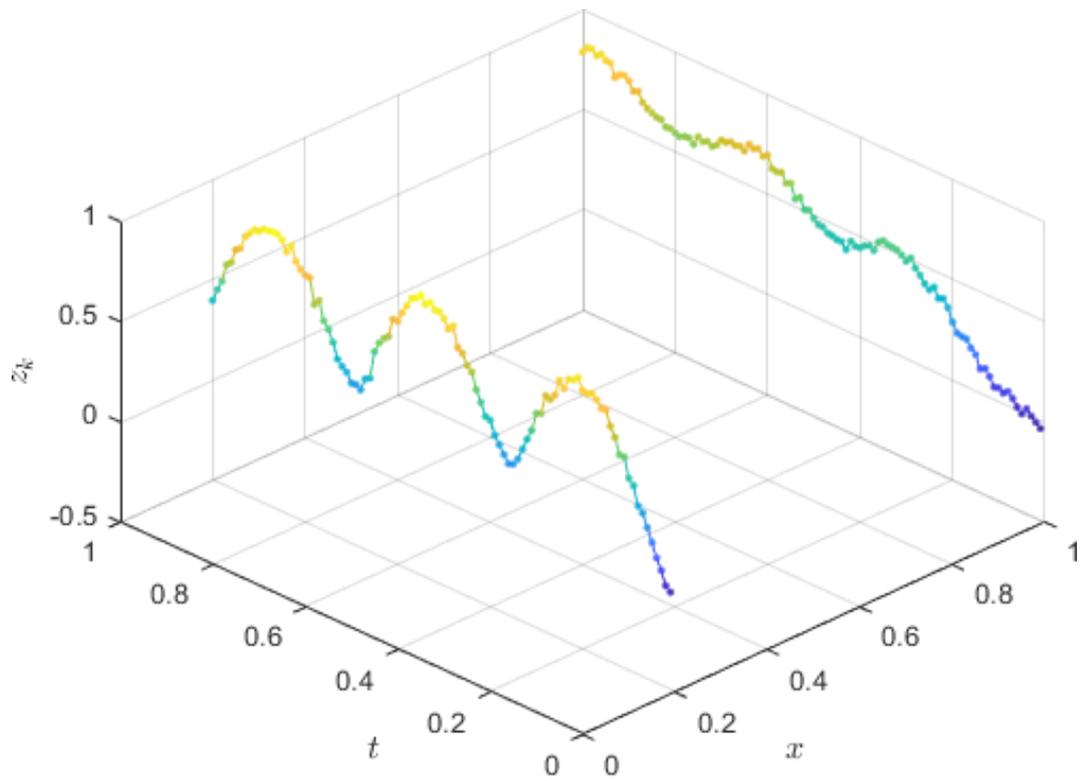


Рис. 4.3. Зашумленные измерения (Z)

## Компонент «Идентификация»

### Идентификация граничных условий

Блок-схема процесса идентификации граничных условий представлена на рисунке 4.4. Для работы алгоритмов необходима предварительная дискретизация задачи и зашумленные измерения значений решения в узлах пространственной сетки.



Рис. 4.4. Процесс идентификации граничных условий

Алгоритмы идентификации граничных условий 3.1, 3.2 и 3.3 реализованы соответственно в виде функций

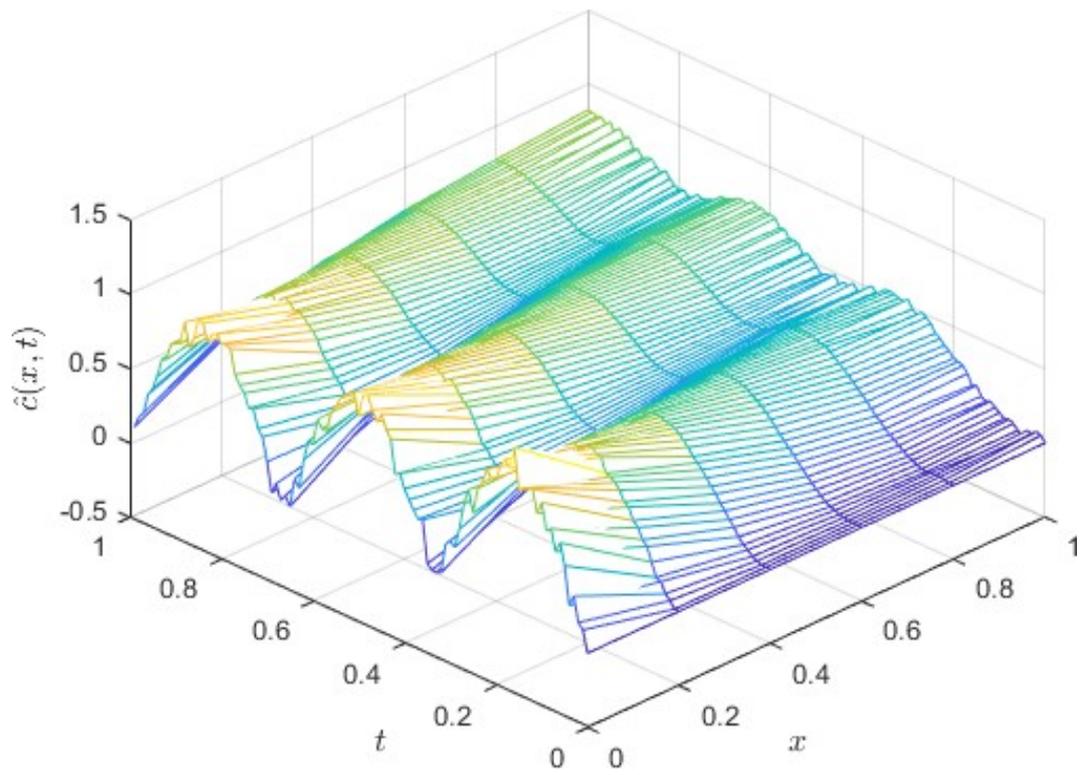
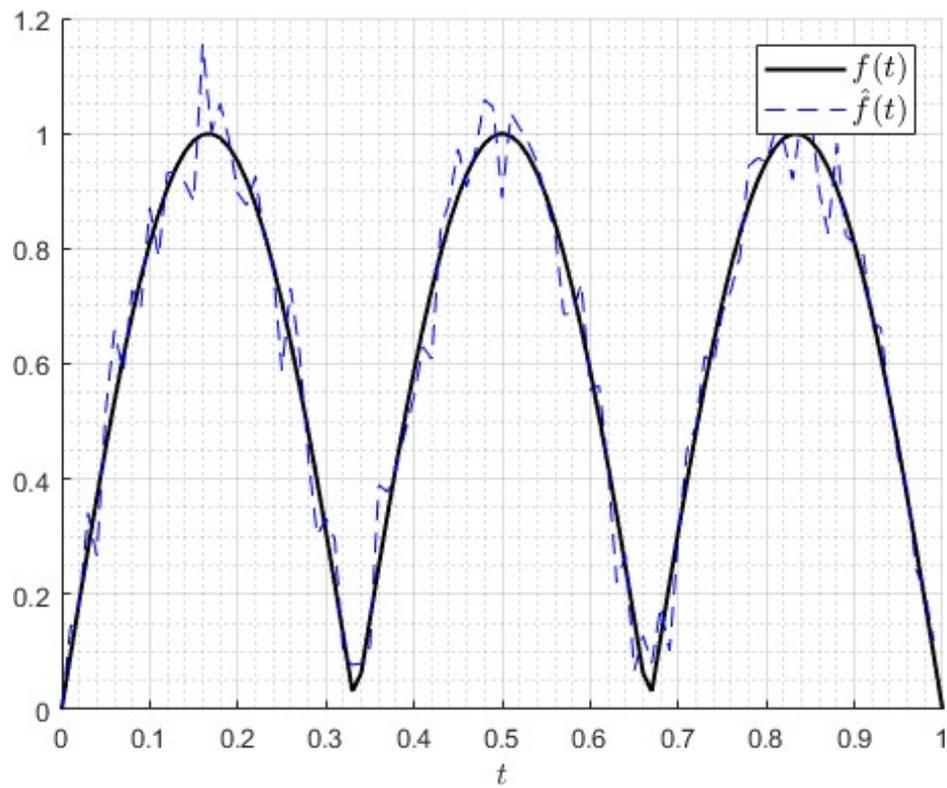
```
[estSol, estBCF2, PHistory, PUNhistory]
      = bc_id_gm_1(ic, bct, grid, lds, sens, Z),
[estSol, estBCF2, PHistory, PUNhistory]
      = bc_id_gm_2(ic, bct, grid, lds, sens, Z),
[estSol, estBCF2, PHistory, PUNhistory]
      = bc_id_gm_srcf(ic, bct, grid, lds, sens, Z).
```

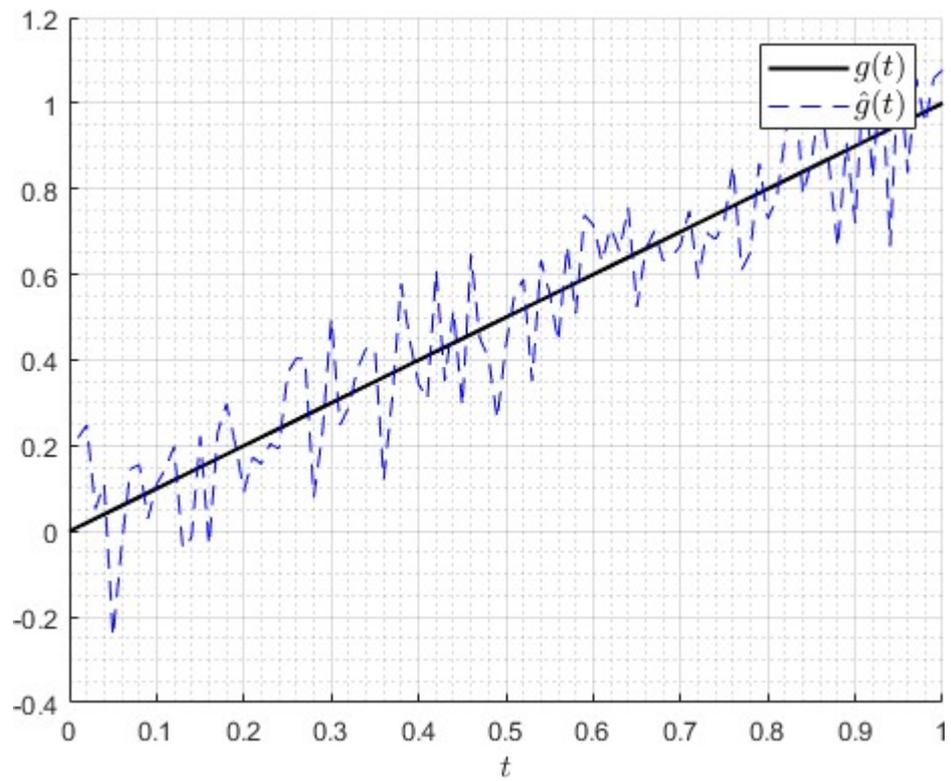
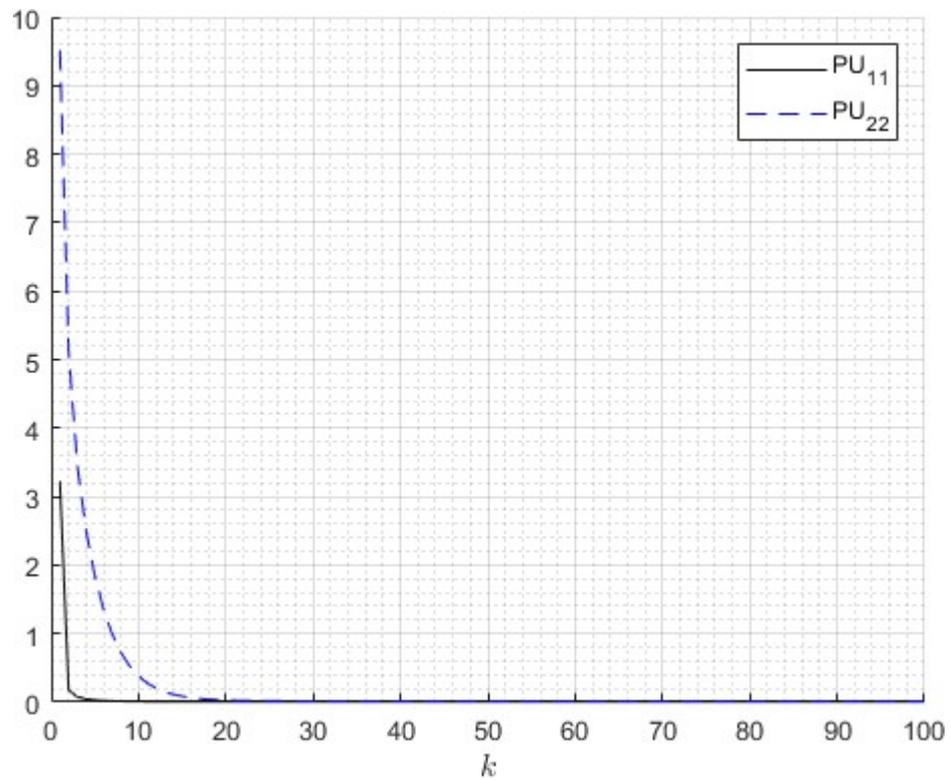
Выходными аргументами функций являются:

- `estSol` — оценка решения в узлах пространственно-временной сетки,
- `estBCF2` — оценка значений функции в граничном условии третьего рода,
- `PHistory` — ковариации ошибок оценивания вектора состояния,
- `PUNhistory` — ковариации ошибок оценивания вектора входных воздействий.

Переменная `estSol` представляет собой матрицу того же размера, что и матрица численного решения `solution`, полученного по методу конечных разностей. В случае, когда решается задача идентификации с двумя граничными условиями первого рода, их оценка сохраняется в первом и последнем столбце `estSol` (вектор `estBCF2` при этом является пустым), а в остальных столбцах хранятся оценки вектора состояния. Если решается задача идентификации с граничными условиями первого и третьего рода, то их оценки сохраняются соответственно в первом столбце `estSol` и векторе `estBCF2`, а в остальных столбцах хранятся оценки вектора состояния. В матрицах `PHistory` и `PUNhistory` по столбцам хранятся ковариации ошибок оценивания вектора состояния и измерений соответственно, характеризующие качество получаемых оценок.

Для запуска процесса идентификации граничных условий имеется скрипт `run_bc_id_gm`, содержимое и параметры которого аналогичны приведенному на листинге 4.3, но с добавлением после моделирования измерений вызова нужной функции `bc_id_gm_1`, `bc_id_gm_2` или `bc_id_gm_srcf` и визуализацией получаемых результатов. На рисунках 4.5–4.8 приведены некоторые результаты его работы для модели (4.1)–(4.3) и функции `bc_id_gm_2`.

Рис. 4.5. Оценка решения (`estSol`)Рис. 4.6. Функция  $f(t)$  и ее оценка (`estSol(:, 1)`)

Рис. 4.7. Функция  $g(t)$  и её оценкаРис. 4.8. Ковариации ошибок оценивания граничных значений (вектора  $u_k$ )

## Идентификация коэффициентов конвекции и диффузии

Блок-схема процесса идентификации коэффициентов конвекции и диффузии представлена на рисунке 4.9.

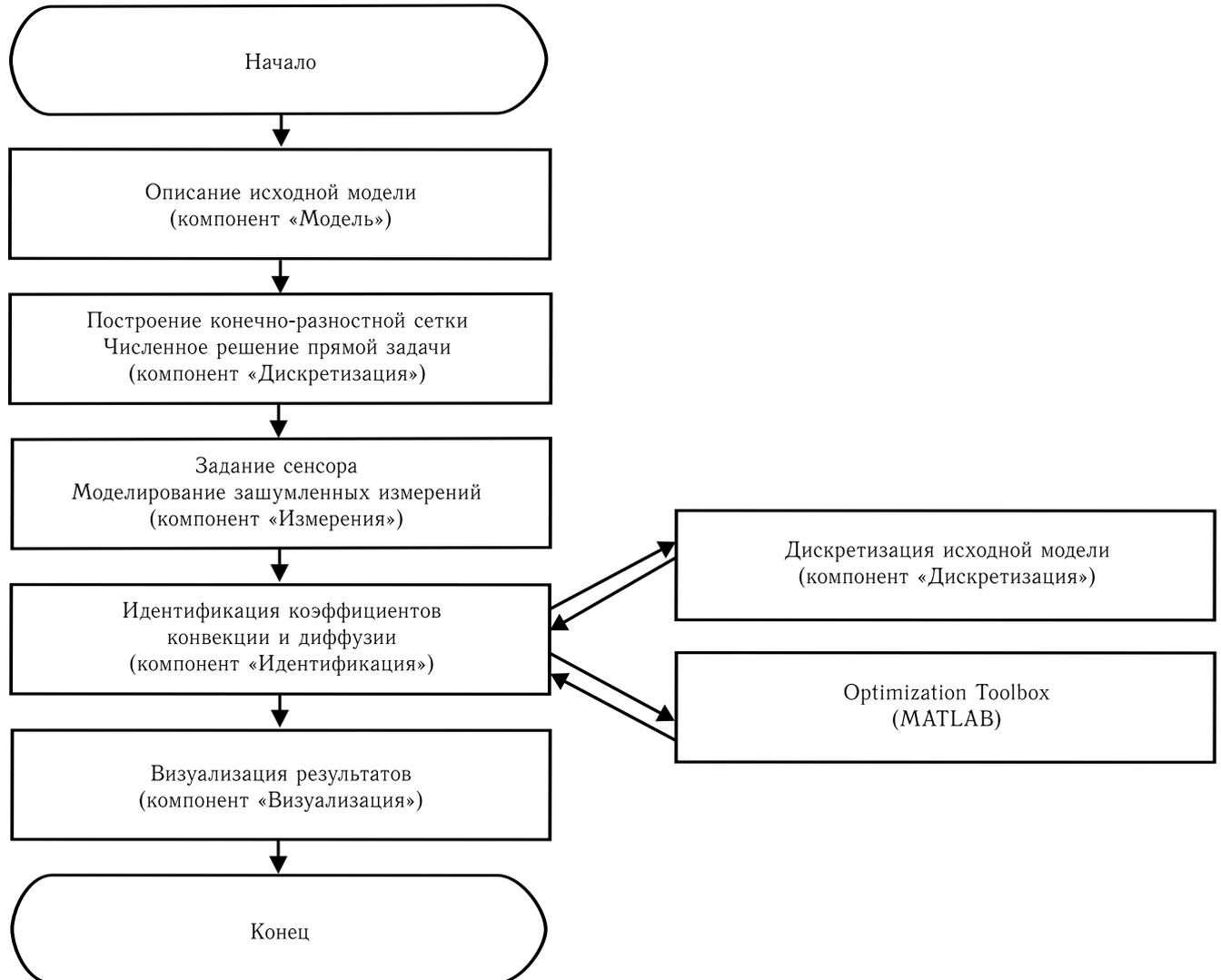


Рис. 4.9. Процесс идентификации коэффициентов конвекции и диффузии

В программном комплексе реализовано несколько методов идентификации коэффициентов уравнения (1.1), описанные в работах [37] и [38]. Данные методы основаны на поиске значений параметров  $v$  и  $\alpha$ , минимизирующих критерий идентификации, заданный в виде логарифмической функции правдоподобия, которая строится на основе величин, вычисляемых фильтром Калмана: вектора невязок измерений и его ковариационной матрицы (см., например, [48]).

Для вычисления значений критерия идентификации на основе стандартно-

го фильтра Калмана реализованы две функции:

```
LR = kf_lr(theta, bc, bct, grid, sens, phi, f1, f2, Z),
[LR, LRG]
= kf_lr_grad(theta, bc, bct, grid, sens, phi, f1, f2, Z).
```

Первый аргумент обеих функций — вектор значений параметров  $\theta = [v, \alpha]^T$ . Вторая функция отличается от первой дополнительным вычислением градиента критерия идентификации.

Минимизация критерия идентификации может выполняться различными функциями MATLAB, предназначенными для минимизации функций нескольких переменных. В программном комплексе для этих целей используются функции MATLAB `simulannealbnd` (Simulated Annealing — метод имитации отжига), `ga` (Genetic Algorithm — генетический алгоритм) и `fmincon` (условная минимизация) из модуля Optimization Toolbox. Первые две функции являются безградиентными метаэвристическими методами оптимизации, и в них в качестве целевой функции используется функция `kf_lr`. Функция `fmincon` может использоваться как с градиентной (`kf_lr_grad`), так и с безградиентной (`kf_lr`) версией функции вычисления значений критерия идентификации (во втором случае градиент целевой функции аппроксимируется самой функцией `fmincon`).

Для идентификации коэффициентов уравнения в программном комплексе реализованы функции `coef_id_ga`, `coef_id_sa`, `coef_id_fmincon` и `coef_id_fmincon_grad` с унифицированным интерфейсом. На листинге 4.4 приведен код функции `coef_id_sa`.

Листинг 4.4. Файл `coef_id_sa.m`

```
1 function theta = coef_id_sa(ic, bc, bct, grid, sens, saOptions, ...
   LB, UB, Z, theta0)
2 vLB = LB(1);
3 vUB = UB(1);
4 alphaLB = LB(2);
5 alphaUB = UB(2);
6
7 phi = ic.f(grid.xGrid); % phi(x)
```

```

8 f1 = bc.f1(grid.tGrid); % f(t)
9 f2 = bc.f2(grid.tGrid); % g(t)
10
11 if nargin < 10
12 v0 = vLB + 0.5*(vUB-vLB);
13 alpha0 = alphaLB + 0.5*(alphaUB - alphaLB);
14 theta0 = [v0 alpha0];
15 end
16
17 objFnc = @(theta) kf_lr(theta, bc, bct, grid, sens, phi, f1, f2, Z);
18
19 [theta, fval, exitflag, output] = simulannealbnd(objFnc, theta0, ...
    [vLB alphaLB], [vUB alphaUB], saOptions);
20 end

```

Настройка параметров и вызов нужного алгоритма идентификации выполняется в скриптах `run_coef_id_ga`, `run_coef_id_sa`, `run_coef_id_fmincon` и `run_coef_id_fmincon_grad`. Ниже приведено содержимое скрипта `run_coef_id_sa`.

Листинг 4.5. Файл `run_coef_id_sa.m`

```

1 [pde, domain, ic, bc, bct] = model();
2
3 rng('default');
4
5 nX = 6; % number of x-grid points
6 variance = 0.02^2; % noise variance
7
8 vLB = 0; % convection speed lower bound
9 vUB = 5; % convection speed upper bound
10 alphaLB = 0; % diffusion coefficient lower bound
11 alphaUB = 5; % diffusion coefficient upper bound
12
13 saOptions = saoptimset('simulannealbnd');
14 saOptions = saoptimset(saOptions, 'TimeLimit', Inf);
15 saOptions = saoptimset(saOptions, 'MaxIter', Inf);
16 saOptions = saoptimset(saOptions, 'ReannealInterval', 100);
17 saOptions = saoptimset(saOptions, 'StallIterLimit', 100);
18 saOptions = saoptimset(saOptions, 'MaxFunEvals', Inf);
19 saOptions = saoptimset(saOptions, 'Display', 'off');
20 saOptions = saoptimset(saOptions, 'DisplayInterval', 1);

```

```

21 saOptions = saoptimset(saOptions, 'OutputFcn', @sa_out);
22
23 grid = compute_grid(pde, domain, nX, 101);
24 solution = solve_findiff(pde, ic, bc, bct, grid);
25
26 sens = sensor(bct, nX, variance, 'eye');
27 Z = measure(solution, bct, sens);
28
29 theta = coef_id_sa(ic, bc, bct, grid, sens, saOptions, [vLB ...
    alphaLB], [vUB alphaUB], Z, 'svd_tm_lr');
30
31 disp(['v = ', num2str(theta(1),15), ', alpha = ', ...
    num2str(theta(2),15)]);
32 plot_solution(solution, grid);

```

### Результат работы скрипта (вывод в командное окно)

```

>> run_coef_id_sa
v = 1.92765518820729, alpha = 1.02136603822657

```

## 4.2. Примеры решения задач идентификации

Рассмотрим решение задач параметрической идентификации для следующей модели.

### Пример 4.2.

$$\frac{\partial c}{\partial t} + 2 \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}, \quad (4.4)$$

$$c(x, 0) = 0, \quad (4.5)$$

$$c(0, t) = |\sin 3\pi t|, \quad c(1, t) = t/2, \quad (4.6)$$

$$x \in [0; 1], \quad t \in [0; 1].$$

Зададим параметры модели, начальное и граничные условия на левом и правом концах в файлах `model.m`, `ic_f.m`, `bc_f1.m`, `bc_f2.m`.

В файле `model.m` в поле `bct.f2` задается тип условия на правом конце отрезка: 1 — условие первого рода, 3 — условие третьего рода (на левом конце предполагается условие первого рода). В поле `bct.known` необходимо указать

известные граничные условия: 0 — оба условия неизвестны, 1 — первое условие известно, 2 — второе условие известно.

Листинг 4.6. Файл model.m

```

1 function [pde, domain, ic, bc, bct] = model()
2 pde.v = 2;           % convection speed
3 pde.alpha = 1;      % diffusion coefficient
4
5 domain.a = 0;       % left x bound
6 domain.b = 1;       % right x bound
7 domain.Tmin = 0;    % start time
8 domain.Tmax = 1;    % stop time
9
10 ic.f = @ic_f;       % initial condition function
11
12 bc.f1 = @bc_f1;     % left boundary condition function
13 bc.c1 = 1.0;        % left boundary condition constant
14 bc.f2 = @bc_f2;     % right boundary condition function
15 bc.c2 = 1.0;        % right boundary condition constant
16
17 bct.t1 = 1;         % left boundary condition type {1|3}
18 bct.t2 = 1;         % right boundary condition type {1|3}
19 bct.known = 0;      % known boundary condition {0|1|2}
20 end

```

Листинг 4.7. Файл ic\_f.m

```

1 function c = ic_f(x)
2 %IC_F initial condition function
3 % Detailed explanation goes here
4
5 c = zeros(size(x));
6 end

```

Листинг 4.8. Файл bc\_f1.m

```

1 function c = bc_f1(t)
2 %BC_F1 left boundary condition function
3 % Detailed explanation goes here
4
5 c = abs(sin(3*pi*t));
6 end

```

Листинг 4.9. Файл bc\_f2.m

```

1 function c = bc_f2(t)
2 %BC_F2 right boundary condition function
3 % Detailed explanation goes here
4
5 c = t/2;
6 end

```

С помощью скрипта `run_problem_demo` построим конечно-разностную сетку, дискретизируем исходную модель с нужными параметрами, зададим модель измерений, построим графики начальных и граничных условий, найдем решение прямой задачи методом конечных разностей и смоделируем зашумленные измерения.

Листинг 4.10. Файл run\_problem\_demo.m

```

1 [pde, domain, ic, bc, bct] = model();
2
3 rng('default');
4
5 nX = 6;           % number of x-grid points
6 variance = 0.02^2;% noise variance
7
8 grid = compute_grid(pde, domain, nX);
9 lds = pde2lds(pde, bc, bct, grid);
10 solution = solve_findiff(pde, ic, bc, bct, grid);
11 sens = sensor(bct, nX, variance);
12 Z = measure(solution, bct, sens);
13
14 print_grid(grid);
15 print_lds(lds);
16 print_sensor(sens);
17
18 plot_conditions(ic, bc, bct, grid);
19 plot_solution(solution, grid);
20 plot_measurements(bct, grid, sens, Z);

```

**Результаты работы скрипта run\_problem\_demo**

Вывод в командное окно:

```
>> run_problem_demo
```

```
nX = 6
```

```
nT = 101
```

```
xGrid = 0:0.2:1
```

```
tGrid = 0:0.01:1
```

```
F(4x4):
```

```
0.5000    0.2000         0         0
```

```
0.3000    0.5000    0.2000         0
```

```
0    0.3000    0.5000    0.2000
```

```
0         0    0.3000    0.5000
```

```
B(4x2):
```

```
0.3000         0
```

```
0         0
```

```
0         0
```

```
0    0.2000
```

```
H(2x4):
```

```
1    0    0    0
```

```
0    0    0    1
```

```
R(2x2):
```

```
1.0e-03 *
```

```
0.4000         0
```

```
0    0.4000
```

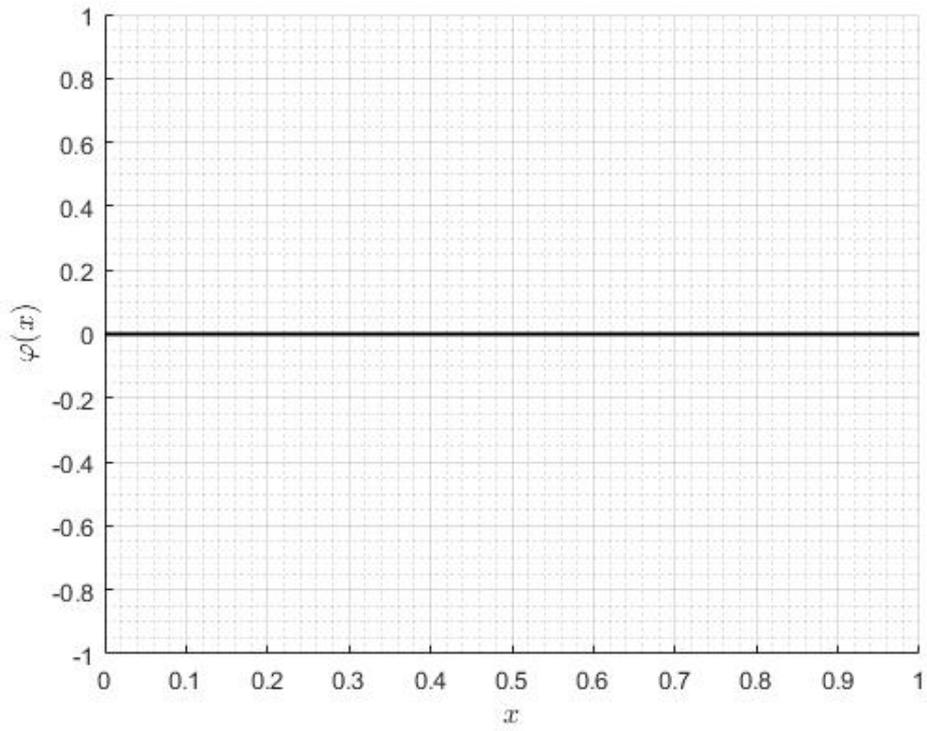


Рис. 4.10. Начальное условие

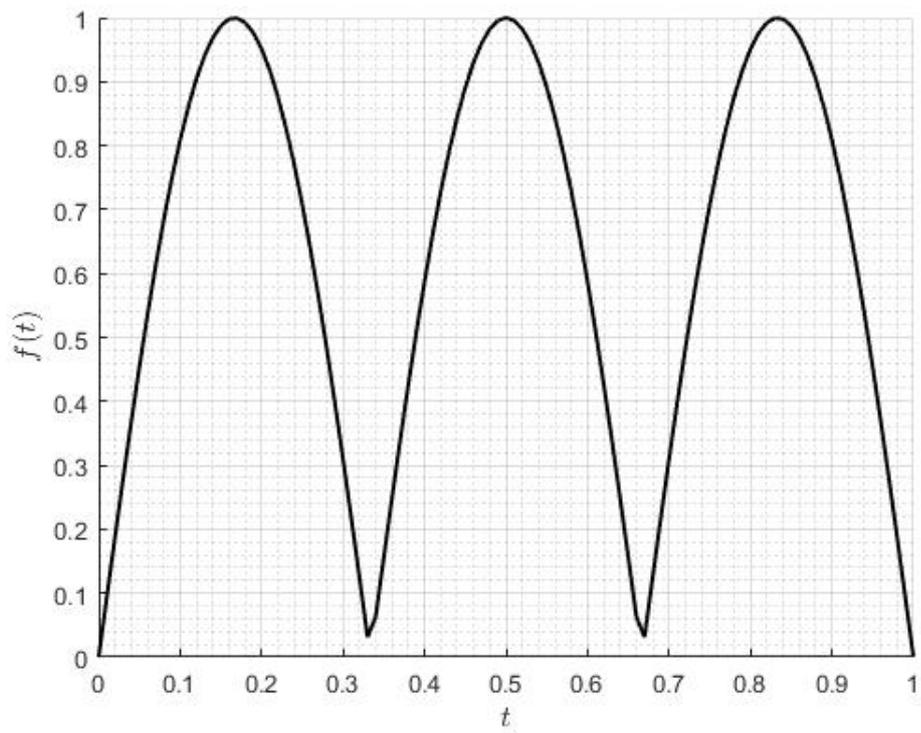


Рис. 4.11. Левое граничное условие

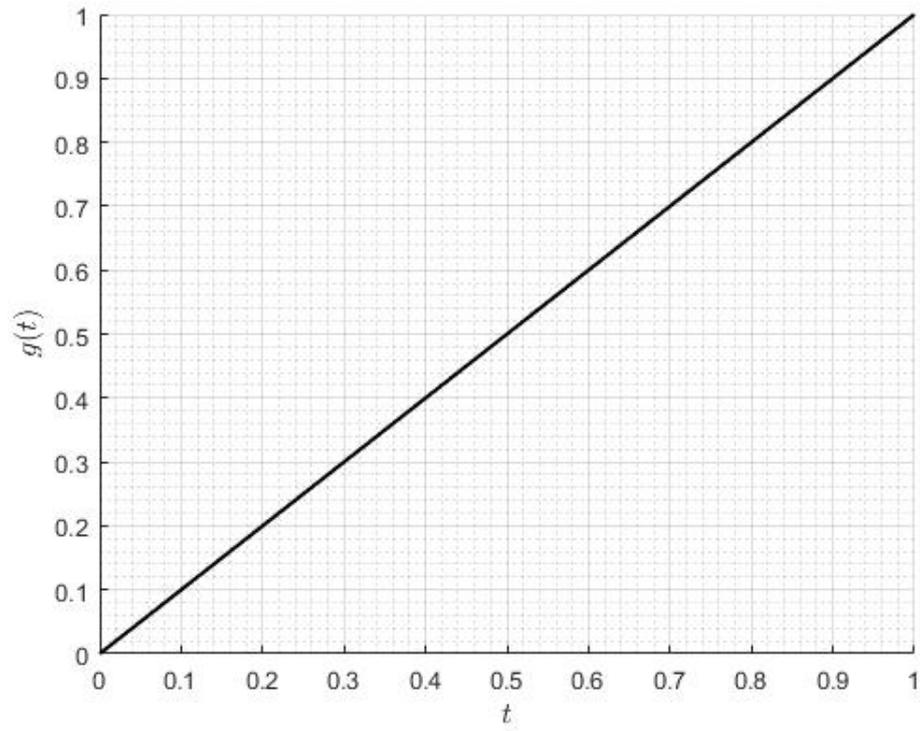


Рис. 4.12. Правое граничное условие

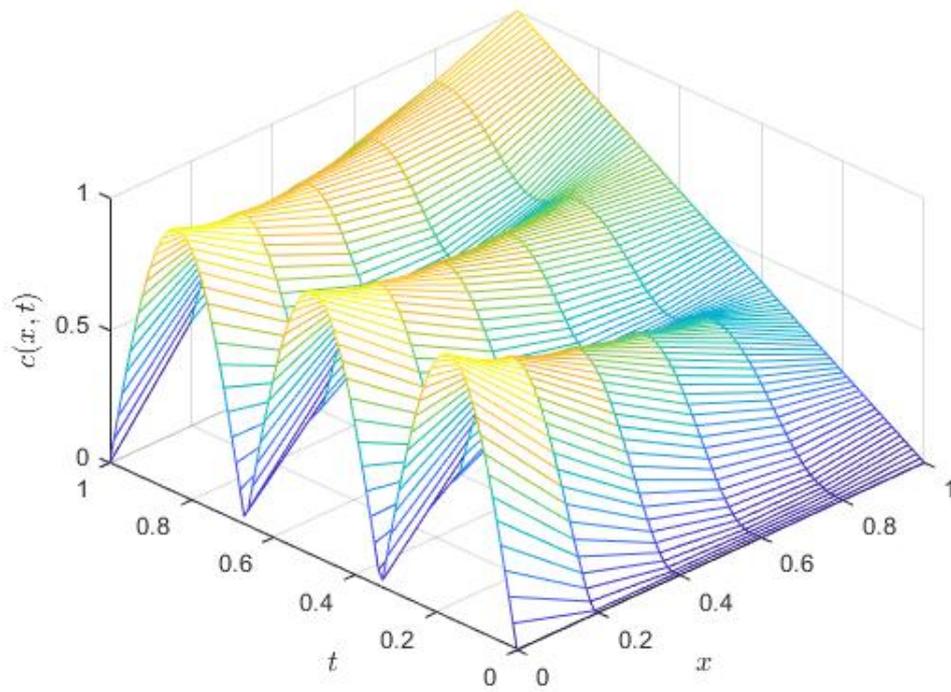


Рис. 4.13. Решение задачи

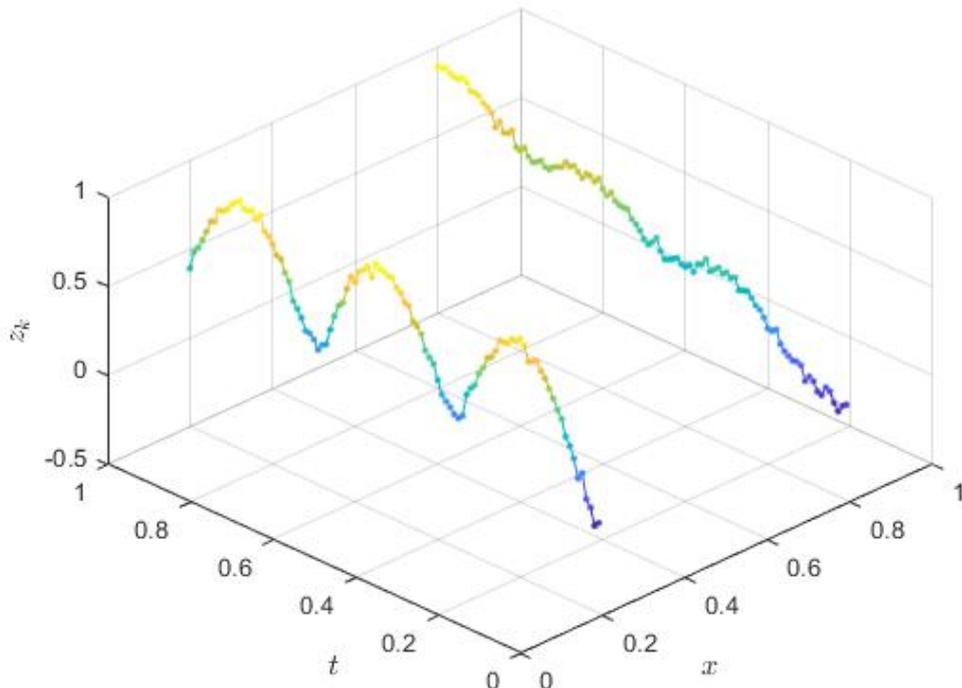


Рис. 4.14. Зашумленные измерения

Идентификацию граничных условий выполним алгоритмом 3.1 с помощью скрипта `run_bc_id_gm`.

Листинг 4.11. Файл `run_bc_id_gm.m`

```

1 [pde, domain, ic, bc, bct] = model();
2
3 rng('default');
4
5 nX = 6;           % number of x-grid points
6 variance = 0.01^2;% noise variance
7
8 grid = compute_grid(pde, domain, nX);
9 lds = pde2lds(pde, bc, bct, grid);
10 solution = solve_findiff(pde, ic, bc, bct, grid);
11 sens = sensor(bct, nX, variance);
12 Z = measure(solution, bct, sens);
13
14 switch bct.known
15 case 0
16 [estSol, estBCF2, PHistory, PUHistory] = bc_id_gm_1(ic, bct, grid, ...
    lds, sens, Z);

```

```
17 case 1
18 [estSol, estBCF2, PHistory, PUHistory] = bc_id_gm_1(ic, bct, grid, ...
    lds, sens, Z, bc.f1);
19 case 2
20 [estSol, estBCF2, PHistory, PUHistory] = bc_id_gm_1(ic, bct, grid, ...
    lds, sens, Z, bc.f2);
21 otherwise
22 error('Wrong parameter bct.known!');
23 end
24
25 plot_solution(solution, grid);
26 plot_measurements(bct, grid, sens, Z);
27 plot_estimates(bc, bct, grid, estSol, estBCF2);
28 plot_p(grid, PHistory, PUHistory);
```

### Результаты работы скрипта run\_bc\_id\_gm

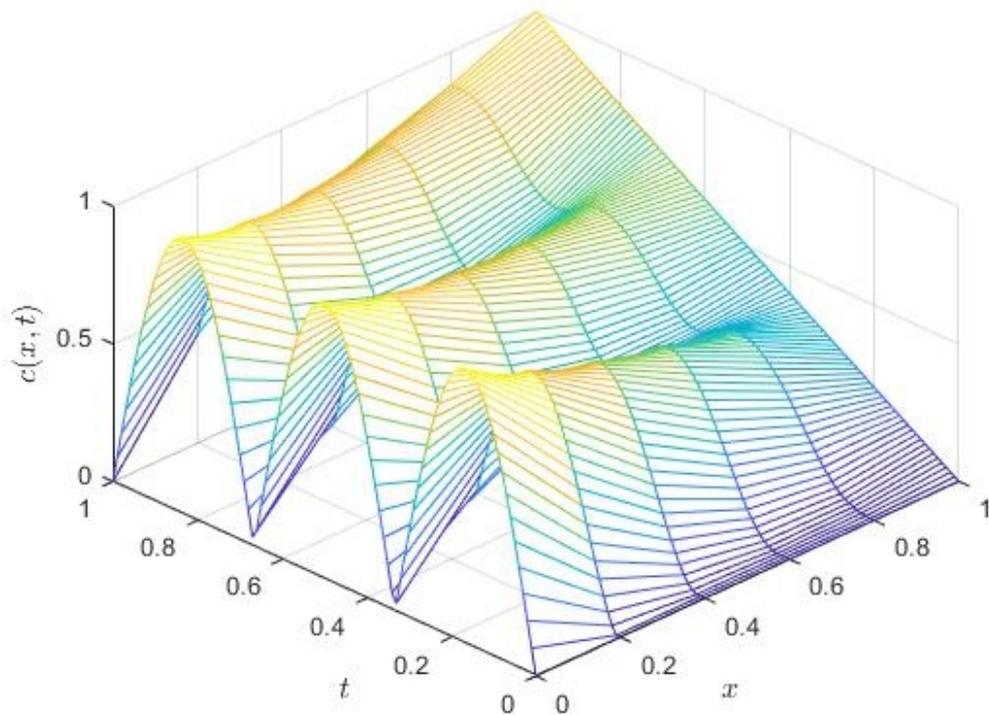


Рис. 4.15. Решение задачи

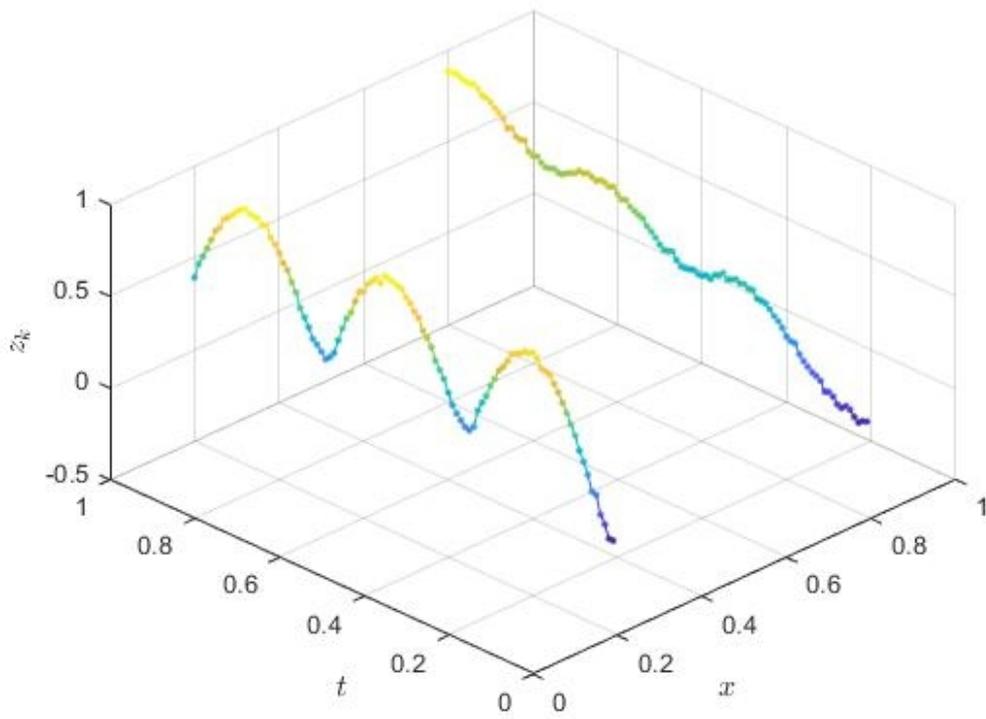


Рис. 4.16. Зашумленные измерения

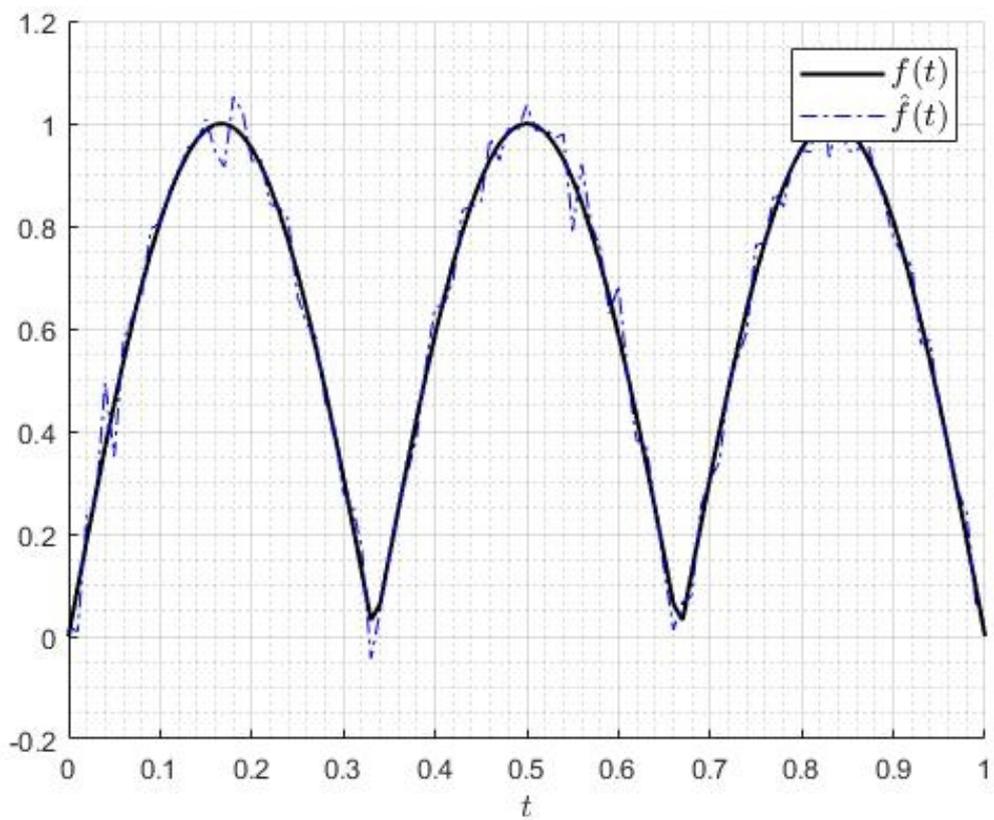


Рис. 4.17. Левое граничное условие и его оценка

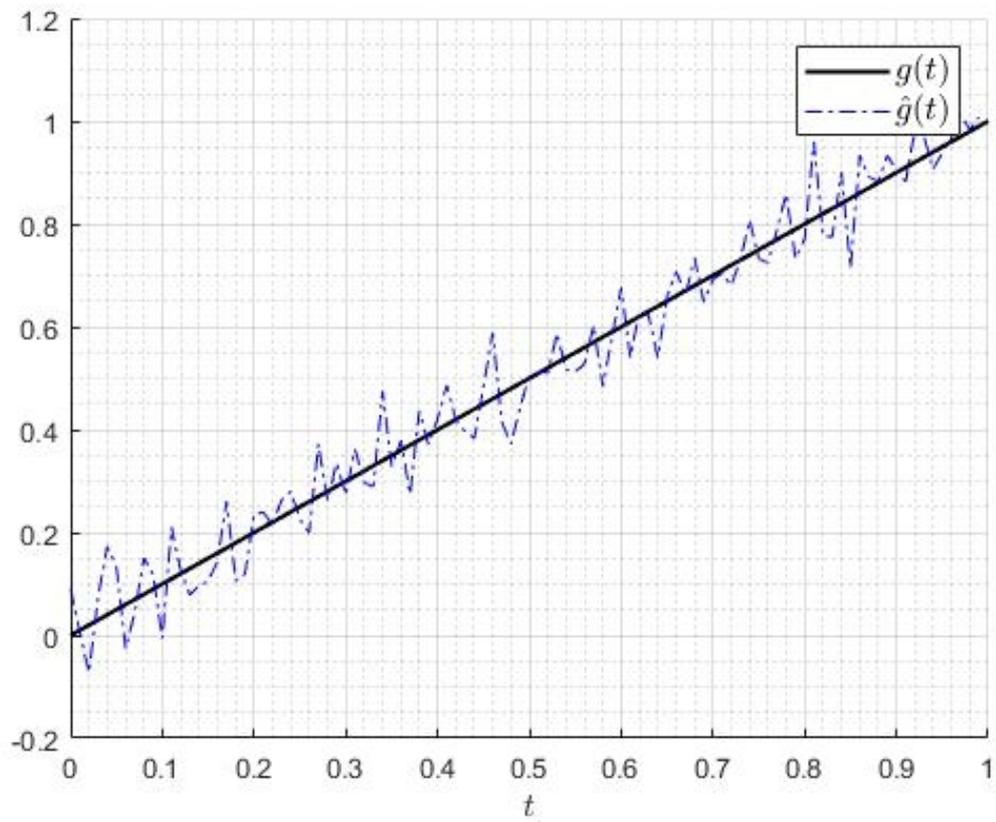


Рис. 4.18. Правое граничное условие и его оценка

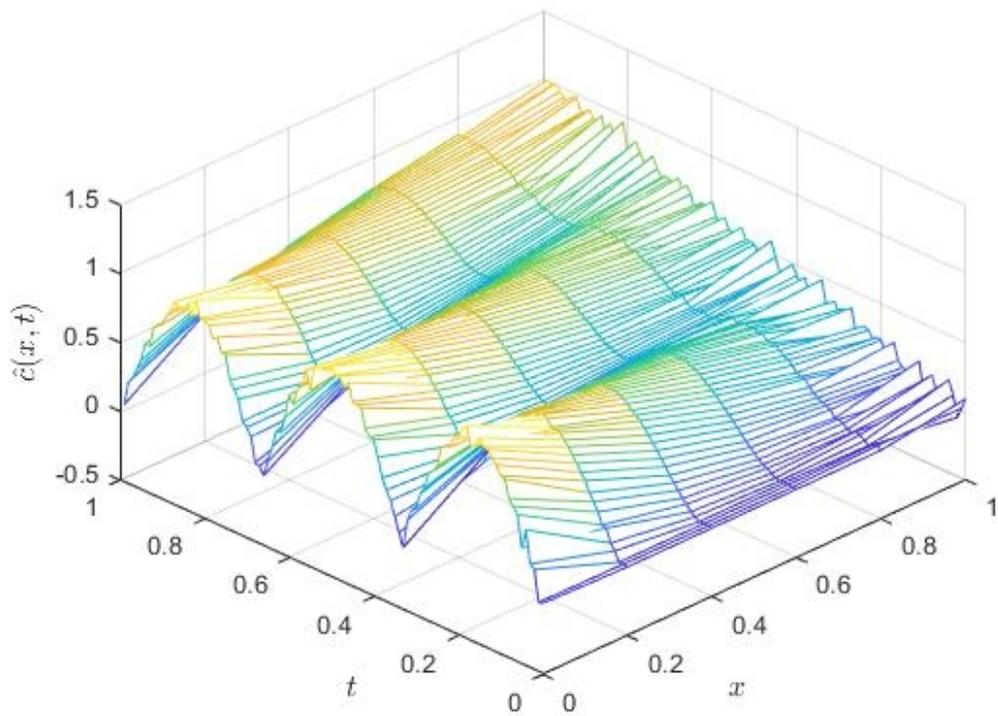


Рис. 4.19. Оценка решения

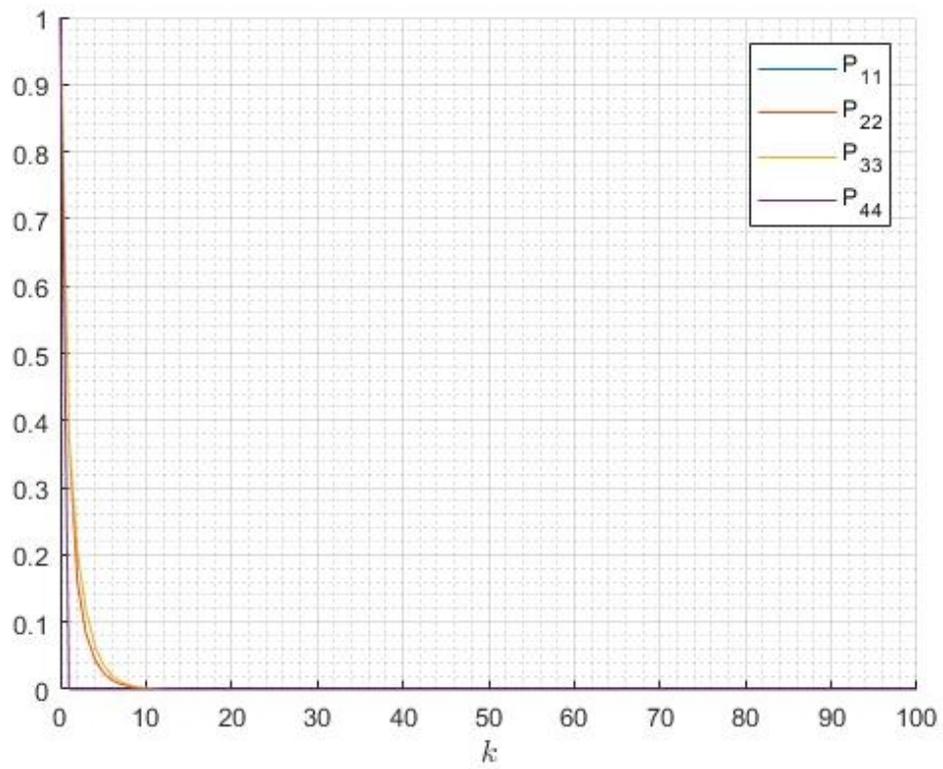


Рис. 4.20. Ковариации ошибок оценивания вектора состояния

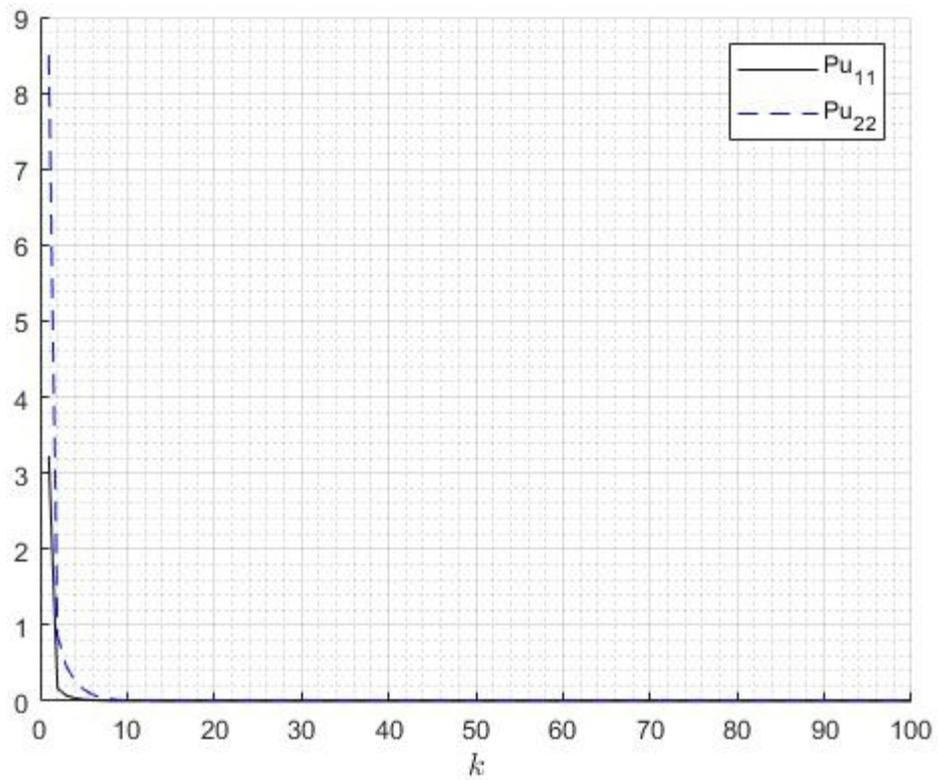


Рис. 4.21. Ковариации ошибок оценивания вектора входных воздействий

Идентификация коэффициентов уравнения может быть выполнена с помощью следующих скриптов, в каждом из которых для минимизации критерия идентификации используется одна из встроенных функций MATLAB:

- `run_coef_id_fmincon` — функция условной минимизации `fmincon`,
- `run_coef_id_fmincon_grad` — функция условной минимизации `fmincon` с градиентом,
- `run_coef_id_ga` — генетический алгоритм `ga`,
- `run_coef_id_sa` — метод имитации отжига `simulannealbnd`.

Во всех скриптах, кроме `run_coef_id_fmincon_grad` в качестве критерия идентификации могут быть указаны критерии как на основе стандартного фильтра Калмана, так и его SVD-модификации.

В качестве примера рассмотрим скрипт `run_coef_id_fmincon`.

Листинг 4.12. Файл `run_coef_id_fmincon.m`

```

1 [pde, domain, ic, bc, bct] = model();
2
3 rng('default');
4
5 nX = 6;           % number of x-grid points
6 variance = 0.02^2;% noise variance
7
8 vLB = 0;         % convection speed lower bound
9 vUB = 5;         % convection speed upper bound
10 alphaLB = 0;    % diffusion coefficient lower bound
11 alphaUB = 5;    % diffusion coefficient upper bound
12
13 minOptions = optimoptions('fmincon');
14 minOptions = optimoptions(minOptions, 'SpecifyObjectiveGradient', ...
    false);
15 %minOptions = optimoptions(minOptions, 'Algorithm','sqp');
16 minOptions = optimoptions(minOptions, 'MaxFunctionEvaluations', 500);
17 minOptions = optimoptions(minOptions, 'OptimalityTolerance', 1e-16);
18 minOptions = optimoptions(minOptions, 'FunctionTolerance', 1e-16);
19 minOptions = optimoptions(minOptions, 'StepTolerance', 1e-16);
20 minOptions = optimoptions(minOptions, 'Display', 'off');
21
22 grid = compute_grid(pde, domain, nX);

```

```

23 sens = sensor(bct, nX, variance, 'coef');
24 Z = measure(solution, bct, sens);
25
26 theta = coef_id_fmincon(ic, bc, bct, grid, sens, minOptions, [vLB ...
    alphaLB], [vUB alphaUB], Z, 'svd_tm_lr');
27
28 disp(['v = ', num2str(theta(1),15), ', alpha = ', ...
    num2str(theta(2),15)]);

```

### Результат работы скрипта run\_coef\_id\_fmincon

```

>> run_coef_id_fmincon
v = 2.00475667413572, alpha = 0.986397436241291

```

С помощью скрипта `exp_plot_criterion` можно построить график выбранного критерия идентификации.

Листинг 4.13. Файл `exp_plot_criterion.m`

```

1 [pde, domain, ic, bc, bct] = model();
2
3 rng('default');
4
5 nX = 6;           % number of x-grid points
6 variance = 0.02e-2;% noise variance
7
8 vLB = 0;         % convection speed lower bound
9 vUB = 5;         % convection speed upper bound
10 alphaLB = 0;    % diffusion coefficient lower bound
11 alphaUB = 5;    % diffusion coefficient upper bound
12
13 dv = 0.2;       % v step
14 dalpha = 0.2;   % alpha step
15
16 grid = compute_grid(pde, domain, nX, 101);
17 sens = sensor(bct, nX, variance, 'coef');
18 Z = measure(solution, bct, sens);
19
20 phi = ic.f(grid.xGrid); % phi(x)
21 f1 = bc.f1(grid.tGrid); % f(t)
22 f2 = bc.f2(grid.tGrid); % g(t)
23
24 vGrid = vLB:dv:vUB;

```

```

25 alphaGrid = alphaLB:dalpha:alphaUB;
26 nV = length(vGrid);
27 nAlpha = length(alphaGrid);
28
29 objFun = zeros(nAlpha, nV);
30
31 for i = 1:nAlpha
32 for j = 1:nV
33 v = vLB + dv*(j-1);
34 alpha = alphaLB + dalpha*(i-1);
35 objFun(i, j) = kf_lr([v, alpha], bc, bct, grid, sens, phi, f1, f2, ...
    Z);
36 end
37 end
38
39 figure
40 [V, ALPHA] = meshgrid(vGrid, alphaGrid);
41 meshc(V, ALPHA, objFun);
42 ax = xlabel('$v$');
43 ay = ylabel('$\alpha$');
44 az = zlabel('$\mathrm{kf\_lr}$');
45 set(ax, 'Interpreter', 'latex', 'FontSize', 12);
46 set(ay, 'Interpreter', 'latex', 'FontSize', 12);
47 set(az, 'Interpreter', 'latex', 'FontSize', 12);

```

## Результат работы скрипта `exp_plot_criterion`

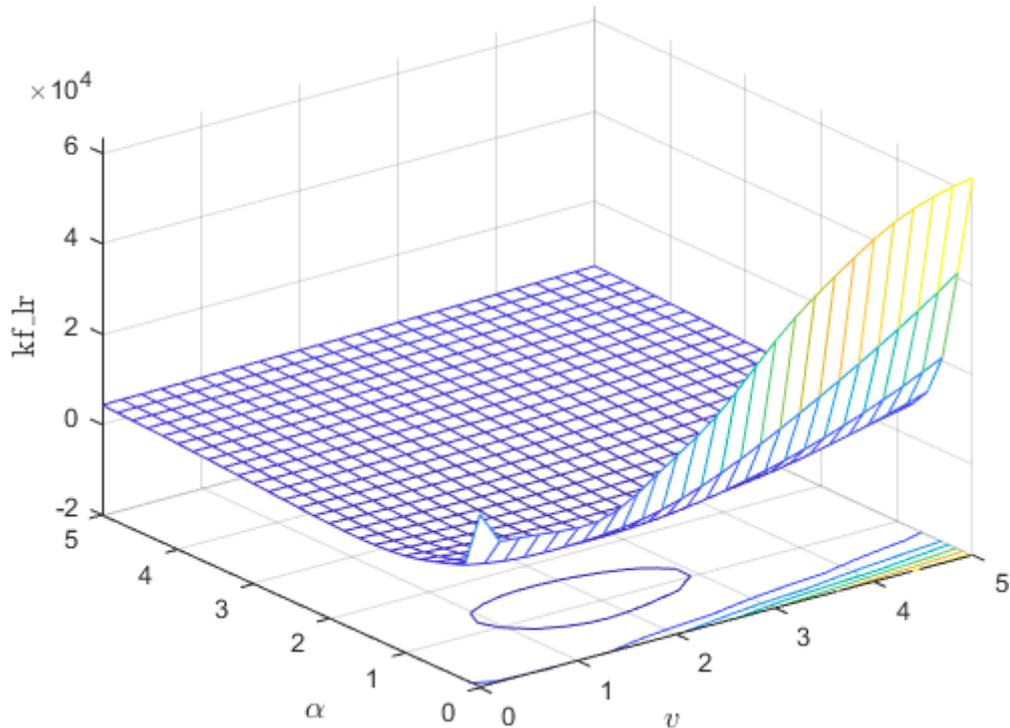


Рис. 4.22. Критерий идентификации на основе стандартного фильтра Калмана

### Пример 4.3.

$$\frac{\partial c}{\partial t} + \frac{\partial c}{\partial x} = \frac{\partial^2 c}{\partial x^2}, \quad (4.7)$$

$$c(x, 0) = 10x(1 - x), \quad (4.8)$$

$$c(0, t) = 5t(1 - t), \quad c(1, t) = 0, \quad (4.9)$$

$$x \in [0; 1], \quad t \in [0; 1].$$

Пусть требуется идентифицировать граничное условие на левом конце отрезка следующей модели при известном правом граничном условии с помощью алгоритма 3.3.

Сначала зададим параметры модели, начальное и граничные условия на левом и правом концах в файлах `model.m`, `ic_f.m`, `bc_f1.m`, `bc_f2.m`.

Листинг 4.14. Файл `model.m`

```

1 function [pde, domain, ic, bc, bct] = model()
2 pde.v = 1;           % convection speed

```

```

3  pde.alpha = 1;      % diffusion coefficient
4
5  domain.a = 0;      % left x bound
6  domain.b = 1;      % right x bound
7  domain.Tmin = 0;   % start time
8  domain.Tmax = 1;   % stop time
9
10 ic.f = @ic_f;      % initial condition function
11
12 bc.f1 = @bc_f1;     % left boundary condition function
13 bc.c1 = 1.0;        % left boundary condition constant
14 bc.f2 = @bc_f2;     % right boundary condition function
15 bc.c2 = 1.0;        % right boundary condition constant
16
17 bct.t1 = 1;         % left boundary condition type {1|3}
18 bct.t2 = 1;         % right boundary condition type {1|3}
19 bct.known = 2;      % known boundary condition {0|1|2}
20 end

```

Листинг 4.15. Файл ic\_f.m

```

1  function c = ic_f(x)
2  %IC_F initial condition function
3  % Detailed explanation goes here
4
5  c = 10*x.*(1-x);
6  end

```

Листинг 4.16. Файл bc\_f1.m

```

1  function c = bc_f1(t)
2  %BC_F1 left boundary condition function
3  % Detailed explanation goes here
4
5  c = 5*t.*(1-t);
6  end

```

Листинг 4.17. Файл bc\_f2.m

```

1  function c = bc_f2(t)
2  %BC_F2 right boundary condition function
3  % Detailed explanation goes here
4

```

```

5 c = zeros(size(t));
6 end

```

Затем воспользуемся скриптом `run_bc_id_gm`.

Листинг 4.18. Файл `run_bc_id_gm.m`

```

1 [pde, domain, ic, bc, bct] = model();
2
3 rng('default');
4
5 nX = 6;           % number of x-grid points
6 variance = 0.01^2;% noise variance
7
8 grid = compute_grid(pde, domain, nX);
9 lds = pde2lds(pde, bc, bct, grid);
10 solution = solve_findiff(pde, ic, bc, bct, grid);
11 sens = sensor(bct, nX, variance);
12 Z = measure(solution, bct, sens);
13
14 switch bct.known
15 case 0
16 [estSol, estBCF2, PHistory, PUHistory] = bc_id_gm_srcf(ic, bct, ...
17     grid, lds, sens, Z);
18 case 1
19 [estSol, estBCF2, PHistory, PUHistory] = bc_id_gm_srcf(ic, bct, ...
20     grid, lds, sens, Z, bc.f1);
21 case 2
22 [estSol, estBCF2, PHistory, PUHistory] = bc_id_gm_srcf(ic, bct, ...
23     grid, lds, sens, Z, bc.f2);
24 otherwise
25 error('Wrong parameter bct.known!');
26 end
27
28
29 print_lds(lds);
30 print_grid(grid);
31 print_sensor(sens);
32
33 plot_solution(solution, grid);
34 plot_measurements(bct, grid, sens, Z);
35 plot_estimates(bc, bct, grid, estSol, estBCF2);
36 plot_p(grid, PHistory, PUHistory);

```

**Результаты работы скрипта run\_bc\_id\_gm**

Вывод в командное окно:

```
>> run_bc_id_gm
```

```
F(4x4):
```

```
0.5000    0.2250         0         0
0.2750    0.5000    0.2250         0
0         0.2750    0.5000    0.2250
0         0         0.2750    0.5000
```

```
B1(4x1):
```

```
0
0
0
0.2250
```

```
B(4x1):
```

```
0.2750
0
0
0
```

```
nX = 6
```

```
nT = 101
```

```
xGrid = 0:0.2:1
```

```
tGrid = 0:0.01:1
```

```
H(1x4):
```

```
1    0    0    0
```

```
R(1x1):
```

```
1.0000e-04
```

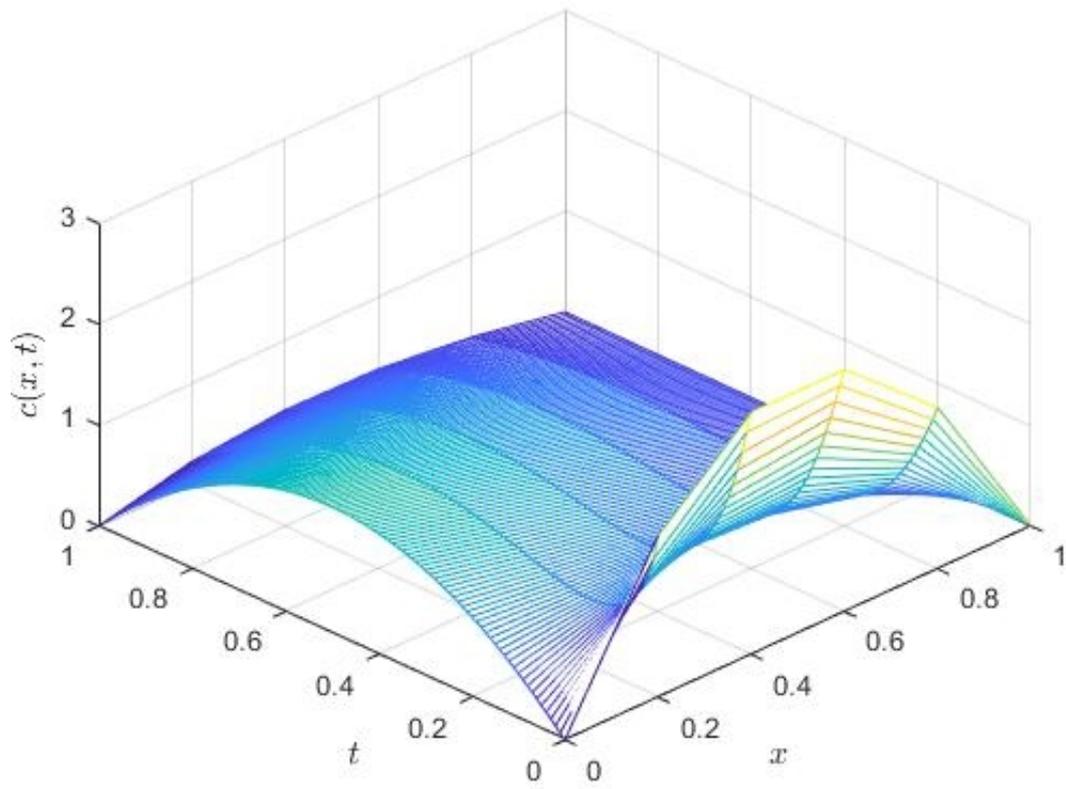


Рис. 4.23. Решение задачи

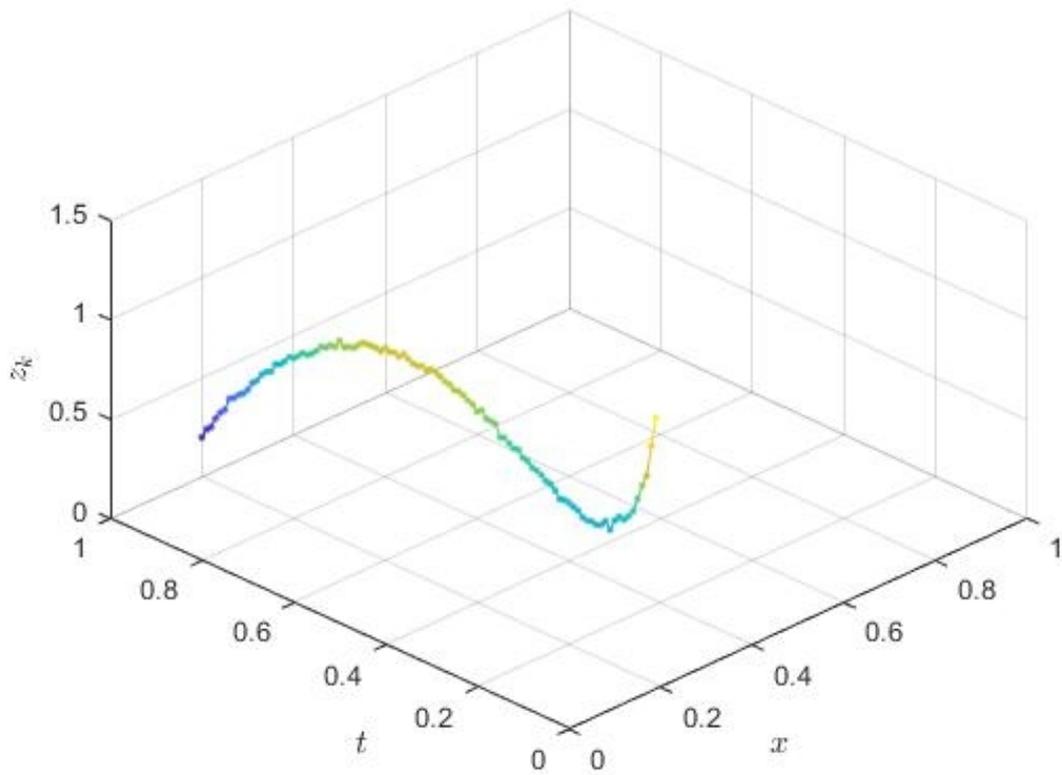


Рис. 4.24. Зашумленные измерения

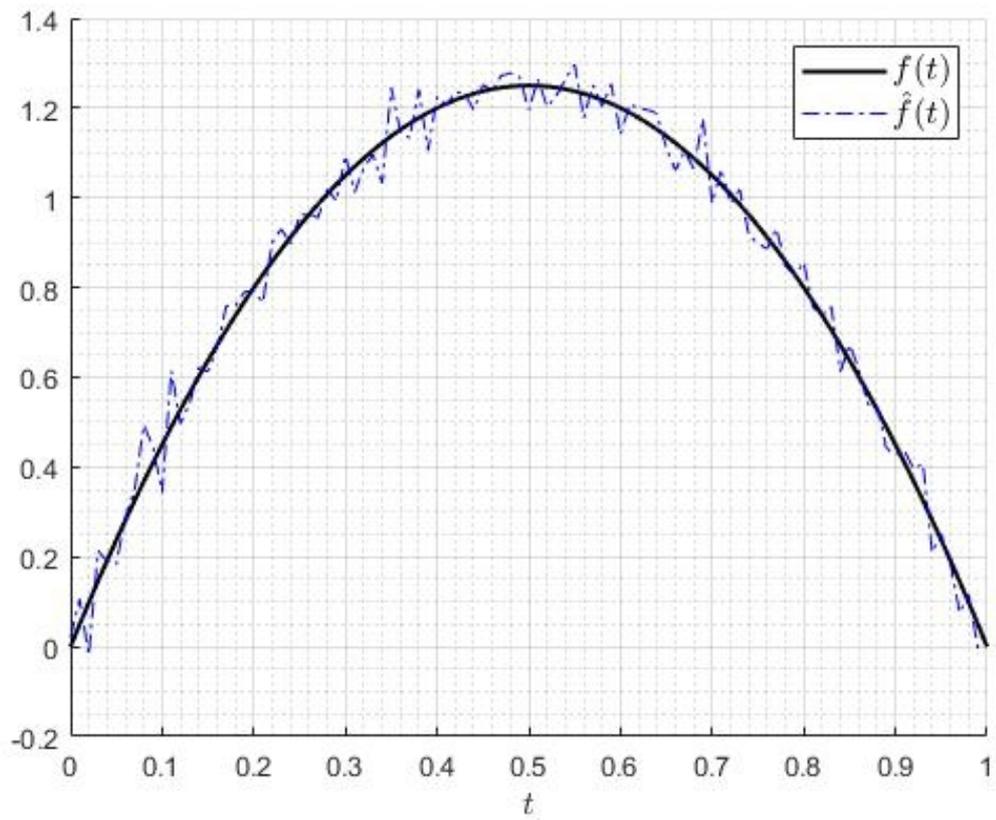


Рис. 4.25. Левое граничное условие и его оценка

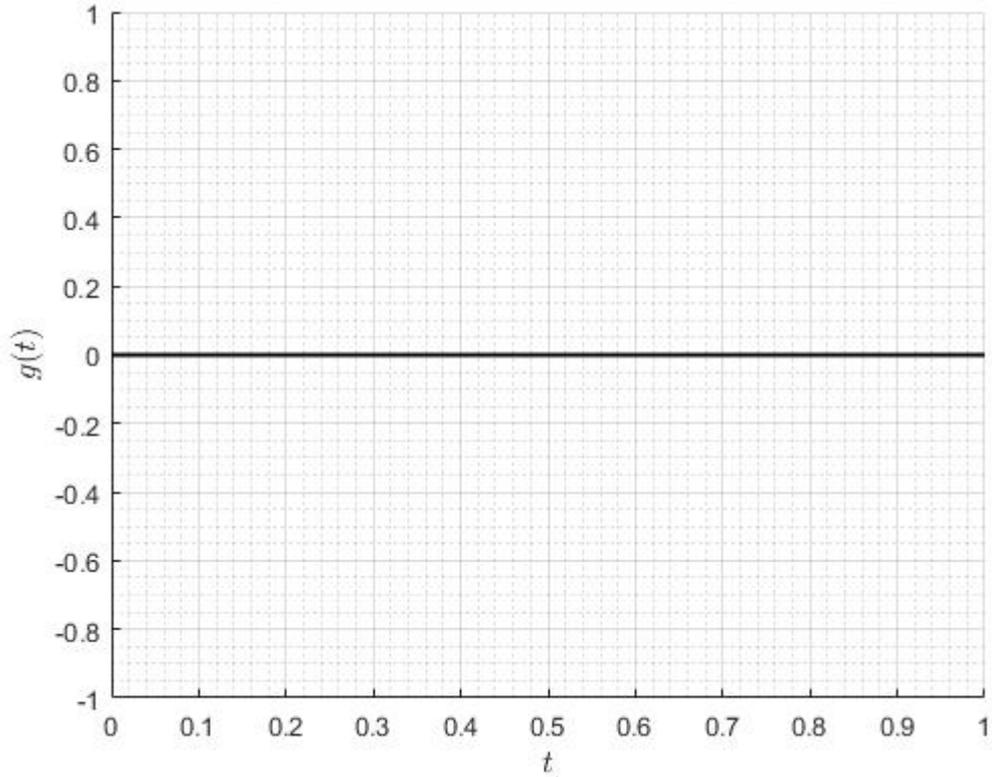


Рис. 4.26. Правое граничное условие

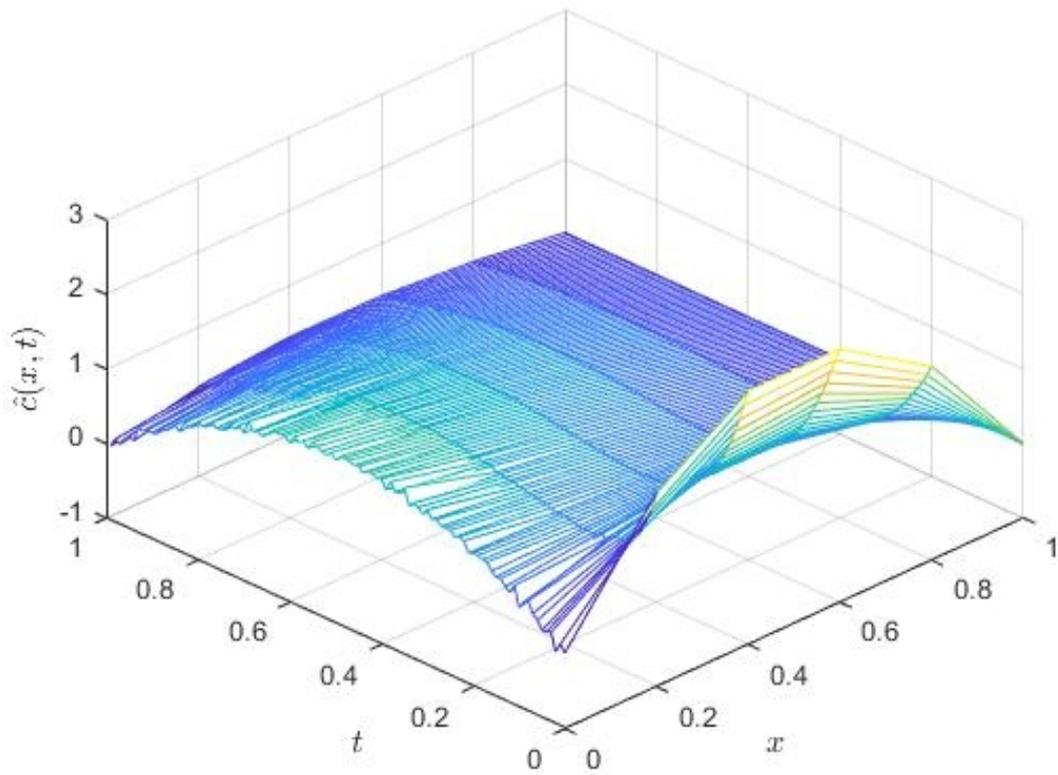


Рис. 4.27. Оценка решения

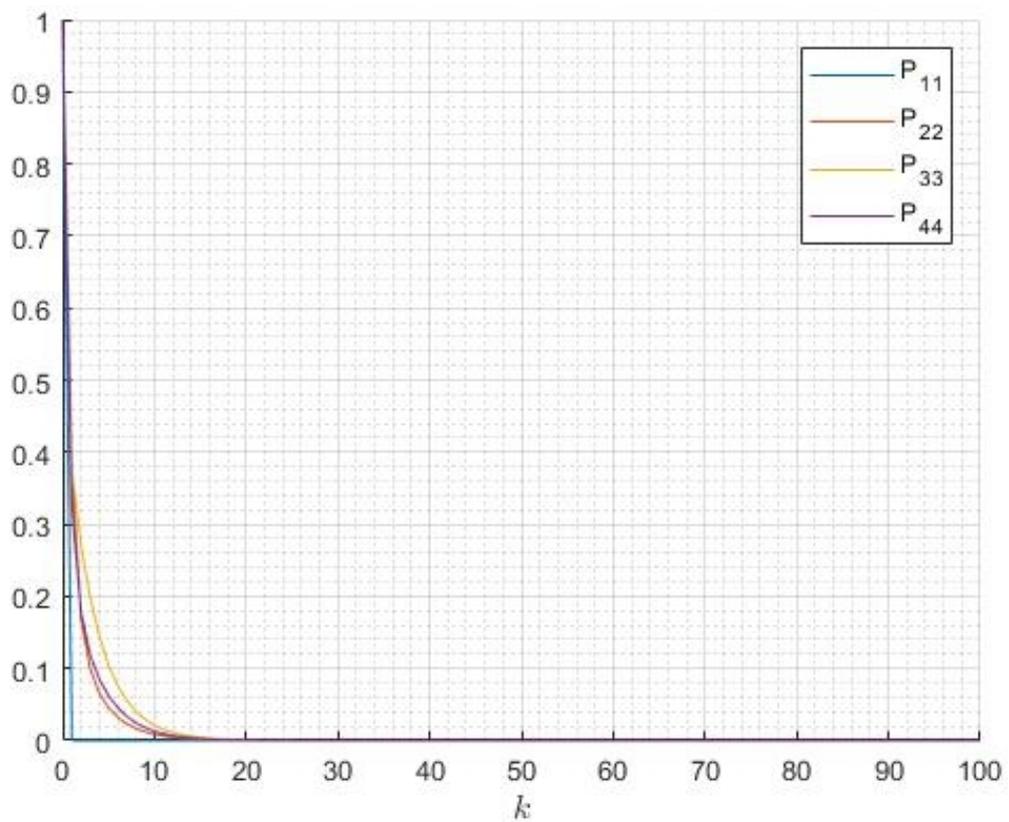


Рис. 4.28. Ковариации ошибок оценивания вектора состояния

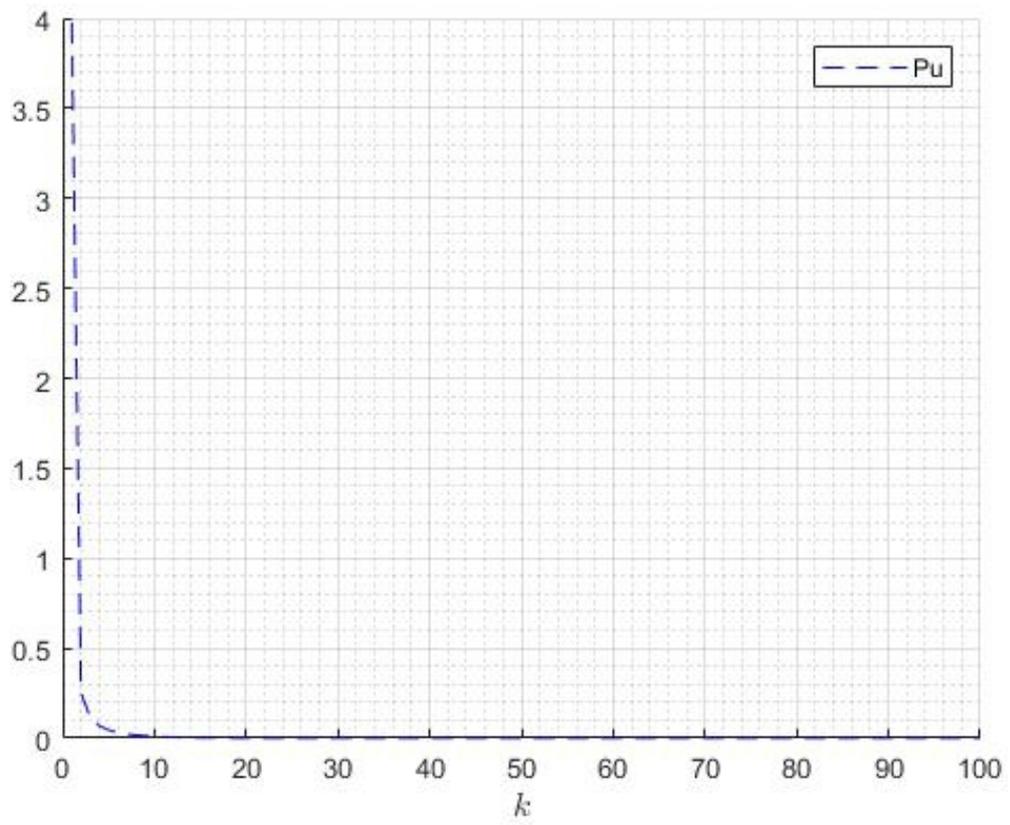


Рис. 4.29. Ковариации ошибок оценивания вектора входных воздействий

### 4.3. Заключение и выводы к главе 4

В данной главе был рассмотрен программный комплекс, предназначенный для компьютерного моделирования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса.

Программный комплекс реализован на языке MATLAB в виде набора скриптов и функций, позволяющих решать следующие задачи:

- 1) описание исходной модели,
- 2) построение конечно-разностной сетки и дискретизация непрерывной модели,
- 3) моделирование экспериментальных данных,
- 4) идентификация коэффициентов конвекции и диффузии,
- 5) идентификация граничных условий,
- 6) визуализация полученных результатов,
- 7) проведение численных экспериментов.

В разделе 4.1 описана структура программного комплекса, состав его компонентов, основные структуры данных и функции, приведены блок-схемы процессов параметрической идентификации.

В разделе 4.2 рассмотрены примеры решения задач параметрической идентификации при помощи программного комплекса.

## Заключение

В диссертации разработаны новые методы параметрической идентификации математических моделей процессов конвективно-диффузионного переноса на основе рекуррентных алгоритмов дискретной фильтрации.

В работе получены новые научные **результаты**:

1. Построены дискретные линейные стохастические модели конвективно-диффузионного переноса в пространстве состояний. Проанализированы свойства полной наблюдаемости и управляемости полученных моделей.

2. Сформулированы алгоритмы вычисления критериев идентификации неизвестных коэффициентов уравнения конвекции-диффузии на основе стандартного фильтра Калмана и его SVD-модификации. Получены формулы для вычисления градиента критерия идентификации на основе стандартного фильтра Калмана.

3. Сформулированы алгоритмы численной идентификации граничных условий на основе алгоритмов Гиллейнса — Де-Мора. Приведены примеры применения данных алгоритмов для случаев, когда оба граничных условия неизвестны и когда одно из граничных условий известно. Сформулирована квадратно-корневая модификация алгоритма Гиллейнса — Де-Мора в ковариационной форме и доказана ее алгебраическая эквивалентность исходному алгоритму.

4. Разработан комплекс программ на языке MATLAB для компьютерного моделирования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса.

**Выводы** из полученных в диссертации результатов заключаются в следующем:

1. Переход от непрерывных моделей конвективно-диффузионного переноса к моделям, описываемым дискретными линейными стохастическими системами в пространстве состояний, позволяет решать задачи параметрической идентификации моделей конвективно-диффузионного переноса в условиях зашумленных измерений.

2. При программной реализации методов параметрической идентифика-

ции предпочтительнее использовать алгоритмы на основе численно устойчивых к ошибкам машинного округления рекуррентных алгоритмов дискретной фильтрации.

3. Результаты численного моделирования подтверждают работоспособность предложенных подходов.

Диссертационное исследование может получить своё **дальнейшее развитие** в следующих направлениях:

1. Полученные в работе результаты и разработанные подходы могут быть применены к другим моделям тепломассопереноса, например: конвекции-реакции-диффузии, двумерным моделям конвекции-диффузии и др.

2. Дискретизация исходной задачи может быть выполнена с использованием альтернативных конечно-разностных схем.

3. Могут быть рассмотрены задачи параметрической идентификации с переменными коэффициентами и с другими наборами граничных условий.

## Список литературы

1. Кожин, И. А. Теория тепломассообмена. Учебник для вузов / И. А. Кожин, С. И. Исаев, А. И. Леонтьев. — 3-е изд. изд. — М. : МГТУ им. Н. Э. Баумана, 2018. — 624 с.
2. Дерюгин, В. В. Тепломассообмен. Учебное пособие / В. В. Дерюгин, В. М. Уляшева, В. Ф. Васильев. — СПб. : Лань, 2018. — 240 с.
3. Фарлоу, С. Д. Уравнения с частными производными для научных работников и инженеров / С. Д. Фарлоу. — М. : Мир, 1985. — 383 р.
4. Денисов, А. М. Введение в теорию обратных задач: учеб. пособие / А. М. Денисов. — М. : Изд-во МГУ, 1994. — 208 с.
5. Самарский, А. А. Численные методы решения обратных задач математической физики / А. А. Самарский, П. Н. Вабищевич. — М. : ЛКИ, 2009.
6. Самарский, А. А. Теория разностных схем / А. А. Самарский. — М. : Наука, 1977.
7. Алифанов, О. В. Обратные задачи теплообмена / О. В. Алифанов. — М. : Машиностроение, 1988.
8. Тихонов, А. Н. Методы решения некорректных задач / А. Н. Тихонов, В. Я. Арсенин. — М. : Наука, 1986.
9. Кабанихин, С. И. Обратные и некорректные задачи / С. И. Кабанихин. — Новосибирск : СНИ, 2009.
10. Daouas, N. An extended version of the discrete Kalman filter applied to a nonlinear inverse heat conduction problem / N. Daouas, M. S. Radhouani // International Journal of Thermal Sciences. — 2000. — Vol. 39. — P. 191–212. — DOI: 10.1016/S1290-0729(00)00239-8
11. Пилипенко, Н. В. Применение фильтра Калмана в нестационарной теплотметрии. Учебное пособие / Н. В. Пилипенко. — СПб. : Университет ИТМО, 2017. — 36 с.
12. Bierman, G. J. Numerical comparison of discrete Kalman filter algorithms: Orbit determination case study / Gerald J. Bierman, Catherine L. Thornton // 1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive

- Processes. — 1976. — P. 859–872. — DOI: 10.1109/CDC.1976.267848
13. Verhaegen, M. Numerical aspects of different Kalman filter implementations / M. Verhaegen, P. Van Dooren // IEEE Transactions on Automatic Control. — 1986. — Vol. 31, no. 10. — P. 907–917. — DOI: 10.1109/TAC.1986.1104128
  14. Grewal, M. S. Kalman Filtering: Theory and Practice Using Matlab / M. S. Grewal, A. P. Andrews. — New Jersey : Prentice Hall, 2001. — 401 p.
  15. Симбирский, Д. В. Температурная диагностика двигателей / Д. В. Симбирский. — Киев : Техника, 1976. — 208 с.
  16. Hodge, J. K. Unsteady heat transfer coefficient estimation for long duration / James K. Hodge, Alice J. Chen, James R. Hayes // Journal of Thermophysics and Heat Transfer. — 1988. — Т. 2, № 3. — С. 218–226. — DOI: 10.2514/3.91
  17. Мацевитый, Ю. М. Идентификация параметров теплообмена методом оптимальной динамической фильтрации / Ю. М. Мацевитый, А. В. Мултановский // Теплофизика высоких температур. — 1979. — Т. 17(5). — С. 1053–1060.
  18. Карпов, А. А. Восстановление нестационарных тепловых потоков по экспериментальным данным / А. А. Карпов, Т. А. Тихонова // Матем. моделирование. — 2000. — Т. 12(5). — С. 101–106.
  19. Симбирский, Д. Ф. Применение цифрового фильтра Калмана для параметрической идентификации высокотемпературного термопреобразователя / Д. Ф. Симбирский, В. К. Лантрат // Автомобиль и электроника. Современные технологии. — 2017. — № 11. — С. 68–75.
  20. Daouas, N. A new approach of the Kalman filter using future temperature measurements for nonlinear inverse heat conduction problems / N. Daouas, M. S. Radhouani // Numerical Heat Transfer Fundamentals. — 2004. — Vol. 45, no. 6. — P. 565–585. — DOI: 10.1080/10407790490430598
  21. Identification of a position and time dependent heat flux by using the Kalman filter and improved lumped analysis in heat conduction / C. C. Pacheco, H. R. B. Orlando, M. J. Colaco, G. S. Dulikravich // Proceedings of the ICCM2014, 28th-30th July 2014. — Cambridge, England, 2014. — P. 801–809.
  22. Solution of inverse radiation-conduction problems using a Kalman filter coupled with the recursive least-square estimator / S. Wen, H. Qi, Y. Ren, et al //

- International Journal of Heat and Mass Transfer. — 2017. — Vol. 111, no. C. — P. 582–592. — DOI: 10.1016/j.ijheatmasstransfer.2017.04.017
23. Параметрическая идентификация дифференциально-разностных моделей теплопереноса в одномерных телах на основе алгоритмов фильтра Калмана / Н. В. Пилипенко, Ю. П. Заричняк, В. А. Иванов, А. М. Халявин // Научно-технический вестник информационных технологий, механики и оптики. — 2020. — Т. 20, № 4(128). — С. 584–588. — DOI: 10.17586/2226-1494-2020-20-4-584-588
  24. Pilipenko, N. Parametric identification of differential-difference heat transfer models in non-stationary thermal measurements / N. Pilipenko // Heat Transfer Research. — 2008. — Vol. 39, no. 4. — P. 311–315.
  25. Пилипенко, Н. В. Методы параметрической идентификации в нестационарной теплотометрии. Учебное пособие / Н. В. Пилипенко. — СПб. : Университет ИТМО, 2016. — 79 с.
  26. Orlande, H. R. B. Application of Bayesian filters to heat conduction problem / H. R. B. Orlande, G. S. Dulikravich, M. J. Colaço // EngOpt 2008 – International Conference on Eng. Optimization, (ed: Herskovitz), June 1-5. — Rio de Janeiro, Brazil, 2008.
  27. State estimation problems in heat transfer / H. R. B. Orlande, M. J. Colaço, G. S. Dulikravich [et al.] // International Journal for Uncertainty Quantification. — 2012. — Vol. 2, no. 3. — P. 239–258. — DOI: 10.1615/Int.J.UncertaintyQuantification.2012003582
  28. A comparison of extended Kalman filter, ultrasound time-of-flight measurement models for heating source localization / M. R. Myers, A. B. Jorge, M. J. Mutton, D. J. Walker // Inverse Problems in Science and Engineering. — 2012. — Vol. 20, no. 7. — P. 991–1016. — DOI: 10.1080/17415977.2012.669272
  29. Vianna, F. Prediction of the temperature field in pipelines with Bayesian filters and non-intrusive measurements / F. Vianna, H. Orlande, G. Dulikravich // In Proceedings of the 20th International Congress of Mechanical Engineering. — Gramado, RS, Brazil, 2009.
  30. Vianna, F. Temperature field prediction of a multilayered composite pipeline

- based on the particle filter method / F. Vianna, H. Orlande, G. Dulikravich // Proceedings of the 14th International Heat Transfer Conference. — 2010.
31. State estimation in bioheat transfer: a comparison of particle filter algorithms / B. Lamien, L. A. B. Varon, H. R. B. Orlande, G. E. Elicabe // International Journal of Numerical Methods for Heat & Fluid Flow. — 2017. — Vol. 27, no. 3. — P. 615–638. — DOI: 10.1108/hff-03-2016-0118
  32. Prediction of the coupled heat radiation and conduction parameters and boundary condition using the unscented Kalman filter / S. Wen, H. Qi, Y. Ren, et al // Sci. China Technol. Sci. — 2020. — Vol. 63. — P. 422–433. — DOI: 10.1007/s11431-019-1437-8
  33. Верификация процесса конвективной диффузии на основе анализа многомерных временных рядов / М. Г. Матвеев, Е. А. Сирота, М. Е. Семенов, А. В. Копытин // Труды XIX Международной конференции «Аналитика и управление данными в областях с интенсивным использованием данных» (DAMDID/ RCDL'2017), Москва, Россия, 10–13 октября 2017 года. — Москва : Федеральный исследовательский центр "Информатика и управление" Российской академии наук, 2017. — С. 354–358.
  34. Матвеев, М. Г. Комбинированный метод идентификации параметров распределенной динамической модели / М. Г. Матвеев, А. В. Копытин, Е. А. Сирота // Сборник трудов IV международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2018), Самара, Россия, 24–27 апреля, 2018. — Самара : Новая техника, 2018. — С. 1651–1656.
  35. Копытин, А. В. Применение расширенного фильтра Калмана для идентификации параметров распределенной динамической системы / А. В. Копытин, Е. А. Копытина, М. Г. Матвеев // Вестник ВГУ. Серия: Системный анализ и информационные технологии. — 2018. — № 3. — С. 44–50. — DOI: 10.17308/sait.2018.3/1229
  36. Фомин, В. Н. Рекуррентное оценивание и адаптивная фильтрация / В. Н. Фомин. — М. : Наука, 1984. — 288 с.
  37. Metaheuristic algorithms for identification of the convection velocity in the convection-diffusion transport model / A. V. Tsyganov, Yu. V. Tsyganova,

- A. N. Kuvshinova, H. R. Tapia Garza // Proceedings of the II International Scientific and Practical Conference “Fuzzy Technologies in the Industry — FTI 2018”, Ulyanovsk, Russia, October 23–25, 2018. — CEUR-WS.org, 2018. — P. 188–196. — URL: <http://ceur-ws.org/Vol-2258/paper24.pdf>.
38. Цыганов, А. В. Динамическая идентификация граничных условий в модели конвективно-диффузионного переноса в условиях зашумленных измерений / А. В. Цыганов, Ю. В. Цыганова, А. Н. Кувшинова // Сборник трудов V международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2019), Самара, Россия, 21–24 мая, 2019. — Т. 3. — Самара : Новая техника, 2019. — С. 169–177.
39. Tsyganov, A. V. Dynamic identification of boundary conditions for convection-diffusion transport model subject to noisy measurements / A. V. Tsyganov, Yu. V. Tsyganova, A. N. Kuvshinova // Journal of Physics: Conference Series. — 2019. — Vol. 1368. — P. 042029. — DOI: 10.1088/1742-6596/1368/4/042029
40. Кувшинова, А. Н. Динамическая идентификация смешанных граничных условий в модели конвективно-диффузионного переноса в условиях зашумленных измерений / А. Н. Кувшинова // Журнал Средневолжского Математического Общества. — 2019. — Т. 21, № 4. — С. 469–479. — DOI: 10.15507/2079-6900.21.201904.469-479
41. Maybeck, P. S. Stochastic models, estimation, and control. Volume 1 / P. S. Maybeck. — New York : Academic Press, 1979. — 445 p.
42. Кувшинова, А. Н. Анализ наблюдаемости дискретной линейной стохастической модели конвективно-диффузионного переноса / А. Н. Кувшинова // Информационные технологии в моделировании и управлении: подходы, методы, решения: Сборник научных статей II Всероссийской научной конференции с международным участием: 22–24 апреля 2019 г. В двух частях. Ч. 1. — Тольятти : Издатель Качалин Александр Васильевич, 2019. — С. 252–257.
43. Кувшинова, А. Н. Анализ дискретной линейной стохастической модели конвективно-диффузионного переноса / А. Н. Кувшинова // Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. — 2019. — № 1. — С. 65–69.

44. Семушин, И. В. Детерминистские модели динамических систем. Методическое пособие / И. В. Семушин, Ю. В. Цыганова. — Ульяновск : УлГТУ, 2007. — 75 с.
45. Дьяконов, В. П. Maple 10/11/12/13/14 в математических расчетах / В. П. Дьяконов. — М. : ДМК Пресс, 2018. — 800 с.
46. Активная параметрическая идентификация стохастических линейных систем: монография / В. И. Денисов, Чубич В. М., Черникова О. С., Бобылева Д. И. — Новосибирск : Изд-во НГТУ, 2009. — 192 с.
47. Gibbs, V. P. Advanced Kalman filtering, least-squares and modeling: a practical handbook / V. P. Gibbs. — Hoboken, New Jersey : John Wiley & Sons, Inc., 2011. — 632 p. — ISBN: 978-0-470-52970-6. — DOI: 10.1002/9780470890042
48. Aström, K. J. Maximum likelihood and prediction error methods / K. J. Aström // Automatica. — 1980. — no. 16(5). — P. 551–574.
49. Алгоритм численной идентификации параметров в модели конвективно-диффузионного переноса / А. Н. Кувшинова, А. В. Цыганов, Ю. В. Цыганова, У. Р. Тапия Гарса // Информационные технологии и нанотехнологии (ИТНТ-2020). Сборник трудов по материалам VI Международной конференции и молодежной школы (г. Самара, 26–29 мая): в 4 т. / Самар. нац.-исслед. ун-т им. С. П. Королева (Самар. ун-т), Ин-т систем. обраб. изобр. РАН-фил. ФНИЦ "Кристаллография и фотоника"РАН; [под ред. В. А. Соболева]. — Т. 3. Математическое моделирование физико-технических процессов и систем. — Самара : Изд-во Самар. ун-та, 2020. — С. 825–832.
50. Parameter identification algorithm for convection-diffusion transport model / A. N. Kuvshinova, A. V. Tsyganov, Yu. V. Tsyganova, H. R. Tapia Garza // Journal of Physics: Conference Series. — 2021. — Vol. 1745. — P. 012110.
51. Кувшинова, А. Н. Адаптивное оценивание дискретной модели конвективно-диффузионного переноса на основе численно эффективных алгоритмов Калмановской фильтрации / А. Н. Кувшинова // Материалы Международного молодежного научного форума «ЛОМОНОСОВ-2021» / Отв. ред. И.А. Алешковский, А.В. Андриянов, Е.А. Антипов, Е.И. Зимакова. [Электронный ресурс]. — Москва : МАКС Пресс, 2021.

52. Кувшинова, А. Н. Математическое моделирование процесса параметрической идентификации моделей конвективно-диффузионного переноса с применением SVD-фильтра Калмана / А. Н. Кувшинова, А. В. Цыганов, Ю. В. Цыганова // Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки. — 2021. — Т. 25, № 4. — С. 716–737. — DOI: 10.14498/vsgtu1876
53. Цыганова, Ю. В. Имитационная нормализация в задаче идентификации параметров стохастической линейной системы / Ю. В. Цыганова, А. В. Цыганов // Стохастическая оптимизация в информатике. — 2010. — Т. 6, № 1. — С. 147–159.
54. Цыганов, А. В. Параллельные гибридные алгоритмы для задачи параметрической идентификации в стохастических линейных системах / А. В. Цыганов, О. И. Булычев, Ю. В. Цыганова // Вектор науки Тольяттинского государственного университета. — 2011. — № 3(17). — С. 45–49.
55. Ivanov, D. V. Genetic algorithm of structural and parametric identification of Gegenbauer autoregressive with noise on output / D. V. Ivanov, V. V. Engelhardt, I. L. Sandler // Proceedings of the 8th International Congress of Information and Communication Technology (ICICT 2018), Xiamen, January 27–28, 2018. — 2018. — P. 619–625.
56. Метаэвристические алгоритмы в задаче идентификации параметров математической модели движущегося объекта / А. В. Цыганов, И. В. Семушин, Ю. В. Цыганова [и др.] // Автоматизация процессов управления. — 2017. — № 1 (47). — С. 16–23.
57. Пантелеев, А. В. Метаэвристические алгоритмы глобальной оптимизации / А. В. Пантелеев, Д. В. Скавинская. — М. : Вузовская книга, 2019. — 332 с.
58. Hromkovič, J. Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics / J. Hromkovič. Texts in Theoretical Computer Science. An EATCS Series. — 2nd edition. — Springer-Verlag, 2004. — XIV, 538 p. — ISBN: 978-3-540-44134-2. — DOI: 10.1007/978-3-662-05269-3
59. Дьяконов, В. П. MATLAB. Полный самоучитель / В. П. Дьяконов. — М. : ДМК Пресс, 2012. — 768 с.

60. Васильев, Ф. П. Численные методы решения экстремальных задач : [Учеб. пособие для вузов по спец. "Прикл. математика"] / Ф. П. Васильев. — 2-е изд., перераб. и доп. изд. — М. : Наука, 1988. — 549 с.
61. Цыганова, Ю. В. О методах реализации UD-фильтра / Ю. В. Цыганова // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. — 2013. — № 3. — С. 84–104.
62. Цыганова, Ю. В. Об одном подходе к построению адаптивного UD-фильтра для параметрической идентификации LQG систем / Ю. В. Цыганова // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014. Москва, 16–19 июня 2014 г. — Москва : ИПУ им. В. А. Трапезникова РАН, 2014. — С. 2741–2751.
63. Grewal, M. S. Kalman filtering: Theory and Practice Using MATLAB / M. S. Grewal, A. P. Andrews. — 4th edition. — John Wiley & Sons, Inc., 2015. — ISBN: 978-1-118-85121-0.
64. Tsyganova, J. V. Some new array information formulations of the UD-based Kalman filter / J. V. Tsyganova, M. V. Kulikova, A. V. Tsyganov // Proceedings of the 18th European Control Conference (ECC), Napoli, Italy, June 25–28, 2019. — IEEE, 2019. — P. 1872–1877. — DOI: 10.23919/ECC.2019.8795648
65. Tsyganov, A. V. UD-based linear filtering for discrete-time systems with multiplicative and additive noises / A. V. Tsyganov, J. V. Tsyganova, T. N. Kureneva // Proceedings of the European Control Conference (ECC), Saint Petersburg, Russia, May 12–15, 2020. — IEEE, 2020. — P. 1389–1394. — DOI: 10.23919/ECC51009.2020.9143804
66. Bjorck, A. Numerical methods in matrix computations / A. Bjorck. — Springer International Publishing, 2015. — XVI, 800 p.
67. Цыганова, Ю. В. О современных ортогонализированных алгоритмах оптимальной дискретной фильтрации / Ю. В. Цыганова, М. В. Куликова // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». — 2018. — Т. 11, № 4. — С. 5–30.
68. Kulikova, M. V. SVD-based state and parameter estimation approach for generalized Kalman filtering with application to garch-in-mean estimation / M. V. Ku-

- likova, J.V. Tsyganova, G. Yu. Kulikov // *Journal of Computational and Applied Mathematics*. — 2021. — Vol. 387. — P. 112487.
69. Oshman, Y. Square root filtering via covariance and information eigenfactors / Y. Oshman, I. Y. Bar-Itzhack // *Automatica*. — 1986. — Vol. 22, no. 5. — P. 599–604.
70. Oshman, Y. Square root information filtering using the covariance spectral decomposition / Y. Oshman // *Proceedings of the 27th IEEE Conf. on Decision and Control*. — Vol. 1. — 1988. — P. 382–387.
71. Oshman, Y. Maximum likelihood state and parameter estimation via derivatives of the V-Lambda filter / Y. Oshman // *Journal of Guidance, Control, and Dynamics*. — 1992. — Vol. 15, no. 3. — P. 717–726.
72. Wang, L. Kalman filter algorithm based on singular value decomposition / L. Wang, J. Libert, P. Manneback // *Proceedings the 31st Conference on Decision and Control*. — Vol. 1. — 1992. — P. 1224–1229.
73. A SVD-based extended Kalman filter and applications to aircraft flight state and parameter estimation / Y. Zhang, G. Dai, H. Zhang, Q. Li // *Proceedings of 1994 American Control Conference – AC'94*. — Vol. 2. — 1994. — P. 1809–1813.
74. Kulikova, M. V. Improved discrete-time Kalman filtering within singular value decomposition / M. V. Kulikova, Y. V. Tsyganova // *IET Control Theory & Applications*. — 2017. — Vol. 11(15). — P. 2412–2418.
75. Tsyganova, Y. V. SVD-based Kalman filter derivative computation / Y. V. Tsyganova, M. V. Kulikova // *IEEE Transactions on Automatic Control*. — 2017. — Vol. 62, no. 9. — P. 4869–4875.
76. Singular value decomposition in embedded systems based on ARM Cortex-M architecture / M. Alessandrini, G. Biagetti, L. Falaschetti [et al.] // *Electronics*. — 2021. — Vol. 10. — P. 34.
77. Кувшинова, А. Н. Программный комплекс для моделирования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса v1.0 / А. Н. Кувшинова, А. В. Цыганов // РОСПАТЕНТ. Свидетельство о государственной регистрации программы для ЭВМ № 2021664064 от 17.08.2021 г.

78. Friedland, B. Treatment of bias in recursive filtering / B. Friedland // *IEEE Transactions Control*. — 1969. — Vol. 14. — P. 359–367.
79. Gillijns, S. Unbiased minimum-variance input and state estimation for linear discrete-time systems / S. Gillijns, B. D. Moor // *Automatica*. — 2007. — Vol. 43. — P. 111–116.
80. Kitanidis, P. K. Unbiased minimum-variance linear state estimation / P. K. Kitanidis // *Automatica*. — 1987. — Vol. 23(6). — P. 775–778.
81. Darouach, M. Unbiased minimum variance estimation for systems with unknown exogenous inputs / M. Darouach, M. Zasadzinski // *Automatica*. — 1997. — no. 33(4). — P. 717–719.
82. Gillijns, S. Joint state and boundary condition estimation in linear data assimilation using basis function expansion / S. Gillijns, B. D. Moor // *Proceedings of the 26th IASTED International Conference on Modelling, Identification, and Control (MIC 2007)*. — Innsbruck, Austria, 2007. — P. 458–463.
83. Кувшинова, А. Н. Численная идентификация граничных условий модели конвективно-диффузионного переноса на основе алгоритма Гиллийнса-Мура / А. Н. Кувшинова // *Дифференциальные уравнения, математическое моделирование и вычислительные алгоритмы : сборник материалов международной конференции, Белгород, 25–29 октября 2021 г. / под ред. В. Б. Васильева, И. С. Ломова*. — Белгород : ИД «БелГУ» НИУ «БелГУ», 2021. — С. 143–145.
84. Кувшинова, А. Н. Об алгоритме дискретной фильтрации для линейных стохастических систем с неизвестными входными воздействиями / А. Н. Кувшинова // *Интегрированные системы управления: сб. науч. тр. науч.-технич. конф., Ульяновск, 18–19 мая 2021 г. / отв. за вып. А. Л. Савкин*. — Ульяновск : ФНПЦ АО «НПО «Марс», 2021. — С. 90–95.
85. Кувшинова, А. Н. О квадратно-корневой модификации алгоритма Гиллейнса–Де-Мора / А. Н. Кувшинова, Д. В. Галушкина // *Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн*. — 2022. — № 1. (в печати)
86. PooGyeon, P. New square-root algorithms for Kalman filtering / Park PooGyeon,

- T. Kailath // IEEE Transactions on Automatic Control. — 1995. — Vol. 40, no. 5. — P. 895–899. — DOI: 10.1109/9.384225
87. Gillijns, S. Information, covariance and square-root filtering in the presence of unknown inputs / Steven Gillijns, Niels Haverbeke, Bart De Moor // 2007 European Control Conference (ECC). — 2007. — P. 2213–2217. — DOI: 10.23919/ECC.2007.7068514
88. Цыганова, Ю. В. Динамическая идентификация граничных условий в модели конвективно-диффузионного переноса v1.0 / Ю. В. Цыганова, А. В. Цыганов, А. Н. Кувшинова // РОСПАТЕНТ. Свидетельство о государственной регистрации программы для ЭВМ № 2019616886 от 30.05.2019 г.
89. Кувшинова, А. Н. Программа для идентификации параметров модели конвективно-диффузионного переноса v1.0 / А. Н. Кувшинова, А. В. Цыганов, Ю. В. Цыганова // РОСПАТЕНТ. Свидетельство о государственной регистрации программы для ЭВМ № 2020619321 от 17.08.2020 г.
90. Кувшинова, А. Н. Программный комплекс для компьютерного моделирования процессов параметрической идентификации математических моделей конвективно-диффузионного переноса / А. Н. Кувшинова, А. В. Цыганов // Программные продукты и системы. — 2021. — Т. 34, № 4. — С. 639–648. — DOI: 10.15827/0236-235X.136.639-648

## Список иллюстративного материала

1.1	Решение задачи . . . . .	18
1.2	Зашумленные измерения (модель 1) . . . . .	20
1.3	Зашумленные измерения (модель 2) . . . . .	20
1.4	Зашумленные измерения (модель 3) . . . . .	21
1.5	Решение задачи . . . . .	24
1.6	Зашумленные измерения . . . . .	26
2.1	Блок-схема алгоритма 2.1 . . . . .	35
2.2	Блок-схема алгоритма 2.2 . . . . .	37
2.3	Точное решение . . . . .	40
2.4	Критерий идентификации для дискретной модели (2.11) . . . . .	42
2.5	Критерий идентификации для дискретной модели (2.12) . . . . .	42
2.6	Блок-схема алгоритма 2.3 . . . . .	51
2.7	Блок-схема алгоритма 2.4 . . . . .	53
2.8	Графики решения задачи с граничными условиями (2.30) (a) и (2.31) (b) . . . . .	55
2.9	Графики измерений для задачи с граничными условиями (2.30) (a) и (2.31) (b) . . . . .	57
3.1	Блок-схема алгоритма 3.1 . . . . .	67
3.2	Блок-схема алгоритма 3.2 . . . . .	69
3.3	Решение задачи . . . . .	72
3.4	Зашумленные измерения . . . . .	72
3.5	Оценка $f(t)$ . . . . .	73
3.6	Оценка $g(t)$ . . . . .	73
3.7	Оценка решения . . . . .	74
3.8	Решение задачи . . . . .	76
3.9	Зашумленные измерения . . . . .	77
3.10	Оценка $f(t)$ . . . . .	77
3.11	Оценка $g(t)$ . . . . .	78
3.12	Оценка решения . . . . .	78
3.13	Решение задачи . . . . .	80
3.14	Зашумленные измерения . . . . .	81
3.15	Оценка $f(t)$ . . . . .	81
3.16	Оценка решения . . . . .	82
3.17	Дисперсия ошибки оценивания граничного условия . . . . .	82
3.18	Блок-схема алгоритма 3.3 . . . . .	85
4.1	Структура программного комплекса . . . . .	90

4.2	Решение задачи (solution) . . . . .	97
4.3	Зашумленные измерения ( $Z$ ) . . . . .	97
4.4	Процесс идентификации граничных условий . . . . .	98
4.5	Оценка решения (estSol) . . . . .	100
4.6	Функция $f(t)$ и ее оценка (estSol(:, 1)) . . . . .	100
4.7	Функция $g(t)$ и её оценка . . . . .	101
4.8	Ковариации ошибок оценивания граничных значений (вектора $u_k$ ) . . . . .	101
4.9	Процесс идентификации коэффициентов конвекции и диффузии . . . . .	102
4.10	Начальное условие . . . . .	109
4.11	Левое граничное условие . . . . .	109
4.12	Правое граничное условие . . . . .	110
4.13	Решение задачи . . . . .	110
4.14	Зашумленные измерения . . . . .	111
4.15	Решение задачи . . . . .	112
4.16	Зашумленные измерения . . . . .	113
4.17	Левое граничное условие и его оценка . . . . .	113
4.18	Правое граничное условие и его оценка . . . . .	114
4.19	Оценка решения . . . . .	114
4.20	Ковариации ошибок оценивания вектора состояния . . . . .	115
4.21	Ковариации ошибок оценивания вектора входных воздействий . . . . .	115
4.22	Критерий идентификации на основе стандартного фильтра Калмана . . . . .	119
4.23	Решение задачи . . . . .	123
4.24	Зашумленные измерения . . . . .	123
4.25	Левое граничное условие и его оценка . . . . .	124
4.26	Правое граничное условие . . . . .	124
4.27	Оценка решения . . . . .	125
4.28	Ковариации ошибок оценивания вектора состояния . . . . .	125
4.29	Ковариации ошибок оценивания вектора входных воздействий . . . . .	126

## Список таблиц

2.1	Настройки алгоритмов . . . . .	41
2.2	Результаты экспериментов для модели (2.11) . . . . .	43
2.3	Результаты экспериментов для модели (2.12) . . . . .	43
2.4	Результаты экспериментов для модели (2.11) . . . . .	47
2.5	Средние значения параметров и ошибки идентификации для $R_k = R_0$ и граничных условий (2.30) . . . . .	58
2.6	Средние значения параметров и ошибки идентификации для $R_k = R_0$ и граничных условий (2.31) . . . . .	58
2.7	Средние значения параметров и ошибки идентификации для $R_k = R_1$ и граничных условий (2.30) . . . . .	59
2.8	Средние значения параметров и ошибки идентификации для $R_k = R_2$ и граничных условий (2.30) . . . . .	59
2.9	Средние значения параметров и ошибки идентификации для $R_k = R_3$ и граничных условий (2.30) . . . . .	60
2.10	Средние значения параметров и ошибки идентификации для $R_k = R_1$ и граничных условий (2.31) . . . . .	60
2.11	Средние значения параметров и ошибки идентификации для $R_k = R_2$ и граничных условий (2.31) . . . . .	61
2.12	Средние значения параметров и ошибки идентификации для $R_k = R_3$ и граничных условий (2.31) . . . . .	61
3.1	Ошибки оценивания . . . . .	71
3.2	Ошибки оценивания . . . . .	76
3.3	Ошибки оценивания . . . . .	88
4.1	Компоненты программного комплекса . . . . .	91

## Список алгоритмов

2.1	Фильтр Калмана в ковариационной форме (СКФ) . . . . .	34
2.2	Вычисление критерия идентификации на основе стандартного фильтра Калмана .	36
2.3	SVD-модификация фильтра Калмана . . . . .	50
2.4	Вычисление критерия идентификации на основе SVD-модификации фильтра Кал- мана . . . . .	52
3.1	Алгоритм Гиллейнса – Де-Мора (вариант 1) . . . . .	66
3.2	Алгоритм Гиллейнса – Де-Мора (вариант 2) . . . . .	68
3.3	Квадратно-корневой ковариационный алгоритм оценивания состояния и входных воздействий . . . . .	84
B.1	Метод имитации отжига (SA) . . . . .	150
B.2	Генетический алгоритм (GA) . . . . .	151

## Приложение А

# Скрипты Maple для автоматической проверки свойства полной наблюдаемости

**Скрипт А.1.** Проверка свойства полной наблюдаемости для граничных условий первого рода

```
interface(rtablesize = infinity) :
with(LinearAlgebra) :
n := 4
```

$$4 \quad (1)$$

```
HL := [1]
```

$$[1] \quad (2)$$

```
H := Matrix([seq([Row(Matrix(n, shape = identity), HL[i]), i = 1 .. nops(HL))])])
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

```
F := BandMatrix([a1, a2, a3], 1, n, storage = rectangular)
```

$$\begin{bmatrix} a_2 & a_3 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 \\ 0 & a_1 & a_2 & a_3 \\ 0 & 0 & a_1 & a_2 \end{bmatrix} \quad (4)$$

```
M_DTI := simplify(Matrix([seq([H.F^{i-1}], i = 1 .. n)]))
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ a_2 & a_3 & 0 & 0 \\ a_1 a_3 + a_2^2 & 2 a_2 a_3 & a_3^2 & 0 \\ 3 a_1 a_2 a_3 + a_2^3 & 2 a_1 a_3^2 + 3 a_2^2 a_3 & 3 a_2 a_3^2 & a_3^3 \end{bmatrix} \quad (5)$$

```
M := GaussianElimination(M_DTI)
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a_3 & 0 & 0 \\ 0 & 0 & a_3^2 & 0 \\ 0 & 0 & 0 & a_3^3 \end{bmatrix} \quad (6)$$

```
Rank(M)
```

$$4 \quad (7)$$

## Скрипт А.2. Проверка свойства полной наблюдаемости для смешанных граничных условий

```
interface (rtabsize = infinity) :
with (LinearAlgebra) :
n := 5
```

$$5 \quad (1)$$

```
HL := [1]
```

$$[1] \quad (2)$$

```
H := Matrix([seq([Row(Matrix(n, shape = identity), HL[i]), i = 1 .. nops(HL))])
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

```
R := Matrix([seq(0, i = 1 .. n - 3), a4*a1, a4*a2, a4*a3]) : C := Matrix([seq([0], i = 1 .. n - 2),
[a3]]) : F1 := BandMatrix([a1, a2, a3], 1, n - 1, storage = rectangular) :
```

```
F := Matrix([[Matrix([F1, C]), [R]])
```

$$\begin{bmatrix} a_2 & a_3 & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 & 0 \\ 0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 \\ 0 & 0 & a_4 a_1 & a_4 a_2 & a_4 a_3 \end{bmatrix} \quad (4)$$

```
M_DTI := simplify(Matrix([seq([H.Fi-1], i = 1 .. n)]))
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ a_2 & a_3 & 0 & 0 & 0 \\ a_1 a_3 + a_2^2 & 2 a_2 a_3 & a_3^2 & 0 & 0 \\ 3 a_1 a_2 a_3 + a_2^3 & 2 a_1 a_3^2 + 3 a_2^2 a_3 & 3 a_2 a_3^2 & a_3^3 & 0 \\ 2 a_1^2 a_3^2 + 6 a_1 a_2^2 a_3 + a_2^4 & 8 a_1 a_2 a_3^2 + 4 a_2^3 a_3 & 3 a_1 a_3^3 + 6 a_2^2 a_3^2 & 4 a_2 a_3^3 & a_3^4 \end{bmatrix} \quad (5)$$

```
M := GaussianElimination(M_DTI)
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & a_3 & 0 & 0 & 0 \\ 0 & 0 & a_3^2 & 0 & 0 \\ 0 & 0 & 0 & a_3^3 & 0 \\ 0 & 0 & 0 & 0 & a_3^4 \end{bmatrix} \quad (6)$$

```
Rank(M)
```

$$5 \quad (7)$$

## Приложение Б

# Скрипты Maple для автоматической проверки свойства полной управляемости

**Скрипт Б.1.** Проверка свойства полной управляемости для граничных условий первого рода

```
interface(rtablesizer = infinity) :
with(LinearAlgebra) :
n := 4
```

$$B := \text{Matrix}([ [a_1, 0], \text{seq}([0, 0], i = 1 .. n - 2), [0, a_3] ]) \quad (1)$$

$$\begin{bmatrix} a_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & a_3 \end{bmatrix} \quad (2)$$

```
F := BandMatrix([a_1, a_2, a_3], 1, n, storage = rectangular)
```

$$\begin{bmatrix} a_2 & a_3 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 \\ 0 & a_1 & a_2 & a_3 \\ 0 & 0 & a_1 & a_2 \end{bmatrix} \quad (3)$$

```
WDTI := simplify(Matrix([seq(Fi-1.B, i = 1 .. n)]))
```

$$\begin{aligned} & [ [a_1, 0, a_2 a_1, 0, (a_1 a_3 + a_2^2) a_1, 0, a_2 (3 a_1 a_3 + a_2^2) a_1, a_3^4], \\ & [0, 0, a_1^2, 0, 2 a_2 a_1^2, a_3^3, a_1^2 (2 a_1 a_3 + 3 a_2^2), 3 a_2 a_3^3], \\ & [0, 0, 0, a_3^2, a_1^3, 2 a_2 a_3^2, 3 a_2 a_1^3, a_3^2 (2 a_1 a_3 + 3 a_2^2)], \\ & [0, a_3, 0, a_2 a_3, 0, (a_1 a_3 + a_2^2) a_3, a_1^4, a_2 (3 a_1 a_3 + a_2^2) a_3] ] \end{aligned} \quad (4)$$

```
W := GaussianElimination(WDTI)
```

$$\begin{aligned} & [ [a_1, 0, a_2 a_1, 0, (a_1 a_3 + a_2^2) a_1, 0, a_2 (3 a_1 a_3 + a_2^2) a_1, a_3^4], \\ & [0, a_3, 0, a_2 a_3, 0, (a_1 a_3 + a_2^2) a_3, a_1^4, a_2 (3 a_1 a_3 + a_2^2) a_3], \\ & [0, 0, a_1^2, 0, 2 a_2 a_1^2, a_3^3, a_1^2 (2 a_1 a_3 + 3 a_2^2), 3 a_2 a_3^3], \\ & [0, 0, 0, a_3^2, a_1^3, 2 a_2 a_3^2, 3 a_2 a_1^3, a_3^2 (2 a_1 a_3 + 3 a_2^2)] ] \end{aligned} \quad (5)$$

```
Rank(W)
```

$$4 \quad (6)$$

## Скрипт Б.2. Проверка свойства полной управляемости для смешанных граничных условий

```
interface(rtablesizе = infinity) :
with(LinearAlgebra) :
n := 5
```

$$B := \text{Matrix}([\text{seq}([0, 0], i = 1 .. n - 2), [0, a_5]]) \quad (1)$$

$$\begin{bmatrix} a_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & a_5 \end{bmatrix} \quad (2)$$

```
R := Matrix([seq(0, i = 1 .. n - 3), a_4*a_1, a_4*a_2, a_4*a_3]) : C := Matrix([seq([0], i = 1 .. n - 2),
[a_3]]) : F_1 := BandMatrix([a_1, a_2, a_3], 1, n - 1, storage = rectangular) :
F := Matrix([[Matrix([F_1, C]), [R]])]
```

$$\begin{bmatrix} a_2 & a_3 & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 & 0 \\ 0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 \\ 0 & 0 & a_4 a_1 & a_4 a_2 & a_4 a_3 \end{bmatrix} \quad (3)$$

```
W_DTI := simplify(Matrix([seq(F^-1.B, i = 1 .. n)]))
```

$$\begin{aligned} & [[a_1, 0, a_2 a_1, 0, (a_1 a_3 + a_2^2) a_1, 0, a_2 (3 a_1 a_3 + a_2^2) a_1, 0, (2 a_1^2 a_3^2 + 6 a_1 a_2^2 a_3 + a_2^4) a_1, \\ & a_3^4 a_5], \\ & [0, 0, a_1^2, 0, 2 a_2 a_1^2, 0, a_1^2 (2 a_1 a_3 + 3 a_2^2), a_3^3 a_5, 4 (2 a_1 a_3 + a_2^2) a_2 a_1^2, a_3^3 (a_3 a_4 \\ & + 3 a_2) a_5], \\ & [0, 0, 0, 0, a_1^3, a_3^2 a_5, 3 a_2 a_1^3, a_3^2 (a_3 a_4 + 2 a_2) a_5, 3 a_1^3 (a_1 a_3 + 2 a_2^2), a_3^2 (a_3^2 a_4^2 \\ & + 3 a_2 a_3 a_4 + 2 a_1 a_3 + 3 a_2^2) a_5], \\ & [0, 0, 0, a_3 a_5, 0, a_3 (a_3 a_4 + a_2) a_5, a_1^4, a_3 (a_3^2 a_4^2 + 2 a_2 a_3 a_4 + a_1 a_3 + a_2^2) a_5, a_1^4 (a_3 a_4 \\ & + 4 a_2), a_3 (a_3^3 a_4^3 + 3 a_2 a_3^2 a_4^2 + 2 a_1 a_3^2 a_4 + 3 a_2^2 a_3 a_4 + 3 a_1 a_2 a_3 + a_2^3) a_5], \\ & [0, a_5, 0, a_4 a_3 a_5, 0, a_4 a_3 (a_3 a_4 + a_2) a_5, a_4 a_1^4, a_3 a_4 (a_3^2 a_4^2 + 2 a_2 a_3 a_4 + a_1 a_3 \\ & + a_2^2) a_5, a_1^4 a_4 (a_3 a_4 + 4 a_2), a_4 a_3 (a_3^3 a_4^3 + 3 a_2 a_3^2 a_4^2 + 2 a_1 a_3^2 a_4 + 3 a_2^2 a_3 a_4 \\ & + 3 a_1 a_2 a_3 + a_2^3) a_5]] \end{aligned} \quad (4)$$

```
W := GaussianElimination(W_DTI)
```

$$\begin{aligned}
& \left[ [a_1, 0, a_2 a_1, 0, (a_1 a_3 + a_2^2) a_1, 0, a_2 (3 a_1 a_3 + a_2^2) a_1, 0, (2 a_1^2 a_3^2 + 6 a_1 a_2^2 a_3 + a_2^4) a_1, \right. \\
& \quad \left. a_3^4 a_5], \right. \\
& \left[ 0, a_5, 0, a_4 a_3 a_5, 0, a_4 a_3 (a_3 a_4 + a_2) a_5, a_4 a_1^4, a_3 a_4 (a_3^2 a_4^2 + 2 a_2 a_3 a_4 + a_1 a_3 \right. \\
& \quad \left. + a_2^2) a_5, a_1^4 a_4 (a_3 a_4 + 4 a_2), a_4 a_3 (a_3^3 a_4^3 + 3 a_2 a_3^2 a_4^2 + 2 a_1 a_3^2 a_4 + 3 a_2^2 a_3 a_4 \right. \\
& \quad \left. + 3 a_1 a_2 a_3 + a_2^3) a_5], \right. \\
& \left[ 0, 0, a_1^2, 0, 2 a_2 a_1^2, 0, a_1^2 (2 a_1 a_3 + 3 a_2^2), a_3^3 a_5, 4 (2 a_1 a_3 + a_2^2) a_2 a_1^2, a_3^3 (a_3 a_4 \right. \\
& \quad \left. + 3 a_2) a_5], \right. \\
& \left[ 0, 0, 0, a_3 a_5, 0, a_3 (a_3 a_4 + a_2) a_5, a_1^4, a_3 (a_3^2 a_4^2 + 2 a_2 a_3 a_4 + a_1 a_3 + a_2^2) a_5, a_1^4 (a_3 a_4 \right. \\
& \quad \left. + 4 a_2), a_3 (a_3^3 a_4^3 + 3 a_2 a_3^2 a_4^2 + 2 a_1 a_3^2 a_4 + 3 a_2^2 a_3 a_4 + 3 a_1 a_2 a_3 + a_2^3) a_5], \right. \\
& \left. \left[ 0, 0, 0, 0, a_1^3, a_3^2 a_5, 3 a_2 a_1^3, a_3^2 (a_3 a_4 + 2 a_2) a_5, 3 a_1^3 (a_1 a_3 + 2 a_2^2), a_3^2 (a_3^2 a_4^2 \right. \right. \\
& \quad \left. \left. + 3 a_2 a_3 a_4 + 2 a_1 a_3 + 3 a_2^2) a_5 \right] \right]
\end{aligned}$$

$\text{Rank}(W)$

5

(6)

## Приложение В

**Метаэвристические алгоритмы оптимизации**

---

**Алгоритм В.1.** Метод имитации отжига (SA)

---

```
1 Solution = InitialSolution()
2 BestSolution = Solution
3 BestCost = Cost(Solution)
4 T = InitialTemperature()
5 n = 0
6 while not StopCondition() do
7   NewSolution = ChooseRandomOf(Neighborhood(Solution))
8   NewCost = Cost(NewSolution)
9   if NewCost < BestCost then
10     BestSolution = NewSolution
11     BestCost = NewCost
12   end if
13   Solution = AcceptWithProbability(Solution, NewSolution, T)
14   n = n + 1
15   T = UpdateTemperature(T, n)
16 end while
```

**Выход:** *BestSolution*

---

---

**Алгоритм В.2.** Генетический алгоритм (GA)

---

```
1 Population = InitialPopulation()
2 for all  $p_i \in Population$  do
3   EvaluateFitness( $p_i$ )
4 end for
5 while not StopCondition() do
6   Parents = SelectParents(Population)
7   Offspring = Crossover(Parents)
8   Offspring = Mutation(Offspring)
9   for all  $p_i \in Offspring$  do
10    EvaluateFitness( $p_i$ )
11  end for
12  Population = UpdatePopulation(Population  $\cup$  Offspring)
13 end while
14 Solution = ChooseBestOf(Population)
```

**Выход:** *Solution*

---

## Приложение Г

## Свидетельство о регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2021664064

**Программный комплекс для моделирования процессов  
параметрической идентификации математических  
моделей конвективно-диффузионного переноса v1.0**

Правообладатель: *Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Ульяновский государственный педагогический  
университет имени И.Н. Ульянова» (RU)*

Авторы: *Кувшинова Анастасия Николаевна (RU), Цыганов  
Андрей Владимирович (RU)*

Заявка № 2021663101

Дата поступления 17 августа 2021 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 30 августа 2021 г.



Руководитель Федеральной службы  
по интеллектуальной собственности

Г.П. Ивлиев