

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ  
НАУКИ ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ ИМ. Г.И.  
МАРЧУКА РОССИЙСКОЙ АКАДЕМИИ НАУК (ИВМ РАН)

На правах рукописи

Христинченко Михаил Юрьевич

**Оптимальные возмущения стационарных и  
периодических решений систем с запаздыванием с  
приложением в математической иммунологии**

Специальность 1.2.2

Математическое моделирование, численные методы и комплексы программ

Диссертация на соискание учёной степени  
кандидата физико-математических наук

Научный руководитель:  
доктор физико-математических наук, доцент  
Нечепуренко Юрий Михайлович

Научный консультант:  
доктор физико-математических наук, доцент  
Бочаров Геннадий Алексеевич

Москва — 2023

## Оглавление

	Стр.
<b>Введение</b> . . . . .	4
<b>Глава 1. Системы с запаздыванием</b> . . . . .	10
1.1 Системы дифференциальных уравнений с запаздыванием . . . . .	10
1.2 Математические модели динамики инфекционных заболеваний . . . . .	11
1.2.1 Модель экспериментальной вирусной инфекции LCMV . . . . .	11
1.2.2 Модель инфекции человека вирусами HBV . . . . .	13
<b>Глава 2. Оптимальные возмущения стационарных решений</b> . . . . .	19
2.1 Вычисление стационарных решений . . . . .	19
2.1.1 Стационарные решения модели LCMV . . . . .	20
2.1.2 Стационарные решения модели HBV . . . . .	20
2.2 Анализ устойчивости стационарного решения . . . . .	23
2.3 Исследование зависимости стационарных решений от параметров . . . . .	26
2.4 Определение оптимального возмущения стационарного решения . . . . .	31
2.5 Базовый алгоритм . . . . .	32
2.6 Метод Ланцоша . . . . .	37
2.7 Метод последовательной максимизации . . . . .	39
2.8 Оценка сложности алгоритмов . . . . .	39
2.8.1 Теоретическая оценка . . . . .	39
2.8.2 Практическая оценка . . . . .	44
<b>Глава 3. Оптимальные возмущения периодических решений</b> . . . . .	54
3.1 Вычисление периодических решений . . . . .	54
3.1.1 Идея метода . . . . .	55
3.1.2 Численная реализация . . . . .	58
3.1.3 Уточнение решения . . . . .	59
3.2 Анализ устойчивости периодического решения . . . . .	61
3.3 Определение оптимального возмущения периодического решения . . . . .	63
3.4 Алгоритмы вычисления оптимального возмущения . . . . .	66

<b>Глава 4. Пакет DEODAN</b> . . . . .	69
4.1 Модельно независимые функции . . . . .	69
4.1.1 Численное интегрирование . . . . .	69
4.1.2 Вычисление и анализ стационарных решений . . . . .	71
4.1.3 Вычисление и анализ периодических решений . . . . .	80
4.2 Модельно зависимые функции . . . . .	84
4.2.1 Задание модели . . . . .	84
4.2.2 Вычисление и анализ стационарных решений . . . . .	91
4.2.3 Вычисление и анализ периодических решений . . . . .	95
4.3 Сравнение с пакетом DDE-BIFTOOL . . . . .	97
<b>Глава 5. Численные эксперименты с моделями инфекционных заболеваний</b> . . . . .	101
5.1 Анализ стационарных решений модели LCMV . . . . .	101
5.2 Оптимальные возмущения стационарных решений модели LCMV	106
5.3 Анализ периодических решений модели LCMV . . . . .	109
5.3.1 Вычисление периодического решения . . . . .	109
5.3.2 Анализ асимптотической орбитальной устойчивости . . . . .	110
5.4 Оптимальные возмущения периодических решений модели LCMV	111
5.5 Зависимость оптимального возмущения от фазы . . . . .	117
5.6 Анализ стационарных решений модели HBV . . . . .	122
5.7 Анализ периодических решений модели HBV . . . . .	128
<b>Заключение</b> . . . . .	131
<b>Список литературы</b> . . . . .	132

## Введение

**Актуальность темы.** Математическое моделирование процессов развития вирусных инфекций в организме человека является быстро развивающейся областью прикладной математики. Изучение закономерностей формирования хронических инфекционных заболеваний и разработка подходов к их лечению были сформулированы как одно из основных направлений исследований в математической иммунологии в основополагающих работах Г.И. Марчука [1; 2]. В результате анализа устойчивости стационарных решений базовой модели инфекционного заболевания был предложен подход к выводу организма из хронического состояния путем обострения болезни, то есть перевода хронической формы в острую с последующим выздоровлением. Этот подход к изучению системы путем возмущения её динамики использовался в дальнейшем для математического анализа иммунных процессов при различных вирусных заболеваниях, в первую очередь — при инфекции вирусами иммунодефицита человека, характеризующейся хронической динамикой и летальным исходом [3; 4].

В настоящее время в качестве компактных моделей динамики инфекционных заболеваний и иммунного ответа для сложных пространственно-распределенных биологических систем широко используются модели, представляющие собой системы обыкновенных дифференциальных уравнений с запаздывающим аргументом [5]. Такие системы также называются системами с запаздыванием, системами с последействием, дифференциально-разностными системами. Обычно имеется несколько возможных вариантов динамики инфекционного заболевания, в том числе с высоким и низким количеством патогенов. Первый вариант соответствует хроническому заболеванию с низким уровнем иммунного ответа, а второй — состоянию выздоровевшего организма с иммунной памятью, которая поддерживается за счет антигенной стимуляции небольшой интенсивности. Такую альтернативность развития заболевания отражает мультистабильность соответствующей модели [6], то есть существование как минимум двух устойчивых стационарных либо периодических решений при одних и тех же значениях параметров. Для мультистабильных систем актуален поиск многокомпонентных воздействий, вызывающих максимальный отклик и переводящих систему из первого состояния во второе. Такие воздействия обычно ищут с помощью

одного из двух подходов. Первый подход заключается в использовании затратных методов анализа чувствительности на основе детерминированного либо стохастического варьирования отдельных параметров модели или их комбинаций [7; 8]. Второй подход заключается в решении задач оптимального управления [9–11].

В диссертационной работе представлен оригинальный, существенно более эффективный подход к построению многокомпонентных воздействий, вызывающих максимальный отклик заданной динамической системы. Этот подход основан на оптимальных возмущениях, которые широко используют для описания механизма докритического ламинарно-турбулентного перехода в теории аэродинамической устойчивости для систем без запаздывания (смотри, например, работы [12–14] и их библиографии). Стоит отметить, что ранее понятия оптимального возмущения для систем с запаздыванием не существовало. В диссертационной работе впервые вводятся понятия оптимальных возмущений стационарных и периодических решений систем с запаздыванием и предлагаются методы вычисления таких возмущений.

Устойчивые стационарные и периодические решения моделей инфекционных заболеваний, сформулированных в виде систем с запаздыванием и использующихся в математической иммунологии, можно интерпретировать как хронические формы заболевания. Поэтому разработка эффективной терапии, основанной на оптимальных возмущениях, должна начинаться с поиска таких решений как функций ее параметров и анализа устойчивости этих состояний, то есть с бифуркационного анализа. В настоящее время наиболее известным и используемым на практике численным программным обеспечением для бифуркационного анализа систем обыкновенных дифференциальных уравнений с запаздыванием является пакет DDE-BIFTOOL, впервые описанный в работе [15]. Однако этот пакет, во-первых, является слишком универсальным и не учитывает специфику моделей динамики инфекционных заболеваний. Во-вторых, он ориентирован на классический анализ бифуркаций решения при варьировании одного скалярного параметра, а его использование требует значительной «ручной работы».

Таким образом, актуальной является задача создания вычислительной технологии, предназначенной для вычисления и анализа стационарных и периодических решений систем с запаздыванием и для вычисления оптимальных

возмущений стационарных и периодических решений, переводящих систему из одного устойчивого состояния в другое.

**Целью** диссертационной работы является разработка оригинальной технологии для вычисления и анализа стационарных и периодических решений систем с запаздыванием и вычисления для них многокомпонентных воздействий, основанных на оптимальных возмущениях.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Разработать методы вычисления стационарных и периодических решений систем с запаздыванием при фиксированных значениях параметров, анализа их устойчивости, трассировки стационарных решений по параметрам.
2. Разработать методы вычисления оптимальных возмущений стационарных и периодических решений систем с запаздыванием.
3. Применить разработанные методы для анализа использующихся в математической иммунологии моделей инфекционных заболеваний, представляющих собой системы с запаздыванием.

**Основные положения, выносимые на защиту:**

1. Впервые вводятся понятия оптимальных возмущений стационарных и периодических решений систем с запаздыванием.
2. Разработаны алгоритмы вычисления оптимальных возмущений. Сравнивается эффективность предложенных алгоритмов.
3. Разработана технология, включающая в себя методы вычисления всех стационарных решений систем с запаздыванием, анализа их устойчивости, исследования их зависимости от параметра, вычисления и анализа устойчивости периодических решений систем с запаздыванием. Также разработанная технология включает упомянутые выше алгоритмы вычисления оптимальных возмущений стационарных и периодических решений систем с запаздыванием.
4. С помощью разработанной технологии выполнен анализ двух широко известных математических моделей вирусных инфекций: модели динамики экспериментальной инфекции, вызванной вирусами лимфоцитарного хориоменингита (далее модель LCMV) и модели Марчука-Петрова противовирусного иммунного ответа на инфекцию вирусами гепатита

В (далее модель HBV). Впервые вычислены стационарные и периодические решения, соответствующие хроническим формам заболевания. Впервые показано наличие у этих моделей свойств бистабильности и гистерезиса. На примере этих моделей выполнен анализ возможности использования оптимальных возмущений для перевода систем, описывающих инфекционное заболевание, из одного устойчивого стационарного или периодического решения в другое.

**Научная и практическая значимость** заключается в разработанных методах, позволяющих, во-первых, эффективно находить стационарные и периодические решения моделей динамики инфекций и иммунного ответа, представляющих собой системы с запаздыванием, которые соответствуют различным хроническим формам заболевания. Во-вторых, разработанные методы позволяют эффективно вычислять оптимальные возмущения, позволяющие перевести систему из состояния, соответствующего хронической форме заболевания, в состояние здорового организма. **Практическая значимость** заключается в реализации предложенных методов в виде численного программного комплекса DEODAN (Delay Equations Optimal Disturbances ANalysis). Разработанные методы и их реализация в рамках пакета DEODAN были применены для анализа моделей LCMV и HBV. С помощью разработанных методов впервые удалось вычислить стационарные и периодические решения этих моделей, соответствующие хроническим формам заболеваний, и показать наличие у этих моделей важных для дальнейшей разработки эффективной терапии свойств бистабильности и гистерезиса.

**Научная новизна:**

1. Предлагается оригинальная технология, включающая в себя методы вычисления всех стационарных решений систем с запаздыванием при фиксированных значениях параметров, их трассировки по параметрам, анализа их устойчивости, вычисления периодических решений и анализа их устойчивости.
2. Вводятся определения оптимальных возмущений стационарных и периодических решений систем с запаздыванием. Предлагаются методы вычисления оптимальных возмущений стационарных и периодических решений систем с запаздыванием.

3. Впервые вычислены стационарные и периодические решения моделей LCMV и HBV, соответствующие различным хроническим формам заболеваний, и впервые найдены условия существования свойств бистабильности, мультистабильности и гистерезиса у этих моделей.

**Достоверность** результатов, полученных в диссертации, обоснована теоретическим анализом предложенных методов, а также всесторонним численным исследованием этих методов на тестовых задачах, включающих в себя численные эксперименты с моделями LCMV и HBV.

**Апробация работы.** Автор лично докладывал основные результаты работы на следующих международных и российских конференциях:

- 60-я, 61-я и 63-я всероссийские конференции МФТИ, Москва, Россия, 2017-2020;
- 17-й международный симпозиум по математической и вычислительной биологии «БИОМАТ 2017», Москва, Россия, 2017;
- международная конференция «Аналитические и численные методы решения задач гидродинамики, математической физики и биологии», Москва, Россия, 2019;
- международная конференция «Mathematical modelling in biomedicine», Москва, Россия, 2019;
- международная конференция «Марчуковские чтения», Новосибирск, Россия, 2021;
- девятая международная конференция по дифференциальным и функционально-дифференциальным уравнениям, Москва, Россия, 2022;
- международная конференция «Bioinformatics of genome regulation and structure/systems biology», Новосибирск, Россия, 2022.

**Публикации.** По теме диссертации опубликовано 25 работ [16–40], из них 14 работ [16–29] — в рецензируемых научных изданиях, в которых должны быть опубликованы основные научные результаты диссертации на соискание учёной степени кандидата наук. Из этих 14 работ 10 работ [16–20; 25–29] опубликованы в научных изданиях, индексируемых в международных базах данных Web of Science или Scopus.

**Личный вклад.** Соискатель участвовал в формулировке цели и задач исследования, разработке и анализе всех предложенных алгоритмов и планировании численных экспериментов. В работах [16–29] соискателем была выполнена



программная реализация разработанных алгоритмов и выполнены все численные эксперименты с моделями LCMV и HBV. В работах [19; 22] соискателем было выполнено теоретическое и численное сравнение эффективности разработанных алгоритмов. В работах [23; 27] соискателем был разработан метод гарантированного вычисления всех стационарных решений модели LCMV и был разработан метод трассирования всех стационарных решений этой модели, который был обобщен соискателем на систему с дискретными запаздываниями общего вида в работе [24]. В работе [25] соискателем был разработан метод ньютоновского типа для уточнения приближенного периодического решения. В работе [20] соискателем были разработаны алгоритмы вычисления оптимального возмущения периодического решения.

**Объем и структура диссертационной работы.** Диссертация состоит из введения, 5 глав и заключения. Полный объем диссертации составляет 138 страниц, включая 34 рисунка и 29 таблиц. Список литературы содержит 61 наименование.

**Содержание работы.** В **первой** главе вводится в рассмотрение общего вида система дифференциальных уравнений с дискретными (постоянными) запаздываниями и кратко описаны рассматриваемые в качестве примеров для демонстрации разработанной технологии математические модели LCMV и HBV. **Вторая** глава диссертации посвящена оптимальным возмущениям стационарных решений системы с запаздыванием. Предлагаются методы вычисления стационарных решений, анализа их устойчивости, анализа их зависимости от параметра и вычисления для них оптимальных возмущений. **Третья** глава диссертации посвящена оптимальным возмущениям периодических решений системы с запаздыванием. Предлагаются методы вычисления периодических решений, анализа их устойчивости и вычисления для них оптимальных возмущений. В **четвертой** главе приводится описание пакета программ DEODAN, в виде которого реализована предложенная технология. В **пятой** главе приводятся результаты численных экспериментов с моделями инфекций, вызванных вирусами LCMV и HBV. В **заключении** перечислены основные результаты работы.

**Благодарности.** Автор выражает благодарность научному руководителю Нечепуренко Ю.М., научному консультанту Бочарову Г.А., соавторам работ Гребенникову Д.С. и Склярской Е.В..

## Глава 1. Системы с запаздыванием

### 1.1 Системы дифференциальных уравнений с запаздыванием

Системы дифференциальных уравнений с дискретными (постоянными) запаздываниями в настоящее время широко используются в качестве моделей динамики инфекционных заболеваний и иммунного ответа. Такие модели имеют форму

$$\frac{du}{dt}(t) = \mathcal{F}(u(t - \tau_0), \dots, u(t - \tau_q), \mathbf{p}), \quad (1.1)$$

где  $\mathcal{F}(v_0, v_1, \dots, v_q, \mathbf{p})$  — рациональная  $n$ -компонентная вектор-функция векторных аргументов  $v_0, v_1, \dots, v_q, \mathbf{p}$ ,  $\tau_0 = 0$ ,  $0 < \tau_1 < \dots < \tau_q$  — запаздывания, а  $\mathbf{p}$  — вектор параметров модели, определяющих скорости различных биологических процессов.

Пусть далее  $C[t_1, t_2]$ , где  $t_1 < t_2$ , означает линейное пространство кусочно непрерывных  $n$ -компонентных векторных функций вида

$$f : [t_1, t_2] \rightarrow \mathbf{R}^n. \quad (1.2)$$

Через  $\mathbf{R}_+$  будем обозначать множество вещественных неотрицательных чисел, а через  $C_+[t_1, t_2]$  — подмножество функций из  $C[t_1, t_2]$ , принимающих только неотрицательные значения. Будем предполагать, что при любой начальной функции  $u^0 \in C_+[-\tau_q, 0]$  на любом конечном временном интервале  $0 \leq t \leq t_{\max}$  решение задачи Коши для системы (1.1) существует, единственно и неотрицательно. А именно, найдется единственная функция  $u \in C_+[-\tau_q, t_{\max}]$ , которая удовлетворяет уравнению (1.1) при  $0 < t < t_{\max}$  и совпадает с  $u^0$  в интервале  $-\tau_q \leq t \leq 0$ . Стоит отметить, что результаты работы справедливы, если функция  $u^0$  имеет разрыв 1-го рода при  $t = 0$ .

Для численного интегрирования системы (1.1) мы будем использовать неявный метод Эйлера

$$\frac{u_k - u_{k-1}}{\delta} = \mathcal{F}(u_k, u_{k-m_1}, \dots, u_{k-m_q}, \mathbf{p}), \quad k \geq 1, \quad (1.3)$$

или неявную схему второго порядка BDF2 [61]

$$\frac{1.5u_k - 2u_{k-1} + 0.5u_{k-2}}{\delta} = \mathcal{F}(u_k, u_{k-m_1}, \dots, u_{k-m_q}, \mathbf{p}), k \geq 1, \quad (1.4)$$

на равномерной сетке

$$\{\delta k : k = -m_q + 1, -m_q + 2, \dots\}, \quad (1.5)$$

построенной в интервале  $(-\tau_q, \infty)$  с шагом  $\delta > 0$ , где  $m_j = [\tau_j/\delta]$  ( $[\cdot]$  означает целую часть действительного числа) являются дискретными аналогами задержек  $\tau_j$ .

## 1.2 Математические модели динамики инфекционных заболеваний

### 1.2.1 Модель экспериментальной вирусной инфекции LCMV

Одной из широко используемых моделей экспериментальных инфекций является модель инфекции мышей вирусами лимфоцитарного хориоменингита. Эта экспериментальная инфекция является базовой вирусной инфекцией современной иммунологии, с помощью которой были установлены важнейшие механизмы развития иммунопатологических процессов и хронизации инфекций. Соответствующие закономерности были описаны на уровне системной динамики вирусов и иммунного ответа (активация, анергия, истощение Т-клеточных реакций) [41]. Для неё были количественно изучены характерные режимы (фенотипы) динамики и показано, что течение и исход заболевания зависят как от скорости размножения и распространения возбудителя, так и от интенсивности развития иммунного ответа. Высокая степень достигнутого понимания процессов в системе вирус-организм хозяина для данной инфекции позволила перейти к построению биологически значимых математических моделей на основе дифференциальных уравнений с запаздывающим аргументом [41; 42; 44]. С помощью таких моделей может быть предсказана чувствительность динамики к противовирусным и иммуномодулирующим воздействиям, что позволит,

в конечном итоге, исследовать возможности оптимальной коррекции динамики хронических вирусных заболеваний, таких как гепатит В и ВИЧ, имеющих ряд общих закономерностей в патогенезе с инфекцией вирусами лимфоцитарного хориоменингита.

Математическая модель LCMV, предложенная и проанализированная в работе [42], сформулирована в виде системы нелинейных дифференциальных уравнений с запаздывающим аргументом:

$$\begin{aligned}
\frac{dV}{dt}(t) &= \beta V(t) \left(1 - \frac{V(t)}{V_{mvc}}\right) - \gamma_{VE} E_e(t) V(t), \\
\frac{dE_p}{dt}(t) &= \alpha_{E_p}(E_p^0 - E_p(t)) + b_p g_p(W) V(t - \tau) E_p(t - \tau) \\
&\quad - \alpha_{AP} V(t - \tau_A) V(t) E_p(t), \\
\frac{dE_e}{dt}(t) &= b_d g_e(W) V(t - \tau) E_p(t - \tau) \\
&\quad - \alpha_{AE} V(t - \tau_A) V(t) E_e(t) - \alpha_{E_e} E_e(t), \\
\frac{dW}{dt}(t) &= b_W V(t) - \alpha_W W(t),
\end{aligned} \tag{1.6}$$

где  $g_p(W) = 1/(1 + W/\theta_p)^2$ ,  $g_e(W) = 1/(1 + W/\theta_E)^2$ . Система уравнений модели описывает динамику следующих переменных: концентрации вирусных частиц  $V$ , двух популяций специфичных к вирусам лимфоцитарного хориоменингита цитотоксических лимфоцитов (ЦТЛ) — клеток-предшественников  $E_p$  и клеток-эффекторов  $E_e$ , а также кумулятивной вирусной нагрузки  $W$ . Биологический смысл параметров системы пояснен в таблице 1.

Для определения решения системы (1.6) при  $t > 0$  достаточно задать значения  $V(t)$  при  $-\tau_A \leq t \leq 0$ , значения  $E_p(t)$  при  $-\tau \leq t \leq 0$ , значения  $E_e(0)$  и  $W(0)$ . Однако для единообразия далее мы будем предполагать, что начальные значения всех переменных заданы при  $-\tau_A \leq t \leq 0$ .

Обозначив вектор переменных системы (1.6) через

$$u(t) = (V(t), E_p(t), E_e(t), W(t))^T,$$

ее можно записать в виде (1.1), где  $n = 4$ ,  $q = 2$ .

Задача Коши для системы (1.6) с неотрицательными начальными значениями и неотрицательными параметрами глобально разрешима на любом

Таблица 1 — Биологический смысл параметров модели (1.6).

Параметр	Биологический смысл
$\beta$	Константа скорости репликации вирусных частиц
$\gamma_{VE}$	Константа скорости элиминации вирусов за счет клеток-эффекторов
$V_{mvc}$	Максимально возможная концентрация вирусных частиц в селезенке
$\tau$	Продолжительность цикла деления ЦТЛ
$b_p$	Константа скорости стимуляции ЦТЛ
$b_d$	Константа скорости дифференцировки ЦТЛ
$\theta_p$	Порог кумулятивной вирусной нагрузки для перехода прекурсоров в состояние анергии
$\theta_E$	Порог кумулятивной вирусной нагрузки для перехода эффекторов в состояние анергии
$\alpha_{E_p}$	Константа скорости естественной гибели прекурсоров
$\alpha_{E_e}$	Константа скорости естественной гибели эффекторов
$E_p^0$	Концентрация прекурсоров в селезенке мыши, не имевшей контакта с вирусами лимфоцитарного хориоменингита
$\tau_A$	Продолжительность перехода ЦТЛ к апоптозу
$\alpha_{AP}$	Константа скорости апоптоза прекурсоров
$\alpha_{AE}$	Константа скорости апоптоза эффекторов
$b_W$	Константа скорости роста кумулятивной вирусной нагрузки
$\alpha_W$	Константа скорости восстановления организма от воздействий кумулятивной вирусной нагрузки

конечном временном интервале. Данное утверждение можно доказать используя технику, описанную в [1], на основе метода шагов Беллмана, рассматривая линейную систему обыкновенных дифференциальных уравнений, мажорирующую правую часть системы уравнений модели.

### 1.2.2 Модель инфекции человека вирусами HBV

В этом разделе мы рассматриваем модель динамики инфекции вирусом гепатита В (HBV), которая является калиброванной версией модели противовирусного иммунного ответа Марчука-Петрова. Модель Марчука-Петрова является более полной с точки зрения описания совокупности иммунофизиологических процессов, чем модель LCMV, рассмотренная в предыдущем разделе. Эта

модель описывает динамику вирусной инфекции и иммунного ответа системой десяти уравнений с десятью переменными. При этом в модели учитываются в виде запаздываний продолжительность процессов деления и дифференцировки клеток иммунной системы. С помощью модели Марчука-Петрова были изучены количественные закономерности динамики острых инфекций человека, вызванных вирусами гриппа А и вирусами гепатита В. В отличие от гриппа, вирусный гепатит В имеет, наряду с острой формой, также хронические варианты течения, отличающиеся уровнем вирусной нагрузки, степенью повреждения клеток печени и интенсивностью иммунных реакций. Высокая степень детализации описания процессов вирусной инфекции и иммунного ответа в многопараметрической модели Марчука-Петрова является предпосылкой для исследования на ее основе более глубоких закономерностей развития хронических форм течения вирусного гепатита В [43]. Также, стоит отметить, что модель Марчука-Петрова и ее последующие модификации, разработанные для описания иммунофизиологических реакций организма, а также смешанных вирусно-бактериальных инфекций [2], являются классическими математическими моделями, используемыми для анализа механизмов развития вирусных заболеваний.

Математическая модель противовирусного иммунного ответа Марчука-Петрова предложена в [2; 43]. Система уравнений модели описывает скорость изменения во времени концентрации следующих популяций: вирусных частиц  $V_f$ ; зараженных вирусами клеток органа-мишени  $C_V$ ; разрушенных клеток органа-мишени  $m$ ; антигенпрезентирующих клеток (макрофагов)  $M_V$ ;  $CD4^+$  Т-лимфоцитов — помощников клеточного иммунитета (Th1)  $H_E$ ;  $CD4^+$  Т-лимфоцитов — помощников гуморального иммунитета (Th2)  $H_B$ ;  $CD8^+$  Т-лимфоцитов — киллеров  $E$ , уничтожающих зараженные вирусами клетки; В-лимфоцитов  $B$ ; плазматических клеток  $P$ , производящих антитела; антител  $F$ , нейтрализующих вирусы. Будем предполагать далее, что переменные  $V_f$  и  $F$  имеют размерность частиц/мл, а остальные переменные — клеток/мл. Система имеет следующий вид:

$$\begin{aligned} \frac{d}{dt}V_f(t) = & \nu C_V(t) + nb_{CE}C_V(t)E(t) - \gamma_{VF}V_f(t)F(t) - \\ & - \gamma_{VM}V_f(t) - \gamma_{VC}V_f(t) [C^0 - C_V(t) - m(t)], \end{aligned}$$

$$\begin{aligned}
\frac{d}{dt}C_V(t) &= \sigma V_f(t) [C^0 - C_V(t) - m(t)] - \\
&\quad - b_{CE}C_V(t)E(t) - b_m C_V(t), \\
\frac{d}{dt}m(t) &= b_{CE}C_V(t)E(t) + b_m C_V(t) - \alpha_m m(t), \\
\frac{d}{dt}M_V(t) &= \gamma_{MV}M^0 V_f(t) - \alpha_M M_V(t), \\
\frac{d}{dt}H_E(t) &= b_H^E[\xi(m)\rho_H^E M_V(t - \tau_H^E)H_E(t - \tau_H^E) - \\
&\quad - M_V(t)H_E(t)] - b_p^{HE} M_V(t)H_E(t)E(t) + \\
&\quad + \alpha_H^E(H_E^0 - H_E(t)), \\
\frac{d}{dt}E(t) &= b_p^E[\xi(m)\rho_E M_V(t - \tau_E)H_E(t - \tau_E)E(t - \tau_E) - \\
&\quad - M_V(t)H_E(t)E(t)] - \\
&\quad - b_{EC}C_V(t)E(t) + \alpha_E(E^0 - E(t)), \\
\frac{d}{dt}H_B(t) &= b_H^B[\xi(m)\rho_H^B M_V(t - \tau_H^B)H_B(t - \tau_H^B) - \\
&\quad - M_V(t)H_B(t)] - b_p^{HB} M_V(t)H_B(t)B(t) + \\
&\quad + \alpha_H^B(H_B^0 - H_B(t)), \\
\frac{d}{dt}B(t) &= b_p^B[\xi(m)\rho_B M_V(t - \tau_B)H_B(t - \tau_B)B(t - \tau_B) - \\
&\quad - M_V(t)H_B(t)B(t)] + \alpha_B(B^0 - B(t)), \\
\frac{d}{dt}P(t) &= b_p^P \xi(m)\rho_P M_V(t - \tau_P)H_B(t - \tau_P)B(t - \tau_P) + \\
&\quad + \alpha_P(P^0 - P(t)), \\
\frac{d}{dt}F(t) &= \rho_F P(t) - \gamma_{FV}F(t)V_f(t) - \alpha_F F(t),
\end{aligned} \tag{1.7}$$

где  $\xi(m) = 1 - m/C^0$ . Эта система состоит из трех блоков уравнений: блока, описывающего процессы развития инфекции, протекающие в органе-мишени (уравнения 1–3), блока, описывающего Т-клеточный иммунный ответ (уравнения 4–6), и блока, описывающего гуморальный иммунитет (уравнения 7–10). Биологический смысл параметров системы пояснен в таблицах 2 и 3. Для определения решения системы при  $t > 0$ , достаточно задать значения  $M_V(t)$  при  $-\tau_5 \leq t \leq 0$ , где  $\tau_5 = \max\{\tau_H^E, \tau_H^B, \tau_E, \tau_B, \tau_P\}$ , значения  $H_E(t)$  при  $-\max\{\tau_H^E, \tau_E\} \leq t \leq 0$ , значения  $E(t)$  при  $-\tau_E \leq t \leq 0$ , значения  $H_B(t)$  при  $-\max\{\tau_H^B, \tau_B, \tau_P\} \leq t \leq 0$ , значения  $B(t)$  при  $-\max\{\tau_B, \tau_P\} \leq t \leq 0$ ,

и значения остальных переменных при  $t = 0$ . Однако для единообразия далее мы будем предполагать, что начальные значения всех переменных заданы при  $-\tau_5 \leq t \leq 0$ . Модель Марчука-Петрова со значениями переменных, представленными в таблице 4 и соответствующими острой инфекции гепатита В, является моделью HBV.

Обозначив вектор переменных системы (1.7) через

$$u(t) = (V_f(t), C_V(t), m(t), M_V(t), H_E(t), E(t), H_B(t), B(t), P(t), F(t))^T,$$

ее можно записать в виде (1.1) где  $n = 10$ ,  $q = 5$ .

Задача Коши с неотрицательными начальными значениями и неотрицательными значениями параметров, как и для системы (1.6), для системы (1.7) глобально разрешима на любом конечном временном интервале.

Таблица 2 — Параметры иммунного гомеостаза и развития вирусной инфекции в клетках-мишенях.

Параметр	Биологический смысл параметра
$b_m$	Константа скорости разрушения зараженных гепатоцитов вследствие цитопатичности вирусов
$\alpha_m$	Константа скорости регенерации гепатоцитов
$M^0$	Концентрация Ia-несущих макрофагов в лимфоузле
$H_E^0$	Концентрация специфических Т-хелперов Th1 в лимфоузле
$H_B^0$	Концентрация специфических Т-хелперов Th2 в лимфоузле
$E^0$	Концентрация специфических предшественников для Т-лимфоцитов-эффекторов в лимфоузле
$B^0$	Концентрация специфических В-лимфоцитов в лимфоузле
$P^0$	Концентрация специфических плазматических клеток в лимфоузле
$C^0$	Концентрация гепатоцитов в печени
$\gamma_{MV}$	Константа скорости антигенной стимуляции макрофагов в лимфоузле
$\gamma_{VM}$	Константа скорости связывания антигенных частиц макрофагами лимфоузла
$\gamma_{FV}$	Константа скорости связывания 1 молекулы IgG с частицей HBsAg
$\gamma_{VF}$	Константа скорости нейтрализации вируса гепатита молекулами IgG
$\sigma$	Константа скорости заражения гепатоцитов
$\nu$	Константа скорости секреции частиц HBsAg одним гепатоцитом в сутки
$n$	Количество частиц HBsAg, высвобождающееся при разрушении гепатоцита Т-лимфоцитом-эффектором
$\gamma_{VC}$	Константа скорости адсорбции вирусов незараженными клетками органа-мишени



Таблица 3 — Параметры развития иммунного ответа.

Параметр	Биологический смысл параметра
$\alpha_M$	Константа скорости потери макрофагом стимулированного состояния
$\alpha_H^E$	Константа скорости потери стимулированного состояния Т-хелперами Th1
$\alpha_H^B$	Константа скорости потери стимулированного состояния Т-хелперами Th2
$\alpha_E$	Константа скорости естественной гибели цитотоксических Т-лимфоцитов-эффекторов
$\alpha_B$	Константа скорости естественной гибели В-лимфоцитов
$\alpha_P$	Константа скорости естественной гибели плазматических клеток
$\alpha_F$	Константа скорости естественной гибели антител
$\tau_H^E$	Продолжительность цикла деления Т-хелперов Th1
$\tau_H^B$	Продолжительность цикла деления Т-хелперов Th2
$\tau_E$	Продолжительность цикла делений Т-лимфоцитов-эффекторов
$\tau_B$	Продолжительность цикла делений В-лимфоцитов
$\tau_P$	Продолжительность цикла делений и дифференцировки В-лимфоцитов до появления плазматических клеток
$\rho_H^E$	Число потомков Т-хелпера Th1 в результате 1-го цикла деления
$\rho_H^B$	Число потомков Т-хелпера Th2 в результате 1-го цикла деления
$\rho_E$	Число потомков Т-лимфоцита-эффектора в результате 1-го цикла делений
$\rho_B$	Число потомков В-лимфоцита в результате 1-го цикла делений
$\rho_P$	Число плазматических клеток потомков В-лимфоцита в результате одного цикла делений
$b_p^B$	Константа скорости стимуляции В-лимфоцита при описании числа В-лимфоцитов
$b_p^P$	Константа скорости стимуляции В-лимфоцита при описании числа плазматических клеток
$b_H^E$	Константа скорости стимуляции Т-хелпера Th1
$b_H^B$	Константа скорости стимуляции Т-хелпера Th2
$b_p^E$	Константа скорости стимуляции Т-лимфоцита-эффектора
$b_p^{HE}$	Коэффициент, характеризующий расход Т-хелперов Th1 на стимуляцию Т-лимфоцитов-эффекторов
$b_p^{HB}$	Коэффициент, характеризующий расход Т-хелперов Th1 на стимуляцию В-лимфоцитов
$\rho_F$	Константа скорости синтеза молекул IgG одной плазматической клеткой
$b_{CE}$	Константа скорости разрушения гепатоцитов Т-лимфоцитами-эффекторами
$b_{EC}$	Константа скорости гибели Т-лимфоцитов-эффекторов вследствие разрушения зараженных клеток

Таблица 4 — Значения параметров модели (1.7), соответствующие острому гепатиту В [55].

Параметр	Значение	Параметр	Значение
$\alpha_M$	$1.2 \text{ сут}^{-1}$	$b_p^{HB}$	$2.2 \cdot 10^{-13} (\text{мл/клеток})^2 \cdot \text{сут}^{-1}$
$\alpha_H^E$	$1.0 \text{ сут}^{-1}$	$\rho_F$	$1.7 \cdot 10^8 (\text{частиц/клеток}) \cdot \text{сут}^{-1}$
$\alpha_H^B$	$1.0 \text{ сут}^{-1}$	$b_{CE}$	$1.1 \cdot 10^{-6} (\text{мл/частиц}) \cdot \text{сут}^{-1}$
$\alpha_E$	$0.4 \text{ сут}^{-1}$	$b_{EC}$	$2.7 \cdot 10^{-7} (\text{мл/клеток}) \cdot \text{сут}^{-1}$
$\alpha_B$	$0.1 \text{ сут}^{-1}$	$b_m$	$0.068 \text{ сут}^{-1}$
$\alpha_P$	$0.4 \text{ сут}^{-1}$	$\alpha_m$	$0.15 \text{ сут}^{-1}$
$\alpha_F$	$0.043 \text{ сут}^{-1}$	$M^0$	$6.0 \cdot 10^5 \text{ клеток/мл}$
$\tau_H^E$	$0.6 \text{ сут}$	$H_E^0$	$6.0 \cdot 10^2 \text{ клеток/мл}$
$\tau_H^B$	$0.6 \text{ сут}$	$H_B^0$	$6.0 \cdot 10^1 \text{ клеток/мл}$
$\tau_E$	$2.0 \text{ сут}$	$E^0$	$6.0 \cdot 10^2 \text{ клеток/мл}$
$\tau_B$	$2.0 \text{ сут}$	$B^0$	$6.0 \cdot 10^2 \text{ клеток/мл}$
$\tau_P$	$3.0 \text{ сут}$	$P^0$	$2.6 \cdot 10^{-1} \text{ клеток/мл}$
$\rho_H^E, \rho_H^B$	2	$C^0$	$3.0 \cdot 10^8 \text{ клеток/мл}$
$\rho_E, \rho_B$	16	$\gamma_{MV}$	$1.6 \cdot 10^{-11} (\text{мл/клеток}) \cdot \text{сут}^{-1}$
$\rho_P$	3	$\gamma_{VM}$	$0.4 \text{ сут}^{-1}$
$b_p^B$	$2.2 \cdot 10^{-9} (\text{мл/клеток})^2 \cdot \text{сут}^{-1}$	$\gamma_{FV}$	$1.4 \cdot 10^{-9} (\text{мл/частиц}) \cdot \text{сут}^{-1}$
$b_p^P$	$4.7 \cdot 10^{-12} (\text{мл/клеток})^2 \cdot \text{сут}^{-1}$	$\gamma_{VF}$	$5.0 \cdot 10^{-10} (\text{мл/частиц}) \cdot \text{сут}^{-1}$
$b_H^E$	$4.5 \cdot 10^{-5} (\text{мл/клеток}) \cdot \text{сут}^{-1}$	$\sigma$	$3.8 \cdot 10^{-12} (\text{мл/клеток}) \cdot \text{сут}^{-1}$
$b_H^B$	$4.5 \cdot 10^{-5} (\text{мл/клеток}) \cdot \text{сут}^{-1}$	$\nu$	$83 \text{ сут}^{-1}$
$b_p^E$	$1.5 \cdot 10^{-8} (\text{мл/клеток})^2 \cdot \text{сут}^{-1}$	$n$	5
$b_p^{HE}$	$1.5 \cdot 10^{-14} (\text{мл/клеток})^2 \cdot \text{сут}^{-1}$	$\gamma_{VC}$	$4.2 \cdot 10^{-14} (\text{мл/клеток}) \cdot \text{сут}^{-1}$

## Глава 2. Оптимальные возмущения стационарных решений

### 2.1 Вычисление стационарных решений

Стационарные решения системы (1.1) при заданных значениях параметров являются решениями системы алгебраических уравнений

$$\mathcal{G}(u, \mathbf{p}) = \mathcal{F}(\underbrace{u, \dots, u}_{q+1}, \mathbf{p}) = 0, \quad (2.1)$$

где  $\mathcal{F}$  — рациональная функция, фигурирующая в правой части уравнения (1.1). Для численного решения систем алгебраических уравнений в настоящее время используют методы вычислительной алгебры [49], которые также называют методами символьных вычислений. В частности такие методы реализованы в процедуре NSolve пакета Mathematica. Эта процедура основана на вычислении базиса Гребнера с использованием мономиального упорядочения и теоретически позволяет получить аппроксимацию решений произвольной системы алгебраических уравнений.

В некоторых случаях при непосредственном применении процедуры NSolve к системе (2.1) алгоритм работает неправильно и в качестве решения выдает пустое множество. Чтобы избежать этого, необходимо выполнить некоторые предварительные преобразования.

Для повышения точности вычислений решения, полученные в результате применения процедуры NSolve к системе (2.1), могут быть уточнены методом Ньютона, который заключается в следующем итерационном процессе:

$$\bar{u}_{i+1} = \bar{u}_i - \left( \frac{\partial \mathcal{G}}{\partial u}(\bar{u}_i, \mathbf{p}) \right)^{-1} \mathcal{G}(\bar{u}_i, \mathbf{p}), \quad (2.2)$$

где  $i$  — номер итерации, а  $\bar{u}$  —  $n$ -компонентный вектор значений переменных в стационарном решении.

### 2.1.1 Стационарные решения модели LCMV

Для модели LCMV (1.6) система (2.1) имеет следующий вид:

$$\begin{aligned} \beta V \left(1 - \frac{V}{V_{mvc}}\right) - \gamma_{VE} E_e V &= 0, \\ \alpha_{E_p} (E_p^0 - E_p) + b_p g_p(W) V E_p - \alpha_{AP} V^2 E_p &= 0, \\ b_d g_e(W) V E_p - \alpha_{AE} V^2 E_e - \alpha_{E_e} E_e &= 0, \\ b_W V - \alpha_W W &= 0. \end{aligned} \quad (2.3)$$

Система (2.3) может быть успешно решена с помощью процедуры NSolve. Кроме того, в [27] был предложен альтернативный, более простой метод решения системы (2.3). Этот метод основан на сведении этой системы к уравнению

$$Y(V) = \alpha_{E_p} E_p^0 - (\alpha_{E_p} - b_p g_p(W) V + \alpha_{AP} V^2) E_p = 0 \quad (2.4)$$

относительно  $V$ , где

$$E_p = \frac{(\alpha_{E_e} + \alpha_{AE} V^2) E_e}{b_d V g_e(W)}, \quad E_e = \frac{\beta(1 - V/V_{mvc})}{\gamma_{VE}}, \quad W = \frac{b_W V}{\alpha_W}, \quad (2.5)$$

и на вычислении решений этих уравнений с использованием стандартных процедур fzero и fmin [48; 54], включенных во многие численные пакеты прикладных программ.

### 2.1.2 Стационарные решения модели HBV

Для модели HBV (1.7) система (2.1) имеет следующий вид:

$$\begin{aligned} \nu C_V + nb_{CE} C_V E - \gamma_{VF} V_f F - \\ - \gamma_{VM} V_f - \gamma_{VC} V_f [C^0 - C_V - m] &= 0, \end{aligned}$$

$$\begin{aligned}
\sigma V_f [C^0 - C_V - m] - b_{CE} C_V E - b_m C_V &= 0, \\
b_{CE} C_V E + b_m C_V - \alpha_m m &= 0, \\
\gamma_{MV} M^0 V_f - \alpha_M M_V &= 0, \\
[b_H^E (\xi(m) \rho_H^E - 1)] M_V H_E - b_p^{H_E} M_V H_E E + \alpha_H^E (H_E^0 - H_E) &= 0, \\
[b_p^E (\xi(m) \rho_E - 1)] M_V H_E E - b_{EC} C_V E + \alpha_E (E^0 - E) &= 0, \\
[b_H^B (\xi(m) \rho_H^B - 1)] M_V H_B - b_p^{H_B} M_V H_B B + \alpha_H^B (H_B^0 - H_B) &= 0, \\
[b_p^B (\xi(m) \rho_B - 1)] M_V H_B B + \alpha_B (B^0 - B) &= 0, \\
[b_p^P \xi(m) \rho_P] M_V H_B B + \alpha_P (P^0 - P) &= 0, \\
\rho_F P - \gamma_{FV} F V_f - \alpha_F F &= 0,
\end{aligned} \tag{2.6}$$

где  $\xi(m) = 1 - m/C^0$ .

При применении процедуры NSolve непосредственно к системе (2.6) алгоритм возвращает пустое множество в качестве решения. Как мы упоминали ранее, чтобы избежать этого, необходимо выполнить некоторые предварительные преобразования.

Отметим, прежде всего, что система (2.6) имеет тривиальное решение

$$\begin{aligned}
V_f = C_V = m = M_V = 0, \quad H_E = H_E^0, \quad E = E^0, \\
H_B = H_B^0, \quad B = B^0, \quad P = P^0, \quad F = \frac{\rho_F P^0}{\alpha_F}.
\end{aligned} \tag{2.7}$$

В работе [59] было показано, что если исключить это решение, то оставшуюся систему можно свести к следующей системе из четырех уравнений относительно переменных  $V_f$ ,  $C_V$ ,  $H_B$  и  $H_E$ :

$$\begin{aligned}
[b_H^B (\xi(m) \rho_H^B - 1)] M_V H_B - b_p^{H_B} R + \alpha_H^B (H_B^0 - H_B) &= 0, \\
M_V H_B B &= R, \\
[b_H^E (\xi(m) \rho_H^E - 1)] M_V H_E - b_p^{H_E} M_V H_E E + \alpha_H^E (H_E^0 - H_E) &= 0, \\
[b_p^E (\xi(m) \rho_E - 1)] M_V H_E E - b_{EC} C_V E + \alpha_E (E^0 - E) &= 0,
\end{aligned} \tag{2.8}$$

где

$$\begin{aligned} m &= \frac{\sigma V_f (C^0 - C_V)}{\alpha_m + \sigma V_f}, \quad E = \frac{\alpha_m m - b_m C_V}{b_{CE} C_V}, \\ M_V &= \frac{\gamma_{MV} M^0 V_f}{\alpha_M}, \quad P = \frac{b_p^P \xi(m) \rho_P R}{\alpha_P} + P^0, \\ F &= \frac{\rho_F P}{\gamma_{FV} V_f + \alpha_F}, \quad B = \frac{[b_p^B (\xi(m) \rho_B - 1)] R}{\alpha_B} + B^0 \end{aligned}$$

есть функции, полученные посредством явного выражения соответствующих переменных системы (2.6) через переменные  $V_f, C_V$ , а  $R$  — алгебраическая функция этих двух переменных, полученная путем решения линейного уравнения

$$\begin{aligned} \nu C_V + n b_{CE} C_V E - \gamma_{VF} V_f G(R) - \gamma_{VM} V_f - \\ - \gamma_{VC} V_f [C^0 - C_V - m] = 0, \end{aligned}$$

где

$$G(R) = \frac{\rho_F}{\gamma_{FV} V_f + \alpha_F} \left( \frac{b_p^P \xi(m) \rho_P}{\alpha_P} R + P^0 \right),$$

что следует из первого уравнения системы (2.6). В отличие от исходной системы (2.6) система (2.8) успешно решается процедурой NSolve. Таким образом, вычисление стационарных решений системы (2.6) при фиксированных значениях параметров можно выполнить с использованием пакета Mathematica.

Кроме того, в [59] был предложен альтернативный метод решения системы (2.6). Этот метод состоит из трех этапов. Первый этап состоит в сведении этой системы к системе двух алгебраических уравнений относительно переменных  $C_V$  и  $V_f$ . Второй этап состоит в локализации области возможных неотрицательных решений полученной системы. Третий этап состоит в построении плоских алгебраических кривых, которые являются наборами решений этих двух уравнений в полученной области их локализации, с использованием метода трассировки решений по параметру, предложенного в [27], и поиске точек пересечения полученных кривых. Этот метод более сложный чем метод основанный на сим-

вольных вычислениях, однако он позволяет находить решения системы (2.6) для нерациональной функции  $\xi(m)$ .

## 2.2 Анализ устойчивости стационарного решения

Пусть  $n$ -компонентный вектор  $\bar{u}$  является стационарным решением системы обыкновенных дифференциальных уравнений с запаздывающим аргументом вида (1.1), причем предполагаем, что функция  $\mathcal{F}(v_0, v_1, \dots, v_p, \mathbf{p})$  непрерывно дифференцируема в окрестности  $(\bar{u}, \bar{u}, \dots, \bar{u}, \mathbf{p})$ . Для анализа устойчивости стационарного решения нас будут интересовать непрерывные решения задачи Коши для линеаризованных уравнений

$$\frac{dw}{dt}(t) = \sum_{j=0}^q L_j w(t - \tau_j), \quad (2.9)$$

где

$$L_j = \frac{\partial \mathcal{F}}{\partial v_j}(\bar{u}, \dots, \bar{u}, \mathbf{p})$$

являются постоянными квадратными матрицами порядка  $n$ , а в качестве начального значения берется непрерывная векторная функция, определенная в интервале  $-\tau_q \leq t \leq 0$ .

Стационарное решение  $\bar{u}$  является асимптотически устойчивым, если любое решение системы (2.9) вида

$$w(t) = \tilde{u}e^{\lambda t}, \quad (2.10)$$

где  $\lambda$  — комплексное число, а  $\tilde{u}$  — постоянный  $n$ -компонентный ненулевой вектор, монотонно убывает при  $t \rightarrow +\infty$ , то есть когда вещественная часть любого такого  $\lambda$  является отрицательной. Подставив (2.10) в (2.9), получим следующую нелинейную проблему собственных значений:

$$A(\lambda)\tilde{u} = 0, \quad (2.11)$$

где  $A(\lambda) = \lambda I - \sum_{j=0}^q L_j e^{-\lambda \tau_j}$ , а  $I$  означает единичную матрицу порядка  $n$ . Таким образом, исследование асимптотической устойчивости стационарного реше-

ния  $\bar{u}$  сводится к решению нелинейной проблемы собственных значений (2.11) и проверке, что все найденные собственные значения лежат строго в левой полуплоскости.

Алгоритмы, позволяющие решить полную нелинейную проблему собственных значений вида (2.11), неизвестны. Более того, эта проблема имеет, вообще говоря, бесконечное число собственных значений. Поэтому мы будем аппроксимировать эту проблему рациональной проблемой собственных значений, которая, в свою очередь, сводится к полиномиальной. Ведущие собственные значения полиномиальной проблемы позволят вычислить приближенные ведущие собственные значения исходной проблемы (2.11), которые затем можно уточнить, используя какой-нибудь из локальных методов ньютоновского типа.

Введем новый спектральный параметр  $\mu = e^{\delta\lambda}$ , где  $\delta$  — заданное достаточно малое положительное число. Формулы, которые аппроксимируют первую производную в неявном методе Эйлера (1.3) или в неявной схеме второго порядка BDF2 (1.4), позволяют получить аппроксимацию  $\lambda$ . Дальнейшее описание приведено для аппроксимации  $\lambda$ , полученной с помощью неявной схемы второго порядка BDF2:

$$\lambda = \frac{1.5 - 2\mu^{-1} + 0.5\mu^{-2}}{\delta} + O(\delta^2).$$

Нелинейную проблему (2.11) будем аппроксимировать следующей рациональной проблемой:

$$\left( \frac{1.5 - 2\mu^{-1} + 0.5\mu^{-2}}{\delta} I - \sum_{j=0}^q \mu^{-m_j} L_j \right) \tilde{u} = 0. \quad (2.12)$$

Умножим уравнение (2.12) на  $\mu^{m_q}$ . Полученное уравнение можно преобразовать к следующей полиномиальной проблеме:

$$(\mu^{m_q} I - \mu^{m_q-1} C_1 - \mu^{m_q-2} C_2 - \sum_{j=1}^q \mu^{m_q-m_j} C_{m_j}) \tilde{u} = 0.$$

где

$$C_1 = 2(1.5I - \delta L_0)^{-1}, \quad C_2 = -0.5(1.5I - \delta L_0)^{-1}, \quad (2.13)$$

$$C_{m_j} = (1.5I - \delta L_0)^{-1} \delta L_j, \quad j = 1, \dots, q.$$



Дополним эту проблему тождествами  $\mu^j \tilde{u} = \mu^j \tilde{u}$ ,  $j = 0, \dots, m_q - 1$ . Полученную таким образом систему из  $m_q$  уравнений можно записать в виде:

$$\mu \tilde{x} = M \tilde{x},$$

где

$$\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_{m_q} \end{pmatrix}, \quad M = \begin{pmatrix} M_{11} & \cdots & M_{1m_q} \\ \vdots & & \vdots \\ M_{m_q 1} & \cdots & M_{m_q m_q} \end{pmatrix}. \quad (2.14)$$

Здесь  $\tilde{x}_i = \mu^{m_q - i} \tilde{u}$ , а матрица  $M$  является блочной, блочного порядка  $m_q$  с блоками порядка  $n$ . Все блоки этой матрицы нулевые, кроме поддиагональных блоков  $M_{i+1,i} = I$  ( $i = 1, \dots, m_q - 1$ ) и  $q + 2$  блоков  $M_{11} = C_1$ ,  $M_{12} = C_2$ ,  $M_{1m_j} = C_{m_j}$ ,  $j = 1, \dots, q$ , стоящих в первой блочной строке.

Таким образом мы свели приближенное решение нелинейной проблемы собственных значений (2.11) к вычислению собственных значений матрицы  $M$ . Для вычисления этих собственных значений можно использовать стандартный QR-алгоритм [50]. Приближенные собственные значения проблемы (2.11) выражаются через собственные значения матрицы  $M$  следующим образом:  $\lambda = \ln(\mu)/\delta$ .

Для анализа устойчивости стационарного решения необходимо найти собственное значение проблемы (2.11) с максимальной вещественной частью. Описанный выше алгоритм позволяет найти приближенные собственные значения проблемы (2.11). Для повышения надежности мы будем отбирать из них определенное число ведущих (с максимальными вещественными частями) и уточнять каждое из них методом последовательных линейных проблем [47], который заключается в следующем итерационном процессе:

$$\lambda_{j+1} = \lambda_j - \nu_j.$$

Здесь  $j$  — номер итерации, а  $\nu_j$  — минимальное по абсолютной величине собственное значение линейного матричного пучка  $\nu B(\lambda_j) - A(\lambda_j)$ , где

$$B(\lambda) = \frac{dA}{d\lambda}(\lambda) = I + \sum_{j=1}^q L_j \tau_j e^{-\lambda \tau_j}.$$

В заключение отметим, что порядок матрицы  $M$  в (2.14) обратно пропорционален величине  $\delta$  и пропорционален числу уравнений рассматриваемой модели. Если он окажется слишком большим для того, чтобы можно было за разумное время решить полную проблему собственных значений для матрицы  $M$ , то можно ограничиться решением частичной проблемы собственных значений, вычисляя методом Стюарта [60] для надежности 10–15 максимальных по абсолютной величине собственных значений матрицы  $M$ .

### 2.3 Исследование зависимости стационарных решений от параметров

Пусть в пространстве значений вектора параметров  $\mathbf{p}$  задана некоторая гладкая рационально параметризуемая кривая, не имеющая самопересечений. И требуется найти решения системы (2.1) как функции параметра этой кривой. Будем обозначать параметр через  $s$ , а интервал его варьирования через  $\eta$ , предполагая этот интервал замкнутым. Введем обозначение  $\mathcal{S}(s, u) = \mathcal{G}(u, \mathbf{p}(s))$ . Поскольку  $\mathcal{S}(s, u)$  является рациональной функцией переменных  $s$  и  $u$ , множество

$$A = \{(s, u) : \mathcal{S}(s, u) = 0, s \in \eta, u \geq 0\}$$

является частью некоторой плоской действительной алгебраической кривой [52]. Будем предполагать для простоты, что эта кривая не содержит кратных компонент. Тогда равенство

$$\det \left( \frac{\partial \mathcal{S}}{\partial u}(s, u) \right) = 0 \quad (2.15)$$

может выполняться лишь в конечном числе точек  $(s, u)$ .

Задача состоит в том, чтобы разбить множество  $A$  на подмножества вида

$$C = \{(s, u(s)) : s \in \eta_C\}, \quad (2.16)$$

где  $u(s)$  — однозначная действительная аналитическая функция переменного  $s$ , определенная в некотором интервале  $\eta_C \subset \eta$ , и точки, в которых выполняется равенство (2.15).

Обозначим через  $\mathcal{U}(s)$  полный набор попарно различных решений системы (2.1). Зададим в интервале  $\eta$  сетку с узлами  $s^1 < \dots < s^N$ , где  $s^1$  — начало, а  $s^N$  — конец этого интервала, и найдем множества

$$\mathcal{U}(s^k) = \{u_j^k : j = 1, \dots, d^k\},$$

где  $d^k$  — число попарно различных решений системы (2.1), методом, описанным в разделе 2.1. Будем предполагать, что сетка по  $s$  выбрана настолько мелкой, что каждое из подмножеств вида (2.16), которые необходимо построить, содержит хотя бы одну точку вида  $(s^k, u_j^k)$ . Кроме того, будем предполагать, что во всех таких точках равенство (2.15) не выполняется.

Для построения подмножеств вида (2.16) нам потребуется стандартная процедура численного интегрирования ode45, основанная на методах Рунге-Кутты [53]. Эту процедуру мы будем использовать для трассировки стационарных решений по параметру путем численного решения задач Коши вида

$$u(s^0) = u^0, \quad \frac{du}{ds} = - \left( \frac{\partial \mathcal{S}}{\partial u} \right)^{-1} \frac{\partial \mathcal{S}}{\partial s}. \quad (2.17)$$

Интегрирование будем останавливать, если переменная  $s$  достигнет заданного финального значения, либо, если в процессе интегрирования обнаружится, что функция  $f(s) = \det(\partial \mathcal{S} / \partial u(s, u))$  обращается в ноль на траектории интегрирования. Отметим, что сетку интегрирования по  $s$  процедура ode45 выбирает самостоятельно. Для проверки второго условия останова, сделав шаг из узла  $s_i$  в узел  $s_{i+1}$  этой сетки, процедура ode45 проверяет не принимает ли функция  $f(s)$  значения разных знаков в этих узлах и, если это так, то находит ноль  $s = s_*$  между этими узлами с помощью стандартной процедуры fzero [48], включенной во многие пакеты прикладных программ. При этом для каждого значения  $s$  соответствующее значение  $u(s)$ , необходимое для вычисления  $f(s)$ , вычисляется одним шагом численного интегрирования из узла  $s_i$ . После того, как ноль найден, процедура ode45 останавливается, а точка  $(s_*, u(s_*))$  считается последней точкой траектории интегрирования.

На первом шаге алгоритма построения подмножеств вида (2.16) мы рассмотрим интервал  $s^1 \leq s \leq s^2$ . Используя процедуру численного интегрирования, решим задачи Коши (2.17) с  $u^0 = u_j^1$ ,  $s^0 = s^1$  и финальным значением  $s = s^2$  при всех  $j = 1, \dots, d^1$ , и сформируем двухточечные наборы

$$\mathcal{B}_j = ((s^1, u_j^1), (\tilde{s}_j^1, \tilde{u}_j^1)),$$

где  $(\tilde{s}_j^1, \tilde{u}_j^1)$  означает последнюю точку на траектории численного решения  $j$ -ой задачи Коши. Отметим, что  $\tilde{s}_j^1 < s^2$ , если при решении  $j$ -ой задачи Коши выполнилось второе условие останова, и  $\tilde{s}_j^1 = s^2$  в противном случае. Не нарушая общности будем считать, что второе условие останова выполнилось для первых  $\tilde{d}^1$  задач Коши и не выполнилось для оставшихся  $d^1 - \tilde{d}^1$ . Уточним элементы множества

$$S = (\tilde{u}_{\tilde{d}^1+1}^1, \dots, \tilde{u}_{d^1}^1)$$

элементами множества  $\mathcal{U}(s^2)$ , путем выбора элементов множества  $\mathcal{U}(s^2)$  наиболее близких по норме к элементам множества  $S$ , и заменим во вторых элементах наборов  $\mathcal{B}_{\tilde{d}^1+1}, \dots, \mathcal{B}_{d^1}$  вектора  $\tilde{u}_j^1$  на их уточненные значения.

В результате мы получим двухточечные наборы  $\mathcal{B}_j$  двух видов

$$\mathcal{B}_j = ((s^1, u_j^1), (\tilde{s}_j^1, \tilde{u}_j^1)), j = 1, \dots, \tilde{d}^1, \quad (2.18)$$

где  $s^1 < \tilde{s}_j^1 < s^2$ , и

$$\mathcal{B}_j = ((s^1, u_j^1), (s^2, u_{k_j}^2)), j = \tilde{d}^1 + 1, \dots, d^1. \quad (2.19)$$

Если  $d^1 - \tilde{d}^1 < d^2$ , то  $\tilde{d}^2 = d^2 - d^1 + \tilde{d}^1$  элементов множества  $\mathcal{U}(s^2)$  оказались не использованными для уточнения. Не нарушая общности будем считать, что не использованными оказались элементы  $u_1^2, \dots, u_{\tilde{d}^2}^2$ . Решим численно задачи Коши (2.17) с  $u^0 = u_j^2$ ,  $s^0 = s^2$  и финальным значением  $s = s^1$  при всех  $j = 1, \dots, \tilde{d}^2$ , и сформируем двухточечные наборы

$$\mathcal{B}_{d^1+j} = ((\tilde{s}_j^2, \tilde{u}_j^2), (s^2, u_j^2)), j = 1, \dots, \tilde{d}^2, \quad (2.20)$$

где  $(\tilde{s}_j^2, \tilde{u}_j^2)$  означает последнюю точку траектории интегрирования. Отметим, что при решении каждой из этих задач Коши обязательно выполнится второе условие останова, то есть  $s^2 > \tilde{s}_j^2 > s^1$ .

На этом первый шаг предлагаемого алгоритма построения подмножеств вида (2.16), состоящий в формировании двухточечных наборов (2.18), (2.19) и (2.20), заканчивается. Каждый из полученных наборов состоит из начальной и конечной точки отдельного подмножества в интервале  $s^1 \leq s \leq s^2$ . При этом, каждая точка множества  $s^1 \times \mathcal{U}(s^1)$  является начальной точкой одного из этих наборов, а каждая точка множества  $s^2 \times \mathcal{U}(s^2)$  является конечной точкой одного из этих наборов.

Рисунок 2.1 поясняет первый шаг алгоритма построения подмножеств вида (2.16). На нем изображен случай, когда  $d^1 = d^2 = 4$ ,  $\tilde{d}^1 = \tilde{d}^2 = 2$ . Круги — начальные точки численного интегрирования, квадраты — конечные. Толстые линии связывают конечные точки численного интегрирования с их уточнениями. В результате было сформировано два двухточечных набора вида (2.18):

$$\mathcal{B}_1 = ((s^1, u_1^1), (\tilde{s}_1^1, \tilde{u}_1^1)), \mathcal{B}_2 = ((s^1, u_2^1), (\tilde{s}_2^1, \tilde{u}_2^1)),$$

два двухточечных набора вида (2.19):

$$\mathcal{B}_3 = ((s^1, u_3^1), (s^2, u_3^2)), \mathcal{B}_4 = ((s^1, u_4^1), (s^2, u_4^2)),$$

и два двухточечных набора вида (2.20):

$$\mathcal{B}_5 = ((\tilde{s}_1^2, \tilde{u}_1^2), (s^2, u_1^2)), \mathcal{B}_6 = ((\tilde{s}_2^2, \tilde{u}_2^2), (s^2, u_2^2)).$$

Далее, мы точно таким же образом строим двухточечные наборы для интервалов  $s^k \leq s \leq s^{k+1}$ ,  $k = 2, \dots, N-1$ . На этом построение двухточечных наборов заканчивается и к полученным наборам начинает применяться следующая процедура склейки: наборы  $(\alpha_1, \dots, \alpha_k)$  и  $(\beta_1, \dots, \beta_l)$  заменяются одним набором  $(\alpha_1, \dots, \alpha_k, \beta_2, \dots, \beta_l)$ , если  $\alpha_k = \beta_1$  и в этой точке не выполняется равенство (2.15).

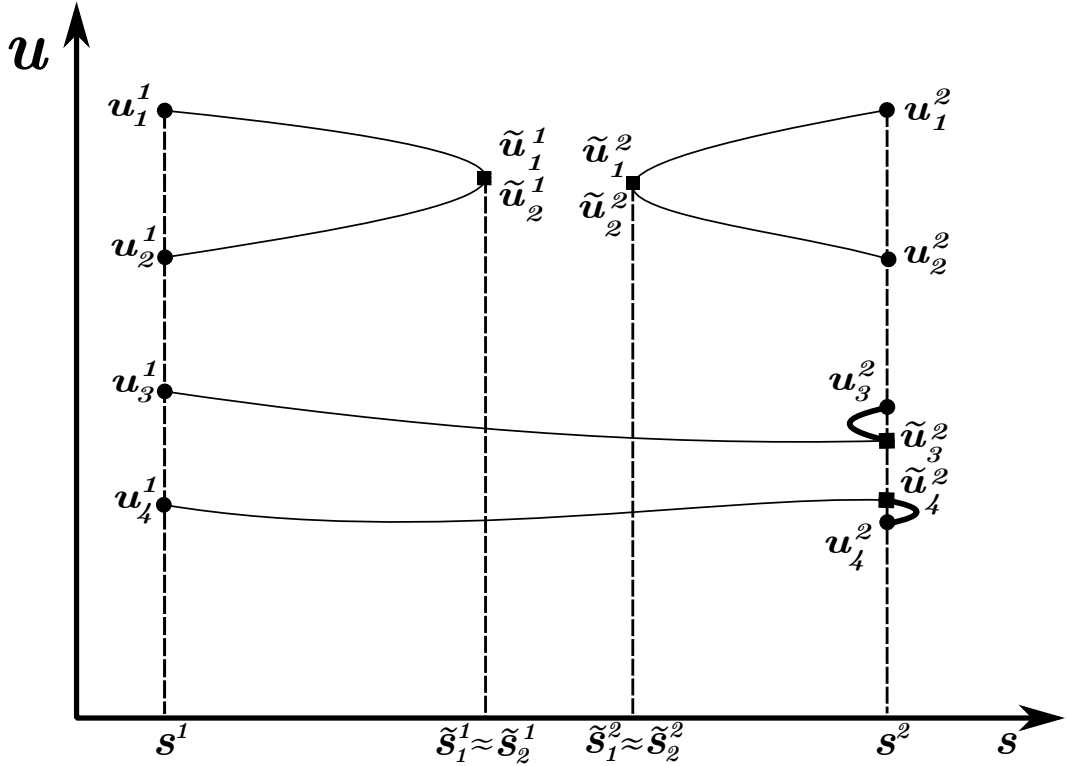


Рисунок 2.1 — Иллюстрация работы первого шага алгоритма построения подмножеств вида (2.16)

Процедура применяется до тех пор, пока остаются наборы, которые можно склеить. В результате мы получим некоторое множество наборов вида

$$\mathcal{C} = ((s_1, u_1), \dots, (s_r, u_r)), \quad (2.21)$$

где  $2 \leq r \leq N$ ,  $s^1 \leq s_1 < \dots < s_r \leq s^N$ . Каждый из этих наборов — это набор точек, с высокой точностью расположенных на отдельном подмножестве вида (2.16), то есть  $u_j \approx u(s_j)$ . Для приближенного построения самого этого подмножества останется построить по этим точкам какой-либо сплайн.

Напомним, что описанный выше алгоритм гарантированно находит все подмножества вида (2.16) множества  $A$ , если сетка  $s^1 < \dots < s^N$  выбрана такой, что каждое из этих подмножеств, содержит хотя бы одну точку вида  $(s^k, u_j^k)$  и во всех таких точках равенство (2.15) не выполняется. Учитывая, что число подмножеств конечно и число точек множества  $A$ , в которых равенство (2.15) выполняется, так же конечно, эти условия несложно удовлетворить выбирая достаточно мелкую сетку по  $s$  и немного сдвигая узел  $s^k$ , если в точке  $(s^k, u_j^k)$  выполняется равенство (2.15).

## 2.4 Определение оптимального возмущения стационарного решения

Для вектора переменных системы (2.9) введем в рассмотрение следующее семейство локальных норм в момент времени  $t$ :

$$\|w\|_{D,\rho,t} = \left( \int_{t-\tau_q}^t \left( \|Dw(\xi)\|_2^2 + \rho \left\| D \frac{dw}{d\xi}(\xi) \right\|_2^2 \right) d\xi \right)^{1/2}, \quad (2.22)$$

где  $D$  — заданная положительно определенная диагональная матрица порядка  $n$ ,  $\|\cdot\|_2$  — вторая (евклидова) векторная норма,  $\rho$  — неотрицательный параметр. В случае  $\rho = 0$  мы будем для краткости называть эту норму  $L_2$ -нормой, а в случае  $\rho = 1$  —  $W_2^1$ -нормой. Стоит отметить, что решение  $w(t)$  может иметь разрыв первого рода в  $t = 0$ , но это не существенно ни при определении семейства локальных норм, ни в рамках дальнейших рассуждений.

Под оптимальным возмущением  $w_{\text{opt}}(t)$  будем понимать решение системы (2.9), обеспечивающее максимальную амплификацию (подскок) локальной нормы (2.22) возмущения по сравнению с ее первоначальным значением, то есть такое решение  $w(t)$ , на котором достигается максимум величины

$$\max_{t \geq 0} \frac{\|w\|_{D,\rho,t}}{\|w\|_{D,\rho,0}}.$$

Оптимальное возмущение по определению является некоторым решением линейной системы (2.9) и, следовательно, полностью определяется своим значением в интервале  $-\tau_q \leq t \leq 0$ . Поэтому, при построении оптимального возмущения наряду с выбором нормы, в которой проводится оптимизация, принципиальным является вопрос о том, из какого пространства функций, заданных в интервале  $-\tau_q \leq t \leq 0$ , мы берем начальные значения. Это пространство функций  $[-\tau_q, 0] \rightarrow \mathbf{R}^n$  мы далее будем обозначать через  $\mathcal{Q}$ . Для гарантированного существования максимума необходимо, чтобы  $\mathcal{Q}$  было полным пространством в норме  $\|\cdot\|_{D,\rho,0}$ . Для этого достаточно выбирать в качестве  $\mathcal{Q}$  линейную оболочку какого-либо конечного набора кусочно-непрерывных базисных функций.

Найденное оптимальное возмущение используют для возмущения стационарного решения исходной нелинейной системы (1.1), выбирая в качестве начального значения

$$u(t) = \bar{u} + \varepsilon \tilde{w}_{\text{opt}}(t) \quad (2.23)$$

при  $-\tau_q \leq t \leq 0$ , где  $\tilde{w}_{\text{opt}}(t)$  означает нормированное в  $L_2$  норме оптимальное возмущение, а  $\varepsilon$  — вещественный параметр. Варьируя абсолютную величину этого параметра, можно увеличивать или уменьшать начальное возмущение. В зависимости же от его знака, заданная компонента решения при  $t > 0$  либо начинает нарастать, либо убывать.

## 2.5 Базовый алгоритм

Искать оптимальное возмущение удобнее в два этапа. Сначала вычисляем максимальную амплификацию

$$\Gamma(t) = \max_w \frac{\|w\|_{D,\rho,t}}{\|w\|_{D,\rho,0}} \quad (2.24)$$

локальной нормы решения системы (2.9), где максимум берется по всем решениям, начальные значения которых ненулевые и принадлежат  $\mathcal{Q}$ , и находим  $t = t_{\text{opt}}$ , при котором функция  $\Gamma(t)$  достигает максимального значения. Если таких  $t$  несколько, то для определенности берем из них минимальное. Таким образом,

$$t_{\text{opt}} = \min \arg \max_{t \geq 0} \Gamma(t).$$

Затем находим

$$w_{\text{opt}} \in \arg \max_w \frac{\|w\|_{D,\rho,t_{\text{opt}}}}{\|w\|_{D,\rho,0}}. \quad (2.25)$$

Если  $D$ ,  $\rho$ , и  $\mathcal{Q}$  фиксированы, то любое найденное оптимальное возмущение обеспечивает одну и ту же максимальную амплификацию локальной нормы решения. Обычно максимальная амплификация имеет только одну точку глобального максимума, а решение оптимизационной задачи (2.25) однозначно с точностью до ненулевой мультипликативной константы.



Оптимальные возмущения можно вычислять на основе любой разностной схемы, подходящей для решения задач Коши для систем обыкновенных дифференциальных уравнений с запаздывающим аргументом, в том числе неявного метода Эйлера (1.3) и неявной схемы второго порядка BDF2 (1.4). Как и в разделе 2.2, мы опишем алгоритмы, использующие неявную схему второго порядка BDF2 на равномерной сетке в качестве примера. После дискретизации система (2.9) принимает вид

$$\frac{1.5w_k - 2w_{k-1} + 0.5w_{k-2}}{\delta} = \sum_{j=0}^q L_{\tau_j} w_{k-m_j}, \quad (2.26)$$

где  $w_k$  — сеточная функция, аппроксимирующая  $w(t_k)$ . В качестве начальных данных для решения разностной задачи Коши нужно задать значения  $w_{-m_q+1}, \dots, w_0$ .

Предполагая шаг  $\delta$  достаточно малым, запишем уравнение (2.26) в виде

$$w_k = C_1 w_{k-1} + C_2 w_{k-2} + \sum_{j=1}^q C_{m_j} w_{k-m_j}, \quad (2.27)$$

где матрицы  $C_1, C_2, C_{m_1}, \dots, C_{m_q}$  такие же, как в (2.13), и дополним уравнение (2.27) тождествами  $w_i = w_i, i = k-1, \dots, k-m_q+1$ . Полученную таким образом систему из  $m_q$  уравнений можно записать в виде

$$x_k = M x_{k-1}, \quad (2.28)$$

где

$$x_k = \begin{pmatrix} w_k \\ \vdots \\ w_{k-m_q+1} \end{pmatrix} \quad (2.29)$$

и  $M$  та же самая, что и в (2.14).

Чтобы ввести сеточный аналог максимальной амплификации (2.24), необходимо определить сеточный аналог нормы (2.22). Для численного интегрирования первого слагаемого под интегралом в (2.22) будем использовать формулу трапеций, а для интегрирования второго — формулу прямоугольников. При этом производную  $dw/d\xi$ , входящую во второе слагаемое, будем аппроксимиро-

вать в точках  $t_{k+1/2}$  с помощью центральной разности:

$$\left(\frac{dw}{d\xi}\right)_{k+1/2} = \frac{w_{k+1} - w_k}{\delta}.$$

Можно показать, что полученный таким образом сеточный аналог локальной нормы (2.22) в точке  $t_k$  может быть записан в следующем виде:

$$\|Hx_k\|_2, \quad (2.30)$$

где  $H = U \otimes D$  — квадратная матрица порядка  $nm_q$ ,  $U$  — верхний треугольный фактор разложения Холецкого симметричной положительно определенной трехдиагональной матрицы

$$\frac{1}{\delta} \begin{pmatrix} \delta^2/2 + \rho & -\rho & & & & \\ -\rho & \delta^2 + 2\rho & -\rho & & & \\ & \ddots & \ddots & \ddots & & \\ & & & -\rho & \delta^2 + 2\rho & -\rho \\ & & & & -\rho & \delta^2/2 + \rho \end{pmatrix}$$

порядка  $m_q$ , а  $\otimes$  — символ Кронекерова произведения.

В силу (2.28), (2.29), (2.30), сеточный аналог  $\Gamma_k$  максимальной амплификации (2.24) нормы решения можно записать следующим образом:

$$\Gamma_k = \max_{x_0 \in \text{span}(Q) \setminus \{0\}} \frac{\|HM^k x_0\|_2}{\|Hx_0\|_2},$$

где  $Q$  — матрица, столбцы которой образуют базис в сеточном аналоге подпространства  $\mathcal{Q}$ , а  $\text{span}(Q)$  означает линейную оболочку столбцов матрицы  $Q$ . Для определенности будем предполагать, что базисные функции для каждой переменной, входящей в вектор  $w$ , выбраны одинаковыми и их количество равно  $d$ , где  $d \leq m_q$ . Обозначим через  $G$  матрицу размера  $m_q \times d$ , столбцы которой являются сеточными аналогами этих базисных функций. Тогда в качестве  $Q$  можно выбрать матрицу  $G \otimes I$  размера  $nm_q \times nd$ .

Учитывая, что  $x_0 = Q\xi$ , где  $\xi$  — некоторый  $nd$ -компонентный вектор, имеем

$$Hx_0 = HQ\xi = (U \otimes D)(G \otimes I)\xi = (UG \otimes D)\xi.$$

Пусть  $UG = \widehat{Q}\widehat{R}$  — QR-разложение, где  $\widehat{Q}$  — ортогональная прямоугольная матрица размера  $m_q \times d$ , а  $\widehat{R}$  — верхняя треугольная квадратная матрица порядка  $d$ . Вводя обозначения  $\widetilde{Q} = \widehat{Q} \otimes I$ ,  $\widetilde{R} = \widehat{R} \otimes D$ , имеем  $Hx_0 = \widetilde{Q}\widetilde{R}\xi = \widetilde{Q}\widetilde{\xi}$ , где  $\widetilde{\xi} = \widetilde{R}\xi$ . Следовательно,

$$\frac{\|HM^k x_0\|_2}{\|Hx_0\|_2} = \frac{\|HM^k H^{-1}Hx_0\|_2}{\|Hx_0\|_2} = \frac{\|HM^k H^{-1}\widetilde{Q}\widetilde{\xi}\|_2}{\|\widetilde{Q}\widetilde{\xi}\|_2} = \frac{\|HM^k H^{-1}\widetilde{Q}\widetilde{\xi}\|_2}{\|\widetilde{\xi}\|_2},$$

и, таким образом имеем

$$\Gamma_k = \max_{\widetilde{\xi} \neq 0} \frac{\|HM^k H^{-1}\widetilde{Q}\widetilde{\xi}\|_2}{\|\widetilde{\xi}\|_2} = \|HM^k H^{-1}\widetilde{Q}\|_2.$$

Для повышения эффективности вычисления  $\Gamma_k$  вместо матрицы  $H^{-1}\widetilde{Q}$  будем использовать равную ей матрицу  $Q\widetilde{R}^{-1}$ . Тогда для вычисления  $\Gamma_k$  окончательно получим следующую формулу:

$$\Gamma_k = \|A_k\|_2,$$

где

$$A_k = HM^k Q\widetilde{R}^{-1}. \quad (2.31)$$

Таким образом, вычисление  $\Gamma_k$  сводится к формированию матрицы  $Y_0 = Q\widetilde{R}^{-1}$  и вычислению матриц  $Y_k$  по рекуррентной формуле  $Y_k = MY_{k-1}$  с одновременным вычислением норм матриц  $HY_k$  для всех натуральных  $k$ , не превосходящих  $N = [T/\delta]$ , где  $T$  — произвольная априорная верхняя оценка  $t_{\text{opt}}$ . При этом, учитывая блочную структуру матрицы  $M$ , ее можно умножать на матрицу  $Y_{k-1}$  следующим образом:

$$Y_{k1} = C_1 Y_{k-1,1} + C_2 Y_{k-1,2} + \sum_{j=1}^q C_{m_j} Y_{k-1,m_j}, \quad Y_{ki} = Y_{k-1,i-1}, \quad i = 2, \dots, m_q,$$

где  $Y_{li}$  — матрицы размера  $n \times nd$ , являющиеся блочными строками матрицы

$$Y_l = \begin{pmatrix} Y_{l1} \\ \vdots \\ Y_{lm_q} \end{pmatrix}.$$

Пусть  $k_{\text{opt}}$  означает значение  $k$ , при котором достигается максимум  $\Gamma_k$ . Вычислив нормированный правый сингулярный вектор  $\eta^{\text{opt}}$  матрицы  $A_{k_{\text{opt}}}$ , отвечающий ее максимальному сингулярному числу [50], начальное значение  $x_0^{\text{opt}}$  сеточного аналога оптимального возмущения можно найти по формуле  $x_0^{\text{opt}} = Q\tilde{R}^{-1}\eta^{\text{opt}}$ .

Отметим, что величина шага сетки по времени, необходимая для достаточно точного численного интегрирования системы уравнений (2.9), а значит и вычисления максимальной амплификации  $\Gamma(t)$ , значительно меньше, чем это требуется для последующего вычисления с приемлемой точностью максимума  $\Gamma(t)$  и оптимального возмущения, на котором достигается этот максимум. Поэтому, для сокращения вычислительных затрат базового алгоритма имеет смысл вычислять  $\|A_k\|_2$  не для всех натуральных  $k$ , не превосходящих  $N = [T/\delta]$ , а лишь для  $k$  кратных  $l$ , где  $l$  — некоторое заданное натуральное число.

Описанный алгоритм можно немного упростить, включив матрицу  $D$  в матрицу  $M$ . Действительно, правая часть равенства (2.31) не изменится, если в матрицах  $H$  и  $\tilde{R}$  заменить матрицу  $D$  единичной, а блоки  $C_i$  матрицы  $M$  заменить на  $DC_iD^{-1}$ . Мы будем считать, что такое предварительное преобразование матрицы  $M$  выполнено и матрица  $D$ , фигурирующая в матрицах  $H$  и  $\tilde{R}$ , — единичная.

Чтобы учесть тот факт, что некоторые переменные могут не варьироваться при вычислении оптимального возмущения, матрица  $Q$  формируется следующим образом:

$$Q = G \otimes I_v, \quad (2.32)$$

где  $I_v = [E_{j_1}, \dots, E_{j_{n_v}}]$ ,  $E_j$  обозначает  $j$ -й столбец единичной матрицы порядка  $n$ , а  $j_1, \dots, j_{n_v}$  ( $n_v \leq n$ ) обозначает номера варьируемых переменных.

Чтобы обеспечить максимизацию только по определенным переменным при вычислении максимальной амплификации, матрица  $A_k$  должна быть модифицирована следующим образом:

$$A_k = \left( I_{m_q} \otimes \hat{I} \right) H M^k Q \tilde{R}^{-1} = \hat{H} M^k Q \tilde{R}^{-1},$$

где  $\hat{I}$  представляет собой квадратную диагональную матрицу порядка  $n$ , диагональные элементы которой с номерами, соответствующими максимизируемым

переменным, равны единице и равны нулю в противном случае и

$$\widehat{H} = (I_{m_q} \otimes \widehat{I})(U \otimes D) = (U \otimes \widehat{I}D).$$

## 2.6 Метод Ланцоша

Учитывая, что  $M$  и остальные матрицы, образующие  $A_k$  в (2.31), являются разреженными матрицами большой размерности, для вычисления максимального сингулярного числа матрицы  $A_k$  и отвечающего ему правого сингулярного вектора можно использовать метод Ланцоша [56], применяя его к матрице  $A_k^T A_k$  для вычисления ее наибольшего собственного значения, корень из которого даст искомое сингулярное число, а собственный вектор матрицы  $A_k^T A_k$ , отвечающий ее максимальному собственному значению, — искомый сингулярный вектор. Формальная схема алгоритма приведена ниже.

Метод Ланцоша строит последовательность векторов  $q_1, q_2, \dots, q_r$ , образующих ортонормированный базис в подпространстве Крылова

$$K_r(A_k^T A_k, q_1) = \text{span} \left( q_1, A_k^T A_k q_1, \dots, (A_k^T A_k)^{r-1} q_1 \right).$$

Пусть  $T_r = \text{tridiag}(\beta_{i-1}, \alpha_i, \beta_i)$  означает вещественную симметричную трехдиагональную матрицу с диагональными и внедиагональными элементами соответственно  $\alpha_i = q_i^T A_k^T A_k q_i$  и  $\beta_i = q_i^T A_k^T A_k q_{i+1}$ . Тогда последовательность  $\lambda_{\max}(T_r)$ , где  $\lambda_{\max}$  означает максимальное собственное значение, не убывает (см., например, [56]) и стремится к наибольшему собственному значению матрицы  $A_k^T A_k$ . Вместе с тем, увеличение  $r$  ведет к увеличению объема данных, которые необходимо хранить, и времени вычислений. Чтобы избежать лишних вычислительных затрат, мы останавливаем алгоритм, когда рост  $\sqrt{\lambda_{\max}(T_r)}$  становится слишком медленным или когда  $r$  достигает заданного максимального значения  $r_{\max}$ . Вектор  $v = [q_1, q_2, \dots, q_r] y_r$ , где  $y_r$  — это собственный вектор, отвечающий  $\lambda_{\max}(T_r)$ , выбираем в качестве искомого приближенного правого сингулярного вектора, отвечающего максимальному сингулярному числу матрицы  $A_k^T A_k$ . В отличие от базового алгоритма, в котором матрицы  $A_k$  формируются явно, его

---

**Алгоритм 1**

Вычисление наибольшего собственного значения матрицы  $A_k^T A_k$  и отвечающего ему собственного вектора методом Ланцоша.

---

**Input:** матрицы  $H, M, Q, \tilde{R}$  в разреженном формате, натуральное число  $k$ , начальный вектор  $v$ , минимальное допустимое относительное увеличение  $\text{tol}$  приближенного сингулярного числа за один шаг, максимальное число шагов  $r_{\max}$  метода Ланцоша.

**Output:** приближенные значения наибольшего сингулярного числа  $s$  и соответствующего правого сингулярного вектора  $v$  матрицы  $A_k$ , определенной в (2.31).

**if**  $r_{\max} > 1$  **then**

$$s_{-1} = s_0 = 0, \quad q_0 = 0, \quad \beta_0 = \|v\|_2, \quad r = 0$$

**while**  $r < r_{\max}$  and  $\beta_r > 0$  and  $s_r \geq (1 + \text{tol})s_{r-1}$  **do**

$$r := r + 1, \quad q_r = v / \beta_{r-1}$$

$$w = A_k^T A_k q_r - \beta_{r-1} q_{r-1}, \quad \alpha_r = q_r^T w$$

$$v = w - \alpha_r q_r, \quad \beta_r = \|v\|_2$$

Сформировать  $r \times r$  трехдиагональную матрицу  $T_r = \text{tridiag}(\beta_{i-1}, \alpha_i, \beta_i)$

$$s_r = \sqrt{\lambda_{\max}(T_r)}$$

**end while**

Вычислить собственный вектор  $y$ , отвечающий максимальному собственному значению матрицы  $T_r$ .

$$v = [q_1, \dots, q_r]y$$

**end if**

Выполнить один шаг степенного метода:

$$w = A_k v, \quad v = w / \|w\|_2$$

$$w = A_k^T v, \quad s = \|w\|_2, \quad v = w / s$$


---

модификация, использующая метод Ланцоша, требует лишь умножения сомножителей, образующих матрицы  $A_k$  и  $A_k^T$ , на вектор, что позволяет обойтись значительно меньшим объемом рабочей памяти.

Для сокращения вычислительных затрат, в качестве начального вектора при  $k > 1$  можно брать сингулярный вектор, найденный при предыдущем значении  $k$ . Кроме того, как и в случае базового алгоритма, имеет смысл вычислять  $\|A_k\|_2$  не при всех натуральных  $k$ , не превосходящих  $N = \lceil T/\delta \rceil$ , а лишь при  $k$  кратных  $l$ , где  $l$  — некоторое заданное натуральное число.

## 2.7 Метод последовательной максимизации

Алгоритм последовательной максимизации принципиально отличается от базового алгоритма и его модификации, описанной в предыдущем разделе, и состоит в следующем. Положим  $k_1 = \lfloor N/2 \rfloor$  и найдем максимальное сингулярное число матрицы  $A_{k_1}$  и отвечающий ему нормированный правый сингулярный вектор  $\eta$  с помощью метода Ланцоша, описанного выше. Вычислив последовательно нормы  $\|A_1\eta\|_2, \dots, \|A_N\eta\|_2$ , положим

$$k_2 = \min \arg \max_{k \geq 1} \|A_k\eta\|_2.$$

Повторим вышеописанную процедуру с  $k_2$  вместо  $k_1$  и так далее до тех пор пока  $k_i$  не совпадет с  $k_{i-1}$ . Это значение  $k_i$  и будет  $k_{\text{opt}}$ , причем по завершении процедуры мы одновременно получим максимальное сингулярное число матрицы  $A_{k_{\text{opt}}}$  и отвечающий ему нормированный правый сингулярный вектор  $\eta^{\text{opt}}$ . Также, как и в модификации базового алгоритма с помощью метода Ланцоша, в алгоритме последовательной максимизации матрицы  $A_k$  и  $A_k^T$  не формируются явно, а используется только умножение этих матриц на вектор. Формальная схема описанного алгоритма приведена ниже.

Для сокращения вычислительных затрат в алгоритме последовательной максимизации можно вычислять нормы  $\|A_k\eta\|_2$  не при всех  $k$ , а лишь при  $k$  кратных  $l$ , где  $l$  — некоторое заданное натуральное число, однако выигрыш от этого будет не столь значительным, как в двух предыдущих алгоритмах, поскольку в данном случае требуется вычислять нормы векторов, а не матриц.

## 2.8 Оценка сложности алгоритмов

### 2.8.1 Теоретическая оценка

Оценим главные члены числа арифметических операций и ячеек памяти, необходимые каждому из трех алгоритмов, описанных в предыдущих разде-

---

**Алгоритм 2**

Вычисление оптимального возмущения методом последовательной максимизации.

---

**Input:** матрицы  $H$ ,  $M$ ,  $Q$ ,  $\tilde{R}$  в разреженном формате, натуральное число  $N$ , параметры  $\text{tol}$  и  $r_{\max}$  Алгоритма 1.

**Output:** приближенное значение сеточного аналога оптимального возмущения  $x_0^{\text{opt}}$ .

$$k_0 = 0$$

$$k_1 = \lceil N/2 \rceil$$

$$i = 1$$

**while**  $k_i \neq k_{i-1}$  **do**

Найти максимальное сингулярное число матрицы  $A_{k_i}$  и отвечающий ему нормированный правый сингулярный вектор  $\eta$  с помощью алгоритма 1.

Вычислить нормы  $\|A_1\eta\|_2, \dots, \|A_N\eta\|_2$  и найти  $k_{i+1} = \min \arg \max_{k \geq 1} \|A_k\eta\|_2$ .

$$i := i + 1$$

**end while**

$$x_0^{\text{opt}} = Q\tilde{R}^{-1}\eta$$


---

лах. Под ячейкой памяти будем понимать 64-х битовое вещественное число с плавающей точкой. Учитывая, что основные вычислительные затраты базового алгоритма, его модификации, использующей метод Ланцоша, приходятся на вычисление максимальной амплификации, затраты на последующее вычисление оптимального возмущения мы для этих алгоритмов учитывать не будем.

В соответствии со сказанным в конце раздела 2.5, будем считать, что блоки матрицы  $M$  преобразованы преобразованием подобия с матрицей  $D$ , а матрицу  $D$  в матрицах  $H$  и  $\tilde{R}$  будем считать единичной порядка  $n$ . Поскольку матрица  $U$  является верхним треугольным фактором разложения Холецкого симметричной положительно определенной трехдиагональной матрицы порядка  $m_q$ , количество ненулевых элементов в этой матрице есть  $2m_q - 1$ . Поэтому, для хранения матрицы  $U$  достаточно  $2m_q$  ячеек памяти, а для умножения на вектор матрицы  $H = U \otimes I$  без ее формирования —  $3nm_q$  арифметических операций. Для хранения ненулевых и не единичных блоков матрицы  $M$  достаточно  $(q + 2)n^2$  ячеек памяти, а для умножения этой матрицы на вектор —  $2(q + 2)n^2$  арифметических операций. Для хранения  $m_q \times d$  матрицы  $G$  достаточно  $m_q d$  ячеек памяти, а для умножения матрицы  $Q = G \otimes I$  на вектор без ее формирования —  $2dnm_q$  арифметических операций. Наконец, поскольку  $\hat{R}$  — верхняя



треугольная квадратная матрица порядка  $d$ , для ее хранения требуется  $d^2/2$  ячеек памяти, а для решения системы с этой матрицей достаточно  $d^2$  арифметических операций. Поэтому, для умножения матрицы  $\tilde{R}^{-1} = \hat{R}^{-1} \otimes I$  на вектор без ее формирования достаточно  $nd^2$  арифметических операций.

Таким образом, для умножения на вектор матриц  $H$ ,  $M$ ,  $Q$  и  $\tilde{R}^{-1}$  без их формирования достаточно соответственно  $3nm_q$ ,  $2(q+2)n^2$ ,  $2nm_qd$  и  $nd^2$  арифметических операций, а для хранения необходимой для этого информации об этих матрицах достаточно соответственно  $2m_q$ ,  $(q+2)n^2$ ,  $m_qd$  и  $d^2/2$  ячеек памяти.

Базовый алгоритм сначала формирует  $nm_q \times nd$  матрицу  $Y_0 = Q\tilde{R}^{-1} = G\hat{R}^{-1} \otimes I$ . Для этого достаточно  $m_qd^2$  арифметических операций. Далее, на каждом  $k$ -ом шаге он вычисляет матрицу  $Y_k = MY_{k-1}$ , для чего достаточно  $2(q+2)n^3d$  арифметических операций. Если  $k$  кратно  $l$ , то на  $k$ -ом шаге вычисляется матрица  $A_k = HY_k$ , для чего достаточно  $3n^2m_qd$  арифметических операций, и ее норма. Для вычисления последней с помощью R-SVD разложения [50] достаточно  $2n^3m_qd^2 + 2n^3d^3$  арифметических операций. Так как формирование матрицы  $Q\tilde{R}^{-1}$  производится один раз и, следовательно, требует небольшого по сравнению с другими шагами числа операций, главный член суммарного числа арифметических операций базового алгоритма составляет

$$2Nn^3d \left( q + 2 + \frac{m_qd}{l} \right). \quad (2.33)$$

Здесь и далее предполагается, что  $1 \ll d \ll m_q$ ,  $m_q \ll N$ ,  $q$  — небольшое натуральное число,  $1 \leq n \ll m_q$ .

Для реализации базового алгоритма необходимо хранить текущую матрицу  $Y_k$ , которая является плотной и требует для хранения  $n^2m_qd$  ячеек памяти. Это значительно больше числа ячеек памяти, необходимых для хранения информации о матрицах  $H$  и  $M$ , необходимой для их умножения на вектор. Поэтому главный член числа ячеек памяти, необходимых для реализации базового алгоритма —  $n^2m_qd$ .

Перейдем к оценке вычислительных затрат модификации базового алгоритма, использующей метод Ланцоша. Из выполненного выше анализа следует, что для умножения матрицы  $A_k$  на вектор  $q_r$  в главном члене достаточно  $2nm_qd + 2k(q+2)n^2$  арифметических операций. Столько же арифметических

операций достаточно и для умножения на вектор матрицы  $A_k^T$ . Остальные вычислительные затраты  $r$ -го шага метода Ланцоша пренебрежимо малы, кроме может быть вычисления максимального собственного значения симметричной трехдиагональной матрицы  $T_r$ , которое требует  $O(r^2)$  арифметических операций с небольшой мультипликативной константой [50]. Однако, как показывает вычислительная практика, для нахождения максимального сингулярного числа матрицы  $A_k$  и соответствующего ему правого сингулярного вектора в среднем достаточно не более полутора десятков шагов метода Ланцоша. Это число шагов плюс один шаг степенного метода мы будем обозначать через  $r_{av}$ . Поэтому главный член числа арифметических операций рассматриваемого алгоритма составляет

$$2r_{av} \left( 2\frac{N}{l}nm_qd + \frac{N^2}{l}(q+2)n^2 \right). \quad (2.34)$$

Для его реализации необходимо хранить матрицы  $H$ ,  $M$ ,  $\tilde{R}^{-1}$ ,  $Q$ , а также векторы  $q_1, \dots, q_{r_{av}}$ , что требует  $ndr_{av}$  ячеек памяти. Так как в силу сделанных ранее предположений количество ячеек памяти, необходимых для хранения информации о матрицах  $Q$  и  $M$ , намного больше чем количество ячеек памяти, необходимое для хранения аналогичной информации о матрицах  $H$ ,  $\tilde{R}^{-1}$ , а также для хранения вышеупомянутых векторов, главный член числа ячеек памяти, достаточного для реализации модификации базового алгоритма, использующей метод Ланцоша, составляет  $m_qd + (q+2)n^2$ .

Оценим теперь число арифметических операций, необходимое алгоритму последовательной максимизации. Каждая итерация этого алгоритма включает в себя нахождение максимального сингулярного числа и соответствующего ему правого сингулярного вектора матрицы  $A_{k_i}$  с помощью метода Ланцоша, что, согласно полученным выше оценкам, можно выполнить за  $4r_{av}(nm_qd + k_i(q+2)n^2)$  арифметических операций. Также каждая итерация требует вычисления  $\|A_l\eta\|_2, \|A_{2l}\eta\|_2, \dots, \|A_N\eta\|_2$ . Для этого, так как произведение  $Q\tilde{R}^{-1}\eta$  вычисляется один раз, в главном члене достаточно  $(2(q+2)n^2 + 5nm_q/l)N$  арифметических операций. Следовательно, главный член числа арифметических операций алгоритма последовательной максимизации составляет

$$g \left( 4r_{av} (nm_qd + (q+2)n^2k_{av}) + \left( 2(q+2)n^2 + \frac{5nm_q}{l} \right) N \right). \quad (2.35)$$

Здесь

$$k_{av} = \frac{1}{g} \sum_{i=1}^q k_i,$$

где  $g$  — число итераций этого алгоритма. Число ячеек памяти, необходимых для реализации метода последовательной максимизации очевидно совпадает с числом ячеек памяти, необходимых для реализации модификации базового алгоритма, использующей метод Ланцоша, то есть в главном члене составляет  $m_q d + (q + 2)n^2$ .

Из выполненного анализа в частности следует, что с ростом числа уравнений  $n$ , число ячеек памяти, необходимых для реализации базового алгоритма, значительно больше, чем требуют два других алгоритма. Оценки числа операций (2.33), (2.34) и (2.35) позволяют заключить, что в этом случае базовый алгоритм проигрывает и по числу арифметических операций, так как с увеличением  $n$  оно растет как  $n^3$ , а для модификации базового алгоритма и алгоритма последовательной максимизации — как  $n^2$ . Кроме того, при достаточно большом числе  $d$  базисных функций число операций базового алгоритма так же значительно больше, чем у остальных алгоритмов, так как с увеличением  $d$  оно растет как  $d^2$ , а для остальных алгоритмов — линейно. Наконец, поскольку  $m_q \sim 1/\delta$  и  $N \sim 1/\delta$ , где  $\delta$  — шаг сетки по времени, число операций каждого алгоритма пропорционально  $1/\delta^2$ . При этом константа пропорциональности базового алгоритма, его модификации, использующей метод Ланцоша, и алгоритма последовательной максимизации равна  $2Tn^3 d^2 \tau_q / l$ ,  $2r_{av}(2Tn\tau_q d + T^2(q + 2)n^2) / l$  и  $5Tn\tau_q / l$  соответственно. То есть, она значительно меньше у алгоритма последовательной максимизации.

Таким образом, в случае систем с большим числом переменных и/или базисных функций и/или малых шагов по времени эффективнее модификация базового алгоритма, использующая метод Ланцоша, и еще более эффективен алгоритм последовательной максимизации.

## 2.8.2 Практическая оценка

Проиллюстрируем работу базового алгоритма, его модификации, использующей метод Ланцоша, и алгоритма последовательной максимизации, которые далее для краткости будем называть соответственно алгоритмами I, II и III, на примере модели LCMV. При значениях параметров, указанных в таблице 5 из [42], кроме  $\beta = 0.102$ ,  $\alpha_{E_e} = 0.3$  и  $\alpha_{E_p} = 0.006$ , данная модель имеет следующее устойчивое стационарное решение:  $V \approx 4.8 \cdot 10^7$ ,  $E_p \approx 1.0$ ,  $E_e \approx 1.8 \cdot 10^{-4}$ ,  $W \approx 4.4 \cdot 10^8$ . В качестве базисных функций для вычисления оптимальных возмущений будем использовать следующие функции, качественно аппроксимирующие поведение препаратов в рамках однокамерных и двухкамерных фармакокинетических моделей:

$$\psi(t, t_j) = \begin{cases} 0, & -\tau_q \leq t < t_j, \\ \exp\{-3(t - t_j)\} - \exp\{-9(t - t_j)\}, & t_j \leq t \leq 0, \end{cases} \quad (2.36)$$

где  $j = -m_q + [1/\delta]i + 1$ ,  $i = 0, 1, \dots, d - 1$ , где  $d$  — число базисных функций.

Для расчетов будем использовать равномерную сетку с шагом  $\delta = 5 \cdot 10^{-3}$ . В качестве априорной верхней оценки  $t_{\text{opt}}$  выберем  $T = 30$ . Таким образом, будем иметь  $n = 4$ ,  $q = 2$ ,  $N = 6000$ ,  $m_q = 1120$ . Для нахождения максимального сингулярного числа и отвечающего ему правого сингулярного вектора методом Ланцоша выберем  $\text{tol} = 10^{-9}$ , а в качестве начального вектора  $v$  всегда будем выбирать случайный вектор.

Рассмотрим четыре тестовые задачи, которые в дальнейшем будем называть тестами 1–4. Между собой эти тесты отличаются значением параметра  $d$ , видом локальной нормы, в которой вычисляется оптимальное возмущение, и весами локальной нормы (диагональные элементы матрицы  $D$ ). А именно,  $d = 14$  в тесте 2 и 56 во всех остальных тестах. В качестве локальной нормы, в тесте 4 выбрана  $L_2$ -норма, а во всех остальных тестах  $W_2^1$ -норма. В тесте 3 веса берутся единичными, а во всех остальных тестах они берутся равными величинам, обратным компонентам рассматриваемого стационарного решения.

Для каждого из тестов с помощью алгоритмов I и II были рассчитаны максимальная амплификация, оптимальное возмущение и локальная норма оп-

тимального возмущения. Максимальная амплификация для случая  $l = 50$  изображена на рисунках 2.2(а), 2.4(а), 2.6(а) и 2.8(а). Там же изображена локальная норма найденного оптимального возмущения, а компоненты его начального значения изображены на рисунках 2.3, 2.5, 2.7 и 2.9. Стоит отметить, что вид максимальной амплификации в случае тестов 1 и 2 (рисунки 2.2(а) и 2.4(а), соответственно) не является типичным. Более типичными являются максимальные амплификации, полученные для тестов 3 и 4 и изображенные на рисунках 2.6(а) и 2.8(а), соответственно.

Рисунки 2.2(б), 2.4(б), 2.6(б) и 2.8(б) демонстрирует работу алгоритма III при  $l = 50$ . Крупные точки соответствуют моментам времени  $k_1\delta$  (зеленая),  $k_2\delta$  (синяя),  $k_3\delta$  (красная), то есть первому, второму и третьему приближениям к  $t_{\text{opt}}$ .

В случае теста 1 все рассматриваемые алгоритмы дали одинаковые результаты  $t_{\text{opt}} \approx 5.50$ ,  $\Gamma(t_{\text{opt}}) \approx 12.5$  с точностью до 16 значащих десятичных цифр вне зависимости от значения  $l$ . В случае тестов 3 и 4 все алгоритмы с точностью до 16 значащих десятичных цифр дали одинаковые результаты, но различные при  $l = 1$  и 50:  $t_{\text{opt}} \approx 12.32$ ,  $\Gamma(t_{\text{opt}}) \approx 7.5498$  при  $l = 1$  и  $t_{\text{opt}} \approx 12.25$ ,  $\Gamma(t_{\text{opt}}) \approx 7.5496$  при  $l = 50$  в случае теста 3, и  $t_{\text{opt}} \approx 11.27$ ,  $\Gamma(t_{\text{opt}}) \approx 39.7884$  при  $l = 1$  и  $t_{\text{opt}} \approx 11.25$ ,  $\Gamma(t_{\text{opt}}) \approx 39.7883$  при  $l = 50$  в случае теста 4.

В случае теста 2 алгоритмы I и II с точностью до 16 значащих десятичных цифр дали одинаковые результаты  $t_{\text{opt}} \approx 4.40$ ,  $\Gamma(t_{\text{opt}}) \approx 4.817$  при  $l = 1$  и  $t_{\text{opt}} \approx 4.25$ ,  $\Gamma(t_{\text{opt}}) \approx 4.815$  при  $l = 50$ . Алгоритм III остановился после  $g = 2$  итераций. При этом были получены следующие результаты:  $t_{\text{opt}} \approx 10.41$ ,  $\Gamma(t_{\text{opt}}) \approx 2.26$  при  $l = 1$  и  $t_{\text{opt}} \approx 10.25$ ,  $\Gamma(t_{\text{opt}}) \approx 2.25$  при  $l = 50$ . То есть в данном случае последовательная максимизация сходилась не к оптимальному возмущению, а к возмущению, на котором достигается локальный максимум.

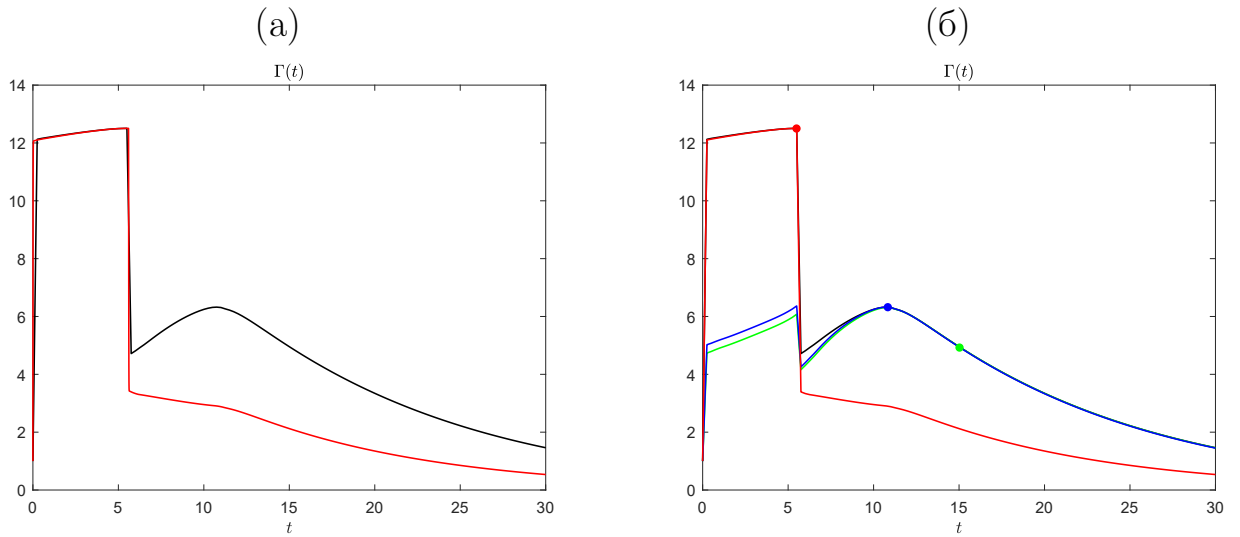


Рисунок 2.2 — Максимальная амплификация (черная линия) для теста 1 и локальная норма оптимального возмущения, полученные алгоритмами I и II, (красная линия) (а) и локальные нормы приближений к оптимальному возмущению, найденных на итерациях алгоритма III: первой — зеленая, второй — синяя, третьей — красная линии (б)

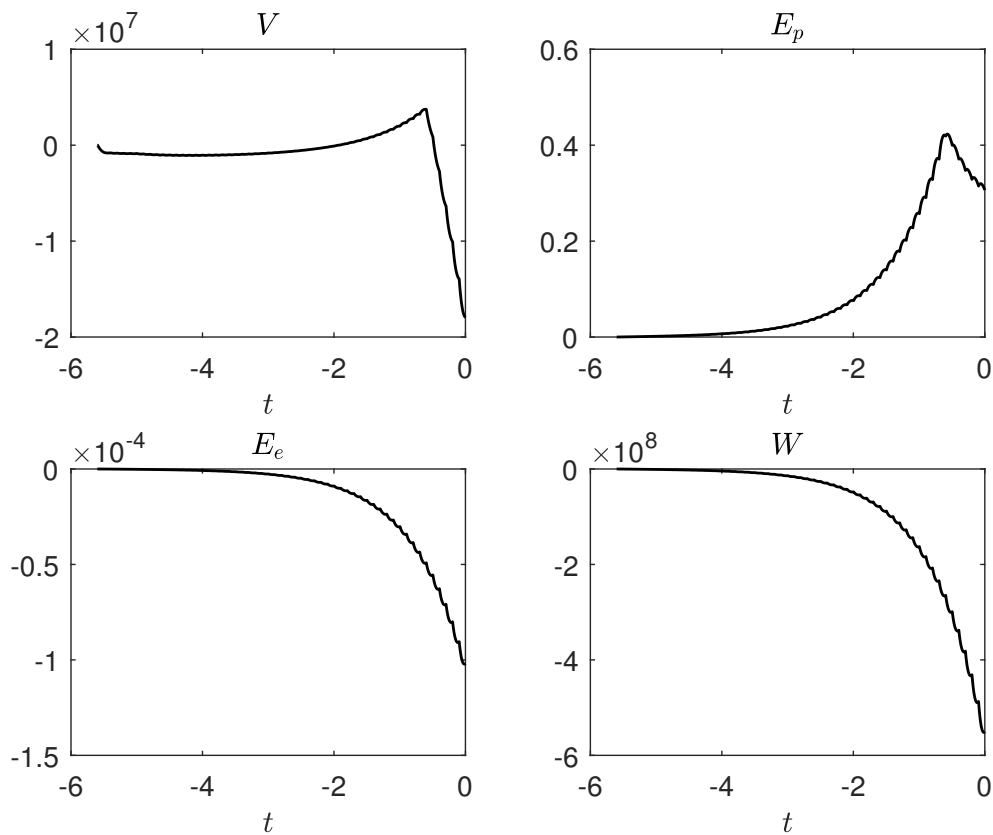


Рисунок 2.3 — Компоненты начального значения оптимального возмущения для теста 1

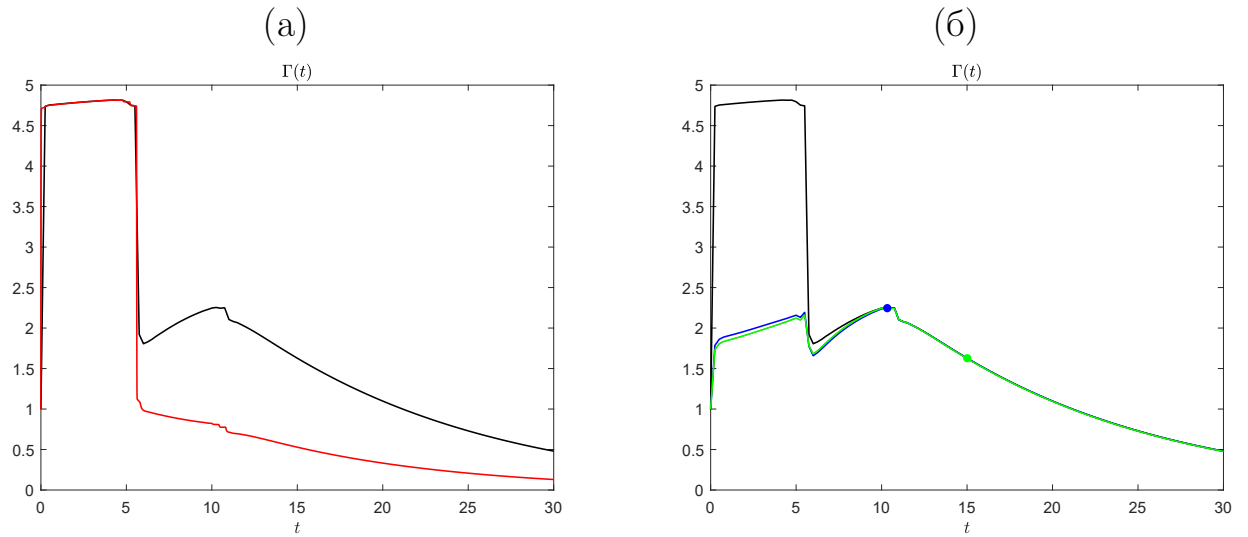


Рисунок 2.4 — Максимальная амплификация (черная линия) для теста 2 и локальная норма оптимального возмущения, полученные алгоритмами I и II, (красная линия) (а) и локальные нормы приближений к оптимальному возмущению, найденных на итерациях алгоритма III: первой — зеленая, второй — синяя линии (б)

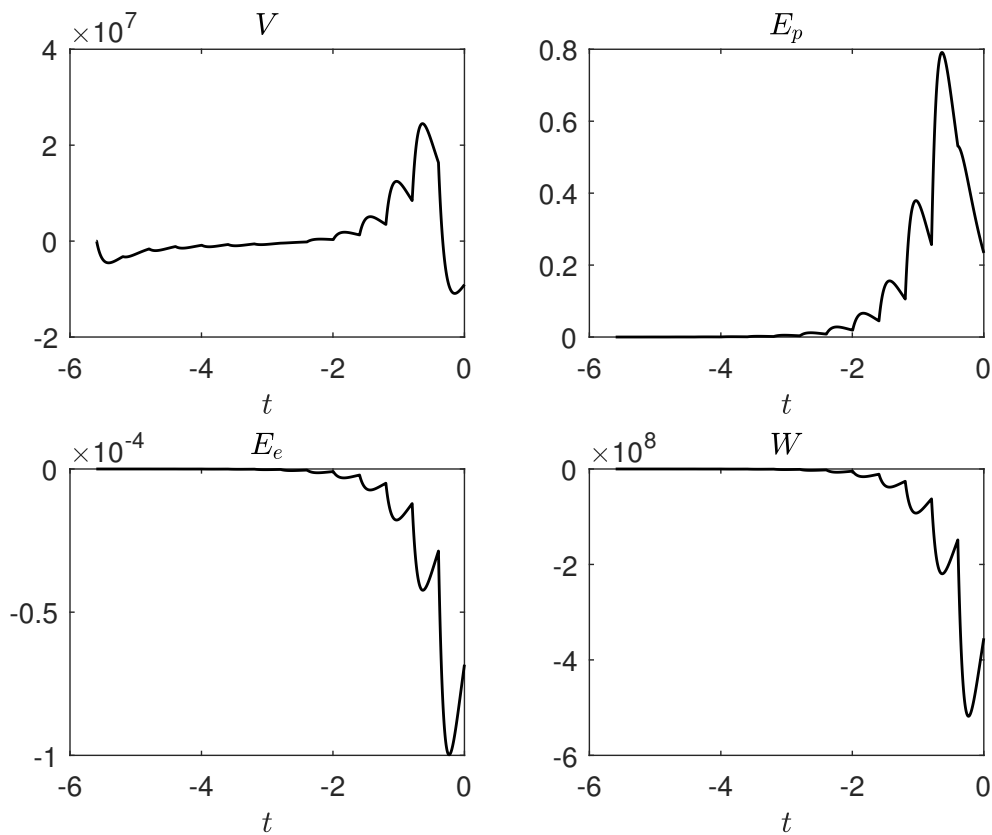


Рисунок 2.5 — Компоненты начального значения оптимального возмущения для теста 2

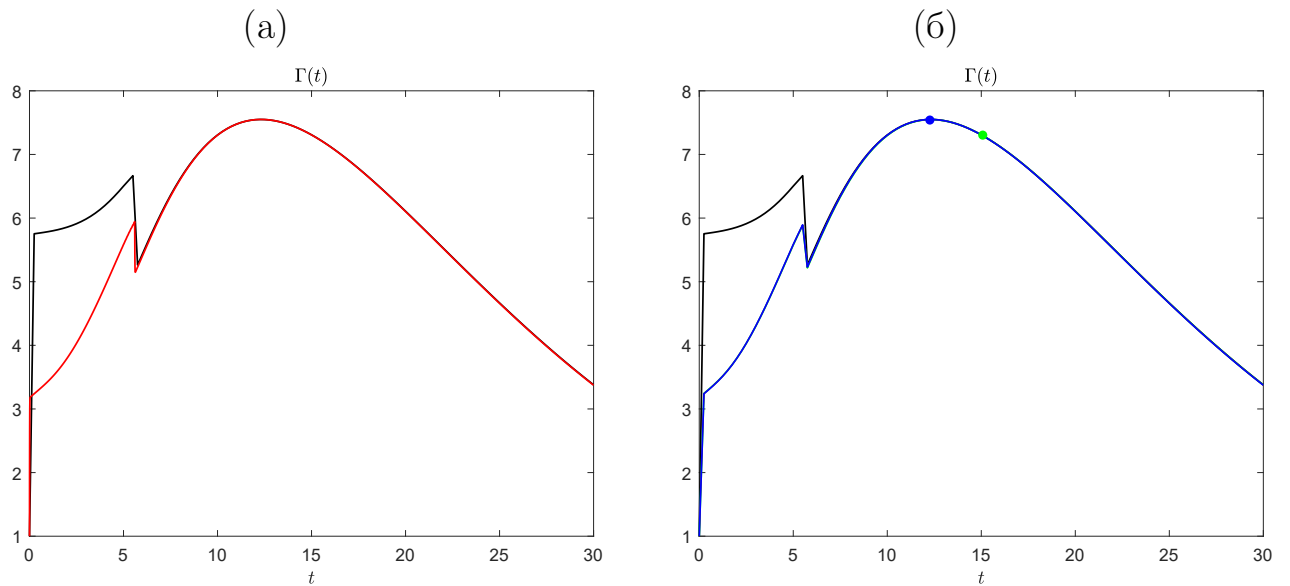


Рисунок 2.6 — Максимальная амплификация (черная линия) для теста 3 и локальная норма оптимального возмущения, полученные алгоритмами I и II, (красная линия) (а) и локальные нормы приближений к оптимальному возмущению, найденных на итерациях алгоритма III: первой — зеленая, второй — синяя (б)

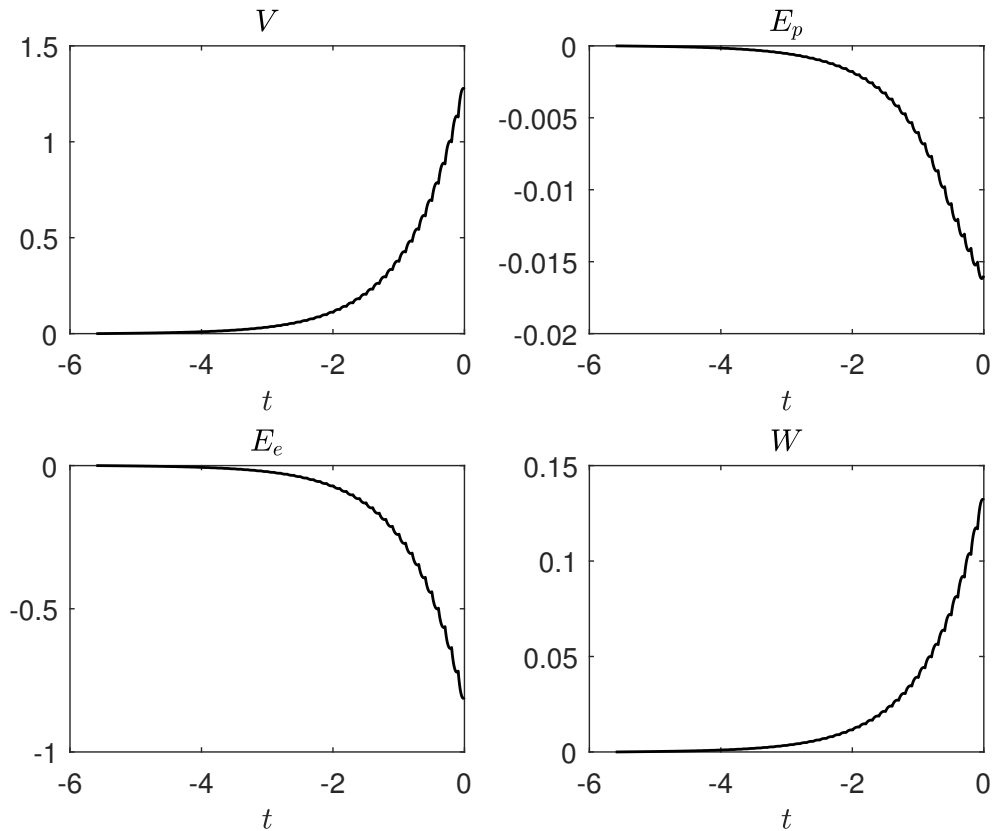


Рисунок 2.7 — Компоненты начального значения оптимального возмущения для теста 3



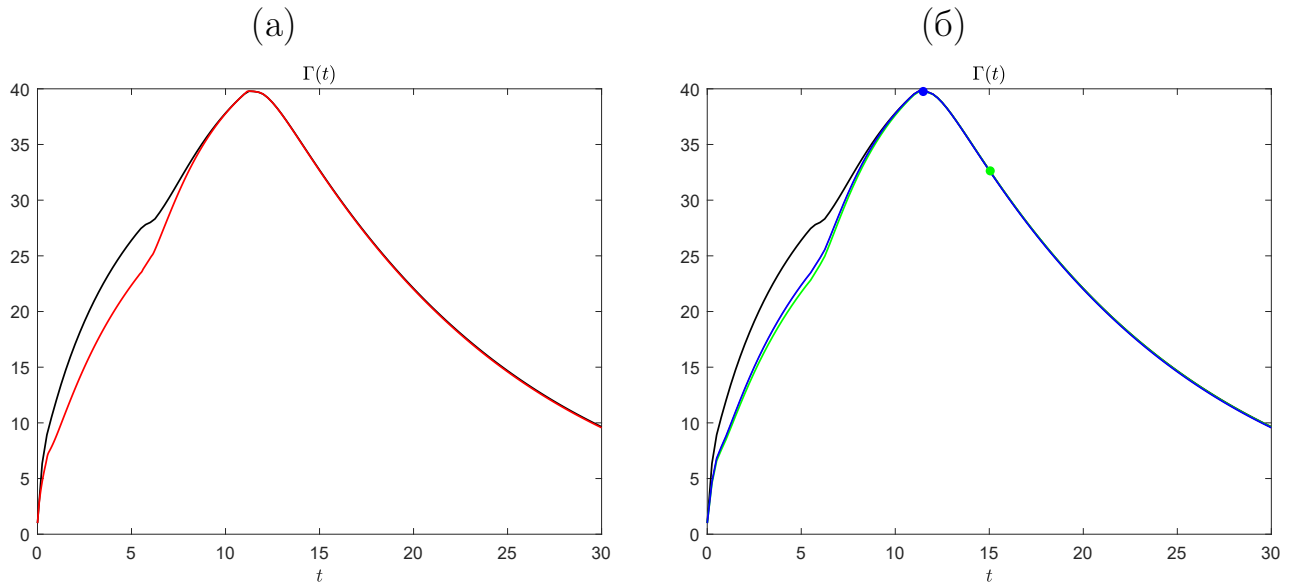


Рисунок 2.8 — Максимальная амплификация (черная линия) для теста 4 и локальная норма оптимального возмущения, полученные алгоритмами I и II, (красная линия) (а) и локальные нормы приближений к оптимальному возмущению, найденных на итерациях алгоритма III: первой — зеленая, второй — синяя (б)

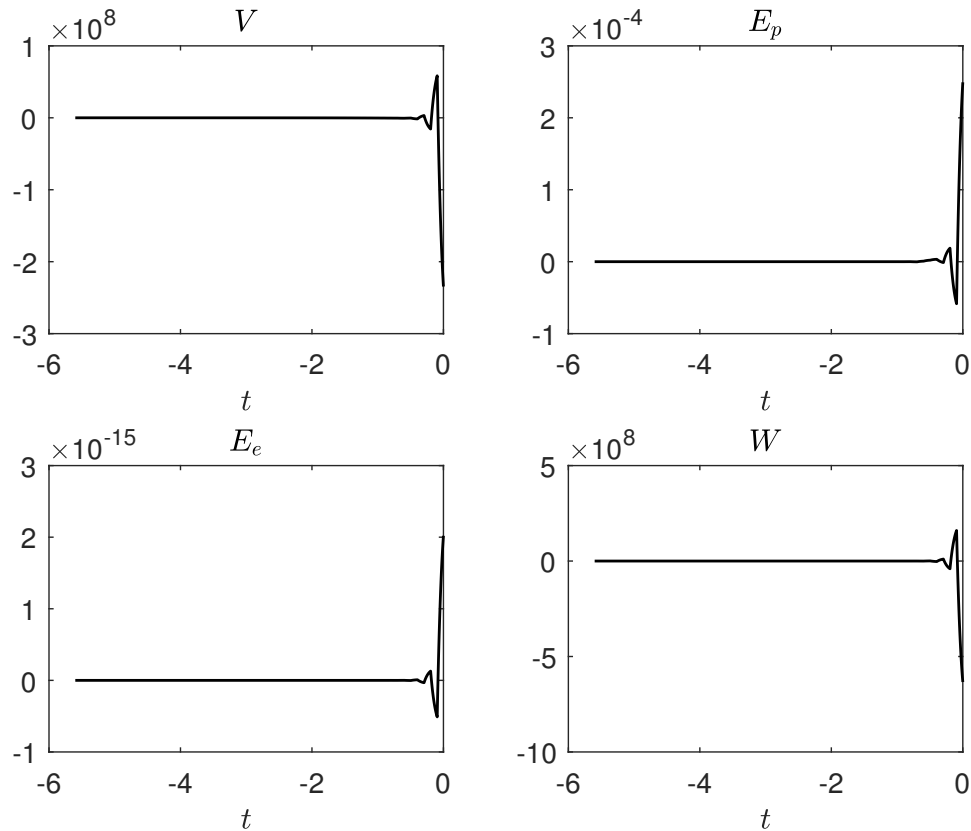


Рисунок 2.9 — Компоненты начального значения оптимального возмущения для теста 4

В таблице 5 приведены среднее число шагов метода Ланцоша  $r_{av}$  (оно оказалось одинаковым для алгоритмов II и III во всех тестах, кроме теста 4), число итераций  $g$  алгоритма III и  $k_{av}$ . Подставляя эти величины и указанные ранее значения параметров расчета в формулы (2.33), (2.34), (2.35) получим приведенные в таблице 6 числа операций, необходимых каждому из рассматриваемых алгоритмов. Для двух первых алгоритмов учитывалось только число операций, необходимых для вычисления максимальной амплификации. В таблице 7 указаны времена работы всех трех алгоритмов в среде MATLAB на компьютере с 2-х ядерным процессором Intel Core i7 с частотой 2.4 GHz. Для двух первых алгоритмов учитывалось только время вычисления максимальной амплификации. Вычисление самого оптимального возмущения требовало еще 4.2 и 0.3 секунд соответственно. Алгоритм III для нахождения оптимального возмущения не требует дополнительных вычислений. Отношение числа операций, необходимых для реализации каждого алгоритма на время его работы, дает производительность, которая приведена в таблице 8.

Таблица 5. Результаты работы алгоритмов II и III

	Тест 1		Тест 2		Тест 3		Тест 4	
	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$
$r_{av}$	6	6	6	6	5	5	6/5	6/5
$g$	4	3	2	2	3	2	4	2
$k_{av}$	1845	2083	2541	2525	2642	2725	2444	2625

Таблица 6. Число операций

Алгоритм	Тест 1		Тест 2		Тест 3		Тест 4	
	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$
I	$2.7 \cdot 10^{12}$	$5.4 \cdot 10^{10}$	$1.7 \cdot 10^{11}$	$3.4 \cdot 10^9$	$2.7 \cdot 10^{12}$	$5.4 \cdot 10^{10}$	$2.7 \cdot 10^{12}$	$5.4 \cdot 10^{10}$
II	$6.4 \cdot 10^{10}$	$1.3 \cdot 10^9$	$3.7 \cdot 10^{10}$	$7.3 \cdot 10^8$	$5.3 \cdot 10^{10}$	$1.1 \cdot 10^9$	$6.4 \cdot 10^{10}$	$1.3 \cdot 10^9$
III	$5.8 \cdot 10^8$	$3.8 \cdot 10^7$	$2.8 \cdot 10^8$	$1.8 \cdot 10^7$	$4.3 \cdot 10^8$	$2.4 \cdot 10^7$	$2.5 \cdot 10^8$	$2.0 \cdot 10^7$

Таблица 7. Время работы (в секундах)

Алгоритм	Тест 1		Тест 2		Тест 3		Тест 4	
	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$
I	184.9	23.2	28.1	5.7	184.9	23.2	178.9	22.1
II	3155	63.2	3064	61.4	3154	63.2	3229	64.7
III	2.4	1.7	1.5	1.3	2.1	1.5	2.5	1.3

Таблица 8. Производительность (число операций в секунду)

Алгоритм	Тест 1		Тест 2		Тест 3		Тест 4	
	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$	$l = 1$	$l = 50$
I	$1.5 \cdot 10^{10}$	$2.3 \cdot 10^9$	$6.0 \cdot 10^9$	$6.0 \cdot 10^8$	$1.5 \cdot 10^{10}$	$2.3 \cdot 10^9$	$1.5 \cdot 10^{10}$	$2.4 \cdot 10^9$
II	$2.0 \cdot 10^7$	$2.1 \cdot 10^7$	$1.2 \cdot 10^7$	$1.2 \cdot 10^7$	$1.7 \cdot 10^7$	$1.7 \cdot 10^7$	$2.0 \cdot 10^7$	$2.0 \cdot 10^7$
III	$2.4 \cdot 10^8$	$2.2 \cdot 10^7$	$1.9 \cdot 10^8$	$1.4 \cdot 10^7$	$2.0 \cdot 10^8$	$1.6 \cdot 10^7$	$1.0 \cdot 10^8$	$1.5 \cdot 10^7$

Из таблицы 4 видно, что производительность на алгоритме I при всех  $l$  и на алгоритме III при  $l = 1$  существенно выше производительности на алгоритме II при всех  $l$  и на алгоритме III при  $l = 50$ . Это связано с тем, что некоторые стандартные функции среды MATLAB, реализующие матричные алгоритмы, особенно эффективно используют вычислительные возможности компьютера. На этих функциях достигается существенно более высокая производительность, чем на других стандартных функциях MATLAB и тем более на пользовательских реализациях тех же и других алгоритмов на языке среды MATLAB.

Поясним сказанное на примере теста 1. Вычисления алгоритма I при  $l = 1$  можно разбить на три части: вычисление  $Y_k = M^k Y_0$  при  $0 \leq k \leq N$ , вычисление матриц  $A_k = H Y_k$  для всех найденных матриц  $Y_k$ , и вычисление вторых норм всех найденных матриц  $A_k$ . Для первой части достаточно  $2 \cdot 10^8$ , для второй —  $1.8 \cdot 10^{10}$ , для третьей —  $2.7 \cdot 10^{12}$  арифметических операций. Первая часть была реализована нами на языке MATLAB без существенного использования стандартных функций, во второй части использовалось стандартное умножение на разреженную матрицу в разреженном формате. В третьей части использовалась очень эффективная стандартная функция `norm` (время ее работы совпадает с временем вычисления сингулярных чисел с помощью стандартной функции `svd` на матрице того же размера). Первая часть потребовала 20.6, вторая — 26, третья — 158 секунд. Таким образом, производительности на

этих трех частях алгоритма I оказались существенно различными и составляли соответственно  $9.7 \cdot 10^6$ ,  $6.9 \cdot 10^8$  и  $1.7 \cdot 10^{10}$  операций в секунду. При  $l = 50$  алгоритм I выполняет в 50 раз меньше умножений на матрицу  $H$  и вычислений матричных норм по сравнению с числом умножений на матрицу  $M$ . Как следствие, производительность на нем уменьшается на порядок.

В алгоритме II число умножений на матрицу  $H$  пренебрежимо мало по сравнению с числом умножений на матрицу  $M$ , а нормы матриц стандартными функциями вообще не вычисляются. Отсюда и сравнительно низкая производительность.

На алгоритме III более высокая производительность при  $l = 1$ , чем при  $l = 50$ , достигается благодаря большему вкладу в общее число операций вычисления  $\|A_1\eta\|_2, \|A_2\eta\|_2, \dots, \|A_N\eta\|_2$ , для которого мы использовали стандартную функцию `norm` и стандартное умножение на разреженную матрицу.

В заключение этого раздела отметим следующее. Алгоритмом, наиболее точно вычисляющим оптимальные возмущения очевидно является базовый алгоритм, поскольку в нем вычисление максимального сингулярного числа и отвечающего ему правого сингулярного вектора выполняются с машинной точностью. Однако, для систем с достаточно большим количеством переменных, он требует больше памяти и числа арифметических операций. Менее точной, но более эффективной, является модификация этого алгоритма с использованием метода Ланцоша, хотя по времени работы наша реализация этой модификации в рассмотренных случаях уступает нашей реализации базового алгоритма в силу используемых в коде стандартных процедур умножения на разреженную матрицу и вычисления второй нормы матрицы, очень эффективно реализованных в среде MATLAB.

Наиболее эффективным алгоритмом вычисления оптимального возмущения оказался алгоритм последовательной максимизации. Он предпочтительнее с точки зрения объема требуемой памяти и количества операций, более того, время работы его реализации значительно меньше времен работы базового алгоритма и его модификации, использующей метод Ланцоша. Хотя алгоритм последовательной максимизации не гарантирует, вообще говоря, получения правильного результата во всех случаях, поскольку этот метод может сойтись к локальному максимуму, тем не менее его имеет смысл использовать при боль-

ших параметрических расчетах, выборочно контролируя результат с помощью одного из двух других алгоритмов.

## Глава 3. Оптимальные возмущения периодических решений

### 3.1 Вычисление периодических решений

В теории бифуркационного анализа [58] вещественных нелинейных систем, представляющих собой системы дифференциальных уравнений и зависящих от параметра, стационарные решения рассматривают как функции этого параметра. Если система, линеаризованная относительно одного из стационарных решений, имеет комплексно-сопряженную пару ведущих (с максимальной вещественной частью) собственных значений с ненулевыми мнимыми частями, которая с ростом значения параметра переходит из левой полуплоскости в правую, то возможны три сценария потери устойчивости стационарного решения. Значение параметра, при котором происходит потеря устойчивости называется точкой бифуркации Андронова-Хопфа. Первый сценарий потери устойчивости соответствует случаю, когда при значении параметра, меньшем точки бифуркации в окрестности устойчивого стационарного решения существует неустойчивое периодическое решение, которое соединяется со стационарным после точки бифуркации, то есть в точке бифуркации амплитуда периодического решения становится равной нулю и периодическое решение становится стационарным решением. Вторым сценарий соответствует случаю, когда устойчивое периодическое решение возникает в окрестности неустойчивого стационарного решения сразу после точки бифуркации. Третий сценарий соответствует случаю, когда при определенном значении параметра, меньшем точки бифуркации, в окрестности устойчивого стационарного решения, возникает одно неустойчивое периодическое решение и одно устойчивое, причем при росте параметра, неустойчивое периодическое решение соединяется со стационарным в точке бифуркации, а устойчивое продолжает существовать в окрестности неустойчивого стационарного решения при значениях параметра, больших точки бифуркации. Данный раздел посвящен описанию метода, который в случае второго и третьего сценариев потери устойчивости позволяет вычислить периодическое решение, находящееся в окрестности неустойчивого стационарного решения. Метод применяется к системе (1.3) или (1.4) и позволяет найти ее приближенное периодическое

решение, являющееся сеточной функцией, заданной на сетке (1.5). Предлагаемый метод состоит из трех этапов. Для простоты изложения опишем сначала идеи первых двух этапов применительно к исходной непрерывной системе (1.1).

### 3.1.1 Идея метода

Будем говорить, что функция  $f$ , заданная в бесконечном интервале  $[0, \infty)$ , является периодической, если найдется такое  $T > 0$ , что  $f(t + T) \equiv f(t)$ . Величину  $T$  будем называть периодом функции  $f$ . Минимальное  $T$ , являющееся периодом функции  $f$  будем называть ее минимальным периодом. Будем предполагать, что минимальный период искомого периодического решения является минимальным периодом и первой переменной  $u^1(t)$  вектора  $u(t)$ .

Любое решение  $u$  системы (1.1) однозначно определяется начальной функцией

$$u(t), \quad -\tau_q \leq t \leq 0. \quad (3.1)$$

Поэтому мы будем искать начальную функцию, отвечающую периодическому решению  $u$ , и его минимальный период  $T$ . Точность найденного приближенного периодического решения и его минимального периода будем определять с помощью следующего функционала невязки:

$$\frac{\left( \int_{-\tau_q}^0 \|D(u(t+T) - u(t))\|_2^2 dt \right)^{1/2}}{\left( \int_{-\tau_q}^0 \|Du(t)\|_2^2 dt \right)^{1/2}}, \quad (3.2)$$

где  $D$  — заданная положительно определенная диагональная матрица. В качестве диагональных элементов матрицы  $D$  будем выбирать величины, обратные к компонентам неустойчивого стационарного решения  $\bar{u}$ , в окрестности которого ищется периодическое решение. Значения функции  $u(t+T)$  на отрезке  $[-\tau_q, 0]$  будем вычислять, решая задачу Коши для системы (1.1) с начальной функцией (3.1).

Первый этап заключается в вычислении приближенного периодического решения методом установления. В качестве начальной выбирается функция

$$u(t) = \bar{u} + \varepsilon \operatorname{Real}(ze^{t\lambda}), \quad -\tau_q \leq t \leq 0, \quad (3.3)$$

где  $\lambda$  — ведущее собственное значение линеаризованной относительно стационарного решения  $\bar{u}$  системы (предполагается, что ведущее собственное значение имеет мнимую часть, отличную от нуля),  $z$  — отвечающий этому собственному значению нормированный собственный вектор,  $\varepsilon$  — некоторая положительная величина (параметр алгоритма). В качестве начального значения минимального периода выбирается  $T = 2\pi/|\operatorname{Imag}(\lambda)|$ .

На первом шаге метода установления система (1.1) интегрируется с начальной функцией (3.3) до момента времени  $N_e T$ , где  $N_e$  — некоторое заданное небольшое натуральное число (параметр алгоритма). На втором шаге система (1.1) интегрируется до момента времени  $2N_e T$  с начальной функцией

$$u(t), \quad N_e T - \tau_q \leq t \leq N_e T,$$

вычисленной на первом шаге. Если величина

$$\frac{|\operatorname{var}u_2^1 - \operatorname{var}u_1^1|}{\operatorname{var}u_1^1} \leq \theta_{u^1},$$

где  $\theta_{u^1}$  — заданная точность (параметр алгоритма), а  $\operatorname{var}u_i^1$  — разность между максимальным и минимальным значениями переменной  $u^1$  на  $i$ -ом интервале интегрирования, то решение считается установившимся и работа метода завершается. Если же заданная точность установления вариации переменной  $u^1$  не достигнута, то работа метода продолжается подобным образом до тех пор пока либо не будет достигнуто некоторое заданное максимальное число шагов  $N_{f,\max}$  (параметр алгоритма), либо пока абсолютная величина относительного изменения за один шаг вариации переменной  $u^1$  не станет меньше чем  $\theta_{u^1}$ . Будем предполагать, что всего выполнено  $N_f$  шагов метода установления, то есть решение рассчитано до момента времени  $N_f N_e T$ .

На втором этапе оценивается минимальный период приближенного периодического решения. В качестве первого шага этого этапа, интервал  $[(N_f - 1)N_e T, N_f N_e T]$  обрезается слева таким образом, чтобы длина полученного ин-



тервала  $[L, N_f N_e T]$  была приближенным периодом приближенного периодического решения. Для этого выбирается наименьшее значение  $L$  в интервале

$$[(N_f - 1)N_e T, N_f N_e T - \tau_q]$$

при котором значение функционала

$$\frac{\left( \int_{-\tau_q}^0 \|(u^1(N_f N_e T + t) - u^1(L + t))\|_2^2 dt \right)^{1/2}}{\left( \int_{-\tau_q}^0 \|u^1(N_f N_e T + t)\|_2^2 dt \right)^{1/2}} \quad (3.4)$$

меньше некоторой малой величины  $tol_{u^1}$  (параметр алгоритма). После этого, значение  $L$  увеличивается до тех пор, пока значение функционала уменьшается.

На втором шаге минимальный период оценивается на основе разложения функции  $u^1$  в ряд Фурье

$$u^1(t) = \sum_{j=-\infty}^{\infty} \widehat{u}^1_j e^{i \frac{2\pi j}{N_f N_e T - L} (t-L)}, \quad (3.5)$$

в интервале  $[L, N_f N_e T]$ , где

$$\widehat{u}^1_j = \frac{1}{N_f N_e T - L} \int_L^{N_f N_e T} u^1(t) e^{-i \frac{2\pi j}{N_f N_e T - L} (t-L)} dt. \quad (3.6)$$

Если бы функция  $u^1$  была точно периодической функцией с минимальным периодом  $(N_f N_e T - L)/s$ , где  $s$  — некоторое натуральное число, то в ряде (3.5) ненулевыми могли бы быть только гармоники с номерами  $j$ , кратными  $s$ . Учитывая это, выберем среди гармоник, соответствующих положительным частотам все гармоники с номерами  $j_1, \dots, j_r$  чьи абсолютные величины  $|\widehat{u}^1_{j_1}|, \dots, |\widehat{u}^1_{j_r}|$  не меньше, чем

$$\rho \max\{|\widehat{u}^1_j| : -\infty \leq j \leq \infty\},$$

где  $\rho$  — некоторая малая величина (параметр алгоритма). Вычислим наибольший общий делитель  $s$  чисел  $j_1, \dots, j_r$  и положим приближенный минимальный период  $\widetilde{T}$  равным  $(N_f N_e T - L)/s$ .

На третьем шаге вычисленный приближенный минимальный период  $\tilde{T}$  корректируется. Сначала, функция  $u$  вычисляется с помощью интегрирования системы (1.1) от момента времени  $T_1 = N_f N_e T$  до момента времени  $T_2 = T_1 + (1 + \varepsilon_T)\tilde{T}$ , где  $\varepsilon_T$  — некоторая малая величина (параметр алгоритма). Затем, из интервала  $[0, 2\varepsilon_T\tilde{T}]$  выбирается  $l$ , при котором невязка

$$\frac{\left( \int_{-\tau_q}^0 \|D(u(T_2 + t) - u(T_1 + l + t))\|_2^2 dt \right)^{1/2}}{\left( \int_{-\tau_q}^0 \|Du(T_2 + t)\|_2^2 dt \right)^{1/2}} \quad (3.7)$$

достигает своего минимума. Скорректированный приближенный минимальный период полагается равным  $T_2 - T_1 - l$ .

### 3.1.2 Численная реализация

Для численной реализации первых двух этапов выберем достаточно малый шаг сетки по времени  $\delta$  и перейдем от системы (1.1) к дискретной системе (1.4) или (1.3). Вместо функций  $u$ ,  $u^1$  будем использовать соответствующие сеточные функции; вместо начального значения минимального периода  $T$  будем использовать величину  $\delta[T/\delta]$ ; в качестве величины  $L$  будем брать узел сетки такой, чтобы  $N = (N_f N_e T - L)/\delta$  было четным числом; вместо функционалов (3.2), (3.4), (3.7) будем использовать их аппроксимации методом трапеций, таким образом, например, сеточный аналог функционала (3.2) будет следующим:

$$\frac{\left( \sum_{k=-m_q+1}^0 h_k \|D(u(\delta k + T) - u(\delta k))\|_2^2 \right)^{1/2}}{\left( \sum_{k=-m_q+1}^0 h_k \|Du(\delta k)\|_2^2 \right)^{1/2}}, \quad (3.8)$$

где  $h_{-m_q+1} = h_0 = \delta/2$  и  $h_k = \delta$  при  $-m_q + 2 \leq k \leq 1$ .

Вместо разложения в ряд Фурье (3.5) будем использовать его аппроксимацию конечным рядом в узлах  $t_k = L + \delta k$  выбранной сетки:

$$u^1(t_k) = \sum_{j=-N/2+1}^{N/2} \widehat{u}^1_j e^{i\frac{2\pi j}{N}k},$$

где

$$\widehat{u}^1_j = \frac{1}{N} \sum_{k=0}^{N-1} u^1(t_k) e^{-i\frac{2\pi j}{N}k}.$$

Таким образом, численная реализация первых двух этапов будет давать кратный шагу сетки  $\delta$  приближенный минимальный период, который мы далее будем обозначать через  $T$ , и значения начальной функции приближенного периодического решения в узлах сетки, которые мы далее будем обозначать через  $u_{-m_q+1}, \dots, u_0$ .

### 3.1.3 Уточнение решения

На третьем этапе начальные значения приближенной периодической сеточной функции и приближенный минимальный период уточняются методом ньютоновского типа. Введем следующие обозначения:

$$y_k = (H \otimes D)x_k,$$

где

$$x_k = (u_k^T, u_{k+1}^T, \dots, u_{k-m_q+1}^T)^T, \quad H = \text{diag}(\delta/2, \delta, \dots, \delta, \delta/2).$$

Эти обозначения позволяют записать сеточный аналог (3.8) функционала невязки (3.2) в виде  $\|y_{T/\delta} - y_0\|_2 / \|y_0\|_2$ .

Задача уточнения начального значения приближенной периодической сеточной функции при фиксированном минимальном периоде  $T$  сводится к решению уравнения

$$G(y_0; T) = y_{T/\delta} - y_0 = 0$$

относительно  $y_0$ , где подвекторы вектора  $x_{T/\delta}$ , необходимые для вычисления вектора  $y_{T/\delta}$ , получены из подвекторов вектора  $x_0$  численным интегрированием по схеме (1.4).

Шаг предлагаемого метода ньютоновского типа имеет следующий вид:

$$y_0^{new} = y_0 - \alpha \Delta,$$

где

$$\Delta = \left( \frac{\partial G}{\partial y_0}(y_0; T) \right)^{-1} G(y_0; T)$$

является ньютоновским шагом, а  $\alpha$  — скалярным параметром, значение которого выбирается так, чтобы минимизировать  $\|G(y_0^{new}; T)\|_2$ . Оптимальное значение ищется в интервале  $[0, \alpha_{\max}]$ , где  $\alpha_{\max}$  — максимальное значение  $\alpha$ , при котором все компоненты вектора  $y_0^{new}$  являются неотрицательными. Если ньютоновский шаг  $\Delta$  уже вычислен, то для нахождения оптимального значения параметра  $\alpha$  можно использовать любой метод поиска минимума непрерывной функции одного переменного. Мы использовали широко известную процедуру `fmin` [48; 54], включенную во многие пакеты прикладных программ.

Ньютоновский шаг  $\Delta$  предлагается вычислять путем решения системы линейных уравнений с матрицей Якоби

$$\frac{\partial G}{\partial y}(y_0; T)$$

и правой частью  $G(y_0; T)$  методом обобщенных минимальных невязок (GMRES) с заданными максимальным числом шагов  $N_{LS}$  и пороговым отношением нормы невязки системы к норме ее правой части  $tol_{LS}$  (параметры алгоритма). Сама матрица Якоби при этом не вычисляется, а используется следующая процедура ее приближенного умножения на заданный вектор  $x$ :

$$\frac{\partial G}{\partial y}(y_0; T)x \approx \frac{G(y_0 + \xi x; T) - G(y_0 - \xi x; T)}{2\xi},$$

где  $\xi$  — малая положительная величина (параметр алгоритма).

После уточнения начальной сеточной функции выполняется корректировка приближенного минимального периода, описанная в разделе 3.1.1 (третий шаг второго этапа). После этого проверяется сходимость. Итерационный про-

цесс останавливается, если норма невязки  $\|G(y_0^{new}; T)\|_2/\|y_0\|_2$  не превосходит заданного значения  $tol_{NM}$  или если выполнено некоторое заданное максимальное число шагов  $N_{NM}$  (параметры алгоритма).

### 3.2 Анализ устойчивости периодического решения

Пусть непрерывно дифференцируемая векторная функция  $\phi$  является периодическим решением системы обыкновенных дифференциальных уравнений с запаздывающим аргументом вида (1.1), причем предполагаем, что функция  $\mathcal{F}(v_0, v_1, \dots, v_p, \mathbf{p})$  непрерывно дифференцируема в окрестности  $(\phi(t), \phi(t - \tau_1), \dots, \phi(t - \tau_p), \mathbf{p})$  при любом фиксированном  $t$  из интервала  $0 \leq t \leq T$ , а  $T$  — наименьший период функции  $\phi$ . Для исследования устойчивости заданного периодического решения нас будут интересовать непрерывные решения задачи Коши для линеаризованных уравнений

$$\frac{dw}{dt}(t) = \sum_{j=0}^q L_j(t)w(t - \tau_j), \quad (3.9)$$

где

$$L_j(t) = \frac{\partial \mathcal{F}}{\partial v_j}(\phi(t - \tau_0), \dots, \phi(t - \tau_q), \mathbf{p})$$

являются квадратными матрицами порядка  $n$ .

Согласно теории устойчивости периодических решений систем с запаздыванием [51] устойчивость периодического решения определяется спектром оператора монодромии  $\mathcal{M}$  системы (3.9). Так как система (1.1) является автономной, у оператора монодромии  $\mathcal{M}$  имеется собственное значение равное единице. Если это собственное значение является простым и единственным собственным значением, по модулю равным 1, а все остальные собственные значения лежат внутри единичного круга, то периодическое решение является асимптотически орбитально устойчивым. Таким образом, исследование асимптотически орбитальной устойчивости периодического решения  $\phi(t)$  сводится к вычислению собственных значений оператора монодромии и проверке, что все найденные

собственные значения лежат внутри единичного круга, за исключением одного простого собственного значения, равного единице.

Для численного анализа устойчивости найденного периодического решения необходимо построить сеточный аналог оператора монодромии. Для этого мы дискретизируем систему (3.9) с помощью неявного метода Эйлера (1.3) или неявной схемы BDF2 (1.4). В качестве примера мы используем неявную схему BDF2. После дискретизации эта система принимает вид:

$$\frac{1.5w_k - 2w_{k-1} + 0.5w_{k-2}}{\delta} = \sum_{j=0}^q L_{jk} w_{k-m_j}, \quad (3.10)$$

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

где  $L_{jk}$  — квадратные матрицы порядка  $n$ , представляющие собой сеточные функции, аппроксимирующие матрицы  $L_j(t)$  соответственно. В качестве начальных данных для решения задачи Коши для этой системы нужно задать значения  $w_{-m_q+1}, \dots, w_0$ .

Запишем уравнение (3.10) в виде

$$w_k = C_{1,k} w_{k-1} + C_{2,k} w_{k-2} + \sum_{j=0}^q C_{m_j,k} w_{k-m_j}, \quad (3.11)$$

где

$$C_{1k} = 2(1.5I - \delta L_{0k})^{-1}, \quad C_{2k} = -0.5(1.5I - \delta L_{0k})^{-1},$$

$$C_{m_j,k} = (1.5I - \delta L_{0k})^{-1} \delta L_{jk},$$

а  $I$  означает единичную матрицу порядка  $n$ , и дополним уравнение (3.11) тождествами  $w_j = w_j$ ,  $j = k-1, \dots, k-m_q+1$ . Полученную таким образом систему из  $m_q$  уравнений можно записать в виде

$$x_k = M_k x_{k-1}, \quad (3.12)$$

где

$$x_k = \begin{pmatrix} w_k \\ \vdots \\ w_{k-m_q+1} \end{pmatrix}, \quad M_k = \begin{pmatrix} M_{11}^{(k)} & \cdots & M_{1m_q}^{(k)} \\ \vdots & & \vdots \\ M_{m_q1}^{(k)} & \cdots & M_{m_qm_q}^{(k)} \end{pmatrix}. \quad (3.13)$$

Матрица  $M_k$  в (3.13) является блочной, блочного порядка  $m_q$  с блоками порядка  $n$ . Все блоки этой матрицы нулевые, кроме поддиагональных блоков  $M_{j+1,j}^{(k)} = I$  ( $j = 1, \dots, m_q - 1$ ) и  $q + 2$  блоков  $M_{11}^{(k)} = C_{1k}$ ,  $M_{12}^{(k)} = C_{2k}$ ,  $M_{1m_j}^{(k)} = C_{m_j k}$ ,  $j = 1, \dots, q$ , стоящих в первой блочной строке.

Сеточным аналогом оператора монодромии является матричное произведение

$$\Pi_l = M_l \dots M_2 M_1, \quad (3.14)$$

где  $l = T/\delta$ , а  $T$  — минимальный период рассматриваемого приближенного периодического решения  $\phi$ . Так как значения  $m_q$  и  $l$  растут при уменьшении шага сетки  $\delta$  как  $O(1/\delta)$ , время требуемое на явное формирование матрицы  $\Pi_l$  будет значительно расти при уменьшении  $\delta$ . Поэтому для вычисления собственных значений матрицы  $\Pi_l$ , которые являются приближенными собственными значениями оператора монодромии, предпочтительней использовать методы, которые не требуют в своей схеме явное формирование матрицы, а использующие только умножение матрицы на вектор. В данной работе в качестве такого метода используется метод Стюарта [60], позволяющий вычислять заданное число максимальных по абсолютной величине собственных значений. В качестве критерия сходимости использовалась вторая норма спектральной невязки  $\Pi_l Z - Z(Z^* \Pi_l Z)$ , где  $Z$  — прямоугольная матрица, столбцы которой образуют унитарный базис в вычисленном приближенном инвариантном подпространстве матрицы  $\Pi_l$ , отвечающем искомым собственным значениями,  $'^*$  обозначает символ сопряженного транспонирования.

### 3.3 Определение оптимального возмущения периодического решения

Пусть линейное пространство  $C[t_1, t_2]$  функций (1.2) снабжено  $L_2$  нормой  $\|f\|_{L_2} = (f, f)_{L_2}^{1/2}$ , где

$$(f, g)_{L_2} = \int_{t_1}^{t_2} (f(t), g(t)) dt,$$

а  $(\cdot, \cdot)$  означает скалярное произведение в  $\mathbf{R}^n$ .

Обозначим через  $\mathcal{M}$  отображение  $C[-\tau_q, 0] \rightarrow C[-\tau_q, 0]$  такое, что для любого  $f \in C[-\tau_q, 0]$  имеем  $\mathcal{M}f = g$ , где  $g(t) = w(T + t)$ ,  $-\tau_q \leq t \leq 0$ , а  $w$  означает решение задачи Коши для системы (3.9) с начальной функцией, равной  $f$ . Отображение  $\mathcal{M}$  называют оператором монодромии системы (3.9). Подставляя периодическое решение  $\phi(t)$  в уравнение (1.1) и дифференцируя полученное равенство, несложно проверить, что функция  $d\phi/dt$  удовлетворяет линейному уравнению (3.9). Следовательно, функция  $g(t) = d\phi/dt(t)$ ,  $-\tau_q \leq t \leq 0$ , удовлетворяет равенству  $\mathcal{M}g = g$ , то есть является собственной функцией оператора  $\mathcal{M}$ , отвечающей его собственному значению 1. Так как периодическое решение  $\phi(t)$  является по предположению устойчивым, то все остальные собственные значения оператора монодромии лежат строго внутри единичного круга. Таким образом, при  $t \rightarrow \infty$  решение задачи Коши для системы (3.9) с начальной функцией  $f \in C[-\tau_q, 0]$  будет стремиться к решению задачи Коши для этой системы с начальной функцией  $Pf$ , где  $P$  означает спектральный проектор на инвариантное подпространство оператора  $\mathcal{M}$ , отвечающее его собственному значению 1.

В отличие от максимальной амплификации локальной нормы возмущения устойчивого стационарного решения, максимальная амплификация локальной нормы возмущения устойчивого периодического решения, вообще говоря, не достигает своего максимума при конечном  $t$ , поскольку в оптимальном возмущении периодического решения присутствует незатухающая мода, отвечающая равному единице собственному значению оператора монодромии. Поэтому, для периодического решения глобальное оптимальное возмущение не имеет смысла. Учитывая это, мы далее будем рассматривать только возмущения, оптимальные в заданный момент времени. Если этот момент времени понятен из контекста либо его значение не принципиально, то такое возмущение мы будем называть оптимальным, не указывая, в какой момент времени оно является оптимальным.

Для функций вида (1.2) наряду с  $L_2$ -нормой введем в рассмотрение следующее семейство локальных норм в момент времени  $t$ :

$$\|f\|_{D,\rho,t} = \left( \int_{t-\tau_q}^t \left( \|Df(\tau)\|_2^2 + \rho \left\| D \frac{df}{d\tau}(\tau) \right\|_2^2 \right) d\tau \right)^{1/2}, \quad (3.15)$$



где  $D$  — заданная положительно определенная диагональная матрица порядка  $n$ ,  $\rho$  — неотрицательный параметр. Под возмущением устойчивого периодического решения системы (1.1), оптимальным в момент времени  $t_* \geq 0$ , будем понимать решение линеаризованной системы (3.9), на котором достигается максимальная амплификация (подскок) локальной нормы (3.15) решения в момент времени  $t_*$  по сравнению с ее первоначальным значением, то есть такое ненулевое решение  $w = w_{\text{opt}, t_*}$ , на котором достигается максимум величины

$$\frac{\|w\|_{D, \rho, t_*}}{\|w\|_{D, \rho, 0}}.$$

Также как при построении оптимального возмущения стационарного решения, при построении оптимального возмущения периодического решения наряду с выбором нормы, в которой проводится оптимизация, принципиальным является вопрос о том, из какого пространства  $\mathcal{Q} \subset C[-\tau_q, 0]$  мы берем начальные функции.

Если  $D$ ,  $\rho$ , и  $\mathcal{Q}$  фиксированы, то любое найденное возмущение, оптимальное в момент времени  $t_*$ , обеспечивает одну и ту же максимальную амплификацию локальной нормы (3.15) решения в этот момент времени. Более того, обычно оптимальное возмущение единственно с точностью до ненулевой мультипликативной константы.

Найденное оптимальное возмущение мы будем нормировать в локальной норме с параметром  $\rho = 0$  и использовать для возмущения периодического решения исходной нелинейной системы (1.1), выбирая в качестве начального значения

$$u(t) = \phi(t) + \varepsilon \tilde{w}_{\text{opt}, t_*} \quad (3.16)$$

при  $-\tau_q \leq t \leq 0$ , где  $\tilde{w}_{\text{opt}, t_*}$  означает нормированное в указанной норме оптимальное возмущение, а  $\varepsilon$  — вещественный параметр. Варьируя абсолютную величину этого параметра, можно увеличивать либо уменьшать начальное возмущение. В зависимости же от его знака, заданная компонента решения при  $t = 0$  либо начинает возрастать, либо убывать с ростом  $t$  по сравнению с той же компонентой невозмущенного периодического решения.

### 3.4 Алгоритмы вычисления оптимального возмущения

Базовый алгоритм вычисления оптимальных возмущений периодических решений систем с запаздыванием является обобщением алгоритма вычисления оптимальных возмущений стационарных решений таких систем. Оптимальные возмущения периодических решений, как и оптимальные возмущения стационарных решений, можно вычислять на основе любой разностной схемы, подходящей для решения задач Коши для систем дифференциальных уравнений с запаздывающим аргументом. В частности их можно вычислять на основе неявного метода Эйлера (1.3) или неявной схемы BDF2 (1.4). Как и в разделе 2.5, мы опишем алгоритм, используя неявную схему второго порядка BDF2 на равномерной сетке в качестве примера.

В разделе 3.2 было показано, что после дискретизации систему (3.9) можно записать в следующем виде:

$$x_k = M_k x_{k-1}, \quad (3.17)$$

где

$$x_k = \begin{pmatrix} w_k \\ \vdots \\ w_{k-m_q+1} \end{pmatrix}, \quad M_k = \begin{pmatrix} M_{11}^{(k)} & \cdots & M_{1m_q}^{(k)} \\ \vdots & & \vdots \\ M_{m_q1}^{(k)} & \cdots & M_{m_qm_q}^{(k)} \end{pmatrix}. \quad (3.18)$$

Также как и сеточный аналог локальной нормы (2.22) сеточный аналог локальной нормы (3.15) в точке  $t_k$  может быть записан в следующем виде:

$$\|Hx_k\|_2, \quad (3.19)$$

где  $H$  — та же квадратная матрица порядка  $nm_q$ , что и в разделе 2.5.

Будем предполагать, что для всех  $i = 1, \dots, n$ ,  $i$ -е компоненты векторов  $u_j$ , составляющих начальное значение  $x_0$  сеточного аналога оптимального возмущения, ищутся как линейные комбинации  $d \leq m_q$  базисных сеточных функций, определенных в узлах  $t_{-m_q+1}, \dots, t_0$ . Обозначим через  $B_i$  матрицу размера  $m_q \times d$  значений этих базисных сеточных функций. Тогда столбцы  $nm_q \times nd$  матрицы

$$Q = [B_1 \otimes E_1, \dots, B_n \otimes E_n],$$

где  $E_j$  означает  $j$ -й столбец единичной матрицы порядка  $n$ , образуют базис в построенном сеточном аналоге подпространства  $\mathcal{Q}$ .

В силу (3.17), (3.18), (3.19), сеточный аналог  $\Gamma_k$  максимальной амплификации (2.24) нормы решения можно записать следующим образом:

$$\Gamma_k = \max_{x_0 \in \text{span}(Q) \setminus \{0\}} \frac{\|H\Pi_k x_0\|_2}{\|Hx_0\|_2},$$

где  $\text{span}(Q)$  означает линейную оболочку столбцов матрицы  $Q$ , а  $\Pi_k = M_k \cdots M_1$ . Пусть  $HQ = \tilde{Q}\tilde{R}$  — QR-разложение, где  $\tilde{Q}$  — ортогональная прямоугольная матрица, а  $\tilde{R}$  — верхняя треугольная квадратная матрица. Учитывая, что  $x_0 = Q\xi$ , где  $\xi$  — некоторый вектор, имеем  $Hx_0 = \tilde{Q}\tilde{R}\xi = \tilde{Q}\tilde{\xi}$ , где  $\tilde{\xi} = \tilde{R}\xi$ . Следовательно,

$$\frac{\|H\Pi_k x_0\|_2}{\|Hx_0\|_2} = \frac{\|H\Pi_k H^{-1}Hx_0\|_2}{\|Hx_0\|_2} = \frac{\|H\Pi_k H^{-1}\tilde{Q}\tilde{\xi}\|_2}{\|\tilde{Q}\tilde{\xi}\|_2} = \frac{\|H\Pi_k H^{-1}\tilde{Q}\tilde{\xi}\|_2}{\|\tilde{\xi}\|_2},$$

и, таким образом имеем

$$\Gamma_k = \max_{\tilde{\xi} \neq 0} \frac{\|H\Pi_k H^{-1}\tilde{Q}\tilde{\xi}\|_2}{\|\tilde{\xi}\|_2} = \|H\Pi_k H^{-1}\tilde{Q}\|_2.$$

Для повышения эффективности вычисления  $\Gamma_k$  вместо матрицы  $H^{-1}\tilde{Q}$  будем использовать равную ей матрицу  $Q\tilde{R}^{-1}$ . Тогда для вычисления  $\Gamma_k$  окончательно получим следующую формулу:

$$\Gamma_k = \|B_k\|_2,$$

где

$$B_k = H\Pi_k Q\tilde{R}^{-1}. \quad (3.20)$$

Таким образом, вычисление  $\Gamma_k$  сводится к формированию матрицы  $Y_0 = Q\tilde{R}^{-1}$  и вычислению матриц  $Y_k$  по рекуррентной формуле  $Y_k = M_k Y_{k-1}$  с одновременным вычислением норм матриц  $HY_k$  для всех натуральных  $k$ , не превосходящих заданное значение.

Для вычисления сеточного аналога возмущения, оптимального в момент времени  $t_k$ , достаточно вычислить правый сингулярный вектор  $\eta$  матрицы

$B_k$ , отвечающий ее максимальному сингулярному числу [50]. Тогда начальное значение сеточного аналога искомого возмущения вычисляется по формуле  $x_0 = Q\tilde{R}^{-1}\eta$ .

Описанный алгоритм можно немного упростить, включив матрицу  $D$  в матрицы  $M_k$ . Действительно, правая часть равенства (3.20) не изменится, если в матрицах  $H$  и  $\tilde{R}$  заменить матрицу  $D$  единичной, а блоки  $C_{lk}$  матрицы  $M_k$  заменить на  $DC_{lk}D^{-1}$ . Мы будем считать, что такое предварительное преобразование матрицы  $M_k$  выполнено и матрица  $D$ , фигурирующая в матрицах  $H$  и  $\tilde{R}$ , — единичная.

Матрица  $B_k$  в (3.20) имеет размер  $nm_q \times nd$ . Так как значение  $m_q$  растет при уменьшении шага сетки  $\delta$  как  $O(1/\delta)$ , размер матрицы  $B_k$  растет при уменьшении  $\delta$  и увеличении значений  $n$  и  $d$  и, следовательно, растет время, требуемое на явное формирование этой матрицы. Поэтому для вычисления максимального сингулярного числа матрицы  $A_k$  и отвечающего ему правого сингулярного вектора предпочтительней использовать методы, которые не требуют в своей схеме явное формирование матрицы, а использующие только умножение матрицы на вектор. Поэтому в рамках второго алгоритма вычисления оптимального возмущения периодического решения, также как и для вычисления оптимального возмущения стационарного решения, в качестве такого метода, предлагается использовать метод Ланцоша, применяя его к матрице  $B_k^T B_k$  для вычисления ее максимального собственного значения, корень из которого даст искомое сингулярное число, а собственный вектор матрицы  $B_k^T B_k$ , отвечающий ее максимальному собственному значению, — искомый сингулярный вектор.

Стоит отметить, что с помощью пакета DEODAN, вычисление оптимальных возмущений периодического решения, описанных в разделах 5.4, 5.5, на компьютере с 2-х ядерным процессором Intel Core i7 с частотой 2.4 GHz при использовании модификации базового алгоритма, использующей метод Ланцоша занимает 1 секунду, а при использовании самого базового алгоритма занимает 4 секунды и значительно большего объема рабочей памяти.

## Глава 4. Пакет DEODAN

Разработанная технология вычисления и анализа стационарных и периодических решений систем с запаздыванием, и вычисления для них оптимальных возмущений была реализована в виде пакета программ DEODAN (Delay Equations Optimal Disturbances ANalysis). Данный пакет разработан в среде MATLAB и состоит из двух типов функций этой среды: модельно независимых и модельно зависимых. Независимые от модели функции предназначены для вычисления и анализа стационарных и периодических решений систем дифференциальных уравнений с дискретным запаздыванием вида (1.1). Эти функции описаны в разделе 4.1. Для их использования пользователю пакета необходимо создать функции, зависящие от модели, которые будут выполнять анализ конкретной модели. Эти функции описаны в разделе 4.2. Следует отметить, что пакет DEODAN в значительной степени использует методы символьных вычислений. Также, пакет DEODAN можно использовать в среде Octave. Пакет DEODAN является открытым пакетом и последнюю версию пакета можно найти по ссылке (URL: <https://github.com/khristichenko-myu/DEODAN>).

### 4.1 Модельно независимые функции

#### 4.1.1 Численное интегрирование

Этот раздел посвящен краткому описанию функции `Solver`, не зависящей от модели. Эта функция предназначена для численного интегрирования по времени дифференциальных уравнений с дискретными запаздываниями вида (1.1) с использованием схем (1.4),(1.3). Синтаксис вызова этой функции показан в Листинге 4.1. Описание входных и выходных аргументов этой функции приведено в таблицах 9 и 10 соответственно.

```
[tnew, Unew]=Solver(coef, delays, RHS, t, U, Tadd, delta, err, ...
```

```
interp1_method , iresult , nfig , iclean , h , v , ...
marks , inc , scheme , varnames , logind)
```

## Листинг 4.1 Синтаксис вызова функции Solver

Таблица 9. Входные аргументы функции Solver

Аргумент	Описание
<code>coef</code>	массив структур с полями, содержащими значения параметров системы, для которых интегрируется модель (см. подраздел 4.2.1.1)
<code>delays</code>	массив структур с полями, содержащими значения задержек в системе (см. подраздел 4.2.1.1)
<code>RHS</code>	функция, определяющая правую часть системы и описанная в подразделе 4.2.1.2
<code>t</code>	$(m_q + 1)$ -компонентный вектор, содержащий часть временной сетки (1.5), на которой задано начальное значение
<code>U</code>	$n \times (m_q + 1)$ матрица, содержащая начальное значение в узлах сетки, заданной параметром <code>t</code>
<code>Tadd</code>	положительная скалярная величина, определяющая время интегрирования
<code>delta</code>	положительная скалярная величина, определяющая шаг $\delta$ сетки (1.5)
<code>err</code>	положительная скалярная величина, определяющая точность метода Ньютона, используемого на каждом шаге и интегрирования по времени
<code>interp1_method</code>	строковый массив, определяющий метод интерполяции: 'linear', 'spline', 'pchip'
<code>iresult</code>	положительная скалярная величина, указывающая, что делать с вычисленным решением. Возможные значения: 1 - дополнить старое решение новым вычисленным, 2 - сохранить только новое вычисление, 3 - сохранить только "хвост" нового расчета, который необходимый для продолжения интеграции, 4 - сохранить новый расчет и исходные данные для него
<code>nfig</code>	положительная скалярная величина, определяющая номер рисунка, на котором будет показано решение (если <code>nfig &lt; 1</code> , то рисунка не будет)
<code>iclean</code>	булева переменная, указывающая, нужна ли предварительная очистка фигур
<code>h</code>	положительная скалярная величина, определяющая число подграфиков, располагающихся горизонтально на рисунке
<code>v</code>	положительная скалярная величина, определяющая число подграфиков, располагающихся вертикально на рисунке
<code>marks</code>	строковый массив, определяющий тип линий рисунка
<code>inc</code>	положительная скалярная величина, определяющая количество шагов, после которых функция выводит на экран информацию о своей работе
<code>scheme</code>	положительная скалярная величина, указывающая схему интегрирования. Возможные значения: 1 - неявный метод Эйлера, 2 - схема BDF2
<code>varnames</code>	массив из $n$ ячеек с именами системных переменных в формате LaTeX для отображения на рисунках (см. подраздел 4.2.1.3)
<code>logind</code>	булева переменная, указывающая на необходимость построения графика в логарифмическом масштабе

Таблица 10. Выходные аргументы функции `Solver`

Аргумент	Описание
<code>tnew</code>	$[Tadd/\delta]$ -компонентный вектор, содержащий сетку, в узлах которой задан результат интегрирования по времени, где $Tadd$ - значение входного аргумента <code>Tadd</code>
<code>Unew</code>	$n \times [Tadd/\delta]$ матрица, содержащая результат интегрирования по времени в узлах сетки, заданной аргументом <code>tnew</code>

### 4.1.2 Вычисление и анализ стационарных решений

Этот раздел посвящен краткому описанию функций, не зависящих от модели, предназначенных для вычисления и анализа устойчивости стационарных решений систем дифференциальных уравнений с дискретными запаздываниями вида (1.1), анализ зависимости этих решений от параметров системы и вычисления для них оптимальных возмущений. Алгоритмы, на которых основаны эти функции, описаны в разделах 2.1 – 2.7.

#### 4.1.2.1 SComputation

Функция `SComputation` предназначена для вычисления всех неотрицательных стационарных решений системы дифференциальных уравнений с запаздыванием. Синтаксис вызова этой функции показан в Листинге 4.2. Описание входных и выходных аргументов этой функции приведено в таблицах 11 и 12 соответственно.

```
[SS, Resid]=SComputation(coef, RHS, SSFinder, tol)
```

Листинг 4.2 Синтаксис вызова функции `SComputation`Таблица 11. Входные аргументы функции `SComputation`

Аргумент	Описание
<code>coef</code>	массив структур с полями, содержащими значения параметров системы, для которых вычисляются стационарные решения (см. подраздел 4.2.1.1)
<code>RHS</code>	функция, определяющая правую часть системы и описанная в подразделе 4.2.1.2
<code>SSFinder</code>	функция, определяющая правую часть системы, модифицированную для вычисления решений нелинейной системы с помощью пакета <code>Mathematica</code>
<code>tol</code>	положительная скалярная величина, определяющая требуемую невязку вычисления стационарных решений

Таблица 12. Выходные аргументы функции `SComputation`

Аргумент	Описание
<code>SS</code>	$n \times S$ матрица вычисленных стационарных решений где $S$ - общее число вычисленных стационарных решений. Каждый столбец этой матрицы — вектор переменных в соответствующем стационарном решении
<code>Resid</code>	$S$ -компонентный вектор невязок вычисленных стационарных решений

```
SSfinal = STracing(RHS, SSFinder, coef, delays, parname, par, tol, ...
                  ATol, RTol)
```

Листинг 4.4 Синтаксис вызова функции `STracing`

#### 4.1.2.2 `SStability`

Функция `SStability` предназначена для вычисления ведущих собственных значений и собственных векторов задачи на собственные значения (2.11). Синтаксис вызова этой функции показан в Листинге 4.3. Описание входных и выходных аргументов этой функции приведено в таблицах 13 и 14 соответственно.

```
[EiVals, EiVects, EiValsRes]=SStability(SS, ...
coef, delays, RHS, EigValNum, apFin, apScheme, R, lmaxlp, ...
resmin, deflation, addinf)
```

Листинг 4.3 Синтаксис вызова функции `SStability`

#### 4.1.2.3 `STracing`

Этот раздел посвящен краткому описанию функции `STracing`, не зависящей от модели. Эта функция предназначена для исследования зависимости стационарных решений систем дифференциальных уравнений с запаздыванием от параметров этих систем. Синтаксис вызова этой функции показан в Листинге 4.4. Описание входных и выходных аргументов этой функции приведено в таблицах 15 и 16 соответственно.



Таблица 13. Входные аргументы функции `SStability`

Аргумент	Описание
<code>SS</code>	$n$ -компонентный вектор значений переменных в стационарном решении
<code>coef</code>	см. таблицу 11
<code>delays</code>	массив структур с полями, содержащими значения задержек системы (см. подраздел 4.2.1.1)
<code>RHS</code>	см. таблицу 11
<code>EigValNum</code>	положительная скалярная величина, определяющая количество ведущих собственных значений, необходимых для вычисления
<code>apFin</code>	достаточно малая положительная скалярная величина, определяющая точность аппроксимации нелинейной собственной проблемы с помощью рациональной собственной проблемы
<code>apScheme</code>	положительная скалярная величина, определяющая схему аппроксимации нелинейной собственной проблемы рациональной собственной проблемой (1 — неявный метод Эйлера, 2 — BDF2)
<code>R</code>	положительная скалярная величина, задающая параметр метода последовательных линейных проблем и определяющая радиус области, в которой должно содержаться вычисленное собственное значение
<code>lmaxlp</code>	положительная скалярная величина, указывающая максимальное число итераций метода последовательных линейных проблем
<code>resmin</code>	достаточно малая положительная скалярная величина, определяющая заданную норму спектральной невязки вычисленных собственных значений
<code>deflation</code>	положительная скалярная величина для выбора между режимами дефляции описанными в [46]
<code>addinf</code>	булева переменная, указывающая, нужно ли показывать информационные сообщения в командном окне

Таблица 14. Выходные аргументы функции `SStability`

Аргумент	Описание
<code>EiVals</code>	$L$ -компонентный вектор собственных значений, где $L$ — число ведущих собственных значений, заданных в аргументах <code>EigValNum</code>
<code>EiVects</code>	$n \times L$ матрица собственных векторов, соответствующих собственным значениям в <code>EiVals</code>
<code>EiValsRes</code>	положительная скалярная величина, значение которой максимальная норма невязки от вычисленных собственных значений

#### 4.1.2.4 `SOptDistComputation`

`SOptDistComputation` предназначена для вычисления оптимального возмущения заданного устойчивого стационарного решения системы дифференциальных уравнений с запаздыванием вида (1.1). Синтаксис вызова этой функции приведен в листинге 4.5. Описание входных и выходных аргументов этой функции приведено в таблицах 17 и 18 соответственно.

```
[optDist, norm2Ud, normW21Ud, Gammaopt, topt, Gamma, ...
localNorms, Sigmas, norm2Udopt] = SOptDistComputation(SS, ...
coefs, delays, RHS, T, delta, method, lanczmax, ...
```

Таблица 15. Входные аргументы функции `STracing`

Аргумент	Описание
<code>RHS</code>	функция, определяющая правую часть системы и описанная в подразделе 4.2.1.2
<code>SSFinder</code>	функция, определяющая правую часть системы, модифицированную для вычисления решений нелинейной системы с помощью пакета <code>Mathematica</code>
<code>coef</code>	массив структур с полями, содержащими значения параметров системы (см. подраздел 4.2.1.1)
<code>delays</code>	массив структур с полями, содержащими значения задержек в системе (см. подраздел 4.2.1.1)
<code>parname</code>	строковый массив определяющий название параметра в формате <code>LaTeX</code> для подписи в картинке
<code>par</code>	$N$ компонентный вектор, содержащий сетку $s^1, \dots, s^N$ , построенную в интервале варьирования параметра
<code>tol</code>	см. таблицу 11
<code>ATol</code>	положительная скалярная величина, определяющая требуемую абсолютную точность решения задачи Коши вида (2.17)
<code>RTol</code>	положительная скалярная величина, определяющая требуемую относительную точность решения задачи Коши вида (2.17)

Таблица 16. Выходные аргументы функции `STracing`

Аргумент	Описание
<code>SSfinal</code>	массив ячеек $S \times 3$ , где $S$ число полученных подмножеств вида (2.16), первый столбец ячеек содержит значения $s_1, \dots, s_r$ для каждого набора вида (2.21), второй столбец ячеек содержит значения $u_1, \dots, u_r$ для каждого набора вида (2.21), третий столбец ячеек содержит невязки вычисленных стационарных решений $u_i$

```

5 lancztol, l, computeM1, difScheme, W21weight, ...
  locNormWghts, diagDM, disturbVars, maximizeVars, Qfuncs, ...
  isClean, marks, vp, hp, varnames)

```

Листинг 4.5 Синтаксис вызова функции `SOptDistComputation`

Таблица 17. Входные аргументы функции `SOptDistComputation`

Аргумент	Описание
<code>SS</code>	см. таблицу 13
<code>coefs</code>	см. таблицу 11
задержки	см. таблицу 13
<code>RHS</code>	см. таблицу 11
<code>T</code>	положительная скалярная величина, определяющая произвольную априорную верхнюю границу времени, на котором максимальная амплификация (2.24) достигает своего максимума
<code>delta</code>	положительная скалярная величина, определяющая шаг $\delta$ для равномерной по времени сетки (1.5) для вычисления оптимального возмущения
<code>method</code>	положительная скалярная величина, указывающая метод вычисления оптимального возмущения (1 — базовый алгоритм, описанный в разделе 2.5, 2 — метод Ланцоша описанный в разделе 2.6, 3 — алгоритм последовательной максимизации, описанный в разделе 2.7)
<code>lanczmax</code>	положительная скалярная величина, определяющая максимальное число шагов $r_{\max}$ метода Ланцоша (используется в случае аргумента <code>method</code> равного 2 или 3)
<code>lancztol</code>	положительная скалярная величина, определяющая минимально допустимое относительное увеличение <code>tol</code> приближенного сингулярного значения за один шаг (используется в случае аргумента <code>method</code> равного 2 или 3)
<code>l</code>	положительная целая скалярная величина, задающая приращение $l$ (см. раздел 2.5)
<code>computeMl</code>	булева переменная, указывающая, нужно ли вычисление матрицы $M^l$ , где $M$ - матрица в (2.29) и $l$ - входной аргумент <code>l</code>
<code>difScheme</code>	положительная скалярная величина, определяющая разностную схему для вычисления оптимального возмущения (1 — неявный метод Эйлера, 2 — BDF2)
<code>W2lweight</code>	положительная скалярная величина в интервале $[0, 1]$ , которая определяет параметр $\rho$ в локальной норме (2.22)
<code>locNormWghts</code>	положительная скалярная величина, определяющая тип диагональной матрицы $D$ в локальной норме (2.22) (1 — $D$ — единичная матрица, 2 — диагональные элементы $D$ являются обратными величинами к компонентам входного вектора <code>SS</code> )
<code>diagDM</code>	$n$ -компонентный вектор, содержащий веса компонент решения
<code>disturbVars</code>	$n$ -компонентный вектор из 0 и 1, содержащий маску, указывающую, какие переменные следует возмущать
<code>maximizeVars</code>	$n$ -компонентный вектор из 0 и 1, содержащий маску, указывающую, какие переменные используются при максимизации
<code>Qfuncs</code>	$n$ -компонентный массив ячеек функций, каждая из которых задает подпространство базисных функций, определенных на равномерной сетке (1.5) для $k = -m_q + 1, -m_q + 2, \dots, 0$ для соответствующей системной переменной. Для определения функций, являющихся элементами такого массива ячеек, рекомендуется использовать функцию <code>BasisSubspace</code> , описанную в подразделе 4.1.2.5
<code>isClean</code>	булева переменная, указывающая, нужна ли предварительная очистка рисунков
<code>marks</code>	массив из 3 ячеек, содержащий отметки для построения графика максимальной амплификации, $L_2$ нормы оптимального возмущения и $W_2^1$ нормы оптимального возмущения соответственно
<code>vp</code>	положительная скалярная величина, определяющая число подграфиков, располагающихся вертикально на рисунке
<code>hp</code>	положительная скалярная величина, определяющая число подграфиков, располагающихся горизонтально на рисунке
<code>varnames</code>	$n$ -компонентный массив ячеек с именами переменных системы в формате LaTeX для отображения на рисунках (см. подраздел 4.2.1.3)

Таблица 18. Выходные аргументы функции `SOptDistComputation`

Аргумент	Описание
<code>optDist</code>	2-компонентный массив ячеек, первая ячейка которого является $n \times N$ матрицей, содержащей сеточный аналог начального значения вычисленного оптимального возмущения для данного стационарного решения <code>SS</code> , где $N$ — число узлов равномерной сетки, построенной в интервале $(-\tau_q, 0]$ с шагом <code>delta</code> и $\tau_q$ - максимальная задержка; вторая ячейка - это $N$ -компонентный вектор, содержащий эту равномерную сетку
<code>norm2Ud</code>	$n$ -компонентный вектор, содержащий $L_2$ норму каждой компоненты начального значения вычисленного оптимального возмущения
<code>normW21Ud</code>	положительная скалярная величина, которая является $W_2^1$ нормой начального значения вычисленного оптимального возмущения
<code>Gammaopt</code>	положительная скалярная величина, которая является максимумом максимальной амплификации в интервале времени $(0, T)$ , где $T$ — значение аргумента <code>T</code>
<code>topt</code>	положительная скалярная величина, которая является временем, в которое максимальная амплификация равна <code>Gammaopt</code>
<code>Gamma</code>	2-компонентный массив ячеек, первая ячейка которого — вектор, содержащий максимальную амплификацию, вторая ячейка — вектор, содержащий временную сетку в узлах $t_k = l\delta k$ , $k = 0, \dots, [T/\delta]/l$ , где $T$ , $l$ , $\delta$ — значения аргументов <code>T</code> , <code>l</code> и <code>delta</code> соответственно
<code>localNorms</code>	3-компонентный массив ячеек, первая ячейка которого — вектор, содержащий локальную $L_2$ норму оптимального возмущения, вторая ячейка — вектор, содержащий локальную $W_2^1$ норму оптимального возмущения, третья ячейка — вектор, содержащий временную сетку в узлах $t_k = \delta k$ , $k = 0, \dots, [T/\delta]$ , в которых вычисляются локальные нормы
<code>Sigmas</code>	2-компонентный вектор, содержащий первое и второе максимальные сингулярные значения
<code>norm2Udopt</code>	$n$ -компонентный вектор, содержащий $L_2$ норму каждой компоненты оптимального возмущения, где оно достигает максимума

#### 4.1.2.5 BasisSubspace

`BasisSubspace` предназначен для получения функции, задающей подпространство базисных функций, определенных на равномерной сетке (1.5) для  $k = -m_q + 1, -m_q + 2, \dots, 0$  одного из следующих типов:

1. Целое пространство.
2. Подпространство гармонических функций.
3. Подпространство кусочно-постоянных функций.
4. Подпространство кусочно-линейных функций.
5. Подпространство кусочно-экспоненциальных функций.

6. Подпространство функций, качественно аппроксимирующих поведение лекарств в одно- и двухкамерных фармакокинетических моделях.

Рекомендуется использовать `BasisSubspace` для определения элементов массива ячеек `Qfuncs`, который является входным аргументом функции `SOptDistComputation`, описанной в подразделе 4.1.2.4.

Первым входным аргументом функции `BasisSubspace` является имя типа подпространства. Остальные аргументы зависят от выбранного типа подпространства. Независимо от типа подпространства, функция `BasisSubspace` всегда имеет только один выходной аргумент — функцию, которая принимает параметры сетки в качестве входных аргументов и возвращает базис на заданной сетке в указанном подпространстве.

Остальная часть этого подраздела посвящена описанию `BasisSubspace` в случаях каждого типа подпространства базисных функций. Синтаксис вызова этой функции в каждом типе подпространства показан в листинге 4.6.

```

% 1. Whole space
basisFunc = BasisSubspace(WholeSpace)

5 % 2. Harmonic functions
basisFunc = BasisSubspace(HarmonicFuncs, pmax)

% 3. Piecewise constant functions
basisFunc = BasisSubspace(PwConst, pwl)
10
% 4. Piecewise linear functions
basisFunc = BasisSubspace(PwLin, pwl)

% 5. Piecewise exponential functions of the form
15 % exp(alpha(t - t0)) - r where r = 0,1

% if r = 0:
basisFunc = BasisSubspace(PwExp, pwl, alpha)

20 % if r = 1:
basisFunc = BasisSubspace(PwExpMinusOne, pwl, alpha)

% 6. Piecewise exponential functions of the form
% exp(Alpha(x - x0)) - exp(Beta(x - x0)) where Beta < Alpha

```

```

25 basisFunc = BasisSubspace(Pharmacokin, pwl, Alpha, Beta)
% with zeroing at the two rightmost grid nodes:
basisFunc = BasisSubspace(PharmacokinZeroing, pwl, Alpha, ...
30                               Beta)

```

Листинг 4.6 Синтаксис вызова функции `BasisSubspace` в каждом типе подпространства базисных функций

**Целое пространство.** В этом случае подпространства функция `BasisSubspace` имеет только один аргумент, который является именем типа подпространства, и она возвращает `basisFunc`, которая является функцией, которая принимает дискретный аналог максимальной задержки  $m_q$  в качестве входного аргумента и возвращает единичную матрицу порядка  $m_q$ .

**Подпространство гармонических функций.** Подпространство гармонических функций, определяемое функцией `BasisSubspace`, можно представить в виде

$$S(t) = \text{span}\{1, \cos(2\pi pt/m_q), \sin(2\pi pt/m_q)\}, \quad p = 1, \dots, p_{\max} \quad (4.1)$$

где  $p_{\max}$  - целое положительное значение, а  $m_q$  - дискретный аналог максимальной задержки.

В данном случае подпространства функция `BasisSubspace` имеет два аргумента: имя типа подпространства и `pmax`, которое является положительным скалярным целым значением, задающим  $p_{\max}$  в (4.1). Она возвращает `basisFunc`, которая является функцией, принимающей дискретный аналог максимального значения задержки  $m_q$  в качестве входного аргумента и возвращающей  $m_q \times (2p_{\max} + 1)$  матрицу, столбцы которой содержат гармонические функции, составляющие подпространство (4.1).

**Подпространство кусочно-постоянных функций.** В этом случае функция `BasisSubspace` имеет два аргумента: имя типа подпространства и `pwl` —  $d$ -компонентный вектор, задающий относительную длину подынтервалов, на которых определены кусочно-постоянные функции, где  $d$  - число подынтервалов. Возвращается `basisFunc`, которая является функцией, принимающей в качестве входного аргумента дискретный аналог максимального значения задержки  $m_q$  и возвращающей матрицу  $m_q \times d$ , столбцы которой содержат кусочно-

постоянные функции, постоянные на подынтервалах, определяемых входным вектором `pwl`.

**Подпространство кусочно-линейных функций.** В этом случае подпространства функция `BasisSubspace` имеет те же аргументы, что и в случае подпространства кусочно-постоянных функций. Она возвращает `basisFunc`, которая является функцией, принимающей в качестве входного аргумента дискретный аналог максимальной задержки  $m_q$  и возвращающей матрицу  $m_q \times d + 1$ , столбцы которой содержат кусочно-линейные функции.

**Подпространство кусочно-экспоненциальных функций.** Подпространство кусочно-экспоненциальных функций, определяемое функцией `BasisSubspace`, может быть представлено как

$$\psi(t, t_j) = \begin{cases} 0 & m_q \leq t < t_j \\ e^{\alpha(t-t_j)} - r & t_j \leq t < 0, \end{cases} \quad (4.2)$$

где  $j = -m_q + [1/\delta]i + 1$ ,  $i = 0, 1, \dots, d - 1$ ,  $d$  - число подынтервалов, заданное в `pwl`,  $\alpha$  и  $r$  - скалярные величины ( $r = 0$  или  $r = 1$ ).

В этом случае подпространства функция `BasisSubspace` имеет те же аргументы, что и в случае подпространства кусочно-постоянных функций и аргумент `alpha`, который является скалярной величиной, задающей  $\alpha$  в (4.2). Она возвращает `basisFunc`, которая является функцией, принимающей в качестве входных аргументов дискретный аналог максимального значения задержки  $m_q$  и шаг сетки  $\delta$  и возвращает матрицу  $m_q \times d$ , столбцы которой содержат кусочно-экспоненциальные функции вида (4.2).

**Подпространство функций, качественно аппроксимирующих поведение лекарств в одно- и двухкомпонентных фармакокинетических моделях.** Это подпространство кусочно-экспоненциальных функций, определяемых функцией `BasisSubspace`, может быть представлено как

$$\psi(t, t_j) = \begin{cases} 0 & m_q \leq t < t_j \\ e^{\alpha(t-t_j)} - e^{\beta(t-t_j)} & t_j \leq t < 0, \end{cases} \quad (4.3)$$

где  $j = -m_q + [1/\delta]i + 1$ ,  $i = 0, 1, \dots, d - 1$ ,  $d$  - число подынтервалов, заданное в `pwl`,  $\alpha$  и  $\beta$  - скалярные величины ( $\beta < \alpha$ ).

В этом случае подпространства функция `BasisSubspace` имеет те же аргументы, что и в случае подпространства кусочно-постоянных функций, аргумент `alpha`, скалярное значение, задающее  $\alpha$  в (4.3) и аргумент `beta`, который является скалярным значением, задающим  $\beta$  в (4.3). Она возвращает `basisFunc`, которая является функцией-ручкой, принимающей в качестве входных аргументов дискретный аналог максимального значения задержки  $m_q$  и шаг сетки  $\delta$  и возвращающей матрицу  $m_q \times d$ , столбцы которой содержат функции вида (4.3).

### 4.1.3 Вычисление и анализ периодических решений

Этот раздел посвящен краткому описанию функций, не зависящих от модели, предназначенных для вычисления и анализа устойчивости периодических решений систем дифференциальных уравнений с дискретными запаздываниями вида (1.1) и вычисления для них оптимальных возмущений. Алгоритмы, на которых основаны эти функции, описаны в разделах 3.1 — 3.4.

#### 4.1.3.1 PComputation

Функция `PComputation` предназначена для вычисления периодических решений системы дифференциальных уравнений с запаздыванием. Синтаксис вызова этой функции показан в Листинге 4.7. Описание входных и выходных аргументов этой функции приведено в таблицах 19 и 20 соответственно. Поскольку функция `PComputation` использует функцию `Solver` (см. подраздел 4.1.1), некоторые входные аргументы функции `PComputation` совпадают с аргументами `Solver` и для краткости не описаны в таблице 19.



Таблица 19. Входные аргументы функции `RComputation`

Аргумент	Описание
<code>coef</code>	массив структур с полями, содержащими значения параметров системы, для которых вычисляется периодическое решение (см. подраздел 4.2.1.1)
<code>delays</code>	массив структур с полями, содержащими значения задержек системы (см. подраздел 4.2.1.1)
<code>RHS</code>	функция, определяющая правую часть системы и описанная в подразделе 4.2.1.2
<code>deltaR</code>	положительная скалярная величина, определяющая шаг сетки для вычисления финальной невязки
<code>maxTE</code>	положительная скалярная величина, определяющая максимально допустимый период ведущей моды
<code>epsilon</code>	положительная скалярная величина, определяющая относительную норму ведущей моды ( $\varepsilon$ в (3.3))
<code>Ne</code>	положительная скалярная величина, определяющая время одного этапа интегрирования возмущенного стационарного решения в периодах ведущей моды (см. подраздел 3.1.1)
<code>Nfmax</code>	положительная скалярная величина, определяющая максимальное число шагов интегрирования (см. подраздел 3.1.1)
<code>thetaV</code>	положительная скалярная величина, определяющая максимальное относительное изменение вариации $u^1$ (см. подраздел 3.1.1)
<code>SS</code>	неустойчивое стационарное решение, в окрестности которого ищется периодическое решение
<code>EiValues</code>	$r$ -компонентный вектор, определяющий $r$ ведущих собственных значений системы, линеаризованной относительно неустойчивого стационарного решения, в окрестности которого ищется периодическое решение
<code>EiVectors</code>	$n \times r$ матрица, определяющая собственные векторы, соответствующие собственным значениям указанным в аргументе <code>EiValues</code>
<code>epsilonT</code>	положительная скалярная величина, определяющая относительную длину полуинтервала периода вариации ( $\varepsilon_T$ в подразделе 3.1.1)
<code>tolV</code>	положительная скалярная величина, определяющая значение $tol_{u^1}$ в подразделе 3.1.1 для уменьшения интервала
<code>rho</code>	положительная скалярная величина, определяющая значение высоты пика, ниже которой пики в спектре периодического решения игнорируются ( $\rho$ в подразделе 3.1.1)
<code>N_NM</code>	положительная скалярная величина, определяющая максимальное число шагов метода Ньютона
<code>tolNM</code>	положительная скалярная величина, определяющая требуемую точность метода Ньютона
<code>ksi</code>	небольшое положительное скалярное значение для аппроксимации произведения матрицы Якоби на вектор
<code>minrat</code>	положительная скалярная величина, определяющая минимально допустимое отношение неязок на двух последовательных шагах метода Ньютона
<code>tolLS</code>	пороговое отношение нормы остатков системы к норме ее правой части в методе GMRES (см. подраздел 3.1.3)
<code>N_LS</code>	положительная скалярная величина, определяющая максимальное количество шагов метода GMRES (см. подраздел 3.1.3)
<code>mrest</code>	положительная скалярная величина, определяющая максимальное количество перезапусков
<code>ExtrFig</code>	булева переменная, указывающая на необходимость дополнительных рисунков

```

function [PS,tPS,T,Res]=
                    PComputation(coef,delays,RHS,delta,deltaR,...
5                    scheme,err,interp1_method,inc,h,v,marks,...
                    varname,logind,maxTE,epsilon,Ne,Nfmax,...
                    thetaV,SS,EiValues,EiVectors,...
                    epsilonT,tolV,rho,N_NM,tolNM,ksi,...
                    minrat,tolLS,N_LS,mrest,ExtrFig)

```

Листинг 4.7 Синтаксис вызова функции PComputation

Таблица 20. Выходные аргументы функции PComputation

Аргумент	Описание
PS	матрица, содержащая вычисленное периодическое решение в интервале $[-\tau_q, T]$
tPS	вектор, содержащий временную сетку, в которой находится найденное периодическое решение
T	наименьший период найденного периодического решения
Res	норма невязки (3.8) найденного периодического решения

### 4.1.3.2 PStability

Функция PStability предназначена для вычисления ведущих собственных значений матрицы монодромии. Синтаксис вызова этой функции показан в Листинге 4.8. Описание входных и выходных аргументов этой функции приведено в таблицах 21 и 22 соответственно.

```

function [EiVals,EiVects,EiValsRes]=
                    PStability(PS,tPS,T,coef,delays,RHS,...
5                    EigValNum,delta,...
                    interp1_method,m,tol,...
                    nsmax,idispl)

```

Листинг 4.8 Синтаксис вызова функции PStability

Таблица 21. Входные аргументы функции PStability

Аргумент	Описание
PS	матрица, содержащая вычисленное периодическое решение в интервале $[-\tau_q, T]$
tPS	вектор, содержащий временную сетку, в которой задано найденное периодическое решение
T	положительная скалярная величина, определяющая наименьший период найденного периодического решения
coef	см. таблицу 19
delays	см. таблицу 19
RHS	см. таблицу 19
EigValNum	положительная скалярная величина, определяющая количество ведущих собственных значений, необходимых для вычисления
delta	положительное скалярное значение, определяющее шаг сетки для аппроксимации оператора монодромии матрицей монодромии
interp1_method	строковый массив, определяющий метод интерполяции: 'linear', 'spline', 'pchip'
m	положительное скалярное значение, определяющее начальную размерность инвариантного подпространства
tol	положительная скалярная величина, определяющая требуемое значение спектральной невязки
nsmax	положительное скалярное значение, определяющее максимальное число итераций метода Стюарта
idisp	булева переменная, указывающая, нужно ли показывать какие-либо информационные сообщения в командном окне

Таблица 22. Выходные аргументы функции PStability

Аргумент	Описание
EiVals	$L$ -компонентный вектор собственных значений, где $L$ - число ведущих собственных значений, заданных в аргументах EigValNum
EiVects	$n \times L$ матрица собственных векторов, соответствующих собственным значениям в EiVals
EiValsRes	положительная скалярная величина, определяющая спектральную невязку вычисленных собственных значений и собственных векторов

#### 4.1.3.3 POptDistComputation

POptDistComputation предназначена для вычисления оптимального возмущения заданного устойчивого периодического решения системы дифференциальных уравнений с запаздыванием вида (1.1). Синтаксис вызова этой функции приведен в листинге 4.9. Описание входных аргументов этой функции приведено в таблице 23. Так как алгоритм вычисления оптимальных возмущений периодических решений является модификацией алгоритма вычисления оптимальных возмущений стационарных решений, то большинство входных аргументов POptDistComputation совпадают с входными аргументами SOptDistComputation и для краткости не приводятся в таблице 23, а выходные аргументы полностью совпадают с аргументами SOptDistComputation.

```
[optDist, norm2Ud, normW21Ud, Gammaopt, topt, Gamma, ...
localNorms, Sigmas, norm2Udopt] = SOptDistComputation(PS,tS ...
coefs, delays, RHS, Topt, T, delta, difScheme, W21weight, ...
5 locNormWghts, diagDM, disturbVars, maximizeVars, Qfuncs, ...
isClean, marks, vp, hp, varnames)
```

Листинг 4.9 Синтаксис вызова функции `POptDistComputation`Таблица 23. Входные аргументы функции `POptDistComputation`

Аргумент	Описание
<code>PS</code>	$n \times M$ матрица, содержащая периодическое решение в узлах сетки, заданной аргументом <code>tS</code> . Здесь $M$ — число узлов сетки <code>tS</code> .
<code>tS</code>	строка длины $M$ , задающая сетку, в узлах которой задано периодическое решение
<code>Topt</code>	положительная скалярная величина, определяющая время, оптимальным в которое будет вычисленное возмущение

## 4.2 Модельно зависимые функции

### 4.2.1 Задание модели

Для задания модели динамики инфекционного заболевания и иммунного ответа, представляющей собой систему дифференциальных уравнений с запаздыванием, в пакете DEODAN необходимо создать две функции MATLAB, `GetPar_*` и `RHS_*`, где  $*$  означает название модели. Например, для модели LCMV, описанной в разделе 1.2.1, имена этих файлов могут быть `GetPar_LCMV` и `RHS_LCMV`, а для модели HBV, описанной в разделе 1.2.2, имена могут быть `GetPar_HBV` и `RHS_HBV`. Функция `GetPar_*` возвращает значения параметров модели, а функция `RHS_*` возвращает ее правую часть. В дополнение к функциям `GetPar_*` и `RHS_*` можно создать две функции MATLAB `LatexNames_*` и `SRHS_*`. Функция `LatexNames_*` возвращает имена переменных и параметров модели в формате LaTeX. Функция `SRHS_*` возвращает правую часть модели, модифицированную для вычисления стационарных решений (см. Раздел 2.1). Для того, чтобы численно интегрировать реализованную модель, необходимо

создать функцию `INTEGR_*`. В следующих подразделах описывается структура этих функций, используя модель HBV в качестве примера.

#### 4.2.1.1 `GetPar_*`

У функции `GetPar_*` нет входных аргументов. В этой функции значения параметров модели должны быть заданы в двух массивах структур `coef` и `delays`. Массив `coef` содержит поля, каждое из которых содержит значение одного из параметров модели, не включая задержки. Массив `delays` содержит поля, каждое из которых содержит значение одной из задержек модели. Для модели HBV пример кода такой функции показан в Листинге 4.10.

#### 4.2.1.2 `RHS_*`

У функции `RHS_*` один входной аргумент `coef` (см. Подраздел 4.2.1.1). В этой функции правая часть модели должна быть задана в двух символьных массивах `UZ`, `F` и одном массиве структур `UD`. Массив `UZ` содержит символьные переменные, соответствующие компонентам вектора  $u(t - \tau_0)$ . Массив `UD` содержит поля, каждое из которых представляет собой массив символьных переменных, соответствующих компонентам вектора  $u(t - \tau_i)$ , где  $i$  - номер поля. Массив `F` содержит символьное выражение правой части модели. Для модели HBV пример кода такой функции показан в листинге 4.11.

#### 4.2.1.3 `LatexNames_*`

У функции `LatexNames_*` нет входных аргументов. В этой функции имена переменных модели и параметров модели в формате LaTeX должны быть заданы в трех массивах ячеек `varnames`, `coefnames` и `delaynames`. Массив `varnames`

```
function [coef,delays]=GetPar_HBV
.
.
5 .
  % Avogadro constant divided by 10e3
  N_A = 6.02e20;

  %parameters excluding delays
10  coef.M_0=1.0e-15*N_A;
    coef.H_E_0=1.0e-18*N_A;
    coef.H_B_0=1.0e-19*N_A;
    coef.E_0=1.0e-18*N_A;
    coef.B_0=1.0e-18*N_A;
15  coef.P_0=4.3e-22*N_A;
    coef.F_0=8.3e-14*N_A;
    coef.C_0=0.5e-12*N_A;

    .
    .
20 .
    delays.tau_H_E=0.6;
    delays.tau_H_B=0.6;
    delays.tau_E=2.0;
    delays.tau_B=2.0;
25  delays.tau_P=3.0;
```

Листинг 4.10 Задание параметров модели

```

function [UZ,UD,F]=RHS_HBV(coef)

syms V_f C_V m M_V H_E E_ H_B B P F_
5 syms V_f_tau_H_E C_V_tau_H_E m_tau_H_E M_V_tau_H_E H_E_tau_H_E
syms E_tau_H_E H_B_tau_H_E B_tau_H_E P_tau_H_E F_tau_H_E
syms V_f_tau_H_B C_V_tau_H_B m_tau_H_B M_V_tau_H_B H_E_tau_H_B
syms E_tau_H_B H_B_tau_H_B B_tau_H_B P_tau_H_B F_tau_H_B
10 syms V_f_tau_E C_V_tau_E m_tau_E M_V_tau_E H_E_tau_E E_tau_E
syms H_B_tau_E B_tau_E P_tau_E F_tau_E
syms V_f_tau_B C_V_tau_B m_tau_B M_V_tau_B H_E_tau_B E_tau_B
syms H_B_tau_B B_tau_B P_tau_B F_tau_B
15 syms V_f_tau_P C_V_tau_P m_tau_P M_V_tau_P H_E_tau_P E_tau_P
syms H_B_tau_P B_tau_P P_tau_P F_tau_P

UZ=[V_f;C_V;m;M_V;H_E;E_;H_B;B;P;F_];
UD.tau_H_E=[V_f_tau_H_E;C_V_tau_H_E;m_tau_H_E;M_V_tau_H_E;...
20 H_E_tau_H_E;E_tau_H_E; H_B_tau_H_E;B_tau_H_E; ...
P_tau_H_E;F_tau_H_E];
UD.tau_H_B=[V_f_tau_H_B;C_V_tau_H_B;m_tau_H_B;M_V_tau_H_B;...
H_E_tau_H_B;E_tau_H_B;H_B_tau_H_B;B_tau_H_B;P_tau_H_B;...
F_tau_H_B];
25 UD.tau_E=[V_f_tau_E;C_V_tau_E;m_tau_E;M_V_tau_E;H_E_tau_E;...
E_tau_E;H_B_tau_E;B_tau_E;P_tau_E;F_tau_E];
UD.tau_B=[V_f_tau_B;C_V_tau_B;m_tau_B;M_V_tau_B;H_E_tau_B;...
E_tau_B;H_B_tau_B;B_tau_B;P_tau_B;F_tau_B];
UD.tau_P=[V_f_tau_P;C_V_tau_P;m_tau_P;M_V_tau_P;H_E_tau_P;...
30 E_tau_P;H_B_tau_P;B_tau_P;P_tau_P;F_tau_P];
%
xi = 1 - m / coef.C_0;
F(1)=(coef.nu*C_V +coef.n*coef.b_CE*C_V*E_) - ...
coef.gamma_VC*V_f*(coef.C_0-C_V-m) - ...
35 coef.gamma_VF*V_f*F_ -coef.gamma_VM*V_f;
F(2)=coef.sigma*V_f*(coef.C_0-C_V-m) -coef.b_CE*C_V*E_ - ...
coef.b_m*C_V;
F(3)=coef.b_CE*C_V*E_ +coef.b_m*C_V -coef.alpha_m*m;
F(4)=coef.gamma_MV*coef.M_0*V_f -coef.alpha_M*M_V;
40 F(5)= (coef.b_H_E*(xi*coef.rho_H_E*M_V_tau_H_E*H_E_tau_H_E - ...
M_V*H_E) -coef.b_p_H_E*M_V*H_E*E_) +coef.alpha_H_E*...
(coef.H_E_0-H_E);
F(6)=coef.b_p_E*(xi*coef.rho_E*M_V_tau_E*H_E_tau_E*E_tau_E - ...
M_V*H_E*E_) - .coef.b_EC*C_V*E_ +coef.alpha_E*(coef.E_0-E_);
45 F(7)=(coef.b_H_B*(xi*coef.rho_H_B*M_V_tau_H_B*H_B_tau_H_B - ...
M_V*H_B) -coef.b_p_H_B*M_V*H_B*B) +coef.alpha_H_B*...
(coef.H_B_0-H_B);
F(8)=coef.b_p_B*(xi*coef.rho_B*M_V_tau_B*H_B_tau_B*B_tau_B - ...
M_V*H_B*B) +coef.alpha_B*(coef.B_0-B);
50 F(9)=coef.b_p_P* xi*coef.rho_P*M_V_tau_P*H_B_tau_P*B_tau_P + ...
coef.alpha_P*(coef.P_0-P);
F(10)=coef.rho_F*P -coef.gamma_FV*F_*V_f -coef.alpha_F*F_;

```

```

function [varnames , coefnames , delaynames]=LatexNames_HBV
.
.
5 .
varnames{1,1}=$V_f$;
varnames{1,2}=$C_V$;
varnames{1,3}=$m$;
varnames{1,4}=$M_V$;
10 varnames{1,5}=$H_E$;
.
.
.
coefnames{1,9}=$\alpha_M$;
15 coefnames{1,10}=$\alpha_H^E$;
coefnames{1,11}=$\alpha_H^B$;
coefnames{1,12}=$\alpha_E$;
coefnames{1,13}=$\alpha_B$;
coefnames{1,14}=$\alpha_P$;
20 .
.
.
delaynames{1, 1} = $\tau_{H_E}$;
delaynames{1, 2} = $\tau_{H_B}$;
25 delaynames{1, 3} = $\tau_E$;
delaynames{1, 4} = $\tau_B$;
delaynames{1, 5} = $\tau_P$;

```

Листинг 4.12 Задание имен переменных и параметров модели в формате LaTeX

содержит ячейки, каждая из которых содержит имя соответствующей переменной модели в формате LaTeX. Массив `coefnames` содержит ячейки, каждая из которых содержит имя соответствующего параметра модели, исключая задержки, в формате LaTeX. Массив `delaynames` содержит ячейки, каждая из которых содержит имя соответствующей задержки модели в формате LaTeX. Для модели HBV пример задания некоторых имен ее переменных и параметров в формате LaTeX показан в листинге 4.12.



#### 4.2.1.4 SRHS\_\*

У функции `SRHS_*` один входной аргумент `coef` (см. Подраздел 4.2.1.1). Файл, в котором содержится эта функция, используется функцией `SInitialization_*` (см. подраздел 4.2.2.2), и в этой функции правая часть модели, модифицированная для вычисления стационарных решений модели, должна быть указана в двух символьных массивах `varsol` и `Fv`. Массив `varsol` содержит символьные переменные, соответствующие переменным системы, для которой должна быть решена нелинейная система (2.1). Массив `Fv` содержит символьное выражение измененной правой части модели. Кроме того, в этой функции должны быть указаны явные выражения для соответствующих переменных модели, как функций переменных из массива `varsol`. Для модели HBV пример кода такой функции показан в листинге 4.13. Следует заметить, что в строках 4–17 этого листинга указаны упомянутые явные выражения (см. Подраздел 2.1.2).

#### 4.2.1.5 INTEGR\_\*

У функции `INTEGER_*` нет входных аргументов. Эта функция численно интегрирует реализованную модель до некоторого заданного момента времени. В начале этой функции должны быть указаны параметры для интегрирования по времени, которые используются функцией `Solver`, не зависящей от модели. Описание этих параметров приведено в подразделе 4.1.1, как описание входных аргументов функции `Solver`. Пример указания некоторых из этих параметров показан в Листинге 4.14.

```

function [varsol, Fv]=SRHS_HBV(coefs)

syms V_f C_V H_E H_B
5 m = (coefs.sigma*V_f*(coefs.C_0 - C_V))/(coefs.alpha_m + ...
    coefs.sigma*V_f);
xi = 1 - m / coefs.C_0;
E_ = (coefs.alpha_m*m - coefs.b_m*C_V)/(coefs.b_CE*C_V);
M_V = coefs.gamma_MV*coefs.M_0*V_f/coefs.alpha_M;
10 K = ((coefs.nu*C_V + coefs.n*coefs.b_CE*C_V*E_ - ...
    coefs.gamma_VM*V_f - coefs.gamma_VC*V_f*(coefs.C_0 - ...
    C_V - m))*(coefs.gamma_FV*V_f + coefs.alpha_F)/...
    (coefs.rho_F*coefs.gamma_VF*V_f) - coefs.P_0)*...
    (coefs.alpha_P/(coefs.b_p_P*xi*coefs.rho_P));
15 P = (coefs.b_p_P*xi*coefs.rho_P*K)/coefs.alpha_P + coefs.P_0;
F_ = (coefs.rho_F*P)/(coefs.gamma_FV*V_f + coefs.alpha_F);
B = (coefs.b_p_B*(xi*coefs.rho_B - 1))*K/coefs.alpha_B + ...
    coefs.B_0;
Fv(1)=(coefs.b_H_B*(xi*coefs.rho_H_B-1))*M_V*H_B- ...
20 coefs.b_p_H_B*K+coefs.alpha_H_B*(coefs.H_B_0-H_B);
Fv(2)=M_V*B*H_B - K;
Fv(3)= (coefs.b_H_E*(xi*coefs.rho_H_E-1))*M_V*H_E- ...
    coefs.b_p_H_E*M_V*H_E*E_+coefs.alpha_H_E*(coefs.H_E_0- ...
    H_E);
25 Fv(4)=(coefs.b_p_E*(xi*coefs.rho_E-1))*M_V*H_E*E_- ...
    coefs.b_EC*C_V*E_+coefs.alpha_E*(coefs.E_0-E_);
varsol=[V_f; C_V; H_E; H_B];

```

Листинг 4.13 Задание правой части модели, модифицированной для вычисления стационарных решений

```

function INTEGR_HBV
.
.
5 .
  % I.Solver parameters
  %
  % Time of integration:
  T=1000;
10 % Grid step:
  delta=1.0e-3;
  % Integration scheme:
  scheme=2; % 1 - implicit Euler method,
           % 2 - BDF2
15
  % Accuracy of the Newton method:
  err=1.0e-9;

```

Листинг 4.14 Задание параметров для интегрирования модели по времени

## 4.2.2 Вычисление и анализ стационарных решений

Для вычисления стационарных решений модели динамики инфекционных заболеваний и иммунного ответа, представляющей собой систему дифференциальных уравнений с запаздыванием, и анализа их устойчивости в рамках пакета DEODAN необходимо создать две функции MATLAB, `SInitialization_*` и `SAnalysis_*`, где `*` означает название модели. Для анализа зависимости стационарных решений модели динамики инфекционных заболеваний и иммунного ответа от параметра в рамках пакета DEODAN необходимо создать функцию MATLAB `STrace_*`. Для вычисления оптимального возмущения стационарного решения модели динамики инфекционной болезни и иммунного ответа в рамках пакета DEODAN необходимо создать функцию MATLAB `SOptDist_*`, где `*` означает название модели. В следующих подразделах кратко описаны функции `SInitialization_*`, `SAnalysis_*`, `STrace_*` и `SOptDist_*`.

### 4.2.2.1 SInitialization\_\*

У функции `SInitialization_*` нет входных аргументов. Эта функция создает файл MATLAB `SSfinder_*.m`, содержащий функцию `SSfinder_*`, используемую для вычисления решения системы (2.1) с помощью процедуры `NSolve` пакета `Mathematica`. Если никаких предварительных преобразований не производится, т.е. если файл `SRHS_*.m` не создан, `SInitialization_*` создает файл `SSfinder.m` из файла `RHS_*.m`, в противном случае она создает файл `SSfinder_*.m` из файла `SRHS_*.m`. В этой функции должны быть указаны четыре строковых массива `ModelFolderName`, `NewSSFinderFileName`, `ModelRHSFileName`, `ModelSRHSFileName`. Массив `ModelFolderName` содержит имя папки, в которой расположены все функции, зависящие от модели. Массив `NewSSFinderFileName` содержит имя файла `SSfinder_*.m`. Массив `ModelRHSFileName` содержит имя файла `RHS_*.m`. Массив `ModelSRHSFileName` содержит имя файла `SRHS_*.m`, если этот файл создан и пуст в противном случае. Для модели HBV пример задания этих массивов показан в Листинге 4.15.

### 4.2.2.2 SAnalysis\_\*

У функции `SAnalysis_*` нет входных аргументов. Эта функция вычисляет все стационарные решения и анализирует устойчивость вычисленных стационарных решений. В начале этой функции должны быть указаны параметры для вычисления стационарных решений и анализа их устойчивости. Эти параметры используются независимыми от модели функциями `SComputation` и `SStability`. Описание этих параметров приведено в подразделах 4.1.2.1, 4.1.2.2 как входных аргументов функций `SComputation` и `SStability`. Пример задания некоторых из этих параметров показан в листинге 4.16.

```

function SInitialization
.
.
5 .
  % SETTING PARAMETERS:
  %
  % Folder name where the model files are located:
  ModelFolderName      = HBV;
10
  % Name of the SSFinder file that will be generated in the
  % given folder:
  NewSSFinderFileName = SSFinder_HBV.m;

15 % Name of the RHS file in the given folder:
  ModelRHSFileName    = RHS_HBV.m;

  % Name of the SRHS file in the given folder (if you dont want
  % use the SRHS file to generate SSFinder write instead of
20 % the file name):
  ModelSRHSFileName   = SRHS_HBV.m;

```

Листинг 4.15 Задание строковых массивов для функции SInitialization\_\*

```

function SAnalysis_LCMV
.
.
5 .
  % I. Parameters for stationary solutions computation:
  tol=1e-6;
  % II. Parameters for stationary solutions stability analysis:
  p=10;
10 delta=1.0e-2;
  scheme=2;
  R=10.0;
  lmaxlp=10;
  resmin=1.0e-12;

```

Листинг 4.16 Задание параметров для вычисления и анализа устойчивости стационарных решений

### 4.2.2.3 STrace\_\*

У функции `STrace_*` нет входных аргументов. Эта функция вычисляет все стационарные решения модели как функции параметров модели. В начале этой функции должны быть указаны параметры для вычисления стационарных решений и анализа их зависимости от параметра. Эти параметры используются независимой от модели функции `STracing`. Описание этих параметров приведено в подразделе 4.1.2.3 как входных аргументов функции `STracing`. Пример задания некоторых из этих параметров показан в листинге 4.17.

```

function STrace_HBV
.
.
5 .
%
% I. Parameters for stationary solutions tracing:
%
10 parname=gamma_MV;
%
N_A=6.02e20;
par=linspace(1e8/N_A,1e10/N_A,30);
15 %
ATol=1e-14;
RTol=1e-14;
%
iclean=1;
20 %
v=2;
h=5;

```

Листинг 4.17 Задание параметров для анализа стационарных решений от параметра

#### 4.2.2.4 SOptDist\_\*

У функции SOptDist\_\* нет входных аргументов. Эта функция вычисляет оптимальные возмущения заданного стационарного решения. В начале этой функции должны быть указаны параметры для вычисления оптимальных возмущений, которые используются независимыми от модели функциями SOptDistComputation и BasisSubspace. Описание этих параметров приведено в подразделах 4.1.2.4, 4.1.2.5, как описание аргументов этих функций. Пример задания некоторых из этих параметров показан в Листинге 4.18.

```
function SOptDist_HBV
.
.
5 .
T = 30;
delta = 5e-3;
difScheme = 2;
optDistCompMethod=1;
10 l = 50;
```

Листинг 4.18 Задание параметров для вычисления оптимальных возмущений

### 4.2.3 Вычисление и анализ периодических решений

Для вычисления периодических решений модели динамики инфекционного заболевания и иммунного ответа, представляющей собой систему дифференциальных уравнений с запаздыванием, и анализа их устойчивости в рамках пакета DEODAN необходимо создать функцию MATLAB PAnalysis\_\*, где \* означает название модели. Для вычисления оптимального возмущения периодического решения модели динамики инфекционной болезни и иммунного ответа в рамках пакета DEODAN необходимо создать функцию MATLAB POptDist\_\*. В этом разделе кратко описаны все эти функции.

```

function PAnalysis_LCMV
.
.
5  % I. Parameters for periodic solution computation:
.
.
.
10 %
% 2. Parameters of the first stage (relaxation method):
%
maxTE=1000;      % maximum allowed period of the leading mode
epsilon=5e3;    % relative norm of the leading mode
15 Ne=30; % time of one stage of integration of the disturbed
           % stationary solution in the periods of the leading mode
Nfmax=15; % maximum number of integration steps
thetaV=1e-7; % maximum relative deviation Vmax at two
           %consecutive stages (for the stopping criterion)

```

Листинг 4.19 Задание параметров для вычисления и анализа устойчивости периодического решения

#### 4.2.3.1 PAnalysis\_\*

У функции `PAnalysis_*` нет входных аргументов. Эта функция вычисляет периодическое решение в окрестности неустойчивого стационарного решения и анализирует устойчивость вычисленного периодического решения. В начале этой функции должны быть указаны параметры для вычисления периодического решения и анализа его устойчивости, которые используются модельными независимыми функциями `PComputation` и `PStability`. Описание этих параметров приведено в подразделах 4.1.3.1, 4.1.3.2, как и аргументов функций `PComputation` и `PStability`. Пример задания некоторых из этих параметров показан в листинге 4.19.



### 4.2.3.2 POptDist\_\*

У функции POptDist\_\* нет входных аргументов. Эта функция вычисляет оптимальные возмущения заданного периодического решения. В начале этой функции должны быть указаны параметры для вычисления оптимальных возмущений, которые используются независимыми от модели функциями POptDistComputation и BasisSubspace. Описание этих параметров приведено в подразделах 4.1.3.3, 4.1.2.5, как описание аргументов этих функций. Пример задания некоторых из этих параметров показан в Листинге 4.20.

```
function POptDist_HBV
.
.
5 .
T = 30;
delta = 5e-3;
difScheme = 2;
W21weight = 1;
```

Листинг 4.20 Задание параметров для вычисления оптимальных возмущений

## 4.3 Сравнение с пакетом DDE-BIFTOOL

Данный раздел посвящен сравнению разработанного пакета программ DEODAN с наиболее совпадающим по функционалу пакетом DDE-BIFTOOL. Совпадение функционала этих пакетов заключается в том, что оба пакета предназначены для бифуркационного анализа систем дифференциальных уравнений с запаздыванием. Однако пакет DDE-BIFTOOL предназначен для работы не только с системами с постоянными запаздываниями вида (1.1), для работы с которыми предназначен пакет DEODAN, но и для работы с системами с переменными запаздываниями, зависящими от состояния системы.

Основными преимуществами пакета DDE-BIFTOOL по сравнению с пакетом DEODAN являются используемые в этом пакете эффективные алгоритмы трассирования одного заданного стационарного или периодического решения по параметру и алгоритмы вычисления и классификации точек бифуркации. Существенными преимуществами пакета DEODAN по сравнению с пакетом DDE-BIFTOOL являются следующие его свойства. Во-первых, пакет DEODAN требует от пользователя существенно меньше ручной работы, так как вычисление и анализ стационарных и периодических решений выполняются в автоматическом режиме. Во-вторых, в рамках пакета DEODAN рассматриваются задачи гарантированного вычисления всех стационарных решений заданной системы с дискретным запаздыванием и одновременного трассирования найденных стационарных решений по параметру, которые не рассматриваются в рамках пакета DDE-BIFTOOL. В-третьих для того, чтобы найти периодическое решение при фиксированных значениях параметров системы в рамках пакета DDE-BIFTOOL необходимо сначала найти параметр, по которому имеется бифуркация и точку бифуркации, вычислить периодическое решение при значении параметра, близком к точке бифуркации, а затем протрассировать найденное периодическое решение от этой точки до требуемого значения параметра, что может оказаться вычислительно затратным. В рамках же пакета DEODAN используется подход, который позволяет вычислять устойчивое периодическое решение в окрестности неустойчивого стационарного состояния при заданных параметрах модели, используя только информацию об этом неустойчивом стационарном состоянии и не требует ни предварительного бифуркационного анализа, ни трассирования по параметру.

Продемонстрируем последнее свойство на примере модели LCMV. Сравним результаты работы метода вычисления устойчивого периодического решения, предложенного в разделе 3.1 данной работы и реализованного в пакете DEODAN, который будем называть методом 1, с результатами работы метода вычисления периодического решения, реализованного в пакете DDE-BIFTOOL, который будем называть методом 2.

Для реализации метода 2 необходимо было выполнить трассирование по параметру и вычислить точку бифуркации. В качестве параметра, по которому выполнялось трассирование, был выбран параметр  $\beta$ , а точкой бифуркации является его значение 1.67. С увеличением значения  $\beta$  при переходе через точ-

ку бифуркации устойчивое стационарное состояние становится неустойчивым и в его окрестности появляется устойчивое периодическое решение с возрастающими с ростом  $\beta$  разбросом значений и минимальным периодом. В пакете DDE-BIFTOOL сначала периодическое решение вычисляется при некотором значении параметра, немного большем, чем точка бифуркации. Затем, это решение трассируется по параметру  $\beta$  до требуемого значения этого параметра. При этом шаг по параметру выбирается автоматически.

Для сравнения точности приближенных периодических решений найденных методами 1 и 2, мы использовали следующую единую и независимую от фазы периодического решения невязку

$$\frac{\left( \sum_{k=0}^{T/\delta} h_k \|D(U(\delta k + T) - U(\delta k))\|_2^2 dt \right)^{1/2}}{\left( \sum_{k=0}^{T/\delta} h_k \|DU(\delta k)\|_2^2 dt \right)^{1/2}}, \quad (4.4)$$

где  $h_0 = h_{T/\delta} = \delta/2$  и  $h_k = \delta$  при  $1 \leq k \leq T/\delta - 1$ . При этом требуемые значения  $U(\delta k)$  для метода 2 вычислялись интерполированием на сетку, используемую для метода 1, найденных значений периодического решения в интервале  $[0, \tau_A]$  и последующим численным интегрированием по схеме (1.4).

Во второй и третьей строках таблицы 24 приведены значения нормы невязки (4.4), отношений минимального и максимального значений переменной  $V$  в полученном периодическом решении к стационарному состоянию и минимальный период  $T$ , полученные с помощью обоих методов для значений параметров описанных в разделе 5.3, а также времена работы этих методов. В четвертой и пятой строках этой таблицы приведены аналогичные результаты для  $\beta = 2.71$ . Из этой таблицы видно, что метод, предложенный в данной работе, позволяет получать приближенное периодическое решение, с меньшей нормой невязки за меньшее время.

С помощью пакета DDE-BIFTOOL нам не удалось подобрать значения параметров метода 2, позволяющие находить достаточно точные приближенные периодические решения для  $\beta > 2.87$ . С помощью же пакета DEODAN достаточно точные приближенные периодические решения удавалось вычислять не меняя начальные значения параметров метода 1, указанные в таблице 25, и

для значений  $\beta > 2.87$ . Например, для  $\beta = 3.79$  были получены результаты, приведенные в шестой строке таблицы 24.

Таблица 24. Результаты вычисления периодического решения методами 1 и 2

$\beta$	Метод	Невязка	$V_{\min}/\bar{V}$	$V_{\max}/\bar{V}$	$T$	Время (сек.)
2.05	1	$5.2 \cdot 10^{-4}$	$9.917 \cdot 10^{-4}$	7.605	13.09	20
	2	$1.7 \cdot 10^{-3}$	$9.945 \cdot 10^{-4}$	7.603	13.09	32
2.71	1	$2.2 \cdot 10^{-4}$	$8.950 \cdot 10^{-12}$	20.135	20.73	24
	2	$2.4 \cdot 10^{-3}$	$9.490 \cdot 10^{-12}$	20.111	20.71	47
3.79	1	$4.5 \cdot 10^{-3}$	$2.734 \cdot 10^{-40}$	59.166	37.87	31

В заключение этого раздела, стоит отметить, что пакет DEODAN является единственным пакетом программ для вычисления оптимальных возмущений и анализа возможности перевода системы из одного устойчивого стационарного или периодического решения в другое с помощью таких возмущений.

## Глава 5. Численные эксперименты с моделями инфекционных заболеваний

### 5.1 Анализ стационарных решений модели LCMV

Продемонстрируем результаты работы алгоритмов вычисления стационарных решений, исследования их зависимости от параметра и исследования их устойчивости на примере модели LCMV (1.6) при значениях параметров, заданных в таблице 5 из работы [42] за исключением параметров  $\alpha_{E_e} = 0.3$ ,  $\alpha_{E_p} = 0.01$  и параметра  $b_p$ , который будем варьировать в диапазоне от  $10^{-5}$  до  $10^{-3}$ .

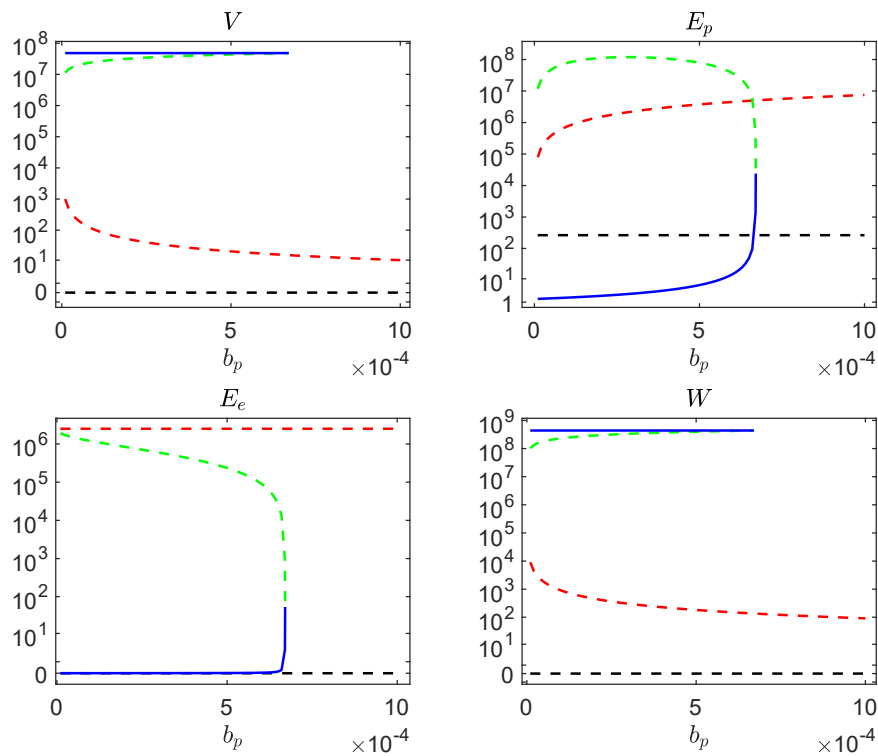


Рисунок 5.1 — Стационарные решения (I — черным, II — красным, III — зеленым, IV — синим) и их устойчивость (сплошная линия) и неустойчивость (пунктирная линия) как функции параметра  $b_p$ .

В результате работы алгоритмов, описанных в разделе 2.3, было найдено четыре различных набора вида (2.21). Подмножества вида (2.16), построенные по этим наборам, показаны на рисунке 5.1. Сплошными линиями обозначены

устойчивые стационарные решения, пунктирными — неустойчивые. Устойчивость для каждого набора исследовалась отдельно при каждом значении параметра  $b_p$ . Видно, что при  $10^{-5} \leq b_p \leq b_p^* \approx 6.7 \cdot 10^{-4}$ , имеется четыре стационарных решения: с  $V = 0$ ,  $V \approx 10^2$ ,  $V \approx 10^7$  и  $V = V^* \approx 4.82 \cdot 10^7$ , которые в дальнейшем будем называть решениями I, II, III и IV соответственно. В точке  $(b_p^*, V^*)$  с точностью до погрешности вычислений выполняется равенство (2.15). Поэтому, в интервале  $b_p^* < b_p \leq 10^{-3}$  остаются только решения I, II.

Рисунок 5.2 иллюстрирует изменение ведущих собственных значений в каждом из стационарных решений при варьировании параметра. На нем изображены 15–16 ведущих (с наибольшими вещественными частями) собственных значений при  $b_p = 10^{-5}$ ,  $5.1 \cdot 10^{-4}$  и  $10^{-3}$  для стационарных решений I, II и при  $b_p = 10^{-5}$ ,  $3.5 \cdot 10^{-4}$ ,  $6.7 \cdot 10^{-4}$  для стационарных решений III, IV. Маркеры «x», «+» и «o» обозначают собственные значения при первом, втором и третьем значениях параметра. Видно, что в решении I имеется лишь четыре различных собственных значения. Эти кратные собственные значения не меняются при варьировании параметра. Действительная часть самого правого из этих собственных значений больше нуля и поэтому решение I не является асимптотически устойчивым при всех рассмотренных значениях параметра.

В решении II четыре крайних правых собственных значения мало меняются при варьировании параметра. Остальные собственные значения при увеличении значения параметра смещаются влево. Действительная часть крайнего правого собственного значения больше нуля и поэтому решение II так же не является асимптотически устойчивым при всех рассмотренных значениях параметра.

В решении III собственные значения в целом смещаются влево при увеличении значения параметра, но при этом действительная часть крайнего правого собственного значения остается больше нуля, поэтому это стационарное решение так же не является асимптотически устойчивым.

В решении IV при увеличении значения параметра собственные значения сначала смещаются влево, а потом вправо. При начальном значении параметра  $b_p$  действительная часть крайнего правого собственного значения меньше нуля. При увеличении значения параметра это собственное значение не меняется, но второе собственное значение смещается вправо и при  $b_p \approx 6 \cdot 10^{-4}$  оно становится

крайним правым. Более того, при  $b_p = 6.7 \cdot 10^{-4}$  это собственное значение уже лежит в правой полуплоскости.

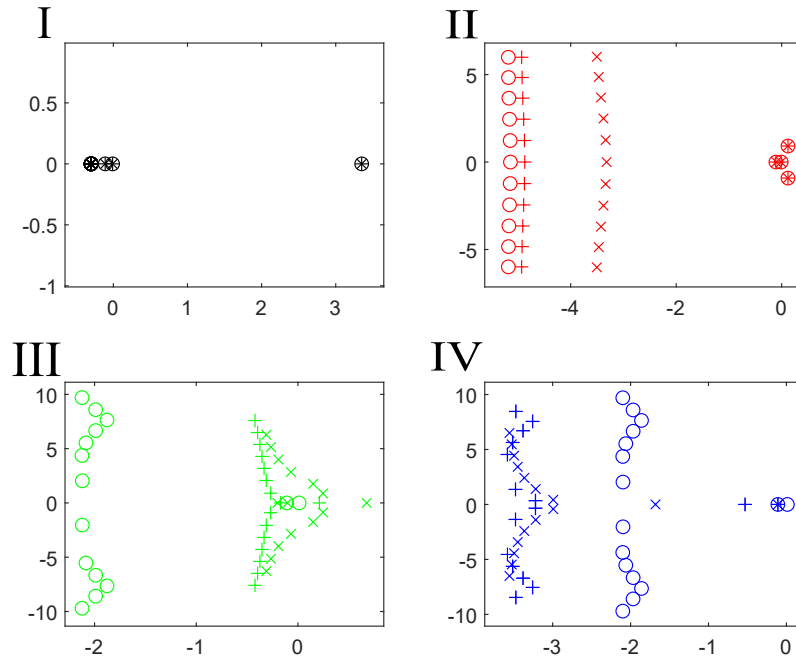


Рисунок 5.2 — Изменение ведущих собственных значений в решениях I, II, III и IV при варьировании параметра.

Таким образом, решение IV является устойчивым при всех рассмотренных значениях параметра, кроме последнего. Отметим, что при последнем значении параметра собственные значения в решениях III, IV визуально совпадают, как и сами эти решения.

Рассмотренный случай — случай, когда только одно из четырех стационарных решений является устойчивым, причем это решение с самой высокой вирусной нагрузкой. Для систем, описывающих динамику вирусных инфекций, важным является определение областей в пространстве параметров модели, где у системы есть как минимум два устойчивых стационарных решения.

Вычислив стационарные решения при значениях параметров, заданных в таблице 5 из работы [42] за исключением параметров  $\alpha_{E_e} = 0.3$ ,  $\alpha_{E_p} = 0.01$  и параметра  $\beta$ , который мы варьировали в диапазоне от  $10^{-3}$  до 5, и исследовав устойчивость этих решений, мы получили результат, изображенный на рисунке 5.3. Из этого рисунка видно, что у системы при указанных значениях параметров существует четыре стационарных решения с  $V = 0$ ,  $V \approx 1.3 \cdot 10^2$ ,  $V \approx 2.3 \cdot 10^7$  и  $V \approx 4.82 \cdot 10^7$ , которые, как и раньше будем нумеровать I, II, III и IV соответственно. Также из этого рисунка видно, что решение IV, является

устойчивым при любом значении параметра в интервале варьирования, а решение II является устойчивым в интервале  $10^{-3} \leq \beta < \beta^* \approx 1.718$ . То есть при значениях параметров, заданных в таблице 5 из работы [42] и любом значении параметра  $\beta$ , удовлетворяющим неравенству  $10^{-3} \leq \beta < \beta^*$  у системы есть два устойчивых стационарных решения.

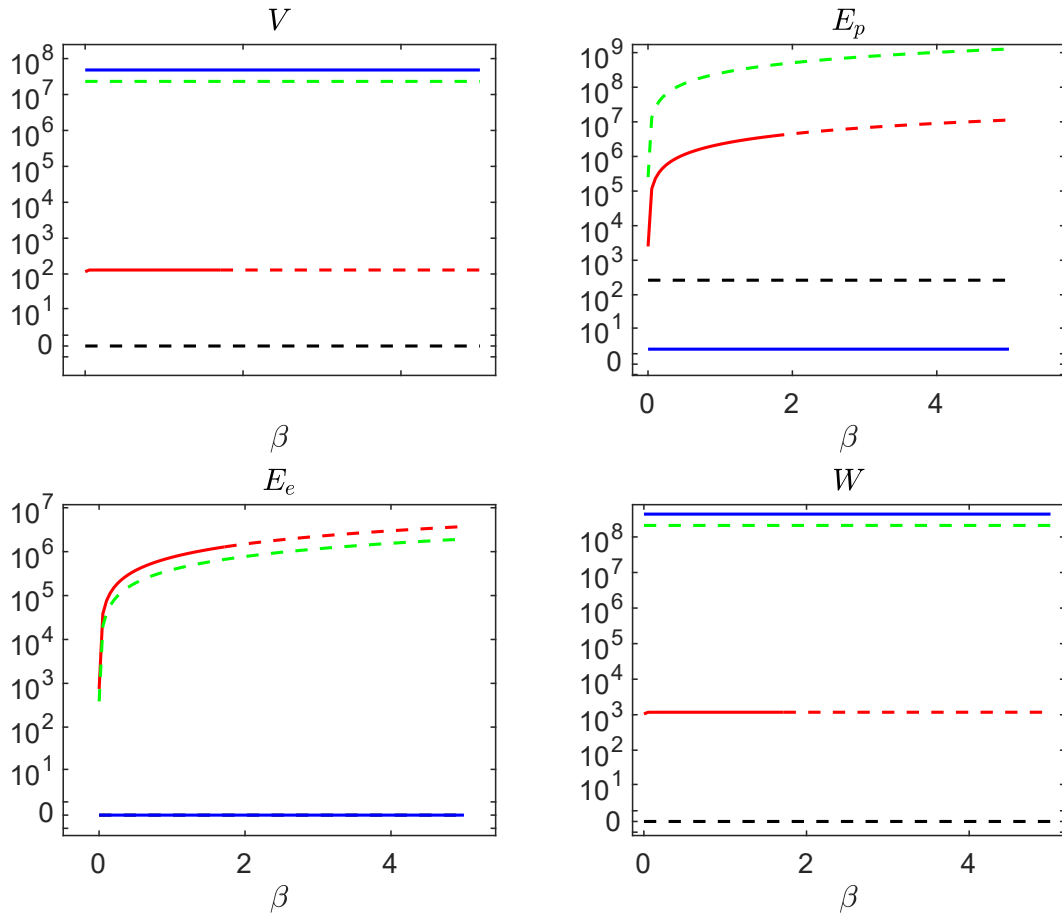


Рисунок 5.3 — Стационарные решения (I — черным, II — красным, III — зеленым, IV — синим) и их устойчивость (сплошная линия) и неустойчивость (пунктирная линия) как функции параметра  $\beta$ .

Поскольку бистабильность является важным свойством функционирования физиологических систем с нормальной регуляцией, важно исследовать возможность перевода системы, которая обладает свойством моностабильности в режим бистабильного функционирования. Одним из возможных подходов является изменение параметров регуляторных процессов. Разработанные методы позволяют осуществить поиск и реализацию подобных параметрических сдвигов. В качестве примера рассмотрим значения параметров из области моностабильности (рисунок 5.1). Путем последовательного уменьшения  $\beta$  (как показано на рисунке 5.4), система может быть переведена в область параметров с биста-



бильным режимом функционирования. На этом рисунке изображена переменная  $V$  в стационарных решениях с параметрами, которые использовались для построения рисунка 5.1, за исключением параметра  $\beta$ , который варьировался.

Таким образом, выявлено важное свойство данной нелинейной модели, которое состоит в возможности управления моно- и бистабильностью системы и в существовании монотонного режима перевода системы в параметрическую подобласть с двумя устойчивыми положениями равновесия. При этом возможность трассировки возникающего второго устойчивого положения равновесия позволяет перейти к задаче определения необходимого параметрического сдвига системы, который обеспечивал бы заданные целевые характеристики нового стационарного решения. В случае сценария, изображенного на рисунке 5.4, если речь идет о необходимости максимального снижения вирусной нагрузки, результаты выполненного численного анализа показывают, что для этого необходимо рассматривать вариант сдвига в пространстве параметров (с учетом возможных ограничений), изображенный на рисунке 5.4D.

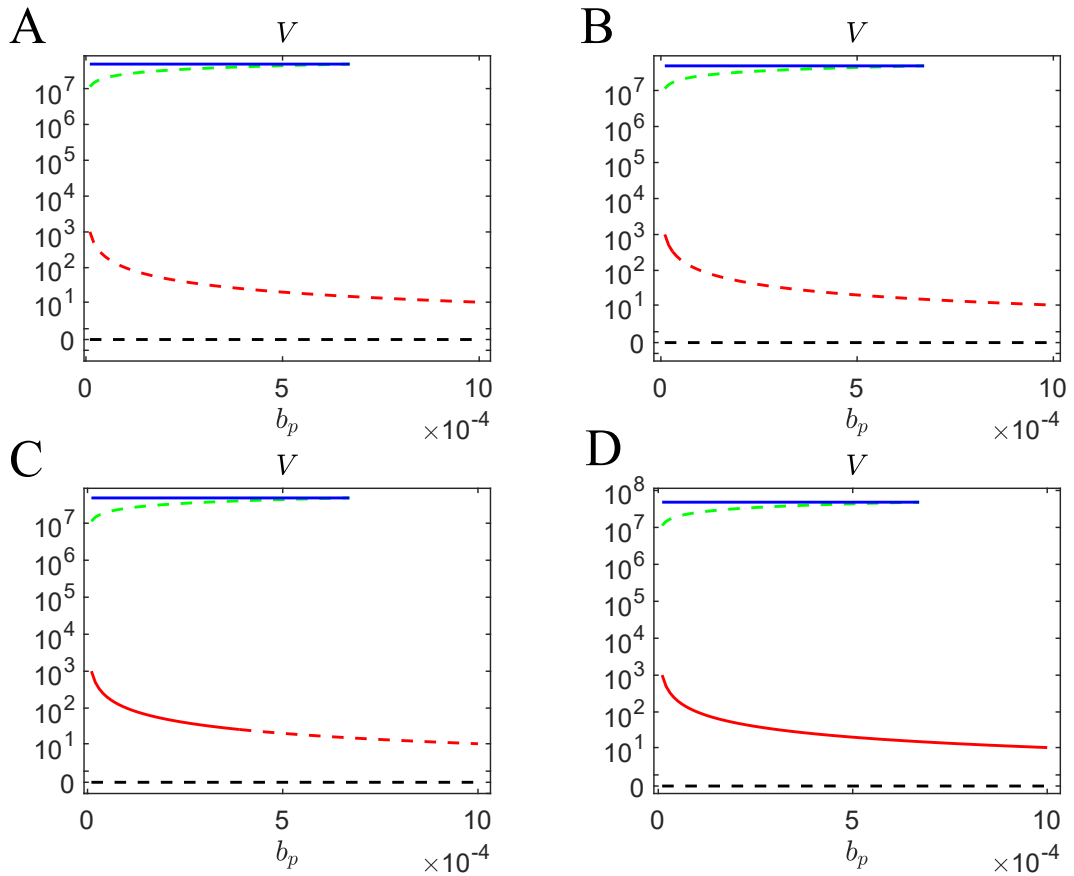


Рисунок 5.4 — Зависимость переменной  $V$  в стационарных решениях от параметра  $b_p$  при следующих значениях параметра  $\beta$ : 1.72 (A), 1.69 (B), 1.671 (C), 1.67 (D)

## 5.2 Оптимальные возмущения стационарных решений модели LCMV

Продemonстрируем результат вычисления оптимальных возмущений стационарных решений и результат воздействия этих возмущений на систему на примере модели LCMV (1.6) при значении параметров, приведенных в таблице 5 из работы [42], за исключением значений параметров  $\alpha_{E_e}$ ,  $\alpha_{E_p}$  и  $\beta$ , которые были выбраны равным 0.3, 0.01 и 0.05 соответственно. С помощью методов вычисления стационарных решений и анализа их устойчивости (разделы 2.1,2.2) было установлено, что у системы (1.6) при выбранных значениях параметров имеется устойчивое стационарное решение (I) с высокой вирусной нагрузкой

$$V \approx 4.82 \cdot 10^7, E_p \approx 1.71, E_e \approx 2.93 \cdot 10^{-4}, W \approx 4.38 \cdot 10^8,$$

и устойчивое стационарное решение (II) с низкой вирусной нагрузкой

$$V \approx 129, E_p \approx 1.13 \cdot 10^5, E_e \approx 3.73 \cdot 10^4, W \approx 1.17 \cdot 10^3.$$

Рисунки 5.5, 5.6 демонстрируют воздействие на решение (I) оптимального возмущения с  $\varepsilon = 1.4$ , рассчитанного в  $W_2^1$  норме и переводящего систему в решение (II) (в правой верхней части каждого рисунка показано начальное возмущение). В качестве диагональных элементов матрицы  $D$  были выбраны величины, обратные значениям соответствующих переменных в решении (I). Для каждой переменной использовалось 56 базисных функций, качественно аппроксимирующих поведение лекарств в одно- и двухкомпонентных фармакокинетических моделях.

Рисунки 5.7, 5.8 демонстрируют аналогичное воздействие оптимального возмущения с  $\varepsilon = 0.2$ , вычисленного в  $L_2$  норме. Видно, что начальное возмущение, в отличие от рассчитанного в  $W_2^1$  норме, значительно лишь в окрестности нулевого момента времени. Поэтому, хотя это возмущение и оказывает большее влияние на последующее поведение решения, его использование для разработки режимов терапии менее предпочтительно, так как "импульсное" воздействие может негативно сказаться на организме больного.

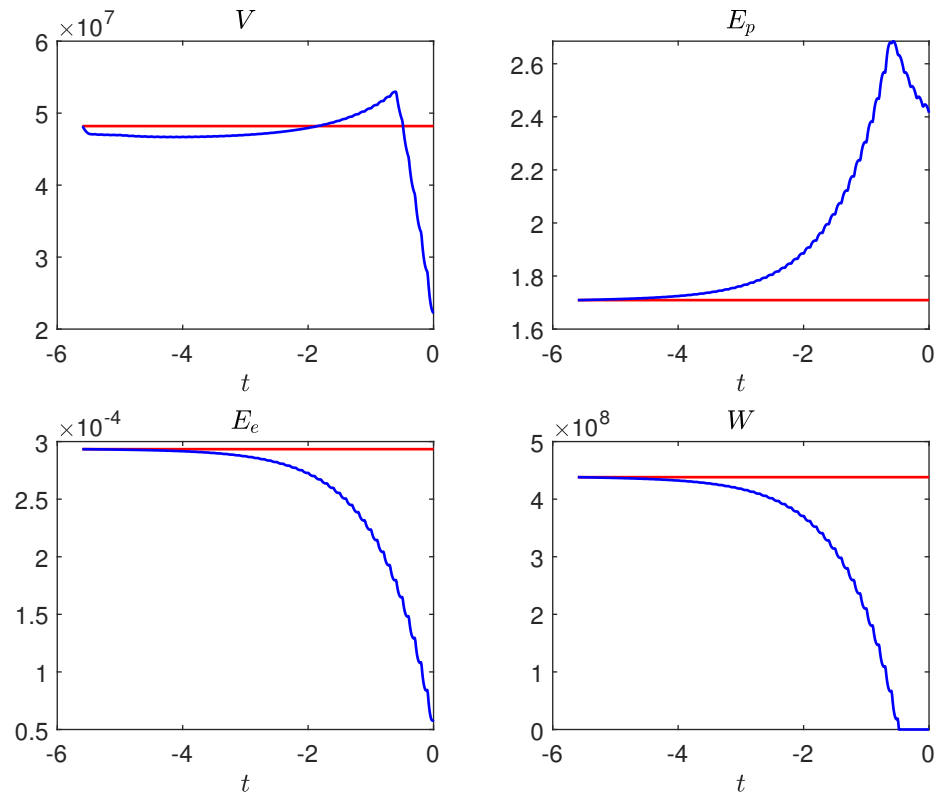


Рисунок 5.5 — Оптимальные возмущения, вычисленные в  $W_2^1$ -норме.

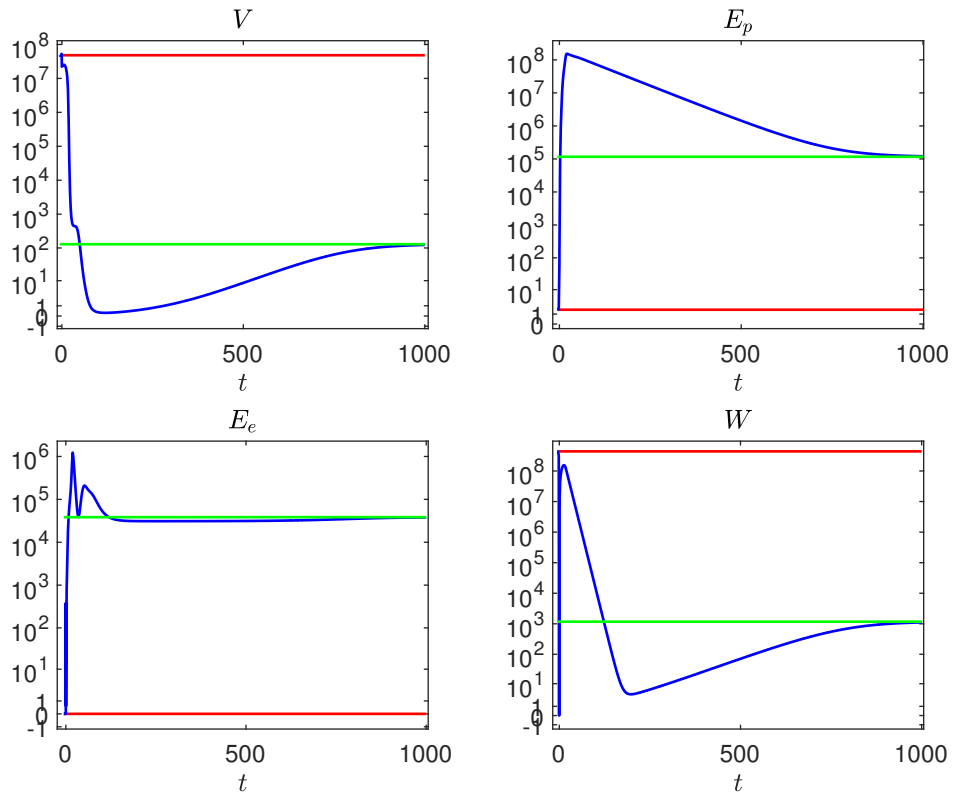


Рисунок 5.6 — Воздействие оптимального возмущения, найденного в  $W_2^1$  норме: переход из решения I (красная линия) в решение II (зеленая линия).

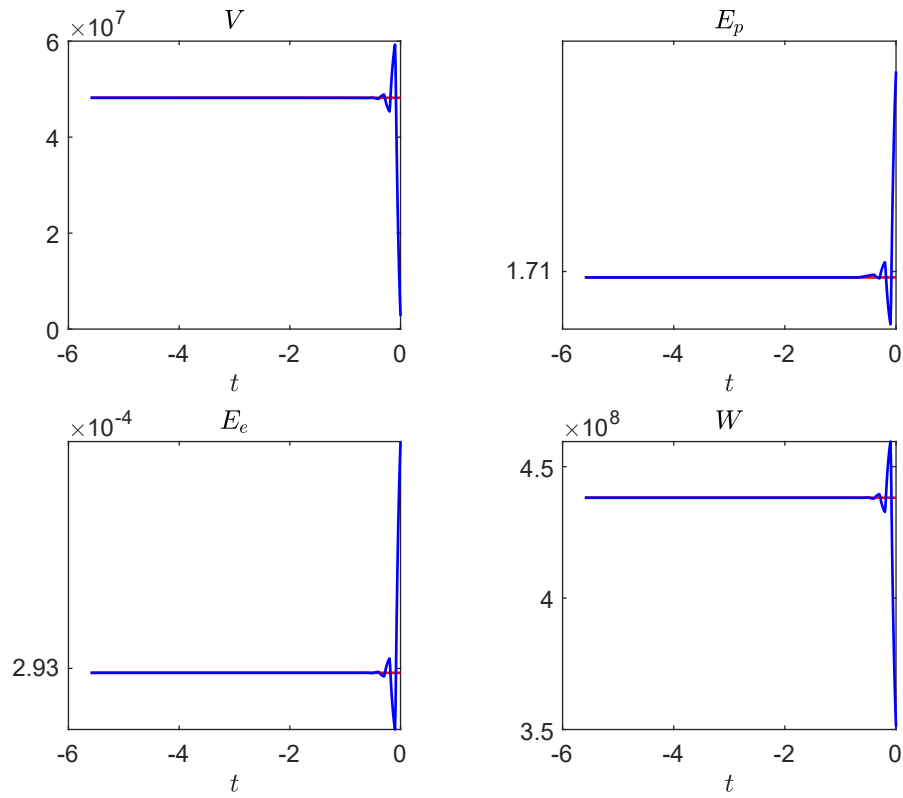


Рисунок 5.7 — Оптимальные возмущения, вычисленные в  $L_2$ -норме.

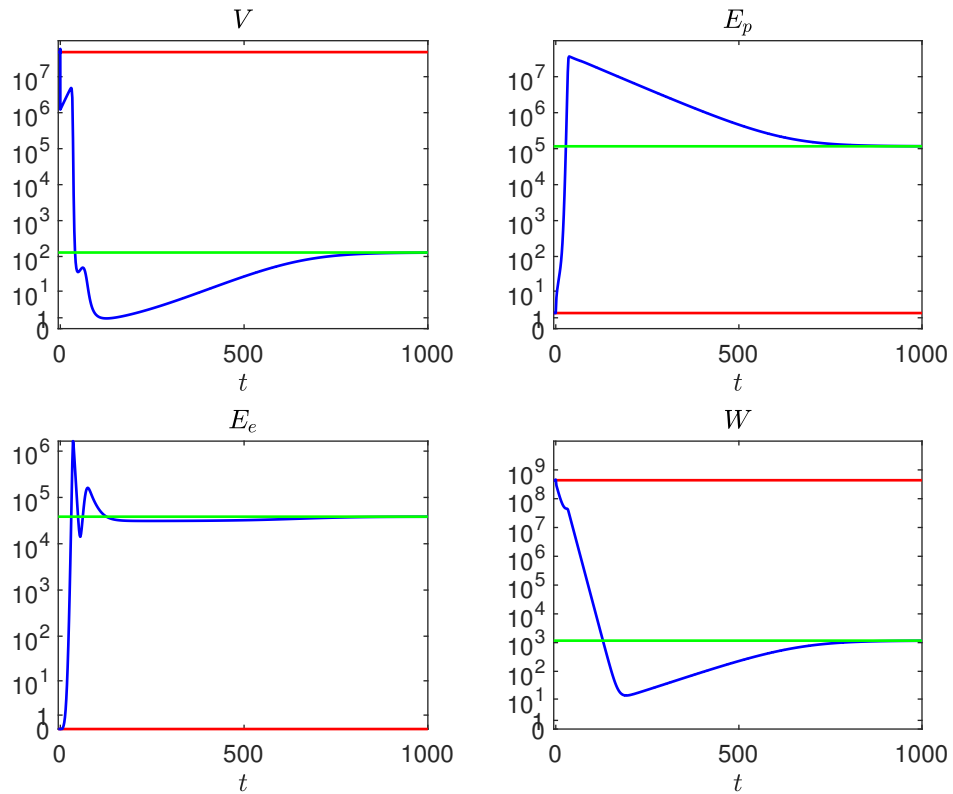


Рисунок 5.8 — Воздействие оптимального возмущения, найденного в  $L_2$  норме: переход из решения I (красная линия) в решение II (зеленая линия).

### 5.3 Анализ периодических решений модели LCMV

Продemonстрируем результаты работы алгоритмов вычисления периодических решений и исследования их устойчивости на примере модели LCMV (1.6) при значениях параметров, приведенных в таблице 5 из работы [42], за исключением значений параметров  $\beta$ ,  $\alpha_{E_p}$ ,  $\alpha_{E_e}$  которые были выбраны равными 2.05, 0.01, 0.3 соответственно.

#### 5.3.1 Вычисление периодического решения

С помощью методов вычисления стационарных решений и анализа их устойчивости было установлено, что у системы (1.6) при выбранных значениях параметров имеется в общей сложности четыре стационарных решения с вирусной нагрузкой  $V = 0$ ,  $V = 1.29 \cdot 10^2$ ,  $V = 2.32 \cdot 10^7$  и  $V = 4.82 \cdot 10^7$ , которые в дальнейшем будем называть решениями I, II, III и IV соответственно. Ведущие собственные значения (то есть с наибольшей действительной частью) для решений I, II, III, IV были равны соответственно 2.05,  $0.03 \pm 0.74i$ , 0.37,  $-0.11$ . Рассмотрим задачу поиска периодического решения в окрестности стационарного решения II. Округленные значения переменных модели в этом решении следующие:  $V = 1.29 \cdot 10^2$ ,  $E_p = 4.62 \cdot 10^6$ ,  $E_e = 1.53 \cdot 10^6$ ,  $W = 1.17 \cdot 10^3$ . Используемые значения параметров алгоритма приведены в таблице 25.

Таблица 25. Значения параметров алгоритма

Параметр	$\delta$	$\varepsilon$	$N_e$	$\theta_{u^1}$	$N_{f, \max}$	$tol_V$	$\rho$
Значение	$10^{-2}$	$5 \cdot 10^3$	30	$10^{-7}$	15	0.1	0.1
Параметр	$\varepsilon_T$	$\xi$	$N_{LS}$	$tol_{LS}$	$N_{NM}$	$tol_{NM}$	
Значение	0.1	$10^{-6}$	50	$10^{-6}$	5	$10^{-5}$	

В таблице 26 приведены значения дискретного аналога (3.8) функционала невязки (3.2), отношений минимального и максимального значений переменной  $V$  в узлах сетки к стационарному решению для начальной функции (3.3) и

начальное значение минимального периода, а также эти данные для каждого из трёх этапов метода вычисления периодического решения. Также в таблице 26 приведено время выполнения каждого из трех этапов в секундах в среде MATLAB на компьютере с 2-х ядерным процессором Intel Core i7 с частотой 2.4 GHz. На рисунке 5.9 изображено найденное периодическое решение в интервалах  $[-\tau_A, 0]$  (красным) и  $[0, T]$  (синим).

Отметим, что работа метода установления завершилась за  $N_f = 7$  шагов, поскольку абсолютная величина относительного изменения за один шаг вариации переменной  $V$  стала меньше чем  $\theta_V$ . Большая величина невязки в конце первого этапа связана с тем, что она вычисляется по найденному в начале этапа приближенному значению минимального периода, которое значительно отличается от установившегося приближенного значения минимального периода, который вычисляется на втором этапе. На третьем этапе для достижения значения нормы невязки, приведенной в последней строке таблицы, потребовалась одна итерация метода ньютоновского типа, описанного в разделе 3.1.3.

Таблица 26. Результаты этапов вычисления периодического решения

Этап	Невязка	$V_{min}/\bar{V}$	$V_{max}/\bar{V}$	$T$	Время (сек.)
0	$8.7 \cdot 10^{-4}$	$3.056 \cdot 10^{-3}$	6.513	8.48	—
1	0.92	$9.934 \cdot 10^{-4}$	7.603	8.48	7
2	$3.7 \cdot 10^{-4}$	$9.934 \cdot 10^{-4}$	7.603	13.09	1
3	$1.2 \cdot 10^{-5}$	$9.917 \cdot 10^{-4}$	7.605	13.09	12

### 5.3.2 Анализ асимптотической орбитальной устойчивости

При исследовании устойчивости полученного периодического решения использовалась равномерная сетка с шагом  $\delta = 10^{-2}$  для достижения нормы спектральной невязки равной  $1.5 \cdot 10^{-13}$  потребовалось три итерации метода Стюарта и 1 секунда. В результате были получены следующие 4 приближенных собственных значения матрицы монодромии: 1.0050, 0.87574, 0.60030, 0.24002. То есть, одно собственное значение матрицы монодромии было близко к 1, а

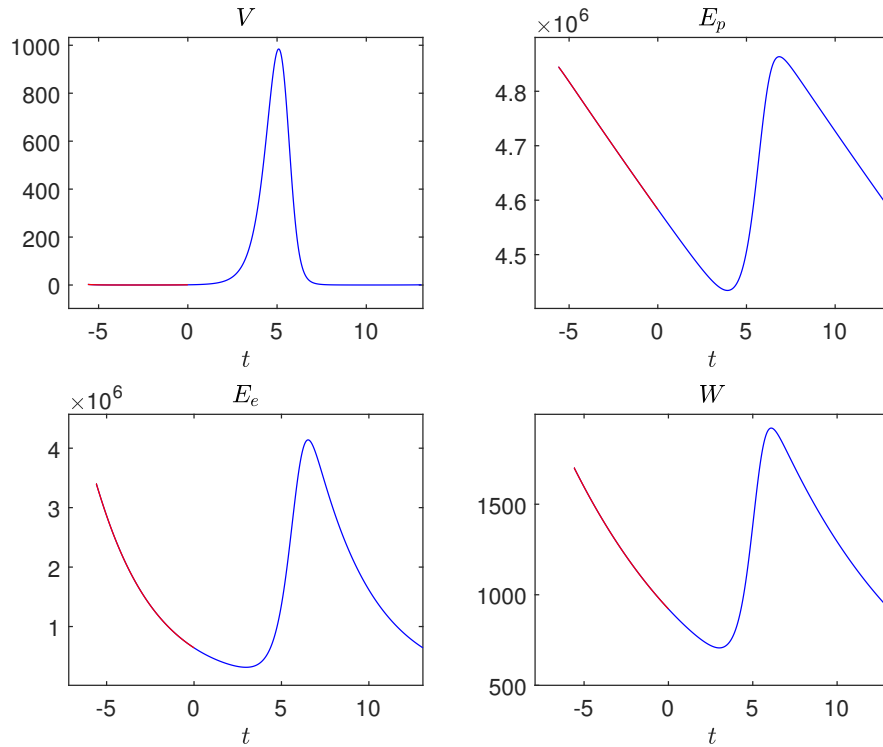


Рисунок 5.9 — Найденное периодическое решение в интервале  $[-\tau_A, T]$ .

все остальные лежали внутри единичного круга. Это означает, что полученное периодическое решение является асимптотически орбитально устойчивым.

Отметим, что при использовании сетки с шагом  $\delta = 10^{-3}$  приближенные собственные значения матрицы монодромии отличаются от значений, полученных при использовании сетки с шагом  $\delta = 10^{-2}$ , в четвертой значащей цифре. Следовательно, выбор шага сетки  $\delta = 10^{-2}$  является достаточным для исследования асимптотической орбитальной устойчивости.

#### 5.4 Оптимальные возмущения периодических решений модели LCMV

Продemonстрируем результаты вычисления оптимального возмущения периодического решения и результат его воздействия на систему на примере модели LCMV (1.6) при значениях параметров, приведенных в таблице 5 из работы [42], за исключением значений параметров  $\beta$ ,  $\alpha_{E_p}$ ,  $\alpha_{E_e}$ , которые выберем равными 2.9, 0.01, 0.3 соответственно. С помощью методов вычисления стац-

онарных решений и анализа их устойчивости (разделы 2.1, 2.2) было установлено, что у системы (1.6) при выбранных значениях параметров имеется в общей сложности четыре стационарных решения с вирусными нагрузками  $V = 0$ ,  $V = 1.29 \cdot 10^2$ ,  $V = 2.32 \cdot 10^7$  и  $V = 4.82 \cdot 10^7$ , причем ведущие собственные значения (то есть с наибольшей действительной частью) системы, линеаризованной относительно этих решений, равны соответственно 2.90,  $0.09 \pm 0.86i$ , 0.40 и  $-0.11$ . Если стационарное решение неустойчиво и ведущее собственное значение линеаризованной относительно него системы имеет ненулевую мнимую часть, то в окрестности такого стационарного решения возможно устойчивое периодическое решение. Этому условию удовлетворяет только второе из перечисленных выше стационарных решений. С помощью метода вычисления периодического решения (раздел 3.1) в окрестности этого стационарного решения было найдено устойчивое периодическое решение  $\phi(t)$ , изображенное на рисунке 5.10 черной линией. Наименьший период этого решения  $T = 23.14$ .

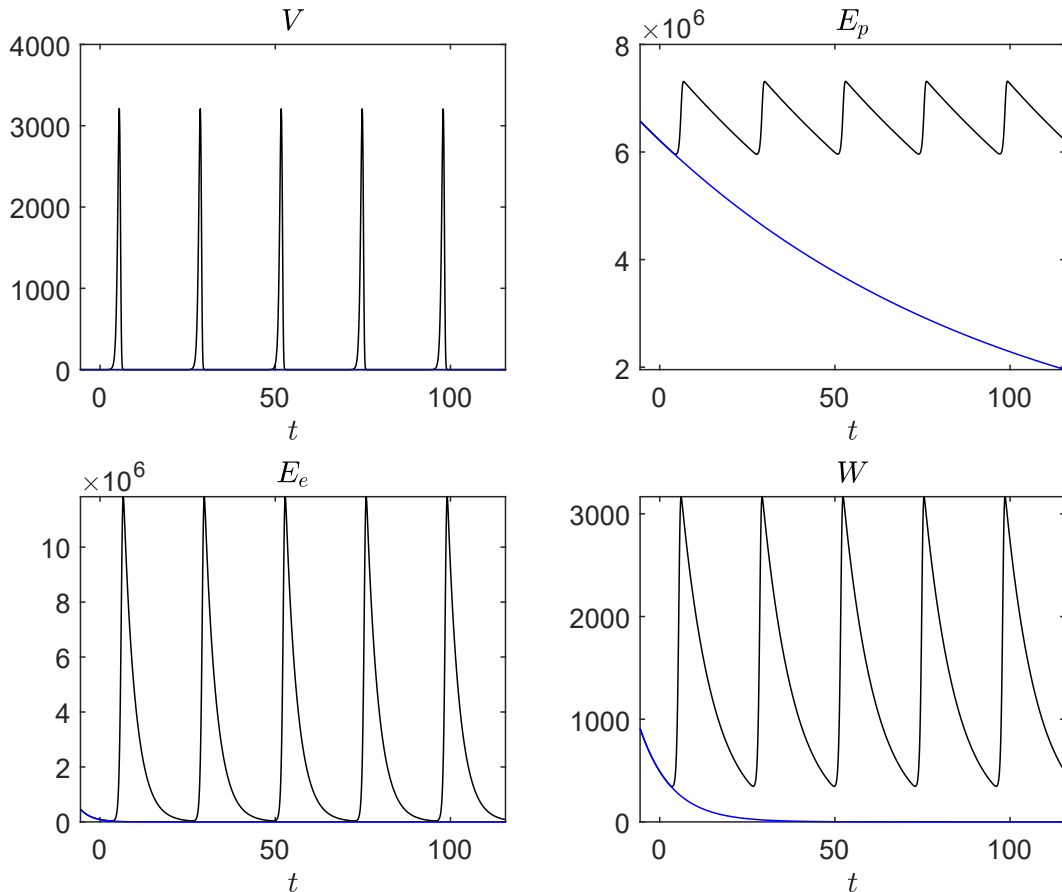


Рисунок 5.10 — Периодическое решение  $\phi(t)$  в интервале  $[-\tau_A, 5T]$  (черная линия) и оно же, возмущенное возмущением, оптимальным в момент времени  $T$ , взятым с весом  $\varepsilon = -9 \cdot 10^{-6}$  (синяя линия).



Рассмотрим возмущение, оптимальное в момент времени  $T$ . Для его вычисления веса локальной нормы (диагональные элементы матрицы  $D$ ) брались равными величинам, обратным компонентам стационарного решения с вирусной нагрузкой  $V = 1.29 \cdot 10^2$ , в окрестности которого расположено рассматриваемое периодическое решение, а параметр  $\rho$  выбирался равным 1. Последнее обеспечивало плавную зависимость начального значения оптимального возмущения от времени. В случае выбора  $\rho = 0$  оптимальное возмущение имеет импульсный характер с существенно большей амплификацией и его использование для разработки режимов терапии менее предпочтительно, так как такое импульсное воздействие может негативно сказаться на организме больного.

Шаг сетки  $\delta$  выбирался равным 0.01, при этом  $m_2 = 560$ . В качестве базисных функций для каждой компоненты оптимального возмущения использовались функции, качественно аппроксимирующие поведение препаратов в рамках однокамерных и двухкамерных фармакокинетических моделей (2.36). Пространство  $\mathcal{Q}$ , из которого искалось оптимальное возмущение, имело размерность  $4 \cdot 56 = 224$ .

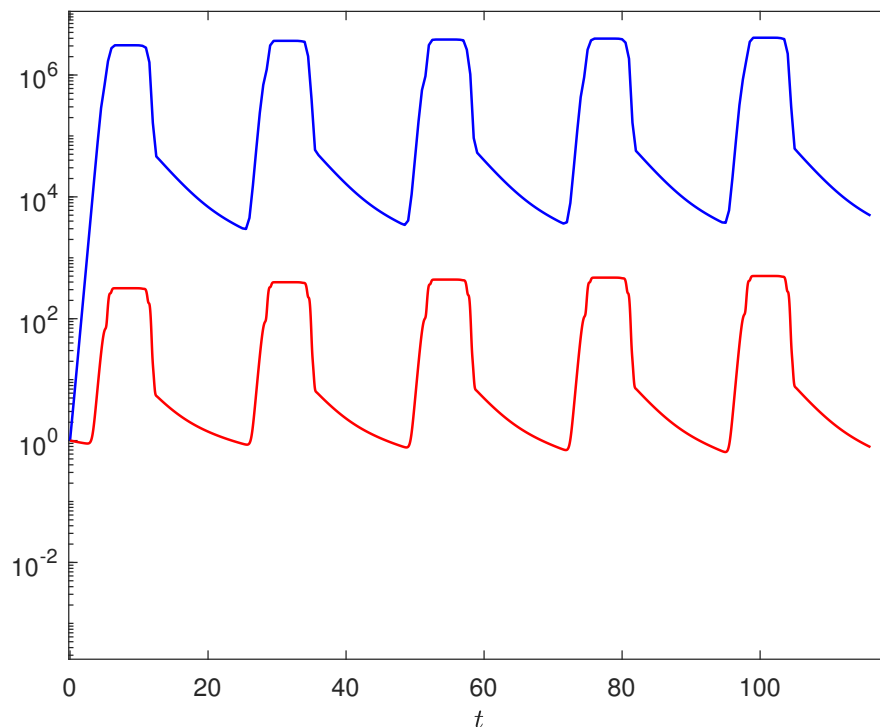


Рисунок 5.11 — Амплификации локальной нормы возмущения, оптимального в момент  $T$ , (синяя линия) и собственной моды, соответствующей собственной функции  $g(t)$  (красная линия).

На рисунке 5.11 изображена зависимость от времени локальной нормы вычисленного оптимального возмущения и собственной моды системы (3.9), соответствующей собственной функции  $g(t)$ , отвечающей собственному значению 1 оператора монодромии. Начальные функции были нормированы в норме, использованной для вычисления оптимального возмущения. Видно, что при  $t \geq T$  эти зависимости близки по форме, но норма оптимального возмущения превосходит норму собственной моды примерно на 4 порядка.

Начальная функция вычисленного оптимального возмущения нормировалась в локальной норме с той же матрицей  $D$ , но параметром  $\rho = 0$ . Такая норма возмущения характеризует общее количество препаратов введенных в организм для формирования этого возмущения и, следовательно, ее можно интерпретировать как меру побочных эффектов, нанесенного организму таким воздействием. Нормированная таким образом начальная функция оптимального возмущения изображена на рисунке 5.12 слева. Справа для сравнения изображена нормированная точно таким же образом собственная функция  $g(t)$  оператора монодромии.

Во второй колонке таблицы 27 приведены значения максимальной амплификации  $\Gamma(t)$  локальной нормы с  $\rho = 1$  возмущения в кратные  $T$  моменты времени от 0 до  $5T$ . Видно, что максимальная амплификация сначала очень быстро возрастает, а затем начинает от периода к периоду возрастать существенно медленнее. Это связано со сходимостью при  $t \rightarrow \infty$  любого возмущения либо к нулю, либо к собственной моде, соответствующей собственной функции  $g(t)$ . Для оптимального возмущения имеет место последнее. Это демонстрируют приведенные в третьей и четвертой колонках таблицы 27 норма спектральной проекции функции

$$v_t(\tau) = \tilde{u}_{\text{opt},T}(t + \tau), \quad -\tau_2 \leq \tau \leq 0, \quad (5.1)$$

на инвариантное подпространство оператора  $\mathcal{M}$ , отвечающее его собственному значению 1, и норма остатка, деленные на норму исходной функции  $v_t$ . При этом были использованы обычная  $L_2$  норма и соответствующее скалярное произведение.

Для вычисления собственного вектора сеточного аналога оператора монодромии, отвечающего собственному значению 1, применялся метод Стюарта. Для вычисления спектральной проекции использовался собственный век-

тор транспонированной матрицы  $\Pi_K^T$ , отвечающий ее собственному значению 1, который так же вычислялся этим методом.

Таблица 27. Максимальная амплификация  $\Gamma(t)$ , относительная норма спектральной проекции возмущения, оптимального в момент времени  $T$ , и относительная норма остатка в различные моменты времени

$t$	$\Gamma(t)$	$\ Pv_t\ /\ v_t\ $	$\ (I - P)v_t\ /\ v_t\ $
0	1	224.02	224.37
$T$	$4.03 \cdot 10^3$	1.20	0.48
$2T$	$4.48 \cdot 10^3$	1.17	0.37
$3T$	$4.78 \cdot 10^3$	1.14	0.28
$4T$	$5.03 \cdot 10^3$	1.11	0.22
$5T$	$5.29 \cdot 10^3$	1.09	0.17

Для демонстрации эффективности воздействия оптимальным возмущением на периодическое решение было проведено сравнение результатов возмущения периодического решения этим возмущением и собственной функцией  $g(t)$ . Оказалось, что для того, чтобы перевести систему из периодического решения в окрестность стационарного решения с вирусной нагрузкой  $V = 0$  необходимо взять изображенное на рисунке 5.12 слева оптимальное возмущение с весом  $-9 \cdot 10^{-6}$ . Результат воздействия оптимальным возмущением показан на рисунке 5.10. Для получения аналогичного результата, используя собственную функцию, изображенную на рисунке 5.12 справа, необходимо взять ее с весом  $-7 \cdot 10^{-2}$ . Результат такого воздействия будет визуально неотличим от результата воздействия оптимальным возмущением, изображенным на рисунке 5.10. Таким образом, оптимальное возмущение позволяет достичь того же эффекта но с гораздо меньшим начальным воздействием на систему, чем собственная мода, отвечающая максимальному собственному значению оператора монодромии.

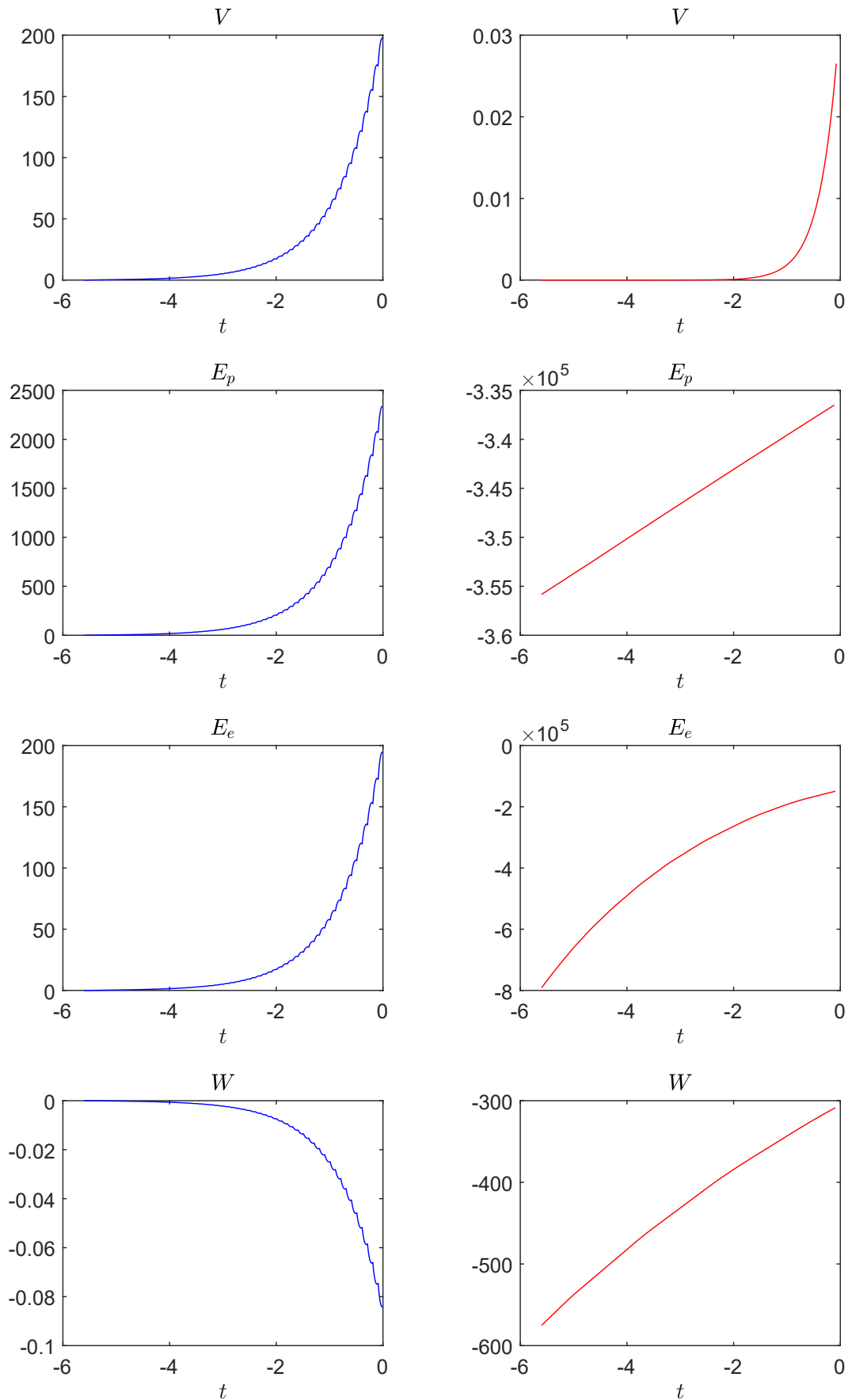


Рисунок 5.12 — Одинаковые по норме начальные значение возмущения, оптимального в момент времени  $T$ , (слева синяя линия) и собственная функция  $g(t)$  (справа красная линия).

## 5.5 Зависимость оптимального возмущения от фазы

С помощью методов вычисления стационарных решений, предложенных в [27] было установлено, что при значениях параметров модели приведенных в таблице 5 из работы [42], за исключением значений параметров  $\beta$ ,  $\alpha_{E_p}$ ,  $\alpha_{E_e}$  которые были выбраны равными 1.9, 0.01, 0.3 соответственно, у рассматриваемой модели есть четыре стационарных решения с вирусной нагрузкой  $V = 0$ ,  $1.29 \cdot 10^2$ ,  $2.32 \cdot 10^7$  и  $4.82 \cdot 10^7$ , причем ведущие (то есть с наибольшей действительной частью) собственные значения системы, линеаризованной относительно этих решений, равны соответственно 1.9,  $0.02 \pm 0.72i$ , 0.37,  $-0.11$ . С помощью метода вычисления периодического решения (раздел 3.1) в окрестности второго стационарного решения было найдено устойчивое периодическое решение  $\phi(t)$ , максимальное и минимальное значение переменных в котором приведены в таблице 28. Наименьший период этого решения  $T = 11.44$ .

Таблица 28. Максимальное и минимальное значения переменных в устойчивом периодическом решении  $\phi(t)$ .

	min	max
$V$	1.99	$6.82 \cdot 10^2$
$E_p$	$4.15 \cdot 10^6$	$4.45 \cdot 10^6$
$E_e$	$4.56 \cdot 10^5$	$3.12 \cdot 10^6$
$W$	$8.07 \cdot 10^2$	$1.71 \cdot 10^3$

Так как решение  $\phi(t + c)$ , где  $c$  — произвольная вещественная константа, является таким же периодическим решением, что и  $\phi(t)$  только сдвинутым по фазе, то при вычислении оптимального возмущения периодического решения принципиальным является вопрос выбора этой фазы. Будем далее обозначать через  $c_{\min}$  и  $c_{\max}$  фазы, такие, что  $V(c_{\min})$  и  $V(c_{\max})$  — соответственно минимальное и максимальное значения  $V(t)$  в периодическом решении  $\phi(t)$ . Для исследования зависимости периодического решения от фазы этого решения вычислим для периодических решений  $\phi_{\min}(t) = \phi(t + c_{\min})$  и  $\phi_{\max}(t) = \phi(t + c_{\max})$  возмущения, оптимальные в момент времени, равный наименьшему периоду  $T$ . Обозначим эти возмущения  $w_{\min}(t)$  и  $w_{\max}(t)$ , соответственно. Веса локальной нормы

(диагональные элементы матрицы  $D$ ) будем брать равными величинам, обратным компонентам стационарного решения с вирусной нагрузкой  $V = 1.29 \cdot 10^2$ , в окрестности которого расположено рассматриваемое периодическое решение, а параметр  $\rho$  возьмем равным 1.

Шаг сетки  $\delta$  и пространство  $\mathcal{Q}$ , из которого мы будем искать оптимальные возмущения выбирались такими же, как и в разделе 5.4.

На Рис. 5.13 изображены начальные значения вычисленных оптимальных возмущений. Видно, что  $w_{\max}(t)$  имеет меньшую вирусную нагрузку  $V$ , чем  $w_{\min}(t)$ , и большие значения переменных иммунного ответа  $E_p$  и  $E_e$ .

На Рис. 5.14—5.15 изображены периодическое решение  $\phi_{\min}(t)$  и оно же, возмущенное возмущением  $w_{\min}(t)$  с положительным и отрицательным весами соответственно. Из этих рисунков видно, что при отрицательном весе, то есть когда возмущение уменьшает вирусную нагрузку в начальный момент, в итоге достигается более значительное уменьшение вирусной нагрузки, причем на более длительное время, чем при положительном весе, хотя норма полученного возмущающего воздействия при отрицательном весе меньше на три порядка. На Рис. 5.16—5.17 изображены периодическое решение  $\phi_{\max}(t)$  и оно же, возмущенное возмущением  $w_{\max}(t)$  с положительным и отрицательным весами соответственно. Из этих рисунков видно, что при отрицательном весе, то есть когда возмущение уменьшает вирусную нагрузку в начальный момент, в итоге происходит подскок вирусной нагрузки на два порядка, а при выборе возмущения с положительным весом итоговый подскок вирусной нагрузки получается меньшим и значение вирусной нагрузки оказывается в окрестности нуля дольше чем в случае невозмущенного периодического решения. Сравнивая Рис. 5.14 и 5.16 можно заключить, что оптимальные возмущения с положительными весами решений  $\phi_{\min}(t)$  и  $\phi_{\max}(t)$ , что естественно интерпретировать как лечение обострением, могут приводить к похожим динамикам, однако для снижения вирусной нагрузки на более длительное время в случае решения  $\phi_{\max}(t)$  возмущение достаточно взять меньшим по норме. Аналогично из Рис. 5.15 и 5.17 следует, что возмущение с отрицательным весом, является наиболее эффективным в случае решения  $\phi_{\min}(t)$ . Наиболее же эффективным из всех рассмотренных случаев является возмущение с отрицательным весом для решения  $\phi_{\min}(t)$ .

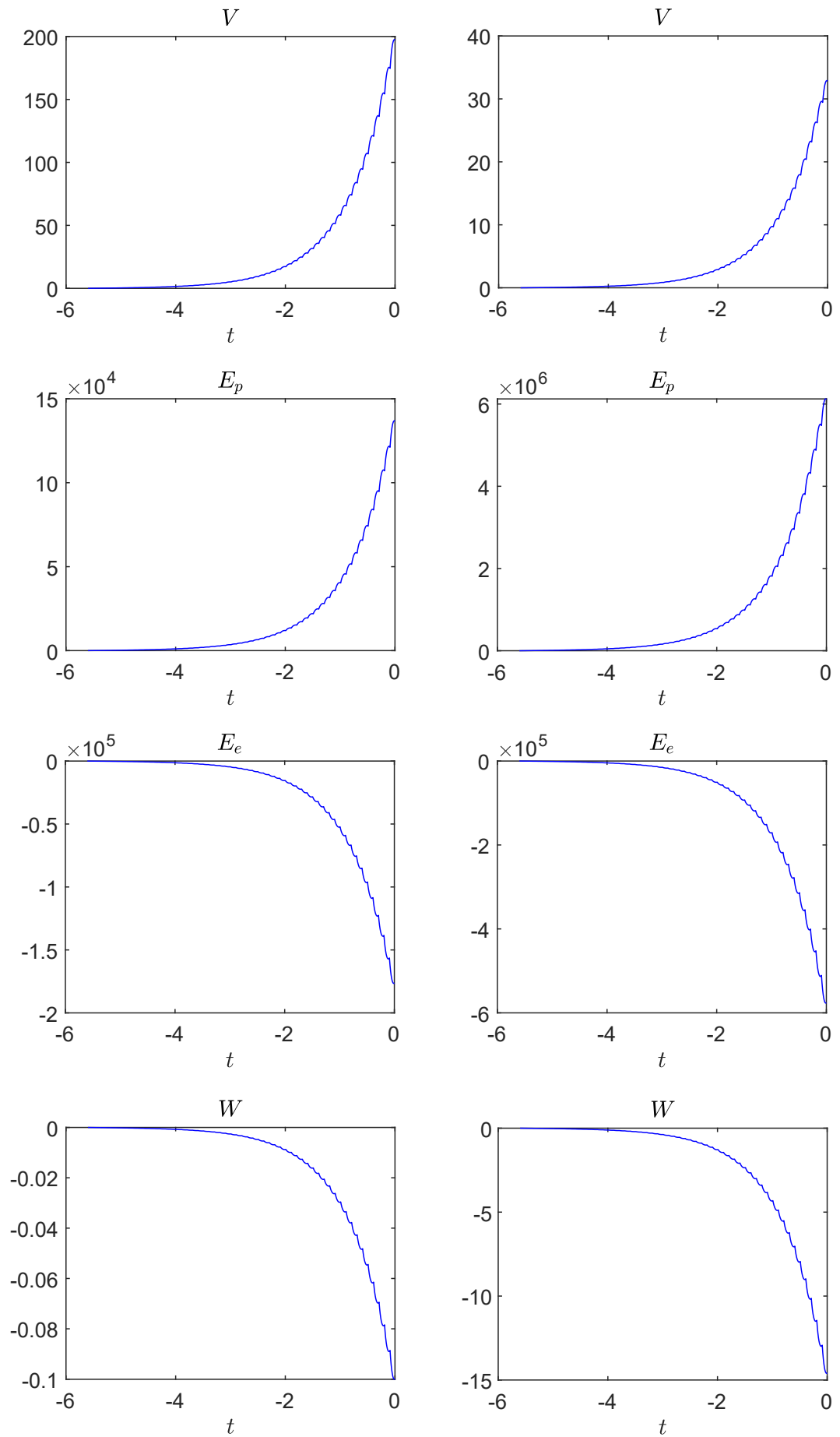


Рисунок 5.13 — Начальные значения оптимальных возмущений  $w_{\min}(t)$  и  $w_{\max}(t)$ .

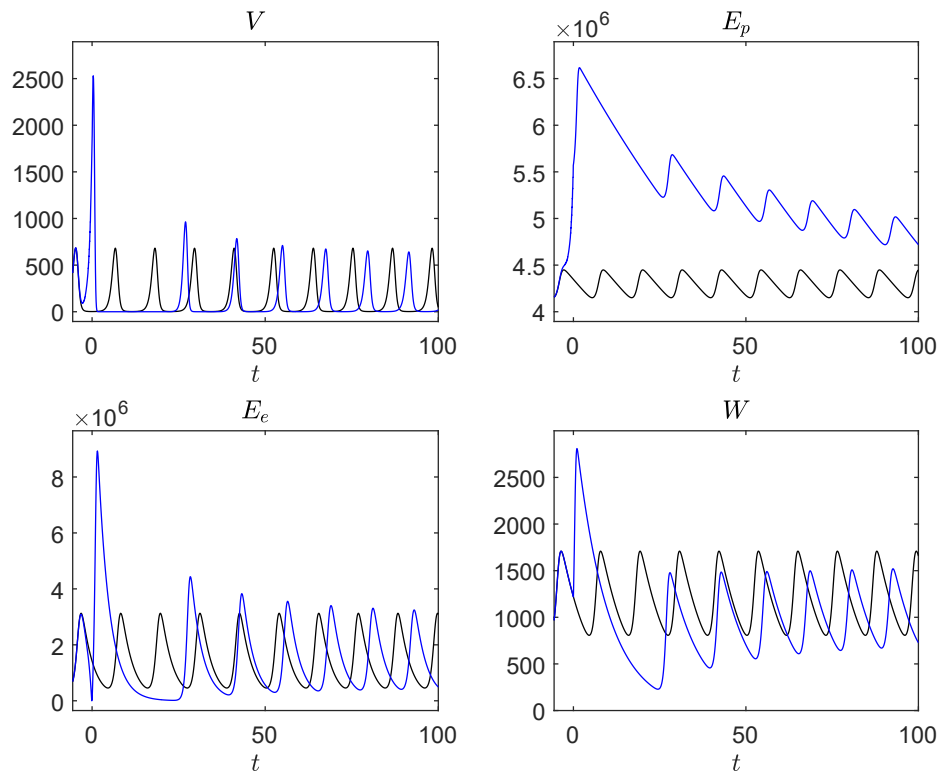


Рисунок 5.14 — Периодическое решение  $\phi_{\min}(t)$  в интервале  $[-\tau_2, 100]$  (черная линия) и оно же, возмущенное возмущением  $w_{\min}(t)$ , взятым с весом  $\varepsilon = 10$  (синяя линия).

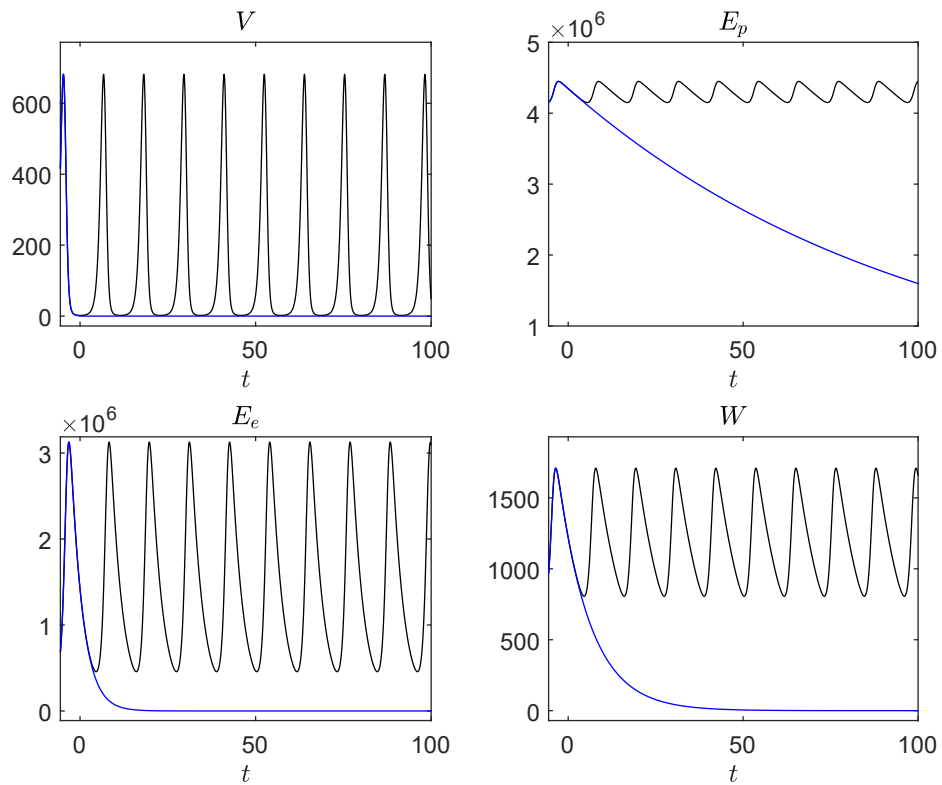


Рисунок 5.15 — Периодическое решение  $\phi_{\min}(t)$  в интервале  $[-\tau_2, 100]$  (черная линия) и оно же, возмущенное возмущением  $w_{\min}(t)$ , взятым с весом  $\varepsilon = -1.02 \cdot 10^{-2}$  (синяя линия).



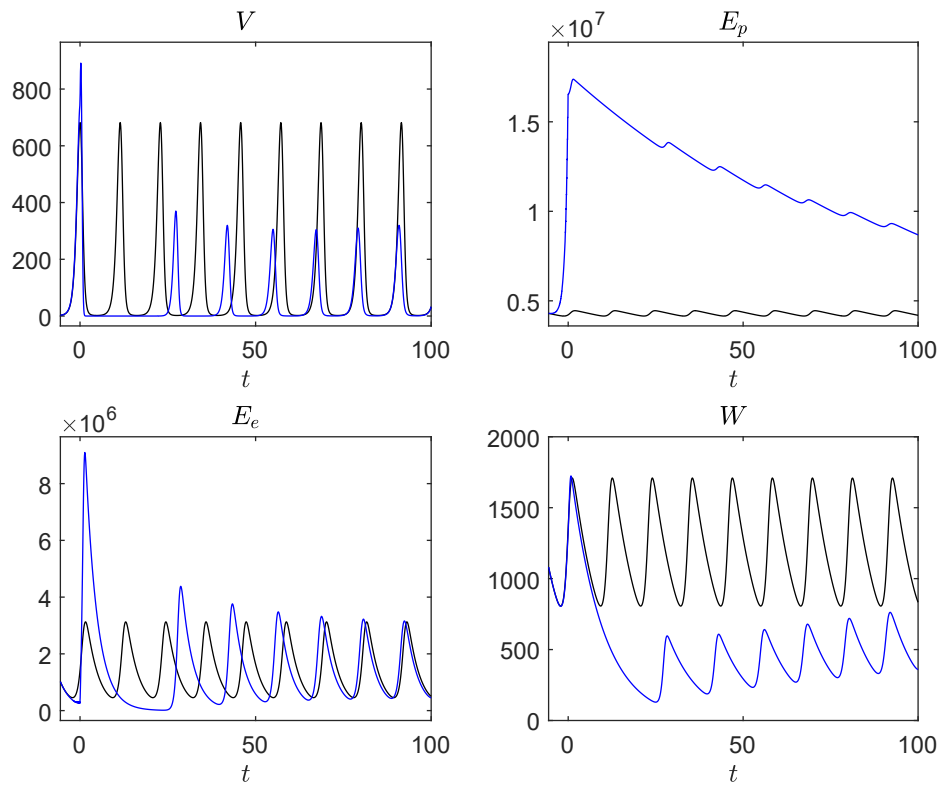


Рисунок 5.16 — Периодическое решение  $\phi_{\max}(t)$  в интервале  $[-\tau_2, 100]$  (черная линия) и оно же, возмущенное возмущением  $w_{\max}(t)$ , взятым с весом  $\varepsilon = 2$  (синяя линия).

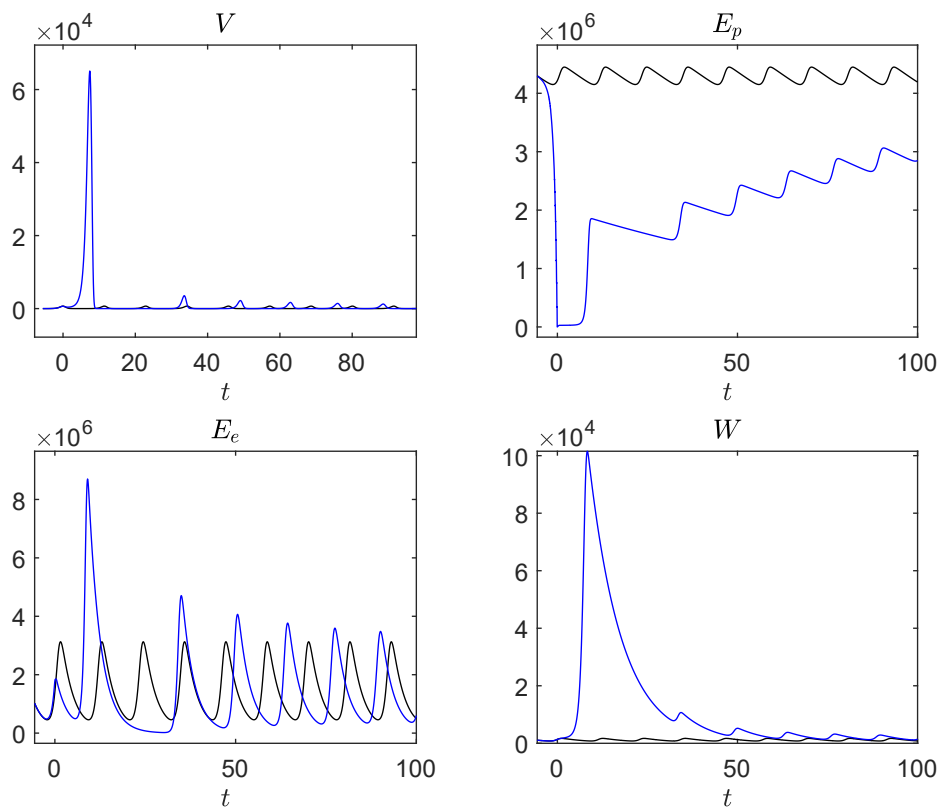


Рисунок 5.17 — Периодическое решение  $\phi_{\max}(t)$  в интервале  $[-\tau_2, 100]$  (черная линия) и оно же, возмущенное возмущением  $w_{\max}(t)$ , взятым с весом  $\varepsilon = -0.7$  (синяя линия).

## 5.6 Анализ стационарных решений модели HBV

С помощью предложенной технологии была исследована зависимость стационарных решений модели Марчука-Петрова от константы скорости антигенной стимуляции макрофагов в лимфоузле  $\gamma_{MV}$ . Этот параметр варьировался в интервале  $[1.7 \cdot 10^{-13}, 1.7 \cdot 10^{-11}]$ . Вычисление стационарных решений и анализ их зависимости от параметра  $\gamma_{MV}$  выполнялся при трех различных наборах параметров, которые мы в дальнейшем будем называть наборами (а), (б) и (с). Набор (а) соответствует острому течению гепатита В. Значения параметров из этого набора приведены в таблице 4. Набор (б) состоит из тех же значений параметров, что и набор (а), за исключением параметра  $\nu$ , который был уменьшен в 3 раза. Набор (с) состоит из тех же значений параметров, что и набор (б), за исключением параметров  $\rho_F$ ,  $b_m$ ,  $b_{CE}$ ,  $b_p^E$ , которые были выбраны равными  $8.5 \cdot 10^6$ ,  $5.2 \cdot 10^{-2}$ ,  $5.2 \cdot 10^{-5}$ ,  $4.1 \cdot 10^{-8}$  соответственно. Биологически, набор значений параметров (б) соответствует вирусной инфекции с малой скоростью размножения вирусов, что является одной из причин хронизации инфекции [1]. Набор значений параметров (с) отвечает ситуации хронической инфекции с более эффективной стимуляцией цитотоксического Т-клеточного иммунного ответа.

В дополнение к варьированию значений параметров модели, мы варьировали чувствительность иммунного ответа к поражению органа мишени, которая описывается с помощью функции  $\xi(m)$ . Так, в экспериментах с набором (с) вместо функции  $\xi(m) = \max\{1 - m/C^0, 0\}$  использовались функции  $\xi(m) = \max\{(1 - m/C^0)^5, 0\}$  и  $\xi(m) = \max\{(1 - m/(0.2C^0)), 0\}$ . На рисунках 5.19—5.20 приведены зависимости стационарных решений модели Марчука-Петрова от параметра  $\gamma_{MV}$  для наборов (а) и (б). На рисунках 5.21 приведена зависимость стационарных решений модели Марчука-Петрова от параметра  $\gamma_{MV}$  в интервале  $[1.5 \cdot 10^{-11}, 1.7 \cdot 10^{-11}]$  для набора (с) при  $\xi(m) = \max\{(1 - m/C^0)^5, 0\}$ . На рисунках 5.22 приведена зависимость стационарных решений модели Марчука-Петрова от параметра  $\gamma_{MV}$  в интервале  $[1.6 \cdot 10^{-11}, 2.5 \cdot 10^{-11}]$  для набора (с) при  $\xi(m) = \max\{(1 - m/(0.2C^0)), 0\}$ .

Выбор альтернативных параметризаций функции  $\xi(m)$  обусловлен следующими соображениями. Поражение клеток печени вследствие цитопатиче-

ских эффектов вируса, а также вследствие их уничтожения цитотоксическими лимфоцитами, вызывает супрессию антиген-специфического иммунного ответа. Данная обратная связь параметризована с помощью функции  $\xi(m)$ , где  $m$  — количество пораженных клеток печени,  $m \in [0, C^0]$ . В исходной модели параметризация имеет вид  $\xi(m) = 1 - m/C^0$ . Функция  $\xi(m)$  принимает значения от 0 (полная супрессия иммунного ответа при полном разрушении печени) до 1 (полноценный иммунный ответ при отсутствии повреждений печени). В данной работе предложены две альтернативные параметризации этой функции:

$$\xi(m) \equiv \xi_k(m) = \left(1 - \frac{m}{C^0}\right)^k, \quad (5.2)$$

а также

$$\xi(m) \equiv \xi_r(m) = \begin{cases} 1 - \frac{m}{rC^0}, & m \leq rC^0, \\ 0, & m > rC^0, \end{cases} \quad (5.3)$$

которые эквивалентны исходной функции при параметрах  $k = 1$  и  $r = 1$ . При увеличении  $k > 1$  или уменьшении  $r < 1$ , функция  $\xi(m)$  быстрее убывает от единицы до нуля (см. рисунок 5.18), что соответствует более сильной чувствительности супрессии иммунного ответа к степени поражения печени. Альтернативные параметризации функции  $\xi(m)$  позволили выявить различные бистабильные режимы со свойством гистерезиса (рисунок 5.21 и 5.22).

Трассировка стационарных решений по параметру  $\gamma_{MV}$ , представленная на рисунке 5.19, показывает, что для возникновения условия хронической инфекции, т.е. устойчивого сосуществования популяции вирусов и иммунного ответа, требуется существенное снижение эффективности антигенпрезентирования до уровня  $\gamma_{MV} \sim 3 \cdot 10^{-11}$ . При этом, уменьшение скорости размножения вирусов  $\nu$  в 3 раза существенно расширяет интервал значений параметра  $\gamma_{MV}$ , в котором возможно формирование стационарных решений, соответствующих хроническому течению вирусного гепатита В с высокой вирусной нагрузкой (рисунок 5.20).

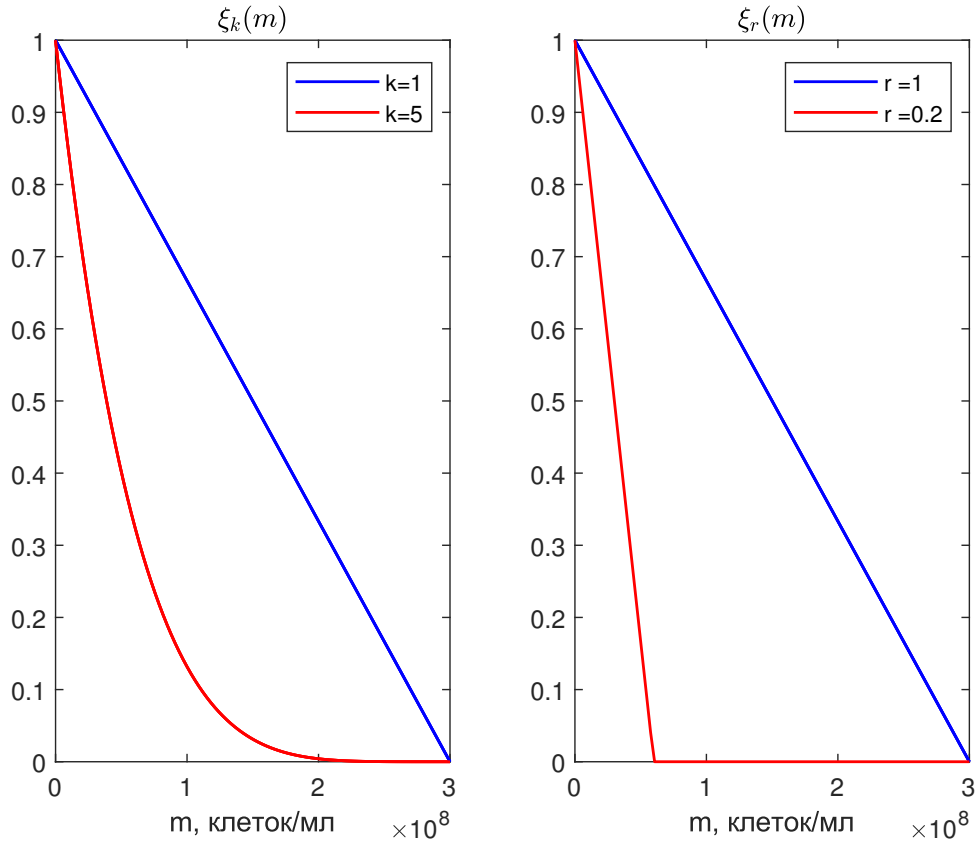


Рисунок 5.18 — Параметризация эффекта супрессии антиген-специфического иммунного ответа при повреждении печени с помощью функций  $\xi_k(m)$  и  $\xi_r(m)$  (уравнения (5.2),(5.3)).

Стоит отметить, что результат, проиллюстрированный на рисунке 5.20, соответствует полученным ранее теоретическим результатам. В работе [43] было приведено следующее условие асимптотической устойчивости тривиального стационарного решения с  $V_f = 0$ :

$$G = \left( \gamma_{VF} \frac{\rho_F P^0}{\alpha_F} + \gamma_{VM} M^0 + \gamma_{VC} C^0 \right) (b_{CE} E^0 + b_m) - \sigma C^0 (\nu + n b_{CE} E^0) > 0.$$

Нетрудно проверить, что это условие выполняется для набора параметров (b) и не выполняется для набора (a), что соответствует результатам, проиллюстрированным на рисунках 5.19–5.20.

Для модели Марчука-Петрова наличие бистабильности возможно в определённых областях параметров модели, поиск которых был осуществлен с использованием разработанных нами методов. На рисунке 5.21, соответствующим набору параметров (c), показано сосуществование трех нетривиальных положений равновесия.

ний равновесия, два из которых являются одновременно устойчивыми для некоторого интервала значений параметра  $\gamma_{MV}$  — константы скорости активации антиген-презентирующих клеток. Ширина интервала сосуществования двух устойчивых стационарных решений достаточно мала, что определяет сложность поиска таких режимов.

Чувствительность иммунного ответа к повреждению клеток печени, описываемая с помощью функции  $\xi(m)$ , существенно влияет на размер области бистабильности по параметру  $\gamma_{MV}$ . Увеличение чувствительности может приводить к появлению свойства ультра-чувствительности, которое важно для реализации бистабильности. На рисунке 5.22 показано, что переход от  $\xi(m) = \max\{(1 - m/C^0)^5, 0\}$  к  $\xi(m) = \max\{(1 - m/(0.2C^0)), 0\}$ , приводящий к увеличению производной данной функции, приводит к сужению области существования устойчивого решения с меньшей вирусной нагрузкой.

Характер зависимости бистабильности от параметра  $\gamma_{MV}$ , продемонстрированный на рисунках 5.21—5.22, позволил выявить свойство гистерезиса в модели Марчука-Петрова. Гистерезис является свойством динамической системы, которое определяет зависимость её состояния при изменении параметров от предыстории. Так, если система находилась в состоянии с более высокой вирусной нагрузкой, как показано на рисунке 5.21 (верхняя ветвь), то увеличение параметра скорости активации макрофагов от значения  $1.5 \cdot 10^{-11}$  до  $1.52 \cdot 10^{-11}$  не приведет к переходу в состояние с более низкой вирусной нагрузкой. Для этого потребуется его увеличение до значения  $1.58 \cdot 10^{-11}$ , при котором происходит потеря устойчивости, и переход системы в другое стационарное состояние, с более низкой вирусной нагрузкой. Соответственно, если имеет место снижение скорости активации макрофагов до значения  $1.52 \cdot 10^{-11}$ , то система остается на нижней ветви устойчивого стационарного состояния. Случайные флуктуации данного параметра в области бистабильности будут проявляться в неоднозначности траектории системы, что связано с тем, на какой из ветвей находилась система в процессе изменений.

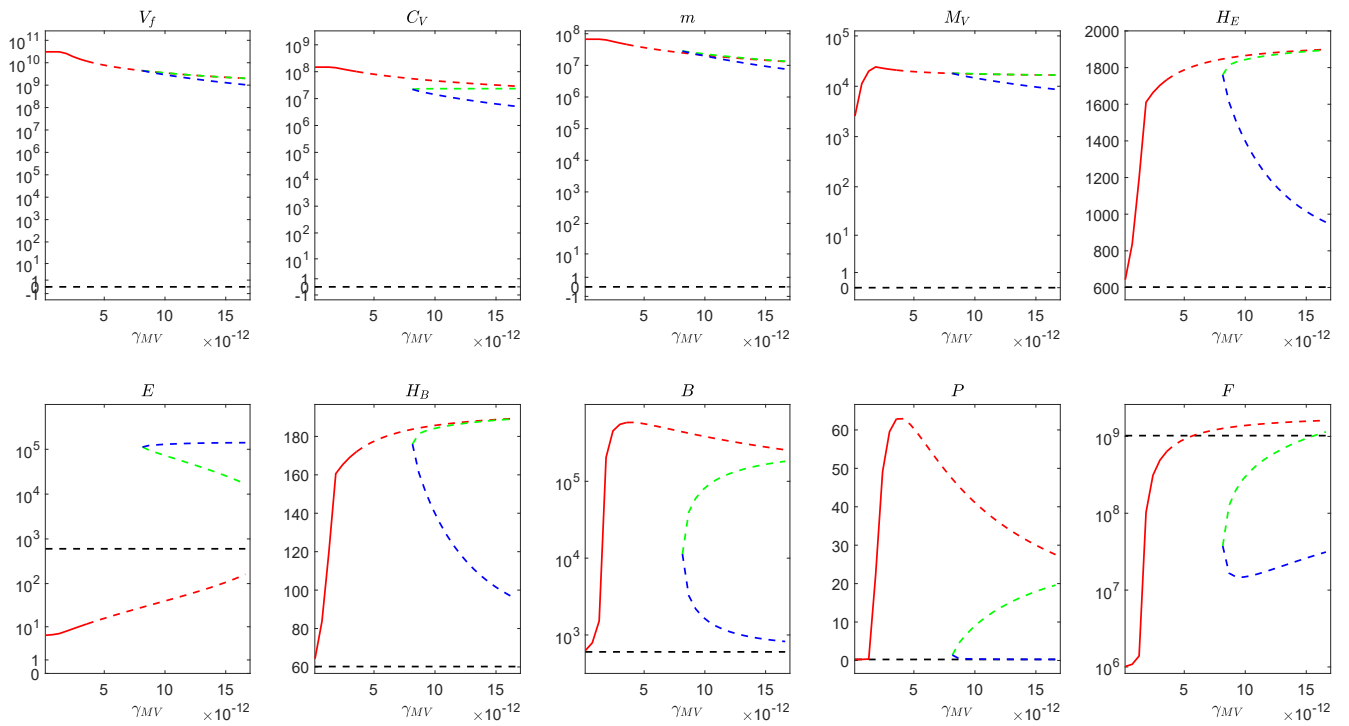


Рисунок 5.19 — Зависимость стационарных решений от параметра  $\gamma_{MV} \in [1.7 \cdot 10^{-13}, 1.7 \cdot 10^{-11}]$  для набора (а) (сплошная линия — устойчивое решение, пунктирная — неустойчивое).

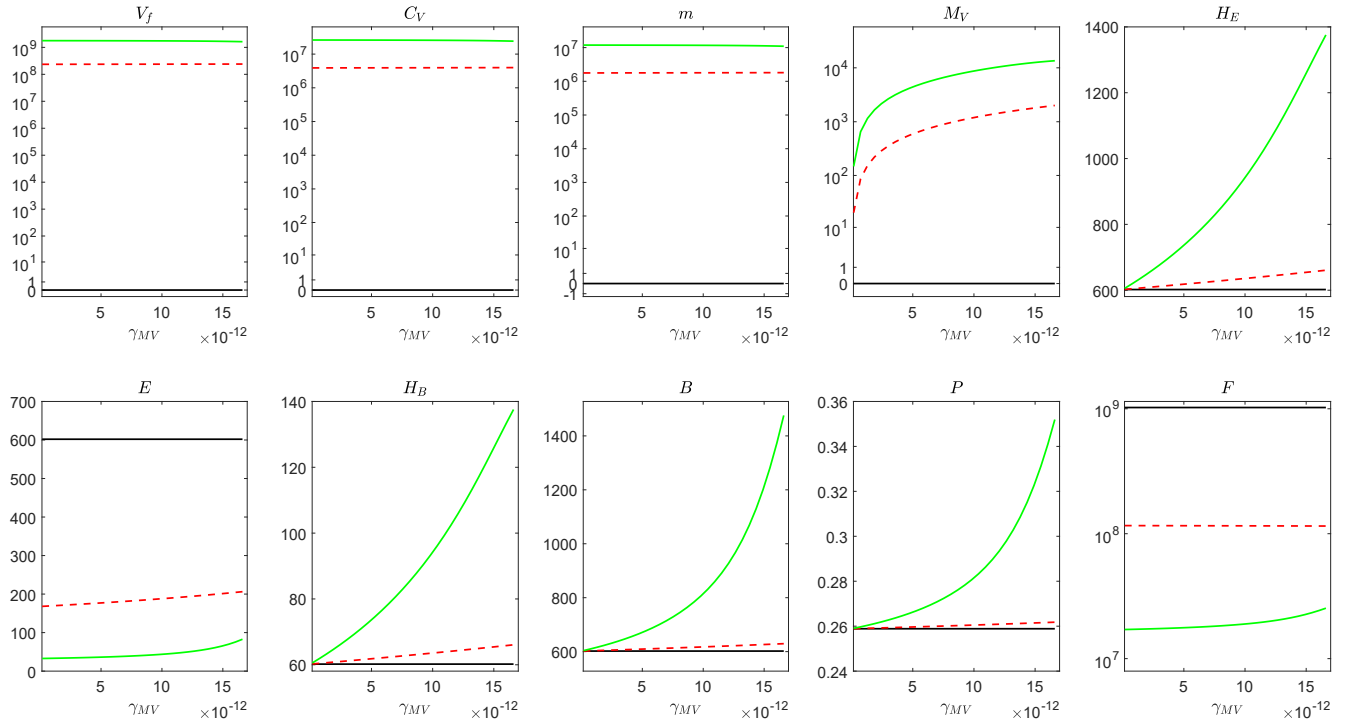


Рисунок 5.20 — Зависимость стационарных решений от параметра  $\gamma_{MV} \in [1.7 \cdot 10^{-13}, 1.7 \cdot 10^{-11}]$  для набора (b) (сплошная линия — устойчивое, пунктирная — неустойчивая)

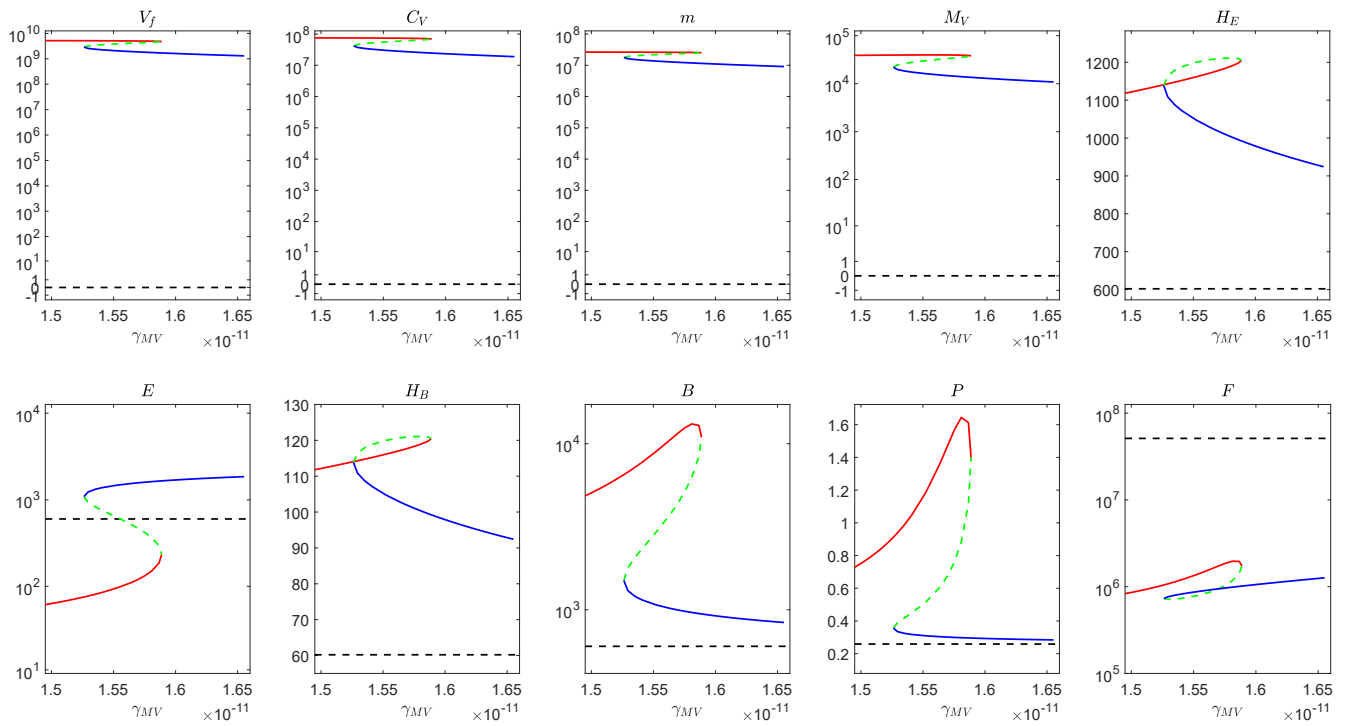


Рисунок 5.21 — Зависимость стационарных решений от параметра  $\gamma_{MV} \in [1.5 \cdot 10^{-11}, 1.7 \cdot 10^{-11}]$  для набора (с) для  $\xi(m) = (1 - m/C^0)^5$  (сплошная линия — устойчивое решение, пунктирная — неустойчивое)

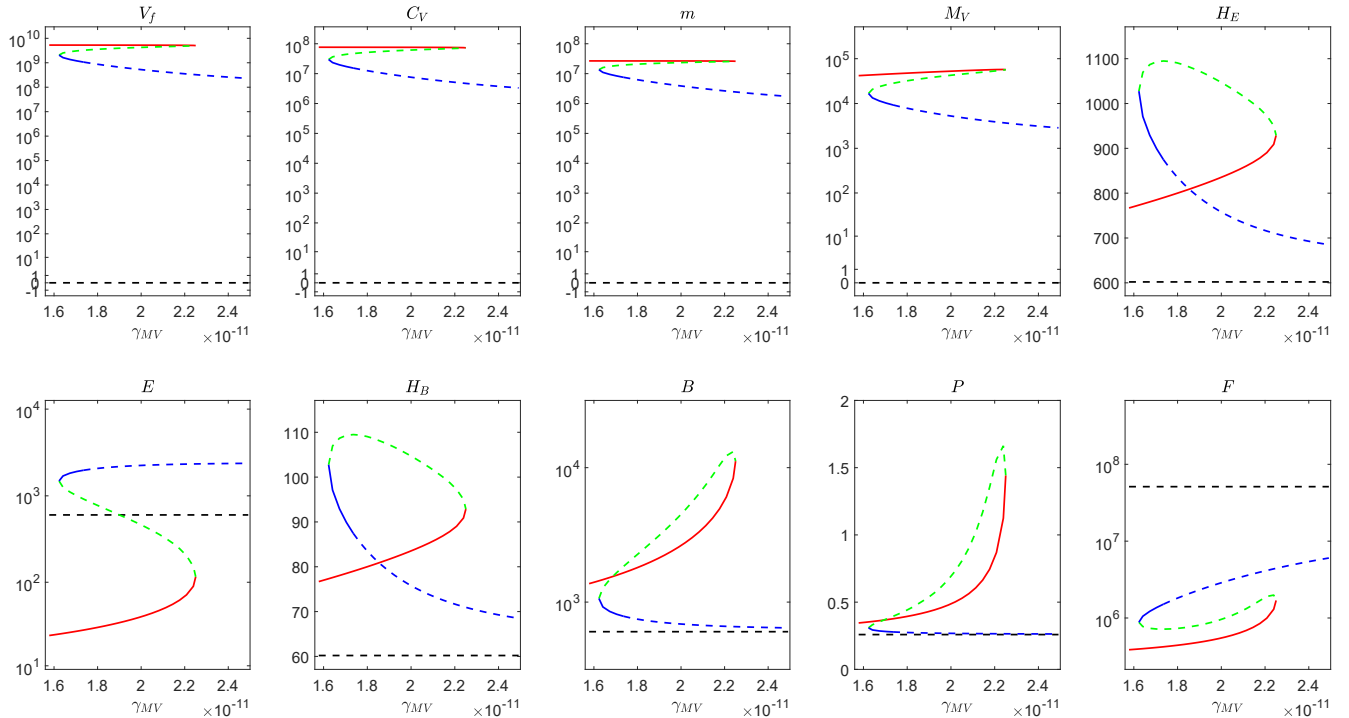


Рисунок 5.22 — Зависимость стационарных решений от параметра  $\gamma_{MV} \in [1.6 \cdot 10^{-11}, 2.5 \cdot 10^{-11}]$  для набора (с) для  $\xi(m) = (1 - m/(0.2C^0))$  (сплошная линия — устойчивое решение, пунктирная — неустойчивое).

## 5.7 Анализ периодических решений модели HBV

С помощью технологии, описанной в главах 2 и 3, стационарные и периодические решения модели HBV были рассчитаны для трех различных наборов значений параметров модели, которые будут называться наборами (а), (b) и (с). Набор (а) состоит из значений, указанных в таблице 4, за исключением значений параметров  $\gamma_{MV}$ ,  $b_p^E$ ,  $b_p^B$ , которые были выбраны равными  $3.09 \cdot 10^{-12}$  (мл/клеток)  $\cdot$  сут $^{-1}$ ,  $7.23 \cdot 10^{-10}$  (мл/клеток) $^2$   $\cdot$  сут $^{-1}$ ,  $1.38 \cdot 10^{-11}$  (мл/клеток) $^2$   $\cdot$  сут $^{-1}$ . Набор (b) состоит из тех же значений параметров, что и набор (а), за исключением параметра  $b_H^B$ , который был выбран равным  $8.31 \cdot 10^{-6}$  (мл/клеток)  $\cdot$  сут $^{-1}$ . Набор (с) состоит из тех же значений параметров, что и набор (b), за исключением параметра  $\gamma_{MV}$ , который был выбран равным  $1.66 \cdot 10^{-9}$  (мл/клеток)  $\cdot$  сут $^{-1}$ . Для каждого из трех наборов система имеет два стационарных решения, которые далее будем обозначать цифрами I и II. Решение I является тривиальным и одинаково во всех трех случаях. Округленные значения переменных в стационарных решениях и ведущие собственные значения  $\lambda$ , соответствующие этим решениям, представлены в таблице 29.

На рисунках 5.23—5.25 показаны периодические решения, вычисленные для значений параметров модели из наборов (а), (b), (с) соответственно, в интервалах  $[-\tau_5, 0]$  (красный) и  $[0, T]$  (синий), где  $T$  обозначает приблизительный наименьший период решения. Для вычисления периодических решений использовались параметры алгоритма, указанные в таблице 25, за исключением параметров  $\epsilon$ ,  $N_e$ ,  $\rho$ , описанных в разделе 3.1 и принятых равными  $10^6$ , 70, 0.01. Отметим, что периодическое решение для каждого из наборов (а), (b), (с) искалось в окрестности неустойчивого стационарного решения II. Периодическое решение, показанное на рисунке 5.23, соответствует набору параметров, соответствующему слабому иммунному ответу. Периодическое решение, показанное на рисунке 5.24, характеризуется большой амплитудой колебаний вирусной нагрузки и значительным поражением печени. Это решение соответствует случаю ослабленного гуморального иммунного ответа. Периодическое решение, показанное на рисунке 5.25, характеризуется меньшей вирусной нагрузкой и меньшим поражением печени, что является следствием того, что значения параметров модели



соответствуют более низкому порогу активации для антиген-презентирующих макрофагов.

Таблица 29. Значения переменных в стационарных решениях.

	I	IIa	IIb	IIc
$V_f$	0	$2.5 \cdot 10^{10}$	$2.7 \cdot 10^{10}$	$2.0 \cdot 10^7$
$C_V$	0	$1.4 \cdot 10^8$	$1.3 \cdot 10^8$	$1.8 \cdot 10^5$
$m$	0	$6.4 \cdot 10^7$	$7.0 \cdot 10^7$	$1.5 \cdot 10^5$
$M_V$	0	$3.8 \cdot 10^4$	$4.1 \cdot 10^4$	$1.7 \cdot 10^4$
$H_E$	602	$1.1 \cdot 10^5$	$1.0 \cdot 10^5$	$2.4 \cdot 10^3$
$E$	602	$1.7 \cdot 10^3$	$1.2 \cdot 10^4$	$5.7 \cdot 10^4$
$H_B$	60.2	$1.1 \cdot 10^4$	73.8	69.9
$B$	602	$2.0 \cdot 10^3$	604.9	603.5
$P$	0.26	24.1	0.31	0.28
$F$	$10^9$	$1.2 \cdot 10^8$	$1.4 \cdot 10^6$	$6.7 \cdot 10^8$
$\lambda$	0.0326	$0.0077 \pm 0.0983i$	$0.0523 \pm 0.1197i$	$0.0708 \pm 0.1280i$

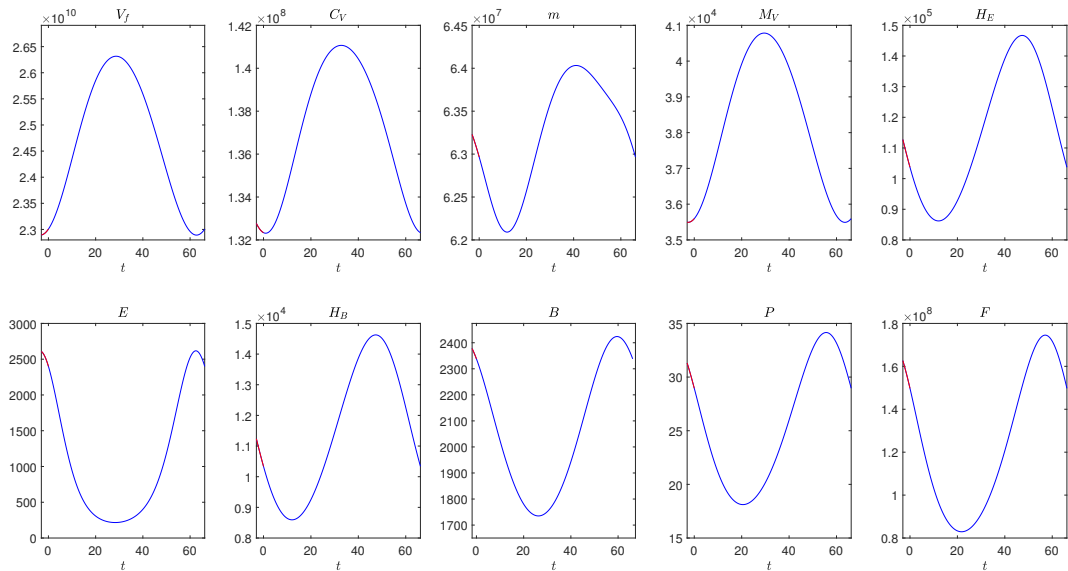


Рисунок 5.23 — Периодическое решение для набора (а) в интервале  $[-\tau_5, T]$ .

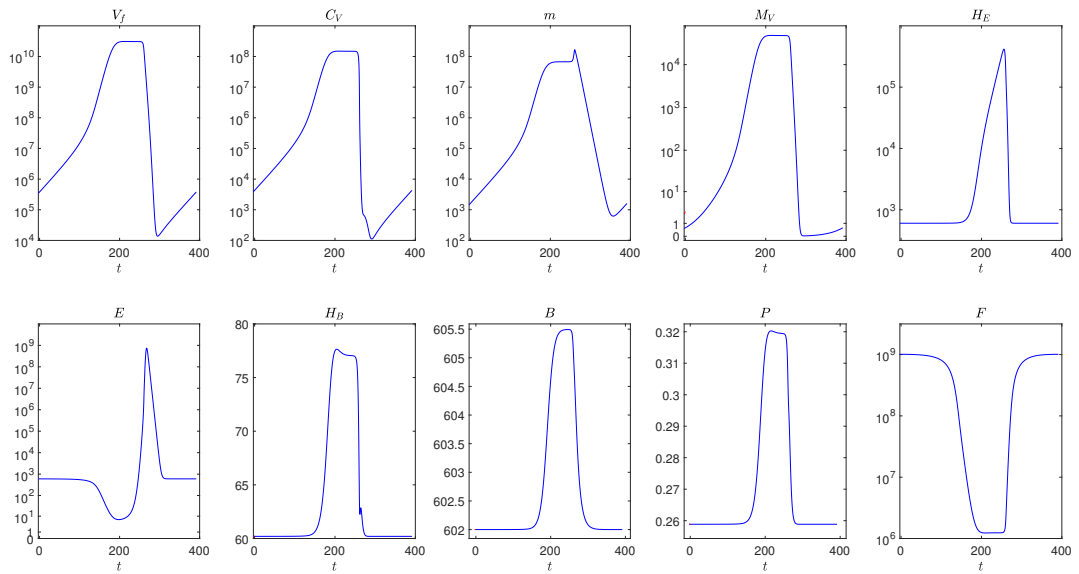


Рисунок 5.24 — Периодическое решение для набора (b) в интервале  $[-\tau_5, T]$ .

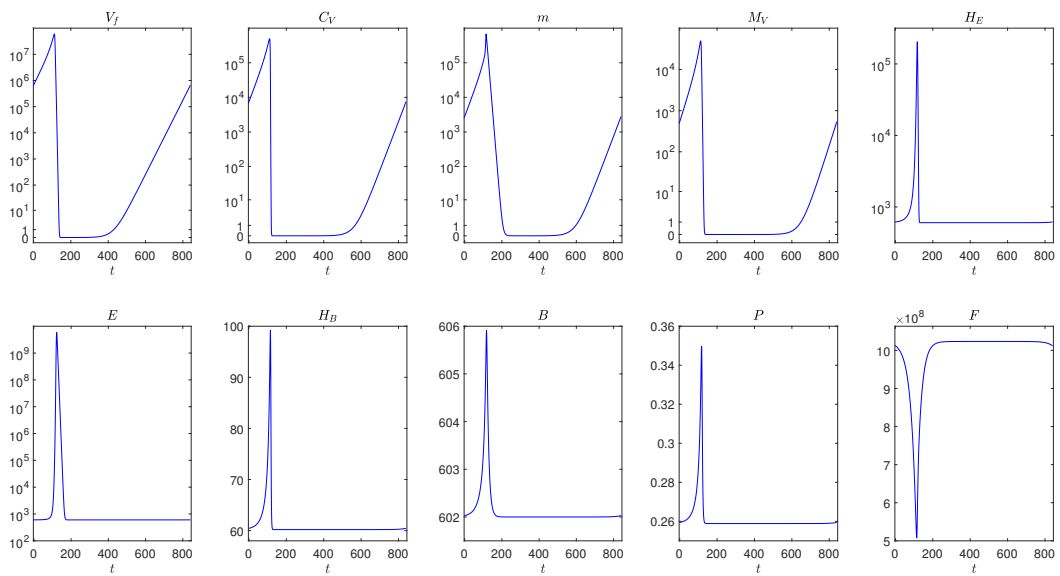


Рисунок 5.25 — Периодическое решение для набора (c) в интервале  $[-\tau_5, T]$ .

## Заключение

Основные результаты работы заключаются в следующем:

- Впервые было введено понятие оптимальных возмущений стационарных и периодических решений систем с запаздыванием.
- Были предложены алгоритмы вычисления оптимальных возмущений и проведено сравнение эффективности предложенных алгоритмов.
- Была разработана технология, включающая в себя методы вычисления всех стационарных решений систем с запаздыванием, анализа их устойчивости, исследования их зависимости от параметра, вычисления и анализа устойчивости периодических решений систем с запаздыванием. Разработанная технология также включает упомянутые выше алгоритмы вычисления оптимальных возмущений стационарных и периодических решений. Предложенная технология была реализована в виде программного комплекса DEODAN (Delay Equations Optimal Disturbances ANalysis).
- С помощью разработанной технологии для модели LCMV и модели HBV были впервые вычислены стационарные и периодические решения, соответствующие хроническим формам заболевания. Для этих моделей впервые было показано существование таких свойств, как бистабильность и гистерезис.
- Был выполнен анализ возможности использования оптимальных возмущений для перевода системы из одного устойчивого стационарного или периодического решения в другое. На примере модели LCMV было показано, что малое по норме оптимальное возмущение позволяет перейти из устойчивого стационарного решения, соответствующего хроническому заболеванию, в устойчивое стационарное решение, соответствующее состоянию здорового организма. Также на примере модели LCMV было показано, что малое по норме оптимальное возмущение устойчивого периодического решения, которое можно интерпретировать как хроническое заболевание, позволяет перевести систему в окрестность стационарного решения, соответствующего состоянию здорового организма.

## Список литературы

1. *Marchuk, G. I.* Mathematical models in immunology [Текст] / G. I. Marchuk. – Springer Verlag, 1983.
2. *Marchuk, G. I.* Mathematical modelling of immune response in infectious diseases [Текст] / G. I. Marchuk. – Springer Science & Business Media, 1997. – Т. 395.
3. *Nowak, M.* Virus dynamics: mathematical principles of immunology and virology: mathematical principles of immunology and virology [Текст] / M. Nowak, R. M. May. – Oxford University Press, 2000.
4. *Perelson, A. S.* Mathematical analysis of HIV-1 dynamics in vivo [Текст] / A. S. Perelson, P. W. Nelson // SIAM review. – 1999. – Т. 41, № 1. – С. 3-44.
5. *Bocharov, G. A.* Numerical modelling in biosciences using delay differential equations [Текст] / G. A. Bocharov, F. A. Rihan // Journal of Computational and Applied Mathematics. – 2000. – Т. 125, № 1-2. – С. 183-199.
6. *it Ferrell Jr, J. E.* Bistability, bifurcations, and Waddington's epigenetic landscape [Текст] / J. E. Ferrell Jr // Current biology. – 2012. – Т. 22, № 11. – С. 458-466.
7. Software for systems biology: from tools to integrated platforms [Текст] / S. Ghosh [и др.] // Nature Reviews Genetics. – 2011. – Т. 12, № 12. – С. 821-832.
8. *Olivier, B. G.* Modeling and simulation tools: from systems biology to systems medicine [Текст] / B. G. Olivier, M. J. Swat, M. J. Moné // Systems Medicine. – 2016. – С. 441-463.
9. HIV dynamics: modeling, data analysis, and optimal treatment protocols [Текст] / B. M. Adams [и др.] // Journal of Computational and Applied Mathematics. – 2005. – Т. 184, № 1. – С. 10-49.
10. An extremal shift method for control of HIV infection dynamics [Текст] / G. Bocharov [и др.] // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2015. – Т. 30, № 1. – С. 11-25.

11. *Hadjandreou, M. M.* Long-term HIV dynamics subject to continuous therapy and structured treatment interruptions [Текст] / M. M. Hadjiandreou, R. Conejeros, D. I. Wilson // *Chemical Engineering Science*. – 2009. – Т. 64, № 7. – С. 1600-1617.
12. *Boiko, A. V.* Fast computation of optimal disturbances for duct flows with a given accuracy [Текст] / A. V. Boiko, Y. M. Nechepurenko, M. Sadkane // *Computational Mathematics and Mathematical Physics*. – 2010. – Т. 50, № 11. – С. 1914-1924.
13. *Boiko, A. V.* Computing the maximum amplification of the solution norm of differential-algebraic systems [Текст] / A. V. Boiko, Y. M. Nechepurenko, M. Sadkane // *Computational Mathematics and Modeling*. – 2012. – Т. 23, № 2. – С. 216-227.
14. *Nechepurenko, Y. M.* A low-rank approximation for computing the matrix exponential norm [Текст] / Y. M. Nechepurenko, M. Sadkane // *SIAM journal on matrix analysis and applications*. – 2011. – Т. 32, № 2. – С. 349-363.
15. *Engelborghs, K.* DDE-BIFTOOL v. 2.00: a Matlab package for bifurcation analysis of delay differential equations [Текст] / K. Engelborghs, T. Luzyanina, G. Samaey // *TW Reports*. – 2001. – С. 61-61.
16. Maximum response perturbation-based control of virus infection model with time-delays [Текст] / G. A. Bocharov [и др.] // *Russian Journal of Numerical Analysis and Mathematical Modelling*. – 2017. – Т. 32, № 5. – С. 275-291.
17. Optimal Perturbations of Systems with Delayed Independent Variables for Control of Dynamics of Infectious Diseases Based on Multicomponent Actions [Текст] / G. A. Bocharov [и др.] // *Journal of Mathematical Sciences*. – 2021. – Т. 253, № 5. – С. 618-641.
18. Optimal disturbances of bistable time-delay systems modeling virus infections [Текст] / G. A. Bocharov [и др.] // *Doklady Mathematics*. – Pleiades Publishing, 2018. – Т. 98, № 1. – С. 313-316.

19. *Nechepurenko, Yu. M.* Computation of optimal disturbances for delay systems [Текст] / Yu. M. Nechepurenko, M. Yu. Khristichenko // Computational Mathematics and Mathematical Physics. – 2019. – Т. 59, № 5. – С. 731-746.
20. *Khristichenko, M. Y.* Optimal disturbances for periodic solutions of time-delay differential equations [Текст] / M. Y. Khristichenko, Y. M. Nechepurenko // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2022. – Т. 37, № 4. – С. 203-212.
21. Управление моделями вирусных инфекций с запаздывающими переменными на основе оптимальных возмущений [Текст] / Г. А. Бочаров [и др.] // Препринты Института прикладной математики им. М.В. Келдыша РАН. – 2017. – № 52.
22. *Нечепуренко, Ю. М.* Разработка и исследование алгоритмов вычисления оптимальных возмущений для систем с запаздыванием [Текст] / Ю. М. Нечепуренко, М. Ю. Христиченко // Препринты Института прикладной математики им. МВ Келдыша РАН. – 2018. – № 120.
23. Анализ бистабильности моделей вирусных инфекций с запаздывающим аргументом [Текст] / Ю. М. Нечепуренко [и др.] // Препринты Института прикладной математики им. М.В. Келдыша РАН. – 2019. – № 17. – С. 17-26.
24. Численный анализ стационарных решений систем с запаздывающим аргументом в математической иммунологии [Текст] / М. Ю. Христиченко [и др.] // Современ. мат. Фундам. направл. – 2022. – Т. 68, № 4. – С. 686-703.
25. *Khristichenko, M. Yu.* Computation of periodic solutions to models of infectious disease dynamics and immune response [Текст] / M. Yu. Khristichenko, Yu. M. Nechepurenko // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2021. – Т. 36, № 2. – С. 87-99.
26. Modelling chronic hepatitis B using the Marchuk-Petrov model [Текст] / M. Yu. Khristichenko [и др.] // Journal of Physics: Conference Series. – IOP Publishing, 2021. – Т. 2099, № 1. – С. 012036.

27. Bistability analysis of virus infection models with time delays [Текст] / Yu. Nechepurenko [и др.] // *Discrete & Continuous Dynamical Systems-S.* – 2020. – Т. 13, № 9. – С. 2385-2401.
28. *Khristichenko, M. Y.* Dependence of optimal disturbances on periodic solution phases for time-delay systems [Текст] / M. Yu. Khristichenko, Yu. M. Nechepurenko, G.A. Bocharov // *Russ. J. Numer. Anal. Math. Modelling.* – 2023. – Т. 38, № 2. – С. 89-98.
29. Numerical study of chronic hepatitis B infection using Marchuk-Petrov model [Текст] / M. Yu. Khristichenko [и др.] // *Journal of Bioinformatics and Computational Biology.* – 2023. – С. 234001.
30. *Христиченко, М. Ю.* Управление моделями вирусных инфекций на основе оптимальных возмущений / М. Ю. Христиченко // Труды 60-й Всероссийской научной конференции МФТИ. - Москва, 2017. – С. 183-184.
31. *Христиченко, М. Ю.* Анализ устойчивости стационарных состояний моделей динамики вирусных инфекций на примере модели ВЛХМ / М. Ю. Христиченко // Труды 61-й Всероссийской научной конференции МФТИ. - Москва, 2018. - С. 166-167.
32. Вычисление стационарных состояний моделей вирусных инфекций / М. Ю. Христиченко [и др.] // Аналитические и численные методы решения задач гидродинамики, математической физики и биологии: тезисы международной конференции. - Москва: ИПМ им. М.В. Келдыша РАН, 2019. - С. 110-113.
33. Computation of steady states of virus infection models / M. Yu. Khristichenko [и др.] // *Mathematical modelling in biomedicine: abstracts of International Conference.* - Moscow: Peoples' Friendship University of Russia, 2019. - С. 47-48.
34. Оптимальные возмущения стационарных состояний моделей вирусных инфекций / М. Ю. Христиченко [и др.] // Аналитические и численные методы решения задач гидродинамики, математической физики и биологии: тезисы международной конференции. - Москва: ИПМ им. М.В. Келдыша РАН, 2019. - С. 84-86.

35. Optimal disturbances of steady states of viral infection models / M. Yu. Khristichenko [и др.] // Mathematical modelling in biomedicine: abstracts of International Conference. - Moscow: Peoples' Friendship University of Russia, 2019. - С. 49-50.
36. *Христиченко, М. Ю.* Вычисление периодических решений систем с запаздыванием на примере модели ВЛХМ / М. Ю. Христиченко // Труды 63-й Всероссийской научной конференции МФТИ. 23–29 ноября 2020 года. Прикладная математика и информатика, Москва : МФТИ, 2020, С. 31-32.
37. Chronic hepatitis B modelling within the Marchuk-Petrov model / M. Yu. Khristichenko [и др.] // Тезисы Международной конференции "Марчуковские научные чтения" . - Институт вычислительной математики и математической геофизики СО РАН, 2021. - С. 169.
38. Optimal disturbances of periodic solutions of viral infections models / M. Yu. Khristichenko [и др.] // Abstracts of International online workshop "Mathematical Modelling in Biomedicine" . - Moscow: RUDN University. - С. 28.
39. Computation and tracing of stationary solutions of Marchuk-Petrov model / M. Yu. Khristichenko [и др.] // Abstracts of The 9th International Conference on Differential and Functional Differential Equations. - Moscow, 2022. - С. 61-62.
40. Computation and tracing of periodic solutions of Marchuk-Petrov model / M. Yu. Khristichenko [и др.] // Abstracts of The Thirteenth International Multiconference "Bioinformatics of Genome Regulation and Structure/Systems Biology (BGRS/SB-2022)" . Novosibirsk, 2022. - С. 212-213.
41. *Bocharov, G.* Understanding experimental LCMV infection of mice: the role of mathematical models [Текст] / G. Bocharov, J. Argilaguet, A. Meyerhans // Journal of Immunology Research. – 2015. – Т. 2015. - С. 1-10.
42. *Bocharov, G. A.* Modelling the dynamics of LCMV infection in mice: conventional and exhaustive CTL responses [Текст] / G. A. Bocharov // Journal of theoretical biology. – 1998. – Т. 192, № 3. – С. 283-308.



43. *Bocharov, G. A.* Applied problems of mathematical modeling in immunology [Текст] / G. A. Bocharov, G. I. Marchuk // Computational mathematics and mathematical physics. – 2000. – Т. 40, № 12. – С. 1830-1844.
44. Mathematical immunology of virus infections [Текст] / G. Bocharov [и др.]. – Springer International Publishing, 2018. – Т. 245.
45. Physics of Transitional Shear Flows: Instability and Laminar–Turbulent Transition in Incompressible Near-Wall Shear Layers [Текст] / A. V. Boiko [и др.]. – Springer Science & Business Media, 2011. – Т. 98.
46. *Demyanko, K. V.* A Newton-type method for non-linear eigenproblems [Текст] / K. V. Demyanko, Y. M. Nechepurenko, M. Sadkane // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2017. – Т. 32, № 4. – С. 237-244.
47. *Effenberger, C.* Robust successive computation of eigenpairs for nonlinear eigenvalue problems [Текст] / C. Effenberger // SIAM Journal on Matrix Analysis and Applications. – 2013. – Т. 34, № 3. – С. 1231-1256.
48. *Forsythe, G. E.* Computer methods for mathematical computations [Текст] / G. E. Forsythe // Prentice-Hall series in automatic computation. – 1977. – Т. 259.
49. *Geddes, K. O.* Algorithms for computer algebra [Текст] / K. O. Geddes, S. R. Czapor, G. Labahn. – Springer Science & Business Media, 1992.
50. *Golub, G. H.* Matrix computations [Текст] / G. H. Golub, C. F. Van Loan. – JHU press, 2013.
51. *Hale, J. K.* Theory of Functional Differential Equations [Текст] / J. K. Hale. – Springer-Verlag, 1977.
52. *Hartshorne, R.* Algebraic geometry [Текст] / R. Hartshorne. - Springer Science & Business Media, 2013.
53. *Dormand, J. R.* A family of embedded Runge-Kutta formulae [Текст] / J. R. Dormand, P. J. Prince // J. Comp. Appl. Math. - 1980. - Т. 6, № 1. - С. 19—26.

54. *Kahaner, D.* Numerical methods and software [Текст] / D. Kahaner, C. Moler, S. Nash. – Prentice-Hall, Inc., 1989.
55. *Marchuk, G. I.* Mathematical model of antiviral immune response. II. Parameters identification for acute viral hepatitis B [Текст] / G. I. Marchuk, A. A. Romanyukha, G. A. Bocharov // Journal of theoretical biology. – 1991. – Т. 151, № 1. – С. 41-69.
56. *Parlett, B. N.* The symmetric eigenvalue problem [Текст] / B. N. Parlett. – Society for Industrial and Applied Mathematics, 1998.
57. *Schmid, P. J.* Stability and transition in shear flows [Текст] / P. J. Schmid, D. S. Henningson. – Springer-Verlag, 2001.
58. *Seydel, R.* Practical bifurcation and stability analysis [Текст] / R. Seydel. – Springer Science & Business Media, 2009. – Т. 5.
59. *Sklyarova, E. V.* Numerical steady state analysis of the Marchuk–Petrov model of antiviral immune response [Текст] / E. V. Sklyarova, Y. M. Nechepurenko, G. A. Bocharov // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2020. – Т. 35, № 2. – С. 95-110.
60. *Stewart, G. W.* Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices [Текст] / G. W. Stewart // Numerische Mathematik. – 1976. – Т. 25, № 2. – С. 123-136.
61. *Wanner, G.* Solving ordinary differential equations II [Текст] / G. Wanner, E. Hairer. – Springer Berlin Heidelberg, 1996. – Т. 375.