

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

Кафедра цифровых технологий

*Использование метода перевала в нестационарных  
задачах квантовой механики*

ВКР Бакалаврская работа  
02.03.01 Математика и компьютерные науки  
Распределенные системы и искусственный интеллект

Допущено к защите в ГЭК

Зав. кафедрой	_____	С.Д. Кургалин, д. ф.-м. н., профессор	___ . ___ . 2017
Обучающийся	_____	А.А. Махно, 4 курс, д/о	
Руководитель	_____	А.В. Флегель, к. ф.-м. н., доцент	

Воронеж 2017

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Факультет компьютерных наук  
Кафедра цифровых технологий

УТВЕРЖДАЮ  
заведующий кафедрой

\_\_\_\_\_ Кургалин С.Д.,  
*подпись* *расшифровка подписи*  
\_\_\_ . \_\_\_ . 2017

**ЗАДАНИЕ**  
**НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**  
**ОБУЧАЮЩЕГОСЯ МАХНО АРТЕМА АНДРЕЕВИЧА**

*фамилия, имя, отчество*

1. Тема работы «Использование метода перевала в нестационарных задачах квантовой механики», утверждена решением ученого совета факультета компьютерных наук от \_\_\_\_ . \_\_\_\_ . 20 \_\_\_\_
2. Направление подготовки / специальность 02.03.01 Математика и компьютерные науки  
*шифр, наименование*
3. Срок сдачи студентом законченной работы \_\_\_\_ . \_\_\_\_ . 20 \_\_\_\_
4. Календарный план: (строится в соответствии со структурой ВКР)

№	Структура ВКР	Сроки выполнения	Примечание
1	Введение	26.10.2016	
2	Глава 1. Постановка задачи	30.11.2016	
3	Глава 2. Описание метода перевала	16.01.2017	
4	Глава 3. Методы вычисления функции $M(\epsilon, t)$	21.03.2017	
5	Глава 4. Анализ результатов	12.04.2017	
6	Заключение	3.05.2017	

Обучающийся \_\_\_\_\_ Махно А.А.  
*подпись* *расшифровка подписи*

Руководители \_\_\_\_\_ Флегель А.В.  
*подпись* *расшифровка подписи*

## Реферат

Бакалаврская работа 47 с., 16 рисунков

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, КВАНТОВАЯ МЕХАНИКА,  
МЕТОД ПЕРЕВАЛА, ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ.

Объектом исследования являются интегралы от быстро осциллирующих функций, возникающие в квантово-механических задачах моделирования взаимодействия атома с лазерным импульсом.

Цель работы – получение аналитических формул для решений нестационарных уравнений квантовой механики с использованием метода перевала, а также выявление границ их применимости.

В процессе выполнения работы была получена аналитическая формула перевальной оценки интеграла от быстро осциллирующей функции, интенсивность осцилляций которой определяется параметрами задачи. Для получения точного результата для интеграла разработан численный метод расчета с использованием библиотек численного интегрирования.

В результате работы была выявлена область параметров задачи, для которой наблюдается хорошее согласие полученной аналитической оценки с исходным интегралом. Проанализированы причины нарушения согласия вне указанной области параметров. Приведенный в работе анализ может рассматриваться как одно из обоснований приближенных аналитических квазиклассических оценок в задачах квантовой механики.

Результаты работы могут использоваться при математическом моделировании взаимодействия атомов и молекул с интенсивными электромагнитными полями, а также в других задачах теоретической физики.

## Содержание

<b>Введение</b>	<b>5</b>
<b>1 Постановка задачи</b>	<b>7</b>
<b>2 Описание метода перевала</b>	<b>11</b>
<b>3 Методы вычисления функции <math>\mathcal{M}(\epsilon, t)</math></b>	<b>16</b>
3.1 Аналитическое решение . . . . .	16
3.2 Численное решение . . . . .	23
<b>4 Анализ результатов</b>	<b>29</b>
<b>Заключение</b>	<b>37</b>
<b>Список используемой литературы</b>	<b>38</b>
<b>Приложение А Листинг программы</b>	<b>40</b>

## Введение

В последние годы активно развиваются высокопроизводительные вычисления с использованием массивных вычислительных кластеров и суперкомпьютеров. Область научных и технических задач, решаемых на параллельных многопроцессорных системах, стремительно расширяется и включает в себя сложные многочастичные динамические задачи, решение которых еще недавно казалось невозможным (см., например, [1]). Тем не менее, математическое и компьютерное моделирование некоторых экспериментально наблюдаемых процессов сталкивается с существенными сложностями, обусловленными, с одной стороны, все еще недостаточной мощностью вычислительных ресурсов, с другой – принципиальной невозможностью исключить ошибки численного решения даже при использовании наиболее продвинутых алгоритмов расчетов.

Примером такой задачи является описание взаимодействия атомов и молекул с интенсивным низкочастотным лазерным полем, влияние которого приводит к появлению ряда нелинейных многофотонных явлений, таких как, генерация высоких гармоник лазерного поля (атом излучает фотон с энергией, в десятки или сотни раз превышающей энергию фотона поля накачки) или надпороговая ионизация атомов (электрон вырывается из связанного состояния, поглощая десятки или сотни фотонов лазерного поля) [2, 3, 4]. Численное решение нестационарного уравнения Шредингера, являющегося математической (квантовомеханической) моделью атома в лазерном поле, является крайне сложной задачей даже в рамках одноэлектронной модели, в которой взаимодействие активного электрона с атомным остовом описывается модельным потенциалом. Сложность этой задачи связана с экспоненциальным ростом размерности задачи (из-за роста числа узлов сетки четырехмерного пространства  $\mathbf{r} \otimes t$ ) при увеличении интенсивности лазерного поля или его длины волны [5]. В связи с этим актуальным остается использование, во-первых, модельных подходов к описанию содержательной части поставленной задачи, снижающих размерность исходных уравнений, и, во-вторых, приближенных методов вычислений, позволяющих получить простые аналитические результаты, удовлетворительно описывающие интересные процессы как качественно, так и количественно.

Одним из приближенных подходов для получения аналитического ре-

зультата для квантовой амплитуды многофотонного процесса в интенсивном низкочастотном лазерном поле является квазиклассический подход, основанный на использовании метода перевала и его модификаций (см., например, [6, 7] и ссылки там). В рамках такого подхода удастся получить замкнутые аналитические формулы для амплитуд физических процессов, ценность которых заключается в прозрачной физической интерпретации ряда сложных нелинейных эффектов, а также адекватной их численной оценки. Как правило, простое физическое объяснение изучаемого явления существенно более сложно извлечь из результатов ресурсозатратных вычислений.

**Целью** настоящей работы является получение перевальной оценки для волновой функции атомного электрона в лазерном поле и проверка применимости такой оценки путем сравнения с точным результатом численного расчета. При формулировании проблемы, будут получены основные соотношения для волновой функции квазистационарного состояния электрона в короткодействующем потенциале и выделены временные интегралы для ключевых элементов модели. При анализе этих интегралов будут рассмотрены решения уравнений на перевальные точки подынтегральных выражений и проанализирован их вклад. Для получения точного численного результата в работе используется алгоритм на базе быстрого преобразования Фурье для интегрирования медленно затухающей осциллирующей функции.

Текст работы организован следующим образом: постановка задачи представлена в главе 1; в главе 2 описан принцип метода перевала; в главе 3 получена аналитическая формула для поставленной задачи, и анализируется метод численного интегрирования исходной функции; в главе 5 проведено сравнение численного и аналитического решения; в заключении резюмируются основные результаты и выводы работы; в приложении приводится листинг расчетной программы.

# 1 Постановка задачи

В квантовой механике одноэлектронная математическая модель атома, подверженного воздействию лазерного импульса, описывается нестационарным уравнением Шредингера [8] для электронной волновой функции  $\Psi(\mathbf{r}, t)$ :

$$\left[ i\hbar \frac{\partial}{\partial t} + \frac{\hbar^2}{2m} \nabla^2 - U(r) - V(\mathbf{r}, t) \right] \Psi(\mathbf{r}, t) = 0, \quad (1.1)$$

где  $U(r)$  – потенциал взаимодействия электрона с атомным остовом,  $V(\mathbf{r}, t) = |e|(\mathbf{F}(t) \cdot \mathbf{r})$  – взаимодействие электрона (с зарядом  $e$  и массой  $m$ ) с лазерным импульсом, электрическое поле которого описывается вектором напряженности  $\mathbf{F}(t)$ . Пусть электрон-атомное взаимодействие моделируется короткодействующим потенциалом  $U(r)$ , поддерживающим одно связанное сферически-симметричное состояние  $\Psi_0(\mathbf{r}, t)$  с энергией  $E_0 = -\hbar^2 \kappa^2 / (2m)$ .

Для больших  $r$ , вне области действия потенциала  $U(r)$  волновая функция  $\Psi(\mathbf{r}, t)$  ведет себя как суперпозиция расходящихся волн, соответствующих потоку уходящих на бесконечность частиц с энергиями  $E = \mathcal{E}(F, \omega) + n\hbar\omega$ , где  $\mathcal{E}(F, \omega)$  – положение атомного уровня в лазерном поле (т.е. энергия эволюционировавшего в поле  $\mathbf{F}(t)$  состояния  $\Psi_0$ ),  $\hbar\omega$  – энергия лазерного фотона, соответствующая несущей частоте  $\omega$ . Функция  $\Psi(\mathbf{r}, t)$  с требуемым асимптотическим поведением при  $r \rightarrow \infty$  может быть выражена через нестационарную функцию Грина  $G(\mathbf{r}, t; \mathbf{r}', t')$  для свободного электрона в поле  $\mathbf{F}(t)$  [9]:

$$\Psi(\mathbf{r}, t) = -\frac{2\pi\hbar^2}{m\kappa} \int_{-\infty}^t e^{-i\epsilon t'/\hbar} G(\mathbf{r}, t; 0, t') f(t') dt', \quad (1.2)$$

где  $f(t)$  – неизвестная функция, которую необходимо определить из граничного условия для  $\Psi(\mathbf{r}, t)$  при  $r \rightarrow 0$ ,  $\epsilon$  – квазиэнергия состояния  $\Psi(\mathbf{r}, t)$ , переходящая в  $E_0$  при выключении лазерного поля. Функция Грина  $G(\mathbf{r}, t; \mathbf{r}', t')$  удовлетворяет уравнению

$$\left[ i\hbar \frac{\partial}{\partial t} + \frac{\hbar^2}{2m} \nabla^2 - V(\mathbf{r}, t) \right] G(\mathbf{r}, t; \mathbf{r}', t') = \delta(\mathbf{r} - \mathbf{r}') \delta(t - t'), \quad (1.3)$$

и имеет следующий вид:

$$G(\mathbf{r}, t; \mathbf{r}', t') = -\theta(t - t') \frac{i}{\hbar} \left[ \frac{m}{2\pi i \hbar (t - t')} \right]^{3/2} e^{iS(\mathbf{r}, t; \mathbf{r}', t')/\hbar}, \quad (1.4)$$

где  $\theta(x)$  – функция Хевисайда, и  $S$  – классическое действие для электрона в лазерном поле  $\mathbf{F}(t)$ :

$$S(\mathbf{r}, t; \mathbf{r}', t') = \frac{m}{2(t - t')} \left( \mathbf{r} - \mathbf{r}' - \frac{e}{mc} \int_{t'}^t \mathbf{A}(\tau) d\tau \right)^2 - \frac{e^2}{2mc^2} \int_{t'}^t \mathbf{A}^2(\tau) d\tau - \frac{e}{c} [\mathbf{r} \mathbf{A}(t) - \mathbf{r}' \mathbf{A}(t')]. \quad (1.5)$$

В (1.5)  $\mathbf{A}(t)$  – векторный потенциал лазерного поля,

$$\mathbf{F}(t) = -\frac{1}{c} \frac{\partial \mathbf{A}(t)}{\partial t}. \quad (1.6)$$

При  $r \rightarrow 0$  функция  $\Psi(\mathbf{r}, t)$  удовлетворяет следующему краевому условию:

$$\Psi(\mathbf{r}, t) \simeq \left( \frac{1}{r} + B(\epsilon) \right) f(t) e^{-i\epsilon t/\hbar}, \quad (1.7)$$

где явный вид зависимости  $B(\epsilon)$  определяется характером взаимодействия  $U(r)$ . Например, для потенциала нулевого радиуса  $B = -\kappa = -\sqrt{2m|E_0|}/\hbar = \text{const}$ .

Сшивая решение (1.2) с краевым поведением (1.7), получим систему уравнений для коэффициентов Фурье функции  $f(t)$ ,

$$f_n = \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} f(t) e^{in\omega_\tau t} dt, \quad \omega_\tau = \frac{2\pi}{\mathcal{T}}, \quad (1.8)$$

где  $\mathcal{T}$  – временной интервал между последовательными импульсами лазерного поля в бесконечной периодической последовательности импульсов.

Результирующая система уравнений для  $f_n$  имеет вид [7]:

$$\sum_{n=-\infty}^{\infty} \mathcal{R}(\epsilon + n\hbar\omega_\tau) f_n e^{-in\omega_\tau t} = \sum_{m=-\infty}^{\infty} e^{-im\omega_\tau t} f_m \mathcal{M}(\epsilon + m\hbar\omega_\tau, t), \quad (1.9)$$

где

$$\mathcal{R}(E) = B(E) - i\sqrt{2mE}/\hbar, \quad (1.10)$$



$$\mathcal{M}(\epsilon, t) = \sqrt{\frac{m}{2\pi i\hbar}} \int_0^\infty \frac{e^{i\epsilon\tau/\hbar}}{\tau^{3/2}} [e^{iS(t, t-\tau)/\hbar} - 1] d\tau. [7] \quad (1.11)$$

В данной работе мы анализируем ключевой элемент системы (1.9) – матричный элемент  $\mathcal{M}$ . Используя метод перевала, будет получена аналитическая оценка для интеграла в (1.11), применимость которой будет проверена численным сравнением с точным результатом для  $\mathcal{M}(\epsilon, t)$ .

В дальнейшем будет использоваться атомная система единиц, в которой  $|e| = m = \hbar = 1$ .

Явный вид напряженности электрического поля лазерного импульса  $\mathbf{F}(t) = \mathbf{e}_z F(t)$  задается следующей функцией:

$$F(t) = F_0 e^{-t^2/a^2} \cos(\omega t), \quad (1.12)$$

где  $F_0$  – амплитуда поля,  $\omega$  – несущая частота, и  $a$  – определяет ширину гауссовской огибающей импульса. Приведем также выражения для вычисления классического действия  $S(t', t)$ :

$$S(t, t') = -\frac{1}{2} \int_{t'}^t [\alpha(\epsilon, t, t')]^2 d\epsilon, \quad (1.13)$$

где

$$\alpha(\epsilon, t, t') = A(\epsilon) - \frac{1}{(t - t')} \int_{t'}^t A(\tau) d\tau, \quad (1.14)$$

$$A(t) = - \int_{-\infty}^t F(\tau) d\tau. \quad (1.15)$$

Функции  $F(t)$  и  $A(t)$  представлены на рисунке (1.1)

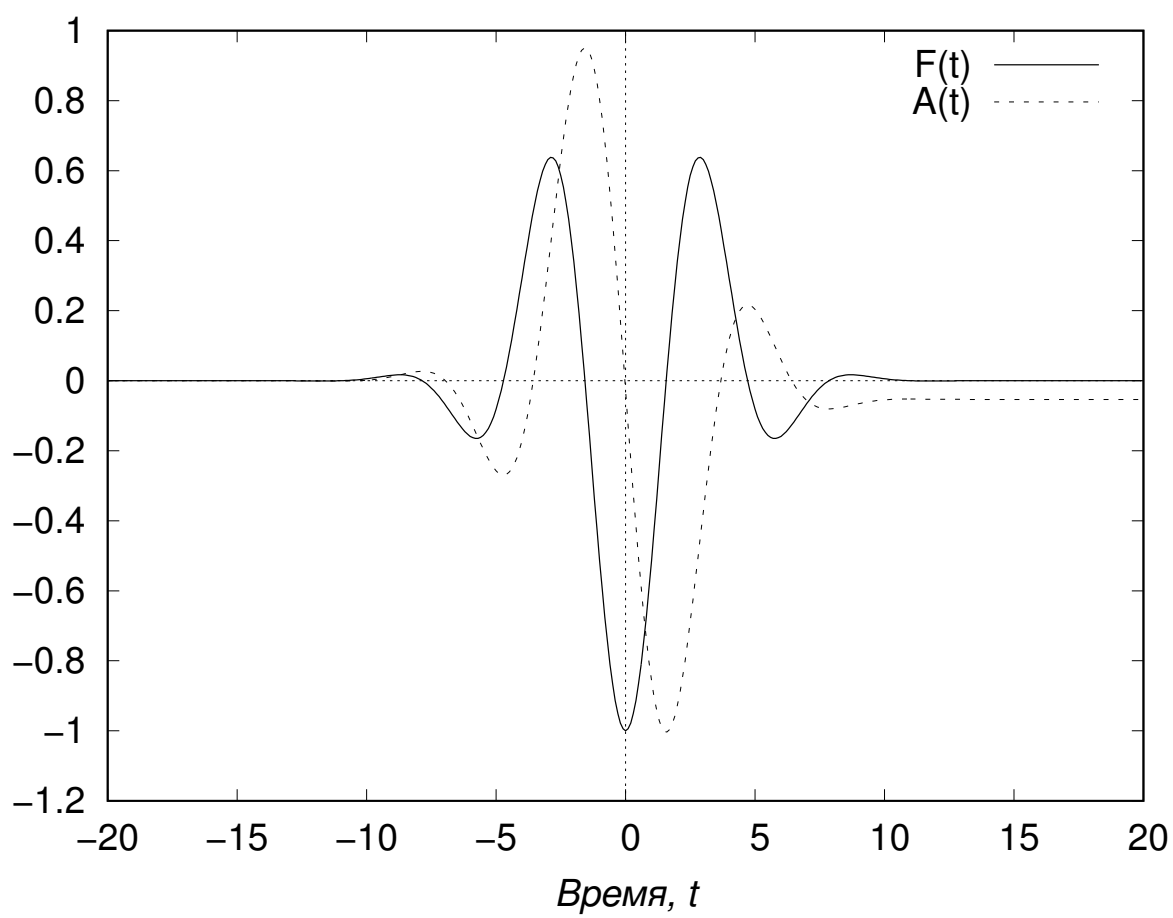


Рисунок 1.1 – Функции  $F(t)$  и  $A(t)$

## 2 Описание метода перевала

В данной главе дается основное содержание метода перевала для оценки интеграла вида 1.11.

Метод перевала применяется для оценки при больших значениях параметра  $\lambda$  контурных интегралов вида

$$I(\lambda) = \int_C \phi(z) e^{\lambda f(z)} dz, \quad (2.1)$$

где  $f(z)$  и  $\phi(z)$  функции, аналитические вдоль линии интегрирования. Интегралы вида (2.1) встречаются также при представлении многих специальных функций, решении дифференциальных уравнений, как обыкновенных, так и с частными производными [10].

Рассмотрим частный случай, а именно - действительные интегралы вида

$$I(\lambda) = \int_a^b \phi(t) e^{\lambda f(t)} dt. \quad (2.2)$$

Предположим, что  $f(t)$  имеет на отрезке  $(a, b)$  один резко выраженный максимум. Чем больше значение параметра  $\lambda$ , тем резче выражается этот максимум, и поэтому ясно, что при больших  $\lambda$  основной вклад в значение интеграла дает окрестность точки максимума.

В основе этого метода лежит лемма:

*Лемма* : Пусть дан интеграл

$$I(\lambda) = \int_0^a \phi(t) e^{-\lambda t^\alpha} dt \quad (0 < a \leq \infty, \alpha > 0),$$

где  $\phi(t)$  при  $|t| < 2h$  представляется сходящимся рядом

$$\phi(t) = t^\beta (c_0 + c_1 t + \dots + c_n t^n + \dots), \quad \beta > -1,$$

причем  $\int_0^a |\phi(t)| e^{-\lambda_0 t^\alpha} dt \leq M$  для некоторого  $\lambda_0$ . Тогда имеет место асимптотическое разложение

$$I(\lambda) \sim \sum_{n=0}^{\infty} \frac{c_n}{\alpha} \Gamma\left(\frac{\beta + n + 1}{\alpha}\right) \lambda^{-\frac{\beta + n + 1}{\alpha}}, \quad (2.3)$$

где  $\Gamma$  - гамма-функция Эйлера.

К доказанной лемме сводится оценка интеграла (2.2).

*Теорема 1.* Пусть интеграл (2.2) абсолютно сходится для некоторого  $\lambda = \lambda_0$ , т. е.

$$\int_a^b |\phi(t)| e^{\lambda_0 f(t)} dt \leq M,$$

и  $f(t)$  достигает своего наибольшего значения во внутренней точке  $t_0$  отрезка  $(, b)$ , в окрестности  $|t - t_0| < \delta$  которой  $f(t)$  представляется рядом

$$f(t) = f(t_0) + a_2(t - t_0)^2 + \dots + a_n(t - t_0)^n + \dots \quad (a_2 < 0),$$

причем существует  $h > 0$  такое, что вне этой окрестности  $f(t_0) - f(t) > h$ . Пусть еще функция  $t = \psi(\tau)$  определяется в окрестности точки  $\tau = 0$  из уравнения  $f(t_0) - f(t) = \tau^2$ , причем в этой окрестности

$$\phi[\psi(\tau)]\psi'(\tau) = \sum_{n=0}^{\infty} c_n \tau^n. \quad (2.4)$$

Тогда интеграл (2.2) имеет асимптотическое разложение

$$I(\lambda) = \int_a^b \phi(t) e^{\lambda f(t)} dt \sim e^{\lambda f(t_0)} \sqrt{\frac{\pi}{\lambda}} \sum_{n=0}^{\infty} \frac{c_2 n (2n)!}{\lambda^n 4^n n!}.$$

Эта теорема относится к случаю, когда наибольшее значение  $f(t)$  достигается во внутренней точке отрезка  $(a, b)$ .

*Теорема 2.* Пусть интеграл (2.2) абсолютно сходится для некоторого  $\lambda = \lambda_0$  (см теорему 1) и  $f(t)$  достигает наибольшего значения в точке  $t = a$ , аналитична в этой точке ( $f'(a) \neq 0$ ), и существует  $h > 0$  такое, что  $f(a) - f(t) > h$  вне некоторой окрестности точки  $a$ . Пусть еще функция  $t = \psi(\tau)$  определяется в окрестности точки  $\tau = 0$  из уравнения  $f(a) - f(t) = \tau$ , причем в этой окрестности имеет место разложение (2.4). Тогда

$$I(\lambda) = \int_a^b \phi(t) e^{\lambda f(t)} dt \sim \frac{e^{f(a)}}{\lambda} \sum_{n=0}^{\infty} \frac{n! c_n}{\lambda^n}. \quad (2.5)$$

Суть метода перевала состоит в том, что при больших значениях параметра  $\lambda$  величина интеграла

$$I(\lambda) = \int_C \phi(t) e^{\lambda f(z)} dz$$

в основном определяется тем участком пути интегрирования  $C$ , на котором  $|e^{\lambda f(z)}| = e^{\lambda \operatorname{Re} f(z)}$ , т. е.  $\operatorname{Re} f(z)$  велика по сравнению со значениями на остальной части. При этом интеграл оценивается тем легче, чем меньше этот участок и чем круче падает величина  $\operatorname{Re} f(z)$ . В соответствии со сказанным, при применении метода перевала стараются деформировать путь интегрирования  $C$  в наиболее удобный путь  $\tilde{C}$ , пользуясь тем, что по теореме Коши такая деформация не меняет величины интеграла.[10]

Для геометрической иллюстрации метода положим  $z = x + iy$ , и представим

$$u = \operatorname{Re} f(z),$$

как поверхность  $S$  в пространстве  $(x, y, u)$ . Так как функция  $u$  гармоническая, то  $S$  не может иметь точек максимума и минимума, а точки, в которых  $f'(z) = 0$ , будут для нее точками перевала (седловыми точками, рисунок 2.1).

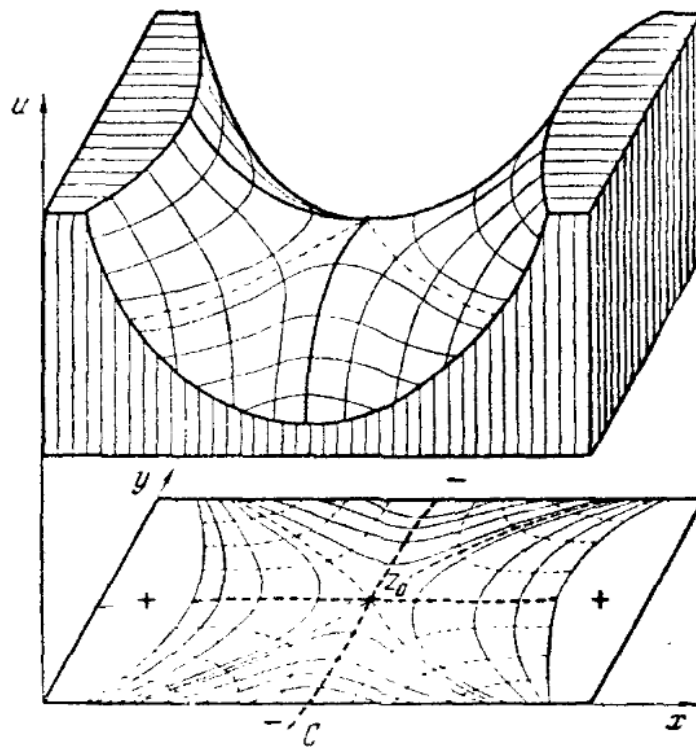


Рисунок 2.1 – Седловые точки

Наиболее удобный для оценки путь интегрирования  $\tilde{C}$ , в каждой точке должен проходить в направлении наиболее быстрого изменения  $\operatorname{Re} f(z)$ , а так как функция  $f(z)$  аналитическая, то это направление должно совпадать с линией, на которой  $\operatorname{Im} f(z) = \text{const}$ .

Также, новый контур  $\tilde{C}$  должен содержать точку  $z_0$ , в которой  $\operatorname{Re} f(z)$  достигает наибольшего значения на  $\tilde{C}$ . Покажем для этого случая, что  $f'(z_0) =$

0, то есть точка линии  $\operatorname{Im} f(z) = \operatorname{const}$ , в которой  $\operatorname{Re} f(z)$  достигает наибольшего значения, является точкой перевала.

Так и есть, ведь в точке  $z_0$ , которая является максимумом для  $\operatorname{Re} f(z)$  производная  $u = 0$  вдоль линии  $\tilde{C}$  должна быть равна 0, т. е.  $\frac{\partial}{\partial s} \operatorname{Re} f(z) = 0$ , а так как  $\operatorname{Im} f(z) = \operatorname{const}$  на  $\tilde{C}$ , то  $\frac{\partial}{\partial s} \operatorname{Im} f(z) \equiv 0$ , а значит и

$$f'(z_0) = \frac{\partial}{\partial s} \operatorname{Re} f(z) + i \frac{\partial}{\partial s} \operatorname{Im} f(z) = 0.$$

Для метода перевала к интегралу (2.1) путь интегрирования следует деформировать в путь  $\tilde{C}$ , проходящий через точку перевала  $z_0$  и в окрестности этой точки идущий вдоль линии наибольшего ската  $\operatorname{Im} f(z) = \operatorname{const}$  (рисунок 2.1).

Есть одно важное обстоятельство, обеспечивающее эффективность применения метода перевала: так как вдоль линии  $\tilde{C}$  имеем  $\arg e^{f(z)} = \operatorname{Im} f(z) = \operatorname{const}$ , то оценка интеграла (2.1) сводится к оценке интеграла от действительной функции, которая может быть проведена по методу Лапласа для интеграла вида (2.2). [11]

Именно это позволяет нам пользоваться полученными результатами теорем 1 и 2.

Рассмотрим случай, когда путь интегрирования можно деформировать в путь  $\tilde{C}$ , проходящий через точку перевала  $z_0$ , где  $f'(z_0) = 0$ ,  $f''(z_0) \neq 0$ , и в окрестности  $z_0$  совпадающий с линией наибольшего ската  $\operatorname{Im} f(z) = \operatorname{const}$ , причем на  $\tilde{C}$  вне этой окрестности  $\operatorname{Re} f(z) < \operatorname{Re} f(z_0) - h$  ( $h > 0$ ). Кроме того, предположим, что интеграл (2.1) абсолютно сходится для достаточно больших значений  $\lambda$ . Тогда образом, оценку интеграла можно провести на основании теоремы 1. Пусть  $z = z(t)$  будет уравнение контура  $\tilde{C}$ ; Тогда,

$$I(\lambda) = \int_C \phi(z) e^{\lambda f(z)} dz = e^{\lambda i \operatorname{Im} f[z(t)]} \int_a^b \phi[z(t)] e^{\lambda \operatorname{Re} f[z(t)]} z' dt. \quad (2.6)$$

и задача сводится к оценки интеграла вида 2.2 действительной области, разложение для которого уже было получено Лапласом, и имеет вид [12]:

$$I(\lambda) = \int_a^b \phi(t) e^{\lambda f(t)} dt \sim \frac{e^{f(a)}}{\lambda} \sum_{n=0}^{\infty} \frac{n! c_n}{\lambda^n}.$$

Выпишем первый член этого разложения. Обозначим  $\phi[z(t)]z' = \tilde{\phi}(t)$ ,  $\text{Re}f[z(t)] = \tilde{f}(t)$  и тогда по формуле (2.5) получаем:

$$\int_a^b \tilde{\phi}(t) e^{\lambda \tilde{f}(t)} dt \sim e^{\lambda \tilde{f}(t_0)} \sqrt{\frac{\pi}{\lambda}} \tilde{c}_0, \quad (2.7)$$

где  $\tilde{c}_0$  - свободный член в разложении функции  $\tilde{\phi}[\tilde{\psi}(\tau)]\tilde{\psi}'(\tau)$ .

Имеем:  $\tilde{\phi}(t_0) = \phi(z_0)z'(t_0)$ , и исходя из того, что  $f[z(t)] = \text{Re}f[z(t)] + i\text{Im}f[z(t)] = \tilde{f}(t) + \text{const}$  вдоль  $\tilde{C}$ , то

$$\tilde{f}''(t_0) = \frac{d^2}{dt^2} f[z(t)]|_{t=t_0} = f''(z_0)z'^2(t_0),$$

причем  $f'[z(t)]z''(t) = 0$  при  $t = t_0$ . Так как эта величина отрицательна, то представив  $z'(t_0) = ke^{i\theta}$ , можно записать ее в виде  $\tilde{f}'' = -|f''(z_0)|k^2$ . Получаем, что

$$\tilde{c}_0 = \tilde{\phi}(t_0) \sqrt{-\frac{2}{\tilde{f}''(z_0)}} = \phi(z_0)e^{i\theta} \sqrt{\frac{2}{|f''(z_0)|}}.$$

Подставим найденное значение в (2.7), а затем в (2.6), получаем искомую формулу

$$I(\lambda) \sim e^{\lambda f(z_0)} \sqrt{\frac{2\pi}{|f''(z_0)|}} \phi(z_0) e^{i\theta} \frac{1}{\sqrt{\lambda}}. \quad (2.8)$$

Как говорилось ранее, точка  $z_0$  - это точка, где  $\text{Re}f(z)$  достигает своего максимального значения. В то же время совершенно обычная ситуация - когда на искомом контуре  $\tilde{C}$  имеется несколько точек перевала, в которых значения  $\text{Re}f(z)$  находятся вблизи к наибольшему, то следует взять сумму выражений (2.8) по всем этим точками.

Тот случай, когда контур интегрирования заканчивается в точке перевала  $z_0$ , аналогичным образом приводится к теореме 2.

Итак, мы получили рабочую формулу, подставляя в которую составляющие наших искомых функций  $\phi(z)$  и  $f(z)$ , мы должны получать приближенные значения интеграла, когда  $\lambda \rightarrow \infty$

В интеграле  $\mathcal{M}(\epsilon, t)$  (1.11) роль большого параметра  $\lambda$  играет величина:

$$\lambda = \frac{1}{2\mathcal{T}} \int_{-\mathcal{T}/2}^{\mathcal{T}/2} A^2(\tau) d\tau. \quad (2.9)$$

### 3 Методы вычисления функции $\mathcal{M}(\epsilon, t)$

#### 3.1 Аналитическое решение

Получим формулу оценки интеграла (1.11) при помощи метода перевала.

Прежде всего заметим, что наряду с перевальными точками вклад в интеграл может давать значение  $t' = t$ , при котором следует учитывать “сингулярность” в подынтегральной функции  $\sim (t - t')^{-3/2}$ . Однако, вклад окрестности  $t' = t$  может рассматриваться независимо от вклада перевальных точек, ввиду того, что функция  $S(t', t)$  при  $t' \rightarrow t$  мала, и метод перевала становится несправедлив.

Функция  $S(t', t)$  при фиксированном  $t$  показана на рисунке 3.1. Из рисунка (3.1) видно, что большое значение функция  $S$  может принимать только в случае  $|t - t'| \gg 1$ .

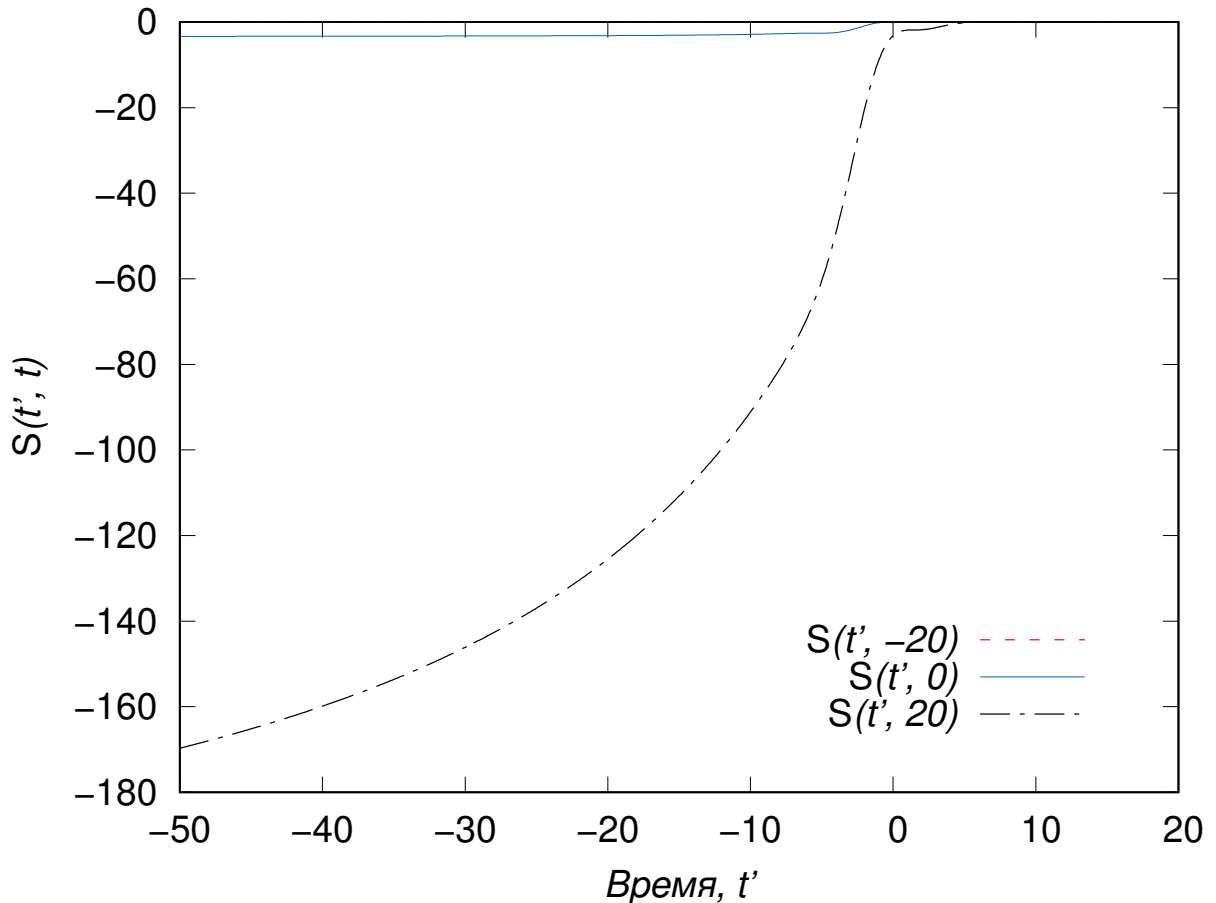


Рисунок 3.1 – Значение функции  $S(t', t)$  для  $t = -20, 0$  и  $20$

Рассмотрим случай  $t \rightarrow t'$  отдельно. В этом случае разложение функции  $S$  в ряд Тейлора принимает вид



$$S(t', t) \sim \frac{1}{24} \left( \frac{d}{dt} A(t) \right)^2 (t' - t)^3 + \frac{1}{24} \frac{d}{dt} A(t) \cdot \frac{d^2}{dt^2} A(t) (t' - t)^4 + \dots$$

В первом неисчезающем порядке, учитывая  $F(t) = -\dot{A}(t)$ :

$$S(t', t) \sim \frac{F^2(t)}{24} (t' - t)^3. \quad (3.1)$$

Подставляя (3.1) в интеграл (1.11) для  $\mathcal{M}$  и вводя замену  $t - t' = \tau$ , получаем

$$\begin{aligned} \mathcal{M}_{\tau=0} &= \frac{1}{\sqrt{2\pi i}} \int_0^\infty \frac{e^{i\epsilon\tau}}{\tau^{3/2}} \left( e^{-i\frac{2(t)}{24}\tau^3} - 1 \right) d\tau = \\ &= \frac{1}{\sqrt{2\pi i}} \int_0^\infty e^{i\epsilon\tau} \tau^{3/2} d\tau \cdot \left( -i \frac{F^2(t)}{24} \right) \end{aligned} \quad (3.2)$$

На рисунке 3.2 отображен график  $\mathcal{M}_{\tau=0}(\epsilon, t)$ . Полученная функция, как видно из графика, очень быстро затухает. Поскольку при недостаточном отдалении от нуля ( $t \approx 0$ ) даже для функции  $S(t', t)$  малый размер функции не позволяет использовать перевальную оценку, то вкладом  $\mathcal{M}_{\tau=0}(\epsilon, t)$  вблизи нуля, а также на остальном участке можно пренебречь (на интервале  $t = (10, \infty)$ :  $\mathcal{M}_{\tau=0}(\epsilon, t) \approx 0$ ).

Для учета перевальных точек рассмотрим интеграл:

$$\mathcal{M} \sim \int_{-\infty}^t \frac{1}{(t - t')^{3/2}} e^{i[\epsilon(t-t') + S(t, t')]} dt',$$

где опущено второе слагаемое -1 в (1.11), регуляризующее сингулярность  $(t - t')^{-3/2}$ . Как следует из оценки  $\mathcal{M}_{\tau=0}$  это слагаемое не существенно в области применимости метода перевала.

После замены  $t - t' = \tau$ , получаем

$$\mathcal{M} \sim \int_0^\infty \frac{1}{\tau^{3/2}} e^{i[\epsilon\tau + S(t, t-\tau)]} d\tau.$$

Если обозначить  $f(t, \tau) = \epsilon\tau + S(t, t - \tau)$  и  $\phi(\tau) = \frac{1}{\tau^{3/2}}$ , то получим формулу вида:

$$\mathcal{M} \sim \int_0^\infty \phi(\tau) e^{if(t, \tau)} d\tau,$$

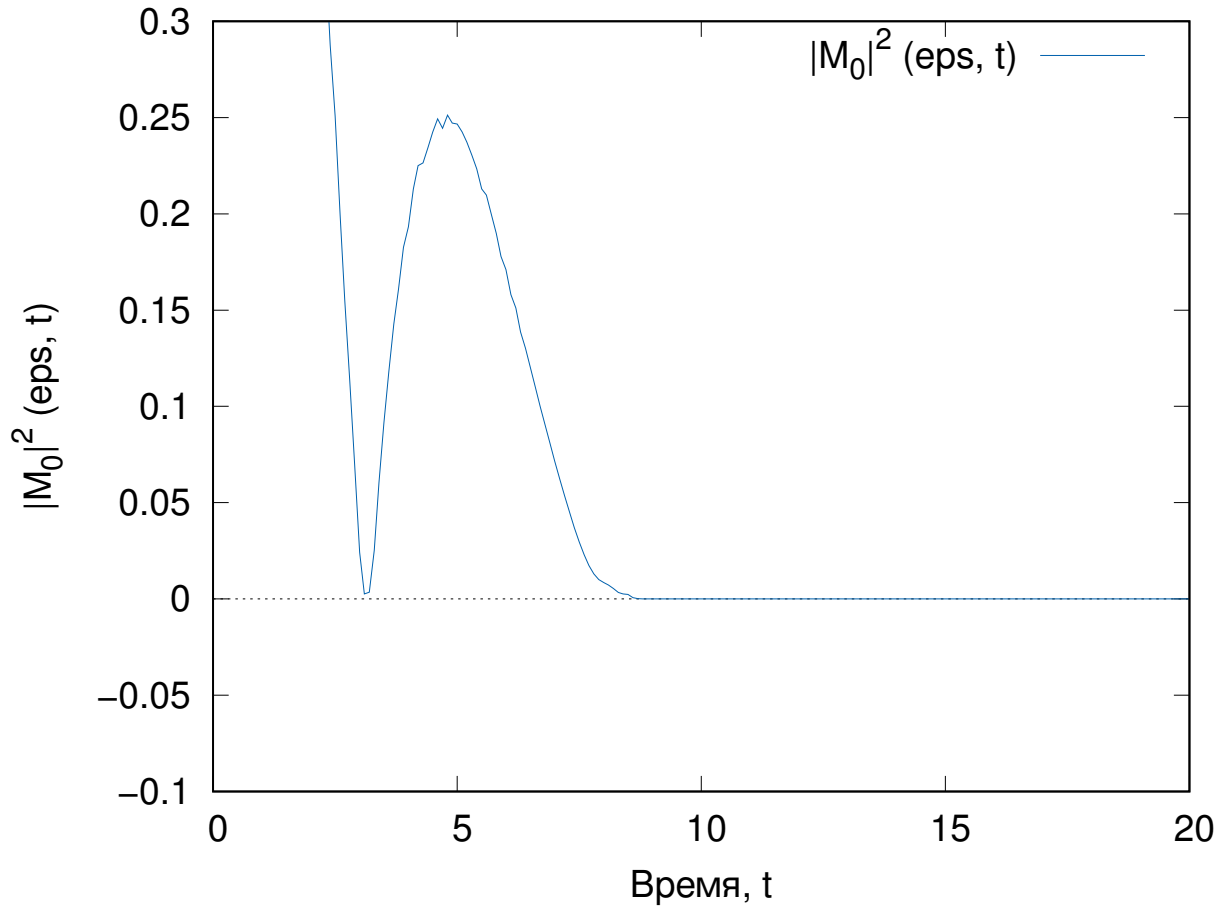


Рисунок 3.2 – Вклад интеграла  $\mathcal{M}_{\tau=0}(\epsilon, t)$

которая формально совпадает с рассмотренным в главе 2 интегралом (2.1).

В результате применения метода перевала, получаем:

$$\mathcal{M}(\epsilon, t) \simeq \sqrt{\frac{1}{2\pi i}} \sum_{\tau_0} e^{f(t, \tau_0)} \sqrt{-\frac{2}{\partial^2 f(t, \tau)/\partial \tau^2|_{\tau=\tau_0}}} \phi(t_0), \quad (3.3)$$

где  $\tau_0$  - корни уравнения  $\partial f(t, \tau)/\partial \tau = 0$ .

Найдем выражение для  $\partial^2 f(t, \tau)/\partial \tau^2$  и решим уравнение на стационарные точки  $\tau_0 \equiv t - t_0$ .

Дифференцируя  $f(t, \tau) = \epsilon\tau + S(t, t - \tau)$ , по  $\tau$  получим:

$$\begin{aligned} \frac{\partial f(t, \tau)}{\partial \tau} &= \epsilon + \frac{\partial S(t, t - \tau)}{\partial \tau}, \\ \frac{\partial f(t, \tau)}{\partial \tau} &= 0 \Rightarrow \frac{\partial S(t, t - \tau)}{\partial \tau} = -\epsilon. \end{aligned} \quad (3.4)$$

С введенной заменой  $S$  примет вид

$$S(t, t - \tau) = -\frac{1}{2} \int_{t-\tau}^t \alpha(\epsilon, t, t - \tau)^2 d\epsilon$$

Дифференцируя  $S(t, t - \tau)$  по  $\tau$ , получаем

$$\frac{\partial S(t, t - \tau)}{\partial \tau} = -\frac{1}{2} \alpha(t - \tau, t, t - \tau)^2.$$

Произведем обратную замену  $t_0 = t - \tau_0$  и с учетом уравнения (3.4), получаем явный вид уравнения на стационарные точки:

$$\alpha^2(t_0, t, t_0) = 2\epsilon. \quad (3.5)$$

Далее получим формулу для  $D \equiv \partial^2 f(t, \tau) / \partial \tau^2$ :

$$D = -\frac{1}{2} [\alpha(t - \tau, t, t - \tau)^2]' = -\alpha(t_0, t, t_0) \alpha(t_0, t, t_0)',$$

и с учетом явного вида  $\alpha$  (1.14):

$$D = \alpha(t_0, t, t_0) \left[ F(t_0) - \frac{1}{(t - t_0)^2} \int_{t_0}^t A(\tau) d\tau + \frac{A(t_0)}{t - t_0} \right].$$

Подставляя полученные соотношения в формулу (3.3), получаем:

$$\mathcal{M}(\epsilon, t) \simeq \sum_{t_0} \frac{e^{i\tilde{S}(t, t_0)}}{\sqrt{D}(t - t_0)^{3/2}}, \quad (3.6)$$

где  $\tilde{S} = \epsilon \tau_0 + S(t, t - \tau_0) = \epsilon(t - t_0) + S(t, t_0)$ .

Видно, что полученная формула включает в себя множество различных элементов, предполагающих численное интегрирование. Также перед ее вычислением необходимо решать уравнение (3.5) на стационарные точки для каждого  $t$ .

Прежде всего заметим, что в соответствии с уравнением (1.9) аргумент  $\epsilon$  рассматриваемой функции  $\mathcal{M}(\epsilon, t)$  может принимать любые вещественные значения. В случае отрицательных  $\epsilon$  решения уравнения (1.9) становятся комплексными, и требуют отдельного детального анализа.

В данной работе мы ограничимся случаем вещественных точек стационарной фазы ( $\text{Im} t_0 = 0$ ), и, соответственно, рассмотрим положительные зна-

чения  $\epsilon > 0$ . Из (3.5) имеем два уравнения:

$$\alpha(t', t, t') - \sqrt{2\epsilon} = 0,$$

$$\alpha(t', t, t') + \sqrt{2\epsilon} = 0,$$

объединяя все решения которых, получаем искомые седловые точки  $t' = t_0$ .

Построим график функции  $\alpha(t', t, t')$  на отрезке  $t' = [-20..20]$  при  $t = 0$  (рисунок 3.3): Видно, что корней несколько и они практически симметричны относительно ветвей оси  $Oy$ . Важно, что в суммировании участвуют только решения на промежутке  $t' = [-20, t]$ . На графике показано, что не для каждого  $\epsilon$  у нас найдутся решения в области действительных чисел. Максимальное значение  $\epsilon$  при данных условиях около 0.6.

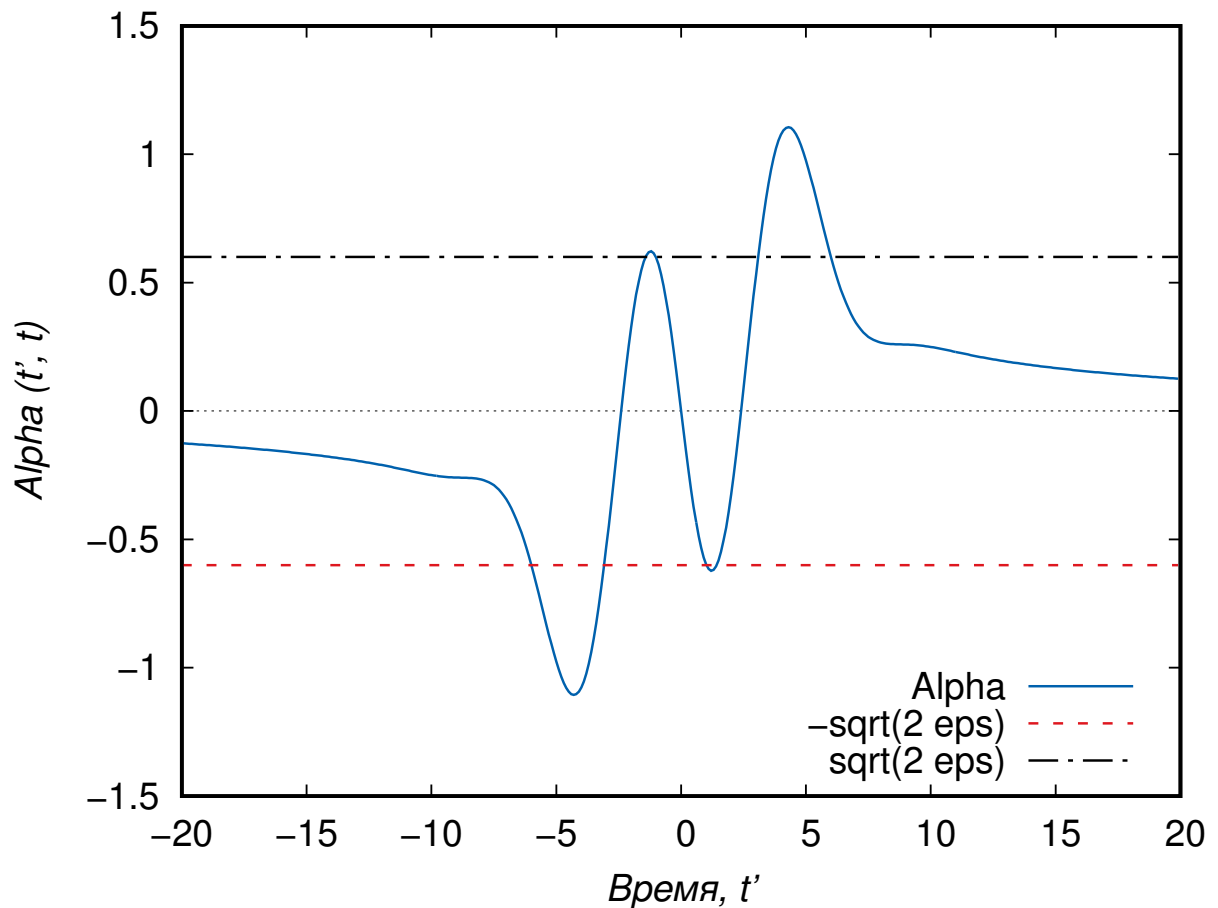


Рисунок 3.3 – Функция  $\alpha$  при  $t = 0$ .

Если корни близки к локальному экстремуму функции  $\alpha(t', t, t')$ , то возможен случай слияний корней, который является еще одним ограничением использования перевальных оценок. Чтобы этого избежать, можно получить оценку не для двух точек, а для одной, которая находится между ними на

равном удалении: равномерная аппроксимация исходного интеграла [11]. В этом случае асимптотическое разложение интеграла  $\mathcal{M}$  примет другой вид в связи с тем, что сделанное ранее предположение о равенстве нулю первой производной фазовой функции в подынтегральном выражении будет не верным. Оценка  $\mathcal{M}$  при равномерной аппроксимации выражается через функцию Эйри [13]. Поскольку явный вид ее объемный, приведен он не будет, но получение не вызывает трудностей.

На рисунке 3.4 представлен график  $\alpha(t', t, t')$  на отрезке  $t' = [-20..20]$  при  $t = 20$ . С ростом  $t$  корней становится меньше, и они дальше друг от друга расположены, чем на рисунке 3.3. Следует заметить, что для каждого значения  $\epsilon$  необходимо проверять существование решения уравнения на рассматриваемом временном интервале. В данной работе для нахождения корней уравнений используется метод Брента [14].

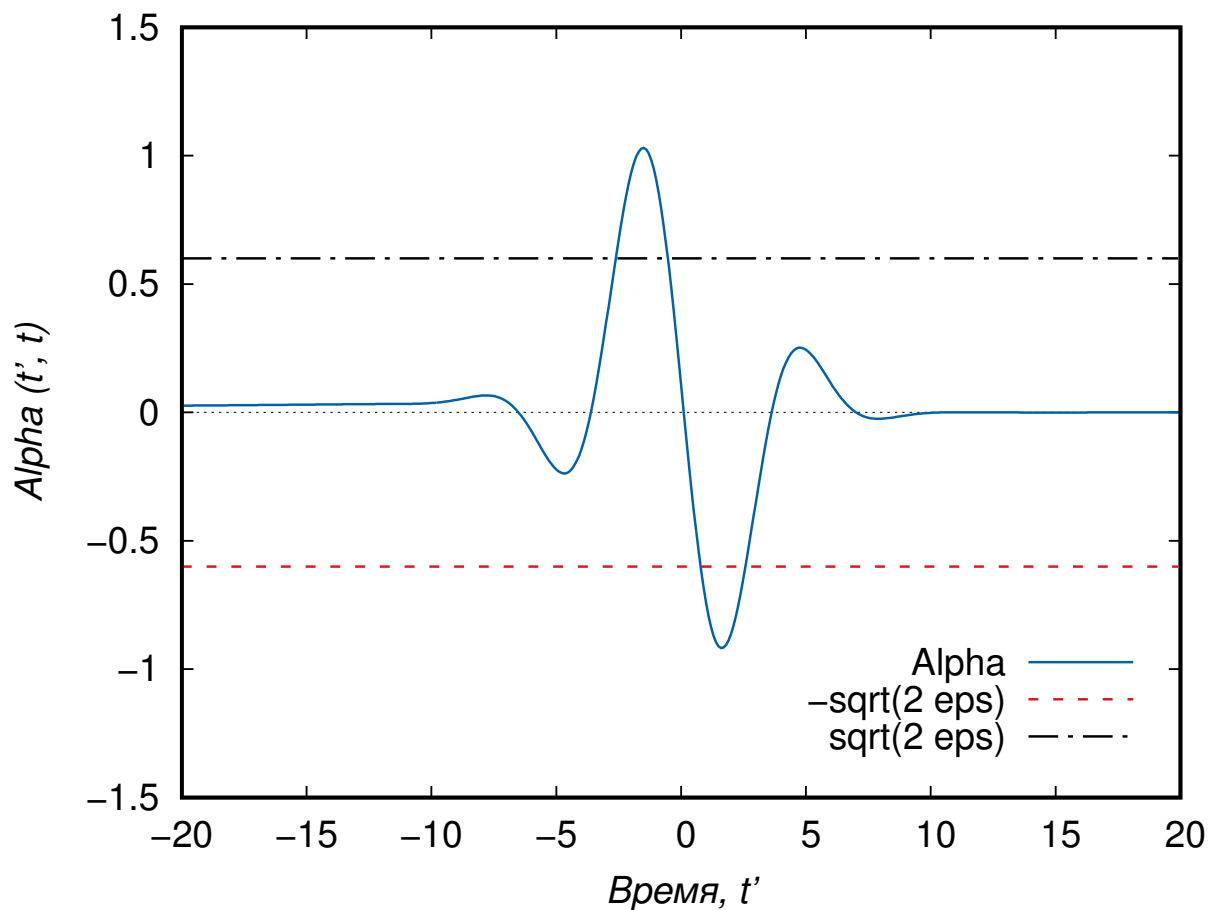


Рисунок 3.4 – Функция  $\alpha$  при  $t = 20$ .

В общем случае корни седлового уравнения могут быть комплексные. Тогда необходимо решать систему (при фиксированном  $t$ ):

$$\begin{cases} \operatorname{Re} \alpha(x + iy, t, x + iy) = 2\epsilon \\ \operatorname{Im} \alpha(x + iy, t, x + iy) = 0 \end{cases}$$

Здесь  $\epsilon$  - любое действительное число. В этом случае потребуется производить интегрирование в области комплексных чисел, что вызывает определенные трудности. Проблема также возникает в поиске множества корней. Основным методом решения уравнений в нескольких измерениях является метод градиентного спуска, то есть должны быть описаны также производные этих функций. Самым трудоемким в таком случае является именно поиск множества корней, так как возможна потеря некоторых из них во время перехода к следующей области поиска.

### 3.2 Численное решение

В данной главе обсуждается алгоритм вычисления выражения (1.11) непосредственным интегрированием несобственного интеграла.

Нижняя граница  $t_b$  интервалов интегрирования для интегралов от величин, определяемых функциями  $F(t)$  или  $A(t)$  с ограниченным носителем (см. выражение (1.12) и рисунок 1.6) определяется заданной точностью  $\delta$ , в соответствии с условием

$$\exp(-t_b^2/a^2) < \delta.$$

Например, интеграл (1.15) принимает вид:

$$A(t) = - \int_{t_b}^t F(\tau) d\tau.$$

Для вычисления этого интеграла воспользуемся гауссовскими квадратурами:

$$\int_{x_1}^{x_2} f(x) dx = \sum_{j=1}^N \omega_j f(x_j),$$

с весами  $\omega_j$  и узлами  $x_j$ , явный вид которых приведен в [13].

Для реализации вычисления интегралов, определяющих классическое действие  $S(t', t)$  [которое в свою очередь входит в подынтегральную функцию для искомой величины  $\mathcal{M}(\epsilon, t)$  (1.11)] необходимо адаптировать гауссовские методы интегрирования под переменные пределы интегрирования и требуемую точность. Для этих целей воспользуемся библиотекой подпрограмм GSL (GNU Scientific Library) [15].

GSL - это открытая библиотека, написанная на языке программирования C для численных вычислений в прикладной математике и науке.

#### Особенности GSL:

1. библиотека написана полностью для стандарта C (также применима для C++) и основана на использовании заголовочных файлов;
2. определяет новые типы, и структуры данных, у которых нет аналогов в Фортране или C;
3. использует порядок хранения данных на многомерных массивах, отличающийся от используемого Фортраном. Единственный способ, исполь-

зование его в Фортране, состоит в том, чтобы написать собственные подпрограммы на С для обеспечения необходимого соответствия между типами данных, структурами и соглашениями хранения их для двух языков.

4. Интерфейсный уровень не поставляется с библиотекой.

Самый главный плюс GSL – скорость вычислений, а также относительная экономия памяти и очистка неиспользуемой памяти. В ней реализовано огромное количество численных методов:

1. Базовые математические функции.
2. Комплексные числа.
3. Специальные функции.
4. Вектора и матрицы.
5. Комбинаторика.
6. Сортировка.
7. Линейная алгебра.
8. Быстрое преобразование Фурье.
9. Численно интегрирование (основанное на пакете QUADPACK).
10. Поиск корней уравнений и др.

Для нашей задачи важны две функции GSL:

*gsl\_integration\_qags* [16], позволяющая интегрировать функцию на отрезке с заданной точностью, и

*gsl\_root\_fsolver*, решающая уравнения методом Брента, не используя информацию о производной функции, задающей решаемое уравнение[16].

Максимальная достигнутая точность вычисления интеграла (1.15) с использованием средств GSL составила  $\sim 10^{-9}$ .

Для вычисления величины  $S(t', t)$  представим её в следующем виде:

$$S(t, t') = \frac{1}{2} \int_{t'}^t (A(\epsilon) - A_1(t', t))^2 d\epsilon, \quad (3.7)$$



где  $A_1(t', t) = (t' - t)^{-1} \int_{t'}^t A(\tau) d\tau$ . Относительно внешнего интеграла по  $\epsilon$   $A_1(t', t)$  остается постоянной, поэтому этот элемент можно посчитать только 1 раз для данных  $t', t$ . Этим преобразованием мы уменьшаем количество операций.

Основную численную сложность представляет расчет внешнего интеграла 1.11 по переменной  $t'$  для  $M(\epsilon, t)$ . Для его вычисления используем алгоритм на базе быстрого преобразования Фурье.

Преобразование Фурье переводит функцию времени (временной сигнал) в спектральное (частотное) распределение:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx. \quad (3.8)$$

При замене непрерывной функции ее отсчетами, вычисленными в дискретные моменты времени, соответствующее преобразование называется дискретным преобразованием Фурье (ДПФ). Для равномерной сетки размера  $N$  ДПФ выражается в виде обрезанного ряда Фурье следующего вида:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}, \quad (3.9)$$

Ширина спектра ДПФ является обратной величиной к длительности временного сигнала. Если исходная последовательность охватывает все ненулевые значения функции, ее ДПФ непрерывно и периодически.

Алгоритм быстрого преобразования Фурье (БПФ) вычисляет ДПФ последовательности  $X_k$  или обратное ДПФ. Алгоритм БПФ вычисляет такие преобразования, разлагая матрицу ДПФ на произведение разреженных сомножителей. В результате это приводит к уменьшению сложности вычислений ДПФ до  $\mathcal{O}(N \log N)$ .

Вычисление интеграла (1.11) или (заменяя  $t - t' \rightarrow \tau$ )

$$\mathcal{M}(\epsilon, t) = \frac{1}{\sqrt{2\pi i}} \int_0^{\infty} \frac{e^{i\epsilon\tau}}{\tau^{3/2}} (e^{iS(t, t-\tau)} - 1) d\tau, \quad (3.10)$$

выполним используя алгоритм БПФ.

Для этого воспользуемся программной библиотекой для дискретных преобразований Фурье FFTW [17]. Библиотека FFTW известна как открытая реализация самых быстрых алгоритмов преобразования Фурье (по регуляр-

ным сравнительным тестам). Одними из плюсов FFTW является адаптивный подбор алгоритма преобразования и возможность сохранения результатов работы библиотеки при многократном повторении ДПФ для последующего использования алгоритма.

С точки зрения практических вычислений, скорость FFTW является определяющим фактором. Возможность сохранения результатов компиляции программы не столь значительна, ввиду малого времени компиляции и слабой зависимости сложности скомпилированного кода программы от вычислительной сложности алгоритма. Основную сложность вычислений составляет не преобразование Фурье, а исходной получение последовательности, над которой выполняется преобразование.

Сравнивая (3.10) с (3.8), следует положить

$$f(\tau) = \frac{1}{\tau^{3/2}}(e^{iS(t,t-\tau)} - 1),$$

и

$$\hat{f}(\xi) = \int_0^\infty e^{i\tau\xi} f(\tau) d\tau.$$

Тогда

$$\mathcal{M}(\epsilon, t) = \frac{1}{\sqrt{2\pi i}} \hat{f}(\epsilon).$$

Учитывая, что FFTW реализует дискретное преобразование, значение  $\epsilon$  может не совпасть с одной из частот преобразования, попадая в интервал  $w_1 < \epsilon < w_2$ . Для устранения этой проблемы используется следующая линейная аппроксимация спектра:

$$\sqrt{2\pi i} \mathcal{M}(\epsilon, t) = \frac{(\epsilon - w_1)[\hat{f}(w_2) - \hat{f}(w_1)]}{w_2 - w_1} + \hat{f}(w_1). \quad (3.11)$$

Отметим также, что использование ДПФ для вычисления интеграла (1.11) позволяет получить за время одного преобразования полный спектр значений функции  $\mathcal{M}(\epsilon)$ .

Для проверки точности интегрирования в зависимости от объема дискретизации подынтегральной функции (или размера сетки, или, фактически, верхней границы интервала интегрирования в (3.10)), были произведены расчеты для разных значений  $N$  при фиксированном шаге дискретизации переменной  $\tau$ . На рисунке (3.5) изображены результаты численного интегриро-

вания с использованием FFTW для интервалов интегрирования разной длины (100 или 200). Как видно из рисунка, отличие результатов для различных верхних границ интегрирования достаточно велико. Численный анализ показал, что оптимальным интервалом интегрирования можно считать  $[0, 350]$ , при котором получается хорошее соотношение между точностью расчетов и затратами. Кроме того, из рисунка (3.5) видно наличие осцилляций у искомого интеграла, фазы которых зависят от размера интегрируемой области. Причиной тому является особенность ДПФ, а именно его периодичность. Численная ошибка, приводящая к осцилляциям результата, связана с величиной функции  $f(\tau)$  на верхней границе интегрирования (в идеальном случае  $f(\tau)$  должна обращаться в нуль:  $f(0) = f(\infty) = 0$ ).

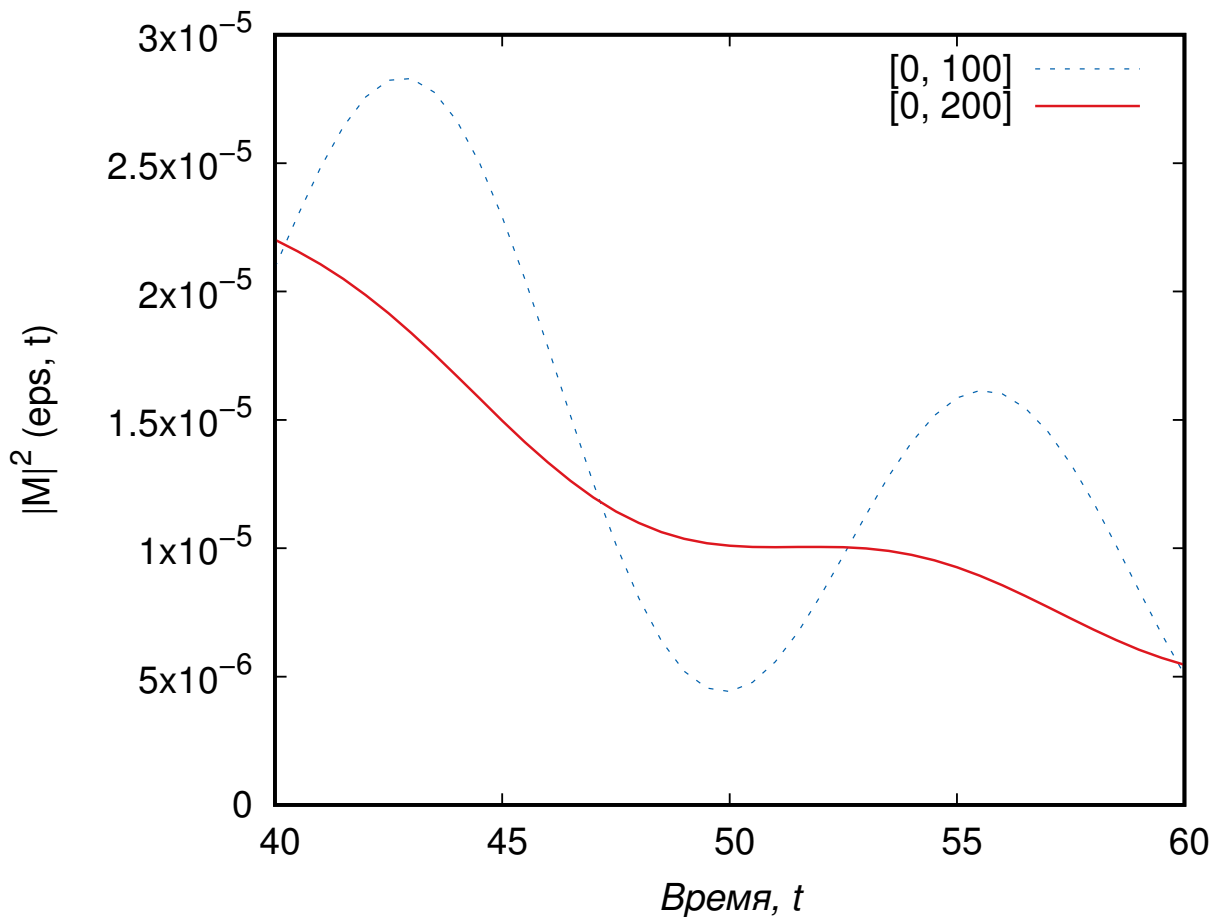


Рисунок 3.5 – Сравнение результатов интегрирования алгоритмом БПФ для разных отрезков интегрирования

Для фиксированного значения параметра  $\epsilon$  вместо использования преобразования Фурье, можно воспользоваться другими методами интегрирования. Для сравнения результатов используем интегрирование по методу прямоугольников (т.е. построим прямую интегральную сумму, являющуюся част-

ным случаем ДПФ для нулевой частоты) [18]. Сравнение результата такого интегрирования с результатом, полученным через FFTW показано на рисунке 3.6. Отличие представленных на рисунке результатов обусловлено неточностью линейной аппроксимации (3.11). Его можно улучшить, используя, например, сплайны второго порядка.

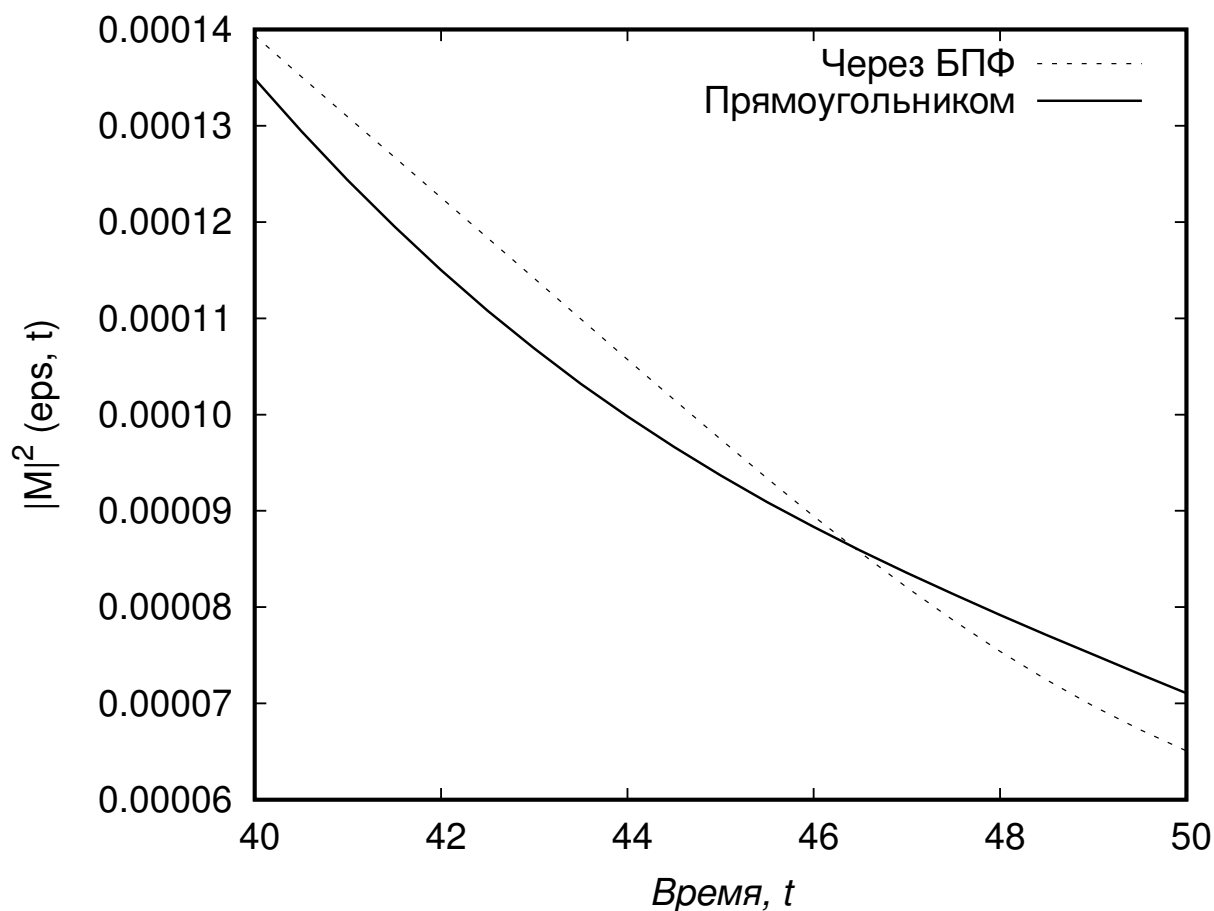


Рисунок 3.6 – Сравнение результатов для разных методов интегрирования

## 4 Анализ результатов

Проведем сравнение аналитического результата (3.6) для  $\mathcal{M}(\epsilon, t)$  с результатом численного интегрирования исходного интеграла в (1.11). Для этого произведем вычисления матричного элемента  $\mathcal{M}(\epsilon, t)$  на отрезке  $t = [0, 70]$ . Для больших значений  $t$  становятся существенными численные ошибки, ввиду малости значения  $\mathcal{M}(\epsilon, t)$ :  $\mathcal{M}(t > 70) \sim 10^{-7}$ . При определении параметров поля  $F(t)$  в (1.12), будем полагать ширину огибающей фиксированной ( $a^2 = 20$ ) и менять амплитуду  $F_0$  и несущую частоту  $\omega$ .

На рисунке 4.1 изображены решения седлового уравнения (3.5) для следующих параметров:  $\epsilon = 0.4$ ,  $F_0 = 2$ ,  $\omega = 0.8$ . Из графика видно, что для  $t > 5$  количество различных корней уравнения равно 4, и с ростом  $t$  новых корней не возникает, а значит рассматривать отрезок такой длины на появление новых корней не имеет смысла.

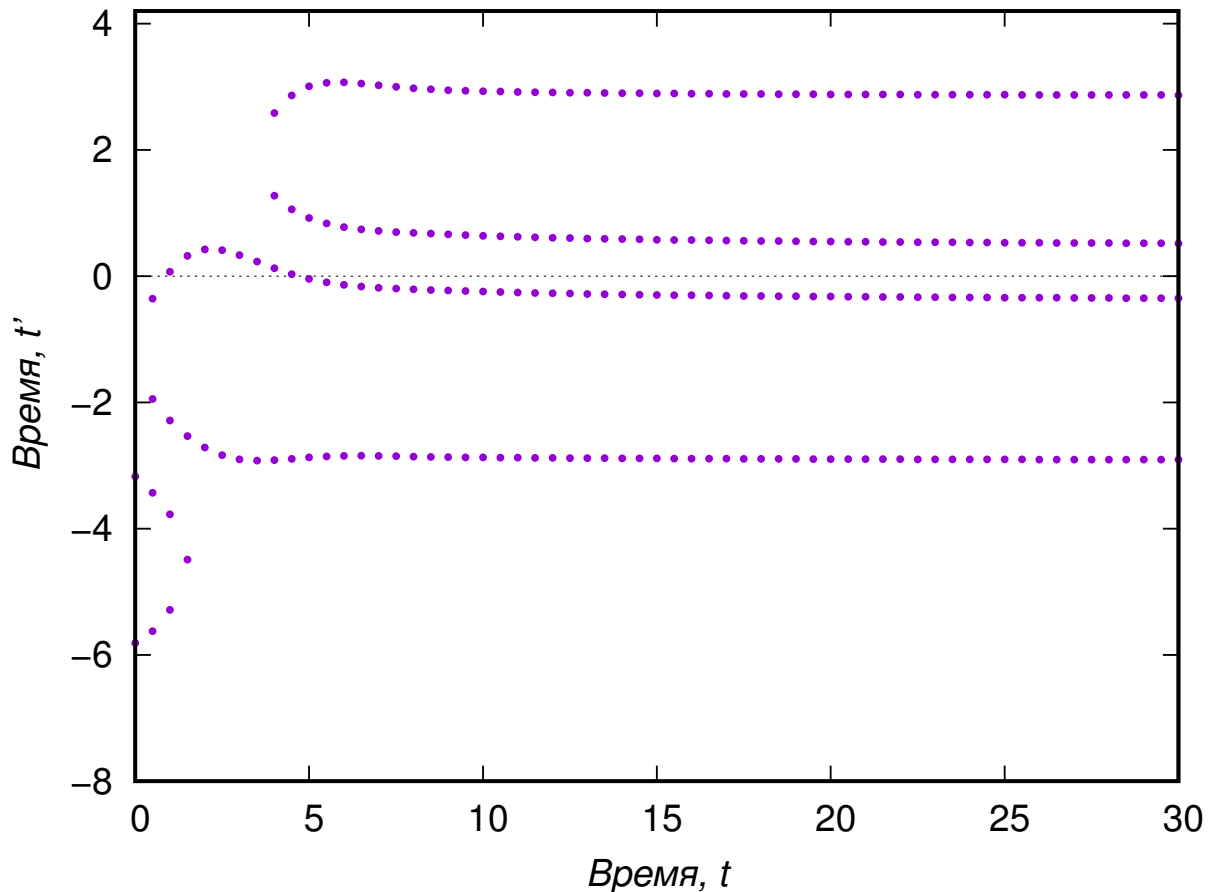


Рисунок 4.1 – Седловые точки для  $\epsilon = 0.4$ ,  $\omega = 0.8$ ,  $F_0 = 2$ .

Для выявления общей закономерности появления новых решений уравнения (3.5) с ростом  $t$  построим график для других значений  $\epsilon$  (рисунок 4.2).

Видно, что для  $\epsilon = 0.3$  корней больше, чем для больших значений  $\epsilon$ : максимум 6, и последний появляется при  $t \approx 11$ . При дальнейшем росте  $t$  количество решений не возрастает. Увеличение числа корней с уменьшением  $\epsilon$  подтверждается зависимостью  $\alpha(t', t, t')$  от  $t'$  на рисунке 3.4.

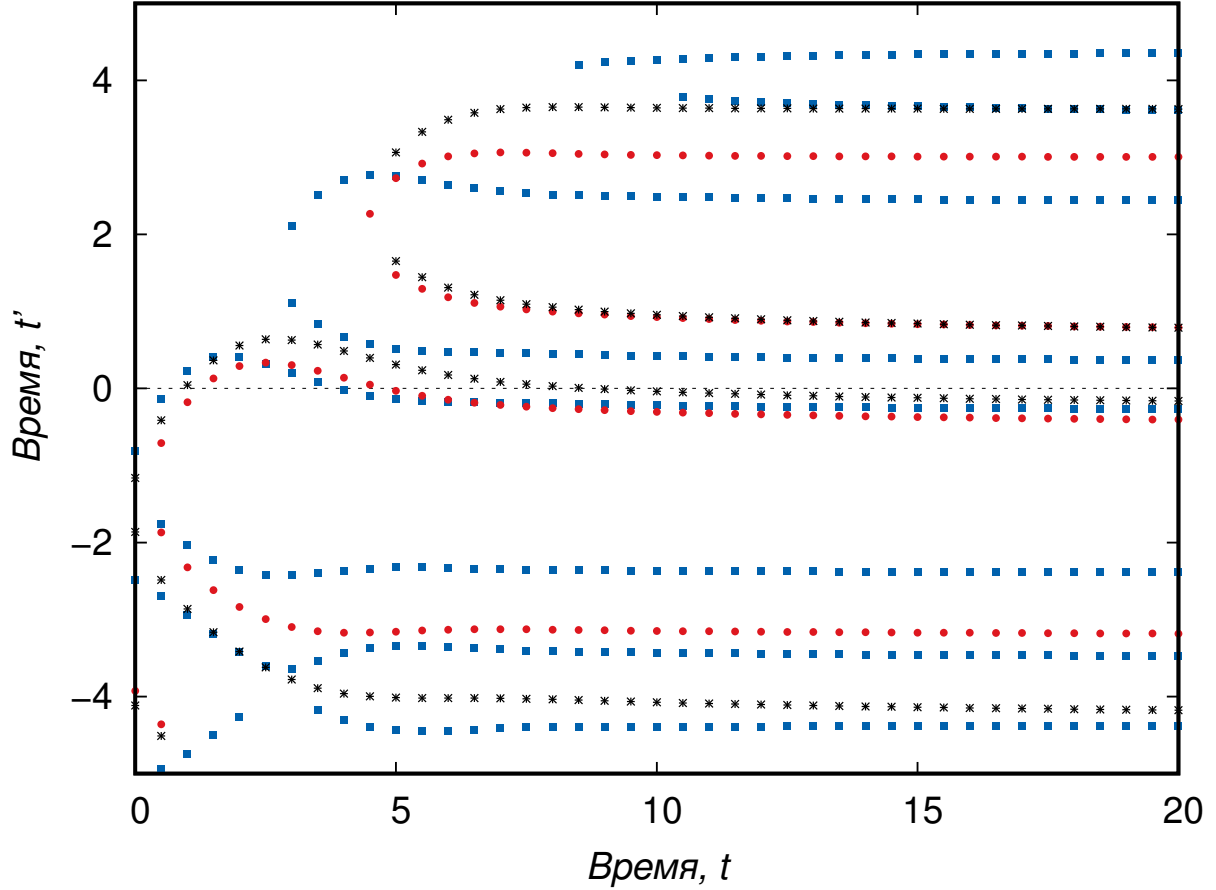


Рисунок 4.2 – Седловые точки для  $\epsilon = 0.3$  (квадраты),  $0.35$  (круги) и  $0.4$  (звездочки). Параметры функции  $F(t)$  те же, что на рисунке 4.1.

Сравнение аналитического и численного результатов для параметров  $\epsilon = 0.4$ ,  $F_0 = 2$ ,  $\omega = 0.8$  показано на рисунке 4.3. Из рисунка видно, что при малых  $t$  в аналитическом решении проявляются пики, отсутствующие в численном решении. Сравнение с рисунком 4.1 показывает, что положение этих пиков совпадает со значениями  $t$ , при которых возникают новые решения седлового уравнения. Причин возникновения пиков может быть две: первая связана с тем, что новые решения появляются при  $t = t'$ , что соответствует сингулярности в подынтегральной функции  $\sim (t - t')^{-3/2}$ ; вторая – при появлении решений парами в точке появления они сливаются, в результате чего производная  $D$  в выражении (3.6) для сливающихся седловых точек обращается в нуль, и метод перевала становится не применим.

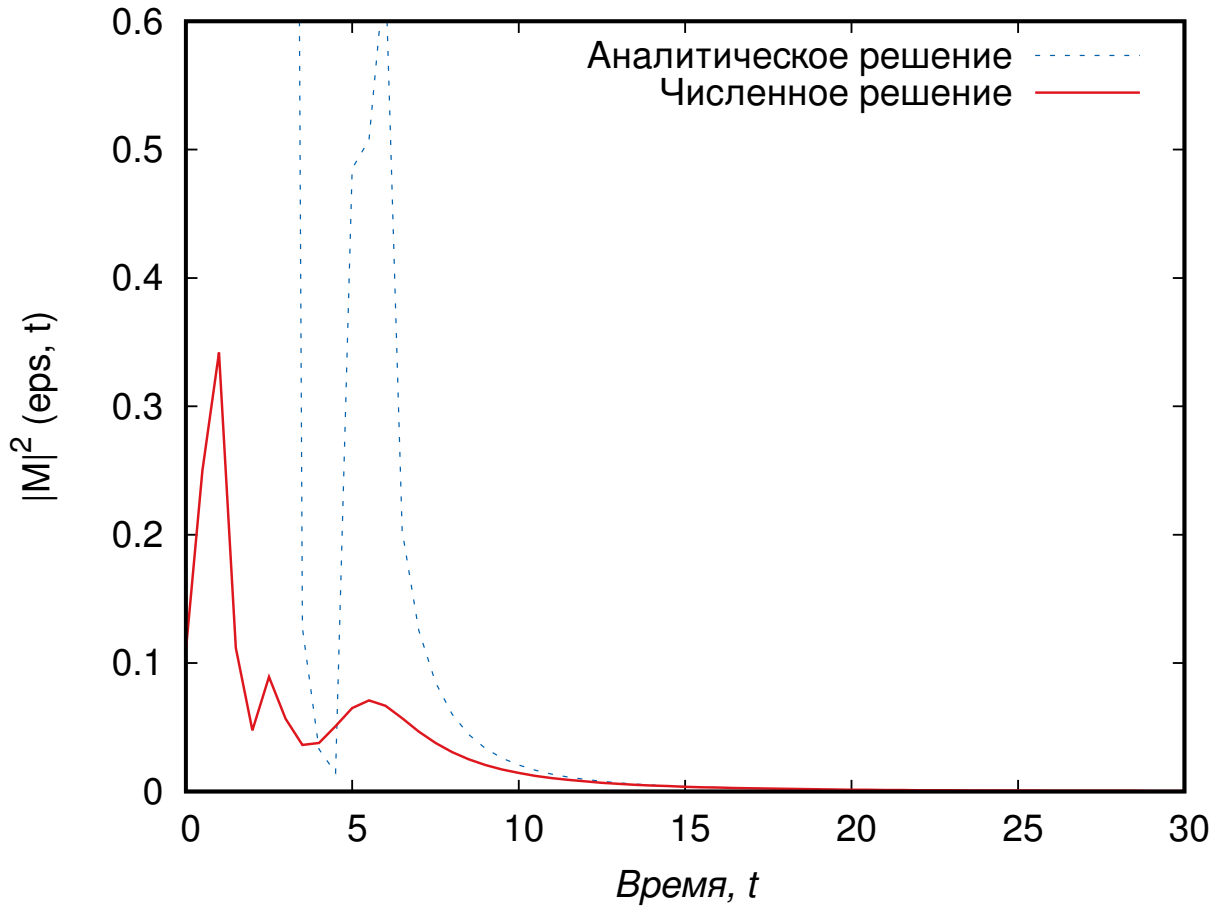


Рисунок 4.3 – Сравнение численного (сплошная линия) и аналитического (пунктирная линия) результатов на временном отрезке  $t \in [0, 30]$  для параметров как на рисунке 4.1

При удалении  $t$  от значения перевальной точки  $t' = t_0$  аналитическое решение показывает хорошее согласие с численным расчетом. Областью удовлетворительного согласия можно считать  $|t - t_0| > 2\pi/\omega$ , то есть при временном удалении на период поля от перевальной точки  $t_0$  поведение матричного элемента  $\mathcal{M}(\epsilon, t)$  хорошо описывается аналитической перевальной оценкой. Согласие аналитического и численного решений при больших значениях  $t$  показано на рисунке 4.4. Заметим, что расхождение результатов на рисунке 4.4 отчасти связано с неточностью численного решения.

Аналогичное согласие результатов наблюдается и для следующих параметров задачи:  $\epsilon = 0.35, F_0 = 2.2, \omega = 0.8$  (см. рисунки 4.5 и 4.6).

Изменим  $F_0$  и получим графики для других параметров задачи:  $\epsilon = 0.35, F_0 = 1, 6, \omega = 1$  (см. рисунки 4.7 и 4.8). Как видно из сравнения рисунков 4.4 и 4.8, с уменьшением  $F_0$  согласие аналитического и численного результатов становится хуже, поскольку величина  $\lambda \propto F_0^2/\omega^2$  [см. (2.9)] для

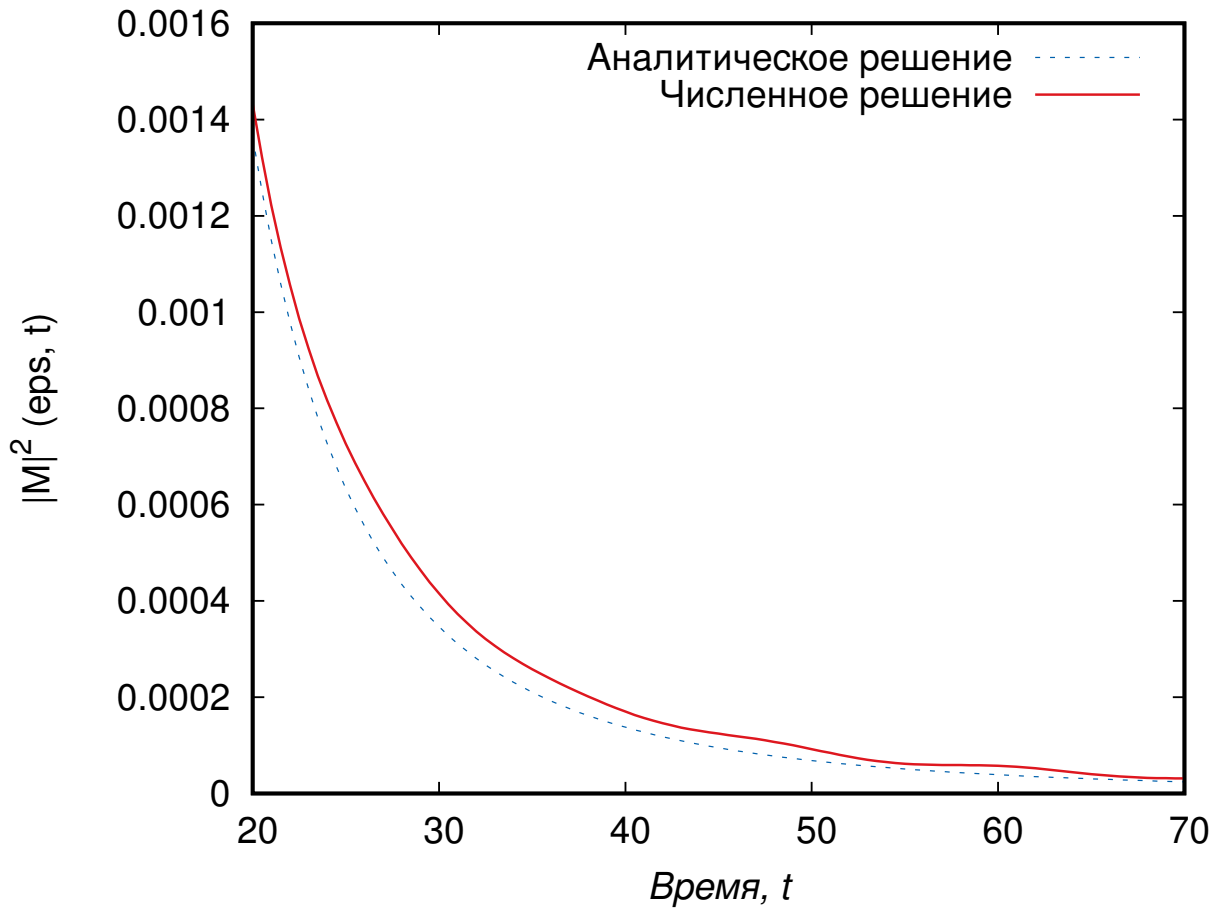


Рисунок 4.4 – То же, что на рисунке 4.3 для  $t \in [20, 70]$

этих условий перестает быть большим параметром, как требуют условия метода перевала (см. главу 2).

В заключение отметим, что с точки зрения времени вычислений при использовании аналитического результата и прямого численного интегрирования аналитическая формула дает выигрыш во времени на порядок величины. Этот результат можно улучшить, используя для поиска корней седлового уравнения более быстрые алгоритмы, чем метод Брента.



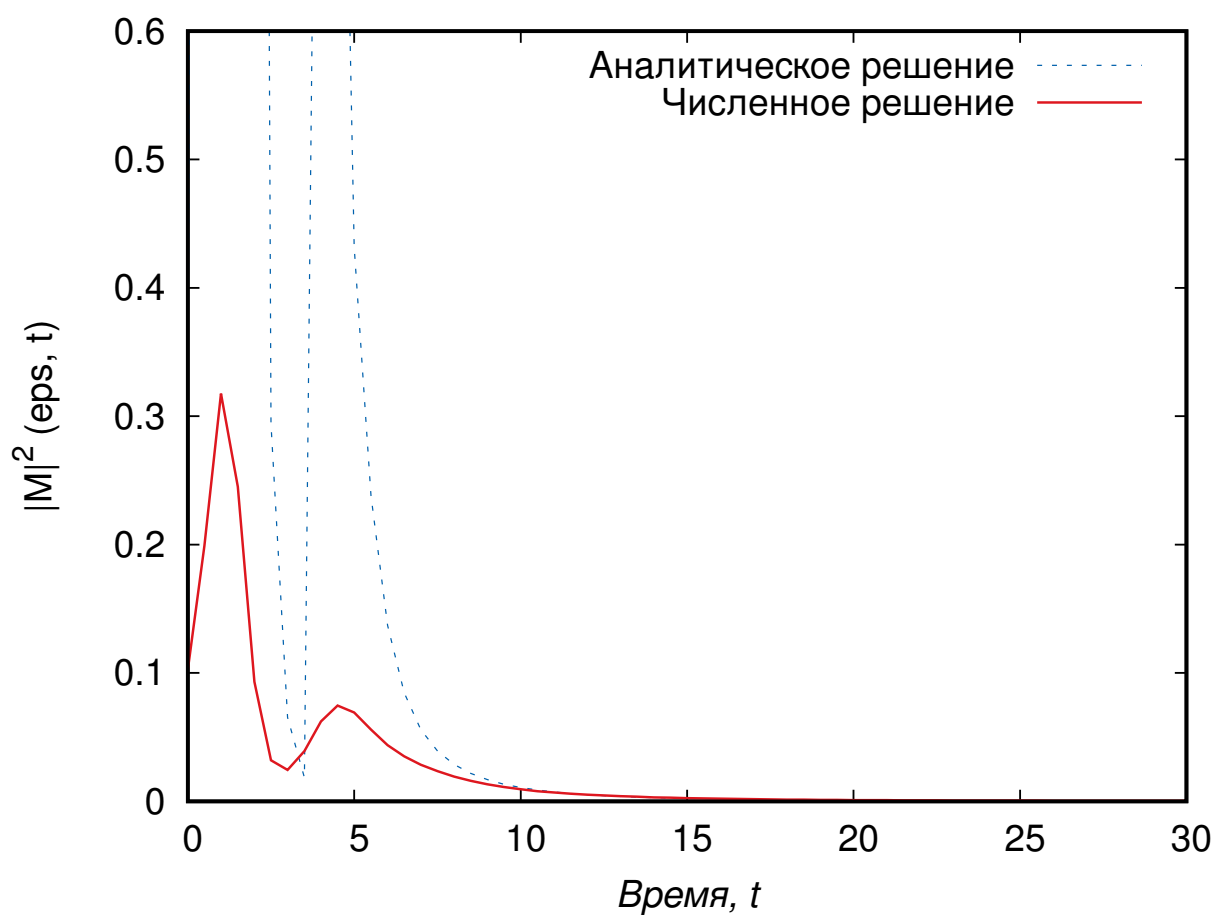


Рисунок 4.5 – Сравнение численного (сплошная линия) и аналитического (пунктирная линия) результатов на временном отрезке  $t \in [0, 30]$  для параметров  $\epsilon = 0.35$ ,  $F_0 = 2.2$ ,  $\omega = 0.8$

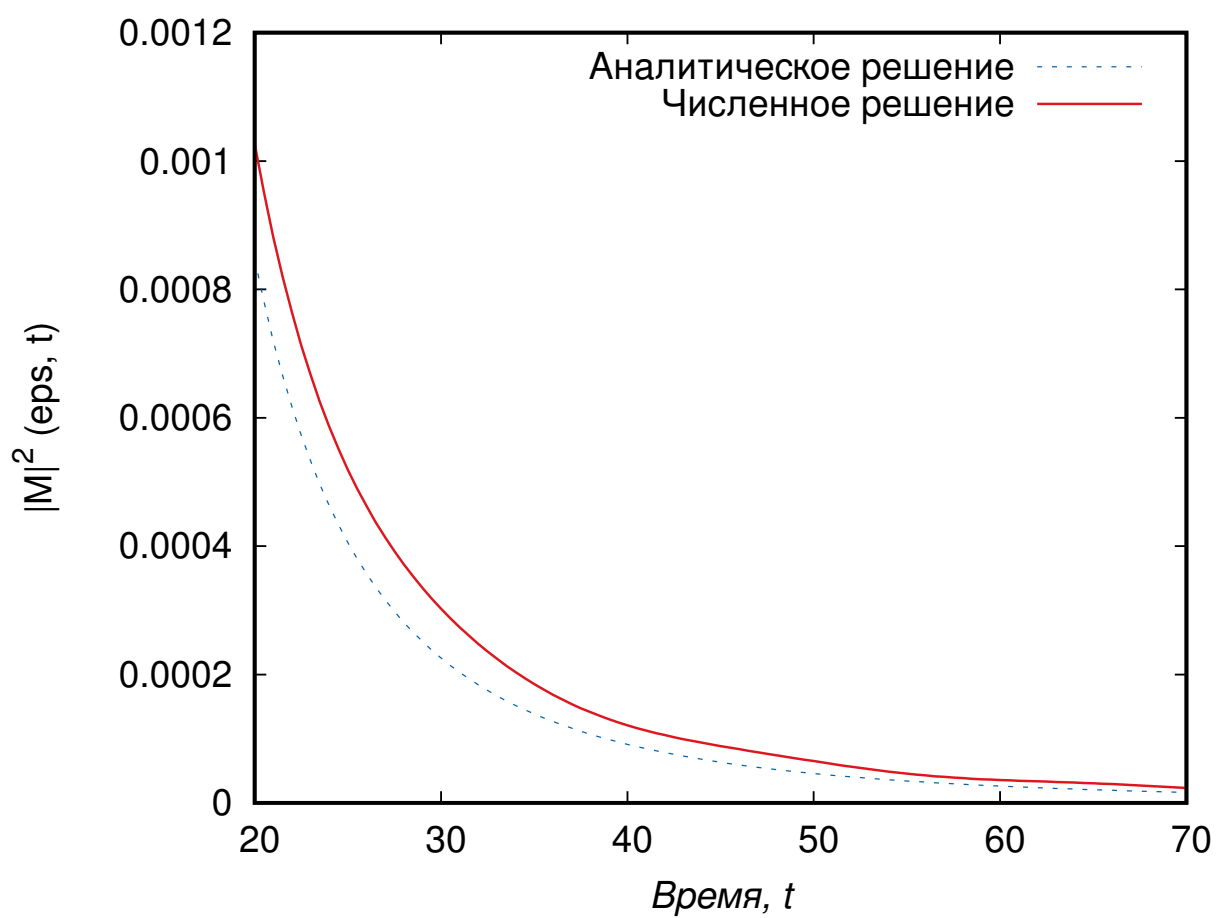


Рисунок 4.6 – То же, что на рисунке 4.5 для  $t \in [20, 70]$

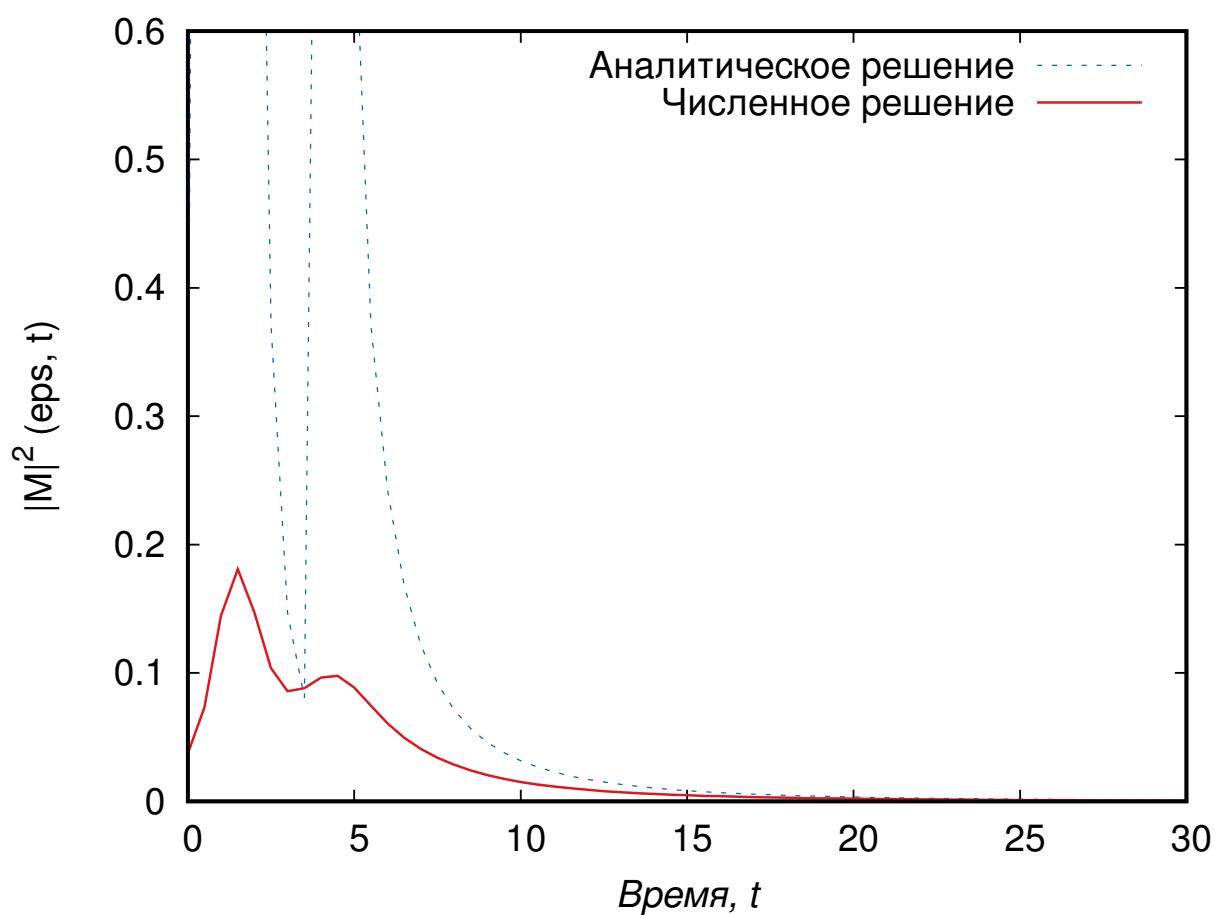


Рисунок 4.7 – Сравнение численного (сплошная линия) и аналитического (пунктирная линия) результатов на временном отрезке  $t \in [0, 30]$  для параметров  $\epsilon = 0.35$ ,  $F_0 = 1, 6$ ,  $\omega = 1$

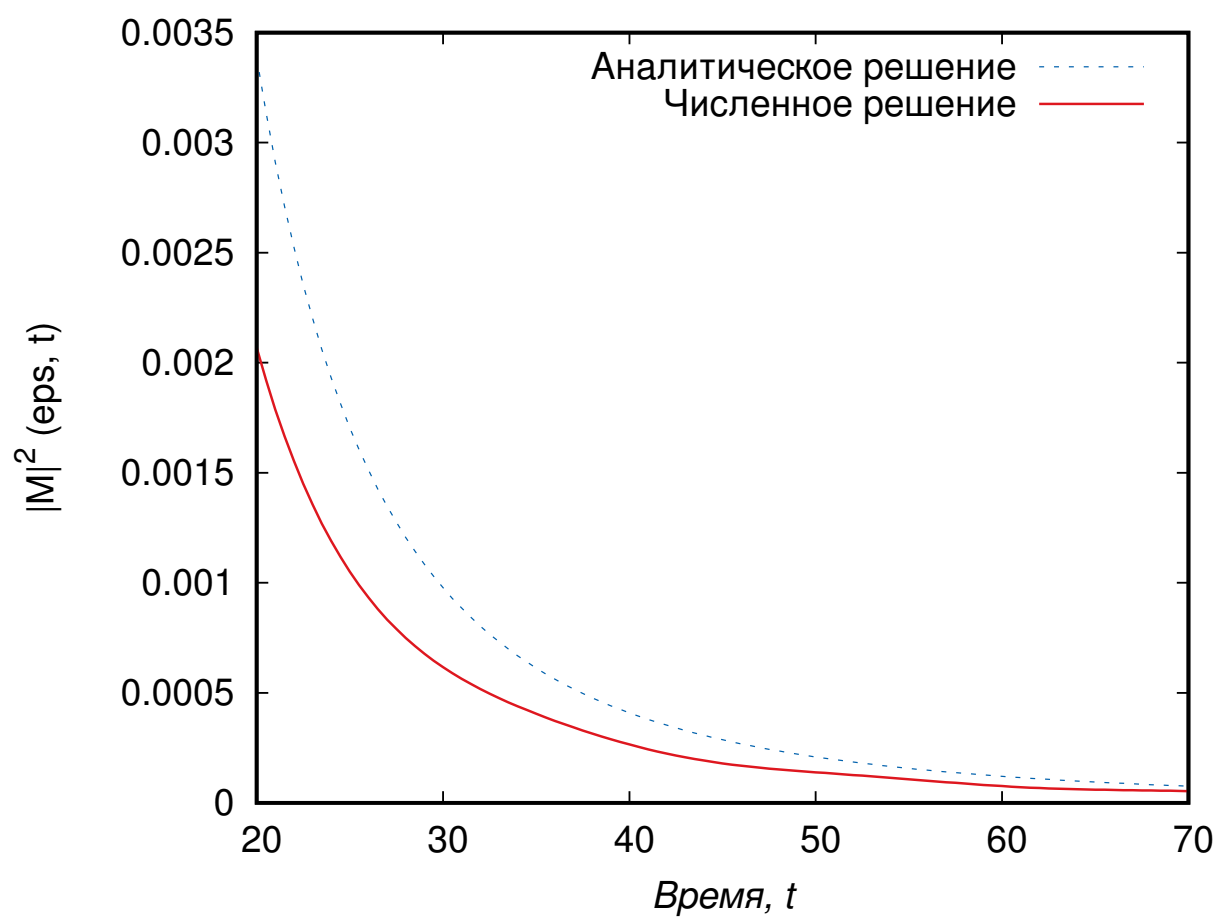


Рисунок 4.8 – То же, что на рисунке 4.7 для  $t \in [20, 70]$

## Заключение

В ходе выполнения данной бакалаврской работы рассматривалась задача моделирования взаимодействия атома с полем лазерного импульса. Были получены следующие результаты:

1. Получена аналитическая перевальная оценка (3.6) для матричного элемента  $\mathcal{M}(\epsilon, t)$ , определяющего поведение волновой функции атомного электрона, на малых расстояниях от атома. Результат представляет собой сумму по перевальным точкам слагаемых, каждое из которых выражается через классические физические величины.
2. Для получения численной оценки полученной аналитической формулы реализован алгоритм решения нелинейного трансцендентного уравнения, в общем случае имеющего произвольное число корней.
3. Разработан и реализован программный модуль на языке C++ для численного нахождения  $\mathcal{M}(\epsilon, t)$  в соответствии с исходным выражением (1.11). Метод численного решения включает в себя использование гауссовских квадратур и алгоритма быстрого преобразования Фурье.
4. Проведено численное сравнение результатов аналитического и численного решений. Выявлены границы применимости метода перевала для оценки  $\mathcal{M}(\epsilon, t)$ . Обнаружено хорошее согласие аналитического результата с численным для времен  $t$ , удаленных от значения перевальной точки  $t_0$  на величину  $|t - t_0| \gtrsim T$ , где  $T$  – период колебаний лазерного поля.

Обнаруженное отличие аналитического результата от численного в окрестности  $t \approx t_0$  может быть устранено уточнением аналитической оценки, путем более аккуратного учета вклада в оцениваемый интеграл граничной точки  $t = t'$  [см. (1.11) и (3.2)], а также использования равномерного приближения для исходного интеграла, учитывающего вклад сливающихся перевальных точек.

## Список использованных источников

- 1 Немухин, А. В. Многоуровневое моделирование сложных химических систем, или за что присуждена Нобелевская премия по химии 2013 года [Текст] / А. В. Немухин, Б. Л. Григоренко // Суперкомпьютерные технологии в науке, образовании и промышленности / Под ред. В. А. Садовничий, Г. И. Савин, Вл. В. Воеводин. — М. : Издательство Московского Государственного Университета, 2013. — С. 5–10.
- 2 Feynman's path-integral approach for intense-laser-atom interactions [Text] / P. Salieres, B. Carre, L. Le Deroff [et al.] // Science. — 2001. — Vol. 292. — P. 902–905.
- 3 Above threshold ionization: From classical features to quantum effects [Text] / W Becker, F Grasbon, R Kopold [et al.] // Adv. At., Mol., Opt. Phys. — 2002. — Vol. 48. — P. 35–98.
- 4 Milošević, D B. Scattering and reaction processes in powerfull laser fields [Text] / D B Milošević, F Ehlotzky // Adv. At., Mol., Opt. Phys. — 2003. — Vol. 49. — P. 373–532.
- 5 Birkeland, T. Parallel Pseudo-Spectral Methods for the Time-Dependent Schrödinger Equation [Text] / T. Birkeland, T. Sørøvik // Parallel Computing. Numerics, Applications, and Trends / Ed. by P. Zinterhof R. Trobec, M. Vajteršic. — Dordrecht, Heidelberg, London, New York : Springer, 2009. — P. 261–279.
- 6 Analytic description of elastic electron-atom scattering in an elliptically polarized laser field [Text] / A. V. Flegel, M. V. Frolov, N. L. Manakov [et al.] // Phys. Rev. A. — 2013. — Vol. 87. — P. 013404.
- 7 Analytic model for the description of above-threshold ionization by an intense short laser pulse [Text] / M. V. Frolov, D. V. Knyazeva, N. L. Manakov [et al.] // Phys. Rev. A. — 2014. — Vol. 89. — P. 063419.
- 8 Ландау, Л. Д. Квантовая механика [Текст] / Л. Д. Ландау, Е. М. Лифшиц. — М. : Наука, 1989. — С. 768.

- 9 Frolov, M. V. Effective-range theory for an electron in a short-range potential and a laser field [Text] / M. V. Frolov, N.L. Manakov, A.F. Starace // Phys. Rev. A. — 2008. — Vol. 78. — 063418.
- 10 Араманович, И. Г. Уравнения математической физики [Текст] / И. Г. Араманович, В. И. Левин. — М. : Наука, 1969. — С. 288.
- 11 Федорюк, М. В. Метод перевала [Текст] / М. В. Федорюк. — М. : Главная редакция физико-математической литературы изд-ва «Наука», 1977.
- 12 Wong, R. Asymptotic Approximations of Integrals [Text] / R. Wong. — NY. : SIAM, 2001. — P. 543.
- 13 Абрамовиц, М. Справочник по специальным функциям [Текст] / М. Абрамовиц, И. Стиган. — М. : Наука, 1979.
- 14 Тарасевич, Ю. Ю. Численные методы на Mathcad'е [Текст] / Ю. Ю. Тарасевич. — Астрахань : Астраханский гос. пед. ун-т, 2000. — С. 70.
- 15 Galassi, Mark. GSL - GNU Scientific Library [Text]. — 2016. — [Сайт] URL: <https://www.gnu.org/software/gsl/> (дата обращения 03.01.17).
- 16 Galassi, Mark. GNU Reference Manual [Text]. — 2016. — [Сайт] URL: [https://www.gnu.org/software/gsl/manual/html\\_node/](https://www.gnu.org/software/gsl/manual/html_node/) (дата обращения 03.01.17).
- 17 Frigo, Matteo. FFTW - Home Page [Text]. — 2017. — [Сайт] URL: <http://www.fftw.org/index.html> (дата обращения 20.12.16).
- 18 Numerical Recipes in C [Text] / William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. — NY : Press Syndicate of the University of Cambridge, 2002. — P. 949.

## Приложение А

### Листинг программы

В данном приложении приведен текст программы, реализующей алгоритмы, описанные в данной работе.

```
1  #include <math.h>
2  #include <stdio.h>
3  #include <malloc.h>
4  #include <float.h>
5  #include <complex.h>
6  #include <fftw3.h>
7  #include <gsl/gsl_integration.h>
8  #include <gsl/gsl_math.h>
9  #include <gsl/gsl_roots.h>
10 #include <gsl/gsl_errno.h>
11
12 #define TBEGIN (50.0)
13 #define TEND (70.0)
14 #define TSTEP (0.5)
15 #define EPS 0.35
16 #define H 1.054572e-27
17 #define PIM4 0.7511255444649425
18 #define PI 3.14159265359
19 #define OMEGA 1
20 #define F0 1.6
21 #define ALPHA 4.472135955
22 #define NPOINTS (1024.0)
23
24 double f(double x, void * params);
25 double A_evaluate(double t, void * params);
26 double A_integrate(double t1, double t);
27 double S_integrate_new(double tau, double t);
28 double S_integrate_func_new(double eps, void * params);
29 double Alpha2(double t1, double t);
30 double GetEvalSolveWithFFTW(double t);
31 double GetEvalSolveM_0(double t);
32 double GetEvalSolveWithoutFFTW(double t);
33 double GetAnalyticSolve(double t, int countOfRoots, double* roots);
34 int FindRoots(double t, double* roots);
35
36 int main()
37 {
38     FILE *f;
39     f = fopen("analytical.txt", "w");
40     FILE *f1;
41     f1 = fopen("numerical.txt", "w");
42     FILE *f2;
```



```

43  f2 = fopen("roots.txt", "w");
44  double tBegin = TBEGIN, tEnd = TEND, tStep = TSTEP;
45  int steps = (int)(tEnd-tBegin)/tStep + 1;
46
47  for(int i = 0; i<steps; i++)
48  {
49      double t = tBegin + i*tStep;
50      printf("T = %f\n", t);
51      double roots[20];
52      int countOfRoots = FindRoots( tBegin + i*tStep , roots );
53
54      for(int j = 0; j<countOfRoots; j++)
55      {
56          fprintf(f2, "%3.10f %3.10f\n", t, roots[j]);
57      }
58
59      double m1 = GetAnalyticSolve( t , countOfRoots , roots );
60      double m2 = GetEvalSolveWithoutFFTW( t );
61      printf("M_analytic(eps , t) = %2.15f\n", m1);
62      printf("M_numerical(eps , t) = %2.15f\n", m2);
63      fprintf(f, "%2.15f %2.15f\n", t , m1);
64      fprintf(f1, "%2.15f %2.15f\n", t, m2);
65  }
66
67  fclose(f);
68  fclose(f1);
69  fclose(f2);
70  return 0;
71 }
72
73
74 //Структура для численного решения уравнения
75 struct quadratic_params
76 {
77     double t;
78 };
79
80 struct S_integrate_params
81 {
82     double Aint;
83 };
84
85 //Структура для численного решения уравнения
86 double quadratic (double t0, void *params)
87 {
88     struct quadratic_params *p
89     = (struct quadratic_params *) params;
90
91     double t = p->t;

```

```

92
93     return Alpha2(t0, t) - sqrt(2*EPS);
94 }
95
96 double quadratic2 (double t0, void *params)
97 {
98     struct quadratic_params *p
99     = (struct quadratic_params *) params;
100
101     double t = p->t;
102
103     return Alpha2(t0, t) + sqrt(2*EPS);
104 }
105
106 int FindRoots(double t, double* roots)
107 {
108     gsl_function AlphaFunc;
109     const gsl_root_fsolver_type *T;
110     gsl_root_fsolver *s;
111     struct quadratic_params params = { t };
112     AlphaFunc.function = &quadratic;
113     AlphaFunc.params = &params;
114     T = gsl_root_fsolver_brent;
115     s = gsl_root_fsolver_alloc (T);
116     double h = 0.5;
117     double t1_hi = t, t1_lo = t - h;
118     int countOfRoots = 0;
119     while( t1_lo > -10 )
120     {
121         if ((quadratic(t1_lo, &params)) * (quadratic(t1_hi, &params)) < 0)
122         {
123             gsl_root_fsolver_set (s, &AlphaFunc, t1_lo, t1_hi);
124             int iter = 0;
125             int status = 0;
126             do
127             {
128                 iter++;
129                 status = gsl_root_fsolver_iterate (s);
130                 roots[countOfRoots] = gsl_root_fsolver_root (s);
131                 t1_lo = gsl_root_fsolver_x_lower (s);
132                 t1_hi = gsl_root_fsolver_x_upper (s);
133                 status = gsl_root_test_interval (t1_lo, t1_hi,
134                                                 0, 0.001);
135             }
136             while (status == GSL_CONTINUE && iter < 1000);
137             countOfRoots++;
138             t1_hi = t1_lo;
139             t1_lo -= h;
140         }

```

```

141     t1_hi = t1_lo;
142     t1_lo -= h;
143 }
144
145
146 AlphaFunc.function = &quadratic2;
147 AlphaFunc.params = &params;
148 h = 0.5;
149 t1_hi = t, t1_lo = t - h;
150 while( t1_lo > -10 )
151 {
152     if ((quadratic2(t1_lo, &params)) * (quadratic2(t1_hi, &params)) < 0)
153     {
154         gsl_root_fsolver_set (s, &AlphaFunc, t1_lo, t1_hi);
155         int iter = 0;
156         int status = 0;
157         do
158         {
159             iter++;
160             status = gsl_root_fsolver_iterate (s);
161             roots[countOfRoots] = gsl_root_fsolver_root (s);
162             t1_lo = gsl_root_fsolver_x_lower (s);
163             t1_hi = gsl_root_fsolver_x_upper (s);
164             status = gsl_root_test_interval (t1_lo, t1_hi,
165                                             0, 0.001);
166         }
167         while (status == GSL_CONTINUE && iter < 1000);
168         countOfRoots++;
169         t1_hi = t1_lo;
170         t1_lo -= h;
171     }
172     t1_hi = t1_lo;
173     t1_lo -= h;
174 }
175 gsl_root_fsolver_free (s);
176 return countOfRoots;
177 }
178
179 double GetAnalyticSolve(double t, int countOfRoots, double* roots)
180 {
181     complex sum = 0;
182
183     int a = 0;
184
185     for(int i = 0; i < countOfRoots; i++)
186     {
187         double S_1 = S_integrate_new(roots[i], t) + EPS*(t-roots[i]);
188         double d = Alpha2(roots[i], t) * (( f(roots[i], 0) ) -
189             ( 1/( (roots[i] - t)*(roots[i] - t) ) ) *

```

```

190     A_integrate(roots[i], t) + A_evaluate(roots[i], 0) / (t - roots[i]));
191     sum += cexp(I * S_1)/(csqrt(d) * cpow( t - roots[i] , 3.0/2.0));
192 }
193 sum *= csqrt(2);
194 return creal(sum)*creal(sum) + cimag(sum)*cimag(sum);
195 }
196
197 double f (double x, void * params)
198 {
199     return -F0*exp(-(x*x)/(ALPHA*ALPHA))*cos(OMEGA*x);
200 }
201
202 //A какинтегралот f
203 double A_evaluate(double t, void * params)
204 {
205     double t1 = -50; // -\infty
206     double result;
207     double error;
208     double alpha = 1.0;
209     gsl_function F;
210     F.function = &f;
211     F.params = &alpha;
212     gsl_integration_workspace * w = gsl_integration_workspace_alloc (1000000);
213     gsl_integration_qags (&F, t1, t, 1e-7, 0, 1000000, w, &result, &error);
214     gsl_integration_workspace_free (w);
215     return result;
216 }
217
218 //A какинтегралот A_eval
219 double A_integrate(double t1, double t)
220 {
221     double result;
222     double error;
223     double alpha = 1.0;
224     gsl_function F;
225     F.function = &A_evaluate;
226     F.params = &alpha;
227     gsl_integration_workspace * a = gsl_integration_workspace_alloc (10000);
228     gsl_integration_qags (&F, t1, t, 1e-7, 0, 10000, a, &result, &error);
229     gsl_integration_workspace_free (a);
230     return result;
231 }
232
233 //Под интегралом функция S_integrate_func
234 double S_integrate_new(double tau, double t)
235 {
236     if (tau >= t)
237     {
238         return 0;

```

```

239 }
240 double Aint = A_integrate(tau , t)/(t-tau);
241 struct S_integrate_params params = { Aint };
242 double result , error;
243 gsl_function F;
244 F.function = &S_integrate_func_new;
245 F.params = &params;
246 gsl_integration_workspace * b = gsl_integration_workspace_alloc (10000);
247 gsl_integration_qags (&F, tau , t, 1e-7, 0, 10000, b, &result , &error);
248 gsl_integration_workspace_free (b);
249 return -result/2.0;
250 }
251
252 double S_integrate_func_new(double eps , void * params)
253 {
254     struct S_integrate_params *p
255     = (struct S_integrate_params *) params;
256
257     double Aint = p->Aint;
258     double temp = A_evaluate(eps , 0) - Aint;
259     return temp*temp;
260 }
261
262 //Функция по которой решается уравнение
263 double Alpha2(double t1 , double t)
264 {
265     return A_evaluate(t1 , 0) - A_integrate(t1 , t) / (t - t1);
266 }
267
268 double S_integrate_func(double eps , void * params)
269 {
270     double temp = A_evaluate(eps , 0);
271     return temp*temp;
272 }
273
274 double GetEvalSolveWithoutFFTW(double t)
275 {
276     double tempEnd = 350;
277     double step = tempEnd / NPOINTS;
278     double tempPower;
279     int i = 0;
280     complex sum = 0;
281
282     for(int i = 1; i < (int)NPOINTS; i++)
283     {
284         sum += cexp(I * EPS * i*step)
285             / cpow(i*step , 3.0/2.0)
286             * (cexp(I * S_integrate_new(t-i*step , t)) - 1);
287     }

```

```

288
289     sum *= step;
290     sum *= 1 / csqrt(2 * PI * I);
291     double result = creal(sum) * creal(sum) + cimag(sum) * cimag(sum);
292     return result;
293 }
294
295 double GetEvalSolveWithFFTW(double t)
296 {
297     fftw_complex *in, *out;
298     fftw_plan p;
299
300     in = (fftw_complex*) fftw_malloc(sizeof(fftw_complex) * NPOINTS);
301     out = (fftw_complex*) fftw_malloc(sizeof(fftw_complex) * NPOINTS);
302
303     double tempEnd = 200;
304     double step = tempEnd / NPOINTS;
305     double tempPower;
306     int i = 0;
307
308     for(double tempT = 0; tempT < tempEnd; tempT += step)
309     {
310         if(!tempT)
311         {
312             in[i] = 0;
313             i++;
314             continue;
315         }
316         in[i] = cexp(I * EPS * tempT)
317             / cpow(tempT, 3.0/2.0)
318             * (cexp(I * S_integrate_new(t - tempT, t)) - 1);
319         i++;
320
321     }
322
323     p = fftw_plan_dft_1d(NPOINTS, in, out, FFTW_FORWARD, FFTW_ESTIMATE);
324     fftw_execute(p);
325     out[0] *= 1.0 / NPOINTS;
326     out[0] *= tempEnd;
327     out[0] *= 1 / csqrt(2 * PI * I);
328     double result = creal(out[0]) * creal(out[0]) + cimag(out[0]) * cimag(out[0]);
329     free(out);
330     fftw_destroy_plan(p);
331     return result;
332 }
333
334 double GetEvalSolveM_0(double t)
335 {
336     fftw_complex *in, *out;

```

```

337     fftw_plan p;
338
339     in = (fftw_complex*) fftw_malloc(sizeof(fftw_complex) * NPOINTS);
340     out = (fftw_complex*) fftw_malloc(sizeof(fftw_complex) * NPOINTS);
341
342     double tempEnd = 90;
343     double step = tempEnd / NPOINTS;
344     double tempPower;
345
346     int i = 0;
347     for(double tempT = 0; tempT < tempEnd; tempT+=step)
348     {
349         if(!tempT)
350         {
351             in[i] = 0;
352             i++;
353             continue;
354         }
355         double S_0=-f(t,0)*f(t,0)/24*tempT*tempT*tempT;
356         in[i] = cexp(I * EPS * tempT)
357             / cpow(tempT, 3.0/2.0)
358             * (cexp(I * S_0) - 1);
359         i++;
360     }
361
362     p = fftw_plan_dft_1d(NPOINTS, in, out, FFTW_FORWARD, FFTW_ESTIMATE);
363
364     fftw_execute(p);
365     out[0]*=1.0/NPOINTS;
366     out[0]*=tempEnd;
367     double result = creal(out[0])*creal(out[0]) + cimag(out[0])*cimag(out[0]);
368     free(out);
369     fftw_destroy_plan(p);
370     return result;
371 }

```