

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)

Институт информационных технологий, математики и механики
Кафедра: Теории управления и динамики систем

Направление подготовки: «Фундаментальная информатика и информационные технологии»
Магистерская программа: «Компьютерная графика»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема:

**«Динамика решёток нелинейных осцилляторов:
развитие математического программного обеспечения
и исследования»**

Заведующий кафедрой:

Доктор физико-математических наук
_____ проф.Осипов Г.В.

Подпись

Выполнила:

Студентка группы 381406м2
_____ Игнашева Е.В.

Подпись

Научный руководитель:

Кандидат физико-математических наук
_____ ст. преп. Лаптева Т.В.

Подпись

Нижний Новгород

2016

Аннотация

В рамках представленной дипломной работы:

- 1) Исследована симплектическая схема интегрирования $SABA_2C$;
- 2) Разработана ускоренная схема вычислений, построенная на основе изученного интегратора;
- 3) Реализована ускоренная версия интегратора $SABA_2C$, использующая GPU;
- 4) Исследована модель связанной динамики FSW;
- 5) Получено утверждение о вероятности локализации в условиях низкой энергии системы;
- 6) Продемонстрирована эффективность разработанной программы на примере доказательства полученного утверждения.

Содержание

§ 1. ВВЕДЕНИЕ	4
§ 2. ПОСТАНОВКА ЗАДАЧИ	5
§ 3. ТЕОРЕТИЧЕСКОЕ ИССЛЕДОВАНИЕ.....	6
3.1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ	6
3.2. УРАВНЕНИЯ ИЗМЕНЕНИЯ КООРДИНАТ И ИМПУЛЬСОВ	6
3.3. ВЕРОЯТНОСТЬ ЛОКАЛИЗАЦИИ.....	7
§ 4. ИНТЕГРАТОР	10
4.1. ДЕМОНСТРАЦИЯ ТОЧНОСТИ ИНТЕГРИРОВАНИЯ	12
4.2. ЭТАПЫ РАЗРАБОТКИ ИНТЕГРАТОРА	13
4.3. МАКСИМАЛЬНО УСКОРЕННАЯ ВЕРСИЯ	17
4.4. ПОЛУЧЕННОЕ УСКОРЕНИЕ ВЫЧИСЛЕНИЙ	19
§ 5. ПРАКТИЧЕСКОЕ ИССЛЕДОВАНИЕ	20
5.1. ДЕМОНСТРАЦИЯ ПОНЯТИЯ ЛОКАЛИЗОВАННОСТИ	21
5.2. КРИТЕРИЙ ЛОКАЛИЗАЦИИ И ЕГО ПРОВЕРКА	22
5.3. ОКОНЧАТЕЛЬНОЕ СТАТИСТИЧЕСКОЕ ИССЛЕДОВАНИЕ	28
§ 6. ЗАКЛЮЧЕНИЕ	30
§ 7. ЛИТЕРАТУРА	31
ПРИЛОЖЕНИЕ	32
А) ОПИСАНИЕ СХЕМЫ SABA.....	32
В) ВЫБОР КОНФИГУРАЦИИ СХЕМЫ SABA	35
С) КОД ОСНОВНЫХ ВЫЧИСЛЕНИЙ.....	37

§ 1. Введение

Андерсоновская локализация — явление, возникающее при распространении волн в среде с пространственными неоднородностями. Оно состоит в том, что при определённой конфигурации беспорядка колебания приобретают характер стоячей волны, локализованной в ограниченной области пространства. Эта теория была предложена Нобелевским лауреатом Филиппом Андерсоном в 1958 году.

Изучение и дальнейшая разработка теории Андерсоновской локализации по-прежнему составляют интерес многих учёных. Теоретические и численные исследования в данной области проводятся для определённого класса задач, характерной чертой которых является необходимость численного моделирования на больших временных и пространственных масштабах, при этом используются колебательные решёточные системы с сохранением энергии.

Проведение численных экспериментов для этих задач требует существенных ресурсов, так как необходимо получать результаты высокой точности и запускать большое количество экспериментов. В связи с этим разработка быстрого и точного инструмента численного интегрирования весьма востребована.

В рамках данной работы на основе схем интегрирования, применявшихся ранее, был разработан и реализован интегратор, выигрывающий по производительности у своих предшественников на два порядка. Он был применён в исследовании утверждения о линейном характере вероятности локализации в условиях низких значений энергии системы. Чтобы подтвердить это утверждение, необходимо получить объемную экспериментальную выборку. Ранее для этой цели было необходимо использование компьютерного кластера. Теперь для получения этого же результата достаточно мощности одного персонального компьютера.

§ 2. Постановка задачи

1) Теоретическая часть

- 1.1) изучить математическую модель связанной динамики FSW;
- 1.2) изучить вопрос о сохранении Андерсоновской локализации.

2) Практическая часть

- 2.1) изучить симплектический метод интегрирования $SABA_2C$;
- 2.2) реализовать интегратор и подтвердить корректность вычислений;
- 2.3) выбрать критерий локализации и подтвердить его корректность;
- 2.4) максимально ускорить процесс вычислений без ущерба результатам.

§ 3. Теоретическое исследование

3.1. Математическая модель

В данной работе рассматривается одномерная решетка связанных осцилляторов. Каждый из них описывается своим значением координаты u и импульса p . Взаимодействие между ними описывается системой FSW¹, которая записана своим гамильтонианом:

$$H_{FSW} = \sum_l \left[\frac{p_l^2}{2} + \frac{e_l u_l^2}{2} + \alpha \frac{(u_m - u_l)^4}{\gamma} \right], \quad (1)$$

Где H_{FSW} - гамильтониан системы, p_l - импульс l -го осциллятора, u_l - координата l -го осциллятора, e_l - величина потенциала l -го осциллятора, α и γ - коэффициенты, с помощью которых регулируется вклад нелинейности в динамику системы.

Потенциал e_l - есть случайная равномерно распределённая величина, плотность распределения которой равна $f_x(x) = \begin{cases} 1, & x \in \left[\frac{1}{2}, \frac{3}{2}\right] \\ 0, & x \notin \left[\frac{1}{2}, \frac{3}{2}\right] \end{cases}$. В случае отсутствия нелинейности система демонстрирует Андерсоновскую локализацию, т.е. собственные моды системы являются экспоненциально локализованными на одном узле решетки.

3.2. Уравнения изменения координат и импульсов

Система называется гамильтоновой, если её уравнения могут быть записаны в форме Гамильтона:

$$\dot{p}_l = -\frac{\partial H}{\partial u_l}, \dot{u}_l = \frac{\partial H}{\partial p_l}, l = \overline{1, N} \quad (2)$$

Будем рассматривать случай, когда вклад в динамику одного осциллятора существенен только со стороны двух его ближайших соседей. Для FSW системы запишем уравнения координат и импульсов для каждого осциллятора:

¹ Frohlich-Spencer-Wayne-type classical lattice [2].

$$\begin{cases} \dot{p}_l = -\frac{\partial H}{\partial u_l} = -e_l u_l + \frac{\alpha}{\gamma}(u_{l+1} - u_l)^3 + \frac{\alpha}{\gamma}(u_{l-1} - u_l)^3 \\ \dot{u}_l = \frac{\partial H}{\partial p_l} = p_l \end{cases}, l = \overline{1, N} \quad (3)$$

Далее будем рассматривать приведённую систему:

$$\ddot{u}_l = -e_l u_l + \frac{\alpha}{\gamma}(u_{l+1} - u_l)^3 + \frac{\alpha}{\gamma}(u_{l-1} - u_l)^3, l = \overline{1, N} \quad (4)$$

3.3. Вероятность локализации

Получим теоретическую вероятность сохранения Андерсоновской локализации при следующих начальных условиях: смещения $u_l = 0$, при $l = \overline{1, N}$, импульсы $p_l = 0$, при $l \in \left\{ \overline{1, N} \setminus \left[\frac{N}{2} \right] \right\}$, $p_{\left[\frac{N}{2} \right]} = \sqrt{2E_{initial}}$. Воспользуемся методами исследования, применяемыми в теории возмущений, предложенными в [1][5]. Будем искать решение в виде суммы последовательных приближений:

$$u_l = u_l^0 + u_l^1 + \dots \quad (5)$$

Получим уравнение изменения координат любого осциллятора без учёта нелинейного воздействия. При $\alpha = 0$:

$$\ddot{u}_l = -e_l u_l \quad (6)$$

$$u_l = A_l \cos(\sqrt{e_0} t) \quad (7)$$

Введем следующую нумерацию: $\dots, -2, -1, 0, 1, 2, \dots$ причем центральный осциллятор будет иметь индекс 0. Тогда относительно центрального амплитуды будут следующими: $A_0 \neq 0, A_{\pm 1} \ll 1, A_{\pm 2} = 0$. Без потери общности примем $\alpha = 1$. Рассмотрим для такого случая уравнение динамики 1-го осциллятора:

$$\ddot{u}_1 = -e_1 u_1 + \frac{1}{\gamma}(u_2 - u_1)^3 + \frac{1}{\gamma}(u_0 - u_1)^3 \quad (8)$$

Для 2-го осциллятора $A_{\pm 2} = 0$, то $u_2 \ll 1$, пренебрежём его влиянием:

$$\ddot{u}_1 = -e_1 u_1 - \frac{u_1^3}{\gamma} + \frac{1}{\gamma}(u_0 - u_1)^3 = -e_1 u_1 - \frac{u_1^3}{\gamma} + \frac{1}{\gamma}[u_0^3 - 3u_0^2 u_1 + 3u_0 u_1^2 - u_1^3] \quad (9)$$

Слагаемые высоких порядков не учитываем, так как они вносят незначительный вклад. Уравнение сводится к следующему виду:

$$\ddot{u}_1 + e_1 u_1 = \frac{1}{\gamma} u_0^3 - \frac{3}{\gamma} u_0^2 u_1 \quad (10)$$

Пренебрежём правой частью уравнения (10): вклад суммы $\frac{1}{\gamma} u_0^3 - \frac{3}{\gamma} u_0^2 u_1$ будет незначительным, так как смещения входят в неё в третьей степени. Для \ddot{u}_1 получим $\ddot{u}_1 \sim -e_0 u_1$. Подставим полученное приближение:

$$-e_0 u_1 + e_1 u_1 = \frac{1}{\gamma} A_0^3 \cos^3(\sqrt{e_0} t) - \frac{3}{\gamma} A_0^2 A_1 \cos^3(\sqrt{e_0} t) \quad (11)$$

Так как значения амплитуд $A_{\pm 1} \ll 1$, последним слагаемым можем пренебречь. Для оставшейся части применим формулу косинусов и продолжим разложение:

$$-e_0 u_1 + e_1 u_1 \cong \frac{1}{\gamma} A_0^3 \left(\frac{3}{4} \cos(\sqrt{e_0} t) + \frac{1}{4} \cos(3\sqrt{e_0} t) \right) \quad (12)$$

Пренебрежём слагаемым $\frac{1}{4} \cos(3\sqrt{e_0} t)$, так как его вклад в конечное решение будет незначительным из-за частоты аргумента. Подставим $u_1 = A_1 \cos(\sqrt{e_0} t)$:

$$-e_0 A_1 \cos(\sqrt{e_0} t) + e_1 A_1 \cos(\sqrt{e_0} t) \cong \frac{1}{\gamma} A_0^3 \frac{3}{4} \cos(\sqrt{e_0} t) \quad (13)$$

В результате для A_1 и A_0 получаем следующее соотношение:

$$\frac{A_1}{A_0} = \frac{3A_0^2}{4\gamma(e_1 - e_0)} \quad (14)$$

Без потери общности примем $e_1 = 1$. Тогда для двух ближайших соседей получаем:

$$\frac{A_{\pm 1}}{A_0} = \frac{3}{4\gamma} \frac{A_0^2}{[e_{\pm 1} - 1]} \quad (15)$$

Энергия центрального осциллятора может быть выражена через его амплитуду следующим образом:

$$E_{ini} = \frac{A_0^2}{2} \quad (16)$$

Введём для удобства $\mu = \frac{3}{2\gamma} E_{ini}$. Решение будем считать локализованным, если не наблюдается существенный рост амплитуд среди ближайших соседей:

$$\frac{A_{\pm 1}}{A_0} \ll 1, \text{ или } \frac{\mu}{[e_{\pm 1} - 1]} < 1 \quad (17)$$

Найдем область делокализации:

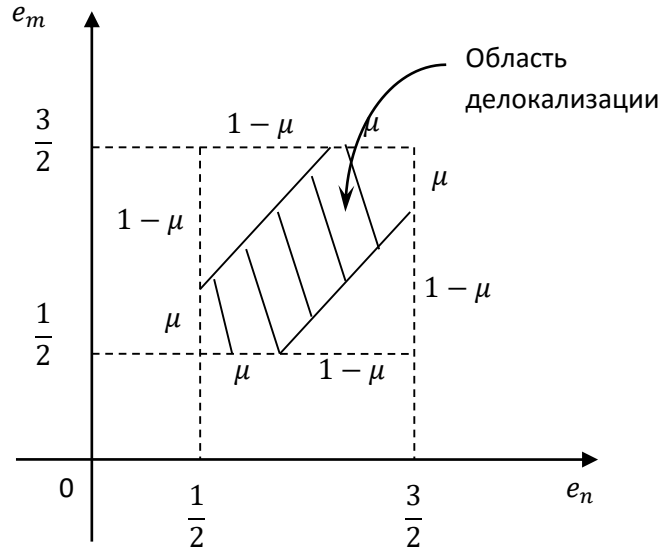


Рис. 1. Область существования локализации.

Найдем вероятность делокализации:

$$P_D = 1 - \frac{1}{2}(1 - \mu)^2 \cdot 2 = \mu(2 - \mu) \quad (18)$$

μ^2 стремится к нулю, поэтому получим:

$$P_D = \frac{3}{2\gamma} A_0^2 \xrightarrow{\gamma=1} P_D = \frac{3}{2} A_0^2 \quad (19)$$

Отсюда получим вероятность локализации:

$$P_L = 1 - P_D = 1 - 3E_{ini} \quad (20)$$

Подведем итог:

Если E_{ini} – полная энергия системы, P_L – вероятность локализации, то:

при $0 < E_{ini} < \frac{1}{3}$: $P_L = 1 - 3E_{ini}$,

при $E_{ini} > \frac{1}{3}$: $P_L \approx 0$

§ 4. Интегратор

Симплектические схемы были разработаны для численного интегрирования гамильтоновых систем. Поскольку нельзя получить аналитическое решение теоретического гамильтониана, строится смещённый по энергии гамильтониан, который решается точно. Разница значений истинного и смещённого гамильтонианов контролируется шагом интегрирования [4]. Благодаря своей надёжности и точности симплектические схемы интегрирования повсеместно применяются для моделирования процессов небесной механики. Основным их достоинством является то, что ошибка по энергии остаётся ниже некоторого порогового значения в течение всего времени интегрирования.

Подход Рута [8], использованный для явного интегрирования, разбивает гамильтониан на два слагаемых $H = A + B$, каждое из которых интегрируется в отдельности. На основе этого подхода Ласкар и Робутель [6] разработали семейство *SABA*-алгоритмов для гамильтонианов, представляемых в виде $H = A + \varepsilon B$ с малым параметром $\varepsilon \ll 1$. Полное описание разностной схемы интегрирования можно найти в Приложении А.

В данной работе была использована и реализована схема интегрирования *SABA* второго порядка с корректором, или *SABA₂C* [1]. Это симплектическая схема интегрирования, где под корректором понимается возможность корректировать ошибку по энергии с помощью изменения шага интегрирования. Сравнение точности и времени работы *SABA₁*, *SABA₂*, *SABA₂C* и *SABA₄* при разных шагах интегрирования было проведено в [3], где был сделан вывод о том, что *SABA₂C* даёт наилучший результат при малых шагах интегрирования. Было получено значение времени, требуемого для проведения вычислений. Относительно прочих конфигураций, при счёте с использованием схемы *SABA₂C* требуется больше времени, но разница не критична. Краткое описание проведённого исследования и прилагающиеся к нему рисунки и комментарии приведены в Приложении В. Опираясь на [3], мы использовали и реализовали схему *SABA₂C* для проведения экспериментов в рамках настоящего исследования.

Для проведения какого-либо эксперимента нам необходимо определить набор начальных условий, с тем чтобы однозначно описать конфигурацию задачи. В качестве шаблона для записи начальных условий эксперимента будем использовать таблицу следующего вида:

Таблица 1:

Шаблон для записи начальных условий эксперимента

E_0	...	N_{osc}	...	N_{run}	...	t_{start}	...	t_{end}	...	n	...
-------	-----	-----------	-----	-----------	-----	-------------	-----	-----------	-----	-----	-----

В таблице 1 представлены следующие характеристики, описывающие единичный запуск: E_0 – полная энергия цепи, N_{osc} – количество осцилляторов в цепи, N_{run} – количество запусков с разным распределением неоднородности, t_{start} – начальное время интегрирования, t_{end} – конечное время интегрирования, n – величина подразбиения шага интегрирования. Данную таблицу, уже заполненную, будем использовать при описании экспериментов и не дифференцировать её, как отдельный элемент повествования, а считать частью описания каждого конкретного рисунка.

Все эксперименты, представленные в работе, были проведены на настольном компьютере, обладающем следующими основными характеристиками:

- Intel Core i7 6770K, 4.2 GHz,
- 16Gb RAM DDR4,
- NVidia GTX 960, 1.2 MHz.

Забегаая вперёд, отметим, что на первых этапах разработки вычисления проводились на CPU, но позже были перенесены на GPU с использованием технологии CUDA, поэтому необходимо описать графический процессор точнее:

- 1024 CUDA – ядра (отдельные физические вычислительные узлы),
- Совместимость с версией CUDA 5.2,
- Базовая тактовая частота 1127 MHz (с ускорением - 1178 MHz),
- 4 Gb видеопамяти.

4.1. Демонстрация точности интегрирования

Так как конфигурация интегратора уже выбрана, остаётся определиться с подразбиением шага интегрирования.

E_0	0.08	N_{osc}	16	N_{run}	1	t_{start}	0	t_{end}	10e8	n	1,5,10,15,20
-------	------	-----------	----	-----------	---	-------------	---	-----------	------	-----	--------------

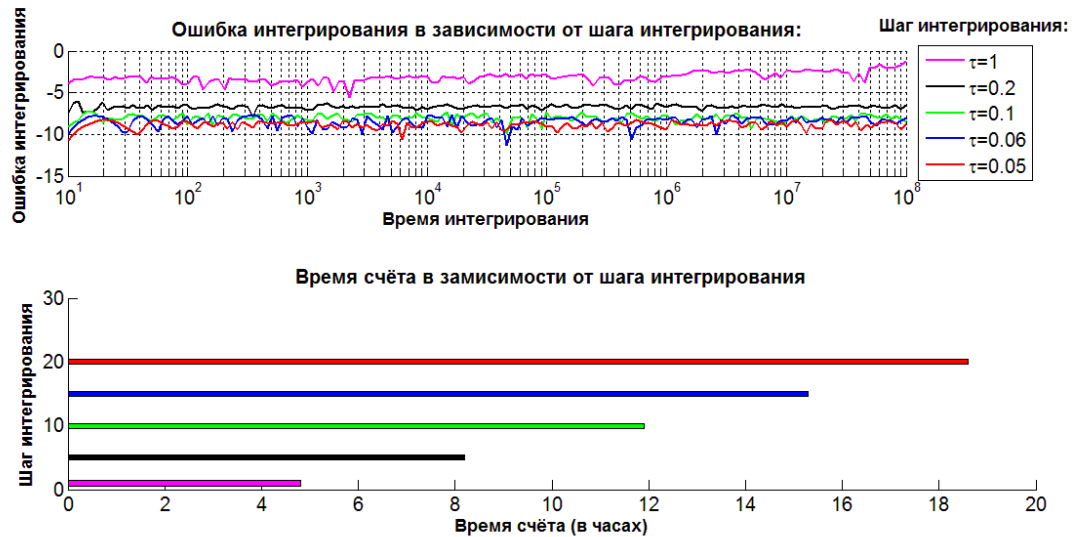


Рис. 2. Сравнение ошибки по энергии и времени счёта при интегрировании до $t=10e8$.

На рисунке 2 представлен график зависимости логарифма ошибки интегрирования от времени интегрирования для различных значений шага интегрирования $\tau = \{1, 0.2, 0.1, 0.06, 0.05\}$, что соответствует подразбиению $n = \{1, 5, 10, 15, 20\}$. По графику видно, что даже на больших временах ошибка не растёт, а остается постоянной, что особенно важно, когда изучается система с беспорядком и нелинейным воздействием. В рамках работы будем считать ошибку в 7-м знаке после запятой достаточной. Как мы видим из рисунка 2, дальнейшее дробление шага интегрирования улучшает точность вычислений, но, к сожалению, время вычислений растёт прямо пропорционально уменьшению шага интегрирования. Поскольку точность вычислений улучшается не так быстро, как растёт время счёта, остановимся на значении $n = 10$, или $\tau = 0.1$, так как в итоге ошибка интегрирования не превышала 7-го знака после запятой. Такая точность позволяет нам в дальнейшем полагать, что вносимая погрешность вычислений не будет качественно влиять на результат.

4.2. Этапы разработки интегратора

Первая реализация представляла из себя последовательно запрограммированную схему интегрирования $SABA_2C$, а именно: был запрограммирован алгоритм интегрирования, проверена корректность работы в смысле точности интегрирования. После того как полученные расчёты были проверены на корректность, начался процесс анализа и оптимизации программы, так как эксперимент, состоящий из одного запуска одного распределения неоднородности в системе из 100 осцилляторов для времени $10e7$ и шагом интегрирования 0.1 требовал для обчёта 120 минут. Такие долгие вычисления не являются продуктивными, так как для получения объемной экспериментальной выборки потребуются огромные ресурсы.

Чтобы читателю было проще понять дальнейшие рассуждения относительно оптимизации алгоритма, приведём схему, отображающую граф потока исполнения программы.

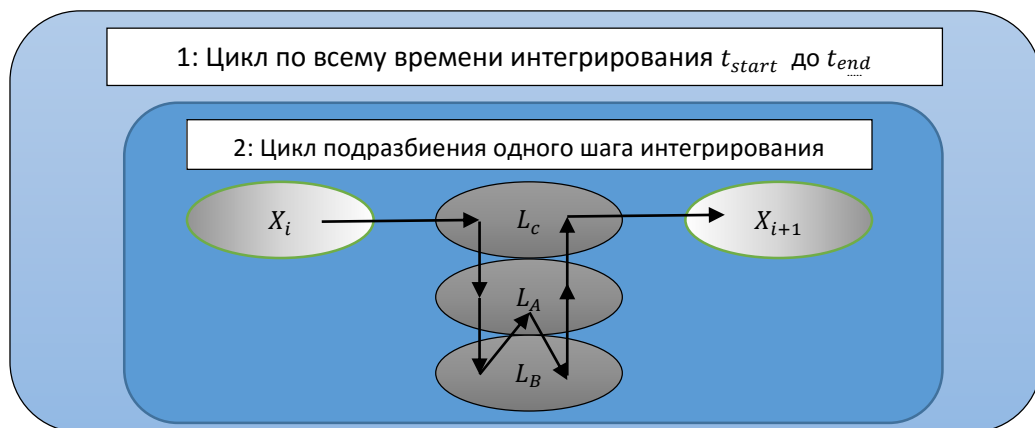


Схема 1. Граф потока исполнения программы.

Схема 1 состоит из двух больших прямоугольных блоков, которые являются циклами: внешний - по всему времени интегрирования, внутренний - по каждому шагу интегрирования.

Входными и выходными данными цикла 2 являются соответственно:

- X_i (состояние осцилляторов в момент времени t),
- X_{i+1} (состояние осцилляторов в момент времени $t + \tau$);

Процесс получения X_{i+1} из X_i состоит из последовательно исполняющихся однотипных функций, которые независимо изменяют каждую ячейку вектора X_i : L_A , L_B и L_C . Эти три функции исполняют роль дифференциальных операторов, описанных в Приложении А, а именно: функции L_A , L_B и L_C соответствуют формулам (A.17), (A.18) и (A.19).

Дальнейшие улучшения требуют анализа применимости к нашей задаче следующих трёх широко известных способов программной оптимизации:

- векторизации циклов;
- распараллеливании циклов;
- распараллеливании запусков.

Скажем пару слов о каждом из них. Эффективная векторизация циклов невозможна, так как вычисления имеют зависимость по данным, а это является критерием неприменимости векторизации. Распараллеливание циклов оказалось возможным и впоследствии дало существенный выигрыш при проведении экспериментов для большого количества осцилляторов. Этот способ оптимизации стал первым, который был применён к уже существующему коду. Он использовал интерфейс OpenMP. Поскольку реализация использовала CPU, мы не получили поразительного ускорения, так как ускорение в этом случае прямо пропорционально количеству ядер центрального процессора. Далее было добавлено распараллеливание запусков, что ускорило исполнение программы, увеличив количество экспериментов на один запуск. Напомним, что на используемом процессоре Intel Core i7-6770K присутствует 8 логических ядер. Максимально эффективным распараллеливанием при таком подходе является 8-поточное исполнение, чего всё же не достаточно.

Дальнейшее ускорение с использованием выбранных способов оптимизации возможно с увеличением количества вычислительных ядер. Для такого требования хорошо подходит архитектура, которой обладают графические вычислительные модули, или GPU. Код был перенесён на CUDA интерфейс, который используется для программирования графических процессоров.

E_0	0.08	N_{osc}	16	N_{run}	1	t_{start}	0	t_{end}	10e6	n	10
-------	------	-----------	----	-----------	---	-------------	---	-----------	------	-----	----

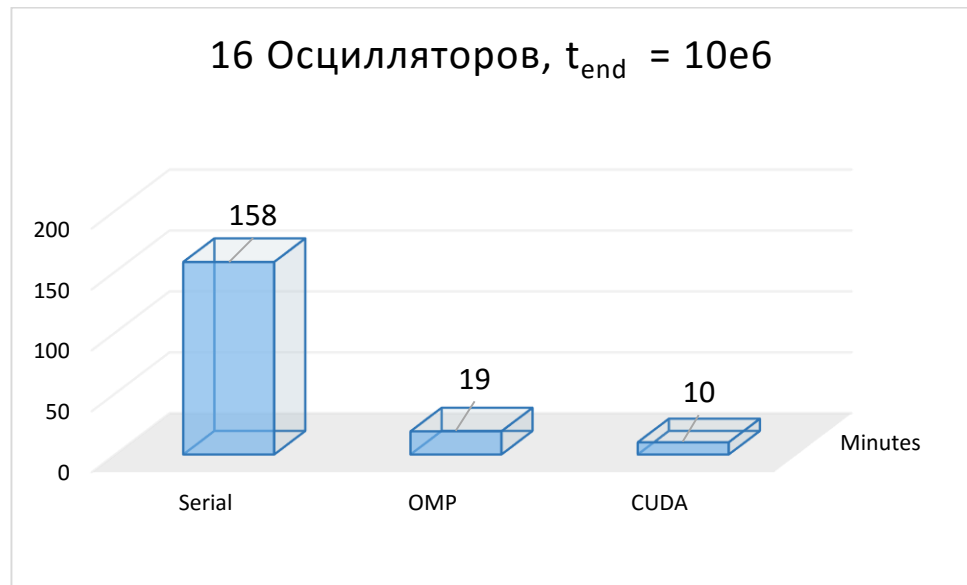


Рис. 3. Сравнение по времени трех реализаций численного интегратора для одинаковых конфигураций экспериментов.

Рисунок 3 показывает нам, разницу между временем, потраченным на получение одного и того же результата с использованием трёх версий программы. Напомним, что Serial и OMP реализации используют CPU, а CUDA реализация использует GPU. Итак, мы можем записать приблизительное значение полученного ускорения:

$$\frac{Time_{Serial}}{Time_{OMP}} \cong 8, \quad \frac{Time_{Serial}}{Time_{CUDA}} \cong 16$$

В предыдущем эксперименте для обчёта использовалась цепь из 16 осцилляторов. Попробуем увеличить объем задачи и проведём эксперимент, в котором исследуем цепь из 128 осцилляторов, причём все остальные начальные условия экспериментов оставим прежними.

E_0	0.08	N_{osc}	128	N_{run}	1	t_{start}	0	t_{end}	10e6	n	10
-------	------	-----------	-----	-----------	---	-------------	---	-----------	------	-----	----

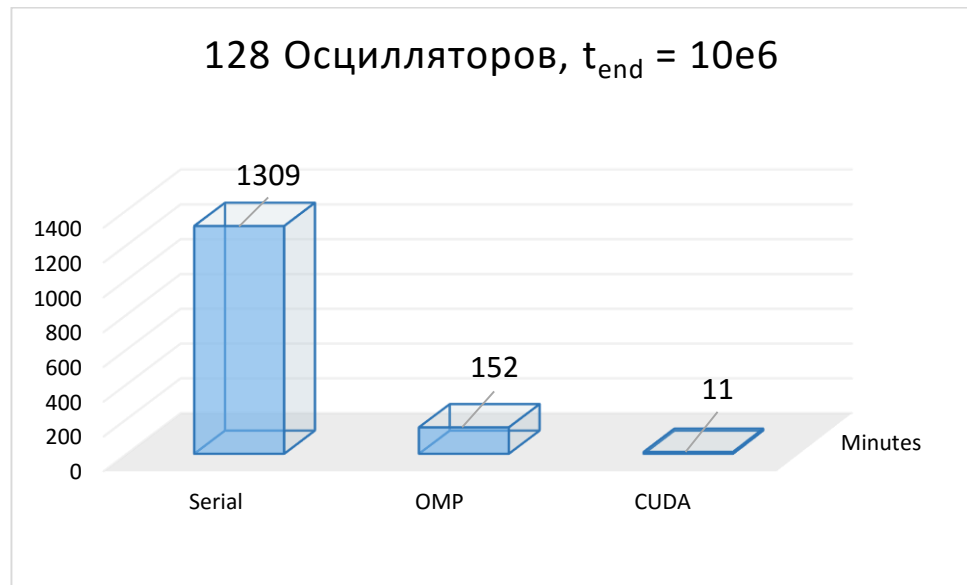


Рис. 4. Сравнение по времени трех реализаций численного интегратора для одинаковых конфигураций экспериментов.

По рисунку 4 видно, что время обсчета Serial версии существенно возросло и потребовало в общей сложности около 20 часов. Время, потраченное на OMP версию заняло около 2 часов, что значительно меньше, но всё равно не удовлетворительно, так как это время, потраченное на 1 запуск. Примечательно то, что CUDA версия заняла качественно столько же времени. Приведём временной выигрыш, достигнутый в этом эксперименте:

$$\frac{Time_{Serial}}{Time_{OMP}} \cong 8, \quad \frac{Time_{Serial}}{Time_{CUDA}} \cong 128$$

В рамках работы нам будет необходимо получить результаты для задачи, соответствующей цепи из 16 осцилляторов, так как нас интересует динамика при низких значениях энергии, а значит в небольшой окрестности возбуждённого осциллятора цепи. Приведем схематичное представление распределения вычислительной нагрузки при запуске нескольких одновременных экспериментов на GPU:

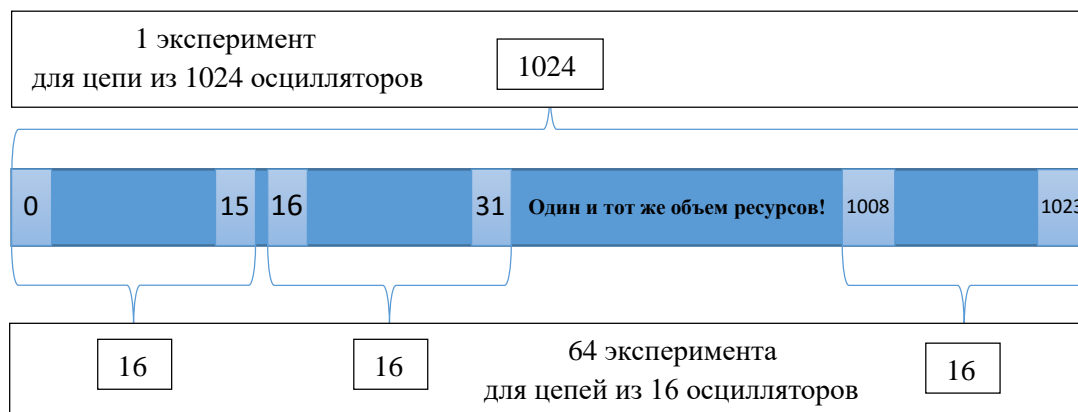


Рис. 5. Распределение вычислительной нагрузки на ядра GPU.

Из рисунка 5 видно, как мы можем изменить объем задачи и выиграть в смысле увеличения количества одновременных запусков. На этом рисунке приведён пример для 64 одновременных запусков. Поскольку на используемом чипе расположено 1024 вычислительных модулей, такой способ максимально эффективно использует ресурсы процессора.

4.3. Максимально ускоренная версия

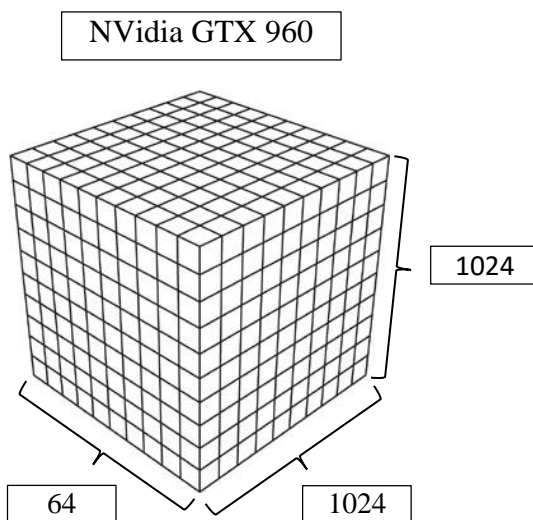


Схема 2. Визуализация сетки потоков процессора NVidia GTX 960.

На схеме 2 приведён размер «сетки потоков» используемого процессора NVidia GTX 960, она ограничена следующими значениями: 1024x1024x64.

Сетка потоков – термин, используемый при вычислениях на графических процессорах. Так как вычисления на GPU происходят по методу SIMD (single instruction, multiple data), запуск любой программы представляет из себя множество нитей, использующих общую

память. Это означает, что они исполняют один программный код, используют общее глобальное адресное пространство. Количество одновременно запускаемых нитей ограничено, и для удобства представляется в виде трёхмерной сетки слотов, которые программист может заполнять задачами, которые необходимо исполнять во время работы программы. В нашем случае такой размер сетки означает, что максимально возможное количество одновременных запусков равно 1024. Продемонстрируем результаты эксперимента с 64-я и 1024-я одновременными запусками на следующем рисунке.

E_0	0.08	N_{osc}	16	N_{run}	64 и 1024	t_{start}	0	t_{end}	10e7	n	10
-------	------	-----------	----	-----------	-----------	-------------	---	-----------	------	-----	----

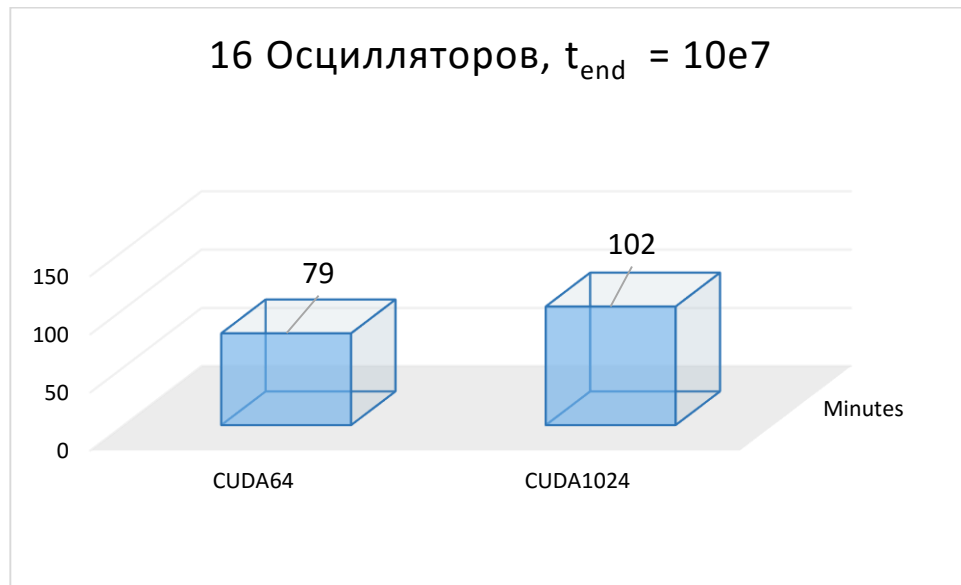


Рис. 6. Сравнение оптимальной (CUDA64) и максимальной (CUDA1024) нагрузки GPU.

На рисунке 6 видно, что при таком распределении нагрузки заметного уменьшения производительности не наблюдается. Причиной лежит в том, что все временные данные загружаются в память GPU в начале эксперимента и постоянного обмена данных с хостом (центральным процессором) не происходит. В итоге полученная программа даёт возможность, потратив одно и то же время, обчислить одновременно 1024 уникальных эксперимента вместо одного.

4.4. Полученное ускорение вычислений

Итак, мы построили численный интегратор и показали временной выигрыш, достигаемый при расчёте конкретной задачи. Отметим, что при внесении необходимых изменений, разработанная программа успешно применима и к другим решёточным системам, для которых возможно использование SABA-схемы интегрирования.

В целях обобщения, приведём окончательное значение доступного ускорения вычислений.

$$Time_{GPU} = \frac{Time_{CPU}}{N_{Osc} \times N_{Run}}. \quad (21)$$

Формула (21) даёт приблизительное значение ускорения. Имеется в виду, что GPU-реализация будет работать производительнее CPU-реализации в $N_{Osc} \times N_{Run}$ раз, где N_{Osc} – количество осцилляторов в цепи, а N_{Run} – количество запусков.

Сложно получить точную формулу ускорения, так как используемые центральный и графический процессоры всегда будут иметь сильные различия. Если принять, что один запуск одной реализации эксперимента будет занимать на CPU и на GPU одинаковое время, то формула будет отражать полученный выигрыш точнее.

Итак, для любой математической модели, описывающей динамику связанной решёточной системы, для которой возможно использование схемы численного интегрирования SABA, применим разработанный программный продукт, причём доступное ускорение времени счёта будет соответствовать (21).

§ 5. Практическое исследование

В данной части работы мы будем проводить практическое исследование результата, полученного в теоретическом блоке. Напомним читателю полученное ранее утверждение: при низких значениях энергии $0 < E_{initial} < \frac{1}{3}$ вероятность локализации траектории при возбуждении центрального осциллятора равна: $P_L = 1 - 3E_{ini}$. Важным является то, что вероятность локализации для описанных условий имеет линейный характер. Так как при выводе этого утверждения мы допускали различные пренебрежения, коэффициенты линейного уравнения в результате эксперимента могут принять иные значения. Главное – подтвердить линейный характер вероятности локализации.

Итак, для получения экспериментальной выборки будем проводить по 1024 запуска для каждого из 8 значений полной энергии системы: $E = \{0.08, 0.05, 0.03, 0.02, 0.008, 0.002\}$. Эти результаты нам необходимы для того, чтобы получить статистическое значение вероятности локализации траектории в зависимости от величины энергии системы.

5.1. Демонстрация понятия локализованности

Для начала, постараемся наглядно продемонстрировать, как выглядит локализованная и нелокализованная траектории, что они из себя представляют.

E_0	0,08	N_{osc}	16	N_{run}	2	t_{start}	0	t_{end}	10e7	n	10
-------	------	-----------	----	-----------	---	-------------	---	-----------	------	-----	----

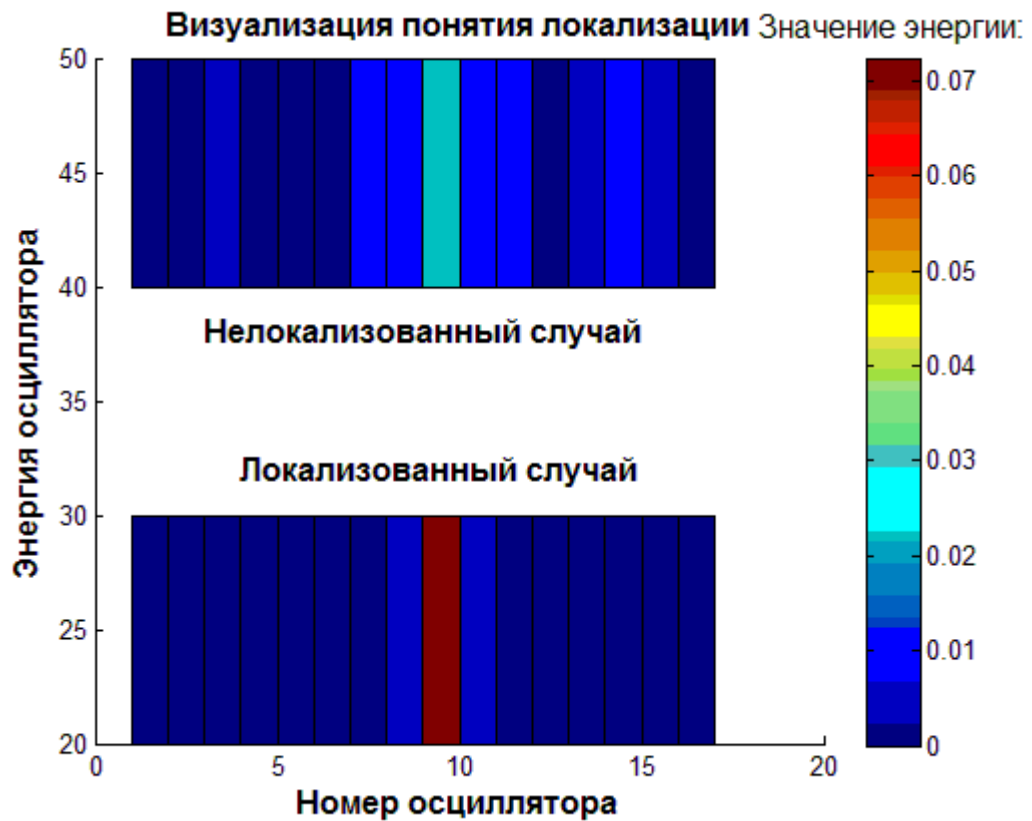


Рис. 7. Энергетические срезы в конечный момент интегрирования нелокализованной (сверху) и локализованной (снизу) траекторий.

На рисунке 7 представлен энергетический портрет решетки в последний момент интегрирования. Эти две реализации различаются только начальным случайным распределением потенциальных уровней. Они имеют одинаковое значение полной энергии и прочие равные характеристики. Для локализованного случая характерно сохранение динамики только на центральном осцилляторе, либо в малой его окрестности, а для нелокализованного случая – распространение энергии по всей цепи.

5.2. Критерий локализации и его проверка

В качестве меры локализованности, введём понятие «число участия». Эта величина будет отображать количество колеблющихся осцилляторов: $P = \frac{(\sum_{i=0}^N E_i)^2}{\sum_{i=0}^N E_i^2}$, где $E_{exp}^2 = (\sum_{i=0}^N E_i)^2$ – полное мгновенное значение энергии (экспериментальное). Будем считать траекторию локализованной, если в конечный момент времени $P \leq 2$. На рисунке 6 число участия для локализованной и нелокализованной траекторий соответственно равно $P_{LOCALIZED} = 1,0342$ и $P_{NOT LOCALIZED} = 8,2452$.

Чтобы подтвердить теоретический результат, нам необходимо будет провести статистическое исследование на больших временах (до $t=10e8$). Мы планируем оценивать наличие локализованности исходя из критерия, основанного на подсчёте числа участия. Но насколько верным является это решение? Сперва нам необходимо подтвердить правильность выбора такого способа оценки локализованности.

Для этого проведём две серии экспериментов: с прямым и обратным временем, причём каждый эксперимент с обратным временем будет соответствовать некоторому эксперименту с прямым временем: для обратного интегрирования мы не будем создавать нового распределения потенциальных уровней осцилляторов, а будем использовать исходное (ранее полученное при прямом интегрировании), а так же, в качестве стартового состояния цепи будем использовать конечное состояние цепи, полученное после прямого интегрирования. Для результата с прямым временем при оценке локализации мы будем использовать критерий, основанный на числе участия, а для результата с обратным временем – понятие евклидовой нормы в пространстве координат осцилляторов. Если результат, полученный после прямого и обратного проходов, будет достаточно близок к начальному состоянию цепи, которое мы знаем, то будем считать «обратную» траекторию локализованной.

Будем считать, что метод, основанный на числе участия, верен, если подавляющее большинство траекторий, локализованных на основе него, вернулись в своё первоначальное состояние. Так как мы исследуем систему с беспорядком, то предполагаем наличие различных ситуаций с совпадением локализации по обоим критериям. Вместе с тем, мы ожидаем, что наличие подобного шума не повлияет на результат качественно.

Так как интегрирование для обратного времени не даст нам удовлетворительных результатов на больших временах, вследствие нелинейного вклада в системе, ограничимся временем $t=10e5$ для того, чтобы проверить корректность критерия.

Проведём множественные эксперименты (по 1024 реализации для каждого значения энергии) с тем, чтобы получить статистическую вероятность локализации с прямым временем.

E_0	0.08, 0.03, 0.008, 0.002	0.05, 0.02,	N_{osc}	16	N_{run}	1024	t_{start}	0	t_{end}	10e5	n	10
-------	-----------------------------------	----------------	-----------	----	-----------	------	-------------	---	-----------	------	-----	----

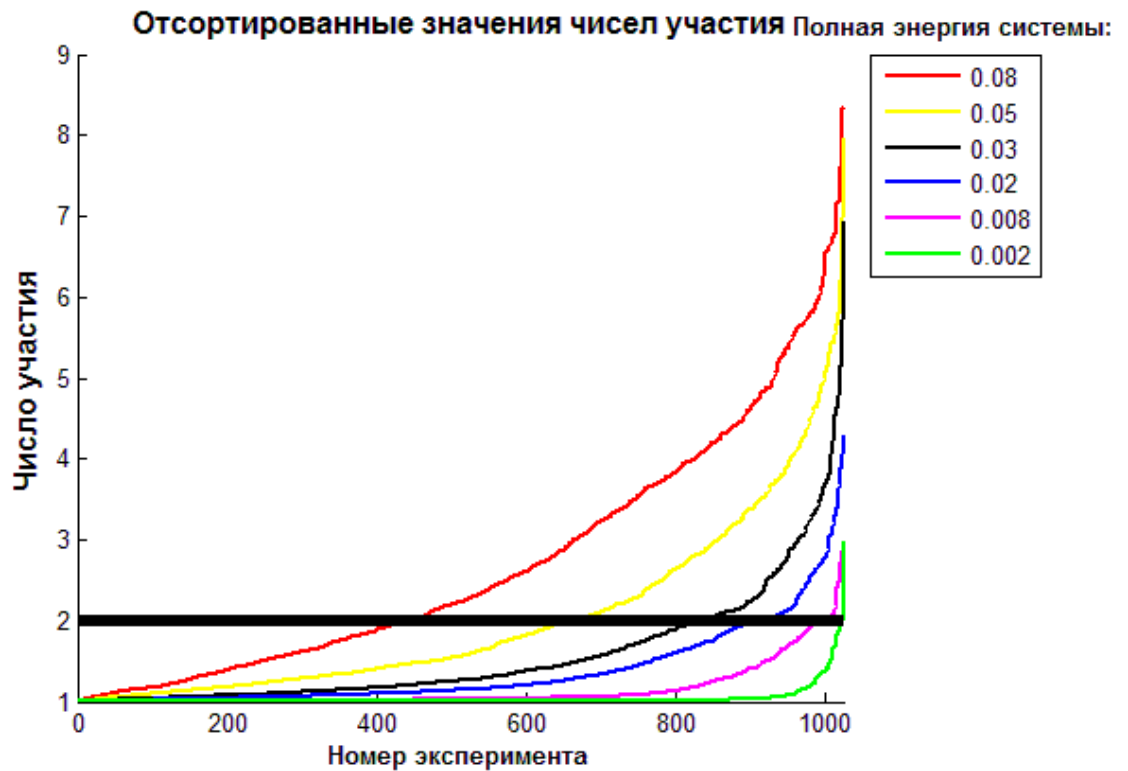


Рис. 8. Отсортированные по возрастанию значения чисел участия для всей экспериментальной выборки при различных значениях энергии.

На рисунке 8 изображены отсортированные значения чисел участия для всех проведенных экспериментов. Линией $P = 2$ мы оставляем «снизу» локализованные реализации, а «сверху» - нелокализованные. Теперь посмотрим на статистическую вероятность, а именно на процентное соотношение локализованных траекторий по отношению ко всем реализациям.

E_0	0.08, 0.03, 0.008, 0.002	0.05, 0.02,	N_{osc}	16	N_{run}	1024	t_{start}	0	t_{end}	10e5	n	10
-------	-----------------------------------	----------------	-----------	----	-----------	------	-------------	---	-----------	------	-----	----

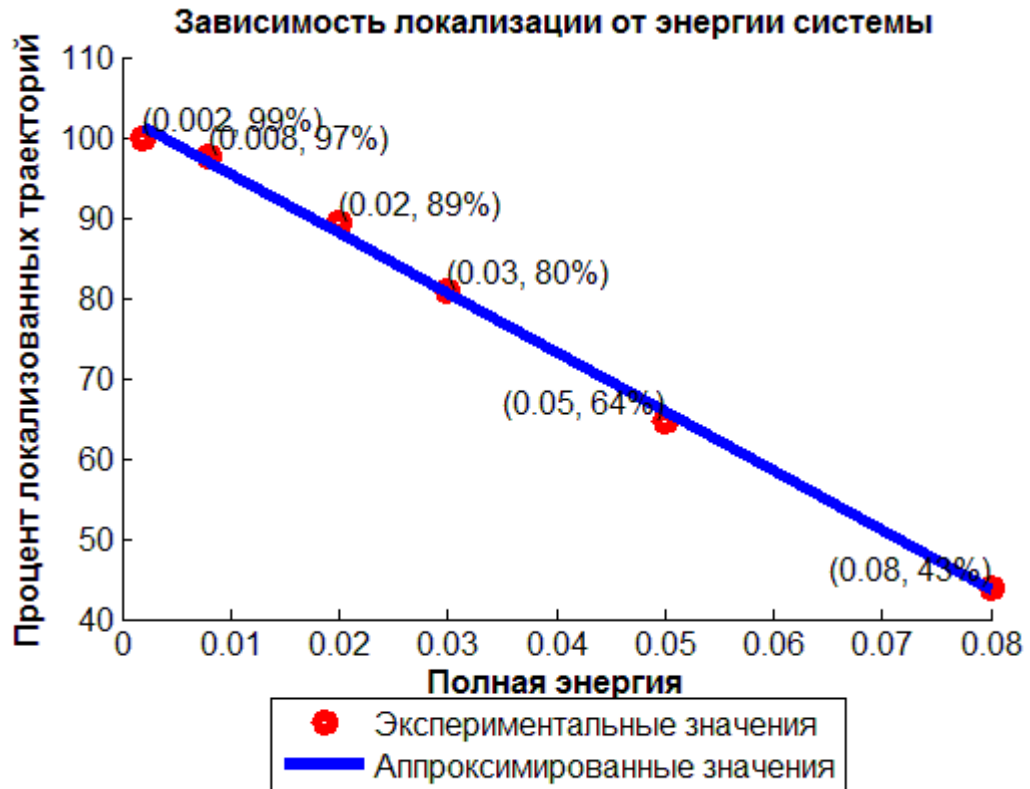


Рис. 9. Статистическая вероятность локализации по числу участия для конкретных значений энергии (красный) и аппроксимация результата (синий).

На рисунке 9 синим цветом изображена аппроксимация полученных значений вероятности. Мы видим наличие важного для нас характера линейности вероятности локализации.

Теперь проведём аналогичный эксперимент с обратным временем. Для этого уточним понятие евклидовой нормы и критерия локализации с её использованием.

Нам известно стартовое состояние системы X_0 : все осцилляторы в покое, кроме центрального, которому сообщается импульс $p_0 = \sqrt{2E_{initial}}$. После «обратного прохода» мы так же получаем некоторое состояние цепи \bar{X}_0 . Будем судить о локализованности траектории для обратного времени, если евклидова норма между состояниями X_0 и \bar{X}_0 равна: $\varepsilon = |X_0 - \bar{X}_0| \leq 10e - 4$. Распишем значение евклидовой нормы:

$$\varepsilon = \sqrt{\sum_{i=0}^N [(u_i - \bar{u}_i)^2 + (p_i - \bar{p}_i)^2]}.$$

Будем называть эту величину невязкой.

Проведём массовые эксперименты для обратного времени интегрирования, используя в качестве начальных условий конечное состояние системы в предыдущем эксперименте и ранее сгенерированный набор случайных потенциальных уровней.

E_0	0.08, 0.03, 0.008, 0.002	0.05, 0.02,	N_{osc}	16	N_{run}	1024	t_{start}	$10\epsilon 5$	t_{end}	0	n	10
-------	-----------------------------------	----------------	-----------	----	-----------	------	-------------	----------------	-----------	---	-----	----

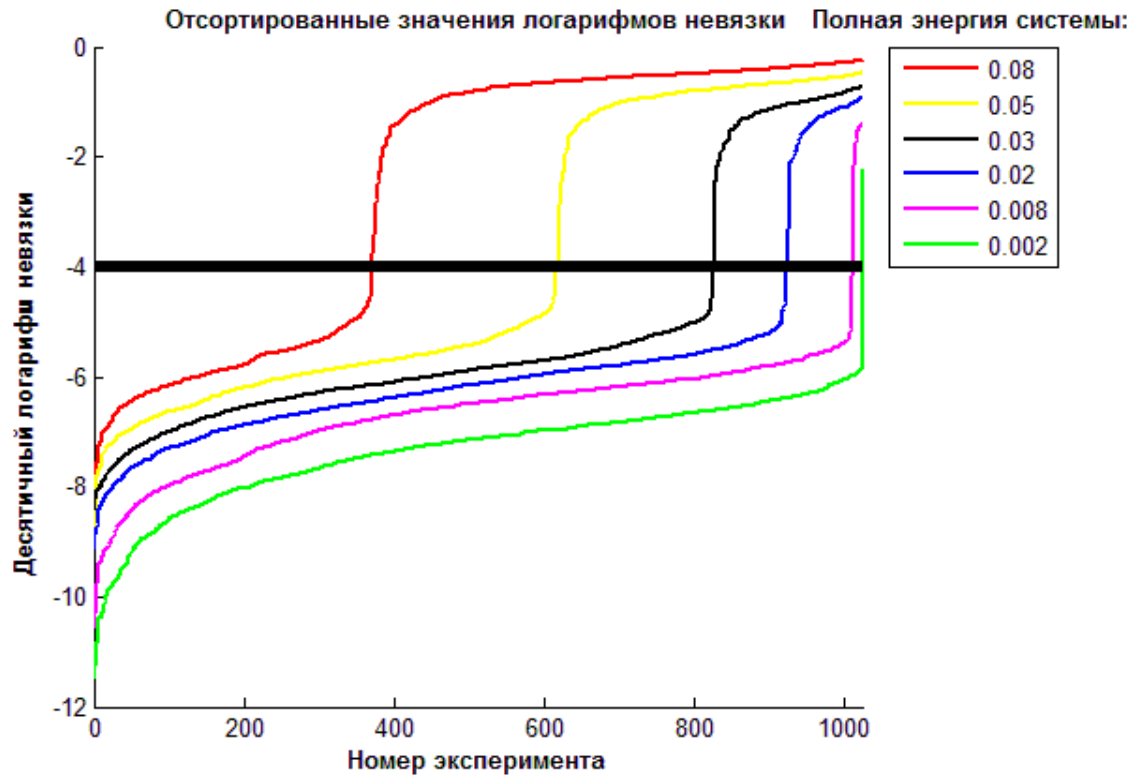


Рис. 10. Отсортированные по возрастанию логарифмы невязок для всех траекторий, полученных при обратном интегрировании.

На рисунке 10 мы видим отсортированные логарифмы невязок по каждому значению энергии. Линия $\log_{10} \varepsilon = -4$ разбивает выборку на локализованные (те, что «под прямой») и нелокализованные (те, что «над прямой»). Но среди тех, что мы считаем локализованными по невязке, могли оказаться траектории, которые были нелокализованы по числу участия и наоборот. Этот интересный факт можно увидеть на следующем рисунке.

E_0	0.08, 0.03, 0.008, 0.002	0.05, 0.02,	N_{osc}	16	N_{run}	1024	t_{start}	0; 10e5	t_{end}	10e5; 0	n	10
-------	-----------------------------------	----------------	-----------	----	-----------	------	-------------	------------	-----------	------------	-----	----

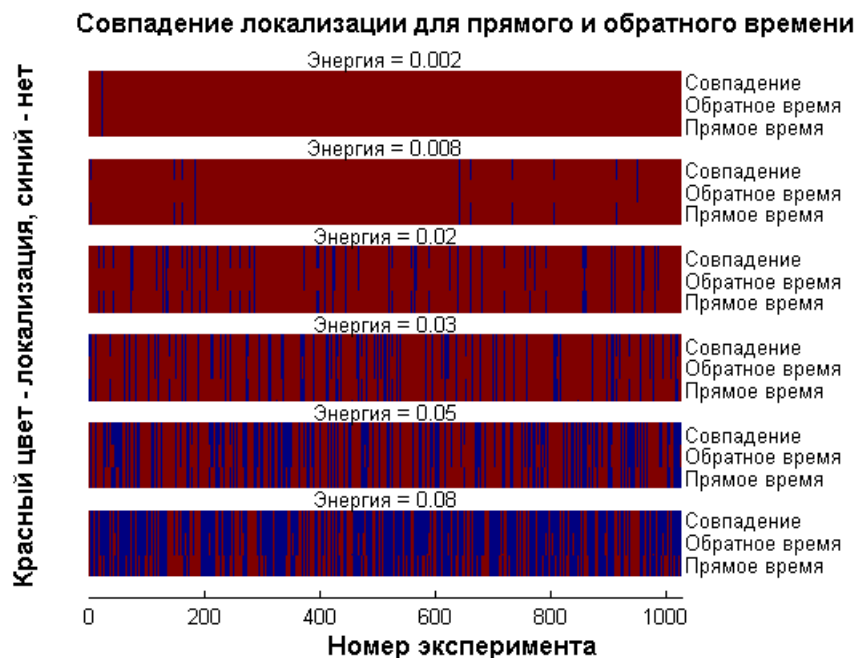


Рис. 11. Совпадение локализованности траекторий по обоим критериям: по числу участия и по невязке.

На рисунке 11 шесть «полос» соответствуют шести значениям энергий, которые мы исследуем. Каждая большая полоса состоит из трёх меньших, которые соответствуют критерию по числу участия (Прямое время), критерию по невязке (Обратное время) и ситуации, когда оба критерия выполняются (Совпадение). Красным цветом обозначены локализованные траектории, а синим – нелокализованные. Красный цвет присутствует в полосе с совпадениями только в том случае, когда мы считаем траекторию локализованной по обоим критериям. Итак, по рисунку 11 видно, что число локализованных по невязке не равно числу локализованных в совпадении.

E_0	0.08, 0.03, 0.008, 0.002	0.05, 0.02,	N_{osc}	16	N_{run}	1024	t_{start}	$10e5$	t_{end}	0	n	10
-------	-----------------------------------	----------------	-----------	----	-----------	------	-------------	--------	-----------	---	-----	----

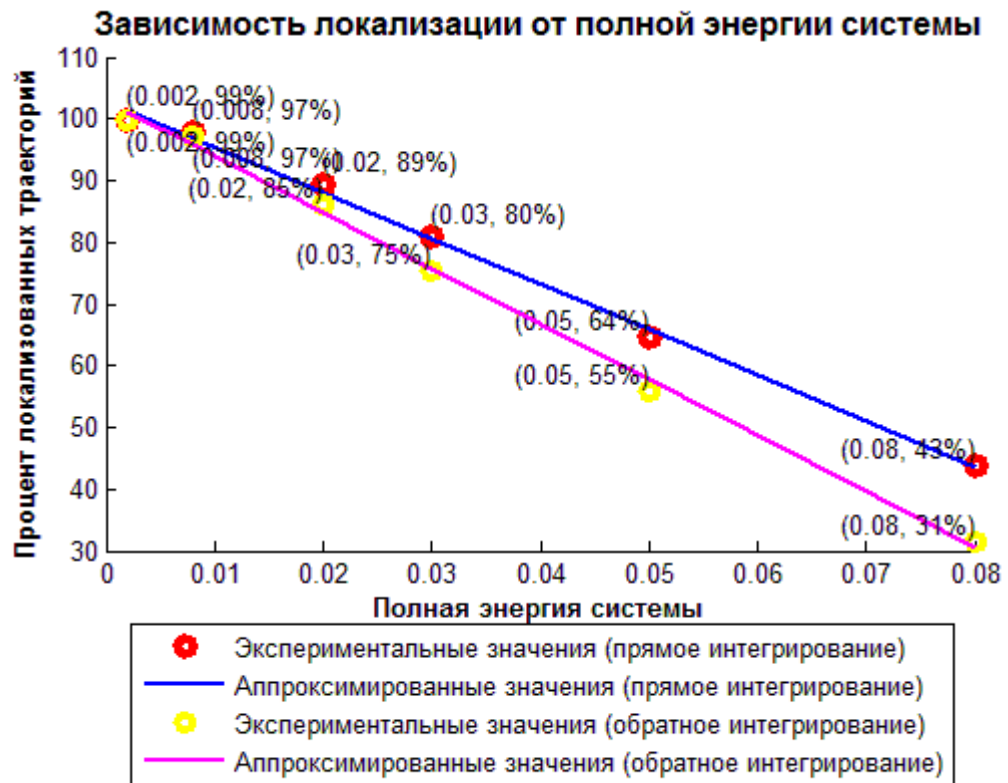


Рис. 12. Статистическая вероятность локализации по числу участия (синий) и по невязке (розовый).

На рисунке 12 изображены результаты, полученные при обратном интегрировании (розовый цвет) и при прямом интегрировании (синий цвет), причём вторые соответствуют рисунку 8. Красным отмечены экспериментальные значения вероятностей и сами значения энергии, синим – аппроксимация этих значений. Желтым – аналогичные вероятности и энергии, только для обратного времени интегрирования с использованием критерия по невязке. Рядом с каждой точкой приведены точные значения в виде: (энергия, вероятность). Рисунок 12 даёт нам важный результат: аппроксимации значений вероятности локализации по обоим критериям близки. Это даёт нам право считать, что мы правильно выбрали критерий локализованности по числу участия.

5.3. Окончательное статистическое исследование

Теперь, когда мы подтвердили критерий локализации по числу участия, воспользуемся им и проведём массовые эксперименты, но в этот раз для больших времён интегрирования.

E_0	0.08, 0.03, 0.008, 0.002	0.05, 0.02,	N_{osc}	16	N_{run}	1024	t_{start}	0	t_{end}	10e8	n	10
-------	-----------------------------------	----------------	-----------	----	-----------	------	-------------	---	-----------	------	-----	----

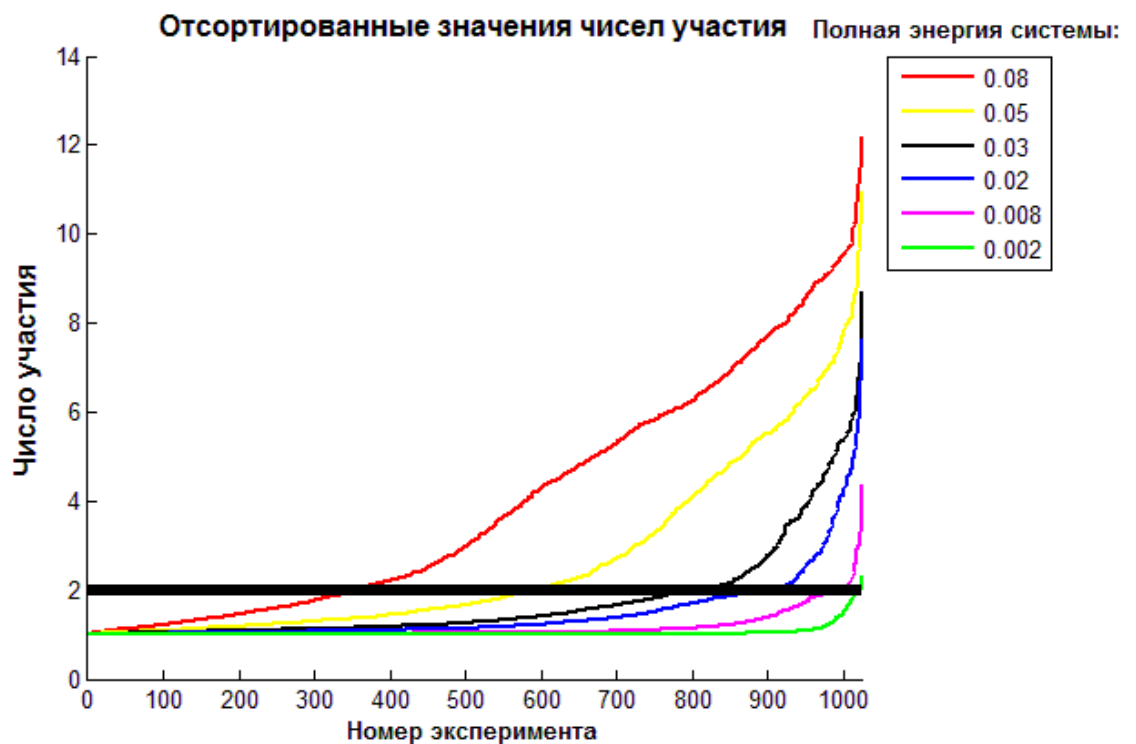


Рис. 13. Отсортированные по возрастанию значения чисел участия для всей экспериментальной выборки при различных значениях энергии.

Разница между рисунками 13 и 8 состоит во времени интегрирования: для рис.13 $t=10e8$, для рис.8 $t=10e5$. Интересно заметить, что кривые на рис.13 менее вогнуты. Это говорит нам о том, что процент локализованных траекторий снизился для всех значений энергии. Причина состоит в том, что в область локализации для эксперимента при $t=10e5$ попали ложно локализованные траектории, то есть им не хватило времени интегрирования, чтобы начать вести себя нелокализованно. В новом эксперименте при $t=10e8$ ложно локализованные траектории практически отсутствуют.

E_0	0.08, 0.03, 0.008, 0.002	0.05, 0.02,	N_{osc}	16	N_{run}	1024	t_{start}	0	t_{end}	10e8	n	10
-------	-----------------------------------	----------------	-----------	----	-----------	------	-------------	---	-----------	------	-----	----

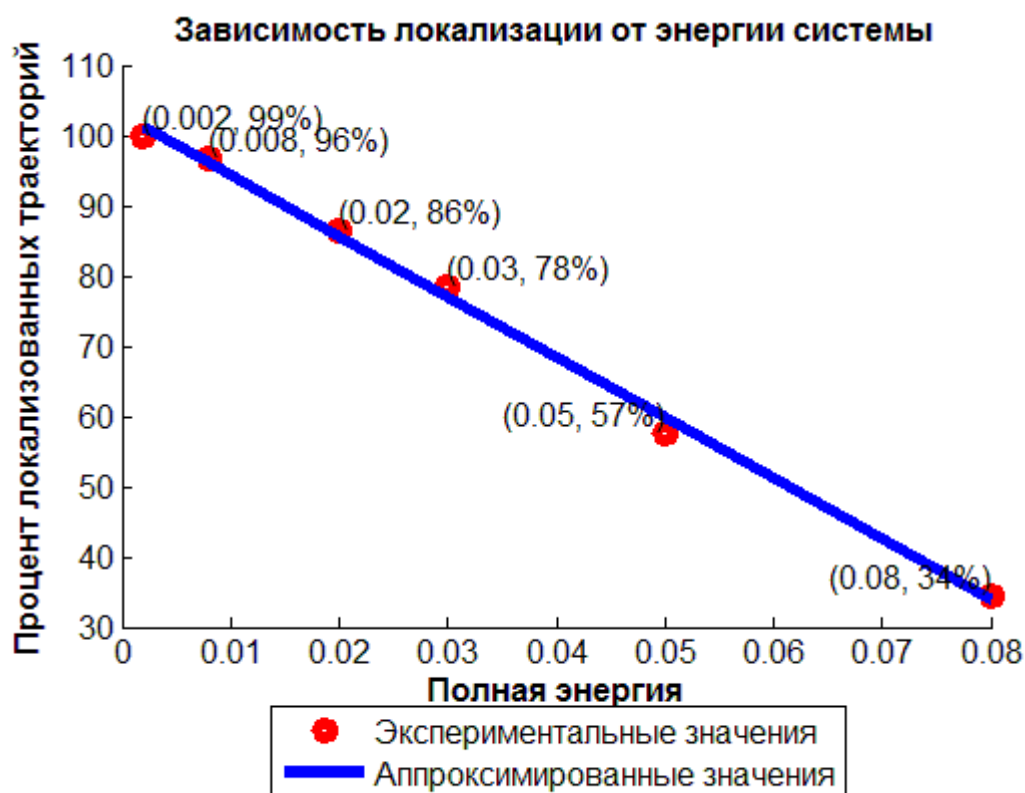


Рис. 14. Статистическая вероятность локализации по числу участия для конкретных значений энергии (красный) и аппроксимация результата (синий).

Рисунок 14 даёт нам итоговый результат практического исследования. Статистическая вероятность локализации, аппроксимированная на основе экспериментальных значений, имеет линейный вид, причём точки, соответствующие энергии системы и значению вероятности, практически принадлежат аппроксимированной (синей) кривой. Заметим, что значения вероятностей стали меньше, чем на рисунке 12. Это ещё раз подтверждает то, что на больших временах интегрирования мы получаем более точный результат, так как ложно локализованные траектории вносят меньший вклад в значение вероятности локализации.

Итак, главным результатом практического исследования является то, что нам удалось подтвердить теоретическое утверждение: при низких значениях энергии вероятность локализации линейно зависит от полной энергии системы.

§ 6. Заключение

В рамках данной работы была исследована математическая модель FSW[2]; было получено утверждение о характере локализации в условиях низких значений общей энергии системы (п.3.3). Был построен производительный инструмент численного интегрирования (§4), использующий схему SABA. Эффективность использования полученного программного продукта была продемонстрирована на примере объемного статистического эксперимента (п.5.3) с целью подтверждения теоретического вывода (п.3.3).

Основной задачей, поставленной в рамках данной работы, было построение эффективного инструмента численного интегрирования для класса решёточных систем с сохранением энергии. Хочется отметить, что эта задача была успешно выполнена. В п.4.4 приведена формула ускорения вычислений для общего случая. Теперь для любой математической модели, описывающей динамику связанной решёточной системы, для которой возможно использование схемы численного интегрирования SABA, применим разработанный программный продукт, причём доступное ускорение времени счёта будет соответствовать (21).

Поле для дальнейшей деятельности по-прежнему велико. Поскольку мы получили хороший инструмент интегрирования математических моделей, описывающих поведение решёточных колебательных систем, хотелось бы исследовать более сложные случаи, например двумерные. В 2D пространстве траектории бы имели намного более разнообразное поведение.

Применяя техники оптимизации вычислений (п.4.2), исследователи могут либо создавать собственные более производительные численные интеграторы для решения описанного класса задач, либо использовать уже существующее решение, внося необходимые изменения в код, соответствующий численному решению конкретной модели. Это позволит получать объёмные экспериментальные выборки, используя ресурсы одного настольного компьютера.

§ 7. Литература

1. Nonlinear lattice waves in heterogeneous media. T. V. Lapyeva, M. V. Ivanchenko and S. Flach. Topical Review, J. Phys. A: Math. Theor. 47, 493001, 2014.
2. J. Frohlich, T. Spencer and C. E. Wayne. Localization in disordered, nonlinear dynamical systems. J. Stat. Phys., vol. 42, page 247, 1986.
3. Яновская А. Самолокализация колебаний в нелинейных пространственно-неоднородных решетках. – Нижний Новгород, ННГУ им.Лобачевского, 2015.
4. Авдюшев В.А. Численные методы интегрирования обыкновенных дифференциальных уравнений: Курс лекций, 2009 год. - 52 с.
5. Кузнецов А.П. Лекции по теории колебаний и волн. Нелинейные колебания / Кузнецов А.П., Кузнецов С.П., Рыскин Н.М. – Саратов, изд-во СГУ им. Чернышевского, 2011. – 314 с.
6. J. Laskar and P. Robutel. High order symplectic integrators for perturbed Hamiltonian systems. Celest. Mech. Dyn. Astron., vol. 80, pages 39, 2001.
7. Л. Мандель, Э. Вольф. Оптическая когерентность и квантовая оптика. Пер. с англ. — М.: Физматлит, 2000. - 896 с.
8. R. D. Ruth. A canonical integration technique. IEEE Trans. Nucl. Sci., vol. NS-30, page 2669, 1983.

Приложение

А) Описание схемы SABA

Симплектические интеграторы были разработаны для численного решения систем дифференциальных уравнений, для которых можно записать гамильтониан:

$$\dot{p} = -\frac{\partial H}{\partial u}, \quad \dot{u} = \frac{\partial H}{\partial p}, \quad (\text{A.1})$$

Где $u = (u_1, \dots, u_N)$ и $p = (p_1, \dots, p_N)$ – обобщённые смещение и импульс соответственно, а H – гамильтониан.

Симплектическая схема интегрирования подразумевает под собой получение приближённого решения путём замены исходного гамильтониана на смещённый с сохранением канонического характера уравнений. Важно то, что смещённый гамильтониан интегрируется точно. Близость гамильтонианов контролируется величиной шага интегрирования.

Подход Рута [8], использованный для явного интегрирования, разбивает гамильтониан на два слагаемых $H = A + B$, каждое из которых интегрируется в отдельности. На основе этого подхода, Ласкар и Робутель [6] разработали семейство SABA-алгоритмов для гамильтонианов $H = A + \varepsilon B$ с малым параметром $\varepsilon \ll 1$.

Определим скобки Пуассона для функций f и g в следующем виде:

$$\{f, g\} = \sum_{l=1}^N \left[\frac{\partial f}{\partial p_l} \frac{\partial g}{\partial u_l} - \frac{\partial f}{\partial u_l} \frac{\partial g}{\partial p_l} \right] \quad (\text{A.2})$$

Канонические координаты представим в виде $x \equiv (p, u)$. Тогда уравнения гамильтониана примут следующий вид:

$$\dot{x} = \{H, x\} = L_H x, \quad (\text{A.3})$$

Где L_H – дифференциальный оператор, определённый как $L_H x = \{x, H\}$. В таком случае, решение может быть записано в виде:

$$x(t) = x_0 e^{tL_H}, \quad (\text{A.4})$$

Где x_0 – начальное состояние системы. Динамика $x(t)$ во времени от 0 до τ , где τ малый параметр, проходит под оператором $e^{\tau L_H}$. Если гамильтониан имеет вид $H = A + B$, решение для него можно записать в виде:

$$x(\tau) = x_0 e^{\tau(L_A + L_B)}. \quad (\text{A.5})$$

По формуле Кемпбелла—Бейкера—Хаусдорфа [7] ператор $e^{\tau L_H}$ тождественно равен: $e^{\tau L_H} \equiv e^{\tau(L_A + L_B)}$. Мы можем аппроксимировать его как:

$$e^{\tau L_A} e^{\tau L_B} = e^{\tau L_Z} \approx e^{\tau(L_A + L_B)}, \quad (\text{A.6})$$

Где $Z = A + B + O(\tau)$ – приближённое значение гамильтониана.

Оператор $S_1 = e^{\tau L_A} e^{\tau L_B}$ даёт нам простейшую симплектическую схему интегрирования для гамильтониана, представленного в виде двух слагаемых. Чтобы уменьшить остаток при Z необходимо использовать симплектические схемы более высоких порядков:

$$S_n = e^{c_1 \tau L_A} e^{d_1 \tau L_B} \dots e^{c_n \tau L_A} e^{d_n \tau L_B}, \quad (\text{A.7})$$

где c_i и d_i выбираются так, чтобы уменьшить порядок остатка.

Рассмотрим конкретный SABA-класс симплектических, обратимых по времени схем интегрирования, предложенных Ласкаром и Робутелем [6]:

$$SABA_{2n}: e^{c_1 \tau L_A} e^{d_1 \tau L_{\varepsilon B}} \dots e^{d_n \tau L_{\varepsilon B}} e^{c_{n+1} \tau L_A} e^{d_n \tau L_{\varepsilon B}} \dots e^{d_1 \tau L_{\varepsilon B}} e^{c_1 \tau L_A}, \quad (\text{A.8})$$

$$SABA_{2n+1}: e^{c_1 \tau L_A} e^{d_1 \tau L_{\varepsilon B}} \dots e^{c_{n+1} \tau L_A} e^{d_{n+1} \tau L_{\varepsilon B}} e^{c_{n+1} \tau L_A} \dots e^{d_1 \tau L_{\varepsilon B}} e^{c_1 \tau L_A}. \quad (\text{A.9})$$

Точность интеграторов SABA может быть существенно улучшена введением корректора. Для этого необходимо, чтобы слагаемое A имело квадратичную зависимость от импульса, а слагаемое B не имело зависимости от смещения. При выполнении этих условий становится возможным применение корректора:

$$SABA_n C = e^{-(\tau^3 \varepsilon^2 g/2) L_C} \cdot SABA_n \cdot e^{-\left(\frac{\tau^3 \varepsilon^2 g}{2}\right) L_C}. \quad (\text{A.10})$$

Подробное описание того, как построить схему SABA можно найти в [1]. В целях краткости изложения, мы приведём окончательный вид для $SABA_2$ и $SABA_2 C$ в общем виде:

$$SABA_2 = e^{c_1 \tau L_A} e^{d_1 \tau L_B} e^{c_2 \tau L_A} e^{d_1 \tau L_B} e^{c_1 \tau L_A} \quad (\text{A.11})$$

$$SABA_2C = e^{-(\tau^3 g/2)L_c} \cdot SABA_2 \cdot e^{-(\tau^3 g/2)L_c} \quad (\text{A.12})$$

Приведём вид схемы $SABA_2C$ для модели FSW, которая была исследована в данной работе.

Гамильтониан системы FSW:

$$H_{FSW} = \sum_l \left[\frac{p_l^2}{2} + \frac{e_l u_l^2}{2} + \alpha \frac{(u_m - u_l)^4}{\gamma} \right], \quad (\text{A.13})$$

Можно разбить на два слагаемых:

$$A = \sum_{l=1}^N \frac{p_l^2}{2}, \quad (\text{A.14})$$

$$B = \sum_{l=1}^N \frac{\tilde{\epsilon}_l u_l^2}{2} + \frac{1}{4} (u_{l+1} - u_l)^4, \quad (\text{A.15})$$

С корректором:

$$C = \{\{A, B\}, B\} = \sum_{l=1}^N [\tilde{\epsilon}_l u_l - (u_{l+1} - u_l)^3 - (u_{l+1} - u_l)^3]^2. \quad (\text{A.16})$$

Итоговая схема вычислений $SABA_2C$ по времени от t до $t + \tau$ представлена ниже:

$$e^{\tau L_A} = \begin{cases} u'_l = u_l + \tau p_l, \\ p'_l = p_l, \end{cases}, \quad (\text{A.17})$$

$$e^{\tau L_B} = \begin{cases} u'_l = u_l \\ p'_l = p_l - \tau Q_l, \end{cases} \quad (\text{A.18})$$

$$e^{\tau L_C} = \begin{cases} u'_l = u_l \\ p'_l = p_l - 2\tau \{ [\tilde{\epsilon}_l + 3(u_{l-1} - u_l)^2 + 3(u_{l+1} - u_l)^2] Q_l - \\ - 3(u_{l-1} - u_l)^2 Q_{l-1} - 3(u_{l+1} - u_l)^2 Q_{l+1} \}, \end{cases} \quad (\text{A.19})$$

где

$$Q_x = \tilde{\epsilon}_x u_x - (u_{x-1} - u_x)^3 - (u_{x+1} - u_x)^3 \quad (\text{A.20})$$

В) Выбор конфигурации схемы SABA

Ниже кратко приведены результаты анализа [3], где был подробно исследован вопрос об эффективности различных SABA схем интегрирования. При проведении экспериментов в [3] была взята за основу решётка Клейна-Гордона. В нашей работе мы используем другую математическую модель, но для того, чтобы оценить точность вычислений, нам неважно, какая система выбрана, так как обе они удовлетворяют условиям, при которых мы можем построить для них SABA схему.

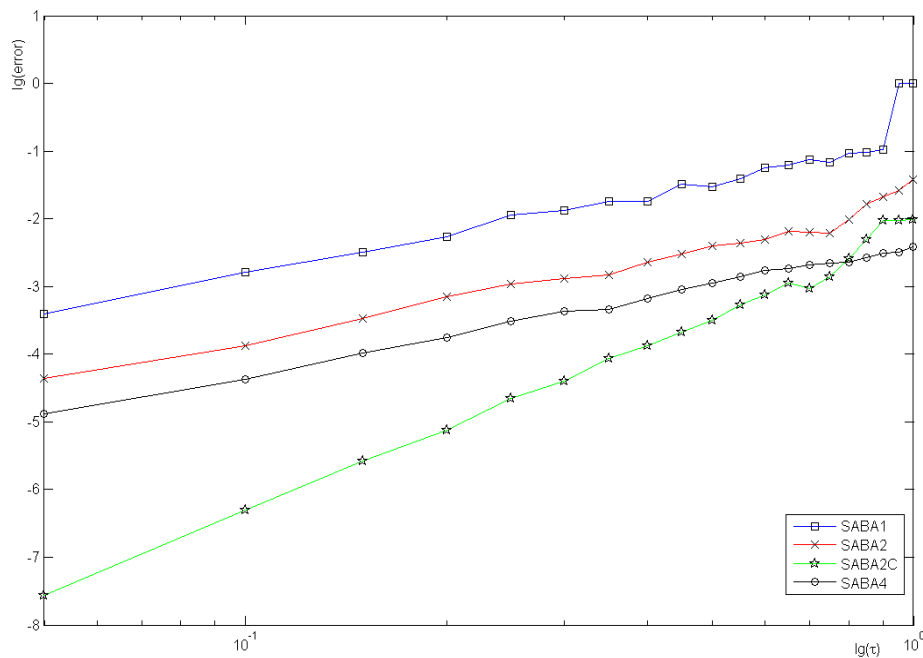


Рис. В.1. Сравнение SABA -интеграторов различного порядка: в двойных логарифмических осях глобальная погрешность интегрирования показана в зависимости от шага интегрирования τ (конечное время интегрирования для всех реализаций составляло $t_{\text{end}} = 10^7$). Параметры решетки КГ (1) были одинаковы для всех реализаций численного эксперимента ($N = 301$, $W = 4$, $E_0 = 0.4$).²

На рисунке В.1 приведено сравнение четырех конфигураций схемы SABA: $SABA_1$, $SABA_2$, $SABA_2C$ и $SABA_4$. По рисунку видно, что при малых шагах интегрирования $SABA_2C$ показывает наиболее точный результат: среднее значение ошибки по энергии менее, чем в седьмом знаке после запятой. Такой выигрыш мы получаем за счёт использования корректора. Но вместе с тем, такие вычисления требуют больше ресурсов. Посмотрим на следующем рисунке, какое время потребуется при счёте с использованием выбранных конфигураций SABA-схемы.

² Рисунок взят из [3].

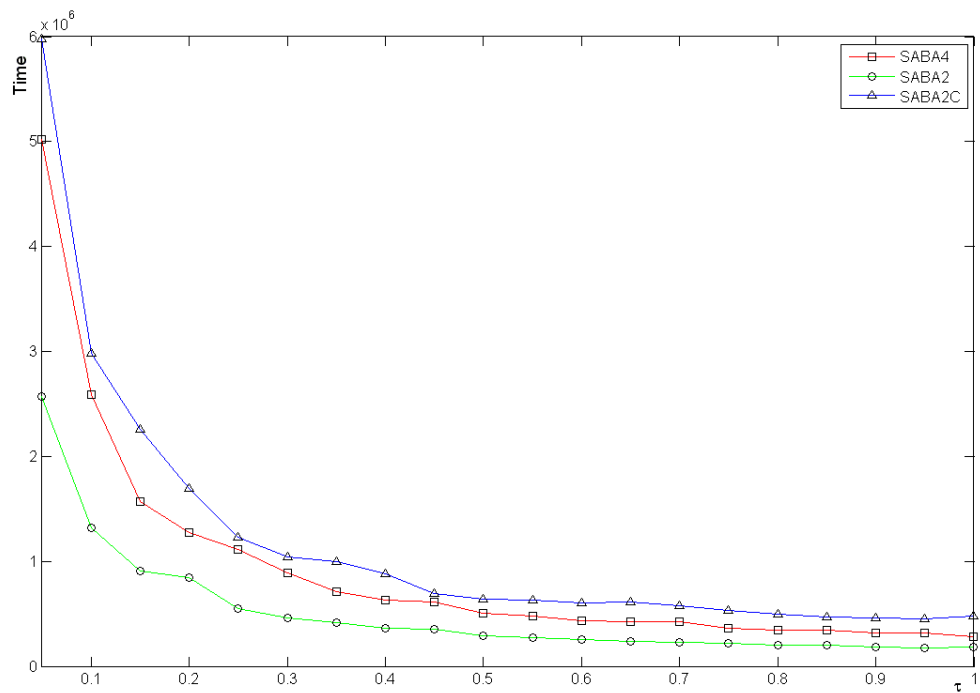


Рис. В.2. Сравнение времени работы SABA -схем интегрирования различного порядка при прочих одинаковых параметрах ($N = 301$, $W = 4$, $E_0 = 0.4$, $t_{\text{end}} = 10^7$).³

По рисунку В.2 видно, что время счёта для различных конфигураций *SABA* различно. Например, при сравнении *SABA₂C* и *SABA₂* видно, что стоимость вычислений *SABA₂C* больше стоимости вычислений *SABA₂* примерно в два раза. Причина такого замедления лежит во внутреннем устройстве этих двух схем: *SABA₂C* на каждом шаге интегрирования требует вычисления корректора, что является достаточно объёмной операцией.

Несмотря на то, что схема с корректором требует больше времени для проведения вычислений, она даёт нам возможность получить желаемую точность. Поэтому, остановим свой выбор на конфигурации *SABA₂C*.

³ Рисунок взят из [3].

С) Код основных вычислений

```
1. __device__ void LA(float tau, int ID, float *X0, float *TempPoint, float *EPS)
2. {
3.     TempPoint[ID * 2]      = X0[ID * 2] + tau * X0[ID * 2 + 1];
4.     TempPoint[ID * 2 + 1] = X0[ID * 2 + 1];
5. }
6. __device__ void LB(float tau, int ID, float *X0, float *TempPoint, float *EPS)
7. {
8.     float ulM1_ul_3 = 0, ulP1_ul_3 = 0, Q1;
9.     if (ID != 0)
10.        ulM1_ul_3 = pow(X0[(ID - 1) * 2] - X0[ID * 2], 3);
11.     else
12.        ulM1_ul_3 = 0;
13.     if (ID != N-1)
14.        ulP1_ul_3 = pow(X0[(ID + 1) * 2] - X0[ID * 2], 3);
15.     else
16.        ulP1_ul_3 = 0;
17.     Q1          = EPS[ID] * X0[ID * 2] - ulM1_ul_3 - ulP1_ul_3;
18.     TempPoint[ID * 2]      = X0[ID * 2];
19.     TempPoint[ID * 2 + 1] = X0[ID * 2 + 1] - tau * Q1;
20. }
21. __device__ void LC(float tau, int ID, float *X0, float *TempPoint, float *EPS)
22. {
23.     float ul_minus_1, el_minus_1, ul_minus_2, ulM1_ul_2, ulM1_ul_3, ulM2_ulM1_3,
24.     ul_plus_1, el_plus_1, ul_plus_2, ulP1_ul_2, ulP1_ul_3, ulP2_ulP1_3, ul, pl, el,
25.     Q1, Q1_minus_1, Q1_plus_1;
26.     ul = X0[ID * 2];
27.     pl = X0[ID * 2 + 1];
28.     el = EPS[ID];
29.     if (ID==0) {
30.         ul_minus_1 = 0;
31.         el_minus_1 = 0;
32.         ul_minus_2 = 0;
33.         ulM1_ul_2  = 0;
34.         ulM1_ul_3  = 0;
35.         ulM2_ulM1_3 = 0;
36.     }
37.     else if (ID==1){
38.         ul_minus_1 = X0[0];
39.         el_minus_1 = EPS[0];
40.         ul_minus_2 = 0;
41.         ulM1_ul_2  = pow(ul_minus_1 - ul, 2);
```

```

42.         ulM1_ul_3    = pow(ul_minus_1 - ul, 3);
43.         ulM2_ulM1_3 = 0;
44.     }
45.     else {
46.         ul_minus_1    = X0[(ID - 1) * 2];
47.         el_minus_1    = EPS[ID - 1];
48.         ul_minus_2    = X0[(ID - 2) * 2];
49.         ulM1_ul_2     = pow(ul_minus_1 - ul, 2);
50.         ulM1_ul_3     = pow(ul_minus_1 - ul, 3);
51.         ulM2_ulM1_3   = pow(ul_minus_2 - ul_minus_1, 3);
52.     }
53.     if (ID==(N-1)) {
54.         ul_plus_1     = 0;
55.         el_plus_1     = 0;
56.         ul_plus_2     = 0;
57.         ulP1_ul_2     = 0;
58.         ulP1_ul_3     = 0;
59.         ulP2_ulP1_3   = 0;
60.     }
61.     else if (ID==(N-2)) {
62.         ul_plus_1     = X0[(ID + 1) * 2];
63.         el_plus_1     = EPS[ID + 1];
64.         ul_plus_2     = 0;
65.         ulP1_ul_2     = pow(ul_plus_1 - ul, 2);
66.         ulP1_ul_3     = pow(ul_plus_1 - ul, 3);
67.         ulP2_ulP1_3   = 0;
68.     }
69.     else {
70.         ul_plus_1     = X0[(ID + 1) * 2];
71.         el_plus_1     = EPS[ID + 1];
72.         ul_plus_2     = X0[(ID + 2) * 2];
73.         ulP1_ul_2     = pow(ul_plus_1 - ul, 2);
74.         ulP1_ul_3     = pow(ul_plus_1 - ul, 3);
75.         ulP2_ulP1_3   = pow(ul_plus_2 - ul_plus_1, 3);
76.     }
77.     Q1                = el * ul      - ulM1_ul_3 - ulP1_ul_3;
78.     Q1_minus_1         = el_minus_1 * ul_minus_1 - ulM2_ulM1_3 + ulM1_ul_3;
79.     Q1_plus_1          = el_plus_1 * ul_plus_1 + ulP1_ul_3 - ulP2_ulP1_3;
80.     TempPoint[ID * 2]  = ul;
81.     TempPoint[ID * 2 + 1] = pl - 2 * tau * ( (el + 3 * ulM1_ul_2 + 3 * ulP1_ul_2 ) *
82.         * Q1 - 3 * Q1_minus_1 * ulM1_ul_2 - 3 * Q1_plus_1 * ulP1_ul_2 );
83. }

```