

УДК 004.057.7, 004.62, 004.63, 004.75

СИМУЛЯТОР КВАНТОВЫХ ВЫЧИСЛЕНИЙ FEYNMAN НА ЯЗЫКЕ MAPLE: ОСНОВНЫЕ ВОЗМОЖНОСТИ

**Воробьев Владимир Владимирович¹, Кореньков Владимир Васильевич²,
Гердт Владимир Петрович³**

¹Инженер-программист;

Объединенный институт ядерных исследований,

Лаборатория информационных технологий;

Россия, 141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6;

Ассистент;

ГБОУ ВО МО «Университет «Дубна»,

Институт системного анализа и управления;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;

e-mail: vlavor@jinr.ru.

²Директор лаборатории;

Объединенный институт ядерных исследований,

Лаборатория информационных технологий;

Россия, 141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6;

Доктор технических наук, профессор, заведующий кафедрой РИВС;

ГБОУ ВО МО «Университет «Дубна»,

Институт системного анализа и управления;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;

e-mail: korenkov@jinr.ru.

³Начальник сектора лаборатории;

Объединенный институт ядерных исследований,

Лаборатория информационных технологий;

Россия, 141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6;

Доктор физико-математических наук, профессор;

ГБОУ ВО МО «Университет «Дубна»,

Институт системного анализа и управления;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;

e-mail: gerdt@jinr.ru.

В статье представлен краткий обзор возможностей симулятора квантовых вычислений *Feynman* на языке системы компьютерной алгебры *Maple*. Он был разработан с целью реализации на высокоуровневом языке программирования пользователя системы, который способен работать с конечными n -кубитными системами в символьном и числовом видах. Данный инструмент позволяет работать с квантовыми состояниями, производить квантовые вычисления, реализовать квантовые алгоритмы. Пакет имеет графический интерфейс для наглядного представления квантовой информации в виде формул, схем и графиков. Описаны основные возможности инструментария *Feynman*, приведены примеры его использования для задач квантовых вычислений и квантового компьютеринга.

Ключевые слова: квантовые вычисления, квантовый компьютеринг, квантовые алгоритмы, симулятор квантовых вычислений, *Maple*, *Feynman*.

FEYNMAN QUANTUM COMPUTING SIMULATOR IN THE MAPLE LANGUAGE: MAIN FEATURES

Vorobyov Vladimir¹, Korenkov Vladimir², Gerdt Vladimir³

¹Software engineer;
Joint institute for nuclear researches,
Laboratory of Information Technologies;
Russia, 141980, Moscow reg., Dubna, 6 Joliot-Curie st.;
Assistant;
Dubna State University,
Institute of system analysis and management;
Russia, 141980, Moscow reg., Dubna, 19 Universitetskaya st.;
e-mail: vlavor@jinr.ru.

²Laboratory Director;
Joint institute for nuclear researches,
Laboratory of Information Technologies;
Russia, 141980, Moscow reg., Dubna, 6 Joliot-Curie st.;
Doctor of technical Sciences, Professor, head of the Department;
Dubna State University,
Institute of system analysis and management;
Russia, 141980, Moscow reg., Dubna, 19 Universitetskaya st.;
e-mail: korenkov@jinr.ru.

³ Head of the Laboratory sector;
Joint institute for nuclear researches,
Laboratory of Information Technologies;
Russia, 141980, Moscow reg., Dubna, 6 Joliot-Curie st.;
Doctor of physical and mathematical Sciences, Professor;
Dubna State University,
Institute of system analysis and management;
Russia, 141980, Moscow reg., Dubna, 19 Universitetskaya st.;
e-mail: gerdt@jinr.ru.

The article presents an overview of the capabilities of the Feynman quantum computing simulator in the language of computer algebra Maple. It was designed to implement a high-level computer language that is able to work with finite p -qubit systems in symbolic and numeric forms. This tool allows to work with quantum states, perform quantum computations, implement quantum algorithms. The package has a graphical interface for visual representation of quantum information in the form of formulas, diagrams and graphs. The main features of the Feynman toolkit are described, and examples of its use for problems of quantum computations and quantum computing are given.

Keywords: quantum computing, quantum computing, quantum algorithms, quantum computing simulator, Maple, Feynman.

Введение

За последние два десятилетия квантовые вычисления и квантовый компьютер развиваются стремительными темпами. Эта область исследования включает в себя разделы из физики, математики и информатики, которые в совокупности изучают свойства квантового мира. Интерес специалистов заключается в достижении более эффективных (квантовых) вычислений и коммуникаций с использованием неклассических состояний материи. В частности, в последние годы был реализован ряд новых протоколов и архитектур, таких как телепортация [1], безопасная связь [2], распределенная квантовая обработка [3] и многие другие. Это говорит о перспективности развития этого направления.

Однако, несмотря на большое количество успешных экспериментов, остается много открытых вопросов и (экспериментальных) проблем в отношении нашего (настоящего) понимания квантовой природы материи. Разными учеными была проведена большая алгебраическая и числовая работа, которая привела к появлению ряда сообществ по квантовым вычислениям, различающихся по методам и интересам. Их требования схожи: необходимо иметь вычислительный инструмент, который облегчает обработку вычислительных методов, но по форме входных и выходных данных максимально приближен к обозначениям, применяемым в литературе.

Хотя несколько таких инструментов и квантовых симуляторов уже существуют, как описано в литературе [4], [5] или доступны через Интернет, многие из них посвящены лишь небольшому числу задач и часто несовместимы с другими платформами и приложениями. Именно поэтому команда под руководством профессора из Германии С. Фрицше разработала универсальную библиотеку-инструмент на базе языка *Maple*. Особое внимание было уделено определению небольшого, но мощного набора команд как для символьных, так и для числовых вычислений. Инструмент достаточно гибкий, и может использоваться в других языках и в разных средах.

1. Общие требования к созданию симуляторов квантовых вычислений

Для реализации модели квантовой системы необходимо понимать, для каких типов задач она подходит и какие вычислительные требования необходимы для их решения. В таблице 1 перечислены несколько ключевых тем из квантовых вычислений и квантовой информатики вместе с их типичными задачами.

Таблица 1. Ключевые темы для изучения квантовых систем и состояний

Тема исследования в квантовых системах	Вычислительные требования и задачи
Декогеренция и рассеивание	Оценка квантовых операций для заданных квантовых состояний; карты между квантовыми операциями и квантовыми состояниями; вычисление супероператоров, матриц Чоя и операторов Крауса
Нелокальность	Оценка вероятностей и неравенств; вычисление (взаимных) информационных мер между парами квантовых состояний
Квантовые алгоритмы	Квантовая фазовая оценка; оценка квантового преобразования Фурье и Адамара и др.
Квантовые корреляции	Вычисление квантовых измерений различных видов
Квантовая криптография и распределение ключей	Эволюция шума в распределении квантовых ключей; поток информации в моделях квантового хранения; эволюция времени квантовой запутанности
Квантовая запутанность	Оптимизация квантовых мер по некоторому полному набору чистых состояний или матриц плотности; параметризация матриц различного вида
Квантовая коррекция ошибок	Построение стабилизаторов и групповых генераторов; проективные измерения в сложных основаниях; частое использование квантовых преобразований

Квантовые измерения	Генерация случайных состояний; квантовые вычисления на основе измерений; моделирование измерений состояния Белла и результатов измерений и т.д.
Квантовые сети и телепортация	Реализация протоколов телепортации в разных базисных наборах; анализ многопланового запутанного состояния; расчет мер расстояния
Квантовое моделирование	Декомпозиция гейтовых операций; Гамильтонова динамика во внешних полях; поведение захваченных ионов и спиновых цепей
Переход квантовых состояний	Преобразование несовершенного запутанного состояния в (меньшее число) максимально запутанного состояния; очищение запутывания
Квантовая томография состояния и процесса	Вычисление вероятностных распределений; протоколы для оценки квантового состояния

За прошедшие годы было проведено множество программных реализаций для решения различных задач. Они представляют собой рабочий лист программы, который демонстрирует эффективность работы различных квантовых алгоритмов или решают какую-то одну задачу из области квантовых вычислений. Тем не менее, несмотря на все усилия, ни одно программное средство не стало универсальным инструментом для работы с квантовыми состояниями и процессами. Причины разные: узкая направленность, малый вычислительный запрос, неудобство пользования, закрытый код. Кроме того, многие из этих программ допускают только численное моделирование, например *Qubit4Matlab* [6].

Подобно многим программным проектам, в 2005 г. команда под руководством С. Фрицше начала разрабатывать свой проект – инструмент *Feynman* для языка компьютерной алгебры *Maple* [7]. Цель состояла в том, чтобы помочь пользователю использовать методы квантовых вычислений для решения задач в символьном и числовом видах. Первоначально новое средство хорошо справлялось со своими задачами, но со временем росло количество процедур и функций, а архитектура библиотеки плохо масштабировалась. Поэтому в 2013 г. команда Фрицше сделала перезагрузку своего проекта *Feynman* [8], который теперь имеет огромный потенциал в масштабировании и претендует на звание универсального симулятора квантовых вычислений.

2. Симулятор квантовых вычислений *Feynman* (краткое описание)

В настоящее время симулятор *Feynman* реализован в рамках языка *Maple*. В этой среде компьютерной алгебры был определен (иерархический) набор процедур для предоставления структур данных, а также команд, которые работают с этими структурами. В частности, акцент был сделан на том, чтобы определить небольшой, но мощный набор команд и использовать базовую структуру данных (иногда называемую вспомогательными процедурами) для того, чтобы объединить соответствующую информацию в одно целое. В *Maple* эти вспомогательные процедуры просто используются в качестве «контейнеров» данных, так как они иным образом предоставляются таблицей записей или производными данными в других языках.

Основная цель реорганизации кода состояла в том, чтобы уменьшить общее количество команд, доступных для пользователя. Это было достигнуто с помощью общих процедур и путем объединения аналогичных функций и задач в одной команде. Все эти высокоуровневые команды обычно ссылаются на некоторый (неоднозначный) глагол, такой как вычислить, преобразовать, определить или оценить. Процедура ожидает «ключевое слово» в качестве своего первого аргумента, чтобы указать на конкретное действие. Этот простой и согласованный дизайн делает выполнение даже сложных задач достаточно описательным, а также облегчает реализацию кода, поскольку общие (основные) команды имеют дело только с обработкой и распределением входных данных. В таблице 2 приведены общие принципы построения архитектуры программного кода *Feynman*, на рисунках 1 и 2 – как это выглядит в системе (руководство для пользователя), на рисунке 3 – цифровой вид (код).

Таблица 2. Общие принципы построения программного кода симулятора *Feynman*

Принцип программирования	Краткое описание
Ключевое слово, строка	Используется для краткого описания действия каждой команды. С каждой цепочкой ключей связан один или несколько списков параметров, которые облегчают использование кода
Небольшой, но мощный набор команд	Процедуры с четким описанием и последовательным использованием параметров. Несмотря на некоторые вычислительные затраты при использовании таких общих имен, этот принцип улучшает читаемость кода и его интерактивное использование; это также облегчает обработку и обслуживание программы
Базовые структуры данных	Квантовые операции и квантовые регистры часто используются для реализации квантовых алгоритмов и протоколов. В симуляторе <i>Feynman</i> процедуры <code>qoperator()</code> и <code>qregister()</code> используются – наряду с несколькими другими – для хранения соответствующей информации и для облегчения ввода и вывода многих команд
Хорошая читаемость кода	Часто кажется более важной для длительного жизненного цикла и успеха программного кода, чем только одна эффективность
Открытый программный код	Позволяет настроить симулятор или расширить его функционал (масштабируемость)

```
> Feynman_define();
```

```
The following 'keystings' are supported by this command:
```

```
-----
a) Linear combinations & mixtures::      state          mixture
b) Diagonal & string states::           string
c) Initialization & random::            equal          Walsh-Hadamard    random
d) Predefined states::                  Bell: Phi+      Bell: Phi-        Bell: Psi+
                                           Bell: Psi-      Bell              Bell diagonal
                                           biseparable    Dicke             GHZ
                                           ground state   Ishizaka          IH
                                           isotropic      Smolin            thermal
                                           W              Werner
e) Maximally mixed & entangled states::  maximally entangled  maximally mixed  MEMS
f) Graph states::                       graph            graph: chain     graph: ring
```

Рис. 1. Справочная информация для команды *Feynman_define()*

```
> Feynman_compute();
```

```
The following 'keystings' are supported by this command:
```

```
-----
a) Adjoint states, etc. ::               adjoint
b) Eigenvectors & values::               eigenvalues      eigenvectors      eigenvectors: normalize
c) Norm & normalized states::           norm            trace norm        Hilbert--Schmidt norm
                                           HS norm         normalized state
d) Flatten & unflatten::                flattened vector (unflattened) matrix matrix
e) Ranks::                              rank
f) Traces & partial traces::           trace           partial trace     partial trace: subsystems
g) Expectation values::                 expectation value variance
h) Matrix transpose, etc.::             transpose        transpose: partial transpose: column
                                           transpose: raw
i) Miscellaneous::                      simplify        sqrt              sqrt: Denman
                                           correlation tensor
```

Рис. 2. Справочная информация для команды *Feynman_compute()*

```
#
Feynman_define := proc(keystring) ## export
#
# Defines (and returns) a quantum registers in various given or predefined states as specified by keystring.
#
local i, keys, wa, wx1, wx2, wx3, wx4;
#
keys := [ "a) Linear combinations & mixtures::",      "state",      "mixture",      " ",
          "b) Diagonal & string states::",          "string",     " ",           " ",
          "c) Initialization & random::",          "equal",     "Walsh-Hadamard", "random",
          "d) Predefined states::",                "Bell: Phi+", "Bell: Phi-",  "Bell: Psi+",
          " ",                                       "Bell: Psi-", "Bell",        "Bell diagonal",
          " ",                                       "biseparable", "Dicke",       "GHZ",
          " ",                                       "ground state", "Ishizaka",   "IH",
          " ",                                       "isotropic",  "Smolin",     "thermal",
          " ",                                       "W",          "Werner",     " ",
          "e) Maximally mixed & entangled states::", "maximally entangled", "maximally mixed", "MEMS",
          "f) Graph states::",                    "graph",     "graph: chain", "graph: ring" ];
#
if nargs = 0 then
print(NULL);
printf("%s\n", "The following `keystrings' are supported by this command:");

```

Рис. 3. Заголовок команды *Feynman_define* ()

3. Примеры использования симулятора *Feynman*

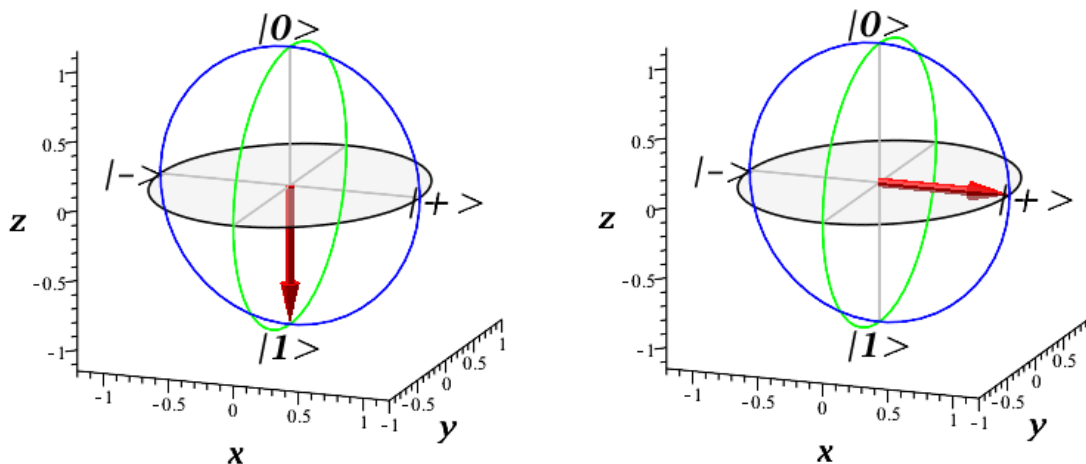
От краткого описания переходим к практике. В данном разделе будут рассмотрены простые примеры использования команд симулятора *Feynman*. Процедура *Feynman_plot* () будет наглядно демонстрировать результаты работы рабочего листа программы в *Maple* 17.

3.1. Векторы состояний

Код программы:

```
wa:= qbit(0,1): Feynman_plot("Bloch vector",wa);
wb:= Feynman_define("string","+"):Feynman_plot("Bloch vector", wb,1);
```

Результаты работы программы:



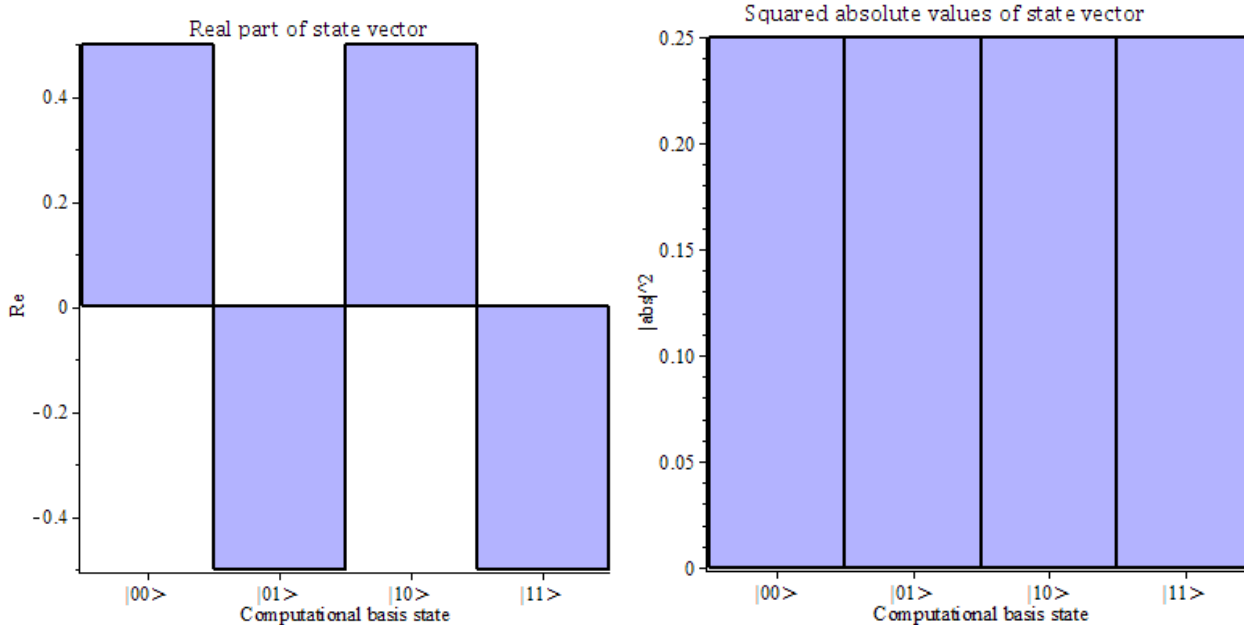
3.2. Матрицы плотности

Код программы (продолжение):

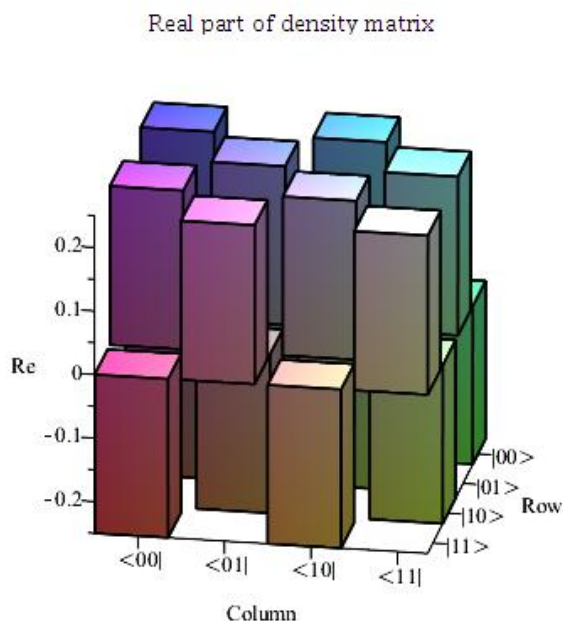
```
Feynman_plot("state: Re", wb); Feynman_plot("probability", wb);
wc:=Feynman_convert("ket-to-rho",wb):Feynman_print(wc);
```

```
Feynman_plot("state: Re", wc);
```

Результаты работы программы:



$$\begin{aligned} & \frac{1}{4} |00\rangle\langle 00| - \frac{1}{4} |00\rangle\langle 01| + \frac{1}{4} |00\rangle\langle 10| - \frac{1}{4} |00\rangle\langle 11| - \frac{1}{4} |01\rangle\langle 00| \\ & + \frac{1}{4} |01\rangle\langle 01| - \frac{1}{4} |01\rangle\langle 10| + \frac{1}{4} |01\rangle\langle 11| + \frac{1}{4} |10\rangle\langle 00| - \frac{1}{4} |10\rangle\langle 01| \\ & + \frac{1}{4} |10\rangle\langle 10| - \frac{1}{4} |10\rangle\langle 11| - \frac{1}{4} |11\rangle\langle 00| + \frac{1}{4} |11\rangle\langle 01| - \frac{1}{4} |11\rangle\langle 10| \\ & + \frac{1}{4} |11\rangle\langle 11| \end{aligned}$$

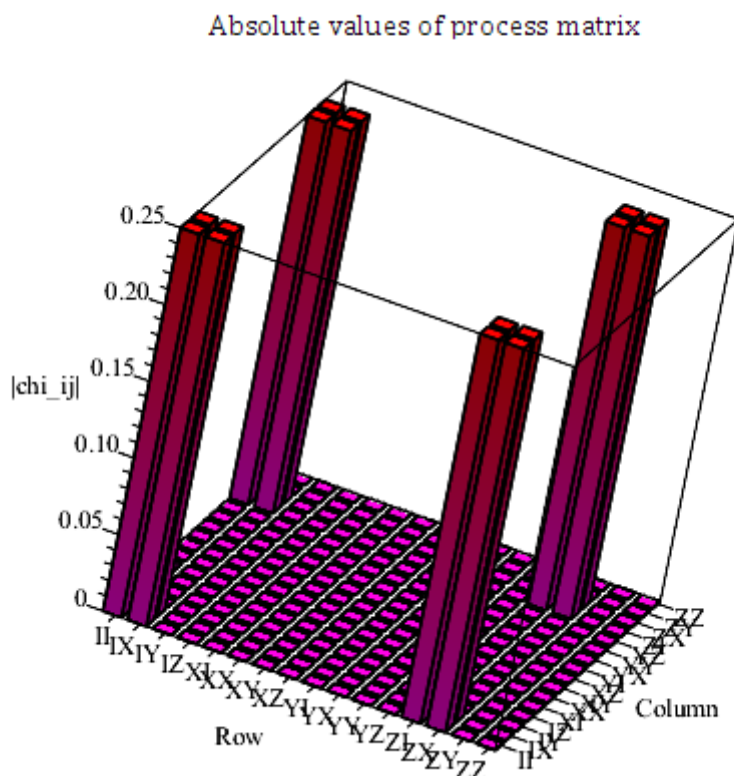


3.3. Квантовая томография данных

Код программы (продолжение):

```
wa:= qoperator("cn"): Feynman_plot("process matrix", wa);
```

Результаты работы программы:



Заключение

В данной статье было представлено краткое описание квантового симулятора *Feynman* на языке компьютерной алгебры *Maple*. Инструмент способен работать с n -кубитными системами и производить квантовые вычисления в удобном символьном, числовом и графическом видах. Код проекта открытый, что позволяет пользователям расширять свои функциональные возможности и добавлять новые процедуры. Данный симулятор может стать универсальным инструментом для моделирования квантовых систем, а также для обучения квантовым вычислениям и квантовому компьютерингу.

Список литературы

1. С.Н. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, W.K. Wootters Phys. Rev. Lett., 70, 1993. – [Электронный ресурс]. URL: <https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.70.1895>.
2. L.-M. Duan, M.D. Lukin, J.I. Cirac, P. Zoller Nature, 414 – 2001, p. 413. – [Электронный ресурс]. URL: <https://media.nature.com/original/nature-assets/nature/journal/v414/n6862/extref/35106513.pdf>.
3. Lim Y.L., Beige A., Kwek L.C. Phys. Rev. Lett., 95. – 2005. – P. 030505. – [Электронный ресурс]. URL: <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.95.030505>.
4. Toth Comput G. Phys. Comm., 179. – 2008. – P. 430. – [Электронный ресурс]. URL: https://ac.els-cdn.com/S0010465508001367/1-s2.0-S0010465508001367-main.pdf?_tid=191b18cc-95f0-4c35-bf9a-635b4d242dd6&acdnat=1550816964_b38f286068a79bde2ead4a4befcdc241.
5. Tabakin F., Julia-Diaz B. Comput. Phys. Commun., 182. – 2011. – P. 1693. – [Электронный ресурс]. URL: https://ac.els-cdn.com/S0010465511001287/1-s2.0-S0010465511001287-main.pdf?_tid=bd71e110-9476-48da-8ad8-f47063e08d3a&acdnat=1550817021_98703320e6c5cfc29f82bcdf0b71a36e.
6. Toth G. Comput. Phys. Comm., 179. – 2008. – P. 430. – [Электронный ресурс]. URL: https://ac.els-cdn.com/S0010465508001367/1-s2.0-S0010465508001367-main.pdf?_tid=34e00cb2-f4f5-4866-863f-775676d4de3a&acdnat=1550817091_afca25a0d1c4518320cb53dd31539455.

7. Radtke T., Fritzsche S. *Comput. Phys. Commun.*, 173. – 2005. – P. 91. – [Электронный ресурс]. URL: https://ac.els-cdn.com/S0010465505004261/1-s2.0-S0010465505004261-main.pdf?_tid=1d553525-6019-4d34-b14f-ff22d814475d&acdnat=1550817166_1c27c2eb30d0a9264d27ffcc8068c3bb.
8. Radtke T., Fritzsche S. *Comput. Phys. Commun.*, 185. – 2014. – P. 1617. – [Электронный ресурс]. URL: https://ac.els-cdn.com/S0010465514000356/1-s2.0-S0010465514000356-main.pdf?_tid=d2a3fe62-6351-4772-b78e-c13e2fca813d&acdnat=1550816160_917098b0a2a1eef820ebced57f80ce02.