

Федеральное государственное учреждение «Федеральный научный центр  
Научно-исследовательский институт системных исследований  
Российской академии наук»  
(ФГУ ФНЦ НИИСИ РАН)

## **ТРУДЫ НИИСИ РАН**

ТОМ 10 № 3

**МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ  
МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ:**

**ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ**

МОСКВА  
2020

**Редакционный совет ФГУ ФНЦ НИИСИ РАН:**

В.Б. Бетелин (председатель),  
Е.П. Велихов, С.Е. Власов, В.А. Галатенко, В.Б. Демидович (отв. секретарь),  
Ю.В. Кузнецов (отв. секретарь), Б.В. Крыжановский, А.Г. Кушниренко,  
А.Г. Мадера, М.В. Михайлюк, В.Я. Панченко, В.П. Платонов, В.Н. Решетников

**Главный редактор журнала:**

В.Б. Бетелин

**Научный редактор номера:**

М.В. Михайлюк

**Тематика номера:**

Технологические проблемы плазмохимического травления микро- и наноструктур, исследование физических процессов и их моделирование, информационные и компьютерные технологии, информационные технологии в экономике

Журнал публикует оригинальные статьи по следующим областям исследований: математическое и компьютерное моделирование, обработка изображений, визуализация, системный анализ, методы обработки сигналов, информационная безопасность, информационные технологии, высокопроизводительные вычисления, оптико-нейронные технологии, микро- и нанoeлектроника, математические исследования и вопросы численного анализа, история науки и техники.

**The topic of the issue:**

Technological problems of plasma-chemical etching processes for micro- and nanostructuring, Research and modeling of physical processes, Information and computer technologies, Information technologies in economy

The Journal publishes novel articles on the following research areas: mathematical and computer modeling, image processing, visualization, system analysis, signal processing, information security, information technologies, high-performance computing, optical-neural technologies, micro- and nanoelectronics, mathematical researches and problems of numerical analysis, history of science and of technique.

Заведующий редакцией: В.Е. Текунов

Издатель: ФГУ ФНЦ НИИСИ РАН,  
117218, Москва, Нахимовский проспект 36, к. 1

## СОДЕРЖАНИЕ

### I. ТЕХНОЛОГИЧЕСКИЕ ПРОБЛЕМЫ ПЛАЗМОХИМИЧЕСКОГО ТРАВЛЕНИЯ МИКРО- И НАНОСТРУКТУР

*Ефремов А. М., Смирнов С. А., Рыбкин В.В., Бетелин В. Б.* Особенности кинетики реактивно-ионного травления Si и SiO<sub>2</sub> в смеси Cl<sub>2</sub>+O<sub>2</sub> .....4

### II. ИССЛЕДОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ И ИХ МОДЕЛИРОВАНИЕ

*Б.В. Крыжановский, Л.Б. Литинский* Собственные значения гамильтониана Изинга с дальним действием.....11

*Б.В. Крыжановский, Л.Б. Литинский* Спектр матрицы связи модели Изинга с произвольным дальним действием.....23

*А.В. Мальцев, Л.А. Финагин* Методы моделирования параметров и фокусировки камеры в виртуальном окружении.....29

*К.Д. Ашмян, С.Г. Вольтин, О.В. Ковалева, Г.А. Ковалева* Создание системы управления с обратной связью при разработке нефтяных месторождений .....34

*Ф.А. Юдкин, А.В. Голубкин* Параметрическая компенсация тепловых сдвигов в сенсорах на основе поверхностного плазмонного резонанса.....39

### III. ИНФОРМАЦИОННЫЕ И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

*М.Г. Фуругян* Составление многопроцессорного расписания в системе с двумя типами ресурсов.....44

*А.А. Бурцев* Особенности программирования троичной машины: новые возможности и новые задачи.....49

*А.А. Бурцев, В.А. Бурцев* О преимуществах троичных машин и эффективности троичных вычислений.....61

*Д.Б. Аглямутдинова, Н.О. Бесшапошников, А.Г. Леонов, К.А. Мащенко, А.Е. Орловский* Механизмы универсального подключения программных исполнителей к системе ПиктоМир.....66

### IV. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ

*Г.Е. Каратаева* Применение технологий бережливого производства в сфере нематериальных услуг .....72

# Особенности кинетики реактивно-ионного травления Si и SiO<sub>2</sub> в смеси Cl<sub>2</sub>+O<sub>2</sub>

<sup>1,2)</sup>Ефремов А. М., <sup>1,2)</sup>Смирнов С. А., <sup>1,2)</sup>Рыбкин В.В., <sup>2)</sup>Бетелин В. Б.

<sup>1)</sup> ФГБОУ ВО «Ивановский государственный химико-технологический университет»,

[amefremov@mail.ru](mailto:amefremov@mail.ru), [sas@isuct.ru](mailto:sas@isuct.ru), [rybkin@isuct.ru](mailto:rybkin@isuct.ru)

<sup>2)</sup> ФГУ ФНЦ НИИСИ РАН

**Аннотация:** Проведено исследование кинетики и механизмов реактивно-ионного травления Si и SiO<sub>2</sub> в плазме смеси Cl<sub>2</sub>+O<sub>2</sub> в условиях индукционного ВЧ (13.56 МГц) разряда. При совместном использовании методов диагностики и моделирования плазмы выявлены механизмы влияния начального состава смеси на стационарные электрофизические параметры и состав газовой фазы. Установлено, что увеличение доли кислорода в плазмообразующем газе приводит к резкому снижению как скоростей травления обоих материалов, так и кинетических коэффициентов, характеризующих гетерогенное взаимодействие – эффективной вероятности и выхода травления. Показано, что причиной последнего эффекта может являться снижение доли свободных центров адсорбции для атомов хлора из-за окисления продуктов травления в менее летучие соединения.

**Ключевые слова:** хлор, кислород, плазма, константа скорости, скорость реакции, концентрация, вероятность взаимодействия.

## 1. Введение

Низкотемпературная плазма галогенсодержащих газов находит широкое применение в технологии интегральной электроники для структурирования (размерного травления) различных неорганических материалов [1–3]. В частности, для процессов реактивно-ионного травления (РИТ) кремния и его соединений наибольшее распространение получила плазма фторуглеродных (C<sub>x</sub>H<sub>y</sub>F<sub>z</sub>) газов и смесей на их основе. Широкая номенклатура таких газов обеспечивает эффективную оптимизацию технологических параметров РИТ (скорость, селективность по отношению к материалу маски) за счет регулирования баланса скоростей травления и поверхностной полимеризации [4, 5]. Основным недостатком РИТ в среде фторсодержащих газов является близкий к изотропному профиль травления кремния, обусловленный спонтанной химической реакцией  $Si + xF \rightarrow SiF_x$  [2–4]. Таким образом, поиск альтернативных плазмообразующих сред для РИТ кремния и его соединений является важной задачей, направленной на совершенствование как самих плазмохимических технологий, так и функциональных характеристик конечных изделий.

В настоящее время существует ряд работ, посвященных исследованиям процессов РИТ Si и SiO<sub>2</sub> в плазме хлорсодержащих газов, в частности – в смесях на основе Cl<sub>2</sub> [6–9]. Результаты этих исследований позволяют сформулировать два основных отличия от процессов, основанных на химии атомов фтора. Во-первых, химическая реакция  $Si + xCl \rightarrow SiCl_x$  протекает

по ионно-стимулированному механизму из-за низкой летучести продуктов взаимодействия. Причиной этого является большой размер атома хлора, который затрудняет его проникновение в решетку кремния, при этом частично хлорированные атомы кремния сохраняют связи с ближайшими соседями. Соответственно, характерные значения скоростей травления (~ 50 нм/мин) существенно ниже по сравнению со фторсодержащей плазмой (~ 200 нм/мин) [1–3]. И, во-вторых, добавка кислорода к активному газу приводит к снижению скорости травления, но к увеличению анизотропии процесса. Эти эффекты обычно относят к образованию на обрабатываемой поверхности соединений вида SiCl<sub>x</sub>O<sub>y</sub>, обладающих еще меньшей летучестью (более высокой пороговой энергией десорбции) по сравнению с SiCl<sub>x</sub>.

Необходимо отметить, что подавляющее большинство упомянутых работ имеют экспериментальную природу и, таким образом, не обеспечивают анализа наблюдаемых эффектов во взаимосвязи с параметрами газовой фазы. Такая ситуация не обеспечивает корректного понимания как механизмов гетерогенного взаимодействия, так и общих принципов регулирования выходных характеристик травления при варьировании условий обработки (давление газа, вкладываемая мощность, начальный состав смеси). В наших предшествующих работах [10–12] было показано, что совместное использование методов диагностики и моделирования плазмы является эффективным инструментом анализа механизмов реактивно-ионных процессов, в том числе – в плазмообразующих смесях

на основе  $\text{Cl}_2$ . В частности, при исследовании травления кремния в трехкомпонентной системе  $\text{HBr} + \text{Cl}_2 + \text{O}_2$  [13–15] было найдено, что 1) при бомбардировке поверхности ионами с энергиями  $\sim 300$  эВ кинетика травления соответствует ионно-стимулированной химической реакции в режиме лимитирования потоком нейтральных частиц; и 2) содержание кислорода в плазмообразующей смеси оказывает существенное влияние на параметры газовой фазы и кинетические коэффициенты гетерогенного взаимодействия – эффективную вероятность и выход травления.

Основная идея данной работы заключалась в использовании подхода, разработанного в работах [10–15], для исследования кинетики реактивно-ионного травления Si и  $\text{SiO}_2$  в плазме бинарной смеси  $\text{Cl}_2 + \text{O}_2$ . В качестве основных целей исследования выступали 1) анализ механизмов влияния начального состава смеси на электрофизические характеристики плазмы и концентрации активных частиц; 2) установление взаимосвязей между параметрами газовой фазы и кинетическими характеристиками гетерогенного взаимодействия; и 3) проверка гипотезы об изменении состава продуктов травления при варьировании содержания кислорода в смеси.

## 2. Методическая часть

Эксперименты проводились в плазмохимическом реакторе планарного типа (в верхнем расположением плоского индуктора), конструкция которого подробно описана в наших работах [10–15]. Плазма возбуждалась на частоте 13.56 МГц. В качестве постоянных параметров эксперимента выступали давление газа ( $p = 6$  мтор), общий расход газа ( $q = 40$  см<sup>3</sup>/мин при станд. усл.), вкладываемая мощность ( $W = 700$  Вт) и мощность смещения на нижнем электроде ( $W_{dc} = 200$  Вт). Смещение обеспечивалось независимым ВЧ (12.56 МГц) генератором и задавало энергию бомбардирующих ионов  $\varepsilon_i$  через величину напряжения смещения  $-U_{dc}$ . В качестве варьируемого параметра использовался начальный состав плазмообразующей смеси  $\text{Cl}_2 + \text{O}_2$ , регулирование которого осуществлялось изменением парциальных расходов газовой смеси. Так, варьирование  $q_{\text{O}_2}$  в диапазоне 0–30 см<sup>3</sup>/мин обеспечивало увеличение доли кислорода в смеси  $y_{\text{O}_2} = q_{\text{O}_2}/q$  до 75%.

Для диагностики плазмы использовался двойной зонд Лангмюра (DLP2000, Plasmart Inc.). Обработка зондовых вольт-амперных характеристик с использованием известных положений теории двойного зонда [16] обеспечивала данные по температуре электронов ( $T_e$ ) и плотности ионного тока ( $J_+$ ).

Измерение скоростей травления Si и  $\text{SiO}_2$  проводили на фрагментах неокисленных и термически окисленных кремниевых пластин размером  $\sim 4$  см<sup>2</sup>, которые располагали в центре нижнего электрода. Малый размер образцов позволял исключить загрузочный эффект и, таким образом, проводить процесс травления в кинетическом режиме. Температуру образца стабилизировали на уровне  $\sim 17$  °С. Скорости травления определяли как  $R = \Delta h/\tau$ , где  $\Delta h$  – высота ступеньки травления на границе маскированной и немаскированной областей, измеренная с помощью профилометра Alpha-Step 500 (Tencor), и  $\tau = 2$  мин – время обработки. Близкий к линейному характер зависимости  $\Delta h = f(\tau)$  свидетельствовал о стационарном режиме травления. Для маскирования использовали фоторезист AZ1512 толщиной  $\sim 1.5$  мкм.

Для определения электрофизических параметров и состава плазмы использовалась 0-мерная кинетическая модель с маквелловской аппроксимацией функции распределения электронов по энергиям [17, 18]. Алгоритм моделирования базировался на совместном решении уравнений химической кинетики для нейтральных и заряженных частиц в квазистационарном приближении. Принципы создания таких моделей и вопросы их применимости для описания плазмы низкого ( $p < 50$  мтор) давления (в том числе – в  $\text{Cl}_2$  и  $\text{O}_2$ ) подробно рассмотрены в предшествующих работах [19, 20]. Кинетическая схема (набор реакций и соответствующих констант скоростей, см. Табл. 1) была заимствована из наших работ по исследованию параметров и состава плазмы  $\text{HBr} + \text{Cl}_2 + \text{O}_2$  [13–15]. Входными параметрами модели служили экспериментальные данные по  $T_e$  и  $J_+$ . В качестве выходных параметров выступали средние по объему плазмы концентрации нейтральных и заряженных (электроны, положительные и отрицательные ионы) частиц, а также плотности их потоков на поверхность обрабатываемого материала.

Основные реакции с участием нейтральных частиц в плазме смеси  $\text{Cl}_2 + \text{O}_2$

Таблица 1

Реакция	$k$	Реакция	$k$
1. $\text{Cl}_2 + e \rightarrow 2\text{Cl} + e$	$f(T_e)$	10. $2\text{ClO} \rightarrow \text{Cl}_2 + \text{O}_2$	$1.5 \times 10^{-13}$

2.	$O_2 + e \rightarrow 2O + e$	$f(T_e)$	11.	$Cl \rightarrow Cl_{ads}$	$f(\gamma), \gamma = 0.05$
3.	$O_2 + e \rightarrow O + O(^1D) + e$	$f(T_e)$		$Cl_{ads} + Cl \rightarrow Cl_2$	
4.	$O + e \rightarrow O(^1D) + e$	$f(T_e)$		$Cl_{ads} + O \rightarrow ClO$	
5.	$ClO + e \rightarrow Cl + O + e$	$f(T_e)$	12.	$O \rightarrow O_{ads}$	$f(\gamma), \gamma = 0.1$
6.	$Cl_2 + O \rightarrow ClO + Cl$	$3.0 \times 10^{-13}$		$O_{ads} + O \rightarrow O_2$	
7.	$Cl_2 + O(^1D) \rightarrow ClO + Cl$	$3.6 \times 10^{-11}$		$O_{ads} + Cl \rightarrow ClO$	
8.	$ClO + O \rightarrow O_2 + Cl$	$7.0 \times 10^{-11}$	13.	$O(^1D) \rightarrow O$	$f(\gamma), \gamma = 1$
9.	$ClO + O(^1D) \rightarrow O_2 + Cl$	$1.0 \times 10^{-10}$			

Примечание: константы скоростей R6–R10 в  $см^3/с$

### 3. Результаты и обсуждение

Эксперименты показали, что увеличение доли кислорода в смеси сопровождается монотонным снижением скоростей травления Si (49–8 нм/мин, или в ~ 6 раз при 0–75%  $O_2$ ) и  $SiO_2$  (16–5.3 нм/мин, или в ~ 3 раза при 0–75%  $O_2$ ), при этом характер обеих зависимостей отличается от простого эффекта разбавления (рис. 1(а)). Отметим также, что селективность травления в системе Si/ $SiO_2$  снижается в диапазоне 3.1–1.5 при  $y_{O_2} = 0–50\%$ , а затем сохраняет постоянное значение вплоть до 75%  $O_2$ .

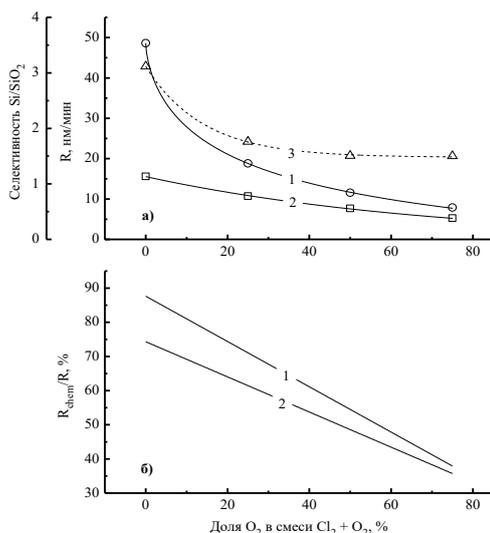


Рис. 1. Абсолютные скорости (а) и вклад химической составляющей в общую скорость травления (б): 1–Si; 2– $SiO_2$ ; 3– селективность Si/ $SiO_2$ .

Из литературы [7, 21] известно, что суммарная скорость реактивно-ионного процесса может быть представлена как  $R = R_{phys} + R_{chem}$ , где  $R_{phys}$  – скорость распыления химически не модифицированных атомов поверхности ионной бомбардировкой, и  $R_{chem}$  – скорость ионно-стимулированной химической реакции. Очевидно, что ненулевые скорости травления в плазме чистого  $O_2$  (~ 5 нм/мин для Si и ~ 3.5 нм/мин для  $SiO_2$ ) не могут быть обусловлены химическим механизмом и представляют собой  $R_{phys}$ . Корректировка этих величин с учетом

изменения энергии бомбардирующих ионов (294–301 эВ при 0–75%  $O_2$ ) и последующий расчет  $R_{chem} = R - R_{phys}$  позволяет заключить, что вплоть до 50%  $O_2$  доминирующим механизмом травления обоих материалов является ионно-стимулированная химическая реакция (Рис. 1(б)). Таким образом, именно изменение  $R_{chem}$  определяет наблюдаемый характер зависимостей скоростей травления Si и  $SiO_2$  от начального состава смеси.

При анализе кинетики ионно-стимулированных химических реакций в бинарных газовых смесях вопросом первостепенной важности является влияние начального состава смеси на параметры электронной (температура электронов, концентрация электронов) и ионной (плотность потока ионов, энергия ионной бомбардировки) компонент газовой фазы. Очевидно, что первая составляющая определяет скорости образования активных частиц в объеме плазмы, а вторая – эффективность гетерогенных стадий, инициируемых ионной бомбардировкой. При диагностике плазмы было найдено, что увеличение  $y_{O_2}$  сопровождается ростом температуры электронов ( $T_e = 3.0–3.4$  эВ при 0–75%  $O_2$ ), но снижением величин  $n_+$  и  $n_e$  ( $1.2 \times 10^{11}–8.8 \times 10^{10} \text{ см}^{-3}$  и  $8.9 \times 10^{10}–7.4 \times 10^{10} \text{ см}^{-3}$ , соответственно, при 0–75%  $O_2$ ) (рис. 2(а)). Причиной первого эффекта являются более высокие энергии электронов при столкновениях с молекулами  $Cl_2$ , которые отличаются низкороговым электронным возбуждением ( $\epsilon_{th} \sim 2.5$  эВ для  $Cl_2 + e \rightarrow Cl_2^*(B^3\Pi) + e$ ) и ионизацией ( $\epsilon_{th} \sim 11.5$  эВ для  $Cl_2 + e \rightarrow Cl_2^+ + 2e$ ) с высокими (~  $10^{16} \text{ см}^2$ ) сечениями соответствующих процессов [22]. Изменение концентраций электронов и положительных ионов обусловлено снижением суммарной частоты ионизации ( $\nu_{iz} = 1.4 \times 10^5–7.8 \times 10^4 \text{ с}^{-1}$  при 0–75%  $O_2$ ). Так как величина  $\nu_{iz}$  однозначно определяется кинетикой ионизации атомов и молекул хлора, ее поведение следует изменению концентраций этих частиц (рис. 3). Из рис. 2(б) можно видеть также, что добавление кислорода приводит к незначительному росту отрицательного смещения на нижнем электроде ( $-U_{dc} = 275–281$  В при 0–75%  $O_2$ ), но к снижению параметра

$\sqrt{M_i \varepsilon_i} \Gamma_+$ , характеризующего эффективность ионной бомбардировки. Это обеспечивается уменьшением массы бомбардирующих ионов  $M_i$  в условиях  $\Gamma_+ \approx \text{const}$ .

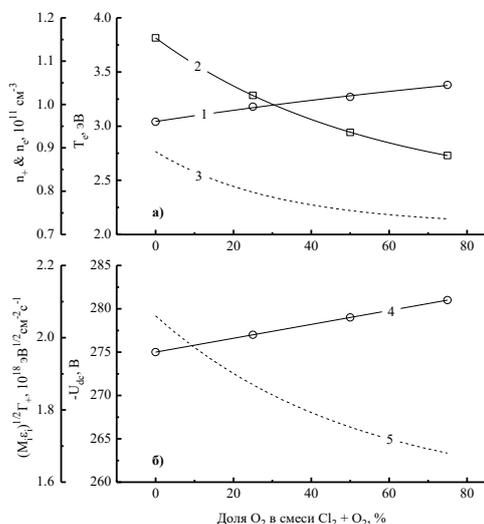


Рис. 2. Электрофизические параметры плазмы в смеси  $\text{Cl}_2 + \text{O}_2$ : 1 – температура электронов; 2 – суммарная концентрация положительных ионов; 3 – концентрация электронов; 4 – напряжения смещения на нижнем электроде; и 5 – параметр  $\sqrt{M_i \varepsilon_i} \Gamma_+$ , характеризующий эффективность ионной бомбардировки.

При анализе кинетики нейтральных частиц было найдено, что характерной особенностью плазмы  $\text{Cl}_2$  в исследованном диапазоне условий являются высокие степени диссоциации исходных молекул, обеспечивающие  $n_{\text{Cl}} > n_{\text{Cl}_2}$  (рис. 3). Причиной этого является высокая константа скорости R1 ( $k_1 \sim 1.2 \times 10^{-8} \text{ см}^3/\text{с}$  при  $T_e = 2.5 \text{ эВ}$ ), обусловленная низкой пороговой энергией ( $\varepsilon_{th} \sim 3 \text{ эВ}$ ) и высоким сечением этого процесса. Увеличение доли  $\text{O}_2$  в смеси сопровождается незначительным ростом частоты диссоциирующих столкновений электронов с молекулами  $\text{Cl}_2$  ( $k_1 n_e = 962\text{--}996 \text{ с}^{-1}$  при 0–75%  $\text{O}_2$ ), но приводит к резкому снижению скорости R1 (в  $\sim 7$  раз при 0–75%  $\text{O}_2$ , см. рис. 3) из-за аналогичного изменения  $n_{\text{Cl}_2}$ . Последний эффект обусловлен быстрым расходом этих частиц в процессах R6 и R7. В то же время, максимальный вклад R6 и R7 в общую скорость генерации атомов хлора не превышает 10% в области  $y_{\text{O}_2} > 50\%$ . Причина этого связана с низкими скоростями образования O и  $\text{O}(\text{D})$  по R2 и R3 (из-за  $k_2 + k_3 \ll k_1$ ), а также с высокими скоростями гетерогенной рекомбинации атомов кислорода в R11–R13. Значительно большей эффективностью с точки зрения образования атомов хлора обладают реакции с участием молекул  $\text{ClO}$  – R5, R8 и R9. Основным

источником этих частиц являются гетерогенные процессы вида  $\text{Cl} + \text{O} \rightarrow \text{ClO}$  в составе R11 и R12, на которые расходуется более 50% атомов кислорода, образующихся по R2 и R3. Кроме этого, из Табл. 1 можно видеть, что константы скоростей R8 и R9 на порядок величины превышают значения для аналогичных процессов с участием  $\text{Cl}_2$ . В результате, суммарный вклад  $R_5 + R_8 + R_9$  в общую скорость образования атомов хлора достигает 50% при 75%  $\text{O}_2$ . В то же время, противоположное действие на величину  $n_{\text{Cl}}$  оказывает рост частоты их гетерогенной рекомбинации из-за  $k_{12} > k_{11}$ . Поэтому результирующее изменение концентрации атомов хлора (в  $\sim 4$  раза при 0–75%  $\text{O}_2$ ) соответствует изменению начального состава смеси.

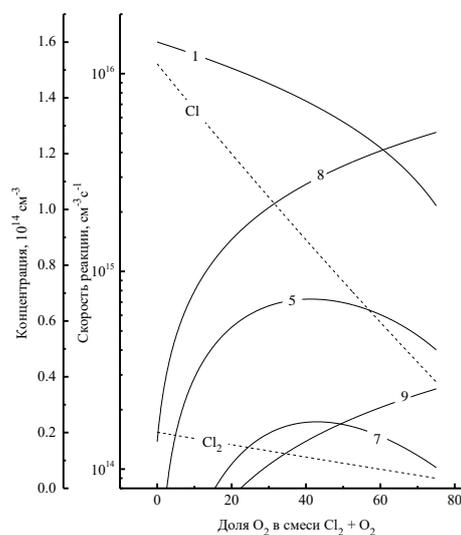


Рис. 3. Скорости образования атомов хлора (сплошные линии) и концентрации хлорсодержащих частиц (пунктир). Цифры на кривых соответствуют номерам реакций в Табл. 1.

Сравнение данных рис. 1 и 3 позволяет заключить, что снижение  $R_{chem}$  при изменении начального состава смеси (в  $\sim 14$  раз для Si и в  $\sim 6$  раз для  $\text{SiO}_2$  при 0–75%  $\text{O}_2$ ) является более резким по сравнению с изменением плотности потока атомов хлора  $\Gamma_{\text{Cl}} \approx 0.25 n_{\text{Cl}} v_T$ , где  $v_T$  – скорость теплового движения атомов в условиях малых изменений температуры газа. Очевидно, что такая ситуация соответствует уменьшению эффективной вероятности взаимодействия атомов хлора с обрабатываемой поверхностью  $\gamma_R = R_{chem} / \Gamma_{\text{Cl}}$ , как показано на рис. 4(a).

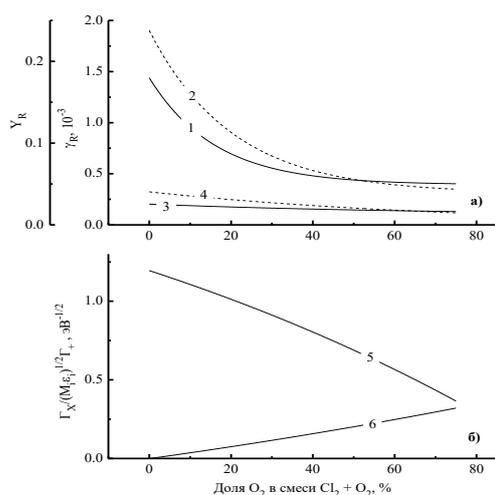
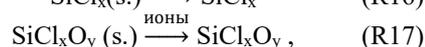
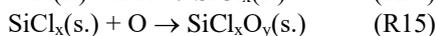


Рис.4. Кинетические характеристики гетерогенного взаимодействия (а) и отношения плотностей потоков активных частиц (б). На рис. а): эффективная вероятность взаимодействия (1, 3) и выход травления (2, 4) для Si (1, 2) и SiO<sub>2</sub> (3, 4). На рис. б): 5 – X = Cl; 6 – X = O.

Из литературы [10, 21, 23] известно, что эффективная вероятность ионно-стимулированного гетерогенного процесса может быть представлена как  $\gamma_R = s_0(1 - \theta)$ , где  $s_0$  – вероятность локализации атома хлора на свободном центре адсорбции, а  $\theta$  – доля таких центров, занятых продуктами взаимодействия. Поэтому можно полагать, что в условиях постоянства температуры поверхности и при доминировании ионного механизма очистки  $\theta \sim \Gamma_{Cl} / \sqrt{M_i \epsilon_i} \Gamma_+$ . Из рис. 4(б) можно видеть, что увеличение доли O<sub>2</sub> в смеси сопровождается монотонным снижением параметра  $\Gamma_{Cl} / \sqrt{M_i \epsilon_i} \Gamma_+$ . Это указывает на увеличение доли свободных центров  $(1 - \theta)$  и противоречит наблюдаемым изменениям  $\gamma_R$  для Si и SiO<sub>2</sub>. Таким образом, реальный механизм травления включает дополнительные гетерогенные процессы, влияющие на величину  $\gamma_R$ , но не связанные с химией атомов хлора. Основываясь на результатах предшествующих работ, можно полагать, что таким процессом является окисление хлоридов кремния в менее летучие оксихлоридные соединения. Соответственно, общая схема гетерогенного взаимодействия на примере Si может быть представлена как:



где индекс (s.) отвечает состоянию частицы на поверхности. Полагая, что скорость R15 пропорциональна плотности потока атомов кислорода  $\Gamma_O$ , а скорость R17 определяется величиной  $\sqrt{M_i \epsilon_i} \Gamma_+$ , увеличение параметра  $\Gamma_O / \sqrt{M_i \epsilon_i} \Gamma_+$  (рис. 4(б)) свидетельствует о накоплении окисленных форм продуктов на обрабатываемой поверхности. Принимая во внимание также а) снижение скоростей R14 и R16 из-за снижения плотности потока атомов хлора; и б) низкую скорость R17 по сравнению с R16 из-за высокой пороговой энергии десорбции SiCl<sub>x</sub>O<sub>y</sub>, можно уверенно говорить об увеличении доли окисленных хлоридов кремния в их общей массе и, следовательно, о снижении эффективной летучести продуктов травления. Как следствие, увеличение  $u_{O_2}$  сопровождается снижением величин  $(1 - \theta)$  и  $\gamma_R$ .

Другой подход к анализу кинетики РИТ рассматривает такие процессы как химически-стимулированное распыление. Основной кинетической характеристикой гетерогенного взаимодействия здесь является выход травления  $Y_R = R / \Gamma_+$ . Из рис. 4(а) можно видеть, что добавление кислорода приводит к снижению в  $Y_R$ , которое не объясняется изменением энергии бомбардирующих ионов. Поэтому можно предположить, что с ростом  $u_{O_2}$  имеет место изменение свойств распыляемой поверхности в сторону увеличения пороговой энергии распыления. Это предположение полностью согласуется с предложенным выше механизмом травления, в частности - с увеличением доли труднолетучих окисленных форм хлоридов кремния в продуктах химического взаимодействия. Таким образом, факт влияния кислорода на гетерогенные стадии процессов травления Si и SiO<sub>2</sub> в плазме смеси Cl<sub>2</sub> + O<sub>2</sub> можно считать достоверно установленным.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (проведение фундаментальных научных исследований) по теме № 0065-2019-0006 «Фундаментальные и прикладные исследования в области субволновой голографической литографии, физико-химических процессов травления 3D нанометровых диэлектрических структур для развития критических технологий производства ЭКБ».

# Features of Si and SiO<sub>2</sub> reactive-ion etching kinetics in Cl<sub>2</sub> + O<sub>2</sub> gas mixture

Efremov A. M., Smirnov S. A., Rybkin V. V., Betelin V. B.

**Abstract:** The investigation of kinetics and mechanisms for Si and SiO<sub>2</sub> reactive-ion etching processes in inductively-coupled rf (13.56 MHz) Cl<sub>2</sub>+O<sub>2</sub> plasma was carried out. The combination of plasma diagnostics and modeling tools allowed one to determine the influence of initial mixture composition on both electro-physical plasma parameters and steady-state densities of plasma active species. It was shown that an increase in O<sub>2</sub> fraction in a feed gas results in a rapid decrease in etching rates for both materials as well as reduces the kinetic coefficients characterizing the heterogeneous interaction pathways, such as effective reaction probability and etching yield. It was found that the last phenomenon is connected with a decrease in the fraction of free adsorption sites for Cl atoms due to the oxidation of reaction products into the lower volatile compounds.

**Keywords:** chlorine, oxygen, plasma, rate coefficient, reaction rate, density, reaction probability.

## Литература

1. K. Nojiri. Dry Etching Technology for Semiconductors. New York, Springer, 2015.
2. S. Wolf, R. N. Tauber. Silicon Processing for the VLSI Era. Volume 1. Process Technology. New York, Lattice Press, 2000.
3. T. Makabe, Z. Petrovic. Plasma electronics: applications in microelectronic device fabrication. New York, Taylor & Francis, 2006.
4. M. A. Lieberman, A. J. Lichtenberg. Principles of plasma discharges and materials processing. New York, John Wiley & Sons Inc., 2005.
5. T. E. F. M. Standaert, C. Hedlund, E. A. Joseph, G. S. Oehrlein and T. J. Dalton. Role of fluorocarbon film formation in the etching of silicon, silicon dioxide, silicon nitride, and amorphous hydrogenated silicon carbide, «Journal of Vacuum Science and Technology A» V. 22 (2004), 53-61.
6. M. C. Chuang, and J. W. Coburn, Molecular-Beam Study of Gas-Surface Chemistry in the Ion-Assisted Etching of Silicon with Atomic and Molecular-Hydrogen and Chlorine, «Journal of Vacuum Science and Technology A» V. 8 (1990), 1969-1977.
7. W. Jin, S. A. Vitale, and H. H. Sawin, Plasma-surface kinetics and simulation of feature profile evolution in Cl<sub>2</sub>+HBr etching of polysilicon, «Journal of Vacuum Science and Technology A» V. 20 (2002), 2106-2015.
8. G. Y. Yeom, Y. Ono, and T. Yamaguchi, Polysilicon Etchback Plasma Process Using HBr, Cl<sub>2</sub>, and SF<sub>6</sub> Gas Mixtures for Deep-trench Isolation, «Journal of Electrochemical Society» V. 139 (1992), № 2, 575-587.
9. S. Tinck, W. Boullart, and A. Bogaerts, Modeling Cl<sub>2</sub>/O<sub>2</sub>/Ar inductively coupled plasmas used for silicon etching: effects of SiO<sub>2</sub> chamber wall coating, «Plasma Sources Science and Technology» V. 20 (2011), 045012.
10. A. M. Efremov, D. P. Kim, C. I. Kim, Simple model for ion-assisted etching using Cl<sub>2</sub>-Ar inductively coupled plasma: Effect of gas mixing ratio, «IEEE Transactions on Plasma Science» V. 32 (2004), № 3, 1344-1355.
11. N. Lim, A. Efremov, K.-H. Kwon, Gas-phase chemistry and etching mechanism of SiN<sub>x</sub> thin films in C<sub>4</sub>F<sub>8</sub> + Ar inductively coupled plasma, «Thin Solid Films» V. 685 (2019), 97-112.
12. J. Lee, A. Efremov, G. Y. Yeom, N. Lim, and K.-H. Kwon, Application of Si and SiO<sub>2</sub> Etching Mechanisms in CF<sub>4</sub>/C<sub>4</sub>F<sub>8</sub>/Ar Inductively Coupled Plasmas for Nanoscale Patterns, «Journal of Nanoscience and Nanotechnology» V. 15 (2015), 8340-8353.
13. B. J. Lee, A. Efremov, J. Kim, C. Kim, K.-H. Kwon, Peculiarities of Si and SiO<sub>2</sub> Etching Kinetics in HBr+Cl<sub>2</sub>+O<sub>2</sub> Inductively Coupled Plasma, «Plasma Chemistry and Plasma Processing» V. 39 (2019), № 1, 339-355.
14. B. J. Lee, A. Efremov, K.-H. Kwon, Plasma Parameters, Gas-Phase Chemistry and Si/SiO<sub>2</sub> Etching Mechanisms in HBr+Cl<sub>2</sub>+O<sub>2</sub> Gas Mixture: Effect of HBr/O<sub>2</sub> mixing ratio, «Vacuum» V. 163 (2019), 110-119.
15. B. J. Lee, A. Efremov, Y. Nam, and K.-H. Kwon, On the Control of Plasma Chemistry and Silicon Etching Kinetics in Ternary HBr + Cl<sub>2</sub> + O<sub>2</sub> Gas System: Effects of HBr/O<sub>2</sub> and Cl<sub>2</sub>/O<sub>2</sub> Mixing Ratios, «Science of Advanced Materials» V. 12 (2020) 628-639.

16. E. V. Shun'ko. Langmuir probe in theory and practice. Universal Publishers, Boca Raton, 2008.
17. C. C. Hsu, M. A. Nierode, J. W. Coburn, and D. B. Graves, Comparison of model and experiment for Ar, Ar/O<sub>2</sub> and Ar/O<sub>2</sub>/Cl<sub>2</sub> inductively coupled plasmas, «Journal of Physics. D: Applied Physics» V. 39 (2006), №15, 3272-2388.
18. C. Lee, and M. A. Lieberman, Global model of Ar, O<sub>2</sub>, Cl<sub>2</sub> and Ar/O<sub>2</sub> high density plasma discharges, «Journal of Vacuum Science and Technology A» V. 13 (1995), № 2, 368-377.
19. A. Efremov, N. K. Min, B. G. Choi, K. H. Baek, and K. H. Kwon. Model-Based Analysis of Plasma Parameters and Active Species Kinetics in Cl<sub>2</sub>/X (X = Ar, He, N<sub>2</sub>) Inductively Coupled Plasmas, «Journal of Electrochemical Society» V. 155 (2008), № 12, D777-D793.
20. A. Efremov, J. Lee, K. H. Kwon, A comparative study of CF<sub>4</sub>, Cl<sub>2</sub> and HBr + Ar inductively coupled plasmas for dry etching applications, «Thin Solid Films » V. 629 (2017), 39-47.
21. D. C. Gray, I. Tepermeister, and H. H. Sawin, Phenomenological Modeling of Ion-Enhanced Surface Kinetics in Fluorine-Based Plasma-Etching, «Journal of Vacuum Science and Technology B» V. 11 (1993), № 4, 1243-1254.
22. L. G. Christophorou, and J. K. Olthoff, Fundamental electron interactions with plasma processing gases. Springer Science+Business Media, New York, 2004.
23. C. Lee, D. B. Graves, and M. A. Lieberman, Role of etch products in polysilicon etching in a high-density chlorine discharge, «Plasma Chemistry and Plasma Processing» V. 16 (1996), № 1, 99-118.

# Собственные значения гамильтониана Изинга с дальним действием

Б.В. Крыжановский<sup>1</sup>, Л.Б. Литинский<sup>2</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, [kryzhanov@mail.ru](mailto:kryzhanov@mail.ru)

<sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, [litin@mail.ru](mailto:litin@mail.ru)

**Аннотация.** Получены аналитические выражения для собственных значений гамильтониана модели Изинга на гиперкубической решетке. Собственные значения многомерного гамильтониана выражаются через константы взаимодействия между спинами и собственные значения одномерного гамильтониана (выражения для этих последних хорошо известны). Собственные векторы многомерного гамильтониана вычисляются как кронекеровы произведения собственных векторов одномерного гамильтониана. В случае периодических граничных условий строгие выражения могут быть получены, когда учитывается взаимодействие с неограниченно большим числом соседних спинов. Мы приводим выражения для собственных значений двумерного и трехмерного гамильтонианов, когда учитывается взаимодействие со спинами пяти ближайших координационных сфер. Для свободных границ собственные значения двумерного и трехмерного гамильтонианов выражаются через собственные значения одномерной модели только в двух исключительных случаях.

**Ключевые слова:** модель Изинга, гиперкубическая решетка, собственные значения и собственные векторы

## 1. Введение

Модель Изинга постоянно находится в фокусе внимания исследователей. Отчасти это связано с тем, что модель является удобным объектом для проверки новых методов вычисления свободной энергии и критических характеристик. Отчасти – с тем, что, несмотря на кажущуюся простоту, в ряде случаев модель не поддается точному решению уже около ста лет. Обзор различных подходов к исследованию модели Изинга и полученных при этом результатов можно найти в [1]. Приложениям к изучению фазовых переходов в конденсированных средах посвящена книга [2]. Приложениям в теории спиновых стекол и нейронных сетей посвящена монография [3], а связям модели Изинга с задачами оптимизации – коллективная монография [4].

В нашей работе получены точные выражения для собственных значений модели Изинга на гиперкубической решетке, когда учитывается взаимодействие с произвольным числом ближайших соседей. Изучена зависимость результатов от типа граничных условий. Наша работа близко примыкает к исследованиям [5].

Для спинов, взаимодействие с которыми учитывают в задаче, обычно используют словесные характеристики: nearest neighbors, next neighbors and next-next neighbors. В наших рассмотрениях будут возникать спины, отстоящие от начального спина очень далеко. Чтобы не наращивать многократно приставку «next-», будем пользоваться номерами координацион-

ных сфер, в которые объединяются узлы решетки [6,7]. Для заданного начального узла  $k$ -я координационная сфера включает все узлы решетки, отстоящие от начального узла на расстоянии, занимающее  $k$ -е место в возрастающей последовательности всех различных расстояний между узлами. Спины в узлах  $k$ -й координационной сферы одинаково взаимодействуют со спином в начальном узле – соответствующую константу взаимодействия будем обозначать  $w_k$ . Координационные сферы с небольшими значениями номеров  $k$  именуют приведенными выше словесными характеристиками: nearest neighbors – это в точности 1-я координационная сфера, next neighbors – 2-я координационная сфера, next-next neighbors – 3-я. Мы будем пользоваться как номерами координационных сфер, так и общеупотребительными названиями.

Во втором разделе изучается одномерная модель Изинга. Вводятся матрицы  $\mathbf{J}(k)$ , отвечающие учету взаимодействия со спинами только  $k$ -й координационной сферы. В случае периодических граничных условий все такие матрицы коммутируют. Тогда легко получить выражение для собственных значений одномерного гамильтониана  $\mathbf{H}_1 = \sum_k w_k \mathbf{J}(k)$ , когда учитывается взаимодействие со спинами нескольких координационных сфер. Иная ситуация в случае свободных границ. Для этого случая будем снабжать значком «тильда» матрицы связи, их собственные значения и собственные

векторы. Матрицы  $\tilde{\mathbf{J}}(k)$  не коммутируют друг с другом, и значит нельзя выразить собственные значения матрицы  $\tilde{\mathbf{H}}_1 = \sum_k w_k \tilde{\mathbf{J}}(k)$  через собственные значения отдельных матриц  $\tilde{\mathbf{J}}(k)$ . Здесь известны только выражения для собственных значений  $\{\tilde{\lambda}_i\}_1^n$  матрицы  $\tilde{\mathbf{J}}(1)$ , которая отвечает учету взаимодействия  $w_1$  с ближайшими соседями.

В третьем разделе мы анализируем двумерную и трехмерную модели Изинга с периодическими граничными условиями. Для двумерной модели учитывается взаимодействие со спинами первой и второй координационных сфер. Для трехмерной модели сверх того учитывается взаимодействие еще и со спинами третьей координационной сферы. Выбор сравнительно простых вариантов двумерной и трехмерной моделей Изинга обусловлен следующим. Во-первых, здесь мы детально излагаем метод, позволяющий выразить собственные значения многомерного гамильтониана через константы взаимодействия и собственные значения одномерного гамильтониана. Изложение является многоэтапным, поэтому лучше проводить его на сравнительно простом материале. Во-вторых, полученные здесь результаты в полном объеме справедливы для тех же вариантов модели Изинга и в случае свободных границ – это будет показано уже в следующем, четвертом разделе. Там же будет установлено, что для свободных границ распространить эти результаты на большее число соседей невозможно.

Наконец, в пятом разделе излагается еще один метод вычисления собственных значений многомерного гамильтониана Изинга. Метод базируется на представлении собственного вектора многомерной задачи в виде кронекерова произведения собственных векторов одномерной модели Изинга. Метод особенно эффективен, когда в многомерной модели учитывают взаимодействие со спинами, принадлежащими большому числу координационных сфер. Для случая периодических граничных условий здесь получены точные выражения для собственных значений гамильтониана, когда для планарной и кубической решеток учитывается взаимодействие со спинами, принадлежащими пяти ближайшим координационным сферам.

Полученные результаты могут быть полезны при исследовании спектральной плотности многомерной модели Изинга и ее зависимости от параметров взаимодействия между спинами. Найденные точные выражения для собственных энергий могут быть востребованы при вычислении свободной энергии.

## 2. 1D-модель Изинга

Рассматривается одномерная модель Изинга, которой описывается линейная цепочка  $n$  последовательно связанных спинов  $s_i = \pm 1$ ,  $i = 1, \dots, n$ . В первом пункте изучаются периодические граничные условия, когда цепочку спинов замыкают в кольцо: последний,  $n$ -й спин цепочки, объявляют ближайшим соседом первого спина «слева»; предпоследний,  $(n-1)$ -й спин цепочки оказывается вторым соседом первого спина слева и т.д. Здесь имеется несколько строгих результатов. Во втором пункте раздела рассматриваются свободные границы, когда никаких дополнительных условий на последние спины цепочки не накладывается.

### 2.1. Периодические граничные условия

Обозначим  $\mathbf{J}(k)$  матрицу связи одномерной модели, которой учитывается взаимодействие *только* со спинами  $k$ -й координационной сферы. Матрицы  $\mathbf{J}(1)$  и  $\mathbf{J}(2)$ , например, имеют вид:

$$\mathbf{J}(1) = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 \end{pmatrix},$$

$$\mathbf{J}(2) = \begin{pmatrix} 0 & 0 & 1 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 1 & \dots & \dots & 1 \\ 1 & 0 & 0 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & 1 \\ 1 & 0 & \dots & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 1 & 0 & 0 \end{pmatrix}.$$

Вообще,  $\mathbf{J}(k)$  есть симметричная матрица, у которой единицы заполняют  $k$ -ю и  $(n-k)$ -ю наддиагонали, параллельные главной диагонали матрицы. Поскольку выполняется тождество  $\mathbf{J}(k) \equiv \mathbf{J}(n-k)$ , ограничим интервал изменения индекса  $k$ :  $k \leq k_0 = \left\lfloor \frac{n}{2} \right\rfloor$ . С помощью матриц

$\{\mathbf{J}(k)\}_1^{k_0}$  легко записать матрицу гамильтониана одномерной модели, в которой учитывается взаимодействие  $w_1$  со спинами первой координационной сферы, взаимодействие  $w_2$  со спинами второй координационной сферы, и так

далее – взаимодействие  $w_k$  со спинами  $k$ -й координационной сферы:

$$\mathbf{H}_1 = w_1 \cdot \mathbf{J}(1) + w_2 \cdot \mathbf{J}(2) + \dots + w_k \cdot \mathbf{J}(k). \quad (1)$$

Займемся задачей на собственные значения матрицы  $\mathbf{J}(1)$ . Соответствующая система уравнений имеет вид:

$$\begin{pmatrix} 0 & + & x_2 & + & 0 & \dots & + & x_n & = & \lambda x_1 \\ x_1 & + & 0 & + & x_3 & \dots & + & 0 & = & \lambda x_2 \\ \dots & \dots \\ x_1 & + & 0 & + & 0 & \dots & x_{n-1} & 0 & = & \lambda x_n \end{pmatrix}. \quad (2)$$

Решение системы уравнений (2) хорошо известно [5, 10]:

$$\lambda_i(1) = 2 \cos(\varphi_i), \quad \varphi_i = \frac{2\pi(i-1)}{n}, \quad i=1, \dots, n. \quad (3)$$

Собственные векторы системы (2) тоже известны [5], [11]. Поскольку почти все собственные значения (3) дважды вырождены, собственные векторы можно записать по-разному. Наиболее удобным представляется следующий вид собственных векторов:

$$\mathbf{f}_i = \begin{pmatrix} f_i^{(1)} \\ f_i^{(2)} \\ \vdots \\ f_i^{(n)} \end{pmatrix}, \quad f_i^{(j)} = \frac{\cos(j-1)\varphi_i + \sin(j-1)\varphi_i}{\sqrt{n}}, \quad i, j = 1, 2, \dots, n. \quad (4)$$

Получим теперь выражение для собственных значений матрицы  $\mathbf{J}(k)$  когда  $k > 1$ . Возводя матрицу  $\mathbf{J}(1)$  в целую степень, получим соотношения:

$$\begin{aligned} \mathbf{J}(2) &= \mathbf{J}^2(1) - 2 \cdot \mathbf{I} \\ \mathbf{J}(3) &= \mathbf{J}^3(1) - 3 \cdot \mathbf{J}(1) \\ \mathbf{J}(4) &= \mathbf{J}^4(1) - 4 \cdot \mathbf{J}^2(1) + 2 \cdot \mathbf{I}, \end{aligned} \quad (5)$$

где  $\mathbf{I} = \mathbf{diag}(1, 1, \dots, 1)$  – единичная  $(n \times n)$ -матрица. Покажем, что при любом значении  $k$  матрица  $\mathbf{J}(k)$  будет полиномом от матрицы  $\mathbf{J}(1)$ . Воспользуемся равенством, которое легко проверить самостоятельно:

$$\mathbf{J}(k) \cdot \mathbf{J}(l) = \begin{cases} \mathbf{J}(k+l) + \mathbf{J}(k-l), & \text{когда } k > l \\ \mathbf{J}(2k) + 2 \cdot \mathbf{I}, & \text{когда } k = l \end{cases}. \quad (6)$$

Пусть для всех  $l \leq k$  показано, что матрица  $\mathbf{J}(l)$  есть полином от матрицы  $\mathbf{J}(1)$ . Тогда из равенства (6) получаем, что и для любого значения  $k+l$  матрицу  $\mathbf{J}(k+l)$  можно представить в виде полинома от  $\mathbf{J}(1)$ . Поскольку при любом значении  $k$  матрица  $\mathbf{J}(k)$  является полиномом от матрицы  $\mathbf{J}(1)$ , матрицы  $\mathbf{J}(k)$  коммутируют друг с другом, и значит все эти матрицы обладают одним и тем же набором собственных векторов (4).

Теперь получим выражения для собственных значений матрицы  $\mathbf{J}(k)$ , которые будем

обозначать как  $\{\lambda_i(k)\}_{i=1}^n$ . Используя первое из соотношений (5) получаем:

$$\lambda_i(2) = \lambda_i^2(1) - 2 = 2 \cos(2 \cdot \varphi_i)$$

Аналогично, с помощью второго соотношения (5) получаем, что  $\lambda_i(3) = 2 \cdot \cos(3 \cdot \varphi_i)$ . Используя (6) теперь легко доказать, что и для всякого  $k$  выполняется равенство:

$$\lambda_i(k) = 2 \cdot \cos(k \cdot \varphi_i), \quad i = 1, 2, \dots, n. \quad (7)$$

Действительно, пусть равенство (7) справедливо для всех  $l \leq k$ . Тогда с помощью соотношения (6) получаем:

$$\lambda_i(k+l) = \dots = 2 \cdot \cos((k+l)\varphi_i).$$

Подведем итог: единая для всех матриц  $\mathbf{J}(k)$  система собственных векторов  $\{\mathbf{f}_i\}_{i=1}^n$  позволяет вычислить собственные значения одномерного гамильтониана  $\mathbf{H}_1$  (1):

$$\begin{aligned} \lambda_i(\mathbf{H}_1) &= w_1 \lambda_i(1) + \dots + w_k \lambda_i(k) = \\ &= 2 \sum_{l=1}^k w_l \cos(l\varphi_i), \quad i = 1, 2, \dots, n. \end{aligned} \quad (8)$$

## 2.2. Свободные границы

В этом случае все характеристики матрицы связи снабдим значком «тильда». При построении матрицы  $\tilde{\mathbf{J}}(k)$ , являющейся аналогом матрицы  $\mathbf{J}(k)$ , учитывают взаимодействие каждого спина только с реальным  $k$ -м соседом в одномерной цепочке, и замыкания цепочки в кольцо не происходит. Матрица  $\tilde{\mathbf{J}}(k)$  симметрична, у нее отлична от нуля  $k$ -я наддиагональ (и симметричная поддиагональ). Характерная для периодических граничных условий  $(n-k)$ -я наддиагональ здесь отсутствует.

Для этих матриц известно выражение для собственных значений и собственных векторов  $\tilde{\mathbf{J}}(1)$  [12, 13]:

$$\tilde{\lambda}_i(1) = 2 \cdot \cos(\tilde{\varphi}_i), \quad \text{где } \tilde{\varphi}_i = \frac{\pi \cdot i}{n+1}, \quad (9.1)$$

а также собственные векторы:

$$\tilde{\mathbf{f}}_i = \begin{pmatrix} \tilde{f}_i^{(1)} \\ \tilde{f}_i^{(2)} \\ \vdots \\ \tilde{f}_i^{(n)} \end{pmatrix}, \quad \text{где } \begin{cases} \tilde{f}_i^{(j)} = \sqrt{\frac{2}{n+1}} \sin(j \cdot \tilde{\varphi}_i) \\ i, j = 1, 2, \dots, n. \end{cases} \quad (9.2)$$

Формулы (9), равно как и (3) – (4), можно получить с помощью несложной тригонометрии.

Оказывается, матрицы данного типа не коммутируют друг с другом (легко убедиться самостоятельно). Это означает, что каждая матрица  $\tilde{\mathbf{J}}(k)$  обладает индивидуальным набором собственных векторов, и все эти наборы различны

– а значит, не существует базиса, в котором все матрицы  $\tilde{\mathbf{J}}(k)$  приводятся к диагональному виду одновременно. Соотношения типа (8) здесь невозможны. Это сильно ограничивает получение точных результатов для матриц данного типа в двумерной и трехмерной моделях Изинга. Грубо говоря, в случае периодических граничных условий собственные значения  $n$ -мерной задачи выражаются через собственные значения одномерной модели  $\{\lambda_i(1)\}_1^n$  при учете взаимодействия со спинами любого числа координационных сфер. Для свободных границ подобное возможно только в двух исключительных случаях – см. конец следующего раздела.

**Замечание.** В каждой строке (и столбце) матрицы  $\mathbf{J}(k)$  стоят две единицы. Однако так будет не всегда: при четном  $n = 2m$  и  $k = m$  нижняя наддиагональ совпадает с верхней:  $k = m = n - k$ . У такой матрицы в каждой строке стоит всего одна единица – см. пример для случая  $n = 6, k = 3$ :

$$\mathbf{J}(k=3) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Здесь возникает противоречие: формула (7) дает  $\lambda_i(3) = 2 \cos(\pi \cdot (i-1)) = \pm 2$ ,  $i = 1, 2, \dots, n$ , а прямым вычислением получаются вдвое меньшие собственные значения:  $\lambda_i(3) = \pm 1$ .

Когда  $n$  достаточно велико, матрица  $\mathbf{J}(n/2)$  практически не возникает в рассмотренных – см. следующие два раздела. Однако, вообще говоря, это противоречие необходимо устранить. Можно слегка изменить алгоритм формирования матриц  $\mathbf{J}(k)$ : не проставлять единицы вдоль  $k$ -й и  $(n-k)$ -й наддиагоналей, а прибавлять единицу к матричным элементам на этих наддиагоналях. Такая формулировка автоматически приведет к появлению двоек на месте единиц в рассматриваемых матрицах, и не затронет остальные матрицы.

### 3. 2D- и 3D-модели Изинга с периодическими граничными условиями: простой случай

Здесь анализируются сравнительно простые варианты двумерной и трехмерной модели Изинга с периодическими граничными условиями. Для двумерной модели учитывается взаимодействие со спинами первой и второй координационных сфер (п.1). Для трехмерной модели учитывается взаимодействие со спинами первой, второй и третьей координационных сфер (п.2).

Условимся об обозначениях. Для обеих решеток спины первой координационной сферы отстоят от начального спина на расстояние ребра решетки, длина которого принимается равной 1. Константу взаимодействия с этими спинами обозначим  $w_1$ . Спины второй координационной сферы отстоят от начального спина на расстояние  $\sqrt{2}$  (см. рис.1). Константу взаимодействия с ними обозначим  $w_2$ . Для кубической решетки потребуется еще и третья координационная сфера, на расстоянии  $\sqrt{3}$  от начального спина (см. рис.2). Константу взаимодействия с ними обозначим  $w_3$ .

Символами  $\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1 \dots$  обозначаются  $(n \times n)$ -матрицы, составленные из взвешенных сумм матриц  $\mathbf{J}(k)$  наподобие (1). Нижний индекс «1» указывает на то, что эти матрицы связаны с одномерной моделью Изинга. Кроме того, возникнут  $(n^2 \times n^2)$ -матрицы, описывающие взаимодействие в двумерной модели Изинга. Эти матрицы снабжаются нижним индексом 2:  $\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2$ . Наконец,  $(n^3 \times n^3)$ -матрица  $\mathbf{A}_3$  описывает взаимодействие в кубической модели Изинга (см. п. 2).

И последнее: упростим обозначение для собственных значений матрицы  $\mathbf{J}(1)$  – вместо  $\lambda_i(1)$  будем писать  $\lambda_i$ . Этим упрощением будем пользоваться до конца статьи.

**1.** Полное число спинов в двумерной модели Изинга равно  $n^2$ , где  $n$  – число спинов на ребре квадрата. Соответствующую  $(n^2 \times n^2)$ -матрицу связи обозначим  $\mathbf{A}_2$ . Выпишем в явном виде ее первые  $n$  строк:

$$\left( \begin{array}{cccccccccccc} 0 & w_1 & 0 & \cdots & w_1 & w_1 & w_2 & 0 & \cdots & w_2 & 0 & \cdots & 0 & \cdots \\ w_1 & 0 & w_1 & \cdots & 0 & w_2 & w_1 & w_2 & \cdots & 0 & 0 & \cdots & 0 & \cdots \\ 0 & w_1 & 0 & w_1 & 0 & 0 & w_2 & w_1 & w_2 & 0 & 0 & \cdots & 0 & \cdots \\ \cdots & 0 & \cdots & 0 & \cdots \\ w_1 & 0 & 0 \dots & w_1 & 0 & w_2 & 0 & 0 \dots & w_2 & w_1 & 0 & \cdots & 0 & \cdots \end{array} \right) \left( \begin{array}{cccccc} w_1 & w_2 & 0 & \cdots & w_2 \\ w_2 & w_1 & w_2 & \cdots & 0 \\ 0 & w_2 & w_1 & w_2 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_2 & 0 & 0 \dots & w_2 & w_1 \end{array} \right), \quad (10)$$

и запишем (10) в блочном виде:

$$(w_1 \mathbf{J}(1) \quad w_1 \mathbf{I} + w_2 \mathbf{J}(1) \quad \mathbf{0} \dots \mathbf{0} \quad w_1 \mathbf{I} + w_2 \mathbf{J}(1)),$$

где  $\mathbf{J}(1)$  и единичная  $(n \times n)$ -матрица  $\mathbf{I}$  введены в предыдущем разделе. Определим  $(n \times n)$ -матрицы

$$\mathbf{A}_1 = w_1 \mathbf{J}(1) \quad \text{и} \quad \mathbf{B}_1 = w_1 \mathbf{I} + w_2 \mathbf{J}(1), \quad (11)$$

и запишем всю матрицу взаимодействий  $\mathbf{A}_2$  в блочном виде:

$$\mathbf{A}_2 = \begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 & 0 & 0 & \cdots & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{B}_1 & 0 & \cdots & 0 & \mathbf{B}_1 & \mathbf{A}_1 \end{pmatrix}. \quad (12)$$

Матрица  $\mathbf{A}_2$  образована с помощью двух  $(n \times n)$ -матричных блоков  $\mathbf{A}_1$  и  $\mathbf{B}_1$  (11): первая «блочная» строка матрицы  $\mathbf{A}_2$  содержит только  $\mathbf{A}_1$  и  $\mathbf{B}_1$ , а каждая следующая строка получается циклическим сдвигом предыдущей строки на одну позицию вправо. Будем называть  $(n \times n)$ -блоки  $\mathbf{A}_1$  и  $\mathbf{B}_1$  образующими блоками матрицы  $\mathbf{A}_2$ , или образующими матрицами, и обозначать это  $\mathbf{A}_2 \sim \{\mathbf{A}_1, \mathbf{B}_1\}$ . Задача на собственные значения принимает вид:

$$\left( \begin{array}{cccccccc} w_1 \lambda_1 & \cdots & 0 & w_1 + w_2 \lambda_1 & \cdots & 0 & w_1 + w_2 \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & w_1 \lambda_n & 0 & \cdots & w_1 + w_2 \lambda_n & 0 & \cdots & w_1 + w_2 \lambda_n \\ w_1 + w_2 \lambda_1 & \cdots & 0 & w_1 \lambda_1 & \cdots & 0 & \cdots & 0 & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & 0 & \cdots \\ 0 & \cdots & 1 + w_2 \lambda_n & 0 & \cdots & w_1 \lambda_n & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ w_1 + w_2 \lambda_1 & \cdots & 0 & \cdots & w_1 \lambda_1 & \cdots & 0 & \cdots & \cdots \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots & \cdots & \cdots \\ 0 & \cdots & w_1 + w_2 \lambda_n & \cdots & 0 & \cdots & w_1 \lambda_n & \cdots & \cdots \end{array} \right) \left( \begin{array}{c} x_1^{(1)} \\ \vdots \\ x_n^{(1)} \\ x_1^{(2)} \\ \vdots \\ x_n^{(2)} \\ \vdots \\ x_1^{(n)} \\ \vdots \\ x_n^{(n)} \end{array} \right) = \mu \left( \begin{array}{c} x_1^{(1)} \\ \vdots \\ x_n^{(1)} \\ x_1^{(2)} \\ \vdots \\ x_n^{(2)} \\ \vdots \\ x_1^{(n)} \\ \vdots \\ x_n^{(n)} \end{array} \right). \quad (14)$$

Система уравнений (14) распадается на  $n$  независимых задач размерности  $n$ . В самом деле, выпишем одну за другой 1-ю строку задачи (14),  $(n+1)$ -ю,  $(2n+1)$ -ю, ... и, наконец

$$\left( \begin{array}{cccccc} \mathbf{A}_1 & \mathbf{B}_1 & 0 & 0 & \cdots & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{B}_1 & 0 & \cdots & 0 & \mathbf{B}_1 & \mathbf{A}_1 \end{array} \right) \left( \begin{array}{c} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \mathbf{x}^{(3)} \\ \vdots \\ \mathbf{x}^{(n)} \end{array} \right) = \mu \left( \begin{array}{c} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \mathbf{x}^{(3)} \\ \vdots \\ \mathbf{x}^{(n)} \end{array} \right)$$

$$\text{где } \mathbf{x}^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ x_3^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix}, \quad i = 1, 2, \dots, n. \quad (13)$$

Здесь  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}$  -  $n$ -мерные векторы-компоненты собственного вектора размерности  $n^2$ .

Пространство  $\mathbf{R}^{n^2}$  представим в виде ортогональной суммы  $n$  подпространств  $\mathbf{R}^n$ , на которых и определены образующие матрицы  $\mathbf{A}_1$  и  $\mathbf{B}_1$ :  $\mathbf{R}^{n^2} = \mathbf{R}^n \oplus \mathbf{R}^n \oplus \dots \oplus \mathbf{R}^n$ . В каждом подпространстве  $\mathbf{R}^n$  перейдем от базиса декартовых ортов к базису из  $n$  собственных векторов (4) матрицы  $\mathbf{J}(1)$ . Поскольку в этом базисе образующие матрицы  $\mathbf{A}_1$  и  $\mathbf{B}_1$  приводятся к диагональному виду, задача (13) принимает вид:

$(n^2 - n + 1)$ -ю строки – получим систему уравнений:

$$\begin{array}{ccccccc}
w_1 \lambda_1 x_1^{(1)} & + & (w_1 + w_2 \lambda_1) x_1^{(2)} & + & 0 & \cdots & (w_1 + w_2 \lambda_1) x_1^{(n)} = \mu x_1^{(1)} \\
(w_1 + w_2 \lambda_1) x_1^{(1)} & + & w_1 \lambda_1 x_1^{(2)} & + & (w_1 + w_2 \lambda_1) x_1^{(3)} & \cdots & 0 = \mu x_1^{(2)} \\
0 & + & (w_1 + w_2 \lambda_1) x_1^{(2)} & + & w_1 \lambda_1 x_1^{(3)} & \cdots & 0 = \mu x_1^{(3)} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
(w_1 + w_2 \lambda_1) x_1^{(1)} & + & 0 & + & 0 & \cdots & w_1 \lambda_1 x_1^{(n)} = \mu x_1^{(n)}
\end{array} \quad (15)$$

Переменные  $x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(n)}$ , фигурирующие в (15), не используются ни в одном другом уравнении системы (14). Следовательно, задачу (15) можно решать независимо от остальных переменных. Аналогично, объединяя вместе 2-ю строку системы (14),  $(n+2)$ -ю,  $(2n+2)$ -ю,  $\dots, (n^2-n+2)$ -ю строки получим аналог системы (15) для  $x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(n)}$ , где вместо  $\lambda_1$

$$\begin{array}{ccccccc}
w_1 \lambda_i x_i^{(1)} & + & (w_1 + w_2 \lambda_i) x_i^{(2)} & + & 0 & \cdots & (w_1 + w_2 \lambda_i) x_i^{(n)} = \mu x_i^{(1)} \\
(w_1 + w_2 \lambda_i) x_i^{(1)} & + & w_1 \lambda_i x_i^{(2)} & + & (w_1 + w_2 \lambda_i) x_i^{(3)} & \cdots & 0 = \mu x_i^{(2)} \\
0 & + & (w_1 + w_2 \lambda_i) x_i^{(2)} & + & w_1 \lambda_i x_i^{(3)} & \cdots & 0 = \mu x_i^{(3)} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
(w_1 + w_2 \lambda_i) x_i^{(1)} & + & 0 & + & 0 & \cdots & w_1 \lambda_i x_i^{(n)} = \mu x_i^{(n)}
\end{array} \quad (16)$$

где  $i=1, 2, \dots, n$ , и эти задачи надо решать независимо друг от друга.

Упростим систему (16), перебросив диагональный член  $w_1 \lambda_i x_i^{(k)}$  каждого уравнения в правую часть, и разделив все коэффициенты на общий множитель  $w_1 + w_2 \lambda_i$ . Получим систему уравнений

$$\begin{array}{ccccccc}
0 & + & x_i^{(2)} & + & 0 & \cdots & + & x_i^{(n)} = v_i \cdot x_i^{(1)} \\
x_i^{(1)} & + & 0 & + & x_i^{(3)} & \cdots & + & 0 = v_i \cdot x_i^{(2)} \\
\cdots & \cdots \\
x_i^{(1)} & + & 0 & + & 0 & \cdots & x_i^{(n-1)} & + 0 = v_i \cdot x_i^{(n)}
\end{array}$$

$$\text{где } v_i = \frac{\mu - w_1 \lambda_i}{w_1 + w_2 \lambda_i}.$$

Эта система совпадает с системой (2), и, согласно формуле (3),  $v_i$  принимает  $n$  различных значений  $\{\lambda_j\}_{j=1}^n : v_i = \lambda_j, j=1, \dots, n$ . Тогда

$$\mu = (w_1 + w_2 \lambda_j) \cdot \lambda_j + w_1 \lambda_j, \text{ где } j=1, \dots, n.$$

Придавая индексу  $i$  все возможные значения, окончательно получаем:

$$\mu_{ij} = w_1 \cdot (\lambda_i + \lambda_j) + w_2 \cdot \lambda_i \lambda_j, \quad i, j=1, 2, \dots, n. \quad (17)$$

Это и есть выражение для собственных значений матрицы гамильтониана 2D модели Изинга с периодическими граничными условиями, когда учитывается взаимодействие со спинами двух ближайших координационных сфер. Не сложно показать, что собственными векторами будут кронекеровы произведения собственных векторов  $\mathbf{f}_i$  и  $\mathbf{f}_j$  (4):

$$\mathbf{F}_{ij} = \mathbf{f}_i \otimes \mathbf{f}_j. \quad (18)$$

фигурирует  $\lambda_2$ , а вместо  $\{x_1^{(i)}\}_1^n$  - переменные  $\{x_2^{(i)}\}_1^n$ ; легко видеть, что эти переменные не используются в других уравнениях системы (14). Продолжая в том же духе, получим, что задача (14) распадается на  $n$  однотипных задач:

**2.** Аналогично анализируется трехмерная модель Изинга. Мы приведем только ключевые моменты рассмотрения и основные формулы.

Полное число спинов в трехмерной модели Изинга равно  $n^3$ , а  $(n^3 \times n^3)$ -матрицу гамильтониана обозначим  $\mathbf{A}_3$ . Ее удобно представить в виде, аналогичном выражению (12):

$$\mathbf{A}_3 = \begin{pmatrix} \mathbf{A}_2 & \mathbf{B}_2 & 0 & 0 & \cdots & \mathbf{B}_2 \\ \mathbf{B}_2 & \mathbf{A}_2 & \mathbf{B}_2 & 0 & \cdots & 0 \\ 0 & \mathbf{B}_2 & \mathbf{A}_2 & \mathbf{B}_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{B}_2 & 0 & \cdots & 0 & \mathbf{B}_2 & \mathbf{A}_2 \end{pmatrix}.$$

Матрица  $\mathbf{A}_3$  образована с помощью образующих  $(n^2 \times n^2)$ -блоков  $\mathbf{A}_2$  и  $\mathbf{B}_2$ :  $\mathbf{A}_3 \sim (\mathbf{A}_2, \mathbf{B}_2)$ . Матрица  $\mathbf{A}_2$  дается выражением (12), ее образующими блоками являются  $(n \times n)$ -матрицы  $\mathbf{A}_1 = w_1 \mathbf{J}(1)$  и  $\mathbf{B}_1 = w_1 \mathbf{I} + w_2 \mathbf{J}(1)$  (11). Нетрудно показать, что  $(n^2 \times n^2)$ -матрица  $\mathbf{B}_2$  имеет вид:

$$\mathbf{B}_2 = \begin{pmatrix} \mathbf{B}_1 & \mathbf{D}_1 & 0 & 0 & \cdots & \mathbf{D}_1 \\ \mathbf{D}_1 & \mathbf{B}_1 & \mathbf{D}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{D}_1 & \mathbf{B}_1 & \mathbf{D}_1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{D}_1 & 0 & \cdots & 0 & \mathbf{D}_1 & \mathbf{B}_1 \end{pmatrix}, \quad (19)$$

где  $\mathbf{D}_1 = w_2 \mathbf{I} + w_3 \mathbf{J}(1)$ . Ее образующими блоками являются матрицы  $\mathbf{B}_1$  и  $\mathbf{D}_1$ :  $\mathbf{B}_2 \sim (\mathbf{B}_1, \mathbf{D}_1)$ .



ла):

$$\begin{cases} \mathbf{A}_2 \sim (\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1): \mathbf{A}_1 = w_1 \mathbf{J}(1) + w_3 \mathbf{J}(2) \\ \mathbf{B}_1 = w_1 \mathbf{I} + w_2 \mathbf{J}(1), \quad \mathbf{C}_1 = w_3 \mathbf{I} \end{cases} \quad (23)$$

В случае периодических граничных условий все эти матрицы приводятся к диагональному виду одновременно в базисе из собственных векторов (4). В результате оказывается возможной декомпозиция большой задачи (14) на  $n$  независимых задач (16), и все остальное, что из этого следует.

Напротив, в случае свободных границ не существует базиса, в котором матрицы  $\tilde{\mathbf{J}}(1)$  и  $\tilde{\mathbf{J}}(2)$  приводятся к диагональному виду одновременно. Значит, невозможно привести к диагональному виду *одновременно* все три образующих блока  $\tilde{\mathbf{A}}_1$ ,  $\tilde{\mathbf{B}}_1$  и  $\tilde{\mathbf{C}}_1$  (23). Следовательно, все соотношения, начиная с (14) и далее, становятся в данном случае просто невозможными.

Для двумерной модели Изинга со свободными границами справедливы формулы (22), а их обобщение на случай взаимодействия с большим числом соседей невозможно.

$$\tilde{\mu}_{ijk} = w_1 \cdot (\tilde{\lambda}_i + \tilde{\lambda}_j + \tilde{\lambda}_k) + w_2 \cdot (\tilde{\lambda}_i \tilde{\lambda}_j + \tilde{\lambda}_j \tilde{\lambda}_k + \tilde{\lambda}_k \tilde{\lambda}_i) + w_3 \cdot \tilde{\lambda}_i \tilde{\lambda}_j \tilde{\lambda}_k, \quad i, j, k = 1, 2, \dots, n. \quad (24)$$

Другое дело, если для трехмерной модели Изинга будем учитывать взаимодействие со спинами, скажем, еще и четвертой координационной сферы. В этом случае образующих блоков всех уровней станет больше, а их конструкция усложнится (см. п.2 следующего раздела). К примеру, образующие блоки нижнего уровня примут вид:

$$\begin{aligned} \tilde{\mathbf{A}}_1 &= w_1 \tilde{\mathbf{J}}(1) + w_4 \tilde{\mathbf{J}}(2), \quad \tilde{\mathbf{B}}_1 = w_1 \mathbf{I} + w_2 \tilde{\mathbf{J}}(1), \\ \tilde{\mathbf{C}}_1 &= w_4 \mathbf{I}, \quad \tilde{\mathbf{D}}_1 = w_2 \mathbf{I} + w_3 \tilde{\mathbf{J}}(1). \end{aligned}$$

Не существует базиса, в котором все эти матрицы одновременно диагонализуются – этому мешает матрица  $\tilde{\mathbf{J}}(2)$  в выражении для  $\tilde{\mathbf{A}}_1$ . Следовательно, кронекеровы произведения  $\tilde{\mathbf{F}}_{ij} = \tilde{\mathbf{f}}_i \otimes \tilde{\mathbf{f}}_j$  не будут собственными векторами  $(n^2 \times n^2)$ -матрицы  $\tilde{\mathbf{A}}_2$ , и это отменяет все построения п.2 предыдущего раздела.

Для трехмерной модели Изинга со свободными границами справедливы формулы (24), а их обобщение на случай взаимодействия с большим числом соседей невозможно. Рассмотрение ситуации со свободными границами на этом завершается. Все дальнейшее относится к периодическим граничным условиям.

2. Аналогично анализируется и трехмерная модель Изинга. Если – как в п.2 предыдущего раздела, – ограничиться взаимодействием со спинами только первых трех координационных сфер все рассуждения этого пункта останутся в силе и для свободных границ. В таком случае образующие блоки первого уровня представляются  $(n \times n)$ -матрицами

$\tilde{\mathbf{A}}_1 = w_1 \tilde{\mathbf{J}}(1)$ ,  $\tilde{\mathbf{B}}_1 = w_1 \mathbf{I} + w_2 \tilde{\mathbf{J}}(1)$ ,  $\tilde{\mathbf{D}}_1 = w_2 \mathbf{I} + w_3 \tilde{\mathbf{J}}(1)$  – см. (11), (19). В базисе из собственных векторов  $\{\tilde{\mathbf{f}}_i\}_1^n$  (9) все три образующих блока одновременно диагонализуются. Тогда  $(n^2 \times n^2)$ -матрицы  $\tilde{\mathbf{A}}_2$  (12) и  $\tilde{\mathbf{B}}_2$  (19) в базисе из кронекеровых произведений  $\tilde{\mathbf{F}}_{ij} = \tilde{\mathbf{f}}_i \otimes \tilde{\mathbf{f}}_j$  примут диагональный вид,  $(n^3 \times n^3)$ -задача на собственные значения примет квазидиагональный вид (20), а выражения для собственных значений  $\tilde{\mu}_{ijk}$  получатся простановкой тильды в нужных местах выражения (21):

## 5. 2D- и 3D-модели Изинга с периодическими граничными условиями: общий случай

Здесь будет развит другой метод вычисления собственных значений многомерного гамильтониана модели Изинга. В вычислительном отношении этот метод более эффективен. Он основан на представлении собственного вектора многомерной задачи в виде кронекерова произведения собственных векторов одномерной модели Изинга (4). Для периодических граничных условий такое представление справедливо всегда.

Планарная решетка является подрешеткой кубической решетки, и расстояния между парами узлов в двух решетках иногда совпадают, но номера узлов при этом разнятся. Чтобы избежать путаницы, которая может возникнуть при использовании в трехмерном случае формул, полученных для планарной модели, здесь мы снабжаем константы взаимодействия дополнительным верхним индексом, указывающим на размерность решетки:  $w_k^{(2)}$  – есть константа взаимодействия со спинами  $k$ -й координационной сферы на планарной решетке, а  $w_k^{(3)}$  – со спинами  $k$ -й координационной сферы на кубической решетке.

### 5.1. Двумерная модель Изинга с пятью ближайшими соседями

Пять наименьших расстояний между узлами квадратной решетки равны  $l_1=1$ ,  $l_2=\sqrt{2}$ ,  $l_3=2$ ,  $l_4=\sqrt{5}$  и  $l_5=2\sqrt{2}$ . На рис.1 цифрами помечены узлы решетки, расстояния до которых от начальной точки в левом нижнем узле равно этим числам.

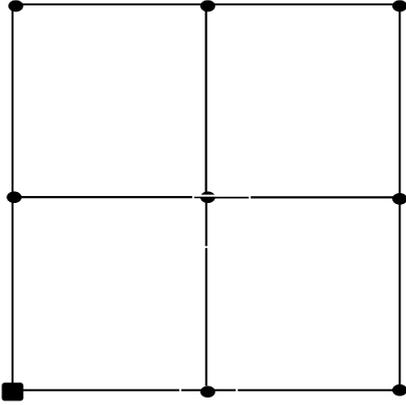


Рис.1. Квадратная решетка: пять ближайших соседей к начальной точке.

Определим  $(n \times n)$ -матрицы  $\mathbf{A}_1$ ,  $\mathbf{B}_1$  и  $\mathbf{C}_1$ :

$$\begin{aligned} \mathbf{A}_1 &= w_1^{(2)}\mathbf{J}(1) + w_3^{(2)}\mathbf{J}(2), \\ \mathbf{B}_1 &= w_1^{(2)}\mathbf{I} + w_2^{(2)}\mathbf{J}(1) + w_4^{(2)}\mathbf{J}(2), \\ \mathbf{C}_1 &= w_3^{(2)}\mathbf{I} + w_4^{(2)}\mathbf{J}(1) + w_5^{(2)}\mathbf{J}(2). \end{aligned} \quad (25)$$

Тогда матрицу гамильтониана можно представить в блочном виде:

$$\mu_{kr} = w_1^{(2)}(\lambda_k + \lambda_r) + w_2^{(2)}\lambda_k\lambda_r + w_3^{(2)}(\lambda_k(2) + \lambda_r(2)) + w_4^{(2)}(\lambda_k\lambda_r(2) + \lambda_r\lambda_k(2)) + w_5^{(2)}\lambda_k(2)\lambda_r(2). \quad (29)$$

Положив здесь  $w_1^{(2)}=1$ ,  $w_2^{(2)}=w_3^{(2)}=w_4^{(2)}=w_5^{(2)}=0$ , получим известные выражения [5], а приравняв нулю только  $w_3^{(2)}$ ,  $w_4^{(2)}$  и  $w_5^{(2)}$  получим формулу (17). Сравнивая (17) и (29) мы видим, что учет взаимодействия с новыми, более удаленными соседями, не влияет на результат, полученный раньше для меньшего числа соседей. Это есть следствие того, что все матрицы  $\mathbf{J}(k)$  одномерной модели имеют один и тот же набор собственных векторов.

### 5.2. Трехмерная модель Изинга с пятью ближайшими соседями

Пять наименьших расстояний между узла-

$$\mathbf{A}_2 = \begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{C}_1 & 0 & \dots & \mathbf{C}_1 & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{C}_1 & \dots & 0 & \mathbf{C}_1 \\ \mathbf{C}_1 & \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \dots & 0 & 0 \\ 0 & \mathbf{C}_1 & \mathbf{B}_1 & \mathbf{A}_1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{C}_1 & 0 & 0 & 0 & \dots & \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{C}_1 & 0 & 0 & \dots & \mathbf{B}_1 & \mathbf{A}_1 \end{pmatrix}. \quad (26)$$

Если  $\mathbf{F}_2$  есть собственный вектор матрицы  $\mathbf{A}_2$ , отвечающее ему собственное значение вычисляется как  $\mu = \mathbf{F}_2^+ \mathbf{A}_2 \mathbf{F}_2$ . Векторы  $\mathbf{F}_2$  будем искать в виде кронекеровых произведений собственных векторов  $\{\mathbf{f}_i\}_i^n$  (4):  $\mathbf{F}_2 = \mathbf{f}_r \otimes \mathbf{f}_k$ . Для собственного значения  $\mu_{kr}$  получаем:

$$\mu_{kr} = \mathbf{f}_k^+ \otimes \mathbf{f}_r^+ \begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{C}_1 & 0 & \dots & \mathbf{C}_1 & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{C}_1 & \dots & 0 & \mathbf{C}_1 \\ \mathbf{C}_1 & \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \dots & 0 & 0 \\ 0 & \mathbf{C}_1 & \mathbf{B}_1 & \mathbf{A}_1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{C}_1 & 0 & 0 & 0 & \dots & \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{C}_1 & 0 & 0 & \dots & \mathbf{B}_1 & \mathbf{A}_1 \end{pmatrix} \mathbf{f}_r \otimes \mathbf{f}_k.$$

Воспользовавшись ассоциативностью матричного умножения, получим:  $\mu_{kr} = \mathbf{f}_k^+ \cdot \mathbf{M} \cdot \mathbf{f}_k$ , где  $(n \times n)$ -матрица

$$\mathbf{M} = (\mathbf{f}_r^+ \mathbf{A}_1 \mathbf{f}_r) \mathbf{I} + (\mathbf{f}_r^+ \mathbf{B}_1 \mathbf{f}_r) \mathbf{J}(1) + (\mathbf{f}_r^+ \mathbf{C}_1 \mathbf{f}_r) \mathbf{J}(2), \quad (27)$$

и, согласно (25), имеем:

$$\begin{aligned} (\mathbf{f}_r^+ \mathbf{A}_1 \mathbf{f}_r) &= w_1^{(2)}\lambda_r(1) + w_3^{(2)}\lambda_r(2), \\ (\mathbf{f}_r^+ \mathbf{B}_1 \mathbf{f}_r) &= w_1^{(2)} + w_2^{(2)}\lambda_r(1) + w_4^{(2)}\lambda_r(2), \\ (\mathbf{f}_r^+ \mathbf{C}_1 \mathbf{f}_r) &= w_3^{(2)} + w_4^{(2)}\lambda_r(1) + w_5^{(2)}\lambda_r(2). \end{aligned} \quad (28)$$

Тогда

$$\mu_{kr} = (\mathbf{f}_r^+ \mathbf{A}_1 \mathbf{f}_r) + (\mathbf{f}_r^+ \mathbf{B}_1 \mathbf{f}_r) \cdot \lambda_k(1) + (\mathbf{f}_r^+ \mathbf{C}_1 \mathbf{f}_r) \cdot \lambda_k(2)$$

и подставляя сюда (28), окончательно получаем:

$$\mu_{kr} = w_1^{(2)}(\lambda_k + \lambda_r) + w_2^{(2)}\lambda_k\lambda_r + w_3^{(2)}(\lambda_k(2) + \lambda_r(2)) + w_4^{(2)}(\lambda_k\lambda_r(2) + \lambda_r\lambda_k(2)) + w_5^{(2)}\lambda_k(2)\lambda_r(2). \quad (29)$$

ми кубической решетки равны  $l_1=1$ ,  $l_2=\sqrt{2}$ ,  $l_3=\sqrt{3}$ ,  $l_4=2$  и  $l_5=\sqrt{5}$  - соответствующие узлы показаны на рис. 2.

Сопоставим рисунки 1 и 2. Расстояние  $l_1$  до ближайшего соседа на двумерной решетке совпадает с аналогичным расстоянием на трехмерной решетке - следовательно, необходимо отождествить константы  $w_1^{(2)}$  и  $w_1^{(3)}$ . Анализируя рисунки 1 и 2 нетрудно отождествить и другие пары констант взаимодействия:

$$w_1^{(2)} \equiv w_1^{(3)}, w_2^{(2)} \equiv w_2^{(3)}, w_3^{(2)} \equiv w_4^{(3)}, w_4^{(2)} \equiv w_5^{(3)}. \quad (30)$$

Нам потребуются эти тождества, чтобы использовать для трехмерной модели выражения

(25), полученные для двумерной модели Изинга. Поскольку пятая координационная сфера планарной модели не попадает в число первых пяти координационных сфер кубической модели, константу  $w_5^{(2)}$  положим равной нулю.

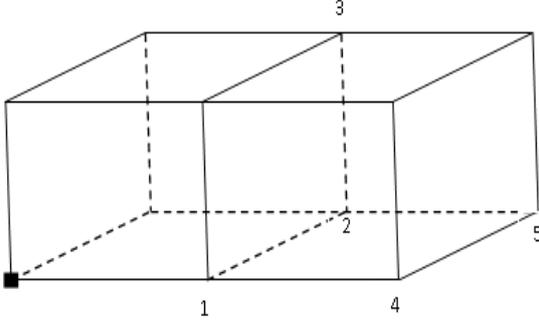


Рис. 2. Кубическая решетка: пять ближайших соседей к начальной точке.

Гамильтониан 3D-модели Изинга задается матрицей  $\mathbf{A}_3$ , которую удобно представить в блочном виде, используя  $(n^2 \times n^2)$ -матрицы  $\mathbf{A}_2$ ,  $\mathbf{B}_2$  и  $\mathbf{C}_2$ :

$$\mathbf{A}_3 = \begin{pmatrix} \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & 0 & \dots & \mathbf{C}_2 & \mathbf{B}_2 \\ \mathbf{B}_2 & \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & \dots & 0 & \mathbf{C}_2 \\ \mathbf{C}_2 & \mathbf{B}_2 & \mathbf{A}_2 & \mathbf{B}_2 & \dots & 0 & 0 \\ 0 & \mathbf{C}_2 & \mathbf{B}_2 & \mathbf{A}_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{C}_2 & 0 & 0 & 0 & \dots & \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{B}_2 & \mathbf{C}_2 & 0 & 0 & \dots & \mathbf{B}_2 & \mathbf{A}_2 \end{pmatrix}.$$

Матрица  $\mathbf{A}_2$  имеет в точности вид матрицы (26), а ее образующие  $(n \times n)$ -матрицы отличаются от (25) только другими обозначениями констант взаимодействия – см. тождества (30):

$$\begin{aligned} \mathbf{A}_1 &= w_1^{(3)} \mathbf{J}(1) + w_4^{(3)} \mathbf{J}(2), \\ \mathbf{B}_1 &= w_1^{(3)} \mathbf{I} + w_2^{(3)} \mathbf{J}(1) + w_5^{(3)} \mathbf{J}(2), \\ \mathbf{C}_1 &= w_4^{(3)} \mathbf{I} + w_5^{(3)} \mathbf{J}(1). \end{aligned}$$

Две остальные  $(n^2 \times n^2)$ -матрицы даются вы-

$$\begin{aligned} \mu_{krm} &= w_1^{(3)} (\lambda_k + \lambda_r + \lambda_m) + w_2^{(3)} (\lambda_k \lambda_r + \lambda_r \lambda_m + \lambda_m \lambda_k) + w_3^{(3)} \lambda_k \lambda_r \lambda_m + w_4^{(3)} [\lambda_k(2) + \lambda_r(2) + \lambda_m(2)] + \\ &+ w_5^{(3)} [(\lambda_k + \lambda_r) \lambda_m(2) + (\lambda_k + \lambda_m) \lambda_r(2) + (\lambda_r + \lambda_m) \lambda_k(2)], \quad k, r, m = 1, \dots, n. \end{aligned} \quad (32)$$

## 6. Обсуждение и выводы

В работе получены аналитические выражения для собственных значений 2D- и 3D-модели Изинга, когда учитывается взаимодействие со спинами нескольких координационных сфер. Собственные значения многомерной задачи

ражениями:

$$\mathbf{B}_2 = \begin{pmatrix} \mathbf{B}_1 & \mathbf{D}_1 & \mathbf{E}_1 & 0 & \dots & \mathbf{E}_1 & \mathbf{D}_1 \\ \mathbf{D}_1 & \mathbf{B}_1 & \mathbf{D}_1 & \mathbf{E}_1 & & 0 & \mathbf{E}_1 \\ \mathbf{E}_1 & \mathbf{D}_1 & \mathbf{B}_1 & \mathbf{D}_1 & & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{D}_1 & \mathbf{E}_1 & 0 & 0 & \dots & \mathbf{D}_1 & \mathbf{B}_1 \end{pmatrix} \text{ и}$$

$$\mathbf{C}_2 = \begin{pmatrix} \mathbf{C}_1 & \mathbf{E}_1 & 0 & \dots & \mathbf{E}_1 \\ \mathbf{E}_1 & \mathbf{C}_1 & \mathbf{E}_1 & \dots & 0 \\ 0 & \mathbf{E}_1 & \mathbf{C}_1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{E}_1 & 0 & 0 & \dots & \mathbf{C}_1 \end{pmatrix},$$

где

$$\mathbf{D}_1 = w_2^{(3)} \mathbf{I} + w_3^{(3)} \mathbf{J}(1), \quad \mathbf{E}_1 = w_5^{(3)} \mathbf{I}.$$

Собственные векторы  $\mathbf{F}_3$  матрицы  $\mathbf{A}_3$  будем искать в виде кронекеровых произведений собственных векторов одномерной и двумерной моделей:

$$\mathbf{F}_3 = \mathbf{F}_2 \otimes \mathbf{f}_m, \quad \text{где } \mathbf{F}_2 = \mathbf{f}_r \otimes \mathbf{f}_k, \quad k, r, m = 1, 2, \dots, n.$$

Тогда для собственного значения  $\mu_{krm} = \mathbf{F}_3^+ \mathbf{A}_3 \mathbf{F}_3$  получим:

$$\begin{aligned} \mu_{krm} &= (\mathbf{F}_2^+ \mathbf{A}_2 \mathbf{F}_2) + (\mathbf{F}_2^+ \mathbf{B}_2 \mathbf{F}_2) \lambda_m(1) \\ &+ (\mathbf{F}_2^+ \mathbf{C}_2 \mathbf{F}_2) \lambda_m(2). \end{aligned} \quad (31)$$

Для вычисления  $(\mathbf{F}_2^+ \mathbf{A}_2 \mathbf{F}_2)$  воспользуемся полученным ранее выражением (29), в котором надо скорректировать веса согласно (30):

$$\begin{aligned} (\mathbf{F}_{kr}^+ \mathbf{A}_2 \mathbf{F}_{kr}) &= w_1^{(3)} (\lambda_k + \lambda_r) + w_2^{(3)} \lambda_k \lambda_r + \\ &+ w_4^{(3)} (\lambda_k(2) + \lambda_r(2)) + w_5^{(3)} (\lambda_k \lambda_r(2) + \lambda_r \lambda_k(2)). \end{aligned}$$

Значения величин  $(\mathbf{F}_2^+ \mathbf{B}_2 \mathbf{F}_2)$  и  $(\mathbf{F}_2^+ \mathbf{C}_2 \mathbf{F}_2)$  вычисляются аналогично тому, как в предыдущем пункте вычислялось значение  $(\mathbf{F}_2^+ \mathbf{A}_2 \mathbf{F}_2)$ . В результате получим:

$$\begin{aligned} (\mathbf{F}_{kr}^+ \mathbf{B}_2 \mathbf{F}_{kr}) &= w_5^{(3)} + w_2^{(3)} (\lambda_k + \lambda_r) + \\ &+ w_3^{(3)} \lambda_k \lambda_r + w_5^{(3)} (\lambda_k(2) + \lambda_r(2)) \end{aligned}$$

$$\text{и } (\mathbf{F}_{kr}^+ \mathbf{C}_2 \mathbf{F}_{kr}) = w_4^{(3)} + w_5^{(3)} (\lambda_k + \lambda_r).$$

Подставляя последние три выражения в (31), окончательно имеем:

выражаются через константы взаимодействия  $\{w_r\}_1^k$  между спинами и собственные значения  $\{\lambda_r\}_1^n$  одномерной модели Изинга. Отметим, что выражения для собственных значений  $\{\lambda_r\}_1^n$  хорошо известны как для периодических граничных условий, так и для свободных границ.

Однако уже для одномерной модели Изинга эти два типа граничных условий приводят к кардинально различным результатам. Затем эти различия транслируются на многомерную модель Изинга.

Обсудим сначала случай периодических граничных условий. Напомним, что  $\mathbf{J}(r)$  есть матрица связи одномерной модели, в которой учитывается взаимодействие только со спинами  $r$ -й координационной сферы (см. раздел 1). Все матрицы  $\mathbf{J}(r)$  имеют один и то же набор собственных векторов, а их собственные значения  $\lambda_i(r) = P_r(\lambda_i)$ , где  $P_r(x)$  есть полином  $r$ -й степени от  $\lambda_i$ . Тогда собственные значения гамильтониана одномерной модели, в котором учитывается взаимодействие со спинами  $k$  ближайших координационных сфер -  $\mathbf{A}_1 = w_1\mathbf{J}(1) + \dots + w_k\mathbf{J}(k)$ , - есть взвешенная сумма собственных значений  $\lambda_i(r)$ :

$$\lambda_i(\mathbf{A}_1) = \sum_{r=1}^k w_r \lambda_i(r), \quad i=1, 2, \dots, n. \quad (33)$$

Это свойство одномерной модели позволяет и для многомерной модели Изинга учесть взаимодействие со спинами любого числа координационных сфер, и в явном виде выразить собственные значения гамильтониана через  $\{\lambda_i\}_1^n$  и  $\{w_r\}_1^k$  - см. выражения (29) и (32). Важно, что добавление взаимодействий с новыми соседями сопровождается лишь появлением новых слагаемых в выражении для собственных значений, а слагаемые, полученные ранее для меньшего числа соседей, никак не изменяются.

Задавшись тем или иным вариантом функциональной зависимости константы взаимодействия от расстояния, можно с помощью (33) получить асимптотическое выражение для спектральной плотности одномерного гамиль-

тониана. Затем этот результат можно распространить на двумерную и трехмерную решетки - мы планируем сделать это в следующей работе.

В случае свободных границ ситуация кардинально отличается уже для одномерной модели: матрицы  $\tilde{\mathbf{J}}(r)$  не коммутируют друг с другом. Все они обладают различными наборами собственных векторов, и даже для простейшей линейной комбинации этих матриц  $\tilde{\mathbf{A}}_1 = w_1\tilde{\mathbf{J}}(1) + w_2\tilde{\mathbf{J}}(2) + \dots + w_k\tilde{\mathbf{J}}(k)$  собственные значения невозможно выразить через собственные значения матриц  $\tilde{\mathbf{J}}(r)$ . То же оказывается справедливым и в многомерном случае - за двумя исключениями (см. раздел 4): для свободных границ собственные значения двумерной модели Изинга выражаются через собственные значения  $\tilde{\lambda}_i$ , когда учитывается взаимодействие только со спинами двух ближайших координационных сфер - см. (22); для трехмерной модели Изинга можно учесть взаимодействие со спинами трех ближайших координационных сфер, и собственные значения гамильтониана будут выражаться через  $w_1, w_2, w_3$  и  $\tilde{\lambda}_i$  - см. (24).

Все эти результаты естественным образом обобщаются для модели Изинга на гиперкубе произвольной размерности  $d > 3$ . Различия между случаями свободных границ и периодических граничных условий при этом сохраняются.

Работа была поддержана грантом РФФИ № 18-07-00750. Авторы благодарны И.М. Кагановой и М.Л. Литинской за полезные дискуссии и помощь в создании статьи.

## Eigenvalues of Ising Hamiltonian with long-range interactions

B.V. Kryzhanovskiy and L.B. Litinskii

**Abstract.** We obtained analytically eigenvalues of a multidimensional Ising Hamiltonian on a hypercube lattice and expressed them in terms of spin-spin interaction constants and the eigenvalues of the one-dimensional Ising Hamiltonian (the latter are well known). To do this we wrote down the multidimensional Hamiltonian eigenvectors as the Kronecker products of the eigenvectors of the one-dimensional Ising Hamiltonian. For periodic boundary conditions, it is possible to obtain exact results taking into account interactions with an unlimited number of neighboring spins. In this paper, we present exact expressions for the eigenvalues for the planar and cubic Ising systems accounting for the first five coordination spheres (that is interactions with the nearest neighbors, the next neighbors, the next-next neighbors, the next-next-next neighbors and the next-next-next-next neighbors). In the case of free-boundary systems, we showed that in the two- and three-dimensions the exact expressions could be

obtained only if we account for interactions with spins of first two coordination spheres and first three coordination spheres, respectively.

**Keywords:** Ising Model, hypercubic lattice, eigenvalues and eigenvectors

## Литература

1. R.J. Baxter. Exactly solved models in statistical mechanics. London, Academic Press, 1982.
2. H.E. Stanley. Introduction to Phase Transitions and Critical Phenomena. Oxford University Press, 1987.
3. V.S. Dotsenko. Introduction to the Theory of Spin-Glasses and Neural Networks. Singapore: World Scientific, 1994.
4. New Optimization Algorithms in Physics. Eds.: A.K. Hartmann and H. Rieger, Weinheim, WILEY-VCH Verlag GmbH & Co. KGaA, 2004.
5. J.M. Dixon, J.A. Tuszynski, M.L.A. Nip. Exact eigenvalues of the Ising Hamiltonian in one-, two- and three-dimensions in the absence of magnetic field. "Physica A", V. 289 (2001), 137-156.
6. Robert S. Knox, Albert Gold (Eds.). Symmetry in the solid state. NY, W.A. Benjamin, Inc., 1964.
7. Joseph Callaway. Energy band theory. NY and London, Academic Press, 1964.
8. S. Nag and A. Garg. Many body localization in the presence of long-range interactions and long-range hopping. "Phys. Rev. B", V. 99 (2019), p. 224203.
9. Kelly Boothby, Paul Bunyk, Jack Raymond, Aidan Roy. Next-Generation Topology of D-Wave Quantum Processors. Technical report, #2019-02-25.  
[https://www.dwavesys.com/sites/default/files/14-1026A-C\\_Next-Generation-Topology-of-DW-Quantum-Processors.pdf](https://www.dwavesys.com/sites/default/files/14-1026A-C_Next-Generation-Topology-of-DW-Quantum-Processors.pdf)
10. М.В. Волькенштейн, Л.А. Грибов, М.А. Ельяшевич, Б.И. Степанов. Колебания молекул. Москва, Наука, 1972.
11. B. V. Kryzhanovsky, L. B. Litinskii. Connection-Matrix Eigenvalues in the Ising Model: Taking into Account Interaction with Next-Nearest Neighbors. "Doklady Physics", V. 64 (2019), P. 414–417.
12. H. Fidler, J. Knoester, D.A. Wiersma. Optical properties of disordered molecular aggregates: a numerical study. "J. Chem. Phys." v. 95 (1991), # 11, P.7880-7890.
13. U. Grenander and G. Szego. Toeplitz forms and their applications. Berkley and Los Angeles, Univ. of California Press, 1958

# Спектр матрицы связи модели Изинга с произвольным дальним действием

Б.В. Крыжановский<sup>1</sup>, Л.Б. Литинский<sup>2</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, [kryzhanov@mail.ru](mailto:kryzhanov@mail.ru)

<sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, [litin@mail.ru](mailto:litin@mail.ru)

**Аннотация.** Изучается матрица связи модели Изинга на  $d$ -мерной гиперкубической решетке с произвольным дальним действием и периодическими граничными условиями. Собственные векторы матрицы связи  $d$ -мерной модели вычисляются в виде кронекеровых произведений собственных векторов матрицы связи одномерной модели Изинга, которые хорошо известны. Получены строгие выражения для собственных значений в виде тригонометрических полиномов степени  $d$ , коэффициентами в которых являются константы взаимодействия между спинами. Установлено, что вырождение собственных значений спектра  $d$ -мерной модели не снимается при включении дальнего действия.

**Ключевые слова:** модель Изинга, гиперкубическая решетка, кронекерово произведение, собственные значения

## 1. Введение

Модель Изинга около 100 лет находится в фокусе внимания исследователей. Она дает идеализированное описание системы взаимодействующих спинов, распределенных по узлам регулярной решетки. Данная модель является удобным объектом для отработки новых методов вычисления свободной энергии и критических характеристик. За годы исследований для нее получен целый ряд точных результатов, на которые можно ориентироваться при разработке новых подходов. Большая часть полученных результатов относится к ситуации, когда в модели учитывается взаимодействие только между ближайшими соседями (nearest neighbors) [1], [2]. За очень редким исключением [3], почти все работы прошлых лет касались модели Изинга с короткодальним действием. В последние годы сформировался устойчивый интерес к тому, как на критических характеристиках сказывается учет взаимодействия с более далекими соседями. В значительной части этих работ исследуется влияние соседей, следующих за ближайшими (next nearest neighbors) [4-8]. Однако имеются исследования, исследующие влияние и более далеких соседей [9, 10]. Настоящая работа является откликом на возникший интерес к модели Изинга с дальним действием.

Подчеркнем, что у нас изучается не сама модель Изинга, а только отвечающая ей матрица связи. Нами получены строгие выражения для собственных значений матрицы связи модели Изинга на  $d$ -мерной гиперкубической решетке, когда учитывается взаимодействие с

произвольно большим числом соседей. Чтобы обеспечить трансляционную инвариантность системы граничные условия всегда выбираются периодическими. В этом случае собственные значения матрицы связи  $d$ -мерной модели удастся выразить через константы взаимодействия между спинами и тригонометрические полиномы. Все рассуждения ведутся для случая анизотропного взаимодействия. В настоящей работе развиваются результаты, полученные ранее в [11, 12].

В следующем разделе задача решается для матриц связи одномерной модели Изинга, которая описывает линейную цепочку спинов. Основным результатом этого раздела заключается в том, что все матрицы связи одномерной модели - учитывающие взаимодействие со все более далекими соседями, - коммутируют друг с другом. Следовательно, все они имеют один и тот же набор собственных векторов, что сильно облегчает дальнейшие рассуждения. В третьем разделе исследуется матрица связи двумерной спиновой системы, когда спины находятся в узлах квадратной решетки на плоскости. Здесь все основано на том, что матрицу связи двумерной модели можно представить в виде линейной комбинации кронекеровых произведений матриц связи одномерной модели. Четвертый раздел посвящен матрице связи 3D-модели Изинга, когда спины помещаются в узлах трехмерной кубической решетки. Здесь получает дальнейшее развитие прием, разработанный в предыдущем разделе для матрицы связи 2D-модели Изинга. В пятом разделе полученные результаты обобщаются для матрицы связи  $d$ -мерной модели Изинга, когда взаимодействуют

щие спины помещаются в узлах гиперкубической решетки произвольной размерности  $d$ . В заключительном разделе обсуждаются некоторые общие свойства спектра матрицы связи модели Изинга, и их зависимость от размерности решетки  $d$ .

## 2. Линейная цепочка спинов

Рассмотрим характерную для одномерной модели Изинга линейную цепочку взаимодействующих спинов. Граничные условия предполагаются периодическими, а число спинов  $L$  - нечётным:  $L=2l+1$ . Условимся, что расстояние между узлами одномерной решетки равно 1.

Учет взаимодействия с  $k$ -ми соседями осуществляется с помощью симметричной матрицы  $\mathbf{J}(k)$ , у которой на  $k$ -й и  $(L-k)$ -й диагоналях, параллельных главной, стоят единицы. Остальные матричные элементы равны нулю. Таким образом,  $\mathbf{J}(k)$  - это симметричная теплица  $(L \times L)$  матрица,  $(l+1)$ -я строка которой имеет вид:

$$(0 \ \dots \ 1 \ \dots \ 0 \ \dots \ 1 \ \dots \ 0) -$$

две единицы в строке располагаются на расстоянии  $k$  от центра. Остальные строки получают последовательными циклическими сдвигами данной строки: сдвиг на одну позицию влево дает  $l$ -ю строку, сдвиг на одну позицию вправо дает  $(l+2)$ -ю строку и так далее. Например, матрицы  $\mathbf{J}(1)$  и  $\mathbf{J}(2)$  для  $L=7$  имеют вид:

$$\mathbf{J}(1) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\mathbf{J}(2) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Легко показать [11, 12], что матрицы  $\mathbf{J}(k)$  коммутируют друг с другом - следовательно, все они обладают одним и тем же набором соб-

ственных векторов  $\{\mathbf{f}_\alpha\}$ ,  $\alpha=1,2,\dots,L$ . Компонентам собственного вектора  $\mathbf{f}_\alpha$  можно придать следующий вид:

$$f_\alpha^{(i)} = \frac{\cos \varphi_{\alpha i} + \sin \varphi_{\alpha i}}{\sqrt{L}}, \quad \varphi_{\alpha i} = \frac{2\pi(\alpha-1)(i-1)}{L}, \quad (1)$$

где  $\alpha, i=1,2,\dots,L$ .

Каждая матрица  $\mathbf{J}(k)$  имеет свой набор собственных значений  $\{\lambda_\alpha(k)\}_{\alpha=1}^L$ , для которых будем указывать номер  $k$  матрицы, с которой они связаны:

$$\mathbf{J}(k)\mathbf{f}_\alpha = \lambda_\alpha(k)\mathbf{f}_\alpha, \quad \lambda_\alpha(k) = 2 \cos\left(\frac{2\pi k(\alpha-1)}{L}\right), \quad (2)$$

где  $\alpha=1,2,\dots,L$  и  $k=1,2,\dots,l$ .

Пусть  $w(k)$  - величина взаимодействия между спинами, отстоящими друг от друга на расстояние  $k$ ,  $k=1,2,\dots,l$ . Запишем матрицу связи, в которой учитывается взаимодействие  $w(1)$  с ближайшими соседями, взаимодействие  $w(2)$  со вторыми соседями и так далее - взаимодействие  $w(l)$  с последней,  $l$ -й парой соседей:

$$\mathbf{A}_0 = \sum_{k=1}^l w(k)\mathbf{J}(k). \quad (3)$$

Естественно, некоторые коэффициенты  $w(k)$  могут равняться нулю. Для одномерной модели Изинга выражение (3) есть самый общий вид матрицы связи. Собственные векторы этой матрицы также даются выражениями (1), а соответствующие собственные значения равны:

$$\mu_\alpha = \sum_{k=1}^l w(k)\lambda_\alpha(k), \quad \alpha=1,2,\dots,L. \quad (4)$$

Для общности выражений условимся на будущее о следующих обозначениях:

$$\mathbf{J}(0) = \mathbf{I}, \quad \lambda_\alpha(0) = 1, \quad \alpha=1,2,\dots,L,$$

где  $\mathbf{I}$  есть единичная  $L \times L$ -матрица, а  $\lambda_\alpha(0) \equiv 1$  - ее собственные значения.

## 3. Планарная спиновая система

Рассмотрим характерную для 2D-модели Изинга систему спинов, распределенных по узлам квадратной решетки. Обозначим через  $w(m,k)$  константу взаимодействия между спинами, сдвинутыми друг относительно друга на расстояние  $m$  по одной оси решетки и на расстоянии  $k$  по другой. Связи в такой системе описываются  $L^2 \times L^2$ -матрицей  $\mathbf{B}_0$ , которую удобно записать в виде блочной  $L \times L$ -матрицы:

$$\mathbf{V}_0 = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \dots & \mathbf{A}_l & \dots & \mathbf{A}_2 & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_0 & \dots & \mathbf{A}_{l-1} & \dots & \mathbf{A}_3 & \mathbf{A}_2 \\ \mathbf{A}_2 & \mathbf{A}_1 & \dots & \mathbf{A}_{l-2} & \dots & \mathbf{A}_4 & \mathbf{A}_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{A}_l & \mathbf{A}_{l-1} & \dots & \mathbf{A}_0 & \dots & \mathbf{A}_{l-1} & \mathbf{A}_l \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{A}_2 & \mathbf{A}_3 & \dots & \mathbf{A}_{l-1} & \dots & \mathbf{A}_0 & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_l & \dots & \mathbf{A}_1 & \mathbf{A}_0 \end{pmatrix}. \quad (5)$$

Здесь  $L \times L$ -матрицы  $\mathbf{A}_m$  имеют вид:

$$\mathbf{A}_m = \sum_{k=0}^l w(m,k) \mathbf{J}(k), \quad m=0,1,\dots,l. \quad (6)$$

Поскольку самодействие в системе отсутствует, здесь и далее полагаем  $w(0,0) = 0$ .

Центральная блочная строка матрицы  $\mathbf{V}_0$  имеет вид

$$(\mathbf{A}_l \ \mathbf{A}_{l-1} \ \dots \ \mathbf{A}_1 \ \mathbf{A}_0 \ \mathbf{A}_1 \ \dots \ \mathbf{A}_{l-1} \ \mathbf{A}_l),$$

а остальные строки получаются ее циклическими перестановками.

Матрицы  $\mathbf{A}_m$  (6) имеют следующий смысл. Двумерную спиновую систему можно рассматривать как набор взаимодействующих одномерных цепочек (например, горизонтальных). Тогда матрица  $\mathbf{A}_0$  описывает взаимодействие между спинами, находящимися в пределах одной горизонтальной цепочки. Матрица  $\mathbf{A}_m$  описывает взаимодействие между спинами, находящимися в разных цепочках, сдвинутых друг относительно друга (по вертикали) на  $m$  узлов.

Отметим, что матрица  $\mathbf{V}_0$  имеет вид блочной тёплицевой матрицы, у которой на главной диагонали стоят  $(L \times L)$ -матрицы  $\mathbf{A}_0$ , а на  $m$ -й и  $(L-m)$ -й диагоналях стоят матрицы  $\mathbf{A}_m$ . Тогда матрицу  $\mathbf{V}_0$  можно представить в виде:

$$\mathbf{V}_0 = \sum_{m=0}^l \mathbf{J}(m) \otimes \mathbf{A}_m = \sum_{m=0}^l \sum_{k=0}^l w(m,k) \cdot \mathbf{J}(m) \otimes \mathbf{J}(k). \quad (7)$$

Ввиду коммутативности матриц  $\mathbf{A}_m$  собственные векторы матрицы  $\mathbf{V}_0$  можно представить в виде кронекерова произведения собственных векторов одномерной модели:

$$\mathbf{F}_{\alpha\beta} = \mathbf{f}_\alpha \otimes \mathbf{f}_\beta, \quad \alpha, \beta = 1, 2, \dots, L. \quad (8)$$

Переходя к вычислению собственных значений матрицы  $\mathbf{V}_0$ , напомним правило произведения матриц и векторов, представленных в виде кронекеровых произведений [13]. Если матрица  $\mathbf{M} = \mathbf{M}_1 \otimes \mathbf{M}_2$  а вектор  $\mathbf{F} = \mathbf{F}_1 \otimes \mathbf{F}_2$ , то:

$$\mathbf{F}^+ \mathbf{M} \mathbf{F} \equiv \mathbf{F}_1^+ \mathbf{M}_1 \mathbf{F}_1 \cdot \mathbf{F}_2^+ \mathbf{M}_2 \mathbf{F}_2. \quad (9)$$

Итак, задачу на собственные значения для

матрицы  $\mathbf{V}_0$  мы свели к вычислению величины  $\mu_{\alpha\beta} = \mathbf{F}_{\alpha\beta}^+ \mathbf{V}_0 \mathbf{F}_{\alpha\beta}$ . Подставляя сюда  $\mathbf{V}_0$  в виде (7) и  $\mathbf{F}_{\alpha\beta}$  в виде (8), и используя тождество (9), получим:

$$\mu_{\alpha\beta} = \sum_{m=0}^l \sum_{k=0}^l w(m,k) \cdot \lambda_\alpha(m) \cdot \lambda_\beta(k), \quad (10)$$

где  $\alpha, \beta = 1, 2, \dots, L$ .

Напомним, что  $w(0,0) = 0$  и  $\lambda_\alpha(0) \equiv 1$ .

Как видим, собственные значения матрицы связи, отвечающей 2D-модели Изинга, являются однородными полиномами второй степени от собственных значений одномерной задачи. Коэффициентами полинома являются константы взаимодействия  $w(m,k)$ .

Например, рассмотрим двумерную спиновую систему с изотропным взаимодействием:

$$w(k,m) = w(m,k).$$

Пусть отличным от нуля будет взаимодействие только с ближайшими и со вторыми соседями:  $w(0,1) = w(1,0) = w_1$ , и  $w(1,1) = w_2$ . В этом случае выражение (10) принимает более обобщимый вид:

$$\mu_{\alpha\beta} = w_1 (\lambda_\alpha(1) + \lambda_\beta(1)) + w_2 \lambda_\alpha(1) \lambda_\beta(1),$$

$\alpha, \beta = 1, 2, \dots, L$ .

Ранее это выражение получено в [11,12], где использовался частный случай настоящего подхода.

## 4. Кубическая решетка

Рассмотрим характерную для 3D-модели Изинга систему взаимодействующих спинов, сидящих в узлах кубической решетки размерами  $L \times L \times L$ . Обозначим через  $w(n,m,k)$  константу взаимодействия между спинами, сдвинутыми друг относительно друга на расстояние  $n$  по одной оси решетки, на расстояние  $m$  по другой оси и на расстояние  $k$  по третьей оси. Отметим, что взаимодействие не предполагается изотропным. Связи в такой системе описываются  $L^3 \times L^3$ -матрицей  $\mathbf{C}_0$ , которую удобно записать в блочном виде:

$$\mathbf{C}_0 = \begin{pmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \dots & \mathbf{B}_l & \dots & \mathbf{B}_2 & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{B}_0 & \dots & \mathbf{B}_{l-1} & \dots & \mathbf{B}_3 & \mathbf{B}_2 \\ \mathbf{B}_2 & \mathbf{B}_1 & \dots & \mathbf{B}_{l-2} & \dots & \mathbf{B}_4 & \mathbf{B}_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{B}_l & \mathbf{B}_{l-1} & \dots & \mathbf{B}_0 & \dots & \mathbf{B}_{l-1} & \mathbf{B}_l \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{B}_2 & \mathbf{B}_3 & \dots & \mathbf{B}_{l-1} & \dots & \mathbf{B}_0 & \mathbf{B}_1 \\ \mathbf{B}_1 & \mathbf{B}_2 & \dots & \mathbf{B}_l & \dots & \mathbf{B}_1 & \mathbf{B}_0 \end{pmatrix}, \quad (11)$$

где  $\mathbf{B}_n$  есть  $(L^2 \times L^2)$ -матрицы, а  $n = 0, 1, 2, \dots, l$ .

Для получения  $\mathbf{B}_n$  необходимо сначала построить набор  $(L \times L)$ -матриц  $\mathbf{A}_m^{(n)}$ , аналогичных (6):

$$\mathbf{A}_m^{(n)} = \sum_{k=0}^l w(n, m, k) \mathbf{J}(k), \quad (12)$$

где  $n, m = 0, 1, 2, \dots, l$  и  $\mathbf{J}(0) = \mathbf{I}$ . Поскольку самодействие в системе отсутствует, положим  $w(0, 0, 0) = 0$ .

Затем, с помощью матриц  $\mathbf{A}_m^{(n)}$  (12) строим набор матриц  $\mathbf{B}_n$ :

$$\mathbf{B}_n = \sum_{m=0}^l \mathbf{J}(m) \otimes \mathbf{A}_m^{(n)} = \sum_{m, k=0}^l w(n, m, k) \mathbf{J}(m) \otimes \mathbf{J}(k). \quad (13)$$

Матрицы  $\mathbf{B}_n$  (13) имеют следующий смысл. Трехмерную систему можно рассматривать как набор взаимодействующих планарных решеток. Тогда матрица  $\mathbf{B}_0$  описывает взаимодействие спинов в пределах одной (например, горизонтальной) плоскости, а матрица  $\mathbf{B}_n$  ( $n \neq 0$ ) описывает взаимодействие спина из одной плоскости со спинами другой плоскости, сдвинутой по вертикали на  $n$  узлов.

Как видим, матрица  $\mathbf{C}_0$  имеет вид блочной теплицевой матрицы, у которой на главной диагонали стоят матрицы  $\mathbf{B}_0$ , а на  $n$ -й и  $(L-n)$ -й диагоналях стоят матрицы  $\mathbf{B}_n$  с  $n \neq 0$ . Матрицу  $\mathbf{C}_0$  можно представить в виде:

$$\begin{aligned} \mathbf{C}_0 &= \sum_{n=0}^l \mathbf{J}(n) \otimes \mathbf{B}_n = \\ &= \sum_{n=0}^l \sum_{m=0}^l \sum_{k=0}^l w(n, m, k) \cdot \mathbf{J}(n) \otimes \mathbf{J}(m) \otimes \mathbf{J}(k). \end{aligned}$$

Ввиду коммутативности матриц  $\mathbf{B}_n$  собственные векторы матрицы  $\mathbf{C}_0$  представимы в виде кронекерова произведения собственных векторов одномерной модели:

$$\mathbf{F}_{\alpha\beta\gamma} = \mathbf{f}_\alpha \otimes \mathbf{f}_\beta \otimes \mathbf{f}_\gamma, \quad \alpha, \beta, \gamma = 1, 2, \dots, L.$$

Тогда задача на собственные значения для матрицы  $\mathbf{C}_0$  сводится к вычислению  $\mu_{\alpha\beta\gamma} = \mathbf{F}_{\alpha\beta\gamma}^+ \mathbf{C}_0 \mathbf{F}_{\alpha\beta\gamma}$ , и с учетом равенства (9) получаем:

$$\mu_{\alpha\beta\gamma} = \sum_{n=0}^l \sum_{m=0}^l \sum_{k=0}^l w(n, m, k) \lambda_\alpha(n) \lambda_\beta(m) \lambda_\gamma(k). \quad (14)$$

Собственные значения матрицы связи для трехмерной модели Изинга представляются в виде однородного полинома третьей степени от собственных значений одномерной модели, коэффициентами которого являются константы взаимодействия между спинами.

Для примера, рассмотрим частный случай 3D-модели Изинга с изотропным взаимодей-

ствием

$$w(n, m, k) = w(k, n, m) = w(m, n, k),$$

причем положим отличным от нуля только взаимодействие с ближайшими соседями:

$$w(0, 0, 1) = w(0, 1, 0) = w(1, 0, 0) = w_1,$$

со следующими соседями:

$$w(0, 1, 1) = w(1, 0, 1) = w(1, 1, 0) = w_2,$$

и с третьими соседями:

$$w(1, 1, 1) = w_3.$$

В этом случае выражение (14) примет более обозримый вид:

$$\begin{aligned} \mu_{\alpha\beta\gamma} &= w_1 (\lambda_\alpha(1) + \lambda_\beta(1) + \lambda_\gamma(1)) + \\ &+ w_2 (\lambda_\alpha(1) \cdot \lambda_\beta(1) + \lambda_\alpha(1) \cdot \lambda_\gamma(1) + \lambda_\beta(1) \cdot \lambda_\gamma(1)) + \\ &+ w_3 \cdot \lambda_\alpha(1) \cdot \lambda_\beta(1) \cdot \lambda_\gamma(1). \end{aligned}$$

Ранее это выражение было получено в [11,12].

## 5. d-мерный гиперкуб

Обобщение полученных выражений на случай  $d$ -мерной решетки очевидно. Для начала зададимся константами взаимодействия  $w(k_1, k_2, \dots, k_d)$  между спинами, сдвинутыми друг относительно друга на расстояние  $k_1$  по одной из осей решетки, на расстояние  $k_2$  по другой и т.д. Затем строится набор  $(L \times L)$ -матриц

$$\mathbf{A}_{k_2}^{(k_3, \dots, k_d)} = \sum_{k_1=0}^l w(k_d, \dots, k_2, k_1) \cdot \mathbf{J}(k_1),$$

где  $k_2, \dots, k_d = 0, 1, \dots, l$ .

С помощью этих матриц строится набор  $(L^2 \times L^2)$ -матриц

$\mathbf{B}_{k_3}^{(k_4, \dots, k_d)} = \sum_{k_2=0}^l \mathbf{J}(k_2) \otimes \mathbf{A}_{k_2}^{(k_3, \dots, k_d)}$ . Затем, с помощью этих матриц  $\mathbf{B}_{k_3}^{(k_4, \dots, k_d)}$ , строятся матрицы трехмерной модели

$\mathbf{C}_{k_4}^{(k_5, \dots, k_d)} = \sum_{k_3=0}^l \mathbf{J}(k_3) \otimes \mathbf{B}_{k_3}^{(k_4, \dots, k_d)}$  - и так далее до тех пор, пока будет построена матрица связи  $d$ -мерной модели  $\mathbf{U}_0$ . Эта матрица, представленная в блочном виде через  $(L^{d-1} \times L^{d-1})$ -матрицы  $(d-1)$ -мерной модели, имеет вид, аналогичный выражениям (5) или (11). Опуская стандартные промежуточные преобразования, для матрицы  $\mathbf{U}_0$  получим:

$$\mathbf{U}_0 = \sum_{k_1, \dots, k_d=0}^l w(k_1, k_2, \dots, k_d) \mathbf{J}(k_1) \otimes \mathbf{J}(k_2) \dots \otimes \mathbf{J}(k_d), \quad (15)$$

где как всегда  $w(0, 0, \dots, 0) = 0$ .

Ввиду коммутативности входящих в выражение (15) матриц  $\mathbf{J}(k)$ , собственные векторы

матрицы  $\mathbf{U}_0$  представимы в виде кронекерова произведения собственных векторов одномерной модели:

$$\mathbf{F}_{\alpha_1\alpha_2\dots\alpha_d} = \mathbf{f}_{\alpha_1} \otimes \mathbf{f}_{\alpha_2} \otimes \dots \otimes \mathbf{f}_{\alpha_d},$$

где  $\alpha_i = 1, 2, \dots, L$ ,  $i = 1, 2, \dots, d$ .

Сводя задачу на собственные значения к вычислению  $\mu_{\alpha_1\alpha_2\dots\alpha_d} = \mathbf{F}_{\alpha_1\alpha_2\dots\alpha_d}^+ \mathbf{U}_0 \mathbf{F}_{\alpha_1\alpha_2\dots\alpha_d}$ , и учитывая соотношения (9), получаем:

$$\mu_{\alpha_1\alpha_2\dots\alpha_d} = \sum_{k_1=0}^l \sum_{k_2=0}^l \dots \sum_{k_d=0}^l w(k_1, k_2, \dots, k_d) \prod_{i=1}^d \lambda_{\alpha_i}(k_i),$$

где  $\alpha_i = 1, 2, \dots, L$ , а  $i = 1, 2, \dots, d$ .

## 6. Обсуждение и выводы

Проведенные рассмотрения показывают, что с ростом размерности решетки  $d$  степень вырожденности спектра собственных значений возрастает: увеличивается как число вырожденных собственных значений, так и степень их вырождения. Рассмотрим, как на вырождение собственных значений влияет дальноедействие, когда учитывается взаимодействие не только с ближайшими соседями (nearest neighbors), но и со следующими соседями (next nearest), и со следующими за ними соседями (next next nearest) - и так далее.

В одномерной модели, за исключением невырожденного первого собственного значения  $\mu_1 = \sum_{k=1}^l w(k) \cdot \lambda_1(k)$ , остальные собственные значения дважды вырождены - см. выражение (4). То же самое имеет место и в случае, когда учитывается взаимодействие только с ближайшими соседями - см. выражение (2) для собственных значений  $\{\lambda_1(1)\}$ . Иными словами, включение дальногодействия не меняет вырождения собственных значений.

Для решеток большей размерности ситуация несколько сложнее. Начнем с изотропного взаимодействия, когда величина связи между спинами  $w(k_1, k_2, \dots, k_d)$  не меняется при перестановке номеров  $k_1, k_2, \dots, k_d$  и, соответственно, величина собственного значения  $\mu_{\alpha_1\alpha_2\dots\alpha_d}$  не зависит от порядка следования индексов  $\alpha_1, \alpha_2$

, ...,  $\alpha_d$ , пробегая значения от 0 до  $L$ . Например, в двумерной модели:  $w(n, m) = w(m, n)$  и  $\mu_{\alpha\beta} = \mu_{\beta\alpha}$ . В такой ситуации, для  $d$ -й модели вырождение собственного значения  $\mu_{\alpha_1\alpha_2\dots\alpha_d}$  равно  $2^{d-r} d!$ , где  $r$  - число индексов, равных 1. Множитель  $2^{d-r}$  обусловлен здесь двукратным вырождением собственных значений одномерной модели, а множитель  $d!$  равен числу перестановок индексов  $\alpha_1, \alpha_2, \dots, \alpha_d$ . Та же самая картина имеет место и при учете взаимодействия с одними только ближайшими соседями, и оказывается, она не меняется при включении дальногодействия.

Обратим внимание на то, что степень вырождения растет экспоненциально по размерности решетки  $d$ .

Ситуация несколько меняется в случае анизотропного взаимодействия, когда величины связей по разным осям решетки различны. В этом случае константа взаимодействия  $w(k_1, k_2, \dots, k_d)$  изменяется при перестановке своих аргументов  $k_1, k_2, \dots, k_d$  и, соответственно, величина собственного значения  $\mu_{\alpha_1\alpha_2\dots\alpha_d}$  существенно зависит от порядка следования индексов  $\alpha_1, \alpha_2, \dots, \alpha_d$ . Например, в двумерной модели  $w(n, m) \neq w(m, n)$ , и  $\mu_{\alpha\beta} \neq \mu_{\beta\alpha}$ . В этой ситуации вырождение собственного значения  $\mu_{\alpha_1\alpha_2\dots\alpha_d}$  равно  $2^{d-r}$ , где  $r$  - число индексов, равных 1. Множитель  $d!$  здесь пропадает. Та же самая картина имеет место при учете взаимодействия с одними только ближайшими соседями, и она не изменяется при включении дальногодействия.

Резюмируем: i) вырождение собственных значений резко возрастает с ростом размерности пространства  $d$ ; ii) включение дальногодействия не снимает вырождения, характерного для взаимодействия только с ближайшими соседями; iii) вырождение уменьшается в  $d!$  раз при появлении анизотропии по разным осям многомерной решетки.

Работа была поддержана грантом РФФИ № 18-07-00750.

# Spectrum of Ising connection matrix with arbitrary long-range interaction

B.V. Kryzhanovsky and L.B. Litinskii

**Abstract.** We examine a connection matrix of Ising model on a d-dimensional hypercube lattice with an arbitrary long-range interaction and periodic boundary conditions. We calculate eigenvectors of this

matrix as the Kronecker products of the well-known eigenvectors of the connection matrix of the one-dimensional Ising model. The obtained rigorous results for eigenvalues are trigonometric polynomials of degree  $d$  with coefficients equal to the interaction constants between spins. We found that a degeneracy of the eigenvalues of the spectrum remains when we include long-range interactions..

**Keywords:** Ising Model, hypercubic lattice, eigenvalue, Kronecker product

## Литература

1. Р. Бэкстер. Точно решаемые модели статистической физики. М., Мир, 1985.
2. Г. Стэнли. Введение в фазовые переходы и критические явления. М., Мир, 1989.
3. В.Г. Вакс, А.В. Ларкин Ю.Н. Овчинников. Модель Изинга при взаимодействии с ближайшими соседями. «А.И. Ларкин, Собрание трудов», т.1, М., МЦНМО, 2009, стр. 245-256.
4. Junqi Yin, D. P. Landau. Phase diagram and critical behavior of the square-lattice Ising model with competing nearest- and next-nearest-neighbor interactions. «Physical Review E», V.80 (2009), p. 05117.
5. А. К. Муртазаев, М. К. Рамазанов, Ф. А. Кассан-Оглы, Д. Р. Курбанова. Фазовые переходы в антиферромагнитной модели Изинга на объемно-центрированной кубической решетке с взаимодействиями вторых ближайших соседей. «ЖЭТФ», Т. 147 (2015), № 1, 127-131, DOI: 10.7868/S0044451015010101.
6. С. В. Белим, И. Б. Ларионов. Критическое поведение антиферромагнитной модели Изинга с эффектами дальнего действия. «Вестник Московского Университета», сер. «ФИЗИКА. АСТРОНОМИЯ», Т.4 (2018), С. 58–62.
7. А. К. Murtazaev, М. К. Ramazanov, М. К. Badiev. Critical properties of the two-dimensional Ising model on a square lattice with competing interactions. “Physica B”. V. 476 (2015), P. 1-5.
8. Boris Kryzhanovsky and Leonid Litinskii. Phase Transition in 2D Ising Model with Next-Neighbor Interaction. “Journal of Physics: Conference Series” V.1391 (2019), P. 012077, doi:10.1088/1742-6596/1391/1/012077.
9. F.A. Kassin-Ogly, A.K. Murtazaev, A.K. Zhuravlev et al. Ising model on a square lattice with second-neighbor and third-neighbor interactions. (2014), arXiv:1407.1578, DOI: 10.1016/j.jmmm.2015.02.030.
10. B.V. Kryzhanovsky, L.B. Litinskii, I.M. Kaganova. n-vicinity method for Ising Model with long-range interaction. Submitted in Chinese Journal of Physics, 2020.
11. Б.В. Крыжановский, Л.Б. Литинский. Собственные значения матрицы связи модели Изинга: учет взаимодействия с соседями, следующими за ближайшими. «ДАН», Т. 489 (2019), № 3, С. 26–29.
12. Б.В. Крыжановский, Л.Б. Литинский. Собственные значения гамильтониана Изинга с дальним действием. «Труды НИИСИ», Т.10 (2020), №3 (настоящий сборник).
13. Р. Беллман. Введение в теорию матриц. М., Наука, 1969.

# Методы моделирования параметров и фокусировки камеры в виртуальном окружении

А.В. Мальцев<sup>1</sup>, Л.А. Финагин<sup>2</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, avmaltcev@mail.ru;

<sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, antifin@mail.ru

**Аннотация.** В работе рассматриваются решения для вычисления углов обзора виртуальной камеры, имитирующей реальное устройство наблюдения с заданным набором параметров его оптики и электроники. Также предлагается метод выполнения автоматической фокусировки изображения, получаемого при распределенном рендеринге трехмерной виртуальной среды на современном многоядерном графическом процессоре.

**Ключевые слова:** виртуальное окружение, рендеринг, реальное время, средство наблюдения, фокусное расстояние, угол обзора

## 1. Введение

Одной из задач, стоящих при создании систем виртуального окружения, является моделирование в них настоящих устройств наблюдения, таких как фото- и видеокамеры. Эквивалентом таких устройств в синтезируемом пространстве выступают виртуальные камеры. Видимое ими изображение формируется с помощью технологий, методов и алгоритмов рендеринга, которые, как правило, основаны на применении распределенных вычислений на современных графических процессорах (GPU).

Реалистичность результата виртуального моделирования средств наблюдения напрямую зависит от степени соответствия генерируемых изображений трехмерных сцен с теми изображениями, которые получаются с помощью реальных устройств. Такое соответствие во многом определяется наличием и качеством имитации различных параметров оптики и электроники этих устройств, а также возникающих в них световых эффектов и помех. Данной проблеме посвящено не мало научных публикаций. Так, в работе [1] рассматриваются методы и распределенные алгоритмы моделирования искажений изображения и помех, свойственных реальным видеокамерам. Статьи [2, 3] описывают решения для реализации глубины резкости на изображениях трехмерной виртуальной среды. Авторы публикации [4] предлагают оригинальные подходы к имитации хроматических аберраций и виньетирования.

В данной работе предлагаются методы и подходы для моделирования ряда основных физических и оптических параметров устройств наблюдения при выполнении распределенной визуализации трехмерных виртуаль-

ных сцен в масштабе реального времени. Рассматривается вопрос вычисления углов обзора виртуальной камеры, соответствующих характеристикам электроники и оптики имитируемого устройства, а также размерам области вывода. Кроме того, описывается метод выполнения автофокусировки виртуальной камеры по центральной точке видимого ей кадра. Рассмотрим предлагаемые решения подробнее.

## 2. Расчет углов обзора

При моделировании реального устройства наблюдения в трехмерной виртуальной среде важной задачей является расчет горизонтального  $\phi_h$  и вертикального  $\phi_v$  углов обзора виртуальной камеры, соответствующей данному устройству. Именно они определяют пирамиду видимости, а значит и тот объем пространства, который будет виден на синтезируемых изображениях.

Пусть необходимо смоделировать работу фото- или видеокамеры, электронная матрица которой имеет ширину  $W_m$  и высоту  $H_m$  (мм), а установленная оптика обеспечивает фокусное расстояние  $f$ . Рассмотрим вычисление угла  $\phi_h$ , на основе которого затем найдем  $\phi_v$ .

Как показано в [5], зависимость угла обзора  $\alpha$  реального устройства наблюдения с «полнокадровой» матрицей 24 x 36 мм (эквивалентна малоформатному кадру 35-миллиметровой фотопленки) от ее размера  $d$  (равного 36 мм для горизонтального угла или 24 мм для вертикального) и фокусного расстояния  $f$  определяется как

$$\alpha = 2 \cdot \operatorname{arctg} \frac{d}{2f}.$$

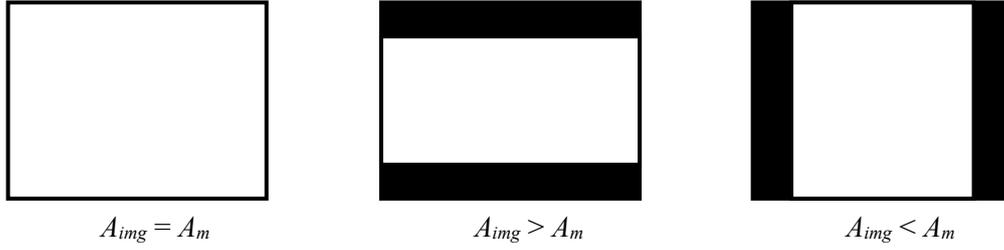


Рис. 1. Эффективные области использования матрицы камеры (выделены белым) при различных соотношениях сторон кадра

Если используемая матрица не совпадает с «полнокадровой», то итоговое фокусное расстояние  $f_{res}$  зависит не только от оптики, но и от размеров матрицы. Данная зависимость выражается коэффициентом  $k_{crop}$ , называемым кроп-фактором, и имеет вид

$$f_{res} = k_{crop} \cdot f,$$

$$k_{crop} = \frac{D_{35}}{\sqrt{W_m^2 + H_m^2}},$$

где  $D_{35} \approx 43.3$  мм – диагональ упомянутого выше малоформатного кадра. Поэтому горизонтальный угол обзора  $\varphi$  для камеры с произвольной матрицей, имеющей ширину  $W_m$ , можно вычислить по формуле

$$\varphi = 2 \cdot \arctg\left(\frac{W_m}{2 \cdot k_{crop} \cdot f}\right). \quad (1)$$

Однако, кроме параметров самой камеры, горизонтальный угол обзора также зависит от размеров  $W_{img}$  и  $H_{img}$  (мм) сторон формируемых

ей кадров, а конкретнее от их соотношения (*aspect ratio*). В реальности изменение этих размеров выполняется в меню устройства посредством установки разрешения кадра по ширине и высоте в пикселах ( $R_w$  и  $R_h$  соответственно). Пользователю, как правило, предоставляется возможность выбора из нескольких вариантов. При визуализации трехмерной сцены с использованием виртуальной камеры значения  $R_w$  и  $R_h$  для формируемого кадра определяются размерами области вывода изображения на экране (*viewport*) в пикселах.

Введем обозначения  $A_{img}$  и  $A_m$  для отношений ширины кадра и электронной матрицы к их высотам. Тогда можно выделить три типа эффективных областей использования матрицы в зависимости от значений  $A_{img}$  и  $A_m$ . Они показаны на рисунке 1. При  $A_{img} \geq A_m$ , кадр распространяется на всю ширину матрицы, поэтому искомый угол обзора  $\varphi_h$  совпадает с вычисленным по формуле (1) углом  $\varphi$ . Если же  $A_{img} < A_m$ , то для задания физически корректной пирамиды видимости виртуальной камеры необходимо рассчитывать  $\varphi_h$  в соответствии с шириной эффективной области использования матрицы, равной  $W_{img}$  (рис. 2). При условии квадратной формы пикселей имеем

$$A_{img} = \frac{W_{img}}{H_{img}} = \frac{R_w}{R_h}, \quad A_m = \frac{W_m}{H_m}, \quad H_{img} = H_m,$$

откуда следуют равенства

$$\frac{A_{img}}{A_m} = \frac{W_{img}}{W_m},$$

$$W_{img} = \frac{R_w}{R_h \cdot A_m} W_m. \quad (2)$$

Вычислив  $W_{img}$  по формуле (2) и принимая во внимание, что

$$\tg \frac{\varphi}{2} / \tg \frac{\varphi_h}{2} = W_m / W_{img},$$

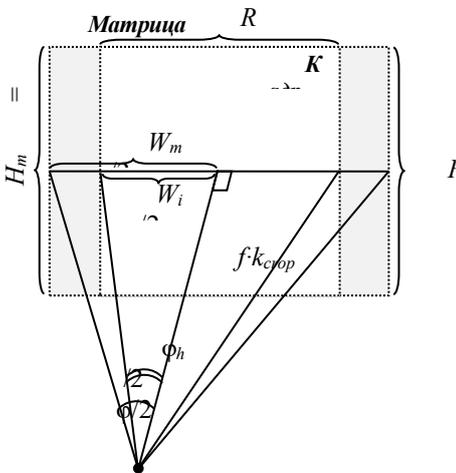


Рис. 2. Расчет угла обзора камеры

получим

$$\varphi_h = 2 \cdot \arctg \left( \frac{R_w}{R_h \cdot A_m} \cdot \tg \frac{\varphi}{2} \right).$$

Суммируя все рассмотренные выше случаи соотношений сторон, горизонтальный угол  $\varphi_h$  обзора виртуальной камеры будем определять по формуле

$$\varphi_h = \begin{cases} 2 \cdot \arctg \left( \frac{R_w}{R_h \cdot A_m} \cdot \tg \frac{\varphi}{2} \right), & A_{img} < A_m \\ \varphi, & A_{img} \geq A_m \end{cases}. \quad (3)$$

Поскольку

$$\tg \frac{\varphi_h}{2} / \tg \frac{\varphi_v}{2} = R_w / R_h$$

то вертикальный угол  $\varphi_v$  ее обзора при этом вычисляется как

$$\varphi_v = 2 \cdot \arctg \left( \frac{R_h}{R_w} \cdot \tg \frac{\varphi_h}{2} \right). \quad (4)$$

На рис. 3а и 3б представлены результаты визуализации трехмерной модели виртуального полигона с применением описанного подхода. В ходе рендеринга была использована одна и та же модель устройства наблюдения с кроп-фактором матрицы равным 1 и постоянным фокусным расстоянием. При этом соотношение размеров сторон области вывода изменялось:  $A_{img} = A_m$  (рис. 3а) и  $A_{img} < A_m$  (рис. 3б). Из рисунков видно, что пересчет горизонтального угла обзора виртуальной камеры по формуле (3) сохраняет оптический зум постоянным.

### 3. Фокусировка камеры

Изменение фокусного расстояния и/или

диафрагмы устройства наблюдения, а также смена композиции кадра в большинстве случаев влечет видимую потерю фокуса изображения. Существует несколько основных способов для его восстановления: ручная, однократная автоматическая или постоянная автоматическая фокусировка. Первая состоит в подборе фокусного расстояния с помощью органов управления устройства наблюдения, например, путем поворотов фокусирующего кольца объектива. Для реализации такого режима в виртуальной среде достаточно ассоциировать изменение параметра  $f$  в большую и меньшую стороны на некоторую величину  $\Delta f$  с элементами используемого устройства управления (клавиатуры, мышки, реального или виртуального пульта управления и т.д.) В данной работе основное внимание уделено моделированию автофокусировки по центральной точке кадра.

Суть предлагаемого подхода заключается в использовании предварительной визуализации трехмерной среды с применением специальной виртуальной камеры и получением значения глубины сцены в направлении центра основного кадра. Данная камера размещается в той же точке, что и текущая камера для основного рендеринга, и имеет такую же видовую систему координат VCS. Ширина и высота формируемого изображения задаются равными одному пикселу. Для этого используется функция *glViewport* из графической библиотеки OpenGL с параметрами (0, 0, 1, 1). Горизонтальный угол обзора добавленной камеры устанавливается равным  $0.01^\circ$ , а вертикальный пересчитывается по формуле (4). Таким образом, данная камера смотрит в центральную точку основного кадра. Предварительный рендеринг сцены с ее применением выполняется в упрощенном виде, не требующем расчета освещенности и наложения никаких текстур, кроме карт прозрачности. По-

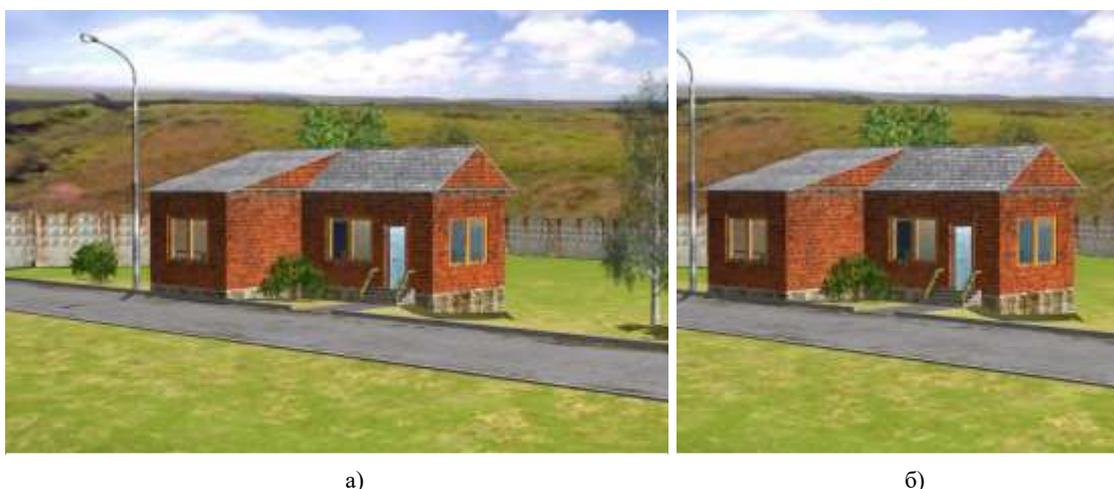


Рис. 3. Зависимость горизонтального угла обзора от соотношения сторон области вывода

следние текстуры необходимо учитывать, поскольку они могут задавать полностью прозрачные (невидимые) части виртуальных объектов, влияя тем самым на общую геометрию сцены. В результате проведенной визуализации графической картой формируется z-буфер размером в 1 пиксел. В этом пикселе записана глубина  $d \in [0, 1]$ , которая соответствует расстоянию  $r$  от основной камеры до точки некоторого объекта виртуальной среды, проецируемой в центр текущего кадра и ближайшей к камере. Для вычисления  $r$  используем формулу из работы [3]:

$$r = \frac{m_{23}}{m_{22} + 2d - 1},$$

где  $m_{22}$ ,  $m_{23}$  – элементы матрицы перспективного преобразования текущей камеры,

$$m_{22} = \frac{f_p + n_p}{n_p - f_p}, \quad m_{23} = \frac{-2 \cdot f_p \cdot n_p}{f_p - n_p},$$

$n_p$  и  $f_p$  – расстояния до ближней и дальней плоскостей отсечения камеры в ее системе координат VCS.

Далее производится распределенный рендеринг изображения сцены, видимого основной камерой, на GPU с расчетом освещенности и применением к объектам необходимых текстур и материалов. Для полученного кадра выполняется моделирование глубины резкости по методу, изложенному в [3]. При этом требуемый фокус изображения по центральной точке кадра обеспечивается использованием значения  $r$  в

качестве расстояния фокусировки. В режиме однократной автофокусировки расчет данного расстояния описанным выше способом и его сохранение в памяти выполняется один раз при поступлении сигнала от заданного элемента устройства управления, например, клавиши на клавиатуре. При постоянной автофокусировке расчет выполняется перед визуализацией каждого кадра.

На рисунке 4 представлены примеры изображений трехмерной виртуальной сцены, сфокусированных на переднем (слева) и заднем (справа) планах по центральной точке кадра с применением описанного метода.

#### 4. Заключение

В данной работе были рассмотрены подходы к моделированию в виртуальной среде реальных устройств наблюдения, обладающих набором характеристик их оптических и электронных компонентов. На базе предложенных решений для расчета углов обзора виртуальной камеры, а также метода ее фокусировки по центральной точке кадра были созданы оригинальные программные модули. Они были протестированы в составе системы визуализации, разработанной в ФГУ ФНЦ НИИСИ РАН. Проведенная апробация показала возможность использования предложенных методов и подходов в системах виртуального окружения.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00393.



Рис. 4. Автофокусировка виртуальной камеры по центральной точке

# Methods for simulation of camera's parameters and focusing in virtual environment

Andrey Maltsev, Leonid Finagin

**Abstract.** The article discusses solutions for computing view angles of a virtual camera, which simulates a real surveillance device with a given set of optical and electronic system parameters. A method for autofocusing images in distributed visualization of three-dimensional virtual environment on multi-core graphics processor is also proposed.

**Keywords:** virtual environment, rendering, real time, surveillance device, focal length, view angle

## Литература

1. А.В. Мальцев. Методы моделирования световых эффектов и искажений видеосигнала в виртуальных средствах наблюдения. «Информационные технологии и вычислительные системы», Т.9 (2019), № 2, 54-61.
2. B.A. Barsky, T.J. Kosloff. Algorithms for rendering depth of field effects in computer graphics. 12th WSEAS International Conference on COMPUTERS, 2008.
3. А.В. Мальцев, М.А. Торгашев. Распределенное моделирование глубины резкости при визуализации трехмерных сцен на GPU. «Труды НИИСИ РАН», Т. 9 (2019), № 5, 100-104.
4. M. Kučič, P. Zemčík. Simulation of Camera Features. 16th Central European Seminar on Computer Graphics, 2012, 117-123.
5. Angle of view, [https://en.wikipedia.org/wiki/Angle\\_of\\_view](https://en.wikipedia.org/wiki/Angle_of_view) (дата обращения: 22.04.2020).

# Создание системы управления с обратной связью при разработке нефтяных месторождений

К.Д. Ашмян<sup>1</sup>, С.Г. Вольпин<sup>2</sup>, О.В. Ковалева<sup>3</sup>, Г.А. Ковалева<sup>4</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, kdashmyan@yandex.ru;

<sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, sergvolpin@gmail.com;

<sup>3</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, olgakovaleva57@mail.ru

<sup>4</sup>ФГБОУ ВО «СамГТУ», Самара, Россия, ga-kovaleva@yandex.ru

**Аннотация.** Рассматривается возможность применения метода, используемого в кибернетике, как науке управления, при создании эффективной системы реализации проекта разработки нефтяных месторождений. Управление реализуемой системой разработки через корректируемые в процессе разработки исходные данные по свойствам нефтенасыщенных кернов и гидродинамических исследований пластов и скважин. А именно, создание системы получения, хранения, преобразования и передачи информации для включения обратной связи. Задача состоит в том, чтобы полученная информация оказывала влияние на реализуемую модель или проект разработки. Т.е. прямое изменение (корректировка) модели для эффективной организации выполнения проектных решений.

**Ключевые слова:** разработка нефтяных месторождений, обратная связь, база исходных данных, пластовые флюиды, керн, гидродинамические исследования

## 1. Введение

При проектировании разработки нефтяных месторождений одним из важнейших аспектов является использование комплекса данных лабораторных исследований пластовых флюидов, и керна, гидродинамических (ГДИ) и промышленно-геофизических (ПГИ) исследований пластов и скважин [1]. Полученные по всем этим направлениям результаты исследований являются взаимосвязанными и позволяют моделировать процессы разработки, составлять проекты разработки и осуществлять контроль и управление процессами разработки [2].

Таким образом, создается основа для управления и целенаправленных действий по технологии процесса добычи нефти, а именно, проекта разработки нефтяного месторождения. В настоящее время корректировки в проекты разработки вносятся не регулярно, а в зависимости от изменения представления о геологическом строении и геологических запасах разрабатываемого объекта и уточнения свойств породы и пластовых флюидов или при изменении фактических показателей разработки от проектных свыше допустимых отклонений. Такой подход к управлению добычи нефти является спонтанным и призван задним числом скорректировать недостатки проектирования и системы добычи нефти с целью исправления экономических показателей.

## 2. Создание системы управления с обратной связью

Изменившиеся в процессе разработки исходные данные являются следствием, так называемого влияния разработки, а именно, изменения, возникшие в процессе добычи, могут значительно отличаться от первоначальных (принятых в проектном документе).

Возможность применения методов, используемых в кибернетике как науке управления, уже на данном уровне получения информации при исследовании пластовых флюидов, нефтенасыщенных кернов, гидродинамических и промышленно-геофизических исследований пластов и скважин, позволит корректировать проектный документ в течение всего времени эксплуатации скважин.

Для этого необходимо создать систему получения, хранения, преобразования и передачи информации для включения обратной связи. Задача состоит в том, чтобы полученная информация оказывала влияние на реализуемую схему или проект разработки. Т.е. прямое изменение (корректировка) модели для эффективной организации выполнения проектных решений.

В кибернетике разработаны общие принципы создания управления и автоматизации умственного труда в различных природных, технических и общественных системах. Однако,

создание научных основ и целенаправленных действий с использованием кибернетики, а именно, путем переработки и учета изменившейся информации относительно исходной, в нефтяной отрасли не проводилось. Основным препятствием на этом пути являлось отсутствие базы данных и технических средств для решения этой задачи (большой объем информации).

Создание современной базы данных, а также применения одного из инструментов кибернетики – метода математической статистики, позволит начать новый этап в развитии проектирования разработки с элементами обратной связи.

В настоящее время практически во всех крупных нефтяных компаниях в той или иной мере ведется сбор первичной информации, которая хранится в различном виде: базы данных, электронный архив, бумажные носители и т.д. Как правило, регулярный анализ полученной информации не проводится, а учитывается только при необходимости составления нового подсчета запасов или проектного документа. Кроме того, чтобы не менять ранее принятые параметры пласта, данные, показывающие изменение свойств флюидов или коллектора в процессе разработки, не учитываются.

Для мониторинга процесса изменения параметров продуктивного пласта, необходим регулярный анализ промысловых данных с построением зависимостей и учетом условий отбора проб флюидов, керн, гидродинамических и промыслово-геофизических исследований. Такое регулярное наблюдение за изменением свойств пласта и флюидов может дать материал для своевременной корректировки системы разработки пласта до того момента, когда произойдет значительное ухудшение процесса выработки запасов.

Важным направлением анализа необходимо считать, как качество проведенных исследований, связанных с условиями отбора и процессом выполнения исследований, так и количеством и периодичностью исследований. В условиях даже крупных нефтяных компаний исследования, в последнее время, проводятся крайне мало по целому ряду причин, связанных с тем, что ведется в основном эксплуатационное бурение, обводненность продукции высокая, вовлекаются в разработку пласты с трудноизвлекаемыми запасами, которые для вызова притока часто приходится подвергать химическим методам интенсификации, после которых адекватные пробы нефти или образцов керн получить невозможно. Результаты проведенных ранее исследований или не сохранились, или они проводились в условиях, которые не соответствуют современным нормативным документам.

В небольших компаниях ситуация еще более

сложная, поскольку исследования проводятся мало, или не проводятся совсем. Для небольших месторождений почти все параметры берутся по аналогии, а выбор аналогов зависит не от объективных причин, связанных с идентичными условиями осадконакопления, а по принципу «что есть». Работы по контролю динамики изменения параметров в процессе разработки вообще не планируются. Хотя на малых месторождениях влияние изменения пластовых условий может быть значительнее, чем на крупных, и приводят к более серьезным последствиям для выработки запасов.

Проблема оценки влияния изменения характеристик пласта и флюидов весьма сложная и многозадачная. В процессе разработки меняются термобарические условия фильтрации флюидов, меняется смачиваемость породы, может изменяться проницаемость поровой среды вследствие выпадения асфальто-смолопарафиновых веществ [3], переноса слабосцементированных разностей породы. Эти эффекты зависят от принятой системы разработки, скоростей фильтрации, технологии закачки вытесняющих агентов.

При анализе разработки решается, как правило, обратная задача: по промысловым данным проводится адаптация параметров коллектора и флюидов. Такой подход, безусловно, оправдан тем, что постоянные замеры проводятся по дебитам скважин, обводненности продукции и, в лучшем случае, забойным и пластовым давлениям. Т.е. решается макрозадача формирования модели потоков в пласте. Но, как и почему меняются потоки в процессе разработки, эти данные не устанавливаются.

Начиная с 2000 годов разными авторами делается попытка решения прямой задачи оценки на микроуровне изменения состава нефти и свойств коллектора. Задачи разделяются по узким направлениям: изменение свойств нефти, изменение смачиваемости, проницаемости и т.д. Найти универсальное решение проблемы сложно в связи с многообразием видов микро неоднородностей, углеводородов и пластовых вод.

Одним из возможных направлений решения этой задачи может стать создание единой базы свойств пластовых флюидов и коллекторов, основанной на результатах отбора и исследования проб флюидов и керн, гидродинамических и промыслово-геофизических исследований, с проведением мониторинга изменения параметров пластов в процессе разработки, приуроченных к единой системе геологического формирования и имеющих сходные начальные свойства.

Создание такой базы возможно только на общей основе, созданной по данным исследо-

ваний разных недропользователей, результаты мониторинга при этом должны быть доступны участникам создания единой базы данных. Эти результаты могут использоваться для оценки процесса разработки целевых пластов даже при наличии малого количества исследований или без них, используя метод аналогии.

На рисунке 1 представлена схема передачи и получения информации по методу обратной связи от недропользователя к исполнителям.



Рис. 1. Схема передачи и получения информации по методу обратной связи от недропользователя к исполнителям

### 3. Принцип создания информационного банка данных для реализации обратной связи в проекте разработки

Одним из основных направлений создания единой базы свойств пластовых флюидов, коллекторов, гидродинамических и промыслово-геофизических исследований является создание информационного банка данных по физико-химическим и PVT свойствам пластовых нефтей [4].

Исходя из современных представлений о физико-химических процессах, происходящих при разработке нефтяных месторождений, можно сделать вывод, что параметры пластовых нефтей изменяются под влиянием следующих факторов: уменьшения газосодержания пластовых нефтей в зоне контакта с закачиваемой водой, захвата адсорбционной поверхностной пленки и различных отложений при увеличении скорости движения пластовой нефти; увеличения охвата нефтяного пласта заводнением при увеличении депрессии на пласт; зачакки в пласт различных агентов, изменения термобарических условий залегания продуктивного объекта (изменения температуры и давления в процессе разработки) и т.д.

Основной принцип работы информационного банка данных - чем больше объем информации, внесенный в базу данных, тем качественней можно выполнить анализ и оценку «истории» и прогноз изменения параметров, а так же восстановить недостаточную или утраченную информацию.

Все полученные по конкретному месторождению исследования физико-химических свойств пластовых флюидов, как правило, со-

храняются в виде научных и технических отчетов, подсчетных и проектных документов. Пользование такой накопленной информацией весьма трудоемко и для составления каждого последующего документа требует привлечения высококвалифицированных специалистов, а время поиска и анализа всей необходимой информации, как правило, ограничено.

Кроме того, формат анализа и выдачи данных, представленных в различных отчетных материалах, выполнен по различным методикам и регламентам, что так же вносит путаницу в правомерности использования имеющихся данных при составлении новых проектных документов.

В настоящее время исследования пластовых флюидов (нефть, газ, вода) выполняются в соответствии со стандартом отрасли ОСТ 153-39.2-048-2003 «Нефть. Типовое исследование пластовых флюидов и сепарированных нефтей» (Объем исследований и формы представления результатов). Поэтому таблицы и графики, используемые для хранения информации, приняты в соответствии с вышеуказанным нормативным документом.

Наиболее рациональным видом анализа, обработки, хранения и выдачи данных по стандартным формам является автоматизированный БАНК. Все имеющиеся данные, перед тем как их заносят в «Базу данных», проходят аналитическую проверку на «качество», т.е. оцениваются условия отбора проб, история разработки, геология объекта, динамика изменения свойств. Параметры PVT пластовых нефтей проверяются расчетом и по корреляционным зависимостям. Отбракованные в результате такой проверки данные из базы не исключаются, т.к. могут служить источником информации по отдельным видам исследований и характеризовать историю исследований проводимых ранее.

Таким образом, в базу данных заносится вся имеющаяся информация, а в банке данных используются только качественные данные, которые позволяют проводить достоверные технологические расчеты и дают возможность оценить изменение свойств по глубине залегания, по площади и во времени в результате воздействия процесса разработки месторождения.

«Банк данных физико-химических свойств пластовых нефтей» включает в себя несколько программ формирования массива исходных данных о параметрах пластовых нефтей, физико-химических свойствах поверхностных нефтей, компонентном составе пластовой и дегазированной нефти и газа. Каждая программа обеспечивает ввод и запись информации в ЭВМ, запись информации в память, сортировку информации по «ключевым» признакам, кон-

троль на наличие ошибок, пополнение массива информации и корректировку отдельных записей.

Выдача информации предусматривается в форме стандартных таблиц, практически соответствующих (с некоторой корректировкой) результирующим таблицам нормативного документа [5]. Предусматриваются также программы, позволяющие получать зависимости по различным свойствам нефтей, характеризующих нефти по объекту, по пласту, по глубине залегания.

Поисковая система должна обеспечивать формирование задания по запросу, поиск информации и выдачу информации. Выдача информации должна осуществляться отдельно по скважинам, горизонтам, месторождениям. В результате можно получать результирующие и статистические отчеты, как по отдельным пробам и параметрам, так и по пластам и месторождениям.

Примерная схема обработки и передачи данных в информационном банке, включенном в обратную связь, действующего проекта разработки нефтяного месторождения, представлена на Рисунке 2.



Рис. 2. Схема обработки и передачи данных в информационном банке, включенном в обратную связь, действующего проекта разработки нефтяного месторождения

Создание банка данных предусматривает регулярное внесение объема информации, необходимого для вывода корреляционных зависимостей по свойствам нефтей по данному региону, по восполнению частично недостающей информации, оперативной оценки степени изу-

ченности того или иного объекта, а так же для введения в разработку Метода Обратной связи.

### 3. Заключение

Начальный состав и распределение компонентов нефти и газа по разрезу и площади продуктивной толщи определяют, прежде всего, запасы компонентов пластовой нефти и газов, возможные объемы их добычи и переработки, систему разработки залежи, технологией промышленной подготовки и переработки пластовых флюидов.

Знание динамики составов флюидов необходимо для корректировки текущих и долгосрочных дебитов исходного сырья для их поставки потребителю и контроля за процессом разработки залежи, включая компонентоотдачу.

Одна из задач при проведении физико-химических и PVT исследований пластовых флюидов является выбор режима работы системы пласт – скважина и скважина – сепаратор, т.е. дебита, депрессии и условий сепарации, при которых обеспечивается наиболее полная и достоверная исходная (начальная) характеристика пластовой нефти. Однако влияние разработки на количественное и качественное изменение состава добываемой нефти требует введение корректирующих параметров в проекты разработки нефтяных месторождений в процессе их реализации.

В настоящее время по данному вопросу нет единой методологической базы и имеющиеся данные часто несопоставимы.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН - Проведение фундаментальных научных исследований (47 ГП) по теме № 0065-2019-0019 «Создание методики выявления невыработанных зон на нефтяных месторождениях и подсчета остаточных запасов нефти на основе комплексирования математического моделирования, анализа разработки с исследованиями скважин и пластов» (рег. № АААА-А19-119020190071-7).

## Creating a feedback management system for oil field development

K.D. Ashmyan, S.G. Volpin, O.V. Kovaleva, G. A. Kovaleva

**Abstract.** The paper considers the possibility of applying the method used in Cybernetics as a management science in creating an effective system for implementing an oil field development project. Management of the implemented development system through initial data on the properties of oil-saturated cores and hydrodynamic studies of reservoirs and wells that are corrected during development. Namely, creating a system for receiving, storing, convert-

ing and transmitting information to enable feedback. The goal is to ensure that the received information has an impact on the implemented model or development project, i.e., direct modification (correction) of the model for effective organization of project decisions.

**Keywords:** oil field development, feedback, source data base, reservoir fluids, core, hydrodynamic research

## Литература

1. Методическое руководство по комплексированию и этапности выполнения геофизических, гидродинамических и геохимических исследований нефтяных и нефтегазовых месторождений. РД 153-39.0-109-01, М, 2002, С. 75.

2. К.Д. Ашмян, А.Б. Баишев., С.Г. Вольпин. Анализ технических требований к исходным геолого-физическим и гидродинамическим данным для моделирования углеводородных систем. Вестник ЦКР РОСНЕДРА, № 1, 2006, 53-59.

3. Методика оценки фазового состояния парафинов в пластовых нефтях. Вестник ЦКР РОСНЕДРА, № 6, 2011, 11–14.

4. К.Д. Ашмян, О.В. Ковалева, Д.О. Казаков. Комплексная система анализа и подготовки данных по свойствам пластовых флюидов для проектных документов с использованием Регионального оперативного банка данных. Вестник ЦКР РОСНЕДРА, № 4, 2013, 46-51.

5. Об утверждении временных методических рекомендаций по подготовке технических проектов разработки месторождений углеводородного сырья. Распоряжение Минприроды России, №12р от 18.05.2016.

# Параметрическая компенсация тепловых сдвигов в сенсорах на основе поверхностного плазмонного резонанса

Ф.А. Юдкин<sup>1</sup>, А.В. Голубкин<sup>2</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, [yudkin.fedor@yandex.ru](mailto:yudkin.fedor@yandex.ru)

<sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, [alexandr\\_go@bk.ru](mailto:alexandr_go@bk.ru)

**Аннотация.** Полупроводниковые лазеры широко применяются в сенсорных приборах, работающих на эффекте поверхностного плазмонного резонанса. Плазмонные сенсоры обладают высокой чувствительностью к смещению длины волны излучения. Особенно это критично для газовых сенсоров при длительных интервалах измерения. В данной статье рассматривается параметрическая компенсация ухода минимума резонансной кривой сенсора вследствие теплового сдвига частоты лазера. Для компенсации предлагается варьировать градиент показателя преломления поверхностного диэлектрического слоя. На основе структуры Glass – Ag – TiO<sub>2</sub> дается оценка влияния смещения длины волны излучения на сдвиг положения минимума ППР. Представлены соотношения, с помощью которых рассчитывается компенсационный градиент при заданных характеристиках сенсора, в том числе толщины диэлектрического слоя. Приведено усредненное значение градиента коэффициента показателя преломления диэлектрического слоя для типичного полупроводникового лазера. Данная методика может быть применена при использовании различных типов полупроводниковых лазеров.

**Ключевые слова:** ППР, температура, смещение лазера, плазмонный резонанс, показатель преломления, длина волны

## 1. Введение

Сенсоры на основе поверхностного плазмонного резонанса (ППР) обладают высокой чувствительностью более  $10^{-6}$  RIU и представляют интерес для различных биологических и экологических применений [1,2,3]. Увеличение чувствительности сенсоров за данные пределы является актуальной проблемой данной категории сенсоров. Физическая модель сенсора показывает, что положение минимума поверхностного плазмонного резонанса существенно зависит от длины волны падающего излучения [4]. Для газовых анализаторов, в связи с их повышенной чувствительностью это влияние наиболее существенно, ограничивая чувствительность сенсора на длительных (десятки минут) интервалах измерений, характерных для биологических и некоторых химических процессов.

Также интерес представляет использование эффекта в проблеме оптической трассировки процессорных шин, где, в связи с плотностью компоновки, шины занимают наибольшую площадь кристалла.

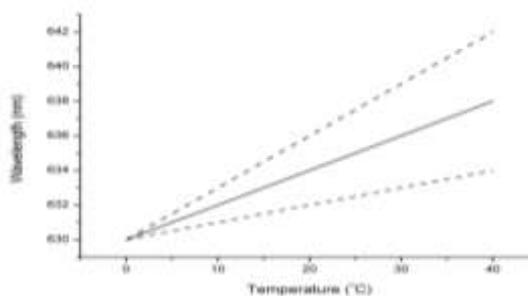
Таким образом, необходимость компенсировать тепловой уход лазера по частоте является важной задачей сенсорики.

В настоящей работе рассматривается вариант компенсации ухода частоты лазера от увеличения температуры рабочего тела путем из-

менения параметров поверхностного слоя сенсора.

## 2. Постановка задачи

Факт того, что сегодня все еще проводятся исследования по минимизации сдвигов длины волны лазеров от температуры, говорит о том, что данная проблема актуальна [5].

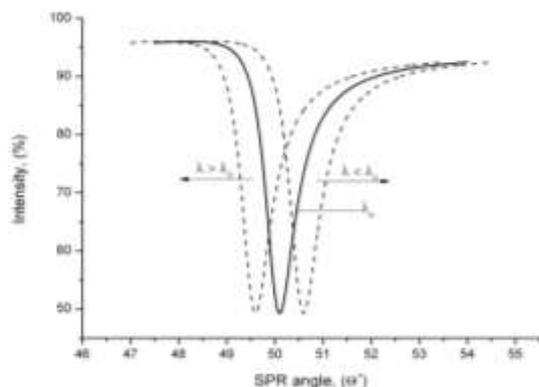


**Рис.1.** Усредненная характеристика температурного ухода полупроводникового лазера по частоте.

Существующие в настоящее время полупроводниковые лазеры обладают значительным температурным смещением частоты излучения. Типичный график такого смещения [6,7,8] приведен на рис.1.

Эти изменения приводят к сдвигу резонанса в плазмонных сенсорах. На рис.2 показана за-

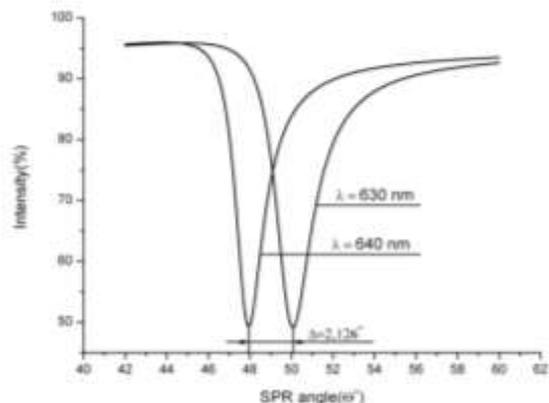
висимость смещения минимума ППР от длины волны падающего излучения.



**Рис.2.** Зависимость сдвига минимума поверхностного плазмонного резонанса от частоты падающего излучения.

Из графика видно, что уменьшение длины волны падающего излучения приводит к увеличению угла плазмонного резонанса.

Оценим количественное влияние сдвига длины волны лазера на положения минимума ППР.

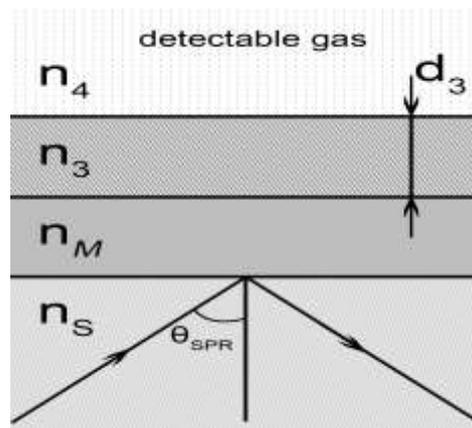


**Рис.3.** Смещение минимума ППР в системе Glass – Ag – TiO<sub>2</sub>.

Для этого проведем простое моделирование поведения ППР с использованием программы TFCalc (Software Spectra) для активного материала сенсора TiO<sub>2</sub>. Как можем видеть на рис. 3, угол ППР увеличивается на 2,126 градуса. При чувствительности сенсора 10<sup>-6</sup> RIU теряется около четырех порядков в значении точности.

### 3. Методика компенсации

Для расчета используется структура сенсора, приведенная на рис.4.



**Рис.4.** Базовая конфигурация для наблюдения поверхностного плазмонного резонанса.

Положение пика поглощения плазмонного резонанса определяется как:

$$\sin \theta_{SPR} \approx \frac{1}{n_S} \left( \frac{n_{M,i}^2 \cdot n_A^2}{n_{M,i}^2 - n_A^2} \right)^{1/2}, \quad (1)$$

где  $n_S$  – показатель преломления диэлектрика  $S$ , из которого падает излучение;  $n_A$  – в данном соотношении полубесконечный эффективный слой, суммирующий показатели преломления  $n_3$  и  $n_4$ ;  $n_{M,i}$  – мнимая часть комплексного показателя преломления металлического слоя [9].

Из соотношения (1) легко сделать вывод о параметрах материала, необходимых для компенсации частотного сдвига лазера.

$$n_A^2 = \frac{n_{M,i}^2 (n_S \cdot \sin \theta_{SPR})^2}{n_{M,i}^2 + (n_S \cdot \sin \theta_{SPR})^2} \quad (2)$$

Имея все данные из правой части выражения можем вычислить необходимые для компенсации оптические характеристики диэлектрика формуле (2). Диэлектрик  $n_A$  это полубесконечный эффективный слой, который заменяет детектируемый диэлектрик, и окружающую среду. Однако в соотношении (2) необходимо разделить  $n_A$  на составляющие её  $n_3$  и  $n_4$ . Так как показатель преломления среды из которой падает свет и металлической пленки помечены узлы решетки, расстояния до которых от начальной точки в левом нижнем узле равно этим числам. считаем заранее известными, а угол ППР, исходя из поставленной задачи, постоянным, то требуется получить показатель преломления тонкослойного диэлектрического покрытия  $n_3$  [10]. С учетом толщины  $d_3$ , выра-

жение для нахождения показателя преломления  $n_A$  будет выглядеть как (3):

$$n_A = \frac{2\pi d_3}{\lambda} \cdot \frac{(-n_{M,i} \cdot n_4^2)^{3/2}}{(n_{M,i} - n_4^2)^2} \cdot \frac{n_3^2 - n_4^2}{n_3^2} + n_4 \quad (3)$$

Тогда для нахождения показателя преломления тонкослойного диэлектрика объединим (2) и (3):

$$\frac{n_3^2 - n_4^2}{n_3^2} = \frac{(n_A - n_4)\lambda(n_{M,i} - n_4^2)^2}{2\pi d_3(-n_{M,i} \cdot n_4^2)^{3/2}}$$

$$n_3^2 = n_4^2 \left( 1 - \frac{(n_A - n_4)\lambda(n_{M,i} - n_4^2)^2}{2\pi d_3(-n_{M,i} \cdot n_4^2)^{3/2}} \right)^{-1} \quad (4)$$

Таким образом, воспользовавшись соотношением (4) и задав толщину диэлектрика, получим уточнённое значение градиента  $n/\lambda$  для каждого конкретного лазера.

Посчитаем крайние точки  $n_A$  для случая, когда среда, из которой падает излучение – стекло, а металл – серебро. Проведя вычисления получим изменение показателя преломления необходимое для компенсации теплового смещения частоты лазера. Для нашего случая градиент составляет  $n/\lambda = 0.0001784113$  (Рис. 5).

## 4. Обсуждение результатов

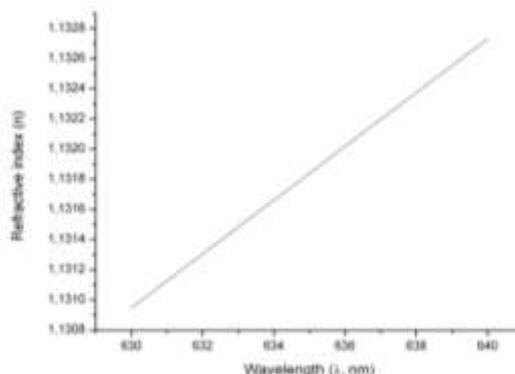
Полученный градиент указывает направление, в котором должен изменяться показатель преломления диэлектрика. Однако, чтобы более точно определить материал диэлектрика, который компенсировал бы уход длины волны лазера, необходимо для каждой модификации сенсора просчитать значение  $n_3$  в рабочих точках диапазона изменения длины волны лазера.

Также смещение пика вызывает температурное изменение толщины слоя диэлектрика, однако эти изменения достаточно малы для того чтобы исключить их из рассмотрения. Например, коэффициент теплового расширения диоксида титана  $7,4 \cdot 10^{-6}$  град<sup>-1</sup>. Синус угла SPR в таком случае увеличится на  $5,476 \cdot 10^{-11}$ .

Изменение длины волны источника излучения касается не только сенсорной области, но и любой другой, где может использоваться плазмонный эффект. Например, в оптических коммутаторах, оптоволоконных кабелях или оптических процессорах.

Использование плазмонного эффекта в коммутационных устройствах позволяет устранить недостатки, среди которых ограничение узлами

коммутации пропускной способности оптической сети. Это объясняется тем, что информационный сигнал преобразуется из оптического в электрический и обратно, поэтому данное направление является перспективным в области информатики [11,12].



**Рис.5.** Расчетный градиент показателя преломления диэлектрика при изменении частоты излучения.

Применение плазмонного резонанса в коммутации более эффективно, так как в условиях резонанса происходит полное поглощение энергии света. Однако резонанс очень чувствителен к изменению внешних условий поэтому смещение частоты лазера при микронных размерах переключающего элемента влечет за собой существенное ухудшение работы коммутатора [13].

Компенсация ухода длины волны лазера также поможет в получении более точных результатов при исследованиях оптических коммутаторов на основе мемристорных наноструктур с учетом временных и температурных воздействий [14].

## 6. Выводы

В данной работе было рассмотрено негативное влияние изменения температуры лазера на смещение пика плазмонного резонанса и последствия этого смещения для длительных измерений. В результате исследований удалось компенсировать сдвиг резонансного провала вследствие температурного дрейфа частоты лазера путем изменения параметров аморфного диэлектрического слоя. Рассмотренная методика позволяет увеличить чувствительность плазмонных сенсоров на длительных интервалах измерений, что характерно для биологических анализов и других применений, требующих стабильных опорных уровней. Параметрическая компенсация позволяет простыми технологическими методами значительно улучшить стабильность приборов на основе плазмонного

резонанса. Так, для чувствительности сенсора  $10^{-6}$  RIU необходимо стабилизировать температуру прибора с точностью  $2,35 \cdot 10^{-7}$  градусов Цельсия, что недостижимо методами внешней стабилизации температуры.

Работа выполнена при поддержке Российского Фонда Фундаментальных исследований.  
Проект №: 19-29-03030.

## Surface plasmon resonance sensors parametric compensation of thermal shift

F.A. Yudkin and A.V. Golubkin

**Abstract.** Semiconductor lasers are widely used in the surface plasmon resonance sensor devices. SPR sensors are highly sensitive to offset wavelength radiation. This is especially critical for gas sensors with long measurement intervals. This article discusses the parametric compensation of the sensor resonance curve minimum departure due to the thermal shift of the laser frequency. To compensate, it is proposed to vary the gradient of the surface dielectric layer refractive index. Based on Glass - Ag - TiO<sub>2</sub>, an assessment of the radiation wavelength shift effect on the minimum SPR position shift is given. The ratio by which the compensation gradient is calculated for the given sensor characteristics, including the dielectric layer thickness are presented. The averaged gradient of the dielectric layer refractive index coefficient for a typical semiconductor laser is presented. This technique can be applied using various types of semiconductor lasers.

**Keywords:** SPR, temperature, laser shift, plasmon resonance, refractive index, wavelength.

### Литература

1. Brilliant Adhi Prabowo, Agnes Purwidyantri, Kou-Chen Liu Surface Plasmon Resonance Optical Sensor: A Review on Light Source Technology Biosensors (Basel). 2018 Sep; 8(3): 80.
2. Chii-WannLin Kuo-PingChen Chien-NanHsiao ShimingLin Chih-KungLee Design and fabrication of an alternating dielectric multi-layer device for surface plasmon resonance sensor / Sensors and Actuators B 113 (2006) 169–176
3. Liu, Y. *et al.* Surface Plasmon Resonance Biosensor Based on Smart Phone Platforms. *Sci. Rep.* **5**, 12864; doi: 10.1038/srep12864 (2015).
4. Nylander, C., Liedberg, B., & Lind, T. (1982). Gas detection by means of surface plasmon resonance. *Sensors and Actuators*, **3**, 79–88. doi:10.1016/0250-6874(82)80008-5
5. J. Carter, D. Snyder, J. Reichenbaugh Temperature dependence of optical wavelength shift as a validation technique for pulsed laser diode array thermal modeling /Nineteenth Annual IEEE Semiconductor Thermal Measurement and Management Symposium, 2003.
6. Sivers, N. L., Van de Workeen, B. C., Lee, S. M., Turner, J. N., & Szarowski, D. H. (2004). Improving fluorescence confocal microscopy with cryogenically-cooled diode lasers. *Optics Express*, **12**(17), 4157. doi:10.1364/opex.12.004157 (delta 0.12 600–650nm)
7. X. Yang , J.B. Heroux , M.J. Jurkovic , W.I. Wang Low-threshold 1.3- $\mu$ m InGaAsN:Sb-GaAs single-quantum-well lasers grown by molecular beam epitaxy (delta 0.4)
8. Daniel Hofstettera, Hans P. Zappe Anomalous longitudinal mode hops in GaAs/AlGaAs distributed Bragg reflector lasers *Appl. Phys. Lett.*, Vol. 71, No. 2, 14 July 1997 (delta 0.1)
9. Weiyong Zong, Mads H. Sørensen, Carsten Thirstrup Integrated holographic grating chip for surface plasmon resonance sensing DK-2630 Taastrup, Denmark *Opt. Eng.* **43**(11) 2505–2510 (November 2004)
10. C. Thirstrup, W. Zong, M. Borre, H. Neff, H.C. Pedersen, G. Holzhueter Diffractive optical coupling element for surface plasmon resonance sensors C. Thirstrup et al. / Sensors and Actuators B **100** (2004) 298–308
11. Гайворонская Г.С., Рябцов А. В. Особенности применения оптических коммутаторов в современных информационных сетях // *Applicable Information Models*. №22. 2011.

- 
12. Barabanov I. O., Maltseva N. S., Barabanova E. A. Switching cell for information transmission optical systems //2016 International Conference on Actual Problems of Electron Devices Engineering (APEDE). – IEEE, 2016. – Т. 2. – С. 1-4.
  13. Гензе Ю. В. и др. Электрические свойства тонких пленок на основе  $\text{TiO}_2$  с нанокластерами золота и меди //Известия Российской академии наук. Серия физическая. – 2015. – Т. 79. – №. 2. – С. 217-217.
  14. A.N. Palagushkin, D.V. Roshchupkin, F.A. Yudkin, D.V. Irzhak, Olga Keplinger, V.V. Privezentsev, «Aspects of the  $\text{TiO}_2$  memristor active medium technology», Journal of Applied Physics, Vol.124, Issue 20, 2018, <https://doi.org/10.1063/1.5038919>.  
ksZwfcH7TLfnuPPfkDhoeLk/.

# Составление многопроцессорного расписания в системе с двумя типами ресурсов

М.Г. Фуругян

ВЦ ФИЦ ИУ РАН, Москва, Россия, rtscas@yandex.ru;

**Аннотация.** Рассматривается задача построения расписания с прерываниями при наличии двух типов ресурсов – возобновляемых (идентичные процессоры) и не возобновляемых, используемых для выполнения комплекса работ. Известны директивные интервалы и объемы выполняемых работ. Количество не возобновляемого ресурса является непостоянным. Для нахождения допустимого распределения ресурсов и построения расписания выполнения работ разработан алгоритм, основанный на нахождении максимального потока в сети.

**Ключевые слова:** возобновляемые и не возобновляемые ресурсы, допустимое расписание с прерываниями, потоковая сеть

## 1. Введение

Задачи составления многопроцессорных расписаний и планирования работ возникают во многих научных и производственных областях. Алгоритмы их решения составляют одну из основных частей математического и программного обеспечения для бортовых вычислительных систем, систем планирования работ в строительстве, при разработке и функционировании сложных технических объектов, в частности систем ПРО и ПВО, в других областях деятельности человека. Примером успешного применения систем планирования работ с использованием не возобновляемых ресурсов могут служить системы МКП и ПЕРТ, разработанные и использованные в конце 50-х годов прошлого века для управления различного рода гражданскими и военными проектами в США.

Имеется большое число публикаций по распределению возобновляемых ресурсов (машин, приборов, процессоров), или составлению расписаний. Отметим работы [1 – 4], в которых решаются задачи составления многопроцессорных расписаний при различных условиях (допускаются или не допускаются прерывания, одинаковые или различные директивные интервалы, идентичные или различные процессоры). В этих работах рассматривается один тип ресурса – возобновляемый, в роли которого выступают процессоры.

В [5, 6] при выполнении работ наряду с процессорами используются также не возобновляемые ресурсы. В отличие от возобновляемых ресурсов, они не могут использоваться многократно. В [5, 6] предполагается, что при выделении такого ресурса работе ее длительность сокращается линейным образом.

В настоящей статье рассматривается задача составления расписания выполнения комплекса работ при наличии двух типов ресурсов – возобновляемых и не возобновляемых. В отличие от [5, 6], предполагается, что не возобновляемый ресурс выделяется работам многократно, в заданные моменты времени. В статье предложен полиномиальный алгоритм нахождения допустимого распределения ресурсов и построения допустимого расписания. Алгоритм основан на сведении исходной задачи к потоковой.

## 2. Формулировка задачи

Имеется множество работ (заданий)  $W = \{w_1, w_2, \dots, w_n\}$ , подлежащих выполнению. Каждая работа  $w_i \in W$ ,  $i = \overline{1, n}$ , характеризуется директивным интервалом  $(b_i, f_i]$  (работа  $w_i$  может выполняться только в этом интервале;  $b_i$  – момент готовности,  $f_i$  – директивный срок) и объемом  $Q_i$ . Для выполнения работ имеется два типа ресурсов – возобновляемых и не возобновляемых. Возобновляемые ресурсы (процессоры) могут использоваться всеми работами многократно. Их количество равно  $m$ . Все процессоры являются идентичными. Объем работы, выполняемый процессором, равен продолжительности его использования. При выполнении заданий допускаются прерывания и переключения с одних процессоров на другие. Предполагается, что прерывания и переключения выполняются мгновенно и не требуют каких-либо затрат. Не допускается параллельное выполнение несколькими процессорами одной работы и одновременное выполнение нескольких работ одним процессором. Не возобновляемый ресурс (электроэнергия, денежные ресур-

сы), выделенный некоторой работе, в дальнейшем использоваться не может.

Величины не возобновляемого ресурса в различные моменты времени могут отличаться, так же как и ограничения на его потребление заданиями  $A$  именно, пусть  $(0; T]$  — период планирования, в котором распределяются ресурсы и строится расписание выполнения работ  $((b_i, f_i] \subseteq (0; T]$  при всех  $i = \overline{1, n}$ ). Не возобновляемый ресурс выделяется работам в моменты времени  $\tau_k, k = \overline{0, K}$ , где  $0 \leq \tau_0 < \tau_1 < \dots < \tau_{K-1} < \tau_K \leq T$ . Имеющаяся величина не возобновляемого ресурса в момент времени  $\tau_k$  составляет  $R_k$ , а максимальная величина этого ресурса, которая может быть использована заданием  $w_i$  в момент времени  $\tau_k$ , равна  $r_{ik}^0, i = \overline{1, n}, k = \overline{1, K}$ . Работе  $w_i$  не возобновляемый ресурс может быть выделен в момент  $\tau_k$  только в том случае, если  $b_i \leq \tau_k \leq f_i - d_{ik}, d_{ik} \geq 0$  — заданная величина (т.е. на освоение этого ресурса должно остаться время, не меньшее  $d_{ik}$ ),  $i = \overline{1, n}, k = \overline{1, K}$ . Выделение в момент времени  $\tau_k$  заданию  $w_i$  не возобновляемого ресурса в количестве  $r_{ik}$  обеспечивает объем работы по выполнению этого задания, равный  $c_k r_{ik}, i = \overline{1, n}, k = \overline{1, K}$ , где  $c_k > 0$  — заданные величины. При этом не должны нарушаться следующие ограничения:

$$r_{ik} \leq r_{ik}^0, i = \overline{1, n}, k = \overline{1, K}, \quad (1)$$

$$\sum_{i=1}^n r_{ik} \leq R_k, k = \overline{1, K}. \quad (2)$$

Выполнение задания с помощью не возобновляемого ресурса не влияет на его выполнение с помощью возобновляемых и не возобновляемых ресурсов, выделенных ему в другие моменты времени.

Распределение ресурсов, а также соответствующее ему расписание выполнения работ  $W$  указывают для каждого задания  $w_i \in W, i = \overline{1, n}$ , и каждого моменты времени  $t \in (0, T]$ , выполняется ли оно каким-либо процессором, а для каждого момента времени  $\tau_k, k = \overline{1, K}$ , определяют количество не возобновляемого ресурса, выделенного ей. При этом требуется выполнение неравенств (1), (2).

Распределение ресурсов, при котором каждое задание  $w_i \in W, i = \overline{1, n}$ , полностью вы-

полняется (т.е. суммарный объем работ, предоставленный ему обоими видами ресурсов, составляет  $Q_i$ ) в своем директивном интервале, будем называть допустимым, а соответствующее расписание — допустимым расписанием.

Требуется определить, существует ли допустимое распределение ресурсов и соответствующее допустимое расписание, и найти их, если они существуют.

Рассмотрим пример сформулированной задачи. Для выполнения комплекса заданий имеется несколько процессоров (возобновляемые ресурсы), каждый из которых может быть использован неограниченное число раз и передаваться от одного задания к другому. Помимо процессоров для выполнения заданий в определенные моменты времени выделяются некоторые дополнительные устройства (не возобновляемый ресурс), которые не могут быть использованы повторно другими работами. Объем работы, выполняемый процессором, равен продолжительности его использования, а объем работы, выполняемый дополнительными устройствами, пропорционален их количеству. Каждое задание характеризуется объемом работ, который необходимо выполнить, моментом готовности, директивным сроком окончания, а также некоторыми ограничениями на использование процессоров и дополнительных устройств. Требуется распределить имеющиеся ресурсы таким образом, чтобы каждое задание было полностью выполнено в заданном для него временном интервале, или установить, что такого распределения ресурсов не существует.

### 3. Построение потоковой сети

Пусть  $0 = x_0 < x_1 < \dots < x_{p-1} < x_p = T$  — все различные величины  $b_1, b_2, \dots, b_n, f_1, f_2, \dots, f_n$ .

Определим интервалы  $I_j = (x_{j-1}, x_j], j = \overline{1, p}$ ,

и пусть  $\delta_j = x_j - x_{j-1}$  — длина интервала  $I_j$ .

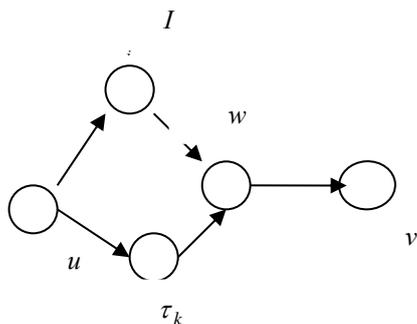
Отметим, что при всех  $j = \overline{1, p}, i = \overline{1, n}$  либо  $I_j \subseteq (b_i, f_i]$ , либо  $I_j \cap (b_i, f_i] = \emptyset$ .

Построим потоковую сеть  $G = (V, A)$ , где  $V = \{u, v, I_j, j = \overline{1, p}, \tau_k, k = \overline{1, K}, w_i, i = \overline{1, n}\}$  — множество узлов,  $u$  — источник,  $v$  — сток,  $A = \{(u, I_j), j = \overline{1, p}, (u, \tau_k), k = \overline{1, K}, (I_j, w_i), j = \overline{1, p}, i = \overline{1, n}, (\tau_k, w_i), k = \overline{1, K}, i = \overline{1, n}, (w_i, v), i = \overline{1, n}\}$  — множество ориентированных дуг.

Дуги  $(I_j, w_i)$  и  $(\tau_k, w_i)$  включаются в сеть  $G$  в том и только том случае, если выполнены усло-

вия  $I_j \subseteq (b_i, f_i]$  и  $b_i \leq \tau_k \leq f_i - d_{ik}$  соответственно. Фрагмент сети  $G$  изображен на рисунке. Пропускные способности дуг сети  $G$  указаны в таблице.

Будем рассматривать поток  $g$  в сети  $G$ . Величину потока  $g$  по дуге  $(a, b) \in A$  будем обозначать через  $g(a, b)$ .



$$j = \overline{1, p}; k = \overline{1, K}; i = \overline{1, n}$$

Рисунок. Фрагмент сети  $G$

Таблица. Параметры дуг сети  $G$

Дуга	Пропускная способность
$(u, I_j)$	$m\delta_j$
$(u, \tau_k)$	$c_k R_k$
$(I_j, w_i)$	$\delta_j$
$(\tau_k, w_i)$	$c_k r_{ik}^0$
$(w_i, v)$	$Q_i$

#### 4. Существование допустимого расписания

Докажем следующую лемму о необходимых и достаточных условиях существования допустимого распределения ресурсов и допустимого расписания. Лемма является обобщением соответствующего утверждения в [1].

*Лемма.* Допустимое распределение ресурсов существует в том и только том случае, когда в сети  $G$  существует поток  $g$ , для которого

$$g(w_i, v) = Q_i \quad (3)$$

при всех  $i = \overline{1, n}$ .

**Доказательство.** 1) Предположим, что в сети  $G$  существует поток  $g$ , для которого выполнены равенства (3). Тогда

$$g(I_j, w_i) \leq \delta_j \quad (4)$$

при всех  $i = \overline{1, n}, j = \overline{1, p}$ . Кроме того,  $g(u, I_j) \leq m\delta_j$  при всех  $j = \overline{1, p}$ . Из сохранения потока в узлах  $I_j$  следует, что  $g(u, I_j) = \sum_{i=1}^n g(I_j, w_i)$  при всех  $j = \overline{1, p}$ . Следовательно,

$$\sum_{i=1}^n g(I_j, w_i) \leq m\delta_j \quad (5)$$

при всех  $j = \overline{1, p}$ . Из (4), (5) следует, что если рассматривать величину  $g(I_j, w_i)$  как объем работы процессоров по выполнению задания  $w_i$  в интервале  $I_j$ , то будут выполнены необходимые и достаточные условия существования допустимого распределения процессоров в интервале  $I_j$ . Это распределение находится с помощью алгоритма упаковки [1] для каждого  $j = \overline{1, p}$

Далее,  $g(\tau_k, w_i) \leq c_k r_{ik}^0$  при всех  $i = \overline{1, n}, k = \overline{1, K}$ . Следовательно, существуют такие величины  $r_{ik}$ ,  $r_{ik} \leq r_{ik}^0$ , что  $g(\tau_k, w_i) = c_k r_{ik}$ , т.е.  $c_k r_{ik} \leq c_k r_{ik}^0$ , откуда следует выполнение неравенства (1). Кроме того,  $g(u, \tau_k) \leq c_k R_k$  при всех  $k = \overline{1, K}$ . Из сохранения потока в узлах  $\tau_k$  следует, что  $g(u, \tau_k) = \sum_{i=1}^n g(\tau_k, w_i)$  при всех  $k = \overline{1, K}$ . Сле-

довательно,  $\sum_{i=1}^n c_k r_{ik} \leq c_k R_k$ , откуда следует неравенство (2). Таким образом, в каждый момент времени  $\tau_k$ ,  $k = \overline{1, K}$ , существует такое распределение не возобновляемого ресурса в количестве, не превосходящем  $R_k$ , что каждому заданию  $w_i$ ,  $i = \overline{1, n}$ , выделяется  $r_{ik}$  единиц этого ресурса, что обеспечивает объем работы, равный  $g(\tau_k, w_i) = c_k r_{ik}$ .

Далее, из сохранения потока в каждом узле  $w_i$ ,  $i = \overline{1, n}$ , следует что

$$\sum_{j=1}^p g(I_j, w_i) + \sum_{k=1}^K g(\tau_k, w_i) = g(w_i, v),$$

а с учетом (3), получаем

$$\sum_{j=1}^p g(I_j, w_i) + \sum_{k=1}^K g(\tau_k, w_i) = Q_i,$$

т.е. каждое задание  $w_i$ ,  $i = \overline{1, n}$ , выполнено полностью. Из структуры сети  $G$  следует, что каждое задание выполняется в своем директивном интервале. Таким образом, допустимое распределение ресурсов и допустимое расписание существуют.

2) Пусть существуют допустимое распределение ресурсов и допустимое расписание, т.е. каждому заданию  $w_i$ ,  $i = \overline{1, n}$ , в интервале  $I_j$ ,  $j = \overline{1, p}$ , предоставляется объем работы процессоров, равный  $h_{ij} \leq \delta_j$ , а в моменты времени  $\tau_k$ ,  $k = \overline{1, K}$ , выделяется не возобновляемый ресурс в количестве  $r_{ik}$ , что обеспечивает объем работы, равный  $c_k r_{ik}$ , при выполнении неравенств (1), (2). В силу допустимости этого распределения ресурсов справедливо равенство

$$\sum_{j=1}^p h_{ij} + \sum_{k=1}^K c_k r_{ik} = Q_i \quad (6)$$

при всех  $i = \overline{1, n}$ . Определим поток  $g$  в сети  $G$  следующим образом:

$$g(I_j, w_i) = h_{ij}, \quad i = \overline{1, n}, \quad j = \overline{1, p},$$

$$g(u, I_j) = \sum_{i=1}^n h_{ij}, \quad j = \overline{1, p},$$

$$g(\tau_k, w_i) = c_k r_{ik}, \quad i = \overline{1, n}, \quad k = \overline{1, K},$$

$$g(u, \tau_k) = \sum_{i=1}^n c_k r_{ik}, \quad k = \overline{1, K},$$

$$g(w_i, v) = \sum_{j=1}^p h_{ij} + \sum_{k=1}^K c_k r_{ik}, \quad i = \overline{1, n}.$$

В каждом внутреннем узле сети  $G$  (т.е. в узлах  $I_j$ ,  $j = \overline{1, p}$ ,  $\tau_k$ ,  $k = \overline{1, K}$ ,  $w_i$ ,  $i = \overline{1, n}$ ) выполнено условие сохранения потока и справедливы соотношения

$$g(I_j, w_i) = h_{ij} \leq \delta_j, \quad i = \overline{1, n}, \quad j = \overline{1, p},$$

$$g(u, I_j) = \delta_j m, \quad j = \overline{1, p},$$

$$g(\tau_k, w_i) = c_k r_{ik} \leq c_k r_{ik}^0, \quad i = \overline{1, n}, \quad k = \overline{1, K},$$

$$g(u, \tau_k) = \sum_{i=1}^n c_k r_{ik} = c_k \sum_{i=1}^n r_{ik} \leq c_k R_k, \quad k = \overline{1, K},$$

(в силу (2)), т.е. выполнены ограничения сверху на потоки по дугам  $(I_j, w_i)$ ,  $(u, I_j)$ ,  $(\tau_k, w_i)$ ,  $(u, \tau_k)$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, p}$ ,  $k = \overline{1, K}$ , сети  $G$ . Кроме того, в силу (6)

$$g(w_i, v) = \sum_{j=1}^p h_{ij} + \sum_{k=1}^K c_k r_{ik} = Q_i, \quad i = \overline{1, n},$$

т.е. выполнено условие (3). Лемма доказана.

## 5. Решение исходной задачи

Опишем алгоритм решения исходной задачи.

1) Построить потоковую сеть  $G$ .

2) Найти в сети  $G$  максимальный поток  $g$ .

3) Если выполнено условие (3) при всех  $i = \overline{1, n}$ , то допустимое распределение ресурсов и допустимое расписание существуют. Величина  $g(I_j, w_i)$  потока  $g$  по дуге  $(I_j, w_i)$  равна объему работы процессоров по выполнению задания  $w_i$  в интервале  $I_j$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, p}$ . Распределение процессоров по заданиям в каждом интервале  $I_j$  строится с помощью алгоритма упаковки [1]. Величина  $g(\tau_k, w_i)$  потока  $g$  по дуге  $(\tau_k, w_i)$  равна объему не возобновляемого ресурса, выделяемого заданию  $w_i$  в момент времени  $\tau_k$ ,  $i = \overline{1, n}$ ,  $k = \overline{1, K}$ .

4) Если условие (3) не выполнено хотя бы для одного  $i$ ,  $1 \leq i \leq n$ , то допустимого распределения ресурсов и допустимого расписания не существует.

Поясним шаг (4) алгоритма. Если для максимального потока  $g$  условие (3) не выполнено хотя бы при одном  $i$ ,  $1 \leq i \leq n$ , то оно не выполнено ни для какого другого потока. Действительно, предположим, что условие (3) выполнено для некоторого потока  $\bar{g}$  при всех  $i = \overline{1, n}$ . Тогда величина потока  $\bar{g}$  через разрез  $C = \{u, I_j, j = \overline{1, p}, \tau_k, k = \overline{1, K}, w_i, i = \overline{1, n}\} \cup \{v\}$  равна величине этого разреза. Следовательно, по теореме о максимальном потоке и минимальном разрезе [7] поток  $\bar{g}$  является максимальным, а разрез  $C$  – минимальным. Но величина потока  $g$  через разрез  $C$  меньше величины этого разреза. Следовательно, поток  $g$  не может быть максимальным. Полученное противоречие обосновывает шаг (4) алгоритма.

Определим вычислительную сложность предложенного алгоритма. Построение сети  $G$  (шаг 1) требует  $O((p+K)n)$  операций. Сложность нахождения максимального потока в сети  $G$  (шаг 2) (например, с помощью алгоритма Карзанова) составляет  $O((p+K+n)^3)$ , а проверка условия (3) и применение алгоритма упаковки для одного интервала  $I_j$ ,  $1 \leq j \leq p$ , тре-

бует  $O(m)$  операций. Следовательно, вычислительная сложность предложенного алгоритма составляет  $O((p + K + n)^3 + pnm)$  операций сложения, вычитания, сравнения и деления. Таким образом, предложенный алгоритм является полиномиальным.

## 6. Заключение

Разработан полиномиальный алгоритм по-

строения расписания с прерываниями выполнения комплекса работ при наличии двух типов ресурсов (возобновляемых и не возобновляемых), заданных директивных интервалов и объемов выполняемых работ. Величина не возобновляемого ресурса может изменяться во времени. Алгоритм основан на сведении исходной задачи к задаче нахождения максимального потока в сети.

# Work planning in a task with mixed resource types

Meran Furugyan

**Abstract.** The problem of preemptive scheduling in the presence of two types of the resources used for performance of complex of jobs – renewable (processors) and not renewable, is considered. Directive intervals and volume of jobs are specified. Not renewable resource levels can vary over time. In order to find a feasible resource allocation and to build a schedule of jobs, an algorithm based on finding the maximum flow in the network has been developed.

**Keywords:** renewable and non-renewable resources, feasible preemptive schedule, network flow

## Литература

1. В.С. Танаев, В.С. Гордон, Я.М. Шафранский. Теория расписаний. Одностадийные системы. М., Наука, 1984.
2. T Gonzales, S. Sahni. Preemptive Scheduling of Uniform Processor Systems // J. Association for Computing Machinery. 1978. V. 25. № 1. P. 92 – 101.
3. A. Federgruen, H. Groenevel. Preemptive Scheduling of Uniform Machines by Ordinary Network Flow Technique // Management Science. 1986. V. 32. № 3. P. 341 – 349.
4. А.А. Лазарев. Теория расписаний. Оценка абсолютной погрешности и схема приближенного решения задач теории расписаний. М., МФТИ, 2008.
5. М.Г. Фуругян. Планирование вычислений в многопроцессорных системах с несколькими типами дополнительных ресурсов и произвольными процессорами // Вестн. МГУ. Сер. 15, Вычисл. математика и кибернетика. 2017. № 3. с. 38 – 45.
6. Д.А. Кононов, М.Г. Фуругян. Эффективное управление региональными проектами: распределение ресурсов и директивное планирование. Материалы двенадцатой международной конференции “Управление развитием крупномасштабных систем. MLSD’2019. (1–3 октября 2019 г., Москва, Россия). С. 811 – 813.
7. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.. Алгоритмы. Построение и анализ. (Второе издание). М., Вильямс, 2005.

# Особенности программирования троичной машины: новые возможности и новые задачи

А.А. Бурцев<sup>1</sup>

<sup>1</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия, burtsev@niisi.msk.ru

**Аннотация.** В статье рассматриваются новые выразительные средства построения программ, которые может предоставить программисту троичная машина и которые обеспечивает система программирования ДССП для троичной виртуальной машины (ТВМ). А также демонстрируется способ построения формулы вычисления троичной логической функции по заданной таблице значений с последующей её реализацией в ДССП-ТВМ.

**Ключевые слова:** троичный компьютер, троичная арифметика, троичная логика, троичная функция, ЦМ “Сетунь” и “Сетунь-70”, троичная виртуальная машина (ТВМ), ДССП, ДССП-ТВМ.

## 1. Введение

В настоящее время общепризнано, что традиционная двоичная микроэлектроника уже почти достигла потолка своих потенциальных возможностей. Поэтому сейчас назрела острая потребность расширить область исследований, направленных на поиск новых путей развития вычислительной техники. В качестве одного из перспективных вариантов дальнейшего развития отечественной компьютерной индустрии может рассматриваться и возможность построения на современной элементной базе троичных компьютеров, т.е. компьютеров, функционирование которых основывается на применении троичной симметричной системы счисления, а также присущей ей троичной логики и троичной арифметики.

Когда-то на заре развития вычислительной техники, называя троичную симметричную систему счисления (с цифрами  $-1, 0, +1$ ) “самой изящной” [1], Д.Кнут отмечал её безусловные преимущества (в частности, перед традиционной двоичной, на которой основана вся современная вычислительная техника) и предрекал ей успешное применение в будущем. А это будущее стало реальностью (правда, лишь на некоторое время) уже во второй половине XX века с разработкой в НИЛ ЭВМ МГУ (под руководством Брусенцова Н.П.) троичных цифровых машин (ЦМ) “Сетунь”(1958) и “Сетунь-70”(1970) [2], позволивших на практике ощутить преимущества троичной логики и арифметики.

Спустя почти полвека сотрудники НИЛ троичной информатики (ТИ) ВМК МГУ предприняли попытки возродить на современной основе троичный компьютер, воплотив в нём харак-

терные черты ЦМ “Сетунь-70”: двухстековую архитектуру с управляющими командами процедурного структурированного программирования. Программисты создали имитатор [3] троичной машины и кросс-систему ДССП-ТВМ [4] разработки программ для неё. А разработчики аппаратуры придумали [5], как можно создать на современной элементной базе такие микросхемы, которые смогли бы функционировать аналогично троичным элементам ЦМ “Сетунь” и “Сетунь-70”, и тем самым обозначили принципиальную возможность реализации в настоящее время подобной троичной машины «в кремнии».

Созданный в МГУ программный комплекс ДССП-ТВМ [6] можно использовать в настоящее время в качестве среды программирования [7] для разработки и прогона учебных программ для троичной машины. Что позволяет готовить практических специалистов в области троичной информатики [8].

Для переноса на троичную машину программ, созданных ранее для двоичных машин, приходится учитывать ряд особенностей, характерных для троичных операций обработки данных. Следует учитывать, что на троичной машине с симметричной системой счисления результаты некоторых операции над числами могут оказаться отличными от тех, которые получались на машине с обычной двоичной позиционной системой счисления (см. п. 2.1)

Но такую специфику троичных операций можно легко освоить. А что же дальше? Что такого нового может предоставить троичная машина, чего не умеет делать двоичная? Подобные ставшие уже традиционными вопросы задают обычно специалисты, сомневающиеся в особых преимуществах троичных компьютеров.

Опыт программирования, полученный авто-

ром при разработке троичных программ для ТВМ (в ДССП-ТВМ) и для ЦМ «Сетунь-70» (на ассемблере), позволяет с уверенностью утверждать, что ассортимент выразительных средств построения программного обеспечения, предоставляемый троичной машиной, не только полностью обеспечивает привычный арсенал средств, обеспечиваемый программисту на двоичной машине, но и значительно перекрывает его, обогащая программиста новыми приёмами разработки программ, которые двоичными машинами и языками программирования для них либо вообще никогда не предусматривались, либо широко не применялись на практике в силу неэффективности их реализации. Вот об этих новых возможностях программирования, предоставляемых троичной машиной, и пойдёт, главным образом, речь в предлагаемой статье.

В качестве таковых далее рассматриваются: команда организации троичного ветвления, новая команда цикла с троичным условием, способы построения троичного логического выражения с применением поразрядных операций троичной логики (и арифметики), а также приёмы эффективной реализации операций для работы с троичными множествами. Приводятся варианты воплощения этих команд и операций в ЦМ «Сетунь-70», в троичной виртуальной машине (ТВМ) и в системе программирования ДССП для ТВМ. Демонстрируются примеры программ, разработанных в кросс-системе ДССП-ТВМ (или в среде интерпретатора ДССП/ТВМ), применяющих такие специфические «троичные» команды и операции. Сравнивая их с аналогичными программами, разработанными для решения тех же задач с применением арсенала традиционных выразительных средств, обеспечиваемых обычно языками программирования для двоичных машин, можно убедиться, что реализация таких программ на троичной машине позволяет не только упростить сам алгоритм решения задачи, но и добиться в итоге улучшения его производительности.

Применение троичных машин неизбежно приводит и к расширению круга решаемых на компьютере задач, а также порождает и новые проблемы, без решения которых невозможно рассчитывать на успешное внедрение троичной вычислительной техники. Образно говоря, новые возможности как бы позволяют выйти из плоскости в пространство и увидеть на горизонте новые задачи. Одной из них является задача построения формулы вычисления троичной функции по заданной таблице её значений на основе имеющихся базовых операций троичной логики и арифметики. В булевой алгебре широко известен универсальный метод постро-

ения формулы логической функции по её таблице истинности, приводящий к получению совершенной ДНФ или совершенной КНФ, пригодных для реализации требуемой функции на основе трёх базисных операций: конъюнкции (&), дизъюнкции (∨) и отрицания (~).

Аналогичный универсальный метод построения формулы вычисления функции троичной логики, возможно, тоже существует, но, к сожалению, он не так широко известен. Автор на практике убедился, что иногда предпочтительнее самому изобрести такой метод заново, чем искать его описание на просторах Интернета или печатных страниц научных книг и журналов. Опробовав такой «заново придуманный» метод на практике при реализации некоторых двухместных функций троичной логики в ДССП-ТВМ, автор предлагает ознакомиться с этим методом тем специалистам, кто проявляет живой интерес к практическому освоению приёмов разработки программ для троичных машин.

## 2. Специфика вычислительной обработки в троичной машине

Допустим, нам требуется перенести на троичную машину какую-то вычислительную программу, которая ранее уже была разработана для двоичной машины (скажем, на языке Си). Какие препятствия или так называемые «подводные камни» нам могут встретиться на этом пути? Рассмотрим их поподробнее.

### 2.1. Целочисленная арифметика

В троичной симметричной системе счисления единообразно представляются положительные и отрицательные числа. И результат выполнения целочисленной операции в такой троичной машине всегда знаковый и потому остаётся корректным, если он укладывается в 27-разрядную сетку, т.е. не превосходит по модулю значение  $(3^{27}-1)/2$ . (Заметим, что  $3^{27} = 7625597484987 \gg 2^{32} = 4294967296$ ).

Но не всякая операция целочисленной арифметики будет исполняться на троичной машине с симметричной системой счисления так же, как на двоичной. Есть исключения.

В первую очередь, следует отметить, что в такой троичной машине операция целочисленного деления вычисляет в качестве результата частное и остаток так, чтобы последний оказался как можно меньше по абсолютной величине. При этом (в отличие от обычной операции деления) значение остатка может оказаться не только положительным, но и отрицательным.

В некоторых случаях результат такого деления на троичной машине может совпадать с результатом, полученным на двоичной. Напри-

мер,  $16\%3 = [5,1]$  (т.к.  $5*3+1=16$ ). А в других случаях он может отличаться. Так, на двоичной машине  $17\%3 = [5,2]$  ( $5*3+2=17$ ), а на троичной машине  $17\%3 = [6,-1]$  ( $6*3-1=17$ ).

В базовом словаре ДССП-ТВМ слово, обозначаемое знаком деления /, выполняет целочисленное деление, получая результат [частное, остаток] как раз по описанным выше правилам, характерным для троичной симметричной системы счисления. А для выполнения привычного целочисленного деления в ДССП-библиотеке предусмотрено определение слова /DivMod:

```
: /DivMod {X,Y}
  C2 ABS C2 ABS {X,Y,X|,|Y} / {X,Y,D,M}
  {корректируем результат, если остаток < 0 : }
  {X,Y,D,M} C IF- D-1 M+|Y|
  {корректируем знаки результатов: }
  {X,Y,D,M} C4 IF- NEG {меняем знак для M}
  {X,Y,D,M} E4 SGN C3 SGN <>
  IF+ NEG {и знак для D, если sign(X)<>sign(Y)}
  {M,Y,D} E2D E2 {D,M} ;
: D-1 M+|Y| {X,Y,D,M}
  C3 ABS + E2 1- E2 {D-1,M+|Y|} ;
```

Другое исключение касается операций сдвигов (<<, >>), которые могут применяться при вычислении арифметического выражения (например, в языке Си) для умножения и деления значения на степень двойки ( $2^n$ ). При исполнении таких операций на троичной машине результат окажется умноженным (или поделённым) уже на степень тройки ( $3^n$ ).

Подобные особенности обработки целых чисел не позволяют, вообще говоря, легко перенести на троичную машину многие программы, уже разработанные для двоичных машин.

## 2.2. Операции доступа к трайтам и троичным словам памяти

Есть ещё одна причина, которая может затруднить автоматический перенос существующего ПО на троичную машину. Это различие в строении и соответственно в адресации памяти у двоичных и троичных машин.

Память в двоичной машине обычно адресуется с точностью до байта. Так что при обработке машинных слов, образованных из нескольких байтов, например, элементов массивов, хранимых в 16-, 32- или 64-разрядных машинных словах, для передвижения к следующему слову требуется прибавлять к адресу-указателю соответственно значение 2, 4 или 8.

Память троичного компьютера будет адресоваться с точностью до трайта. А диапазон адресов будет охватывать пространство от отрицательного значения  $-P$ , до положительного значения  $+P$ , где  $P=(3^{27}-1)/2$ , если для адреса отводится 27 разрядов. Причём нулевое значение адреса будет указывать не на начало всего пространства памяти (как это обычно бывает в

двоичной машине), а на его середину. Что тоже приходится учитывать при размещении программ и данных в памяти троичной машины.

В троичной машине при обработке элементов массивов, занимающих два трайта или полное 27-титное машинное слово (3 трайта), для передвижения к следующему элементу к адресу-указателю придётся прибавлять уже совсем другие значения (2 или 3 соответственно). А значит, все программы, работающие с массивами и указателями, потребуются тщательно переделывать для переноса на троичную машину.

В ТВМ, разработанной в НИИ троичной информатики на ВМК МГУ, можно читать и записывать в память по указанному адресу троичные слова, трайты и двухтрайтовые значения. Для этого предусмотрены команды (слова в языке ДССП-Т) @W, @T, @TT для взятия значения из памяти в вершину стека, и команды !W, !T, !TT для записи в память троичного значения из стека.

В качестве примера демонстрации обработки троичных слов в памяти приведём простую процедуру сортировки по возрастанию вектора 27-титных значений (на языке ДССП-Т):

```
{A - адрес вектора, n= кол-во элементов
: SortVctr{A,n} 1- DO- Sorti D { } ;
: Sorti{A,i} 1+ Maxi {A,i,k}
  C2 C2 <> IF+ Exchi D 1- {A,i} ;
: Maxi {опред индекс k макс.эл-та среди A0..Ai}
  {A,i} C C {A,i,k,i} DO- Maxj {A,i,k} ;
: Maxj{A,i,k,j} C4 C {~,A,A} C3 @[x] E2
  {~,Aj,A} C4 @[x] {~,Aj,Ak}
  > IF+ k:=j {A,i,k,j} ; {j=i-1,..,0}
: k:=j {k,j} E2 D C {j,j} ;
: @[x] {A,x} *3 + @W {A[x]} ;
: Exchi {A,i,k} {переставляет A[i]↔A[k]}
  C3 C3 *3 + {~, 'Ai} C4 C3 *3 +
  {~, 'Ai, 'Ak} C @W C3 @W E4
  {~,Ai,'Ak,Ak,'Ai} !W {A[i]:=Ak}
  !W {A[k]:=Ai} {A,i,k} ;
```

Процедура SortVctr сортирует вектор методом простого выбора. На каждом  $i$ -ом шаге цикла ( $i=n-1, \dots, 1$ ) она находит среди элементов вектора  $A0..Ai$  максимальный, используя процедуру Maxi для определения его индекса  $k$ , и переставляет этот  $k$ -ый элемент с  $i$ -ым элементом, используя процедуру Exchi.

## 2.3. Поразрядная (потритная) обработка троичного слова

Ещё одну проблему при переносе ПО на троичную машину могут предоставить программы, в которых как отдельные величины обрабатываются значения, размещённые внутри одного машинного слова.

Для доступа к таким величинам в программах обычно применяются операции логических сдвигов (<<, >>), поразрядной конъюнкции (&),

дизъюнкции ( $\vee$ ), и отрицания ( $\sim$ ), которые позволяют выделять в машинном слове отдельные биты или битовые поля, образуемые соседними разрядами одного машинного слова.

При разработке программ для троичной машины нужны аналогичные операции, чтобы обеспечить доступ к отдельным тритам и троичным значениям полей, образованных соседними тритами машинного слова. Для поразрядной обработки данных внутри троичного машинного слова ДССП-ТВМ предоставляет набор команд для поразрядного умножения, сложения, инверсии (**TMUL**, **TADD**, **NEG**), а также команды сдвигов влево и вправо на одну или несколько позиций (**SHL**, **SHR**, **SHT**).

Продемонстрируем характерные приёмы использования этих троичных операций для обработки отдельных разрядов и полей троичного машинного слова. И приведём для сравнения примеры аналогичных действий на языке Си с битами и битовыми полями:

взять значение V из разрядов [9:12] слова S	
ДССП-Т	{S} -9 SHT #44 TMUL {V}
Си	V = (S >> 9) & (0xF)

записать значение V в разряды [9:12] слова S	
ДССП-Т	{S,V} #44 TMUL +9 SHT E2 #144444443014444 TMUL TADD{S'}
Си	S = S & (0xFFFFE1FF)   ((V & 0xF) << 9)

взять значение L из 5-го разряда слова S	
ДССП-Т	{S} #300 TMUL -5 SHT {L}
Си	L = ((S & 0x2F) >> 5)

записать значение L в 5-й разряд слова S	
ДССП-Т	{S,L} #1 TMUL +5 SHT E2 #144444444444144 TMUL TADD{S'}
Си	S = S & (0xFFFFFDF)   ((L & 0x1) << 5)

Сложнее реализовать операции чтения и записи троичного флага, если разряд этого флага в троичном слове не фиксируется заранее, а определяется в ходе исполнения программы. Тогда необходимые значения масок нельзя приготовить заранее в виде констант: их приходится формировать динамически в зависимости от заданного номера (n) разряда троичного флага.

Покажем, например, как можно определить в ДССП-ТВМ такие операции чтения и записи над произвольным разрядом вершины стека:

```
{ запись значения флага L в n-й трит вершины }
: T!I {V[27:0],L,n} E2 #1 TMUL
  C2 SHT E3 {L<<n,n,V[27:0]} E2
  M+j0 TMUL TADD {V'[27:0]} ;
{ M+j0 =маска, где M[j]=0, M[i]=+1 для i<>j }
```

```
{ чтение значения n-го трита вершины стека }
: T@I {V[27:0],n}
  NEG SHT #1 TMUL {V[n]} ;
```

### 3. Троичное условие и троичное логическое выражение

В нашей повседневной жизни достаточно часто встречается необходимость троичного ответа на вопрос. Например, на вопрос «как закончился матч?» возможны три варианта ответа: выигрыш – ничья – проигрыш. На хрестоматийный вопрос богатырю из русских сказок: «куда двигаться?» возможен ответ: влево – прямо – направо. Ещё примеры ответов с тремя значениями: дружба – нейтралитет – вражда; больше – равно – меньше; выше – вровень – ниже; тяжелее – равновесно – легче; горячее – одинаковой температуры – холоднее; лучше – нормально – хуже; вчера – сегодня – завтра.

По сути на любой вопрос, ответить на который нельзя ни утвердительно («да»), ни отрицательно («нет»), не получив нужной информации, в качестве третьего варианта надо предусматривать ответ «не знаю». Например, в настоящее время наука вряд ли может чётко ответить (да или нет) на вопрос «есть ли жизнь на Марсе?». И тут ответ «неизвестно» будет наиболее подходящим.

Вычислять выражения троичной логики, предусматривающие не два, а три варианта ответа, на троичной машине можно непосредственно (напрямую). При этом значения +1, 0, -1 можно трактовать по-разному. Например, так: «да» (+1), «нет» (-1), «не знаю» (0); или немного иначе: «да» (+1), «нет» (0), «не знаю» (-1). А путём составления троичных логических выражений (условий) можно более эффективно решать некоторые задачи.

В базовом наборе команд ДССП-ТВМ имеются операции, выдающие в качестве результата троичное логическое значение. Это операция оценки знака числа **SGN** и операция сравнения двух чисел **CMR**. А также предусмотрены операции, которые позволяют производить поразрядные действия с отдельными тритами машинного слова. Это потритная инверсия **NEG**, потритный минимум **TMIN** и потритный максимум **TMAX**. С помощью этих операций можно составлять (на языке ДССП-Т) сколь угодно сложные троичные логические выражения.

В качестве первого примера определим в ДССП для ТВМ слово-выражение **TSEG** для троичной проверки попадания точки **z** в сегмент прямой **[x,y]**, которое в качестве результата **L** выдаёт одно из трёх значений: 1)  $L=+1$ , если точка **z** расположена внутри отрезка ( $x < z < y$ ); 2)  $L=0$ , если точка **z** лежит на одной из границ отрезка ( $z=x$  или  $z=y$ ); 3)  $L=-1$ , если точка **z** находится вне отрезка ( $z < x$  или  $z > y$ ):

```
: TSEG{x,y,z} C E4
{z,y,z,x} CMP {z?x} E3{z?x,y,z}
CMP {z?x,y?z} TMIN {L} ;
```

и сравним его с фрагментом (на языке Си), который позволяет решить эту задачу на двоичной машине:

```
if((x<z)&&(z<y)) L=+1; else
if((z=x)|| (z=y)) L= 0; else L=-1;
```

Заметим, что для двоичной машины (на языке Си) при выполнении такой троичной проверки приходится использовать команды ветвления, а на языке ДССП-Т (для троичной машины) можно без них обойтись, используя лишь логические операции.

Аналогично можно составить логические выражения для троичной проверки попадания точки с координатами (x,y) в заданную фигуру на плоскости. Результат такой проверки = +1 (если точка внутри фигуры); 0 (если точка на границе фигуры); -1 (если точка вне фигуры).

В качестве примеров составим определения слов (на языке ДССП-Т) для выполнения троичных проверок (условий) попадания точки с заданными координатами в следующие две фигуры: А) прямоугольник размерами g×h, левый нижний угол которого расположен в начале координат; и В) круг радиуса r с центром в начале координат:

```
троичное условие попадания точки (x,y) в фигуру А:
прямоугольник высотой h и шириной g, левый нижний
угол которого расположен в начале координат
: inFigA? {x,y,g,h} E4 0 E3 E2
{h,y,0,g,x} TSEG {h,y,x?in[0,g]}
E3 0 E3 {x?in[0,g],0,h,y} TSEG
{x?in[0,g],y?in[0,h]} TMIN {z=+1|0|-1} ;
```

```
троичное условие попадания точки (x,y) в фигуру В:
круг радиуса r с центром в начале координат
: inFigB? {x,y,r} C * E3 C * E2 C * +
{r^2,x^2+y^2} CMP {z= +1| 0|-1} ;
```

Применяя дополнительные операции потритной инверсии, а также потритного максимума и минимума как логические связки, можно выразить и более сложные троичные выражения-условия попадания точки в объединение (A∪B) или пересечение двух фигур (A∩B). А также составить соответствующие проверки для фигур, образованных вырезанием фигуры В из фигуры А (A\B), или наоборот, фигуры А из фигуры В (B\A):

```
троичное условие попадания точки в фигуру A∪B
: inFigA+B? {x,y,r,g,h}
E3 C4 6 CT E3 {x,y,h,g,x,y,r} inFigB?
5 ET {(x,y)inFigA?,y,h,g,x} E4 E3
{(x,y)inFigA?,x,y,g,h} inFigA?
{(x,y)inFigA?,(x,y)inFigB?} TMAX
{z= (x,y)inFig(A+B)?} ;
```

```
троичное условие попадания точки в фигуру A∩B
```

```
: inFigA*B? {x,y,r,g,h}
E3 C4 6 CT E3 {x,y,h,g,x,y,r} inFigB?
5 ET {(x,y)inFigA?,y,h,g,x} E4 E3
{(x,y)inFigA?,x,y,g,h} inFigA?
{(x,y)inFigA?,(x,y)inFigB?} TMIN
{z= (x,y)inFig(A*B)?} ;
```

```
троичное условие попадания точки в фигуру A\B
```

```
: inFigA\B? {x,y,r,g,h}
E3 C4 6 CT E3 {x,y,h,g,x,y,r} inFigB?
NEG 5 ET {~(x,y)inFigA?,y,h,g,x} E4 E3
{~(x,y)inFigA?,x,y,g,h} inFigA?
{~(x,y)inFigA?,(x,y)inFigB?} TMIN
{z= (x,y)inFig(A\B)?} ;
```

```
троичная проверка попадания точки в фигуру B\A
```

```
: inFigB\A? {x,y,r,g,h}
E3 C4 6 CT E3 {x,y,h,g,x,y,r} inFigB?
5 ET {(x,y)inFigA?,y,h,g,x} E4 E3
{(x,y)inFigA?,x,y,g,h} inFigA?
{(x,y)inFigA?,(x,y)inFigB?} NEG TMIN
{z= (x,y)inFig(B\A)?} ;
```

Другие примеры составления троичных логических выражений для вычисления функций троичной логики рассматриваются в п.6.

## 4. Троичное множество

Для представления обычных множеств на двоичных машинах обычно резервируется массив машинных слов, в котором для каждого потенциально возможного элемента отводится один бит. Присутствие элемента в множестве фиксируется значением 1 в этом бите, а отсутствие – значением 0.

На троичной машине можно эффективно обрабатывать такие множества, относительно вхождения элемента в которые предусматривается не два (да, нет), а три варианта ответа: (да, нет, неизвестно). Будем называть такие множества троичными. Для работы с ними каждому элементу, претендующему на вхождение в множество, будем отводить в памяти один трит.

Троичное множество можно образно представить некой фигурой на плоскости с обозначенной границей. Элементу, входящему в него, соответствует некая точка на плоскости внутри этой фигуры (а его трит = +1). Элементу, который не входит в него, соответствует некая точка на плоскости вне этой фигуры (а его трит = -1). Если про элемент неизвестно, входит ли он в это множество, то ему соответствует точка на границе этой фигуры (а его трит 0).

Для множества, количество элементов в котором будет не более 27, в ДССП-ТБМ достаточно объявить переменную (типа TWORD), занимающую в памяти одно 27-тритное ма-

шинное слово. Каждому  $j$ -му элементу множества будет сопоставлен  $j$ -ый трит такого слова ( $j=0,1,\dots,26$ ):

```
{ объявление множества A } TWORD VAR SA
{ объявление множества B } TWORD VAR SB
```

При заполнении множества требуется записывать в  $j$ -ый трит ответ на вопрос о вхождении  $j$ -ого элемента в это множество. Это можно сделать с помощью рассмотренной ранее операции записи (**T!I**) значения отдельного  $j$ -ого трита в троичное слово, помещённое в вершину стека.

Вопрос, входит ли  $j$ -ый элемент в множество (+1), не входит (-1) или про это ничего неизвестно (0), можно выяснить, применяя операцию чтения **T@I** значения отдельного  $j$ -ого трита троичного слова, предварительно записав значение множества в вершину стека.

Продemonстрируем, как можно первоначально сформировать значения множеств. Пусть в одномерных массивах (векторах)  $X[0:26]$  и  $Y[0:26]$  заданы значения координат некоторого набора точек плоскости:

```
{ вектора, задающие координат точек: }
26 VCTR X 26 VCTR Y
```

И пусть на плоскости представлены две фигуры. Фигура  $A$  – это прямоугольник высотой  $h=12$  и шириной  $g=16$ , левый нижний угол которого расположен в начале координат, а фигура  $B$  – это круг радиуса  $r=10$  с центром в начале координат.

Сформируем множества  $SA$  и  $SB$ , проверяя, какие из заданных точек попадают в фигуры  $A$  и  $B$ . Чтобы сформировать множество  $SA$ , проверим каждую точку  $(X_i, Y_i)$  ( $i=0,1,\dots,26$ ) на попадание в фигуру  $A$  и отметим этот факт в переменной  $SA$ , представляющей множество  $A$ :

```
{ для каждой j-ой точки j=0,...,26 выполним: }
SA j X j Y {Xj,Yj} 16 12 inFigA?
j T!I ! SA
```

Аналогично проверим каждую точку на попадание в фигуру  $B$  и отметим этот факт в переменной  $SB$ , представляющей множество  $B$ :

```
{ для каждой j-ой точки j=0,...,26 выполним: }
SB j X j Y {Xj,Yj} 10 inFigB?
j T!I ! SB
```

С троичными множествами можно выполнять такой же набор характерных операций, который обеспечивался для работы с обычными множествами на двоичной машине. Можно проверять принадлежность элемента множеству (операцией **T@I**), включать или исключать элементы из множества (**T!I**), а также получать из двух множеств новое множество путём их объединения (**TMAX**), пересечения (**TMIN**) или вычитания одного множества из другого (**NEG TMIN**).

Продemonстрируем выполнение таких операций над множествами на языке ДССП-Т:

```
{ проверка, есть ли j-й элемент в множестве A }
SA j T@I
{ проверка, есть ли j-й элемент в множестве B }
SB j T@I
{ включить j-ый элемент в множество A }
SA +1 j T!I ! SA
{ включить j-ый элемент в множество B }
SB +1 j T!I ! SB
{ исключить j-ый элемент из множества A }
SA -1 j T!I ! SA
{ исключить j-ый элемент из множества B }
SB -1 j T!I ! SB
{ объявление нового множества C } VAR SC
{ пересечение множеств: C=A∩B }
SA SB TMIN ! SC
{ объединение множеств: C=A∪B }
SA SB TMAX ! SC
{ вычитание множества B из A: C=A\B }
SA SB NEG TMIN ! SC
{ вычитание множества A из B: C=B\A }
SA NEG SB TMIN ! SC
```

## 5. Троичные управляющие конструкции

Благодаря возможности формулировать логическое условие с тремя возможными вариантами ответа на троичной машине можно применять новые виды ветвлений и циклов: троичное ветвление и цикл с троичным условием.

### 5.1. Троичное ветвление

Троичное ветвление означает, что для выбора дальнейшего хода исполнения программы предусматривается не два, а три возможных пути её продолжения. В ДССП для троичной машины (и в самой ТВМ) для организации такого ветвления предусмотрена команда **BRS**. Она позволяет выбрать на исполнение одну из трёх операций (процедур) в зависимости от знака значения, сформированного в вершине стека (см. блок-схему варианта  $A$  на рис. 1).

Следует упомянуть, что подобную команду **<BR P1,P2,P3>** было предложено реализовать ещё в ЦМ «Сетунь-70» [9]. Впоследствии она сразу же появилась в самой первой версии ДССП, реализованной для микрокомпьютера «Электроника-ИЦ03Д».

В языках программирования высокого уровня (для двоичных машин) оператор троичного ветвления предлагается крайне редко (хотя он был когда-то предусмотрен в языке Фортран), видимо, из-за его неэффективности. Ведь для его реализации на двоичной машине придётся использовать (в сгенерированном коде) две проверки и соответственно две команды двоичного ветвления (см. блок-схему варианта  $B$  на рис. 1). А при необходимости организации троичного ветвления в программе просто предла-

гается воспользоваться оператором двоичного ветвления 2 раза.

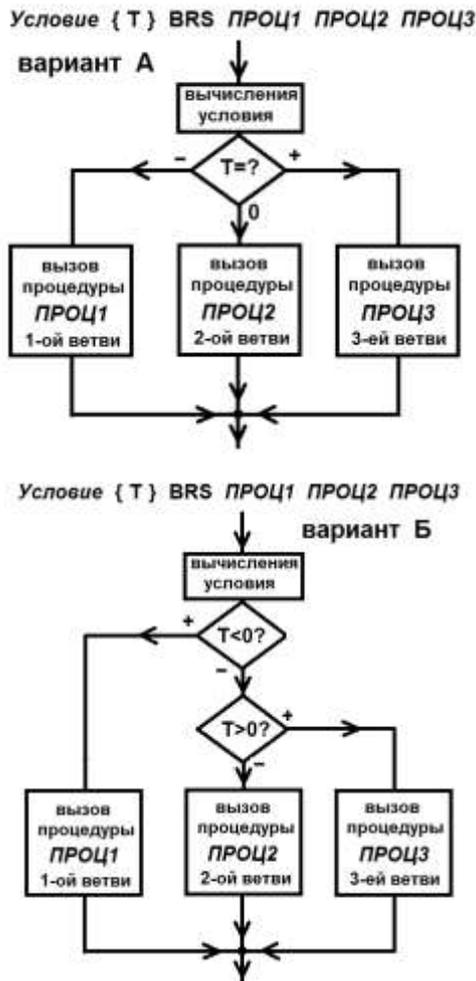


Рис. 1. Блок-схема реализации троичного ветвления на троичной (А) и на двоичной машине (Б).

Пожалуй, простейшим примером, демонстрирующим потребность троичного ветвления, является формула вычисления знака числа:

$$z = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ +1, & x > 0 \end{cases}$$

Сравним варианты её реализации для двоичной (на языке Си) и троичной машины (на языке ДССП-Т):

Си	<code>if (x&lt;0) z=-1; else if (x&gt;0) z=+1; else z=0;</code>
ДССП-Т	<code>: SGN{x} BRS -1 0 +1 {z};</code>

В качестве ещё одного примера продемонстрируем варианты реализаций (на языке ДССП-Т для троичной машины и на языке Си для двоичной машины) троичной операции сравнения двух чисел:

$$z = \begin{cases} -1, & x < y \\ 0, & x = y \\ +1, & x > y \end{cases}$$

Си	<code>if (x&lt;y) z=-1; else if (x&gt;y) z=+1; else z=0;</code>
ДССП-Т	<code>: CMP{x,y} - {x-y} BRS -1 0 +1 {z};</code>

Следует сказать, что слова **SGN** и **CMP** уже предусмотрены в базовом словаре ДССП. И в ассемблерном ядре ДССП их тела представлены вызовами соответствующих команд (**SIGN** и **CMP**) троичной машины. А здесь их определения (на языке ДССП-Т) приведены лишь для демонстрации применения команды **BRS**.

Сами же эти операции **SGN** и **CMP** могут, в свою очередь, применяться для формирования более сложного троичного логического выражения (условия), по результату которого организуется троичное ветвление командой **BRS**. Например, они могут пригодиться при определении слова, предназначенного для вычисления функции  $z=F(x,y)$ :

$$z = \begin{cases} x + y, & x < y \\ x, & x = y \\ x - y, & x > y \end{cases}$$

Си	<code>if (x&lt;y) z=x+y; else if (x&gt;y) z=x-y; else z=x;</code>
ДССП-Т	<code>: F{x,y} C2 C2 CMP {x,y,x?y} BRS + D - {z=x+y x x-y} ;</code>

Заметим, что при определении в ДССП слова **F**, как и при определении слов **CMP** и **SGN**, использовалась лишь одна команда троичного ветвления. А для двоичной машины (на языке Си) для реализации тех же действий приходится использовать две команды ветвления.

Выигрыш, обеспечиваемый применением команд троичного ветвления, становится ещё более ощутимым с возрастанием количества анализируемых вариантов продолжения программы. Рассмотрим пример функции, при вычислении которой потребуется проанализировать 9 вариантов:

$$F(x,y) = \begin{cases} \begin{cases} x^2 + y^2 + 1 & \text{при } y > 0, x > 0 \\ x^4 & \text{при } y = 0, x > 0 \\ x^2 + y^3 & \text{при } y < 0, x > 0 \end{cases} \\ \begin{cases} y^2 - 1 & \text{при } y > 0, x = 0 \\ 1 & \text{при } y = 0, x = 0 \\ y^3 + 1 & \text{при } y < 0, x = 0 \end{cases} \\ \begin{cases} x^4 + y^4 & \text{при } y > 0, x < 0 \\ x & \text{при } y = 0, x < 0 \\ x^3 + y^3 & \text{при } y < 0, x < 0 \end{cases} \end{cases}$$

Приведём пример её определения в ДССП для ТВМ с использованием команд троичного ветвления:

<code>: F{x,y} C2 BRS x- x0 x+ {z} ; : x-{x,y} C BRS x-y- x-y0 x-y+ {z}; : x-y-{x,y} ^3 E2 ^3 + {z=x^3+y^3}; : x-y0{x,y} D {z=x} ;</code>
---

```

: x-y{x,y} ^4 E2 ^4 + {z=x^4+y^4};
: x0{x,y} C BRS x0y- x0y0 x0y+ {z};
: x0y-{x,y} ^3 1+ E2D {z=y^3+1};
: x0y0{x,y} D T1 {z=1} ;
: x0y+{x,y} ^2 1- E2D {z=y^2-1};
: x+{x,y} C BRS x+y- x+y0 x+y+ {z};
: x+y-{x,y} ^3 E2 ^2 + {z=x^2+y^3};
: x+y0{x,y} D ^4 {z=x^4} ;
: x+y+{x,y} ^2 1+ E2 ^2 +
      {z=x^2+y^2+1};
: ^2 {a} C * {a^2} ;
: ^3 {a} C C * * {a^3} ;
: ^4 {a} ^2 ^2 {a^4} ;

```

И сравним его с вариантом реализации этой функции на языке Си для двоичной машины:

```

int FC(int x, int y) {
int z;
if (x>0) {
  if (y>0) z = x*x + y*y + 1; else
  if (y==0) z = x*x*x*x; else
  /*if(y<0)*/ z = x*x + y*y*y;
} else if (x==0) {
  if (y>0) z = y*y - 1; else
  if (y==0) z = 1; else
  /*if(y<0)*/ z = y*y*y + 1;
} else { /*if(x<0)*/
  if (y>0) z=x*x*x*x+y*y*y*y-1; else
  if (y==0) z = x; else
  /*if(y<0)*/ z = x*x*x + y*y*y ;
}
return(z);
} //F

```

В слове **F** в ДССП-ТВМ будет всегда исполнено 2 команды троичного ветвления. А Си-функции **FC** может потребоваться исполнить от 2-х до 4-х команд двоичного ветвления.

Наконец, заметим, что при любом сколь угодно большом количестве вариантов **P** продолжения программы анализ, по какому варианту её следует продолжать, будет производиться быстрее с помощью команд троичного ветвления, т.к. таких команд потребуется выполнить значительно меньше. Ведь количество требуемых троичных команд ветвления определяется величиной  $\log_3 P$  (округлённой до ближайшего большего целого), а количество двоичных команд ветвления исчисляется величиной  $\log_2 P$ , а для любого целого  $P > 1$  всегда справедливо соотношение:  $\log_3 P < \log_2 P$ .

## 5.2. Цикл с троичным условием

Представляемый здесь цикл с троичным условием является принципиально новым видом цикла. Такой вид цикла совмещает в себе проверку, требуется ли прекращать его повторение, с выбором одного из двух действий, предусмотренных для исполнения в качестве тела цикла на очередном его шаге.

Такой вид цикл не встречался ранее в системе команд какой-либо вычислительной машины (ни двоичной, ни троичной). Не было такой

команды цикла и на троичной ЦМ «Сетунь-70», в системе команд которой впервые (в 1975 г.) появились управляющие команды, эквивалентные основным конструкциям структурированного программирования.

Вместе с командой обычного вызова процедуры **<SR P>** в системе команд ЦМ «Сетунь-70» появились команда **<BR P1,P2,P3>** для условного вызова одной из процедур и команда **<DW P>** повторяемого вызова процедуры, действующая как цикл с предусловием (**while do**), которые и обеспечили возможность структурированного программирования на уровне машинного кода. Но команда цикла **DW** в ЦМ «Сетунь-70» анализировала вершину стека как двоичное условие (равно нулю или нет) и вызывала в качестве тела цикла всегда одну и ту же процедуру. А предлагаемая команда нового вида цикла анализирует вершину стека уже как троичное условие.

В широко распространённых языках программирования цикл с троичным условием тоже не встречался ранее. Можно сказать, что практическую полезность такого вида цикла неявно отметил когда-то Эдсгер Дейкстра. В своей знаменитой книге «Дисциплина программирования» [10] он предложил конструкцию цикла с множественными условиями и телами, применяя которую продемонстрировал [10, с.70], в частности, как можно упростить известный алгоритма Евклида для вычисления наибольшего общего делителя (НОД) двух натуральных чисел:

НОД(X,Y) на основе цикла с двоичным условием и ветвлением
<pre> x, y := X, Y; do x≠y → if x&gt;y → x := x-y            ■ y&gt;x → y := y-x            fi od печатать (x) </pre>

НОД(X,Y) на основе цикла с двумя телами
<pre> x, y := X, Y; do x&gt;y → x := x-y    ■ y&gt;x → y := y-x od печатать (x) </pre>

Цикл с троичным условием впервые был предложен в ДССП для троичной машины (ТВМ). И был сначала реализован в ней, как новое слово **DW+**, определённое сначала на языке ДССП-Т, а также в коде ассемблера [11]. А затем уже (после его апробации в ДССП-ТВМ) было предложено обеспечить функционирование такого цикла соответствующей командой (**DWMP**) самой троичной виртуальной машины.

Объясним подробно правила функционирования (семантику) цикла с троичным условием и познакомимся с соответствующей ему командой **DW++** языка ДССП-Т (см. блок-схему варианта А на рис. 2). Эта команда проверяет, надо ли завершать цикл, и при этом тут же выбирает одну из двух процедур в качестве тела цикла на случай его продолжения. Такая совмещённая троичная проверка осуществляется в зависимости от знака значения, сформированного в вершине стека. Заметим, что после проверки значения вершины стека само это значение из стека удаляется перед вызовом одной из предусмотренных процедур (или при завершении цикла).

Для обеспечения реализации такой конструкции цикла на двоичной машине потребовалось бы использовать (в сгенерированном коде) две проверки (одну на прекращение цикла, а другую для выбора одного из тел) и соответственно две команды двоичного ветвления (см. блок-схему варианта Б на рис. 2). Поэтому никакого выигрыша в производительности такой вид цикла на двоичной машине не сможет обеспечить. (Правда, такой вид цикла мог бы упростить хотя бы запись алгоритма, освободив его от лишнего ветвления).

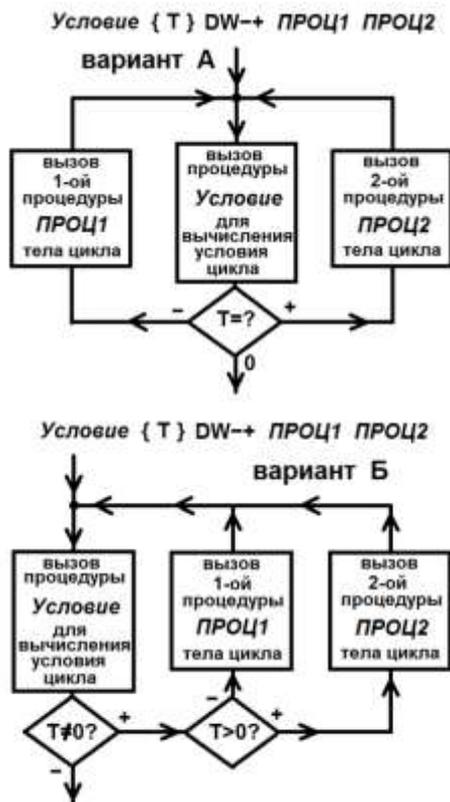


Рис. 2. Блок-схема реализации цикла с троичным условием на троичной (А) и на двоичной машине (Б)

А в троичной машине оценка знака верши-

ны стека вместе с принятием решения о прекращении цикла или его продолжении с выбором соответствующей процедуры на исполнение может выполняться за одну команду. Поэтому на троичной машине такой вид цикла может повысить производительность исполняемой программы.

Продемонстрируем, как применение команды цикла **DW++** с троичным условием в программе на языке ДССП-Т позволяет не только упростить её алгоритм, но и ускорить её исполнение. В качестве примера рассмотрим задачу вычисления наибольшего общего делителя двух натуральных чисел НОД(Х,У) алгоритмом Евклида. Сначала приведём вариант её решения, в котором применяется команда цикла **DW** с двоичным условием и команда **BR-** условного выбора одной из двух процедур:

```
: NOD1 {x,y} x-y? DW y-x|x-y D ;
: y-x|x-y{x,y} x-y? BR- y-x x-y ;
: x-y?{x,y} C2 C2 - ;
: y-x{x,y} C2 - {x,y-x};
: x-y{x,y} E2 C2 - E2 {x-y,y};
```

А вот вариант решения той же задачи с применением команды троичного цикла:

```
: NOD2 x?y DW++ y-x x-y D ;
: x?y{x,y} C2 C2 CMP ;
: y-x{x,y} C2 - {x,y-x} ;
: x-y{x,y} E2 C2 - E2 {x-y,y};
```

Теперь сравним процедуры NOD1 и NOD2. Прежде всего, отметим, что запись алгоритма процедуры NOD2 выглядит заметно проще: в нём стало на одну строку меньше, так как отпала необходимость определения слова  $y-x|x-y$ , задающего общее тело цикла. Далее сравним их по производительности.

Таблица 1. Сравнение производительностей алгоритмов NOD1(X,Y) и NOD2(X,Y)

аргументы		результат Z=	Количество команд		KB
X	Y	NOD(X,Y)	NOD1	NOD2	
1019	123	1	590	358	1.65
1020	3560	20	544	312	1.74
2015	135	5	574	350	1.64
39366	99	9	8054	4838	1.66
5000	250	250	398	246	1.62
7777	49	1	3254	1958	1.66

KB – коэффициент выигрыша, который обеспечивает алгоритм NOD2 для троичной машины по отношению к алгоритму NOD1 для двоичной машины

Процедуры вычисления НОД обоих этих вариантов (NOD1 и NOD2) многократно запускались на программном имитаторе троичной машины (ТБМ) с различными аргументами. Для оценки их производительности измерялись по-

казания счётчика исполненных команд ТВМ до и после исполнения каждой процедуры. Результаты проведённых испытаний представлены в таблице 1.

Как видно из этой таблицы, процедура **NOB2**, применяющая цикл с троичным условием, работает значительно быстрее (более, чем в полтора раза), так как исполняется за меньшее число команд ТВМ.

Таким образом, реализация подобных процедур на троичной машине с применением троичного цикла может принести существенный выигрыш в производительности.

## 6. Способы реализации функций троичной логики

Да, троичный компьютер открывает разработчикам мир новых возможностей. Но при создании программного и аппаратного обеспечения, задающего алгоритмы поведения троичной машины, разработчикам требуется также решать и новые задачи. Причём, некоторые из известных задач, решённые когда-то для двоичных машин, для троичной машины приходится решать почти что заново. Одной из таких является задача построения формулы вычисления функции троичной логики по заданной для неё таблице значений.

В двоичной логике (булевой алгебре) широко известны как минимум два конструктивных метода получения формулы вычисления функции по заданной для неё таблице истинности: СДНФ (совершенная дизъюнктивная нормальная форма) и СКНФ (совершенная конъюнктивная нормальная форма). Для получения формулы функции можно также применять универсальный метод так называемых «полиномов Жегалкина», который пригоден для любой  $k$ -значной логики. Но он неудобен тем, что для каждой новой функции вынуждает снова решать систему линейных уравнений, чтобы определить коэффициенты того конкретного полинома, который нужно получить для вычисления заданной функции.

Столкнувшись с необходимостью получения формул для некоторых функций троичной логики, автор сначала попытался отыскать (на просторах Интернета и в доступной ему печатной литературе) некоторые аналоги методов СДНФ и СКНФ применительно к троичной симметричной логике (+1,0,-1). А своевременно не найдя их, сам «изобрёл» подобные им методы получения требуемой формулы, объяснению которых и посвящён данный раздел.

Поясним суть одного из возможных предлагаемых методов, ограничив задачу получением формулы вычисления лишь для двуместной

троичной функции. (Заметим, что в троичной логике количество различных функций от двух переменных исчисляется величиной  $3^9=19683$ , в то время как в двоичной логике таких двуместных функций всего  $2^4=16$ ).

Итак, пусть троичная функция  $F(x,y)$  от двух переменных задана таблицей своих значений TF на всевозможных наборах пар  $(x_i, y_j)$ :

$$TF = \{ f_{ij} = F(x_i, y_j), i = -1, 0, +1; j = -1, 0, +1 \}.$$

Формулу вычисления этой функции представим в виде троичной суммы, каждый из слагаемых которой представляется характеристической функцией  $P_k(x,y)$ , принимающей значение 1 только на одном выделенном наборе  $(x_{1k}, y_{1k})$  и 0 на остальных, или функцией  $N_m(x,y)$ , принимающей значение -1 только на одном выделенном наборе  $(x_{1m}, y_{1m})$  и 0 на остальных:

$$\sum_k^t P_k(x,y) +^t \sum_k^t N_m(x,y)$$

В изображении этой формулы используются следующие обозначения:  $+^t$  – это операция троичного сложения, а  $\sum_k^t$  – это операция троичного сложения всех операндов, образуемых перебором всевозможных значений  $k$ .

В правую часть этой суммы включаются функции вида  $P_k(x,y)$  для тех наборов, на которых заданная функция  $F$  принимает значения +1, а в левую часть включаются функции вида  $N_m(x,y)$  для тех наборов, где заданная функция  $F$  принимает значения -1.

А формулы самих характеристических функций  $P_k(x,y)$  и  $N_m(x,y)$  представим в виде троичного произведения двух одноместных характеристических функций:

$$P_k(x,y) = N_a(x) \times^t N_b(y) \text{ (где } a=I_k, b=J_k),$$

$N_m(x,y) = \sim^t N_a(x) \times^t N_b(y)$  (где  $a=I_m, b=J_m$ ), используя операции троичного умножения ( $\times^t$ ) и троичной инверсии ( $\sim^t$ ).

Каждая из трёх одноместных характеристических функций ( $N_{+1}, N_0, N_{-1}$ ) призвана принимать значение +1 на выделенном входном значении и 0 на остальных. Для их вычисления можно предложить следующие формулы:

$$N_{+1}(x) = \sim^t x \times^t (x +^t 1), N_0(x) = (x +^t 1) \times^t (1 -^t x),$$

$$N_{-1}(x) = x \times^t (1 -^t x),$$

учитывая, что операцию троичного вычитания  $-^t$  можно реализовать с использованием операций троичного сложения и инверсии (смены знака):  $a -^t b = a +^t (\sim^t b)$ .

В результате получим формулу, вычисляющую значение троичной функции в виде многочлена на основе троичных операций сложения  $+^t$ , умножения  $\times^t$ , и инверсии  $\sim^t$  (смены знака). Полученную таким способом формулу далее можно упростить, используя правила эквивалентных преобразований для троичной симметричной логики.

Продemonстрируем описанный метод на примере получения формулы вычисления двухместной троичной функции, заданной следующей таблицей значений (см. таблицу 2). На просторах Интернета она упоминается под названием «функция следования Брусенцова» (см., например, <http://phg.su/basis2/X19.HTM>).

Таблица 2. Таблица значений двухместной троичной функции следования Брусенцова.

аргументы	X	-1	-1	-1	0	0	0	+1	+1	+1
	Y	-1	0	+1	-1	0	+1	-1	0	+1
результат	Z	+1	0	-1	0	0	0	0	0	+1

Соберём требуемую формулу поэтапно. В левой части троичной суммы поставим в качестве слагаемых двуместные характеристические функции, соответствующие наборам аргументов  $(-1, -1)$  и  $(+1, +1)$ , на которых функция принимает значение  $+1$ . В правой части суммы поставим характеристическую функцию, соответствующую набору  $(-1, +1)$ , на котором функция принимает значение  $-1$ . А для наборов, на которых функция принимает значение  $0$ , в итоговую сумму никаких слагаемых не ставим вообще. И представим каждое их 3-х слагаемых в виде произведения двух одноместных характеристических функций.

В результате для нашей функции получим следующую первоначальную формулу:

$$\mathbf{tSLBr1}(x,y) = (\mathbf{H}_{-1}(x) \times^t \mathbf{H}_{-1}(y) +^t +^t \mathbf{H}_{+1}(x) \times^t \mathbf{H}_{+1}(y)) +^t (-^t \mathbf{H}_{-1}(x) \times^t \mathbf{H}_{+1}(y))$$

По ней уже можно вычислять требуемую функцию:

```

: tSLBr1 {x, y} {1-й вариант}
C2 tH- C2 tH- TMUL {x,y,x(x-1)*y(y-1)}
C3 tH+ C3 tH+ TMUL + E3
{x(x-1)*y(y-1) + x(x-1)*y(y-1), y, x }
tH- E2 tH+ TMUL tSUB ;
: tH- {x} { x*(1-x) = if x=-1 then +1 else 0 }
C Mask--- TADD TMUL NEG ;
: tH+ {x} { -x*(x+1) = if x=1 then +1 else 0 }
Mask+++ C2 TADD TMUL NEG;
: tSUB {x, y} NEG TADD {x-y} ;

```

используя определения слов **tH-** и **tH+** для вспомогательных функций  $\mathbf{H}_{-1}$  и  $\mathbf{H}_{+1}$ . Но этот вариант её вычисления не будет эффективным.

В первоначально полученную формулу подставим формулы для вычислений  $\mathbf{H}_{-1}$  и  $\mathbf{H}_{+1}$  и выполним эквивалентные преобразования:

$$\begin{aligned} & x \times^t (1 - {}^t x) \times^t y \times^t (1 - {}^t y) +^t \sim^t x \times^t (x + {}^t 1) \\ & \times^t \sim^t y \times^t (y + {}^t 1) -^t x \times^t (1 - {}^t x) \times^t \sim^t y \times^t (y + {}^t 1) \approx \\ & \approx x \times^t y \times^t ((x - {}^t 1) \times^t (y - {}^t 1) +^t (x + {}^t 1) \times^t \\ & (y + {}^t 1) -^t (x - {}^t 1) \times^t (y + {}^t 1)) \approx \\ & \approx x \times^t y \times^t ((x - {}^t 1) \times^t ((y - {}^t 1) -^t (y + {}^t 1)) \\ & +^t (x + {}^t 1) \times^t (y + {}^t 1)) \approx \\ & \approx x \times^t y \times^t ((x - {}^t 1) \times^t ((-1 -^t -1) \\ & +^t x \times^t y +^t x +^t y + {}^t 1)) \approx \\ & \approx x \times^t y \times^t (x - {}^t 1 +^t x \times^t y +^t x +^t y + {}^t 1) \approx \end{aligned}$$

$$\approx x \times^t y \times^t (x \times^t y -^t x +^t y) \\ (\text{т.к. } -1 - {}^t 1 = +1 \text{ и } x + {}^t x = \sim^t x)$$

Запишем итоговую полученную формулу:

$$\mathbf{tSLBr2}(x,y) = x \times^t y \times^t (x \times^t y -^t x +^t y)$$

И определим на языке ДССП-Г слово, которое будет вычислять троичную функцию следования Брусенцова по этой формуле:

```

: tSLBr2 {2-й вариант}
{x, y} C2 C2 TMUL C E4
{x*y, y, x*y, x} NEG TADD
{x*y, y, x*y-x} TADD TMUL
{x*y*(y+x*y-x)} ;

```

Раскрыв скобки и обозначив  $x \times^t x = |x|$ , можем получить ещё один вариант формулы вычисления функции следования Брусенцова:

$$\mathbf{tSLBr3}(x,y) = |x \times^t y| -^t |x| \times^t y +^t x \times^t |y|$$

Но этот 3-й вариант будет уступать предыдущему (2-му) по количеству команд:

```

: tSLBr3 {3-й вариант}
C2 C2 TMUL tABS {x, y, |x*y|}
C3 tABS C3 TMUL tSUB E3
{|x*y|-|x|*y,y,x} E2 tABS TMUL TADD
{|xy|-|x|*y+x*|y|} ;
: tABS {x} C TMUL { |x|= x*x } ;

```

Так что самым оптимальным (по числу команд) будет именно 2-ой вариант вычисления функции, определённый словом **tSLBr2**.

Описанный способ получения формулы вычисления троичной функции по заданной таблице её значений на всевозможных наборах аргументов можно распространить и на функции от многих переменных. Например, для случая трёх переменных надо будет формулы функций  $\mathbf{Pk}(x,y,z)$  и  $\mathbf{Nm}(x,y,z)$  представлять в виде произведения не двух, а уже трёх одноместных характеристических функций:

$$\mathbf{Pk}(x,y,z) = \mathbf{H}_a(x) \times^t \mathbf{H}_b(y) \times^t \mathbf{H}_c(z), \text{ где } a=I_k, b=J_k, c=L_k \\ \mathbf{Nm}(x,y,z) = \sim^t \mathbf{H}_a(x) \times^t \mathbf{H}_b(y) \times^t \mathbf{H}_c(z), a=I_m, b=J_m, c=J_m$$

## 7. Заключение

В статье охарактеризованы специфические особенности программирования, которые приходится учитывать при создании программного оснащения для троичного компьютера. На примерах ряда характерных программ, разработанных в ДССП для троичной виртуальной машины, продемонстрированы новые возможности, которые может предоставить программисту троичная цифровая машина. А также отмечено, что на определённом классе задач троичный компьютер по ряду параметров может превосходить двоичный. И это даёт основание полагать, что с созданием аппаратной реализации троичной цифровой машины человечество обретёт более совершенный инструмент обработки информации.

# Features of programming of ternary computer: new capabilities and new tasks

A.A.Burtsev

**Abstract.** The article considers new expressive facilities of constructing programs, which can be provided to the programmer by the ternary computer and which are provided now by the programming system DSSP for the ternary virtual machine (TVM). It also demonstrates a method of constructing a formula for calculating a ternary logic function according to a given table of values with its subsequent implementation in DSSP-TVM.

**Keywords:** ternary computer, ternary arithmetics, ternary logic, ternary function, "Setun", "Setun-70", ternary virtual machine (TVM), DSSP, DSSP-TVM.

## Литература

1. Д. Кнут. Искусство программирования на ЭВМ. М., Мир, 1978, Т.2, с. 216-219.
2. Н.П. Брусенцов, Х. Рамиль Альварес. Троичные ЭВМ "Сетунь" и "Сетунь 70". «Первая Международная конференция "Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы" SORUCOM-2006». Россия, г. Петрозаводск, 3-7 июля 2006 г. (труды конференции в 2-х ч.). Изд-во ПетрГУ, 2006. ч.1, 45-51.
3. С.А. Сидоров, Ю.С. Владимирова. Троичная виртуальная машина. «Программные системы и инструменты. Тематический сборник», №12 (2011), 46-55.
4. А.А. Бурцев, Х. Рамиль Альварес. Кросс-система разработки программ на языке ДССП для троичной виртуальной машины. «Программные системы и инструменты. Тематический сборник», №12 (2011), 183-193.
5. С.П. Маслов. Троичная схемотехника. «Программные системы и инструменты. Тематический сборник», №13 (2012), 152-158.
6. А. А. Бурцев, С. А. Сидоров. Троичная виртуальная машина и троичная ДССП. «Программные системы: теория и приложения», Т.6 (2015), №4, 29–97, [http://psta.psisras.ru/read/psta2015\\_4\\_29-97.pdf](http://psta.psisras.ru/read/psta2015_4_29-97.pdf).
7. А.А. Бурцев, С.А. Сидоров. Программный комплекс ДССП-ТВМ для структурированного программирования троичной [виртуальной] машины. «Современные информационные технологии и ИТ-образование», Т. 2 (2015), № 11, 371–379.
8. Ю.С. Владимирова. Введение в троичную информатику: учебное пособие. М., АРГАМАК-МЕДИА, 2015.
9. Н.П. Брусенцов, Х. Рамиль Альварес. Структурированное программирование на малой цифровой машине. «Вычислительная техника и вопросы кибернетики», Вып. 15. М., Изд-во МГУ, 1978, 3-8.
10. Э. Дейкстра. Дисциплина программирования. М., Мир, 1978.
11. А.А. Бурцев. Разработка собственных управляющих конструкций в среде ДССП для троичной машины. «Труды НИИСИ РАН», Т. 8 (2018), № 2, 52–64.

# О преимуществах троичных машин и эффективности троичных вычислений

А.А. Бурцев<sup>1</sup>, В.А. Бурцев<sup>2</sup>

<sup>1</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия, [burtsev@niisi.msk.ru](mailto:burtsev@niisi.msk.ru)

<sup>2</sup> Калужский филиал МГТУ имени Н.Э. Баумана, Калуга, Россия, [basel99@mail.ru](mailto:basel99@mail.ru)

**Аннотация.** В статье рассматриваются преимущества троичных машин и демонстрируется эффективность троичных вычислений на примерах разработанных авторами программ решения ряда учебных задач в диалоговой системе структурированного программирования (ДССП), созданной в МГУ им. М.В. Ломоносова для троичной виртуальной машины (ТВМ).

**Ключевые слова:** троичная симметричная система счисления, троичная виртуальная машина, ДССП, троичная логика, троичное ветвление, троичный цикл.

## 1. Введение

Общепризнано, что в настоящее время двоичная микроэлектроника приблизилась к пределу своих технологических возможностей. Поэтому возрождается интерес к исследованиям путей построения компьютерной техники на основе других систем счисления.

Одной из наиболее перспективных считается троичная симметричная система счисления (с цифрами  $-1, 0, +1$ ). На заре развития вычислительной техники её безусловные преимущества отмечал ещё Дональд Кнут [1], предвещая ей успешное применение в будущем.

Более полувека назад эффективность вычислительной техники, построенной на основе такой системы счисления, была убедительно подтверждена практической разработкой и применением троичных цифровых машин (ЦМ) «Сетунь» и «Сетунь-70» [2], разработанных в НИЛ ЭВМ МГУ под руководством Брусенцова Н. П. О преимуществах троичных машин Брусенцовым была написана серия статей под общим названием «Заметки о троичной цифровой технике» [3]. А относительно недавно в преимуществах троичной вычислительной техники убедились и разработчики квантовых компьютеров, заявивших, что “будущее квантовых компьютеров — в троичных вычислениях” [4].

В настоящее время в качестве среды разработки и прогона учебных программ можно использовать программный комплекс ДССП-ТВМ [5], созданный в НИЛ ТИ на факультете ВМК МГУ. Он включает имитатор троичной машины (ТВМ) и систему разработки программ для нее на языке ДССП-Т – троичном варианте языка ДССП.

Далее в статье перечисляются преимущества

троичных машин, основанных на троичной симметричной системе счисления, и демонстрируется эффективность троичных вычислений на примерах разработанных авторами программ решения ряда учебных задач в ДССП-ТВМ.

## 2. Преимущества троичных компьютеров

Кратко перечислим основные преимущества троичных компьютеров (более подробно они объясняются в [3] и [6]).

1. Среди целочисленных позиционных систем счисления наиболее экономичной является троичная. Доказательство этого утверждения можно найти во многих источниках, например, в [6] (см. с.29), поэтому приводить его здесь не будем, а дадим лишь некоторые пояснения.

Понятие экономичности системы счисления непосредственно связано с объемом оборудования, которое требуется заготовить для работы с машинными словами как с числами. Представим себе аппаратуру как магнитную азбуку (кассу), из знаков которой нам требуется составлять всякое  $p$ -ичное число из  $n$  разрядов. Сколько литер (знаков)  $z$  необходимо заготовить в такой кассе? Очевидно,  $z = n * p$  знаков (по  $p$  знаков на каждую позицию). А какой диапазон чисел  $d$  можно будет при этом представить? Ясно, что  $d = p^n$ . В предположении, что  $z$  делится на  $p$ , можно выразить  $d$  формулой  $p^{z/p}$ .

Вот такой величиной  $d$  и оценивается экономичность системы счисления по основанию  $p$ . Она определяет количество чисел, которое можно представить в этой системе счисления  $p$  при помощи заданного числа знаков  $z$ . Наиболее экономичной считается система счисления,

у которой эта величина  $d=p^{(z/p)}$  максимальна.

Оценим величину экономичности систем счисления по основанию 2, 3, 4, 5, 10 для 60 знаков и представим результаты в таблице 1.

Таблица 1. Оценка экономичности систем счисления по основанию  $p=2, 3, 4, 5, 10$  для  $z=60$  знаков.

$p$	число разрядов $n$	диапазон чисел $p^n$
2	30	$2^{30} = 1073741824$
3	20	$3^{20} = 3486784401$
4	15	$4^{15} = 1073741824$
5	12	$5^{12} = 244140625$
10	6	$10^6 = 1000000$

Как видно из этой таблицы, наибольшая величина экономичности (среди всех рассмотренных) оказывается у троичной системы счисления. Заметим также, что у 2-ной и 4-ной систем счисления эта величина одинаковая.

При проведении строгого доказательства рассматривают функцию  $F(x) = x^{(z/x)}$  на диапазоне вещественных чисел, и определяют, при каком  $x$  она принимает максимальное значение. Для поиска экстремума приравнивают её производную нулю и получают в качестве ответа число Эйлера  $e=2,71828\dots$ . А среди целых чисел к ней ближе всего число 3. Вот поэтому троичная система оказывается самой экономичной системой счисления среди всех целочисленных.

2. В троичном машинном слове из  $n$  разрядов можно закодировать значительно больший диапазон значений:  $3^n$ , а в двоичном – только  $2^n$ , т.е. троичный код имеет выигрыш в  $(1,5)^n$  у двоичного, что демонстрируется таблицей 2:

Таблица 2. Диапазоны значений для  $3^n$  и  $2^n$

$n$	$2^n$	$3^n$	$3^n / 2^n$
5	32	243	$\approx 7,6$
8	256	6561	$\approx 25,6$
16	65536	43046721	$\approx 656,8$
27	134217728	7625597484987	$\approx 56815,1$
32	4294967296	1853020188851841	$\approx 431439,9$

Из приведённой таблицы, в частности, можно увидеть, что 27-тритным словом можно перекрыть диапазон 32-битного слова уже примерно в 1775 раз ( $3^{27}/2^{32}$ ).

3. Троичная симметричная система является минимальной, в которой можно единообразно представлять положительные и отрицательные числа (без каких-либо «ухищрений» в форме дополнительного кода).

4. В троичной симметричной системе счисления упрощаются реализации операции сдвигов и сравнения чисел. И не требуется различать арифметические и логические команды сдвигов (благодаря единообразному представлению положительных и отрицательных чисел).

5. А также естественно выражаются резуль-

таты трехзначных операций сравнения чисел ( $<, =, >$ ) и функции «знак числа» ( $- , 0, +$ ).

6. Можно непосредственно воплощать троичную логику, т.е. вычислять логические выражения с тремя значениями:  $+1$  («да»),  $-1$  («нет»),  $0$  («не знаю» или «может быть»).

7. Для округления числа достаточно просто отсечь младшие незначимые разряды, т.к. отсекаемый остаток не будет превышать (по абсолютной величине) половины значения последнего значимого разряда.

В этих преимуществах можно наглядно убедиться, выполняя троичные вычисления, т.е. вычисления на троичных машинах.

### 3. Примеры эффективности троичных вычислений

Прежде всего, заметим, что троичная машина может исполнить всё то, что может двоичная. Но при этом она может предоставить для построения компьютерной программы гораздо более широкие возможности. В их числе следует отметить возможность эффективной реализации троичного ветвления, цикла с троичным условием, операций троичной арифметики и троичной логики.

Эффективность троичных вычислений продемонстрируем на примерах разработки в ДССП-ТВМ программ для решения ряда учебных задач в двух вариантах: 1) для троичной машины и 2) для двоичной, т.е. с использованием только тех операций, которые можно исполнить на двоичной машине.

#### 3.1. Пример №1

Рассмотрим задачу попадания точки в определенную область (заштрихованную фигуру) на плоскости (см. рис. 1) с трёхзначным вариантом ответа: точка попадает внутрь области ( $+1$ ), на границу области ( $0$ ), вне области ( $-1$ ).

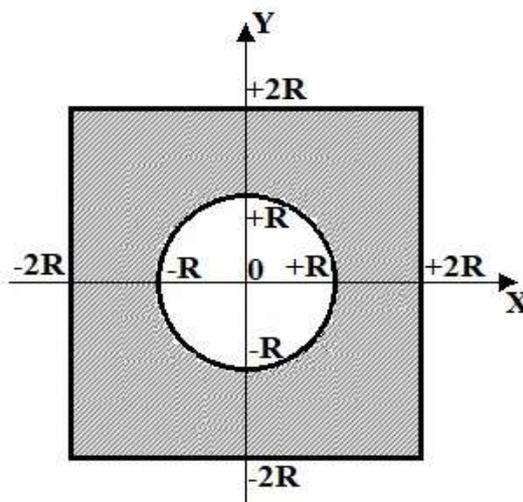


Рис. 1. Заданная фигура на плоскости

Для решения этой задачи на троичной машине можно просто составить троичное логическое выражение:

```
inSqr(x,y,R) = (sign(2R-|x|)) &' (sign(2R-|y|))
inCircle(x,y,R) = sign(R*R-x*x-y*y)
inFigt(x,y,R) = inSqr(x,y,R)&'~inCircle(x,y,R)
```

используя троичные логические операции:

```
sign(x) – определение знака числа x
(SGN – на языке ДССП-Т)
&' – погритная конъюнкция
(TMIN – на языке ДССП-Т)
~ – отрицание
(NEG – на языке ДССП-Т)
```

Для вычисления этого выражения составим программу на языке ДССП-Т:

```
: inSquare?t {x,y,R}
*2 E3 ABS E2 ABS C3 E2
CMP E3 E2 CMP TMIN
{(y|<?>2R)&t(x|<?>2R)} ;
: inCircle?t {x,y,R}
2^ E3 2^ E2 2^ +
CMP {R^2<?>x^2+y^2} ;
: inFig?t {x,y,R}
C3 C3 C3 inSquare?t
E4 E3 E2 inCircle?t NEG TMIN
{(y|<?>2R)&t(x|<?>2R)&t~(R^2<?>x^2+y^2)} ;
```

А для двоичной машины решение не получится сформулировать в виде логического выражения, но можно предложить вычислить нужный результат с помощью арифметических операций с применением функций  $\min(x,y)$  и  $\text{sign}(z)$ :

```
inSqr(x,y,R) = min(sign(2R-|x|),sign(2R-|y|))
inCircle(x,y,R) = sign(R*R-x*x-y*y)
inFig(x,y,R) = min(inSqr(x,y,R),-inCircle(x,y,R))
```

и составить соответствующую программу на языке ДССП-Т:

```
: sign{x} C {x,x} BR- -1 sign01 {z};
: sign01{x} BR0 0 1 {z} ;
: min{a,b} C2 C2 < BR+ D E2D {min(a,b)};
: inSquare{x,y,R}
*2 E3 ABS E2 ABS C3 E2 - sign
E3 E2 - sign min
{min(sign(2*R-|x|),sign(2*R-|y|))} ;
: inCircle{x,y,R}
2^ E3 2^ E2 2^ + - sign
{sign(R^2-x^2+y^2)} ;
: inFig{x,y,R}
C3 C3 C3 inSquare E4 E3 E2
inCircle NEG min
{min(inSquare(x,y,R),-inCircle(x,y,R))} ;
```

Поскольку для реализации функций  $\text{sign}$  и  $\min$  приходится использовать команды ветвления, решение этой задачи с применением операций, присущих двоичной машине, потребует исполнения большего числа команд, т.е. будет менее производительным.

## 3.2. Пример №2

Рассмотрим задачу вычисления значения многовариантной функции, заданной следующей формулой:

$$F(x,y) = \begin{cases} x + y + 1 & \text{при } y > 0, x > 0 \\ x^3 & \text{при } y = 0, x > 0 \\ x + y^2 - 1 & \text{при } y < 0, x > 0 \\ y^3 & \text{при } y > 0, x = 0 \\ 100 & \text{при } y = 0, x = 0 \\ y^2 & \text{при } y < 0, x = 0 \\ y + x^2 + 1 & \text{при } y > 0, x < 0 \\ x^2 & \text{при } y = 0, x < 0 \\ y^2 + x^2 - 1 & \text{при } y < 0, x < 0 \end{cases}$$

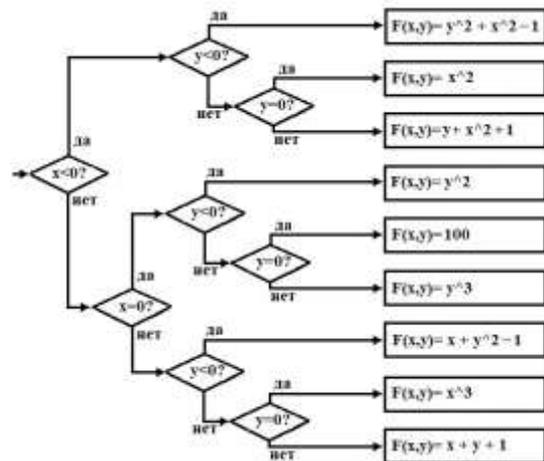


Рис. 2. Блок-схема с двоичными ветвлениями.

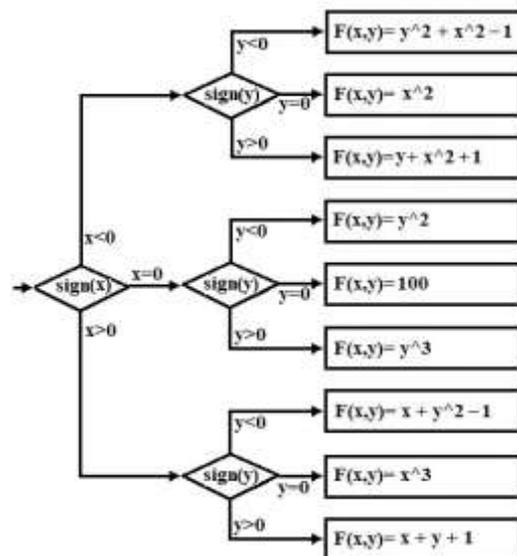


Рис. 3. Блок-схема с троичными ветвлениями.

Представим решение этой задачи в виде блок-схем программ. Для двоичной машины мы можем применять только команду двоичного ветвления (см. рис. 2). А в варианте для троичной машины можно применить команду троич-

ного ветвления (см. рис.3).

Процедуру вычисления функции с применением троичного ветвления можно оформить на языке ДССП-Т:

```

: MVF{x,y} C2 BRS x< x0 x> {z} ;
: x<{x,y} C BRS x<y< x<y0 x<y> {z};
: x<y<{x,y} ^2 E2 ^2 + 1-
      {z=y^2+x^2-1};
: x<y0{x,y} D ^2 {z=x^2};
: x<y>{x,y} E2 ^2 + 1+ {z=y+x^2+1};
: x0{x,y} C BRS x0y< x0y0 x0y> {z};
: x0y<{x,y} ^2 E2D {z=y^2};
: x0y0{x,y} DD 100 {z=100};
: x0y>{x,y} ^3 E2D {z=y^3};
: x>{x,y} C BRS x>y< x>y0 x>y> {z};
: x>y<{x,y} ^2 + 1- {z=x+y^2-1};
: x>y0{x,y} D ^3 {z=x^3};
: x>y>{x,y} + 1+ {z=x+y+1};
: ^2 {x} C * {x^2};
: ^3 {x} C C * * {x^3};

```

Сравнивая эти две блок-схемы, изображенные в виде деревьев, можно легко увидеть, что на рис. 2 представлено 4-х ярусное дерево, а на рис. 3 – 2-х ярусное дерево. Это означает, что при вычислении данной функции на троичной машине (согласно рис. 3) будет всегда исполняться 2 троичных ветвления, а в версии для двоичной машины (по рис. 2) может потребоваться исполнить операцию ветвления 2, 3 или даже 4 раза. Значит, на троичной машине вычисление этой функции в большинстве случаев будет исполнено за меньшее число команд, т.е. будет более производительным.

### 3.3. Пример №3

Решение следующей задачи для троичной машины основывается на применении цикла с троичным условием. В ДССП-ТБМ такой вид цикла обеспечивается командой повторения **DW-+**. Эта команда цикла совмещает в себе проверку, надо ли прекращать его повторение, с выбором одного из двух действий, предусмотренных для исполнения в качестве тела цикла на очередном его шаге.

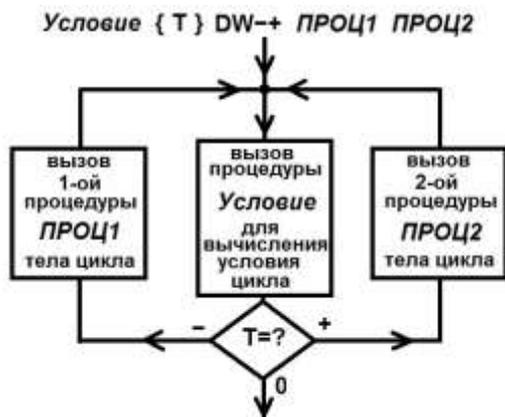


Рис. 4. Блок-схема выполнения цикла с троичным условием

Блок-схема выполнения этого цикла приведена на рис. 4 (подробнее он объясняется в [7]).

Продемонстрируем эффективность применения троичного цикла на примере задачи подсчёта разницы между количествами положительных и отрицательных чисел заданной последовательности (в предположении, что последовательность чисел задаётся в стеке до первого встреченного нулевого элемента).

Представим на языке ДССП-Т два варианта программы решения этой задачи: А) с применением троичного цикла (**DW-+**); Б) с применением двоичного цикла (**DW**) и команды ветвления.

Вариант А	
: CounNPt	{~, 0, X1, ..., Xn-1, Xn}
0 {d=0}	{~, 0, X1, ..., Xn-1, Xn, d}
E2 {Xi?}	<b>DW-+ 1- 1+</b> {~, d};

Вариант Б	
: CounNP	{~, 0, X1, ..., Xn-1, Xn}
0 {d=0}	{~, 0, X1, ..., Xn-1, Xn, d}
C2{Xn}	<b>DW</b> CounNP1 {~, 0, d}E2D{d};
: CounNP1	{~, Xi, d} E2
{~, d, Xi}	<b>BR+ 1+ 1-</b> {~, d};

Чтобы сравнить производительность этих процедур (**CounNPt** и **CounNP**) выполнения задачи, на программном имитаторе троичной машины (ТБМ) они запускались многократно для одинаковых последовательностей чисел различной длины с замером количества исполняемых ими команд. В результате проведённых испытаний было подтверждено, что алгоритм процедуры **CounNPt**, применяющий цикл с троичным условием, работает быстрее (более, чем в 2 раза), так как исполняется за меньшее число команд ТБМ (см. таблицу 3).

Таблица 3. Сравнение производительности процедур вариантов А и Б.

число элементов	количество исполненных команд	
	вариант А	варианта Б
10	41	85
20	71	155
40	131	295
50	161	365
100	311	715

## • Заключение

В статье представлены преимущества троичного компьютера. А также продемонстрировано, что в ряде случаев на таком компьютере можно более эффективно решать вычислительные задачи, применяя новые возможности, которые может предоставить программисту троичная машина и её система программирования.

# About the benefits of ternary machines and the effectiveness of ternary computing

A.A.Burtsev, B.A.Burtsev

**Abstract.** The article considers the advantages of ternary computers and demonstrates the effectiveness of ternary calculations on the basis of examples of programs developed by the authors to solve some of educational tasks in the dialogue system of structured programming (DSSP), created at the Moscow State University named after M.V. Lomonosov for the ternary virtual machine (TVM).

**Keywords:** ternary symmetric notation system, ternary virtual machine, DSSP, ternary logic, ternary branching, ternary loop.

## Литература

1. Д. Кнут. Искусство программирования на ЭВМ. М., Мир, 1978, Т.2, с. 216-219.
2. Н.П. Брусенцов, Х. Рамиль Альварес. Троичные ЭВМ “Сетунь” и “Сетунь 70”. «Первая Международная конференция "Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы" SORUCOM-2006». Россия, г. Петрозаводск, 3-7 июля 2006 г. (труды конференции в 2-х ч.). Изд-во ПетрГУ, 2006. ч.1, 45-51.
3. Н.П. Брусенцов. Заметки о троичной цифровой технике. «Вычислительная техника и вопросы кибернетики», Вып. 15 (1978), 145–155.
4. Будущее квантовых компьютеров — в троичных вычислениях, [http://www.infuture.ru/news.php?news\\_id=475pdf](http://www.infuture.ru/news.php?news_id=475pdf).
5. А. А. Бурцев, С. А. Сидоров. Троичная виртуальная машина и троичная ДССП. «Программные системы: теория и приложения», Т.6 (2015), №4, 29–97, [http://psta.pstiras.ru/read/psta2015\\_4\\_29-97.pdf](http://psta.pstiras.ru/read/psta2015_4_29-97.pdf).
6. Ю.С. Владимирова. Введение в троичную информатику: учебное пособие. М., АРГАМАК-МЕДИА, 2015.
7. А.А. Бурцев. Особенности программирования в ДССП для троичной машины. «Современные информационные технологии и ИТ-образование», Т. 13 (2017), № 14, 188–196.

# Механизмы универсального подключения программных исполнителей к системе ПиктоМир

Д.Б. Аглямутдинова<sup>1</sup>, Н.О. Бешшапошников<sup>2</sup>, А.Г. Леонов<sup>3</sup>, К.А. Машенко<sup>4</sup>,  
А.Е. Орловский<sup>5</sup>

<sup>1</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, dagliamutdinova@vip.niisi.ru;

<sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия, nbesshaposnikov@vip.niisi.ru;

<sup>3</sup> ФГУ ФНЦ НИИСИ РАН, Москва, Россия, МГУ им. М.В.Ломоносова, Москва, Россия,  
МПГУ, Москва, Россия, dr.l@vip.niisi.ru;

<sup>4</sup> ФГУ ФНЦ НИИСИ РАН, МГУ им. М.В.Ломоносова, Москва, Россия, kirill010399@vip.niisi.ru;

<sup>5</sup> ФГУ ФНЦ НИИСИ РАН, МГУ им. М.В.Ломоносова, Москва, Россия, orlovskiy@niisi.ru

**Аннотация.** В пропедевтическом курсе алгоритмики, в рамках которого дошкольники и младшие школьники осваивают основные понятия последовательного программирования, упор делается на составление программ для управления реальными и виртуальными роботами-исполнителями. В основной школе, при переходе на текстовый язык программирования ученики составляют программы в том числе и для управления теми же знакомыми им исполнителями. В данной статье рассматриваются методы подключения программных исполнителей, используемых в цифровой образовательной среде текстового программирования, КуМир, к цифровым образовательным средам ПиктоМир (пиктографическое программирование) и ПиктоМир-К (гибридное пиктограммно-текстовое программирование). Описаны некоторые детали реализации механизма подключения роботов-исполнителей, проблемы и методы их решения. Рассмотрены примеры подключения исполнителей. Знакомство со статьей поможет педагогам и программистам в создании и подключении своих исполнителей к средам ПиктоМир, ПиктоМир-К и КуМир.

**Ключевые слова:** ПиктоМир, КуМир, ПиктоМир-К, роботы-исполнители

## 1. Введение

В настоящее время образовательный процесс в России подвержен широкомасштабным сдвигам, основанным на повсеместном внедрении цифровых педагогических программных продуктов с автоматической проверкой заданий и формированием индивидуальных траекторий обучаемого. В этот процесс вовлечены все периоды обучения: от дошкольного до вузовского и послевузовского образования. В связи с этим возрастают потребности в разработке новых педагогических программных систем и переработке уже показавших свою эффективность существующих образовательных сред не только во внедрении методов искусственного интеллекта для автоматической проверки заданий и построения индивидуальных образовательных траекторий, но и в наполнении новым образовательным контентом существующих курсов.

Цифровая образовательная среда (ЦОС) ПиктоМир широко используется как в школьном и дошкольном образовании, так и в области высшего и дополнительного профессионально-

го образования для освоения основных понятий программирования и пропедевтики текстовых алгоритмических языков. Авторы цифровой образовательной среды ПиктоМир продолжают развитие этой системы как в области методики преподавания основ программирования, так и внедряя современные методы разработки программных систем в саму ЦОС ПиктоМир, расширяя область использования системы и модифицируя методы разработки ее частей [1], [2]. При этом ЦОС ПиктоМир продолжает сохранять продуманный и удобный для пользователей интерфейс, простое управление даже для самых маленьких детей и наполняется все новыми и увлекательными заданиями.

## 2. Интеграция систем

Отдельно от ЦОС ПиктоМир функционирует хорошо себя показавшая в образовательном процессе цифровая образовательная среда программирования КуМир, предоставляющая пользователю-ученику широкий выбор программных исполнителей, а также инструментов алгоритмического языка программирования [3]. В

этой среде используется школьный алгоритмический язык, который в научной литературе часто называют языком программирования КуМир. Этот учебный язык программирования по возможностям и синтаксису родственен другим объектно-ориентированным языкам [4]. Поскольку ЦОС КуМир и школьный алгоритмический язык обладают большим набором возможностей для составления программ, то от ученика, в отличие от ЦОС ПиктоМир, уже требуется иметь навыки чтения и печати на клавиатуре компьютера. Дополнительной нагрузкой на обучаемого ложится также требование соблюдения синтаксиса школьного алгоритмического языка, то есть умение исправлять синтаксические ошибки в его программе. Соответственно, обучение в этой среде рекомендуется проводить в основной школе, после успешного освоения преподавательского курса по алгоритмике в дошкольных образовательных учреждениях и начальной школы на ЦОС ПиктоМир и ПиктоМир-К.

Учебная среда программирования ПиктоМир-К служит своеобразной ступенью, которая облегчает переход из полностью пиктограммной среды ПиктоМир в текстовую среду КуМир. В ней присутствуют все программные исполнители из ЦОС ПиктоМир, а кроме того, становится доступным дополнительный функционал, помогающий освоить текстовое представление команд и управляющих конструкций алгоритмического языка программирования в системе КуМир. Более того, в этой системе появляется дополнительный инструментарий языка программирования: более сложные и часто используемые в производственных проектах конструкции, математические и иные вспомогательные функции, структуры и методы обработки данных.

Уже освоенный учениками в ЦОС ПиктоМир метод собирания программ из интуитивно понятных пиктограмм унаследован в ЦОС ПиктоМир-К, где перенос пиктограмм в программу приводит к их преобразованию в текстовое представление, очень напоминающее синтаксис школьного алгоритмического языка, при этом как и в ЦОС ПиктоМир, формы пиктограмм фактически указывают на возможные слоты для их размещения в программе [5], [6]. Такой подход позволил начинающему программисту конструировать свою программу, не обращая внимания на возможные синтаксические ошибки. При таком методе «сборки» учебной программы синтаксические ошибки просто не могут возникнуть даже в арифметических и логических выражениях.

### 3. Механизм подключения

#### 3.1 Система модулизации ЦОС ПиктоМир

ЦОС ПиктоМир предоставляет возможность педагогам и разработчикам самостоятельно создавать разнообразных роботов (программных исполнителей) для формирования у учеников более полной картины процесса программного управления, а также способствует удержанию мотивации при работе с системой, поскольку длительная работа с одним и тем же роботом снижает вовлеченность ребенка в образовательный процесс. Специально для этого сконструирован механизм добавления новых исполнителей, например, предназначенных для демонстрации решения конкретной, в том числе около производственной задачи, такой как работа на складе и др. В этом случае задача разработки и имплементации нового программного исполнителя может быть разделена на две подзадачи:

1. Автономная реализация и визуализация нового исполнителя, что осуществляется при владении навыками программирования на языке JavaScript [7].

2. Подключение автономного модуля к ЦОС ПиктоМир. Для решения последней подзадачи разработан специальный интерфейс – модулизация, позволяющий провести интеграцию нового исполнителя в ЦОС ПиктоМир без детального изучения реализации ядра системы [8]. Это система быстрого модульного подключения исполнителей к среде при помощи автоматического встраивания кода в критические места проекта в зависимости от выбранного робота.

Интеграция новых исполнителей является часто используемой процедурой.

Система ПиктоМир подразделена на модули, где модулем здесь называется набор исходных файлов на языке JavaScript и ресурсов, которыми являются файлы реализации программного исполнителя и обстановки. В них содержатся классы, наследники базовых классов из ядра базовой системы, реализующие необходимый для работы API. Это также могут быть наборы картинок в формате SpriteSheet и анимаций, используемых для визуализации роботов [9]. Основной ресурс, служащий связующим звеном нового робота и имеющейся системы, – файл описания модуля, который содержит наследник класса **pm.ModuleInfo** с наименованиями функций, их назначением и спецификацией возвращаемых данных. При этом набор функций-методов у всех модулей одинаковый. В классе модуля также должны содержаться все уникальные идентификаторы программных

исполнителей, обстановок, объектов и заданий.

### 3.2 Роботы системы КуМир

Роботы-исполнители в системе КуМир разработаны с использованием единой технологии с единым программным интерфейсом [10]. Они обладают набором свойств: собственными командами, условиями, заданиями, уникальным полем, которое либо является цельным, либо состоит из клеток, формой и цветовой раскраской робота. Таким образом, педагоги и разработчики получили возможность самостоятельно проектировать и создавать программных исполнителей с их методами-предписаниями и уникальной обстановкой в ЦОС КуМир.

При этом основная задача при подключении сторонних исполнителей в ЦОС ПиктоМир и ПиктоМир-К заключается в следующем: имея данный набор так называемых свойств программного исполнителя, предоставить простой механизм интеграции данного исполнителя в систему. Это осуществимо благодаря описанной ранее системе модулизации. Определив все свойства исполнителя в терминах системы ПиктоМир, система модулей способна порождать полноценного робота со всей описанной функциональностью на своем уникальном уровне. Для этого описываются все параметры и функции отдельного модуля, которые в дальнейшем используются и вычисляются при работе с данным исполнителем.

В итоге для переноса существующего программного исполнителя в систему КуМир необходимо выполнить следующие действия:

1. Описать реализацию исполнителя для модуля ЦОС ПиктоМир в системе программирования cocos2d-x.
2. Подключить вновь созданный модуль с помощью интерфейса, предоставляемого системой ПиктоМир.

### 3.3 Подключение программного исполнителя на примере Робота

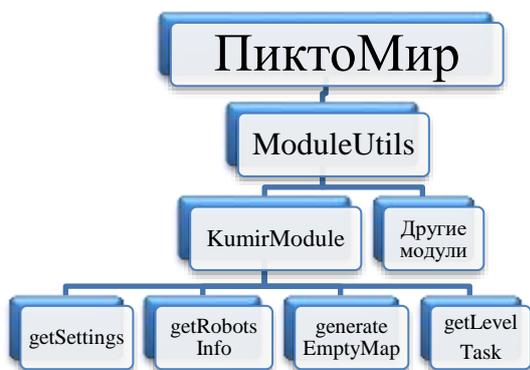


Рис. 1. Модульная схема ПиктоМир

Рассмотрим подробно пример подключения к ЦОС ПиктоМир(-К) исполнителя Робот. (см. рис. 1) Итак, первоначально происходит создание модуля для данного исполнителя в системе ПиктоМир. Для этого наследуется следующий класс, содержащий функции, которые определяют свойства уровня Робота:

```
pm.KumirLevelModule =
pm.ModuleInfo.extend()
```

Внутри данной библиотеки описываются такие свойства и функции, как тип уровня **getType()**, генерация поля **generateEmptyMap()**, создание уровня **getEmptyLevel()**, расположение графических файлов и наборов картинок в формате SpriteSheet **getRobotSpriteSheet()**, **getResources()**, **getMapTileset()**. Также здесь описываются файлы локализации программного исполнителя **getLocalization()** и создание задания для данного робота **getLevelTask()**. В конце файла данный модуль регистрируется, тем самым он добавляется в систему модулизации ПиктоМир:

```
pm.moduleUtils.register(new KumirLevelModule)
```

Затем можно приступить к формированию самого робота. Создается класс, содержащий команды и условия исполнителя:

```
pm.data.KumirRobot =
pm.PlayerRobot2D.extend()
```

Так как Робот умеет передвигаться в четырех направлениях и закрашивать клетки под ним, то создаются его методы:

```
NativeMethod.MoveUp, NativeMethod.MoveDown, NativeMethod.MoveRight, NativeMethod.MoveLeft, NativeMethod.Paint.
```

Также исходя из существующих условий программного исполнителя, а именно: проверки пустоты клетки с каждой из сторон от робота и проверки закрашенности клетки, создаются условия:

```
Condition.ClearUp, Condition.ClearDown, Condition.ClearLeft, Condition.ClearRight, Condition.IsPainted.
```

Аналогично создаются и заполняются такие классы исполнителя, как **KumirLevel**, **KumirMap**, **KumirMapElement**, **KumirTask**.

После выполнения данных действий можно полноценно использовать исполнителя Робот в системе ПиктоМир и ПиктоМир-К. Однако у рассмотренного робота могут использоваться методы, вызов которых предполагает передачу аргументов. Их реализация происходит уже в системе ПиктоМир-К. Для этого необходимо уже в созданном классе **pm.data.KumirRobot** добавить методы **NativeMethod.Temperature** и **NativeMethod.Radiation**, которые возвращают значение температуры и радиации соответ-

ственно в заданной клетке поля. В классе **KumirMapElement** появляются такие члены класса, как **\_temperature** и **\_radiation**, которые хранят информацию о температуре и радиации в каждой клетке поля.

Таким образом, в реализации Робота в ЦОС ПиктоМир-К к основным методам программного исполнителя добавляются еще два метода, которые пользователь может в полной мере использовать в своей программе.

### 3.4 Сложности и пути их решения

Во время переноса исполнителей из одной системы в другую неизбежно преодоление различных трудностей. Рассмотрим некоторые из них с примерами решения.

Исполнитель Водолей отличается от других исполнителей тем, что в нем нет поля, по которому бы передвигался робот. Все команды выполняются абстрактным роботом, а пользователь просто наблюдает визуализацию результата его действий. Проблема робота без поля удалось преодолеть следующим образом: так как система ПиктоМир обладает структурой наследования, было решено унаследовать класс, описывающий поле данного робота, от родительского класса **pm.AbstractMap**, который содержит в себе основные свойства и функционал, что позволяет работать с ним, как и с полем обычного исполнителя без существенных изменений. (см. рис. 2)

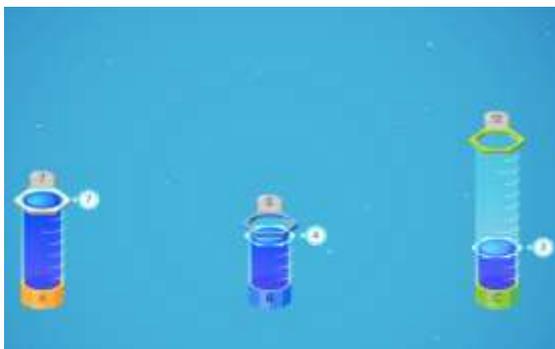


Рис. 2. Исполнитель Водолей

Другая ситуация была с исполнителями Чертежник и Черепаха. Данные исполнители-роботы могут составлять на своем поле некоторые фигуры при использовании инструментов Перо или Хвост соответственно. То есть возникает задача рисования произвольных линий на уровне при перемещении роботов по полю. Данная задача решается с помощью использования возможностей системы программирования **socos2d-x**, которая используется в ЦОС ПиктоМир. А именно, необходимо задействовать методы класса **cc.DrawNode**, позволяющие

рисовать точки, сегменты и многоугольники. (см. рис. 3)

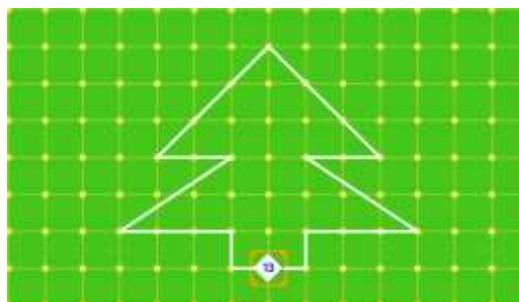


Рис. 3. Исполнитель Чертежник

Также стоит отметить нестандартную для системы ПиктоМир систему координат исполнителя Черепаха. Так как данный робот использует полярные координаты, то он должен “уметь” поворачиваться на заданный угол, а также команды Вперед и Назад должны осуществлять перемещение в направлении относительно робота, а не всей карты. В результате исполнитель приобрел свойство **\_direction**, которое изменяется при осуществлении поворота. Также были имплементированы методы перемещения робота по полю. (см. рис. 4)

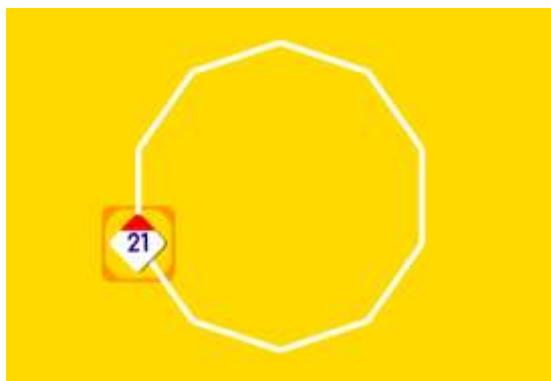


Рис. 4. Исполнитель Черепаха

## 4. Дальнейшее развитие

В связи с появлением и развитием различных приложений становится актуальным вопрос интеграции и внедрения новых сервисов. В настоящее время приоритетно развитие идеи о выделении исполнителей ЦОС ПиктоМир в отдельный независимый сервис, доступ к которому может легко осуществляться по определенному протоколу.

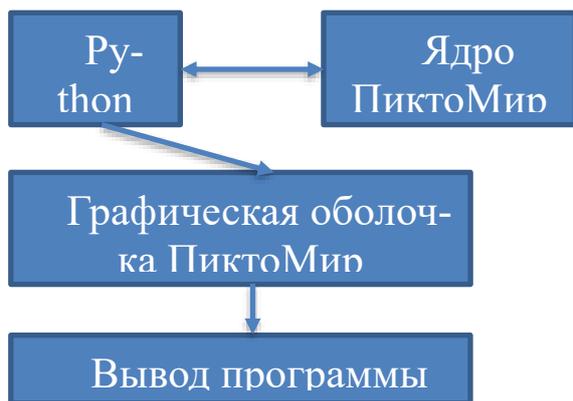


Рис. 5. Схема взаимодействия ПиктоМир и Python

Идея проекта заключается в универсальном использовании исполнителей, содержащихся в ядре системы ПиктоМир, в совершенно различных языках программирования в независимости от платформы.

В качестве примера предлагается рассмотреть язык программирования Python и исполнитель Вертун [11]. (см. рис. 5) Чтобы начать взаимодействие с исполнителями системы ПиктоМир пользователь Python объявляет в привычном для него формате импорт библиотеки: `import PiktoMir`. Затем, используя доступные функции из имеющегося у него протокола взаимодействия с исполнителями, пользователь пишет полноценную программу. При ее запуске пользователь может наблюдать появление в отдельном окне знакомой обстановки

уровня Вертун и самого робота. В этом окне робот начинает выполнять написанную пользователем программу в пошаговом режиме. По окончании работы программы пользователь получает информацию об успехе прохождения уровня или же о возможной ошибке в программе.

## 5. Заключение

ЦОС ПиктоМир позволяет не только создавать новых исполнителей для дальнейшего их использования, но и интегрировать существующих исполнителей из системы КуМир. Этот процесс не требует большого количества времени. Также он не является затратным в смысле энергии и ресурсов. Достаточно описать существующего робота в терминах системы ПиктоМир и предоставить его стилистику и визуализацию. После этого остается лишь воспользоваться системой модулей и получить готового к использованию исполнителя.

Публикация выполнена в рамках государственного задания по проведению фундаментальных исследований по теме «Разработка, реализация и внедрение семейства интегрированных многоязыковых сред программирования с автоматизированной проверкой заданий для учащихся образовательных организаций, ДОО, младшей, основной и старшей школы и студентов педагогических университетов.» (№ 0065-2019-0010).

# Mechanisms of universal connection of software executors to the PiktoMir system

Diana Agliamutdinova, Nikita Besshaposhnikov, Alexander Leonov, Kirill Mashchenko, Anton Orlovskii

**Abstract.** In the propaedeutic course of algorithms, in which preschoolers and primary schoolchildren study the basic concepts of sequential programming, the base is on creating programs for controlling real and virtual robots-executors. In a basic school, when switching to a text-based programming language, students compose programs, including for managing the same performers they know. This article discusses methods for connecting executors used in the digital educational environment of textual programming of the KuMir system to the digital educational environments of the PiktoMir system (pictographic programming) and PiktoMir-K (hybrid icon-text programming). Some specific details of the implementation of this mechanism for connecting robot executors, problems and methods for solving them are described. Examples of connecting executors are considered. Familiarity with the article will help teachers and programmers to create and connect their own executors to the environments of PiktoMir, PiktoMir-K and KuMir.

**Keywords:** PiktoMir, KuMir, PiktoMir-K, robots, executors

## Литература

1. Методическое пособие по системе ПиктоМир [Электронный ресурс] / Режим доступа: <https://www.niisi.ru/piktomir/m2019.pdf> свободный.
2. А.Г. Кушниренко, А.Г. Леонов, М.А. Ройтберг. Знакомим дошкольников младших школьников с азами алгоритмики с помощью систем ПиктоМир и КуМир. «Труды НИИСИ РАН», Т. 5 (2015), № 1, 134-137.
3. А. Г. Кушниренко, А. Г. Леонов. КуМир – учебное программное обеспечение базового курса информатики. «Информатика в школе», (2017), № 8.
4. Ian Graham. Object-Oriented Methods: Principles and Practice. “Addison-Wesley – Third Edition” (2004).
5. А. Г. Леонов, Н. О. Бешапошников. Пиктограммный язык программирования Пикто. «Вестник кибернетики», Т. 28 (2017), № 4.
6. А. Г. Кушниренко, А. Г. Леонов. Архитектура смешанной пиктограммно-текстовой системы программирования для дошкольников и младших школьников. «Вестник кибернетики», Т. 28 (2017), № 4.
7. Dr. Axel Rauschmayer. The Nature of JavaScript. “Speaking JavaScript. – First Edition. – O’Reilly Media”, (2014).
8. Н. О. Бешапошников, К. А. Мащенко, А. Е. Орловский. Разработка программных исполнителей системы ПиктоМир. «Труды НИИСИ РАН», Т. 8 (2018), № 6, 155-158.
9. Creating a Sprite from a Sprite Sheet [Электронный ресурс] / Режим доступа: <https://docs.cocos2d-x.org/cocos2d-x/v3/en/sprites/spritesheets.html> свободный
10. А. Г. Леонов, Ю. А. Первин, Я. Н. Зайдельман. Программные исполнители в цифровых образовательных средах ПиктоМир, Роботландия и КуМир. «Информатика в школе», Т. 152 (2019), № 9.
11. A. Sweigart. Core Python Applications Programming. 2012.

# Применение технологий бережливого производства в сфере нематериальных услуг

Г.Е. Каратаева

*БУ ВО Ханты-мансийского автономного округа – Югры «Сургутский государственный университет», [karataeva\\_ge@surgu.ru](mailto:karataeva_ge@surgu.ru), 8922653702*

**Аннотация** Анализ практических кейсов внедрения технологий бережливого производства в непромышленной сфере приводит к заключению, что главной причиной провала проектов является проблема определения ценности для клиента результатов деятельности ее субъектов. В статье раскрыты основные элементы методологии бережливого производства, обобщены методики оценки результатов внедрения лин-технологий, обоснованы ограничения их применения в сфере нематериальных услуг. Автор приходит к выводу, что главным ограничением выступает доверительный характер услуг. Невозможность установления прямой зависимости факторов создания ценности (кадры, материальная база, финансовые ресурсы и т.п.) с ее измеримой величиной названо основным препятствием эффективного применения бережливых технологий в этой сфере.

**Ключевые слова:** сфера нематериальных услуг, производительность труда, бережливое производство, ценность услуг, потери, инструменты бережливого производства

## 1. Введение

Внедрение бережливых технологий в сфере государственного управления и социальной сфере стало популярным направлением в России в середине 200-х гг. с началом административной реформы [1]. Во многих субъектах Российской Федерации были приняты соответствующие документы. В частности, пилотами стали Республика Татарстан, Ханты-Мансийский автономный округ – Югра, Ульяновская область, позже Липецкая, Ростовская, Нижегородская, Белгородская области и др. Подробно хронология реализации основных проектов бережливого производства в регионах, государственных компаниях и компаниях с государственным участием в России рассмотрена, например, в работах А.С. Царенко и О.Ю. Гусельниковой [2], С.Г. Тяглова и И.В. Такмашевой [3].

Региональные документы по внедрению бережливых технологий были разработаны в основном на период до 2019 года. Анализ официальных сайтов региональных органов исполнительной власти показывает, что к 2019 году интерес к этой теме заметно снизился. Так в Ханты-Мансийском автономном округе – Югре информация о бережливых технологиях последний раз обновлялась в первом квартале 2018 г. (портал «Бережливометр» <https://dereconom.admhmao.ru/berezhlivometr>) Позже это направление было включено в региональные портфели национального проекта «Повышение производительности труда и поддержка занятости» (далее – Национальный проект) [4].

Национальный проект, в частности, предусматривает в рамках Федерального проекта «Системные меры по повышению производительности труда» предоставление финансовых и нефинансовых мер поддержки предприятиям, реализующим программы повышения производительности труда, снижение административно-правовых ограничений для роста производительности труда и формирование системы подготовки кадров, направленной на обучение управленческого звена предприятий и служб занятости населения [4]. Пока согласно отчетам Счетной палаты Российской Федерации, реализация национального проекта не набрала необходимого темпа [5].

Анализируя проблемы Национального проекта, эксперты Счетной палаты справедливо отмечают, что «в документах стратегического планирования, разработанных в рамках целеполагания по отраслевому принципу, данные о производительности труда измеряются с помощью различных показателей, в результате чего они оказываются несопоставимыми в отраслевом разрезе и в динамике» [5].

Еще одной существенной проблемой является методологическая несогласованность государственных программ. Производительность труда в них рассматривается «в виде выпуска продукции (выработки) на одного сотрудника в стоимостных или натуральных величинах, или же в виде валовой добавленной стоимости, приходящейся на одного жителя региона, а также темпов прироста таких величин к предыдущему/базовому году» [5] (табл. 1).

Таблица 1.

Примеры показателя производительности труда в государственных программах  
Российской Федерации

Государственная программа	Показатель производительности труда	Интерпретация показателя
Развитие физической культуры и спорта	Производительность труда в сфере физической культуры и спорта	численность населения Российской Федерации, систематически занимающегося физической культурой и спортом, в расчете на одного штатного работника, %; объем платных услуг, предоставляемых гражданам Российской Федерации, в расчете на одного штатного работника, руб./1 шт. раб.
Экономическое развитие и инновационная экономика	Рост производительности труда на средних и крупных предприятиях базовых несырьевых отраслей экономики	Рост по отношению к предыдущему году, %
Социально-экономическое развитие Калининградской области	Производительность труда	ВРП на 1 занятого в экономике, тыс. руб.

Анализ региональных государственных программ в редакции 2019 года демонстрирует исчезновение из их паспортов показателя «Производительность труда» и упоминаний о бережливом производстве практически в большинстве из них. От прежних редакций остался блок, посвященный обучению бережливому производству управленческих кадров на базе региональных центров компетенций производительности труда под руководством АНО «Федеральный центр компетенций в сфере производительности труда» (ФЦК).

Несмотря на отмеченные проблемы государственных программных документов стратегического планирования и не всегда очевидные результаты их реализации, в России есть успешные практики внедрения бережливого производства, в том числе, в сфере услуг. Речь идет о ПАО «Сбербанк», которому удалось реализовать уникальный проект за относительно небольшой временной интервал [6]. С 2018 года активно продвигается проект АО «Росатом»,

предусматривающий широкое внедрение бережливых технологий в социальной сфере: здравоохранении и образовании (Бережливая поликлиника, Бережливый вуз) [7,8]. Результаты этого проекта еще предстоит оценить.

Концепция бережливого производства достаточно подробно изложена в современной экономической литературе. В рамках настоящей статьи обозначим ее основные элементы, имеющие принципиальное значение для ее реализации в непромышленной сфере.

Бережливое производство основано на

1) анализ объекта (в частности, производственных процессов предприятия) на предмет наличия потерь;

2) устранение потерь с помощью специальных инструментов;

3) непрерывное наблюдение за объектом с целью его постоянного совершенствования.

Классификация потерь включает 7 основных групп (табл. 2.):

Таблица 2.

Группировка потерь в концепции бережливого производства

Группа потерь	Содержание
Традиционная классификация потерь (muda)	
1. Перепроизводство	продукт/услуга производится в большем объеме, чем требуется заказчику
2. Избыток запасов	хранение любых запасов в количестве, существенно превышающем минимально необходимое
3. Транспортировка	лишнее движение материалов
4. Задержки	большие простои между этапами производства продукта/выполнения услуги
5. Дополнительная обработка	лишняя обработка/действия из-за несоответствующих инструментов или плохой конструкции продукта (из-за несоответствующего планирования и проектирования услуги)
6. Перемещения	лишние движения человека, потери при подборе материалов, поиске компонентов, инструментов, информации, документов
7. Дефекты	доработка и отбраковка несоответствующей продукции/ненадлежащее выполнение услуги

Дополнительные виды потерь	
8. изменчивость (mura)	неравномерность выполнения работы, колебания спроса, поставок, нестабильность характеристик продукции
9. перегрузку (muri)	излишняя загруженность оборудования или операторов, возникающая при работе с большей скоростью или темпом и с большими усилиями в течение долгого периода времени по сравнению с расчетной нагрузкой
10. незадействованный потенциал персонала	неспособность в полной мере использовать талант и способности людей
11. транзакционные издержки	издержки, связанные с договорной деятельностью, а также менеджментом
12. недостаточную ценность продукции	несоответствие продукции ожиданиям потребителя и других заинтересованных сторон

Составлено автором по ГОСТ Р 56020-2014 Национальный стандарт Российской Федерации Бережливое производство. Основные положения и словарь.

Основные инструменты бережливого производства широко известны, признаны эффективными во многих сферах и хорошо зарекомендовали себя в отдельных проектах в России. Среди них «5S», TPM, канбан, стандартизированные операционные инструкции, вытягивание, картирование потока создания ценности и др.

Стоит отметить, что часто бережливые технологии упоминаются одновременно с цифровой трансформацией. Есть мнение, что цифровая трансформация позволяет создать полностью прозрачную систему управления, лишённую транзакционных издержек и других потерь, что само по себе отменяет необходимость в бережливых технологиях. В теории это можно рассматривать как некую формулу идеального газа. Здесь уместно вспомнить уже ставшее крылатым выражение об оцифровке бардака. Цифровая трансформация не может стать полной заменой бережливого производства. Однако это мощное дополнение. В реализации проектов по цифровой трансформации залогом успеха является правильно выбранная мера, которая в итоге способствует конечному улучшению процессов.

В настоящем исследовании на основе анализа особенностей сферы услуг планируется установить, возможно ли в ней полноценное внедрение технологий бережливого производства.

## 2. Ценность услуг, производительность труда и потери в сфере нематериальных услуг

Экономическая наука имеет богатейший пул исследований, посвященных проблемам непроизводственной сферы. Тем не менее, многие вопросы носят дискуссионный характер. На пример, в части определения структуры непроизводственной сферы можно выделить два подхода [9]:

- 1) отождествление непроизводственной сферы со сферой услуг; общественное производство подразделяется на два сектора: материальное производство и непроизводственная сфера или сфера услуг;
- 2) отождествление сферы услуг с общественно организованным производством потребительских услуг; общественное производство подразделяется на три сектора: материальное производство (производство вещных благ), сфера услуг (производство потребительских услуг) и непроизводственная сфера.

Далее мы будем придерживаться второго подхода. Не ставя перед собой задачу, рассматривать эволюцию развития науки в этом направлении выделим характеристики сферы услуг, определяющие их ценность для потребителя (табл. 3).

Таблица 3.  
Характеристики сферы услуг, определяющие специфику экономических отношений в ней

№ п/п	Характеристика	Содержание
1	Функциональное назначение	Труд работников сферы услуг осуществляется в форме услуги. Она выступает, с одной стороны, как труд в качестве действия, а с другой стороны, как результат трудовой деятельности в виде полезного эффекта труда. Эти две стороны услуги представляют собой две стороны единого целого. Выделяют: нематериальные (чистые) услуги (не получают предметно осязаемую форму (услуги социально-культурной

		сферы) и услуги, которые создаются отраслями материального производства (торговля, общественное питание, транспорт и связь, а также производственные услуги бытового обслуживания и жилищно-коммунального хозяйства).
2	Моменты процесса труда (целесообразная деятельность человека, предметы и средства труда)	Труд работников сферы услуг индивидуализирован. Сфера услуг имеет дело с индивидуальными запросами людей, с их разнообразными вкусами и настроениями. Это предъявляет особые требования к личным качествам работников сферы услуг. Она требует от них умения и способности работать с людьми. Труд работников многих отраслей сферы услуг отличается высоким интеллектуальным содержанием, характерен высокий удельный вес работников высококвалифицированного труда.
3	Труд в меньшей степени поддается автоматизации	Повышенная трудоемкость. Расширение объема услуг и улучшение качества обслуживания обеспечиваются главным образом за счет увеличения численности трудовых ресурсов в непромышленной сфере, введения прогрессивных методов организации труда и управления.
4	Доверительный характер и синергизм услуг	Качество медицинских и образовательных услуг невозможно полностью проверить сразу в момент их потребления. Оно зависит не только от производителя, но и от потребителя услуги, что может изменить процесс и результат ее оказания. Выигрыш от потребления услуги зависит не только от преподавателя (врача), но и от обучающегося (пациента) (все люди имеют различные способности к восприятию информации и, следовательно, к обучению; обладают индивидуальными особенностями и состоянием здоровья и т.п.)
5	Услуги являются предметами потребления	По К. Марксу «в каждый данный момент, в числе предметов потребления, наряду с предметами потребления, существующими в виде товаров, имеется известное количество предметов потребления в виде услуг» Платные услуги, как и товары, удовлетворяют платежеспособный спрос населения на потребительские блага. Сфера платных услуг представляет собой главным образом индивидуальную форму удовлетворения соответствующих потребностей населения. Платные услуги характеризуются значительно большим ассортиментом и представляют большую свободу их выбора. Развитие платных услуг обеспечивается за счет привлечения денежных средств населения и организаций. Потребность в платных услугах и фактическое их потребление дифференцируются в зависимости от уровня среднедушевых доходов населения и других социально-экономических факторов. Поскольку услуги не могут накапливаться, невозможно иметь запасы нематериальных благ (без учета возможностей хранения определенной информации, как составляющей услуги). Неравномерность потока потребителей и сезонность в спросе на услуги значительно сказываются на использовании ресурсов в сфере услуг. Поэтому при установлении цен и тарифов на услуги, режима работы учреждений и работников и т. д. важное значение имеет учет фактора времени.
6	Большая территориальная рассредоточенность организаций сферы услуг, что вызывается необходимостью обеспечения широкого доступа населения к ним.	Сфера услуг имеет дело с обслуживанием населения. Поэтому территориальное размещение ее организаций в первую очередь зависит от демографических факторов (численности, плотности и состава населения, расселения людей, уровня урбанизации и интенсивности миграции и т. д.), затем от территориальной организации производственной инфраструктуры и в конечном счете определяется размещением производственного потенциала. Территориальный подход к управлению сферой услуг призван реализовать принцип комплексности ее развития в пределах соответствующих административно-территориальных единиц. Он предусматривает застройку населенных пунктов, которая обеспечивает рациональное сочетание производственных зон с

		жилыми районами, с сетью общественных, культурных и учебно-воспитательных учреждений, спортивных сооружений, торговых и бытовых предприятий, транспорта и общественного питания. Все это способствует созданию наилучших условий для жизни людей.
7	Методы финансирования	<p>Бюджетное финансирование является основным методом финансового обеспечения оказания государственных (муниципальных) услуг в социально-культурных отраслях непроизводственной сферы, и финансирование их осуществляется преимущественно через систему местных бюджетов (школы, поликлиники, учреждения культуры и спорта).</p> <p>В сфере услуг (особенно социальных) действие товарно-денежных отношений ограничено. По сравнению с материальным производством имеется сравнительно меньше частных организаций. Также применяются методы смешанного финансирования (частные организации получают доступ к бюджетным субсидиям; государственные (муниципальные) организации оказывают платные услуги наряду с услугами в рамках государственного (муниципального) задания).</p>
8	Организация и оплата труда работников сферы услуг.	<p>В качестве источников фонда оплаты труда в организациях сферы социальных услуг выступают в большей степени бюджетные субсидии на финансовое обеспечение государственного (муниципального) задания и в меньшей степени доходы от платных услуг.</p> <p>Трудности количественной оценки труда и его нормирование в отраслях сферы социальных услуг обусловили широкое применение здесь таких показателей, которые характеризуют не полезный эффект трудовой деятельности, а потенциальные возможности работников в достижении соответствующего полезного эффекта: категории должностей и организаций, уровень образования и стаж.</p> <p>Творческий характер многих видов труда в сфере социальных услуг определяет особенности организации заработной платы в ней. Здесь широко применяются штатно-окладная система оплаты труда, надбавки за интенсивность и высокое профессиональное мастерство, а также специфические формы вознаграждения за труд в виде гонорара, выплат разового характера за консультацию и экспертизу.</p>
9	Ценообразование	Необходимые затраты на создание услуги (себестоимость) не воплощаются в стоимости, а цена приобретает относительно самостоятельную форму своего существования. Цена выполняет функцию регулирования и учета бюджетных расходов на оказание услуги. Одним из ключевых факторов ценообразования в сфере услуг является учет их социальной значимости.
10	Оценка эффективности	Социальный эффект, социальные показатели ставятся выше по отношению к хозяйственным результатам.

Составлено автором по [10]

Представленные характеристики услуг могут быть дополнены философской проблемой восприятия таких ключевых понятий социальной сферы как образование, здравоохранение, культура. Здесь надо отметить двойственную природу этих фундаментальных характеристик развития общества. Их сущность гораздо шире, чем понятие услуги. В целях формализации экономических отношений в этой сфере в рамках гражданского и бюджетного законодательства введены соответствующие понятия: образовательная услуга, медицинская услуга и т.п. Однако формализация отношений не может решить проблему определения ценности социальных услуг. Например, в договоре на оказание

образовательной услуги (подготовка по программе высшего образования, бакалавриат) устанавливается, что «исполнитель обязуется предоставить образовательную услугу, а заказчик оплатить обучение по основной образовательной программе (наименование образовательной программы высшего образования, объем в час.) с присвоением обучающемуся соответствующей квалификации». Предполагается, что качество услуги подтверждено аккредитацией и наличием соответствующей лицензии у вуза. В свою очередь аккредитация подтверждает соответствие образовательной программы вуза требованиям Федерального государственного образовательного стандарта высшего

образования (далее - ФГОС ВО). Формально все характеристики, необходимые для определения качества и объема услуги установлены. Вместе с тем ценность услуги, а также ее потребителя невозможно определить. Кто выступает в данном случае потребителем обучающийся, заказчики (родители, организации-работодатели, государство в лице соответствующего органа власти) налогоплательщики? Отметим, что в отношении образовательных услуг, оказываемых в рамках государственного задания договоры вообще не составляются. Таким образом, с точки зрения гражданского законодательства отношения между исполнителем (вузом) и обучающимся не возникают. Финансовые отношения фиксируются только в соглашении о предоставлении субсидии на выполнение государственного задания, а показатели качества и количества услуг указывается в государственном задании. Показатели качества могут отличаться от показателей, предусмотренных ФГОС ВО.

Ценность услуги для каждой из перечисленных сторон-участников этого процесса будет разной. Отдельно стоит отметить стандартизацию качества услуг. Образовательная программа согласно ФГОС ВО должна формировать определенные компетенции. Набор компетенций формируется заинтересованными в подготовке выпускников сторонами. Акцент делается

на работодателя. Возникает вопрос, может ли работодатель сформулировать потребность в компетенциях на 4-6 лет вперед? Способны ли преподаватели обеспечить формирование этих компетенций? Как это стандартизировать? Это только вершина проблемы определения ценности и качества образовательной услуги в высшем образовании.

Все это вносит неопределенность в понятие ценности этих услуг для клиента и создает ограничения для стандартизации этих услуг, оценки эффективности деятельности работников и пр., а также и применения технологий бережливого производства

Отдельно остановимся на понятии производительность труда в сфере услуг. Динамика среднегодовой численности занятых в России по отраслям экономики за период 1990–2018 гг. (рис 1.), демонстрирует общемировую тенденцию перехода рабочей силы из производственной в непроизводственную сферу. При этом численность занятых растет в сфере потребительских услуг (в торговле, транспорте, финансовых, страховых, административных и гостиничных услуг, а также услуг), в то время как численность занятых и ее доля в структуре занятости в сфере нематериальных услуг (здравоохранение, образование, культура, наука) снижается.

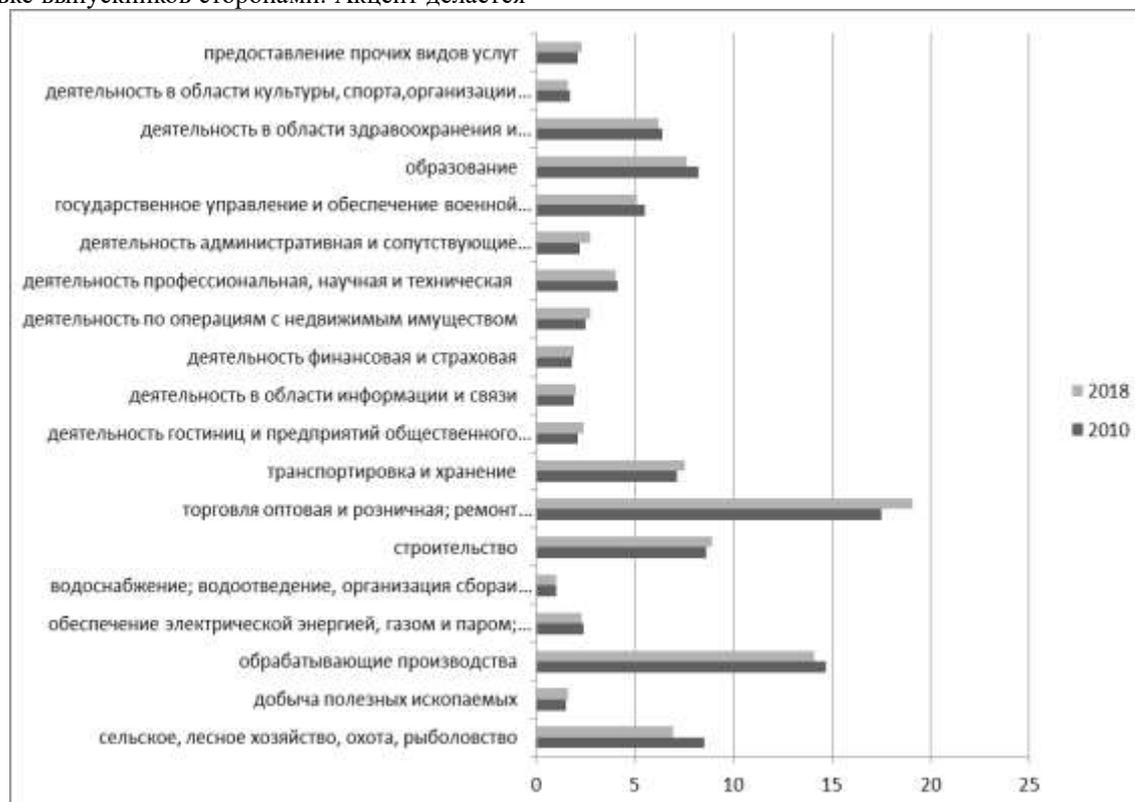


Рис 1. Динамика среднегодовой численности занятых в России по отраслям экономики за период 2010–2018 гг. в %

Фиксирующаяся тенденция в развитых странах при снижении занятости, общих затрат времени и продолжительности отработанного времени на одного занятого в промышленности объем производимых и потребляемых обществом благ не снижается, а растет, что обусловлено повышением производительности труда вследствие научно-технического процесса. Для России эта тенденция не характерна. Индекс производительности труда в целом по экономи-

ке России имеет отрицательную динамику (рис 2).

Отмечается рост производительности труда в сфере транспортировки и хранения, торговли, строительства и сельского хозяйства.

В сфере непроизводственных услуг производительность труда, как показатель эффективности использования трудовых ресурсов не определяется. Это связано опять же с природой труда в этой сфере (см. табл.3).

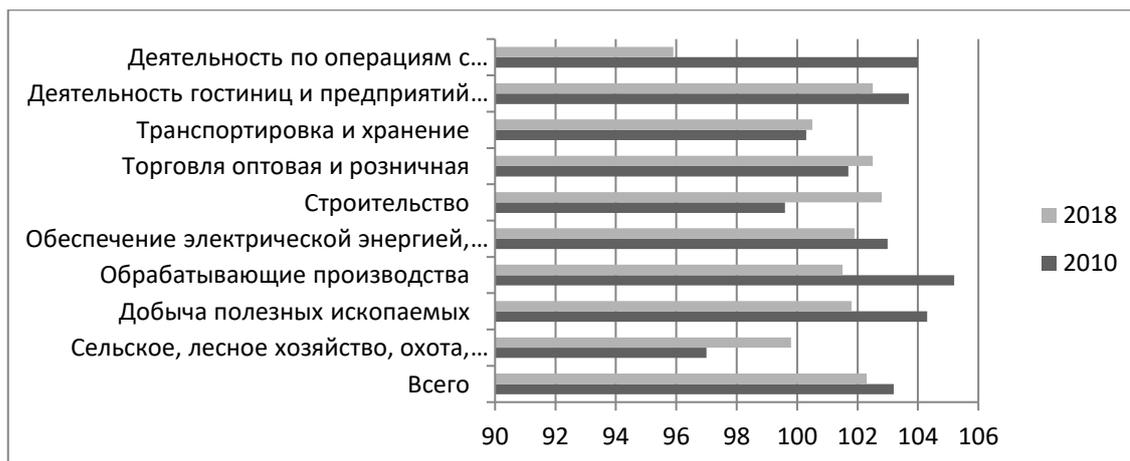


Рис. 2. Динамика производительности труда по видам экономической деятельности, в % к предыдущему году

Одним из показателей производительности труда выступают затраты времени на производство продукции и оказание услуг, или временные затраты. С экономической точки зрения рабочее время регламентируется в установленных нормативах величинах затрат труда определенной квалификации посредством использования таких единиц измерения времени, как минута, час, день, год. В сфере нематериальных услуг это основной показатель для установления оплаты труда работников.

Нематериальность результата трудовой деятельности, слабая зависимость стоимостных показателей деятельности организации (выручки, прибыли) от результатов труда конкретного работника делает затруднительным установление конкретных измеримых показателей для оценки эффективности труда работников сферы нематериальных услуг. По этой причине проводимая реформа систем оплаты труда в здравоохранении, образовании, культуре и социальном обслуживании не дала ожидаемых результатов и не смогла заменить существовавшую в СССР тарифную систему. По нашему мнению для работников этой сферы следует сохранить систему профессиональной аттестации кадров. Бережливые технологии могут быть реализованы здесь в полной мере, потому что именно человеческий

фактор играет главную роль как в рассматриваемой нами сфере, так и идеологии «lean».

При внедрении принципов бережливого производства одновременно с улучшением основных процессов преобразуется мышление людей, меняется организационная культура организации. Переход к бережливой организации труда вносит существенные коррективы в кадровую политику организации, что обусловлено следующим:

- изменение организационной структуры;
- снижение численности персонала до оптимальных размеров, создание новых рабочих мест через расширение видов деятельности;
- обучение персонала принципам бережливого производства;
- обогащение труда и изменение методов мотивации [12].

Если говорить о потерях, то в организациях сферы нематериальных услуг в большей степени присутствуют потери, обозначенные в табл. 3 как дополнительные. В начале проекта внедрения технологий бережливого производства необходимо в первую очередь идентифицировать их и понять, каким образом они могут быть устранены.

Рассматривая методики оценки эффективности проектов внедрения бережливых техно-

логий (например, табл.4) приходим к выводу, что большинство из них ориентированы на промышленную сферу. Применить их к основным «производственным» процессам в сфере

нематериальных услуг так же сложно, как определить результат труда врача или учителя, но можно с некоторыми ограничениями.

Таблица 4.

Критерии эффективности инструментов бережливого производства )		
Система	Показатель	Единица измерения
TQS	Количество рацпредложений на 1 работника в год	штук
	Вариация значений качественных характеристик	-
TPM	Общая эффективность оборудования	%
	Доля плановых ремонтов в общих простоях оборудования	%
Just-in-time	Доля времени обработки материалов	%
	Оборачиваемость запасов	дней
	Переналадка	минут
Многопроцессная работа	Потери рабочего времени в действиях рабочих	%
	Загрузка рабочих	%

Источник: [13,14]

Главным оценочным показателем здесь будет являться «удовлетворенность потребителя». Согласно отчетам учреждений о выполнении государственных (муниципальных) заданий, размещенных на официальном портале bus.gov.ru, такая удовлетворенность растет. Сегодня получить объективную информацию для проведения оценки крайне сложно. К вспомогательным (обеспечивающим) процессам инструменты бережливого производства могут применяться в полном объеме, как в офисе обычного производственного предприятия. В книге Т. Шукета и Д. Теппинга «Бережливый офис. Управление потоками создания ценности» изложен общий подход [15]. Дополнив его особенностями конкретного учреждения можно получить необходимый методический инструментарий по этому вопросу.

### 3. Вывод

Недостаточное понимание сущности сферы нематериальных услуг, особенно ее социальной составляющей (здравоохранение, образование, наука, культура) приводят к практикам, когда внедрение проектов бережливого производства за редким исключением сводится, в основном, к оптимизации материальных затрат. Специалисты, реализующие проекты внедрения бережливых технологий в офисах компаний, отмечают: «пока вы убираете лишние карандаши, люди тратят часы рабочего времени на совершенно бессмысленную и бесполезную работу: они неделями согласуют заказы с полудюжиной директоров, они носятся по отделам в надежде хоть ненадолго ускорить выполнение срочного заказа, они пишут многостраничные отчеты, которые никто не читает, они заполня-

ют сотни журналов и бланков, вместо того, чтобы делать свою основную работу. И поэтому попытка сократить потери компании путем сокращения количества ручек или введением правила «использовать обратную сторону бумаги для печати черновиков» всегда выглядит как игра в бирюльки» [11]. То же самое можно сказать и о сфере социальных услуг. В частности, преподаватель, учитель, врач или социальный работник испытывает те же сложности в своей повседневной деятельности. Необходимо начать с начала, следуя принципам бережливого производства: изучить процесс на предмет возможности его усовершенствования с точки зрения создания ценности для клиента, а уже потом переходить к устранению потерь (в том числе, оцифровке процесса) и созданию системы мониторинга за состоянием системы. Пока же внедрение бережливых технологий наряду с цифровизацией в социальной сфере «пропускает» эту первую стадию, стремясь сразу оптимизировать «очевидные» потери и создавать системы мониторинга. Наглядным примером являются отчеты по проекту Бережливая поликлиника. Согласно отчету о реализации проекта «в среднем время нахождения пациентов в регистратуре снизилось в 2-4 раза, сокращение времени нахождения в очереди к врачу в 3-7 раз, количество посещений поликлиники по вопросам прохождения диспансеризации, медицинских осмотров сократилось в 2-3 раза, время ожидания забора крови сократилось в 1,5-2 раза. Медицинский персонал отмечает снижение бумажного документооборота, компьютеризацию рабочих мест, помощь в оснащении современным медицинским оборудованием» [7]. Смогло ли это кардинально увели-

чить ценность для клиента? Отчасти, да. Однако наличие цифровой записи к врачу не решает проблемы доступности медицинской помощи при отсутствии врача. Статистика удовлетворенности граждан медицинскими услугами в России, особенно в регионах, свидетельствует о существовании системных проблем, которые требуют системных же решений.

## Application of lean manufacturing technologies in the field of intangible services

G. Karataeva

**Abstract** An analysis of practical cases of implementing lean manufacturing technologies in the non-manufacturing sector leads to the conclusion that the main reason for the failure of the projects is the problem of determining the value for the client of the results of the activities of its entities. The article reveals the main elements of the lean manufacturing methodology, summarizes the methods for evaluating the results of lean-technology implementation, and substantiates the limitations of their use in the field of intangible services. The impossibility of establishing a direct relationship between the factors of creating value (personnel, material base, financial resources, etc.) with its measurable value is called the main obstacle to the effective use of lean technologies in this area.

**Keywords** intangible services, labor productivity, lean manufacturing, value of services, losses, lean manufacturing tools

### Литература

1. Распоряжение Правительства РФ от 25.10.2005 N 1789-р (ред. от 10.03.2009) «О Концепции административной реформы в Российской Федерации в 2006 - 2010 годах». [Электронный ресурс] Доступ из справ.-правовой системы «КонсультантПлюс» URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_56259/](http://www.consultant.ru/document/cons_doc_LAW_56259/) (дата обращения: 24.06.2020).
2. А.С. Царенко, О.Ю. Гусельникова Проекты «Бережливый регион», «Бережливая поликлиника», «Бережливый город» как шаги на пути к созданию «Бережливого Правительства»: оценка реализации лин-инициатив в государственном секторе РФ // Государственное управление.-2019.-№ 73.-С.167-202
3. С.Г. Тяглов, И.В. Такмашева Использование технологий бережливого производства в качестве основы инновационного развития регионов // Вопросы регулирования экономик. - 2017. - Том 8, № 2. – С. 97-110. DOI: 10.17835/2078-5429.2017.8.2.097-110.
4. Паспорт национального проекта «Повышение производительности труда и поддержка занятости», утверждён решением президиума Совета при Президенте Российской Федерации по стратегическому развитию и национальным проектам 24 декабря 2018 года.
5. Отчет о промежуточных результатах экспертно-аналитического мероприятия «Анализ планирования и реализации мероприятий национального проекта «Производительность труда и поддержка занятости», включая оценку сбалансированности целей, задач, показателей, мероприятий и финансовых ресурсов, а также его соответствия долгосрочным целям социально-экономического развития Российской Федерации [Электронный ресурс] Счетная палата Российской Федерации URL: <http://audit.gov.ru/upload/iblock/316/316dfb87833f8a900e3672099f4fd75d.pdf> (дата обращения: 24.06.2020).
6. Итоги реализации Стратегии развития Сбербанка на период 2014–2018. [Электронный ресурс] URL: [https://2017.report-sberbank.ru/pdf/ar/ru/strategic-report\\_development-strategy-results.pdf](https://2017.report-sberbank.ru/pdf/ar/ru/strategic-report_development-strategy-results.pdf) (дата обращения: 24.06.2020).
7. Проект «Бережливая поликлиника». [Электронный ресурс] URL: <https://rosatom.ru/sustainability/proekt-berezhlivaya-poliklinika/> (дата обращения: 24.06.2020).
8. Ассоциация бережливых вузов. [Электронный ресурс] URL: <http://lean-vuz.udsu.ru/> (дата обращения: 24.06.2020).
9. Л.К. Климович, И.А. Ткаченко Роль и место сферы услуг в общественном производстве . [Электронный ресурс] URL: <http://edoc.bseu.by:8080/bitstream/edoc/5215/2/Klimovich%20L.%20Tkachenko%20I.%20Rol%27%20i%20mesto%20sfery%20uslug%20v%20obshchestvennom%20proizvodstve.pdf> (дата обращения: 24.06.2020).

10. А.И. Панов. О роли сферы услуг в системе рыночной экономики \ [Электронный ресурс] URL: [http://www.unn.ru/pages/issues/vestnik/99990193\\_West\\_econ\\_finans\\_2002\\_1\(4\)/B\\_3-6.pdf](http://www.unn.ru/pages/issues/vestnik/99990193_West_econ_finans_2002_1(4)/B_3-6.pdf) (дата обращения: 24.06.2020).
11. В. Казарин. Как и почему бережливое производство терпит крах в офисе? [Электронный ресурс] URL: <http://wkazarin.ru/2015/02/17/how-and-why-lean-production-crashes-in-the-office/> (дата обращения: 24.06.2020).
12. Н.В. Просвирина, А.И. Тихонов. Роль человеческого фактора при внедрении принципов бережливого производства на предприятиях высокотехнологичного производства // Идеи К.Э. Циолковского в инновациях науки и техники, с. 448–450.
13. А.С. Птускин, В.Ю. Анцев, Н.А. Витчук. Методика расчета показателей эффективности проекта внедрения инструментов бережливого производства \ Известия Тульского государственного университета. Экономические и юридические науки. 2014. – С. 253-266 [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/metodika-rascheta-pokazateley-effektivnosti-proekta-vnedreniya-instrumentov-berezhlivogo-proizvodstva> (дата обращения 26.06.2020).
14. А.Б. Николаева. Оценка эффективности внедрения бережливого производства на промышленных предприятиях. \ Вестник экономики, права и социологии, 2016 . – С. 69-72. [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/otsenka-effektivnosti-vnedreniya-berezhlivogo-proizvodstva-na-promyshlennyh-predpriyatiyah> (дата обращения 26.06.2020).
15. Shuker, T., Tepping, D. Berezhlivyi ofi s. Upravlenie potokami sozdaniya tsennosti (Lean office. Value creating processes management). Moscow: Standarty i kachestv, 2009, 205 p

Подписано в печать 20.7.2020 г.  
Формат 60x90/8  
Печать цифровая. Печатных листов 8  
Тираж 100 экз. Заказ № 518

Отпечатано в ППП «Типография «Наука»  
121099, Москва, Шубинский пер., 6