

## Моделирование физики в условиях неполноты

А.В. Каминский

(Получена 13 сентября 2004; опубликована 16 сентября 2004)

Развивается принципиально новый подход к построению физической теории, основанный на концепции, которую можно назвать субъективной физикой. Предметом ее исследования является субъект-объектное отношение в замкнутом конечном мире. Исследуются свойства простейшей алгоритмической модели мира, построенной на основе избыточного совершенного кода. Показано, что модель проявляет свойства, характерные для квантовой физики.

### Введение

Как должен выглядеть мир для его субъекта? Какие физические законы сможет открыть гипотетический физик – субъект замкнутого конечного мира? Будут ли эти законы похожи на открываемые нами законы физики? Ответам на эти вопросы и посвящена настоящая статья.

Мы предполагаем существование некоей объективной реальности в форме замкнутой конечной системы, которую далее будем именовать словом "Мир". Состояния мира не являются физическими, как и он сам не является физическим объектом. Физическими объектами или явлениями будем считать только принципиально наблюдаемые проявления реальности. Легко видеть, что для субъекта, являющегося частью той же самой замкнутой системы должны иметь место очевидные ограничения на наблюдаемость, связанные с неполнотой, подобной той, которая имеет место в замкнутых аксиоматических системах, исследуемых математической логикой [1].

Физическая неполнота является ключевым понятием субъективной физики. Знание субъекта о мире в принципе не может быть исчерпывающим. Поэтому в отличие от объективной реальности, существующей для гипотетического внешнего (его еще можно назвать – объективным) наблюдателя, для субъективного наблюдателя возникает слой физической реальности со свойственной ему неполнотой, которая преломляется в наших физических законах в форме неопределенности, индетерминизма, необратимости и др. Физическая реальность не существует сама по себе, но возникает как конфигурационная субъект-объектная сущность. Идея физической неполноты оказывается чрезвычайно плодотворной. На ее основе могут быть сделаны вполне конкретные выводы о характере физических законов, возможных в нашем мире.

В статье на основе этих идей изучается частная проблема обоснования квантовой механики (КМ).

### Структура модели

Мы продолжим наше исследование методом компьютерного моделирования. То есть займемся тем, что по мнению Р. Фейнмана [2] невозможно. В статье [3] нами отмечалась чрезмерная категоричность Р. Фейнмана в этом вопросе. В статье [4] строится модель клеточного автомата с псевдоквантовой эволюцией. Что имел в виду Фейнман, утверждая, что моделировать физику на обычном компьютере (универсальной машине Тьюринга) нельзя? Эти интуитивные догадки Фейнмана впоследствии развивал Д. Дойч [5]. Заслуга Дойча состоит в том, что он обратил наше внимание на то, что вычисление – физический процесс. Поэтому наш успех в решении задачи моделирования физической реальности должен определяться процессами, положенными в основу функционирования самого

вычислительного устройства. Полагают, что возможности универсальной машины Тьюринга недостаточны для описания квантовой реальности, но соответствующую теорему никто пока что не доказал. Так можно ли моделировать физику на обычном (МТ) компьютере? На наш взгляд сама постановка вопроса требует уточнения. От модели не следует ожидать исчерпывающего описания предмета.

Предположим, что компьютер или мозг физика - теоретика представляет собой часть замкнутого конечного мира. Вследствие неполноты такая ситуация исключает возможность исчерпывающего моделирования реальности, – средствами части мира (компьютер) не может быть описан весь мир, включающий этот компьютер в себя, как составную часть. Физическая реальность не может быть полна.

Однако *мы всегда можем построить некую метатеорию, которая включит описанную выше ситуацию как целое в состав своих объектов исследования*. Построив модель наблюдателя, мы можем понять, каковым будет его мир. В этом состоит наш метод, который будем далее именовать *теорией субъективной физики*. В отличие от стандартной научной методологии, отрывающей мир от субъекта (исследователя) и страдающей очевидными ограничениями, связанными с неполнотой, новая методология позволяет построить модель субъект-объектного мира, а саму неполноту сделать объектом исследования.

Если говорить о компьютерном моделировании, то наш метод может быть реализован следующим образом. Некоторая программа должна имитировать объективную эволюцию мира в объективном времени (алгоритмическом). Роли субъекта и объекта могут играть две взаимодействующие подпрограммы. На подпрограмму, выполняющую роль субъекта в этой модели, распространяются все ограничения неполноты, подобные тем, о которых мы говорили выше. Таким образом, можно получить представление об устройстве реального мира. Именно это мы имеем в виду, оппонировав Фейнману и Дойчу в их выводах.

На простом примере рассмотрим, каким образом на классическом компьютере могут быть смоделированы квантовые корреляции ЭПР-типа [6].

### **Имитация квантовой нелокальности на классическом компьютере.**

Сущность явления состоит в следующем. Пусть, мы проводим измерение над квантовой системой, состоящей из двух разнесенных в пространстве частей, так что события измерения разделены пространственно-подобным интервалом. И пусть изначально система приготовлена в состоянии суперпозиции чистых состояний  $\psi_1$  и  $\psi_2$ . Пусть, далее, для измерения над первой частью мы применяем либо прибор “A1”, либо прибор “A2”, а над второй либо “B1”, либо “B2”. Собственные значения операторов “A1, A2, B1, B2” дают измеряемые этими приборами величины “a1, a2, b1, b2”.

Достаточно произвести измерение только над одной из частей системы, чтобы другая часть немедленно зафиксировала свое состояние, невзирая на то, что эти части могут быть причинно не связаны. Но важен не сам факт корреляции, которая может выражать закон сохранения энергии или углового момента, но факт более сильной корреляции, не объяснимой классической физикой. Критерий отличия классических корреляций от квантовых был найден Беллом и выражается в виде неравенства:

$$|\langle ab \rangle - \langle a_1 b \rangle + \langle ab_1 \rangle + \langle a_1 b_1 \rangle| \leq 2.$$

Угловыми скобками здесь обозначены корреляционные функции соответствующих наблюдаемых. Ключевой эксперимент Аспекта по двухфотонной корреляции показал, что передача информации осуществляется именно в момент коллапса волновой функции и, следовательно, никакой априорной информации части системы не несут. Получается, что первая часть системы как-то передает второй части информацию о том, проводилось ли над ней измерение и какой оно дало результат именно в момент коллапса, нарушая все наши представления о локальности и причинности.

Рассмотрим, для простоты обычную схему эксперимента по корреляции фотонов. Пусть в результате двухфотонного перехода источник света излучает 2 фотона, разлетающихся в противоположных направлениях к приборам “А” и “В” соответственно. Закон сохранения четности гарантирует, что, если первый фотон Х- поляризован, то другой тоже Х-поляризован. Угол же поляризации  $\Phi$  относительно условно выбранного направления случаен. Пусть первый и второй наблюдатели применяют приборы “А<sub>1</sub>” и “В<sub>1</sub>” соответственно, ориентированные под углами  $\Phi_1$  и  $\Phi_2$ . Вероятность того, что первый наблюдатель (прибор “А<sub>1</sub>”) обнаружит Х-поляризацию, равна:

$$P_{1x} = 1/2 \cos^2(\Phi - \Phi_1)$$

Вероятность, что при этом второй наблюдатель, применив прибор “В<sub>2</sub>”, ориентированный под углом  $\Phi_2$ , обнаружит Х-поляризацию равна

$$P_{2x} = 1/2 \cos^2(\Phi_1 - \Phi_2)$$

Можно написать простейшую программу, имитирующую такое поведение. В псевдокоде программа может иметь вид:

```

Procedure MESSURE (Φ,Φ0)
Begin
If Random(1)>Cos2(Φ - Φ0) then Φ=Φ0
Else Φ=Φ0+90°
End.

```

```

-----
MAIN PROGRAM
Begin
Read (Φ1,Φ2)
Φ=Random(360°)
Call MESSURE(Φ,Φ1)
Print Φ
Call MESSURE(Φ,Φ2)
Print Φ
End.
-----

```

Здесь вначале описана процедура измерения MESSURE(), Передача параметров осуществляется через переменную  $\Phi$  в обоих направлениях. Процедура управляет функцией распределения вероятности обнаружения той или иной поляризации. В теле процедуры использован генератор псевдослучайных чисел. Основная программа MAIN PROGRAM является протоколом проведения эксперимента. В начале экспериментатор задает углы  $\Phi_1$  и  $\Phi_2$  ориентации применяемых приборов “А”, и “В”. Далее источник света

испускает фотоны со случайной (0-360°) поляризацией  $\Phi$ . Затем один за другим производят измерения оба наблюдателя. Значения углов полученных поляризаций выводятся на печать. Легко убедиться, что программа действительно описывает квантовые корреляции и, как и следовало ожидать, нарушает неравенства Белла. Каким же образом программа имитирует квантовую нелокальность? Секрет очень прост. Все дело в том, что результат измерения  $\Phi$ , полученный первым наблюдателем, «нелокально» передается в процедуру измерения второго наблюдателя. Такой подход к интерпретации квантовой механики известен как теории скрытых переменных. Обычно считается, что нарушение неравенства Белла доказывает невозможность описания реальности посредством локальных скрытых переменных. И если бы удалось «изобрести» нелокальный способ передачи параметров, – квантовая механика была бы обоснована.

- Но то, что вы сделали - это обман! – Скажете вы. – Вы не моделируете нелокальность, а «подсовываете» ее нам в готовом виде.

Однако не торопитесь с выводами. Действительно все, что мы пока сделали это даже не модель, а некая наивная аналогия. Нелокальность в ней возникает благодаря тому, что мы просто игнорируем сам механизм передачи параметра  $\Phi$ . Давайте подумаем о том, какой будет физическая реальность для наблюдателя, который даже не догадывается о том, что в его мире может иметь место некий механизм, реализующий нелокальность. Кроме того, пусть эта физическая реальность такова, что любая попытка обнаружить этот механизм будет безуспешной. Очевидно, что наш наблюдатель вынужден будет «изобрести» свою Копенгагенскую интерпретацию.

Но, что за причина может помешать нам обнаружить этот самый скрытый механизм? Очевидно, эта причина должна быть весьма принципиального характера. Продолжим рассмотрение нашей компьютерной аналогии. Пользователю, как правило, доступна только внешняя сторона работы программы, тогда как процессы, осуществляемые на более глубоком уровне иерархии программного обеспечения, скрыты от него. Однако нам необходимо как-то онтологизировать это невосприятие субъектом-наблюдателем «тонких» процессов, происходящих в мировом "компьютере" (в противном случае, обнаружив эти процессы, наш наблюдатель «доберется» и до механизма передачи параметра « $\Phi$ » и констатирует его локальный характер). Что касается модели, то этого легко можно добиться, допустив, что для нашего наблюдателя минимальным временным интервалом, или квантом времени, является один шаг приведенной выше программы, тогда как вся необычайно сложная динамика работы компьютера, характеризующаяся значительно меньшими характерными временами, останется скрытой от него вследствие невосприятия им столь малых временных интервалов. В реальности, проще всего было бы постулировать наличие кванта времени. Например, взяв за основу планковское время  $\approx 10^{-43}$  сек. Однако, в модели субъект-объектного мира, которую мы рассматриваем ниже, в таком искусственном приеме не возникает необходимости, ибо возможности восприятия субъектом реальности в ней ограничиваются естественным образом. Далее делается попытка формализации идеи субъективной физики, которую до сих пор мы рассматривали на качественном уровне.

### **Квантовая механика и коды, исправляющие ошибки. (математическая модель)**

Введем в рассмотрение гипотетического внешнего (объективного) наблюдателя, который может изучать взаимодействие физика (субъективного наблюдателя) с объектом.

Для объективного наблюдателя мир может быть конечен.

Аксиома 1: Число состояний мира конечно.

Для описания конечного мира мы воспользуемся алгеброй на конечных полях. Такой аппарат достаточно хорошо разработан и является разделом современной теории чисел. Эварист Галуа ввел в рассмотрение поля  $GF(q)$  с конечным числом элементов  $q=p^n$ , где  $p$  - простое, а  $n$ -натуральное [7]. Это поле классов эквивалентности многочленов. На этих полях сохраняются обычные свойства арифметических операций над вещественным полем  $R$ .

Рассмотрим пространство, в котором каждая точка представлена  $n$ -числами, принимающими значения из элементов поля классов вычетов по модулю некоего неприводимого полинома. Рассмотрим частный случай  $GF(2^n)$ . Элементы этого поля образуют  $n$ -мерное двоичное пространство. Для наглядности в качестве примера рассмотрим случай  $n=3$ . На рисунке 3 показаны все 8 точек этого пространства, являющиеся вершинами 3-х мерного куба. Базисом этого пространства является ортонормированная система векторов, которые условно обозначим как степени двойки.  $2^0, 2^1, 2^2$ . Будем считать, что 8 вершин этого куба полностью описывают все состояния нашего модельного мира. Запишем все 8 векторов этого пространства в виде столбцов матрицы.

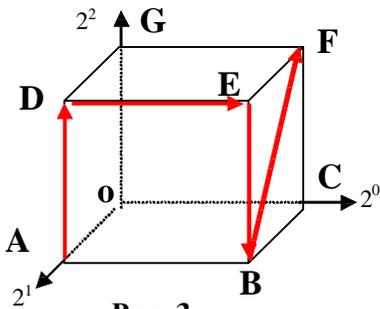


Рис. 3

$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} 2^0 \\ 2^1 \\ 2^2 \end{matrix}$$

Если сложить, например, 6-ой столбец (вектор OF) с 7-ым (вектор OD) получим 4-ый столбец (вектор OG). То есть элемент того же поля. Заметим, что складывать координаты векторов нужно по модулю 2 и, очевидно, без переноса в старшие разряды. Элементы рассматриваемого пространства - вектора,

полиномы или числа. Поэтому, если мы и будем пользоваться двоичной позиционной системой счисления или нумеровать вершины куба десятичными числами, то всегда нужно будет помнить о том, что это просто обозначения элементов поля, на котором действует своя особая арифметика. **Назовем элементы поля  $GF(2^n)$  алгоритмическими состояниями, имея в виду, что они могут быть связаны некоторым алгоритмом, действующим над этим полем.** Сопоставим множество этих состояний с **фундаментальными состояниями мира**. Будем считать, что эти состояния столь просты, что нахождение системы в том или ином фундаментальном состоянии однозначно характеризуется булевыми значениями  $\{0,1\}$ . Напомним, что **физические состояния** описываются амплитудами на комплексном поле. Ниже мы покажем, что они конструктивны и могут быть выражены через эти самые фундаментальные или алгоритмические состояния.

Рассмотрим вектор алгоритмического состояния  $\xi$ . Он может меняться со временем  $\xi(t^a)$  согласно некоторому алгоритму. Здесь  $t^a$ - алгоритмическое время или просто номер шага алгоритма. На рисунке 2 жирными красными стрелками показан фрагмент одной из возможных траекторий системы ADEBF в пространстве алгоритмических состояний.

Аксиома 2: На фундаментальном уровне мир детерминистичен.

Отсюда сразу же можно сделать ряд выводов о характере алгоритмической динамики: Алгоритмическая траектория не может самопересекаться. Это следует из того факта, что каждое последующее состояние алгоритма однозначно определяется предыдущим. По аналогии с классической механикой такое движение можно назвать эргодическим. При этом максимальное время системы из  $N$  состояний будет  $T=2^N$ , где  $N$ -размерность пространства состояний.  $N$ - мерный куб объективного пространства можно рассматривать, как граф, а траекторию движения системы в пространстве состояний, как гамильтонов цикл на этом графе [8]. Напомним, что у каждого графа с количеством точек больше двух есть гамильтонов цикл - это способ соединения всех вершин графа одной кривой, проходящий по его ребрам и не проходящий через одну вершину два раза. Такие траектории эргодичны и образуют множество кодов Грея. Мы будем называть их траекториями Грея. Эти коды характерны тем, что расстояние Хемминга между соседними словами всегда равно 1. Траектории Грея замечательны и тем, что дают в двоичном пространстве путь наименьшей длины. Другими словами описывают движение по геодезической.

Как мы уже говорили, некоторые алгоритмические состояния системы не могут быть различены наблюдателем вследствие неполноты. Субъективный наблюдатель не способен различить каждую вершину такого графа. Однако, ему будет доступна более грубая структура меньшей размерности. Эту структуру образуют группы близко расположенных вершин, объединенных в гранулы – **физические состояния**. Переходы системы между этими гранулами или «физическими» состояниями с точки зрения субъекта не могут быть детерминированы, так как определяются недоступной ему динамикой скрытых алгоритмических состояний. Этот подход открывает путь для понимания связи случайного и детерминированного в физике. Согласно нашей модели **природа квантового индетерминизма обусловлена субъективной неполнотой мира**.

Рассмотрим для простоты модель 7 – битного мира, имеющего всего 128 состояний рис 4.



Каждое состояние такого мира может быть описано 7-ми битным словом. Пусть число состояний субъекта ограничено 16 состояниями. То есть 4 бита из 7 занимает субъект. Конечно, такой субъект будет обладать весьма ограниченными интеллектуальными способностями! Но это не мешает нам сделать достаточно общие выводы об особенностях восприятия конечного мира субъектом и характере физических законов, открываемых им. Число состояний мира, очевидно, можно определить как\*

$$N_{\text{wor}} = N_{\text{sub}} \cdot N_{\text{obj}}, \quad (1)$$

где  $N_{\text{sub}}$  – число состояний субъекта, а  $N_{\text{obj}}$  – число состояний объекта. Длина мирового слова, соответственно, определится как сумма длин слов субъекта и объекта, (т.к. длина= $\log_2 N$ ). Субъект, находясь с остальной частью мира в информационном отношении (а только такой тип отношений возможен в рассматриваемой модели), занимает часть «ячеек» или разрядов мирового слова, уменьшая тем самым число состояний оставшейся части мира.

Какие выводы можно сделать из этого элементарного рассмотрения? Очевидно, что такой субъект будет способен различить только 16 внешних по отношению к нему физических состояний, приписав им номера в соответствии со своими внутренними состояниями 1,2,3...16. Заметим, что все числа большие 16 для нашего субъекта будут трансфинитны\*\*. Это следует из того очевидного факта, что ни прочитать, ни сохранить в памяти наш 4-х битный субъект их не сможет по причине отсутствия достаточных для этого вычислительных ресурсов. Мы видим, как естественно понятие бесконечности возникает в конечной модели! [9].

Объективно каждое физическое состояние – гранула, содержит  $128/16=8$  алгоритмических состояний. Структура этого вырождения не доступна субъективному наблюдателю. Это означает, что наблюдатель в принципе не может определить точно, в каком состоянии находится система в каждый момент физического времени.

Как вы уже поняли, согласно нашей модели, рассматриваются 2 уровня реальности. На объективном, скрытом от субъекта уровне, процессы описываются как переходы между  $2^n$  точками конечного двоичного пространства в соответствии с некоторым алгоритмом. Процессы, имеющие место на уровне физического мира, являются проекцией объективной алгоритмической динамики на подпространство физического мира. Как может быть описана взаимосвязь этих двух слоев реальности?

В решении этой задачи нам поможет несколько далекая от теоретической физики задача построения кодов, исправляющих ошибки [10]. Эта область прикладной информатики в последнее время бурно развивалась в связи с нуждами космической радиосвязи и вычислительной техники. Напомним вкратце суть проблемы. Читатель, знакомый с теорией кодов, исправляющих ошибки, без ущерба для понимания дальнейшего материала может пропустить следующие два абзаца.

\* Говоря о субъективном, ни здесь, ни далее, мы не имеем в виду субъективный характер восприятия наблюдателя в психологическом смысле. В понятие "субъективность" мы вкладываем ровно столько смысла сколько содержится в выражении (1).

\*\* Левенгейм и Сколем показали, что если непрерывность связывается с несчетностью, то она оказывается относительной. То есть, если в некоторой аксиоматике множество несчетно, то в другой аксиоматике оно может оказаться счетным. В условиях физической неполноты имеет место сходная ситуация.

Допустим, мы передаем сообщения 4-х битными словами. Всего в нашем словаре будет  $2^4=16$  слов. Допустим, что каждое слово в нашем словаре значимо. Очевидно, в таком случае ошибка даже в один бит совершенно недопустима, так как в результате искажения информации корреспондент получит совершенно другое слово. Проблема решается путем передачи по тому же каналу избыточной информации. Делается это следующим образом. Вместо 4-х бит для передачи тех же 16 слов будем использовать 7- битные слова. При помощи 7 битов, как известно, можно передать  $2^7=128$  слов, но мы будем передавать те же 16 слов. Для этого “лишним” словам припишем те же самые значения из нашего словаря. Сгруппируем “лишние” слова таким образом, чтобы ошибка при передаче слова X в один бит переводила его в одно из слов группы с тем же смыслом. Таким образом, искаженная информация может быть восстановлена.

Далее для удобства будем пользоваться векторным представлением. Так 7- битное слово будет одним из векторов в 7-мерном двоичном пространстве. Выберем из 128 векторов 16 векторов, максимально удаленных друг от друга. Возьмем их в качестве базы кода. Все остальные 128-16 слов представляют собой всевозможные ошибочные сочетания битов.

В теории информации принято расстояние между векторами – кодовыми словами измерять по числу двоичных знаков, которыми они отличаются. Определенную таким образом метрическую меру называют расстоянием Хэмминга. При описанном выше способе выбора базы все его вектора удалены друг от друга на расстояние Хэмминга  $\Delta s \geq 4$ . Матрица базовых слов может быть очень легко построена для любых  $N=2$  или  $N=4k$ , где  $k$ -целое. Дело в том, что она составляется из матриц Адамара  $H_N$ , итеративная процедура получения которых хорошо известна [11].

$$F = \frac{J - [H_N ; -H_N]}{2} . \quad (2)$$

Здесь  $J$ - матрица, составленная из единиц. Заметим следующее. Из 16 столбцов матрицы  $F$  4 - линейно независимы. Они образуют четырехмерное подпространство семимерного пространства над  $GF(2)$ . Этот факт и позволяет использовать эти 16 векторов в качестве полезного кода, исправляющего ошибки. Действительно, если как в нашем примере, расстояние между кодовыми словами  $\Delta s > 4$ , и известно также, что при передаче информации происходит не более одной ошибки, то такую ошибку всегда можно исправить.

Теория кодов, исправляющих ошибки, привлекает нас по той причине, что наблюдатель, как мы показали, в силу физической неполноты не может с абсолютной точностью определять состояние системы. Можно сказать, что *вследствие фундаментальной физической неполноты канал «связи» между физической реальностью и объективной реальностью принципиально «зашумлен»*. Система, для описания которой требуется  $N$ -двоичных разрядов, не может быть исчерпывающе описана меньшим числом разрядов. Мы здесь говорим о принципиальном ограничении на точность измерения и задания любых величин. Напомним, что КМ, так же как и классическая теория, легкомысленно допускает бесконечно точные измерения и вычисления. Наша модель устраняет этот недостаток.

В нашем примере мы выбрали 7 битный мир, в котором 4 бита занимает субъект не случайно. Это так называемый совершенный (7,4) код, осуществляющий упаковку двоичного пространства без перекрытий и пропусков. Этот код, если обнаруживает, то обязательно исправляет ошибку. Запишем матрицу (2) для (7,4) модели в явном виде:

$$\begin{array}{c}
 \text{Столбцы – вектора физических состояний} \\
 \downarrow \downarrow \downarrow \quad \dots \quad \downarrow \\
 \mathbf{F} = \begin{pmatrix}
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1
 \end{pmatrix} \quad (3)
 \end{array}$$

Мы имеем 16 векторов, которые мы сопоставим физическим состояниям (столбцы). В двоичном пространстве каждый такой вектор указывает на центр дискретной сферы, образуемой 7 точками. Таким образом, физическое состояние в нашей модели представляет собой сферу, образованную фундаментальными состояниями.

16 векторов, указывающие на центры этих сфер, образуют 4-х мерное  $\eta$ -пространство физических состояний. Позже мы покажем, что 4-х мерное подпространство 7-мерного двоичного  $\xi$ - пространства, натянутое на множество физических состояний, является аналогом 4-х мерного Гильбертова пространства КМ.

Все 16 физических состояний являются линейными комбинациями четырех базовых состояний. Физический базис здесь не ортогонален. (Это легко проверить, вычислив скалярные произведения столбцов матрицы (3).) Но его всегда можно ортогонализировать, воспользовавшись стандартной рекуррентной процедурой ортогонализации [12].

В общем случае нужно рассматривать  $m$ -мерное физическое двоичное пространство, вложенное в  $n$ - мерное пространство большей размерности, которое мы называем алгоритмическим. Полученную  $(n,m)$  структуру будем далее именовать  $(n,m)$  - моделью.

Такого рода структуры хорошо изучены и имеют практическое применение в теории кодирования. Как уже говорилось, описанная структура представляет собой не что иное, как хорошо известный  $(7,4)$  избыточный совершенный код. Напомним, что совершенными или плотно упакованными называют коды, в которых выполняются соотношения

$$\sum_{\sigma=0}^{\Delta} C_n^{\sigma} (b-1)^{\sigma} = b^r - 1, \quad (4)$$

где  $\Delta$  - максимальная кратность исправляемых ошибок;  $b$  – основание кода  $r$  - число проверочных символов. Мы полагаем, что только класс совершенных  $(n,m)$  моделей пригоден для описания физической реальности. Дело в том, что в модели не должно быть фундаментальных алгоритмических состояний, не относящихся ни к одному физическому состоянию. Во всяком случае, у нас нет оснований предполагать существование слоя реальности, никак не связанного с физическим миром. Число известных совершенных кодов ограничено кодами Хэмминга значности

$$n = \frac{b^r - 1}{b - 1} \quad (5)$$

и бинарным циклическим кодом Голея. Проблема получения более точных нижних оценок для числа совершенных кодов привлекает внимание математиков давно, однако в полной мере эта задача не решена.

Математический аппарат, применяемый в теории избыточных линейных кодов, оказывается пригоден для описания рассматриваемой ситуации с физической неполнотой. Вектор физического состояния (кодовое слово) не дает исчерпывающего описания системы, так как это состояние вырождено по соответствующим алгоритмическим состояниям (ошибкам, которые могут быть исправлены). Физическое и алгоритмическое пространства связаны друг с другом проекционными операторами, которые мы будем называть операторами редукции и антиредукции. Проекционный оператор редукции  $\Omega$  понижает размерность вектора, осуществляя "спуск" из алгоритмического пространства в подпространство физической реальности.

$$\eta = \Omega \cdot \xi. \quad (6)$$

Действие этого оператора имитирует процесс редукции квантового состояния. Воздействуя на алгоритмическое состояние оператором  $\Omega$ , получим соответствующее физическое состояние. В теории кодирования этой операцией пользуются для декодирования сообщений. Для совершенных кодов вектор любого алгоритмического состояния мира всегда лежит внутри одной из сфер физических состояний. И механизм действия оператора редукции, который мы отождествляем с процедурой измерения, сводится к определению этой принадлежности. Оператор редукции для (7,4)-модели может быть представлен (7×4) матрицей. Например:

$$\Omega = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

Этот оператор является прямоугольной матрицей и поэтому, воздействуя на вектор, меняет его размерность. Здесь он осуществляет "спуск" из алгоритмического пространства в физическое пространство. Вид матрицы (7) определяет *представление*. Представление определяется конфигурационным отношением наблюдатель – объект. Характер этого отношения, в свою очередь определяемый измерительной аппаратурой, системой отсчета и прочее, задает способ гранулирования алгоритмических состояний в физическое пространство. Обратная операция осуществляет "подъем" из физического в алгоритмическое пространство:

$$\xi = \Omega^{-1} \cdot \eta. \quad (8)$$

В теории кодирования такие матрицы используют для генерирования избыточных кодов.

Итак, модель в основном построена. Она описывает два слоя реальности – скрытый алгоритмический и физический миры, связанные посредством операций (6) и (8). Ниже мы покажем, что поведение модели, во многом напоминает квантовое. Во-первых, сразу можно сказать, что модели присущ индетерминизм. Этот индетерминизм имеет место

только в физическом слое реальности. Во-вторых, нелокальность, которая объясняется связью физических состояний на уровне скрытой реальности. Хотя наш подход многое объясняет, его практическое применение в таком виде, как описано выше, весьма затруднительно. Дело в том, что алгоритмическая динамика  $\xi(t^d)$  для физического наблюдателя не доступна. Мы не можем работать с этой функцией, как мы работаем с волновыми функциями (ВФ). Физическая же динамика  $\eta(t)$ , получаемая проецированием (б) уже чисто классическая и так же не пригодна для описания.

Таким образом, нам необходимо найти аналог ВФ. В процессе решения этого вопроса, мы обнаружим еще более тесную связь нашей модели с квантовой механикой.

### Физическое поле, смысл вектора состояния

Фундаментальная физическая категория поля неизбежно возникает при переходе к субъективному наблюдателю. Здесь важным моментом является понятие скрытого времени. Два или более последовательных моментов алгоритмического времени, которые мы не способны различить (по причине неполноты), сливаются в один, образуя скрытый интервал времени.

Таким образом, целые участки алгоритмической траектории могут соответствовать всего одной точке на оси физического времени. Состояния  $\xi_1, \xi_2, \xi_3 \dots \xi_n$ , проходимые алгоритмической траекторией, последовательно могут принадлежать к разным физическим состояниям и разным моментам физического времени. Вектор  $\xi(t^d)$  полностью описывает систему, но для субъекта это описание недоступно. Как мы уже говорили, субъекту доступна только совокупность состояний, которую он называет физическим состоянием, скажем  $x$ . То есть  $x \leftrightarrow \{\xi_1, \xi_2, \xi_3 \dots \xi_n\}$ . Определим физическое состояние следующей конструкцией. Она формирует множество состояний, соответствующих наблюдаемой  $x$ :

$$\Phi_i(x) = \delta(\Omega \cdot \xi_i - x) \xi_i, \quad (9)$$

$\delta$ -функция здесь фильтрует состояния, принадлежащие физическому состоянию  $x$ .  $i$  - номер шага алгоритма или алгоритмическое время.  $\Phi_i(x)$  – компоненты вектора:

$$\Phi(x, t) = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_n \end{pmatrix} \quad (10)$$

Размерность этого вектора равна скрытому времени в алгоритмических единицах. Будучи определен уже в физическом времени, этот математический объект будет вести себя локально, и подчиняться дифференциальным закономерностям. Если мы, таким образом, опишем движение частицы, например, в импульсном состоянии, то в физическом слое реальности она будет выглядеть, как протяженный, распределенный в пространстве объект. Этот объект имеет много общих черт с физическим полем. Есть ли что-либо общее между функцией  $\Phi$  и квантово-механической волновой функцией  $\psi$  ?

Прежде всего, из самой конструкции (9) следует ортогональность функций, относящихся к разным собственным значениям оператора физического состояния. Так в нашем случае для всех  $i \neq j$ , очевидно

$$\sum_k \Phi_k(x_i) \cdot \Phi_k^*(x_j) = 0.$$

Вычислим квадрат нормы вектора  $\Phi$ .  $|\Phi|^2 = \Phi\Phi^*$ . Здесь звездочкой обозначено, конечно же, не комплексное сопряжение, а обратный элемент поля  $GF(2^n)$ . Ясно, что мы получим число алгоритмических состояний, соответствующих физическому состоянию  $\Phi(x)$ . То есть степень его вырождения. Понятно так же, что вероятность при измерении обнаружить физическое состояние  $\Phi(x)$  будет пропорциональна этому числу. Таким образом, вектор  $\Phi$  подобно КМ вектору состояния может служить для описания состояния системы. В некотором роде, он является аналогом вектора состояния. Однако, в таком виде, как он представлен в (10), он содержит избыточную информацию, которая недоступна субъективному наблюдателю. Для того чтобы получить средство для описания системы, доступное субъекту, мы должны осуществить переход к субъективному наблюдателю. Прежде всего, заметим, что вектор  $\Phi$ , как и любой другой, это некоторая совокупность компонент. Поэтому можно говорить о некоторых добавочных свойствах этой совокупности, не связанных непосредственно со значением этих компонент. Норма вектора – это одна из таких характеристик.

О векторе  $\Phi$  можно говорить как о функции алгоритмического времени, принимающей последовательно значения  $\xi_1, \xi_2, \xi_3 \dots \xi_n$ . Если мы изменим порядок этой последовательности, осуществив, циклическую перестановку, то тем самым мы изменим фазу функции  $\Phi$ . Другим способом измениться порядок не может, так как состояния последовательны в алгоритмическом времени. Таким образом, еще одной характеристикой является – фаза. Наблюдатель не способен различить значения  $\xi_1, \xi_2, \xi_3 \dots \xi_n$ , соответствующие одному физическому состоянию. Поэтому для него информативны только фаза и норма, которые и определяют функцию  $\psi$  для субъективного наблюдателя:

$$\psi(x) = \begin{pmatrix} |\Phi| \\ \varphi \end{pmatrix}. \quad (11)$$

В наших рассуждениях мы опирались на принцип физической неполноты, но как известно, к пониманию недостаточности действительного поля для адекватного описания физического состояния физики ранее пришли эмпирически, введя в рассмотрение комплексные волновые функции:  $\Phi = R \exp(ikx)$ . То есть каждому чистому состоянию, подобно (11) сопоставляют пары чисел (норма вектора и угол на плоскости Аргана). Теперь смысл такого описания проясняется.

По-видимому, на поле  $GF(2^n)$  справедливы все выводы КМ, однако, это требует строгого доказательства. Мы же, приняв это за гипотезу, просто приведем еще ряд примеров. Например, среднее значение измеряемой величины, так же как и в КМ можно вычислить по формуле:

$$\bar{X} = \Phi X \Phi^*, \quad (12)$$

где  $X$  оператор, подобный квантовомеханическому оператору наблюдаемой. Напоминаем еще раз, что звездочкой здесь обозначены обратные элементы поля  $GF(2^n)$ . По-видимому, "устройство" функции  $\Phi(x)$  обеспечивает изоморфность ее КМ вектору состояния. Так,

если над полем функций  $\Phi(x)$  построить пространство состояний, то оно будет полным аналогом Гильбертова пространства КМ. Любая функция  $\psi(x)$  может быть разложена по некоторому базису в этом пространстве. Например, в базисе собственных функций оператора координаты  $X$ :

$$\psi(x) = \sum_i \Phi(x_i) \delta(x - x_i). \quad (13)$$

Напомним, что механизм измерения квантовая механика не описывает. Проблема решается формально введением проекционных операторов [13]. Любая наблюдаемая  $X$  может быть разложена по операторам проектирования. Например, для четырехмерного Гильбертова пространства, представленного матрицей физических состояний (2) такой оператор будет иметь вид:

$$X = x_1 \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 0 & \\ & & & 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 1 & \\ & & & 0 \end{bmatrix} + x_4 \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & 1 \end{bmatrix}, \quad (14)$$

где  $x_1, x_2, x_3, x_4$  – собственные значения оператора  $X$ . Сами операторы проектирования в сумме образуют единичный оператор. Каждый из них проектирует вектор физического состояния на одно из базовых состояний. Объективный наблюдатель должен уметь описывать механизм, который стоит за тем, что в процессе измерения «срабатывает» только один из проекторов. Напомним, что в настоящий момент это остается за пределами формализма КМ. Полная теория, которая объясняла бы, почему в результате измерения мы получили одно значение, а не другое, по идее, должна различать проекторы друг от друга. То есть должна существовать Булева функция «В», такая, что  $\sum B(P_i) = 1$ , если  $\sum P_i = I$ , где  $P_i$  – разложение единицы  $I$  [14].

Как вы уже поняли, наша модель дает простое решение этой проблемы. По сути дела, измерение и есть проецирование вектора алгоритмического состояния на подпространство физических состояний.

Поддействуем оператором  $X$  на физическое состояние (13), получим уравнение на собственные значения оператора  $X$

$$X \cdot \psi_i(x) = x_i \cdot \Phi(x_i). \quad (15)$$

Здесь  $\Phi(x_i)$  – собственные функции, а  $x_i$  – собственные значения оператора  $X$  в  $x$ -представлении. Зависимость от времени можно перенести на оператор наблюдаемой. Тогда мы получим Гейзенберговское описание. В своем собственном представлении он будет иметь вид:

$$X = \begin{bmatrix} x_1 \delta(\Omega_\xi(t^a) - x_1) & & & \\ & x_2 \delta(\Omega_\xi(t^a) - x_2) & & \\ & & x_3 \delta(\Omega_\xi(t^a) - x_3) & \\ & & & \ddots \end{bmatrix}. \quad (16)$$

Можно представить себе, что единица, задаваемая булевой функцией  $\delta(\Omega \cdot \xi_i - x)$ , «бегает» по главной диагонали матрицы (16) вне физического времени. Для субъективного наблюдателя все выглядит так, как если бы он сфотографировал быстродвижущийся объект с малой выдержкой. Ведь все изменения происходят за нулевой промежуток физического времени! На фотографии этот объект, несмотря на объективно детерминированный характер движения, проявился бы в совершенно непредсказуемом месте, точно так же, как электрон, когда его «фотографируют» в двухщелевом эксперименте.

### Квантовая динамика

Для того, чтобы узнать, как меняются физические состояния во времени, необходимо знать, как меняются алгоритмические состояния. Однако, субъективный наблюдатель этого знать не может. Оператор эволюции системы во времени  $U$ , как обычно, можно записать в виде некоторой линейной по времени добавки к единичному оператору:

$$U = I + H \cdot \Delta t, \quad (17)$$

где  $H$  - аналог гамильтониана.  $I$  - диагональная единичная матрица. Тогда уравнение для  $\Phi(x)$  будет иметь вид:

$$\Phi(x_i, t + dt) = \sum U_{ij} \Phi(x_j, t). \quad (18)$$

Это уравнение по сути ничем не отличается от обычного КМ уравнения, описывающего унитарную эволюцию вектора состояния, за исключением того, что входящие в него величины определены на поле Галуа. В классической КМ это уравнение в зависимости от вида гамильтониана имеет некие периодические решения. Если гамильтонова матрица диагональна, то решение стационарно. В нашем случае, решение уравнения (18) можно получить численно, совершая итерации с шагом, равным скрытому интервалу времени (этот интервал можно назвать квантом времени). Для  $(n,m)$ -модели размерность вектора  $\Phi$  равна  $2^n/2^m$ . Так для  $(7,4)$ -модели вектор  $\Phi$  имеет 8 компонент, а размерность матрицы  $U_{ij}$  будет  $8 \times 8$ . Прежде всего, заметим, что на поле Галуа решение всегда будет периодическим, так как само поле представляет собой циклическую группу. Рассмотрим простой пример. Пусть гамильтонова матрица диагональна. Тогда рекуррентное уравнение (18) даст решение с частотой  $\omega = 1/2^n$ , которую можно отождествить с частотой  $H_{ij}/\hbar$  для стационарного состояния. При наличии недиагональных элементов будет иметь место более сложная динамика вектора состояния, и период будет превышать собственное время осцилляции системы. Физически это означает наличие интерференции между близкими энергетическими уровнями, то есть нестационарность. Рецепт отыскания гамильтониана на конечном поле мы не знаем, но, по всей видимости, это можно сделать, соответствующим образом адаптировав каноническую процедуру.

### Принцип неопределенности и преобразование Фурье над полем Галуа

Рассмотрим источник частиц, генерирующий чистое КМ состояние  $\psi(x_1)$ , являющееся собственным значением оператора координаты. Ему соответствует измеряемая величина " $x_1$ ". Это означает, что попадая на экран, частица всегда будет

засвечивать одну единственную точку "x<sub>1</sub>". Теперь рассмотрим другой источник, генерирующий суперпозицию всех возможных состояний  $\psi(x_1), \psi(x_2), \psi(x_3) \dots$ . Такое квантовое состояние называют импульсным. При попадании на экран такого пучка частиц вся его поверхность окажется засвеченной. КМ в этом случае рассматривает непрерывные спектры собственных функций  $\psi(x)$ , используя аппарат математического анализа, адаптированный для работы на Гильбертовых пространствах с размерностью, равной мощности континуума. Мы и в случае координатного представления рассматриваем счетное множество состояний.

Описанные состояния на континууме связаны линейным преобразованием Фурье:

$$\psi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{ikx} \psi(k) dk . \quad (19)$$

Если  $\psi(x)$  отлична от 0 только при одном значении аргумента  $x=x_1$ , то спектр функции  $\psi(k)$  будет сплошным. Этот факт выражает собой соотношение неопределенности для пары сопряженных величин  $x$  и  $p=hk$ , где  $k$  – волновой вектор или пространственная частота.

В случае дискретного спектра базовые функции  $F_{kx} = e^{\frac{2\pi ikx}{T}}$  образуют квадратную матрицу  $N \times N$ . Соответствующие линейные преобразования пространства квантовых состояний можно записать в матричном виде:

$$X = F_{kx} * K ; \quad K = F_{kx} * X, \quad (20)$$

где  $X$  и  $K$  – вектора (столбцы) значений функций  $\psi(x)$  и  $\psi(k)$  соответственно. Существуют аналогичные выражения и для конечного двоичного пространства над  $GF(2^n)$ .

Рассмотрим в качестве простого примера поле  $GF(2^7)$ . Это поле классов эквивалентных многочленов, дающих при делении на простой многочлен  $x^3+x+1$  следующие остатки:  $\{0, 1, x, x^2, 1+x, x+x^2, 1+x+x^2, 1+x^2\}$ . Обычно для обозначения элементов поля используют порядковые номера  $\{0, 1, 2, 3 \dots 7\}$ . Рассмотрим вектор над этим полем:

$$\psi(x) = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} . \quad (21)$$

Для этого случая матрица дискретного преобразования Фурье [15] имеет вид \*\*\* :

\*\*\* Матрица вычислена средствами программного пакета "Matlab", Смотрите документацию к пакету "Matlab"

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 & 6 & 7 & 5 \\ 1 & 4 & 6 & 5 & 2 & 3 & 7 \\ 1 & 3 & 5 & 4 & 7 & 2 & 6 \\ 1 & 6 & 2 & 7 & 4 & 5 & 3 \\ 1 & 7 & 3 & 2 & 5 & 6 & 4 \\ 1 & 5 & 7 & 6 & 3 & 4 & 2 \end{bmatrix} \quad (22)$$

Преобразование Фурье  $\psi(p)=T\psi(x)$  дает:

$$\psi(p) = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (23)$$

Как и следовало ожидать, мы получили спектр с максимумом на нулевой частоте. Этому тривиальному результату на конечном поле, на континууме соответствует введенная ad hoc обобщенная дельта-функция Дирака:

$$\int_{-\infty}^{+\infty} \exp(-ikx)\delta(x)dx = \psi(k) = const. \quad (24)$$

### Простейшие имитационные эксперименты

(7,4) – модель, которую мы здесь исследуем, конечно, "игрушка" и результаты, которые мы получим, будут носить лишь иллюстративный характер. Тем не менее, эта простейшая модель демонстрирует реальное квантовое поведение. Прежде всего, нам понадобится процедура измерения:

$$f = \text{measur}(a), \quad (25)$$

производящая редукцию вектора алгоритмического состояния "a" к ближайшему физическому состоянию "f".\*\*\*\*

Для задания начальных условий нам будет нужна также процедура, генерирующая по заданному физическому состоянию на входе произвольное алгоритмическое состояние, лежащее в пределах субъективной ошибки наблюдателя на выходе. Эту процедуру можно сконструировать из оператора антиредукции  $\Omega$ , который мы обсуждали выше и специальной рандомизирующей операции "зашумляющей" фазу получившегося вектора алгоритмического состояния таким образом, чтобы ошибка не вывела его за пределы сферы данного физического состояния\*\*\*\*. По сути, эта процедура обратна процедуре

\*\*\*\* С точки зрения информатики эта процедура является декодированием с исправлением ошибок.

\*\*\*\*\* Эта процедура с точки зрения информатики аналогична кодированию сообщения.

измерения "measur". Эта процедура будет имитировать подготовку экспериментатором объекта исследования. Назовем эту процедуру "ameasur":

$$a_0 = \text{ameasur}(f). \quad (26)$$

Описание алгоритмов этих процедур можно найти в приложении. Эта процедура по заданному физическому вектору "f" генерирует некое алгоритмическое состояние "a0", скрытое от наблюдателя. О нем наблюдателю известно только то, что оно лежит в пределах его субъективной ошибки. Если сразу же за процедурой "ameasur" применить процедуру измерения "measur", то с вероятностью 1 мы вернемся к исходному физическому состоянию. Этим оператором, например, можно имитировать испускание одиночных частиц со случайной фазой. Не зная состояния "a0", мы не можем предсказать дальнейшую эволюцию системы, ибо через сферу физического состояния "f" проходит множество непересекающихся алгоритмических траекторий.

В результате, начав с неизвестного стартового состояния "a0", принадлежащего данной сфере, через некоторое время траектория окажется внутри сферы другого физического состояния, которое может быть выявлено только путем измерения. Состояние "a0", конечно, известно нам – объективному наблюдателю, но совершенно недоступно субъективному. В реальном мире такой ситуации соответствуют эксперименты с одиночными фотонами. Место попадания одиночного фотона на экран мы в принципе не способны предсказать, – вероятность зарегистрировать фотон в любой точке "x" на поверхности экрана будет одинакова. Можно сказать, что это наилучший генератор случайных чисел, изобретенный самой природой. Наша модель совершенно адекватно описывает этот случай. Этот эксперимент может быть описан следующей последовательностью процедур – операторов.

$$\left. \begin{array}{l} a_0 = \text{ameasur}(f) \\ a = \text{geval}(a_0, t) \\ f_1 = \text{measur}(a) \end{array} \right\} \text{Программа эксперимента 1.} \quad (27)$$

Эволюционный оператор "geval" описывает движение системы вдоль траектории Грея. В нашем случае, воздействуя на начальное состояние "a0", он возвращает некое алгоритмическое состояние, в которое придет система через время t. Оператор измерения "measur", воздействуя на текущее алгоритмическое состояние, возвращает физическое состояние, которому оно принадлежит. **Таким образом, полностью детерминированный процесс на уровне объективной реальности (в алгоритмическом пространстве) генерирует индетерминированный результат в слое физической реальности.**

## Моделирование интерференции

Что может интерферировать в нашей модели, ведь у нас нет волн? Разве траектории могут интерферировать? Оказывается да, могут! Траектория в конечном двоичном пространстве это ничто иное, как полином над полем Галуа. Очевидно, что сложение любых двух траекторий (полиномов) даст траекторию из того же множества всевозможных траекторий.

Рассмотрим классический двухщелевой эксперимент. На рисунках приведены результаты вычисления по программе 27. Этот случай соответствует одной открытой щели в перегородке.

Программа запускалась по 50 раз для каждого из 128 моментов алгоритмического времени, и вычислялось распределение вероятности физических состояний.

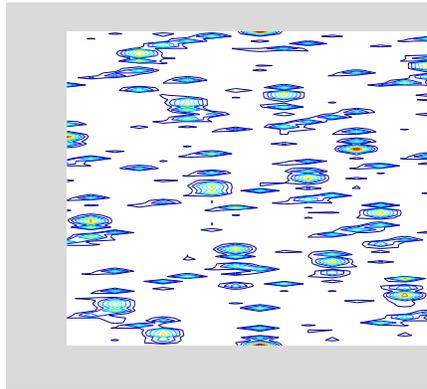


Рис. 5

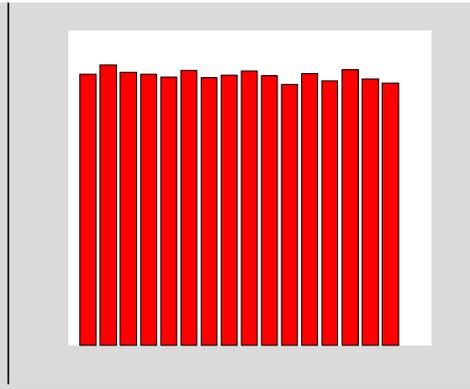


Рис. 6

На рисунке 5 приведена рассчитанная в Matlab контурная диаграмма посещаемости физических состояний во времени. На рисунке 6 функция распределения, усредненная по времени.

Из рисунка 6 видно, что распределение не совсем равномерно. Это вполне естественно и обусловлено несостоятельным характером распределения. Аналогичный эффект флуктуаций можно наблюдать в экспериментах с малыми световыми потоками. В нашей модели это является результатом работы генератора псевдослучайных чисел в операторе "ameasur", при помощи которого приготавливается ансамбль "фотонов".

Далее описан эксперимент с двумя открытыми щелями. Как и в первом примере, приготовим физическое состояние "f" и при помощи оператора "ameasur" произведем от него некое недетерминированное алгоритмическое состояние "a0". Только на этот раз нам понадобятся 2 копии исходного алгоритмического состояния. "a1" и "a2", соответствующие расщеплению пучка на делительной пластинке или на двух открытых щелях интерферометра. На рисунке ниже приведена программа, соответствующая этому случаю:

$$\left. \begin{array}{l}
 a1=ameasur(f) \\
 a2=a1 \\
 a3=geval(a1,t) \\
 a4=geval(a2,t+\delta) \\
 a=XOR(a3,a4) \\
 f1=measur(a)
 \end{array} \right\} \text{Программа эксперимента 2.} \quad (28)$$

Итак, мы имеем 2 начальных идентичных алгоритмических состояния. Предоставим теперь возможность системе развиваться во времени без вмешательства извне. Так как алгоритмическая динамика эргодична, обе алгоритмические точки будут двигаться вдоль

одной и той же траектории с некоторым отставанием, соответствующим разности оптических путей или базе интерференции  $\delta$ . Время  $t$  может меняться от 0 до  $2^n$ . Программа протоколирует следующую последовательность событий. Первый оператор (строка 1) создает на основе исходного физического состояния "f" начальное алгоритмическое состояние "a1", которое затем дублируется в строке 2. (расщепление пучка на два когерентных). В строках 3и 4 полученные состояния "a1" и "a2" подвергается эволюции со сдвигом по фазе на  $\delta$  (движение частиц в пространстве базы интерферометра). Далее (строка 5) полученные алгоритмические состояния складываются на поле Галуа (сложение пучков на второй пластинке или прямо на экране) и возникшая сумма "a" подвергается измерению (строка 6) (измерение света на экране). В результате получаем некое новое физическое состояние "f1" (вспышка на экране). На рисунке приведен результат расчета по этой программе:



Рис. 7

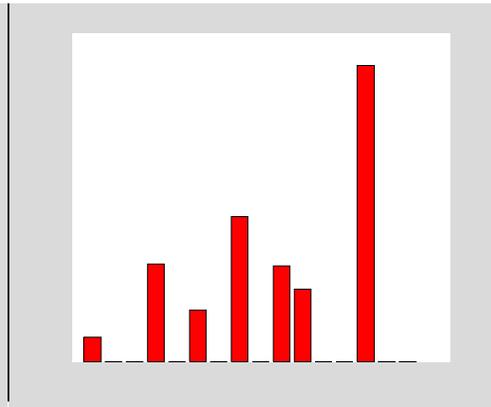


Рис. 8

Видно, что равномерное распределение трансформировалась в вертикальные полосы. Сравните рисунки 6 и 8. Если раньше при измерении в произвольный момент физического времени мы могли с равной вероятностью получить любое из 16 физических состояний, то теперь спектр возможных состояний сузился. В результате деструктивной интерференции некоторые состояния стали невозможны, а вероятность других наоборот возросла. Этот результат, действительно, очень напоминает интерференцию в реальных физических системах.

## Выводы

- В настоящей работе, взяв за основу представление о конечности мира, мы исследовали вопрос о том, как будет выглядеть этот мир "изнутри" То есть с точки зрения наблюдателя, являющегося составной частью этого же мира. Мы показали, что физическая реальность для этого наблюдателя будет неизбежно – квантовой. Мы исходили из того, что методологически, физика - это способ описания природы ее субъектом. Но описание мира субъектом, являющимся его же составной частью, очевидно, не может быть полным. Такое положение вещей мы назвали – физической неполнотой. Оказалось, что физика, построенная для такого субъекта изоморфна квантовой механике. Неустраняемая неполнота нашего знания приводит к физической реальности со свойственной ей квантовой неопределенностью, индетерминизмом и т.д.

- Развиваемую теорию субъективной физики можно классифицировать, как теорию скрытых переменных. Однако, в отличие от теорий такого рода в которых скрытые переменные вводятся формально или ad hoc, у нас они появляются совершенно естественным образом как следствие фундаментальной субъективной неполноты мира.
- Физическая реальность (реальность, данная субъекту) в нашей модели связана с объективной физической реальностью проективным образом. Поэтому для того, чтобы построить субъективную физику необходимо иметь представление о законах, действующих в объективном мире. Но, как мы уже говорили, объективный мир нельзя считать физическим объектом. К решению этого вопроса мы подошли, формально отождествив объективный мир с конечным полем фундаментальных состояний, над которым определили некоторый детерминированный алгоритмический процесс. Согласно нашей модели, физическая неполнота, совершенно естественно приводит к квантовому индетерминизму в слое физической реальности. Поэтому не возникает необходимости постулировать индетерминизм ad hoc.
- В статье показано, что математической основой субъективной физики может быть алгебра и геометрия на конечных полях. В частности, алгебраическая теория кодирования может быть непосредственно использована для описания структуры субъект – объектного мира.
- Методом имитационного моделирования мы изучили простейший нефизический процесс – движение точки в пространстве объективной реальности, подчиняющееся алгоритму, генерирующему последовательность Грея, что соответствует движению по кратчайшему пути в этом пространстве. Нам удалось продемонстрировать, что проекция этого движения на физическое пространство субъективной физической реальности приводит к динамике, напоминающей квантовую.
- В статье мы не преследовали цель построить модель, описывающую физические процессы на количественном уровне. На данном этапе развития теории это просто невозможно. Однако, нетривиальные качественные результаты, полученные нами для простейшей  $(n,m)$  модели, служат дополнительным аргументом, вселяющим в нас уверенность, что субъективная физика, построенная на конечном пространстве, действительно может служить основой для построения более глубокой модели квантовой реальности.

## Приложение

### (Коды процедур)

Здесь приводятся коды процедур, используемых в вычислениях. Вы можете скопировать эти процедуры в виде m-файлов в папку WORK директории MATLAB и поставить собственные эксперименты с физикой на конечном поле.

```
function y=ameasur(x)
% Оператор антиредукции y=ameasur(x)
% y-вектор алгоритмического состояния с субъективной ошибкой
%
```

```
% Copyright Kaminski A.V.
% $Revision: $Date: 2003/09/22 $
n=numel(x);
j=1+fix(n*rand);
if rand < .5
    x(j)=~x(j);
end
y=x;
```

```
function [y, z] = measur(x)
% Оператор редукции
% Преобразовывает алгоритмический вектор n-мерного пространства
% в ближайший (с точностью до субъективной ошибки) вектор физического
% состояния.
% Y-вектор физического состояния
% X-вектор алгоритмического состояния
% n-размерность алгоритмического пространства
%
% Copyright Kaminski A.V.
% $Revision: $Date: 2003/09/22 $
n=numel(x);
H=hadamard(n+1);
H=(H+1)/2;
H=H(2:n+1,:);
H=[H,~H];
for i=1:(n+1)*2;
    R=xor(H(:,i),x);
    if sum(R)<=1 ;
        y=H(:,i);
        z=i;
    end;
end
```

```
function y=geval(x,n)
% функция эволюции. Перемещает алгоритмическую точку на n шагов вдоль
% траектории Грея в m - мерном двоичном пространстве. x - входной вектор, y - выходной.
% Copyright Kaminski A.V.
% $Revision: $Date: 2003/09/22 $
```

```
n1=numel(x);
b=graybin(x);
a=arbit(b);
a=mod(a+n,2^n1);
c=bitarr(a);
yy=bingray(c);
n2=numel(yy);
n=n1-n2;
z((1:n),1)=0;
y=[yy;z];
```

```
function y=bitarr(x)
% Функция целого числа. Возвращает массив коэффициентов разложения этого
% числа по степеням 2. В первом элементе массива - младший разряд.
% Copyright Kaminski A.V.
% $Revision: $Date: 2003/09/22 $
```

```
i=0;
z=x;
```

```

a(1)=0;
while z
    i=i+1;
    r=mod(z,2);
    z=fix(z/2);
    a(i)=r;
end
y=a';

```

```

function y=arrbit(x)
%Функция конвертирует бинарный вектор в число в естественной позиционной
%двоичной системе счисления. Обратная ей bitarr.
% Copyright Kaminski A.V.
% $Revision: $Date: 2003/09/22 $

n=numel(x);
y=0;
for i=1:n;
    y=y+x(i)*2^(i-1);
end

```

```

function y=bingray(x)
%Преобразует двоичный позиционный код в код Грея. x - входной вектор, y -
%выходной. Первый элемент вектора - младший разряд.
% Copyright Kaminski A.V.
% $Revision: $Date: 2003/09/22 $

```

```

m=numel(x);
x(m+1)=0;
for i=1:m
    b(i)=xor(x(i+1),x(i));
end
y=b';

```

```

function y=graybin(x)
%Преобразует код Грея в двоичный позиционный. x - входной вектор, y - выходной.
% Copyright Kaminski A.V.
% $Revision: $Date: 2003/09/22 $

```

```

m=numel(x);
b(m+1)=0;
for i=(m+1):-1:2
    b(i-1)=xor(b(i),x(i-1));
end
y=b(1:m)';

```

**Программа перекодировки физических состояний использовалась для перехода от метрики естественной нумерации к метрике Грея.**

```

F=getphis(7);
for i=1:16
    ff=graybin(F(:,i));
    z(i)=arrbit(ff);
end
Rez=sort(z)
for i=1:16
    for j=1:16
        if z(i)==Rez(j);

```

```
dec(i)=j;
end
end
end
```

**Это случай без интерференции (открыта одна щель)**

```
F=getphis(7);
f= F(:,5);
y=0;
for s=1:128
for n=1:50
a0=ameasur( f);
a1= geval(a0,s);
[Rez,h]=measur(a1);
num(n)=dec(h);
end
y(s+1,1:16)=hist(num,1:16);
end
contour(y)
```

**Это случай интерференции с фиксированной базой.**

```
F=getphis(7);
y=0;
f= F(:,5);
for s=0:128
for n=0:100
a0=ameasur( f);
a1= geval(a0,s);
a2= geval(a0,s+39);
a3= xor(a1,a2);

[Rez,h]=measur(a3);
num(n+1)=dec(h);
end
y(s+1,1:16)=hist(num,1:16);
end
contour(y)
```

## Благодарности

Считаю своим долгом поблагодарить аспиранта Института программных систем Березуева Р.И. за полезные дискуссии на стадии постановки задачи, а также мою супругу Каминскую Н.А. за помощь в редактировании рукописи.

## Литература

1. Kurt Godel, "Uber formal unentscheidbare Satze der Pincipia Mathematica und verwandter Systeme I", Monatshefte fur Mathematik und Physik, 38, 173-198, 1931.
2. R. Feynman, "Simuating Physics with Computers", Department of physics, California Institute of Technology, Pasadena, California 91107.
3. А.В. Каминский "Возможно ли алгоритмическое описание мира"  
<http://www.trinitas.ru/rus/doc/0016/001a/00160077.htm> .

4. P.V. Kurakin, G.G. Malinetsky, Toy quantum mechanics with hidden variables. Keldysh Institute of Applied Mathematics (KIAM) Russian Academy of Sciences Moscow, Miusskaya sq., 4.
5. Д. Дойч, "Квантовая теория, принцип Чёрча-Тьюринга и универсальный квантовый компьютер" // Квантовый компьютер и квантовые вычисления, т. II, Ижевск, 1999, с.157, [http://old.rcd.ru/qc/contents/v99-2\\_r.html](http://old.rcd.ru/qc/contents/v99-2_r.html) .
6. A. Einstein, B. Podolsky, N. Rosen, Phys. Rev. , 1935, v.47, p.7.
7. L.E. Dickson, American Mathematical Monthly, 1911, vol. 18, p.109.
8. <http://www.caravan.ru/~alexch/graphs/Литература#Литература>
9. С.К. Клини, *Введение в математику*, с.378.
10. Э. Берлекемп, *Алгебраическая теория кодирования*.- М.: Мир, 1971.
11. см. например, <http://www.osp.ru/nets/2000/11/022.htm> .
12. В.И. Смирнов, *Курс высшей математики*, т.3 ч.1 стр120.
13. Х. Грин, *Матричная Квантовая механика*, Изд. "Мир", Москва 1968г.
14. L. Brillouin, *Science and Information Theory*, Academic Press Inc., New York, 1956.
15. Lang, Serge, *Algebra*, Third Edition, Reading, Mass., Addison-Wesley, 1993.