Содержание

А.С. нам рассказывает вещи, которые не должны быть утеряны. Каждый раз он рассказывает, каковой (по-видимому) на самом деле должна быть парадигма. Здесь будет записываться «передний фронт разработки», как в программировании. (Таким образом, это не конспект нашего семинара.)

Пожалуйста, **вносите изменения** в этот файл, **дополняйте** его. Пусть он станет вашим рабочим столом. Нет никакого другого способа что-то понять, чем попытаться изложить это листу бумаги/соседу/уточке в ванной. Но у изложения листу бумаги есть два дополнительных бонуса:

- не пропадёт ваш скорбный труд и дум высокое стремленье
- возможность конструктивного фидбека от n слушателей семинара

0 Как вносить изменения в файл

Если вы обнаруживаете, что чего-то не понимаете — предлагаю задать вопрос сноской. Делается это с помощью синтаксиса \que{почему?}. Пример[1] заданного таким образом вопроса.

Если вы обнаруживаете, что что-то понимаете — просто берёте и редактируете текст.

Если у вас не компилится этот tex-файл — закомментируйте строку \usepackage{pscyr} и попробуйте снова; пакет pscyr, содержащий красивые кириллические шрифты, не все себе устанавливали.

В качестве системы контроля версий мы будем использовать GitHub (сейчас это фактически стандарт индустрии).

0.1 Зачем нужна система контроля версий?

Зачем Волга впадает в Каспийское море, а хлоропласты — зелёные?

GitHub основан на Git. Их плюсы:

- У Git не бывает ошибок при синхронизации, в отличие от многих других систем совместной разработки (например, shared Dropbox folder творит ерунду, когда трое или больше человек одновременно работают над файлом)
- Git сохраняет абсолютно все версии отслеживаемых файлов. Каждый раз, когда вы делаете коммит, вы сохраняете в специальной скрытой папке снимок состояния вашего проекта, который можете позже восстановить или с которым можно сравнить текущее состояние.
- Удобно, что проект публичен и имеет короткий, простой и понятный, милый нашему сердцу веб-адрес — github.com/m-theory/book.

Минус у Git/GitHub только один — относительная сложность освоения. Но вас никакие сложности не коснутся, потому что ниже по пунктам расписано, что и как делать.

[1] почему?

0.2 Linux-мануал

0.2.1 При первом использовании

Сначала необходимо установить git. Для этого, как знает каждый Linux-юзер, нужно запустить Terminal/Konsole и установить соответствующий пакет. В Ubuntu 14.04 это делается командой

```
sudo apt-get install git
```

В других дистрибутивах эта команда может выглядеть по-другому, но каждый Linux-пользователь знает, как она выглядит для его дистрибутива.

Далее необходимо сделать клон на своём локальном компьютере текущей версии репозитория m-theory:

```
git clone https://github.com/m-theory/book.git
```

Такой клон называется локальным репозиторием. *(В Git/GitHub у каждого из проектов, над которым работает разработчик, есть по репозиторию; именно в репозиториях хранятся файлы.)*

В результате выполнения такой команды репозиторий будет склонирован в папку ∼/book. Зайдём же в неё.

```
cd book
```

Полюбуемся на файлы в ней:

```
ls
```

Попробуем внести изменение в tex-файл. Откроем его texmaker'ом (или вашим любимым текстовым редактором — vim, nano, kate, gedit, emacs):

```
texmaker paradigm.tex
```

Внесли изменения. Внесли? Отлично.

Давайте полюбуемся на внесённые изменения:

```
git diff paradigm.tex
```

Более чем наглядно, не правда ли? *(Выйти из процесса git:diff можно нажатием клавиши q.)*

Далее необходимо пометить интересные нам файлы как отслеживаемые (добавить их под версионный контроль). Если пометить как отслеживаемые все файлы репозитория командой git add . и затем делать коммиты, то на диске такого разработчика будут сохраняться не только 654 версии файлов paradigm.tex и paradigm.pdf, но и 654 версии таких невероятно полезных файлов, как paradigm.aux, paradigm.log, paradigm.blg, paradigm.toc, (я могу долго перечислять), возникающих во время процедуры PDFLaTeX. Это была бы лишь трата места на жёстком диске. К счастью, в нашем случае мы можем заранее назвать все файлы, которые мы хотим индексировать, так что удобно просто выполнить две команды

```
git add paradigm.tex
```

```
git add paradigm.pdf
```

Что это? Зачем это нужно? Здесь уместно ещё раз проговорить, что Git — это система контроля версий. Командами git add мы создали указатель на текущие версии файлов paradigm.tex и paradigm.pdf. После совершения так называемого коммита в скрытой папке /.git/ будут созданы слепки этих текущих версий. Оттуда Git способен полностью восстановить любую из версий любого из отслеживаемых файлов. Каждый раз, когда вы делаете коммит, вы сохраняете снимок состояния вашего проекта, который можете позже восстановить или с которым можно сравнить текущее состояние.

Пытаемся сделать коммит

```
git commit -m "Внёс небольшие правки; в частности, задал пару вопросов в сносках"
```

но, внезапно, нам выкатывают такую предъяву:

Please tell me who you are.

git config --global user.email "you@example.com"

fatal: unable to auto-detect email address

Действительно, GitHub просит публиковать e-mail'ы тех, кто делает коммиты (=разработчиков). Это необходимо, чтобы, увидев на гитхабе восхитительный проект, заинтересованные лица могли связаться с его авторами. Словом, ничего не поделаешь, придётся послушаться и выполнить команду

```
git config --global user.email "yourmail@xxx.xxx"
```

и теперь всё получится

```
git commit -m "Внёс небольшие правки; в частности, задал пару вопросов в сносках"
```

Сообщение, отражающее суть изменения (commit message), надо писать обязательно!

Ура, теперь текущее состояние файлов, с которыми вы работаете, сохранено в репозиторий. Сделан слепок, из которого вы это текущее состояние сможете в любой момент восстановить.

Последнее, что осталось сделать —

```
git push
```

Здесь вы «толкаете» свой локальный репозиторий на удалённый сервер. Здесь вас спросят username и password. Ваш юзернейм m-theory, ваш пассворд вы знаете (а если не знаете, спросите у кого-нибудь из нас).

Всё! Файлы с изменениями публично доступны по адресу github.com/m-theory/book.

0.2.2 При использованиях последующих

Работа начинается с

```
git pull
```

Здесь отличие. git pull «тянет» удалённый репозиторий (расположенный по адресу github.com/m-theory/book — более точно, по тому адресу, с которого вы в своё время сделали git clone) на локальную машину, при этом локальный репозиторий будет перезаписан. Это необходимо, когда вы не уверены в том, что именно вы внесли последнюю правку в файлы нашего «официального» репозитория github.com/m-theory/book (т.е. необходимо почти всегда).

```
cd book

texmaker paradigm.tex
```

(вносим изменения; необязательный шаг — полюбоваться на них командой git diff paradigm.tex)
(выйти из открывшегося процесса book:git можно нажатием клавиши q)

```
git commit -a -m "Внёс небольшие правки; в частности, задал пару вопросов в сносках"
```

Сообщение, отражающее суть изменения (commit message), надо писать обязательно! Директива -a автоматически делает git add для всех файлов, которые уже являются отслеживаемыми.

Q: Зачем git add, зачем снова добавлять файлы в отслеживаемые?

A: git commit создаёт слепок файлов в том состоянии, в котором они были на момент последнего выполнения git add. Так что следует сначала вносить изменения, и лишь затем выполнять git add filename(s) и git commit. К счастью, последние две команды можно объединить в одну (git commit -a).

Мы не предполагаем, что в проекте появятся другие значимые файлы, помимо paradigm.tex и paradigm.pdf, что появятся другие файлы, которые следует отслеживать. Если они появятся, то непосредственно перед коммитом необходимо сделать одну или несколько команд вида git add newfile.tex.

Осталось лишь отправить результат наших правок на «официальный репозиторий»:

```
git push
```

Здесь всё то же самое: username m-theory, пароль вы знаете.

0.3 Windows-мануал

0.3.1 При первом использовании

Заходим на github.com, листаем вниз, долистываем до кнопки «Download GitHub for Windows». Устанавливаем.

После установки запустится замечательный графический интерфейс. Он скажет «Welcome» и предложит залогиниться. Username наш m-theory, пароль вы знаете (а если не знаете, то спросите у кого-нибудь из нас).

На шаге «Configure» вам просто нужно нажать Continue.

На шаге поиска локальных репозиториев нажать Skip. Всё, вас отпустило, и вам пишут «Get started by adding a repository».

В левом верхнем углу вашего внимания добивается пульсирующая кнопка «+». Жмём на неё, выбираем вкладку Clone, нажимаем на book. Clone book.

Отлично! Вы в раю. Ваши файлы находятся, скорее всего, в папке /Документы/GitHub/book/. Вы можете изменять их там как хотите и копировать в эту папку что хотите.

Как только содержимое этой папки малейшим образом изменится, в приложении GitHub (надеюсь, вы его ещё не закрыли?..) появится оповещение «Uncommitted changes». Нажимаем на Show. Пишем что-нибудь в «Message» и «Description», делаем коммит. После чего замечаем, что справа от кнопки Sync есть ↑1, что как бы намекает нам залить изменения с локальной машины на «официальный» репозиторий github.com/m-theory/book.

Вы восхитительны! Ваши изменения вписаны золотыми буквами в историю github.com/m-theory/book. Просто пройдите туда и взгляните.

0.3.2 При использованиях последующих

Если в публичном репозитории github.com/m-theory/book кипела работа, пока вы наслаждались жизнью, то у кнопки Sync в правом верхнем углу будет ненавязчиво привлекать к себе внимание циферка (количество коммитов, сделанных другими людьми). Кнопку Sync в таком случае неплохо б нажать, чтобы загрузить на свой компьютер результаты работы других людей (самое свежее состояние проекта).

Важный момент, о котором осталось сказать. Если вы компилируете этот tex-файл в той же папке (/Документы/GitHub/book/), у вас постоянно обновляются разные автоматически генерируемые файлы (обычно это paradigm.aux, paradigm.log, paradigm.blg, paradigm.toc и paradigm.synctex.gz). Как уже было сказано, Git сохраняет слепок каждого из файлов, за которым он «следит». Эти слепки хороши в том смысле, что по ним можно восстановить каждую из версий файлов, но в случае .aux, .log, ... файлов они просто засоряют ваш диск. Можно приказать Git не делать слепки этих файлов. Для этого нажимаем на (символизирующую настройки) шестерёнку, выбираем «Repository settings». Выбираем Add a default .gitignore file, НО НЕ нажимаем Update. В этот файл необходимо добавить следующие строки:

```
*.aux
*.blg
*.log
*.synctex
*.synctex.gz
*.toc
*.bbl
*.nav
*.out
*.snm
```

Проследите за тем, чтобы эти строки не начинались со знака # (иначе Git их проигнорирует как комментарий). Теперь нажимаем Update. В чём смысл? В том, что Git теперь будет игнорировать изменения в файлах с такими расширениями, т.е. не будет записывать их слепки на жёсткий диск. Чего и требовалось добиться.

1 A_∞-структуры, Δ_{BV}-оператор, поливекторные поля

1.1 Наша задача — понять, что вместо вопроса

(Пока что наша задача — великолепно переписать этот огрызок.)

Есть нечётные [2] векторные поля $v_н$:

$$v_н \in \oplus_k V^{\otimes k} \to V.$$

И главное уравнение A_∞-структуры [3] имеет вид: $v_н^2 = 0$. Или $\{v, v\}_G = 0$.

Но неправильно говорить просто об этом уравнении. Нужно ещё сфакторизовать по соотношению эквивалентности $v \sim v + \{v, w\}$ — автоморфизмам этого векторного пространства.

$$v = v_0 + Pol, \qquad Pol \ll v_0.$$

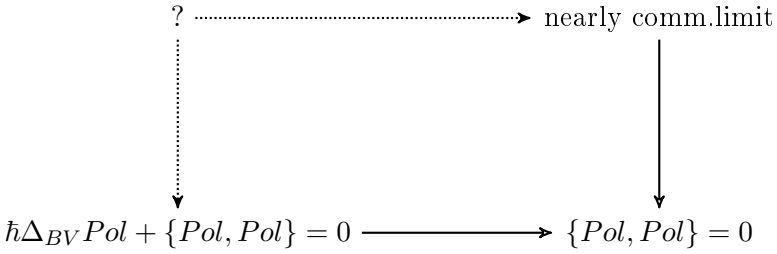
Получится уравнение $\{v_0, Pol\} + \{Pol, Pol\} = 0$.

Далее, можно сказать, что $Pol = P_0 + \omega$. Т.е. $v = v_0 + P_0 + \omega$. Именно ω, этот чёртов третий член, даёт нам когомологии. Первые же два члена этого разложения определяют мир, который рассматривается.

Стоит отметить, что $v_0 : V^{\otimes 2} \to V$ — обычное умножение. Поэтому его обычно обозначают m_2.

Если мы работаем с кольцом $\mathbb{C}[x] \otimes \mathbb{C}[\lambda, \theta]/(\lambda\gamma\lambda)$ (пространственных переменных x 10, чётных полей λ 16, нечётных полей θ тоже 16), а в качестве P_0 мы берём $\lambda^\alpha \frac{\partial}{\partial\theta^\alpha}$, то получится $\mathcal{N} = 1$ $D = 10$ SYM [4]. Низкоэнергетическое приближение — $D = 10$ супергравитация.

Получается такой коммутативный квадрат:



И наша задача — понять, что вместо вопроса.

1.2 Задача матричной факторизации

Точки на нашем пространстве модулей могут подъезжать к границе, и нужно предложить какое-то граничное условие. Граница является браной.

Сама по себе задача матричной факторизации давно известна в теории особенностей и сводится к элементарному утверждению. Именно, нужно решить матричное уравнение на N вида $W = N^2$. $W(x)$ — суперпотенциал, N — матрица, которую необходимо найти.[5] Высший смысл происходящего формулируется так: триангулированные категории особенностей слоёв отображения W эквивалентны категориям B-бран.

[2] Что это такое?

[3] Что это такое?

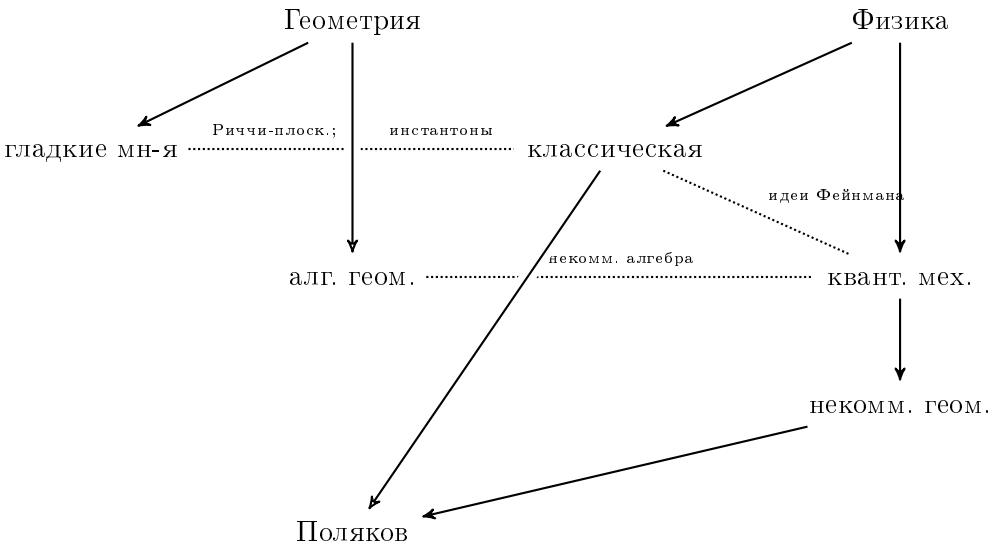
[4] интересно, какая из пяти?..

[5] Тема матриц не раскрыта

2 Новый взгляд на геометрию

2.1 Собственно взгляд

Классическая геометрия является вырождением геометрии Полякова.



Есть две точки зрения на то, что изучает геометрия:

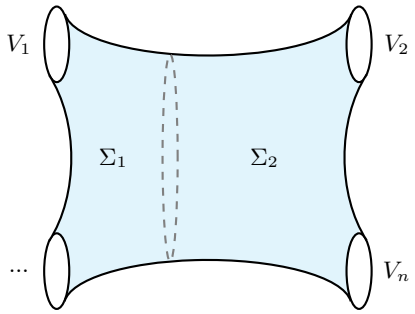
- пространство
- пространство и какая-нибудь геометрическая структура на нём

Согласно Полякову, (гомотопическая) конформная теория — это и есть пространство + геометрия на нём.

2.2 Напоминание о том, что такое конформная теория поля

Конформные теории поля — теории поля, инвариантные относительно конформных преобразований метрики. В них есть «основной объект» I, зависящий от поверхности Σ (вид поверхности зависит от того, открытая или замкнутая струна, а также от количества наблюдаемых $V_1, ..., V_n$ в рассматриваемом коррелляторе). Метрик, сфакторизованных по соотношению эквивалентности «конформно связанные метрики эквивалентны», столько же, сколько комплексных структур, поэтому основной объект I зависит также от комплексной структуры (т.е. от дифференциала Бельтрами μ, который параметризует модули комплексных структур на Σ). Комплексные структуры, очевидно, надо изучать по модулю диффеоморфизмов. Физический смысл нижеследующей картинки вот каков: для того, чтобы вычислить коррелятор операторов V_1, ..., V_n, нужно вырезать малые диски вокруг точек worldsheet'а и вычислять инварианты Громова—Виттена, являющиеся подсчётом композиций кобордизмов комплексно одномерных многообразий[6].

[6] я правильно понимаю, что основной объект $I(\Sigma, V_1, ..., V_n)$ и соответствующий ему инвариант Громова—Виттена — это одно и то же?



Т.е. вот где основной объект принимает значения:

$$I(\Sigma, \mu) \in V_1 \otimes ... \otimes V_n \otimes \Big(\mu(\Sigma)/\mathrm{diff}\Big)$$

и при этом есть единственная аксиома: что если эту поверхность разрезать, то

$$I(\Sigma, \mu) = I(\Sigma_1, \mu) \circ I(\Sigma_2, \mu).$$

Тензор энергии-импульса же можно вытащить из такой теории вариацией основного объекта по дифференциалу Бельтрами:

$$T = \frac{\delta I(\Sigma, \mu)}{\delta \mu}$$

Польчинский пишет действие, которое мы назовём действием старой струнной геометрии:

$$S = \int g_{\mu\nu}(x) dx^\mu * dx^\nu + B_{\mu\nu}(x) dx^\mu \wedge dx^\nu,$$

второй член — поле Калба—Рамона, 2-форма на таргете. Действие новой же струнной геометрии таково:

$$S = \int\limits_\Sigma \left(P_i \bar\partial X^i + \bar P_{\bar i} \partial \bar X^{\bar i} + \boxed{g^{i\bar j}(x, \bar x) P_i P_{\bar j} + \mu^i_{\bar j} P_i \bar\partial \bar X^{\bar j} + \mu^{\bar i}_j P_{\bar i} \partial X^j + b_{i\bar j} \partial X^i \bar\partial \bar X^{\bar j}} \right). \tag{2.1}$$

С помощью комплексной структуры J можно переходить от старой геометрии к новой и, почти всегда, наоборот:

$$(G, B) \overset{J}{\leftrightarrow} (\ g\ ,\ \mu\ ,\ \bar\mu\ ,\ b\ ). \tag{2.2}$$

Поговорим про теорию

$$S = \int\limits_\Sigma P_i \bar\partial X^i.$$

В ней есть поля размерности $(\bullet, 0)$. Это, конечно, функции $f(x) \in (0, 0)$, но также и $(\mathrm{Vect} \oplus \Omega) \in (1, 0)$ — алгебра векторных полей, расширенная своими представлениями. Такие элементы образуют алгебру Ли, назовём её L.

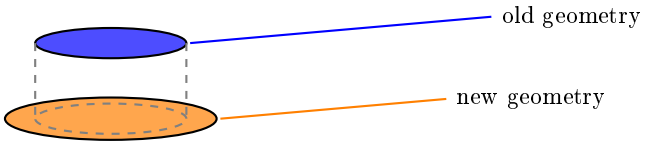
Новая геометрия лежит в $L \otimes \bar L$. Именно, произведём следующее несложное вычисление:

$$(V \oplus \Omega^1) \otimes (\bar V \oplus \bar\Omega^1) =$$

$$= \boxed{V \otimes \bar V} \oplus \boxed{V \otimes \bar\Omega^1} \oplus \boxed{\Omega^1 \otimes \bar V} \oplus \boxed{\Omega^1 \otimes \bar\Omega^1}.$$

Как уже стало понятно из боевой раскраски, $g \in V \otimes \bar V$, $\mu \in V \otimes \bar\Omega^1$, $\bar\mu \in \Omega^1 \otimes \bar V$, $b \in \Omega^1 \otimes \bar\Omega^1$.

Новая геометрия «чуть» больше старой:



Например, в старой геометрии годилась только Риччи-плоская метрика.

Новая геометрия существует на 0-мерных схемах. (А 0-мерные схемы — это по разным причинам хорошо.)

Говоря «схема», мы, в первую очередь, держим в уме следующий пример:

$$\mathbb{C}[x_1, ..., x_n]/I_{F(x)}.$$

Посредством резольвенты Кошуля это эквивалентно $\mathbb{C}[x_1, ..., x_n, \theta]$ с дифференциалом $Q = F\frac{\partial}{\partial\theta}$.

Пространство является гомологическим многообразием с гомологическим векторным полем. (Гомологическое векторное поле — такое векторное поле $Q = v^i(x)\frac{\partial}{\partial x^i}$, что $Q^2 = 0$.) Для вещественно двумерного многообразия с комплексной структурой условие гомологичности векторного поля означает интегрируемость структуры. Деформация многообразия — это деформация гомологического векторного поля.[7]

Традиционного пространства в CFT нет.

Пусть есть семейство CFT_t. Пусть, когда $t \to t_0$, некоторая группа полей неожиданно приобретает размерность 0.

Назовём эти поля $\tilde{\varphi}$. У них есть операторное разложение

$$\tilde{\varphi}_a(t)\tilde{\varphi}_b(0) = c^c_{ab}(z,t)\tilde{\varphi}_c + ...$$

Мы видим, что пространство возникает алгебраически. Возникает как аффинная схема («по Гротендику»), а не как набор дисков, склеенных между собой.

Классическая физика и связанная с ней дифференциальная геометрия умерли. Фейнман как великий контрреволюционер.

Пусть $\gamma \in L \otimes \bar{L}, a \in L, \bar{a} \in \bar{L}$. Определим скобку $[[,]]$:

$$[[a \otimes \bar{a}, b \otimes \bar{b}]] := [a,b]_L \otimes [\bar{a},\bar{b}]_{\bar{L}}$$

Уравнение струнной гравитации, предположительно, выглядит так:

$$(d+Q)(\gamma) + [[\gamma,\gamma]] + \mathcal{O}(\gamma^3) = 0 \tag{2.3}$$

$\gamma \in (g, \mu, \bar{\mu}, b)$, так что это действительно уравнение струнной гравитации (струнной — потому что с полем Калба—Рамона b, гравитации — потому что с метрикой). У этого уравнения есть решения на схемах, которые можно изучать.

То есть, по-видимому, уравнения струнной гравитации имеют вид уравнения Маурера—Картана.

Не так же ли выглядят уравнения М-теории? Этот вопрос, естественно, открыт.

Изучение этого уравнения и его симметрий — это и есть более-менее изучение струнной геометрии пространства-времени.

[7] Тут где-то ещё мимо проходили обобщённые деформации комплексных структур по Баранникову—Концевичу, но, где конкретно, я не понимаю.