



От САДАВРА К САДАВРА II — НОВЫЕ ВОЗМОЖНОСТИ АНАЛИТИЧЕСКИХ ТЕНЗОРНЫХ ВЫЧИСЛЕНИЙ

Кулябов Дмитрий Сергеевич^{1,2}

Семинар «Компьютерная алгебра» факультета ВМК МГУ и ВЦ РАН,
15 февраля 2017

¹Кафедра прикладной информатики и теории вероятностей, Российский университет дружбы народов, Москва, Россия

²Лаборатория информационных технологий, Объединённый институт ядерных исследований, Дубна, Россия

Виды тензорных вычислений

Cadabra

Cadabra II

Исследовательская задача

Виды ТЕНЗОРНЫХ ВЫЧИСЛЕНИЙ

Абстрактные вычисления

- Безындексный формализм. Тензор рассматривается как целостный объект.
- Формализм абстрактных индексов. Тензоры рассматриваются как целостный объект.

Компонентные вычисления

Формализм компонентных индексов. Расчёты проводятся с компонентами тензоров.

ВИДЫ ТЕНЗОРНЫХ ВЫЧИСЛЕНИЙ

ТЕНЗОР КАК ЦЕЛОСТНЫЙ ОБЪЕКТ

Абстрактные вычисления

- Симметрии тензоров.
- Связности.
- Вычисление и упрощение выражений.

Ключевой момент

- Канонизация с учётом симметрий.
- Канонизация с учётом немых индексов.

ВИДЫ ТЕНЗОРНЫХ ВЫЧИСЛЕНИЙ

ТЕНЗОР КАК НАБОР КОМПОНЕНТ

Компонентные вычисления

- Выбор конкретной координатной системы.
- Выбор метрики.
- Вычисление тензоров в данной метрике.

Ключевой момент

- Вычисление компонент как скалярных объектов.
- Как учитывать симметрии?

ВИДЫ ТЕНЗОРНЫХ ВЫЧИСЛЕНИЙ

ТИПЫ СИММЕТРИЙ

Простая перестановочная симметрия (monoterm symmetries)

- Тензор Римана:

$$R_{bacd} = R_{abcd}, \quad R_{cdab} = R_{abcd}.$$

- Присутствует во многих пакетах.

Многоэлементная симметрия (multiterm symmetries)

- Внутренние циклические симметрии:

$$R_{abcd} + R_{acdb} + R_{adbc} = 0.$$

- Тожество Бьянки:

$$\nabla_a R_{bcde} + \nabla_b R_{cade} + \nabla_c R_{abde} = 0.$$

- Тожество Риччи:

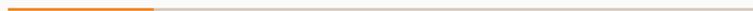
$$\nabla_a \nabla_b v^c = \nabla_b \nabla_a v^c + R_{abd}{}^c v^d.$$

CADABRA

САДАВРА

ГРАФИЧЕСКИЙ ИНТЕРФЕЙС

САДАВРА



РЕАЛИЗОВАНО В САДАВРА

Реализация и интероперабельность

- Монолитная программа.
- Вызов системы *Maxima* из интерфейса *Cadabra*.

Интерфейс пользователя

- Графический интерфейс `xcadabra`.
- Использование \TeX -нотации для ввода и вывода.
- Ковариантные и контравариантные индексы.

Тензорные расчёты

- Абстрактные безындексные вычисления.
- Абстрактные индексные вычисления.
- Простые симметрии (симметричные, антисимметричные).
- Многоэлементные симметрии (тождества Бьянки).
- Пространственные симметрии (тождество Схоутена).
- Диаграммы Юнга.

Другие формализмы

- Спинорные вычисления (вейлевские, майорановские).
- γ -матрицы (произвольной размерности).

САДАВРА



ПРИМЕРЫ

$\{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u\}::\text{Indices}.$

$g_{\{a\ b\}}::\text{Metric}.$

$\backslash\text{partial}_{\{\#\}}::\text{PartialDerivative}.$

```
\nabla := \partial_c\{g_{a b}\} -  
g_{a d}\Gamma^d_{b c} - g_{d b}\Gamma^d_{a c};
```

$$\nabla := \partial_c g_{ab} - g_{ad} \Gamma^d_{bc} - g_{db} \Gamma^d_{ac};$$

```
Gamma:=\Gamma^{a}_{b c} -> (1/2) g^{a d}
( \partial_{b}\{g_{d c}\} + \partial_{c}\{g_{b d}\} -
\partial_{d}\{g_{b c}\} );
```

$$Gamma := \Gamma^a_{bc} \rightarrow \frac{1}{2} g^{ad} (\partial_b g_{dc} + \partial_c g_{bd} - \partial_d g_{bc});$$

`@substitute!(\nabla)(\Gamma);`

$$\nabla := \partial_c g_{ab} - \frac{1}{2} g_{ad} g^{de} (\partial_b g_{ec} + \partial_c g_{be} - \partial_e g_{bc}) \\ - \frac{1}{2} g_{db} g^{de} (\partial_a g_{ec} + \partial_c g_{ae} - \partial_e g_{ac});$$

@distribute!(%);

$$\begin{aligned} \nabla := & \partial_c g_{ab} - \frac{1}{2} g_{ad} g^{de} \partial_b g_{ec} - \frac{1}{2} g_{ad} g^{de} \partial_c g_{be} + \frac{1}{2} g_{ad} g^{de} \partial_e g_{bc} \\ & - \frac{1}{2} g_{db} g^{de} \partial_a g_{ec} - \frac{1}{2} g_{db} g^{de} \partial_c g_{ae} + \frac{1}{2} g_{db} g^{de} \partial_e g_{ac}; \end{aligned}$$

```
@eliminate_metric!!(%);
```

$$\begin{aligned}\nabla := & \partial_c g_{ab} - \frac{1}{2} \partial_b g_{ac} - \frac{1}{2} \partial_c g_{ba} + \frac{1}{2} \partial_a g_{bc} \\ & - \frac{1}{2} \partial_a g_{bc} - \frac{1}{2} \partial_c g_{ab} + \frac{1}{2} \partial_b g_{ac};\end{aligned}$$

```
@canonicalise!(%);
```

$$\begin{aligned} \nabla := & \partial_c g_{ab} - \frac{1}{2} \partial_b g_{ac} - \frac{1}{2} \partial_c g_{ab} + \frac{1}{2} \partial_a g_{bc} \\ & - \frac{1}{2} \partial_a g_{bc} - \frac{1}{2} \partial_c g_{ab} + \frac{1}{2} \partial_b g_{ac}; \end{aligned}$$

```
@collect_terms!(%);
```

$$\nabla := 0;$$

CADABRA



НЕДОСТАТКИ САДАВРА

Недостатки Cadabra

- Отсутствуют компонентные вычисления.
- Низкая интероперабельность.
- Проблемы с расширяемостью.

CADABRA II

CADABRA II

HOBUE B CADABRA II

Реализация и интероперабельность

- Набор библиотек с обвязкой на *python3*.
- Вызов системы *SymPy* из интерфейса *Cadabra II*.
- Интерфейс для пакетов расширения.
- Подпрограммы внутри *Cadabra II*.

Интерфейс пользователя

- Графический интерфейс `cadabra2-gtk`.
- Интерфейс *IPython/Jupyter notebook*.

Тензорные расчёты

- Компонентные индексы.

САДАВРА II

ГРАФИЧЕСКИЙ ИНТЕРФЕЙС



Gamma matrix algebra (Clifford algebras)

Cadabra can do gamma matrix algebra in an arbitrary number of dimensions. The example below shows how to work out the particular gamma matrix contraction

$$\gamma_{\mu}\gamma_{\nu}\gamma_{\lambda}\gamma_{\sigma}$$

which is not impossible to do by hand, but certainly much nicer with Cadabra. As always, we first need to declare the symbols we are going to use.

```
(s, r, l, k, m, n) :: Indices(vector);
(i, r, l, k, m, n) :: Integer(1..4);
gamma_(i) :: GammaMatrix(metr(i+delta));
delta_(i, n) :: KroneckerDelta;

Attached property Indices(position-free) to (s, r, l, k, m, n).
Attached property Integer to (i, r, l, k, m, n).
Attached property GammaMatrix to  $\gamma_i$ .
Attached property KroneckerDelta to  $\delta_{ij}$ .
```

After every step in which we work out the product of two generalised gamma matrices, we need a number of algorithms to simplify the result. We will use the mechanism of defining a `post_process` function, which gets called after every Cadabra function call. Note how every algorithm that is called in here acts on the same expression; you do not have to do things like `ex = sort_product(ex)`.

```
def post_process(ex):
    sort_product(ex)
    eliminate_kronecker(ex)
    canonicalize(ex)
    collect_terms(ex)
```

The expression which we want to simplify reads

```
ex:=gamma_(s r) gamma_(r l) gamma_(k m) gamma_(m s);
--gamma_nu_kappa_nu
```

We need to join and distribute three times to merge all four generalised gamma matrices, so

```
for i in range(3):
    join_gamma(i)
    distribute(i)
    factor_out(i, 405);

gamma(-18d+8(d*d)+12-(d*d*d)+4d(-3+6d-4(d*d)+(d*d*d))

collect_factors(ex);
gamma(-18d+8d^2+12-d^3)+4d(-3+6d-4d^2+d^3)
```

That's the answer we wanted to compute, much easier than by hand.

САДАВРА II

ПРИМЕРЫ

```
{s,r,l,k,m,n}::Indices(vector);  
{s,r,l,k,m,n}::Integer(0..d1);  
\gamma_{#}::GammaMatrix(metric=\delta);  
\delta_{m n}::KroneckerDelta;
```

Attached property Indices(position=free) to (s, r, l, k, m, n) .

Attached property Integer to (s, r, l, k, m, n) .

Attached property GammaMatrix to $\gamma_{(\#)}$.

Attached property KroneckerDelta to $\delta_{(mn)}$.

```
def post_process(ex):  
    sort_product(ex)  
    eliminate_kronecker(ex)  
    canonicalise(ex)  
    collect_terms(ex)
```

```
ex:=\gamma_{s r} \gamma_{r l} \gamma_{k m} \gamma_{m s};
```

$$- \gamma_{mr}\gamma_{lm}\gamma_{ks}\gamma_{rs}$$

```
for i in range(3):  
    join_gamma(_)  
    distribute(_)  
factor_in(_, $d$);
```

$$\gamma_{kl}(-18d + 8(dd) + 12 - (ddd)) + \delta_{kl}(-3 + 6d - 4(dd) + (ddd))$$

```
collect_factors(ex);
```

$$\gamma_{kl} (-18d + 8d^2 + 12 - d^3) + \delta_{kl} (-3 + 6d - 4d^2 + d^3)$$

ИССЛЕДОВАТЕЛЬСКАЯ ЗАДАЧА

ИССЛЕДОВАТЕЛЬСКАЯ ЗАДАЧА

Цели

Цель

Разработка геометрических методов расчёта оптических (электромагнитных) приборов.

ИССЛЕДОВАТЕЛЬСКАЯ ЗАДАЧА

Задачи

Задачи

- Геометрическое описание уравнений Максвелла.
- Геометризация материальных уравнений.
- Лагранжев формализм для уравнений Максвелла.
- Гамильтонов формализм для уравнений Максвелла.
- Методика расчёта оптических приборов.

ИССЛЕДОВАТЕЛЬСКАЯ ЗАДАЧА

МЕТОДЫ И ПОДХОДЫ

Метод

Формализм расслоенных пространств.

Формализм расслоенных пространств

Уравнения Максвелла как структуры на расслоениях.

Метод

Эффективный геометрический формализм.

Варианты геометрий

- Квадратичные геометрии. Риманова геометрия.
- Пространства с кручением.
- Неквадратичные геометрии. Метрика Бервальда–Моора.

Метод

Гамильтонов формализм.

Виды гамильтоновых формализмов

- Симплектические гамильтоновы системы.
- Системы со связями. Формализм Дирака–Бергмана.
- Гамильтонов формализм Гамильтона–де Дондера.
- Многоимпульсный гамильтонов формализм.

ИССЛЕДОВАТЕЛЬСКАЯ ЗАДАЧА

Приложения

Задачи оптики

Прямая задача по параметрам среды получить путь распространения электромагнитных волн.

Обратная задача по заданным путям распространения электромагнитных волн получить параметры среды.

Задачи геометризованной оптики

Прямая задача по заданным путям распространения электромагнитных волн получить параметры среды.

Обратная задача по параметрам среды получить путь распространения электромагнитных волн.

Приложения

- Расчёт линз.
- Расчёт устройств трансформационной оптики (шапка-невидимка).

ИССЛЕДОВАТЕЛЬСКАЯ ЗАДАЧА

ГЕОМЕТРИЗАЦИЯ УРАВНЕНИЙ МАКСВЕЛЛА

Лагранжиан уравнений Максвелла

$$L = -\frac{1}{16\pi c} G^{\mu\nu} F_{\mu\nu} - \frac{1}{c^2} j^\mu A_\mu.$$

Материальные уравнения

$$G^{\mu\nu} = \frac{1}{2} \lambda^{\mu\nu\alpha\beta} F_{\alpha\beta}.$$

Лагранжиан Янга–Миллса уравнений Максвелла

$$L = -\frac{1}{16\pi c} g^{\lambda\mu} g^{\beta\nu} F_{\lambda\beta} F_{\mu\nu} \sqrt{-g} - \frac{1}{c^2} j^\mu A_\mu \sqrt{-g}.$$

Материальные уравнения

$$G^{\alpha\beta} = \sqrt{-g} (g^{\alpha\gamma} g^{\beta\delta} - g^{\alpha\delta} g^{\beta\gamma}) F_{\gamma\delta}.$$

$$\begin{aligned}
 G^{0i} &= \sqrt{-g} (g^{00} g^{ij} g^{0i} g^{0j}) F_{0j} + \sqrt{-g} (g^{0j} g^{ik} g^{0k} g^{ij}) F_{jk}, \\
 G^{ij} &= \sqrt{-g} (g^{i0} g^{jk} g^{0j} g^{ik}) F_{0k} + \sqrt{-g} (g^{ik} g^{jl} g^{il} g^{jk}) F_{kl}.
 \end{aligned}$$

ИССЛЕДОВАТЕЛЬСКАЯ ЗАДАЧА

НЕОБХОДИМЫЕ СООТНОШЕНИЯ

$$g_{\underline{\alpha}\underline{\delta}} g^{\underline{\delta}\underline{\beta}} = \delta_{\underline{\alpha}}^{\underline{\beta}}$$

$$g_{0\underline{\delta}} g^{\underline{\delta}i} = g_{00} g^{0i} + g_{0\underline{k}} g^{\underline{k}i} = \delta_0^i = 0,$$

$$g_{\underline{i}\underline{\delta}} g^{\underline{\delta}j} = g_{\underline{i}0} g^{0j} + g_{\underline{i}\underline{k}} g^{\underline{k}j} = \delta_{\underline{i}}^j.$$

$$\left(g_{\underline{i}\underline{k}} - \frac{1}{g_{00}} g_{0\underline{i}} g_{0\underline{k}} \right) g^{\underline{k}\underline{j}} = \delta_{\underline{i}}^{\underline{j}}.$$

Wer's nicht glaubt, bezahlt einen Taler