Системы алгебраических уравнений

Владимир Борисенко

Мехмат МГУ

vladimir_borisen@mail.ru

Ассоциативное кольцо

Определение кольца

Ассоциативное кольцо — это множество с двумя операциями: сложение и умножение. По сложению оно образует *абелеву группу*, т.е. выполнима операция вычитания. Операции умножения и сложения связаны законом *дистрибутивности*:

$$a \cdot (b + c) = a \cdot b + a \cdot c.$$

В ассоциативных кольцах операция умножения *ассоциативна*:

$$a(bc) = (ab)c.$$

Мы будем рассматривать только *коммутативные кольца*, в которых операция умножения коммутативна: $ab = ba$.

Алгебра

Определение алгебры

Алгебра — это кольцо плюс одновременно и линейное пространство над некоторым полем.

Примеры ассоциативных алгебр: алгебра многочленов от нескольких переменных над полем (можно рассматривать поля рациональных, вещественных, комплексных чисел); алгебра матриц размера $n \times n$ над полем (это некоммутативная алгебра).

Идеал

Определение идеала

Идеалом в кольце/алгебре R называется подкольцо $I \subseteq R$, которое выдерживает умножения на любые элементы кольца:

$$\forall x \in R, i \in I \quad xi \in I.$$

Тот факт, что подмножество I является идеалом в R, обозначается как $I \lhd R$.

В некоммутативном кольце различают левые, правые и двусторонние идеалы, выдерживающие умножения на элементы кольца слева, справа и с обеих сторон; для коммутативного кольца эти понятия совпадают.
Примеры идеалов: в кольце целых чисел $\mathbb{Z}$ это подмножество чисел, которые делятся на фиксированное число m:

$$m\mathbb{Z} = \{mn, n = 0, \pm 1, \pm 2, \dots\}.$$

В кольце многочленов $F[x]$ от одной переменной над полем F любой идеал также состоит из всех многочленов, которые делятся на фиксированный многочлен $g \in F[x]$:

$$I = gF[x].$$

Кольцо целых чисел и кольцо многочленов от одной переменной являются *кольцами главных идеалов*: любой идеал в них порождается одним элементом.

Определение факторкольца

Пусть в кольце/алгебре R есть идеал $I \lhd R$. Тогда можно рассматривать факторкольцо/факторалгебру R/I. Ее элементами являются классы эквивалентности элементов кольца по отношению

$$x \sim y \Leftrightarrow x - y \in I.$$

Пример: кольцо вычетов по модулю m есть факторкольцо кольца целых чисел $\mathbb{Z}$ по идеалу, порожденному элементом $m \in \mathbb{Z}$:

$$\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}.$$

Любая конечно порожденная коммутативная алгебра над полем F есть факторалгебра алгебры многочленов от нескольких переменных по некоторому идеалу:

$$A = F\langle a_1, \ldots a_n\rangle \cong F[x_1, \ldots x_n] \lhd I.$$

Теорема Гильберта о конечности базисов

В кольце многочленов от нескольких переменных над полем $F[x_1, x_2, \ldots, x_n]$ любой идеал порождается конечным числом элементов.

Теореме Гильберта о конечности базисов можно дать следующую интерпретацию. Пусть мы имеем бесконечную систему полиномиальных уравнений от нескольких переменных:

$$g_i(x_1, \ldots, x_n) = 0, \quad g_i \in F[x_1, x_2, \ldots, x_n],$$

где i пробегает бесконечное множество индексов. Тогда в этой системе можно выбрать конечную подсистему уравнений, эквивалентную исходной системе.

Отметим, что вместо системы полиномиальных уравнений всегда можно рассматривать идеал в кольце многочленов, порожденный левыми частями уравнений. Множество решений системы будет совпадать с множество нулей этого идеала, т.е. наборами чисел $x_0, \ldots, x_n$, на которых все элементы идеала обращаются в ноль.

Алгебраическое многообразие

Определение

Аффинным алгебраическим многообразием называется множество решений системы полиномиальных уравнений от нескольких переменных или, что то же самое, множество корней некоторого идеала $I \lhd F[x_1, x_2, \ldots, x_n]$.

Понятие алгебраического многообразия обобщает кривые и поверхности второго порядка, знакомые нам по курсу аналитической геометрии.

Теорема Гильберта о нулях

Замечательная Теорема Гильберта о нулях имеет несколько эквивалентных формулировок, дадим сначала наиболее простую и в то же время наиболее важную формулировку.

Первый вариант теоремы Гильберта о нулях

Пусть $I \lhd \mathbb{C}[x_1, x_2, \ldots, x_n]$ — собственный идеал в кольце многочленов от нескольких переменных над полем комплексных чисел $\mathbb{C}$, отличный от всего кольца многочленов:

$$I \neq \mathbb{C}[x_1, x_2, \ldots, x_n] \Leftrightarrow 1 \notin I.$$

Тогда идеал I имеет корень, т.е. аффинное алгебраическое многообразие, соответствующее идеалу I, не пусто.

Идея доказательства первого варианта теоремы Гильберта о нулях состоит в том, что мы сначала погружаем идеал I в максимальный собственный идеал M: $1 \notin M$, $I \subseteq M$. Факторалгебра $K = \mathbb{C}[x_1, \ldots, x_n]/M$ является полем, и элемент $(\bar{x_1}, \ldots, \bar{x_n})$ является корнем идеала I в поле K. Осталось только показать, что $K = \mathbb{C}$. Это следует из следующей замечательной теоремы, которую мы приведем без доказательства (оно основано на рассмотрении алгебраических расширений полей, ее доказательство восходит еще к Гауссу).

Теорема

Пусть A — конечно порожденная подалгебра над произвольным полем K. Пусть алгебра A и сама также является полем. Тогда A конечномерна как векторное пространство над полем K, т.е поле A является конечным расширением поля K.

Поскольку поле комплексных чисел алгебраически замкнуто, у него не бывает конечных расширений, т.е. в условии теоремы, если $K = \mathbb{C}$, то $A = K$.

Возвращаясь к доказательству первого варианта теоремы Гильберта, имеем

$$I \subseteq M \lhd \mathbb{C}[x_1, \ldots, x_n], \quad M \text{ — максимальный идеал,}$$
$$\mathbb{C}[x_1, \ldots, x_n]/M \cong \mathbb{C} \text{ по указанной теореме,}$$
$$(\bar{x_1}, \ldots, \bar{x_n}) \in \mathbb{C}[x_1, \ldots, x_n]/M \text{ — корень идеала } I.$$

Теперь приведем и докажем традиционную формулировку теоремы Гильберта о нулях.

Традиционный вариант теоремы Гильберта о нулях

Пусть $I \lhd \mathbb{C}[x_1, x_2, \ldots, x_n]$ — собственный идеал в кольце многочленов от нескольких переменных на полем комплексных чисел. Пусть $f \in \mathbb{C}[x_1, \ldots, x_n]$ — некоторый многочлен. Следующие условия эквивалентны:

(i) многочлен f обращается в 0 на всех корнях идеала I;

(ii) некоторая степень многочлена f принадлежит идеалу I:

$$\exists m \in \mathbb{N} : \ f^m \in I.$$

Множество многочленов $\mathrm{Rad}(I) = \{f : \ f^m \in I \text{ для некоторого } m\}$ называется *радикалом идеала* I. Из теоремы Гильберта следует, что радикал идеала I — это в точности идеал, соответствующий алгебраическому многообразию, заданному идеалом I.

Доказательство теоремы Гильберта о нулях

Из того, что $f^m = 0$, вытекает, что и $f = 0$, так что из второго условия следует первое. Нетривиальная часть теоремы состоит в доказательстве импликации (i)$\to$(ii).
Пусть идеал I порождается конечным множеством многочленов $g_1, \ldots, g_k \in I$. Добавим к кольцу многочленов еще одну переменную y и рассмотрим идеал $J \in \mathbb{C}[x_1, \ldots, x_n, y]$, порожденный многочленами

$$J = \mathrm{Ideal}(g_1, \ldots, g_k, 1 - yf).$$

Поскольку по условию теоремы многочлен f обращается в ноль на общих корнях многочленов $g_1, \ldots, g_k \in I$, у идеала J корней нет. Значит, по первому варианту теоремы Гильберта, J совпадает со всем кольцом многочленов, т.е. $1 \in J$. Имеем:

$$1 = g_1 u_1 + \cdots + g_k u_k + (1 - yf)v, \quad u_i, v \in \mathbb{C}[x_1, \ldots, x_n, y].$$

Рассмотрим теперь поле рациональных функций от переменных $x_1, \ldots, x_n, y$ и подставим в это равенство $y = f^{-1}$. Получим

$$1 = g_1 \tilde{u_1} + \cdots + g_k \tilde{u_k}, \quad \tilde{u_i} = u_i|_{y=f^{-1}}.$$

Элементы $\tilde{u_i}$ представляют собой частные двух многочленов от переменных $x_1, \ldots, x_n$, где в знаменателях стоят степени многочлена f. Пусть m — максимальная из этих степеней. Домножив равенство на f^m, получим уже равенство не рациональных дробей, а многочленов:

$$f^m = g_1 w_1 + \cdots + g_k w_k, \quad w_i = f^m \tilde{u_i}.$$

Теорема доказана.

Базис Грёбнера идеала

Рассмотрим алгебру многочленов $F[X]$ от нескольких переменных; через X мы обозначаем множество переменных, $X = \{x_1, x_2, \ldots, x_n\}$.
Упорядочим мономы *лексикографически*: у двух мономов сравниваем сначала их степени по x_1, если они равны, по x_2 и т.д. Например,

$$x_1^3 x_2 > x_1^2 x_3^5, \quad x_1 > x_2^3.$$

Такой порядок называется *чисто лексикографическим*. Применяется еще *однородный лексикографический* порядок, когда сначала сравниваются степени мономов и только потом мономы равной степени сравниваются лексикографически, а также и другие порядки.
Все упорядочения мономов удовлетворяют *условию обрыва убывающих цепей*: не существует бесконечной строго убывающей цепочки мономов $u_1 > u_2 > u_3 > \ldots$.
При зафиксированном порядке мономов у каждого многочлена p можно выделить старший моном $\mathrm{lm}(p)$ (от слов leading monomial), например

$$lm(x_1^3 x_2 x_3 - x_1^2 x_2^4 + x_3^7) = x_1^3 x_2 x_3.$$

Пусть $I \lhd F[X]$ — идеал в алгебре многочленов. Пусть $p \in I$ — многочлен, принадлежащий идеалу, со старшим коэффициентом 1. Представим его в виде

$$p = u - v, \quad \text{где } u \text{ — старший моном } p.$$

Тогда в фактор-алгебре $A = F[X]/I$ справедливо сравнение

$$u \equiv v \pmod{I}.$$

Таким образом, в фактор-алгебре A элемент u можно заменить на v. Поскольку старший моном многочела v строго меньше u, это означает, что в некотором смысле многочлен v устроен проще, чем u.

Определение

Пусть $f = u - v$ — многочлен от нескольких переменных, где u — его старший моном. Пусть w — некоторый моном, который делится на u. *Редукцией* $\mathrm{red}(w, f)$ называется линейное отображение алгебры многочленов в себя, которое в любом многочлене заменяет моном $w = su$ на многочлен sv:

$$\mathrm{red}(w, f) : w = su \mapsto sv.$$

Пусть идеал $I \lhd F[X]$ порождается системой полиномов $J = \{f_1, f_2, \ldots, f_k\}$. Можно рассмотреть множество всевозможных редукций $\mathrm{red}(w, f_i)$ для многочленов $f_i \in J$. Любой многочлен $p \in F[X]$ такими редукциями приводится к форме $\tilde{p}$, в которой все мономы, входящие в его запись, не делятся ни на один из старших членов многочленов $f_i \in J$. Многочлен $\tilde{p}$ называется *нормальной формой* многочлена p.

Определение

Система полиномов $J = \{f_1, f_2, \ldots, f_k\} \subset I$ называется *базисом Грёбнера* идеала $I \lhd F[X]$, если для любого полинома $p \in F[X]$ его нормальная форма $\tilde{p}$ единственна, т.е. не зависит от того, в каком порядке выполнялись редукции при приведении p к нормальной форме.

Эквивалентное определение: J является базисом Грёбнера идеала I тогда и только тогда, когда для любого полинома $g \in I$ его нормальная форма равна нулю: $\tilde{g} = 0$.

Пусть мы имеем систему полиномов $J = \{f_1, f_2, \ldots, f_k\}$. Для каждой пары полиномов $f, g \in J$ можно рассмотреть так называемый *s-полином, или зацепление*. Пусть $f = u_1 - v1$, $g = u_2 - v_2$, где u_1, u_2 — старшие мономы. Пусть u — наименьшее общее кратное мономов u_1, u_2, т.е. $u = s_1 u_1 = s_2 u_2$. Тогда

$$s(f, g) = s_1 v_1 - s_2 v_2.$$

Критерий базиса Грёбнера (бриллиантовая лемма)

Система полиномов $J = \{f_1, f_2, \ldots, f_k\} \subset I$ является базисом Грёбнера тогда и только тогда, когда для любой пары полиномов $f, g \in J$ их зацепление $s(f, g)$ редуцируется к нулю.

Теорема. Для любого идеала $I \lhd F[X]$ может быть построен конечный базис Грёбнера.

Начинаем с любой системы полиномов J, порождающей идеал I. Если какой-то s-полином не редуцируется к нулю, то мы добавляем его нормальную форму к системе полиномов J. Рано или поздно этот процесс оборвется.

Вычисления базисов Грёбнера в SageMath

В системе компьютерной математики SageMath особенно по́лно и удобно реализованы алгебраические вычисления. В частности, многие задачи из коммутативной алгебры и алгебраической геометрии можно решать, вычисляя базисы Грёбнера идеалов в алгебре многочленов от нескольких переменных.
Для задания алгебры многочленов в Sage надо указать основное поле, число и имена переменных, а также порядок на мономах (лексикографический, однородный лексикографический и др.).
Например, следующая команда определяет R как кольцо многочленов над полем комплексных чисел от переменных x, y, z с лексикографическим порядком на мономах:

```
sage: R = PolynomialRing(CC, "x, y, z", order="lex")
sage: R
Multivariate Polynomial Ring in x, y, z over Complex Field
with 53 bits of precision
```

Здесь CC — поле комплексных чисел; варианты QQ — рациональные, RR — вещественные, AA — алгебраические вещественные числа.
Вместо строки с именами переменных можно указать префикс и количество переменных, например

```
sage: R = PolynomialRing(CC, 3, "x", order="lex")
sage: R
Multivariate Polynomial Ring in x0, x1, x2 over Complex Field
with 53 bits of precision
```

Можно также использовать список или кортеж в стиле Python'а с именами переменных:

```
sage: R = PolynomialRing(CC, 3, ["x0", "x1", "x2"], order="lex")
```

Имена переменных — это просто буквы, которые используются при печати. Для получения многочленов, соответствующих переменным, следует использовать метод *gens*() класса *PolynomialRing*, который возвращает кортеж многочленов, представляющих эти переменные. Метод *gen*(i) возвращает многочлен, соответствующий i-й переменной:

```
sage: R = PolynomialRing(CC, 3, "x", order="lex")
sage: R.gens()
(x0, x1, x2)
sage: x0 = R.gen(0)
sage: type(x0)
<class 'sage.rings.polynomial.multi_polynomial_element.
MPolynomial_polydict'>
```

Можно в одной строке определить имя кольца R и порождающих его переменных x_0, x_1, x_2:

```
sage: R.<x0, x1, x2> = PolynomialRing(CC, 3, "x", order="lex")
```

После того, как определены многочлены, представляющие переменные, можно записывать любые многочлены от этих переменных, используя операции +, -, *, а также возведение в степень ^:

```
sage: R.<x0, x1, x2> = PolynomialRing(CC, 3, "x", order="lex")
sage: f = (x0 - 1)^2 + (x1 - 2)^2 + (x2 + 1)^2 - 100
sage: g = (x0 + 1)^2 + (x1 + 3)^2 + (x2 - 1)^2 - 64
sage: h = x0 + x1 + x2
```

Функция *ideal* вычисляет идеал, порожденный набором многочлены, ее аргументами могут быть либо многочлены, перечисленные через запятую, либо список или кортеж. Две следующие строки эквивалентны:

```
sage: I = ideal(f, g, h)
sage: I = ideal([f, g, h])
```

Метод *I.groebner_basis*() вычисляет базис Грёбнера идеала *I*:

```
sage: J = I.groebner_basis()
sage: J
[x0 + 2.333333333333333*x2 - 6.833333333333333,
x1 + (-1.333333333333333)*x2 + 6.833333333333333,
x2^2 + (-5.932432432432432)*x2 + 1.587837837837840]
```

Отметим, что, вычислив базис Грёбнера J идеала I, мы в данном случае сразу можем сказать, что I имеет ровно 2 вещественных корня (корни — это точки 3-мерного пространства). Последний многочлен из J — это квадратный трехчлен, зависящий только от переменной x_2 и имеющий 2 вещественных корня. Подставляя их во второй и первый многочлены, мы найдем соответствующие значения x_1 и x_0.
Базис Грёбнера J идеала I позволяет определить, принадлежит ли произвольный многочлен $t \in R$ идеалу I. Для этого t надо редуцировать к нормальной форме с помощью системы многочленов J. Нормальная форма равна нулю тогда и только тогда, когда $t \in I$. Пример: рассмотрим многочлен $t = f^2 + g + h^3$. Убедимся, что он принадлежит идеалу I:

```
sage: t = f^2 + g + h^3
sage: t.reduce(J)
0
```

А многочлен $s = (f + 1)^2 + g^2$ идеалу I не принадлежит:

```
sage: s = (f+1)^2 + g^2
sage: s.reduce(J)
0.999999999999847
```

Задачи, решаемые с помощью базисов Грёбнера в SageMath

Проверка совместности системы алгебраических уравнений над $\mathbb{C}$

Строим базис Грёбнера идеала, порожденного левыми частями уравнений. Система несовместна тогда и только тогда, когда базис Грёбнера состоит из одной единицы.
Пример. Проверим, совместна ли система

$$\begin{cases} (x-1)^2+(y-1)^2-1=0 \\ ((x-1)^2+(y-1)^2-1)^2-1=0 \end{cases}$$

```
sage: R.<x, y> = PolynomialRing(CC, "x, y")
sage: f = (x - 1)^2 + (y - 1)^2 - 1
sage: g = ((x - 1)^2 + (y - 1)^2 - 1)^2 - 1
sage: ideal(f, g).groebner_basis()
[1.00000000000000]
```

Значит, система несовместна.

Проверка эквивалентности двух систем алгебраических уравнений над $\mathbb{C}$

Пусть мы имеем 2 системы алгебраических уравнений:

$$\begin{cases} f_1 = 0 \\ \dots \\ f_k = 0 \end{cases} \qquad \begin{cases} g_1 = 0 \\ \dots \\ g_s = 0 \end{cases}$$

Пусть $I_1 = \text{ideal}(f_1, f_2, \dots, f_k)$, $I_2 = \text{ideal}(g_1, g_2, \dots, g_s)$. Системы эквивалентны тогда и только тогда, когда всякий многочлен g_i, $i = 1, \dots, s$ обращается в 0 на корнях идеала I_1 и, обратно, всякий многочлен f_j, $j = 1, \dots, k$ обращается в 0 на корнях идеала I_2. Таким образом, задача сводится к следующей: для заданного многочлена g и идеала $I = \text{ideal}(f_1, f_2, \dots, f_k)$ определить, обращается ли g в 0 на всех корнях идеала I.

Из доказательства теоремы Гильберта о нулях вытекает, что эта задача в свою очередь эквивалентна следующей:

определить, содержит ли единицу идеал в кольце многочленов над $\mathbb{C}$, порожденный многочленами

$$f_1, f_2, \dots, f_k, 1 - yg.$$

Здесь y — дополнительная переменная, которую мы добавили к кольцу многочленов.

Пример. Пусть I — идеал, порожденный многочленами

$$f_1 = (x_0 + x_1)(x_0 - x_1) - 1, \quad f_2 = x_0 - 2x_1.$$

Проверим, обращается ли в 0 на корнях идеала I следующий многочлен:

$$g = (x_0^2 - x_1^2)^2 - 1.$$

```
sage: R.<x0, x1> = PolynomialRing(CC, "x0, x1")
sage: f1 = (x0 + x1)*(x0 - x1) - 1
sage: f2 = x0 - 2*x1
sage: g = (x0^2 - x1^2)^2 - 1
sage: R1.<x0, x1, y> = PolynomialRing(CC, "x0, x1, y")
sage: ideal(f1, f2, 1-y*g).groebner_basis()
[1.00000000000000]
```

Таким образом, ответ “да” (многочлен g принадлежит радикалу идеала I).

Если мы хотим написать функцию, которая определяет, принадлежит ли многочлен g радикалу идеала I, порожденному списком многочленов $s = [f_1, f_2, \ldots, f_k]$, то нам необходимо добавить новую переменную к списку переменных кольца многочленов, которому принадлежат многочлены f_i из списка s. Это можно сделать в помощью следующего фрагмента программы на sage:

```
def inRadical(s, g):
    '''Определить, содержится ли многочлен g в радикале
    идеала, порожденного списком многочленов s'''
    f = sum(s)              # Сумма всех многочленов из списка
    R = f.parent()          # Исходное кольцо многочленов
    vars = [str(v) for v in R.gens()] # Список имен переменных
    vars.append('y')        # Добавим 'y' к списку имен переменных
    R1 = PolynomialRing(CC, vars) # Кольцо с добавленной перем.
    y = R1.gens()[-1]       # Многочлен, соотв. y (последней перем.)
```

И теперь можно решить нашу задачу:

```
    s1 = s + [1 - y*g]  # Добавим многочлен 1 - y*g к списку s
    J = ideal(s1).groebner_basis()
    return J == [1.0]
```