УДК 519.6:004.9

Рожков А. В., Филонцева Е. О.

МНОГОЧЛЕН ЭЙЛЕРА И ГЕЛЕНДЖИКСКАЯ ГИПОТЕЗА*

Александр Викторович Рожков

доктор физико-математических наук, профессор

great.rose.marine2@Gmail.com

ФГБОУ ВО «Кубанский государственный университет», Россия, Краснодар

Екатерина Олеговна Филонцева

магистрант факультета математики и компьютерных наук

filontseva99@mail.ru

ФГБОУ ВО «Кубанский государственный университет», Россия, Краснодар

THE EULER POLYNOMIAL AND THE GELENDZHIK HYPOTHESIS

Alexander Viktorovich Rozhkov

Kuban State University, Russia, Krasnodar

Ekaterina Olegovna Filontseva

Kuban State University, Russia, Krasnodar

Аннотация*. Исследование локального расположения простых чисел, проводимое в КубГУ с 2015 г. Поддержано Благотворительным фондом Владимира Потанина. В данной статье исследуются простые числа, связанные с многочленом Эйлера.*

Abstract*. A study of the local location of prime numbers conducted at Kabuga since 2015. Supported by the Vladimir Potanin Charitable Foundation. In this paper, we study the prime numbers associated with the Euler polynomial.*

Ключевые слова: *Теория чисел, язык программирования Julia, функция Эйлера, локальное распределение простых чисел.*

Keywords: *Number theory, Julia programming language, Euler's function, local distribution of prime numbers.*

Цель проекта — проведение разведочных вычислений в области нерешенных проблем алгебры и теории чисел с использованием нового перспективного языка программирования Julia. Официальный сайт https://julialang.org/ текущая версия 1.8.5. Методологией исследований является Геленджикская гипотеза [1]. Проект поддержан Благотворительным фондом Владимира Потанина. Частичные итоги проделанной работы представлены в [2; 3]. В данной статье речь идет о локальном распределении простых чисел, порождаемых многочленом Эйлера.

Введение

Проблема глобального распределения простых чисел решена вполне удовлетворительно и описывается формулой $\pi(n) \approx \frac{n}{\ln(n)-1}$, где $\pi(n)$ — число простых чисел, не превосходящих n. Однако локальное расположение простых чисел, важное для криптографии, не поддается изучению, поскольку нет инструментов для подобных исследований.

Давно замечено, что простые числа склонны к «сгущению». Например, очень популярны исследования близнецов [p, p+2], сдвоенных близнецов [p, p+2, p+6, p+8], плотных шестерок [p, p+4, p+6, p+10, p+12, p+16] и др.

Наибольшее по количеству чисел наиболее плотное сгущение простых чисел, известное в настоящее время (02.02.2023), содержит 21 элемент и имеет вид:

[p+d, d = 0, 2, 8, 12, 14, 18, 24, 30, 32, 38, 42, 44, 50, 54, 60, 68, 72, 74, 78, 80, 84], где p = 39433867730216371575457664399, (https://pzktupel.de/ktuplets.php).

Это p — наименьшее число, всего же их известно лишь 4, — очень мало. Поиск таких чисел возможен только на суперкомпьютерах со специальным программным обеспечением.

Обратим внимание, что p — число 29-значное. Приведенная выше плотная 21-ка расположена на отрезке длины 85. Из этих 85 чисел нечетных ровно 43, таким образом, в этой 21-ке каждое второе нечетное число — простое.

В то же время, в силу формулы глобального распределения простых чисел, в районе 29-значных чисел простым является только каждое 66 число, т. е. каждое 33 нечетное.

Это очень существенное уплотнение простых чисел, даже для таких малых размерностей, как 29-значные числа.

Плотные n-ки очень важны для теории. Например, если существует плотная 447-ка, то она расположена на отрезке меньшей длины, чем отрезок, включающий первые 447 простых чисел в начале координат.

В 2018 г. на Международной конференции по алгебре, которая проводилась в г. Геленджик, нами была предложена методология, как можно искать «сгущения простых чисел».

Геленджикская гипотеза [1]. *Пусть n — натуральное число, зафиксируем его. Пусть M — подмножество четных чисел, содержащее 0, принадлежащее отрезку [0, 2n]. Пусть M не содержит полной системы вычетов ни по какому нечетному простому модулю q < 2n. Тогда существует бесконечно много простых чисел p, таких, что все числа* $\{p + a | a \in M\}$ *являются простыми.*

Многочлен Эйлера

Рассмотрим знаменитый многочлен Эйлера $x^2 + x + 41$, который для $x = 0, 1, 2, \dots, 39$ принимает подряд 40 простых значений. Легко проверить, что по любому простому модулю, меньшему 43, множество M

$$M = \{x^2 + x | x = 0, 1, 2, \dots, 39\} = [0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132,$$
$$156, 182, 210, 240, 272, 306, 342, 380, 420, 462, 506, 552, 600, 650, 702,$$
$$756, 812, 870, 930, 992, 1056, 1122, 1190, 1260, 1332, 1406, 1482, 1560]$$

не содержит полной системы вычетов.

Значит, по Геленджикской гипотезе, существует бесконечно много простых чисел p таких, что p+m простое число для всех m из M.

Нами был проведен обширный вычислительный эксперимент. Вычисления велись на 16 ядерном Intel Core i9 12900k с 64 Гб оперативной памяти.

Компьютер был приобретен на средства гранта Благотворительного фонда Потанина.

Вычисления заняли около 1000 часов, было задействовано до 10 потоков процессора из доступных 24. Были проведены вычисления до $6.3 \cdot 10^{16}$, реально проверено 10 трлн. чисел на простоту.

Следует отметить огромную вычислительную мощь процессора Core i9. Один поток проверяет на простоту 5 млрд. 16-значных чисел в час — это почти 1,5 млн. чисел в секунду.

Конечно, нужного простого числа p, который бы давал 40 простых чисел, мы не смогли найти.

Если такое число и существует, то оно имеет примерно 70–80 знаков в десятичной записи. Но нам удалось найти числа такие, что первые 12, 13, 14, 15 и 16 чисел являются простыми.

Результаты вычислений. *Для чисел меньших* $6.3 \cdot 10^{16}$ *(т.е. до 17-значных чисел включительно) имеют место утверждения:*

1. Пусть M15 = [0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210], тогда все числа множества $\{p + a | a \in M15\}$ *являются простыми, где простое число p принимает 25 значений {291598227841757, 521999251772081, 600274478310161,710383753620701, 1152474895783691, 1170223195594547, 1571972138215091, 3900396511179251, 7565432363941841, 8002271580759497, 11686517862287621, 14230300630910801, 16264557864942701, 19491385050908711, 20852175851423297, 22903600507745801, 28238184261279257, 32049219951869291, 44143478480535671, 48643236534002807, 50353340234905187, 54214679066608211, 54603189310073477, 60917113539762917, 61821409654538201}.*

2. Пусть M16 = [0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240], тогда все числа множества $\{p + a | a \in M16\}$ *являются простыми, где простое число p принимает 4 значения {521999251772081, 19491385050908711, 22903600507746041, 32049219951869531}.*

3. *Ни одно их перечисленных чисел не дает 17 простых чисел с M17.*

Для вычислений мы использовали три несложные программы на языке программирования Julia. Эти программы у нас есть и на языке GAP https://www.gap-system.org/ - GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra, но на Julia они работают в несколько раз быстрее.

Применять программы можно для любого множества M, из Геленджикской гипотезы, которое должно задавать множество простых чисел.

Назовем это множество M ***шаблоном***. Программа Ros1 проверяет все ли числа, входящие в ***шаблон***, являются простыми — на выходе выдает да или нет.

```
using Nemo
function Ros1(M,p)
j = true
for i in M
    if isprobable_prime(ZZ(p+i))
 j=true
       else j=false
break;
    end
end
 return j
end
```

Рисунок 1 — Программа, проверяющая, что все числа p+m простые

Кажется, дальше все просто. Перебираем все простые числа p, а вернее все нечетные числа p, потому что на первом шаге проверяется, является ли число p простым, и задача решена.

Но это ведет к чудовищному количеству лишних вычислений. Дело в том, что число p+m не должно делиться ни на какое простое число q, и значит число p по модулю q имеет столько запретных значений, сколько элементов в шаблоне M. И чем больше мощность шаблона, тем больше чисел-претендентов мы можем отбросить и сократить свои вычисления в десятки, сотни и тысячи раз.

Мы вначале находим допустимые числа по модулю 2, потом из получившихся по модулю 3, т. е. реально по модулю 6, потом по модулю 5, т. е. результирующее по модулю 30 и т.д. При этом используется китайская теорема

об остатках, поэтому нужен пакет Mods, пакет Nemo нужен для проверки на простоту.

Приведем пример уменьшения количества чисел для проверки претендентов, взяв самый важный в этой статье шаблон M15.

По модулю 210 = 2·3·5·7 множество претендентов на простое число p равно 6. То есть нужно проверять только каждое 35-е число.

По модулю 2310 = 210·11 претендентов 30, т.е. проверять нужно каждое 77 число и т. д.

По модулю 6469693230 претендентов почти 2 млн. — 1995840, но проверять нужно только одно число из 3241.

Мы реально вычисляли по модулю 200 млрд., точнее 200560490130. Здесь претендентов почти 32 млн. — 31933440. И мы имеем самый больший выигрыш в уменьшении числа проверок — проверять нужно одно число из 6280.

При этом Julia занимает больше 2 Гб памяти и дальше увеличивать модуль не получится. На следующем шаге по модулю 7 трлн. претендентов будет около млрд. и памяти потребуется больше — 50 Гб. Поэтому операционная система (даже Linux Debian) убивает (официальный термин программирования) прожорливую программу. Но если бы у нас было оперативной памяти 128 Гб, удалось бы проверять только одно число из 10 тыс.

Ниже приведена эта самая важная программа Ros0, которая по шаблону M отбирает претендентов на простое число p по растущим модулям:

```
using Mods
using Nemo
function Ros0(M,m)
l=1; K=[1];D=[];S=[];
Pprime = [2,3,5,7,11,13,17,19,23,29,31,37,41];
for i=1:m
   l= l*Pprime[i]
    L= M.% Pprime[i+1]
     L=sort(unique(L))
      L= setdiff(0:Pprime[i+1]-1,L)
    for j in L
        for k in K
           d=crt(ZZ(k),ZZ(l),ZZ(Pprime[i+1]-j),ZZ(Pprime[i+1]))
              D=push!(D,d)
        end
    end
          K=sort(unique(D))
    D=[]
         end
        S= [K,length(K),l*Pprime[m+1]]
             return(S)
end
```

Рисунок 2 — Программа Ros0 поиска претендентов на простое число p

Стоит отметить одну маленькую, но чрезвычайно важную для вычислений деталь. Массив D, содержащий претендентов на поиск простого числа, с точки зрения математики является одномерным массивом, т. е. вектором. Но вектор — это упорядоченная структура и в Julia его пополнение выполняется командой vcat. Однако использование команды D = vcat(D, d) вместо D = push!(D, d) сильно увеличивает время работы программы: для миллионной размерности вектора D в 1000 раз, для 10 миллионной в 10 тыс. раз, при больших размерностях программа просто останавливается.

Представление данных в языках программирования — наиважнейшая для практики программирования вещь.

И теперь итоговая программа, которая использует шаблон M, массив претендентов S и отрезок [m, n], на котором планируется проводить реальные вычисления.

Кстати, такая программа легко распараллеливается. Просто запускается несколько копий программы на разных отрезках числовой прямой. Что и было нами сделано — на 10 независимых отрезках:

```
using Nemo
function Ros2(S,M,m,n)
  for q in m:n
     for   s in S[1]
                  t= s +S[3]*q
               if Ros1(M,t)
                  println(t,",")
               end
      end
  end
end
```

Рисунок 3 — Итоговая программа Ros2, пригодная для распараллеливания

Результаты

Проведены обширные вычисления в поисках решения классической задачи теории чисел. Предложена методика и инструментарий решения вопросов локального расположения простых чисел. Исследования будут продолжены, в том числе и в связи с многочленом Эйлера.

* Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина

Список литературы

1. *Рожков, А. В.* Автоморфизмы графа вложений сгущений простых чисел / А. В. Рожков, Н. В. Потапова. Текст: непосредственный // Теория групп и ее приложения: материалы XII международной школы конференции по теории групп, посвященной 65-летию А. А. Махнева. Краснодар: Кубан. гос. ун-т, 2018. С. 132–136.

2. *Рожков, А. В.* Экспериментальная математика в КубГУ – первые результаты / А. В. Рожков. Текст: электронный // Новые информационные технологии в образовании и науке: материалы XIV международной научно-практической конференции, Екатеринбург, 1–5 марта 2021 г. Екатеринбург: Рос. гос. проф.-пед. ун-т, 2021. С. 163–172. URL: https://elar.rsvpu.ru/handle/123456789/34886?ysclid=lhuqn38c2f962958690.

3. *Рожков, А. В.* Экспериментальная математика и язык Julia – локальное распределение простых чисел / А. В. Рожков, А. Барсукова. Текст: электронный // Новые информационные технологии в образовании и науке. 2022. № 2 (6). С. 82–88. URL: https://elar.rsvpu.ru/handle/123456789/42071?ysclid=lhuqr6ztpg5814552.