# ЭЛЛИПТИЧЕСКАЯ КРИВАЯ – ВЫЧИСЛЕНИЯ НА ПЛАТФОРМЕ JULIA[1]

## ELLIPTICAL CURVE – JULIA PLATFORM COMPUTING

**А.В. Рожков, Е.О. Филонцева, Р.Л. Репин**

**A.V. Rozhkov, E.O. Filontseva, R.L. Repin**

***Поля Галуа, примитивный элемент, язык программирования Julia, криптография, эллиптическая кривая.***

**В рамках освоения магистерского курса по криптографии производятся вычисления в полях Галуа средствами нового языка программирования Julia.**

***Galois fields, primitive element, Julia programming language, cryptography,*** *elliptic curve.*

**As part of the master's course in cryptography, calculations are made in the Galois fields using the new Julia programming language.**

В рамках реализации проекта, поддержанного грантом фонда Владимира Потанина ГСГК-0072-21, разрабатывается ряд курсов для магистерской программы «Алгебраические методы защиты информации», открытой в Кубанском государственном университете в 2013 г. В данной работе речь идет о курсе «Эллиптическая кривая и электронная подпись». Основой курса является теория эллиптических кривых над полями Галуа.

Julia – высокопроизводительный язык программирования с динамической типизацией, созданный для математических вычислений. Язык имеет встроенную поддержку распределенных и параллельных вычислений. Более того, в код Julia можно включать модули и библиотеки, написанные на языках C/C++, Fortran, Python, Java.

Julia включает в себя множество пакетов, с помощью которых можно производить алгебраические вычисления. Язык активно развивается, и уже имеется 6500 официально принятых расширяющих пакетов и более 10 тыс. еще не сертифицированных. Обширный функционал позволяет использовать экосистему Julia как систему компьютерной алгебры.

Базовые математические пакеты Nemo v0.27.0, AbstractAlgebra v0.22.1, GaloisFields v1.1.1, DarkCurve v0.2.0, LinearAlgebra, Hecke, SymPy v1.0.52. Для построения графиков удобно использовать мощный пакет Plots.

В Windows Julia по умолчанию ставится по адресу C:\Users\user\AppData\Local\Programs\Julia-1.6.3, а расширяющие ее пакеты по адресу C:\Users\user\.julia. При этом, даже если вы не установили ни одного пакета большинство пакетов уже будет на вашем компьютере – их объем больше 7 Gb. Причина в том, что многие пакеты между собой связаны перекрестными ссылками.

Работа с пакетами. Запускаем Julia в терминале REPL. Нажимаем кнопку **“]”** и попадаем в менеджер пакетов `(@v1.6) pkg>`

[1] Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Для добавления, удаления, тестирования пакета, обновления и выяснения статуса пакетов выполняем следующие команды:

```
(@v1.6) pkg> add Nemo
(@v1.6) pkg> rm Nemo
(@v1.6) pkg> test Nemo
(@v1.6) pkg> up
(@v1.6) pkg> st
Status `C:\Users\rosav\.julia\environments\v1.6\Project.toml`
  [eb74ef6d] DarkCurves v0.2.0
  [8d0d7f98] GaloisFields v1.1.1
  [7073ff75] IJulia v1.23.2
  [2edaba10] Nemo v0.27.0
```

Менеджер помощи вызывается клавишей “**?**”

```
help?>
```

Чтобы работать в привычной среде браузера нужно набрать команды

```
julia> using IJulia
julia> notebook()
```

## МОДЕЛЬНЫЕ ЗАДАЧИ

**Задача № 1.** Вычислить количество точек на кривой L, заданной уравнением $y^2 = x^3 + 3x + 8$ над полем GF(199).

**Теорема**. (Хассе) *Если эллиптическая кривая L задана над полем, содержащим q элементов, то число точек на ней удовлетворяет неравенству*

$$|q+1-\#L| \le 2\sqrt{q}\ .$$

В нашем случае у кривой точек будет от 172 до 228.

```
julia> using Nemo
Welcome to Nemo version 0.27.0
Nemo comes with absolutely no warranty whatsoever
julia> function ros(p)
       F=GF(p)
       t=0
          for i in F
              for j in F
                  s= j^2
                  s1= i^3+3*i+8
                     if s == s1
                        t = t+1
                        print("(",i, ",",j, "),")
                     end
                end
              end
         print(t)
       end
ros(199)
```

```
(0,±40),(6,±21),(10,±21),(11,±24),(12,±58),(14,±40),(15,±29),(16,±42),(17,±14
),(18,±83),(19,±77),(21,±24),(22,±37),(29,±87),(33,±5),(35,±2),(36,±87),(38,±
95),(40,±99),(45,±46),(47,±33),(48,±10),(51,±26),(52,±26),(54,±15),(55,±69),(
57,±74),(59,±14),(64,±96),(65,±3),(68,±5),(69,±37),(71,±23),(73,0),(74,±84),(
75,±32),(79,±19),(82,±81),(88,±13),(89,±28),(91,±55),(92,±22),(94,±7),(95,±66
),(96,±26),(98,±5),(99,±9),(102,±97),(106,±39),(108,±37),(110,±25),(111,±65),
(114,±90),(115,±79),(118,±98),(119,±81),(120,±75),(121,±6),(123,±14),(125,±70
),(126,±4),(130,±55),(132,±23),(134,±87),(136,±78),(138,±36),(141,±90),(142,±
50),(143,±90),(144,±35),(146,±53),(149,±32),(150,±47),(151,±48),(152,±11),(15
4,±41),(155,±62),(159,±31),(161,±44),(162,±63),(163,±3),(165,±2),(167,±24),(1
70,±3),(171,±85),(172,±12),(173,±57),(174,±32),(177,±55),(180,±16),(181,±93),
(183,±21),(185,±40),(186,±89),(187,±45),(189,±42),(193,±42),(195,±23),(197,±8
1),(198,±2), 199
```

Получилось 199 решений и плюс бесконечно удаленная точка – всего 200 точек. Очень редкая кривая, у нее ровно q+1 точка.

**Задача № 2.** *Построить график эллиптической кривой.*

Используем полученные результаты. Обратим внимание, что она совсем не похожа на эллиптическую кривую над полем действительных чисел

```
    using Plots
x=[0,6,10,11,12,14,15,16,17,18,19,21,22,29,33,35,36,38,40,45,47,48,51,52,54,
55,57,59,64,65,68,69,71,73,74,75,79,82,88,89,91,92,94,95,96,98,99,102,106,
108,110,111,114,115,118,119,120,121,123,125,126,130,132,134,136,138,141,142,1
43,144,146,149,150,151,152,154,155,159,161,162,163,165,167,170,171,172,173,
174,177,180,181,183,185,186,187,189,193,195,197,198];
y=[40,21,21,24,58,40,29,42,14,83,77,24,37,87,5,2,87,95,99,46,33,10,26,26,15,6
9,74,14,96,3,5,37,23,0,84,32,19,81,13,28,55,22,7,66,26,5,9,97,39,37,25,65,90,
79,98,81,75,6,14,70,4,55,23,87,78,36,90,50,90,35,53,32,47,48,11,41,62,31,44,6
3,3,2,24,3,85,12,57,32,55,16,93,21,40,89,45,42,42,23,8,2];
     data=[y,-y]
      plot(x,data)
     --- savefig("ros.png")
```

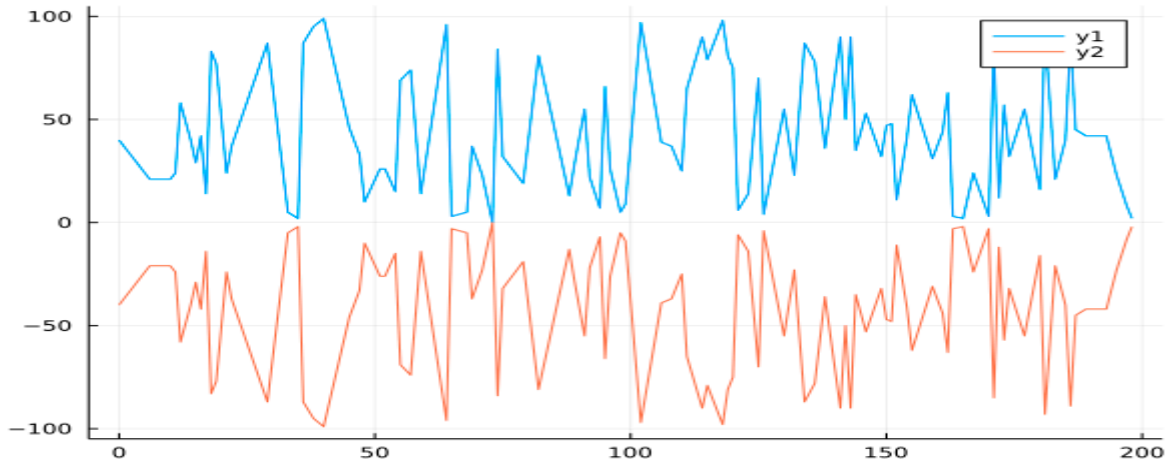


*Рис. 1. График кривой L над полем GF(199)*

**Задача № 3.** *Найти элемент максимального порядка на эллиптической кривой*. Наша кривая $y^2 = x^3 + ax + b$, $P(x_1, y_1) + Q(x_2, y_2) = R(x_3, y_3)$ – ее точки.

Тогда при P = Q $\begin{cases} x_3 = k^2 - 2x_1 \\ y_3 = k(x_1 - x_3) - y_1 \end{cases}, k = \frac{3x_1^2 + a}{2y_1},$

при $P \neq Q$ $\begin{cases} x_3 = k^2 - (x_1 + x_2) \\ y_3 = k(x_1 - x_3) - y_1 \end{cases}, k = \frac{y_2 - y_1}{x_2 - x_1}.$

Поскольку кривая имеет 200 элементов, то порядок максимального элемента может быть 10, 20, 25, 40, 50, 100, 200.

Первая программа – удвоение точки s

```
using Nemo
function rosa(p::Int,s::Vector{gfp_elem})
         F=GF(p)
         a= F(s[1]); b= F(s[2])
         k= (F(3)*a^2+F(3))*(F(2)*b)^(-1)
         a1 = k^2-F(2)*a; b1 = k*(a-a1)-b
   return [a1,b1]
end
```

Вторая – сложение точек s +S

```
function rosA(p::Int,s::Vector{gfp_elem},S::Vector{gfp_elem})
         F=GF(p)
         a= F(s[1]); b= F(s[2]); A= F(S[1]); B= F(S[2])
         if s == S || s==-S
         return F(0)
         else
         k= (B-b)*(A-a)^(-1)
         A = k^2-(a+A)
         B = k*(a-A)-b;
   return [A,B]
end
end
```

Программа нахождения порядка точки s

```
function rosN(p::Int,s::Vector{gfp_elem})
         S= rosa(p,s)
          for i in 1:p
                 S= rosA(p,s,S)
                     if S[1] == s[1]
                   println("s=",s, "->","N=",i+3)
                   break
                   end
            end
           end
julia> rosN(199,[GF(199)(102),GF(199)(97)])
s=gfp_elem[102, 97]->N=100
```

Значит точка [102,97] имеет порядок 100.

Пусть M – это множество точек кривой, без нулевой – они перечислены выше.

```
julia> for m in M
       rosN(199,[GF(199)(m[1]),GF(199)(m[2])])
       end
s=gfp_elem[0, 40]->N=100
s=gfp_elem[6, 21]->N=20
s=gfp_elem[10, 21]->N=200
```

Кривая очень хороша, поскольку как группа является циклической. Точка [10,21] имеет порядок 200. Так как функция Эйлера от 200 равна 80, то точек порядка 200 ровно 80 штук.

**Выводы**

Никакие из вышепроведенных вычислений не могут быть проведены вручную за разумное время. Язык Julia лаконичен и ориентирован на математические вычисления. И может быть применен в любой области математики как хорошее вспомогательное иллюстрационное средство.

## Библиографический список


1. Глухов М.М., Круглов И.А., Пичкур А.Б., Черемушкин А.В. Введение в теоретико-числовые методы криптографии [Электронный ресурс]. СПб.: Лань, 2021. – URL: https://e.lanbook.com/reader/book/153680
2. Рожков А.В. Экспериментальная математика в КубГУ первые результаты // Новые информационные технологии в образовании и науке: материалы XIV междунар. науч.-практ. конф., Екатеринбург, 1–5 марта 2021 г. // ФГАОУ ВО «Рос. гос. проф.-пед. ун-т». Екатеринбург, 2021. С. 163–172. URL: https://www.elibrary.ru/item.asp?id=45825056