ИвГУ, ф-т МиКН, курс 2

"КОМПЬЮТЕРНАЯ АЛГЕБРА"

Тема 6.

Простые элементы

в кольцах квадратичных

целых чисел

Лектор: Н. И. Яцкин, 2014


the fellowship of the ring
the Lord of the Rings
the two towers
the return of the king
www
Themes.
RU
copyright Themes.ru 2001

КОЛЬЦА

КОММУТАТИВНЫЕ
КОЛЬЦА С ЕДИНИЦЕЙ

ЦЕЛОСТНЫЕ КОЛЬЦА

ФАКТОРИАЛЬНЫЕ КОЛЬЦА

КОЛЬЦА ГЛАВНЫХ ИДЕАЛОВ

ЕВКЛИДОВЫ КОЛЬЦА

$$\{\text{К}\} \supseteq \{\text{ККсЕ}\} \supseteq$$

$$\supseteq \{\text{ЦК}\} \supseteq \{\text{ФК}\} \supseteq$$

$$\supseteq \{\text{КГИ}\} \supseteq \{\text{ЕК}\}$$

Определение целостного кольца

ЦК: $[a \cdot b = 0] \Rightarrow [(a = 0) \vee (b = 0)]$.

Целостные кольца "похожи" на кольцо целых чисел $\mathbb{Z}$.

Всякое подкольцо в ***поле*** является ***целостным***. Обратно, для всякого ***целостного*** кольца можно построить его ***поле частных*** ("наименьшее" из полей, содержащих данное кольцо).

Для кольца $\mathbb{Z}$ ***полем частных является*** $\mathbb{Q}$.

Три типа элементов в целостном кольце K:

(1) ***нулевой*** элемент: $0 \in K$;

(2) ***обратимые*** элементы:

$$(a \in K^*) \Leftrightarrow (\exists b \in K)(a \cdot b = 1);$$

(3) ненулевые ***необратимые*** элементы.

Ненулевые ***необратимые*** элементы делятся на **два подтипа**:

(3.1) *неразложимые*;

(3.2) *разложимые*.

Разложимый элемент: представим в виде произведения двух ***необратимых.***

Обратимые элементы образуют ***мультипликативную группу*** K^* кольца K.

Ассоциированность элементов:

$$(a \sim b) \Leftrightarrow (\exists u \in K^*)(b = au)$$

(отличаются обратимым множителем)

Делимость элементов:

$$(a \mid b) \Leftrightarrow (\exists c \in K)(b = ac)$$

Ассоциированность ≡ ***Взаимная делимость***:

$$(a \sim b) \Leftrightarrow (a \mid b) \wedge (b \mid a)$$

НЭ (***неразложимый*** элемент a):

$$[a = b \cdot c] \Rightarrow [(b \text{ обратим}) \vee (c \text{ обратим})]$$

ПЭ (***простой*** элемент a):

$$[a \mid b \cdot c] \Rightarrow [(a \mid b) \vee (a \mid c)]$$

ПЭ $\Rightarrow$ НЭ **ПЭ $\not\Leftarrow$ НЭ**

Определение факториального кольца

ФК: ***Любой ненулевой необратимый элемент однозначно (с точностью до порядка сомножителей и их ассоциированности) разлагается в произведение неразложимых элементов.***

Факториальные кольца "очень похожи" на кольцо целых чисел $\mathbb{Z}$.

Во всяком **ф.к.** ***неразложимость*** эквивалентна ***простоте***:

$$\textbf{ПЭ} \Leftrightarrow \textbf{НЭ}.$$

Определение идеала в кольце

Идеал в (коммутативном) кольце $\boldsymbol{K}$- это подмножество $\boldsymbol{A \subseteq K}$, являющееся подгруппой по сложению:

$$(\boldsymbol{a, b \in A}) \Rightarrow (\boldsymbol{a + b \in A, -a \in A}),$$

и устойчивое относительно умножения на элементы из $\boldsymbol{K}$:

$$(\boldsymbol{k \in K, a \in A}) \Rightarrow (\boldsymbol{k \cdot a \in A}).$$

Среди идеалов выделяются ***главные*:**

$$\boldsymbol{(a) = Ka = \{ka : k \in K\}.}$$

Определение кольца главных идеалов

КГИ: ***Всякий идеал является главным.***

Известные факты:

1. Всякое **к.г.и.** является **ф.к.**
2. Для любых двух элементов **к.г.и.** существует **НОД**, допускающий ***линейное представление***.
3. Кольцо целых чисел $\mathbb{Z}$ является **к.г.и.**

Определение евклидова кольца

ЕК: Заданы ***евклидова норма*** – функция

$$\boldsymbol{\nu : K \setminus \{0\} \to \mathbb{N} \cup \{0\}},$$

позволяющая организовать ***деление с остатком***:

$$\boldsymbol{b = a \cdot q + r}, \text{ где } \boldsymbol{r = 0} \text{ или } \boldsymbol{\nu(r) < \nu(a)}.$$

Известные факты:

1. Всякое **е.к.** является **к.г.и.**, и, следоваетльно, **ф.к.**
2. **НОД**, для элементов **е.к.** можно найти ***алгоритмически***.
3. Кольцо целых чисел $\mathbb{Z}$ является **е.к.**

Примеры *евклидовых* колец:

кольца *квадратичных целых* чисел

Гаусс:

$$\Gamma = \mathbb{Z}[i] = \{z = a + bi : a, b \in \mathbb{Z}\}.$$

Эйзенштейн:

$$(1)\ Э = \mathbb{Z}[\omega] = \{z = \alpha + \beta\omega : \alpha, \beta \in \mathbb{Z}\};\ \omega = \zeta_3\ ;$$

$$(2)\ Э = \{z = \frac{a+ib\sqrt{3}}{2} :\ a, b \in \mathbb{Z}; a \equiv b \bmod 2\}.$$

Пелль:

$$П_2 = \mathbb{Z}[\sqrt{2}] = \{z = a + b\sqrt{2} : a, b \in \mathbb{Z}\}.$$

Евклидовы нормы (следящие функции)

Гаусс:

$$N(z) = a^2 + b^2;\ z = a + bi \in \Gamma.$$

Эйзенштейн:

$$(1)\ N(z) = \alpha^2 - \alpha\beta + \beta^2; z = \alpha + \beta\omega \in Э;$$

$$(2)\ N(z) = \frac{a^2+3b^2}{4};\ z = \frac{a+ib\sqrt{3}}{2} \in Э.$$

Пелль:

$$N(z) = \left|a^2 - 2b^2\right|;\ z = a + b\sqrt{2} \in \Pi_2.$$

Во всех примерах:

$[N(z) = 0] \Leftrightarrow [z = 0]$;

$[N(z) = 1] \Leftrightarrow [z -$ обратимое число$]$;

$N(z \cdot w) = N(z) \cdot N(w)$ (мультипликативность);

$[z \mid w] \Rightarrow [N(z) \mid N(w)]$.

Группы обратимых элементов:

$$\Gamma^* = C_4 = \{1,\ i,\ -1,\ -i\};$$

$$Э^* = C_6 = \{1,\ \zeta_6,\ \zeta_6^2,\ -1,\ \zeta_6^4,\ \zeta_6^5\},$$

где $\zeta_6 = \frac{1}{2}(1 + i\sqrt{3})$;

$$\Pi_2^* = \{\pm\varepsilon^n : n \in \mathbb{Z}\} \cong C_2 \times C,$$

где $\varepsilon = 1 + \sqrt{2}$.

(*C – бесконечная, C_m – конечные циклические группы*)

Критерии неразложимости (≡ простоты):

Гаусс:

$N(z) = p$ – простое натуральное

или

$N(z) = p^2$, где p – простое натуральное, сравнимое с **3 mod 4.**

Эйзенштейн:

$N(z) = p$ – простое натуральное

или

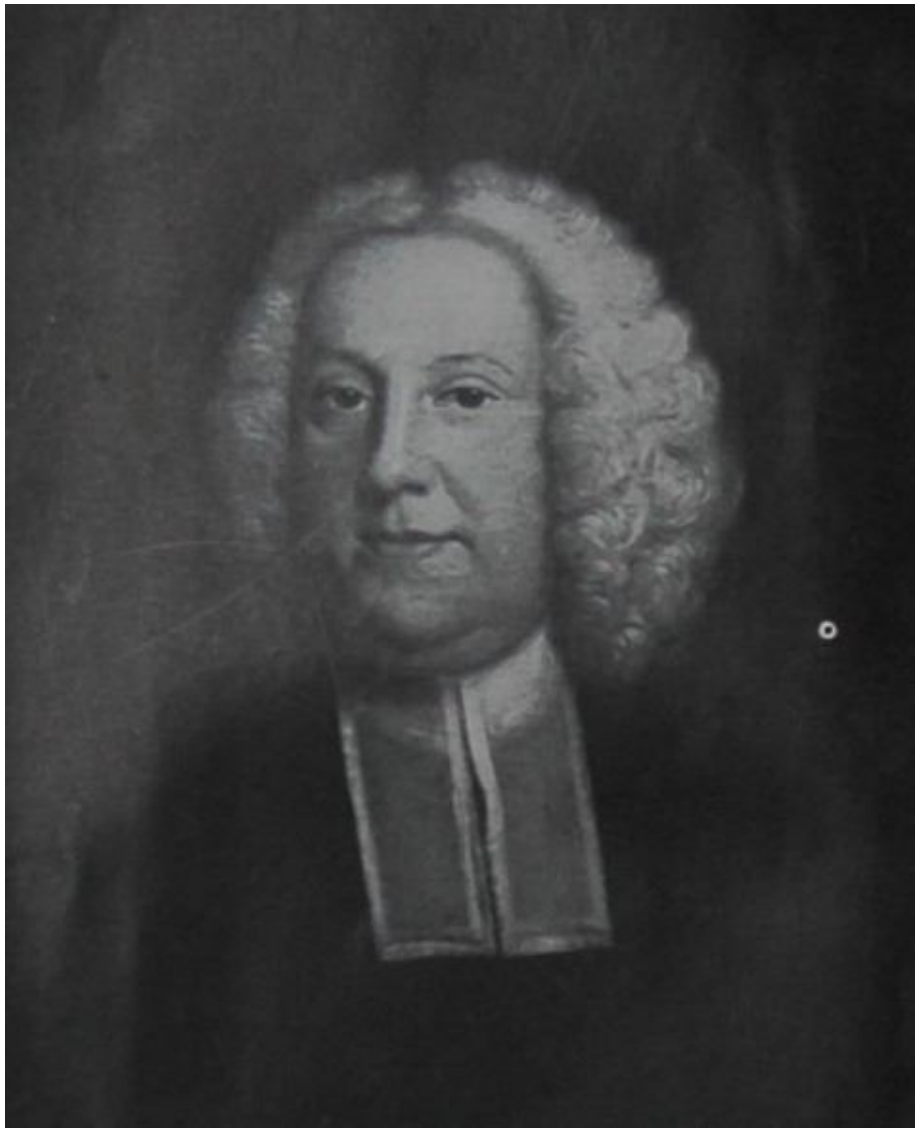
$N(z) = p^2$, где p – простое натуральное, сравнимое с **2 mod 3.**

Пелль:

$N(z) = p$ – простое натуральное

или

$N(z) = p^2$, где p – простое натуральное, сравнимое с **3** или **5 mod 8.**

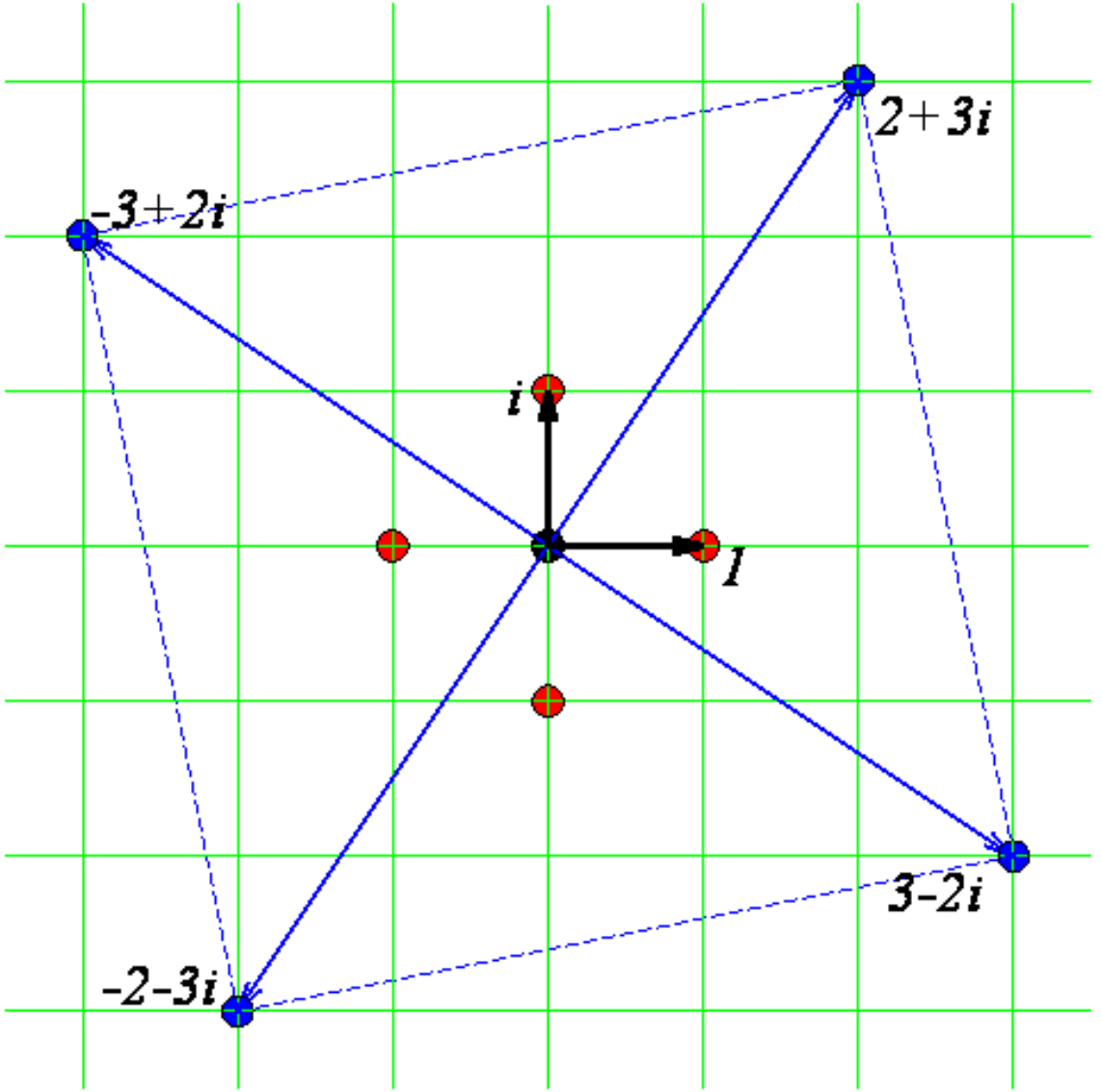


Доказательство приведенных выше фактов
можно найти в учебной литературе:

Айерлэнд К., Роузен М.
Классическое введение в современную теорию чисел.
М.: Мир, 1987. 416 с.

Hardy G. H., Wright E. M.
An Introduction to the Theory of Numbers.
Oxford: Clarendon Press. 1975. 421 p.

Вычисления с числами Гаусса



Задача 1. Представить процедуру **GINORM**, вычисляющую евклидову *норму* целого гауссова числа.

Решение.

```
> GINORM:=proc(z::complex);
if not (type(Re(z),integer) and
          type(Im(z),integer)) then
  ERROR();
end if;
RETURN(Re(z)^2+Im(z)^2);
end proc;
```

```
GINORM := proc(z::complex)
    if not (type(ℜ(z), integer) and type(ℑ(z), integer)) then ERROR( ) end if
    ;
    RETURN(ℜ(z)^2 + ℑ(z)^2)
end proc
```

Пример.

```
> GINORM(1+5*I);
```

$$26$$

Замечание. Можно подгрузить специализированный ***пакет***:

```
> with(GaussInt);
```

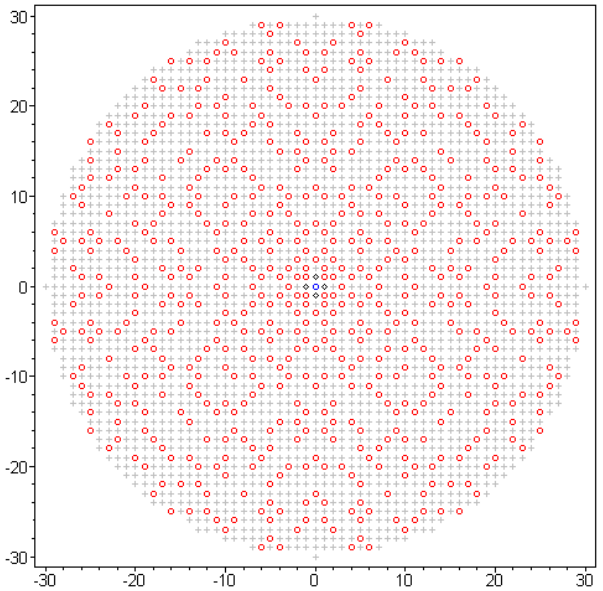
[*GIbasis*, *GIchrem*, *GIdivisor*, *GIfacpoly*, *GIfacset*, *GIfactor*, *GIfactors*, *GIgcd*, *GIgcdex*, *GIhermite*, *GIissqr*, *GIlcm*, *GImcmbine*, *GImod*, *GInearest*, *GInodiv*, *GInorm*, *GInormal*, *GIorder*, *GIphi*, *GIprime*, *GIquadres*, *GIquo*, *GIrem*, *GIroots*, *GIsieve*, *GIsmith*, *GIsqrfree*, *GIsqrt*, *GIunitnormal*]

и воспользоваться "штатной" функцией `GaussInt[GInorm]`.

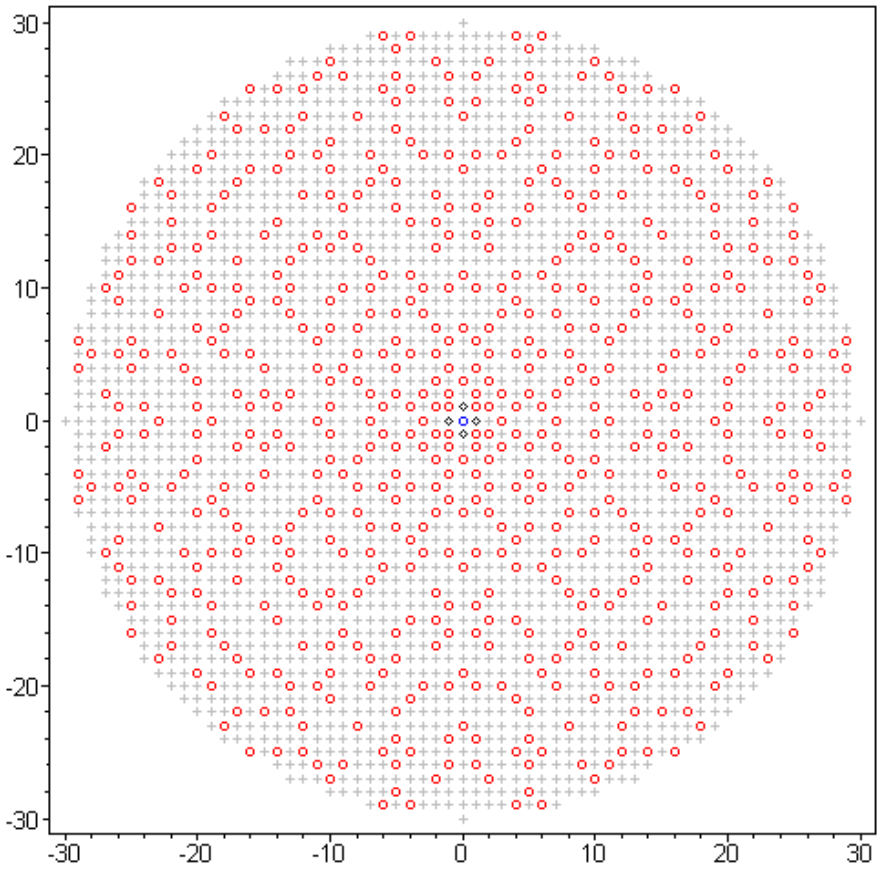
Задача 2. Представить процедуру-тест **GIPRIME** на ***простоту*** целого гауссова числа. (*См. также стандартную версию* **GaussInt[GIprime]**.)

Решение и пример.

```
> GIPRIME:=proc(z::complex)
   local nrm,sqr,ans;
if not (type(Re(z),integer) and
      type(Im(z),integer)) then
   ERROR();
end if;
```



```
ans:=false;
nrm:=GINORM(z);
sqr:=sqrt(nrm);
if type(nrm,prime) or (type(sqr,prime) and
                       sqr mod 4=3) then
  ans:=true;
end if;
RETURN(ans);
end proc;
```



```
GIPRIME := proc(z::complex)
local nrm, sqr, ans;
    if not (type(ℜ(z), integer) and type(ℑ(z), integer)) then ERROR( ) end if
    ;
    ans := false;
    nrm := GINORM(z);
    sqr := sqrt(nrm);
    if type(nrm, prime) or type(sqr, prime) and sqr mod 4 = 3 then ans := true
    end if ;
    RETURN(ans)
end proc
```

```
> map(GIPRIME,[1,2,3,1+5*I]);
```

$$[false, false, true, false]$$

Задача 3. Представить процедуру **`GIFACTORS`**, возвращающую ***сгруппированное*** разложение целого гауссова числа на ***простые множители***.

Указание. *Если* $z \mid w$, *то* $N(z) \mid N(w)$. *Могут понадобиться:* ***(1)*** *вспомогательная процедура, определяющая все гауссовы целые числа (если таковые существуют) по заданной* ***норме***; ***(2)*** *вспомогательная процедура-тест на* ***ассоциированность*** *гауссовых чисел.* (*См. также стандартную версию* **`GaussInt[GIfactors]`**.)

Задача 4. Представить графическую процедуру **`GIpict`**, возвращающую изображение фрагмента комплексной плоскости, содержащего все гауссовы целые числа с нормой, не превышающей N^2, расклассифицированные по категориям:

- ***нуль;***
- ***обратимые числа (единицы);***
- ***простые числа;***
- ***составные числа.***

Указание. *Использовать графические пакеты:*

`plots, plottools.`

Решение. *Загрузка пакетов:*

```
> with(plots);with(plottools);
```

[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *densityplot*, *display*, *dualaxisplot*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *graphplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *interactive*, *interactiveparams*, *intersectplot*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *multiple*, *odeplot*, *pareto*, *plotcompare*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedra_supported*, *polyhedraplot*, *rootlocus*, *semilogplot*, *setcolors*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

[*arc*, *arrow*, *circle*, *cone*, *cuboid*, *curve*, *cutin*, *cutout*, *cylinder*, *disk*, *dodecahedron*, *ellipse*, *ellipticArc*, *hemisphere*, *hexahedron*, *homothety*, *hyperbola*, *icosahedron*, *line*, *octahedron*, *parallelepiped*, *pieslice*, *point*, *polygon*, *project*, *rectangle*, *reflect*, *rotate*, *scale*, *semitorus*, *sphere*, *stellate*, *tetrahedron*, *torus*, *transform*, *translate*]

Понадобятся: **`plots[display]`**, **`plottools[point]`**.

```
> GIpict:=proc(N::posint)
  local pts,a,b,nrm;

# Заготовка для списка "геометрических точек".
pts:=[];
# Далее следует просмотр заданной области.
for a from -N to N do
  for b from -N to N do
    nrm:=GINORM(a+b*I);
    if nrm<=N^2 then
```

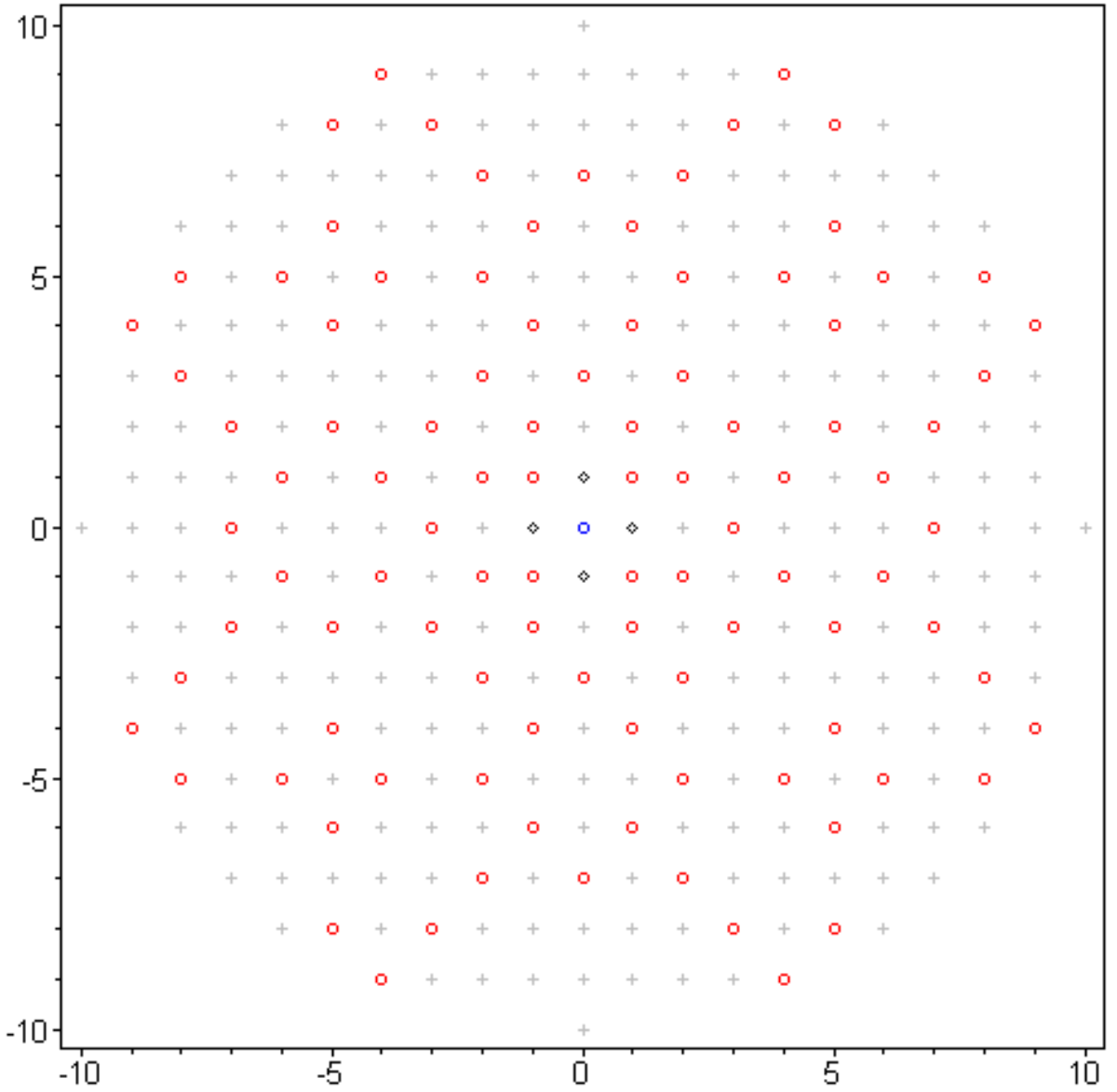
```
      if nrm=0 then
# Задается изображение нулевого элемента.
        pts:=[pts[],point([a,b],
                          symbol=CIRCLE,color=BLUE)];
      elif nrm=1 then
# Задаются изображения обратимых элементов.
        pts:=[pts[],point([a,b],
                         symbol=DIAMOND,color=BLACK)];
      elif GIPRIME(a+b*I) then
# Задаются изображения простых элементов.
        pts:=[pts[],point([a,b],
                          symbol=CIRCLE,color=RED)]:
      else
# Задаются изображения составных элементов.
        pts:=[pts[],point([a,b],
                          symbol=CROSS,color=GRAY)];
```

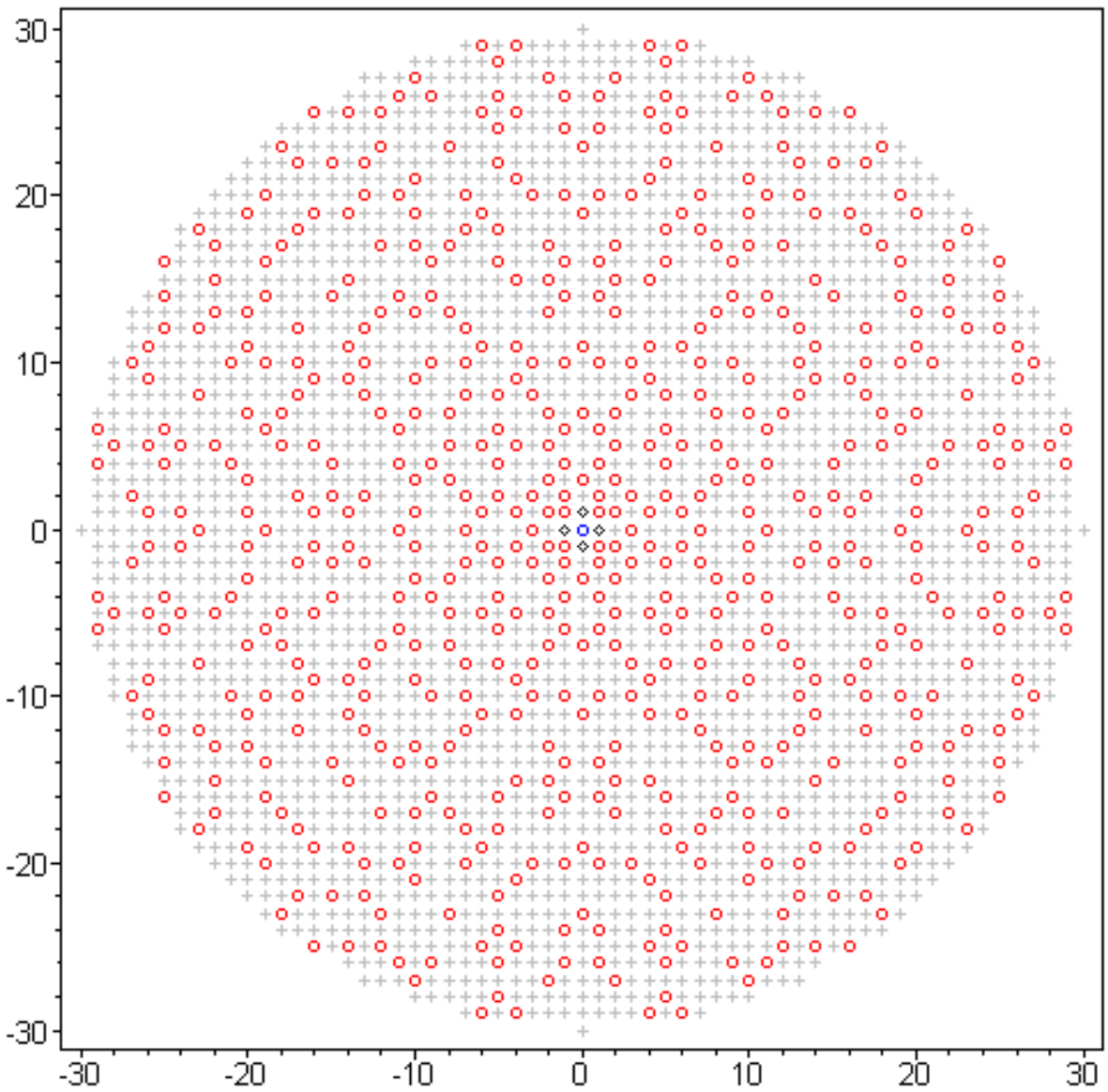
```
      end if;
    end if;
  end do;
end do;

# Отображаются все просмотренные точки.
display(pts,axes=BOXED);
end proc;
```

```
GIpict := proc (N::posint)
local pts, a, b, nrm;
      pts := [ ];
      for a from −N to N do for b from −N to N do
                nrm := GINORM( a + b×I );
               if nrm ≤ N^2 then
                    if nrm = 0 then pts := [pts[ ],
                          plottools:-point( [a, b ], symbol = CIRCLE, color = BLUE ) ]
                    elif nrm = 1 then pts := [pts[ ], plottools:-point( [a, b ],
                          symbol = DIAMOND, color = BLACK ) ]
                    elif GIPRIME( a + b×I ) then pts := [pts[ ],
                          plottools:-point( [a, b ], symbol = CIRCLE, color = RED ) ]
                    else pts := [pts[ ],
                          plottools:-point( [a, b ], symbol = CROSS, color = GRAY ) ]
                    end if
               end if
          end do
     end do ;
     plots:-display( pts, axes = BOXED )
end proc
```

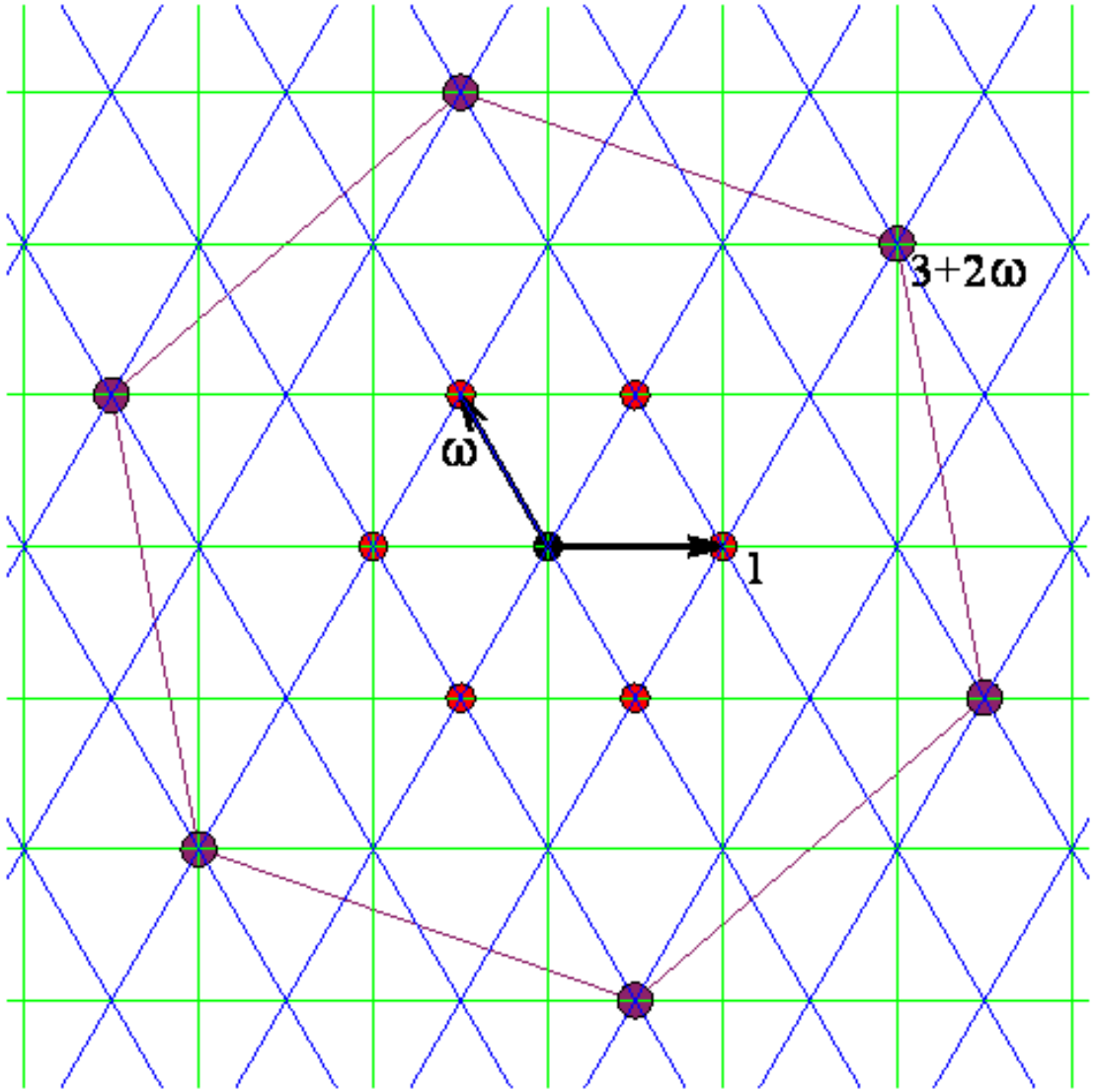
Рисунки 1 - 2. Числа с нормой, не превышающей **100** и **900**.

```
> GIpict(10);
   GIpict(30);
```




30
20
10
0
-10
-20
-30
-30
-20
-10
0
10
20
30

Вычисления с числами Эйзенштейна



*Выберем **второй способ** представления чисел Эйзенштейна:*

$$Э = \{z = \frac{a+ib\sqrt{3}}{2} : \ a, b \in \mathbb{Z}; a \equiv b \bmod 2\}.$$

*Избегая вычислений с радикалами зададим **кодировку** элементов кольца Э списками вида* $z = [a, b]$, где a, b *– целые числа **одинаковой четности**. (**Внимание! Единица** кодируется так:* **[2,0].**)

Нам понадобятся вспомогательные функции:

`EIcompl` - *переход от кодировки элементов Э списками к **радикальной** записи;*

`EIpt` - *переход от кодировки списками к заданию координат **геометрических** точек по типу* **`` `float` ``** (*с плавающей точкой; см. функцию* **`evalf`**).

Задача 5. Представить процедуру **EIcompl**, конвертации *списка* в *радикальную* запись числа Эйзенштейна.

Решение.

```
> EIcomp:=proc(z::list);
if not (nops(z)=2 and type(z[1],integer)
                  and type(z[2],integer)
                  and z[1]-z[2] mod 2=0) then
  ERROR();
end if;
RETURN((z[1]+z[2]*I*sqrt(3))/2);
end proc;
```

```
EIcompl := proc (z::list)
    if not (nops(z) = 2 and type(z[1], integer) and type(z[2], integer) and
    (z[1] - z[2]) mod 2 = 0) then ERROR( )
    end if ;
    RETURN(1/2×z[1] + 1/2×I×z[2]×sqrt(3))
end proc
```

Пример.

```
> EIcompl([3,-5]);
```

$$\frac{3}{2} - \frac{5}{2} I \sqrt{3}$$

Задача 6. Представить процедуру **EIpt**, конвертации ***целочисленного списка***, задающего число Эйзенштейна, в ***список координат*** (в формате **`float`**) соответствующей ***точки*** на комплексной плоскости.

Решение.

```
> EIpt:=proc(z::list);
if not (nops(z)=2 and type(z[1],integer)
                  and type(z[2],integer)
                  and z[1]-z[2] mod 2=0) then
  ERROR();
fi;
RETURN([evalf(Re(EIcompl(z))),
        evalf(Im(EIcompl(z)))]);
end proc;
```

Пример.

```
> EIpt([3,-5]);EIpt([3,-4]);
```

[1.500000000, -4.330127020]

```
Error, (in EIpt)
```

Еще одна вспомогательная функция **EInorm** *должна вычислять* ***норму*** *числа Эйзенштейна по формуле второго способа задания:*

$$N(z) = \frac{a^2+3b^2}{4};\ z = \frac{a+ib\sqrt{3}}{2} \in Э.$$

Задача 7. Представить процедуру **EInorm**, вычисляющую ***норму*** числа Эйзенштейна, заданного ***целочисленным списком***.

Решение.

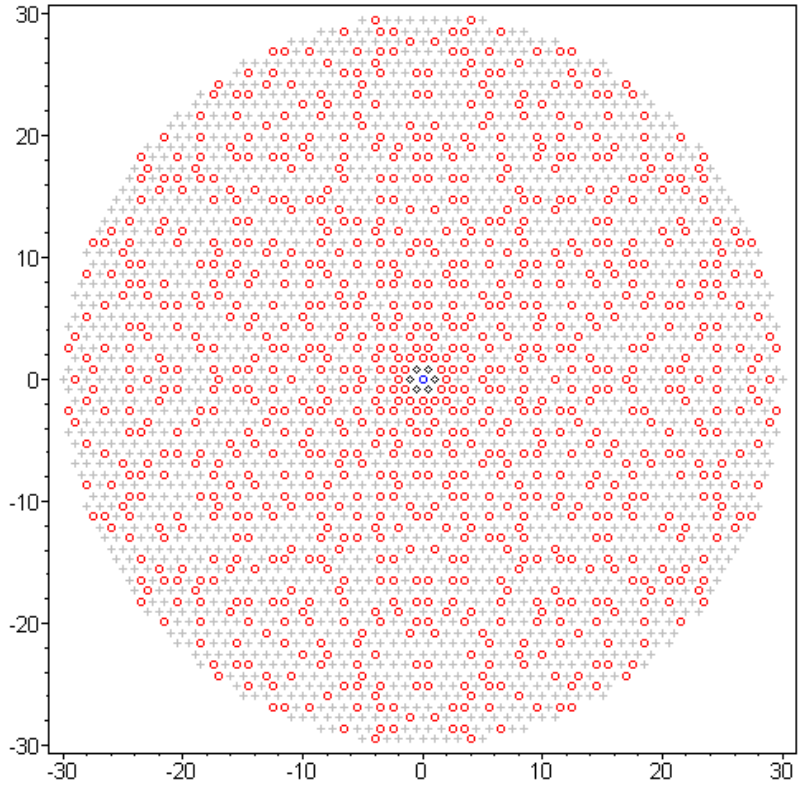
```
> EInorm:=proc(z::list);
if not (nops(z)=2 and type(z[1],integer)
                  and type(z[2],integer)
                  and z[1]-z[2] mod 2=0) then
  ERROR();
fi;
RETURN((z[1]^2+3*z[2]^2)/4);
end proc;
```

Пример.

```
> EInorm([3,-5]);
```

И, наконец, ключевая процедура **EIprime** *– **тест на простоту** числа Эйзенштейна **z**, соответствующий **критерию**:*

норма числа z должна быть
- либо простым натуральным числом p,
- либо квадратом p^2, простого числа p, сравнимого с 2 mod 3.



Задача 8. Представить процедуру-тест **EIprime.**

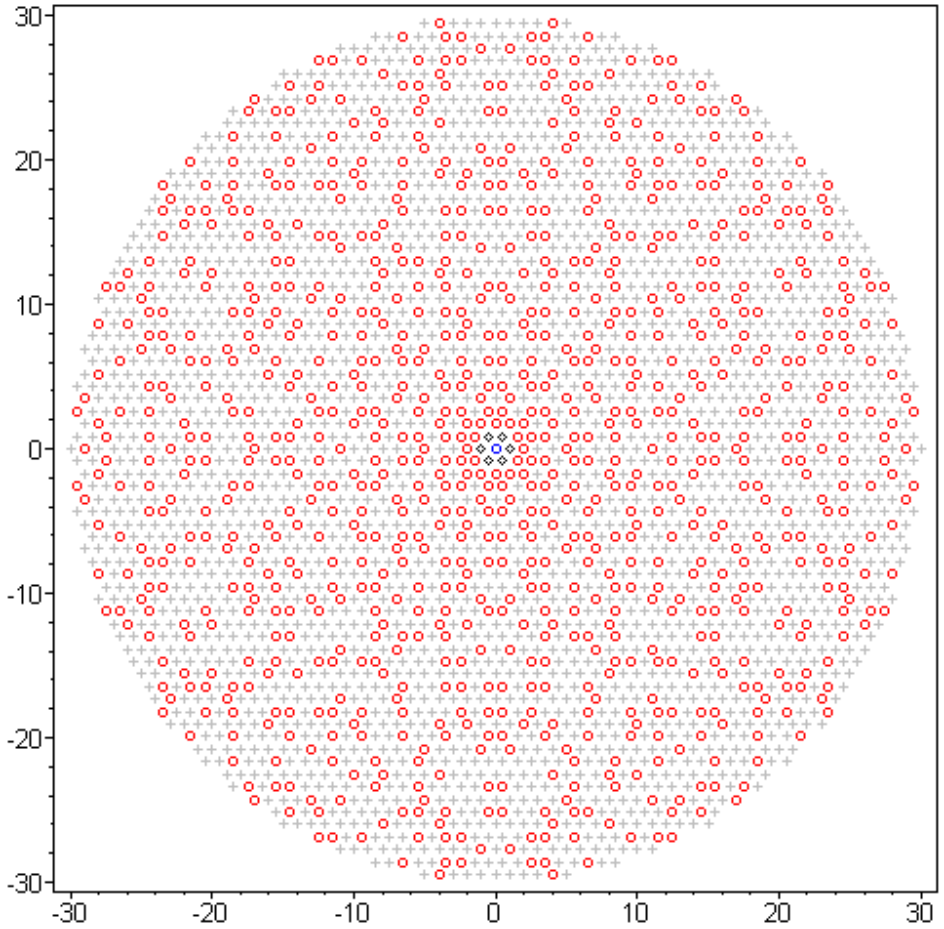
Решение.

```
> EIprime:=proc(z::list)
  local nrm,sqr,ans;
if not (nops(z)=2 and type(z[1],integer)
                  and type(z[2],integer)
                  and z[1]-z[2] mod 2=0) then
  ERROR();
end if;
ans:=false;
nrm:=EInorm(z);
sqr:=sqrt(nrm);
if type(nrm,prime) or (type(sqr,prime) and
                              sqr mod 3=2) then
  ans:=true;
end if;
RETURN(ans);
end proc;
```

Пример.

```
> map(EIprime,[[3,1],[2,2],[2,0],[75,13]]);
```

[*true*, *true*, *false*, *false*]



Рисующая процедура.

Задача 9. Представить графическую процедуру **`EIpict`**, возвращающую изображение фрагмента комплексной плоскости, содержащего все целые числа Эйзенштейна с нормой, не превышающей N^2, расклассифицированные по категориям:

- ***нуль;***
- ***обратимые числа (единицы);***
- ***простые числа;***
- ***составные числа.***

Решение. *Загрузка пакетов:*

```
> with(plots):with(plottools):

> EIpict:=proc(N::posint)
  local pts,a,b,z,nrm,pt;
pts:=[];
# Ограничения для a и для b здесь отличаются;
# это связано с тем, что коэффициенты при этих
# переменных тоже отличаются: 1/2 и sqrt(3)/2.
# Функция ceil округляет число до целого
# в большую сторону.
```

```
for a from -2*N to 2*N do
  for b from -ceil(2*N/sqrt(3))
        to ceil(2*N/sqrt(3)) do
# Проверка того, что a и b одинаковой четности.
    if (a-b) mod 2=0 then
      z:=[a,b];
# Вычисление нормы текущего числа.
      nrm:=EInorm(z);
# Определение соответствующей точки на плоскости.
      pt:=EIpt(z);
      if nrm<=N^2 then
        if nrm=0 then
```

```
# Описание изображения нулевого элемента.
          pts:=[pts[],point(pt,symbol=CIRCLE,
                                color=BLUE)];
        elif nrm=1 then
# Описание изображений обратимых элементов.
          pts:=[pts[],point(pt,symbol=DIAMOND,
                                color=BLACK)];
        elif EIprime(z) then
# Описание изображений простых элементов.
          pts:=[pts[],point(pt,symbol=CIRCLE,
                                color=RED)]:
        else
# Описание изображений составных элементов.
          pts:=[pts[],point(pt,symbol=CROSS,
                                color=GRAY)];
```

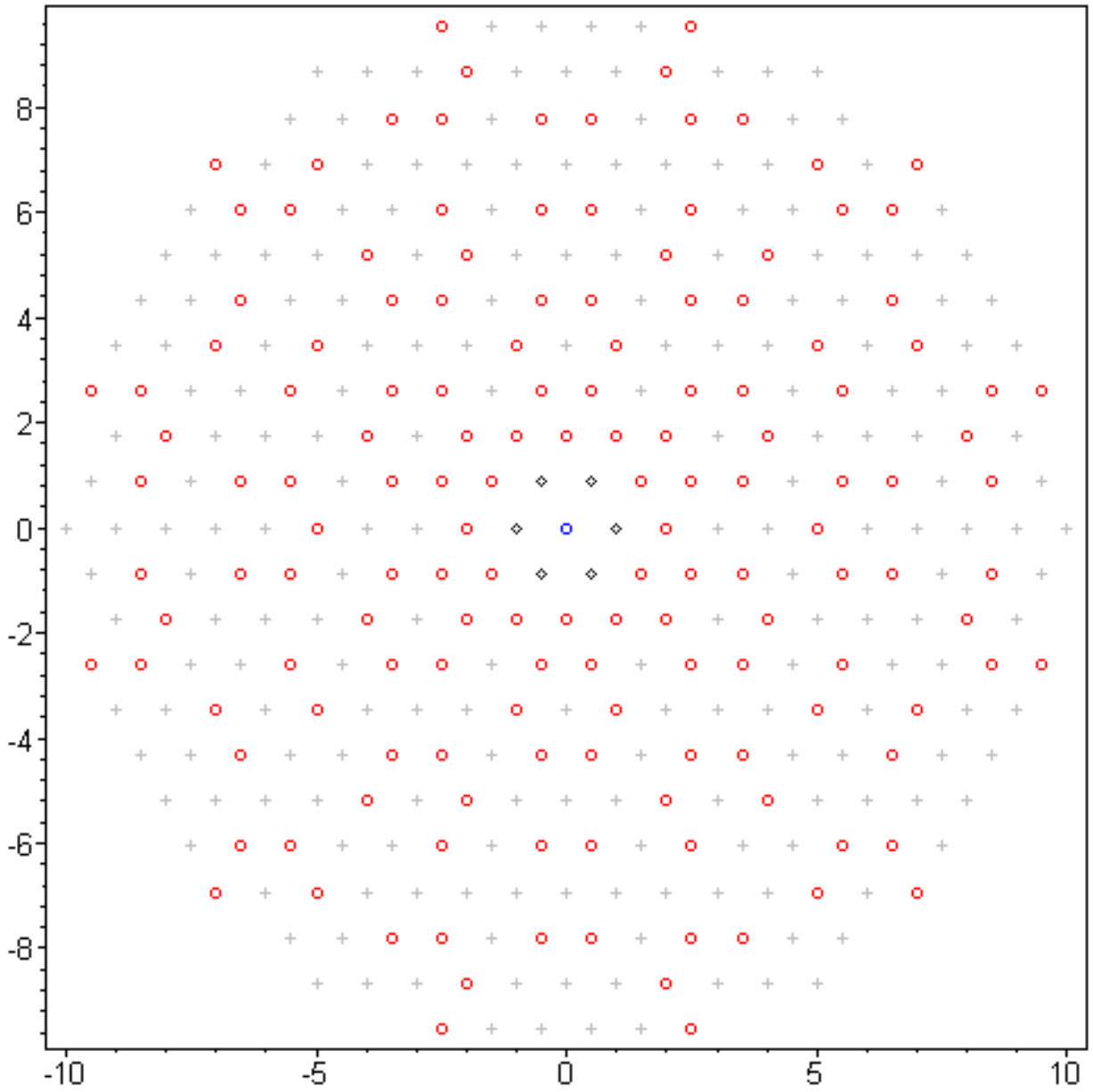
```
        end if;
      end if;
    end if;
  end do;
end do;
# Вывод всех описанных элементов на дисплей.
display(pts,axes=BOXED);
end proc;
```

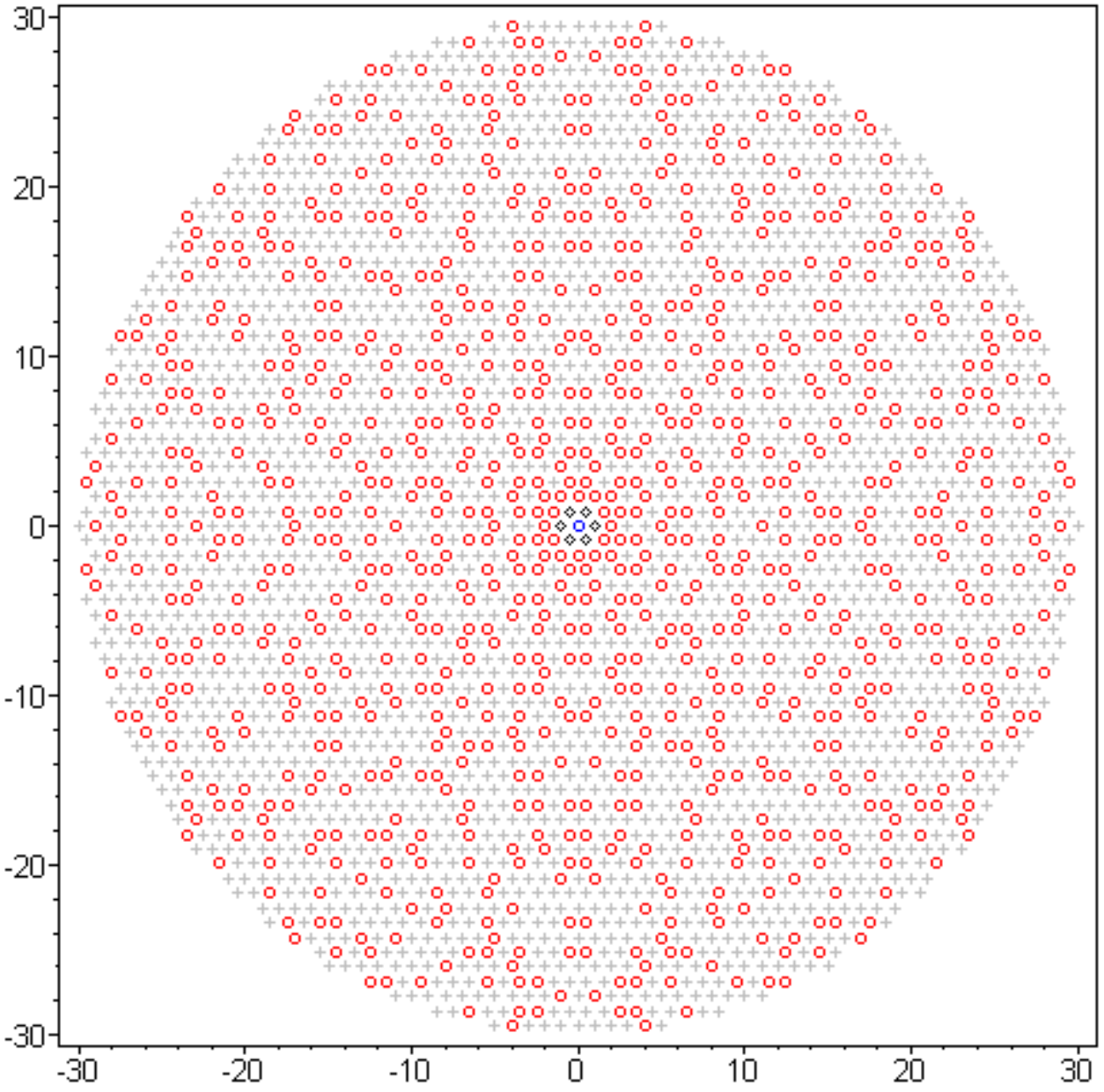
```
EIpict := proc (N::posint)
local pts, a, b, z, nrm, pt;
    pts := [ ];
    for a from −2×N to 2×N do for b from −floor( 2×N/sqrt( 3 ) ) to
        floor( 2×N/sqrt( 3 ) ) do
```

```
            if (a − b) mod 2 = 0 then
                z := [a, b];
                nrm := EInorm(z);
                pt := EIpt(z);
                if nrm ≤ N^2 then
                    if nrm = 0 then pts := [pts[ ],
                        plottools:-point(pt, symbol = CIRCLE, color = BLUE)
                        ]
                    elif nrm = 1 then pts := [pts[ ], plottools:-point(pt,
                        symbol = DIAMOND, color = BLACK)]
                    elif EIprime(z) then pts := [pts[ ],
                        plottools:-point(pt, symbol = CIRCLE, color = RED)]
                    else pts := [pts[ ],
                        plottools:-point(pt, symbol = CROSS, color = GRAY)]
                    end if
                end if
            end if
        end do
    end do ;
    plots:-display(pts, axes = BOXED)
end proc
```

Рисунки 3 - 4. Числа Эйзенштейна с нормой, не превышающей **100** и **900**.

```
> EIpict(10);
   EIpict(30);
```




30
20
10
0
-10
-20
-30
-30
-20
-10
0
10
20
30

Вычисления с числами Пелля

Числа **Пелля**,

$$z = a + b\sqrt{2} \in \Pi_2,$$

в отличие от чисел **Гаусса** и **Эйзенштейна**,
являются ***действительными***

$$\Pi_2 \subset \mathbb{R}$$

и представление их совокупности точками на обычной ***комплексной плоскости*** невозможно.

Тем не менее, поскольку поле $\mathbb{R}$ является (бесконечномерным) линейным пространством над полем $\mathbb{Q}$, рассматривая $\sqrt{2}$ как некий аналог "обычной" мнимой единицы ($i = \sqrt{-1}$), можно рассмотреть двумерное линейное подпространство (плоскость, изоморфную $\mathbb{Q}^2$), содержащее 1 и $\sqrt{2}$, как

гиперболическую

комплексную

плоскость.

На этой плоскости вводится ***"знакопеременная метрика"***, в которой вектор $(\boldsymbol{a}, \boldsymbol{b})$ имеет ***"знакопеременную длину"*** $\boldsymbol{a^2 - 2b^2}$, модуль которой задает ***евклидову норму***

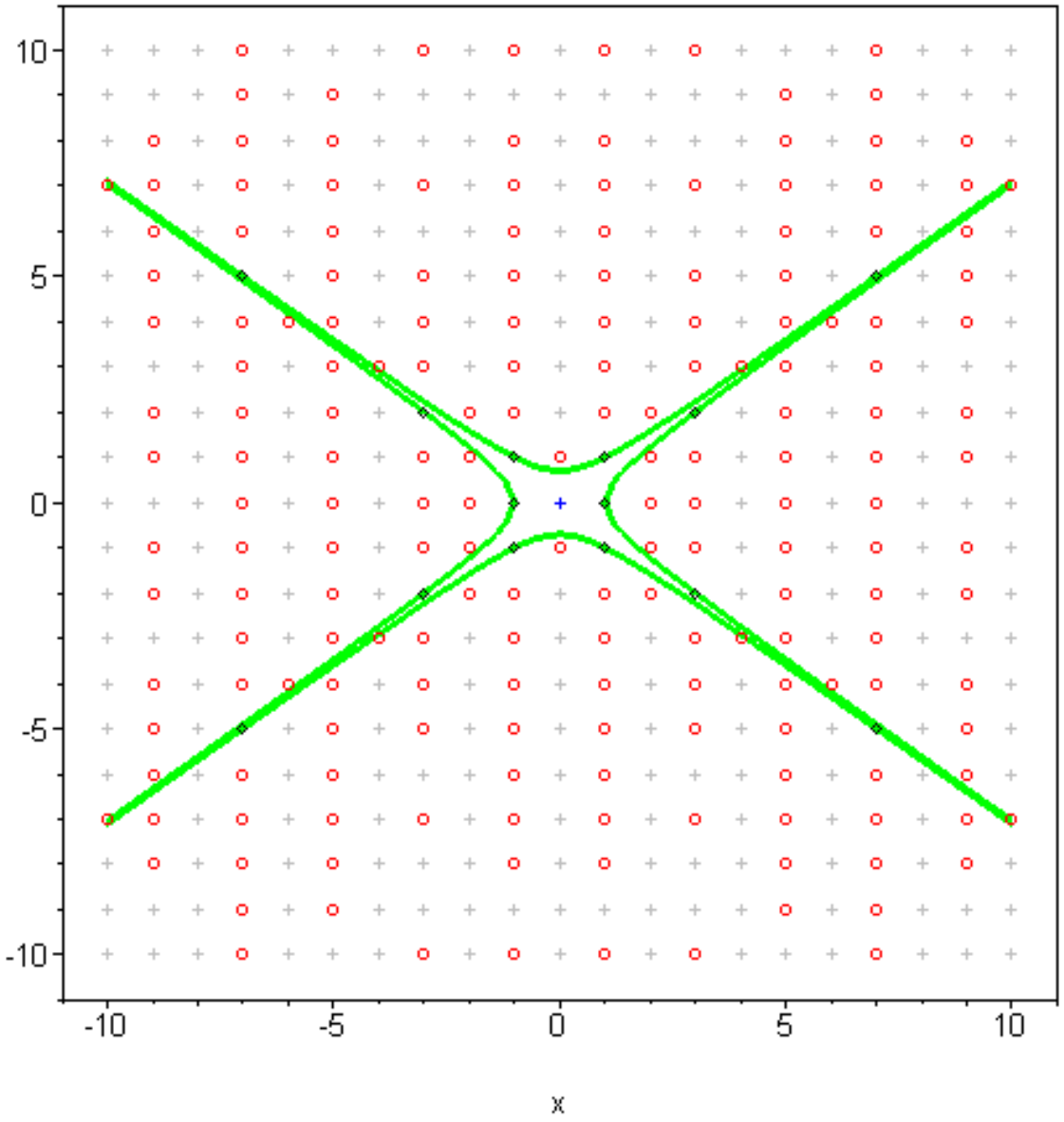
$$\boldsymbol{N\left(a + b\sqrt{2}\right) = |a^2 - 2b^2|},$$

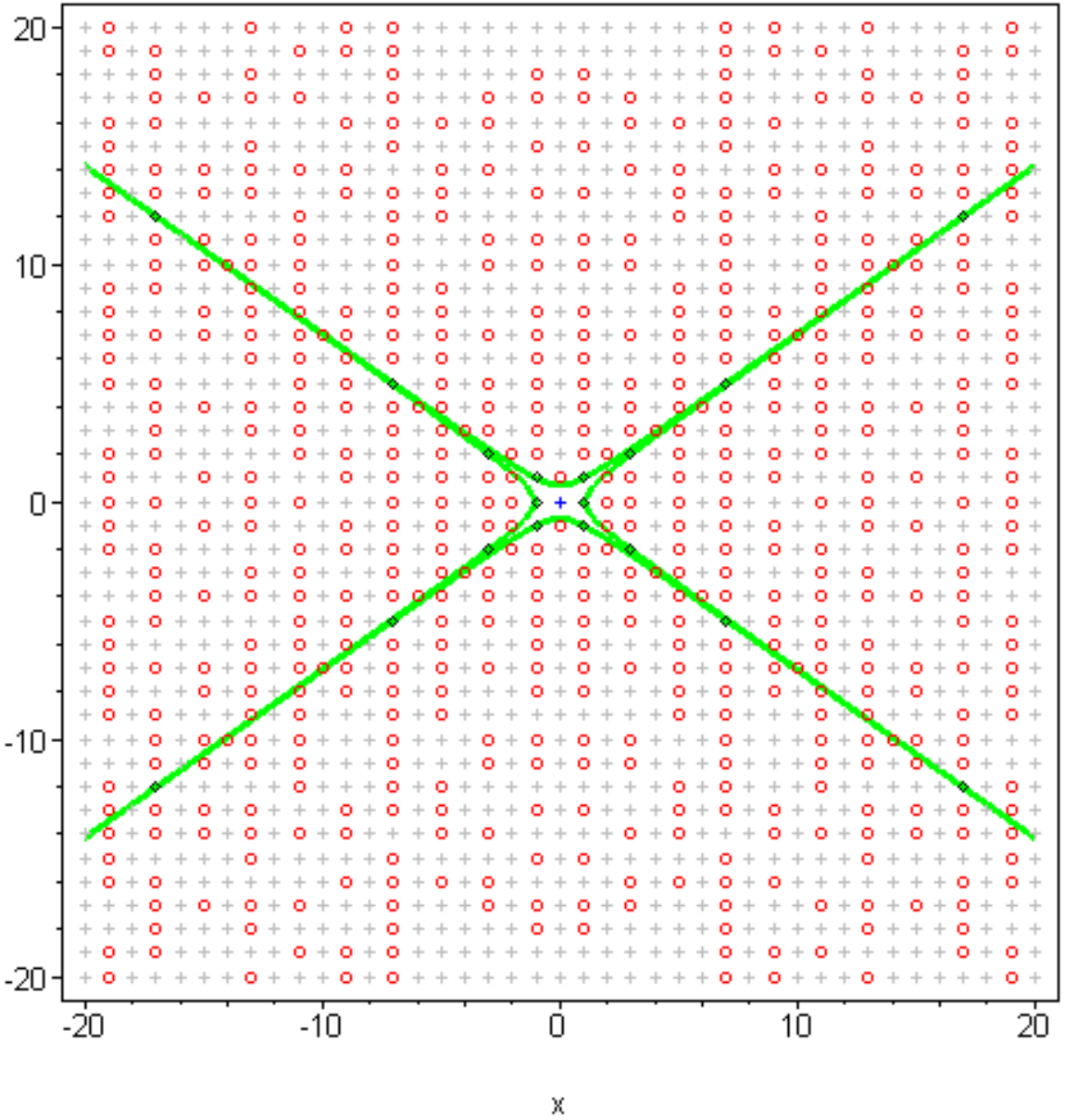
обеспечивающую деление с остатком в решетке

$$\boldsymbol{\Pi_2 \cong \mathbb{Z}^2 \subset \mathbb{Q}^2}.$$

Именно на этой гиперболической плоскости $\boldsymbol{\Pi_2}$ выполняется последующее рисование. Вы увидите, в частности, пару сопряженных гипербол, на которых располагаются обратимые числа Пелля.

Рисунки 5 - 6. Числа Пелля $z = a + b\sqrt{2}$ в области $|a|, |b| \leq 10$ и в области $|a|, |b| \leq 20$ (в плоской модели, обратимые – на гиперболах).




20
10
0
-10
-20
-20
-10
0
10
20
x