сравнительно произвольной формы с достаточно гладкой границей. В языке FreeFem++ реализован эффективный алгоритм триангуляции области, требующий для генерации треугольной сетки задания лишь количества вершин треугольников на границах. Замечательно то, что основная часть кода программы для решения задачи в прямоугольнике остается прежней. Переход к другой области осуществляется заменой только строк 6–9, определяющих границу.

### 3.2.1 Решение задачи для круга

Пусть область $\overline{D}$ представляет собой круг $x^2+y^2 \leqslant 1$ единичного радиуса. Границу области $\overline{D}$ можно задать параметрически при помощи соотношений $x = \cos t$, $y = \sin t$, $t \in [0, 2\pi]$. Предположим, что граница области разбита на четыре фрагмента

$$\begin{array}{llll}
\Gamma_{11}: & x = \cos t, & y = \sin t, & t \in [0,\ 0{,}33\pi]; \\
\Gamma_{2}: & x = \cos t, & y = \sin t, & t \in [0{,}33\pi,\ \pi]; \\
\Gamma_{3}: & x = \cos t, & y = \sin t, & t \in [\pi,\ 1{,}41\pi]; \\
\Gamma_{12}: & x = \cos t, & y = \sin t, & t \in [1{,}41\pi,\ 2\pi].
\end{array}$$

Операторы, задающие границу, в этом случае будут следующими:

```
border Gamma11(t=0,0.33*pi)      { x=cos(t); y=sin(t); };
border Gamma2 (t=0.33*pi,pi)     { x=cos(t); y=sin(t); };
border Gamma3 (t=pi,1.41*pi)     { x=cos(t); y=sin(t); };
border Gamma12(t=1.41*pi,2.0*pi){ x=cos(t); y=sin(t); };
```

Решение задачи (изолинии температуры) (3.1)–(3.4) для параметров, определенных соотношениями (3.14), приведено на рис. 3.3.

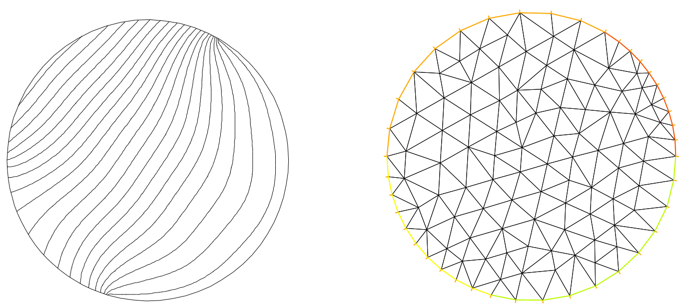

Рис. 3.3. Изолинии температуры для задачи в круге и триангуляция

### 3.2.2 Распределение температуры в четырехугольнике

Пусть $D$ представляет собой четырехугольник с координатами вершин $(x_i, y_i)$ $(i = 0, \ldots, 3)$. В этом случае код, задающий границу, может быть записан в виде

```
int m=4;
real[int] xx(m), yy(m);
xx[0]=0;    yy[0]=0.2;    xx[1]=0.5;  yy[1]=0;
xx[2]=0.8;  yy[2]=0.8;    xx[3]=0.5;  yy[3]=1;
border Gamma2(t=0,1) { x=xx[0]*(1-t)+xx[1]*t; y=yy[0]*(1-t)+yy[1]*t; };
border Gamma11(t=0,1){ x=xx[1]*(1-t)+xx[2]*t; y=yy[1]*(1-t)+yy[2]*t; };
border Gamma3(t=0,1) { x=xx[2]*(1-t)+xx[3]*t; y=yy[2]*(1-t)+yy[3]*t; };
border Gamma12(t=0,1){ x=xx[3]*(1-t)+xx[0]*t; y=yy[3]*(1-t)+yy[0]*t; };
```

Решение задачи (изолинии температуры) (3.1)–(3.4) для параметров, определенных соотношениями (3.14), приведено на рис. 3.4.

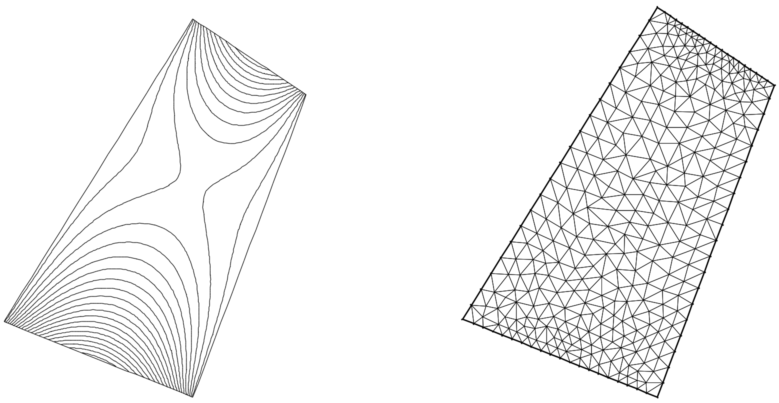

Рис. 3.4. Изолинии температуры для задачи в четырехугольнике и триангуляция

Для области, представляющей собой полигон, координаты вершин которого $(x_i, y_i)$ $(i = 0, \ldots, m)$ известны, граница может быть задана аналогичным образом. Однако, при этом потребуются и другие изменения в программе. Необходимо будет переписать строку с оператором генерации сетки `mesh Th = buildmesh(...)` и строки, задающие краевые условия на соответствующих участках границы полигона.

### 3.2.3 Решение задачи в криволинейной области

Две точки на плоскости $(x_1, y_1)$, $(x_2, y_2)$ можно соединить кривой линией, используя любое параметрическое представление кривой, например,

$$x = x_1 \cos t + x_2 \sin t, \qquad y = y_1 \cos t + y_2 \sin t, \quad t \in [0, \pi/2]$$

или

$$x = x_1(1 - t^2) + x_2 t^2, \qquad y = y_1(1 - t^2) + y_2 t^2, \quad t \in [0, 1].$$

Приведем код для задания криволинейной области (см. рис. 3.5)

```
int m=4;
real[int] xx(m), yy(m);
xx[0]=0;     yy[0]=0.4;   xx[1]=0.5;    yy[1]=0;
xx[2]=1;     yy[2]=1;     xx[3]=0.15;   yy[3]=0.35;
```

```
border Gamma2(t=0,1) { x=xx[0]*(1-t)+xx[1]*t;
                              y=yy[0]*(1-t)+yy[1]*t; };
border Gamma11(t=0,1){ x=xx[1]*(1-t)+xx[2]*t;
                              y=yy[1]*(1-t)+yy[2]*t; };
border Gamma3(t=0,1) { x=xx[2]*(1-t^2)+xx[3]*t^2;
                              y=yy[2]*(1-t^2)+yy[3]*t^2; };
border Gamma12(t=0,0.5*pi){ x=xx[3]*cos(t)+xx[0]*sin(t);
                              y=yy[3]*cos(t)+yy[0]*sin(t); };
```

Решение задачи (изолинии температуры) (3.1)–(3.4) для параметров, определенных соотношениями (3.14), приведено на рис. 3.5.

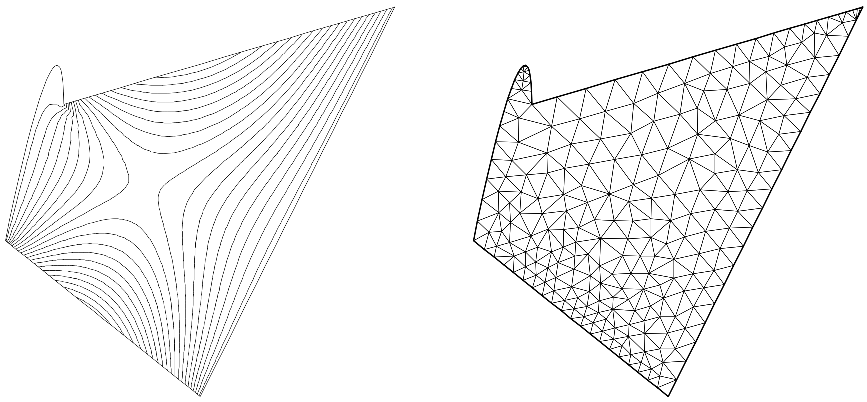

Рис. 3.5. Изолинии температуры для задачи в криволинейной области и триангуляция

### 3.2.4 Решение задачи в области с отверстием

Приведем фрагмент кода, позволяющий рассмотреть задачу, например, в области $D$ с круглым отверстием (см. рис. 3.6).

```
border Gamma11(t=0,0.33*pi) { x=cos(t); y=sin(t); };
border Gamma2 (t=0.33*pi,pi){ x=cos(t); y=sin(t); };
border Gamma3 (t=pi,2*pi)   { x=cos(t); y=sin(t); };
border Gamma12(t=0,2*pi)    { x=0.5*cos(t); y=0.5*sin(t); };
```

Решение задачи (изолинии температуры) (3.1)–(3.4) для параметров, определенных соотношениями (3.14), приведено на рис. 3.6.

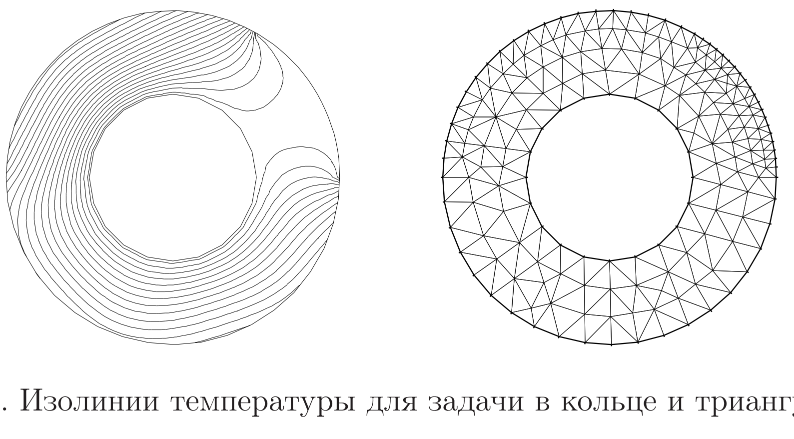

Рис. 3.6. Изолинии температуры для задачи в кольце и триангуляция

Обратим внимание, что для сохранения ориентации контура в операторе `mesh` следует записывать внутренний контур в следующем формате `Gamma12(-5*n)` (подробнее см. гл. 17). Соответствующая строка кода, генерирующая сетку, имеет вид

```
mesh Th = buildmesh(Gamma2(5*n)+Gamma11(5*n)+Gamma3(5*n)+Gamma12(-5*n));
```

Еще один пример сложной области с отверстиями, для которой можно решать задачи (в частности, уравнение теплопроводности с заданными краевыми условиями), показан на рис. 3.7.

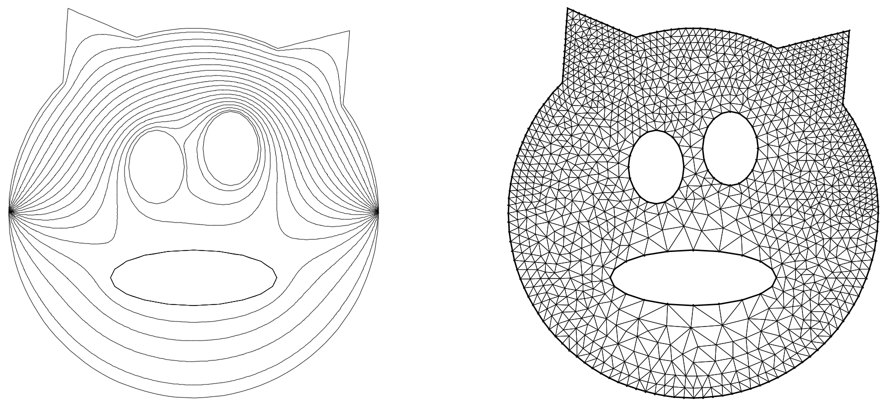

Рис. 3.7. Изолинии температуры в области сложной формы и триангуляция

## 3.3 Физические задачи, приводящие к уравнению Лапласа

Этот раздел носит справочный характер и предназначен в первую очередь для тех, кто, написав первую программу на языке FreeFem++, хочет расширить свой физический кругозор и исследовать некоторые модели реальных прикладных задач. Более полные сведения о постановке задач, конечно же, можно найти практически в любом курсе уравнений математической физики. Однако, на наш взгляд, полезно иметь возможность почерпнуть некоторую информацию непосредственно в книге о языке FreeFem++. Это, в частности, позволит сразу же провести вычислительные эксперименты и получить наглядное представление о многих физических процессах.

### 3.3.1 Теплопроводность

Стационарное распределение температуры в некоторой области $D$ описывается уравнением (уравнение баланса энергии)

$$\operatorname{div} \boldsymbol{q} = f, \tag{3.15}$$

где $\boldsymbol{q}$ — плотность потока тепла, $f$ — плотность внутренних источников.

Плотность потока тепла $\boldsymbol{q}$ связана с температурой $\theta$ законом теплопроводности Фурье

$$\boldsymbol{q} = -\varkappa \nabla \theta, \tag{3.16}$$