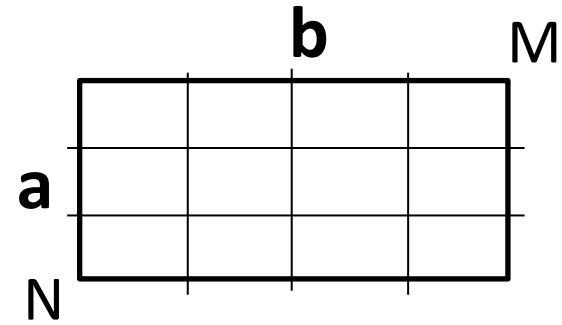


1. Целочисленное деление в формулах для  $i1$ ,  $i2$ ,  $j1$ ,  $j2$ :

$$i1 = \frac{N}{\sqrt{size}} * (rank / \sqrt{size})$$

$$j1 = \frac{M}{\sqrt{size}} * (rank \% \sqrt{size})$$

в программе:  $\text{int}(\text{sqrt}(\text{size}))$  или  
 $\text{int } k = \text{sqrt}(\text{size});$



2. Кратность размеров матрицы количеству процессов по строкам и столбцам:

$$\frac{N}{\sqrt{size}}, \quad \frac{M}{\sqrt{size}}$$

$$\frac{N}{a}, \quad \frac{M}{b}, \quad \text{когда } size = a * b$$

- все эти числа  
должны быть целыми!!!

# Структура программы:

1. Начальные условия
2. Граничные условия
3. Основная часть:

```
for (t=0; t<T; t++)  // временной цикл (while)
{
    3.1  for (i=i1; i<i2; i++)
          for (j=j1; j<j2; j++)
          {
              // расчеты
          }

    3.2  РАССЫЛКА ГРАНИЧНЫХ СТРОК И СТОЛБЦОВ (Send и Recv)

    3.3  for (i=i1; i<i2; i++)
          for (j=j1; j<j2; j++)
              oldU[i][j] = newU[i][j];          // переобозначение для перехода
                                                    на следующий временной слой
}
```

4. Вывод стационарного устойчивого результата (cout)

## 3.1 Расчеты:

```
for (i = i1 + 1; i < i2; i++)  
    for (j = j1 ; j < j2-1; j++)  
    {  
        newU[i][j] = oldU[i][j] - dt * (a*(oldU[i][j] - oldU[i - 1][j]) / dx  
  
        + b * (oldU[i][j] - u_old[i][j + 1]) / dy);  
    }
```

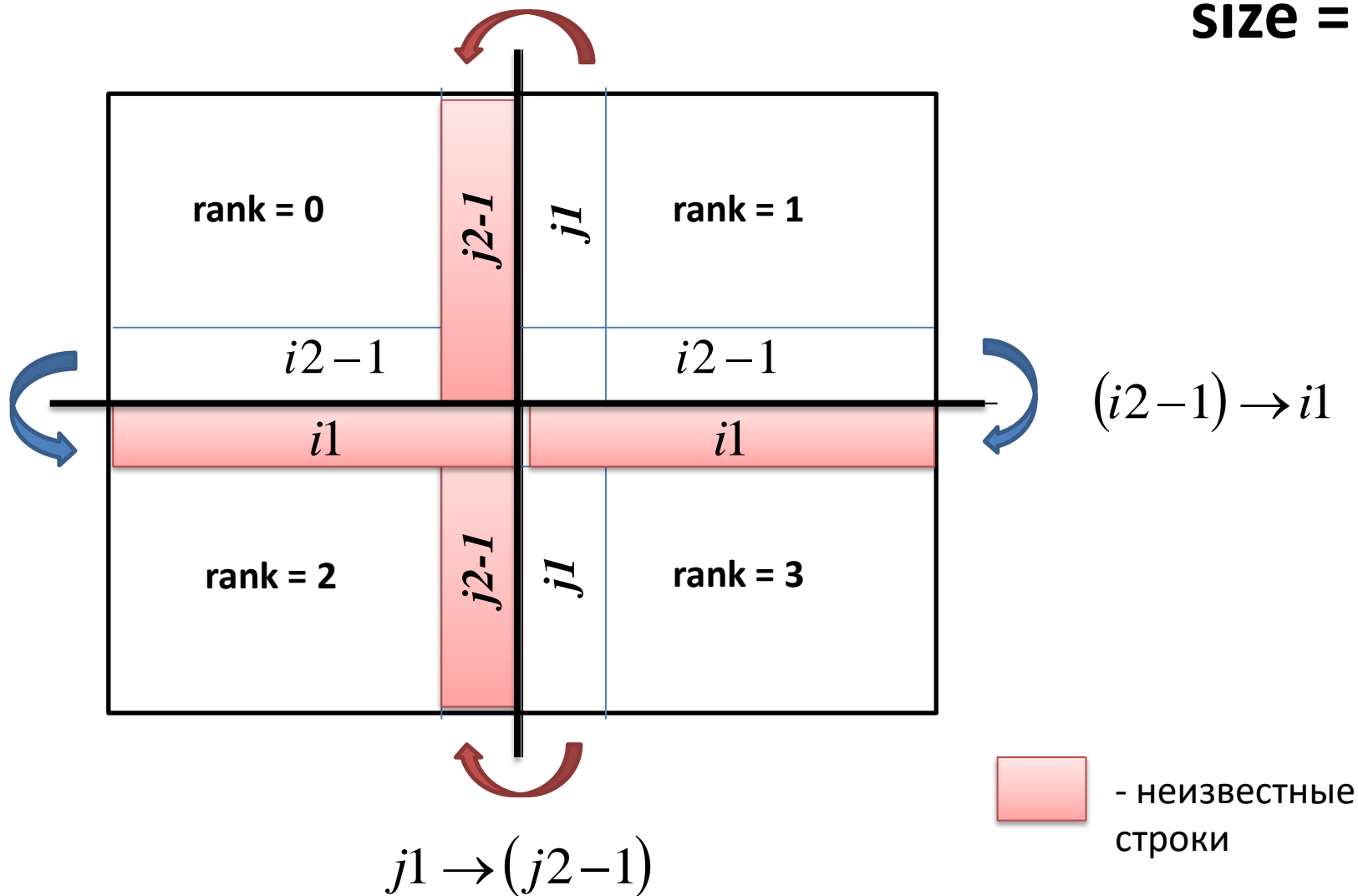
Тогда появляются не вычисляемые строки **i1** и столбцы **j2-1** на каждом процессе.

Поэтому и **возникает необходимость** рассылки граничных строк для заполнения полученных неизвестных строк и столбцов!!!

(для разных задач и разных схем аппроксимации это могут быть **разные строки и столбцы**)

### 3.2 РАССЫЛКА ГРАНИЧНЫХ СТРОК И СТОЛБЦОВ (для данной схемы 3.1)

**size = 4**



- Реализация рассылки на C++ **без цикла**:

$$\text{if } \left( \frac{\text{rank}}{\sqrt{\text{size}}} \neq \sqrt{\text{size}} - 1 \right)$$

$$\text{MPI\_Send} \left( \mathbf{u}[\mathbf{i2}-1][\mathbf{0}], \frac{\mathbf{N}}{\sqrt{\mathbf{size}}}, \dots, \text{rank} + \sqrt{\text{size}}, \dots \right);$$

$$\text{if } \left( \frac{\text{rank}}{\sqrt{\text{size}}} \neq 0 \right)$$

$$\text{MPI\_Recv} \left( \mathbf{u}[\mathbf{i1}][\mathbf{0}], \frac{\mathbf{N}}{\sqrt{\mathbf{size}}}, \dots, \text{rank} - \sqrt{\text{size}}, \dots \right);$$

---



$$\text{if } \left( \text{rank} \% \sqrt{\text{size}} \neq \sqrt{\text{size}} - 1 \right)$$

$$\text{MPI\_Send} \left( \mathbf{u}[\mathbf{0}][\mathbf{j1}], \frac{\mathbf{M}}{\sqrt{\mathbf{size}}}, \dots, \text{rank} + 1, \dots \right);$$


$$\text{if } \left( \text{rank} \% \sqrt{\text{size}} \neq 0 \right)$$

$$\text{MPI\_Recv} \left( \mathbf{u}[\mathbf{0}][\mathbf{j2}-1], \frac{\mathbf{M}}{\sqrt{\mathbf{size}}}, \dots, \text{rank} - 1, \dots \right);$$

- Реализация рассылки на C++ **с циклом**:



```
for(j = j1; j < j2; j++)
{
    if (...)
        MPI_Send(u[i2-1][j], 1, ..., rank + sqrt(size), ...);
    if (...)
        MPI_Recv(u[i1][j], 1, ..., rank - sqrt(size), ...);
}
```



```
for(i = i1; i < i2; i++)
{
    if (...)
        MPI_Send(u[i][j1], 1, ..., rank + 1, ...);
    if (...)
        MPI_Recv(u[i][j2-1], 1, ..., rank - 1, ...);
}
```

Выбираете один из двух вариантов рассылки (с циклом или без)

- 1. Целочисленное деление
- 2.  $N/a$ ,  $M/b$  – целые числа
- 3. Структура программы
- 4.  $i_1+1$ ,  $i_2-1$  (шаг вперед и назад)
- 5. Не высчитываемые значения (в зависимости от граничных условий) и их рассылка
- 6. Рассылка с циклом или без цикла