

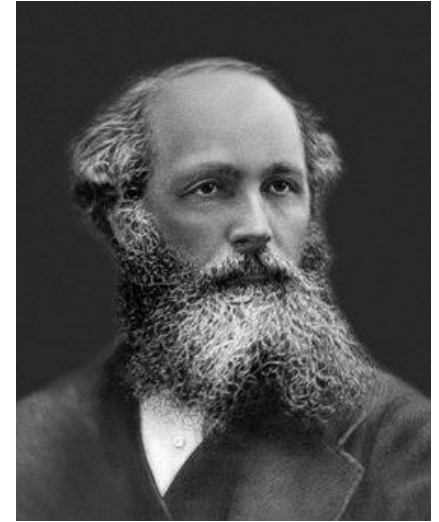
# **КЛЕТОЧНО-АВТОМАТАЯ МОДЕЛЬ ПЛОСКОЙ МОНОХРОМНОЙ ЭЛЕКТРОМАГНИТНОЙ ВОЛНЫ**

**Докладчик:** Бурнышев Егор, магистрант 1 курса ФПМИ НГТУ

**Научный руководитель:** Медведев Юрий Геннадьевич, к.т.н.,  
с.н.с. лаб. синтеза параллельных программ ИВМиМГ СО РАН

# Понятия электромагнетизма

- *Электромагнитное излучение (ЭМИ)* — распространяющееся в пространстве возмущение (изменение состояния) электромагнитного поля. Может рассматриваться как электромагнитная волна или как поток фотонов, в зависимости от характера решаемой задачи.



Джеймс Клерк Максвелл

- *Монохромное излучение* — колебания вектора электрической напряженности  $E$  происходят с одной частотой  $\nu$ .
- *Когерентное излучение* — колебания вектора  $E$  происходят синхронно (с одной фазой  $\phi$  в каждой точке пространства).

# Структура клеток массива

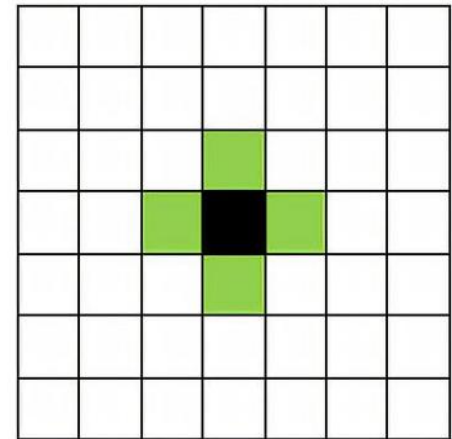
В основу разработанной модели лег подход, базирующийся на дискретном распространении частиц внутри клеточного массива.

**Типы клеток:** источник, среда.

**Константные** характеристики клетки:

- индексы клетки  $(i, j)$  в клеточном массиве, соответствующие некоторым координатам  $(x, y)$  в двумерном пространстве;
- коэффициент поглощения  $\alpha$ ;
- коэффициент преломления  $n$ ;
- направление полета порождаемых частиц  $\theta$  (только для источников);

**Переменная** характеристика клетки – совокупность находящихся в клетке модельных частиц;



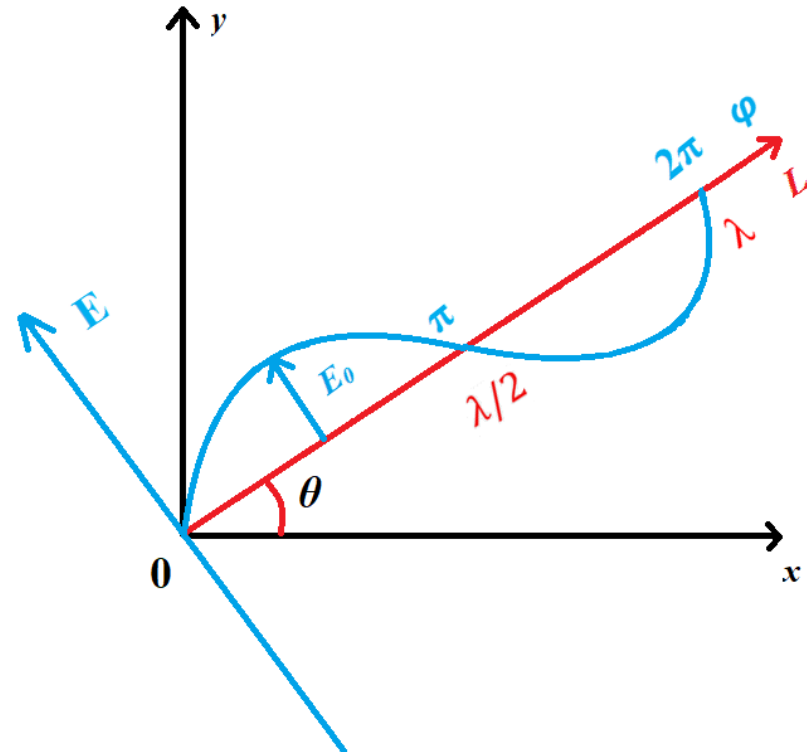
# Характеристики частицы

## Константные:

- амплитуда  $E_0$  вектора  $E$ ;
- направление движения  $\theta$ ;
- начальная фаза  $\varphi_0$ ;
- частота  $\nu$ .

## Переменные:

- значение вектора  $E$ ;
- текущая фаза  $\varphi$ ;
- длина волны  $\lambda$ ;
- координаты  $x$ ,  $y$ ;
- время жизни.



Частица и направление её движения ( $\varphi_0 = 0$ )

# Функция переходов клеточного автомата

Режим функционирования – **синхронный**.

Функция переходов работает в три этапа: **порождение, посылка, приём**.

1. Порождение – генерация новых частиц в клетках источников.
2. Посылка – пересчет характеристик модельных частиц в клетке.
3. Прием – обход всех соседей клетки и перемещение частиц, которые летят от соседа в ее направлении.

# Программная реализация

Главная трудность, которая возникает при программировании КА – задача корректного представления непрерывных вещественных характеристик в дискретном целом виде.

координаты  $x, y$

время жизни

начальная фаза  $\varphi_0$

частота  $\nu$

амплитуда  $E_0$  вектора  $E$

направление движения  $\theta$

```
typedef struct Particle {
    uint16_t    lifetime;
    uint16_t    count_transitions;
    int32_t     sin;
    int32_t     cos;
    int32_t     direction_deviation;
    float       array_speed;
    float       start_phase;
    float       frequency;
    float       E_0;
} Particle;
```

# Программная реализация

На этапе **порождения** осуществляется генерация новых частиц в клетках источников. При этом характеристики этих частиц однозначно определяются параметрами источника.

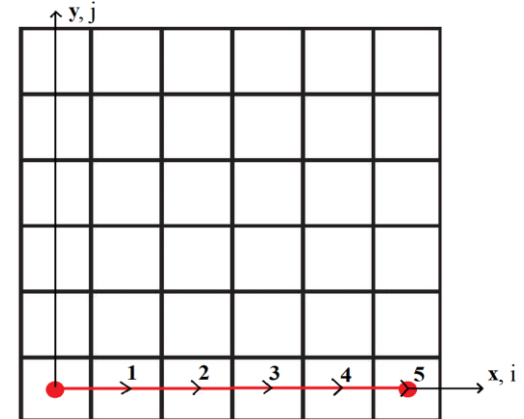
```
void generation_source(Cell* cell) {
    size_t new_count = cell->particles_count + PARTICLES_PER_ITERATION;
    if (new_count > MAX_PARTICLES_PER_CELL) {
        new_count = MAX_PARTICLES_PER_CELL;
    }

    float k = calculate_k(cell->sin, cell->cos);

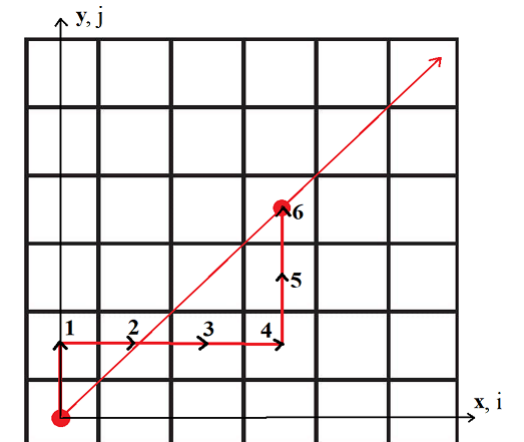
    Particle new_particle = {
        .lifetime = 0,
        .count_transitions = 0,
        .sin = cell->sin,
        .cos = cell->cos,
        .direction_deviation = 0,
        .array_speed = k,
        .start_phase = 0.0f,
        .wavelength = WAVELENGTH,
        .E_0 = E_AMPLITUDE,
    };

    for (size_t i = cell->particles_count; i < new_count; ++i) {
        cell->particles[i] = new_particle;
    }

    cell->particles_count = new_count;
}
```



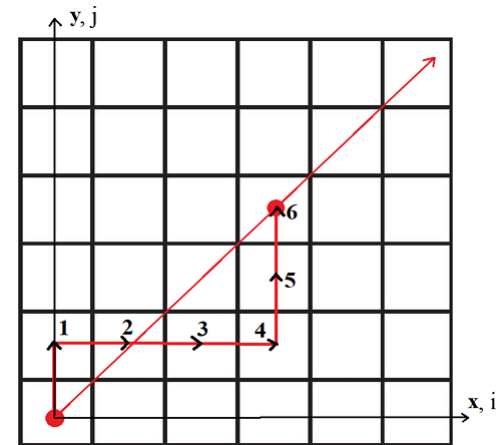
Частица летящая под углом 0  
(5 итераций)



Частица летящая под углом  $\pi/4$   
(6 итераций)

# Программная реализация

На этапе **посылки** осуществляется сравнение пройденного вещественного расстояния с числом переходов частицы между клетками массива, а также выбор направления полета в зависимости от текущего значения `direction_deviation`.



```
int send(void *n) {
    Cell* cell = (Cell*) n;
    for (size_t i = 0; i < cell->particles_count; ++i) {
        Particle* p = &cell->particles[i];
        float total_distance = p->array_speed * cell->environment_speed * (float)p->lifetime;
        if (total_distance >= (float)(p->count_transitions + 1)) {
            if (p->sin == 0 || p->direction_deviation <= 0) {
                p->direction_deviation += abs(p->sin);
            } else {
                p->direction_deviation -= abs(p->cos);
            }
        }
    }
    return 0;
}
```

# Программная реализация

В момент **приема** клетка сначала удаляет все улетающие из нее частицы и высвобождает ресурсы. Затем осуществляется обход соседей клетки и все летящие в ее направлении частицы осуществляют переход.

```
// Define neighbors and their direction flags/signs
Cell* neighbours[4] = {cell + 1, cell + 2, cell + 3, cell + 4};
bool x_flags[4] = {true, false, true, false};
int sin_signs[4] = {0, -1, 0, 1};
int cos_signs[4] = {1, 0, -1, 0};
bool is_fly_by_x = true;

for (int dir = 0; dir < 4; ++dir) {
    Cell* neighbour = neighbours[dir];
    for (size_t i = 0; i < neighbour->particles_count; ++i) {
        Particle* p = &neighbour->particles[i];
        float distance = p->array_speed * neighbour->environment_speed * (float)p->lifetime;

        if (distance >= (float)(p->count_transitions + 1)) {
            is_fly_by_x = (p->sin == 0 || p->direction_deviation <= 0);
            if (is_fly_by_x == x_flags[dir]) {
                if ((x_flags[dir] && p->cos * cos_signs[dir] > 0) ||
                    (!x_flags[dir] && p->sin * sin_signs[dir] > 0)) {
                    move_particle_from_neighbour(cell, neighbour, i);
                }
            }
        }
    }
}
```

# Постпроцессинг. Норма вектора электрической напряженности

$$\|\mathbf{E}_{\text{particle}}\|_2 = |E_0 \cdot \sin \varphi|,$$

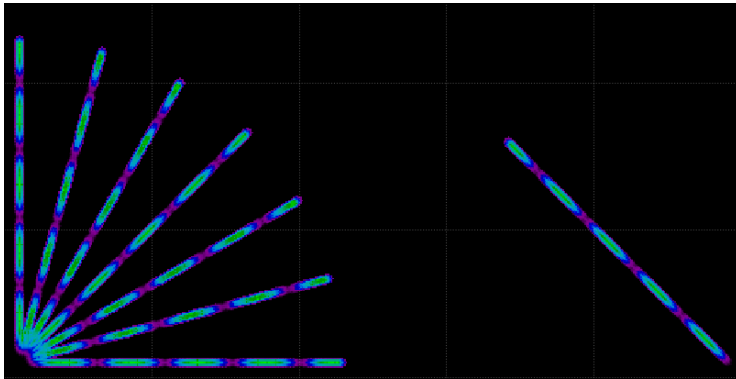
$$\|\mathbf{E}_{\text{cell}}\|_2 = \left\| \sum \mathbf{E}_{\text{particle}} \right\|_2,$$

$$\|\mathbf{E}_{\text{cell},R}(i_{\text{cell}}, j_{\text{cell}})\|_2 = \frac{1}{N} \left\| \sum_{i=-R}^R \sum_{j=-R}^R \mathbf{E}_{\text{cell}}(i_{\text{cell}} + i, j_{\text{cell}} + j) \right\|_2,$$

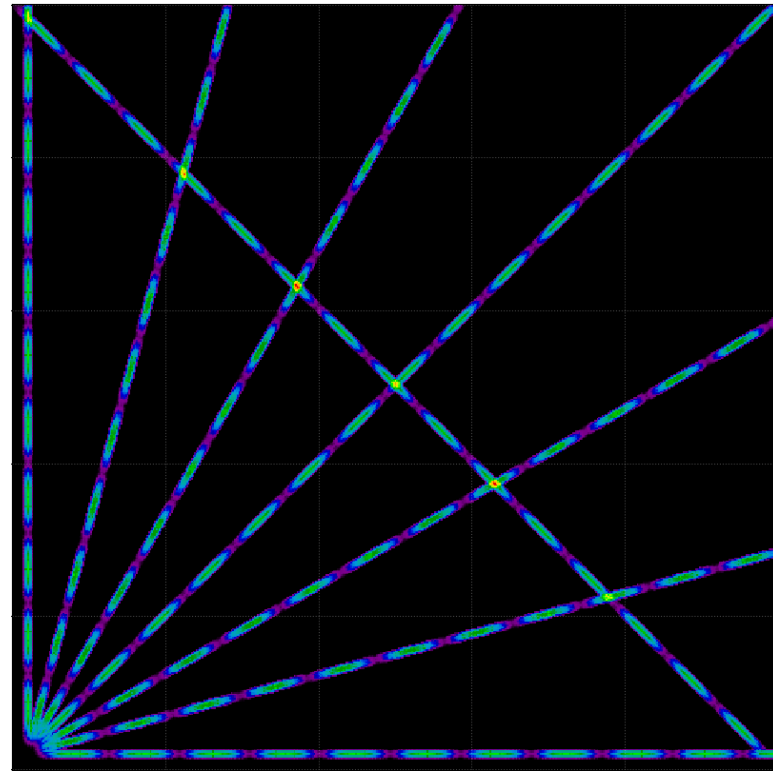
где  $\sqrt{i^2 + j^2} \leq R$ .

# Тестирование. Изотропность

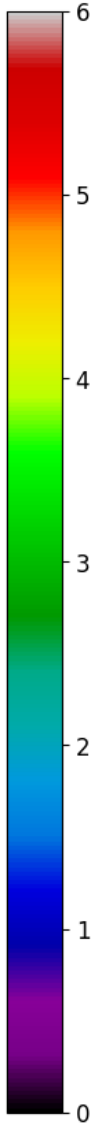
$\lambda = 92$  клетки;  
 $\varphi_0 = 0$ ;  
 $E_0 = 10$ ;  
Размер 500 на 500 клеток.



300 итераций

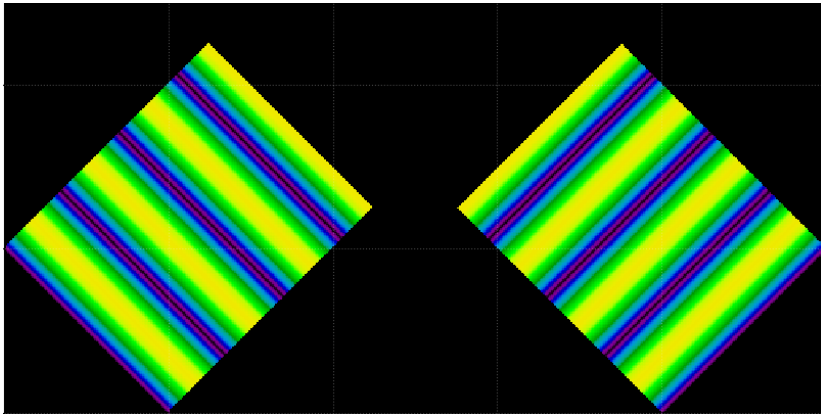


1000 итераций

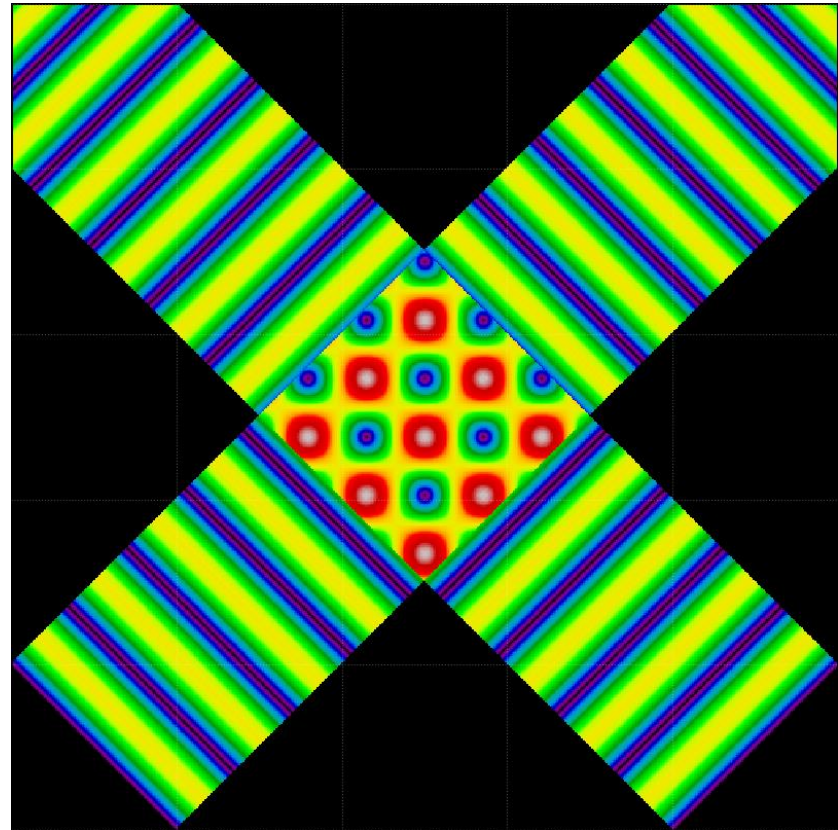


# Тестирование. Интерференция

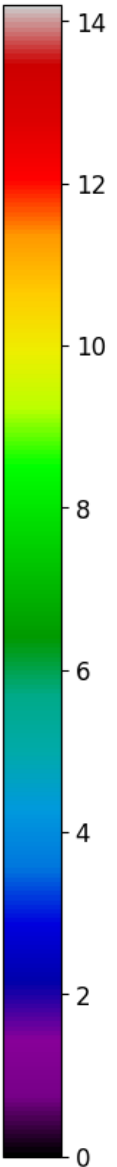
$\lambda = 100$  клетки;  
 $\varphi_0 = 0$ ;  
 $E_0 = 10$ ;  
Размер 500 на 500 клеток.



250 итераций



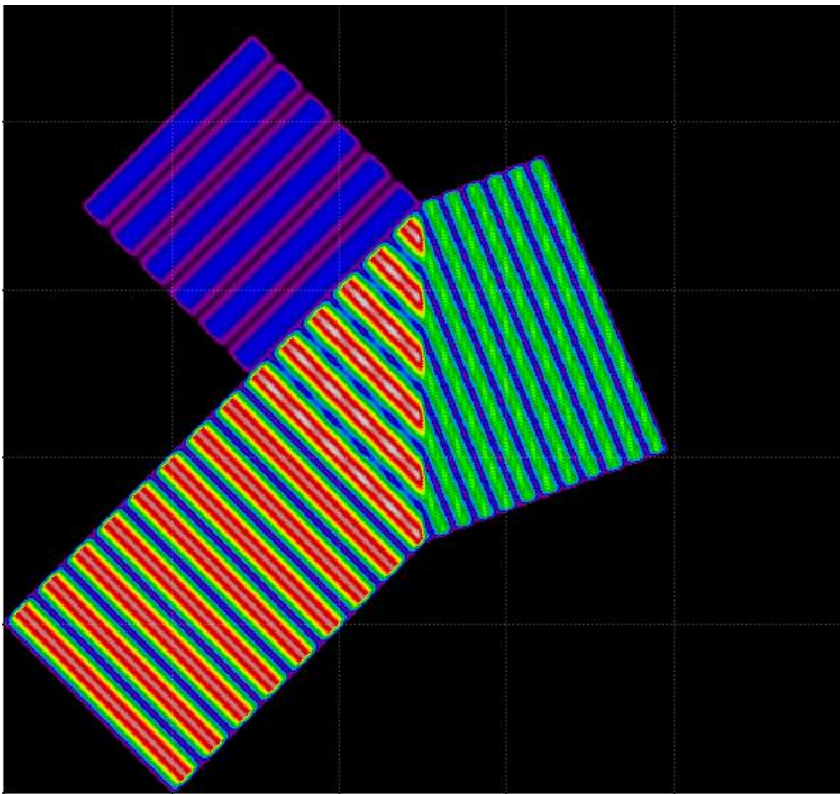
1000 итераций



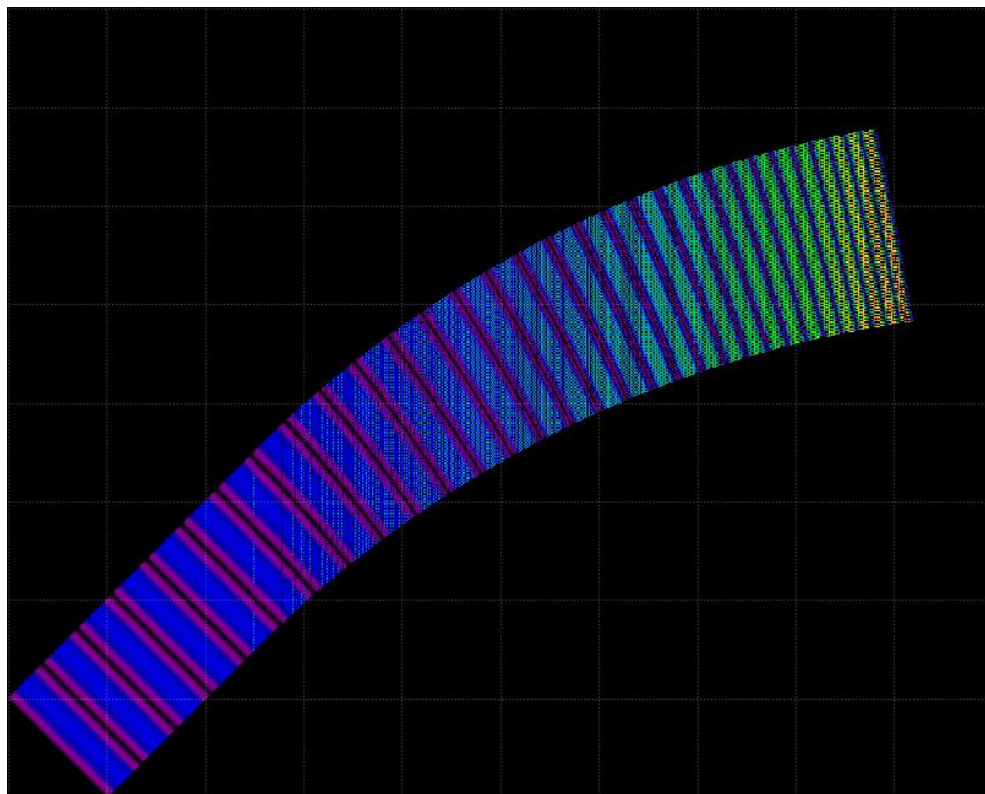
# Дальнейшие планы

1. Сферические волны;
2. Среды с различными характеристиками (преломление, отражение);
3. Дифракция;
4. Динамическое изменение параметров среды;
5. Нагрев и тепловое излучение.

# Преломление



строгая граница раздела



градиент коэффициента преломления

Спасибо за внимание!