

Моделирование эффекта саморепродукции. "Ковёр Талбота"

Выполнил: Мулихов Артур Б01-818

Цели:

- посмотреть на интенсивность в зоне $z \leq z_t$
- убедиться что оценка на максимальное число плоскостей саморепродукции адекватна
- убедиться что вся картина саморепродукции лежит в зоне дифракции Френеля



документация библиотеки:

- <https://pypi.org/project/diffractio/> (<https://pypi.org/project/diffractio/>)
- <https://diffractio.readthedocs.io/en/latest/> (<https://diffractio.readthedocs.io/en/latest/>)

In [1]:

```
1 #!pip install diffractio
2 #!pip install hickle
```

Подгружаем библиотеки

In [2]:

```

1 from diffractio import plt, sp, np, um, mm, degrees, num_max_processors
2
3 from diffractio.scalar_masks_X import Scalar_mask_X
4 from diffractio.scalar_masks_XZ import Scalar_mask_XZ
5 from diffractio.scalar_sources_X import Scalar_source_X
6 from diffractio.scalar_fields_XZ import Scalar_field_XZ
7
8 from diffractio.utils_multiprocessing import (_pickle_method, _unpickle_method,
9                                              execute_multiprocessing)
10 %matplotlib inline
11
12 from matplotlib import rcParams
13 rcParams['figure.dpi']=500

```

number of processors: 8

In [3]:

```

1 def RunExperiment(x, z, period, wavelength):
2     # инициализируем источник
3     u0 = Scalar_source_X(x, wavelength)
4     # создаём плоскую волну с максимальной амплитудой A=1
5     u0.plane_wave(A=1)
6
7     # инициализируем функцию пропускания транспоранта
8     t = Scalar_mask_X(x, wavelength)
9     # решетка рончи или наша обычная дифр решетка
10    t.ronchi_grating(period=period)
11
12    # инициализируем поля после решётки.
13    talbot_effect = Scalar_field_XZ(x, z, wavelength)
14    talbot_effect.incident_field(u0*t)
15    # метод просчёта поля (beam propagation method https://en.wikipedia.org/wiki/Beam\_propagation\_method)
16    talbot_effect.BPM()
17
18    return talbot_effect

```

Параметры системы:

In [4]:

```

1 wavelength = 0.6238 * um
2 # период решетки
3 period = 40 * um
4 # размер решетки
5 D = 800 * um

```

Для дифракционной решетки:

$$z_d = \frac{D^2}{\lambda} \text{ -- начиная с } z_d \text{ -- зона фраунгофера}$$

In [5]:

```
1 z_d = D**2/wavelength
2 print('Z_d:', round(z_d, 2), 'um')
```

Z_d: 1025969.86 um

 z_t -- Длина тальбота

$$z_t = \frac{2d^2}{\lambda}$$

In [6]:

```
1 z_talbot = 2 * period**2 / wavelength
2 print('Z_talbot:', round(z_talbot, 2), 'um')
```

Z_talbot: 5129.85 um

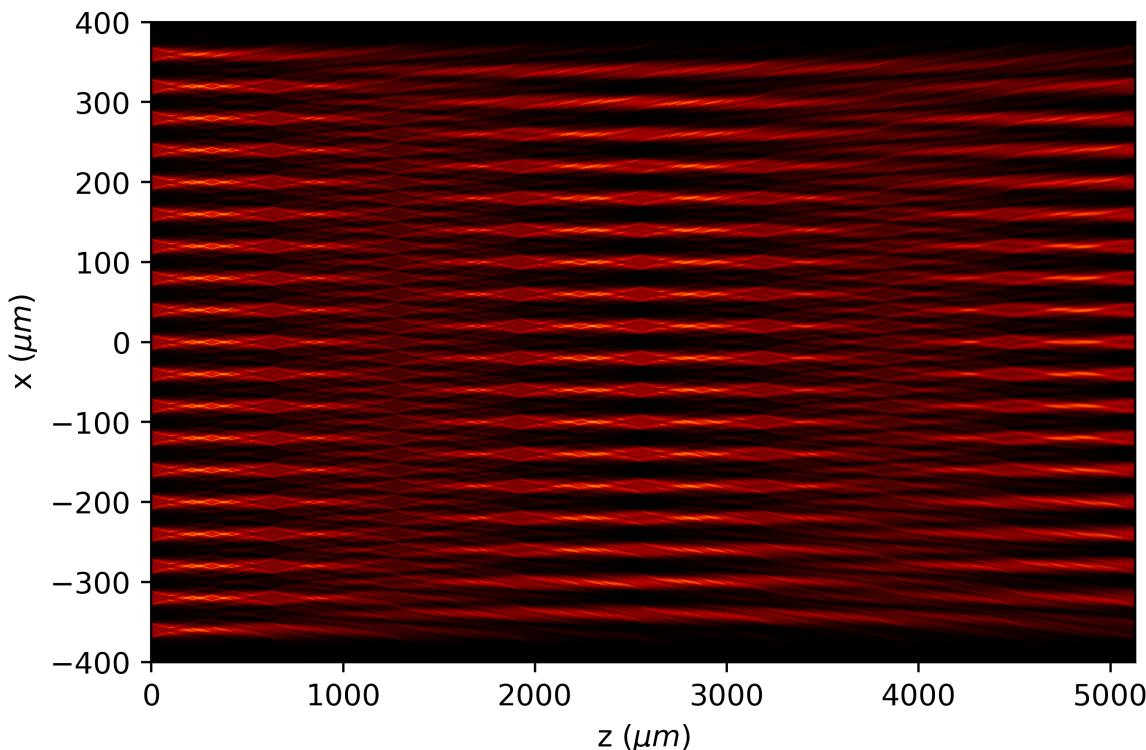
Нарисуем ковёр

In [7]:

```
1 x = np.linspace(-D/2, D/2, 2048)
2 z = np.linspace(0*um, z_talbot, 2048)
```

In [8]:

```
1 talbot_effect = RunExperiment(x, z, period, wavelength)
2 talbot_effect.draw(kind='intensity')
3 plt.show()
```



Проверим, что на L

$$L \approx \frac{D}{2\theta} \approx \frac{Dd}{2\lambda}$$

саморепродукция исчезает

Число плоскостей саморепродукции:

$$N \approx \frac{L}{z_T} = \frac{D}{4d}$$

In [9]:

```
1 x = np.linspace(-D/2, D/2, 2048)
2 z = np.linspace(0*um, 50*mm, 2048)
```

In [10]:

```
1 L = D * period / (2*wavelength)
2 N = L / z_talbot
```

In [11]:

```
1 print('На L:', round(L, 2), 'um от решетки саморепродукция исчезает, число плоскостей саморепродукции: ', N)
```

На L: 25649.25 um от решетки саморепродукция исчезает, число плоскостей саморепродукции: 5.0

Видим, что $L < z_d$, ожидаемо вся картина лежит в зоне френеля

Ожидаем увидеть примерно 5 пл-тей саморепродукции

In [12]:

```
1 talbot_effect = RunExperiment(x, z, period, wavelength)
2 talbot_effect.draw(kind='intensity')
3 plt.ylim(-350*um, 350*um)
4 plt.axvline(x = z_d)
5 plt.show()
```

