In [1]:

```
import numpy as np
from diffractio import mm, um, degrees
from diffractio.scalar_sources_XY import Scalar_source_XY
from diffractio.scalar_masks_XY import Scalar_mask_XY
# Setting up
length = 1 * mm
num_data = 512
x0 = np.linspace(-length / 2, length / 2, num_data)
y0 = np.linspace(-length / 2, length / 2, num_data)
wavelength = 0.633 * um
```

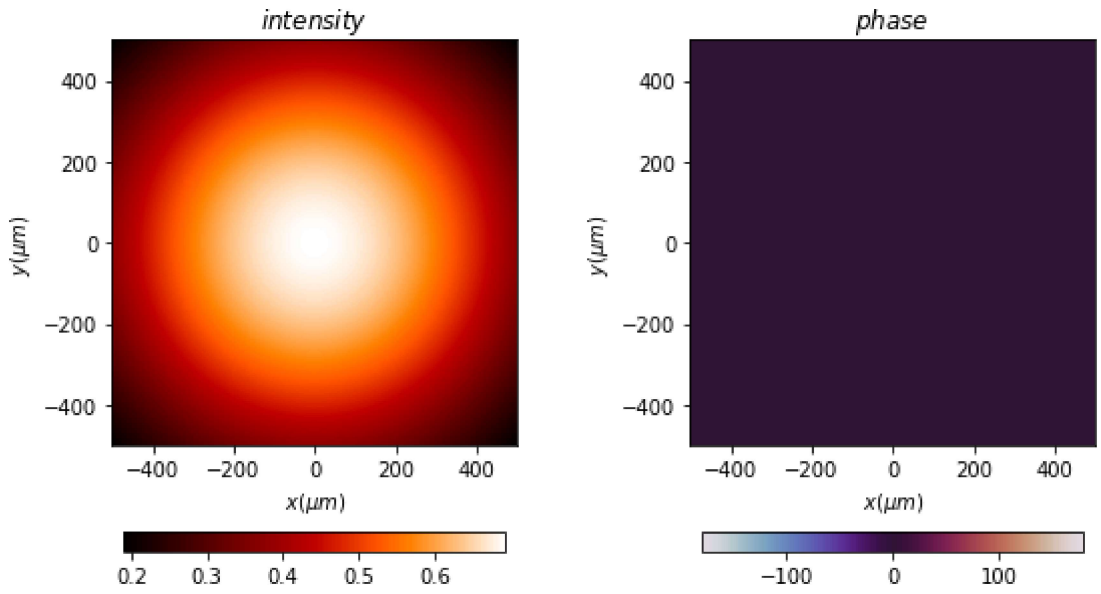
```
number of processors: 16
total memory        : 15.3 Gb
available memory    : 41 %
max frequency       : 1901 GHz
screeninfo not imported.
cv2 not imported. Function send_image_screen cannot be used
```

In [2]:

```
# Gaussian Beam Source - like a LASER
u0 = Scalar_source_XY(x=x0, y=y0, wavelength=wavelength)
u0.gauss_beam(r0=(0, 0), w0=(800 * um, 800 * um), z0=0.0)
u0.draw(kind='field', logarithm=True)
```

Out[2]:

```
((<matplotlib.image.AxesImage at 0x2167fed9f30>,
  <matplotlib.image.AxesImage at 0x2161989f940>),
 None,
 None)
```

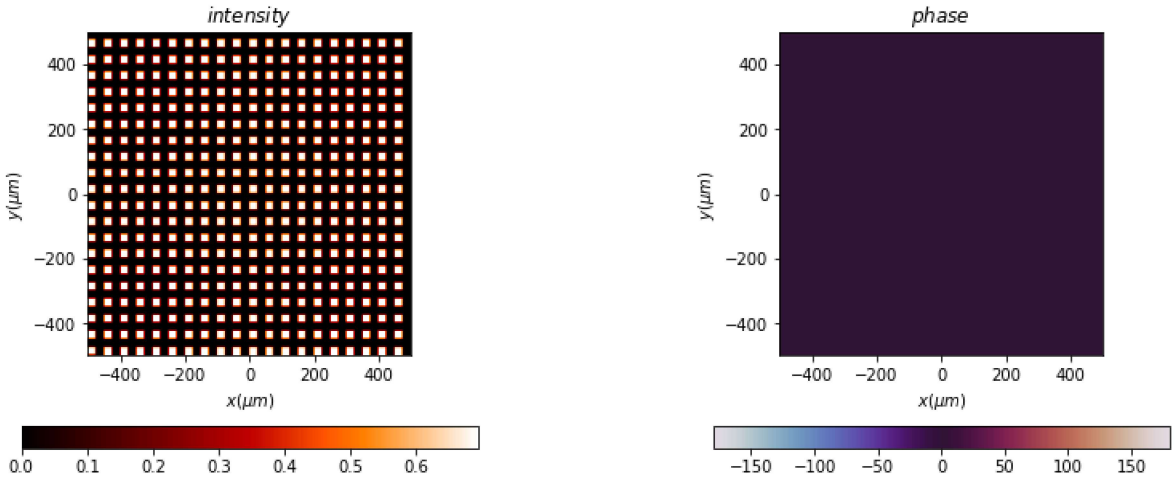


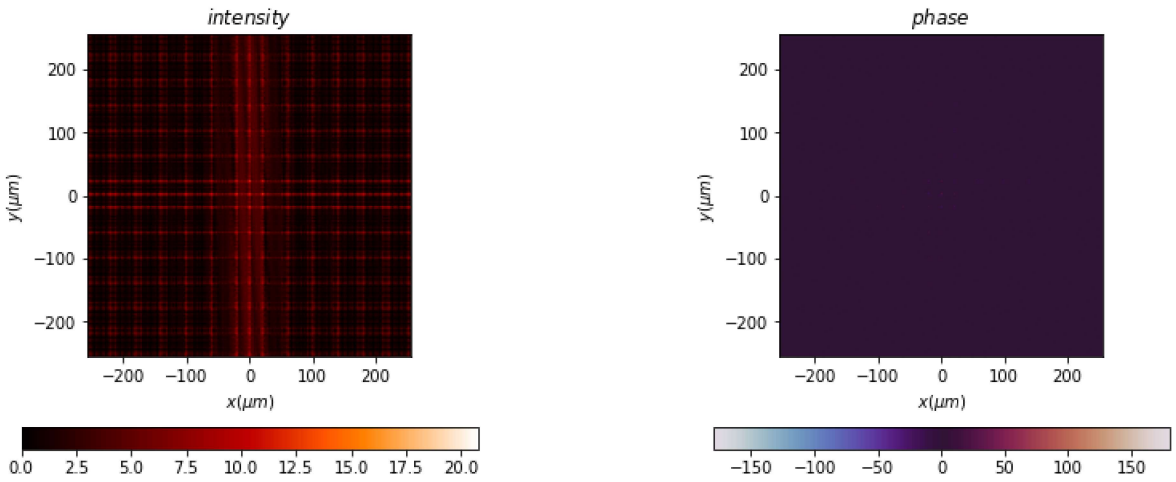
In [3]:

```
gratingx = Scalar_mask_XY(x0, y0, wavelength)
gratingx.ronchi_grating(
    period=50 * um, x0=0 * um,
)
gratingy = Scalar_mask_XY(x0, y0, wavelength)
gratingy.ronchi_grating(
    period=50 * um, x0=0 * um, angle = 90*degrees
)
varSlit =gratingx*gratingy
varSlit.draw(kind='field', logarithm=True)
a_varSlit = (u0 * varSlit).fft(z=1 * mm, new_field=True)
a_varSlit.draw( kind='field', logarithm=True)
```

Out[3]:

```
((<matplotlib.image.AxesImage at 0x2161cf508b0>,
  <matplotlib.image.AxesImage at 0x2161cff61d0>),
 None,
 None)
```

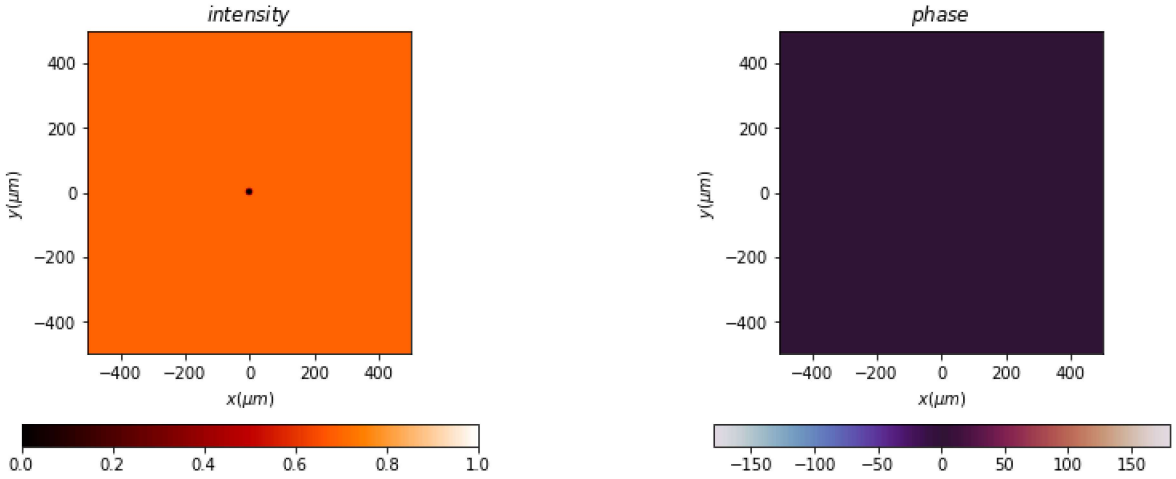




Masks

In [4]:

```
cDot = Scalar_mask_XY(x0, y0, wavelength)
cDot.ring(
 r0=(0 * um, 0 * um),
 radius1=(10 * um, 10 * um),
 radius2=(1000 * um, 1000 * um)
)
cDot.draw(kind='field', logarithm=True)

vert = Scalar_mask_XY(x0, y0, wavelength)
vert.slit(
 x0=0 * um,
 size=75 * um
)
```

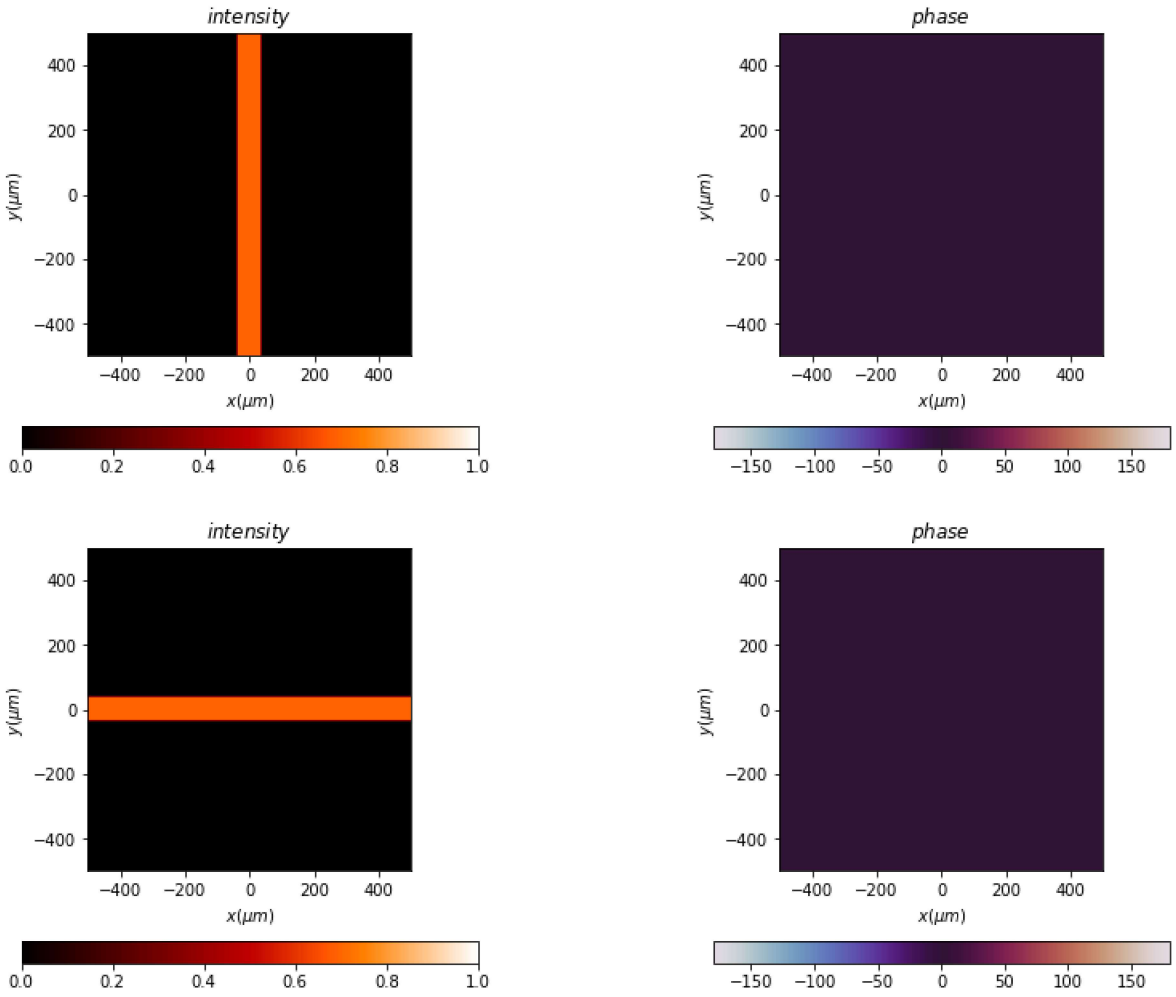


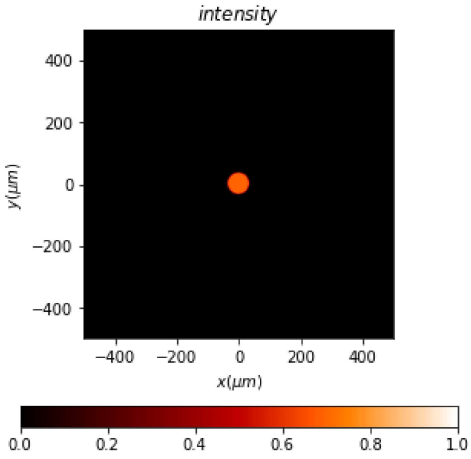
In [5]:

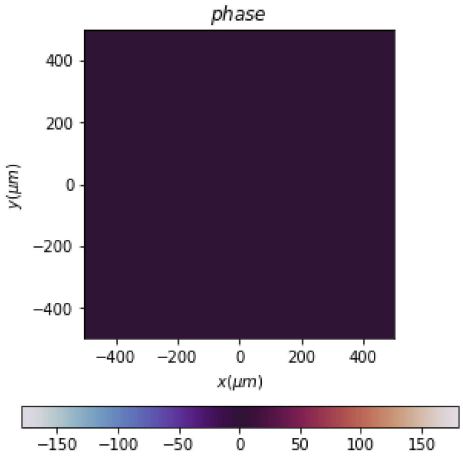
```
vert.draw(kind='field', logarithm=True)

horiz = Scalar_mask_XY(x0, y0, wavelength)
horiz.slit(
 x0=0 * um,
 size=75 * um,
 angle=np.pi / 2
)
horiz.draw(kind='field', logarithm=True)

circ = Scalar_mask_XY(x0, y0, wavelength)
circ.circle(
 r0=(0 * um, 0 * um),
 radius=(35 * um, 35 * um),
 angle=0
)
circ.draw(kind='field', logarithm=True)
```

Out[5]:

```
((<matplotlib.image.AxesImage at 0x2161dc46b60>,
  <matplotlib.image.AxesImage at 0x2161d06cd60>),
 None,
 None)
```







In [6]:

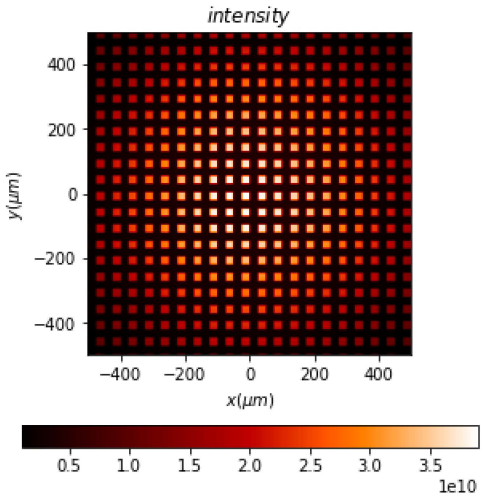
```
def sim(a_L1, mask):
    a_L1_Mask = a_L1 * mask
    a_L1_Mask_L2 = a_L1_Mask.fft(z=1 * mm, shift=False, remove0=False, new_field=True)
    return a_L1_Mask, a_L1_Mask_L2
```

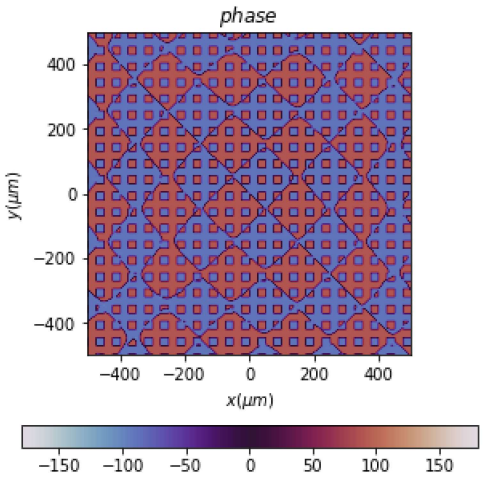
In [7]:

```
a_L1_cD, a_L1_cD_L2 = sim(a_varSlit, cDot)
a_L1_cD_L2.draw(kind='field', logarithm=False)
```

Out[7]:

```
((<matplotlib.image.AxesImage at 0x2161f42e080>,
  <matplotlib.image.AxesImage at 0x2161f4ed840>),
 None,
 None)
```



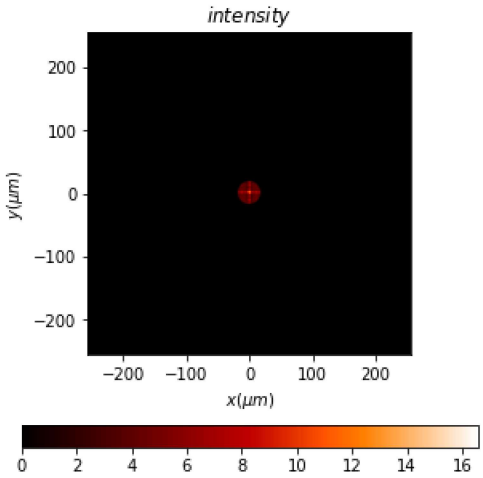


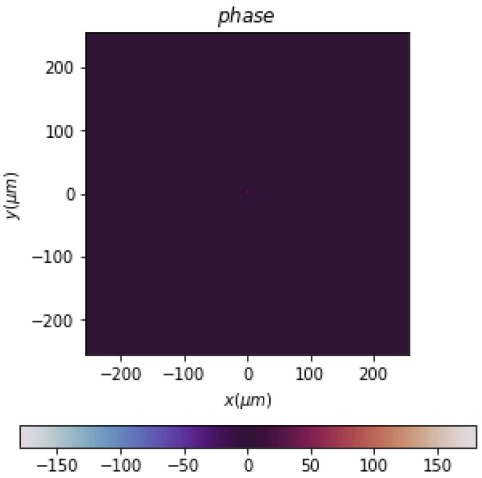
In [8]:

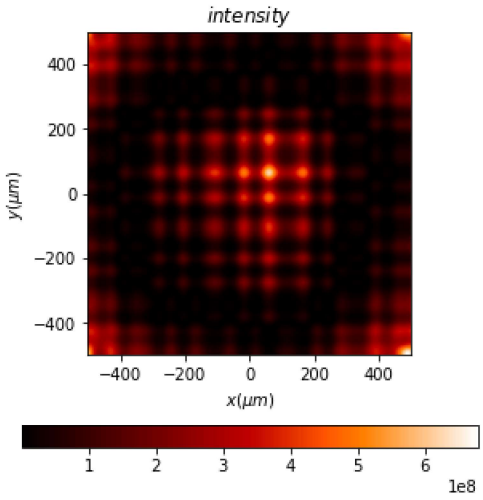
```
a_L1_Circ, a_L1_Circ_L2 = sim(a_varSlit, circ)
a_L1_Circ.draw(kind='field', logarithm=True)
a_L1_Circ_L2.draw(kind='field', logarithm=False)
```

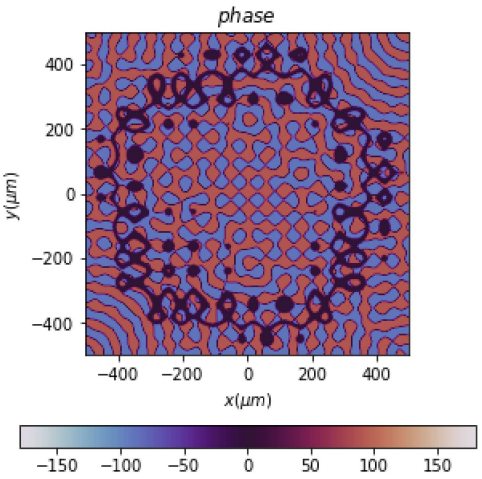
Out[8]:

```
((<matplotlib.image.AxesImage at 0x2161f583010>,
  <matplotlib.image.AxesImage at 0x2161cea29e0>),
 None,
 None)
```







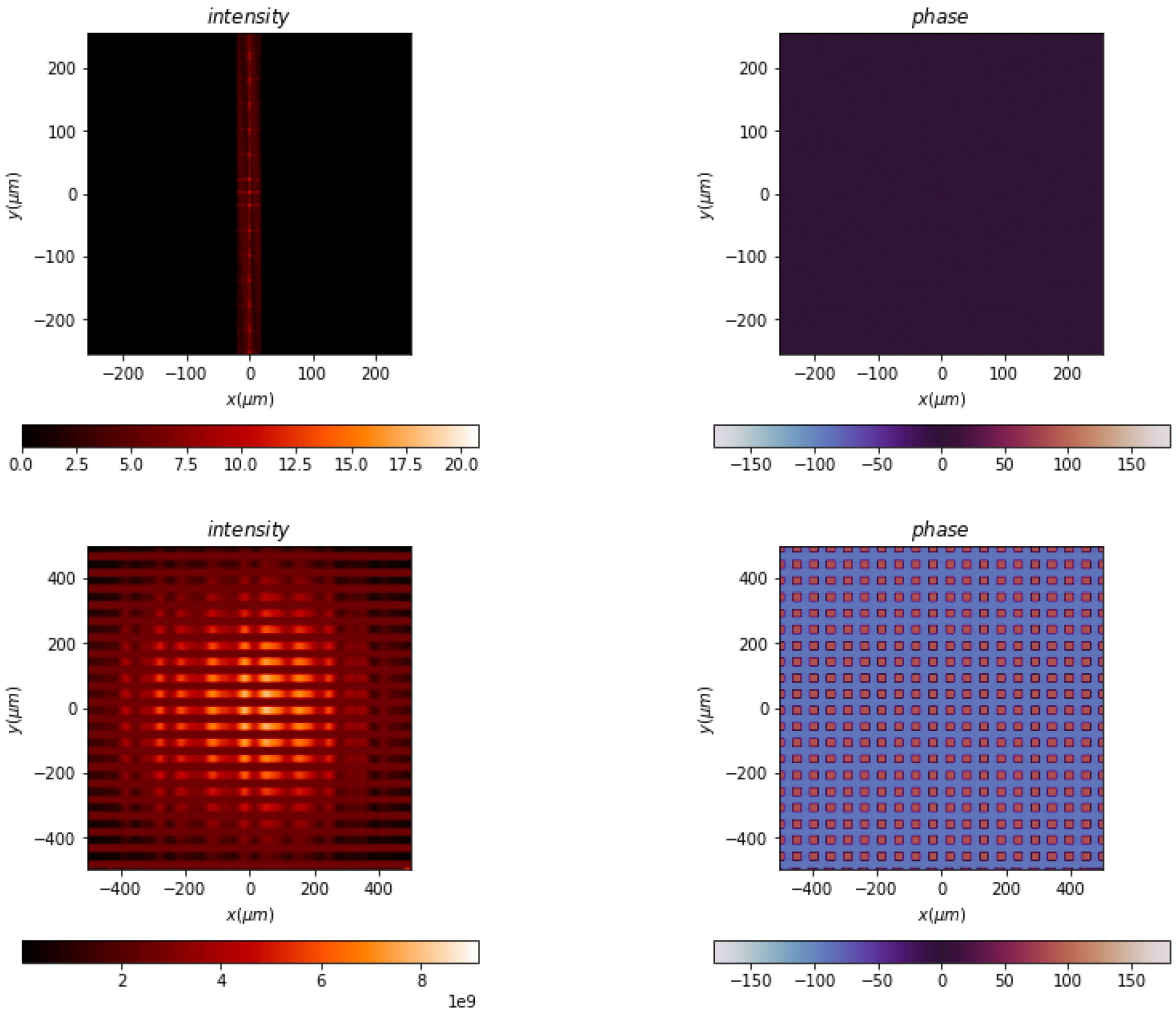


In [9]:

```
a_L1_Vert, a_L1_Vert_L2 = sim(a_varSlit, vert)
a_L1_Vert.draw(kind='field', logarithm=True)
a_L1_Vert_L2.draw(kind='field', logarithm=False)
```

Out[9]:

```
((<matplotlib.image.AxesImage at 0x2161e63ffd0>,
  <matplotlib.image.AxesImage at 0x21620f48280>),
 None,
 None)
```

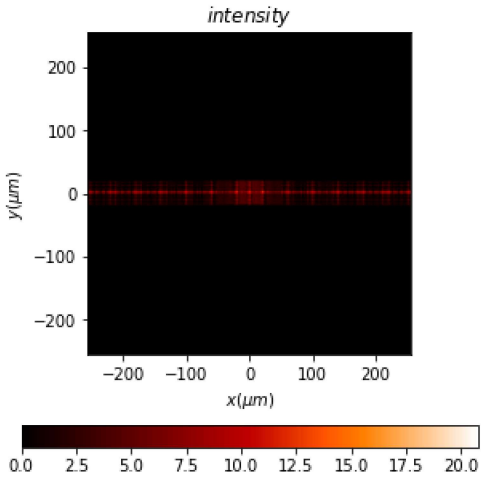


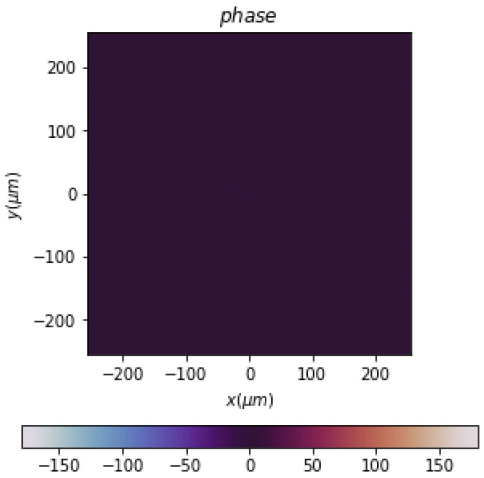
In [10]:

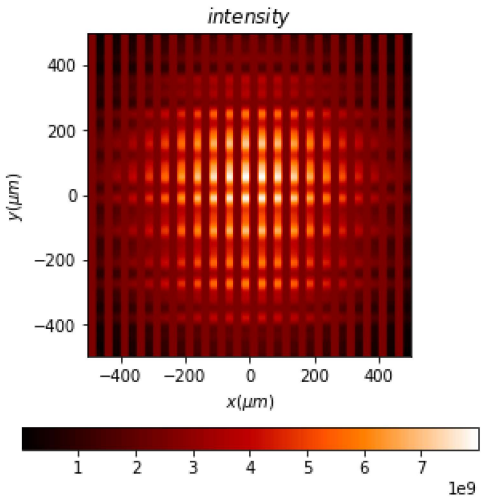
```
a_L1_Horiz, a_L1_Horiz_L2 = sim(a_varSlit, horiz)
a_L1_Horiz.draw(kind='field', logarithm=True)
a_L1_Horiz_L2.draw(kind='field', logarithm=False)
```

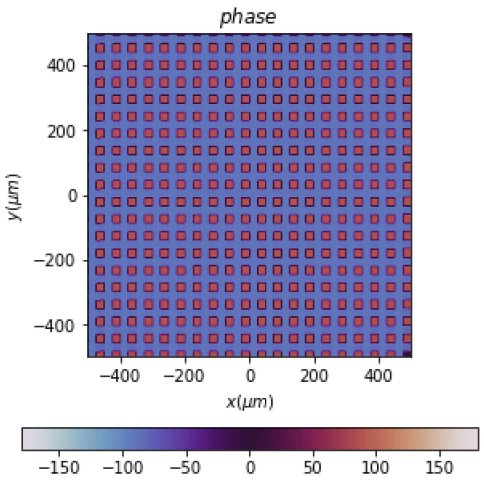
Out[10]:

```
((<matplotlib.image.AxesImage at 0x2162157f8b0>,
  <matplotlib.image.AxesImage at 0x21621e95000>),
 None,
 None)
```







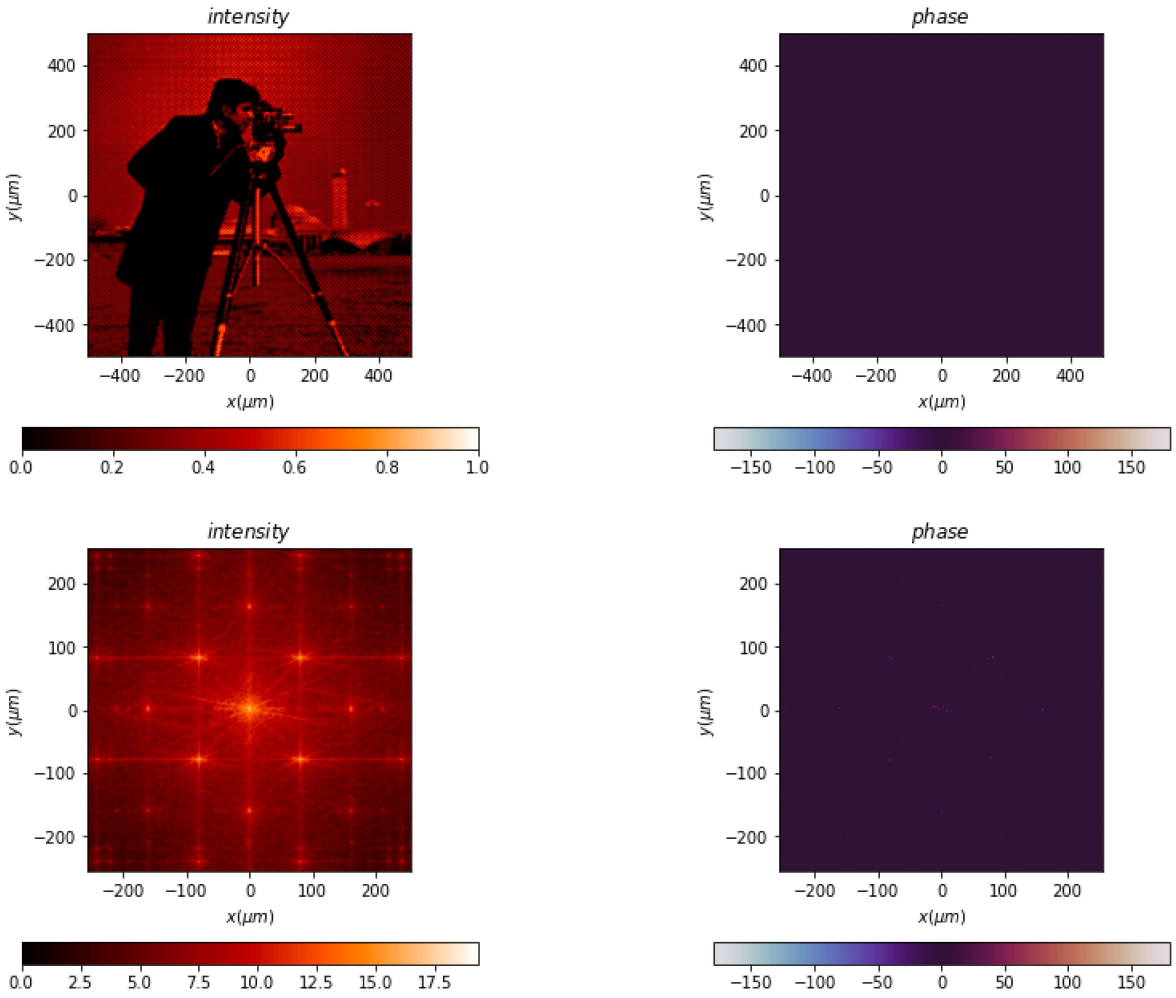


In [11]:

```
y15 = Scalar_mask_XY(x0, y0, wavelength)
y15.image(
 filename="Img.png",
 normalize=True,
 canal=2,
)
y15.draw(kind='field', logarithm=True)

# Fourier Plane - No Filter
a_L1 = (u0 * y15).fft(z=1 * mm, new_field=True)
a_L1.draw(kind='field', logarithm=True)
```

Out[11]:

```
((<matplotlib.image.AxesImage at 0x21620e6ded0>,
  <matplotlib.image.AxesImage at 0x21620db38b0>),
 None,
 None)
```

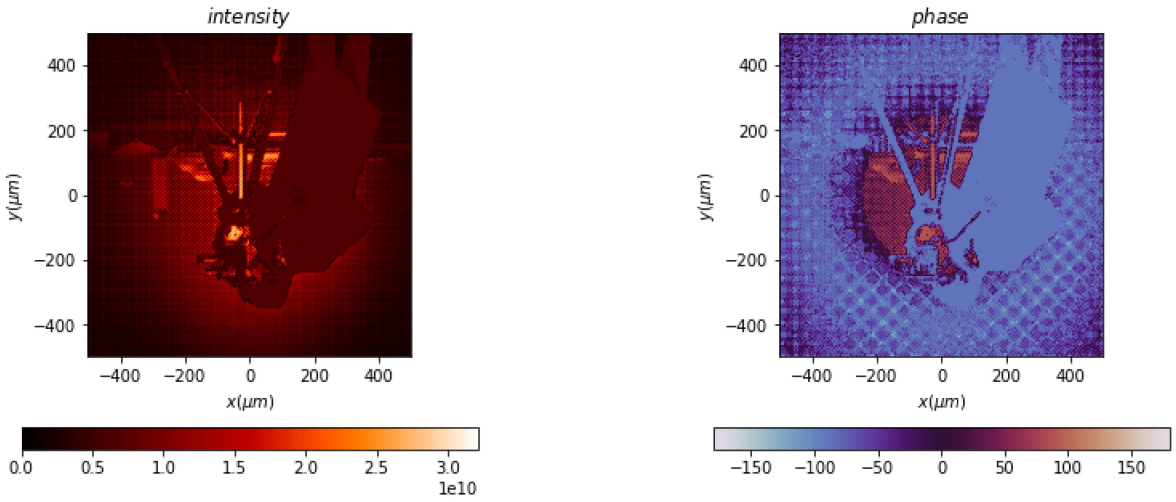


In [12]:

```
# This is how, it'd look without any filter, at the Observation screen
a_L2 = a_L1.fft(z=1 * mm, shift=False, remove0=False, new_field=True)
a_L2.draw(kind='field', logarithm=False)
```

Out[12]:

```
((<matplotlib.image.AxesImage at 0x21621faa740>,
  <matplotlib.image.AxesImage at 0x2162789c250>),
 None,
 None)
```



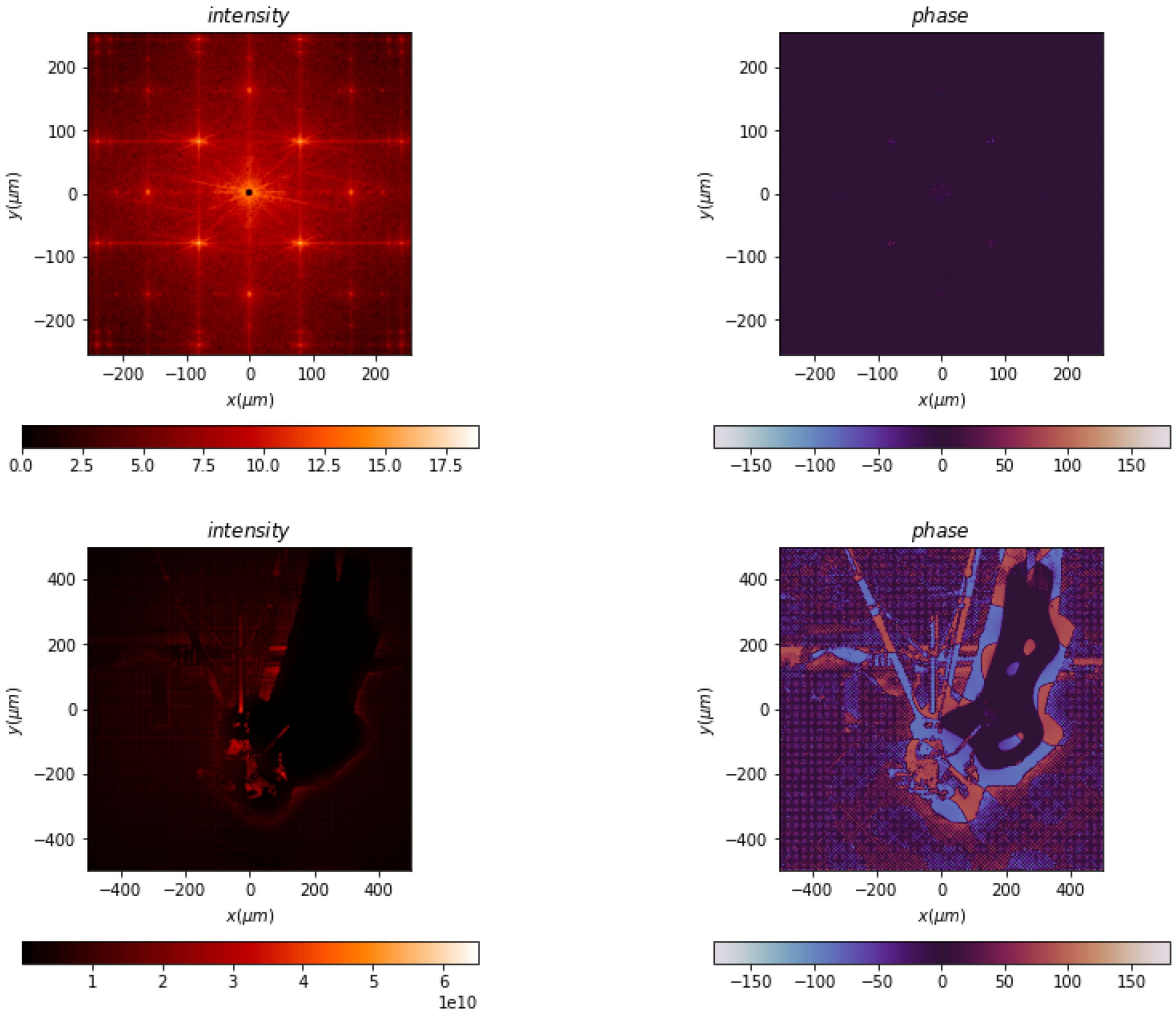
High Pass Filter

In [13]:

```
a_L1_cD, a_L1_cD_L2 = sim(a_L1, cDot)
a_L1_cD.draw(kind='field', logarithm=True)
a_L1_cD_L2.draw(kind='field', logarithm=False)
```

Out[13]:

```
((<matplotlib.image.AxesImage at 0x21622a6d4e0>,
  <matplotlib.image.AxesImage at 0x21622b0ee30>),
 None,
 None)
```



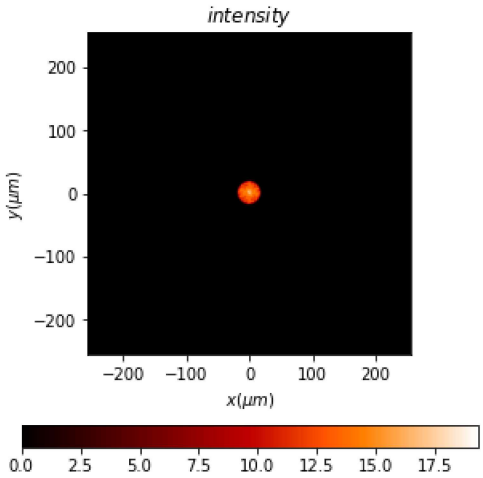
Low Pass Filter

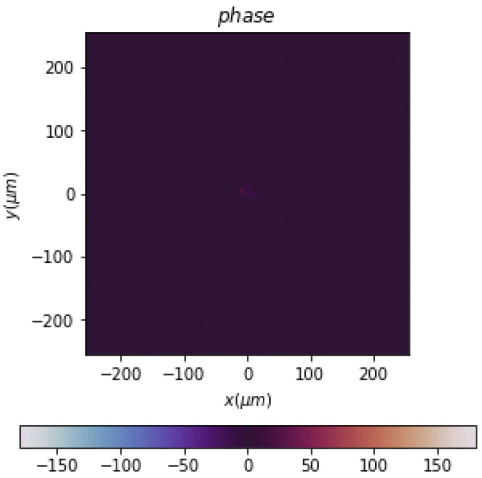
In [14]:

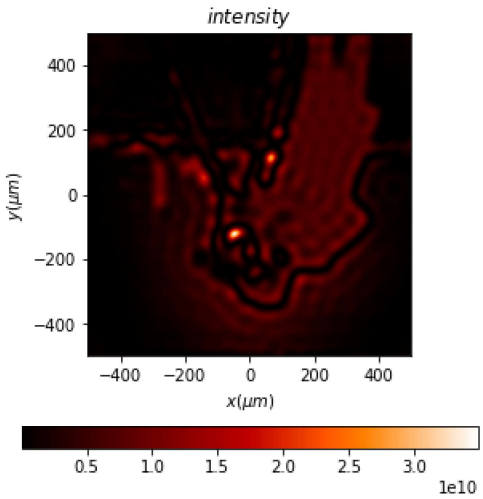
```
a_L1_Circ, a_L1_Circ_L2 = sim(a_L1, circ)
a_L1_Circ.draw(kind='field', logarithm=True)
a_L1_Circ_L2.draw(kind='field', logarithm=False)
```

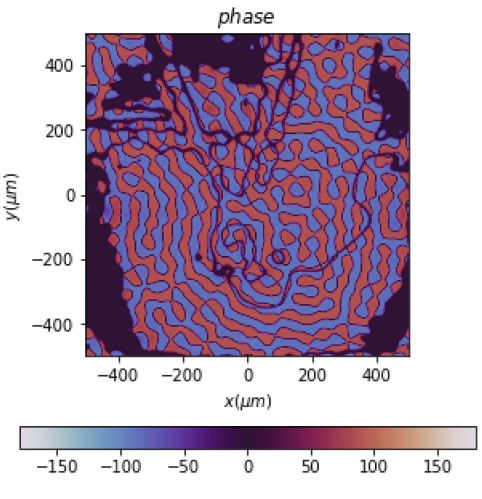
Out[14]:

```
((<matplotlib.image.AxesImage at 0x216214c7250>,
  <matplotlib.image.AxesImage at 0x21621554760>),
 None,
 None)
```









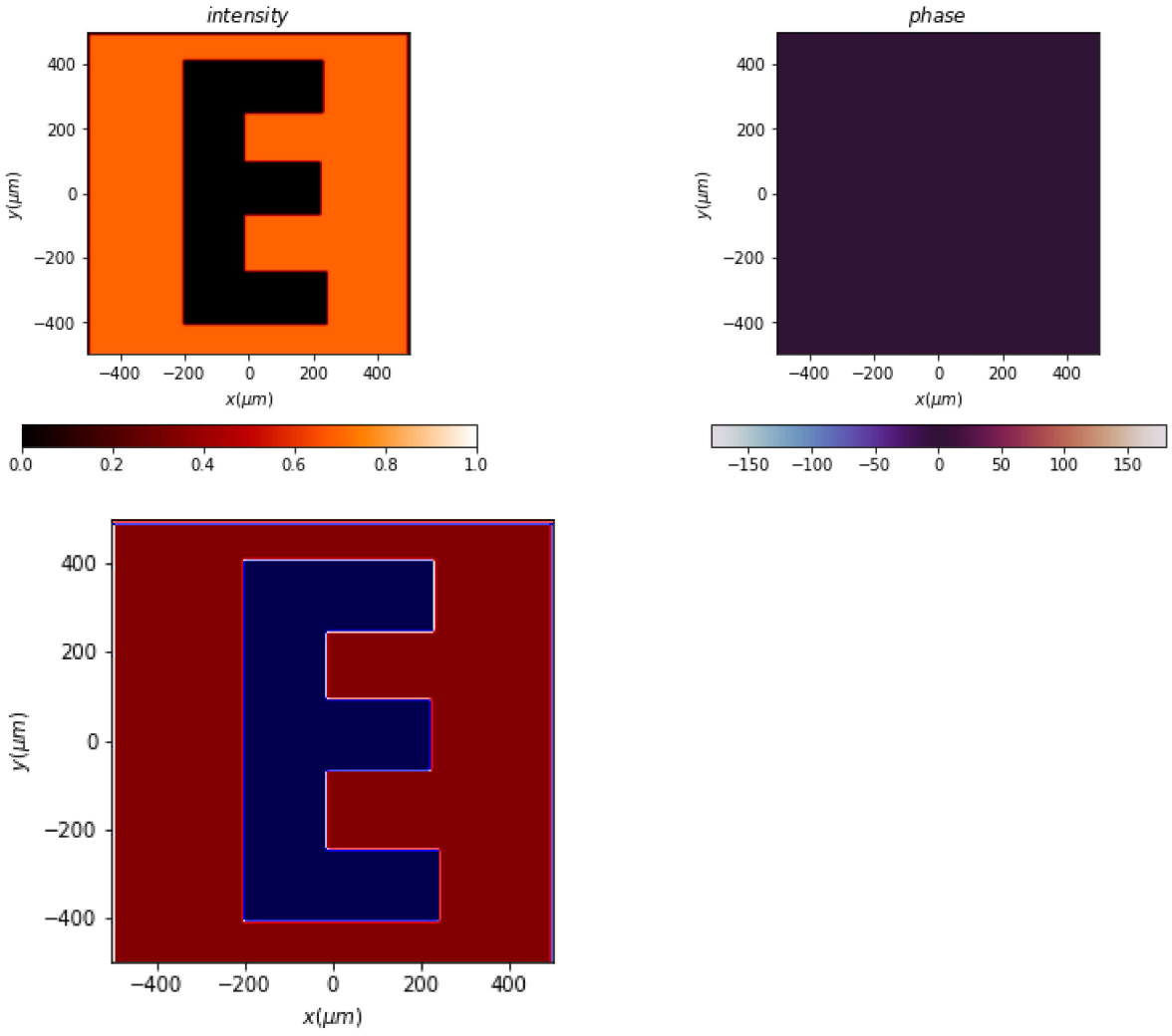
Alphabets - Spatial Filtering and Character Recognition

In [15]:

```
E = Scalar_mask_XY(x0, y0, wavelength)
E.image(
 filename="E.png",
 normalize=True,
 canal=0,
)
E.draw(kind='field', logarithm=True)
E.draw(kind='real_field', logarithm=True)
```

Out[15]:

```
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x21623986fb0>)
```

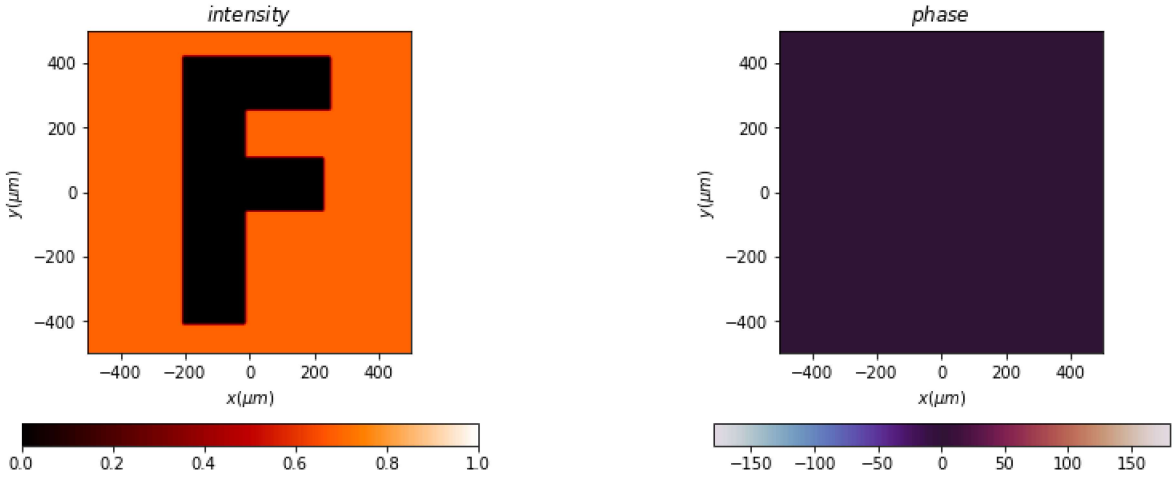


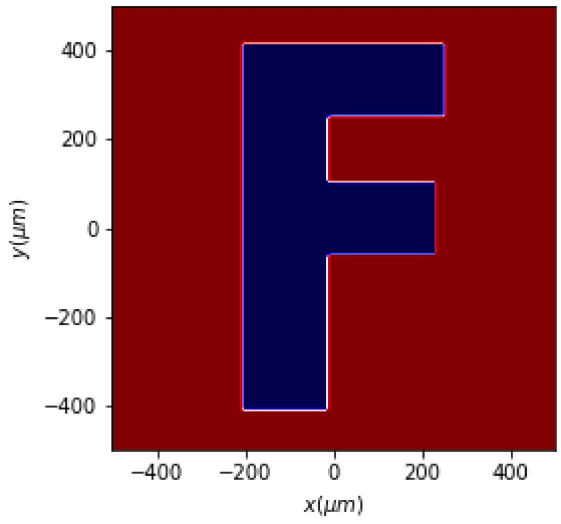
In [16]:

```
F = Scalar_mask_XY(x0, y0, wavelength)
F.image(
 filename="F.png",
 normalize=True,
 canal=0,
)
F.draw(kind='field', logarithm=True)
F.draw(kind='real_field', logarithm=True)
```

Out[16]:

```
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x216244a3e50>)
```





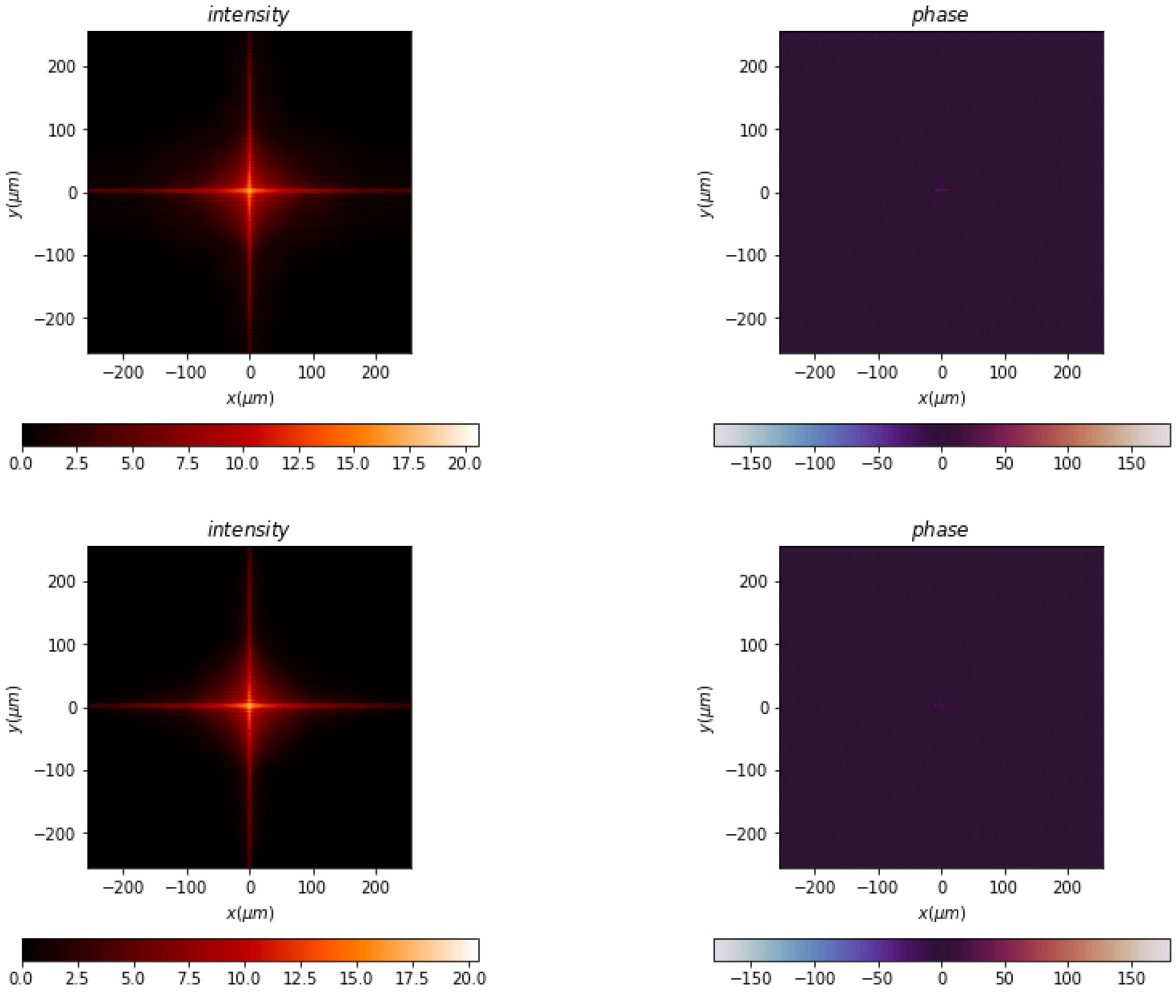
Fourier Image of E and F

In [17]:

```
# Fourier Plane - No Filter
E_a_L1 = (u0 * E).fft(z=1 * mm, new_field=True)
E_a_L1.draw(kind='field', logarithm=True)

F_a_L1 = (u0 * F).fft(z=1 * mm, new_field=True)
F_a_L1.draw(kind='field', logarithm=True)
```

Out[17]:

```
((<matplotlib.image.AxesImage at 0x216284c5690>,
  <matplotlib.image.AxesImage at 0x216239dc490>),
 None,
 None)
```



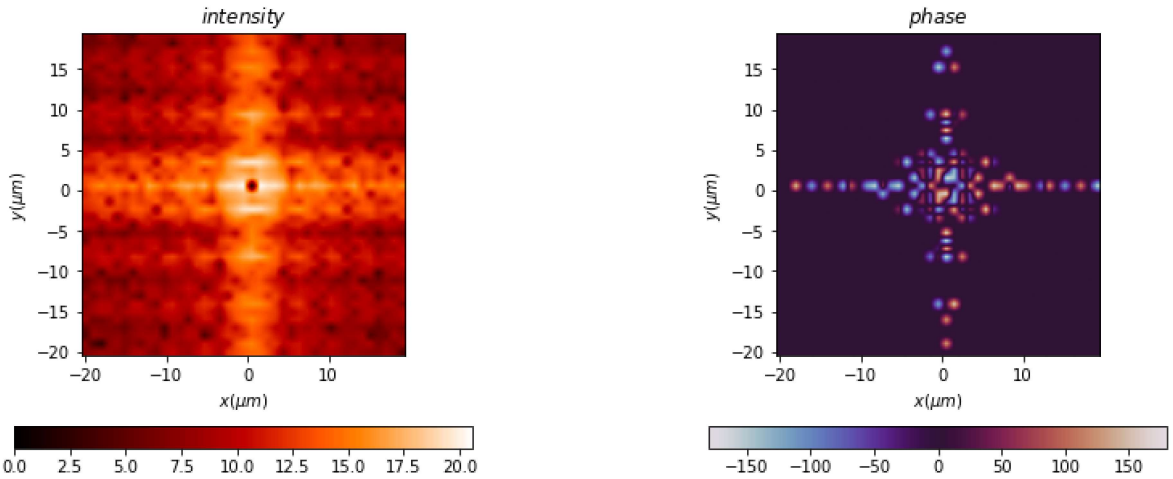
It can be seen that the Fourier Image is same. But in a closer look, the phase information if different.

In [18]:

```
# E closer look
E_a_L1_Crop = E_a_L1.cut_resample(x_limits=(-20, 20), y_limits=(-20, 20), new_field=True)
E_a_L1_Crop.draw(kind='field', logarithm=True)
```

Out[18]:

```
((<matplotlib.image.AxesImage at 0x2161dba3c40>,
  <matplotlib.image.AxesImage at 0x21621579690>),
 None,
 None)
```

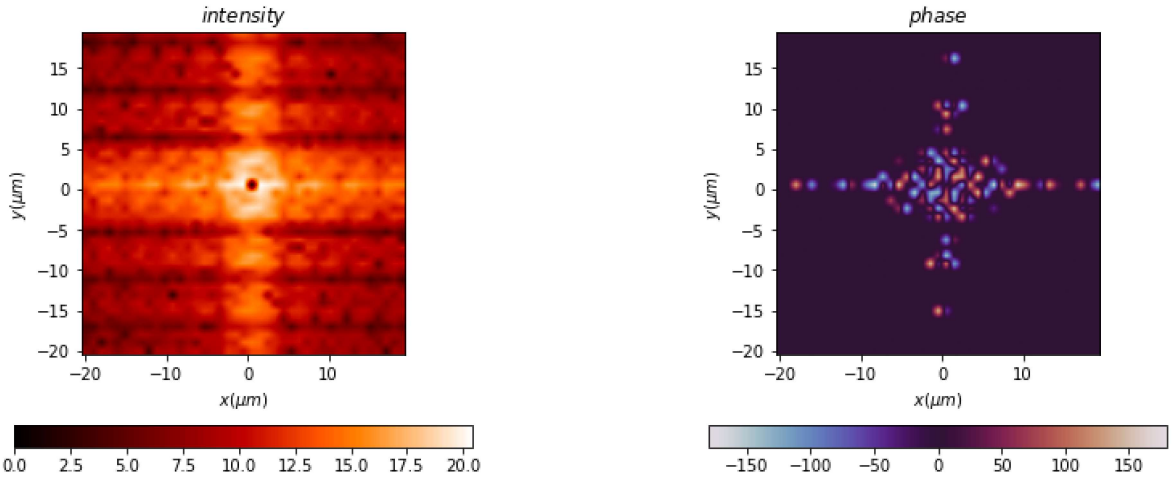


In [19]:

```
# F closer look
F_a_L1_Crop = F_a_L1.cut_resample(x_limits=(-20, 20), y_limits=(-20, 20), new_field=True)
F_a_L1_Crop.draw(kind='field', logarithm=True)
```

Out[19]:

```
((<matplotlib.image.AxesImage at 0x2161f3d58a0>,
  <matplotlib.image.AxesImage at 0x216279bdff0>),
 None,
 None)
```



Phase information is different for each alphabets, which can be later utilised.

In [ ]: