In [47]:

```
from numpy import *
from diffractio import degrees, mm, plt, sp, um, np
from diffractio.scalar_fields_XY import Scalar_field_XY
from diffractio.scalar_masks_XY import Scalar_mask_XY
from diffractio.scalar_sources_XY import Scalar_source_XY

from scipy.special import j1
from numpy.fft import fft,ifft,fft2,ifft2,fftshift,ifftshift
```

lens transform

low NA

In [2]:

```
diameter = 2 * mm
focal = 25 * mm

x0 = np.linspace(-diameter / 2, diameter / 2, 1024)
y0 = np.linspace(-diameter / 2, diameter / 2, 1024)
wavelength = 0.6238 * um
```

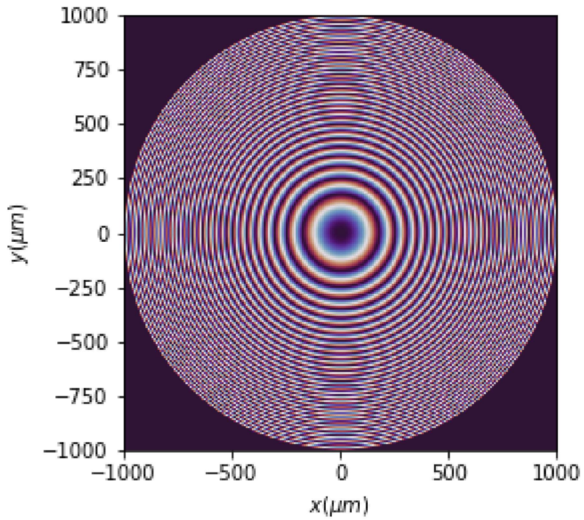
In [37]:

```
print("NA = ",1/25)
```

```
NA =  0.04
```

In [8]:

```
u0 = Scalar_source_XY(x=x0, y=y0, wavelength=wavelength)
u0.plane_wave()

t0 = Scalar_mask_XY(x=x0, y=y0, wavelength=wavelength)
t0.lens(r0=(0 * um, 0 * um),
        radius=(diameter / 2, diameter / 2),
        focal=(focal, focal))

t0.draw('phase')

u1 = u0 * t0
```



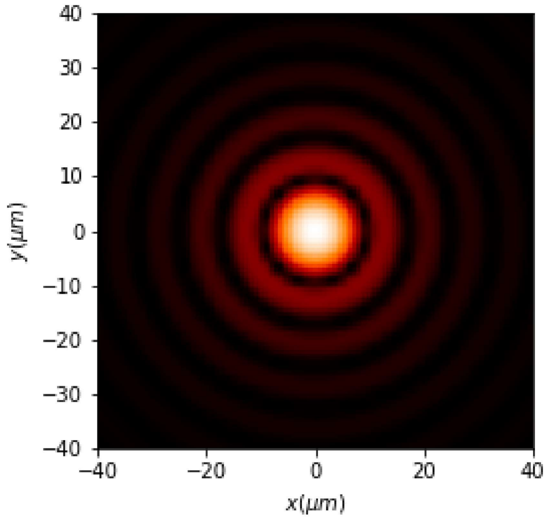
In [9]:

```
%%time
u2_rs = u1.RS(z=focal)

u2_rs.cut_resample(x_limits=(-40, 40),
                   y_limits=(-40, 40),
                   num_points=(128, 128),
                   new_field=False,
                   interp_kind=(3, 1))

u2_rs.draw(logarithm=1e-2)
```

```
Wall time: 2.33 s
```

Out[9]:

```
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x2b04fd26a90>)
```

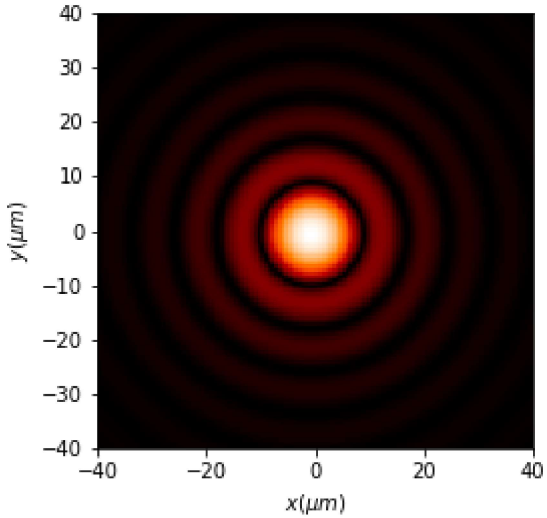


In [10]:

```
%%time
xout=np.linspace(-40,40,128)
yout=np.linspace(-40,40,128)
u2_czt = u1.CZT(z=focal, xout=xout, yout=yout)

u2_czt.draw(logarithm=1e-2)
```

Out[10]:

```
Wall time: 350 ms
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x2b04f3d17c0>)
```

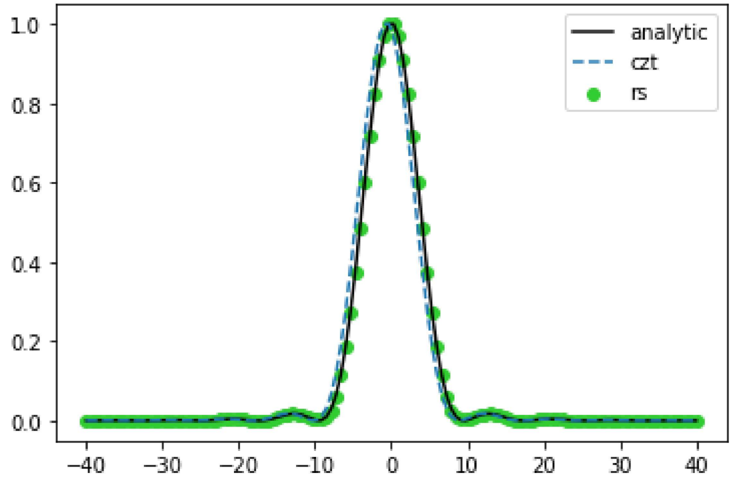


In [34]:

```
# analytical result
a = diameter/2
k = 2*np.pi/wavelength
f = focal
int_func = lambda x: (j1(a*x*k/f)/x)**2
int_analytic = int_func(xout)
int_analytic /= max(int_analytic)
# czt result
int_czt = abs(u2_czt.u[len(xout)//2])**2
norm = max(int_czt)
int_czt /= norm
int_rs = abs(u2_rs.u[len(xout)//2])**2
norm = max(int_rs)
int_rs /= norm
plt.plot(xout,int_analytic,label='analytic',c='k')
plt.plot(xout,int_czt,'--',label='czt')
plt.scatter(xout,int_rs,label='rs',c='limegreen')
plt.legend()
```

Out[34]:

```
<matplotlib.legend.Legend at 0x2b052d82070>
```



In [ ]:

high NA

fails to produce diffraction limited spot size for input grid size up to 4096 wide, but both RS and CZT remain in agreement

In [44]:

```
diameter = 2 * mm
focal = sqrt(3)*(diameter/2)

input_samples = 2**13
x0 = np.linspace(-diameter / 2, diameter / 2, input_samples)
y0 = np.linspace(-diameter / 2, diameter / 2, input_samples)
wavelength = 0.6238 * um
k = 2*pi/wavelength
```

In [96]:

```
2**12
```

Out[96]:

```
4096
```

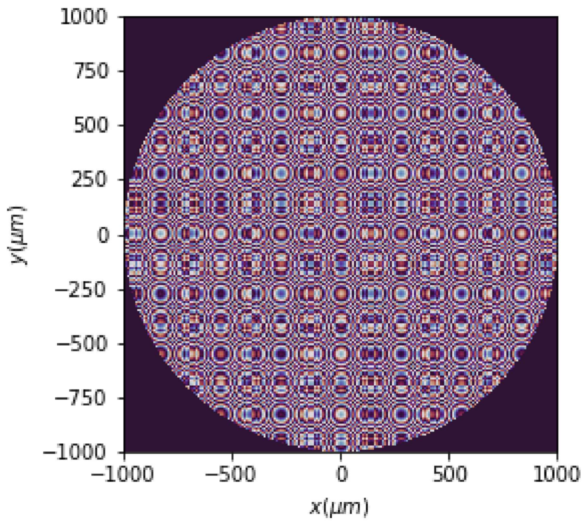
In [4]:

```
NA = sin(arctan((diameter/2)/focal))
print("NA = ",NA)
```

```
NA =  0.5
```

In [5]:

```
u0 = Scalar_source_XY(x=x0, y=y0, wavelength=wavelength)
u0.plane_wave()

t0 = Scalar_mask_XY(x=x0, y=y0, wavelength=wavelength)
t0.lens(r0=(0 * um, 0 * um),
        radius=(diameter / 2, diameter / 2),
        focal=(focal, focal))

t0.draw('phase')

u1 = u0 * t0
```

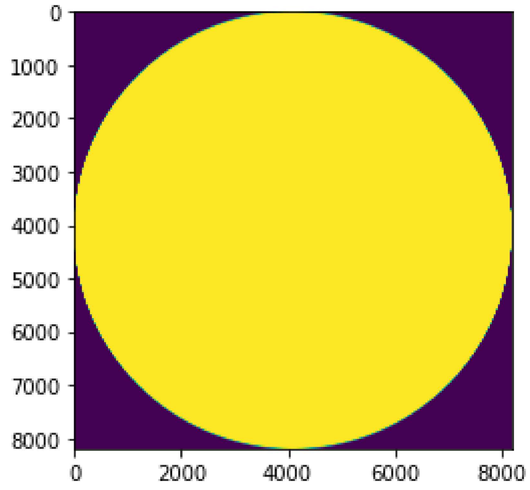


In [35]:

```
plt.imshow(abs(u1.u)**2)
```

Out[35]:

```
<matplotlib.image.AxesImage at 0x24751350460>
```



In [6]:

```
xmin = -10
xmax = -xmin
ymin = xmin
ymax = xmax
samples=512
```

In [48]:

```
%%time
N = input_samples
xspan = diameter
dx = (xspan/N)
# define k space grid and field
dfx = 1/(xspan)
fx_list = arange(-N//2,N//2,1)*dfx
kx_list = 2*pi*fx_list
ky_list = -kx_list
KX, KY = meshgrid(kx_list, ky_list)
KZ_real = real(sqrt(k**2 - KX**2 - KY**2 + 0j))
KZ_imag = imag(sqrt(k**2 - KX**2 - KY**2 + 0j))
KZ = KZ_real + 1j*abs(KZ_imag)
u1_kspace = fftshift(fft2(u1.u))

# propagate
z_distance = focal
phase = f*KZ_real % 2*pi
u2_kspace = u1_kspace*exp(1j*z_distance*KZ)
```

```python
# transform back to real space
u2 = ifft2(u2_kspace)
int_asm = real(conjugate(u2)*u2)
int_asm /= amax(u2)
```

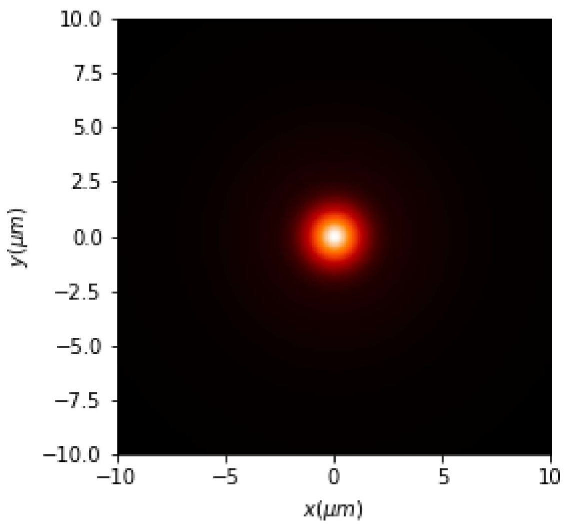
```
Wall time: 28.1 s
```

In [51]:

```python
u2_asm = Scalar_source_XY(x=x0, y=y0, wavelength=wavelength)
u2_asm.u = u2
u2_asm.cut_resample(x_limits=(xmin, xmax),
                    y_limits=(ymin, ymax),
                    num_points=(samples,samples),
                    new_field=False,
                    interp_kind=(3, 1))
u2_asm.draw(logarithm=False)
```

Out[51]:

```
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x24751006c70>)
```



In [34]:

```python
dx = (diameter/input_samples)/um
assert wavelength/(2*dx) < 1, wavelength/(2*dx)
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_5564/4059008104.py in <module>
      1 dx = (diameter/input_samples)/um
```

```
----> 2 assert wavelength/(2*dx) < 1, wavelength/(2*dx)

AssertionError: 1.2775424
```

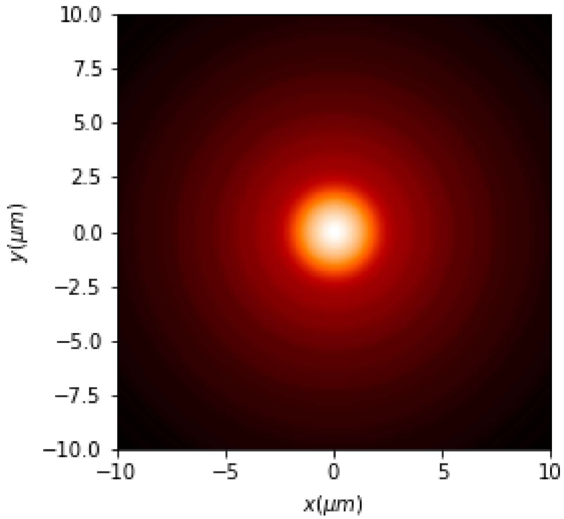
In [7]:

```
%%time
u2_rs = u1.RS(z=focal, verbose=True)

u2_rs.cut_resample(x_limits=(xmin, xmax),
                   y_limits=(ymin, ymax),
                   num_points=(samples,samples),
                   new_field=False,
                   interp_kind=(3, 1))
u2_rs.draw(logarithm=False)
```

```
Wall time: 5min 51s 2.86
```

Out[7]:

```
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x2474f9f9a00>)
```

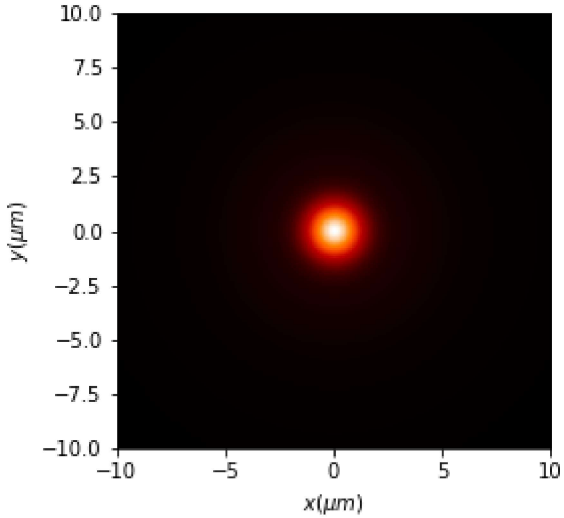


In [29]:

```
u2_rs.draw(logarithm=False)
```

Out[29]:

```
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x24751296e50>)
```



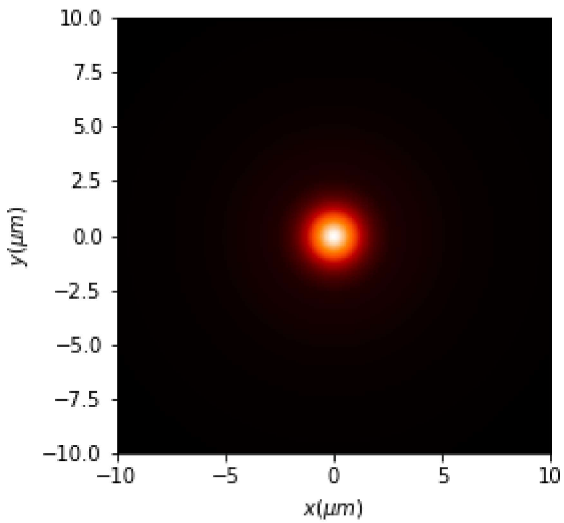
In [12]:

```
%%time
xout=np.linspace(xmin,xmax,samples)
yout=np.linspace(ymin,ymax,samples)
u2_czt = u1.CZT(z=focal, xout=xout, yout=yout,
               verbose=True)

u2_czt.draw(logarithm=False) #logarithm=1e-2)
```

```
num x, num y, num z = 512, 512, 1
Wall time: 17.7 s
```

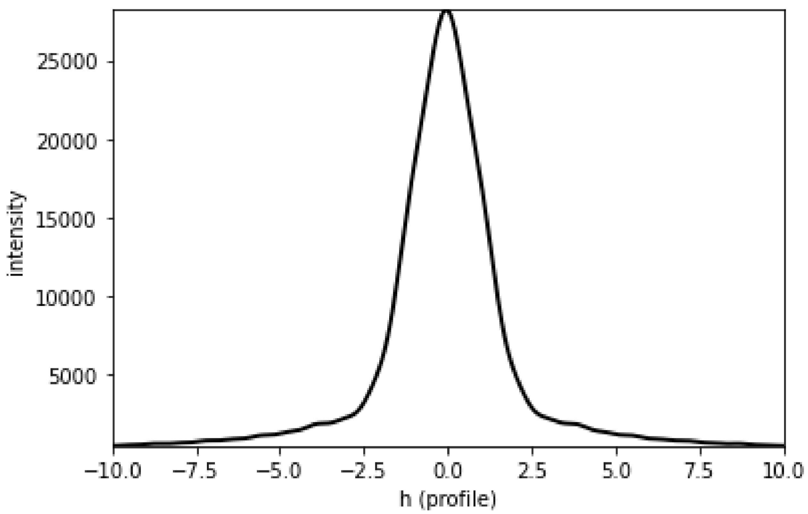
Out[12]:

```
(<Figure size 432x288 with 1 Axes>,
 <AxesSubplot:xlabel='$x  (\\mu m)$', ylabel='$y  (\\mu m)$'>,
 <matplotlib.image.AxesImage at 0x2474ff99e20>)
```



In [28]:

```
u2_czt.draw_profile(point1=(xmin,0),point2=(xmax,0))
plt.show()
```



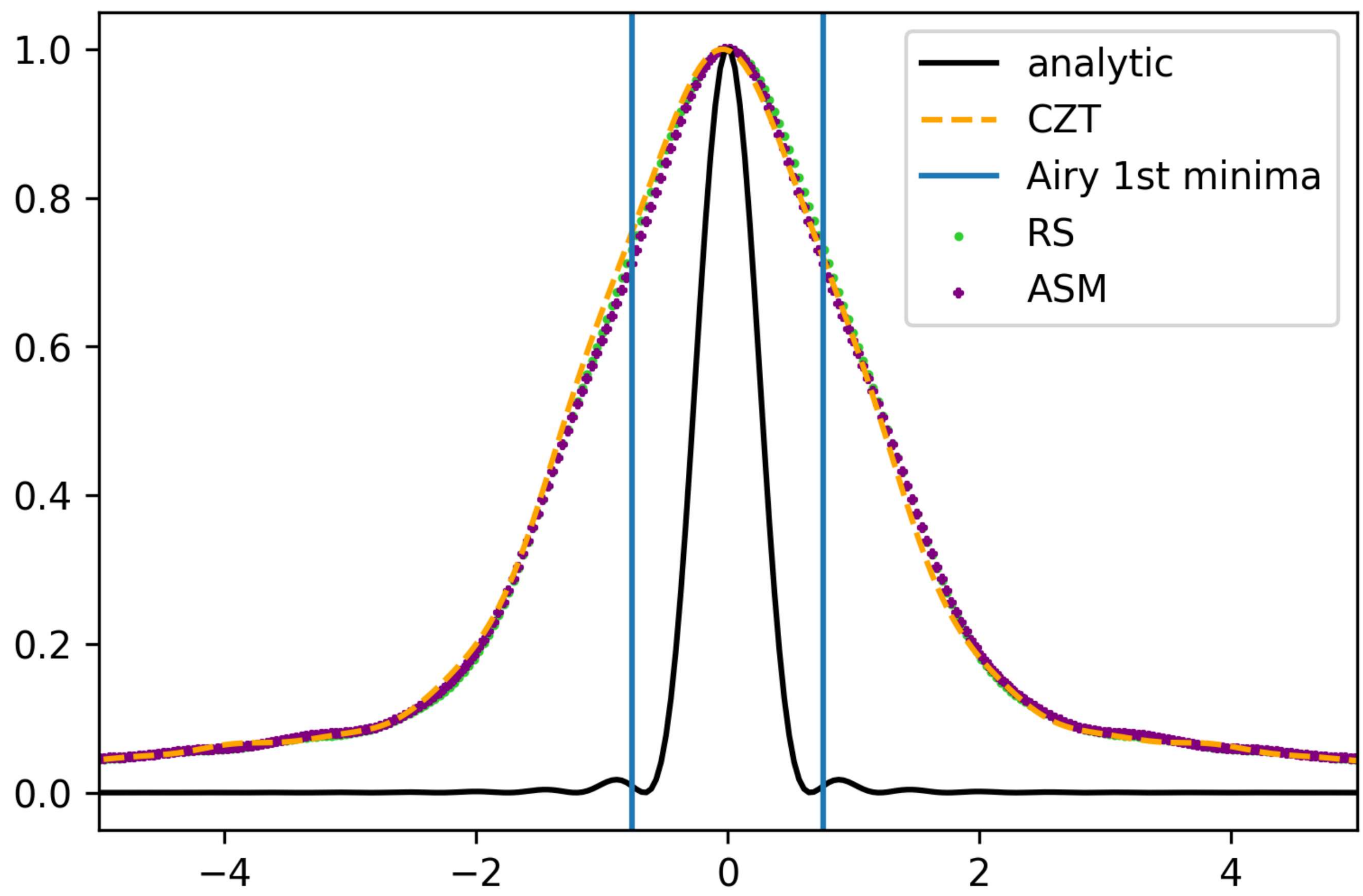
In [78]:

```
# analytical result
a = diameter/2
k = 2*np.pi/wavelength
f = focal
int_func = lambda x: (j1(a*x*k/f)/x)**2
int_analytic = int_func(xout)
```

```
int_analytic /= max(int_analytic)

int_czt = abs(u2_czt.u[len(xout)//2])**2
norm = max(int_czt)
int_czt /= norm

int_asm = abs(u2_asm.u[len(xout)//2])**2
norm = max(int_asm)
int_asm /= norm

int_rs = abs(u2_rs.u[len(xout)//2])**2
norm = max(int_rs)
int_rs /= norm

fig,ax = plt.subplots(dpi=300)
ax.plot(xout,int_analytic,label='analytic',c='k')
ax.plot(xout,int_czt,'--',label='CZT',c='orange')
ax.scatter(xout,int_rs,label='RS',c='limegreen',s=2)
ax.scatter(xout,int_asm,label='ASM',c='purple',s=4,marker="P")
ax.set_xlim((-5,5))
plt.axvline(-1.22*wavelength/(2*NA),label='Airy 1st minima')
plt.axvline(1.22*wavelength/(2*NA))
plt.legend()
```

Out[78]:

```
<matplotlib.legend.Legend at 0x247549f1730>
```



In [110…

```
1.22*0.78*um/0.6
```

Out[110…

```
1.586
```

In [6]:

```
xmin = -10
xmax = -xmin
```

```
ymin = xmin
ymax = xmax
samples=512
```