

```
In [1]: %matplotlib inline
```

```
In [2]: import numpy as np
from matplotlib import pyplot as plt
from matplotlib.colors import LogNorm, Normalize
from pathlib import Path
```

```
In [3]: basefile = Path("./LWA")
filename = "EPIC_1518451851.248535_25.610MHz.npz"
```

```
In [4]: original_file = basefile / "master_output" / "no_dft" / filename
orig = np.load(original_file, allow_pickle=True)
```

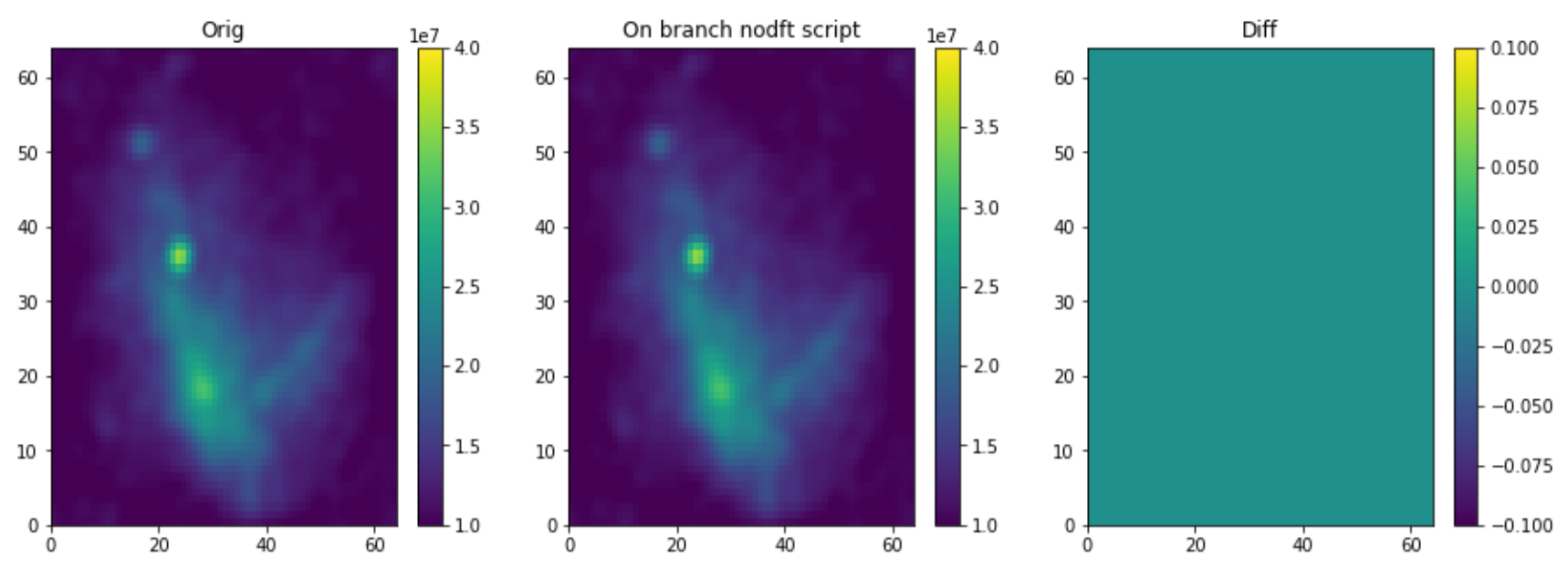
```
In [5]: new_file = basefile / "script_consolidation" / "single_script" / "no_dft" / filename
# new_file = basefile / "script_consolidation" / "single_script" / "no_dft" / filename
new = np.load(new_file, allow_pickle=True)
```

```
In [6]: fig, ax = plt.subplots(ncols=3, figsize=(15,5))

# norm = LogNorm(vmax=4e7, vmin=1e7)
norm = Normalize(vmax=4e7, vmin=1e7)
im = ax[0].pcolorfast(orig["image"][0,0,0].real, norm=norm)
fig.colorbar(im, ax=ax[0]);
ax[0].set_title("Orig");

im = ax[1].pcolorfast(new["image"][0,0,0].real, norm=norm)
fig.colorbar(im, ax=ax[1]);
ax[1].set_title("On branch nodft script");

im = ax[2].pcolorfast((orig["image"][0,0,0] - new["image"][0,0,0]).real)
fig.colorbar(im, ax=ax[2]);
ax[2].set_title("Diff");
```



```
In [7]: np.allclose(orig["image"], new["image"])
Out[7]: True
```

Check the python 3 execution.

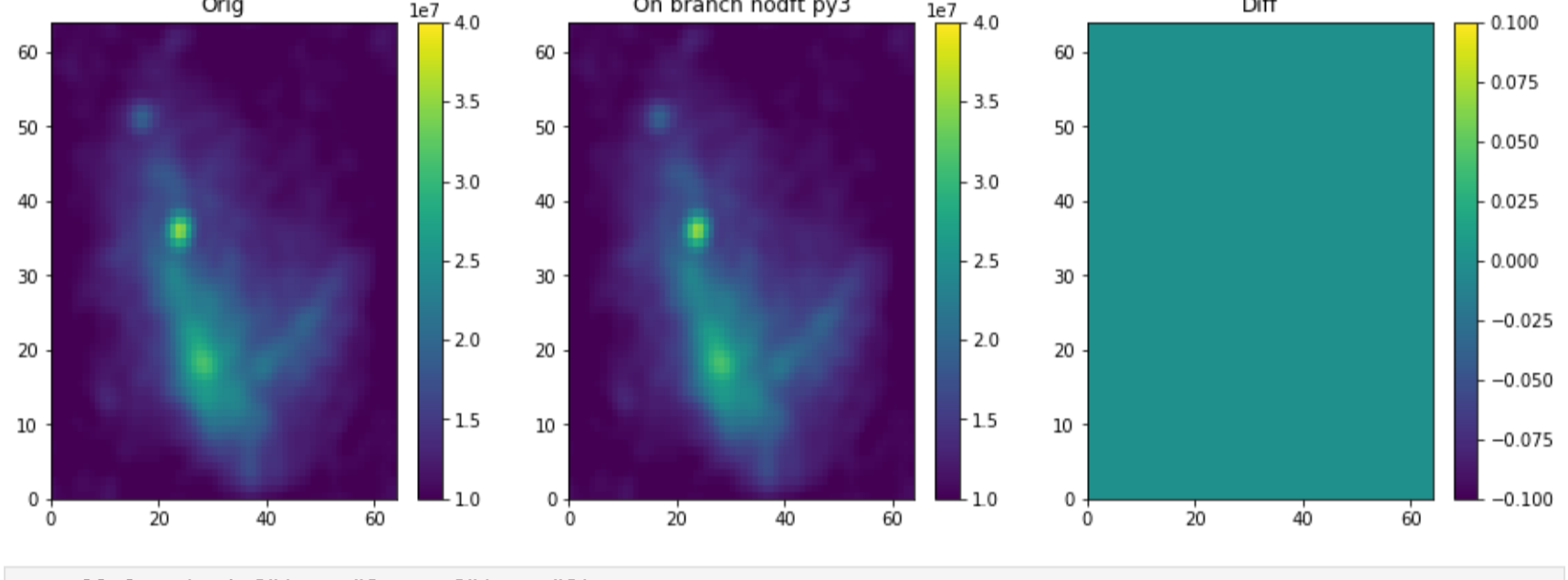
```
In [8]: new_file = basefile / "py3_test" / "no_dft" / filename
# new_file = basefile / "script_consolidation" / "single_script" / "no_dft" / filename
new = np.load(new_file, allow_pickle=True)
```

```
In [17]: fig, ax = plt.subplots(ncols=3, figsize=(15,5))

# norm = LogNorm(vmax=4e7, vmin=1e7)
norm = Normalize(vmax=4e7, vmin=1e7)
im = ax[0].pcolorfast(orig["image"][0,0,0].real, norm=norm)
fig.colorbar(im, ax=ax[0]);
ax[0].set_title("Orig");

im = ax[1].pcolorfast(new["image"][0,0,0].real, norm=norm)
fig.colorbar(im, ax=ax[1]);
ax[1].set_title("On branch nodft py3");

im = ax[2].pcolorfast((orig["image"][0,0,0] - new["image"][0,0,0]).real)
fig.colorbar(im, ax=ax[2]);
ax[2].set_title("Diff");
```



```
In [10]: np.allclose(orig["image"], new["image"])
Out[10]: True
```

```
In [11]: orig_dft_file = basefile / "master_output" / "dft" / filename
orig_dft = np.load(orig_dft_file, allow_pickle=True)
```

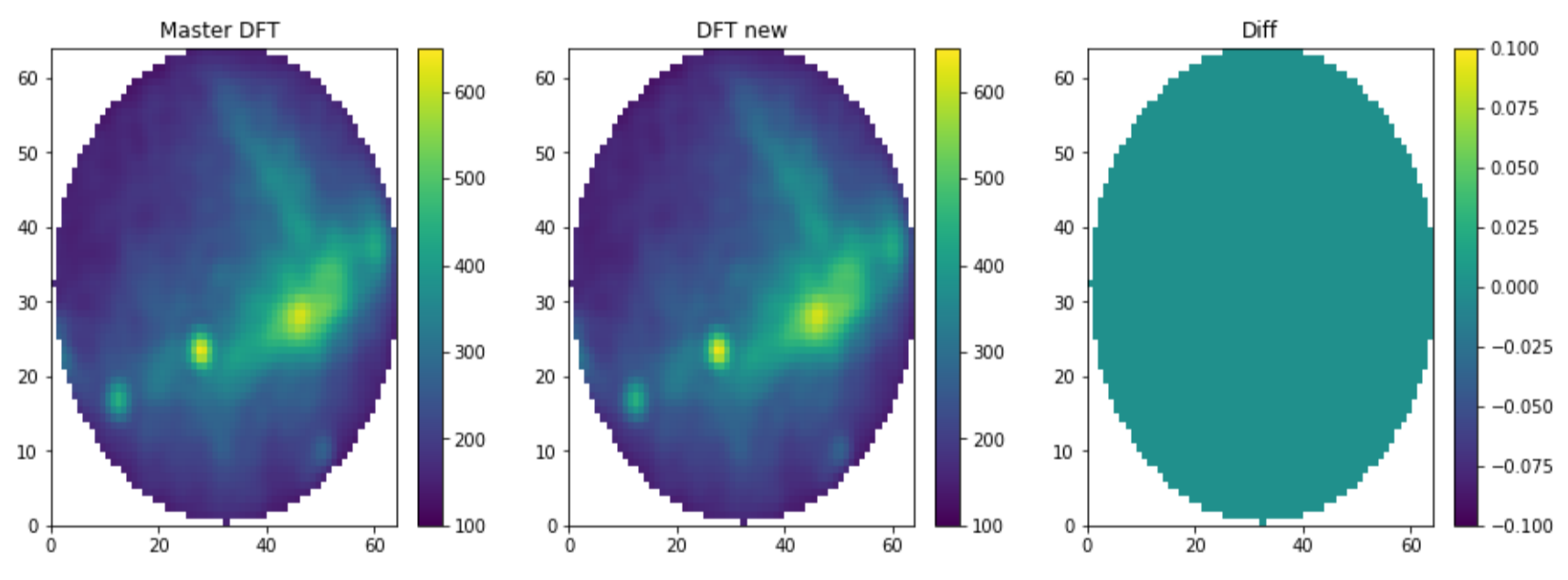
```
In [12]: new_dft_file = basefile / "script_consolidation" / "single_script" / "dft" / filename
dft = np.load(new_dft_file, allow_pickle=True)
```

```
In [13]: fig, ax = plt.subplots(ncols=3, figsize=(15,5))

inds = (0, 0, 0)
# norm = LogNorm(vmax=4e7, vmin=1e7)
norm = Normalize(vmax=650, vmin=100)
im = ax[0].pcolorfast(orig_dft["image"][inds].reshape(64, 64).real, norm=norm)
fig.colorbar(im, ax=ax[0]);
ax[0].set_title("Master DFT ");

im = ax[1].pcolorfast(dft["image"][inds].reshape(64, 64).real, norm=norm)
fig.colorbar(im, ax=ax[1]);
ax[1].set_title("DFT new");

im = ax[2].pcolorfast(
    (orig_dft["image"][inds] - dft["image"][inds]).reshape(64, 64).real
)
fig.colorbar(im, ax=ax[2]);
ax[2].set_title("Diff");
```



```
In [14]: for i in range(40):
    print(i,
          np.allclose(
              np.ma.masked_invalid(orig_dft["image"][i]),
              np.ma.masked_invalid(dft["image"][i])
          )
    )
```

```
0 True
1 True
2 False
3 False
4 False
5 False
6 True
7 True
8 True
9 False
10 True
11 False
12 False
13 False
14 False
15 True
16 True
17 False
18 True
19 False
20 True
21 False
22 True
23 True
24 False
25 False
26 True
27 False
28 True
29 False
30 True
31 True
32 False
33 False
34 False
35 False
36 True
37 True
38 True
39 True
```

```
In [15]: np.max(np.abs(np.ma.masked_invalid(orig_dft["image"] - dft["image"])))
```

410.21875

```
In [16]: np.unravel_index(
    np.argmax(
        np.abs(
            np.ma.masked_invalid(orig_dft["image"] - dft["image"])
        )
    ),
    dft["image"].shape,
)
```

(24, 2, 0, 1500)

```
In [ ]:
```