

dft_comparison

September 16, 2020

```
[1]: %matplotlib inline

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

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

[4]: run1_file = basefile / "script_consolidation" / "run_1" / filename
      run1 = np.load(run1_file, allow_pickle=True)

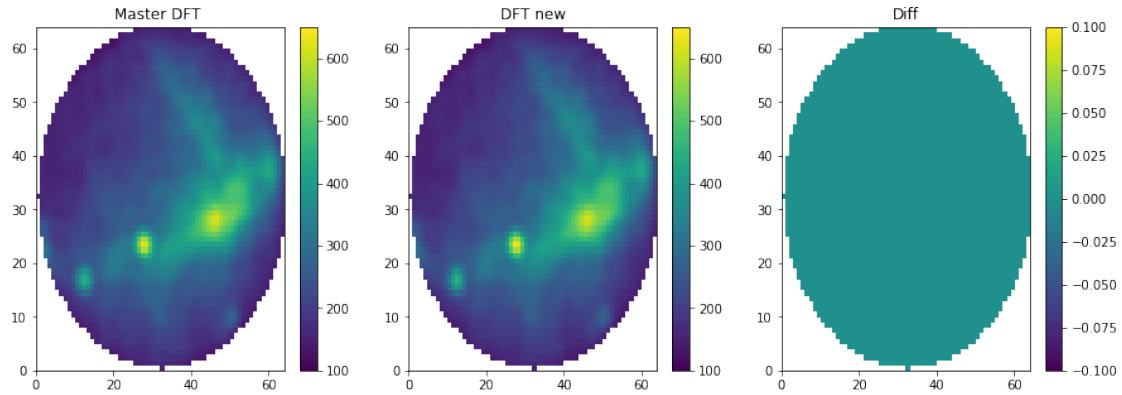
[5]: run2_file = basefile / "script_consolidation" / "run_2" / filename
      run2 = np.load(run2_file, allow_pickle=True)

[6]: 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(run1["image"][inds].reshape(64, 64).real, norm=norm)
      fig.colorbar(im, ax=ax[0]);
      ax[0].set_title("Master DFT ");

      im = ax[1].pcolorfast(run2["image"][inds].reshape(64,64).real, norm=norm)
      fig.colorbar(im, ax=ax[1]);
      ax[1].set_title("DFT new");

      im = ax[2].pcolorfast(
          (run1["image"][inds] - run2["image"][inds]).reshape(64, 64).real
      )
      fig.colorbar(im, ax=ax[2]);
      ax[2].set_title("Diff");
```



```
[7]: for i in range(40):
      print(i,
            np.allclose(
                np.ma.masked_invalid(run1["image"][i]),
                np.ma.masked_invalid(run2["image"][i])
            )
      )
```

```
0 True
1 True
2 True
3 True
4 True
5 True
6 True
7 True
8 True
9 True
10 True
11 True
12 True
13 True
14 True
15 True
16 True
17 True
18 True
19 True
20 True
21 True
22 True
23 True
24 True
```

```
25 True
26 True
27 True
28 True
29 True
30 True
31 True
32 True
33 True
34 True
35 True
36 True
37 True
38 True
39 True
```

```
[8]: for i in range(run1["image"].shape[0]):
    _max = np.max(np.abs(np.ma.masked_invalid(run1["image"][i] -
    ↪run2["image"][i])))
    _std = np.std(np.abs(np.ma.masked_invalid(run1["image"][i] -
    ↪run2["image"][i])))
    _ind = np.unravel_index(
        np.argmax(
            np.abs(
                np.ma.masked_invalid(run1["image"][i]-run2["image"][i])
            )
        ),
        run1["image"][i].shape,
    )
    print(i, _ind, _std, _max)
```

```
0 (0, 0, 32) 0.0 0.0
1 (0, 0, 32) 0.0 0.0
2 (0, 0, 32) 0.0 0.0
3 (0, 0, 32) 0.0 0.0
4 (0, 0, 32) 0.0 0.0
5 (0, 0, 32) 0.0 0.0
6 (0, 0, 32) 0.0 0.0
7 (0, 0, 32) 0.0 0.0
8 (0, 0, 32) 0.0 0.0
9 (0, 0, 32) 0.0 0.0
10 (0, 0, 32) 0.0 0.0
11 (0, 0, 32) 0.0 0.0
12 (0, 0, 32) 0.0 0.0
13 (0, 0, 32) 0.0 0.0
14 (0, 0, 32) 0.0 0.0
15 (0, 0, 32) 0.0 0.0
16 (0, 0, 32) 0.0 0.0
```

```
17 (0, 0, 32) 0.0 0.0
18 (0, 0, 32) 0.0 0.0
19 (0, 0, 32) 0.0 0.0
20 (0, 0, 32) 0.0 0.0
21 (0, 0, 32) 0.0 0.0
22 (0, 0, 32) 0.0 0.0
23 (0, 0, 32) 0.0 0.0
24 (0, 0, 32) 0.0 0.0
25 (0, 0, 32) 0.0 0.0
26 (0, 0, 32) 0.0 0.0
27 (0, 0, 32) 0.0 0.0
28 (0, 0, 32) 0.0 0.0
29 (0, 0, 32) 0.0 0.0
30 (0, 0, 32) 0.0 0.0
31 (0, 0, 32) 0.0 0.0
32 (0, 0, 32) 0.0 0.0
33 (0, 0, 32) 0.0 0.0
34 (0, 0, 32) 0.0 0.0
35 (0, 0, 32) 0.0 0.0
36 (0, 0, 32) 0.0 0.0
37 (0, 0, 32) 0.0 0.0
38 (0, 0, 32) 0.0 0.0
39 (0, 0, 32) 0.0 0.0
```

```
[34]: np.unravel_index(
      np.argmax(
        np.abs(
          np.ma.masked_invalid(run1["image"]-run2["image"])
        )
      ),
      run1["image"].shape,
    )
```

```
[34]: (21, 2, 0, 1500)
```

```
[ ]:
```