In [3]:
```
%load_ext line_profiler
```

In [4]:
```
%load_ext memory_profiler
```

In [5]:
```
import numpy as np
```

In [6]:
```
def make_lm_matrix(skymodes1d):
    lm_matrix = np.zeros((skymodes1d,skymodes1d, 3))
    lm_step = 2.0 / skymodes1d
    for i in np.arange(lm_matrix.shape[0]):
        for j in np.arange(lm_matrix.shape[0]):
            lm_matrix[i, j] = np.asarray([i * lm_step - 1.0, j * lm_step - 1.0, 0.0])
    return lm_matrix
```

In [7]:
```
def new_lm_matrix(skymodes1d):
    lm_step = 2.0 / skymodes1d
    i, j = np.meshgrid(np.arange(skymodes1d), np.arange(skymodes1d))
    lm_matrix = np.asarray([i * lm_step - 1.0, j * lm_step - 1.0, np.zeros_like(j)])
    return lm_matrix.transpose([2, 1, 0])
```

In [8]:
```
np.array_equal(make_lm_matrix(64), new_lm_matrix(64))
```

Out[8]:
```
True
```

In [9]:
```
%lprun -f make_lm_matrix make_lm_matrix(64)
```

```
Timer unit: 1e-06 s

Total time: 0.054255 s
File: <ipython-input-6-38cd79932587>
Function: make_lm_matrix at line 1

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
     1                                           def make_lm_matrix(skymodes1d):
     2         1         19.0     19.0      0.0      lm_matrix = np.zeros((skymodes1d,skymodes1d, 3))
     3         1          2.0      2.0      0.0      lm_step = 2.0 / skymodes1d
     4        65         45.0      0.7      0.1      for i in np.arange(lm_matrix.shape[0]):
     5      4160       3232.0      0.8      6.0          for j in np.arange(lm_matrix.shape[0]):
     6      4096      50956.0     12.4     93.9              lm_matrix[i, j] = np.asarray([i * lm_step - 1.
0, j * lm_step - 1.0, 0.0])
     7         1          1.0      1.0      0.0      return lm_matrix
```

In [10]:
```
%lprun -f new_lm_matrix new_lm_matrix(64)
```

```
Timer unit: 1e-06 s

Total time: 0.000748 s
File: <ipython-input-7-006bf9753be6>
Function: new_lm_matrix at line 1

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
     1                                           def new_lm_matrix(skymodes1d):
     2         1          3.0      3.0      0.4      lm_step = 2.0 / skymodes1d
     3         1        210.0    210.0     28.1      i, j = np.meshgrid(np.arange(skymodes1d), np.arange(sky
modes1d))
     4         1        530.0    530.0     70.9      lm_matrix = np.asarray([i * lm_step - 1.0, j * lm_step
- 1.0, np.zeros_like(j)])
     5         1          5.0      5.0      0.7      return lm_matrix.transpose([2, 1, 0])
```

In [11]:
```
%timeit -r 10 make_lm_matrix(64)
```

```
10 loops, best of 10: 30.6 ms per loop
```

In [26]:
```
res_new = %timeit -r 10 -o new_lm_matrix(64)
```

```
The slowest run took 18.17 times longer than the fastest. This could mean that an intermediate result is bei
ng cached.
10000 loops, best of 10: 90.7 µs per loop
```

In [13]:
```
%memit make_lm_matrix(64)
```

```
peak memory: 72.82 MiB, increment: 0.02 MiB
```

In [14]:
```
%memit new_lm_matrix(64)
```

```
peak memory: 72.82 MiB, increment: 0.00 MiB
```

In [25]:
```
res_new.repeat
```

Out[25]:
```
10
```

In [ ]:
```

```