In [99]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.manifold import Isomap, LocallyLinearEmbedding, SpectralEmbedding, TSN
E, MDS
from sklearn.decomposition import PCA

from time import time
from matplotlib.ticker import NullFormatter
```

In [100]:

```
import scipy.io
file = scipy.io.loadmat('face.mat')
file.keys()
```

Out[100]:

```
dict_keys(['__header__', '__version__', '__globals__', 'Y', 'id'])
```

In [101]:

```
Y = file['Y']
print(Y.shape)
```

```
(112, 92, 33)
```

In [102]:

```
## the "id" is the original order
orig_idx = file['id']
orig_idx=np.reshape(orig_idx,(33,))
print(orig_idx)
```

```
[29 21 15  2 32 16  6 31 17 27 33 18 22 24 28 13 12  8  1 23 26 10 25  7
  5 11 20  4  3 14 19  9 30]
```

In [103]:

```
# the indices of the sorted manual_idx (from small to large)
arg_orig_idx = np.argsort(orig_idx)
# from large to small
arg_orig_idx = arg_orig_idx[::-1]
print(arg_orig_idx)
```

```
[10  4  7 32  0 14  9 20 22 13 19 12  1 26 30 11  8  5  2 29 15 16 25 21
 31 17 23  6 24 27 28  3 18]
```

In [104]:

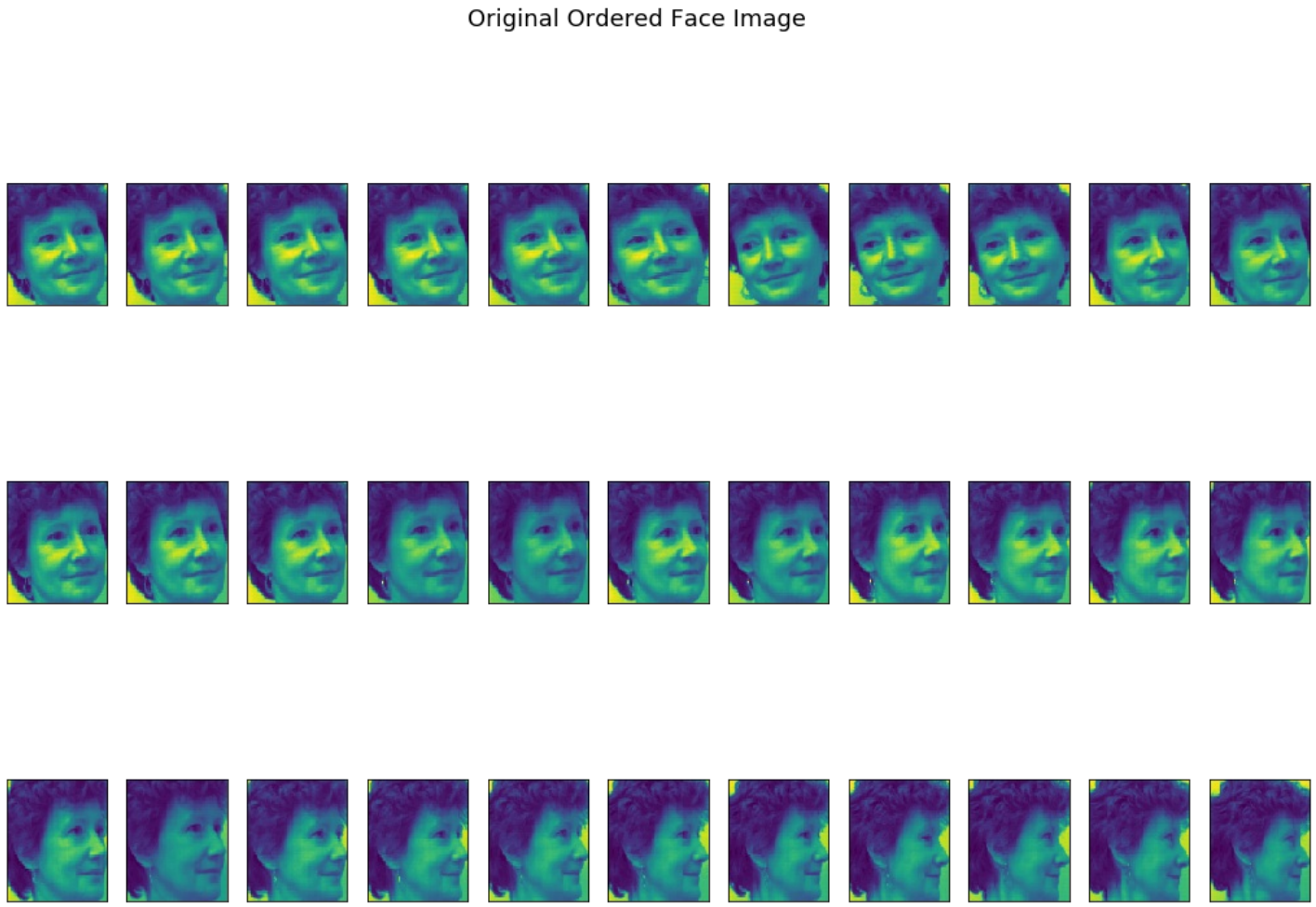
```
X=np.reshape(Y,[10304,33]).T
print(X.shape)
```

```
(33, 10304)
```

See the 33 faces in the original order

In [105]:

```
plt.figure(figsize=(18,12))
plt.suptitle("Original Ordered Face Image", fontsize=18)

for i in range (33):
    plt.subplot(3, 11, i+1)
    plt.imshow(Y[:,:,arg_orig_idx[i]])
    plt.xticks([])
    plt.yticks([])
plt.show()
```



We can see that indeed they are ordered as the woman is turning her head to the left. But the 7th, 8th, 9th ones seem not right.

MDS Embedding

In [106]:

```
n_neighbors = 5
n_components = 2
```

In [110]:

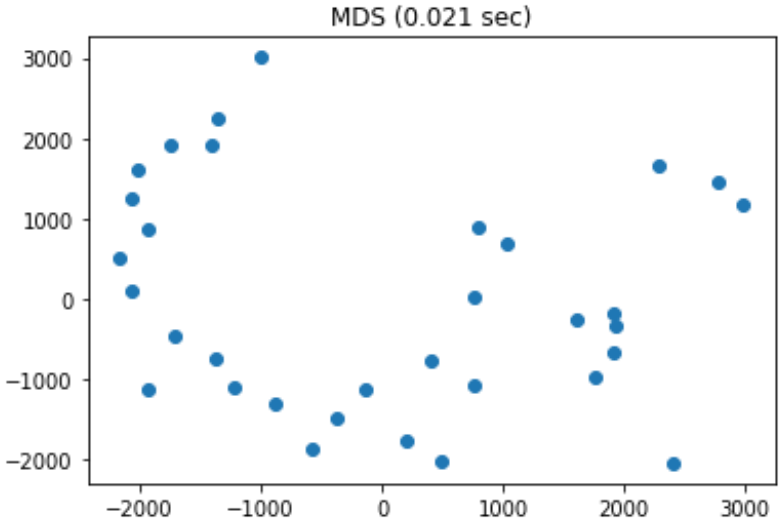
```
t0 = time()
mds = MDS(n_components, max_iter=2000, n_init=1)
face_mds = mds.fit_transform(X)
t1 = time()
print("MDS: %.2g sec" % (t1 - t0))

plt.scatter(face_mds[:, 0], face_mds[:, 1])
```

```
plt.title("MDS (%.2g sec)" % (t1 - t0))
```

```
MDS: 0.021 sec
```

Out[110]:

```
Text(0.5,1,'MDS (0.021 sec)')
```



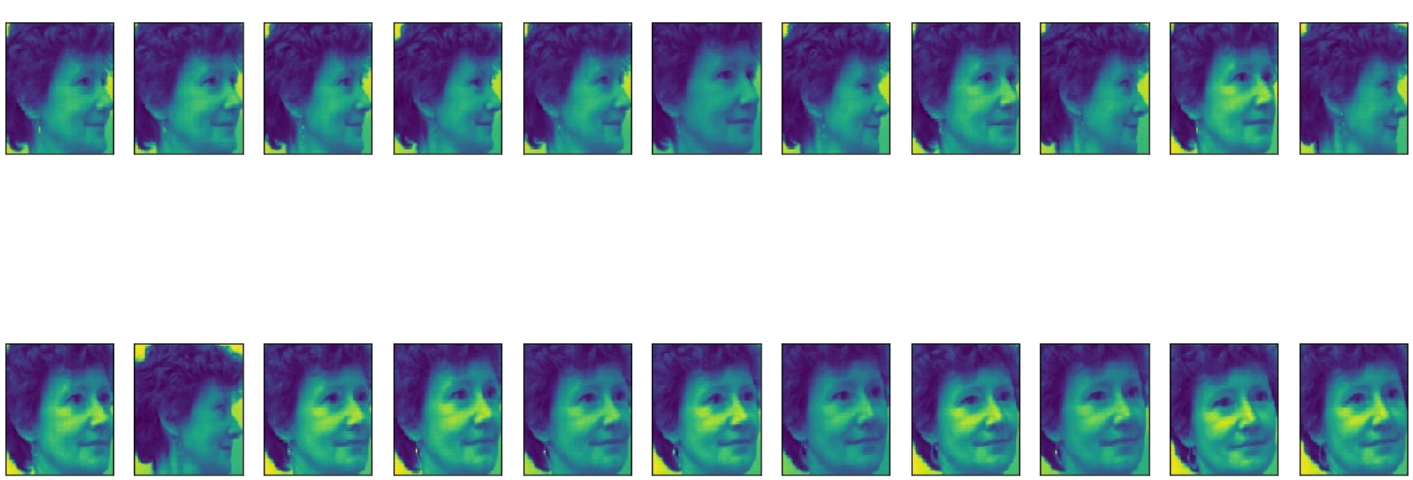
In [111]:

```
mds_idx = np.argsort(face_mds[:,0])
print(mds_idx)
```

```
[17 31  6 24 23 21 27 25 28 16  3 15 18 29  2  5  8 11 26 30 12  1 19 13
 32  0  4  7 10 22 14 20  9]
```

In [112]:

```
plt.figure(figsize=(18,12))
plt.suptitle("Ordered Face Image by MDS", fontsize=18)

for i in range (33):
    plt.subplot(3, 11, i+1)
    plt.imshow(Y[:,:,mds_idx[i]])
    plt.xticks([])
    plt.yticks([])
plt.show()
```



MDS can order most figures corrctly.

ISOMAP Embedding

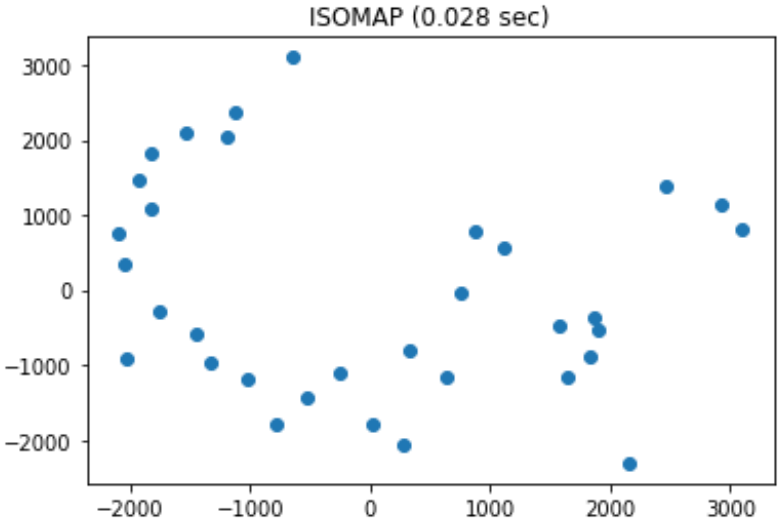
In [118]:

```
t0 = time()
isomap = Isomap(n_neighbors, n_components)
face_isomap = mds.fit_transform(X)
t1 = time()
print("ISOMAP: %.2g sec" % (t1 - t0))

plt.scatter(face_isomap[:, 0], face_isomap[:, 1])
plt.title("ISOMAP (%.2g sec)" % (t1 - t0))
```

```
ISOMAP: 0.028 sec
```

Out[118]:

```
Text(0.5,1,'ISOMAP (0.028 sec)')
```



In [119]:

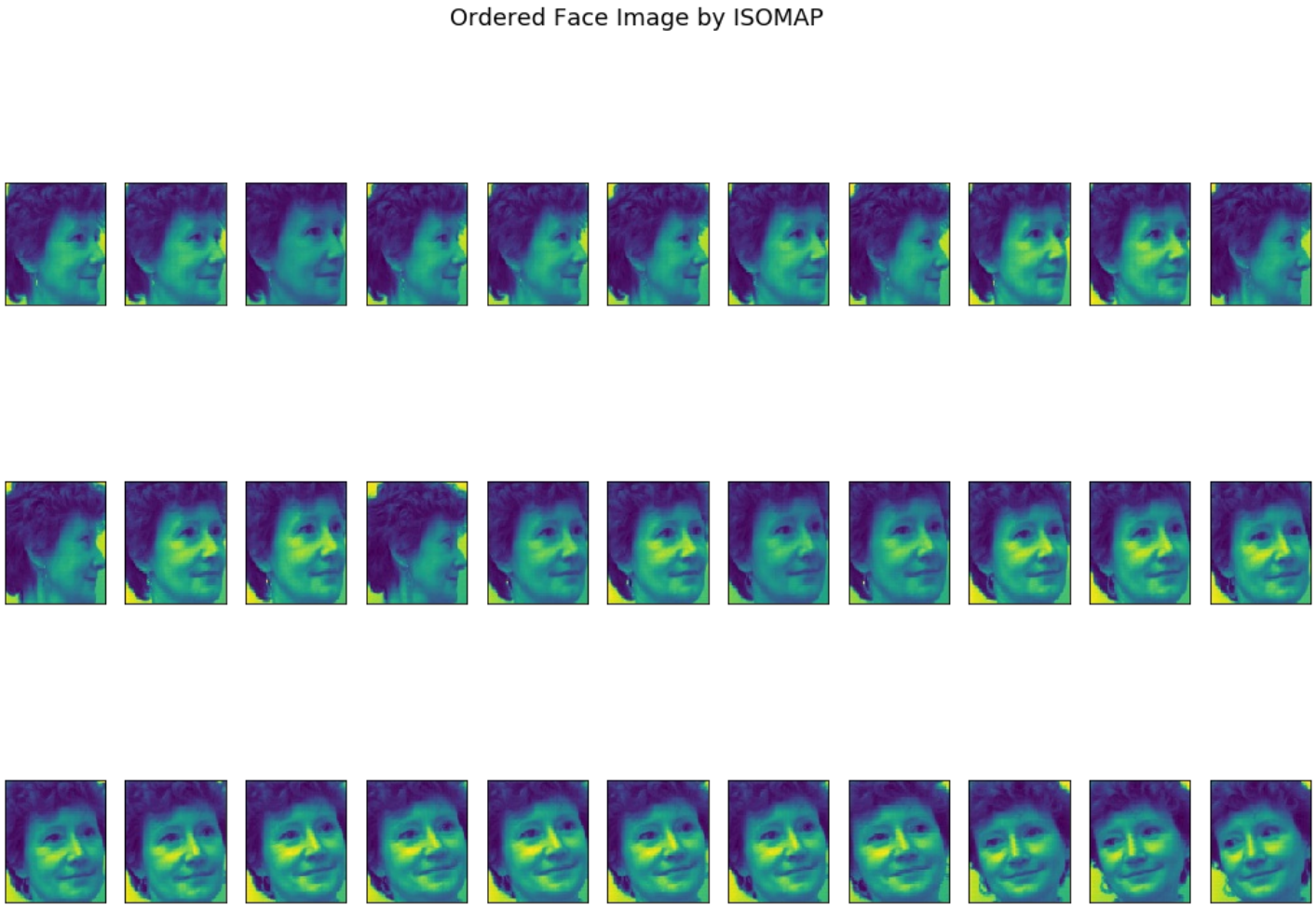
```
isomap_idx = np.argsort(face_isomap[:,0])
print(isomap_idx)
```

```
[17 31 21  6 23 24 25 27 16 15 28  3 29  2 18  5  8 11 30 26  1 12 19 13
 32  0  7 10  4 14 22 20  9]
```

In [120]:

```
plt.figure(figsize=(18,12))
plt.suptitle("Ordered Face Image by ISOMAP", fontsize=18)

for i in range (33):
    plt.subplot(3, 11, i+1)
    plt.imshow(Y[:,:,isomap_idx[i]])
```

```
    plt.xticks([])
    plt.yticks([])
plt.show()
```



In the second line, figure 1 and 4 are obviously ordered wrong. The third line seems good. It is almost as good as MDS.

LLE Embedding

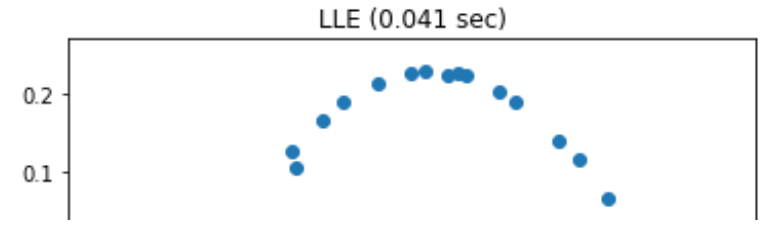
In [121]:

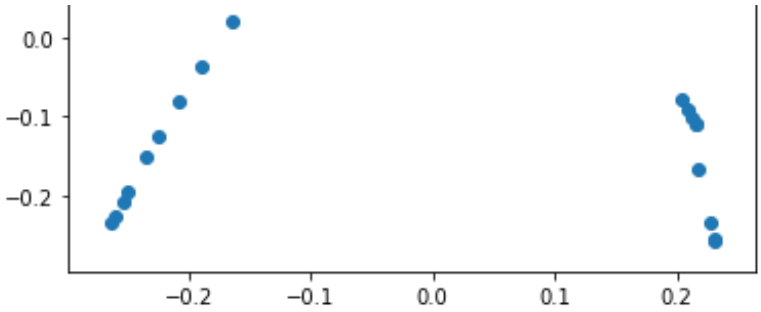
```
t0 = time()
lle = LocallyLinearEmbedding(n_neighbors, n_components, method='standard', eigen_so
lver='auto')
face_lle = lle.fit_transform(X)
t1 = time()
print("LLE: %.2g sec" % (t1 - t0))

plt.scatter(face_lle[:, 0], face_lle[:, 1])
plt.title("LLE (%.2g sec)" % (t1 - t0))
```

```
LLE: 0.041 sec
```

Out[121]:

```
Text(0.5,1,'LLE (0.041 sec)')
```





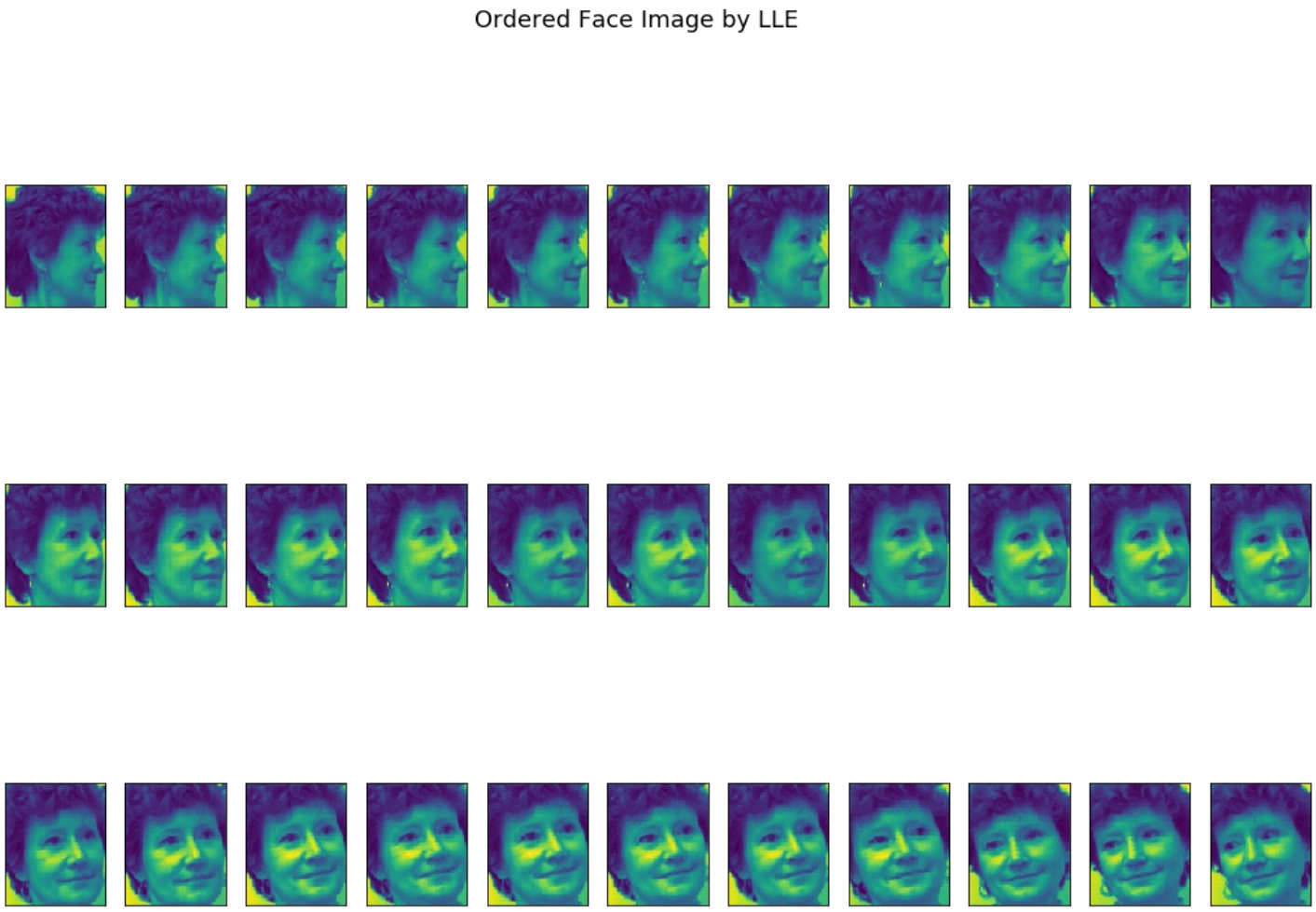
In [122]:

```
lle_idx = np.argsort(face_lle[:,0])
print(lle_idx)
```

```
[18  3 28 27 24  6 23 17 31 25 21 16 15 29  2  5  8 11 30 26  1 12 19 13
 32  0  7 10  4 14 22 20  9]
```

In [123]:

```
plt.figure(figsize=(18,12))
plt.suptitle("Ordered Face Image by LLE", fontsize=18)

for i in range (33):
    plt.subplot(3, 11, i+1)
    plt.imshow(Y[:,:,lle_idx[i]])
    plt.xticks([])
    plt.yticks([])
plt.show()
```



We can see that LLE is very good, even better than the original order since there is no obvious error.

In [ ]: