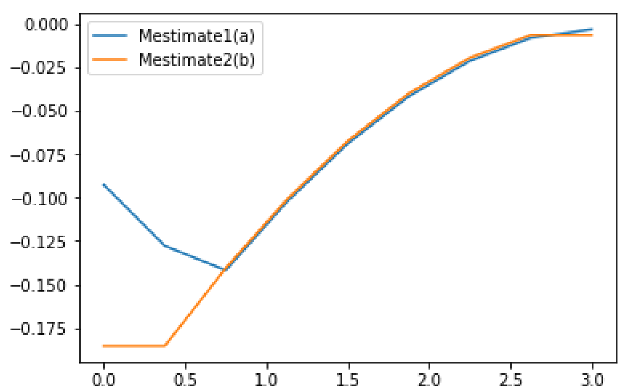
```
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/hoops/OneDrive/Documents/School/ME EN 2450 Numerical Methods/
HW7/HW7.py', wdir='C:/Users/hoops/OneDrive/Documents/School/ME EN 2450 Numerical Methods/
HW7')
Exercise 1:
Using Richardson Extrapolation with step sizes of pi/3 and pi/6 for the derivative of
cos(x) at x = pi/4 the estimate is -0.70539 with a true relative error of 0.24249%


Exercise 2:
       x       y  theta approx  M estimate 1  M estimate 2
0  0.000  0.0000     -0.685600     -0.092629     -0.185259
1  0.375 -0.2571     -1.264533     -0.127744     -0.185259
2  0.750 -0.9484     -2.282400     -0.141803     -0.140459
3  1.125 -1.9689     -3.037067     -0.102475     -0.101035
4  1.500 -3.2262     -3.563333     -0.068939     -0.067371
5  1.875 -4.6414     -3.898800     -0.041728     -0.039979
6  2.250 -6.1503     -4.084933     -0.021397     -0.019584
7  2.625 -7.7051     -4.166267     -0.008117     -0.006443
8  3.000 -9.2750     -4.186400     -0.003221     -0.006443
```




```
It appears that the second estimate using the finite difference approximation is more
accurate since it seems more reasonable that the bending moment is 0 at x = 0 it is also
likely more accurate because the first estimate accrues 2 * O(h^2) error while the second
only accrues O(h^2) error


Exercise 3:
Checking ex22.3.  Two point gauss quadrature:1.8225777777777772  Actual: 1.640533
Checking ex22.4.  Three point gauss quadrature:1.6405333333333294  Actual: 1.640533


Exercise 4:
13.11
a) Using Newton's method the estimated minimum is 1.0689693688871522 at x =
-0.5866826961920031
b) Using Newton's method with finite differences the estimated minimum is
1.0689693703449121 at x = -0.5866688904649386
```

```
In [2]:
```