The purpose of this assignment is to show that one can obtain the Legendre Polynomials numerically from the QR decompostion

The Legendre polynomials are orthogonal on [-1,1] and have many applications. Please see http://en.wikipedia.org/wiki/Legendre_polynomials (http://en.wikipedia.org/wiki/Legendre_polynomials)

In [1]:

```
# Let x be a vector of equally spaced points from -1 to 1
# Let the columns of A be the powers of x
# One can think of A as having first column 1, second column x,
 third column x^2, etc.
N = 1000
h = 1/N
k = 5
A = zeros(2*N+1,k+1)
x = -1: 1/N : 1
for i=1:2N+1,j=1:k
        A[i,j] = x[i]^(j-1)
end
```

In [5]:

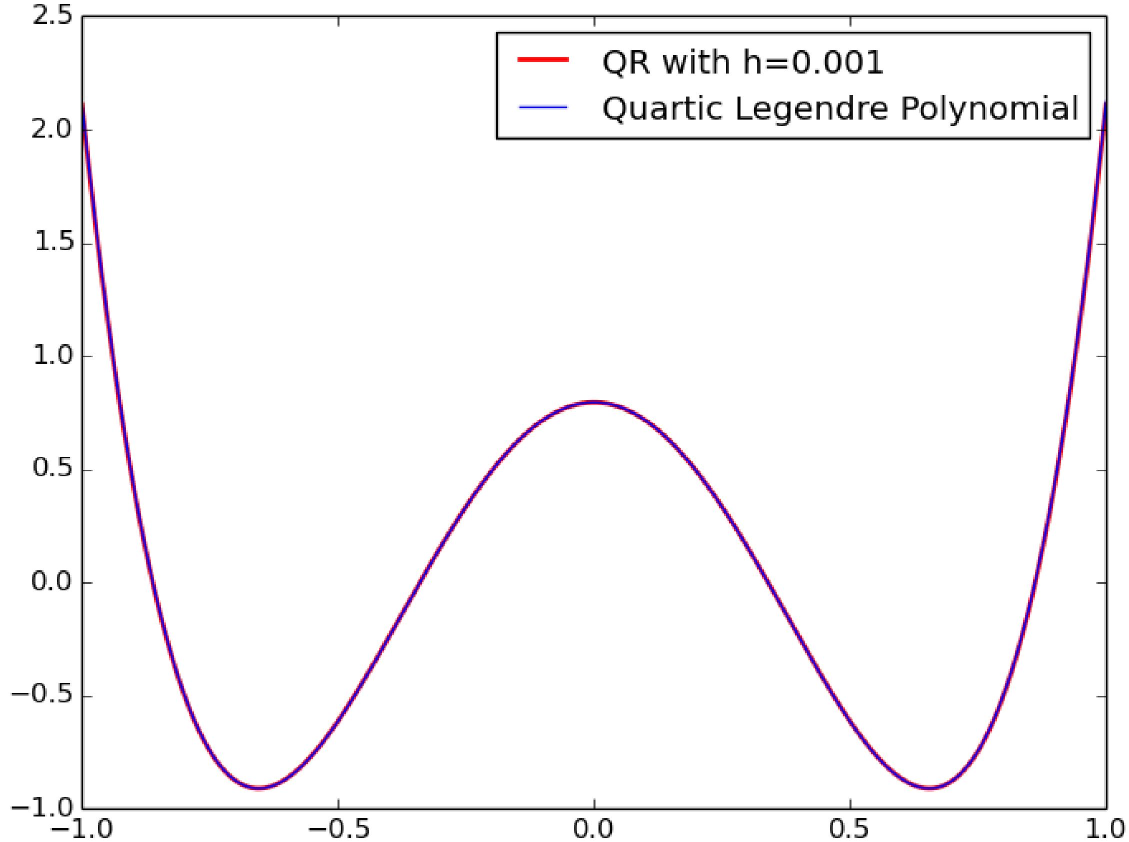
```
# Compute the QR decomposition of A (Gram-Schmidt)
(q,r) = qr(A)
q=q/sqrt(h); # Normalize q to integrate to 1;
```

You are asked to compare graphically with the first five columns of Q with the Legendre polynomials to see if they line up Note: You must compare with the nth Legendre polynomial times sqrt(n+.5). Do you see why? Note that it is possible you may be off by a factor of -1. Can you explain why?

In [23]:

```
using PyPlot;
```

In [22]:
```
plot(x, -q[:,5],        linewidth=2,"r")
plot(x, (35*x.^4 - 30*x.^2 + 3)/8 * sqrt(4.5))
legend({"QR with h=$h","Quartic Legendre Polynomial"})
```



Out[22]: PyObject <matplotlib.legend.Legend object at 0x7fe61c0d1c10>

Please submit pictures up to the quartic like the one above and answer the two questions.

In [ ]: