1. 計算內積:

$< f,\ g >:= \int_{-\pi}^{\pi} f(x)g(x)dx$

$where\ f(x) = sin(x),\ g(x) = cos(x)$

In [1]:

```
import numpy as np
```

In [2]:

```
import scipy.integrate as integrate
import scipy.special as special
```

In [3]:

```
result = integrate.quad(lambda x: np.sin(x)*np.cos(x), -np.pi, np.pi)
```

In [4]:

```
result # the first term is the result, the second term is the error
```

Out[4]:

```
(0.0, 2.2102770810800183e-14)
```

2. Gram-Schmidt procedure:

GS 參考資料 (http://moocs.nccu.edu.tw/media/19162#tab-step)
QR decomposition 參考資料 (http://moocs.nccu.edu.tw/media/19163)

In [9]:

```
import numpy.random as rn
import numpy.linalg as la
```

In [26]:

```
def GS(A, tol=10**-15):
    m, n = A.shape
    Idx = list() # store the
    Q = np.mat(np.zeros((m, n))) # store the result matrix
    Q[:,0] = A[:,0] / la.norm(A[:,0]) # the first vector of the original basis
    for i in range(1, n): # start from the 2nd column till nth column
        Q[:,i] = A[:,i].copy()
        for j in range(i): # inner product
            Q[:,i] = Q[:,i] - Q[:,j] * float(Q[:,j].T * Q[:,i])
        if la.norm(Q[:,i]) > tol: # normalize
            Q[:,i] = Q[:,i] / la.norm(Q[:,i])
        else:
            Idx.append(i)
    return Q, Idx
```

In [35]:

```
x = rn.randn(3)
A = np.mat([x, x**2, x**3, x**4, x**5])
print(A)
print(A.shape)
```

```
[[-1.20608003  0.89880674  2.09276926]
 [ 1.45462904  0.80785356  4.37968318]
 [-1.75439904  0.72610422  9.16566633]
 [ 2.11594564  0.65262737 19.18162475]
 [-2.55199979  0.58658588 40.14271465]]
(5, 3)
```

In [36]:

```
q, idx = GS(A)
q
```

Out[36]:

```
matrix([[-0.28717478,  0.49644677, -0.48315358],
        [ 0.34635576,  0.56078365, -0.12655933],
        [-0.41773277,  0.36585077, -0.23534973],
        [ 0.50381915,  0.49549174,  0.45730232],
        [-0.60764622,  0.24434288,  0.69715884]])
```

3. $P_U(sin(x))$

$U = span(1, x, x^2, x^3, x^4, x^5)$

In [85]:

```
x = rn.randn(5)
B = np.mat([x, x**2, x**3, x**4, x**5])
```

In [86]:

```
q2, idx2 = GS(B)
```

In [87]:

```
q2.shape
```

Out[87]:

$(5,\ 5)$

In [89]:

```
res = np.mat(np.zeros((5, 1)))
for i in range(q2.shape[1]): # column of q2
    sin = np.sin(x).reshape(1, 5)
    res = q2[:,i]*(sin * q2[:,i])

p_u_sinx = res
p_u_sinx
```

Out[89]:

```
matrix([[-0.02099247],
        [-0.16076548],
        [ 0.05517341],
        [ 0.21959042],
        [-0.11417569]])
```