Problem 1

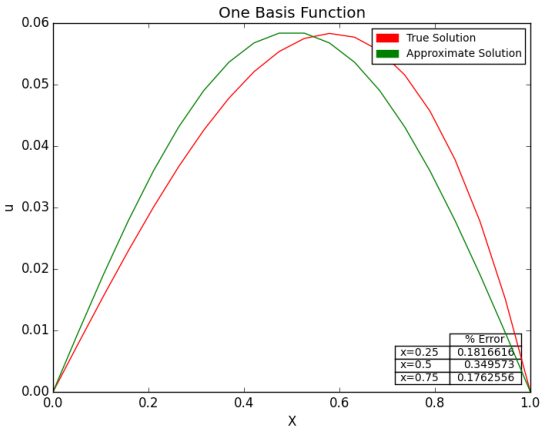
A)

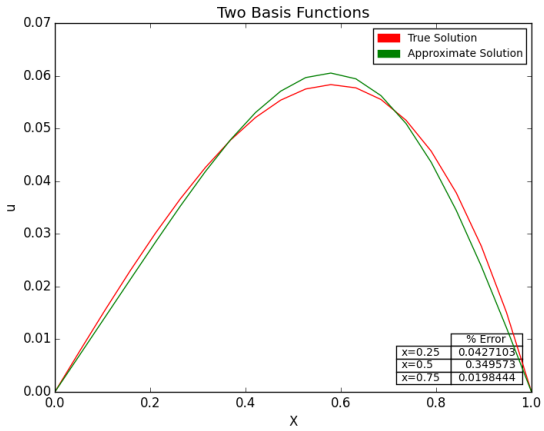
$$u(0) = 0 - \frac{sinh(0)}{sinh(1)} = 0$$
$$u(1) = 1 - \frac{sinh(1)}{sinh(1)} = 0$$
$$u'(x) = 1 - \frac{cosh(x)}{sinh(1)}$$
$$u''(x) = -\frac{sinh(x)}{sinh(1)}$$
$$\therefore -u''(x) + u = \frac{sinh(x)}{sinh(1)} + x - \frac{sinh(x)}{sinh(1)} = x$$

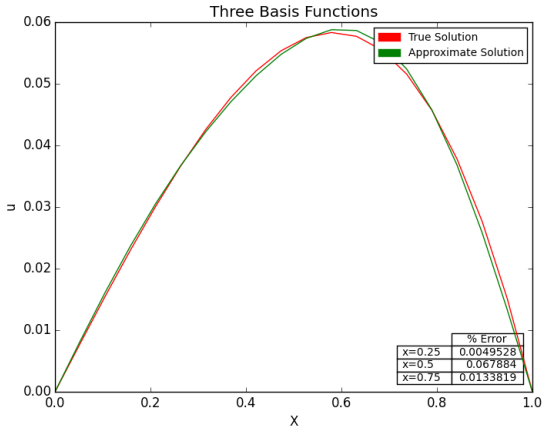
B)

$$\left(-\frac{\partial^2 u}{\partial x^2} + u \right) v = xv \text{ for } v \in \mathcal{H}_0^1$$
$$\int_0^1 -\frac{\partial^2 u}{\partial x^2} v + uv \, dx = \int_0^1 xv \, dx$$
$$\int_0^1 \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} \, dx + \int_0^1 uv \, dx = \int_0^1 xv \, dx$$

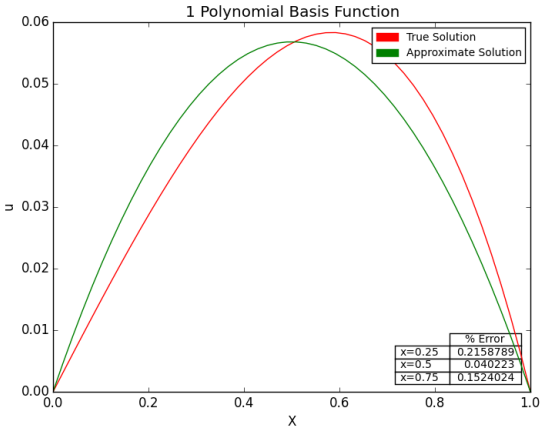
C) The code for these next two sections is very long. It is attached as an appendix.

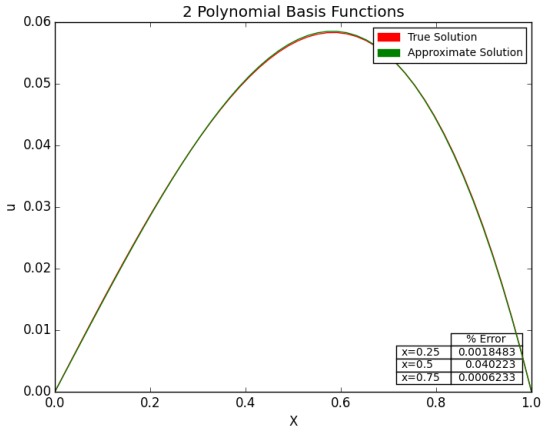


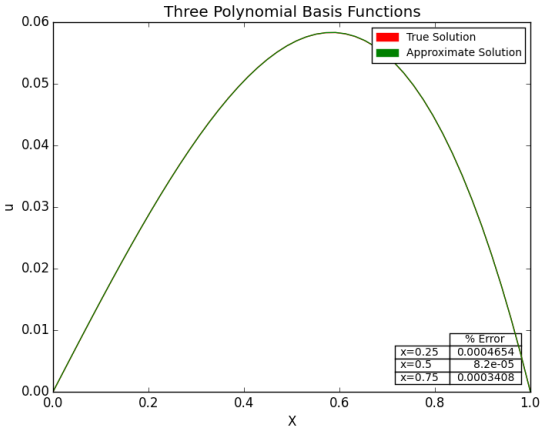




D)







Problem 2

A) B) C) ϕ_1 does not satisfy the boundary conditions.

Problem 1 Part C Code.

What can Python do? It can do anything.

```
from scipy import sin, cos, sinh
import scipy.integrate as DI
from numpy import pi, linspace
import matplotlib.pyplot as plt

phi1   = lambda x: sin(pi*x)
dphi1  = lambda x: pi*cos(pi*x)
phi2   = lambda x: sin(2*pi*x)
dphi2  = lambda x : 2*pi*cos(2*pi*x)
phi3   = lambda x: sin(3*pi*x)
dphi3  = lambda x: 3*pi*cos(3*pi*x)

phi1sq   = lambda x: phi1(x)**2
dphi1sq  = lambda x: dphi1(x)**2
phi2sq   = lambda x: phi2(x)**2
dphi2sq  = lambda x: dphi2(x)**2
phi3sq   = lambda x: phi3(x)**2
```

```
dphi3sq = lambda x: dphi3(x)**2

phi12  = lambda x: phi1(x) * phi2(x)
phi13  = lambda x: phi1(x) * phi3(x)
phi23  = lambda x: phi2(x) * phi3(x)
dphi12 = lambda x: dphi1(x) * dphi2(x)
dphi13 = lambda x: dphi1(x) * dphi3(x)
dphi23 = lambda x: dphi2(x) * dphi3(x)

f  = lambda x: x
f1 = lambda x: f(x) * phi1(x)
f2 = lambda x: f(x) * phi2(x)
f3 = lambda x: f(x) * phi3(x)

X = linspace(0, 1, 20)

u = lambda x: x - sinh(x)/sinh(1)

# N = 1
alpha = DI.quad(f1, 0, 1)[0] /(DI.quad(phi1sq, 0, 1)[0] + DI.quad(dphi1sq, 0, 1)[0])
phi = lambda x:  alpha*phi1(x)

plt.plot(X, u(X),'r--', X, phi(X),'g')
plt.show()

# N = 2
```

Problem 1 Part D Codes.

This is essentially the same as he part c code, except I made some small changes to handle the polynomials.