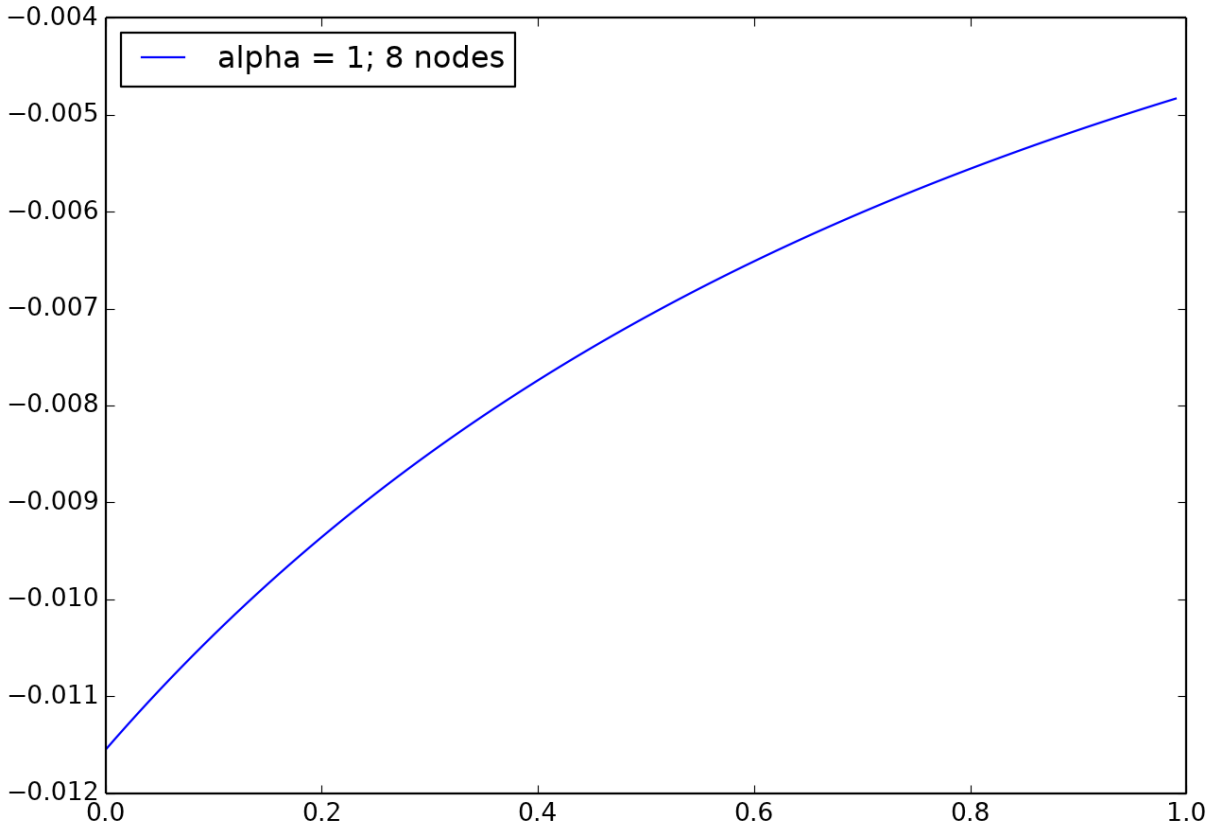
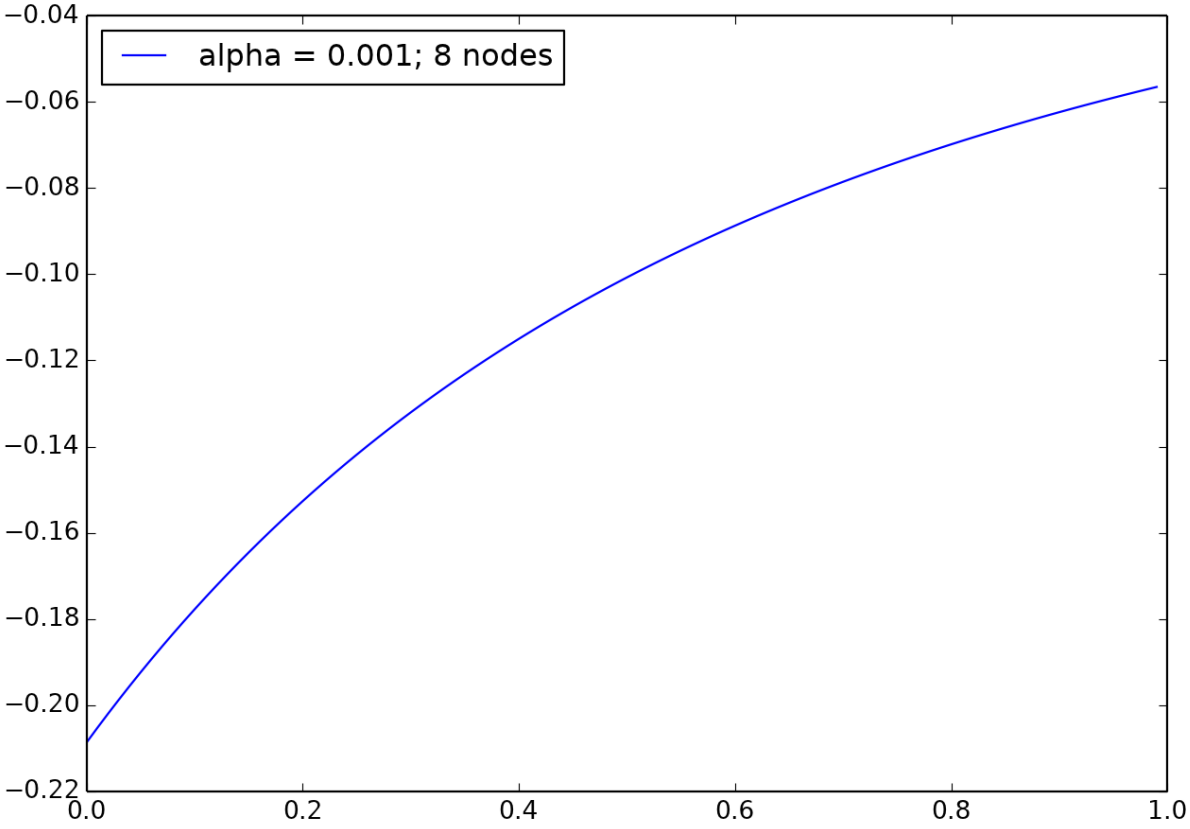
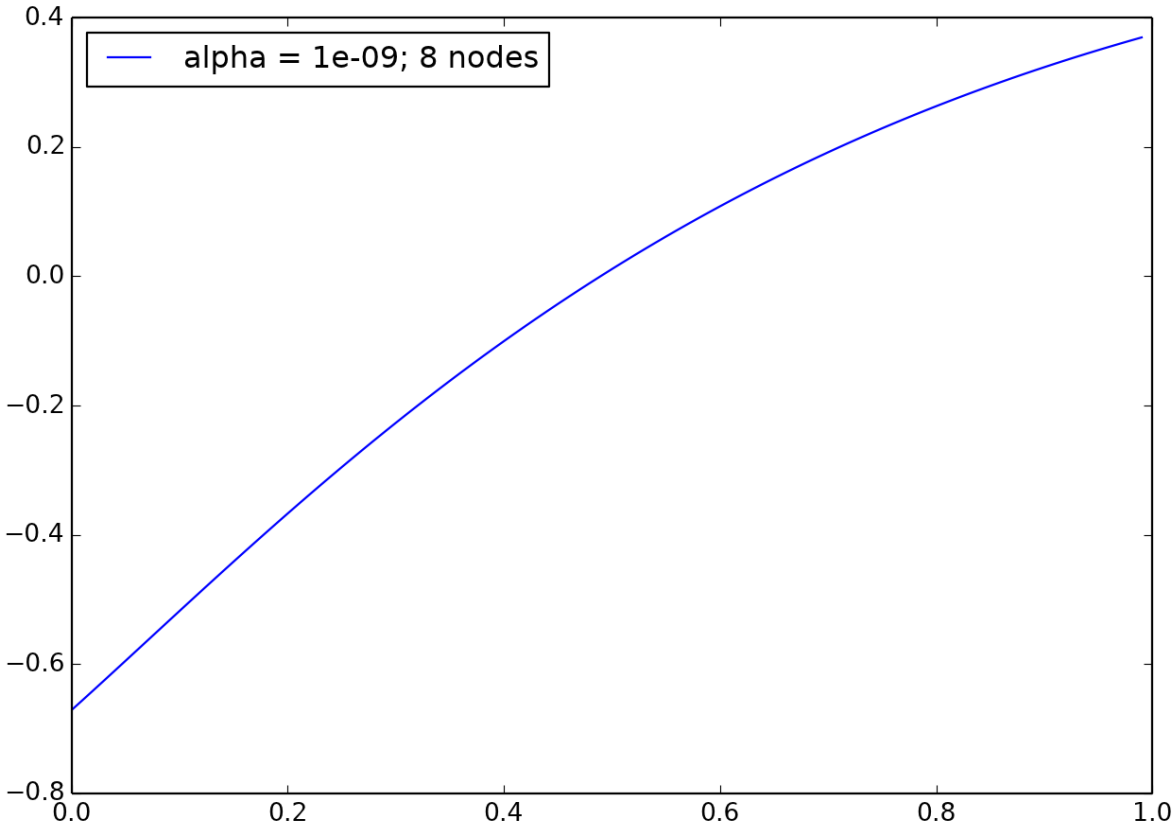
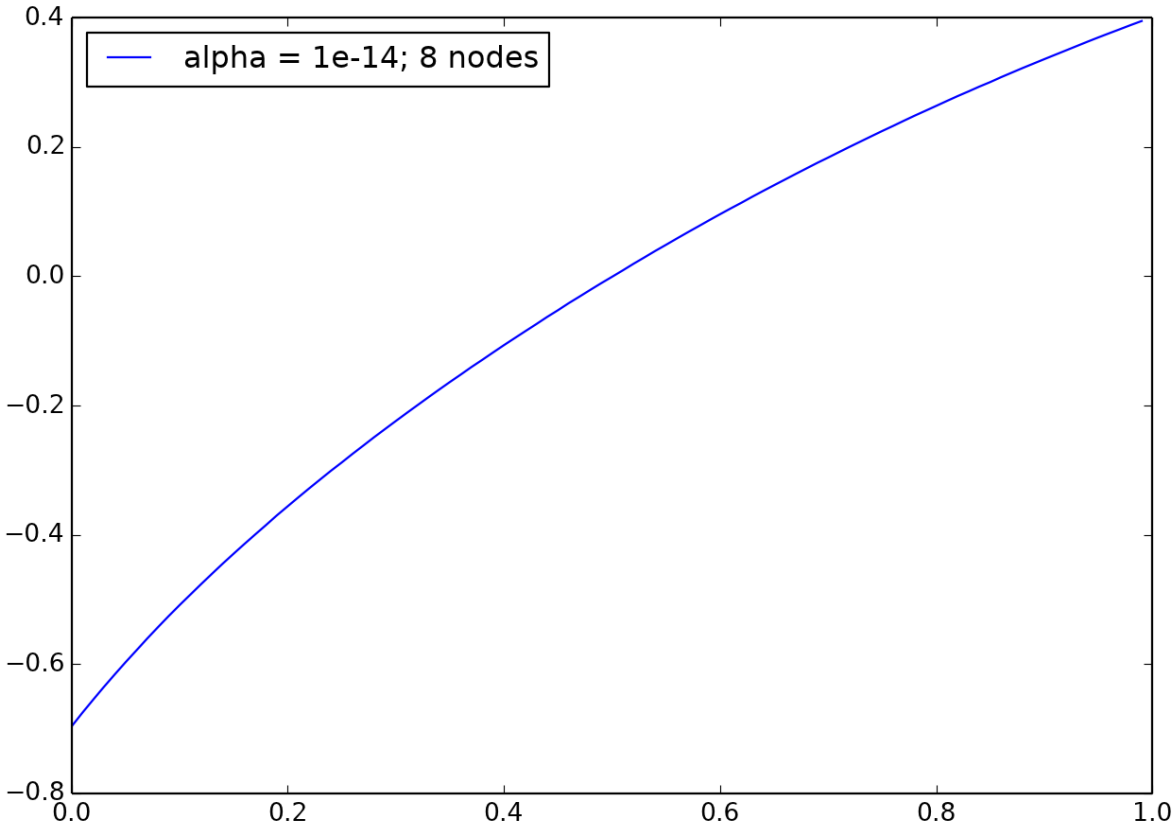
```python
# coding: utf-8

from math import log
import Slozhna as sl
from numpy import empty
from numpy.linalg import solve
import matplotlib.pyplot as plt

def K(x,t):
    return 1 / (x + t + 1.5)**2

def f(x):
    return log(3*(x + 1.5)/(x + 2.5)) / (1.0 + x) - log(2) / (x + 1.5) - log(1.5) / (x + 2.5)

alpha = .00000000000001
nodes = 8
leg_roots = sl.Leg_roots(nodes)
weights = sl.Gauss_weights(nodes)

leg_roots = [(item + 1)/2.0 for item in leg_roots]
weights = [item/2.0 for item in weights]

def K_K(x,t):
    ar = [K(root,x) * K(root,t) for root in leg_roots]
    return sl.Gauss_integr_ar(nodes,ar)

def fun(x):
    ar = [K(root, x) * f(root) for root in leg_roots]
    return sl.Gauss_integr_ar(nodes,ar)

prav = [fun(root) for root in leg_roots]

A = empty([nodes,nodes])

for i in range(nodes):
    for k in range(nodes):
        A[i][k] = weights[k] * K_K(leg_roots[i], leg_roots[k])
    A[i][i] += alpha

u = solve(A, prav)

def tmp(x):
    return sum([weights[i] * K_K(x, leg_roots[i]) * u[i] for i in range(nodes)])

resh = [(fun(i/100.0) - tmp(i/100.0)) / alpha for i in range(100)]

plt.figure(1)
#plt.plot(leg_roots, u, label='alpha = '+str(alpha)+'; '+str(nodes)+' nodes')
plt.plot([i/100.0 for i in range(100)], resh, label='alpha = '+str(alpha)+'; '+str(nodes)+' nodes')
legend = plt.legend(loc='upper left')
plt.show()
```


alpha = 1; 8 nodes
−0.004
−0.005
−0.006
−0.007
−0.008
−0.009
−0.010
−0.011
−0.012
0.0
0.2
0.4
0.6
0.8
1.0


alpha = 0.001; 8 nodes
−0.04
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
−0.18
−0.20
−0.22
0.0
0.2
0.4
0.6
0.8
1.0


alpha = 1e-09; 8 nodes
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.0
0.2
0.4
0.6
0.8
1.0


alpha = 1e-14; 8 nodes
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.0
0.2
0.4
0.6
0.8
1.0