stripload_comparison

Unknown Author

February 13, 2014

In [1]:

```
%pylab inline
from scipy.integrate import trapz,quad,romberg,dblquad
from math import *
```

```
Populating the interactive namespace from numpy and matplotlib
```

1 Original boussinesq with ν parameter

Equation 11.20 from Bowles, σ_x component

In [2]:

```
def b(Q, x, y, z, nu):
    R = sqrt(x**2+y**2+z**2)
    r = sqrt(x**2+y**2)
    return Q/(2.0*pi)*(3.0*r**2*z/R**5-(1.0-2.0*nu)/(R*(R+z)))*(x/r)

#integrate from x1 to x2 and from -inf to inf to from the strip load
def boussinesq_stripload(q,x1,x2,H,z,nu):
    return dblquad(lambda y, x: b(q, x, y, z, nu), x1, x2, \
                   lambda x: float("-inf"), lambda x: float("inf"))[0]
```

2 NAVFAC DM 7.01 strip load

In [3]:

```
def navfac_stripload(q, x1, x2, z):
    gamma = atan(x1/z)
    alpha = atan(x2/z) - gamma
    return (q/pi)*(alpha-sin(alpha)*cos(alpha+2*gamma))
```

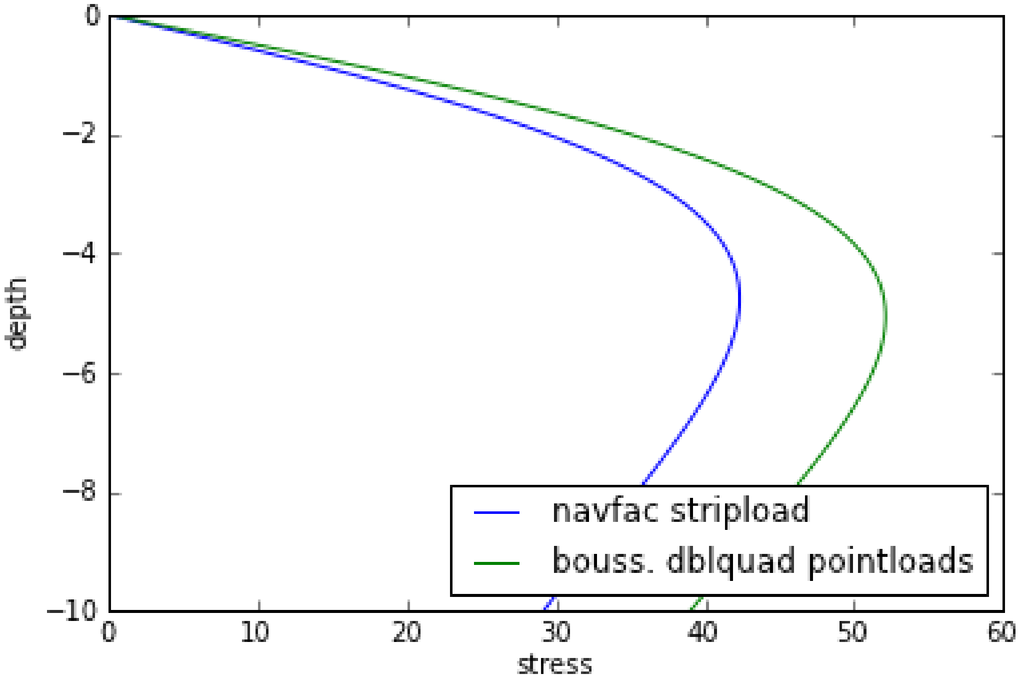
3 Comparison Boussinesq $\nu = 0.5$ with NAVFAC

In [4]:

```
q = 200.0  # 200 psf strip load
x1 = 5.0 # starting at 5' from wall
x2 = x1 + 10.0 # ending at 15' from wall
H = 10.0

nu = 0.5

z = np.arange(0.0,H+H/100.0, H/100.0)
sigma1 = [navfac_stripload(q,x1,x2,i) for i in z]
sigma2 = [boussinesq_stripload(q,x1,x2,H,i,nu) for i in z]

xlabel(u"stress")
ylabel("depth")
```

```
plot(sigma1,-z,label="navfac stripload")
plot(sigma2,-z,label="bouss. dblquad pointloads")


legend(loc=4)
print "total Navfac = %.3f plf" % (trapz(sigma1,z))
print "total Boussinesq from pointloads = %.3f plf" % (trapz(sigma2,z))
```

```
total Navfac = 330.482 plf
total Boussinesq from pointloads = 411.920 plf
```



4 Check Boussinesq sum of point loads vs. sum of line loads ($\nu = 0.5$)

In [5]:

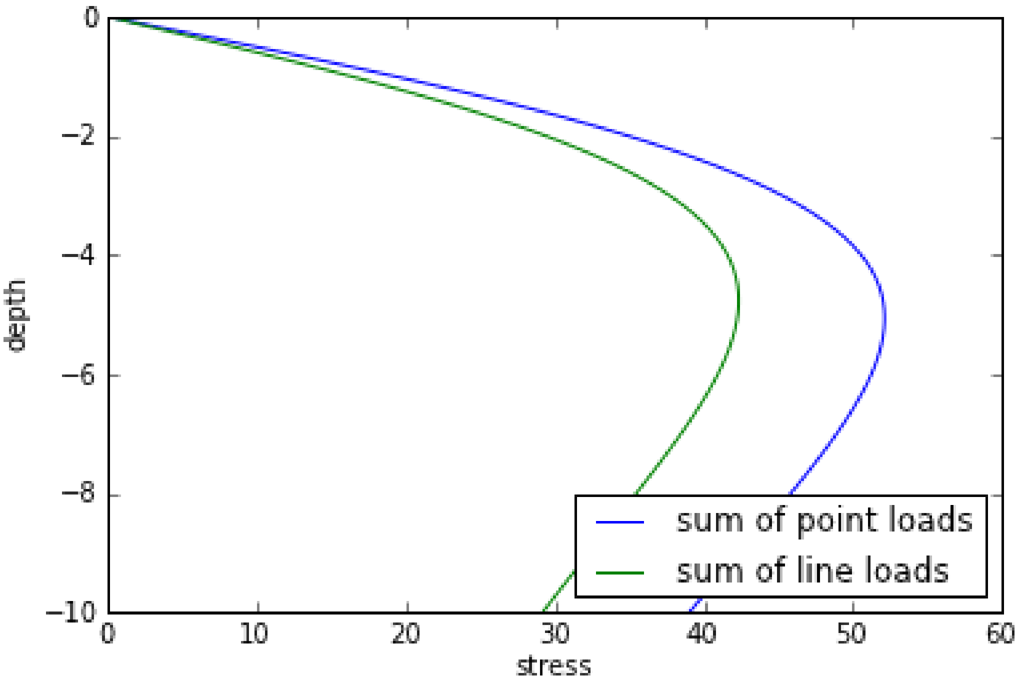
```
def boussinesq_lineload(p,x,z):
    return (2*p/pi)*x**2*z/(x**2+z**2)**2

def boussinesq_stripload2(q,x1,x2,H,z):
    return quad(lambda x: boussinesq_lineload(q,x,z), x1, x2)[0]
```

In [6]:

```
q = 200.0  # 200 psf strip load
x1 = 5.0 # starting at 5’ from wall
x2 = x1 + 10.0 # ending at 15’ from wall
H = 10.0

nu = 0.5

z = np.arange(0.0,H+H/100.0, H/100.0)
sigma1 = [boussinesq_stripload(q,x1,x2,H,i,nu) for i in z]
sigma2 = [boussinesq_stripload2(q,x1,x2,H,i) for i in z]

xlabel(u"stress")
ylabel("depth")

plot(sigma1,-z,label="sum of point loads")
plot(sigma2,-z,label="sum of line loads")
```

```
legend(loc=4)
print "total (sum of point loads) = %.3f plf" % (trapz(sigma1,z))
print "total (sum of line  loads) = %.3f plf" % (trapz(sigma2,z))
```

```
total (sum of point loads) = 411.920 plf
total (sum of line loads) = 330.482 plf
```



The double integration of a pointload or the simple integration of a line load does not give the same result !

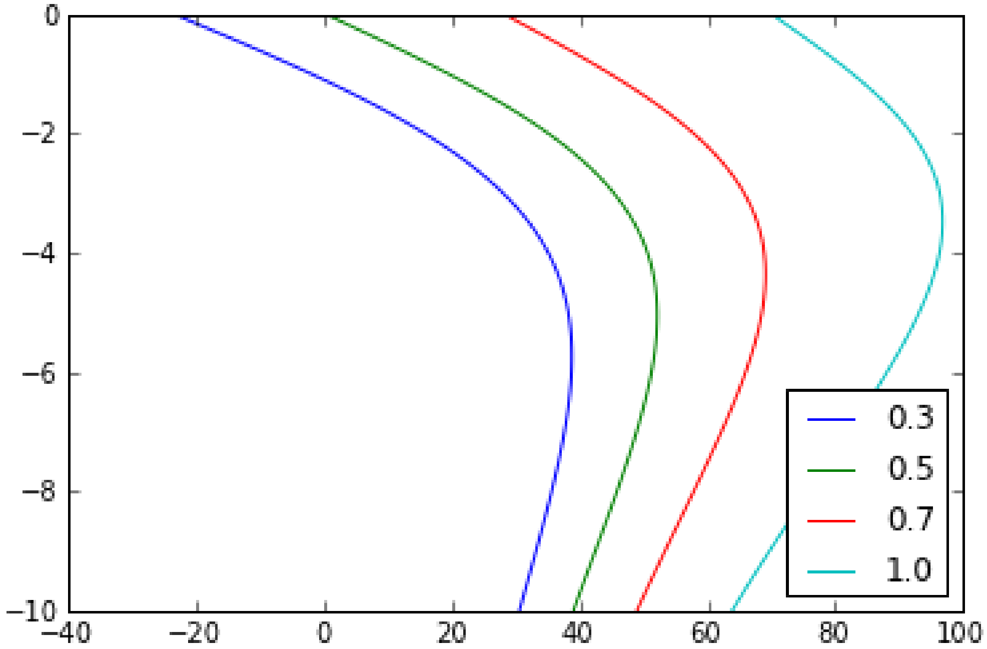
5 Boussinesq for varying ν

In [7]:

```
nu = [0.33, 0.5, 0.7, 1.0]

q = 200.0   # 200 psf strip load
x1 = 5.0 # starting at 5' from wall
x2 = x1 + 10.0 # ending at 15' from wall
H = 10.0

z = np.arange(0.0,H+H/100.0, H/100.0)
sigma=[]
for j in nu:
    sigma.append([boussinesq_stripload(q,x1,x2,H,i,j) for i in z])

for i in range(len(sigma)):
    plot(sigma[i],-z,label="%.1f" % (nu[i]))
    print "nu = %.1f -> total = %.3f" % (nu[i], trapz(sigma[i], z))
legend(loc=4)
```

```
nu = 0.3 -> total = 264.735
nu = 0.5 -> total = 411.920
nu = 0.7 -> total = 585.078
nu = 1.0 -> total = 844.815
<matplotlib.legend.Legend at 0x4064790>
```

```
Out [7]:
```



Note: negative lateral stress is not possible