Solving a singular integral equation using Chebyshev Polynomial Approximation

Harshit Garg - 2013A4PS332G

Abstract

Broberg [Broberg, 1999] derived an equation for the variation of elastostatic crack aperture with the prescribed stress on the crack faces. In this report, we attempt to solve that equation to obtain the relation between crack aperture $v(x)$ and distance x.

1. Introduction

The equation which we attempt to solve is

$$\frac{\partial v(x)}{\partial x} = \frac{-1}{2(1-k^2)G\sqrt{(x-b)(c-x)}} \left\{ \frac{1}{\pi} \int_b^c \frac{\sigma_y^0(\xi)\sqrt{(\xi-b)(c-\xi)}}{\xi - x}\, d\xi \;+\; \sigma_{yy}^{\infty}[x - \frac{b+c}{2}] \right\}$$

Here, $v(x)$ is opening of the crack.

2. Numerical Solution using Chebyshev Approximation

$$\frac{\partial v(x)}{\partial x} = \frac{-1}{2(1-k^2)G\sqrt{(x-b)(c-x)}} \left\{ \frac{1}{\pi} \int_b^c \frac{\sigma_y^0(\xi)\sqrt{(\xi-b)(c-\xi)}}{\xi - x}\, d\xi \;+\; \sigma_{yy}^{\infty}[x - \frac{b+c}{2}] \right\} \tag{1}$$

as crack extends from $-a$ to a, replacing $b = -a$ & $c = a$ in eq. (1) and integrating from $-a$ *to* x. The singular integral equation which we will get is,

$$v(x) = \frac{-1}{2(1-k^2)G} \left\{ \frac{1}{\pi} \int_{-a}^{x} \frac{1}{\sqrt{a^2 - x^2}} \left[\int_{-a}^{a} \frac{\sigma_y^0(\xi)\sqrt{a^2 - \xi^2}}{\xi - x} d\xi \right] dx \;+\; \int_{-a}^{x} \frac{\sigma_{yy}^{\infty}\, x}{\sqrt{a^2 - x^2}}\, dx \right\}$$

Taking linear relation between stress and crack aperture, i.e. $\sigma_y^0(\xi) = 2\alpha v(\xi)$, we get,

$$v(x) = \frac{-1}{2(1-k^2)G} \left\{ \frac{2\alpha}{\pi} \int_{-a}^{x} \frac{1}{\sqrt{a^2 - x^2}} \left[\int_{-a}^{a} \frac{v(\xi)\sqrt{a^2 - \xi^2}}{\xi - x} d\xi \right] dx \;+\; \int_{-a}^{x} \frac{\sigma_{yy}^{\infty}\, x}{\sqrt{a^2 - x^2}}\, dx \right\} \tag{2}$$

There is a Cauchy singularity in the first term of RHS of eq. (2) which have to be removed,

$$\int_{-a}^{a} \frac{v(\xi)\sqrt{a^2-\xi^2}}{\xi - x}\, d\xi \tag{3}$$

To remove this singularity we will use analytical relations defined for Chebyshev Polynomials. For that we will use Chebyshev Approximation.
We will approximate $v(x)$ as linear combination of N Chebyshev Polynomials with degrees 0 to N,

$$v(x) \approx \sum_{n=0}^{N} b_n U_n(x) \tag{4}$$

where $U_k(x)$ is Chebyshev Polynomial of second kind of degree k. And b_k's will be in $[-1, 1]$.
Substituting eq. (4) in eq. (2), we will get,

$$\sum_{n=0}^{N} b_n U_n(x) \approx \frac{-1}{2(1-k^2)G} \left\{ \frac{2\alpha}{\pi} \int_{-a}^{x} \frac{1}{\sqrt{a^2-x^2}} \left[\int_{-a}^{a} \frac{\sum_{n=0}^{N} b_n U_n(\xi)\sqrt{a^2-\xi^2}}{\xi - x}\, d\xi \right] dx + \int_{-a}^{x} \frac{\sigma_{yy}^{\infty} x}{\sqrt{a^2-x^2}}\, dx \right\}$$

Re-arranging,

$$\sum_{n=0}^{N} b_n U_n(\xi) \approx \frac{-1}{2(1-k^2)G} \left\{ \frac{2\alpha}{\pi} \sum_{n=0}^{N} \int_{-a}^{x} \frac{b_n}{\sqrt{a^2-x^2}} \left[\int_{-a}^{a} \frac{U_n(\xi)\sqrt{a^2-\xi^2}}{\xi - x}\, d\xi \right] dx + \sigma_{yy}^{\infty} \int_{-a}^{x} \frac{x}{\sqrt{a^2-x^2}}\, dx \right\} \tag{5}$$

Using Relation between Chebyshev Polynomials $T_n(\xi)$ and $U_n(\xi)$ [Mason and Handscomb, 2003],

$$\int_{-1}^{1} \frac{U_n(\xi)\sqrt{1-\xi^2}}{\xi - x}\, d\xi = -\pi T_{n+1}(x)$$

where $T_k(x)$ is Chebyshev Polynomial of first kind of degree k.

Here we have,

$$\int_{-a}^{a} \frac{U_n(\xi)\sqrt{a^2-\xi^2}}{\xi - x}\, d\xi \tag{6}$$

Substituting ξ to $a\xi$ in eq. (6) we get,

$$\int_{-1}^{1} \frac{U_n(a\xi)\sqrt{a^2-a^2\xi^2}}{a\xi - x}\, a d\xi = a \int_{-1}^{1} \frac{U_n(a\xi)\sqrt{1-\xi^2}}{\xi - \frac{x}{a}}\, d\xi = -a\pi T_{n+1}(x) \tag{7}$$

Substituting (7) in (5),

$$\sum_{n=0}^{N} b_n U_n(x) \approx \frac{-1}{2(1-k^2)G} \left\{ \frac{2\alpha}{\pi} \sum_{n=0}^{N} \int_{-a}^{x} \frac{b_n}{\sqrt{a^2-x^2}} \left[-a\pi T_{n+1}(x)\right] dx \; + \sigma_{yy}^{\infty} \int_{-a}^{x} \frac{x}{\sqrt{a^2-x^2}} dx \right\}$$

Re-arranging,

$$\sum_{n=0}^{N} b_n U_n(x) \;=\; \frac{a\alpha}{(1-k^2)G} \sum_{n=0}^{N} \int_{-a}^{x} \frac{b_n\, T_{n+1}(x)}{\sqrt{a^2-x^2}}\, dx \;-\; \frac{\sigma_{yy}^{\infty}}{2(1-k^2)G} \int_{-a}^{x} \frac{x}{\sqrt{a^2-x^2}}\, dx \tag{8}$$

we can evaluate the second integral of eq. (8) directly,

$$\int_{-a}^{x} \frac{x}{\sqrt{a^2-x^2}}\, dx \;=\; -\sqrt{a^2-x^2} \tag{9}$$

Substituting eq. (9) in eq. (8) and also $\frac{a\alpha}{G} = \alpha^*$,

$$\sum_{n=0}^{N} b_n U_n(x) \;=\; \frac{\alpha^*}{(1-k^2)} \sum_{n=0}^{N} b_n \int_{-a}^{x} \frac{T_{n+1}(x)}{\sqrt{a^2-x^2}}\, dx \;+\; \frac{\sigma_{yy}^{\infty} \sqrt{a^2-x^2}}{2(1-k^2)G}$$

Re-arranging

$$\boxed{\sum_{n=0}^{N} b_n \left(U_n(x) \;-\; \frac{\alpha^*}{(1-k^2)} \int_{-a}^{x} \frac{T_{n+1}(x)}{\sqrt{a^2-x^2}}\, dx \right) \;=\; \frac{\sigma_{yy}^{\infty}}{2(1-k^2)G} \sqrt{a^2-x^2}} \tag{10}$$

Code for solving this equation numerically is given in Appendix- A which take crack-half length$=$ a , α^* , G , k , σ_{yy}^{∞} as input.

3. Comparision with Perturbation Solution

An exact formulation is known for the crack aperture in the presence of a known spatially-variable normal hydrostatic pressure acting along the crack faces [Sneddon and Lowengrub, 1969]. The present problem is different, in the sense that the cohesive stress is not known. A regular perturbation technique [Hinch, 1991] will be utilised to address the problem. The classical solution for the case of no cohesive force will be taken as the zeroth-order solution. The aperture that is obtained in the zeroth-order solution is then used to calculate the cohesive force, according to the law $\sigma_y^0(\xi) = 2\alpha v(\xi)$. This force is then applied to the surfaces of the crack, and Sneddon's formulation is used to calculate the additional crack aperture and stress field due to this surface force. This new incremental aperture will give rise to an additional cohesive force and so this process would need to be iterated until convergence is obtained. Convergence of this perturbation process could be defined. for example, as the stage at which the incremental aperture compute at step N is less than some prescribed fraction of the zeroth-order aperture. For simplicity, only one step of this iterative process will be considered, which is expected to be sufficient for validating the accuracy of the numerical solution procedure.

For the case of a spatially variable pressure acting on the crack face, the aperture variable for the zeroth-order solution plus the first-order perturbation correction can be written as [Sneddon and Lowengrub, 1969]

$$v_{sn}(x) \;=\; \frac{2p_0(1-\nu^2)a}{E}\left[\sqrt{1-(x^2/a^2)} - \frac{2(1-\nu^2)a}{E}\int\limits_{x/a}^{1} \frac{tq(t)}{\sqrt{t^2-(x/a)^2}}\,dt\right] \tag{11}$$

where,

$$q(t) = \frac{2}{\pi}\int\limits_{0}^{t} \frac{p(u)/p_0}{\sqrt{t^2-u^2}}\,du \tag{12}$$

with,

$$p(u) = 2\alpha p_0\sqrt{1-(u^2/a^2)} \tag{13}$$

Here,crack-half length is a, Young's Modulus E, and Poisson's Ration ν.

for chebyshev approximation solution we use condition $\sigma_{yy}^{\infty} \,=\, 1$ and
for perturbation solution we use condition $p_0 = 1$.
These both condtions are same as [Reason i don't know]

4. Results

4.1. Crack Aperture

A line crack that extends from $b = -a$ to $c = a$ with $a = 1$ (for concrete numerical purposes). is considered. Validation of the numerical results can be done by comparision with the first order perturbation solution. The two approaches should closely agree for small values of α, which can be expressed in dimensionless form as $\alpha^* = \alpha a/G$

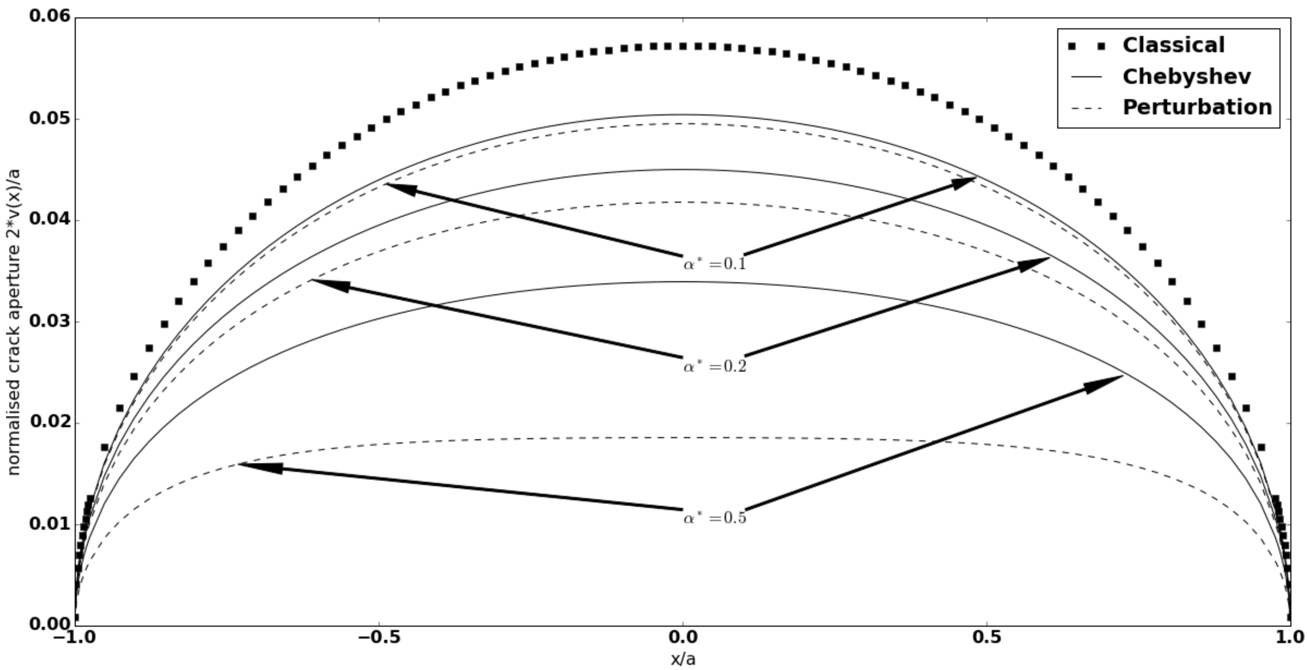


Figure 1: Crack face separation in presence of cohesive stress with parameter α^* when applied load is σ_{yy}^{∞}

As can be observed in fig. 1, for small values of α^*, the perturbation solution agrees closely with the full cohesive stress solution. As α^* is increased, the perturbation solution becomes less accurate, so that for $\alpha^* = 0.5$ and beyond, the comparision are no longer meaningful.

4.2. Error

A further test to validate the results is to track the error in the calculation of correction aperture in the cohesive solution at the mid point of crack i.e $x - 0$.

$$Error_{x=0} = \frac{(Aperture_{Cohesive} \; - \; Aperture_{Perturbation})_{x=0}}{(Aperture_{Classical} \; - \; Aperture_{Perturbation})_{x=0}}$$

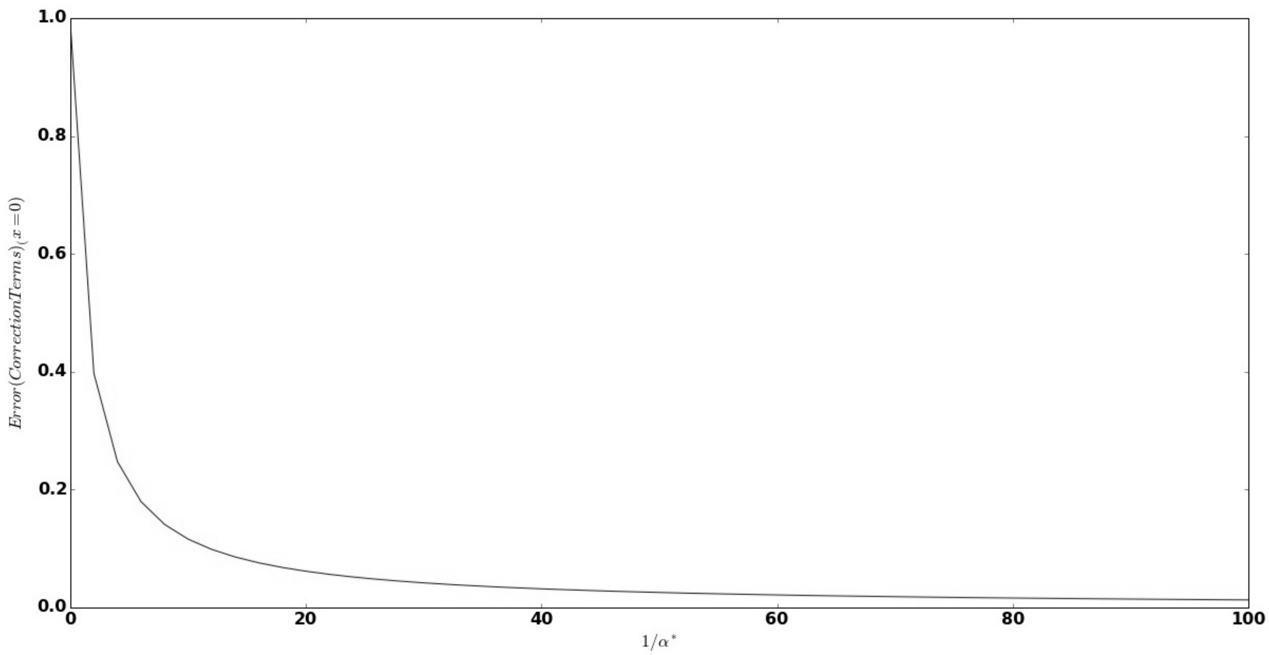


Figure 2: Variation of Error at $x = 0$ between the Chebyshev cohesive solution and the first order perturbation solution as a function of α^*

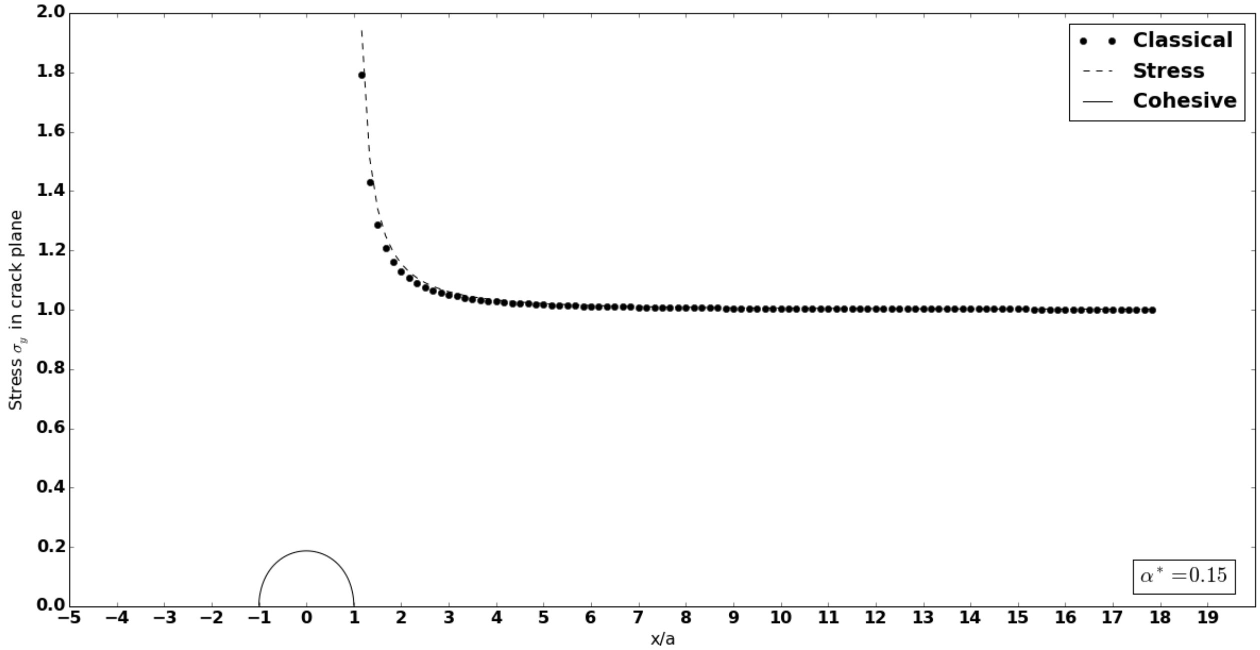
As can be observed from fig. 2, the error reduces as α^* decreases which implies that as α^* decreases perturbation solution comes closer to cohesive solution.

4.3. Stress Along the Crack Line

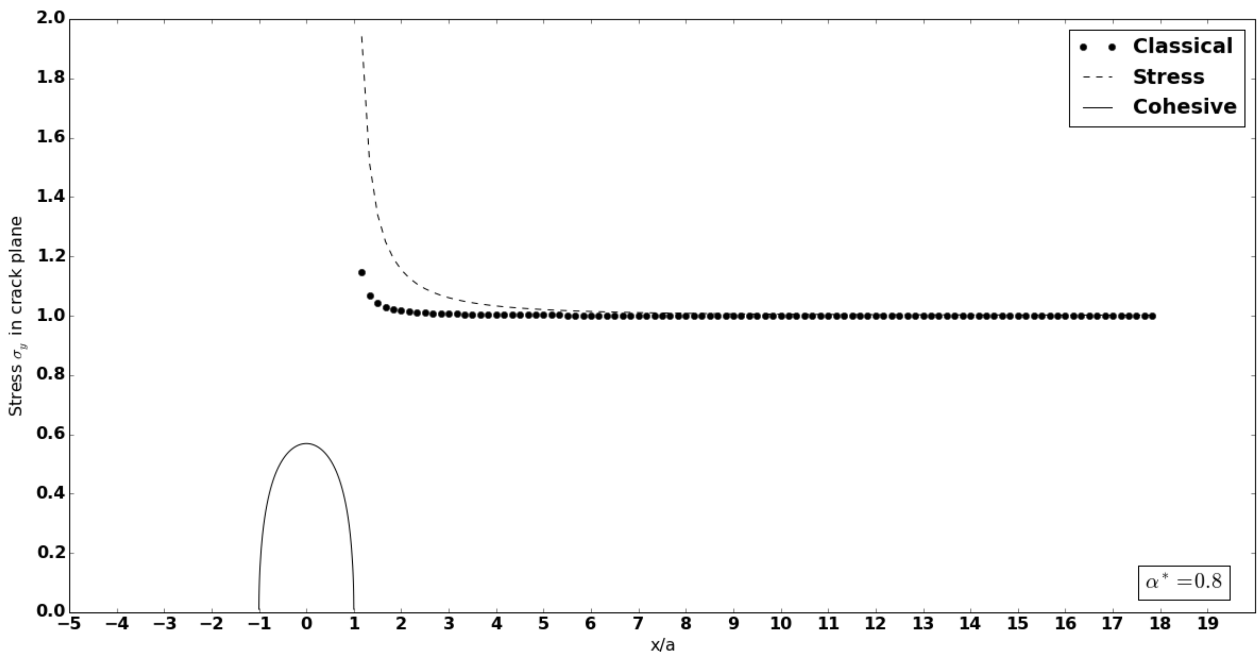
After the aperture ($2v_+$) and cohesive stress $\sigma_y^0(\xi) = 2\alpha v_+(\xi)$ have been calculated.

$$\sigma_y = 2\Re p'(x) = \pm \frac{1}{\sqrt{(x-b)(x-c)}} \left\{ \frac{1}{\pi} \int_b^c \frac{\sigma_y^0(\xi)\sqrt{(\xi-b)(c-\xi)}}{\xi - x} \, d\xi \; + \; \sigma_{yy}^{\infty}[x - \frac{b+c}{2}] \right\} \tag{14}$$

eq. (14) can be solved for the stress σ_{yy} along the crack line.



(a)



(b)

Figure 3: Variation of σ_y in eq. (14) across the crack line for different values of cohesive stress

5. Appendix

```
#packages used

#packages to be downloaded
import numpy as np

import scipy.integrate as integrate
from scipy import linalg as la
import scipy.special as Cheby
from scipy.special import ellipe

import matplotlib.pyplot as plt
font = {'family' : 'normal',
        'weight' : 'bold',
        'size'   : 16}
plt.rc('font', **font)

#packages pre-installed
from math import pi
print 'Step 1'
#constants
a = 1               #Crack Half-Length
nu = 0.25           #Poisson's Ratio
G = 26.2            #Shear Stress
E = 2*G*(1+nu)      #Young's Modulus
ks=(1-2*nu)/(2*(1-nu)) #Plain Strain Condition

degree = 80                 #degree of Chebyshev's Approximation
order = degree + 1          #order of Chebyshev's Approximation

#values of x to use for plot
X = np.linspace(-a,a,order+1,False)
X = np.delete(X,0)
Y1 = np.linspace(-0.9999,X[0],10,False)
Y2 = np.linspace(0.9999,X[-1],10,False)
X = np.concatenate((X,Y1,Y2), axis=0)
X.sort()
order += 20

def First(x):            #For finding first term of LHS for given x
    first = np.zeros([order,])
    for n in xrange(order):
        first[n] = Cheby.eval_chebyu( n , x )
    return first
```

```python
def Second(x):          #For finding second term of LHS for given x
    second = np.zeros([order,])
    for n in range(order):
        func = lambda z:(Cheby.eval_chebyt(n + 1,z))/(np.sqrt(1-z**2))
        second[n] = integrate.quad(func,-1,x)[0]
    return second

#To find Chebyshev Approximation Coefficients for given alpha*
def Chebysol(alphas):
    A = np.zeros([order,order])

    B = (np.sqrt(np.power(a,2)-np.power(X,2)))/(2*G*(1-ks))

    for x in range(order):
        A[x] = First(X[x])

    for x in range(order):
        A[x] -= alphas*Second(X[x])/(1-ks)

    b = la.solve(A,B)
    return b

#To find value for given Chebyshev coefficients and x
def Chebysol_Mat(x,b):
    Sol_Mat = 0.0
    for i in range(order):
            Sol_Mat += b[i]*Cheby.eval_chebyu( i , x )
    return Sol_Mat

def Persol(alphas):      #To find Perturbation Solution for given alpha
    per = np.zeros([order,])
    for i in range(order):
        x = X[i]
        func = lambda t : (t * ellipe(t**2)/((t**2-x**2)**0.5))
        ans0=integrate.quad(func,abs(x),1)[0]
        ans = 4*alphas*(1-nu)*ans0/pi
        ans1 = 2*(1-nu**2)*((1-x**2)**0.5 - ans)/E
        per[i] = 2*ans1
    return per
print 'Step 2'
#Classical Solution i.e alpha* = 0
b_Classical = Chebysol(0)
Classical_Mat = np.zeros([order,])
for i in range(order):
    Classical_Mat[i] = Chebysol_Mat(X[i],b_Classical)
print 'Step 3'
#Chebyshev Solution for alpha* = 0.1
```

```
b_Cheby_1 = Chebysol(0.1)
Cheby_Mat_1 = np.zeros([order,])
for i in range(order):
    Cheby_Mat_1[i] = Chebysol_Mat(X[i],b_Cheby_1)
print 'Step 4'
#Chebyshev Solution for alpha* = 0.2
b_Cheby_2 = Chebysol(0.2)
Cheby_Mat_2 = np.zeros([order,])
for i in range(order):
    Cheby_Mat_2[i] = Chebysol_Mat(X[i],b_Cheby_2)
print 'Step 5'
#Chebyshev Solution for alpha* = 0.5
b_Cheby_3 = Chebysol(0.5)
Cheby_Mat_3 = np.zeros([order,])
for i in range(order):
    Cheby_Mat_3[i] = Chebysol_Mat(X[i],b_Cheby_3)
print 'Step 6'
#Perturbation Solutions for above 3 alpha*
Per_Mat_1 = Persol(0.1)
Per_Mat_2 = Persol(0.2)
Per_Mat_3 = Persol(0.5)
print 'Step 7'
#Graph Plot for Classical ,Cohesive and Perturbation
fig1 = plt.figure(num=1, figsize=(20, 10),
                  dpi=50, facecolor='w', edgecolor='k')
ax1 = plt.subplot(111)
ax1.set_xlabel('x/a')
ax1.set_ylabel('normalised crack aperture 2*v(x)/a')
ax1.plot(X,2*Classical_Mat,'s',color='k',label="Classical")
ax1.plot(X,2*Cheby_Mat_1,'-',color='k',label="Cohesive")
ax1.plot(X,Per_Mat_1,'--',color='k',label="Perturbation")

ax1.plot(X,2*Cheby_Mat_2,'-',color='k')
ax1.plot(X,Per_Mat_2,'--',color='k')

ax1.plot(X,2*Cheby_Mat_3,'-',color='k')
ax1.plot(X,Per_Mat_3,'--',color='k')

ax1.annotate(r'$\alpha^* = 0.1$',xy=(X[30],Per_Mat_1[30]), xytext=(X[50],
        0.035),arrowprops=dict(width = 2,headwidth = 10,facecolor='black'))
ax1.annotate('',xy=(X[70],2*Cheby_Mat_1[70]), xytext=(X[54],0.0365),
             arrowprops=dict(width = 2,headwidth = 10,facecolor='black'))

ax1.annotate(r'$\alpha^* = 0.2$',xy=(X[25],Per_Mat_2[25]), xytext=(X[50],
        0.025),arrowprops=dict(width = 2,headwidth = 10,facecolor='black'))
ax1.annotate('',xy=(X[75],2*Cheby_Mat_2[75]), xytext=(X[54],0.0265),
             arrowprops=dict(width = 2,headwidth = 10,facecolor='black'))
```

```
ax1.annotate(r'$\alpha^* = 0.5$',xy=(X[20],Per_Mat_3[20]), xytext=(X[50],
        0.010),arrowprops=dict(width = 2,headwidth = 10,facecolor='black'))
ax1.annotate('',xy=(X[80],2*Cheby_Mat_3[80]), xytext=(X[54],0.0113),
             arrowprops=dict(width = 2,headwidth = 10,facecolor='black'))

plt.legend(loc=1)
fig1.savefig("Result.png")
print 'Step 8 (Result Image Saved)'
#For plotting error graph
alp = np.zeros([100,])  #different alpha* matrix
C = Classical_Mat[50]

for i in range(1,51):
    alp[i-1] = 1.0/(i*2)
for i in range(51,101):
    alp[i-1] = float(i)

alp.sort()
print 'Step 9'
#Finding Error for given alpha*
err = np.zeros([100,])
for i in range(100):
    b_temp = Chebysol(alp[i])
    P = Persol(alp[i])[50]
    err[i] = (2*Chebysol_Mat(X[50],b_temp) - P)/(2*C - P)
print 'Step 10'
#Graph Plot for error vs 1/alpha*
fig2 = plt.figure(num=2, figsize=(20, 10),
                  dpi=50, facecolor='w', edgecolor='k')
ax2 = plt.subplot(111)
ax2.set_xlabel(r'$1/\alpha^*$')
ax2.set_ylabel(r'$Error(Correction Terms) (x=0)$')
ax2 = plt.subplot(111)
ax2.plot(1/alp,err)
fig2.savefig("Error.png")
print 'Step 11 (Error Image Saved)'
X_2 = np.linspace(1,18,order+1,False)
X_2 = np.delete(X_2,0)

#Finding values of stress for classical solution outside crack
C_stress = np.zeros([order,])
for i in range(order):
    C_stress[i] = (X_2[i])/(X_2[i]**2 - 1)**0.5
print 'Step 12'
#Finding values of stress outside crack
stress_1 = np.zeros([order,])
```

```
for i in range(order):
    temp = 0.0
    for n in range(order):
        func = lambda z:b_Cheby_1[n]*Cheby.eval_chebyu(n,z)
                        *((1 - z**2)**0.5)/(z - X_2[i])
        temp += integrate.quad(func,-1,1)[0]
    stress_1[i] = (2*0.15*G*temp/pi + X_2[i])/(X_2[i]**2 - 1)**0.5
print 'Step 13'
stress_2 = np.zeros([order,])
for i in range(order):
    temp = 0.0
    for n in range(order):
        func = lambda z:b_Cheby_1[n]*Cheby.eval_chebyu(n,z)
                        *((1 - z**2)**0.5)/(z - X_2[i])
        temp += integrate.quad(func,-1,1)[0]
    stress_2[i] = (2*0.8*G*temp/pi + X_2[i])/(X_2[i]**2 - 1)**0.5
print 'Step 14'
#Chebyshev Solution for alpha* = 0.15
b_Cheby_4 = Chebysol(0.15)
Cheby_Mat_4 = np.zeros([order,])
for i in range(order):
    Cheby_Mat_4[i] = Chebysol_Mat(X[i],b_Cheby_4)
print 'Step 15'
#Chebyshev Solution for alpha* = 0.8
b_Cheby_5 = Chebysol(0.8)
Cheby_Mat_5 = np.zeros([order,])
for i in range(order):
    Cheby_Mat_5[i] = Chebysol_Mat(X[i],b_Cheby_5)
print 'Step 16'
#Graph Plot for stress with alpha* = 0.15
fig3 = plt.figure(num=3, figsize=(20, 10), dpi=50,
                  facecolor='w', edgecolor='k')
ax3 = plt.subplot(111)
ax3.set_xlabel('x/a')
ax3.set_ylabel(r'Stress $\sigma_y$ in crack plane')
ax3.plot(X_2,stress_1,'o',color='k',label="Classical")
ax3.plot(X_2,C_stress,'--',color='k',label="Stress")
ax3.plot(X,4*0.15*G*Cheby_Mat_4,color='k',label="Cohesive")
plt.legend(loc=1)
plt.text(18.5, 0.1,r'$\alpha^* = 0.15$', size=20,ha="center",
         va="center",bbox = dict(boxstyle="square",
                                 ec=(0., 0., 0.),fc=(1., 1., 1.)))
fig3.savefig("Stress_1.png")
print 'Step 17 (Stress Image 1 is saved)'
#Graph Plot for stress with alpha* = 0.8
fig4 = plt.figure(num=4, figsize=(20, 10), dpi=50,
                  facecolor='w', edgecolor='k')
```

```
ax4 = plt.subplot(111)
ax4.set_xlabel('x/a')
ax4.set_ylabel(r'Stress $\sigma_y$ in crack plane')
ax4.plot(X_2,stress_2 ,'o',color='k',label="Classical")
ax4.plot(X_2,C_stress ,'--',color='k',label="Stress")
ax4.plot(X,4*0.8*G*Cheby_Mat_5,color='k',label="Cohesive")
plt.legend(loc=1)
plt.text(18.5, 0.1,r'$\alpha^* = 0.8$', size=20,ha="center",
         va="center",bbox = dict(boxstyle="square",
                                 ec=(0., 0., 0.),fc=(1., 1., 1.)))
fig4.savefig("Stress_2.png")
print 'Step 18 (Stress Image 2 Saved)'
print 'Completed'
```

Broberg, K. B., 1999. 4 - elastostatic cracks. In: Broberg, K. B. (Ed.), Cracks and Fracture. Academic Press, San Diego, pp. 99 – 246.

Hinch, E., 1991. Perturbation Methods. Cambridge Texts in Applied Mathematics. Cambridge University Press.

Mason, J. C., Handscomb, D. C., 2003. Chebyshev Polynomials. CRC Press LLC, Boca Raton, Florida.

Sneddon, I., Lowengrub, M., 1969. Crack problems in the classical theory of elasticity.