K_I^E calculations according to Rice's theory

.

·Division of Mechanics, Lund University

June 20, 2022

Abstract

Here we calculate the critical stress intensity factor value according to Rice's crack-tip plasticity theory (i.e. $\mathrm{K_{IE}}$) for different single crystal crack systems.

Keywords Critical stress intensity factor, crack-tip plasticity, dislocation, stacking fault, slip plane and slip direction.

1 Introduction

Crack-tip plasticity is characterized by nucleation of dislocations and twins at preexisting crack-tips.

2 Rice's brittle fracture

Rice's theory [1] suggests that the critical stress intensity factor K_{IE} at which a dislocation is nucleated at crack-tip eventually moving on to an accessible slip plane under mode I loading and plane strain conditions is given by [2, 3]

$$K_{IE} = \sqrt{\frac{G_{IE}}{B}}, \tag{1}$$

with $G_{IE} = 8\frac{1+(1-\nu_{yx})\tan^2\phi}{(1+\cos\theta)\sin^2\theta}\gamma_{us}$ where γ_{us} is the unstable stacking fault energy. The angles θ and ϕ are shown in Figure 1 [2].

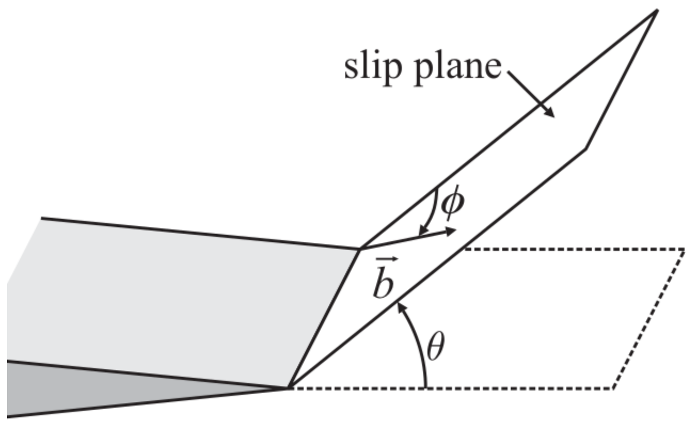

Figure 1: Crack plane, slip system with slip direction $\vec{b}$ and angles θ and ϕ.

2.1 Measurement of angles θ and ϕ and calculation of K_I^E

2.1.1 Procedure to find angles θ and ϕ

- Input appropriate vectors for crack-plane ($\vec{CP}$) and crack-front ($\vec{CF}$) e.g. (indices are multiplied by a,a and c) $\vec{CP}$ = [0*a, 0*a, 4*c] = [u,v,w] and $\vec{CF}$ = [1*a, 0*a, 0*c] = [m,n,o] where a and c are the magnitudes of lattice vectors (Now orthogonal basis vectors are [a,0,0], [0,a,0] and [0,0,c]) instead of [a,0,0], [0,a,0] and [0,0,a]). Then crack propagation direction $\vec{CP}r = \vec{CP} \times \vec{CF}$. Also input appropriate vectors for slip planes ($\vec{SP}$) and directions ($\vec{b} = [b_1, b_2, b_3]$).

$$\theta = \frac{\pi}{2} - cos^{-1}\left\{\frac{\vec{CP} \cdot \vec{SP}}{|\vec{CP}||\vec{SP}|}\right\} \tag{2}$$

- Let the burger vector, $\vec{b}$ make an angle ϕ with the vector $[p, q, r]$ (Unknown at this point). θ and ϕ must also satisfy the following conditions.
(i) With $|[p, q, r]| = \sqrt{p^2 + q^2 + r^2} = \sqrt{|[u, v, w]|}sin\theta = (\sqrt{u^2 + v^2 + w^2})sin\theta$,
$[p, q, r] \cdot \vec{CP} = |[p, q, r]||[u, v, w]|cos(\frac{\pi}{2} - \theta) = |[p, q, r]||[u, v, w]|sin(\theta)$ i.e.

$$pu + qv + rw = (u^2 + v^2 + w^2)sin\theta \tag{3}$$

(ii) $\vec{SP} = [k, l, m]$ is perpendicular to $[p, q, r]$. Therefore, $[p, q, r] \times [k, l, m] = 0$ i.e.

$$pk + ql + rm = 0 \tag{4}$$

(iii) Cross product $[p, q, r] \times \vec{CF} = [p, q, r] \times [m, n, o] = 0$ i.e.

$$pm + qn + ro = 0 \tag{5}$$

- The following equation can be used to solve the system of linear equations formed by equations (3)-(5).

$$\begin{bmatrix} k & l & m \\ u & v & w \\ m & n & o \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ (u^2 + v^2 + w^2)sin\theta \\ 0 \end{bmatrix} \tag{6}$$

- By solving equation (6), we get $[p, q, r]$.
Since $[p, q, r] \cdot \vec{b} = [p, q, r] \cdot [b_1, b_2, b_3] = \sqrt{p^2 + q^2 + r^2}\sqrt{b_1^2 + b_2^2 + b_3^2}cos\phi$, we have,

$$\phi = cos^{-1}\left\{\frac{pb_1 + qb_2 + rb_3}{\sqrt{p^2 + q^2 + r^2}\sqrt{b_1^2 + b_2^2 + b_3^2}}\right\} \tag{7}$$

2.1.2 K_I^E calculation

Insert the θ, ϕ, γ_{us} and ν_{yx} (we will choose crack system such $\phi = 0$ which in turn leaves ν_{yx} obsolete) in equation (1), to calculate K_I^E. Therefore, we need to calculate only θ.

3 Results and discussion

In the following, K_I^E for dislocation slip on (330)[001] (stacking fault energy curve in Figure 9 of manuscript in preparation. $\gamma_{us,(330)} = 2.44$ Jm^{-2} $< \gamma_{us,(004)} = 3.5$ Jm^{-2})(010)[001] crack system

For σ-phase with 0% Re, a=9.538 Å and c=5.070 Å are chosen.

Structure 9 of σ-phase:

The elastic constants' matrix is $C = \begin{pmatrix} 533 & 246 & 205 & 0.0 & 0.0 & 0.0 \\ 246 & 533 & 205 & 0.0 & 0.0 & 0.0 \\ 205 & 205 & 562 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 105 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 105 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 122 \end{pmatrix}$

4 How to use the code

Enter crack plane and crack front on the lines-11 and 16 respectively in the following 'Kie_code.py'. Then enter slip planes and directions on lines 24 and 25 respectively. To calculate K_I^E, γ_{us} should be given on line 118 and rotated elastic constants on line 128.

```
1 from __future__ import division
2 import numpy as np
3 import io
4 import scipy
5 from array import *
6 import os
7 import math
8
9
10 #crack_prop=np.array([0,-1,0]) #x-axis
11 perp_crack_plane=np.array([1,1,2]) #y-axis  Enter the vectors such that [1*a,1*a,2*c] i.e indices are multiplied by a,a and c.
12 u=perp_crack_plane[0]
13 v=perp_crack_plane[1]
14 w=perp_crack_plane[2]
15
16 crack_front=np.array([1,1,-1]) #z-axis    Enter the vectors such that indices are multiplied by a,a and c.
17 m=crack_front[0]
18 n=crack_front[1]
19 o=crack_front[2]
20
21 #Find crack propagation plane
22 crack_prop=np.cross(perp_crack_plane,crack_front)
23
24 Slip_planes=np.array([[1,-1,0],[-1,1,0]])  # Enter the vectors such that indices are multiplied by a,a and c.
25 Slip_dirs=np.array([[1,1,1],[1,1,1]])      # Enter the vectors such that indices are multiplied by a,a and c.
26
27
28 a=Slip_planes[:,0]  #x-component of slip plane vectors
29 b=Slip_planes[:,1]  #y-component of slip plane vectors
30 c=Slip_planes[:,2]  #z-component of slip plane vectors
31
32
33 #burger vectors components
34 b1=Slip_dirs[:,0]   #x-component of burger vectors
35 b2=Slip_dirs[:,1]   #y-component of burger vectors
36 b3=Slip_dirs[:,2]   #z-component of burger vectors
37
38
39 theta_angles=np.zeros(len(Slip_dirs))
40 theta_angles_radian=np.zeros(len(Slip_dirs))
41 phi_angles=np.zeros(len(Slip_dirs))
42
43 result_text=[]*len(Slip_dirs)
44
45 for i in range(0,len(Slip_dirs),1):
46   dot_prod=0.0
47   mag_prop=0.0
48   mag_slip_plane=0.0
49
50   dot_prod=np.dot(crack_prop,Slip_planes[i,:])                    # Dot product of the two vectors
51   mag_prop=np.linalg.norm(crack_prop)                             # Magnitude of the vector
52   mag_slip_plane=np.linalg.norm(Slip_planes[i,:])                 # Magnitude of the vector
53
54 # The following block uses equation (2) in report for calculation of Theta angle.
55   cos_theta1=(dot_prod)/(mag_prop*mag_slip_plane)                 # Finding angle theta
56   cos_theta1=np.round(cos_theta1,3)
57   theta1=(180*np.arccos(cos_theta1))/math.pi                      # Converting to degrees
58   theta=90-theta1
```

```
  theta_angles[i]=theta


#converting into radians
  theta1=((math.pi)/180)*theta1
  theta=((math.pi)/180)*theta
  theta_angles_radian[i]=theta


### Now for phi angle using equations (3), (4), (5) and (6)
  A=np.zeros(3)
  A=np.array([[a[i],b[i],c[i]],[u,v,w],[m,n,o]])    # 3X3 matrix from equation (6) in report


  C=np.zeros(3)
  C=np.array([0,((u*u)+(v*v)+(w*w)*math.sin(theta)),0])  # 3X1 matrix (vector) on the RHS of '=' in equation (6) of report


  if (theta!=0.0):
    X=np.zeros(3)
    X=np.linalg.solve(A, C)       # Solving for [p,q,r] in equation (6) in report


# Following block uses equation (2) in report
    cos_phi=((X[0]*b1[i])+(X[1]*b2[i])+(X[2]*b3[i]))/((np.sqrt((X[0]*X[0])+(X[1]*X[1])+(X[2]*X[2])))*(np.sqrt((b1[i]*b1[i])+(b1[i]*b1[i])+(b1[i]*b1[i]))))
    cos_phi=np.round(cos_phi,3)
    phi=(180*np.arccos(cos_phi))/math.pi
    phi_angles[i]=phi

  else:
    phi_angles[i]=0.0

#  print ('Slip plane: ('+str(Slip_planes[i,0])+str(Slip_planes[i,1])+str(Slip_planes[i,2])+')\t'+'Slip direction(burger vector): ['+ str(Slip_dirs[i,0])+str(Slip_dirs[i,1])+str(Slip_dirs[i,2])+']'+'\t angle theta: '+str(theta_angles[i])+'\t angle phi: '+str(phi_angles[i])+'\n')


################################### KIE calculation using equation (1) in report ###########################################################


K_emit_110=[]*int(len(Slip_dirs)*1)
K_emit_110_test=[]*int(len(Slip_dirs)*1)

print ('theta_angles=',theta_angles)
print ('phi_angles=',phi_angles)

j=0

theta_new=theta_angles
phi_new=phi_angles
G_G=0.0
for j in range(0,int(len(Slip_dirs)*1),1):
    angle_theta=theta_angles[j]
  angle_phi=phi_angles[j]
  if ((angle_theta>90.0)):
    angle_theta=180-angle_theta
        if (angle_phi>90):
    angle_phi=180-angle_phi

        print (angle_theta, angle_phi)
  nu_yx=0.278     # This will be zero for our cases
  gamma_us=1.5775
  theta=((math.pi)/180)*angle_theta
  phi=((math.pi)/180)*angle_phi
  denom=((1+math.cos(theta))*(pow(math.sin(theta),2)))*gamma_us
  #if (denom==0.0):
    #    G_G=0.0
  if (denom!=0.0):
```

```
    G_G=8*((1+(1-nu_yx)*pow(math.tan(phi),2))/((1+math.cos(theta))*(pow(math.sin(theta),2))))*gamma_us


  C=np.array([[532.55*1e9,204.95*1e9,204.95*1e9,0.0,0.0,0.0],[204.95*1e9,532.55*1e9,204.95*1e9,0.0,0.0,0.0],[204.95*1e9,204.95*1e9,532.55*1e9,0.0,0.0,0.0],[0.0,0.0,0.0,163.13*1e9,0.0,0.0],[0.0,0.0,0.0,0.0,163.13*1e9,0.0],[0.0,0.0,0.0,0.0,0.0,163.13*1e9]])   # Elastic constant matrix in appropriate orientation


  S=np.linalg.inv(C)     # Compliance constant matrix

  s11=S[0,0]
  s12=S[0,1]
  s13=S[0,2]
  s22=S[1,1]
  s33=S[2,2]
  s23=S[1,2]
  s26=S[1,5]
  s66=S[5,5]


##bij are in GPa^-1
  b_11=((s11*s33)-(s13*s13))/s33
  b_22=((s22*s33)-(s23*s23))/s33
  b_12=((s12*s33)-(s13*s23))/s33
  b_66=((s66*s33)-(s26*s26))/s33

#### B from equation (1) in report will be in GPa^-1
  temp_1=(b_11*b_22*0.5)
  temp_2=np.sqrt(b_22/b_11)
  temp_3=((2*b_12)+b_66)
  temp_4=(2*b_11)

  B_1=np.sqrt((temp_1)*(temp_2+(temp_3/temp_4)))
  K_G_N=np.sqrt(G_G/B_1)     # equation (1) in report
  K_G_N_MPa_root_m=K_G_N/1e6



  print('Slip plane: ('+str(Slip_planes[j,0])+str(Slip_planes[j,1])+str(Slip_planes[j,2])+')\t Slip direction (burger vector): '+'['+ str(Slip_dirs[j,0])+str(Slip_dirs[j,1])+str(Slip_dirs[j,2])+']'+'\t angle theta: '+str(theta_angles[j])+'\t angle phi: '+str(phi_angles[j])+'\t K_IE: '+str(K_G_N_MPa_root_m)+'\n')
```

Acknowledgements

References

[1] Alan Arnold Griffith. VI. the phenomena of rupture and flow in solids. *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character*, 221(582-593):163–198, 1921.

[2] James R Rice. Dislocation nucleation from a crack tip: an analysis based on the peierls concept. *Journal of the Mechanics and Physics of Solids*, 40(2):239–271, 1992.

[3] James R Rice. Mathematical analysis in the mechanics of fracture. *Fracture: An Advanced Treatise*, 2:191–311, 1968.