

# Gamze Gumuslu - Mini Project 1

November 2, 2012

## Contents

### 1 Project definition

Implement a linear programming approach to phase stability. You may use the formation energies for Cu-Pd alloys from dft-book, or other data, e.g. from <http://materialsproject.org>. Prepare a figure that shows what phases are present as a function of composition from pure Cu to pure Pd.

### 2 Steps for solution of the problem

#### 2.1 Delaunay triangulation and convex hull

The composition versus formation energy of the alloy data were taken from <http://alloy.phys.cmu.edu/cgi-bin/mkp.csh>.

Table 1: Formation energy of Cu-Pd alloys

Phase	Cu fraction	Formation energy(eV/at)
Cu	1.0	0.0
$\text{Cu}_3\text{Pd}$	0.75	-0.1185
CuPd	0.50	-0.134
Pd	0.0	0

These points are the corners of the convex hull. The convex hull is drawn by making use of delaunay triangulation.

## 2.2 Setting up the optimization problem

Any point that is within the convex hull shows that the formation energy of the alloy at the given composition is not a minimum, so that the phase is not stable. Hence, the energy will be minimized by phase separation. The aim in this project is to find the phase separation that gives the minimum energy. The minimized H will be the combination of the formation energies of the alloys of whose phase structure the calculated alloy is separated into. The objective function to be minimized can be given as follows:

$$H_{\min} = 0 \cdot X_{\text{Cu}} - 0.1185 \cdot X_{\text{Cu}_3\text{Pd}} - 0.134 \cdot X_{\text{CuPd}} + 0 \cdot X_{\text{Pd}}$$

where X shows the fraction of phase present. Thus, the fractions of the phases the calculated alloy should separate in order to minimize its energy will be calculated. However, there are some constraints on this minimization. First one is that the summation of all the fractions should be equal to 1.

$$\text{constraint 1} \dots X_{\text{Cu}} + X_{\text{Cu}_3\text{Pd}} + X_{\text{CuPd}} + X_{\text{Pd}} = 1.0$$

The second constraint comes from the Lever Rule on phase diagrams. While applying Lever rule, it should also be kept in mind that the total number of atoms of the components of the alloy should be conserved when phase separation occurs. It would be more clear to write it in equation. Let's think over the total number of Cu atoms present. Let  $x_c$  be the fractions of the Cu atoms in an arbitrary Cu-Pd alloy where the total number of atoms is given as N. we will make the use of the Cu concentration in the stable phases given in Table 1.

$$\text{constraint 2} \dots N \cdot 1.0 \cdot X_{\text{Cu}} + N \cdot 0.75 \cdot X_{\text{Cu}_3\text{Pd}} + N \cdot 0.5 \cdot X_{\text{CuPd}} + N \cdot 0 \cdot X_{\text{Pd}} = N \cdot x_c$$

This equation can be written as:

$$\text{constraint 2} \dots 1.0 \cdot X_{\text{Cu}} + 0.75 \cdot X_{\text{Cu}_3\text{Pd}} + 0.5 \cdot X_{\text{CuPd}} + 0 \cdot X_{\text{Pd}} = x_c$$

Also, we have upper and lower boundaries. Each parameter (fraction of the phase) should be between 0 and 1. So, the object function should be minimized with respect to these two constraints and the given boundaries. To do this optimization in python, `fmin_slsqp` is used since this is an optimization with constraints and boundaries.

## 3 Code and results

---

```
1 import numpy as np
2 from scipy.optimize import fmin_slsqp # import the optimization
3 from scipy.spatial import Delaunay    # import the Delaunay to form the convex hull
4 import matplotlib.pyplot as plt
5 from matplotlib.collections import LineCollection
6
```

```

7  #-----
8  #---DELAUNAY TRIANGULATION &-----
9  #-----CONVEX HULL-----
10 #-----
11 # points in 2-d, these points are the values taken from Table 1
12 points = np.array([[0,0],[0.75, -0.1185],[0.5,-0.134],[1,0]])
13 tri = Delaunay(points) # compute the Delaunay triangulation of the given points
14 # by Delaunay triangulation the convex hull region is determined
15
16 # Make a list of line segments:
17 # edge_points = [ ((x1_1, y1_1), (x2_1, y2_1)),
18 #                 ((x1_2, y1_2), (x2_2, y2_2)),
19 #                 ... ]
20 edge_points = []
21 edges = set()
22
23 def add_edge(i, j):
24     """Add a line between the i-th and j-th points, if not in the list already"""
25     if (i, j) in edges or (j, i) in edges:
26         # already added
27         return
28     edges.add( (i, j) )
29     edge_points.append(points[ [i, j] ])
30
31 for ia, ib in tri.convex_hull:
32     add_edge(ia, ib)
33
34 lines = LineCollection(edge_points)
35 plt.figure()
36 plt.title('Convex hull')
37 plt.gca().add_collection(lines)
38 plt.plot(points[:,0], points[:,1], 'o', hold=1)
39 plt.xlim(0, 1)
40 plt.ylim(-0.2, 0.1)
41 plt.xlabel('Cu fraction')
42 plt.ylabel('H (eV/at)')
43 plt.savefig('cupd-cvxhull.png')
44 plt.show()
45 #-----
46 #-----
47 #---OPTIMIZATION---
48 #-----
49 # create a list which contains the Pd-Cu alloy compositions
50 # list contains only Cu fraction in the alloy with 0.01 increments
51 mylist = [h/100.0 for h in range(101)]
52
53 # define the function to be minimized
54 def func(x):
55     return (0*x[0]-0.1185*x[1]-0.134*x[2]+0*x[3]) # objective function to minimize
56
57 # create a list of initial guesses
58 g = [0.1, 0.2, 0.3, 0.4]
59
60 # bounds on each phase fraction
61 # the lower bound is 0, upper bound is 1, since these are fractions
62 b = [(0,1),(0,1),(0,1),(0,1)]

```

```

63
64 # these lists will keep the final values
65 finalx=[] # of the fractions of the present phases
66 finalhf=[] # minimized formation energy of the given alloy concentration
67
68 # minimize the objective function for each Cu concentration
69 for i in mylist:
70     # b1 and b2 are the constraints on the objective function
71     b1 = lambda x: (x[0]+x[1]+x[2]+x[3]-1)
72     b2 = lambda x: (x[0]+0.75*x[1]+0.5*x[2]+0*x[3]-i)
73     # this line invokes the optimization calculation
74     res= fmin_slsqp(func,g, eqcons=[b1,b2], bounds=b, iprint=0)
75     finalx.append(res)
76     finalhf.append(func(res))
77
78
79 finalx = np.array(finalx)
80
81 print '#+caption: Minimized formation energy and phase separation at each Cu fraction'
82 print '| X_Cu    | H_m_i_n (eV/at) | Cu    | Cu_3Pd | CuPd   | Pd    |'
83 print '|-'
84 for i,h,r in zip(mylist,finalhf,finalx):
85     print '| {0:1.2f} : | {1}    | {2:1.2f} | {3:1.2f} | {4:1.2f} | {5:1.2f} | '.format(i,h,r[0],

```

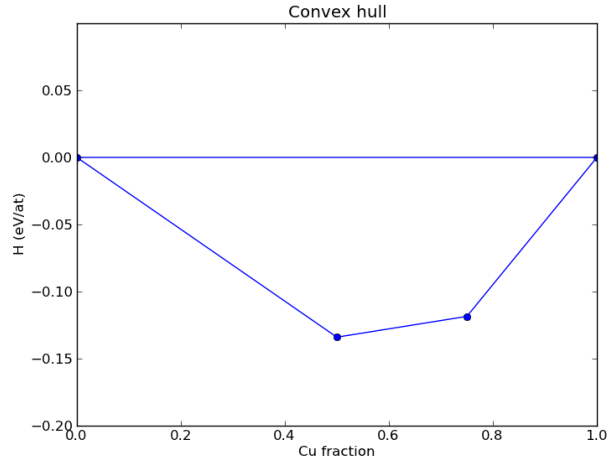


Figure 1: The convex hull for Cu-Pd.

Above in the table, sample calculated results are given, the complete table is provided in the appendix. The results show the minimized value of the energy and the fractions of the existing phases for each Cu composition. For any given composition, separation occurs at most into two phases. The

Table 2: Sample minimized formation energy and phase separation at each Cu fraction

$X_{Cu}$	$H_{min}(eV/at)$	Cu	$Cu_3Pd$	CuPd	Pd
0.00 :	1.31561428418e-17	0.00	-0.00	0.00	1.00
0.10 :	-0.0268	0.00	0.00	0.20	0.80
0.20 :	-0.0536	0.00	0.00	0.40	0.60
0.30 :	-0.0804	0.00	-0.00	0.60	0.40
0.40 :	-0.1072	-0.00	0.00	0.80	0.20
0.90 :	-0.0474	0.60	0.40	0.00	-0.00
1.00 :	6.92224055854e-17	1.00	-0.00	-0.00	0.00

plot which shows the phase separation at each composition, is prepared in Origin and given as the following figure.

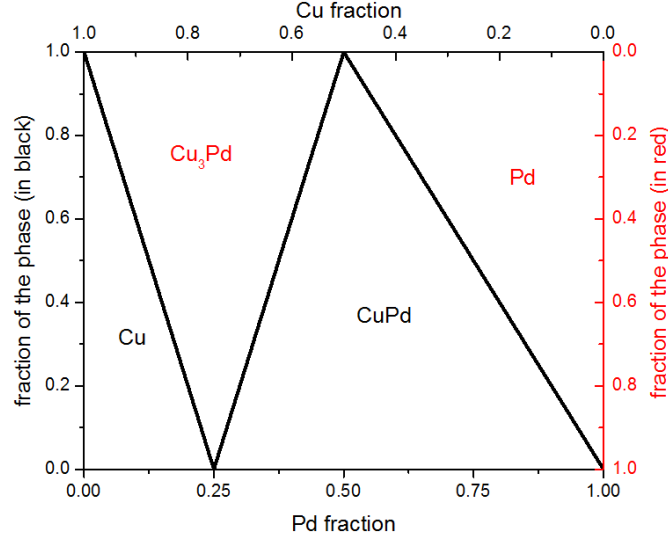


Figure 2: Fraction of the phases present as a function of Pd or Cu composition.

When we want to find the phase separation at 0 K for a given composition what should be done is as follows:

Let's think we are interested in finding the phase separation information for Pd<sub>30</sub>Cu<sub>70</sub> at 0 K. First, draw a line from either 0.3 on Pd fraction axis

or 0.7 on Cu fraction axis till that line intersects one of the lines on the plot. From the intersection point draw lines to the axes on the left (black) and the right (red). The values at which the lines intersect the axes are the fractions of the phases in accordance with the color used to present each phase on the figure. So for  $\text{Pd}_{30}\text{Cu}_{70}$ , only  $\text{Cu}_3\text{Pd}$  (red) and  $\text{CuPd}$  (black) phases will exist and the fraction of the phases will be 0.8 (red) and 0.2 (black), respectively.

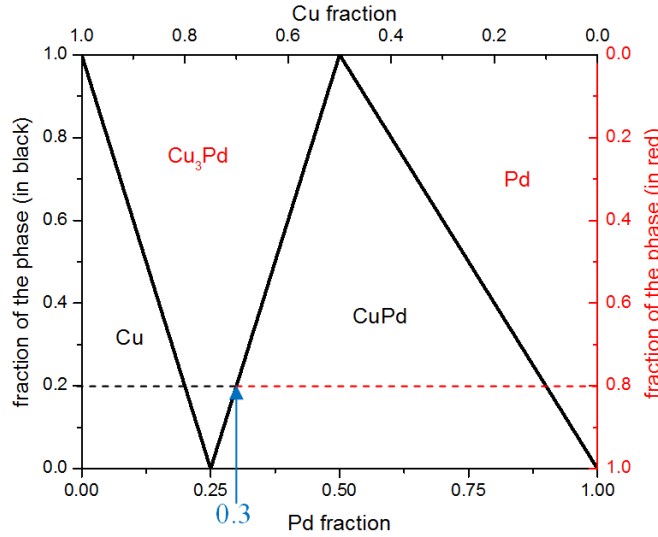


Figure 3: Phase separation for  $\text{Pd}_{30}\text{Cu}_{70}$ .

### 3.1 Use of the computation

It is possible to calculate which phases exist with what fraction at each Cu composition. Once the convex hull is formed for an alloy, the calculated formation energy can be tested if it is within the boundary. If that is the case, then the phase separation can be calculated. If the calculated value is below the boundary of the convex hull, then the hull is recalculated. As an example, let's take  $\text{Cu}_{30}\text{Pd}_{70}$ . The Cu fraction is 0.3. And let's assume that the calculated formation energy from DFT is -0.05 eV/at. First look at its place within the convex hull.

---

```

1 import numpy as np
2 from scipy.spatial import Delaunay      # import the Delaunay to form the convex hull
3 import matplotlib.pyplot as plt
4 from matplotlib.collections import LineCollection

```

```

5
6 # Enter the fraction and the formation energy information to check if it is in convex hull. First number is the Cu
7 # fraction and the second number is the energy of formation calculated at that composition.
8 i=[0.3, -0.02]
9 #-----
10 #---DELAUNAY TRIANGULATION &-----
11 #-----CONVEX HULL-----
12 #-----
13 # points in 2-d, these points now contain the calculated data point in addition to the values taken from Table 1
14 # you can change the last point according to your wish.
15 points=np.array([[0,0],[0.75, -0.1185],[0.5,-0.134],[1,0],i])
16 tri = Delaunay(points) # compute the Delaunay triangulation of the given points
17
18 # Make a list of line segments:
19 # edge_points = [ ((x1_1, y1_1), (x2_1, y2_1)),
20 #                ((x1_2, y1_2), (x2_2, y2_2)),
21 #                ... ]
22 edge_points = []
23 edges = set()
24
25 def add_edge(i, j):
26     """Add a line between the i-th and j-th points, if not in the list already"""
27     if (i, j) in edges or (j, i) in edges:
28         # already added
29         return
30     edges.add( (i, j) )
31     edge_points.append(points[ [i, j] ])
32
33 for ia, ib in tri.convex_hull:
34     add_edge(ia, ib)
35
36 lines = LineCollection(edge_points)
37 plt.figure()
38 plt.title('Convex hull')
39 plt.gca().add_collection(lines)
40 plt.plot(points[:,0], points[:,1], 'o', hold=1)
41 plt.xlim(0, 1)
42 plt.ylim(-0.2, 0.1)
43 plt.xlabel('Cu fraction')
44 plt.ylabel('H (eV/at)')
45 plt.savefig('cu0_3-cvxhull.png')
46 plt.show()

```

---

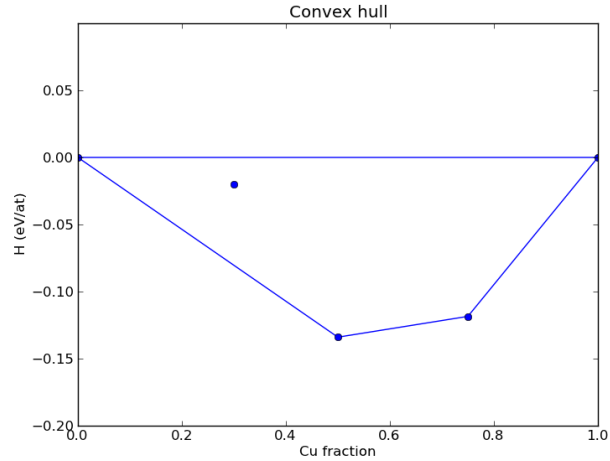


Figure 4: The convex hull for Cu-Pd with the calculated point at 0.3 Cu fraction.

It is seen in the figure that the energy is in the convex hull but not a minimum so further calculation can be done to find the minimized the energy (-0.0804 eV/at) and the fractions of the separation into other phases (0.6 into CuPd and 0.4 into Pd phase).

---

```

1  import numpy as np
2  from scipy.optimize import fmin_slsqp # import the optimization
3  #-----
4  #-----
5  #---OPTIMIZATION---
6  #-----
7
8  # define the function to be minimized
9  def func(x):
10     return (0*x[0]-0.1185*x[1]-0.134*x[2]+0*x[3]) # objective function to minimize
11
12  # create a list of initial guesses
13  g = [0.1, 0.2, 0.3, 0.4]
14
15  # bounds on each phase fraction
16  # the lower bound is 0, upper bound is 1, since these are fractions
17  b = [(0,1),(0,1),(0,1),(0,1)]
18
19  i= 0.3 # enter the Cu fraction here
20
21
22  # b1 and b2 are the constraints on the objective function
23  b1 = lambda x: (x[0]+x[1]+x[2]+x[3]-1)
24  b2 = lambda x: (x[0]+0.75*x[1]+0.5*x[2]+0*x[3]-i)

```



```

25 # this line invokes the optimization calculation
26 res = fmin_slsqp(func,g, eqcons=[b1,b2], bounds=b, iprint=0)
27
28 hf=func(res)
29
30 r = np.array(res)
31
32 print '#+caption: Minimized energy and separated pahses for 30 % Cu alloy'
33 print '| X_C_u | H_m_i_n (eV/at) | Cu | Cu_3Pd | CuPd | Pd |'
34 print '|-----|'
35 print '| {0:1.2f} : | {1} | {2:1.2f} | {3:1.2f} | {4:1.2f} | {5:1.2f} |'.format(i,hf, r[0], r[1], r[2], r[3], r[4], r[5])

```

Table 3: Total energy of  $\text{TiO}_2$  vs. volume.

$X_{Cu}$	$H_{min}(eV/at)$	Cu	$\text{Cu}_3\text{Pd}$	CuPd	Pd
0.30 :	-0.0804	0.00	-0.00	0.60	0.40

If recalculation of the convex hull occurs, this shows that there exists another stable phase. This means that the constraints should be updated with the formation energy and the composition of this stable phase.

## 4 Appendix for results

Table 4: Minimized formation energy and phase separation at each Cu fraction

$X_{Cu}$	$H_{min}(eV/at)$	Cu	$Cu_3Pd$	CuPd	Pd
0.00 :	1.31561428418e-17	0.00	-0.00	0.00	1.00
0.01 :	-0.00268	0.00	-0.00	0.02	0.98
0.02 :	-0.00536	-0.00	0.00	0.04	0.96
0.03 :	-0.00804	-0.00	0.00	0.06	0.94
0.04 :	-0.01072	-0.00	-0.00	0.08	0.92
0.05 :	-0.0134	-0.00	0.00	0.10	0.90
0.06 :	-0.01608	-0.00	-0.00	0.12	0.88
0.07 :	-0.01876	0.00	-0.00	0.14	0.86
0.08 :	-0.02144	0.00	-0.00	0.16	0.84
0.09 :	-0.02412	0.00	0.00	0.18	0.82
0.10 :	-0.0268	0.00	0.00	0.20	0.80
0.11 :	-0.02948	0.00	-0.00	0.22	0.78
0.12 :	-0.03216	-0.00	0.00	0.24	0.76
0.13 :	-0.03484	0.00	-0.00	0.26	0.74
0.14 :	-0.03752	-0.00	-0.00	0.28	0.72
0.15 :	-0.0402	-0.00	0.00	0.30	0.70
0.16 :	-0.04288	0.00	-0.00	0.32	0.68
0.17 :	-0.04556	0.00	-0.00	0.34	0.66
0.18 :	-0.04824	0.00	0.00	0.36	0.64
0.19 :	-0.05092	-0.00	-0.00	0.38	0.62
0.20 :	-0.0536	0.00	0.00	0.40	0.60
0.21 :	-0.05628	-0.00	0.00	0.42	0.58
0.22 :	-0.05896	-0.00	0.00	0.44	0.56
0.23 :	-0.06164	0.00	0.00	0.46	0.54
0.24 :	-0.06432	-0.00	0.00	0.48	0.52
0.25 :	-0.067	0.00	0.00	0.50	0.50

Table 5: Minimized formation energy and phase separation at each Cu fraction (cont'd)

$X_{Cu}$	$H_{min}(eV/at)$	Cu	$Cu_3Pd$	CuPd	Pd
0.26 :	-0.06968	0.00	-0.00	0.52	0.48
0.27 :	-0.07236	0.00	-0.00	0.54	0.46
0.28 :	-0.07504	-0.00	0.00	0.56	0.44
0.29 :	-0.07772	-0.00	0.00	0.58	0.42
0.30 :	-0.0804	0.00	-0.00	0.60	0.40
0.31 :	-0.08308	0.00	-0.00	0.62	0.38
0.32 :	-0.08576	0.00	0.00	0.64	0.36
0.33 :	-0.08844	0.00	-0.00	0.66	0.34
0.34 :	-0.09112	-0.00	-0.00	0.68	0.32
0.35 :	-0.0938	-0.00	0.00	0.70	0.30
0.36 :	-0.09648	-0.00	0.00	0.72	0.28
0.37 :	-0.09916	-0.00	0.00	0.74	0.26
0.38 :	-0.10184	0.00	0.00	0.76	0.24
0.39 :	-0.10452	-0.00	0.00	0.78	0.22
0.40 :	-0.1072	-0.00	0.00	0.80	0.20
0.41 :	-0.10988	-0.00	-0.00	0.82	0.18
0.42 :	-0.11256	0.00	-0.00	0.84	0.16
0.43 :	-0.11524	-0.00	0.00	0.86	0.14
0.44 :	-0.11792	0.00	-0.00	0.88	0.12
0.45 :	-0.1206	0.00	0.00	0.90	0.10
0.46 :	-0.12328	-0.00	0.00	0.92	0.08
0.47 :	-0.12596	0.00	-0.00	0.94	0.06
0.48 :	-0.12864	-0.00	0.00	0.96	0.04
0.49 :	-0.13132	-0.00	0.00	0.98	0.02
0.50 :	-0.134	-0.00	0.00	1.00	-0.00

Table 6: Minimized formation energy and phase separation at each Cu fraction (cont'd)

$X_{Cu}$	$H_{min}(eV/at)$	Cu	$Cu_3Pd$	CuPd	Pd
0.51 :	-0.13338	0.00	0.04	0.96	0.00
0.52 :	-0.13276	0.00	0.08	0.92	-0.00
0.53 :	-0.13214	-0.00	0.12	0.88	-0.00
0.54 :	-0.13152	-0.00	0.16	0.84	-0.00
0.55 :	-0.1309	-0.00	0.20	0.80	-0.00
0.56 :	-0.13028	-0.00	0.24	0.76	-0.00
0.57 :	-0.12966	-0.00	0.28	0.72	0.00
0.58 :	-0.12904	0.00	0.32	0.68	0.00
0.59 :	-0.12842	-0.00	0.36	0.64	-0.00
0.60 :	-0.1278	0.00	0.40	0.60	0.00
0.61 :	-0.12718	0.00	0.44	0.56	0.00
0.62 :	-0.12656	0.00	0.48	0.52	0.00
0.63 :	-0.12594	0.00	0.52	0.48	0.00
0.64 :	-0.12532	0.00	0.56	0.44	-0.00
0.65 :	-0.1247	0.00	0.60	0.40	0.00
0.66 :	-0.12408	0.00	0.64	0.36	0.00
0.67 :	-0.12346	-0.00	0.68	0.32	-0.00
0.68 :	-0.12284	0.00	0.72	0.28	0.00
0.69 :	-0.12222	-0.00	0.76	0.24	-0.00
0.70 :	-0.1216	-0.00	0.80	0.20	-0.00
0.71 :	-0.12098	-0.00	0.84	0.16	-0.00
0.72 :	-0.12036	0.00	0.88	0.12	0.00
0.73 :	-0.11974	-0.00	0.92	0.08	0.00
0.74 :	-0.11912	0.00	0.96	0.04	0.00
0.75 :	-0.1185	-0.00	1.00	0.00	-0.00

Table 7: Minimized formation energy and phase separation at each Cu fraction (cont'd)

$X_{Cu}$	$H_{min}(eV/at)$	Cu	$Cu_3Pd$	CuPd	Pd
0.76 :	-0.11376	0.04	0.96	0.00	0.00
0.77 :	-0.10902	0.08	0.92	-0.00	0.00
0.78 :	-0.10428	0.12	0.88	-0.00	0.00
0.79 :	-0.09954	0.16	0.84	0.00	0.00
0.80 :	-0.0948	0.20	0.80	-0.00	0.00
0.81 :	-0.09006	0.24	0.76	-0.00	0.00
0.82 :	-0.08532	0.28	0.72	0.00	0.00
0.83 :	-0.08058	0.32	0.68	0.00	0.00
0.84 :	-0.07584	0.36	0.64	0.00	0.00
0.85 :	-0.0711	0.40	0.60	-0.00	0.00
0.86 :	-0.06636	0.44	0.56	0.00	0.00
0.87 :	-0.06162	0.48	0.52	0.00	0.00
0.88 :	-0.05688	0.52	0.48	0.00	0.00
0.89 :	-0.05214	0.56	0.44	0.00	-0.00
0.90 :	-0.0474	0.60	0.40	0.00	-0.00
0.91 :	-0.04266	0.64	0.36	0.00	0.00
0.92 :	-0.03792	0.68	0.32	-0.00	0.00
0.93 :	-0.03318	0.72	0.28	-0.00	0.00
0.94 :	-0.02844	0.76	0.24	0.00	-0.00
0.95 :	-0.0237	0.80	0.20	0.00	-0.00
0.96 :	-0.01896	0.84	0.16	-0.00	0.00
0.97 :	-0.01422	0.88	0.12	-0.00	0.00
0.98 :	-0.00948	0.92	0.08	-0.00	0.00
0.99 :	-0.00474	0.96	0.04	-0.00	0.00
1.00 :	6.92224055854e-17	1.00	-0.00	-0.00	0.00