Part I. Data Generation

In this part, we followed the formulas derived by Ce-Wen Nan *et al.,* to generate data. W.l.o.g, we first studied the case of Completely misoriented ellipsoidal particles. Five parameters are regarded in this case: K_p (the thermal conductivity of disperse phase), K_m (the thermal conductivity of matrix phase), a_k (the Kapitza radius, which characterizes the interface thermal property), f (the volume fraction of the disperse phase) and shape (a vector that contains the length of semi-axes of the ellipsoid). Data was generated randomly, with $K_p, K_m \in (0{,}1000)$; $a_k = 0; f \in (0, 1); semiaxis \in (0, 10)$. Since it is not always possible and convenient to represent a shape with its semi-axes vector, the shape of the particle is characterized by sphericity, which is defined as the ratio of the surface area of a sphere with the same volume as the given particle to the surface area of the particle.

$$\varphi = \frac{\pi^{\frac{1}{3}}(6V_p)^{\frac{2}{3}}}{A_p}.$$

However, sphericity is not sufficient to distinct different shapes, because some particularly stretched shape may have the same sphericity as other shape. Take ellipsoid as an example, we set two axes of the ellipsoid have the same length and vary the third axes. In this way, two kinds of ellipsoid can be distinguished as oblate ($a_1 = a_2 > a_3$) and prolate ($a_1 = a_2 < a_3$). In following figure 1, we set $a_1 = a_2 = 1$ and varies a_3 from 0 to 10. The figure indicates that the oblate and prolate may have the same sphericity with different value of a_3.

Figure 2 shows the variation of thermal conductivity with respect to sphericity. It clearly indicates the problem that the with only sphericity to describe the shape, we will not be able to find the bijection between the parameters and the result.

Therefore, we introduced another parameter: p, which is defined as $p = \frac{a_3}{a_1}$. As is presented above, p > 1 means the particle is prolate, while p < 1 means the particle is oblate.

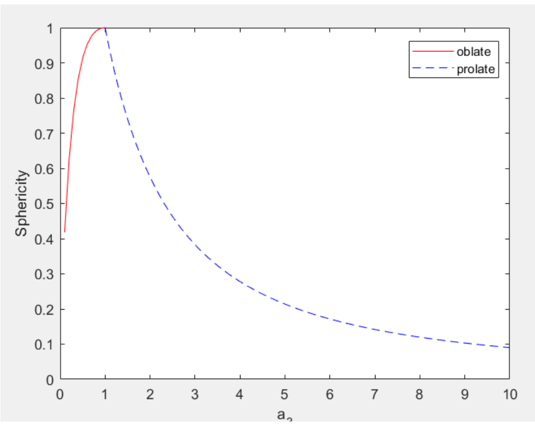


Figure 1

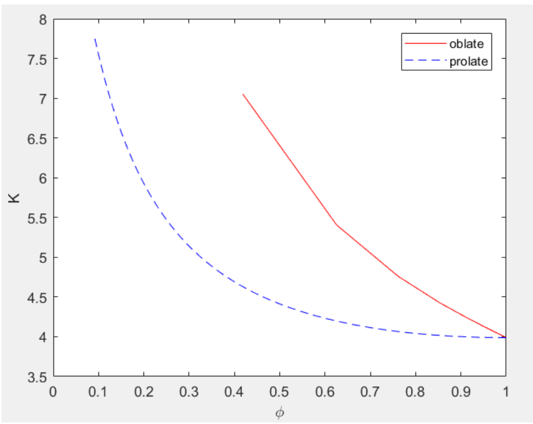


Figure 2

The dataset for training is acquired to this point, with parameters: K_p, K_m, f, φ, and p.

- The code for data generation is attached at the end of this document.

Part II. Model Training

In this part, we have tried three machine learning algorithms: SVR and Neural Network. We used the python library scikit_learn for the learning algorithm.

1. SVR (Support Vector Regression)
   - Number of parameters: 5
   - Number of samples: 100000
   - Model: SVR(kernel='rbf', C=10000, gamma='auto', epsilon=10)
   - With feature normalization
   - Test result:

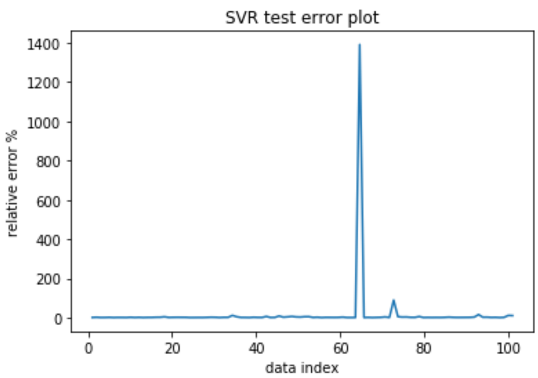


Figure 3

   - Top 5 error samples:

     1: error: 1391.472892

     [479.4632, 0.5224, 0.8985, 0.9333, 0.5467]

     2: error: 89.170988

     [949.3039, 82.0712, 0.0103, 0.0047, 93.0]

     3: error: 15.376204

     [89.9507, 30.8895, 0.4852, 0.2545, 4.3158]

     4: error: 11.246088

     [861.1398, 16.9829, 0.5619, 0.5253, 2.2]

     5: error: 10.974315

     [147.6082, 25.2282, 0.7984, 0.861, 0.4118]

The result shows that most of the samples have pretty good predict result, but several samples with high Kp/Km ratios have great prediction errors. The problem is derived from the reason that there's little points have such extreme value. The machine learning algorithm have failed to fit them. To address this problem, we chose Neural Network, which, with multiple number of neural nodes and layers, can fit relative complex functions. We also restrained the range of K_p and K_m to (0, 100) to make the samples more concentrated.

2. Neural Network
   - Number of parameters: 5
   - Number of samples: 1000000
   - Model:

     ```
     MLPRegressor(hidden_layer_sizes=(100,100,100), alpha = 0, activation='relu', verbose = True, learning_rate_init = 0.001, random_state = random.seed(0), max_iter = 1000)
     ```

   - With feature normalization
   - Result:

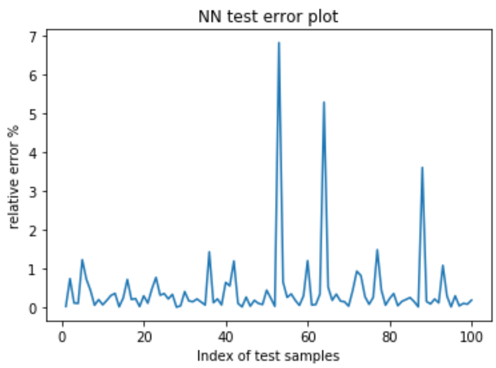


Figure 4

   - Top 5 error samples:

     1: error: 6.831268

     [43.7035, 2.1519, 0.3617, 0.9119, 1.1429]

     2: error: 5.292724

     [43.4833, 2.7735, 0.1217, 0.9848, 0.75]

     3: error: 3.606294

     [79.5504, 3.2859, 0.0448, 0.9957, 0.8571]

     4: error: 1.483032

     [20.4486, 24.1336, 0.908, 1.0, 1.0]

     5: error: 1.428724

     [27.3519, 3.3344, 0.2084, 0.4628, 2.5]

The result shows that although there's still some errors, they are within 10% and is considered acceptable.

Part III. Testing

In this part, we tested our training model with results from finite element software COMSOL. We built a model with a cube of 4mm long at each side as matrix, and 64 particles of specific shapes dispersed evenly into the cube. We applied a surface heating source of 30000W/m^2 and a sink of heat conductance 300000W/(m^{2}K). Before reaching this model, we have tried different size, heat source and heat sink conduction, and found that the influence of these parameters are small.

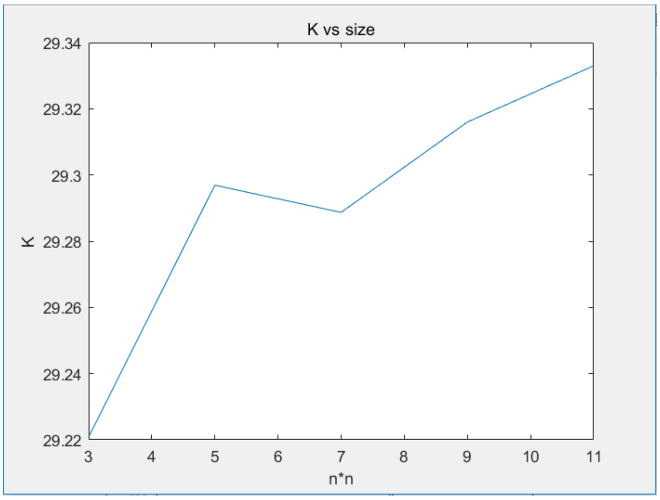


Figure 5

As the figure 3 presents, the overall thermal conductivity K does not vary much with respect to the size of the cube. Also different heating source and sink conduction would lead to the similar result. Then we tested how does the results of Nan's formula matches the results of COMSOL. We found that they match pretty well and the deviation is within the 10% model error, which is closed to the error of Machine learning model.

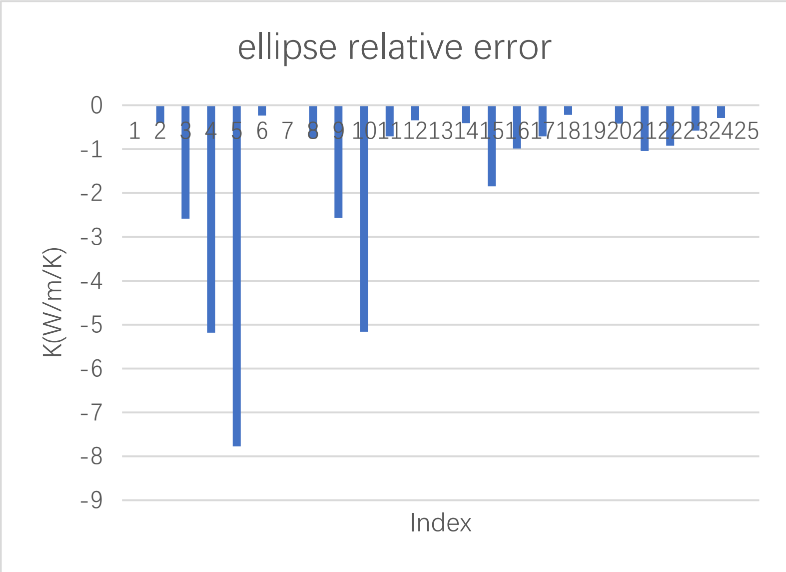


Figure 6

To this point, we have verified that three methods: Analytical Formula by Effective Medium Theory, Machine Learning Method and Finite Element Method match each other, despite the acceptable error.

Next, we tested whether Nan's formula can be applied to other geometrical shapes other than ellipses. To better distinguish and study the effect of various shapes, we chose three special shapes: cube, donut and cone. Intuitively, they are quite different from the ellipse, and we tried to examine whether the machine learning method is also applicable in these cases. The results are closed if the requirements are low as assumed.

Since the analytical formula given by Ce-Wen Nan *et al.* is only valid for the ellipsoid case, we tried to verify our model by comparing the results from Finite element method and Machine Learning model.

The results turn out to be satisfied. By applying different sphericity and p values to the model, the results fit with the Finite element method quite well.

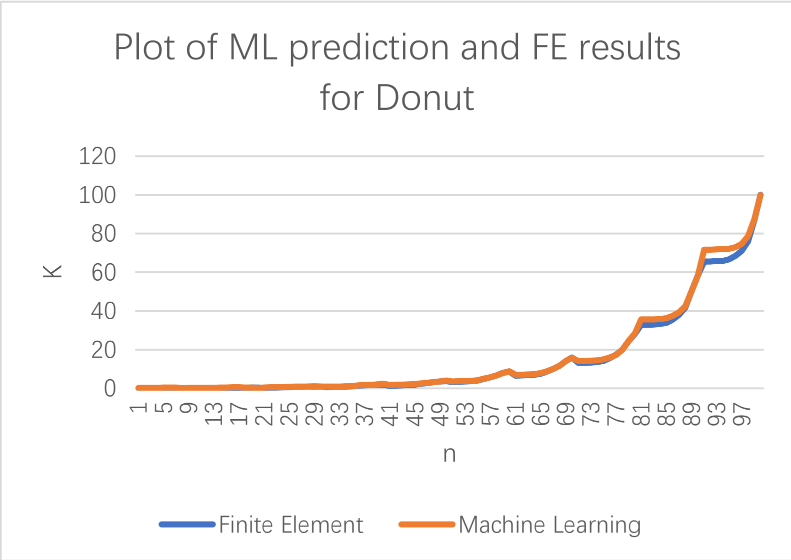


Figure 7

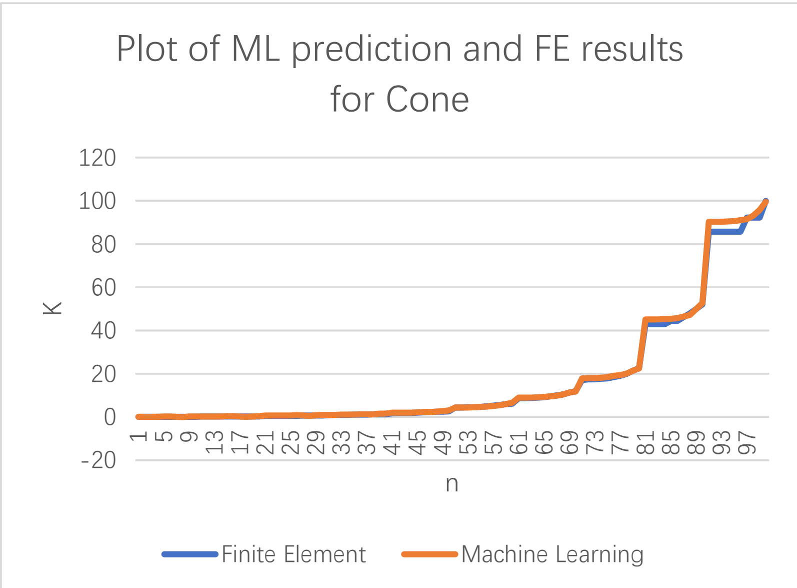


Figure 8

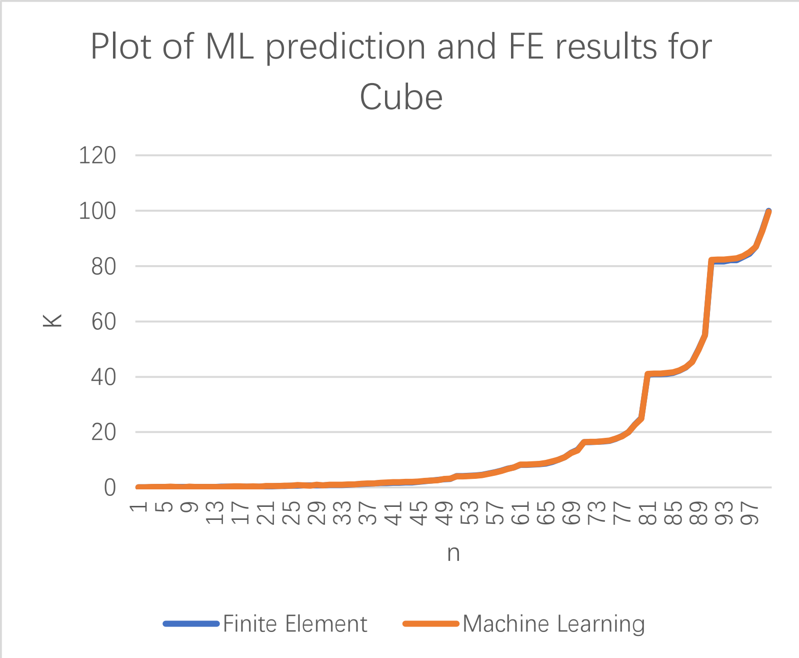


Figure 9

Note: The above plots can be further improved by replace the x-axis with some meaningful variable, such as Kp, Km.

However, there is something worth notice. By comparing the results of various shape from the finite element method, we found that the effect of shape to the conductivity is not quite crucial. On the other hand, the dimension ratio p has a relative significant effect on the composite material conductivity. Thus we re-trained our machine learning model with only four parameters: K_p, K_m, f and p.

- Parameters: K_p, K_m, f and p.
- Number of samples: 1000000
- Model:

  MLPRegressor(hidden_layer_sizes=(100,100,100), alpha = 0, activation='relu', verbose = True, learning_rate_init = 0.001, random_state = random.seed(0), max_iter = 1000)
- With feature normalization
- Result:

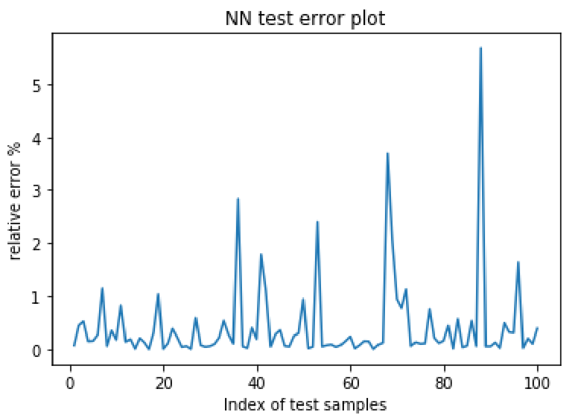


Figure 10

The result fit quite well the analytical formula. Therefore, with only four parameters, we can get a model which predicts thermal conductivity well.

With this standing, it seemed valid to apply Nan's analytical formula to various shapes. We compared the results of analytical formula and finite element method.

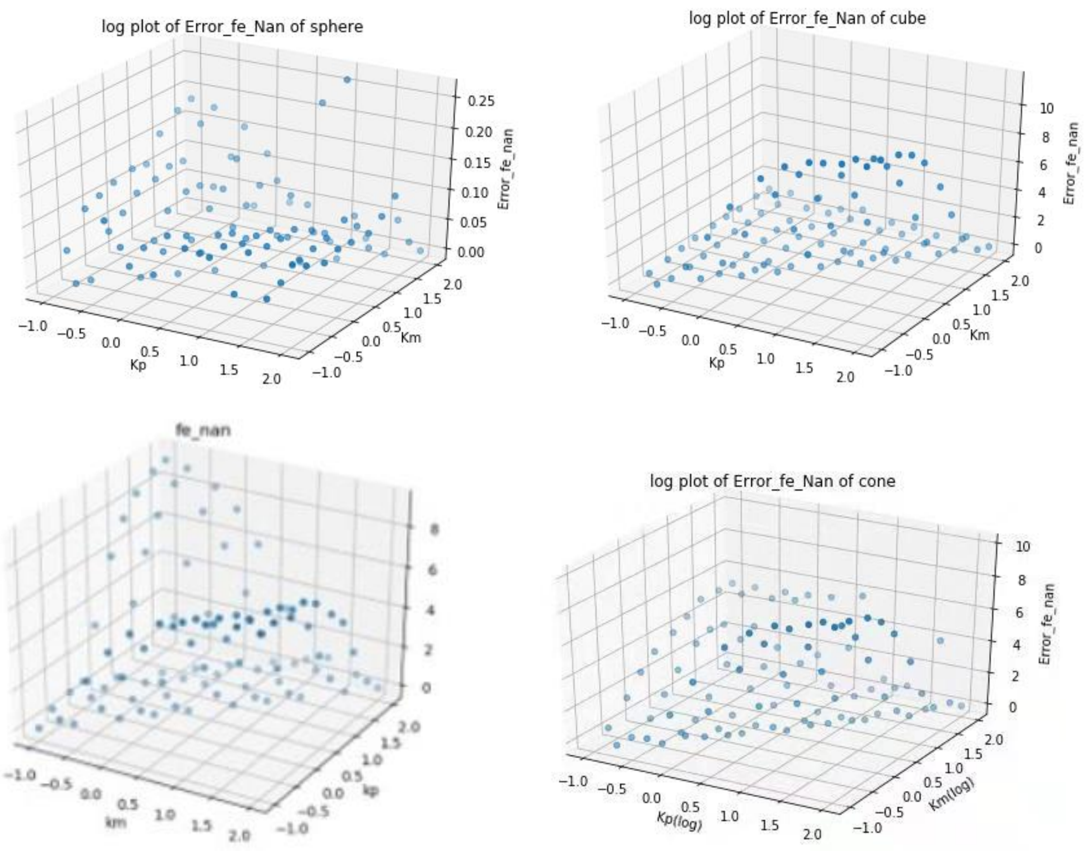
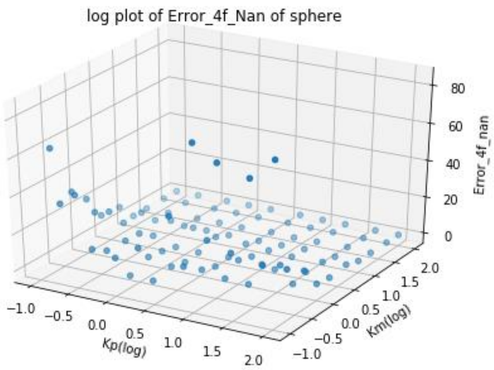


Figure 11

The plots above show that the relative errors between finite element method and analytical formula are quite low (within 10%). Thus, the analytical formula derived for ellipsoid is applicable to other shapes. There is something worth more comment: it can be found that there are lines in the above plots where abrupt increment of error can be observed.

They are the lines where $K_p = K_m$. The observation indicates that when K_m is much smaller than K_p, the error of analytical formula will increase.

Then we took a few steps further to verify the applicability of machine learning in thermal conductivity prediction. Since Machine Learning model collapse in some extreme cases. We first test extreme big K_m, K_p, and extreme small K_m, K_p cases.



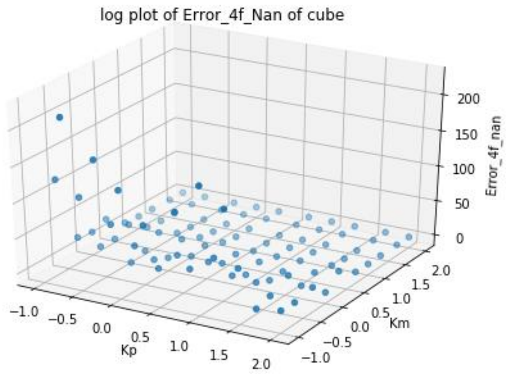


Figure 12

The above plots shows that when K_m is extremely small (< 1) the machine learning model gives extremely high errors. However, when K_p is extremely small, while K_m is greater than 10, the machine learning model is still applicable.

Then we find the up and bottom bound of the results by aligning the particles in a line with specific angles to the x or y axis. We find some interesting observations. The up and bottom bound of the results happen when the particles direction are parallel or normal to the x or y axis.

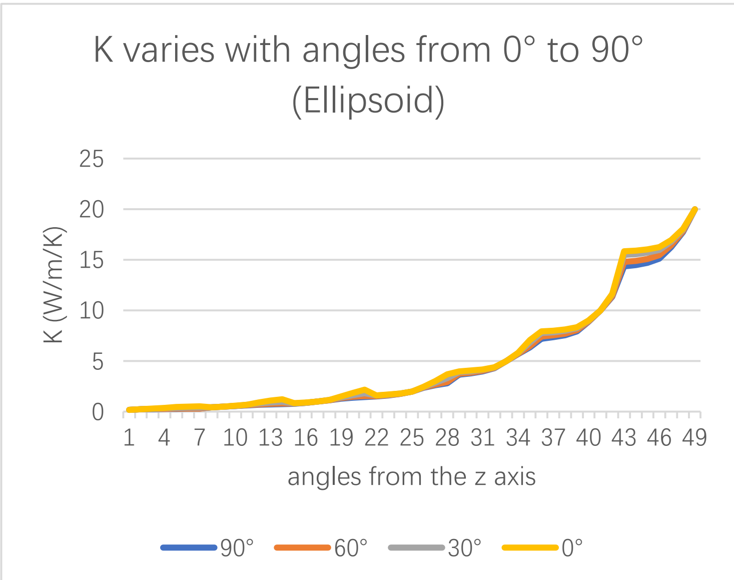


Figure 13

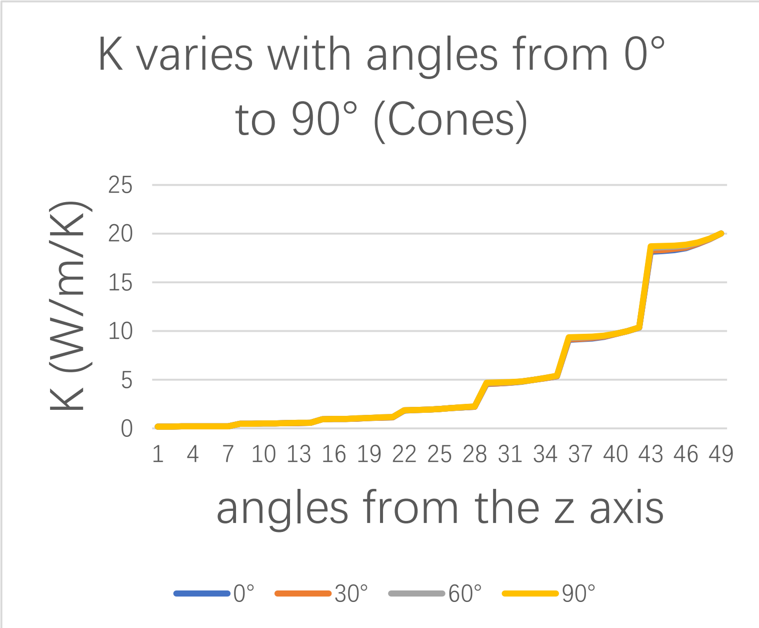


Figure 14

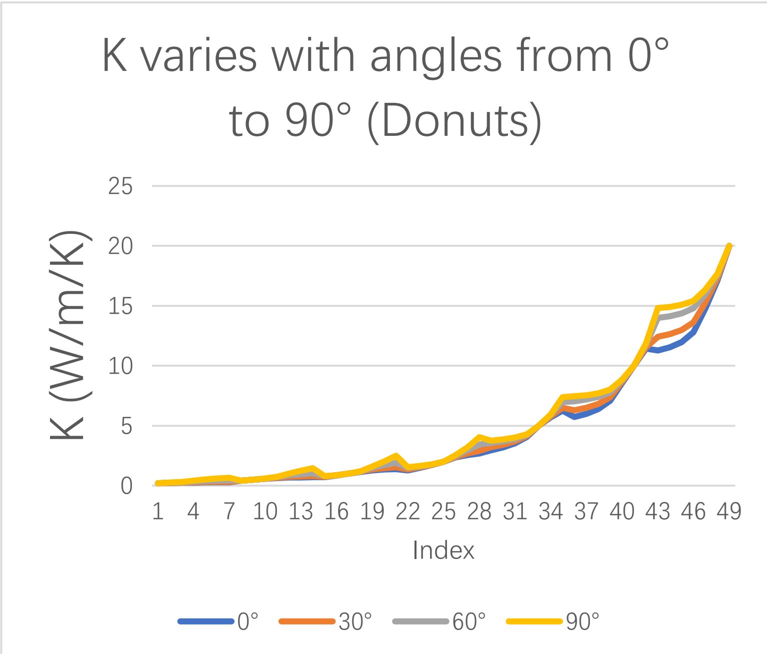


Figure 15

As the figure13 shows that, for the case of ellipsoid, the up bound occurs when all the particles line parallel to the z axis, the bottom bound occurs when all the particles align perpendicularly to the z axis. However, the figure14 and figure15 show that, for the cases of cone and donut, the up bound occurs when all the particles align perpendicularly to the z axis, while the bottom bound occurs when the particles are parallel to the z axis.

If the particles are misoriented ellipsoid, both the results of COMSOL and Nan's formula lie at the middle of the up and bottom bound. However, in donut and cone cases, the result of Nan's formula does not match the result of COMSOL. The band gap between the up and bottom bound is pressed and shifts upwards.

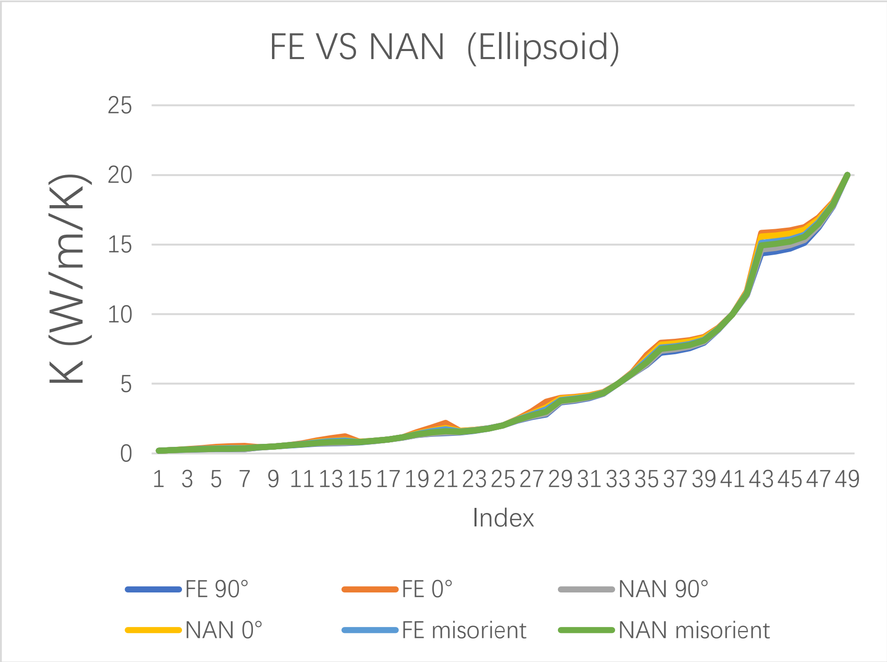


Figure 16

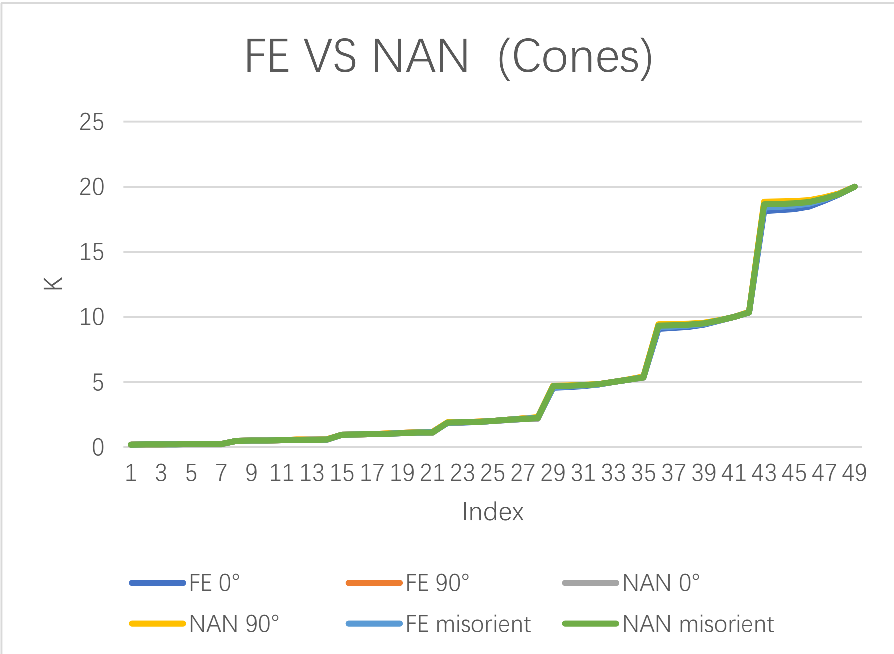


Figure 17

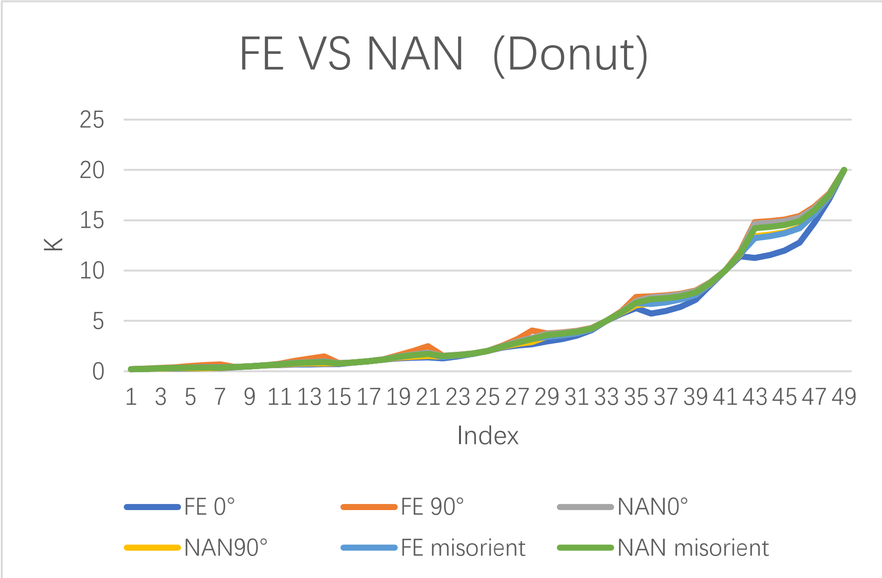


Figure 18

As it can be seen clearly in the Donut graph, the bandgap is large in FE cases while small in Nan's case. Also, relative to the FE case, the band is shifted upword. This may due to the unsuitable p definition. We need to find another suitable definition of p in cone and donut case so that the band can match the FE results.

Part IV. Current Contribution

At current stage, we have showed that it is possible to apply the Nan's analytical formula in the calculation of thermal conductivity of other particle shapes. The specific shape of the particle is not crucial; however, it is the volume fraction f and dimension ratio p that contribute to the variation of thermal conductivity when K_m and K_p are fixed.

Part V. Future Expectation

The current vague definition of dimension ratio p is the main barrier for us to obtain a unified equation. We expected that with p being defined rigorously, the analytical formula can be applied to calculate the thermal conductivity as accurate as the finite element method.

The method we figured out is trying to extend the definition for p in the ellipsoid case. For each case we have tested, i.e. cone, donut and cube, try to find a p that can fit the formula with the finite element results. Then, we can collect these p-definition to see if there's any relationship among them.

Finally, the machine learning method should be better involved in the following work to find the relationship between the old analytical model and the new model with dimension ratio included as a factor.