Measurements of Anisotropic Thermal Conductivity of a Multilayer Ceramic Substrate Over a Uniform Spatial Distribution

Lillie Clark, Tom Marinis, and David Hanson

Charles Stark Draper Laboratory
555 Technology Square
Cambridge, MA 02139-3563
Tel: (617) 258-3479   E-mail: tmarinis@draper.com

Abstract

The thermal conductivity of multilayer packages and printed wiring boards is often modeled as a volumetric weighting of thermal conductivities of the component materials. We demonstrate that this approach overestimates effective thermal conductivity, especially normal to the plane of the package. The reason is that the interfacial heat flux between metal and dielectric layers is significantly less than that within a metal layer. As electronic packaging becomes more intricate and complex, thermal performance analyses and predictions must be more accurate than ever for the industry to progress. In comparison to the ceramic packages we subsequently analyze, printed circuit board layups typically include conductive and dielectric materials that have a higher differential of thermal conductivity, more materials, thinner layers, vias, or embedded components. Each aspect further complicates the analysis and increases the likelihood of error in overestimating the resultant thermal conductivity. We developed an experimental procedure that confirmed our prediction of overestimation by measuring the thermal conductivity in x, y and z directions on 5 millimeter square segments cut from a 31 millimeter square multilayer ceramic package. The data we collected was used to refine a thermal model of our package, and to validate finite element thermal models of the 5 millimeter square substrate pieces. We are proceeding to use parametric finite element models to compute the thermal performance of a large number of hypothetical package segments. The computed thermal conductivity vectors from these models will be combined with attributes of each segment, such as metal fraction, number of layers, conductor line length, etc. to create a data set that can be used to train an inference engine to predict the thermal performance of an advanced package design from its artwork data.

Key Words:
Thermal Conductivity, Co-Fired Ceramic , Machine Learning

Introduction

Thermal conductivity of a material is a crucial aspect of electromechanical design due to its effect on reliability and performance of the part and higher assembly. Often the thermal conductivity of a composite comprised of metal traces on dielectric layers is approximated by a volumetric weighting of the thermal conductivities of the component materials.

Gori and Corasaniti calculated the anisotropic behavior of thermal conductivities of fiber-reinforced composites along two axes [1]. Pietrak and Wiśniewski inquired into the accuracy of numerical approximations for interfacial thermal resistances of several particle shapes [2]. Terentyeva, Perera and Narendran compared theoretical predictions to experimental data for three methods of calculating the thermal conductivities for composites of various filler particle sizes and volume fractions [3]. Wang developed an analytical solution for the case of thin metal strips on the surface of a substrate. It is noteworthy that his solution is applicable even for a staggered array of strips [4]. Procter and Solc compared the thermal conductivity predicted by the Nielsen Model with experimental measurements of composites with various filler particles [5]. Each of these papers have identified the inaccuracies and underestimations of numerical methods used for predicting thermal conductivity of multi-material substrates. Thus, it is necessary to compare simulated results to experimental data to confirm the thermal conductivity of the composite ceramic substrate.

In-Plane Heat Transport

Case I – Parallel Conduction Paths

A section of a substrate layer is shown in Figure 1. Its dimensions are: $x = H,\ y = W$, and $z = d$. It contains three equally spaced metal conductors of width w_m.

If the temperature of the substrate face at $x = 0$ is maintained at a temperature T_1 and that at $x = H$ a temperature T_2, then the flow of heat along the x direction, J_x is given by

$$J_x = -\left(\frac{T_2 - T_1}{H}\right)\left[k_m \sum w_m + k_c\left(W - \sum w_m\right)\right]d \quad (1)$$

In this expression, k_m and k_c, represent the thermal conductivities of the metal and

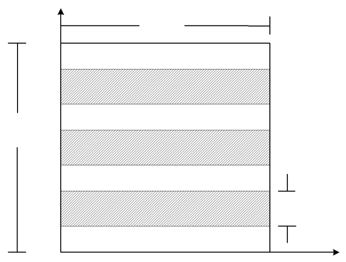


Figure 1 Composite substrate with heat flow parallel to x axis

ceramic materials, respectively. The heat flux in the x direction, j_x is obtained by dividing J_x by the cross-sectional area Wd to obtain

$$j_x = -\left(\frac{T_2 - T_1}{HW}\right)\left[k_m \sum w_m + k_c\left(W - \sum w_m\right)\right] \quad (2)$$

The heat flux can also be expressed in terms of an effective thermal conductivity, k_e of the substrate

$$j_x = -\left(\frac{T_2 - T_1}{H}\right)k_e. \quad (3)$$

By equating 2 and 3, the following expression for k_e is obtained

$$k_e = \frac{k_m \sum w_m + k_c\left(W - \sum w_m\right)}{W}. \quad (4)$$

Multiplication of numerator and denominator by Hd transforms 4 to

$$k_e = \frac{k_m V_{m+} k_c V_c}{V}$$
(5)

in which V_m and V_c are the metal and ceramic volumes and $V = V_m + V_c$ is the substrate volume. By letting f_m and f_c represent the metal and ceramic volume fractions, respectively, 5 reduces to

$$k_e = k_m f_m + k_c f_c.$$
(6)

Case II – Series Conduction Path

Figure 2 shows a composite substrate with three metal stripes of equal width that are arranged so that there are also three ceramic stripes of uniform length. The faces at $y = 0$ and $y = W$ are fixed at temperatures T_1 and T_2, to induce a heat flux, J_j along y, perpendicular to the metal traces.

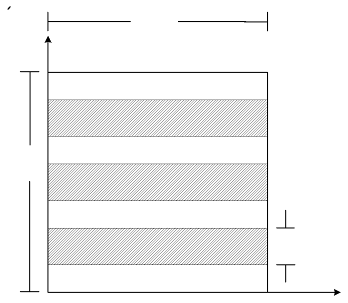


Figure 2 Composite substrate with heat flow parallel to y axis

The flow of heat through the substrate in the y direction is given by

$$J_y = -Hd\left(\frac{T_2 - T_1}{W}\right)k_e.$$
(7)

This flow is equal to the flow of heat through each strip of metal and ceramic such that

$$J_y = -Hd\frac{\Delta T_m}{w_m}k_m,$$
(8)

$$J_y = -Hd\frac{\Delta T_c}{w_c}k_c,$$
(9)

in which ΔT_m and ΔT_c are the respective temperature drops across the metal and ceramic strips. By equating the sum of 8 and 9 to two times 7 we obtain

$$-2Hd\left(\frac{T_2 - T_1}{W}\right)k_e = -Hd\frac{\Delta T_m}{w_m}k_m - Hd\frac{\Delta T_c}{w_c}k_c.$$
(10)

Elimination of common terms yields

$$2\left(\frac{T_2 - T_1}{W}\right)k_e = \frac{\Delta T_m}{w_m}k_m + \frac{\Delta T_c}{w_c}k_c.$$
(11)

If there are n metal and ceramic strips, 11 can be written as

$$2\left(\frac{n(\Delta T_m + \Delta T_c)}{n(w_m + w_c)}\right)k_e = \frac{\Delta T_m}{w_m}k_m + \frac{\Delta T_c}{w_c}k_c,$$
(12)

which by rearranging terms becomes

$$2(\Delta T_m + \Delta T_c)k_e = \left(\frac{w_m + w_c}{w_m}\right)\Delta T_m k_m + \left(\frac{w_m + w_c}{w_c}\right)\Delta T_c k_c.$$
(13)

By letting

$$f_m = \frac{w_m}{w_m + w_c} \text{ and } f_c = \frac{w_c}{w_m + w_c}$$
(14)

Equation 13 can be rewritten as

$$2(\Delta T_m + \Delta T_c)k_e = \frac{k_m}{f_m}\Delta T_m + \frac{k_c}{f_c}\Delta T_c,$$
(15)

which yields the following expression for the effective thermal conductivity

$$k_e = \frac{1}{2}\left[\frac{k_m}{f_m}\frac{\Delta T_m}{(\Delta T_m + \Delta T_c)} + \frac{k_c}{f_c}\frac{\Delta T_c}{(\Delta T_m + \Delta T_c)}\right].$$
(16)

Rewriting the temperature fractions, 16 becomes

$$k_e = \frac{1}{2}\left[\frac{k_m}{f_m}\frac{1}{\left(1+\frac{\Delta T_c}{\Delta T_m}\right)} + \frac{k_c}{f_c}\frac{1}{\left(\frac{\Delta T_m}{\Delta T_c}+1\right)}\right].$$
(17)

By equating equations 8 and 9, we can write

$$\frac{\Delta T_c}{\Delta T_m} = \frac{k_m}{k_c}\frac{w_c}{w_m}.$$
(18)

Substitution of this result into 17 yields

$$k_e = \frac{1}{2}\left[\frac{k_m}{f_m}\frac{1}{\left(1+\frac{k_m}{k_c}\frac{w_c}{w_m}\right)} + \frac{k_c}{f_c}\frac{1}{\left(\frac{k_c}{k_m}\frac{w_m}{w_c}+1\right)}\right].$$
(19)

Multiplying the volume fractions in the denominator yields

$$k_e = \frac{1}{2}\left[\frac{k_m}{\left(f_m+f_c\frac{k_m}{k_c}\right)} + \frac{k_c}{\left(f_m\frac{k_c}{k_m}+f_c\right)}\right].$$
(20)

Letting $k_m = \alpha k_c$, 20 can be written as

$$\frac{k_e}{k_c} = \frac{1}{2}\left[\frac{\alpha}{(f_m+f_c\alpha)} + \frac{1}{\left(f_m\frac{1}{\alpha}+f_c\right)}\right]. \qquad (21)$$

Making this same substitution for k_m in equation 6 yields the following expression for k_e when heat flow is parallel to the metal stripes

$$\frac{k_e}{k_c} = \alpha f_m + f_c.$$
(22)

The values of $\frac{k_e}{k_c}$ given by equations 21 and 22 are plotted against the metal volume fraction in Figures 3 and 4, for values of $\alpha = 2, 5$ and 10.

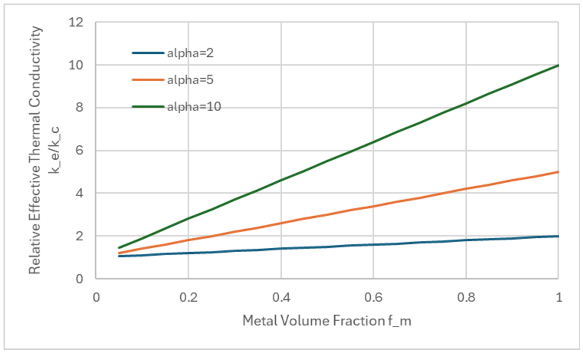


Figure 3 Heat flow parallel to metal stripes

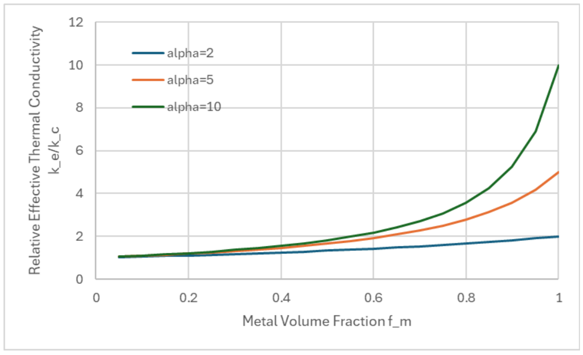


Figure 4 Heat flow normal to metal stripes

Experimental Setup

A wire saw was used to cut a multilayer ceramic substrate into 5 mm square samples with nominal thickness 2.9 mm. Each sample was unique in composition due to the pad geometry and internal trace locations within the substrate; 3 of the 25 samples were chosen for testing. Sample 1 was chosen due to its lack of surface geometry. Sample 2 was chosen for the full footprint of micro-BGA pads spanning the top side, and sample 3 was chosen for the larger pad geometry.

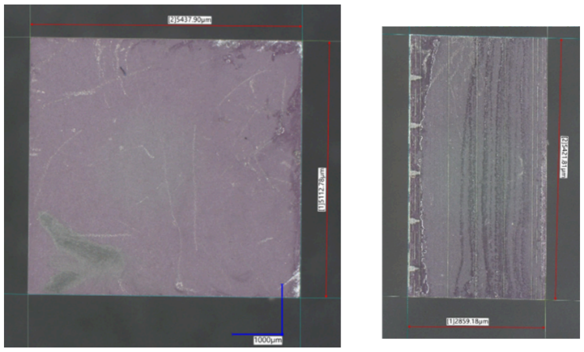


Figure 5 Sample 1, top view (left) and side view (right)

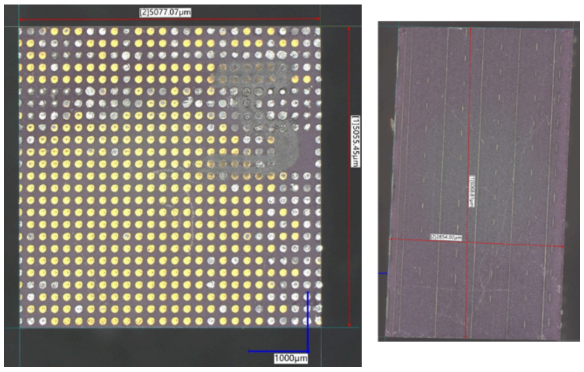


Figure 6 Sample 2 top view (left) and side view (right)

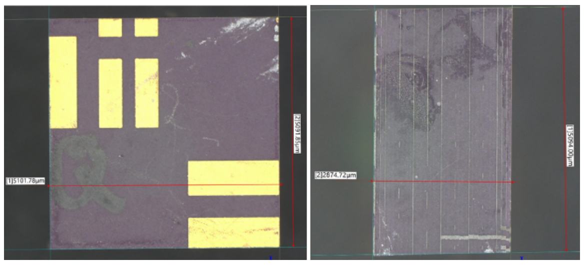


Figure 7 Sample 3 top view (left) side view (right)

A schematic diagram of the experimental setup appears in Figure 8 and pictures of the hardware are shown in Figures 9 and 10. To reduce unintentional convective and radiative heat, the experiment was performed in a pressure-monitored vacuum chamber. The sample was seated between two aluminum blocks containing thermocouples embedded near the top and bottom surfaces, with an additional thermocouple near the heat sink. A heater was wired to a power supply and an in-series resistor to monitor the current into the system. The stack up was mechanically secured between two plates by threaded rods. The thermocouple blocks and heater sleeve were machined from aluminum, and the heat sink was a large piece of steel.

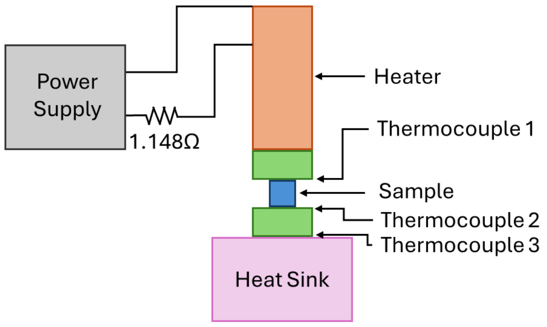


Figure 8. Experiment Schematic Diagram

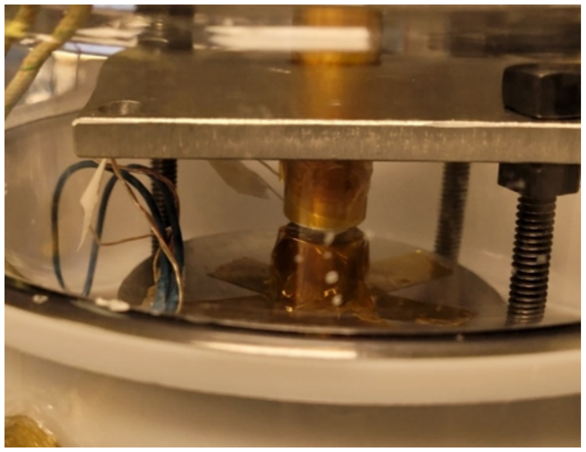


Figure 9 Picture of unit under test looking through sidewall of vacuum chamber

The data was recorded using a Dataq Instruments acquisition unit and WinDaq software. The data was exported to excel for processing after each run. The experiment was first run with a 99% alumina sample to identify the calibration number for the experimental setup, which is calculated next.

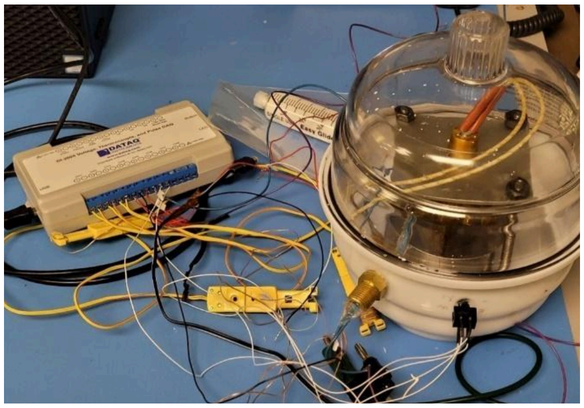


Figure 10. Data Acquisition Unit (left) and vacuum chamber setup (right)

Calculation of Calibration Number

The heat transfer of the overall system, J_{PS}, can be defined as the sum of the heat fluxes through the fixture and alumina sample in the Z-direction, $j_{A,z}$

$$J_{PS} = J_F + j_{A,z}$$
(23)

The heat flux through the fixture, J_F, is the sum of the fluxes through its components

$$J_F \;=\; -\,\Delta T_{2,1}\left[\sum\left(k_F \frac{a_F}{t_F}\right)\right]$$
(24)

in which k_F is thermal conductivity, a_F is cross sectional area and t_F is thickness.

The calibration number, η_O, is defined as the effect of the test fixture on the system, wherein each expression is a property of the fixture itself

$$\eta_O = \sum\left(k_F \frac{a_F}{t_F}\right) \qquad .$$
(25)

The expression for J_F simplifies to

$$J_F \;=\; -\,\Delta T_{2,1}\left[\eta_O\right].$$
(26)

The heat flux in the z direction, $j_{A,z}$, can be expressed in terms of the thermal conductivity, $k_{A,z}$, cross sectional area, $a_{A,z}$, and thickness, $t_{A,z}$ of the alumina sample

$$j_{A,z} \;=\; -\,\Delta T_{2,1} k_{A,z}\left[\frac{a_{A,z}}{t_{A,z}}\right].$$
(27)

The heat flux through the system can also be expressed in terms of the voltage V_{PS} and current, I_{PS}, applied to the system heater

$$J_{PS} = V_{PS} I_{PS}.$$
(28)

By replacing the terms from equations for J_F and $j_{A,z}$ into J_{PS} overall, equation 23 becomes

$$V_{PS} I_{PS} \;=\; -\,\Delta T_{2,1}\left[\eta_O\right] - \Delta T_{2,1} k_{A,z}\left[\frac{a_{A,z}}{t_{A,z}}\right].$$
(29)

Solving for η_O, the following expression for the calibration number is obtained

$$\eta_O \;=\; -\frac{V_{PS} I_{PS}}{\Delta T_{2,1}} - k_{A,z}\left[\frac{a_{A,z}}{t_{A,z}}\right].$$
(30)

The sample used for calibration was 5mm square by 3.75mm thick, with a minimum datasheet conductivity value of 25 $\frac{W}{mK}$, and maximum value of 46 $\frac{W}{mK}$. Replacing these known values of the alumina sample after normalization yields minimum and maximum values of η_O, which were calculated to be -0.117 $[\frac{W}{K}]$ and 0.02 $[\frac{W}{K}]$ respectively. However, since η_0 is associated with the flow of heat through a path that bypasses the sample under test, its value must always be positive. Setting $\eta_0 = 0$, and solving for $k_{A,z}$ yields an upper boundary of 28.38 $\frac{W}{mK}$ for the thermal conductivity of our alumina sample.

Calculation of k from test data

Using the same formula for J_{PS}, j_S can be defined as the heat flux through the substrate sample of unknown thermal conductivity along the test axis,

$$J_{PS} = J_F + j_S \,. \qquad (31)$$

Rewriting the fluxes in terms of components yields

$$V_{PS}I_{PS} = -\Delta T_{2,1}[\eta_0] - \Delta T_{2,1}k_S\left[\frac{a_S}{t_S}\right] \qquad (32)$$

and solving for k_S yields

$$k_S = -[\frac{t_s}{a_s}][\frac{V_{PS}I_{PS}}{\Delta T_{2,1}} + \eta_0] \,. \qquad (33)$$

Equation 34 was input into excel and the results are plotted below. The graphs depict the calculated values of thermal conductivity over time as the sample heats up and reaches equilibrium. The values for thermal conductivity in Table 3 are calculated from the data after it has reached equilibrium.

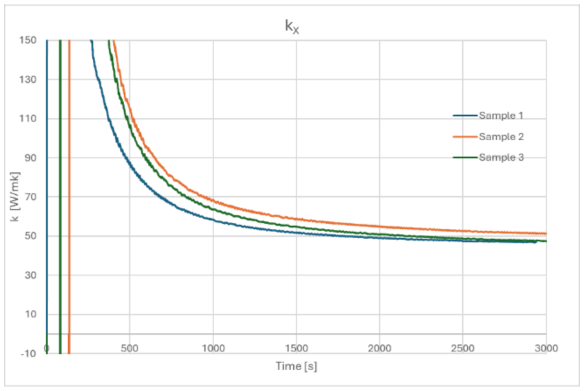


Figure 11 Thermal conductivity along x axis over time, using η_0 *= 0.02 [W/K]*

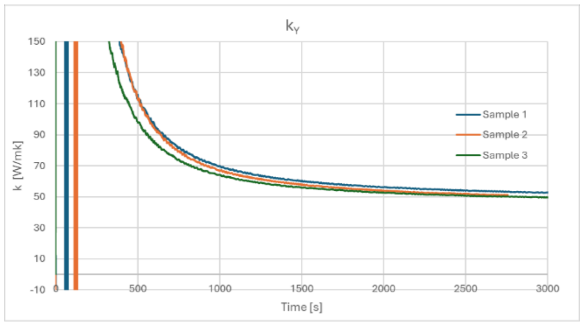


Figure 12. Thermal Conductivity along Y axis over time, using η_0 *= 0.02 [W/K]*

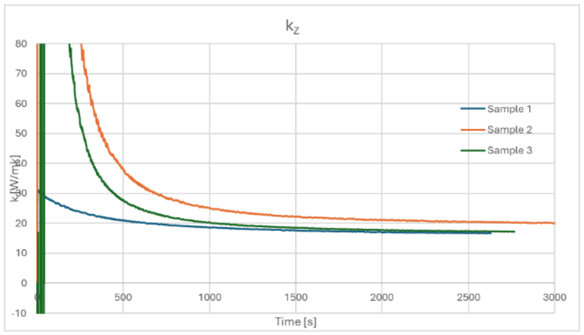


Figure 13. Thermal Conductivity along Z axis over time, using η_0 *= 0.02 [W/K]*

Table 1. Measured thermal conductivity values of substrate samples

ηo [W/K]	Axis	Sample K [W/mK]			Std Dev
		1	2	3	

0.02	X	43.65	49.81	43.80	15.13
	Y	49.62	48.58	46.72	14.03
	Z	16.53	20.83	17.30	4.86
0.00	X	50.99	57.59	51.58	15.17
	Y	57.85	56.35	54.50	14.05
	Z	18.91	23.34	19.82	4.89

The multilayer substrate is fabricated from Kyocera's proprietary ceramic material A440, which their design guide lists as having a thermal conductivity value of 14 [*W/mK*] [6].

Estimation of thermal conductivity from substrate artwork

Previous calculations for in-plane heat transport showed that effective thermal conductivity is impacted not only by metal area fraction, but by the metal pattern as well. Measurements of thermal conductivity along three orthogonal axes of multilayer substrate samples cut from a ceramic package demonstrated the validity of the calculations. The experimental measurement methodology would be prohibitively expensive and time consuming for routine application to large substrate designs. This motivated our investigation into the use of machine learning to predict thermal conductivity from substrate artwork.

Neural network methodology

Development of neural network optimization methodology has been ongoing for several decades, with a good historical overview of algorithm development written by, Ananthaswamy [6]. In Figure 14, we present an interconnect diagram for a linear neural network comprised of eight input nodes, labeled $x1$ through $x8$, two hidden layers, with ten neurons each, and a single neuron output layer. Only a portion of the neurons and their interconnect are shown in the figure for the sake of clarity.

Each input to the network is applied with a weighting factor, $w_1^{i,j}$, to every neuron in the first layer. A unique bias factor, b_1^{j}, is also applied to each neuron. The superscripts, i and j, denote, respectively, the numbers of the input node and the neuron to which the input is applied. The expression for the combined input , z_1^{1}, to the first neuron in the first layer is,

$$z_1^{1} = w_1^{11}x1 + w_1^{21}x2 + w_1^{31}x3 + \cdots + \qquad w_1^{71}x$$
. (34)

The subscript 1 in this expression identifies the weights and bias terms as being applied to layer 1. Similar equations can be written for the other nine neurons in the first layer or alternatively,

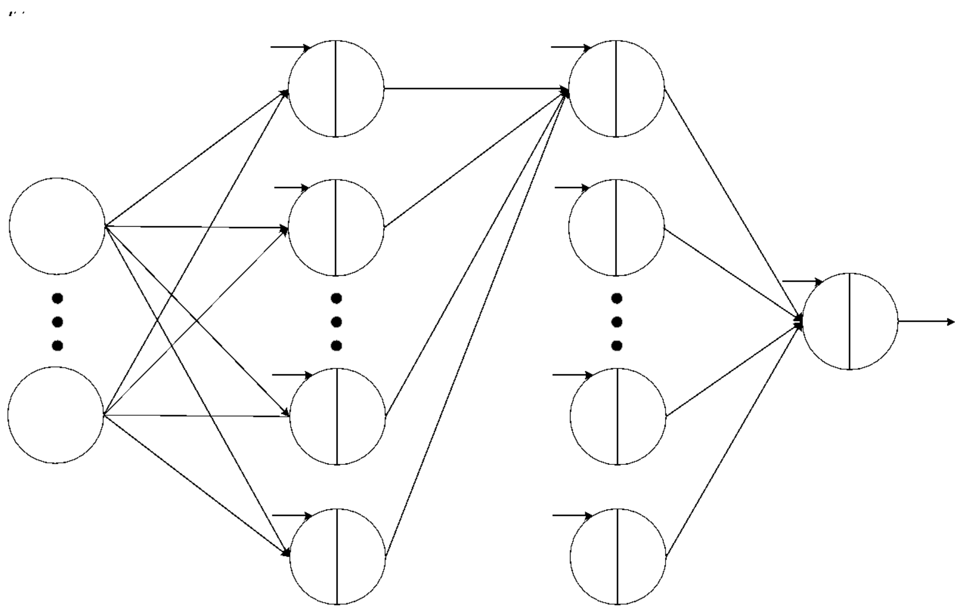


Figure 14 Connection diagram for a neural network with eight inputs, two hidden layers with ten neurons each, and a single neuron output. Only a portion of the nodes and connections are shown for the sake of clarity.

all ten equations can be expressed as a single vector equation

$$\underline{Z_1} = [W_1]^T \underline{X} + \underline{b_1}$$

$($ *SEQ Equation* \ * *ARABIC* 35)

in which $[W_1]^T$ is the transpose of the matrix

$$[W_1] = \left| \frac{w_1^{11}}{w_1^{21} \vdots w_1^{71}\ w_1^{81}} \frac{w_1^{12}}{w_1^{22} \vdots w_1^{72}\ w_1^{82}} \frac{\cdots}{\cdots\ \cdots\cdots} \frac{w}{w_1^{29} \vdots w} \right.$$

.

The combined inputs, $\underline{Z_1}$, to the neurons in the first layer are transformed to outputs, $\underline{a_1}$, by a differentiable function, σ, such that

$$\underline{a_1} = \sigma(Z_1) \qquad ($$

SEQ Equation \ * *ARABIC* 36)

One possible form for σ is

$$\sigma(Z_1) = \frac{1}{1 + e^{-z}}.$$

(37)

The inputs, $\underline{Z_2}$ to the neurons in the second layer are given by

$$\underline{Z_2} = [W_2]^T \underline{X} + \underline{b_2}$$

(38)

in which $[W_2]$, unlike $[W_1]$, is a square matrix with ten columns and rows of weighting factors. The output, $\underline{a_2}$ of the second layer is

$$\underline{a_2} = \sigma(Z_2) \qquad ($$

SEQ Equation \ * *ARABIC* 39)

which is applied to the single neuron of the third layer of the network, such that

$$Z_3 = [W_3]^T \underline{a_2} + b_3$$

(40)

in which Z_3 and b_3 are scalars and $\left[W_3\right]$ is single column array

$$\underline{W_3} = \left| w_3^{11}\ w_3^{21}\ \vdots\ w_3^{91}\ w_3^{10,1} \right| .$$

The output, $\hat{y}_1$, of the network is

$$\hat{y}_1 = a_3 = \sigma(z_3) . \quad (41)$$

In this neural network, there are 211 unknown factors, 80 weights and 10 bias terms in the first layer, 100 weights and 10 bias terms in the second layer and 10 weights and one bias term in the third layer. The process for learning the optimum values for the 211 weights and bias terms proceeds as follows. First, a random number generator is used to populate the weight matrices and bias vectors are set to zero. Then a single case is randomly selected from the set of training data. The input values for the selected case are applied to the input nodes to compute a thermal conductivity estimate $\hat{y}$. An error estimate, e, is computed by taking the difference between $\hat{y}$ and the and the actual thermal conductivity value y . This error value is squared to generate the loss function L, which is used to generate new values for the weights and bias values. For example, the updated estimate of the weight w_1^{11} after training step i is

$$\left(w_1^{11}\right)_{i+1} = \left(w_1^{11}\right)_i + \alpha\frac{\partial L}{\partial w_1^{11}} \quad (42)$$

in which α is a learning rate constant, which typically has a value of 0.01 and $\frac{\partial L}{\partial w_1^{11}}$ is the partial derivative of L with respect to w_1^{11}.

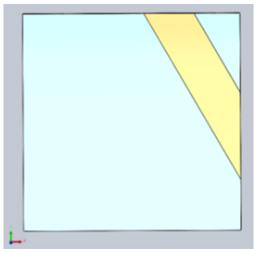
The partial derivative $\frac{\partial L}{\partial w_1^{11}}$ is computed by multiplying the partial derivatives of the network elements to arrive at the partial derivative of L with respect to w_1^{11} via the chain rule of calculus, in a process called Back Propagation. The name derives from the fact that all the partial derivatives are calculated in the forward propagation calculation of $\hat{y}$. As an example,

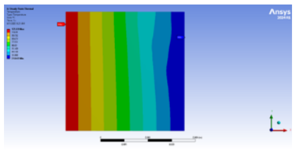
$$\frac{\partial L}{\partial w_1^{11}} = \frac{\partial L}{\partial e}\frac{\partial e}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial z_3}\frac{\partial z_3}{\partial a_2^1}\frac{\partial a_2^1}{\partial z_2^1}\frac{\partial z_2^1}{\partial a_1^1}\frac{\partial a_1^1}{\partial z_1^1}\frac{\partial z_1^1}{\partial w_1^{11}} . \quad (43)$$

In this process, the current weight and bias values must be retained until all the new weight and bias terms are computed. After these terms are updated, the process repeats with another randomly selected case. This learning phase of the computation continues until the value of the loss function attains a sufficiently low value. At this point the weight and bias values of the neural network are tested by applying inputs from a set of test cases that were not used in the training. If the loss function values for these test cases are low, then the neural network is ready for use. Otherwise, training resumes with more randomly selected training cases.

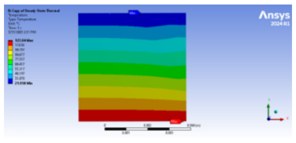
Generation of Training Data

Training data was created by using a solid modeling tool to create a metallization pattern that was laid on a much smaller square substrate as an assembly. Many different substrate metallization patterns were then generated by rotating or shifting the large metallization pattern and trimming it to the size of the substrate. High thermal conductivity strips were added to the four sides of the substrate so that a uniform constant temperature boundary condition could be imposed on the metallization and substrate of any side. The thermal conductivity of these models was computed by using a finite element program to calculate the steady state heat flux between pairs of parallel faces held at different temperatures, with perfect insulator boundary conditions imposed on the other two faces. Two heat flux values, j_x and j_y were computed for each model. In Figures 15 through 17, we present some representative images of metallization patterns and temperature distributions, with their computed heat flux values.


a

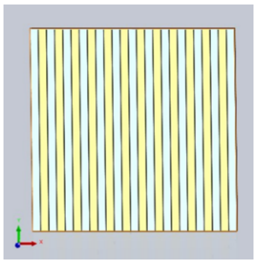

b

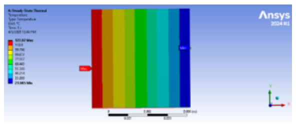
J_x = 2.8330e5 [W m^{-2}]


c

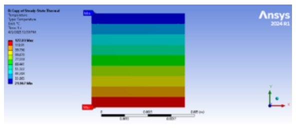
J_y = 2.8685e5 [W m^{-2}]

Figure 15 (a) Single conductor stripe (b), (c) Temperature gradient along x and y


a

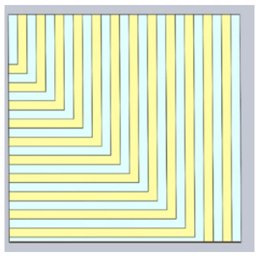

b

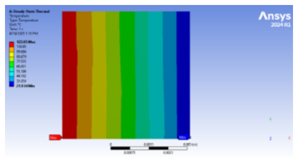
J_x = 3.0009e5 [W m^{-2}]


c

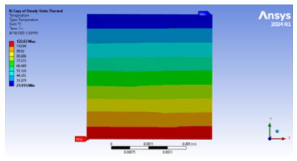
J_y = 3.6870e5 [W m^{-2}]

Figure 16 (a) Substrate with 12 conductors (b), (c) Temperature gradients along x and y


a


b

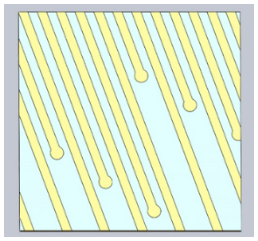
J_x = 2.563e5 [W m^{-3}]

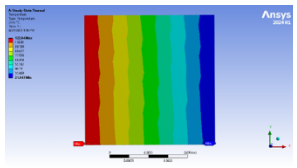

c

Jy = 2.575e5 [W m^{-3}]

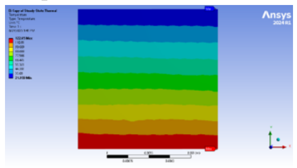
Figure 17 (a) Substrate with 12 conductors that contain right angle jogs (b), (c) Temperature gradients along x and y

The computed heat fluxes, j_i, were


a


b

J_x = 2.7004e5 [W m^{-3}]


c

Jy = 2.7935e5 [W m^{-3}]

Figure 18 (a) Conductors containing via terminations (b), (c) Temperature gradients along x and y

converted to thermal conductivities, k_i, using Fick's First Law

$$k_i = j_i\left(\frac{L_{sub}}{\Delta T}\right), \qquad (44)$$

in which, L_{sub}, is the edge length of the substrate and, ΔT, is the temperature difference between parallel substrate side faces. Eight attributes of the metallization pattern were associated with each computed thermal conductivity value. They were: (1) metallization area, A_m, (2) total conductor line length, l_c, (3) average angle between conductor lines and direction of heat flux, θ_{lj}, (4) average conductor line width, w_c, (5) number of jogs i.e. changes in conductor line orientation, N_{jog}, (6) average jog angle, $\emptyset_{jog}$, (7) number of line terminations at vias, M_T, and (8) average line spacing, h_s. The conductor line length, l_c , was estimated from the perimeter values, p_i, that the solid modeling program calculated for each of the N_c, conductor segments, such that

$$l_c = \frac{1}{2}\sum_{i=1}^{N_c}\left(p_i - 2w_i\right).$$
(45)

Average conductor spacing, h_s, was estimated as,

$$h_s = \frac{\left(L_{sub}^2 - A_m\right)}{l_c}.$$
(46)

The efficiency and robustness of the optimization process is enhanced by normalizing the quantitative measures of the metallization attributes. The reason is that the parameter weighting factors are adjusted based on the derivative of the target value with respect to the attribute values. These derivatives should be of comparable orders of magnitude to avoid biasing the gradient decent optimization path. The normalized metallization attributes and their scaling formulas are listed in Table 2.

Table 2 Normalized Metallization Attributes

Attribute			Formula
#	Symbo l	Description	
1	α	Metal area fraction	$\alpha = \frac{A_m}{\left(L_{sub}\right)^2}$
2	η	Normalized conductor length	$\eta = \frac{l_c}{4L_{sub}}$
3	γ	Normalized angle between conductors and heat flux	$\gamma = \frac{\theta_{lj}}{\left(\frac{\pi}{2}\right)}$
4	ω	Normalized conductor width	$\omega = \frac{2w_c}{L_{sub}}$
5	χ	Jogs per unit conductor length	$\chi = \frac{N_{jog}}{l_c}$
6	ϕ	Normalized jog angle	$\phi = \frac{\emptyset_{jog}}{\left(\frac{\pi}{2}\right)}$
7	ζ	Terminations/ vias per unit line length	$\zeta = \frac{M_T}{l_c}$
8	λ	Normalized line spacing	$\lambda = \frac{2h_s}{L_{sub}}$

Program Description

The machine learning program used to implement a neural network for predicting thermal conductivity from features of a metallization pattern was written by James McCaffrey. He published a detailed description of the program in a two-part article that appeared in Visual Studio Magazine [7] [8]. The source code as well as two data sets, one for training and another for testing, are available on the Visual Studio website [9]. The original application of the program was to predict a person's income based on attributes such as age, sex, state of residence, and political persuasion, which underscores the fact that a neural network is a versatile tool for predicting a number based on a weighted set of input numbers.

The program is written in the Python programming language and uses functions from the NumPy and PyTorch libraries to accomplish the complex tasks [10] [11] [12]. The first step in the program execution is to create the dataset objects which it does by using the NumPy function, loadtext() to read the plain text data files into an eight-column array of metal pattern attributes and a single column array of thermal conductivity values. Two different files are processed to create training and test dataset objects. The NumPy arrays are then converted to tensor objects by calling the PyTorch function, Tensor().

After loading the dataset objects, the program creates the neural network by calling the PyTorch function, nn.Linear() to create an 8 by 10 input layer, a 10 by 10 hidden layer, and a 10 by 1 output layer. The layers are initialized with random weight values by calling the PyTorch function, nn.xavier_uniform(). All network bias values are set to zero by calling the PyTorch function, nn.init.zeros(). These two functions are called three times, once for each layer in the network.

Training the neural network is the compute intensive portion of the program execution. The PyTorch function, util.data.DataLoader() randomly selects a set of training data which is processed by calling the PyTorch functions, nn.MSELoss and optim.Adam() to compute the loss between actual and predicted conductivity values and adjust the weight and bias value of the neural network. After all data in the batch is processed, the program evaluates the accuracy of the thermal conductivity estimates for each case to show how quickly the neural network is learning from the training data. Then the set of test data is processed in the same way, but without making any adjustments to weight or bias values. The average of the loss values for all cases in the test data is computed to give a sense of the predicted thermal conductivity accuracy. Then the program estimates the thermal conductivity for a single case that is not included in either the training or testing data sets.

The final step in the program is to call the PyTorch function, save() to write the weight and bias values to a file. The trained model can be restored later, by the function call load_state_dict(Torch.load(<file>)).

Results

The neural network regression program was run in training mode with different sets of data. For data generated by variations on the patterns shown in Figure 19, the results after 1000 passes through the data are given in Figure 20. For this set of data, The loss function drops from an initial value of 528 to 3, with an accuracy of 0.9.

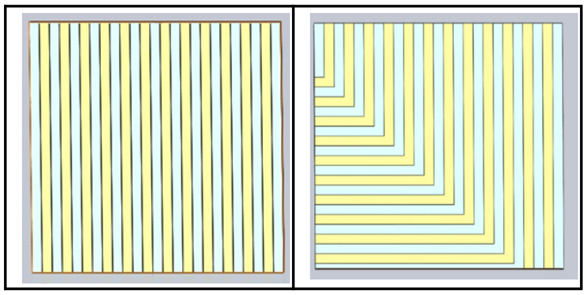


Figure 19 Metallization patterns used to generate initial training dataset

```
Begin Thermal Conductivity Prediction
Creating Thermal Conductivity Dataset objects
Creating 8-(10-10)-1 neural network
```

```
bat_size = 10
loss = MSELoss()
optimizer = Adam
lrn_rate = 0.01
Starting training
epoch =    0  |  loss = 528.2427
epoch =  100  |  loss = 5.6345
epoch =  200  |  loss = 5.6246
epoch =  300  |  loss = 5.5267
epoch =  400  |  loss = 5.1695
epoch =  500  |  loss = 5.0004
epoch =  600  |  loss = 4.3413
epoch =  700  |  loss = 3.6431
epoch =  800  |  loss = 3.0952
epoch =  900  |  loss = 2.9976
Done
Computing model accuracy (within 0.10 of true)
Accuracy on train data = 0.9000
```

Figure 20 Results obtained using the dataset generated by the patterns shown in Figure 19

Adding to the dataset, data generated by the patterns shown in Figure 21, yields the output displayed in Figure 22.

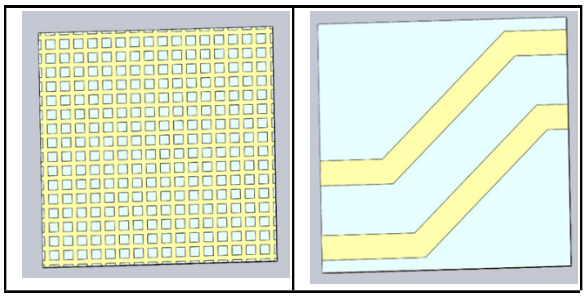


Figure 21 Patterns used to generate additional data for the training dataset

With the expanded dataset, the loss function decreases from 800 to 4 after 1000 passes through the training data and the accuracy is 0.8958.

```
python3 thermal_1.py
Begin Thermal Conductivity Prediction
Creating Thermal Conductivity Dataset objects
Creating 8-(10-10)-1 neural network
bat_size = 10
loss = MSELoss()
optimizer = Adam
lrn_rate = 0.01
Starting training
epoch =    0  |  loss = 800.8136
epoch =  100  |  loss = 7.7311
epoch =  200  |  loss = 7.1173
epoch =  300  |  loss = 7.7795
epoch =  400  |  loss = 6.8060
epoch =  500  |  loss = 5.9627
epoch =  600  |  loss = 5.0783
epoch =  700  |  loss = 4.6052
epoch =  800  |  loss = 4.4057
epoch =  900  |  loss = 4.0972
Done
Computing model accuracy (within 0.10 of true)
Accuracy on train data = 0.8958
End thermal_1
```

Figure 22 Results obtained with the expanded dataset

Conclusions

By developing expressions for thermal conductivity of substrates with metal stripes aligned with and normal to the heat flux, we demonstrated that the metallization pattern can significantly impact the flow of heat. The impact of the metallization pattern on heat flow increases as the difference in the thermal conductivities of the substrate and metallization increase.

Guided by the paper of Terentyeva, et. al., we constructed a simple apparatus to measure the thermal conductivity along three axes of 5 mm squares cut from a large multilayer cofired substrate [3]. Consistent with our analytical calculations, we found large differences in the thermal conductivities measured normal to and in the plane of the substrate. While this measurement technique is useful for validating calculations of thermal conductivities, the effort required to prepare samples and make the measurements, precludes its use to routinely measure all segments of a large substrate.

Ideally, it would be preferred to have a map of the thermal conductivity of a substrate prior to its fabrication for use in thermal studies to assess the performance of a system. While finite element analysis of models constructed from substrate artwork could be used to generate thermal conductivity maps, the calculations often take several days to run on a workstation.

An alternative approach that we explored was to calculate thermal conductivities for 100 small tiles with simple metallization patterns. The patterns were characterized in terms of eight attributes, that included metal area fraction, conductor line length, average line length, etc. This data was used to train a three-layer neural network to predict the thermal conductivity of a tile given its metallization attributes. We used a regression program written by McCaffrey, based on the Python libraries PyTorch and Numpy [7].

When we used a subset of data that only included similar patterns, the program achieved an accuracy of 90% after 1000 passes through the training data. Expanding the training data to include all patterns, such as dense and sparse conductor lines, ground planes, and conductors with via terminations, resulted in a slight reduction in accuracy of 89.6%.

References

[1] F. Gori and S. Corasaniti, "Effective Thermal Conductivity of Composites," *International Journal of Heat and Mass Transfer,* vol. 77, pp. 653-661, 2014.

[2] K. Pietrak and T. S. Wisniewski, "A Review of Models for Effective Thermal Conductivity of Composite Materials," *Journal of Power Technologies,* vol. 95, no. 1, pp. 14-24, 2015.

[3] V. Terentyeva, I. U. Perera and N. Narendran, "Analyzing Theoretical Models for Predicting Thermal Conductivity of Composite Materials for LED Heat Sink Applications," in *Proceedings of IES 2017 Annual Conference*, Portland Oregan, August 10-12, 2017.

[4] C. Y. Wang, "Thermal Conductivity of a Material Containing a layer of Thin Strips or a Staggered Array of Thin Strips," *Transactions of the ASME,* vol. 121, pp. 174-177, 1999.

[5] P. Proctor and J. Solc, "Improved Thermal Conductivity in Microelectronic Encapsulants," *IEEE Transactions on Components, Hybrids and Manufacturing Technology,* vol. 14, no. 4, pp. 708-713, 1991.

[6] A. Ananthaswamy, Why Machines Learn: The Elegant Math Behind Modern AI, New York: Dutton, 2024.

[7] J. McCaffrey, "Regression Using PyTorch, Part 1: New Best Practices," 1 November 2022. [Online]. Available: https://visualstudiomagazine.com/articles/2022/11/01/pytorch-regression.aspx . [Accessed 17 June 2025].

[8] J. McCaffrey, "Regression Using PyTorch New Best Practices, Part 2: Training, Accuracy, Predictions," 14 November 2022. [Online]. Available: https://visualstudiomagazine.com/articles/2022/11/14/pytorch-regression-2.aspx . [Accessed 17 June 2025].

[9] J. McCaffrey, "people_income.zip," [Online]. Available: https://visualstudiomagazine.com/CodeDownload/2022/11/people_income.zip. [Accessed 19 August 2025].

[10] Python Software Foundation, "python," [Online]. Available: https://www.python.org. [Accessed 19 August 2025].

[11] NumPy Steering Council, "NumPy," [Online]. Available: https://numpy.org. [Accessed 19 August 2025].

[12] PyTorch Foundation, "PyTorch," [Online]. Available: https://pytorch.org/. [Accessed 19 August 2025].