Module 4C - Estimation and Optimization of the Effective Properties of Food Mixtures

Omar Betancourt, Payton Goodrich, Emre Mengi

July 24, 2021

Contents

Objectives: To demonstrate understanding of genetic algorithms by applying it to a real-world food system.
Prerequisite Knowledge: N/A
Prerequisite Modules: 1A - Calculus, 1B - Linear Algebra, 1D - Differential Equations, 2B - Continuum Mechanics, 3C - Generic Time Stepping, 4A - Genetic Algorithms, 4B - Gradient-Based Optimization
Summary: In this module, you will learn how to model food as a two-phase heterogenous slurry, estimate material properties of the heterogeous slurry, and how to optimize the material properties by tuning the ratio of solid-phase to liquid-phase food particles using a genetic algorithm.

1 Theory

1.1 Summary

This module was written in a manner to build understanding of the basic principles to utilize mixture-materials comprised of particles and a binding matrix. There are seven pieces to the overall code:

- **COMPONENT 1**: An expression that predicts the overall effective electrical properties of a mixture of particles and matrix.
- **COMPONENT 2**: An expression that predicts the overall effective thermal properties of a mixture of particles and matrix.
- **COMPONENT 3**: An expression that predicts the overall effective mechanical properties of a mixture of particles and matrix.
- **COMPONENT 4**: An expression that predicts the distribution of the electrical load carried by the particles and matrix.
- **COMPONENT 5**: An expression that predicts the distribution of the thermal load carried by the particles and matrix.
- **COMPONENT 6**: An expression that predicts the distribution of the mechanical load carried by the particles and matrix.
- **COMPONENT 7**: A Machine-Learning/Genetic Algorithm that optimizes the the combinations of matrix and particles to deliver a desired overall (macro) response while controlling the (micro) load distributions and the internal heating of the material

In this project, we introduce estimates on the effective properties of particle-functionalized materials. One of the primary properties of interest is the overall "effective" electrical conductivity:

$$\langle \boldsymbol{J} \rangle_\Omega = \boldsymbol{\sigma}^{c,*} \cdot \langle \boldsymbol{E} \rangle_\Omega, \tag{1.1}$$

is sought, where

$$\langle \cdot \rangle_\Omega \overset{\text{def}}{=} \frac{1}{|\Omega|} \int_\Omega \cdot \, d\Omega \,, \tag{1.2}$$

and where $\boldsymbol{\sigma}^{c,*}$ is the effective conductivity for the mixture, $\langle \boldsymbol{E} \rangle_\Omega$ is the volume averaged electric field, $\langle \boldsymbol{J} \rangle_\Omega$ is the volume averaged current, the averaging operator is defined as $\langle \cdot \rangle_\Omega \overset{\text{def}}{=} \frac{1}{|\Omega|} \int_\Omega (\cdot) \, d\Omega$ over a statistically representative volume element with domain Ω. When the materials is isotropic, $\boldsymbol{\sigma}^{c,*} = \sigma^{c,*} \mathbf{1}$. Other properties that we will study are:

- Overall mechanical stiffness:

$$\langle \boldsymbol{\sigma} \rangle_\Omega = \mathbb{E}^* : \langle \boldsymbol{\epsilon} \rangle_\Omega, \tag{1.3}$$

  is sought where, throughout the structure, the mechanical properties of microheterogeneous materials are characterized by a spatially variable elasticity tensor $\mathbb{E} = \mathbb{E}(\boldsymbol{x})$ and $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$ are the stress and

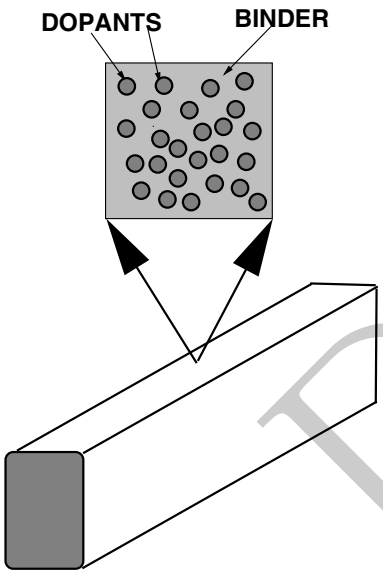

Figure 1.1: A functionalized material for electromagnetic applications.

strain tensor fields within a Representative Volume Element (RVE) of volume $|\Omega|$. The quantity $\mathbb{E}^*$ is known as the effective property. We will concentrate on isotropic materials. In the case of isotropic overall responses, we may write

$$\langle \frac{tr\boldsymbol{\sigma}}{3} \rangle_\Omega = 3\kappa^* \langle \frac{tr\boldsymbol{\epsilon}}{3} \rangle_\Omega \tag{1.4}$$

and

$$\langle \boldsymbol{\sigma}' \rangle_\Omega = 2\mu^* \langle \boldsymbol{\epsilon}' \rangle_\Omega, \tag{1.5}$$

where κ^* and μ^* are the effective bulk and shear moduli, $\frac{tr\boldsymbol{\sigma}}{3}$ is the dilatational stress, $\frac{tr\boldsymbol{\epsilon}}{3}$ is the dilatational strain, $\boldsymbol{\sigma}'$ is the deviatoric stress and $\boldsymbol{\epsilon}'$ is the deviatoric strain. Please see the appendix.

- Overall thermal conductivity:

$$\langle \boldsymbol{q} \rangle_\Omega = -\mathbb{K}^* \cdot \langle \nabla\theta \rangle_\Omega, \tag{1.6}$$

where $\mathbb{K}*$ is the effective thermal conductivity for the mixture, ($\mathbb{K}* = \mathbb{K}^*\mathbf{1}$ when isotropic), $\langle \boldsymbol{q} \rangle_\Omega$ is the volume averaged thermal flux field, $\langle \nabla\theta \rangle_\Omega$ is the volume averaged thermal gradient field flux.

Additionally effective mechanical and thermal conductivity will also be introduced later.
Remark: Other properties, although not needed immediately in the analysis are:

- Overall electrical permittivity:

$$\langle \boldsymbol{D} \rangle_\Omega = \boldsymbol{\epsilon}^* \cdot \langle \boldsymbol{E} \rangle_\Omega, \tag{1.7}$$

  where $\boldsymbol{\epsilon}^*$ is the effective electrical permittivity for the mixture, $\langle \boldsymbol{E} \rangle_\Omega$ is the volume averaged electric field, $\langle \boldsymbol{D} \rangle_\Omega$ is the volume averaged electric field flux,

- Overall magnetic permeability:

$$\langle \boldsymbol{B} \rangle_\Omega = \boldsymbol{\mu}^* \cdot \langle \boldsymbol{H} \rangle_\Omega, \tag{1.8}$$

  where μ^* is the effective magnetic permeability for the mixture, $\langle \boldsymbol{H} \rangle_\Omega$ is the volume averaged magnetic field, $\langle \boldsymbol{B} \rangle_\Omega$ is the volume averaged magnetic field flux,

Additionally effective mechanical and thermal conductivity will also be introduced later.

1.2 Computing the effective electrical conductivity

In order to make estimates of the overall properties of a mixture, we consider the widely used Hashin and Shtrikman bounds (see appendix) for isotropic materials with isotropic effective responses. These estimates provide one with upper and lower bounds on the overall response of the material. For two isotropic materials with an overall isotropic response we utilize the following estimates.

$$\underbrace{\sigma_1^c + \frac{v_2^c}{\frac{1}{\sigma_2^c - \sigma_1^c} + \frac{1-v_2^c}{3\sigma_1^c}}}_{\sigma^{c,*,-}} \leq \sigma^{c,*} \leq \underbrace{\sigma_2^c + \frac{1-v_2^c}{\frac{1}{\sigma_1^c - \sigma_2^c} + \frac{v_2^c}{3\sigma_2^c}}}_{\sigma^{c,*,+}}, \tag{1.9}$$

where the conductivity of phase 2 (with volume fraction v_2^c) is larger than phase 1 ($\sigma_2^c \geq \sigma_1^c$). Usually, v_2^c corresponds to the particle material, although there can be applications where the matrix is more conductive that the particles. In that case, v_2^c would correspond to the matrix material. Provided that the volume fractions and constituent conductivities are the only known information about the microstructure, the expressions are the tightest bounds for the overall isotropic effective responses for two phase media, where the constituents are both isotropic. A critical observation is that the lower bound is more accurate when the material is composed of high conductivity particles that are surrounded by a low conductivity matrix (denoted case 1) and the upper bound is more accurate for a high conductivity matrix surrounding low conductivity particles (denoted case 2).
This can be explained by considering two cases of material combinations, one with with 50 % low conductivity material and 50 % high conductivity material. A material with a continuous low conductivity (fine-scale powder) binder (50 %) will isolate the high conductivity particles ((50 %), and the overall system will not conduct electricity well (this is case 1 and the lower bound is more accurate), while a material formed by a continuous high conductivity (fine-scale powder) binder (50 %) surrounding low conductivity particles (50 %, case 2) will, in an overall sense, conduct electricity better than case 1. Thus, case 2 is more closely approximated by the upper bound and case 1 is closer to the lower bound. Since the true effective property lies between the upper and lower bounds, one can construct the following approximation

$$\sigma^{c,*} \approx \phi^c \sigma^{c,*,+} + (1 - \phi^c)\sigma^{c,*,-}, \tag{1.10}$$

where $0 \leq \phi^c \leq 1$. *ϕ^c is function of the microstructure, and must be calibrated.*
As mentioned, for high conductivity spherical particles, at low volume fractions, for example under 15 %, where the particles are not making contact, the lower bound is more accurate. Thus, one would pick $\phi^c = \phi^s \leq 0.5$ to bias the estimate to the lower bound. However, if we take the same volume fraction of particles, but make the flat flakes, they will certainly touch, and produce high-conductivity pathways. Their overall conductivity will cerianly be higher than those of sphere at the same volume fraction. Thus, one

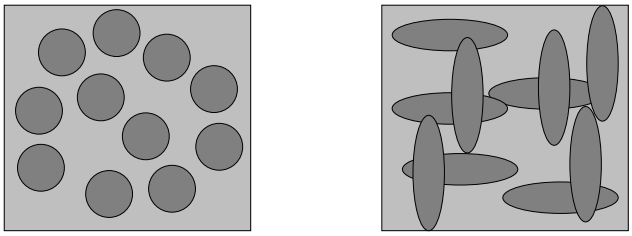

Figure 1.2: Comparing microstructures with the same volume fractions. Flakes touch more, and thus need a higher value of ϕ^c.

would pick $\phi^c = \phi^f > \phi^s$. One can calibrate ϕ^c by comparing it to different experiments. This was done before, for example for mechanical propeties in the work below:

Zohdi, T. I., Monteiro, P. J. M. and Lamour, V. (2002). Extraction of elastic moduli from granular compacts. The International Journal of Fracture/Letters in Micromechanics.115, L49-L54.

Essentially, the more the particles interact (for example physically touch) the more the upper bound become relevant. The general trends are (a) for cases where the upper bound is more accurate, $\phi^c > \frac{1}{2}$ and (b) for cases when the lower bound is more accurate, $\phi^c < \frac{1}{2}$. ϕ^c indicates the degree of interaction of the particulate constituents.

1.3 Concentration tensors and load sharing

For such a sample of material, one can decompose the electrical field carried by each phase in the material as follows

$$\langle \boldsymbol{E} \rangle_\Omega = \frac{1}{|\Omega|}\left(\int_{\Omega_1} \boldsymbol{E}\, d\Omega + \int_{\Omega_2} \boldsymbol{E}\, d\Omega\right) = v_1 \langle \boldsymbol{E} \rangle_{\Omega_1} + v_2 \langle \boldsymbol{E} \rangle_{\Omega_2} \tag{1.11}$$

and the current can be decomposed as

$$\langle \boldsymbol{J} \rangle_\Omega = \frac{1}{|\Omega|}\left(\int_{\Omega_1} \boldsymbol{J}\, d\Omega + \int_{\Omega_2} \boldsymbol{J}\, d\Omega\right) = v_1 \langle \boldsymbol{J} \rangle_{\Omega_1} + v_2 \langle \boldsymbol{J} \rangle_{\Omega_2}, \tag{1.12}$$

The effective conductivity, $\boldsymbol{\sigma}^{c,*}$, defined via[1]

$$\langle \boldsymbol{J} \rangle_\Omega = \boldsymbol{\sigma}^{c,*} \cdot \langle \boldsymbol{E} \rangle_\Omega, \tag{1.13}$$

can be re-written in the following manner

[1] Implicitly, we assume that (a) the contact between the phases is perfect and (b) the ergodicity hypothesis is satisfied (see Kröner [**??**] or Torquato [**??**]).

$$\begin{aligned}\langle \boldsymbol{J}\rangle_{\Omega} &= v_1^c\langle \boldsymbol{J}\rangle_{\Omega_1} + v_2^c\langle \boldsymbol{J}\rangle_{\Omega_2}\\ &= v_1^c\boldsymbol{\sigma}_1^c\cdot\langle \boldsymbol{E}\rangle_{\Omega_1} + v_2^c\boldsymbol{\sigma}_2^c\cdot\langle \boldsymbol{E}\rangle_{\Omega_2}\\ &= \boldsymbol{\sigma}_1^c\cdot(\langle \boldsymbol{E}\rangle_{\Omega} - v_2^c\langle \boldsymbol{E}\rangle_{\Omega_2}) + v_2^c\boldsymbol{\sigma}_2^c\cdot\langle \boldsymbol{E}\rangle_{\Omega_2}\\ &= \underbrace{(\boldsymbol{\sigma}_1^c + v_2^c(\boldsymbol{\sigma}_2^c-\boldsymbol{\sigma}_1^c)\cdot\boldsymbol{C}_{E,2})}_{\boldsymbol{\sigma}^{c,*}}\cdot\langle \boldsymbol{E}\rangle_{\Omega}, \end{aligned} \tag{1.14}$$

where

$$\underbrace{\left(\frac{1}{v_2^c}(\boldsymbol{\sigma}_2^c-\boldsymbol{\sigma}_1^c)^{-1}\cdot(\boldsymbol{\sigma}^{c,*}-\boldsymbol{\sigma}_1^c)\right)}_{\stackrel{\text{def}}{=}\boldsymbol{C}_{E,2}}\cdot\langle \boldsymbol{E}\rangle_{\Omega} = \langle \boldsymbol{E}\rangle_{\Omega_2}. \tag{1.15}$$

$\boldsymbol{C}_{E,2}$ is known as the electric field concentration tensor. Thus, the product of $\boldsymbol{C}_{E,2}$ with $\langle \boldsymbol{E}\rangle_{\Omega}$ yields $\langle \boldsymbol{E}\rangle_{\Omega_2}$. *It is important to realize that once either $\boldsymbol{C}_{E,2}$ or $\boldsymbol{\sigma}^{c,*}$ are known, the other can be computed.*
In order to determine the concentration tensor for phase 1, we have from Equation 1.11

$$\langle \boldsymbol{E}\rangle_{\Omega_1} = \frac{\langle \boldsymbol{E}\rangle_{\Omega} - v_2^c\langle \boldsymbol{E}\rangle_{\Omega_2}}{v_1^c} = \frac{(\mathbf{1}-v_2^c\boldsymbol{C}_{E,2})\cdot\langle \boldsymbol{E}\rangle_{\Omega}}{v_1^c} \stackrel{\text{def}}{=} \boldsymbol{C}_{E,1}\cdot\langle \boldsymbol{E}\rangle_{\Omega}, \tag{1.16}$$

where

$$\boldsymbol{C}_{E,1} = \frac{1}{v_1^c}(\mathbf{1}-v_2^c\boldsymbol{C}_{E,2}) = \frac{\mathbf{1}-v_2^c\boldsymbol{C}_{E,2}}{1-v_2^c}. \tag{1.17}$$

Note that Equation 1.17 implies

$$\underbrace{v_1^c\boldsymbol{C}_{E,1}}_{\text{phase}-1\text{ contribution}} + \underbrace{v_2^c\boldsymbol{C}_{E,2}}_{\text{phase}-2\text{ contribution}} = \mathbf{1}. \tag{1.18}$$

Similarly, for the current, we have

$$\langle \boldsymbol{J}\rangle_{\Omega} = \boldsymbol{\sigma}^{c,*}\cdot\langle \boldsymbol{E}\rangle_{\Omega} \Rightarrow \boldsymbol{\sigma}^{c,*-1}\cdot\langle \boldsymbol{J}\rangle_{\Omega} = \boldsymbol{C}_{E,2}^{-1}\cdot\langle \boldsymbol{E}\rangle_{\Omega_2} = \boldsymbol{C}_{E,2}^{-1}\cdot\boldsymbol{\sigma}_2^{-1}\cdot\langle \boldsymbol{J}\rangle_{\Omega_2}. \tag{1.19}$$

Thus,

$$\underbrace{\boldsymbol{\sigma}_2^c\cdot\boldsymbol{C}_{E,2}\cdot\boldsymbol{\sigma}^{c,*-1}}_{\boldsymbol{C}_{J,2}}\cdot\langle \boldsymbol{J}\rangle_{\Omega} = \langle \boldsymbol{J}\rangle_{\Omega_2}, \tag{1.20}$$

and

$$\boldsymbol{C}_{J,1}\cdot\langle \boldsymbol{J}\rangle_{\Omega} = \langle \boldsymbol{J}\rangle_{\Omega_1} \tag{1.21}$$

where

$$\boldsymbol{C}_{J,1} = \frac{\mathbf{1}-v_2^c\boldsymbol{C}_{J,2}}{1-v_2^c} = \boldsymbol{\sigma}_1^c\cdot\boldsymbol{C}_{E,1}\cdot\boldsymbol{\sigma}^{c,*-1}. \tag{1.22}$$

We remark that from Equation 1.22 implies

$$\underbrace{v_1^c\boldsymbol{C}_{J,1}}_{\text{phase}-1\text{ contribution}} + \underbrace{v_2^c\boldsymbol{C}_{J,2}}_{\text{phase}-2\text{ contribution}} = \mathbf{1}. \tag{1.23}$$

Summarizing, we have the following results:

- $\boldsymbol{C}_{E,1}\cdot\langle \boldsymbol{E}\rangle_{\Omega} = \langle \boldsymbol{E}\rangle_{\Omega_1}$ where $\boldsymbol{C}_{E,1} = \frac{1}{v_1^c}(\mathbf{1}-v_2^c\boldsymbol{C}_{E,2}) = \frac{\mathbf{1}-v_2^c\boldsymbol{C}_{E,2}}{1-v_2^c}$,

- $\boldsymbol{C}_{E,2} \cdot \langle \boldsymbol{E} \rangle_{\Omega} = \langle \boldsymbol{E} \rangle_{\Omega_2}$ where $\boldsymbol{C}_{E,2} = \frac{1}{v_2^c}(\boldsymbol{\sigma}_2^c - \boldsymbol{\sigma}_1^c)^{-1} \cdot (\boldsymbol{\sigma}^{c,*} - \boldsymbol{\sigma}_1^c)$,
- $\boldsymbol{C}_{J,1} \cdot \langle \boldsymbol{J} \rangle_{\Omega} = \langle \boldsymbol{J} \rangle_{\Omega_1}$ where $\boldsymbol{C}_{J,1} = \frac{\mathbf{1} - v_2^c \boldsymbol{C}_{J,2}}{1 - v_2^c} = \boldsymbol{\sigma}_1^c \cdot \boldsymbol{C}_{E,1} \cdot \boldsymbol{\sigma}^{c,*-1}$,
- $\boldsymbol{C}_{J,2} \cdot \langle \boldsymbol{J} \rangle_{\Omega} = \langle \boldsymbol{J} \rangle_{\Omega_2}$ where $\boldsymbol{C}_{J,2} = \boldsymbol{\sigma}_2^c \cdot \boldsymbol{C}_{E,2} \cdot \boldsymbol{\sigma}^{c,*-1}$.

1.4 "Load sharing" interpretation

One may write

$$\begin{aligned} \underbrace{v_1^c \boldsymbol{C}_{E,1}}_{\text{phase}-1 \text{ contribution}} + \underbrace{v_2^c \boldsymbol{C}_{E,2}}_{\text{phase}-2 \text{ contribution}} &= \mathbf{1} \\ \underbrace{v_1^c \boldsymbol{C}_{J,1}}_{\text{phase}-1 \text{ contribution}} + \underbrace{v_2^c \boldsymbol{C}_{J,2}}_{\text{phase}-2 \text{ contribution}} &= \mathbf{1} \end{aligned} \tag{1.24}$$

Frequently, the first term in the above expressions is referred to as "phase-1's" share, while the second term is "phase-2's" share. Thus, the concentration tensor tells you how much of the overall load is carried by the particles and how much by the matrix. This is useful shortly, when we want to compute the Joule heating in the system. In the Appendix, other material properties, not directlty needed here, are discussed.

1.5 Joule Heating

One of the key quantities of interest here is the amount of heat generated from running a current through a material, denoted H (a rate), which feeds into first law of thermodynamics,

$$\rho \dot{w} - \boldsymbol{\sigma} : \nabla \dot{\boldsymbol{u}} + \nabla \cdot \boldsymbol{q} = \underbrace{H.}_{Joule-heating\ source} \tag{1.25}$$

In Equation 1.25, ρ is the mass density, w is the stored energy per unit mass, $\boldsymbol{\sigma}$ is Cauchy stress, $\boldsymbol{u}$ is the displacement field, $\boldsymbol{q}$ is heat flux, and $H = a\,(\boldsymbol{J} \cdot \boldsymbol{E})$, is the rate of electrical energy absorbed due to Joule-heating, where $\boldsymbol{J}$ is the current, $\boldsymbol{E}$ is the electric field and $0 \leq a \leq 1$ is an absorption constant. Our objective in this work is to determine the phase-wise load-shares of the Joule-field, denoted $H = \boldsymbol{J} \cdot \boldsymbol{E}$, carried by the components in the heterogeneous mixture. The Joule-heating Material 1 (eventually assuming isotropy)

$$H_1 = \langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \langle \boldsymbol{E} \rangle_{\Omega_1} = \langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \boldsymbol{\sigma}_1^{c,-1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_1} = \frac{\langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_1}}{\sigma_1^c} \tag{1.26}$$

The Joule-heating Material 2:

$$H_2 = \langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \langle \boldsymbol{E} \rangle_{\Omega_2} = \langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \boldsymbol{\sigma}_2^{c,-1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_2} = \frac{\langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \langle \boldsymbol{J} \rangle_{\Omega_2}}{\sigma_2^c} \tag{1.27}$$

These can be rewritten in terms of macroscale quantities: $\langle \boldsymbol{J} \rangle_{\Omega}$ and $\langle \boldsymbol{E} \rangle_{\Omega}$:

$$H_1 = (\boldsymbol{C}_{J,1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_1}) \cdot (\boldsymbol{C}_{E,1} \cdot \langle \boldsymbol{E} \rangle_{\Omega_1}) = C_{J,1} C_{E,1} \langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \langle \boldsymbol{E} \rangle_{\Omega_1}. \tag{1.28}$$

and

$$H_2 = (\boldsymbol{C}_{J,2} \cdot \langle \boldsymbol{J} \rangle_{\Omega_2}) \cdot (\boldsymbol{C}_{E,2} \cdot \langle \boldsymbol{E} \rangle_{\Omega_2}) = C_{J,2} C_{E,2} \langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \langle \boldsymbol{E} \rangle_{\Omega_2}. \tag{1.29}$$

If the overall property is isotropic (assumed above), and each of the constituents is isotropic (for example a microstructure comprised of a continuous isotropic binder embedded with randomly distributed isotropic particles), then we have the following, $\boldsymbol{C}_{E,i} = C_{E,i}\mathbf{1}$ where, for a two-phase material

$$C_{E,1} = \frac{1}{1 - v_2^c} \frac{\sigma_2^c - \sigma^{c,*}}{\sigma_2^c - \sigma_1^c} \text{ and } C_{E,2} = \frac{1}{v_2^c} \frac{\sigma^{c,*} - \sigma_1^c}{\sigma_2^c - \sigma_1^c}, \tag{1.30}$$

and $\boldsymbol{C}_{J,i} = C_{J,i}\mathbf{1}$, leading to

$$C_{J,1} = \frac{\sigma_1^c}{\sigma^{c,*}(1-v_2^c)}\left(\frac{\sigma_2^c - \sigma^{c,*}}{\sigma_2^c - \sigma_1^c}\right) \text{ and } C_{J,2} = \frac{\sigma_2^c}{\sigma^{c,*}v_2^c}\left(\frac{\sigma^{c,*} - \sigma_1^c}{\sigma_2^c - \sigma_1^c}\right). \tag{1.31}$$

The product of the concentration functions take the following form:

$$C_{E,1}C_{J,1} = \frac{\sigma_1^c}{\sigma^{c,*}}\left(\frac{1}{(1-v_2^c)}\left(\frac{\sigma_2^c - \sigma^{c,*}}{\sigma_2^c - \sigma_1^c}\right)\right)^2 \tag{1.32}$$

and

$$C_{E,2}C_{J,2} = \frac{\sigma_2^c}{\sigma^{c,*}}\left(\frac{1}{v_2^c}\left(\frac{\sigma^{c,*} - \sigma_1^c}{\sigma_2^c - \sigma_1^c}\right)\right)^2. \tag{1.33}$$

Because the concentration functions depend on $\sigma^{c,*}$, which in turn depends on σ_1^c, σ_2^c, v_2^c, and the microstructure, we need to employ estimates for $\sigma^{c,*}$.

Remark: Recall, as introduced earlier, one class of estimates are the Hashin-Shtrikman bounds (Hashin and Shtrikman [23]) for two isotropic materials with an overall isotropic response

$$\underbrace{\sigma_1^c + \frac{v_2^c}{\frac{1}{\sigma_2^c - \sigma_1^c} + \frac{1-v_2^c}{3\sigma_1^c}}}_{\sigma^{c,*,-}} \leq \sigma^{c,*} \leq \underbrace{\sigma_2^c + \frac{1-v_2^c}{\frac{1}{\sigma_1^c - \sigma_2^c} + \frac{v_2^c}{3\sigma_2^c}}}_{\sigma^{c,*,+}}, \tag{1.34}$$

where the conductivity of phase 2 (with volume fraction v_2^c) is larger than phase 1 ($\sigma_2^c \geq \sigma_1^c$). Provided that the volume fractions and constituent conductivities are the only known information about the microstructure, the expressions are the tightest bounds for the overall isotropic effective responses for two phase media, where the constituents are both isotropic. A critical observation is that the lower bound is more accurate when the material is composed of high conductivity particles that are surrounded by a low conductivity matrix (denoted case 1) and the upper bound is more accurate for a high conductivity matrix surrounding low conductivity particles (denoted case 2). Since the true effective property lies between the upper and lower bounds, one can construct the following approximation

$$\sigma^{c,*} \approx \phi^c \sigma^{c,*,+} + (1-\phi^c)\sigma^{c,*,-}, \tag{1.35}$$

where $0 \leq \phi^c \leq 1$. *ϕ^c is an unknown function of the microstructure.* However, the general trends are (a) for cases where the upper bound is more accurate, $\phi^c > \frac{1}{2}$ and (b) for cases when the lower bound is more accurate, $\phi^c < \frac{1}{2}$. Explicitly, for the product of concentration functions, embedding the effective property estimates, we have

$$C_{E,1}C_{J,1} \approx \frac{\sigma_1^c}{(\phi^c\sigma^{c,*,+} + (1-\phi^c)\sigma^{c,*,-})}\left(\frac{1}{(1-v_2^c)}\left(\frac{\sigma_2^c - (\phi^c\sigma^{c,*,+} + (1-\phi^c)\sigma^{c,*,-})}{\sigma_2^c - \sigma_1^c}\right)\right)^2 \tag{1.36}$$

and

$$C_{E,2}C_{J,2} \approx \frac{\sigma_2^c}{(\phi^c\sigma^{c,*,+} + (1-\phi^c)\sigma^{c,*,-})}\left(\frac{1}{v_2^c}\left(\frac{(\phi^c\sigma^{c,*,+} + (1-\phi^c)\sigma^{c,*,-}) - \sigma_1^c}{\sigma_2^c - \sigma_1^c}\right)\right)^2. \tag{1.37}$$

Remark: There are a vast literature of methods, dating back to Maxwell [**??**], [**??**] and Lord Rayleigh [**??**], to estimate the overall macroscopic properties of heterogeneous materials. For an authoritative review of (a) the general theory of random heterogeneous media see Torquato [**??**], (b) for more mathematical homogenization aspects see Jikov et al. [25], (c) for solid-mechanics inclined accounts of the subject see Hashin [**??**], Mura [26], Nemat-Nasser and Hori [**??**], (d) for analyses of cracked media see Sevostianov et al. [**??**] and (e) for computational aspects see Ghosh [**??**], Ghosh and Dimiduk [**??**] and Zohdi and Wriggers [**??**]. Tighter estimates, including generalized N-phase bounds can be found in Torquato [**??**]. [2]

[2]Such N-phase bounds go well beyond the the simple Wiener Bounds [**??**], $\left(\sum_{i=1}^N v_i^c \boldsymbol{\sigma}_i^{-1}\right)^{-1} \leq \boldsymbol{\sigma}^* \leq \sum_{i=1}^N v_i^c \boldsymbol{\sigma}_i$.

1.6 Optimization formulation for electrical properties

In order to utilize the previous results, we now provide examples of how to optimize the material properties. In particular, we will optimize the overall electrical conductivity of a mixture of material, subject to Joule-heating constraints on each phase. The effective conductivity is given by $\sigma^{c,*}$ using convex combinations of the Hashin-Shtrikman bounds as approximations for the effective moduli $\sigma^{c,*} \approx \phi^c \sigma^{c,*+} + (1-\phi^c)\sigma^{c,*-}$, where $0 \leq \phi^c \leq 1$. The micro-macro objective function is

$$\boxed{\Pi^{elec} = w_1 |\frac{\sigma^{c,*,Des} - \sigma^{c,*}}{\sigma^{c,*,Des}}| + \hat{w}_2 |\frac{C_{J1}C_{E1} - TOL_1}{TOL_1}| + \hat{w}_3 |\frac{C_{J2}C_{E2} - TOL_2}{TOL_2}|,} \tag{1.38}$$

where the constraints are unilaterally-activated in the following manner:

- $\frac{C_{J1}C_{E1}-TOL_1}{TOL_1} \leq 0$ then $\hat{w}_2 = 0$,
- $\frac{C_{J1}C_{E1}-TOL_1}{TOL_1} > 0$ then $\hat{w}_2 = w_2$,
- $\frac{C_{J2}C_{E2}-TOL_2}{TOL_2} \leq 0$ then $\hat{w}_3 = 0$,
- $\frac{C_{J2}C_{E2}-TOL_2}{TOL_2} > 0$ then $\hat{w}_3 = w_3$.

Here the design variables are $\mathbf{\Lambda} = \{\sigma_2^c, \sigma_1^c\, v_2^c\}$, and their constrained ranges are

- $(\sigma_1^c)^{(-)} \leq \sigma_1^c \leq (\sigma_1^c)^{(+)}$,
- $(\sigma_2^c)^{(-)} \leq \sigma_2^c \leq (\sigma_2^c)^{(+)}$ and
- $(v_2^c)^{(-)} \leq v_2^c \leq (v_2^c)^{(+)}$.

1.7 Optimization formulation for thermal properties

In order to utilize the previous results, we now provide examples of how to optimize the material properties. In particular, we will optimize the overall thermal conductivity of a mixture of material, subject to thermal constraints on each phase. The effective thermal conductivity is given by $\mathbb{K}^*$ using convex combinations of the Hashin-Shtrikman bounds as approximations for the effective moduli $\mathbb{K}^* \approx \phi^K \mathbb{K}^{*+} + (1-\phi^K)\mathbb{K}^{*-}$, where $0 \leq \phi^K \leq 1$. The micro-macro objective function is

$$\boxed{\Pi^{thermal} = w_1 |\frac{\mathbb{K}^{*,Des} - \mathbb{K}^{c,*}}{\mathbb{K}^{*,Des}}| + \hat{w}_2 |\frac{C_{q1} - TOL_1}{TOL_1}| + \hat{w}_3 |\frac{C_{q2} - TOL_2}{TOL_2}|} \tag{1.39}$$

where the constraints are unilaterally-activated in the following manner:

- $\frac{C_{q1}-TOL_1}{TOL_1} \leq 0$ then $\hat{w}_2 = 0$,
- $\frac{C_{q1}-TOL_1}{TOL_1} > 0$ then $\hat{w}_2 = w_2$,
- $\frac{C_{q2}-TOL_2}{TOL_2} \leq 0$ then $\hat{w}_3 = 0$,
- $\frac{C_{q2}-TOL_2}{TOL_2} > 0$ then $\hat{w}_3 = w_3$.

Here the design variables are $\mathbf{\Lambda} = \{\mathbb{K}_2, \mathbb{K}_1\, v_2^K\}$, and their constrained ranges are

- $(\mathbb{K}_1)^{(-)} \leq \mathbb{K}_1 \leq (\mathbb{K}_1)^{(+)}$,
- $(\mathbb{K}_2)^{(-)} \leq \mathbb{K}_2 \leq (\mathbb{K}_2)^{(+)}$ and
- $(v_2^K)^{(-)} \leq v_2^K \leq (v_2^K)^{(+)}$.

Remark: The effective thermal conductivity estimates are, for two isotropic materials with an overall isotropic response we utilize the following estimates.

$$\underbrace{\mathbb{K}_1 + \frac{v_2^K}{\frac{1}{\mathbb{K}_2-\mathbb{K}_1} + \frac{1-v_2^K}{3\mathbb{K}_1}}}_{\mathbb{K}^{*,-}} \leq \mathbb{K}^* \leq \underbrace{\mathbb{K}_2 + \frac{1-v_2^K}{\frac{1}{\mathbb{K}_1-\mathbb{K}_2} + \frac{v_2^K}{3\mathbb{K}_2}}}_{\mathbb{K}^{*,+}}, \tag{1.40}$$

where the conductivity of phase 2 (with volume fraction v_2^K) is larger than phase 1 ($\mathbb{K}_2 \geq \mathbb{K}_1$). Usually, v_2^K corresponds to the particle material, although there can be applications where the matrix is more conductive that the particles. In that case, v_2^K would correspond to the matrix material. To obtain the concentration of the thermal fields in each phase, the exact procedure holds with $\nabla\theta$ replacing $\boldsymbol{E}$, $\boldsymbol{q}$ replacing $\boldsymbol{J}$, $\mathbb{K}$ replacing $\boldsymbol{\sigma}^c$, v_1^K replacing v_1^c and v_2^K replacing v_2^c. For thermal properties following concentration tensors:

- $\boldsymbol{C}_{\theta,2} \cdot \langle\nabla\theta\rangle_\Omega = \langle\nabla\theta\rangle_{\Omega_2}$ where $\boldsymbol{C}_{\theta,2} = \frac{1}{v_2^K}(\mathbb{K}_2 - \mathbb{K}_1)^{-1} \cdot (\mathbb{K}^* - \mathbb{K}_1)$
- $\boldsymbol{C}_{\theta,1} \cdot \langle\nabla\theta\rangle_\Omega = \langle\nabla\theta\rangle_{\Omega_1}$ where $\boldsymbol{C}_{\theta,1} = \frac{1}{v_1^K}(\mathbf{1} - v_2^K\boldsymbol{C}_{\theta,2}) = \frac{\mathbf{1}-v_2^K\boldsymbol{C}_{\theta,2}}{1-v_2^K}$
- $\boldsymbol{C}_{q,2} \cdot \langle\boldsymbol{q}\rangle_\Omega = \langle\boldsymbol{q}\rangle_{\Omega_2}$ where $\boldsymbol{C}_{q,2} = \mathbb{K}_2 \cdot \boldsymbol{C}_{\theta,2} \cdot \mathbb{K}^{*-1}$
- $\boldsymbol{C}_{q,1} \cdot \langle\boldsymbol{q}\rangle_\Omega = \langle\boldsymbol{q}\rangle_{\Omega_1}$ where $\boldsymbol{C}_{q,1} = \frac{\mathbf{1}-v_2^K\boldsymbol{C}_{q,2}}{1-v_2^K}$

1.8 Estimation and optimization of the effective properties of mechanical mixtures

1.8.1 Combining bounds

The typical use of the bounds from the previous chapter is to make an estimate of the effective properties by forming a convex combination of them in the following manner:

$$\kappa^* \approx \phi^E\kappa^{*,+} + (1-\phi^E)\kappa^{*,-} \tag{1.41}$$

and

$$\mu^* \approx \phi^E\mu^{*,+} + (1-\phi^E)\mu^{*,-}, \tag{1.42}$$

where, for isotropic materials with isotropic effective (mechanical) responses, the Hashin-Shtrikman bounds (for a two-phase material) are as follows for the bulk modulus

$$\kappa^{*,-} \overset{\text{def}}{=} \kappa_1 + \frac{v_2^E}{\frac{1}{\kappa_2-\kappa_1} + \frac{3(1-v_2^E)}{3\kappa_1+4\mu_1}} \leq \kappa^* \leq \kappa_2 + \frac{1-v_2^E}{\frac{1}{\kappa_1-\kappa_2} + \frac{3v_2^E}{3\kappa_2+4\mu_2}} \overset{\text{def}}{=} \kappa^{*,+} \tag{1.43}$$

and for the shear modulus

$$\mu^{*,-} \overset{\text{def}}{=} \mu_1 + \frac{v_2^E}{\frac{1}{\mu_2-\mu_1} + \frac{6(1-v_2^E)(\kappa_1+2\mu_1)}{5\mu_1(3\kappa_1+4\mu_1)}} \leq \mu^* \leq \mu_2 + \frac{(1-v_2^E)}{\frac{1}{\mu_1-\mu_2} + \frac{6v_2^E(\kappa_2+2\mu_2)}{5\mu_2(3\kappa_2+4\mu_2)}} \overset{\text{def}}{=} \mu^{*,+}, \tag{1.44}$$

where κ_2 and κ_1 are the bulk moduli and μ_2 and μ_1 are the shear moduli of the respective phases ($\kappa_2 \geq \kappa_1$ and $\mu_2 \geq \mu_1$), and where v_2^E is the second phase volume fraction. Note that no geometric or other microstructural information is required for the bounds.
The parameter $0 \leq \phi^E \leq 1$ is such that:

- If $\phi^E = 0$ we have the lower bound,
- If $\phi^E = 1$ we have the upper bound,
- If $\phi^E = 1/2$ we have the average of the bounds.

ϕ^E is a function of the microstructure, and must be calibrated. As before, a critical observation is that the lower bound is more accurate when the material is composed of stiff particles that are surrounded by a soft matrix (denoted case 1) and the upper bound is more accurate for a stiff matrix surrounding soft particles (denoted case 2). This can be explained by considering two cases of material combinations, one with 50 % soft material and 50 % stiff material. A material with a continuous soft binder (50 %) will isolate the stiff particles (50 %), and the overall system will not be stiff (this is case 1 and the lower bound is more accurate), while a material formed by a continuous stiff binder (50 %) surrounding soft particles (50 %, case 2) will, in an overall sense, are stiffer than case 1. Thus, case 2 is more closely approximated by the upper bound and case 1 is closer to the lower bound.

1.8.2 Local fields: stresses and strains

The determination of the average load sharing between phases at the microstructural scale can be obtained from the overall effective mechanical properties of the microheterogeneous material, for example, comprised of particles suspended in a binding matrix.
The load carried by each phase in the microstructure is characterized via stress and strain concentration tensors, which we now discuss. These provide a measure of the deviation away from the mean fields throughout the material. One can decompose averages of an arbitrary quantity over Ω into averages over the each of the phases in the following manner: $\langle \boldsymbol{A} \rangle_\Omega = (1/|\Omega|) \left(\int_{\Omega_1} \boldsymbol{A}\, d\Omega + \int_{\Omega_2} \boldsymbol{A}\, d\Omega \right) = v_1^E \langle \boldsymbol{A} \rangle_{\Omega_1} + v_2^E \langle \boldsymbol{A} \rangle_{\Omega_2}$. If we make use of this decomposition, we have

$$\begin{aligned} \langle \boldsymbol{\sigma} \rangle_\Omega &= v_1^E \langle \boldsymbol{\sigma} \rangle_{\Omega_1} + v_2^E \langle \boldsymbol{\sigma} \rangle_{\Omega_2} \\ &= v_1^E \mathbb{E}_1 : \langle \boldsymbol{\epsilon} \rangle_{\Omega_1} + v_2^E \mathbb{E}_2 : \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \\ &= \mathbb{E}_1 : (\langle \boldsymbol{\epsilon} \rangle_\Omega - v_2^E \langle \boldsymbol{\epsilon} \rangle_{\Omega_2}) + v_2^E \mathbb{E}_2 : \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \\ &= \left(\mathbb{E}_1 + v_2^E (\mathbb{E}_2 - \mathbb{E}_1) : \boldsymbol{C}^{\epsilon,2} \right) : \langle \boldsymbol{\epsilon} \rangle_\Omega, \end{aligned} \tag{1.45}$$

where $\boldsymbol{C}^{\epsilon,2} \stackrel{\text{def}}{=} \left(\frac{1}{v_2^E} (\mathbb{E}_2 - \mathbb{E}_1)^{-1} : (\mathbb{E}^* - \mathbb{E}_1) \right)$ with $\boldsymbol{C}^{\epsilon,2} : \langle \boldsymbol{\epsilon} \rangle_\Omega = \langle \boldsymbol{\epsilon} \rangle_{\Omega_2}$. The strain concentration tensor $\boldsymbol{C}^{\epsilon,2}$ relates the average strain over the particle phase (2) to the average strain over all phases. Similarly, for the variation in the stress we have $\boldsymbol{C}^{\epsilon,2} : \mathbb{E}^{*-1} : \langle \boldsymbol{\sigma} \rangle_\Omega = \mathbb{E}_2^{-1} : \langle \boldsymbol{\sigma} \rangle_{\Omega_2}$, which reduces to $\mathbb{E}_2 : \boldsymbol{C}^{\epsilon,2} : \mathbb{E}^{*-1} : \langle \boldsymbol{\sigma} \rangle_\Omega \stackrel{\text{def}}{=} \boldsymbol{C}^{\sigma,2} : \langle \boldsymbol{\sigma} \rangle_\Omega = \langle \boldsymbol{\sigma} \rangle_{\Omega_2}$. $\boldsymbol{C}^{\sigma,2}$ is known as the stress concentration tensor; it relates the average stress in the particle phase to that in the whole RVE. Note that once either $\boldsymbol{C}^{\epsilon,2}$ or $\mathbb{E}^*$ are known, the other can be determined. In the case of isotropy we may write

$$C_\kappa^{\sigma,2} \stackrel{\text{def}}{=} \frac{1}{v_2^E} \frac{\kappa_2}{\kappa^*} \frac{\kappa^* - \kappa_1}{\kappa_2 - \kappa_1} \qquad \text{and} \qquad C_\mu^{\sigma,2} \stackrel{\text{def}}{=} \frac{1}{v_2^E} \frac{\mu_2}{\mu^*} \frac{\mu^* - \mu_1}{\mu_2 - \mu_1} \tag{1.46}$$

where $C_\kappa^{\sigma,2} \langle \frac{tr \boldsymbol{\sigma}}{3} \rangle_\Omega = \langle \frac{tr \boldsymbol{\sigma}}{3} \rangle_{\Omega_2}$ and where $C_\mu^{\sigma,2} \langle \boldsymbol{\sigma}' \rangle_\Omega = \langle \boldsymbol{\sigma}' \rangle_{\Omega_2}$. Clearly, the microstress fields are minimally distorted when $C_\kappa^{\sigma,2} = C_\mu^{\sigma,2} = 1$; there are no stress concentrations in a homogeneous material. For the matrix,

$$\langle \boldsymbol{\sigma} \rangle_{\Omega_1} = \frac{\langle \boldsymbol{\sigma} \rangle_\Omega - v_2^E \langle \boldsymbol{\sigma} \rangle_{\Omega_2}}{v_1^E} = \frac{\langle \boldsymbol{\sigma} \rangle_\Omega - v_2^E \boldsymbol{C}^{\sigma,2} : \langle \boldsymbol{\sigma} \rangle_\Omega}{v_1^E} = \frac{(\mathbf{1} - v_2^E \boldsymbol{C}^{\sigma,2}) : \langle \boldsymbol{\sigma} \rangle_\Omega}{v_1^E} \stackrel{\text{def}}{=} \boldsymbol{C}^{\sigma,1} : \langle \boldsymbol{\sigma} \rangle_\Omega. \tag{1.47}$$

Therefore, in the case of isotropy,

$$C_\kappa^{\sigma,1} \stackrel{\text{def}}{=} \frac{1}{v_1^E} (1 - v_2^E C_\kappa^{\sigma,2}) \qquad \text{and} \qquad C_\mu^{\sigma,1} \stackrel{\text{def}}{=} \frac{1}{v_1^E} (1 - v_2^E C_\mu^{\sigma,2}). \tag{1.48}$$

The fraction of the total stress carried by each phase can be determined by multiplying the concentration factors by the corresponding volume fractions

$$\begin{aligned} \langle \boldsymbol{\sigma} \rangle_\Omega &= v_1^E \langle \boldsymbol{\sigma} \rangle_{\Omega_1} + v_2^E \langle \boldsymbol{\sigma} \rangle_{\Omega_2} \\ &= v_1^E \boldsymbol{C}^{\sigma,1} : \langle \boldsymbol{\sigma} \rangle_\Omega + v_2^E \boldsymbol{C}^{\sigma,2} : \langle \boldsymbol{\sigma} \rangle_\Omega. \end{aligned} \tag{1.49}$$

Remark: Similar to the stress, for the strain, we have for the matrix,

$$\langle \boldsymbol{\epsilon} \rangle_{\Omega_1} = \frac{\langle \boldsymbol{\epsilon} \rangle_{\Omega} - v_2^E \langle \boldsymbol{\epsilon} \rangle_{\Omega_2}}{v_1^E} = \frac{\langle \boldsymbol{\epsilon} \rangle_{\Omega} - v_2^E \boldsymbol{C}^{\epsilon,2} : \langle \boldsymbol{\epsilon} \rangle_{\Omega}}{v_1^E} = \frac{(\mathbf{1} - v_2^E \boldsymbol{C}^{\epsilon,2}) : \langle \boldsymbol{\epsilon} \rangle_{\Omega}}{v_1^E} \stackrel{\text{def}}{=} \boldsymbol{C}^{\epsilon,1} : \langle \boldsymbol{\epsilon} \rangle_{\Omega}. \tag{1.50}$$

Therefore, in the case of isotropy,

$$C_{\kappa}^{\epsilon,1} \stackrel{\text{def}}{=} \frac{1}{v_1^E}(1 - v_2^E C_{\kappa}^{\epsilon,2}) \qquad \text{and} \qquad C_{\mu}^{\epsilon,1} \stackrel{\text{def}}{=} \frac{1}{v_1^E}(1 - v_2^E C_{\mu}^{\epsilon,2}). \tag{1.51}$$

The fraction of the total strain carried by each phase can be determined by multiplying the concentration factors by the corresponding volume fractions

$$\begin{aligned} \langle \boldsymbol{\epsilon} \rangle_{\Omega} &= v_1^E \langle \boldsymbol{\epsilon} \rangle_{\Omega_1} + v_2^E \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \\ &= v_1^E \boldsymbol{C}^{\epsilon,1} : \langle \boldsymbol{\epsilon} \rangle_{\Omega} + v_2^E \boldsymbol{C}^{\epsilon,2} : \langle \boldsymbol{\epsilon} \rangle_{\Omega}. \end{aligned} \tag{1.52}$$

1.9 Optimization: formulation of a mechanical cost-function

The ratio the particulate stress fields to the mean value is

$$\frac{\langle tr\boldsymbol{\sigma} \rangle_{\Omega_2}}{\langle tr\boldsymbol{\sigma} \rangle_{\Omega}} = C_{\kappa}^{\sigma,2} \tag{1.53}$$

and

$$\sqrt{\frac{(\langle \boldsymbol{\sigma}' \rangle_{\Omega_2} : \langle \boldsymbol{\sigma}' \rangle_{\Omega_2})}{\langle \boldsymbol{\sigma}' \rangle_{\Omega} : \langle \boldsymbol{\sigma}' \rangle_{\Omega}}} = C_{\mu}^{\sigma,2}, \tag{1.54}$$

and for the matrix material

$$|\frac{\langle tr\boldsymbol{\sigma} \rangle_{\Omega_1}}{\langle tr\boldsymbol{\sigma} \rangle_{\Omega}}| = C_{\kappa}^{\sigma,1} \tag{1.55}$$

and

$$\sqrt{\frac{(\langle \boldsymbol{\sigma}' \rangle_{\Omega_1} : \langle \boldsymbol{\sigma}' \rangle_{\Omega_1})}{\langle \boldsymbol{\sigma}' \rangle_{\Omega} : \langle \boldsymbol{\sigma}' \rangle_{\Omega}}} = C_{\mu}^{\sigma,1}. \tag{1.56}$$

In order to incorporate the deviation into a cost function, we introduce a tolerance where, ideally,

$$C_{\kappa}^{\sigma,2} \leq TOL_{\kappa} \qquad \text{and} \qquad C_{\mu}^{\sigma,2} \leq TOL_{\mu} \tag{1.57}$$

and

$$C_{\kappa}^{\sigma,1} \leq TOL_{\kappa} \qquad \text{and} \qquad C_{\mu}^{\sigma,1} \leq TOL_{\mu}. \tag{1.58}$$

If the normalized deviation exceeds the corresponding TOL, then the level of violation is incorporated as a multilateral constraint to the macroscopic objectives. As an example, our immediate goal is to computationally design the macroscale effective bulk and shear moduli κ^* and μ^*, using convex combinations of

the Hashin-Shtrikman bounds as approximations for the effective moduli $\kappa^* \approx \phi^E \kappa^{*,+} + (1-\phi^E)\kappa^{*,-}$ and $\mu^* \approx \phi^E \mu^{*,+} + (1-\phi^E)\mu^{*,-}$, where $0 \leq \phi^E \leq 1$. The micro-macro objective function is

$$\boxed{\begin{array}{l}\Pi^{mechanical} = w_1|\dfrac{\kappa^{*D}-\kappa^*}{\kappa^*}| + w_2|\dfrac{\mu^{*D}-\mu^*}{\mu^*}| + \hat{w}_3|\dfrac{C_\kappa^{\sigma,2}-TOL_\kappa}{TOL_\kappa}| + \hat{w}_4|\dfrac{C_\mu^{\sigma,2}-TOL_\mu}{TOL_\mu}| \\ \\ +\hat{w}_5|\dfrac{C_\kappa^{\sigma,1}-TOL_\kappa}{TOL_\kappa}| + \hat{w}_6|\dfrac{C_\mu^{\sigma,1}-TOL_\mu}{TOL_\mu}|\end{array}}$$

where

- (I) if $C_\kappa^{\sigma,2} \leq TOL_\kappa$, then $\hat{w}_3 = 0$,
- (II) if $C_\kappa^{\sigma,2} > TOL_\kappa$, then $\hat{w}_3 = w_3$,
- (III) if $C_\mu^{\sigma,2} \leq TOL_\mu$, then $\hat{w}_4 = 0$,
- (IV) if $C_\mu^{\sigma,2} > TOL_\mu$, then $\hat{w}_4 = w_4$,
- (V) if $C_\kappa^{\sigma,1} \leq TOL_\kappa$, then $\hat{w}_5 = 0$,
- (VI) if $C_\kappa^{\sigma,1} > TOL_\kappa$, then $\hat{w}_5 = w_5$,
- (VII) if $C_\mu^{\sigma,1} \leq TOL_\mu$, then $\hat{w}_6 = 0$,
- (VIII) if $C_\mu^{\sigma,1} > TOL_\mu$, then $\hat{w}_6 = w_6$.

Here the design variables are $\mathbf{\Lambda} = \{\kappa_1, \mu_1, \kappa_2, \mu_2\, v_2^E\}$, and their constrained ranges are $\kappa_1^{(-)} \leq \kappa_1 \leq \kappa_1^{(+)}$, $\mu_1^{(-)} \leq \mu_1 \leq \mu_1^{(+)}$ $\kappa_2^{(-)} \leq \kappa_2 \leq \kappa_2^{(+)}$, $\mu_2^{(-)} \leq \mu_2 \leq \mu_2^{(+)}$ and $(v_2^E)^{(-)} \leq v_2^E \leq (v_2^E)^{(+)}$.

1.9.1 Auxilliary properties of materials

In many models for the deformation and failure material parameters such as effective Young's modulus and the effective Poisson ratio are needed. For example, the effect of an increase in the directional (extensional) stiffness of the material, $\mathbb{E}^*$. The effective moduli are inter-related by

- $\kappa^* = \frac{\mathbb{E}^*}{3(1-2\nu^*)}$,
- $\mu^* = \frac{\mathbb{E}^*}{2(1+\nu^*)}$,
- $\nu^* = \frac{3\kappa^*-2\mu^*}{2(3\kappa+\mu)}$,
- $\frac{\kappa^*}{\mu^*} = \frac{2(1+\nu^*)}{3(1-2\nu^*)}$,
- $\mathbb{E}^* = \frac{9\kappa^*\mu^*}{3\kappa^*+\mu^*}$.

1.10 Optimization of ALL PROPERTIES using Machine-Learning-Genetic algorithms

In order to optimize the overall electrical conductivity, thermal conductivity and mechanical stiffness of a mixture of material, subject to constraints on each phase, the micro-macro objective function is

$$\boxed{\Pi^{total} = W_1\Pi^{electrical} + W_2\Pi^{thermal} + W_3\Pi^{mechanical}} \tag{1.59}$$

There are two characteristics of such a formulation which make the application of standard gradient type minimization schemes, such as Newton's method, difficult:

- (I) the incorporation of limits on the microfield behavior, as well as design search space restrictions, renders the objective function not continuously differentiable in design space and
- (II) the objective function is nonconvex, i.e. the system Hessian is not positive definite (invertible) throughout design space.

1.11 Algorithmic specifics

Following Zohdi [45, 46, **??**], the algorithm is as follows:

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^i, (i = 1, 2, 3, ..., S)$: $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \Lambda_4^i, ..., \Lambda_N^i\} \stackrel{\text{def}}{=} \{\Lambda_1, \Lambda_2, \Lambda_3, ...\}$
- **STEP 2:** Compute fitness of each string $\Pi(\boldsymbol{\Lambda}^i)$, (i=1, ..., S)
- **STEP 3:** Rank genetic strings: $\boldsymbol{\Lambda}^i$, (i=1, ..., S)
- **STEP 4:** Mate nearest pairs and produce two offspring, (i=1, ..., S)
  $\boldsymbol{\lambda}^i \stackrel{\text{def}}{=} \Phi^{(I)}\boldsymbol{\Lambda}^i + (1 - \Phi^{(I)})\boldsymbol{\Lambda}^{i+1}, \quad \boldsymbol{\lambda}^{i+1} \stackrel{\text{def}}{=} \Phi^{(II)}\boldsymbol{\Lambda}^i + (1 - \Phi^{(II)})\boldsymbol{\Lambda}^{i+1}$
- **STEP 5:** Eliminate the bottom $M < S$ strings and keep top $K < N$ parents and top K offspring (K offspring+K parents+M=S)
- **STEP 6:** Repeat STEPS 1-6 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings
- **NOTE:** $\Phi^{(I)}$ and $\Phi^{(II)}$ are random numbers, such that $0 \leq \Phi^{(I)} \leq 1$, $0 \leq \Phi^{(II)} \leq 1$, which are different for each component of each genetic string
- **OPTION:** Rescale and restart search around best performing parameter set every few generations

Remark 1: If one selects the mating parameter Φ to be greater that one and/or less than zero, one can induce "mutations". i.e. characteristics that neither parent possesses. However, this is somewhat redundant with introduction of new random members of the population in the current algorithm.
Remark 2: If one does not retain the parents in the algorithm above, in is possible that inferior performing offspring may replace superior parents. Thus, top parents should be kept for the next generation. This guarantees a monotone reduction in the cost function. Furthermore, retained parents do not need to be re-evaluated-making the algorithm less computationally less expensive, since these parameter sets do not have to be reevaluated (or ranked) in the next generation. Numerous studies of the author have shown that advantages parent retention outweighs inbreeding, for sufficiently large population sizes. Finally, we remark that this algorithm is easily parallelizable.

2 Example

3 Assignment

In this assignment, you will expand upon your genetic algorithm code to find a combination of materials that achieves several objectives simultaneously. The results of this code could be used to guide early-stage experimental trials in the design of a particle-doped composite. The objective is to match a set of desired material properties while minimizing "concentration factors" that limit the effectiveness of load sharing between different phases.
In addition to adapting your GA to solve a problem with more inputs or design variables, you will also test variations on the basic GA framework used in Project 5.

3.1 Effective Properties

3.1.1 Hashin-Shtrikman Bounds

A classical approximation for effective properties is given by the Reuss-Voigt-Wiener bounds. Improved bounds were developed in 1962 by Hashin Shtrikman based on variational principles using the concept of polarization or "filtering" of micro-macro fields. The HS bounds are the tightest possible bounds for the effective material properties of isotropic mixtures with isotropic constituents where only the properties and volume fractions of the constituents are known. The HS bounds can estimate effective property y^* as follows:

$$y^{*,-} \leq y^* \leq y^{*,+} \tag{3.1}$$

In this notation:

- y is some arbitrary property (we will use several properties in this project).
- y^* represents the *effective*, or average, property in the mixture.
- $y^{*,-}$ represents the *lower bound* for this property.
- $y^{*,+}$ represents the *upper bound* for this property.

The formulas for the upper and lower bounds are not the same for all physical properties and will always be provided for you. If we know the upper and lower bounds, it must be possible to write the true effective property's value as a weighted average of the bounds:

$$y^* \approx \phi y^{*,-} + (1-\phi) y^{*,+} \stackrel{\text{def}}{=} y^{*,\phi} \tag{3.2}$$

For real-world applications, $0 \leq \phi \leq 1$ can be experimentally determined for a specific pair of materials.

- If $\phi = 0$ we have the lower bound,
- If $\phi = 1$ we have the upper bound,
- If $\phi = 1/2$ we have the average of the bounds.

If no experimental data exists, selecting $\phi = 1/2$ as a starting estimate reduces the average error.
Remark: The original proofs can be found in Hashin and Shtrikman (1962). In the derivation of the HS bounds, the body is assumed to be *infinite*, the microstructure *isotropic*, and the effective responses isotropic. The bounds are the tightest possible bounds on effective material properties of mixtures using these assumptions and without microstructural information. You are *not responsible* for showing or understanding the derivation of the HS bounds but should know what physical assumptions they rely on.
In this project, we will use the Hashin and Shtrikman (HS) bounds to estimate the mechanical, thermal and electrical properties of mixtures of isotropic materials with an overall isotropic response.

3.1.2 Mechanical Properties

The HS bounds for the effective bulk modulus of such a mixture is given by:

$$\kappa^{*,-} \stackrel{\text{def}}{=} \kappa_1 + \frac{v_2}{\frac{1}{\kappa_2-\kappa_1} + \frac{3(1-v_2)}{3\kappa_1+4\mu_1}} \leq \kappa^* \leq \kappa_2 + \frac{1-v_2}{\frac{1}{\kappa_1-\kappa_2} + \frac{3v_2}{3\kappa_2+4\mu_2}} \stackrel{\text{def}}{=} \kappa^{*,+}, \tag{3.3}$$

The analogous expression for the shear modulus is:

$$\mu^{*,-} \stackrel{\text{def}}{=} \mu_1 + \frac{v_2}{\frac{1}{\mu_2-\mu_1} + \frac{6(1-v_2)(\kappa_1+2\mu_1)}{5\mu_1(3\kappa_1+4\mu_1)}} \leq \mu^* \leq \mu_2 + \frac{(1-v_2)}{\frac{1}{\mu_1-\mu_2} + \frac{6v_2(\kappa_2+2\mu_2)}{5\mu_2(3\kappa_2+4\mu_2)}} \stackrel{\text{def}}{=} \mu^{*,+}, \tag{3.4}$$

where κ_2 and κ_1 are the bulk moduli and μ_2 and μ_1 are the shear moduli of the respective phases ($(\kappa_2 \geq \kappa_1)$ and $(\mu_2 \geq \mu_1)$), and where v_2 is the second phase volume fraction. These bounds are the tightest known on isotropic effective responses, with isotropic two-phase mixtures, where only the volume fractions and material properties of the constituents are known. As a model problem, our goal is to computationally design the macroscale effective bulk and shear moduli κ^* and μ^*, using convex combinations of the HS bounds as approximations for the effective moduli $\kappa^* \approx \phi\kappa^{*+} + (1-\phi)\kappa^{*-}$ and $\mu^* \approx \phi\mu^{*+} + (1-\phi)\mu^{*-}$, where $0 \leq \phi \leq 1$.

3.1.3 Electrical Properties

We can also use the HS bounds for the overall electrical and thermal conductivity σ_e^* and $I\!K^*$

$$\langle \sigma_e^{-1}(\mathbf{x})\rangle_{\mathbf{\Omega}}^{-\mathbf{1}} \leq \underbrace{\sigma_{1,e} + \frac{v_2}{\frac{1}{\sigma_{2,e}-\sigma_{1,e}} + \frac{1-v_2}{3\sigma_{1,e}}}}_{\sigma_{\mathbf{e}}^{*,-}} \leq \sigma_{\mathbf{e}}^* \leq \underbrace{\sigma_{2,e} + \frac{1-v_2}{\frac{1}{\sigma_{1,e}-\sigma_{2,e}} + \frac{v_2}{3\sigma_{2,e}}}}_{\sigma_{\mathbf{e}}^{*,+}} \leq \langle \sigma_{\mathbf{e}}(\mathbf{x})\rangle_{\mathbf{\Omega}} \tag{3.5}$$

where $\sigma_{2,e} \geq \sigma_{1,e}$. Note that the subscript in σ_e^* denotes "electric" and distinguishes this variable from mechanical stress.

3.1.4 Thermal Properties

The analogous expression for the effective thermal conductivity is:

$$\underbrace{I\!K_1 + \frac{v_2^K}{\frac{1}{I\!K_2-I\!K_1} + \frac{1-v_2^K}{3I\!K_1}}}_{I\!K^{*,-}} \leq I\!K^* \leq \underbrace{I\!K_2 + \frac{1-v_2^K}{\frac{1}{I\!K_1-I\!K_2} + \frac{v_2}{3I\!K_2}}}_{I\!K^{*,+}} \tag{3.6}$$

3.2 Concentration Factors

The average properties of a mixture are not the sole predictor of performance. If the mismatch of a property between two phases is very high, loads are not efficiently shared in the overall material, leading to locally elevated mechanical, thermal, or electrical loads. To predict the effectiveness of a mixture of materials, we need to estimate how extreme these load-concentrating factors are.
All load sharing or concentration terms rely on effective property estimates from the previous section.

3.2.1 Mechanical Load Sharing

The load carried by each phase in the microstructure is characterized by stress and strain concentration tensors, which we now discuss. These provide a measure of the deviation away from the mean fields throughout the material. One can decompose averages of an arbitrary quantity over Ω into averages over the each of the phases in the following manner: $\langle \boldsymbol{A}\rangle_\Omega = (1/|\Omega|)\left(\int_{\Omega_1} \boldsymbol{A}\, d\Omega + \int_{\Omega_2} \boldsymbol{A}\, d\Omega\right) = v_1\langle \boldsymbol{A}\rangle_{\Omega_1} + v_2\langle \boldsymbol{A}\rangle_{\Omega_2}$. Using this decomposition:

$$\begin{aligned}
\langle \boldsymbol{\sigma}\rangle_\Omega &= v_1\langle \boldsymbol{\sigma}\rangle_{\Omega_1} + v_2\langle \boldsymbol{\sigma}\rangle_{\Omega_2} \\
&= v_1\, I\!\boldsymbol{E}_1 : \langle \boldsymbol{\epsilon}\rangle_{\Omega_1} + v_2\, I\!\boldsymbol{E}_2 : \langle \boldsymbol{\epsilon}\rangle_{\Omega_2} \\
&= I\!\boldsymbol{E}_1 : (\langle \boldsymbol{\epsilon}\rangle_\Omega - v_2\langle \boldsymbol{\epsilon}\rangle_{\Omega_2}) + v_2\, I\!\boldsymbol{E}_2 : \langle \boldsymbol{\epsilon}\rangle_{\Omega_2} \\
&= \left(\ I\!\boldsymbol{E}_1 + v_2\ (\ I\!\boldsymbol{E}_2 - I\!\boldsymbol{E}_1) : \boldsymbol{C}^{\epsilon,2}\right) : \langle \boldsymbol{\epsilon}\rangle_\Omega,
\end{aligned} \tag{3.7}$$

where $\boldsymbol{C}^{\epsilon,2} \stackrel{\text{def}}{=} \left(\frac{1}{v_2}(\ I\!\boldsymbol{E}_2 - I\!\boldsymbol{E}_1)^{-1} : (\ I\!\boldsymbol{E}^* - I\!\boldsymbol{E}_1)\right)$ with $\boldsymbol{C}^{\epsilon,2} : \langle \boldsymbol{\epsilon}\rangle_\Omega = \langle \boldsymbol{\epsilon}\rangle_{\Omega_2}$. The strain concentration tensor $\boldsymbol{C}^{\epsilon,2}$ relates the average strain over phase 2 to the average strain over all phases. Similarly, for the variation in the stress we have $\boldsymbol{C}^{\epsilon,2} : I\!\boldsymbol{E}^{*-1} : \langle \boldsymbol{\sigma}\rangle_\Omega = I\!\boldsymbol{E}_2^{-1} : \langle \boldsymbol{\sigma}\rangle_{\Omega_2}$, which reduces to $I\!\boldsymbol{E}_2 : \boldsymbol{C}^{\epsilon,2} : I\!\boldsymbol{E}^{*-1} : \langle \boldsymbol{\sigma}\rangle_\Omega \stackrel{\text{def}}{=} \boldsymbol{C}^{\sigma,2} : \langle \boldsymbol{\sigma}\rangle_\Omega = \langle \boldsymbol{\sigma}\rangle_{\Omega_2}$. $\boldsymbol{C}^{\sigma,2}$ is known as the stress concentration tensor; it relates the average stress in the particle phase to that in the whole RVE. Note that once either $\boldsymbol{C}^{\epsilon,2}$ or $I\!\boldsymbol{E}^*$ are known, the other can be determined. In the case of isotropy we may write:

$$C_\kappa^{\sigma,2} \stackrel{\text{def}}{=} \frac{1}{v_2}\frac{\kappa_2}{\kappa^*}\frac{\kappa^*-\kappa_1}{\kappa_2-\kappa_1} \qquad \text{and} \qquad C_\mu^{\sigma,2} \stackrel{\text{def}}{=} \frac{1}{v_2}\frac{\mu_2}{\mu^*}\frac{\mu^*-\mu_1}{\mu_2-\mu_1} \tag{3.8}$$

where $C_\kappa^{\sigma,2}\langle\frac{tr\boldsymbol{\sigma}}{3}\rangle_\Omega = \langle\frac{tr\boldsymbol{\sigma}}{3}\rangle_{\Omega_2}$ and where $C_\mu^{\sigma,2}\langle\boldsymbol{\sigma'}\rangle_\Omega = \langle\boldsymbol{\sigma'}\rangle_{\Omega_2}$. Clearly, the microstress fields are minimally distorted when $C_\kappa^{\sigma,2} = C_\mu^{\sigma,2} = 1$; there are no stress concentrations in a homogeneous material. For the matrix,

$$\langle \boldsymbol{\sigma}\rangle_{\Omega_1} = \frac{\langle \boldsymbol{\sigma}\rangle_\Omega - v_2\langle \boldsymbol{\sigma}\rangle_{\Omega_2}}{v_1} = \frac{\langle \boldsymbol{\sigma}\rangle_\Omega - v_2\boldsymbol{C}^{\sigma,2} : \langle \boldsymbol{\sigma}\rangle_\Omega}{v_1} = \frac{(\mathbf{1} - v_2\boldsymbol{C}^{\sigma,2}) : \langle \boldsymbol{\sigma}\rangle_\Omega}{v_1} \stackrel{\text{def}}{=} \boldsymbol{C}^{\sigma,1} : \langle \boldsymbol{\sigma}\rangle_\Omega. \tag{3.9}$$

If the material is isotropic:

$$C_\kappa^{\sigma,1} \overset{\text{def}}{=} \frac{1}{v_1}(1 - v_2 C_\kappa^{\sigma,2}) \qquad \text{and} \qquad C_\mu^{\sigma,1} \overset{\text{def}}{=} \frac{1}{v_1}(1 - v_2 C_\mu^{\sigma,2}). \tag{3.10}$$

The fraction of the total stress carried by each phase can be determined by multiplying the concentration factors by the corresponding volume fractions:

$$\begin{aligned} \langle \boldsymbol{\sigma} \rangle_\Omega &= v_1 \langle \boldsymbol{\sigma} \rangle_{\Omega_1} + v_2 \langle \boldsymbol{\sigma} \rangle_{\Omega_2} \\ &= v_1 \boldsymbol{C}^{\sigma,1} : \langle \boldsymbol{\sigma} \rangle_\Omega + v_2 \boldsymbol{C}^{\sigma,2} : \langle \boldsymbol{\sigma} \rangle_\Omega. \end{aligned} \tag{3.11}$$

The strain in the matrix can be described by:

$$\langle \boldsymbol{\epsilon} \rangle_{\Omega_1} = \frac{\langle \boldsymbol{\epsilon} \rangle_\Omega - v_2 \langle \boldsymbol{\epsilon} \rangle_{\Omega_2}}{v_1} = \frac{\langle \boldsymbol{\epsilon} \rangle_\Omega - v_2 \boldsymbol{C}^{\epsilon,2} : \langle \boldsymbol{\epsilon} \rangle_\Omega}{v_1} = \frac{(\mathbf{1} - v_2 \boldsymbol{C}^{\epsilon,2}) : \langle \boldsymbol{\epsilon} \rangle_\Omega}{v_1} \overset{\text{def}}{=} \boldsymbol{C}^{\epsilon,1} : \langle \boldsymbol{\epsilon} \rangle_\Omega. \tag{3.12}$$

If the overall material is isotropic:

$$C_\kappa^{\epsilon,1} \overset{\text{def}}{=} \frac{1}{v_1}(1 - v_2 C_\kappa^{\epsilon,2}) \qquad \text{and} \qquad C_\mu^{\epsilon,1} \overset{\text{def}}{=} \frac{1}{v_1}(1 - v_2 C_\mu^{\epsilon,2}). \tag{3.13}$$

The fraction of the total strain in each phase can be determined by multiplying the concentration factors by their respective volume fractions:

$$\begin{aligned} \langle \boldsymbol{\epsilon} \rangle_\Omega &= v_1 \langle \boldsymbol{\epsilon} \rangle_{\Omega_1} + v_2 \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \\ &= v_1 \boldsymbol{C}^{\epsilon,1} : \langle \boldsymbol{\epsilon} \rangle_\Omega + v_2 \boldsymbol{C}^{\epsilon,2} : \langle \boldsymbol{\epsilon} \rangle_\Omega. \end{aligned} \tag{3.14}$$

3.2.2 Electrical Load Sharing

A key quantity of interest for many industrial applications is the amount of heat generated from running a current through a material, denoted H. This can be used to analyze the overall balance of energy in a system using the first law of thermodynamics:

$$\rho \dot{w} - \boldsymbol{\sigma} : \nabla \dot{\boldsymbol{u}} + \nabla \cdot q = \underbrace{H}_{\text{Joule heating source}} \tag{3.15}$$

In Equation 3.15, ρ is the mass density, w is the stored energy per unit mass, $\boldsymbol{\sigma}$ is Cauchy stress, u is the displacement field, q is heat flux, and $H = a(\mathbf{J} \cdot \mathbf{E})$, is the rate of electrical energy absorbed due to Joule heating, where $\mathbf{J}$ is the current, $\mathbf{E}$ is the electric field and $0 \leq a \leq 1$ is an absorption constant. Our objective is to determine the phase-wise load-shares of the Joule field, denoted $H = \mathbf{J} \cdot \mathbf{E}$, carried by the components in the heterogeneous mixture. The Joule heating of phase 1 is given by:

$$H_1 = \langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \langle \boldsymbol{E} \rangle_{\Omega_1} = \langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \boldsymbol{\sigma}_1^{c,-1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_1} = \frac{\langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_1}}{\sigma_1^c} \tag{3.16}$$

The Joule heating phase 2 is given by:

$$H_2 = \langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \langle \boldsymbol{E} \rangle_{\Omega_2} = \langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \boldsymbol{\sigma}_2^{c,-1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_2} = \frac{\langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \langle \boldsymbol{J} \rangle_{\Omega_2}}{\sigma_2^c} \tag{3.17}$$

These can be rewritten in terms of macroscale quantities $\big(\langle \boldsymbol{J} \rangle_\Omega$ and $\langle \boldsymbol{E} \rangle_\Omega\big)$:

$$H_1 = (\boldsymbol{C}_{J,1} \cdot \langle \boldsymbol{J} \rangle_{\Omega_1}) \cdot (\boldsymbol{C}_{E,1} \cdot \langle \boldsymbol{E} \rangle_{\Omega_1}) = C_{J,1} C_{E,1} \langle \boldsymbol{J} \rangle_{\Omega_1} \cdot \langle \boldsymbol{E} \rangle_{\Omega_1} \tag{3.18}$$

$$H_2 = (\boldsymbol{C}_{J,2} \cdot \langle \boldsymbol{J} \rangle_{\Omega_2}) \cdot (\boldsymbol{C}_{E,2} \cdot \langle \boldsymbol{E} \rangle_{\Omega_2}) = C_{J,2} C_{E,2} \langle \boldsymbol{J} \rangle_{\Omega_2} \cdot \langle \boldsymbol{E} \rangle_{\Omega_2} \tag{3.19}$$

If the overall mixture is isotropic, and each of the constituents is isotropic, then the load concentration tensors become multiples of the identity matrix **1** given by:

$$\boldsymbol{C}_{E,i} = C_{E,i}\mathbf{1}$$
$$\boldsymbol{C}_{J,i} = C_{J,i}\mathbf{1}$$

For a two-phase mixture, the concentration factors are given by:

$$C_{E,1} = \frac{1}{1-v_2}\frac{\sigma_{e,2}-\sigma_e^*}{\sigma_{e,2}-\sigma_{e,1}} \text{ and } C_{E,2} = \frac{1}{v_2}\frac{\sigma_e^*-\sigma_{e,1}}{\sigma_{e,2}-\sigma_{e,1}}, \tag{3.20}$$

$$C_{J,1} = \frac{\sigma_{e,1}}{\sigma_e^*(1-v_2)}\left(\frac{\sigma_{e,2}-\sigma_e^*}{\sigma_{e,2}-\sigma_{e,1}}\right) \text{ and } C_{J,2} = \frac{\sigma_{e,2}}{\sigma_e^* v_2}\left(\frac{\sigma_e^*-\sigma_{e,1}}{\sigma_{e,2}-\sigma_{e,1}}\right). \tag{3.21}$$

The product of the concentration factors take the following form:

$$C_{E,1}C_{J,1} = \frac{\sigma_{e,1}}{\sigma_e^*}\left(\frac{1}{(1-v_2)}\left(\frac{\sigma_{e,2}-\sigma_e^*}{\sigma_{e,2}-\sigma_{e,1}}\right)\right)^2 \tag{3.22}$$

$$C_{E,2}C_{J,2} = \frac{\sigma_{e,2}}{\sigma_e^*}\left(\frac{1}{v_2}\left(\frac{\sigma_e^*-\sigma_{e,1}}{\sigma_{e,2}-\sigma_{e,1}}\right)\right)^2. \tag{3.23}$$

3.2.3 Thermal Load Sharing

The effective thermal conductivity bounds for a two-phase mixture are given by equation **??**. Recall that the conductivity of phase 2 (with volume fraction v_2^K) is larger than that of phase 1 ($\mathbb{K}_2 \geq \mathbb{K}_1$). Usually, v_2^K corresponds to the particle material, although there can be applications where the matrix is more conductive that the particles. In that case, v_2^K would correspond to the matrix material. To obtain the concentration of the thermal fields in each phase, the exact procedure holds with $\nabla\theta$ replacing $\mathbf{E}$, $\mathbf{q}$ replacing $\mathbf{J}$, $\mathbb{K}$ replacing $\sigma^{\mathbf{c}}$, v_1^K replacing v_1^σ and v_2^K replacing v_2^σ. For thermal properties, use the following concentration tensors:

- $\boldsymbol{C}_{\theta,2}\cdot\langle\nabla\theta\rangle_\Omega = \langle\nabla\theta\rangle_{\Omega_2}$ where $\boldsymbol{C}_{\theta,2} = \frac{1}{v_2^K}(\mathbb{K}_2-\mathbb{K}_1)^{-1}\cdot(\mathbb{K}^*-\mathbb{K}_1)$
- $\boldsymbol{C}_{\theta,1}\cdot\langle\nabla\theta\rangle_\Omega = \langle\nabla\theta\rangle_{\Omega_1}$ where $\boldsymbol{C}_{\theta,1} = \frac{1}{v_1^K}(1-v_2^K\boldsymbol{C}_{\theta,2}) = \frac{1-v_2^K\boldsymbol{C}_{\theta,2}}{1-v_2^K}$
- $\boldsymbol{C}_{q,2}\cdot\langle\boldsymbol{q}\rangle_\Omega = \langle\boldsymbol{q}\rangle_{\Omega_2}$ where $\boldsymbol{C}_{q,2} = \mathbb{K}_2\cdot\boldsymbol{C}_{\theta,2}\cdot\mathbb{K}^{*-1}$
- $\boldsymbol{C}_{q,1}\cdot\langle\boldsymbol{q}\rangle_\Omega = \langle\boldsymbol{q}\rangle_{\Omega_1}$ where $\boldsymbol{C}_{q,1} = \frac{\mathbf{1}-v_2^K\boldsymbol{C}_{q,2}}{1-v_2^K}$

Optimization

With many separate goals of the project defined, all the goals must be combined into a single scalar value to be minimized by the genetic algorithm. In general, all the weights used could be distinct. For this problem, all weights will be identical for simplicity. All necessary values are defined in the variable glossary.

Optimization of Mechanical Properties

Our immediate goal is to computationally design the macroscale effective bulk and shear moduli κ^* and μ^*, using convex combinations of the Hashin-Shtrikman bounds as approximations for the effective moduli $\kappa^* \approx \phi\kappa^{*,+} + (1-\phi)\kappa^{*,-}$ and $\mu^* \approx \phi\mu^{*,+} + (1-\phi)\mu^{*,-}$, where $0 \leq \phi \leq 1$. The micro-macro objective function is:

$$\Pi^{mechanical} = w_1\left|\frac{\kappa^{*,Des}-\kappa^*}{\kappa^{*,Des}}\right| + w_1\left|\frac{\mu^{*,Des}-\mu^*}{\mu^{*,Des}}\right| + \hat{w}_3\left|\frac{C_\kappa^{\sigma,2}-TOL_\kappa}{TOL_\kappa}\right| + \hat{w}_4\left|\frac{C_\mu^{\sigma,2}-TOL_\mu}{TOL_\mu}\right|$$
$$+\hat{w}_5\left|\frac{C_\kappa^{\sigma,1}-TOL_\kappa}{TOL_\kappa}\right| + \hat{w}_6\left|\frac{C_\mu^{\sigma,1}-TOL_\mu}{TOL_\mu}\right|$$

where:

$$\hat{w}_3 = \begin{cases} w_3 & C_\kappa^{\sigma,2} > TOL_\kappa \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

$$\hat{w}_4 = \begin{cases} w_4 & C_\mu^{\sigma,2} > TOL_\mu \\ 0 & \text{otherwise} \end{cases} \tag{3.25}$$

$$\hat{w}_5 = \begin{cases} w_5 & C_\kappa^{\sigma,1} > TOL_\kappa \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

$$\hat{w}_6 = \begin{cases} w_6 & C_\mu^{\sigma,1} > TOL_\mu \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

In general, the weights on the μ and κ terms could be different. They are set to be the same here to prevent confusion with the weights for the step functions in the other components of the overall cost function.
Here the design variables are $\mathbf{\Lambda} = \{\kappa_2, \mu_2\, v_2\}$, and their constrained ranges are: $\kappa_2^{(-)} \leq \kappa_2 \leq \kappa_2^{(+)}$, $\mu_2^{(-)} \leq \mu_2 \leq \mu_2^{(+)}$ and $v_2^{(-)} \leq v_2 \leq v_2^{(+)}$. All values are specified in the glossary on the last page of this assignment.

Optimization of Electrical Properties

In order to utilize the previous results, we now provide examples of how to optimize the material properties. In particular, we will optimize the overall electrical conductivity of a mixture of material, subject to Joule heating constraints on each phase. The effective conductivity is given by σ_e^* using convex combinations of the HS bounds as approximations for the effective moduli $\sigma_e^* \approx \phi\sigma_e^{*+} + (1-\phi)\sigma_e^{*-}$, where $0 \leq \phi \leq 1$. The micro-macro objective function is:

$$\boxed{\Pi^{elec} = w_1 |\frac{\sigma_e^{*,Des} - \sigma_e^*}{\sigma_e^{*,Des}}| + \hat{w}_2 |\frac{C_{J1}C_{E1} - TOL_\sigma}{TOL_\sigma}| + \hat{w}_3 |\frac{C_{J2}C_{E2} - TOL_\sigma}{TOL_\sigma}|,} \tag{3.28}$$

where:

$$\hat{w}_2 = \begin{cases} w_2 & \frac{C_{J1}C_{E1} - TOL_\sigma}{TOL_\sigma} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.29}$$

$$\hat{w}_3 = \begin{cases} w_3 & \frac{C_{J2}C_{E2} - TOL_\sigma}{TOL_\sigma} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.30}$$

Here the design variables are $\mathbf{\Lambda} = \{\sigma_{e,2}, v_2\}$, and their constrained ranges are:

- $(\sigma_{e,2})^{(-)} \leq \sigma_{e,2} \leq (\sigma_{e,2})^{(+)}$ and
- $v_2^{(-)} \leq v_2 \leq v_2^{(+)}$.

Optimization of Thermal Properties

In order to utilize the previous results, we now provide examples of how to optimize the material properties. In particular, we will optimize the overall thermal conductivity of a multi-phase mixture, subject to thermal constraints on each phase. The effective thermal conductivity is given by $\mathbb{K}^*$ using convex combinations of the HS bounds as approximations for the effective property $\mathbb{K}^* \approx \phi\mathbb{K}^{*+} + (1-\phi)\mathbb{K}^{*-}$, where $0 \leq \phi \leq 1$. The micro-macro objective function is:

$$\boxed{\Pi^{thermal} = w_1 |\frac{\mathbb{K}^{*,Des} - \mathbb{K}^*}{\mathbb{K}^{*,Des}}| + \hat{w}_2 |\frac{C_{q1} - TOL_{\mathbb{K}}}{TOL_{\mathbb{K}}}| + \hat{w}_3 |\frac{C_{q2} - TOL_{\mathbb{K}}}{TOL_{\mathbb{K}}}|} \tag{3.31}$$

Any weight of the form w_i represents a scalar constant. Any weight of the form $\hat{w}_i$ represents a unilateral constraint, defined as a piecewise constant function or "step function."

$$\hat{w}_2 = \begin{cases} w_2 & \frac{C_{q1} - TOL_{\mathbb{K}}}{TOL_{\mathbb{K}}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.32}$$

$$\hat{w}_3 = \begin{cases} w_3 & \frac{C_{q2} - TOL_{\mathbb{K}}}{TOL_{\mathbb{K}}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.33}$$

Here the design variables are $\mathbf{\Lambda} = \{\mathbb{K}_2, v_2\}$, and their constrained ranges are

- $(\mathbb{K}_2)^{(-)} \leq \mathbb{K}_2 \leq (\mathbb{K}_2)^{(+)}$ and
- $v_2^{(-)} \leq v_2 \leq v_2^{(+)}$.

Multi-Property Optimization

We will now combine the cost functions for each material property to optimize all properties simultaneously. The micro-macro objective function is:

$$\boxed{\Pi^{total} = W_1 \Pi^{electrical} + W_2 \Pi^{thermal} + W_3 \Pi^{mechanical}} \tag{3.34}$$

There are two characteristics of such a formulation which make the application of gradient-based minimization schemes, such as Newton's method, difficult:

- (I) the incorporation of limits on the microfield behavior, as well as design search space restrictions, renders the objective function not continuously differentiable in design space.
- (II) the objective function is nonconvex, i.e. the system Hessian is not positive definite (invertible) throughout design space.

Project Instructions and Deliverables

Create a concise but complete, typed technical report written for an undergraduate student who is not in this class. Follow the outline provided with previous assignments. Provide the necessary background to understand the problem and your approach. Integrate all requested plots and tables in the results and discussion section, and explain any unusual or incorrect results.
Consider a base matrix material of fixed material values ($\kappa_1, \mu_1, \sigma_{e,1}, \mathbb{K}_1$ are all fixed). Use $\theta = 0.5$ for estimating the effective material properties. Use a genetic algorithm to find the optimal values for the second phase material parameters as well as the optimal volume fraction required to achieve desired material properties. For this genetic algorithm, your design string will be:

$$\mathbf{\Lambda}^i = \{\kappa_2, \mu_2, \sigma_{e,2}, \mathbb{K}_2, v_2\} \tag{3.35}$$

and you will be optimizing based on the cost function given in equation 3.34. Set the number of genetic strings to 100, keeping the offspring of the top 10 parents (5 pairs) after each generation. Three cases should be considered:

- Keeping the top $P = 10$ parents after each generation (10 parents, 10 offspring, 80 new random strings).
- Not keeping top $P = 10$ parents after each generation, (10 offspring, 90 new genetic strings).
- Keeping the top $P = 10$ parents after each generation, and generating new children with mutations. More specifically, instead of creating children using combinations of parents with random weights $0 \leq \phi \leq 1$, use $-.5 \leq \phi \leq 1.5$ instead. This allows for offspring to be generated outside of the hyperrectangle or "higher-dimensional box" defined by parent designs in design space.

For *each* of the 3 cases above, provide the following information.
Label your tables and figures clearly with the GA variation that produced them!

1. Run the genetic algorithm for 5000 generations. Due to the random nature of the initial parameters, you may have to run the code several times to get good convergence. There is no required tolerance, but the cost can go to zero.

2. Provide convergence plots showing the cost of the best design for each generation, the mean cost of the top 10 designs, and the mean cost of the entire population. Plots should be appropriately labelled and captioned in your report.

   These plots should be on a log-log scale to be intelligible.

3. Discuss the performance of each GA variation.

4. Report your best-performing 4 designs for all 3 cases into 3 separate tables similar to the following.

   Round data to reasonable precision. Do not copy and paste 16-digit numbers.

DESIGN	Λ_1	Λ_2	Λ_3	*etc*	Π
1					
2					
3					
4					

Table 3.1: The top 4 system parameter performers.

.

VARIABLE GLOSSARY

Symbol	Type	Units	Value	Description
κ_1	Scalar	GPa	80	Phase 1 bulk modulus
μ_1	Scalar	GPa	30	Phase 1 shear modulus
$\sigma_{1,e}$	Scalar	S/m	1.0×10^7	Phase 1 electrical conductivity
$\mathbb{K}_1$	Scalar	W/m-K	4.3	Phase 1 thermal conductivity
$\kappa^{*,D}$	Scalar	GPa	111	Desired Effective Bulk Modulus
$\mu^{*,D}$	Scalar	GPa	47	Desired Effective Shear Modulus
$\sigma_e^{*,D}$	Scalar	S/m	2.0×10^7	Desired Effective Electrical Conductivity
$\mathbb{K}^{*,D}$	Scalar	W/m-K	6.2	Desired Effective Thermal Conductivity
TOL_κ	scalar	none	0.5	Bulk Modulus Tolerance
TOL_μ	scalar	none	0.5	Shear Modulus Tolerance
$TOL_{\mathbb{K}}$	scalar	none	0.5	Thermal Tolerance
TOL_σ	scalar	none	0.8	Electrical Tolerance
κ_2^-, κ_2^+	scalar	GPa	$[\kappa_1, 10\kappa_1]$	Search bounds for Bulk Modulus
μ_2^-, μ_2^+	scalar	GPa	$[\mu_1, 10\mu_1]$	Search bounds for Shear Modulus
$\mathbb{K}_2^-, \mathbb{K}_2^+$	scalar	W/m-k	$[\mathbb{K}_1, 10\mathbb{K}_1]$	Search bounds for Thermal Conductivity
$\sigma_{2,e}^-, \sigma_{2,e}^+$	scalar	S/m	$[\sigma_{1,e}, 10\sigma_{1,e}]$	Search bounds for Electrical Conductivity
v_2^-, v_2^+	scalar	none	$[0, \frac{2}{3}]$	Search bounds for second phase volume fraction
W_1, W_2, W_3	scalar	none	$\frac{1}{3}$	Total cost function weights
w_1	Scalar	none	1	Material property matching cost function weights
$\hat{w}_j$	Scalar	none	0.5 **OR** 0	Concentration tensor weights
$w_j,\ j > 1$	Scalar	none	0.5	Concentration tensor weights
ϕ	Scalar	none	0.5	Hashin Shtrikman averaging constant

4 Solution

The solution to the assignment is encoded below.

```
%% Case1
tic %timekeeping
clear all; clc; close all; %housekeeping
% Load Variables
kappa1 = 80e9; %phase 1 bulk modulus
mu1 = 30e9; %phase 1 shear modulus
sig1e = 1e7; %phase 1 electrical conductivity
K1 = 4.3; %phase 1 thermal conductivity
kappa_des = 111e9; %desired effective bulk modulus
mu_des = 47e9; %desired effective shear modulus
sige_des = 2e7; %desired effective electrical conductivity
K_des = 6.2; %desired effective thermal conductivity
TOL_kappa = 0.5; %bulk modulus tolerance
TOL_mu = 0.5; %shear modulus tolerance
TOL_K = 0.5; %thermal tolerance
TOL_sig = 0.8; %electrical tolerance

kappa2 = [kappa1 10*kappa1]; %search bounds for bulk modulus
mu2 = [mu1 10*mu1]; %search bounds for shear modulus
K2 = [K1 10*K1]; %search bounds for thermal conductivity
sig2e = [sig1e 10*sig1e]; %search bounds for electrical conductivity
v2 = [0 2/3]; %search bounds for second phase volume fraction
W1 = 1/3; W2 = 1/3; W3 = 1/3; %total cost function weights
```

```
w1 = 1; %material property matching cost function weights
phi_HS = 0.5; %Hashin-Shtrikman averaging constant

% Genetic Algorithm

%general GA parameters
S = 100; % # of genetic strings
P = 10; % # of parents
C = 10; % # of children
%TOL = 5E-5; % error tolerance
GEN_LIMIT = 5000; % generation limit

%%Step 1: Create S design strings
Lambda = NaN(S,5);

for i = 1:S
    Lambda(i,1) = kappa2(1) + (kappa2(2)-kappa2(1))*rand; %kappa2
    Lambda(i,2) = mu2(1) + (mu2(2)-mu2(1))*rand; %mu2
    Lambda(i,3) = sig2e(1) + (sig2e(2)-sig2e(1))*rand; %sig2e
    Lambda(i,4) = K2(1) + (K2(2)-K2(1))*rand; %K2
    Lambda(i,5) = v2(1) + (v2(2)-v2(1))*rand; %v2
end

%pre-allocation
Pi = NaN(S,1);

n_gen = 0; %number of generations counter
j = 1; %while loop counter
Pi(1)= 1000; %high Pi to start while loop

for z=1:GEN_LIMIT

%     if n_gen >= GEN_LIMIT
%         break
%     end

    for k = 1:S %string loop

        kappa2_str = Lambda(k,1);%kappa2
        mu2_str = Lambda(k,2);%mu2
        sig2e_str = Lambda(k,3);%sig2e
        K2_str = Lambda(k,4);%K2
        v2_str = Lambda(k,5);%v2

        [kappa_lo] = myKappaLo(kappa1,kappa2_str,mu1,v2_str); %kappa lower HS
            bound
        [kappa_up] = myKappaUp(kappa1,kappa2_str,mu2_str,v2_str); %kappa upper
            HS bound
        [mu_lo] = myMuLo(mu1,mu2_str,kappa1,v2_str); %mu lower HS bound
        [mu_up] = myMuUp(mu1,mu2_str,kappa2_str,v2_str); %mu upper HS bound
        %electrical properties
        [sige_lo] = mySigLo(sig1e,sig2e_str,v2_str); %sig_e lower HS bound
        [sige_up] = mySigUp(sig1e,sig2e_str,v2_str); %sig_e upper HS bound
        %thermal properties
```

```
76          [K_lo] = myKLo(K1,K2_str,v2_str); %K lower HS bound
77          [K_up] = myKUp(K1,K2_str,v2_str); %K upper HS bound
78          [kappa_eff,mu_eff,sige_eff,K_eff] = myEffectiveProp(kappa_lo,kappa_up
                ,...
79              mu_lo,mu_up,sige_lo,sige_up,K_lo,K_up,phi_HS); %effective
                    properties
80          %mechanical load sharing
81          [C_K1,C_K2] = myMechLoadKappa(kappa1,kappa2_str,kappa_eff,v2_str);
82          [C_M1,C_M2] = myMechLoadMu(mu1,mu2_str,mu_eff,v2_str);
83          [C_JE1,C_JE2] = myElecLoad(sig1e,sig2e_str,sige_eff,v2_str);
84          %thermal load sharing
85          [C_q1, C_q2] = myThermLoad(K1,K2_str,K_eff,v2_str);
86          %%Step 2: Evaluate fitness of strings
87          Pi_mech = myMechCost(w1,kappa_des,mu_des,kappa_eff,mu_eff,C_K2,...
88              C_K1,C_M2,C_M1,TOL_kappa,TOL_mu); % mechanical property fitness
                    function
89          Pi_elec = myElecCost(w1,sige_des,sige_eff,C_JE1,C_JE2,TOL_sig); %
                electrical property fitness function
90          Pi_therm = myThermCost(w1,K_des,K_eff,C_q1,C_q2,TOL_K); % thermal
                property fitness function
91          Pi(k) = myTotalCost(Pi_mech,Pi_elec,Pi_therm,W1,W2,W3); %total cost
92      end
93
94      %%Step 3: Rank the strings by ascending value of Pi
95      [Pi,ind] = sort(Pi); % keep the original indices for finding associated A
            string
96      %align des strings and sol vectors in order it was sorted above
97      Lambda = Lambda(ind,:);
98
99      %%Step 4: Mate top strings
100
101 %      if n_gen>=GEN_LIMIT*3/4 % enable mutation if number of generations reach
        3/4
102 %          phi = -0.5+(1.5-(-0.5)).*rand(P,3); % calculate phi - MUTATION
        ENABLED
103 %      else
104 %          phi = rand(P,3); % calculate phi - MUTATION DISABLED
105 %      end
106     phi = rand(P,3); % calculate phi - MUTATION DISABLED
107
108     ind1 = [1:2:P]'; % child 1 index pair
109     ind2 = [2:2:P]'; % child 2 index pair
110     Lambda = vertcat(Lambda(1:P,:),phi(ind1).*Lambda(ind1,:)+(1-phi(ind1)).*
            Lambda(ind2,:),...
111         phi(ind2).*Lambda(ind2,:)+(1-phi(ind2)).*Lambda(ind1,:)); %
                concatenate c on p
112
113     %%Step 5: Generate S-2P new strings
114     Lambda_new = NaN(S-2*P,5);
115     for i = 1:S-2*P
116         Lambda_new(i,1) = kappa2(1) + (kappa2(2)-kappa2(1))*rand; %kappa2
117         Lambda_new(i,2) = mu2(1) + (mu2(2)-mu2(1))*rand; %mu2
118         Lambda_new(i,3) = sig2e(1) + (sig2e(2)-sig2e(1))*rand; %sig2e
119         Lambda_new(i,4) = K2(1) + (K2(2)-K2(1))*rand; %K2
```

```
120             Lambda_new(i,5) = v2(1) + (v2(2)-v2(1))*rand; %v2
121         end
122         Lambda = vertcat(Lambda,Lambda_new);
123 
124         %Save generation-based outputs
125         Min(j)=min(Pi);
126         Ave(j)=mean(Pi);
127         Ave_parent(j)=mean(Pi(1:P));
128 
129         n_gen = n_gen + 1; %number of generations counter
130         j = j + 1; %while loop counter
131 
132 end
133 
134 %%%% Gen Plotting %%%%%%%%%%%%%%%%%%%%%%%%
135 figure(1)
136 loglog(1:n_gen,Min,'b',1:n_gen,Ave,'r',1:n_gen,Ave_parent,'g')
137 legend('Best Design','Mean of Population','Mean of Parents')
138 xlabel('Generation')
139 ylabel('Error')
140 title('GA Results')
141 xtickformat('%.0f')
142 hold off
143 
144 %%%%Obtain Best Design Parameters%%%%%%%%%
145 Lambda_best = Lambda(1,:);
146 Error = Pi(1);
147 
148 toc; %timekeeping
149 
150 %% Case2
151 tic %timekeeping
152 clear all; clc; close all; %housekeeping
153 % Load Variables
154 kappa1 = 80e9; %phase 1 bulk modulus
155 mu1 = 30e9; %phase 1 shear modulus
156 sig1e = 1e7; %phase 1 electrical conductivity
157 K1 = 4.3; %phase 1 thermal conductivity
158 kappa_des = 111e9; %desired effective bulk modulus
159 mu_des = 47e9; %desired effective shear modulus
160 sige_des = 2e7; %desired effective electrical conductivity
161 K_des = 6.2; %desired effective thermal conductivity
162 TOL_kappa = 0.5; %bulk modulus tolerance
163 TOL_mu = 0.5; %shear modulus tolerance
164 TOL_K = 0.5; %thermal tolerance
165 TOL_sig = 0.8; %electrical tolerance
166 
167 kappa2 = [kappa1 10*kappa1]; %search bounds for bulk modulus
168 mu2 = [mu1 10*mu1]; %search bounds for shear modulus
169 K2 = [K1 10*K1]; %search bounds for thermal conductivity
170 sig2e = [sig1e 10*sig1e]; %search bounds for electrical conductivity
171 v2 = [0 2/3]; %search bounds for second phase volume fraction
172 W1 = 1/3; W2 = 1/3; W3 = 1/3; %total cost function weights
173 w1 = 1; %material property matching cost function weights
```

```
174 phi_HS = 0.5; %Hashin-Shtrikman averaging constant
175
176 % Genetic Algorithm
177
178 %general GA parameters
179 S = 100; % # of genetic strings
180 P = 10; % # of parents
181 C = 10; % # of children
182 %TOL = 5E-5; % error tolerance
183 GEN_LIMIT = 5000; % generation limit
184
185 %%Step 1: Create S design strings
186 Lambda = NaN(S,5);
187
188 for i = 1:S
189     Lambda(i,1) = kappa2(1) + (kappa2(2)-kappa2(1))*rand; %kappa2
190     Lambda(i,2) = mu2(1) + (mu2(2)-mu2(1))*rand; %mu2
191     Lambda(i,3) = sig2e(1) + (sig2e(2)-sig2e(1))*rand; %sig2e
192     Lambda(i,4) = K2(1) + (K2(2)-K2(1))*rand; %K2
193     Lambda(i,5) = v2(1) + (v2(2)-v2(1))*rand; %v2
194 end
195
196 %pre-allocation
197 Pi = NaN(S,1);
198
199 n_gen = 0; %number of generations counter
200 j = 1; %while loop counter
201 Pi(1)= 1000; %high Pi to start while loop
202
203 for z=1:GEN_LIMIT
204
205 %     if n_gen >= GEN_LIMIT
206 %         break
207 %     end
208
209     for k = 1:S %string loop
210
211         kappa2_str = Lambda(k,1);%kappa2
212         mu2_str = Lambda(k,2);%mu2
213         sig2e_str = Lambda(k,3);%sig2e
214         K2_str = Lambda(k,4);%K2
215         v2_str = Lambda(k,5);%v2
216
217         [kappa_lo] = myKappaLo(kappa1,kappa2_str,mu1,v2_str); %kappa lower HS
                bound
218         [kappa_up] = myKappaUp(kappa1,kappa2_str,mu2_str,v2_str); %kappa upper
                HS bound
219         [mu_lo] = myMuLo(mu1,mu2_str,kappa1,v2_str); %mu lower HS bound
220         [mu_up] = myMuUp(mu1,mu2_str,kappa2_str,v2_str); %mu upper HS bound
221         %electrical properties
222         [sige_lo] = mySigLo(sig1e,sig2e_str,v2_str); %sig_e lower HS bound
223         [sige_up] = mySigUp(sig1e,sig2e_str,v2_str); %sig_e upper HS bound
224         %thermal properties
225         [K_lo] = myKLo(K1,K2_str,v2_str); %K lower HS bound
```

```
226             [K_up] = myKUp(K1,K2_str,v2_str); %K upper HS bound
227             [kappa_eff,mu_eff,sige_eff,K_eff] = myEffectiveProp(kappa_lo,kappa_up
                    ,...
228                 mu_lo,mu_up,sige_lo,sige_up,K_lo,K_up,phi_HS); %effective
                        properties
229             %mechanical load sharing
230             [C_K1,C_K2] = myMechLoadKappa(kappa1,kappa2_str,kappa_eff,v2_str);
231             [C_M1,C_M2] = myMechLoadMu(mu1,mu2_str,mu_eff,v2_str);
232             [C_JE1,C_JE2] = myElecLoad(sig1e,sig2e_str,sige_eff,v2_str);
233             %thermal load sharing
234             [C_q1, C_q2] = myThermLoad(K1,K2_str,K_eff,v2_str);
235             %%Step 2: Evaluate fitness of strings
236             Pi_mech = myMechCost(w1,kappa_des,mu_des,kappa_eff,mu_eff,C_K2,...
237                 C_K1,C_M2,C_M1,TOL_kappa,TOL_mu); % mechanical property fitness
                        function
238             Pi_elec = myElecCost(w1,sige_des,sige_eff,C_JE1,C_JE2,TOL_sig); %
                    electrical property fitness function
239             Pi_therm = myThermCost(w1,K_des,K_eff,C_q1,C_q2,TOL_K); % thermal
                    property fitness function
240             Pi(k) = myTotalCost(Pi_mech,Pi_elec,Pi_therm,W1,W2,W3); %total cost
241         end
242
243         %%Step 3: Rank the strings by ascending value of Pi
244         [Pi,ind] = sort(Pi); % keep the original indices for finding associated A
                string
245         %align des strings and sol vectors in order it was sorted above
246         Lambda = Lambda(ind,:);
247
248         %%Step 4: Mate top strings
249
250 %       if n_gen>=GEN_LIMIT*3/4 % enable mutation if number of generations reach
            3/4
251 %           phi = -0.5+(1.5-(-0.5)).*rand(P,3); % calculate phi - MUTATION
        ENABLED
252 %       else
253 %           phi = rand(P,3); % calculate phi - MUTATION DISABLED
254 %       end
255         phi = rand(P,3); % calculate phi - MUTATION DISABLED
256
257         ind1 = [1:2:P]'; % child 1 index pair
258         ind2 = [2:2:P]'; % child 2 index pair
259         Lambda = vertcat(phi(ind1).*Lambda(ind1,:)+(1-phi(ind1)).*Lambda(ind2,:)
                ,...
260             phi(ind2).*Lambda(ind2,:)+(1-phi(ind2)).*Lambda(ind1,:)); %
                concatenate c on p
261
262         %%Step 5: Generate S-2P new strings
263         Lambda_new = NaN(S-P,5);
264         for i = 1:S-P
265             Lambda_new(i,1) = kappa2(1) + (kappa2(2)-kappa2(1))*rand; %kappa2
266             Lambda_new(i,2) = mu2(1) + (mu2(2)-mu2(1))*rand; %mu2
267             Lambda_new(i,3) = sig2e(1) + (sig2e(2)-sig2e(1))*rand; %sig2e
268             Lambda_new(i,4) = K2(1) + (K2(2)-K2(1))*rand; %K2
269             Lambda_new(i,5) = v2(1) + (v2(2)-v2(1))*rand; %v2
```

```
270         end
271         Lambda = vertcat(Lambda,Lambda_new);
272
273         %Save generation-based outputs
274         Min(j)=min(Pi);
275         Ave(j)=mean(Pi);
276         Ave_parent(j)=mean(Pi(1:P));
277
278         n_gen = n_gen + 1; %number of generations counter
279         j = j + 1; %while loop counter
280
281     end
282
283     %%%% Gen Plotting %%%%%%%%%%%%%%%%%%%%%%%%%
284     figure(2)
285     loglog(1:n_gen,Min,'b',1:n_gen,Ave,'r',1:n_gen,Ave_parent,'g')
286     legend('Best Design','Mean of Population','Mean of Parents')
287     xlabel('Generation')
288     ylabel('Error')
289     title('GA Results')
290     xtickformat('%.0f')
291     hold off
292
293     %%%%Obtain Best Design Parameters%%%%%%%%%%
294     Lambda_best = Lambda(1,:);
295     Error = Pi(1);
296
297     toc; %timekeeping
298
299     %% Case3
300     tic %timekeeping
301     clear all; clc; close all; %housekeeping
302     % Load Variables
303     kappa1 = 80e9; %phase 1 bulk modulus
304     mu1 = 30e9; %phase 1 shear modulus
305     sig1e = 1e7; %phase 1 electrical conductivity
306     K1 = 4.3; %phase 1 thermal conductivity
307     kappa_des = 111e9; %desired effective bulk modulus
308     mu_des = 47e9; %desired effective shear modulus
309     sige_des = 2e7; %desired effective electrical conductivity
310     K_des = 6.2; %desired effective thermal conductivity
311     TOL_kappa = 0.5; %bulk modulus tolerance
312     TOL_mu = 0.5; %shear modulus tolerance
313     TOL_K = 0.5; %thermal tolerance
314     TOL_sig = 0.8; %electrical tolerance
315
316     kappa2 = [kappa1 10*kappa1]; %search bounds for bulk modulus
317     mu2 = [mu1 10*mu1]; %search bounds for shear modulus
318     K2 = [K1 10*K1]; %search bounds for thermal conductivity
319     sig2e = [sig1e 10*sig1e]; %search bounds for electrical conductivity
320     v2 = [0 2/3]; %search bounds for second phase volume fraction
321     W1 = 1/3; W2 = 1/3; W3 = 1/3; %total cost function weights
322     w1 = 1; %material property matching cost function weights
323     phi_HS = 0.5; %Hashin-Shtrikman averaging constant
```

```
324
325 % Genetic Algorithm
326
327 %general GA parameters
328 S = 100; % # of genetic strings
329 P = 10; % # of parents
330 C = 10; % # of children
331 %TOL = 5E-5; % error tolerance
332 GEN_LIMIT = 5000; % generation limit
333
334 %%Step 1: Create S design strings
335 Lambda = NaN(S,5);
336
337 for i = 1:S
338     Lambda(i,1) = kappa2(1) + (kappa2(2)-kappa2(1))*rand; %kappa2
339     Lambda(i,2) = mu2(1) + (mu2(2)-mu2(1))*rand; %mu2
340     Lambda(i,3) = sig2e(1) + (sig2e(2)-sig2e(1))*rand; %sig2e
341     Lambda(i,4) = K2(1) + (K2(2)-K2(1))*rand; %K2
342     Lambda(i,5) = v2(1) + (v2(2)-v2(1))*rand; %v2
343 end
344
345 %pre-allocation
346 Pi = NaN(S,1);
347
348 n_gen = 0; %number of generations counter
349 j = 1; %while loop counter
350 Pi(1)= 1000; %high Pi to start while loop
351
352 for z=1:GEN_LIMIT
353
354 %     if n_gen >= GEN_LIMIT
355 %         break
356 %     end
357
358     for k = 1:S %string loop
359
360         kappa2_str = Lambda(k,1);%kappa2
361         mu2_str = Lambda(k,2);%mu2
362         sig2e_str = Lambda(k,3);%sig2e
363         K2_str = Lambda(k,4);%K2
364         v2_str = Lambda(k,5);%v2
365
366         [kappa_lo] = myKappaLo(kappa1,kappa2_str,mu1,v2_str); %kappa lower HS
              bound
367         [kappa_up] = myKappaUp(kappa1,kappa2_str,mu2_str,v2_str); %kappa upper
              HS bound
368         [mu_lo] = myMuLo(mu1,mu2_str,kappa1,v2_str); %mu lower HS bound
369         [mu_up] = myMuUp(mu1,mu2_str,kappa2_str,v2_str); %mu upper HS bound
370         %electrical properties
371         [sige_lo] = mySigLo(sig1e,sig2e_str,v2_str); %sig_e lower HS bound
372         [sige_up] = mySigUp(sig1e,sig2e_str,v2_str); %sig_e upper HS bound
373         %thermal properties
374         [K_lo] = myKLo(K1,K2_str,v2_str); %K lower HS bound
375         [K_up] = myKUp(K1,K2_str,v2_str); %K upper HS bound
```

```
376             [kappa_eff,mu_eff,sige_eff,K_eff] = myEffectiveProp(kappa_lo,kappa_up
                    ,...
377                 mu_lo,mu_up,sige_lo,sige_up,K_lo,K_up,phi_HS); %effective
                        properties
378             %mechanical load sharing
379             [C_K1,C_K2] = myMechLoadKappa(kappa1,kappa2_str,kappa_eff,v2_str);
380             [C_M1,C_M2] = myMechLoadMu(mu1,mu2_str,mu_eff,v2_str);
381             [C_JE1,C_JE2] = myElecLoad(sig1e,sig2e_str,sige_eff,v2_str);
382             %thermal load sharing
383             [C_q1, C_q2] = myThermLoad(K1,K2_str,K_eff,v2_str);
384             %%Step 2: Evaluate fitness of strings
385             Pi_mech = myMechCost(w1,kappa_des,mu_des,kappa_eff,mu_eff,C_K2,...
386                 C_K1,C_M2,C_M1,TOL_kappa,TOL_mu); % mechanical property fitness
                        function
387             Pi_elec = myElecCost(w1,sige_des,sige_eff,C_JE1,C_JE2,TOL_sig); %
                    electrical property fitness function
388             Pi_therm = myThermCost(w1,K_des,K_eff,C_q1,C_q2,TOL_K); % thermal
                    property fitness function
389             Pi(k) = myTotalCost(Pi_mech,Pi_elec,Pi_therm,W1,W2,W3); %total cost
390         end
391
392         %%Step 3: Rank the strings by ascending value of Pi
393         [Pi,ind] = sort(Pi); % keep the original indices for finding associated A
                string
394         %align des strings and sol vectors in order it was sorted above
395         Lambda = Lambda(ind,:);
396
397         %%Step 4: Mate top strings
398
399 %       if n_gen>=GEN_LIMIT*3/4 % enable mutation if number of generations reach
            3/4
400 %           phi = -0.5+(1.5-(-0.5)).*rand(P,3); % calculate phi - MUTATION
        ENABLED
401 %       else
402 %           phi = rand(P,3); % calculate phi - MUTATION DISABLED
403 %       end
404         phi = -0.5+(1.5-(-0.5)).*rand(P,3); % calculate phi - MUTATION ENABLED
405
406         ind1 = [1:2:P]'; % child 1 index pair
407         ind2 = [2:2:P]'; % child 2 index pair
408         Lambda = vertcat(Lambda(1:P,:),phi(ind1).*Lambda(ind1,:)+(1-phi(ind1)).*
                Lambda(ind2,:),...
409             phi(ind2).*Lambda(ind2,:)+(1-phi(ind2)).*Lambda(ind1,:)); %
                    concatenate c on p
410
411         %%Step 5: Generate S-2P new strings
412         Lambda_new = NaN(S-2*P,5);
413         for i = 1:S-2*P
414             Lambda_new(i,1) = kappa2(1) + (kappa2(2)-kappa2(1))*rand; %kappa2
415             Lambda_new(i,2) = mu2(1) + (mu2(2)-mu2(1))*rand; %mu2
416             Lambda_new(i,3) = sig2e(1) + (sig2e(2)-sig2e(1))*rand; %sig2e
417             Lambda_new(i,4) = K2(1) + (K2(2)-K2(1))*rand; %K2
418             Lambda_new(i,5) = v2(1) + (v2(2)-v2(1))*rand; %v2
419         end
```

```
420         Lambda = vertcat(Lambda,Lambda_new);
421
422         %Save generation-based outputs
423         Min(j)=min(Pi);
424         Ave(j)=mean(Pi);
425         Ave_parent(j)=mean(Pi(1:P));
426
427         n_gen = n_gen + 1; %number of generations counter
428         j = j + 1; %while loop counter
429
430     end
431
432     %%%% Gen Plotting %%%%%%%%%%%%%%%%%%%%%%%%
433     figure(3)
434     loglog(1:n_gen,Min,'b',1:n_gen,Ave,'r',1:n_gen,Ave_parent,'g')
435     legend('Best Design','Mean of Population','Mean of Parents')
436     xlabel('Generation')
437     ylabel('Error')
438     title('GA Results')
439     xtickformat('%.0f')
440     hold off
441
442     %%%%Obtain Best Design Parameters%%%%%%%%%%
443     Lambda_best = Lambda(1,:);
444     Error = Pi(1);
445
446     toc; %timekeeping
447     %% Functions
448
449     %general functions
450     function [kappa_eff,mu_eff,sige_eff,K_eff] = myEffectiveProp(kappa_lo,kappa_up
        ,...
451         mu_lo,mu_up,sige_lo,sige_up,K_lo,K_up,phi) %effective properties
452
453     kappa_eff = phi*kappa_lo + (1 - phi) * kappa_up;
454     mu_eff = phi*mu_lo + (1 - phi) * mu_up;
455     sige_eff = phi*sige_lo + (1 - phi) * sige_up;
456     K_eff = phi*K_lo + (1 - phi) * K_up;
457
458     end
459
460     %mechanical properties
461     function [kappa_lo] = myKappaLo(kappa1,kappa2,mu1,v2) %kappa lower HS bound
462
463     kappa_lo = kappa1 + v2 / (1/(kappa2-kappa1) + (3*(1-v2))/(3*kappa1+4*mu1));
464
465     end
466
467     function [kappa_up] = myKappaUp(kappa1,kappa2,mu2,v2) %kappa upper HS bound
468
469     kappa_up = kappa2 + (1-v2) / (1/(kappa1-kappa2) + (3*v2)/(3*kappa2+4*mu2));
470
471     end
472
```

```
473 function [mu_lo] = myMuLo(mu1,mu2,kappa1,v2) %mu lower HS bound
474
475 mu_lo = mu1 + v2 / (1/(mu2-mu1) + (6*(1-v2)*(kappa1+2*mu1))/(5*mu1*(3*kappa1
        +4*mu1)));
476
477 end
478
479 function [mu_up] = myMuUp(mu1,mu2,kappa2,v2) %mu upper HS bound
480
481 mu_up = mu2 + (1-v2) / (1/(mu1-mu2) + (6*v2*(kappa2+2*mu2))/(5*mu2*(3*kappa2
        +4*mu2)));
482
483 end
484
485 %electrical properties
486 function [sige_lo] = mySigLo(sig1e,sig2e,v2) %sig_e lower HS bound
487
488 sige_lo = sig1e + v2 / (1/(sig2e-sig1e) + (1-v2)/(3*sig1e));
489
490 end
491
492 function [sige_up] = mySigUp(sig1e,sig2e,v2) %sig_e upper HS bound
493
494 sige_up = sig2e + (1-v2) / (1/(sig1e-sig2e) + v2/(3*sig2e));
495
496 end
497
498 %thermal properties
499 function [K_lo] = myKLo(K1,K2,v2) %K lower HS bound
500
501 K_lo = K1 + v2 / (1/(K2-K1) + (1-v2)/(3*K1));
502
503 end
504
505 function [K_up] = myKUp(K1,K2,v2) %K upper HS bound
506
507 K_up = K2 + (1-v2) / (1/(K1-K2) + v2/(3*K2));
508
509 end
510
511 %mechanical load sharing
512 function [C_K1,C_K2] = myMechLoadKappa(kappa1,kappa2,kappa_eff,v2)
513
514 v1 = 1 - v2;
515 C_K2 = (1/v2)*(kappa2/kappa_eff)*(kappa_eff-kappa1)/(kappa2-kappa1);
516 C_K1 = (1 / v1) * (1 - v2 * C_K2);
517
518 end
519
520 function [C_M1,C_M2] = myMechLoadMu(mu1,mu2,mu_eff,v2)
521
522 v1 = 1 - v2;
523 C_M2 = (1/v2)*(mu2/mu_eff)*(mu_eff-mu1)/(mu2-mu1);
524 C_M1 = (1 / v1) * (1 - v2 * C_M2);
```

```
525
526 end
527
528 function [C_JE1,C_JE2] = myElecLoad(sig1e,sig2e,sige_eff,v2)
529
530 C_JE1 = (sig1e/sige_eff)*(1/(1-v2)*((sig2e-sige_eff)/(sig2e-sig1e)))^2;
531 C_JE2 = (sig2e/sige_eff)*(1/v2*((sige_eff-sig1e)/(sig2e-sig1e)))^2;
532
533 end
534
535 %thermal load sharing
536 function [C_q1, C_q2] = myThermLoad(K1,K2,K_eff,v2)
537
538 C_theta2 = (1/v2)*inv(K2-K1)*(K_eff-K1);
539 C_q2 = K2 * C_theta2 * inv(K_eff);
540 C_q1 = (1 - v2 * C_q2) / (1 - v2);
541
542 end
543
544 %cost functions
545 function Pi_mech = myMechCost(w1,kappa_des,mu_des,kappa_eff,mu_eff,C_K2,...
546     C_K1,C_M2,C_M1,TOL_kappa,TOL_mu) % mechanical property fitness function
547
548 if C_K2 > TOL_kappa
549     w3 = 0.5;
550 else
551     w3 = 0;
552 end
553
554 if C_M2 > TOL_mu
555     w4 = 0.5;
556 else
557     w4 = 0;
558 end
559
560 if C_K1 > TOL_kappa
561     w5 = 0.5;
562 else
563     w5 = 0;
564 end
565
566 if C_M1 > TOL_mu
567     w6 = 0.5;
568 else
569     w6 = 0;
570 end
571
572 Pi_mech = w1 * abs((kappa_des - kappa_eff)/kappa_des) + w1 * abs((mu_des -
    mu_eff)/mu_des)...
573     + w3 * abs((C_K2-TOL_kappa)/TOL_kappa) + w4 * abs((C_M2-TOL_mu)/TOL_mu) +
         ...
574     + w5 * abs((C_K1-TOL_kappa)/TOL_kappa) + w6 * abs((C_M1-TOL_mu)/TOL_mu);
575
576 end
```

```
577
578 function Pi_elec = myElecCost(w1,sige_des,sige_eff,C_JE1,C_JE2,TOL_sig) % electrical property fitness function
579
580 if (C_JE1 - TOL_sig)/TOL_sig > 0
581     w2 = 0.5;
582 else
583     w2 = 0;
584 end
585
586 if (C_JE2 - TOL_sig)/TOL_sig > 0
587     w3 = 0.5;
588 else
589     w3 = 0;
590 end
591
592 Pi_elec = w1 * abs((sige_des - sige_eff)/sige_des) + w2 * abs((C_JE1 - TOL_sig)/TOL_sig)...
593     + w3 * abs((C_JE2 - TOL_sig)/TOL_sig);
594
595 end
596
597 function Pi_therm = myThermCost(w1,K_des,K_eff,C_q1,C_q2,TOL_K) % thermal property fitness function
598
599 if (C_q1 - TOL_K)/TOL_K > 0
600     w2 = 0.5;
601 else
602     w2 = 0;
603 end
604
605 if (C_q2 - TOL_K)/TOL_K > 0
606     w3 = 0.5;
607 else
608     w3 = 0;
609 end
610
611 Pi_therm = w1 * abs((K_des - K_eff)/K_des) + w2 * abs((C_q1 - TOL_K)/TOL_K)...
612     + w3 * abs((C_q2 - TOL_K)/TOL_K);
613
614 end
615
616 function Pi_tot = myTotalCost(Pi_mech,Pi_elec,Pi_therm,W1,W2,W3) %total cost
617
618 Pi_tot = W1 * Pi_mech + W2 * Pi_elec + W3 * Pi_therm;
619
620 end
```

5 References

1. Zohdi, T. I. and Dornfeld D. A. (2015) Future Synergy between Computational Mechanics and Advanced Additive Manufacturing. US National Academies Report:

   http://sites.nationalacademies.org/cs/groups/pgasite/documents/webpage/pga$_1$66813*.pdf*

2. Huang, Y. Leu, M. C., Mazumdar, J. and Donmez, A. (2015). Additive manufacturing: current state, future potential, gaps and needs, and recommendation. *Journal of Manufacturing Science and Engineering.* vol 137, 014001-1.

3. Hunt, K. H. (1979). Kinematic Geometry of Mechanisms, Oxford Engineering Science Series.

4. Hartenberg, R. S. and Denavit, J. (1964). Kinematic Synthesis of Linkages, McGraw-Hill, New York.

5. Howell, L. L. (2001). Compliant mechanisms, John Wiley & Sons.

6. McCarthy, J. M. and Soh, G. S. (2010). Geometric Design of Linkages, Springer, New York.

7. McCarthy, J. M. (1990). Introduction to Theoretical Kinematics, MIT Press, Cambridge, MA.

8. Reuleaux, F., (1876). The Kinematics of Machinery, (trans. and annotated by A. B. W. Kennedy), reprinted by Dover, New York (1963).

9. Sandor, G.N. and Erdman,A.G. (1984). Advanced Mechanism Design: Analysis and Synthesis, Vol. 2. Prentice-Hall, Englewood Cliffs, NJ.

10. Slocum, A. (1992). Precision Machine Design, SME

11. Suh, C. H. and Radcliffe, C. W. (1978). Kinematics and Mechanism Design, John Wiley and Sons, New York.

12. Uicker, J. J., Pennock, G. R. and Shigley, J. E. (2003). Theory of Machines and Mechanisms, Oxford University Press, New York.

13. Papageorgiou, D. T. (1995). On the breakup of viscous liquid threads. Physics of Fluids. 7 (7): 1529-1521.

14. Eggers, J. (1997). Nonlinear dynamics and breakup of free-surface flows. Reviews of Modern Physics. 69 (3): 865.

15. Zohdi, T. I. (2017). Laser-induced heating of dynamic depositions in additive manufacturing. Comp. Meth. in Appl. Mech. and Eng. https://doi.org/10.1016/j.cma.2017.11.003

16. Zohdi, T. I. (2017).On simple scaling laws for pumping fluids with electrically-charged particles. Int. Journal of Eng. Sci: https://doi.org/10.1016/j.ijengsci.2017.11.003

17. Jackson, J. D. (1998). Classical Electrodynamics. Third Edition. Wiley.

18. Chow, C. Y. (1980). An introduction to computational fluid dynamics. New York, Wiley.

19. Schlichtling, H. (1979). Boundary-layer theory. 7th edition. New York: McGraw-Hill.

20. Whitaker, S. (1972) Forced convection heat transfer correlations for flow in pipes, past flat plates, single cylinders, single spheres, and flow in packed beds and tube bundles. AIChE J. 18, 361–371.

21. Hashin, Z. and Shtrikman, S. (1962) On some variational principles in anisotropic and nonhomogeneous elasticity. Journal of the Mechanics and Physics of Solids. **10**, 335-342.

22. Hashin, Z. and Shtrikman, S. (1963) A variational approach to the theory of the elastic behaviour of multiphase materials. Journal of the Mechanics and Physics of Solids. **11**, 127-140.

23. Hashin, Z. (1983) Analysis of composite materials: a survey. ASME Journal of Applied Mechanics. **50**, 481-505.

24. Torquato, S. (2002) Random Heterogeneous Materials: Microstructure and Macroscopic Properties Springer-Verlag, New York.

25. Jikov, V. V., Kozlov, S. M., Olenik, O. A.(1994) Homogenization of differential operators and integral functionals. Springer-Verlag.

26. Mura, T. (1993) Micromechanics of defects in solids, 2nd edition. Kluwer Academic Publishers.

27. Markov, K. Z. 2000. Elementary micromechanics of heterogeneous media. In Heterogeneous Media: Micromechanics Modeling Methods and Simulations (K. Z. Markov, and L. Preziozi, Eds.), pp. 1 162. Birkhauser, Boston.

28. Ghosh, S. (2011). Micromechanical Analysis and Multi-Scale Modeling Using the Voronoi Cell Finite Element Method. CRC Press/Taylor & Francis.

29. Ghosh, S. and Dimiduk, D. (2011). Computational Methods for Microstructure-Property Relations. Springer NY.

30. Matous, K. Geers, M., Kouznetsova, V. G. and Gillman, A. (in press). A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *Journal of Computational Physics.*

31. Zohdi, T. I. and Wriggers, P. (*Book*, 2005, 2008) Introduction to computational micromechanics. Second Reprinting (*Peer Reviewed*). Springer-Verlag.

32. Zohdi, T. I. (*Book*, 2012). Electromagnetic properties of multiphase dielectrics. A primer on modeling, theory and computation. Springer-Verlag.

33. Zohdi, T. I. (2002). An adaptive-recursive staggering strategy for simulating multifield coupled processes in microheterogeneous solids. *The International Journal of Numerical Methods in Engineering.* **53**, 1511-1532.

34. Zohdi, T. I. (2004). Modeling and simulation of a class of coupled thermo-chemo-mechanical processes in multiphase solids. *Computer Methods in Applied Mechanics and Engineering.* Vol. 193/6-8 679-699.

35. Zohdi, T. I. (2004). Modeling and direct simulation of near-field granular flows. *The International Journal of Solids and Structures.* Vol 42/2 pp 539-564.

36. Zohdi, T. I. (2005). Charge-induced clustering in multifield particulate flow *The International Journal of Numerical Methods in Engineering.* Volume 62, Issue 7, Pages 870-898

37. Zohdi, T. I. (2007). Computation of strongly coupled multifield interaction in particle-fluid systems. *Computer Methods in Applied Mechanics and Engineering.* Volume 196, 3927-3950.

38. Zohdi, T. I. (2010) On the dynamics of charged electromagnetic particulate jets. *Archives of Computational Methods in Engineering.* Volume 17, Number 2, 109-135

39. Zohdi, T. I. (2010) Simulation of coupled microscale multiphysical-fields in particulate-doped dielectrics with staggered adaptive FDTD. *Computer Methods in Applied Mechanics and Engineering.* Volume 199, 79-101.

40. Zohdi, T. I. (2013) Numerical simulation of charged particulate cluster-droplet impact on electrified surfaces. *Journal of Computational Physics.* 233, 509-526.

41. Zohdi, T. I. (2014) Additive particle deposition and selective laser processing-a computational manufacturing framework. *Computational Mechanics.* Vol 54, 171-191.

42. Zohdi, T. I. (2014) Embedded electromagnetically sensitive particle motion in functionalized fluids. *Computational Particle Mechanics.* Vol 1, 27-45.

43. Zohdi, T. I. (2015). Modeling and simulation of cooling-induced residual stresses in heated particulate mixture depositions. *Computational Mechanics.* Volume 56, 613-630.

44. Zohdi, T. I. (2015). Modeling and efficient simulation of the deposition of particulate flows onto compliant substrates. *The International Journal of Engineering Science.* Volume 99, 74-91. doi:10.1016/j.ijengsci.2015.10.012

45. Zohdi, T. I. (2003). Genetic design of solids possessing a random-particulate microstructure. *Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences.* Vol: 361, No: 1806, 1021-1043.

46. Zohdi, T. I. (2017). Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning. Computer Methods in Applied Mechanics and Engineering. https://doi.org/10.1016/j.cma.2017.11.030

47. Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor, Mich. University of Michigan Press.

48. Goldberg, D. E. (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley.

49. Davis, L (1991). Handbook of Genetic Algorithms. Thompson Computer Press.

50. Onwubiko, C. (2000) Introduction to engineering design optimization. Prentice Hall.

51. Lagaros, N., Papadrakakis, M. and Kokossalakis, G. (2002). Structural optimization using evolutionary algorithms. *Computers & Structures.* 80, 571-589.

52. Papadrakakis, M., Lagaros, N., Thierauf, G. and Cai, J. (1998a). Advanced solution methods in structural optimisation using evolution strategies, *Engineering Computational Journal.* **15** (1), 12-34.

53. Papadrakakis, M., Lagaros, N. and Tsompanakis, Y. (1998b). Structural optimization using evolution strategies and neutral networks, *Computer Methods in Applied Mechanics and Engineering.* **156**, (1), 309-335.

54. Papadrakakis, M., Lagaros, N. and Tsompanakis, Y. (1999a). Optimization of large-scale 3D trusses using Evolution Strategies and Neural Networks. *Int. J. Space Structures.* **14** (3), 211-223.

55. Papadrakakis, M., Tsompanakis, J. and Lagaros, N. (1999b). Structural shape optimisation using evolution strategies. *Eng. Optimization.* **31**, 515-540.

56. Goldberg, D. E. and Deb, K. (2000) Special issue on Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering. 186 (2-4) 121-124.

57. Thomas, Daniel (24 February 2017). Could 3D bioprinted tissues offer future hope for microtia treatment?. International Journal of Surgery. Retrieved 24 February 2017.

58. Nakashima, Yasuharu; Okazak, Ken; Nakayama, Koichiet; Okada, Seiji; Mizu-uchi, Hideki (January 2017). Bone and Joint Diseases in Present and Future. Fukuoka Igaku Zasshi = Hukuoka Acta Medica. 108 (1): 1-7. ISSN 0016-254X. PMID 29226660.

59. Doyle, Ken (15 May 2014). Bioprinting: From patches to parts. Gen. Eng. Biotechnol. News. 34 (10): 1, 34-35. doi:10.1089/gen.34.10.02.

60. US patent 7051654, Boland, Thomas; Wilson, Jr., William Crisp; Xu, Tao, Ink-jet printing of viable cells, issued 2006-05-30

61. Shafiee, Ashkan; Atala, Anthony (2016-03-01). Printing Technologies for Medical Applications. Trends in Molecular Medicine. 22 (3): 254-265. doi:10.1016/j.molmed.2016.01.003.

62. Ozbolat, Ibrahim T. (2015-07-01). Bioprinting scale-up tissue and organ constructs for transplantation. Trends in Biotechnology. 33 (7): 395-400. doi:10.1016/j.tibtech.2015.04.005.

63. Chua, C.K.; Yeong, W.Y. (2015). Bioprinting: Principles and Applications. Singapore: World Scientific Publishing Co. p. 296. ISBN 9789814612104. Retrieved 17 February 2016.

64. Harmon, K. (2013). A sweet solution for replacing organs (PDF). Scientific American. 308 (4): 54-55. doi:10.1038/scientificamerican0413-54. Retrieved 17 February 2016.

65. Murphy, Sean; Atala, Anthony (August 5, 2014). 3D bioprinting of tissues and organs. Nature Biotechnology. 32: 773-85. doi:10.1038/nbt.2958. PMID 25093879.

66. Crawford, M. (May 2013). Creating Valve Tissue Using 3-D Bioprinting. ASME.org. American Society of Mechanical Engineers. Retrieved 17 February 2016.

67. Murphy, S.V.; Skardal, A.; Atala, A. (2013). Evaluation of hydrogels for bio-printing applications. Journal of Biomedical Materials Research Part A. 101A (1): 272-284. doi:10.1002/jbm.a.34326. PMID 22941807.

68. Souza, G. R. et al Three-dimensional tissue culture based on magnetic cell levitation. Nat. Nanotechnol. 5, 291-296 (2010)

69. Haisler, W. L. et al. Three-dimensional cell culturing by magnetic levitation. Nat. Protoc. 8, 1940-1949 (2013)

70. Friedrich, J., Seidel, C., Ebner, R. & Kunz-Schughart, L. A. Spheroid-based drug screen: considerations and practical approach. Nat. Protoc. 4, 309-324 (2009)

71. Seiler, A. E. M. & Spielmann, H. The validated embryonic stem cell test to predict embryotoxicity in vitro. Nat. Protoc. 6, 961-978 (2011)

72. Tseng, H. et al.Assembly of a three-dimensional multitype bronchiole coculture model using magnetic levitation. Tissue Eng. Part C. Methods 19, 665-675 (2013)

73. Tseng, H. et al. A three-dimensional co-culture model of the aortic valve using magnetic levitation. Acta Biomater. 10, 173-182 (2014)

74. Daquinag, A. C., Souza, G. R. & Kolonin, M. G, Adipose tissue engineering in three-dimensional levitation tissue culture system based on magnetic nanoparticles. Tissue Eng. Part C. Methods 19, 336-344 (2013)

75. Timm, D. M. et al. A high-throughput three-dimensional cell migration assay for toxicity screening with mobile device-based macroscopic image analysis. Sci. Rep. 3, 3000 (2013)

76. Gwathmey, J. K., Tsaioun, K. & Hajjar, R. J. Cardionomics: a new integrative approach for screening cardiotoxicity of drug candidates. Expert Opin. Drug Metab. Toxicol. 5, 647-660 (2009)

77. Feynman, Richard P.; Leighton, Robert B.; Sands, Matthew (2006). The Feynman Lectures on Physics 2. ISBN 0-8053-9045-6.

78. Cullity, C. D. Graham (2008). Introduction to Magnetic Materials (2 ed.). Wiley-IEEE Press. p. 103. ISBN 0-471-47741-9.

79. Boyer, Timothy H. (1988). The Force on a Magnetic Dipole. American Journal of Physics 56 (8): 688-692. Bibcode:1988AmJPh.56.688B. doi:10.1119/1.15501.

80. Zienkiewicz, O. C. (1984). Coupled problems & their numerical solution, in R. W. Lewis, P. Bettes & E. Hinton eds *Numerical methods in coupled systems* Wiley, Chichester, 35–58

81. Zienkiewicz, O. C., Paul, D. K. and Chan, A. H. C. (1988). Unconditionally stable staggered solution procedure for soil-pore fluid interaction problems. *The International Journal of Numerical Methods in Engineering.* **26**, 1039-1055.

82. Lewis, R. W., Schrefler, B. A. and Simoni, L (1992) Coupling versus uncoupling in soil consolidation. Int. J. Num. Anal. Metho. Geomech. **15**, 533-548.

83. Lewis, R. W. and Schrefler, B. A. (1998) The finite element method in the static and dynamic deformation and consolidation of porous media. 2nd edition. Wiley press.

84. Park, K. C. & Felippa, C. A. (1983). Partitioned analysis of coupled systems. In *Computational methods for transient analysis.* T. Belytschko & T. J. R. Hughes, editors.

85. Farhat, C., Lesoinne, M. and Maman, N. (1995). Mixed Explicit/Implicit Time Integration of Coupled Aeroelastic Problems: Three-Field Formulation, Geometric Conservation and Distributed Solution. *International Journal for Numerical Methods in Fluids,* Vol. 21, pp. 807-835.

86. Farhat, C. and Lesoinne, M. (2000). Two Efficient Staggered Procedures for the Serial and Parallel Solution of Three-Dimensional Nonlinear Transient Aeroelastic Problems. *Computer Methods in Applied Mechanics and Engineering*, Vol. 182, pp. 499-516.

87. Farhat, C., van der Zee, G. & Geuzaine, P. (2006). Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics & Engineering.* **195**, 1973-2001.

88. Piperno, S. (1997). Explicit/implicit fluid/structure staggered procedures with a structural predictor & fluid subcycling for 2D inviscid aeroelastic simulations. *Int. J. Num. Meth. Fluids.* **25**, 1207-1226.

89. Piperno, S., Farhat, C. and Larrouturou, B. (1995). Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems - Part I: Model Problem, Theory, and Two-Dimensional Application. *Computer Methods in Applied Mechanics and Engineering*, Vol. 124, Nos. 1-2, pp. 79-112.

90. Piperno, S. and Farhat, C. (2001). Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems - Part II: Energy Transfer Analysis and Three-Dimensional Applications. *Computer Methods in Applied Mechanics and Engineering,* Vol. 190, pp. 3147-3170.

91. Michopoulos, G., Farhat, C. and Fish, J. (2005). Survey on Modeling and Simulation of Multiphysics Systems, *Journal of Computing and Information Science in Engineering* Vol 5, Issue 3, 198-213.

92. Lesoinne, M. and Farhat, C. (1998). Free Staggered Algorithm for Nonlinear Transient Aeroelastic Problems. *AIAA Journal*, Vol. 36, No. 9, pp. 1754-1756.

93. Le Tallec, P. & Mouro, J. (2001). Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics & Engineering* **190** 24-25 3039-3067.

94. Frenklach, M. & Carmer, C. S. 1999. Molecular dynamics using combined quantum & empirical forces: application to surface reactions. *Advances in classical trajectory methods.* Volume 4, 27-63.

95. Haile, J. M. 1992. *Molecular Dynamics Simulations: Elementary Methods.* Wiley.

96. Hase, W. L. 1999. *Molecular Dynamics of Clusters, Surfaces, Liquids, & Interfaces. Advances in classical trajectory methods.* Volume 4. JAI Press.

97. Rapaport, D. C. 1995. *The Art of Molecular Dynamics Simulation.* Cambridge University Press.

98. Schlick, T. 2000. *Molecular modeling & simulation. An interdisciplinary guide.* Springer-Verlag, New York.

99. Moelwyn-Hughes, E. A. 1961. *Physical Chemistry.* Pergamon.

100. Tersoff, J. 1988. Empirical interatomic potential for carbon, with applications to amorphous carbon. *Phys. Rev. Lett.* Vol. 61, pp. 2879-2882.

101. Stillinger, F. H. & Weber, T. A. 1985. Computer simulation of local order in condensed phases of silicon. *Phys. Rev. B* Vol. 31, pp. 5262-5271.