Topological Optimization Using the SIMP Method

Mikal Nelson

University of Kansas
Department of Mathematics
mikal.nelson@ku.edu

M.A. Thesis Defense
July 26th, 2021

Outline

- The Volume-to-Point (VP) Heat Conduction Problem
- The Heat Equation
- The Finite Volume Method (FVM)
  - FVM Discretization of Heat Equation
- Optimization
  - The Method of Moving Asymptotes (MMA)
- Solid Isotropic Material with Penalization (SIMP)
- Algorithm Results

The Volume-to-Point (VP) Heat Conduction Problem

The Volume-to-Point (VP) Heat Conduction Problem

Consider a finite-size volume in which heat is being generated at *every* point, and which is cooled through a small patch (the heat-sink) located on its boundary.

The Volume-to-Point (VP) Heat Conduction Problem

Consider a finite-size volume in which heat is being generated at *every* point, and which is cooled through a small patch (the heat-sink) located on its boundary.

Suppose that we have a finite amount of high-conductivity (k_+) material available.

The Volume-to-Point (VP) Heat Conduction Problem

Consider a finite-size volume in which heat is being generated at *every* point, and which is cooled through a small patch (the heat-sink) located on its boundary.

Suppose that we have a finite amount of high-conductivity (k_+) material available.

Our goal is to determine the optimal distribution of the k_+ material through the given volume such that *the average temperature is minimized*.

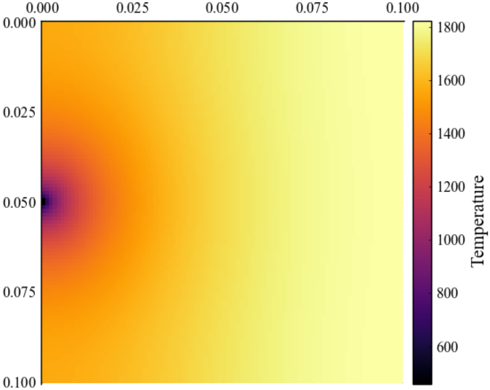


Figure 1: Heatmap for a 0.1m × 0.1m object with uniform heat generation and a heat sink at the center of its west boundary. This map was produced via the Finite Volume Method using 100×100 uniform control volumes.

The Heat Equation

The temperature (T) at any point in such an object is a function of both space $(\mathbf{x})$ and time (t) coordinates: $T(\mathbf{x}, t)$.

The Heat Equation

The temperature (T) at any point in such an object is a function of both space $(\mathbf{x})$ and time (t) coordinates: $T(\mathbf{x}, t)$.

Physical principles demand that such a temperature function must satisfy the equation

$$\frac{\partial T}{\partial t} = \nabla \cdot \big(k(\mathbf{x}) \nabla T\big)\,, \tag{1}$$

where ∇ is the gradient operator and the function k represents the thermal diffusivity at a point in our object.

The Finite Volume Method (FVM)

To numerically solve the heat equation, we employ the Finite Volume Method (FVM).

The Finite Volume Method (FVM)

To numerically solve the heat equation, we employ the Finite Volume Method (FVM).

Given a mesh of polygons (rectangles) on a domain Ω with sample points at $\{x_i\} \subset \Omega$, we create a set of *control volumes* around each x_i. This set of control volumes discretize the partial differential equation.

The Finite Volume Method (FVM)

To numerically solve the heat equation, we employ the Finite Volume Method (FVM).

Given a mesh of polygons (rectangles) on a domain Ω with sample points at $\{x_i\} \subset \Omega$, we create a set of *control volumes* around each x_i. This set of control volumes discretize the partial differential equation.

The main idea of FVM integrate the heat equation over each control volume and then use the Divergence Theorem to convert volume integrals into surface integrals involving the fluxes across the boundaries of the control volumes.

Theorem (The Divergence Theorem)

Suppose that $\mathcal{V}$ is a compact subset of $\mathbb{R}^n$ that has a piecewise smooth boundary $\mathcal{S}$ (i.e. $\partial\mathcal{V} = \mathcal{S}$) with outward pointing normal vectors. If $\mathbf{F}$ is a continously differentiable vector field defined on a neighborhood of $\mathcal{V}$, then

$$\iiint_{\mathcal{V}} (\nabla \cdot \mathbf{F}) \, \mathrm{d}\mathcal{V} = \oiint_{\mathcal{S}} (\mathbf{F} \cdot \hat{\mathbf{n}}) \, \mathrm{d}\mathcal{S} \tag{2}$$

where $\hat{\mathbf{n}}$ is the outwards pointing normal vector to the boundary.

Discretization of Heat Equation

$$\int_{V_i} \frac{\partial T}{\partial t}\,\mathrm{d}\mathbf{x} = \int_{V_i} \nabla \cdot \big(k(\mathbf{x})\nabla T\big)\,\mathrm{d}\mathbf{x} \underset{(2)}{=} \int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}\,\mathrm{d}s \tag{3}$$

Discretization of Heat Equation

$$\int_{V_i} \frac{\partial T}{\partial t}\,\mathrm{d}\mathbf{x} = \int_{V_i} \nabla \cdot \big(k(\mathbf{x})\nabla T\big)\,\mathrm{d}\mathbf{x} \underset{(2)}{=} \int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}\,\mathrm{d}s \tag{3}$$

In a rectangular grid there are only four neighboring cells which we can label as North, South, East, West.

Discretization of Heat Equation

$$\int_{V_i} \frac{\partial T}{\partial t} \,\mathrm{d}\mathbf{x} = \int_{V_i} \nabla \cdot \big(k(\mathbf{x})\nabla T\big) \,\mathrm{d}\mathbf{x} \underset{(2)}{=} \int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}} \,\mathrm{d}s \qquad (3)$$

In a rectangular grid there are only four neighboring cells which we can label as North, South, East, West.

For a control volume V_i we'll label the North boundary as ∂V_N, the South boundary as ∂V_S, the East boundary as ∂V_E, and the West boundary as ∂V_W.

Discretization of Heat Equation

$$\int_{V_i} \frac{\partial T}{\partial t}\,\mathrm{d}\mathbf{x} = \int_{V_i} \nabla \cdot \big(k(\mathbf{x})\nabla T\big)\,\mathrm{d}\mathbf{x} \underset{(2)}{=} \int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}\,\mathrm{d}s \tag{3}$$

In a rectangular grid there are only four neighboring cells which we can label as North, South, East, West.

For a control volume V_i we'll label the North boundary as ∂V_N, the South boundary as ∂V_S, the East boundary as ∂V_E, and the West boundary as ∂V_W.

Additionally, let Δx be the length of the North and South boundaries, and Δy the length of the East and West boundaries.

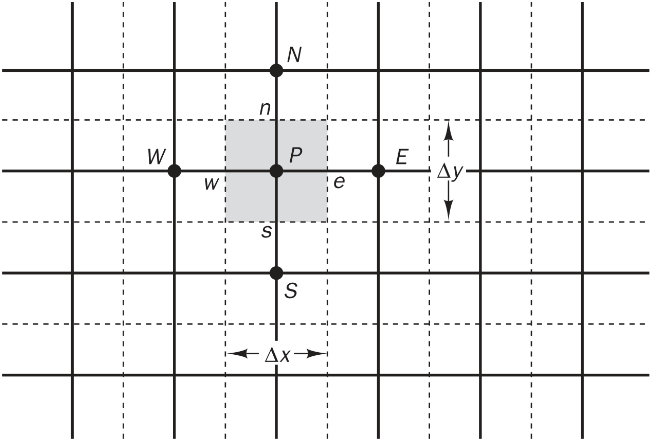


Figure 2: Portion of two-dimensional grid using in the Finite Volume Method for a control volume V_P. [7, p. 129]

Discretization of Heat Equation

$$\int_{V_i} \frac{\partial T}{\partial t} \, \mathrm{d}\mathbf{x} = \int_{V_i} \nabla \cdot \big(k(\mathbf{x}) \nabla T\big) \, \mathrm{d}\mathbf{x} \underset{(2)}{=} \int_{\partial V_i} k(\mathbf{x}) \nabla T \cdot \hat{\mathbf{n}} \, \mathrm{d}s$$

Discretization of Heat Equation

$$\int_{V_i} \frac{\partial T}{\partial t}\,\mathrm{d}\mathbf{x} = \int_{V_i} \nabla \cdot \big(k(\mathbf{x})\nabla T\big)\,\mathrm{d}\mathbf{x} \underset{(2)}{=} \int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}\,\mathrm{d}s$$

Recognizing that $\partial V_i = \partial V_N \cup \partial V_S \cup \partial V_E \cup \partial V_W \ldots$

Discretization of Heat Equation

$$\int_{V_i} \frac{\partial T}{\partial t}\,\mathrm{d}\mathbf{x} = \int_{V_i} \nabla \cdot \big(k(\mathbf{x})\nabla T\big)\,\mathrm{d}\mathbf{x} \underset{(2)}{=} \int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}\,\mathrm{d}s$$

Recognizing that $\partial V_i = \partial V_N \cup \partial V_S \cup \partial V_E \cup \partial V_W \ldots$

$$\begin{aligned}\int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}\,\mathrm{d}s \quad = \quad & \int_{\partial V_N} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_N\,\mathrm{d}s + \int_{\partial V_S} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_S\,\mathrm{d}s \\ & + \int_{\partial V_E} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_E\,\mathrm{d}s + \int_{\partial V_W} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_W\,\mathrm{d}s\end{aligned}$$

Discretization of Heat Equation

$$\begin{aligned}\int_{\partial V_i} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}\,\mathrm{d}s \quad &= \quad \int_{\partial V_N} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_N\,\mathrm{d}s + \int_{\partial V_S} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_S\,\mathrm{d}s \\ &\quad + \int_{\partial V_E} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_E\,\mathrm{d}s + \int_{\partial V_W} k(\mathbf{x})\nabla T \cdot \hat{\mathbf{n}}_W\,\mathrm{d}s\end{aligned}$$

Discretization of Heat Equation

$$\begin{aligned}\int_{\partial V_i} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}\,\mathrm{d}s \quad &= \quad \int_{\partial V_N} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_N\,\mathrm{d}s + \int_{\partial V_S} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_S\,\mathrm{d}s \\ &\quad + \int_{\partial V_E} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_E\,\mathrm{d}s + \int_{\partial V_W} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_W\,\mathrm{d}s\end{aligned}$$

$$\begin{aligned}\int_{V_i}\frac{\partial T}{\partial t}\,\mathrm{d}\mathbf{x} \quad &\approx \quad k_N\frac{T_N - T_i}{\|\mathbf{x}_N - \mathbf{x}_i\|}\Delta x + k_S\frac{T_S - T_i}{\|\mathbf{x}_S - \mathbf{x}_i\|}\Delta x \\ &\quad + k_E\frac{T_E - T_i}{\|\mathbf{x}_E - \mathbf{x}_i\|}\Delta y + k_W\frac{T_W - T_i}{\|\mathbf{x}_W - \mathbf{x}_i\|}\Delta y\end{aligned}$$

Discretization of Heat Equation

$$
\begin{aligned}
\int_{\partial V_i} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}\,\mathrm{d}s \quad &= \quad \int_{\partial V_N} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_N\,\mathrm{d}s + \int_{\partial V_S} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_S\,\mathrm{d}s \\
&\quad + \int_{\partial V_E} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_E\,\mathrm{d}s + \int_{\partial V_W} k(\mathbf{x})\nabla T\cdot\hat{\mathbf{n}}_W\,\mathrm{d}s \\
\int_{V_i}\frac{\partial T}{\partial t}\,\mathrm{d}\mathbf{x} \quad &\approx \quad k_N\frac{T_N-T_i}{\|\mathbf{x}_N-\mathbf{x}_i\|}\Delta x + k_S\frac{T_S-T_i}{\|\mathbf{x}_S-\mathbf{x}_i\|}\Delta x \\
&\quad + k_E\frac{T_E-T_i}{\|\mathbf{x}_E-\mathbf{x}_i\|}\Delta y + k_W\frac{T_W-T_i}{\|\mathbf{x}_W-\mathbf{x}_i\|}\Delta y \\
\implies \Delta x\Delta y\frac{\mathrm{d}T_i}{\mathrm{d}t} \quad &= \quad \left(k_N\frac{T_N-T_i}{\|\mathbf{x}_N-\mathbf{x}_i\|} + k_S\frac{T_S-T_i}{\|\mathbf{x}_S-\mathbf{x}_i\|}\right)\Delta x \\
&\quad + \left(k_E\frac{T_E-T_i}{\|\mathbf{x}_E-\mathbf{x}_i\|} + k_W\frac{T_W-T_i}{\|\mathbf{x}_W-\mathbf{x}_i\|}\right)\Delta y
\end{aligned}
$$

The Optimization Problem

The Optimization Problem

Definition

An optimization problem (in standard form) has the form

$$\begin{array}{lll} \text{minimize} & f_0(\mathbf{x}) & \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, & i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, & i = 1, \ldots, p \end{array} \tag{4}$$

where

- $\mathbf{x} = (x_1, \ldots, x_n)$ are the optimization variables,
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function,
- $f_i : \mathbb{R}^n \to \mathbb{R}$ are the inequality constraint functions, and
- $h_i : \mathbb{R}^n \to \mathbb{R}$ are the equality constraint functions.

The Optimization Problem

Definition

An optimization problem (in standard form) has the form

$$\begin{array}{lll} \text{minimize} & f_0(\mathbf{x}) & \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, & i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, & i = 1, \ldots, p \end{array} \tag{4}$$

where

- $\mathbf{x} = (x_1, \ldots, x_n)$ are the optimization variables,
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function,
- $f_i : \mathbb{R}^n \to \mathbb{R}$ are the inequality constraint functions, and
- $h_i : \mathbb{R}^n \to \mathbb{R}$ are the equality constraint functions.

In our VP problem we are concerned with minimizing the function $f_{av}(T)$, the average temperature, subject to an inequality constraint on the maximum volume of conductive material.

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

1. Choose an initial “guess” for the optimal value x

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

1. Choose an initial "guess" for the optimal value x
2. Find a Descent Direction Δx

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

1. Choose an initial “guess” for the optimal value x
2. Find a Descent Direction Δx
3. Use a line search method to determine an appropriate step-size (t) in the descent direction

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

1. Choose an initial “guess” for the optimal value x
2. Find a Descent Direction Δx
3. Use a line search method to determine an appropriate step-size (t) in the descent direction
4. Compute new guess value: $x + t\Delta x$

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

1. Choose an initial "guess" for the optimal value x
2. Find a Descent Direction Δx
3. Use a line search method to determine an appropriate step-size (t) in the descent direction
4. Compute new guess value: $x + t\Delta x$

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

1. Choose an initial "guess" for the optimal value x
2. Find a Descent Direction Δx
3. Use a line search method to determine an appropriate step-size (t) in the descent direction
4. Compute new guess value: $x + t\Delta x$

The main differentiating factor between methods is how the descent direction is determined.

Optimization Methods

Unconstrained optimizations algorithms follow a general blueprint:

1. Choose an initial "guess" for the optimal value x
2. Find a Descent Direction Δx
3. Use a line search method to determine an appropriate step-size (t) in the descent direction
4. Compute new guess value: $x + t\Delta x$

The main differentiating factor between methods is how the descent direction is determined.

For example, the Gradient Descent algorithm chooses $\Delta x = -\nabla f(x)$ as the descent direction, while the Conjugate Gradient method requires that descent directions be conjugate to one another.

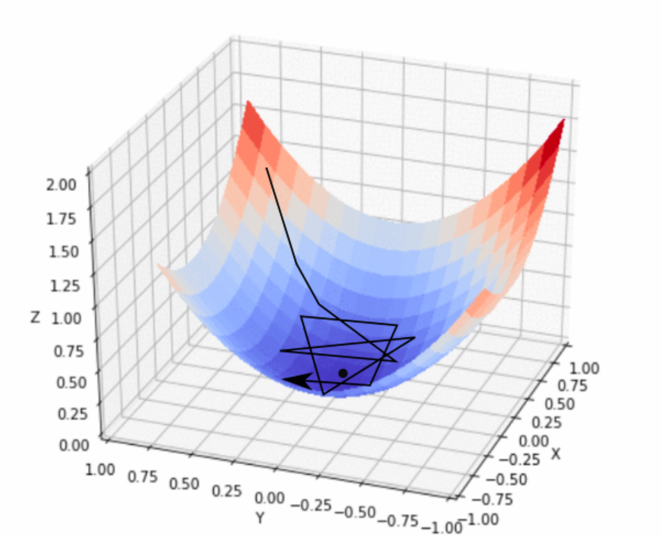


Figure 3: A gradient descent algorithm in action. [3]

The VP problem is a *constrained* optimization problem.

The VP problem is a *constrained* optimization problem.

Additionally, it is not a convex problem.

The VP problem is a *constrained* optimization problem.

Additionally, it is not a convex problem.

(Convex problems have a lot of properties we can take advantage of that make the optimization problem easier to solve. Most notably, for a convex optimization problem, all local optima are global optima.)

The VP problem is a *constrained* optimization problem.

Additionally, it is not a convex problem.

(Convex problems have a lot of properties we can take advantage of that make the optimization problem easier to solve. Most notably, for a convex optimization problem, all local optima are global optima.)

Hence, we will employ a more complicated optimization algorithm.

The Method of Moving Asymptotes (MMA)

The Method of Moving Asymptotes is a method of non-linear programming, originally developed for structural optimization. MMA is designed for constrained optimization problems.

The Method of Moving Asymptotes (MMA)

The Method of Moving Asymptotes is a method of non-linear programming, originally developed for structural optimization. MMA is designed for constrained optimization problems.

The method uses an iterative process which creates a convex subproblem that is solved in each iteration. Each of these subproblems is an approximation of the original problem with parameters that change the curvature of the approximation.

The Method of Moving Asymptotes (MMA)

The Method of Moving Asymptotes is a method of non-linear programming, originally developed for structural optimization. MMA is designed for constrained optimization problems.

The method uses an iterative process which creates a convex subproblem that is solved in each iteration. Each of these subproblems is an approximation of the original problem with parameters that change the curvature of the approximation.

These parameters act as asymptotes for the subproblem and moving the asymptotes between iterations stablizes the convergence of the entire process.

The Method of Moving Asymptotes (MMA)

Step 0: Choose a starting point $\mathbf{x}^{(0)}$, and let the iteration index $k = 0$.

Step 1: Given an iteration point $\mathbf{x}^{(k)}$, calculate $f_i(\mathbf{x}^{(k)})$ and the gradients $\nabla f_i(\mathbf{x}^{(k)})$ for $i = 0, 1, \ldots, m$.

Step 2: Generate a subproblem $P^{(k)}$ by replacing the functions f_i by approximating functions $f_i^{(k)}$, based on calculations from Step 1.

Step 3: Solve $P^{(k)}$ and let the optimal solution of this subproblem be the next iteration point $\mathbf{x}^{(k+1)}$. Let $k = k + 1$ and go to Step 1.

The Method of Moving Asymptotes (MMA)

Step 0: Choose a starting point $\mathbf{x}^{(0)}$, and let the iteration index $k = 0$.

Step 1: Given an iteration point $\mathbf{x}^{(k)}$, calculate $f_i(\mathbf{x}^{(k)})$ and the gradients $\nabla f_i(\mathbf{x}^{(k)})$ for $i = 0, 1, \ldots, m$.

Step 2: Generate a subproblem $P^{(k)}$ by replacing the functions f_i by approximating functions $f_i^{(k)}$, based on calculations from Step 1.

Step 3: Solve $P^{(k)}$ and let the optimal solution of this subproblem be the next iteration point $\mathbf{x}^{(k+1)}$. Let $k = k + 1$ and go to Step 1.

Each $f_i^{(k)}$ is obtained by a linearization of f_i in variables of the type

$$\frac{1}{x_j - L_j} \quad \text{or} \quad \frac{1}{U_j - x_j}$$

dependent on the signs of the derivatives of f_i at $\mathbf{x}^{(k)}$. The values of L_j and U_j are normally changed between iterations and are referred to as moving asymptotes.

The Method of Moving Asymptotes (MMA)

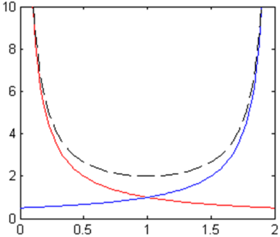
Essentially the algorithm is creating *convex* approximations to the objective and constraint functions:

$$f_i^{(k)}(\mathbf{x}) = r_i^{(k)} + \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right)$$

The Method of Moving Asymptotes (MMA)

Essentially the algorithm is creating *convex* approximations to the objective and constraint functions:

$$f_i^{(k)}(\mathbf{x}) = r_i^{(k)} + \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right)$$

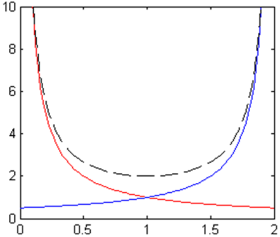


[6]

The Method of Moving Asymptotes (MMA)

Essentially the algorithm is creating *convex* approximations to the objective and constraint functions:

$$f_i^{(k)}(\mathbf{x}) = r_i^{(k)} + \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right)$$



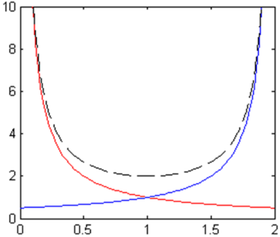
[6]

These convex approximations can then be efficiently solved using methods such as gradient descent.

The Method of Moving Asymptotes (MMA)

Essentially the algorithm is creating *convex* approximations to the objective and constraint functions:

$$f_i^{(k)}(\mathbf{x}) = r_i^{(k)} + \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right)$$



[6]

These convex approximations can then be efficiently solved using methods such as gradient descent.

For our implementation of the SIMP method, the MMA algorithm was run from within the NLopt optimization package in Julia.

Solid Isotropic Material with Penalization (SIMP)

Solid Isotropic Material with Penalization (SIMP) is a method based on topology optimization that can be used to solve the VP Heat Conduction Problem.

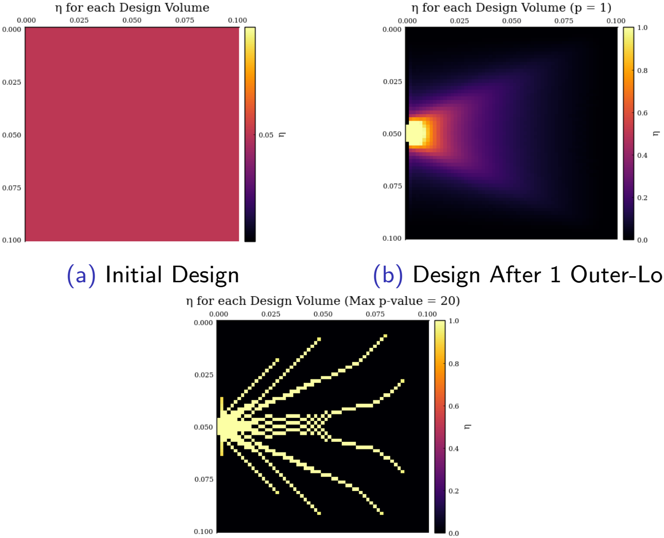
Solid Isotropic Material with Penalization (SIMP)

Solid Isotropic Material with Penalization (SIMP) is a method based on topology optimization that can be used to solve the VP Heat Conduction Problem.

In each step of the SIMP method we increase or decrease the proportion of high-conductivity material used in our object by a small quantity.

This allows us to apply methods designed for continuous optimization problems to the discrete VP problem as it transforms the binary 1—0 problem into a sequence of continuous problems.

Solid Isotropic Material with Penalization (SIMP)



(a) Initial Design

(b) Design After 1 Outer-Loop

(c) Final Design

Preliminary Parameters

First of all, the energy differential equation driving the heat-flux inside the finite-volume requires:

1. All calculations are run under steady-state conditions. That is, we seek a stable solution where quantities are independent of time.
2. All heat produced in the volume is evacuated through the heat sink.
3. Low-conductivity materials (k_0) and high-conductivity materials (k_+) are treated as homogeneous and isotropic on their respective conductivities.

Preliminary Parameters

First of all, the energy differential equation driving the heat-flux inside the finite-volume requires:

1. All calculations are run under steady-state conditions. That is, we seek a stable solution where quantities are independent of time.
2. All heat produced in the volume is evacuated through the heat sink.
3. Low-conductivity materials (k_0) and high-conductivity materials (k_+) are treated as homogeneous and isotropic on their respective conductivities.

We also set the following conditions:

- Thermal conductivities depend only on the material, and therefore are constant:
$$k_0 = 1\,\frac{\mathrm{W}}{\mathrm{m}^2\,\mathrm{K}} \qquad \text{and} \qquad k_+ = 100\,\frac{\mathrm{W}}{\mathrm{m}^2\,\mathrm{K}}$$
- All structures have a square aspect ratio with $L = H = 0.1\mathrm{m}$
- The heat-sink is located on the middle of the west side of the structure
- The heat-sink has Dirichlet boundary conditions: $T_S = 0°\mathrm{C}$
- All other boundaries are adiabatic (Neumann boundary conditions): $\nabla T \cdot \mathbf{n} = 0$

Notation

We have the following sets to describe the VP-problem:

- $\mathbf{x} \in \Omega =$ two-dimensional spatial field

  We set $\Omega = \Omega_0 \cup \Omega_+$ where

  - $\Omega_0 =$ portion of Ω that has conductivity k_0.
  - $\Omega_+ =$ portion of Ω with conductivity k_+. This is the portion of the space with high-conductivity material.

Notation

We have the following sets to describe the VP-problem:

- $\mathbf{x} \in \Omega =$ two-dimensional spatial field

  We set $\Omega = \Omega_0 \cup \Omega_+$ where

  - $\Omega_0 =$ portion of Ω that has conductivity k_0.
  - $\Omega_+ =$ portion of Ω with conductivity k_+. This is the portion of the space with high-conductivity material.

Using the above established notation, we develop the following optimization problem:

$$\begin{array}{lll} \text{minimize} & f(T) & \text{for } \ \Omega_+ \\ \text{subject to} & \nabla \cdot (k \nabla T) + q = 0 & \\ & \mathbf{x} \in \Omega & \end{array} \tag{5}$$

Notation

We have the following sets to describe the VP-problem:

- $\mathbf{x} \in \Omega =$ two-dimensional spatial field

  We set $\Omega = \Omega_0 \cup \Omega_+$ where

  - $\Omega_0 =$ portion of Ω that has conductivity k_0.
  - $\Omega_+ =$ portion of Ω with conductivity k_+. This is the portion of the space with high-conductivity material.

Using the above established notation, we develop the following optimization problem:

$$\begin{array}{lll} \text{minimize} & f(T) & \text{for } \ \Omega_+ \\ \text{subject to} & \nabla \cdot (k \nabla T) + q = 0 & \\ & \mathbf{x} \in \Omega & \end{array} \tag{5}$$

The objective function $f(T)$ varies depending on desired design outcomes. Some possible objective functions include average temperature (used in our implementation), temperature variance, and maximum temperature.

Porosity Constraint

We impose a constraint upon this problem to limit the quantity of k_+ material available:

$$\int_{\Omega_+} \mathrm{d}\mathbf{x} = \int_{\Omega} \delta_+ \, \mathrm{d}\mathbf{x} \leq V \quad \text{where } \begin{cases} \delta_+ = 0 & \text{if } \mathbf{x} \in \Omega_0 \\ \delta_+ = 1 & \text{if } \mathbf{x} \in \Omega_+ \end{cases} \tag{6}$$

Porosity Constraint

We impose a constraint upon this problem to limit the quantity of k_+ material available:

$$\int_{\Omega_+} \mathrm{d}\mathbf{x} = \int_{\Omega} \delta_+ \, \mathrm{d}\mathbf{x} \leq V \quad \text{where } \begin{cases} \delta_+ = 0 & \text{if } \mathbf{x} \in \Omega_0 \\ \delta_+ = 1 & \text{if } \mathbf{x} \in \Omega_+ \end{cases} \tag{6}$$

Inequality (6) sets a cap on the maximum volume (V) of Ω_+ and hence limits the available amount of high-conductivity material that can be applied to the domain.

If we did not have this constraint, the optimal solution would be to set $\Omega_+ = \Omega$, making the entire domain have high conductivity material.

Penalization Process

The problem of whether to place high conductivity material in a particular location or not is discrete in nature. This is unfortunate as continuous optimization problems are generally easier to solve.

Penalization Process

The problem of whether to place high conductivity material in a particular location or not is discrete in nature. This is unfortunate as continuous optimization problems are generally easier to solve.

The SIMP method has a clever way of getting around this particular issue of discrete variables: create a continuous function that allows for a “mix” of the two conductive materials.

Penalization Process

The problem of whether to place high conductivity material in a particular location or not is discrete in nature. This is unfortunate as continuous optimization problems are generally easier to solve.

The SIMP method has a clever way of getting around this particular issue of discrete variables: create a continuous function that allows for a "mix" of the two conductive materials.

$$k\left(\eta\right) = k_0 + \left(k_+ - k_0\right)\eta^p \qquad \text{with} \qquad 0 \leq \eta \leq 1 \qquad \text{and} \qquad p \geq 1. \tag{7}$$

η is called the *design parameter* and p the *penalization parameter*.

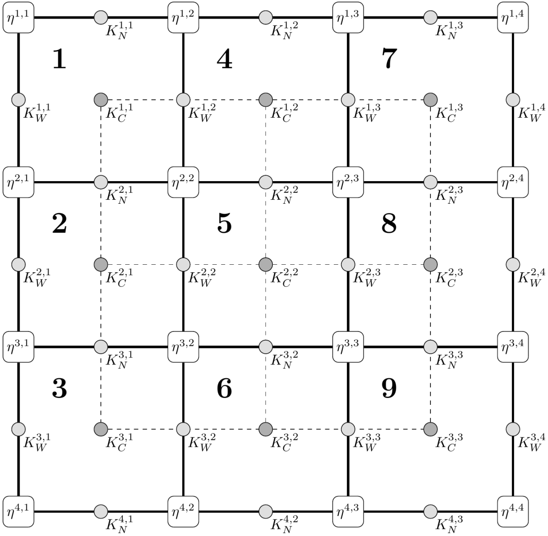


Figure 5: Overlayed Temperature (—) and Design (−−−) grids with 4×4 Design Element ($\eta^{i,j}$) and 3×3 Temperature Control Volume ($K_C^{i,j}$) Nodes. $K_N^{i,j}$ and $K_W^{i,j}$ indicate nodes at the North and West boundaries, respectively, of each Temperature Control Volume. Each Temperature control volume is numbered beginning in the upper left and continuing column-by-column, left-to-right.

Discretization (Again!)

$$\nabla \cdot (k \nabla T) + q = 0 \tag{8}$$

Discretization (Again!)

$$\nabla \cdot (k \nabla T) + q = 0 \tag{8}$$

For temperature control volume (i, j), the finite volume method discretizes (8) into the following linear equation

$$K_C^{i,j} T^{i,j} = K_W^{i,j} T^{i,j-1} + K_W^{i,j+1} T^{i,j+1} + K_N^{i,j} T^{i-1,j} + K_N^{i+1,j} T^{i+1,j} + \Delta x \Delta y Q^{i,j} \tag{9}$$

Discretization (Again!)

$$\nabla \cdot (k \nabla T) + q = 0 \tag{8}$$

For temperature control volume (i, j), the finite volume method discretizes (8) into the following linear equation

$$K_C^{i,j} T^{i,j} = K_W^{i,j} T^{i,j-1} + K_W^{i,j+1} T^{i,j+1} + K_N^{i,j} T^{i-1,j} + K_N^{i+1,j} T^{i+1,j} + \Delta x \Delta y Q^{i,j} \tag{9}$$

The K_W and K_N coefficients are dependent on the thermal conductivity and cross-sectional area of their corresponding faces:

$$K_W^{i,j} = \frac{k_W^{i,j} \Delta y}{\Delta x} \quad \text{and} \quad K_N^{i,j} = \frac{k_N^{i,j} \Delta x}{\Delta y} \tag{10}$$

$$k_W^{i,j} = \frac{k^{i,j} + k^{i+1,j}}{2} \qquad k_N^{i,j} = \frac{k^{i,j} + k^{i,j+1}}{2} \tag{11}$$

Discretization (Again!)

$$\nabla \cdot (k \nabla T) + q = 0 \tag{8}$$

For temperature control volume (i, j), the finite volume method discretizes (8) into the following linear equation

$$K_C^{i,j} T^{i,j} = K_W^{i,j} T^{i,j-1} + K_W^{i,j+1} T^{i,j+1} + K_N^{i,j} T^{i-1,j} + K_N^{i+1,j} T^{i+1,j} + \Delta x \Delta y Q^{i,j} \tag{9}$$

The K_W and K_N coefficients are dependent on the thermal conductivity and cross-sectional area of their corresponding faces:

$$K_W^{i,j} = \frac{k_W^{i,j} \Delta y}{\Delta x} \qquad \text{and} \qquad K_N^{i,j} = \frac{k_N^{i,j} \Delta x}{\Delta y} \tag{10}$$

$$k_W^{i,j} = \frac{k^{i,j} + k^{i+1,j}}{2} \qquad k_N^{i,j} = \frac{k^{i,j} + k^{i,j+1}}{2} \tag{11}$$

$$K_C^{i,j} = K_W^{i,j} + K_W^{i,j+1} + K_N^{i,j} + K_N^{i+1,j} \tag{12}$$

Discretized Optimization Problem

$$\mathbf{K}\mathbf{T} = \Delta x \Delta y \mathbf{Q}. \tag{13}$$

Discretized Optimization Problem

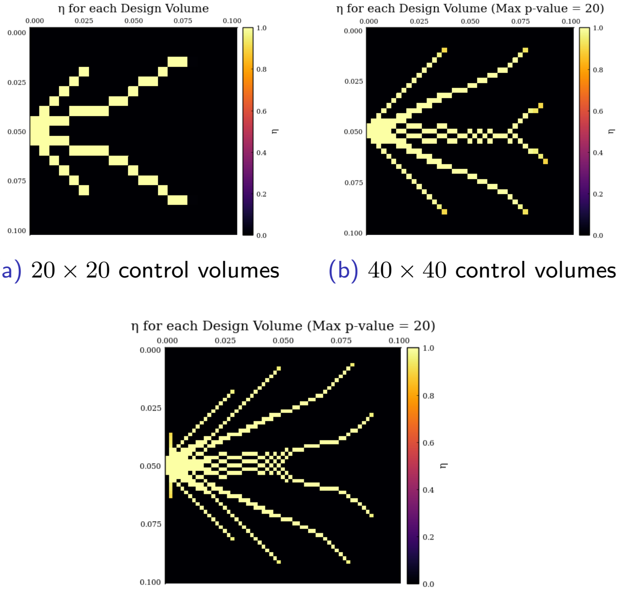
$$\mathbf{K}\mathbf{T} = \Delta x \Delta y \mathbf{Q}. \tag{13}$$

Rearranging what we have done into matrix equations, we can get the following discretized formulation of the average temperature optimization problem:

$$\begin{array}{ll} \text{minimize} & f_{av}\left(\mathbf{T}\right) = \frac{1}{N_T}\mathbf{1}^T\mathbf{T} \\ \text{subject to} & \mathbf{K}\mathbf{T} = \Delta x \Delta y \mathbf{Q} \\ & \mathbf{1}^T\boldsymbol{\eta} - N_T\overline{\phi} \leq 0 \\ & \mathbf{k} = k_0\mathbf{1} + (k_+ - k_0)\boldsymbol{\eta}^p \\ & 0 \leq \boldsymbol{\eta} \leq 1 \text{ and } p \geq 1 \end{array} \tag{14}$$

where N_T represents the number of temperature control volumes and $\overline{\phi}$ represents the maximum porosity (the cap on the amount of high-conductivity material).

Results

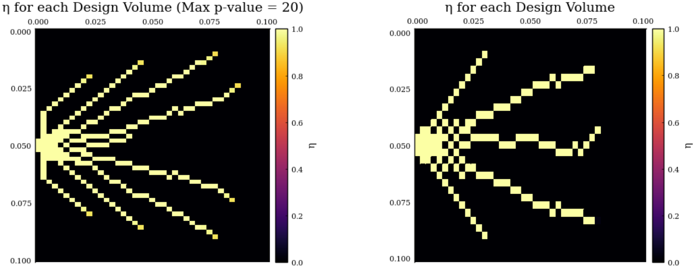


(a) 20×20 control volumes

(b) 40×40 control volumes

(c) 60×60 control volumes

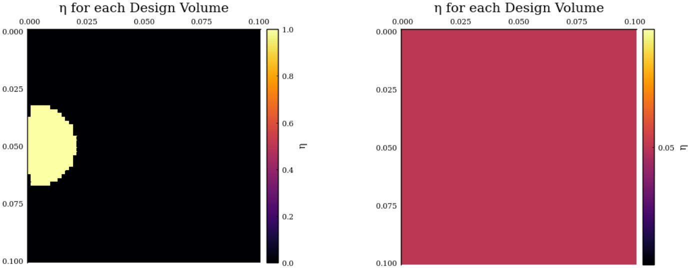
Non-Square Grids



(a) 50×40 control volumes

(b) 30×40 control volumes

Higher Initial Penalization



(a) 60×60 Control Volumes with Initial $p = 5$.

(b) 60×60 Control Volumes with Initial $p = 19$.

Figure 8: Design outputs for inital p-values greater than 1.

Minimum Average Temperature Convergence

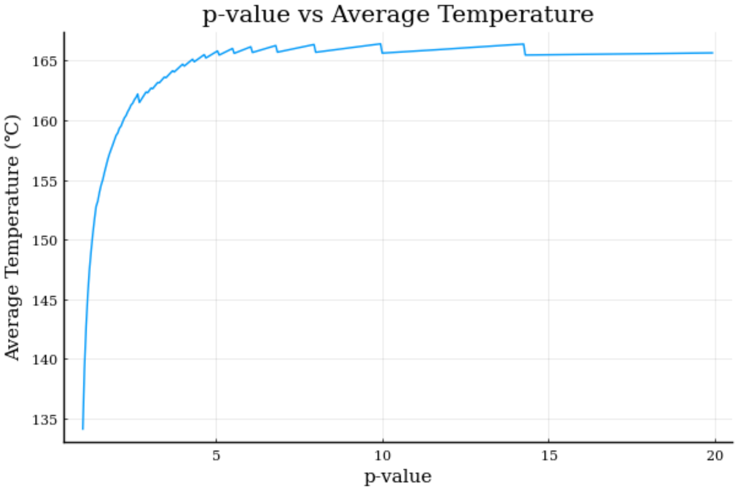
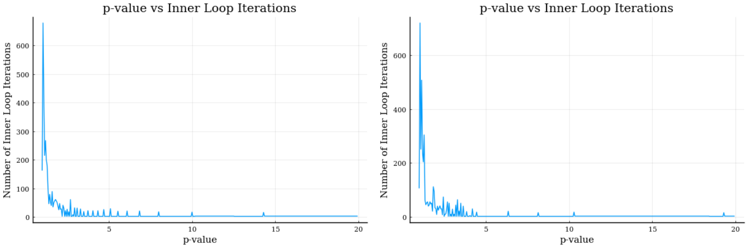


Figure 9: p-value plotted against Average Temperature Evaluation for 60×60 control volumes with a maximum p-value of 20.



(a) p-value plotted against the number of inner loop iterations for 60×60 control volumes with a maximum p-value of 20.

(b) p-value plotted against the number of inner loop iterations for 50×40 control volumes with a maximum p-value of 20.

Figure 10: p-value vs. Number of Inner Loop Iterations for Square and Rectangular Control Volumes

Algorithm Runtime

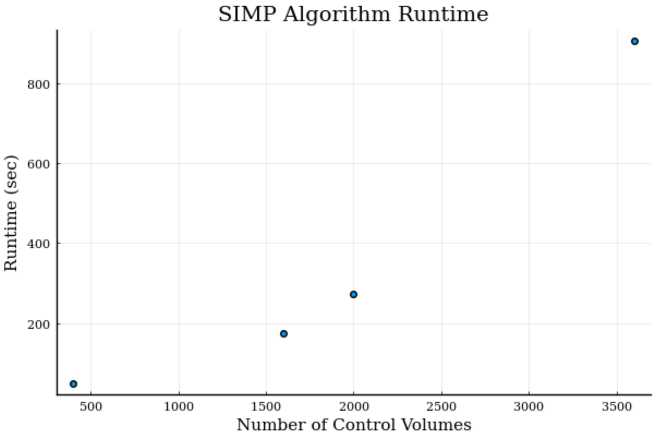


Figure 11: Plot of runtime for SIMP algorithm for various numbers of control volumes.

Bibliography

Stephen P. Boyd and Lieven Vandenberghe.
Convex Optimization.
Cambridge Univ. Pr., 2004.

Martin J. Gander and Felix Kwok.
Numerical Analysis of Partial Differential Equations Using Maple and MATLAB.
Society for Industrial and Applied Mathematics, Aug 2018.

Ayoosh Kathuria.
Intro to optimization in deep learning: Gradient descent.
WebPage, Dec 2020.

Gilles Marck, Maroun Nemer, Jean-Luc Harion, Serge Russeil, and Daniel Bougeard.
Topology optimization using the SIMP method for multiobjective conductive problems.
Numerical Heat Transfer, Part B: Fundamentals, 61(6):439–470, jun 2012.

Krister Svanberg.
The method of moving asymptotes—a new method for structural optimization.
International Journal for Numerical Methods in Engineering, 24(2):359–373, feb 1987.

Fred van Keulen and Matthijs Langelaar.
Engineering optimization – concepts and applications.
PowerPoint.

H. K. Versteeg.
An Introduction to Computational Fluid Dynamics : The Finite Volume Method.
Pearson Education Ltd, Harlow, England New York, 2007.

Questions?