Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
Project five		SZO

Simulation inversion and analysis of global thermal conductivity of two-dimensional space materials based on particle swarm optimization

Summary

Thermal conductivity is an important thermodynamic index of a material, but since it is a derived quantity that cannot be measured directly, the inversion of thermal conductivity is a difficult task. Based on Fourier's law of thermal conductivity, we can somewhat invert the global thermal conductivity of a material by the variation of its temperature distribution. In this paper, we limit the space of the material to two dimensions, i.e., only x and y directions, and simulate the thermal conduction process of the material in the air to reflect the change of its temperature distribution with position and time, and carry out the perturbation analysis of the heat conduction equation, so that we can limit the optimization region to the final temperature distribution, and combine with particle swarm optimization algorithms to invert the coefficient of thermal conductivity that best meets the temperature distribution.

Keywords: Thermal conductivity, particle swarm optimization(PSO), Temperature distribution simulation

Contents

1 Heat conduction equation and its perturbative analysis

We consider here the relationship between the temperature $u_{(t,x,y)}$ with time t, the spatial coordinates x, y

$$\frac{\partial u}{\partial t} = \frac{k}{c\rho}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$$

where c is the material specific heat capacity and ρ is the material density

Assume we have two identical materials with thermal conductivities k_1 and k_2, where $k_1 \neq k_2$. Let $u_0(x, y)$ denote the initial temperature field. We perturb the thermal conductivity as $k(x, y, t) = k_0 + \epsilon\delta k(x, y, t)$, where ϵ is a small perturbation parameter and $\delta k(x, y, t)$ is the perturbation function.

We expand the solution as $u(x, y, t) = u_0(x, y) + \epsilon v(x, y, t)$, where $v(x, y, t)$ is the perturbation term.

Substituting the perturbed solution into the original equation, and collecting terms of different orders of ϵ, we obtain:

$$\begin{aligned}&\frac{\partial}{\partial t}(u_0(x,y) + \epsilon v(x,y,t)) \\ &= \frac{k_0 + \epsilon\delta k(x,y,t)}{c\rho}\left(\frac{\partial^2}{\partial x^2}(u_0(x,y) + \epsilon v(x,y,t)) + \frac{\partial^2}{\partial y^2}(u_0(x,y) + \epsilon v(x,y,t))\right)\end{aligned}$$

Expanding and rearranging, we get:

$$\frac{\partial u_0}{\partial t} + \epsilon\frac{\partial v}{\partial t} = \frac{k_0}{c\rho}\left(\frac{\partial^2 u_0}{\partial x^2} + \frac{\partial^2 u_0}{\partial y^2}\right) + \frac{\delta k}{c\rho}\left(\frac{\partial^2 u_0}{\partial x^2} + \frac{\partial^2 u_0}{\partial y^2}\right) + \frac{k_0}{c\rho}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

We split the equation into zeroth and first-order terms:

Zeroth-order term:

$$\frac{\partial u_0}{\partial t} = \frac{k_0}{c\rho}\left(\frac{\partial^2 u_0}{\partial x^2} + \frac{\partial^2 u_0}{\partial y^2}\right)$$

This is the original heat conduction equation describing the evolution of the initial temperature field $u_0(x, y)$.

First-order term:

$$\frac{\partial v}{\partial t} = \frac{k_0}{c\rho}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{\delta k}{k_0}\frac{k_0}{c\rho}\left(\frac{\partial^2 u_0}{\partial x^2} + \frac{\partial^2 u_0}{\partial y^2}\right)$$

This equation describes the evolution of the perturbation term $v(x, y, t)$. The first term represents the heat conduction of the perturbation v itself, while the second term accounts for the effect of the perturbation in the thermal conductivity k.

By solving these equations, we can obtain the evolution of the initial temperature field $u_0(x, y)$ and the perturbation term $v(x, y, t)$, thereby understanding the changes in the temperature field due to small variations in the initial thermal conductivity k.

To illustrate the effect of different initial thermal conductivities k on the final temperature distribution at a given time T, we can employ numerical methods, specifically the finite difference method. We discretize the rectangular domain into a grid and compute the temperature at each grid node. The 2D heat conduction equation is discretized as follows:

$$u_{i,j}^{n+1} = u_{i,j}^n + \frac{k}{c\rho}\frac{\Delta t}{\Delta x^2}(u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n) + \frac{k}{c\rho}\frac{\Delta t}{\Delta y^2}(u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n)$$

where $u_{i,j}^n$ represents the temperature at grid point (i, j) at time step n, Δt is the time step, Δx and Δy are the spatial steps.

The procedure involves:

1. Initialize the temperature on the grid according to the initial condition $u_0(x, y)$.
2. Use explicit Euler method to iteratively update the temperature field based on the discretized heat conduction equation.
3. Repeat step 2 until the specified end time T is reached.
4. Compare the temperature distributions on the grid for different initial thermal conductivities k.

As long as T is greater than 0, a small perturbation δ acting on k,i.e.$k + \delta$ can make the final temperature distribution $u_{(T,x,y)}$ turn out to be different.

Then, based on this analysis, we only need to simulate the temperature distribution of the material at the end of the heat conduction time over a period of time in order to inversely perform the thermal conductivity of the material k.

2 Material Heat Transfer Simulation

In the simulation, we assume that the length and width of the material are 1 meter and the total simulation duration is 100 seconds, with a spatial step dx,dy, and a time step dt of 0.01, and the ambient temperature of the material is 20 degrees Celsius and the initial internal temperature of the material is 100 degrees Celsius.

simulates the heat conduction process and temperature distribution in a two-dimensional space using numerical methods. The main idea is to discretize the heat conduction equation and iteratively update the temperature field.

The heat conduction equation describes the propagation of heat in space and is typically represented as a partial differential equation. In this program, an explicit finite difference method is used to approximate the solution to the heat conduction equation. Specifically, a second-order central differencing scheme is applied to discretize the temperature gradient in space, and an explicit Euler method is used for time stepping.

At each time step, the temperature field at the next time step is computed by calculating the temperature gradient around each grid point. This process is performed iteratively, updating the entire temperature field at each iteration. Initially, boundary conditions and an initial temperature field are provided, and the temperature field evolves over time through iteration. To accelerate computation, the program utilizes parallel process-

```
def evolve(u, u_previous, dt, dx2, dy2, alpha):
    nx, ny = u.shape
    for i in range(1, nx - 1):
        for j in range(1, ny - 1):
            u[i, j] = u_previous[i, j] + alpha * dt * (
                (u_previous[i + 1, j] - 2 * u_previous[i, j] + u_previous[i - 1, j]) / dx2 +
                (u_previous[i, j + 1] - 2 * u_previous[i, j] + u_previous[i, j - 1]) / dy2)
    return u
```

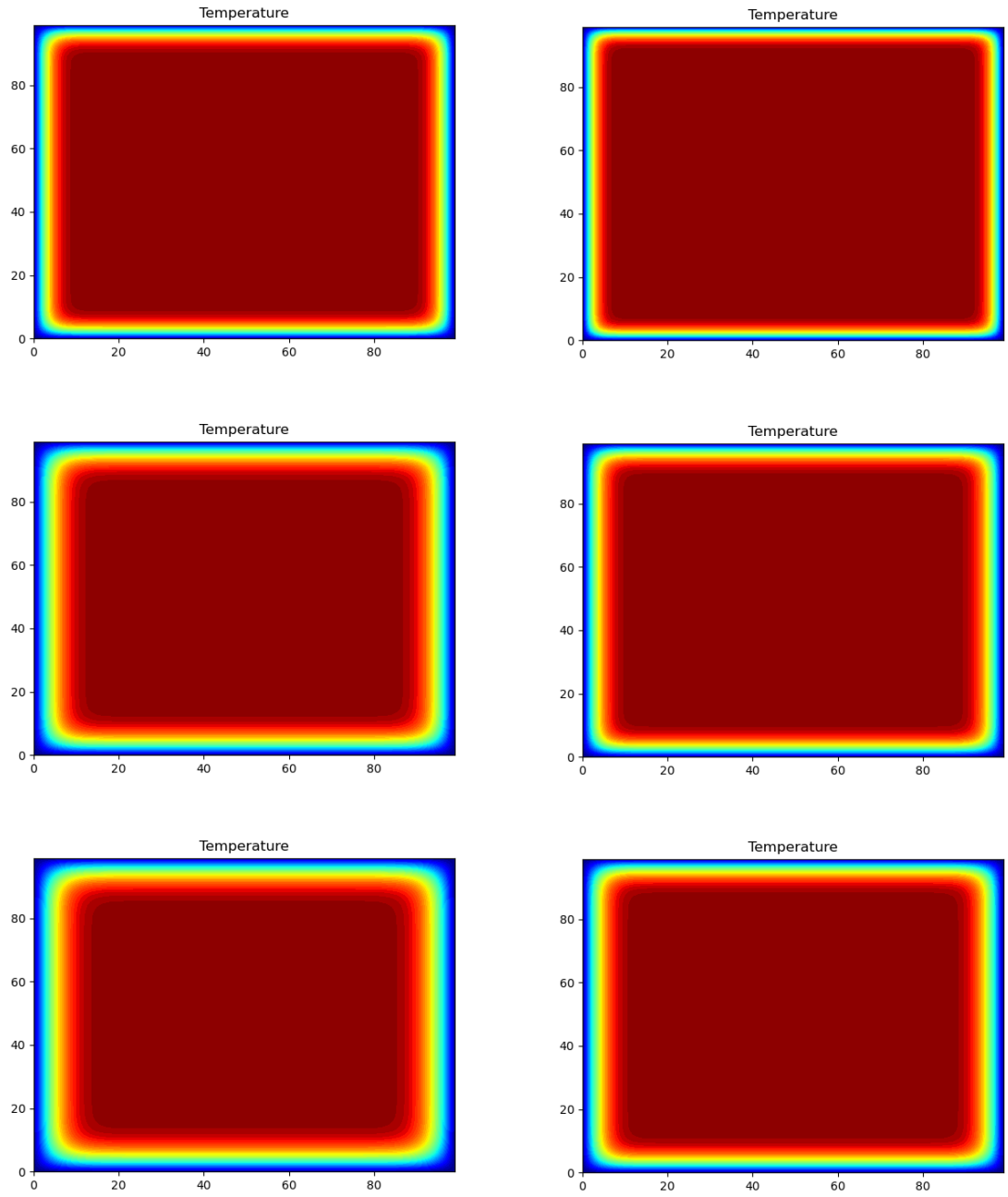
Figure 1: heat simulation

ing, employing multiple processes to simultaneously handle different parts of the computation. This allows for more efficient utilization of computational resources, speeding up the simulation.

Additionally, the program offers visualization of the temperature field data, allowing for the generation of images representing the temperature distribution, as well as the capability to export temperature field data to an Excel file for further analysis and processing.

Simulation results of heat conduction of some materials(The units of k are all $W/(m \cdot K)$, the units of c are all $J/(kg \cdot K)$, and the units of ρ are all kg/m^3):

Aluminum, 6061,Temper-O($k = 180$**,**$c = 900$**,**$\rho = 2700$**);Brass, Red,** $\%85$**Cu,**$\%15$**Zn(**$k = 151$**,**$c = 380$**,**$\rho = 8800$**)**



Aluminum, 6061,Temper-O

Brass, Red, %85Cu,%15Zn

3 Introduction to Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a universal method for solving optimization problems, characterized by fast solution speed, numerical stability, and simplicity of the algorithm. It is a commonly used algorithm in mathematical modeling.

3.1 Bionics of Particle Swarm Optimization

During the foraging process, members of a flock of birds can communicate and share information with each other to gain insights into the discoveries and flight experiences of others. Under conditions where food sources are sporadically distributed and unpredictable, the advantages brought by this collaborative mechanism are decisive, far outweighing the disadvantages caused by competition for food. Inspired by the foraging behavior of birds, PSO mimics this behavior, analogizing the search space of the optimization problem to the flight space of birds. Each bird is abstracted as a particle, which is massless and volumeless, representing a feasible solution to the problem. The optimal solution sought by the optimization problem is equivalent to the food source that the birds are searching for. PSO establishes simple behavioral rules for each particle, similar to the movement of birds, making the movement of the entire swarm exhibit characteristics similar to bird foraging, thus solving complex optimization problems.

3.2 Particle Swarm Optimization Modeling

PSO first initializes the swarm randomly in a given solution space, where the number of variables of the optimization problem determines the dimension of the solution space. Each particle has an initial position and initial velocity (the most important aspect of PSO is the update of the velocity formula because velocity determines the direction of update, thereby determining the value of the solution), and then seeks optimization through iteration. In each iteration, each particle updates its position and velocity in the solution space by tracking two "extremes": one extreme is the optimal solution that the individual particle has found in the iteration process, which is called the individual extreme; (the meaning here is that each particle has memory, remembering two values, one is the optimal value of all particles' distance from the target, and the other is the best value this particle has ever achieved in history) the other extreme is the optimal solution that all particles in the swarm have found in the iteration process, which is the global extreme.

3.3 Features of Particle Swarm Optimization

The essence of PSO is a random search algorithm, and it is an emerging intelligent optimization technology. The algorithm is likely to converge to the global optimal solution with a high probability. Practice has proven that it is suitable for optimization in dynamic and multi-objective environments, and compared with traditional optimization algorithms, it has faster computational speed and better global search capability.

Additionally, PSO has parallelism, and because each particle has memory, at the end of the algorithm iteration, not only can the optimal value be obtained, but also the suboptimal value. Therefore, for scheduling and decision-making problems, it can provide multiple meaningful solutions. Moreover, PSO is not sensitive to the size of the swarm, and the performance degradation is not very significant.

4 Inversion of heat transfer coefficient based on particle swarm optimization algorithm

To solve the difficult inverse problem of inverting the thermal conductivity, we convert it to an optimized positive problem, i.e., given the temperature distribution of the material at the end of the simulation time, u(T,x,y), and assuming that the true thermal conductivity of the material is k, the thermal conductivity of the material is determined by the

$$\min_k ||\hat{u}_k(T,x,y) - u(T,x,y)||^2$$

where $\hat{u}_k(T,x,y)$ is the end-state temperature distribution for a thermal conductivity of k, and the $u_{(T,x,y)}$ is the real end-state temperature distribution of materials.The idea of optimization is that for each k in an iteration, the data of its end-state temperature distribution is generated on-the-fly to compare with the true temperature distribution in order to minimize the objective function and obtain the optimal k, i.e., the true global thermal conductivity of the material.

The overall realization is as follows

1. **Particle Swarm Optimization (PSO) Algorithm**: PSO algorithm is a heuristic optimization algorithm inspired by the foraging behavior of birds. In this algorithm, each "particle" represents a potential solution in the solution space, and the entire swarm of particles searches for the optimal solution in the solution space. By continuously updating the velocity and position of particles, the swarm can move towards the global optimum.

2. **Parameter Settings**: - Number of Particles ($num\_particles$): Controls the density of exploration in the search space. A larger number covers more possibilities in the search space. - Number of Iterations ($num\_iterations$): Specifies the number of iterations the PSO algorithm performs, affecting the speed at which the algorithm converges to the optimal solution. - Space Dimensions ($lenX, lenY$): Define the spatial dimensions of the simulation. - Time Steps per Process ($timesteps\_per\_process$): The number of time steps simulated by each process, affecting the time range and precision of the simulation. - Initial Guess of Thermal Conductivity ($initial\_k$): The initial position of particles at the beginning of the PSO algorithm, i.e., the initial guess of the thermal conductivity value.

3. **PSO Parameter Adjustment**: - Inertia Weight ($inertia$): Controls the inertia of particle motion. A larger value increases the exploration of particles in the solution space, while a smaller value helps accelerate convergence. - Cognitive Weight ($cognitive\_weight$) and Social Weight ($social\_weight$): Influence the degree to which particles move towards individual and global best positions, respectively. Adjusting these two weights balances the ability of local and global search.

By appropriately adjusting these parameters, the algorithm can better explore the solution space and find the optimal thermal conductivity value. Additionally, due to the parallel nature of PSO algorithm, multiple processors or processes can be utilized to speed up computation, improving the efficiency of the algorithm.

In order to achieve only the purpose of verification of the correctness of the model,

without affecting the judgment of the correctness of the model, we change the length and width of the material to 0.2 meters, and in the spatio-temporal mesh, the time step and spatial step are all changed to 0.8. The following table shows the inverse thermal conductivity results for several materials:

Table 1: Thermal conductivity inversion results

Material	Specific Heat	Density	Real Thermal Conductivity	PSO's result
Aluminum, 6061,Temper-O	900	2710	180	180
Brass, Red(%85Cu,%15Zn)	380	8800	151	151
Invar,4J36(%65Fe,%35Ni)	515	8130	13.8	14
Nichrome,(%80Ni,%20Cr)	460	8400	12	12
Pure iron	460	7900	71.8	72

From the above simulation test results, it can be seen that the model's inversion of the thermal conductivity of the material is very accurate in this simulation mechanism.

Then, in practical applications, the end-state temperature distribution of the material can be obtained directly through thermal experimental measurements, and then the data can be imported into the optimization program, which can obtain the inverse results of the thermal conductivity of the material.

5 Model limitations and future perspectives

Although the model can give good inversion results in that two-dimensional space, the real problem is more complex.And in real problems, the material thermal conductivity may change during the heat transfer process (e.g., steel melting process), which involves the analytical inversion of the local thermal conductivity of the material, whereas our model is based on the consideration of the global thermal conductivity, i.e., assuming that the material state of matter does not change and the global thermal conductivity remains unchanged.Then there are future improvements that can be made to extend the two-dimensional space to a real three-dimensional space, which of course requires more computing resources to support. Meanwhile, the discretization of the heat conduction equation is optimized to take a more accurate numerical method, and the heat source term $S_{(t,x,y)}$,obtain the complete heat conduction equation:

$$\rho c \frac{\partial T(x,y,t)}{\partial t} = \nabla \cdot (k_{(t)} \nabla T(x,y,t)) + S$$

and the thermal convection coefficient are introduced, which is more in line with the heat conduction process in reality. Then we can adjust the optimization algorithm so that it can handle the large amount of experimental data more quickly.