拓扑优化

孙天阳

中国科学技术大学数学科学学院

tysun@mail.ustc.edu.cn

2025 年 2 月 20 日

目录

Chapter 1

宏观结构设计

拓扑优化是一种结构优化方法, 通过确定材料在空间中的分布, 使得结构在满足某些约束条件（如体积、应力等）下, 达到最优的性能指标.

$$
\begin{tikzcd}
 & \text{边界条件} \arrow[dr] & & & \text{约束条件} \arrow[dr] & \\
 & & \text{有限元方程} \arrow[r] & \text{物理场} \arrow[r] & \text{目标函数} \arrow[r] & \text{更新设计变量} \\
\text{设计变量} \arrow[r] & \text{材料属性} \arrow[ur] & & & &
\end{tikzcd}
$$

1　第一个例子

给定一个结构, 给定材料的属性（如杨氏模量和泊松比）, 给定固定的点, 给定载荷（也就是什么位置受什么方向什么大小的力）, 可以通过线弹性静力学的方程组及有限元方法求解出该结构在该边界条件下的位移场. 我们希望设计一个轻量化结构在该载荷下抵抗变形的能力最强. 回忆

$$\int_\Omega C_{ijkl}\epsilon_{kl}(u)\epsilon_{ij}(v)\mathrm{d}x = \int_\Omega f_i v_i \mathrm{d}x + \int_{\Gamma_2} g_i v_i \mathrm{d}S \Longleftrightarrow V^T K U = V^T F.$$

我们选择结构在载荷下达到平衡时的弹性势能或者叫应变能 $U^T KU/2$ 作为抵抗变形的能力的量度, 弹性势能越小我们认为抵抗变形的能力越强. 常数不起作用所以可以略去, 再结合方程 $KU = F$, 所以最后我们的目标函数是 $c = F^T U$, 被称作柔度.

我们考虑的集合是形状不超出某个边框的所有结构, 而目标函数就是定义在这个集合上的函数, 我们要找这个函数的最小值, 这当然是一个优化问题. 现在要给所有结构的集合一个数学表达, 简单起见, 设考虑的区域 Ω 是一个矩形, 将其划分为 nely 行 nelx 列的正方形小单元, 那么所有的结构的集合就是 $\{0,1\}^{\mathrm{nelx}\times\mathrm{nely}}$, 也就是说我们只需要决定每个单元有没有材料, 0 代表没有材料 1 代表有材料, 我们就决定出了一个结构. 这显然是一个离散优化问题, 但我们并不把它当成一个离散优化问题来做（可能是因为计算效率的问题, 暂时我给不了什么 comment）, 而是使用取值在 $[0,1]$ 的连续设计变量（以便利用梯度信息）, 再通过各种技术得到离散的结果.

算法倾向于添加材料以最小化目标函数. 为了对这一点建立直观感受, 可以思考弹簧的例子. 在相同的外力下, 弹簧的弹性系数 k 越大, 平衡状态时的弹性势能就越小. 而根据我们之后要介绍的材料属性的插值模型, 设计变量的值越靠近 1, 该单元的杨氏模量就越大. 所以我们其实是在“轻量化”

和“高刚度”这两个矛盾的目标之间寻求一个平衡, 这也是拓扑优化方法的魅力所在. 为了避免收敛到每个位置都填充材料的平凡解, 为了实现“轻量化”的目标, 我们要给优化问题加上体积约束.

$$
\begin{tikzcd}
 & \text{边界条件} \arrow[dr] & & & \text{体积约束} \arrow[dr] & \\
 & & \text{有限元方程} \arrow[r] & U \arrow[r] & c = F^T U \arrow[r] & \text{更新设计变量} \\
x \arrow[r] & E, \nu \arrow[ur] & & & &
\end{tikzcd}
$$

边界条件

接下来思考边界条件的施加. 我想说边界条件的施加有两类方式, 第一类是在微分方程的阶段提出明确的边界条件, 再看根据有限元的操作能对这些节点向量如 U, F 说些什么. 第二类是不管具体的细节, 直接对节点向量提要求.

$$\int_\Omega C_{ijkl}\epsilon_{kl}(u)\epsilon_{ij}(v)\mathrm{d}x = \int_\Omega f_i v_i \mathrm{d}x + \int_{\Gamma_2} g_i v_i \mathrm{d}S$$

这个式子已经用上了 Γ_1 的 Dirichlet 边界条件和 Γ_2 的 Neumann 边界条件. 通过有限元得到向量的等式 $V^T K U = V^T F$, 对任意满足 Dirichelt 边界条件的 V（也就是有些位置的值需要是 0）成立. 这个等式是左侧很多项的求和最后等于右侧很多项的求和, 但因为 V, U 要满足 Dirichlet 边界条件所以其中很多项都为 0, 丢掉这些为零的项等式依然成立, 并且剩下的项依然能写成矩阵相乘的形式, 这时我们得到 $\tilde{V}^T \tilde{K} \tilde{U} = \tilde{V}^T \tilde{F}$, 此时的 $\tilde{V}$ 是一个任意的向量, 我们可以根据它的任意性消去它得到 $\tilde{F} = \tilde{K}\tilde{U}$. 注意是不能够根据 $V^T K U = V^T F$ 得到 $F = KU$ 的, 因为这里的 V 不具有任意性. 综上所述我们在有限元编程中施加 Dirichlet 边界条件的方式就是指定 U 的哪些位置为 0 然后解方程的时候丢掉相应的位置. 而我们施加 Neumann 边界条件的方式也并不是指定面力 g 然后仔细去算会得到什么样的 F, 而是直接对 F 赋予想要的值. 这个操作也是解释的通的, 因为如果物理上只在某个节点施加某个方向的力, 去算相应的面力 g 应该会算出来狄拉克测度 δ, 再按部就班根据有限元去算应该正好能算出来相应的节点向量 F.

材料插值：SIMP

优化方法：OC

2 移动渐近线法

3 柔性逆变机构

4 Min-Max 叙述

5 鲁棒拓扑优化

6 应力约束

7 局部体积约束

8 3 维拓扑优化

Chapter 2

微结构设计