

RAE2822 翼型减阻优化

一. 优化问题描述

该问题旨在研究在工况为攻角 2.31° ，马赫数 0.729，雷诺数 $6.4e+6$ 的情况下，对 RAE2822 翼型进行减阻优化。由于 MATLAB 无法直接进行气动力的计算，引入了外部 main 应用程序，根据该程序读取的固定几何参数格式，对每次迭代后的翼型坐标进行输入，从而得到相应的阻力系数。该问题解决的步骤如下：

1. 采用 CST 参数化方法对 RAE2822 翼型进行几何建模，通过 12 个控制参数（上、下翼面各 6 个）完整表征翼型外形特征。在参数化过程中，采用余弦变换取点策略，在 $[0,1]$ 区间内生成非均匀分布的离散点，使采样点在前缘和后缘区域更加密集，从而更精确地捕捉翼型关键几何特征。这些 CST 参数构成了优化问题的设计变量空间，每个参数对应 Bernstein 多项式的一项系数。在优化过程中，算法通过智能搜索这 12 维参数空间，寻找能够使目标函数（阻力系数）最小化的最优参数组合，同时满足给定的升力性能和几何厚度约束条件。这种基于余弦变换取点的参数化方法不仅保证了翼型几何关键区域的高精度描述，还维持了整体几何的光滑性和合理性。

2. 按照指定的几何参数格式，将翼型外形坐标输入到文本框中。之后，main 应用程序会读取这些输入数据并开展计算工作，随后在 result 文本框输出升力系数、阻力系数以及厚度系数。在遗传算法的主循环里，针对种群评估部分实施了并行化改造，使得每个个体的评估任务就能被分配到多个核心上并行执行，从而显著加快评估速度。此外，在处理多个翼型数据时，先把所有翼型数据拼接成一个完整的字符串，再一次性将该字符串写入文件，有效减少了文件读写的时间。

3. 利用遗传算法对阻力系数进行寻优，在每次迭代过程中都会输出新翼型的外形参数，在计算气动系数后返回给 MATLAB 进行优化。

4. 在目标函数 Target 里，会对升力系数 C_l 与厚度系数 C_m 展开检查。一旦检测到不满足预设条件的情况，就会施加惩罚项。具体而言，当翼型的升力系数和厚度系数均符合既定要求时，该翼型会被保留，进入后续的遗传算法流程。反之，若有任何一项不满足要求，就会将其目标函数值乘以一个较大的惩罚因子。在遗传算法的优化进程中，这种惩罚机制发挥着关键作用，不满足约束条件的个体（即翼型），其目标函数值会显著增大。由于在选择操作时，算法倾向于选择目标函数值较小的个体，因此这些不满足条件的个体被淘汰的概率大幅增加，进而算法会不断朝着既能满足约束条件，又能使阻力系数达到较小值的方向进行搜索，最终找到最优解。

二. 数学模型建立

2.1 CST 参数化翼型

CST 方法通过类别函数和形状函数相结合的方式描述翼型几何形状，其中，类别函数描述了翼型的通用形状特征，形状函数则通过一组 Bernstein 多项式系数来对翼型的具体形状进行调整，通过将这两个函数相乘，再加上后缘厚度修正项，就可以得到翼型上下表面的坐标。

在进行参数化时，选取 RAE2822 作为初始翼型，作为常见的圆头尖尾翼型，将其类别函数指数 N_1 、 N_2 和后缘厚度修正设置为默认值，该数值分别为 0.5、1.0。在空气动力学中，用于描述翼型通用形状特征的表达式为：

在上述类别函数中，假设 X_i 为一个向量 $X = [x_1, x_2, \dots, x_n]$ ， N_1 和 N_2 是标量值， C_i 为计算结果向量中的第 i 个元素，通过不同 N_1 和 N_2 的取值可以得到不同的基础形状。

在计算形状函数时， $S(x)$ 由一组 Bernstein 多项式组成。Bernstein 多项式是一种常用的多项式基函数，其表达式为 $B = nchoosek(n, i) \times x^i \times (1 - x)^{n-i}$ ，其中 $nchoosek(n, i)$ 是组合数，表示从 n 个元素中选择 i 个元素的组合方式。

为了确保翼型的后缘闭合，将上表面和下表面的最后一个坐标值取平均值，作为修正后的后缘坐标。该算法通过 CST 方法，利用类别函数和形状函数的组合，结合后缘厚度修正，生成了翼型的上下表面坐标。通过调整输入的 Bernstein 多项式系数，可以灵活地改变翼型的形状，适用于翼型设计和优化等领域。

2.2 遗传算法

2.2.1 算法原理

遗传算法是一种模拟生物进化过程的全局优化算法，其理论基础源于达尔文的“自然选择”和孟德尔的“遗传机制”。它通过模拟种群中个体的选择、交叉、变异等操作，逐步朝着问题的最优解逼近。

该算法的核心思想包括：一是适应度，即用于衡量个体优劣，通常可借助目标函数值来体现；二是染色体编码，其作用是把解空间映射为基因型，常见的如二进制串；三是种群进化，通过迭代的方式不断生成更优的下一代种群。

遗传算法具有诸多特点：它无需目标函数的梯度信息，这使得它能适用于不可导、非凸以及多峰函数的优化问题；具备强大的全局搜索能力，通过引入随机性有效避免陷入局部最优；还具有并行性，可同时对多个解进行评估，从而提高

优化效率。

2.2.2 关键操作步骤

1. 编码与初始化

编码方式：把设计变量转换为染色体，常见的编码方式有二进制编码和实数编码等。

初始化种群：随机生成若干个体，由这些个体构成初始种群。

2. 选择

从当前种群里挑选适应度高的个体进入交配池，采用锦标赛选择方法。此方法的核心思路是从种群里随机挑选若干个体组成一个“锦标赛小组”，接着选出适应度最优的个体作为父代，参与后续的交叉和变异操作。

3. 交叉

为了让子代个体能够继承父代的优良特性，同时产生新的基因组合，代码采用了单点交叉的方法。在进行交叉操作时，会随机挑选一个交叉点，该交叉点将父代的染色体一分为二，然后交换两个父代在交叉点之后的基因片段，从而生成两个全新的子代个体。这种单点交叉方式能够有效促进基因的交流与重组，增加种群的多样性，推动算法朝着更优解的方向搜索。

4. 变异

引入随机扰动以增加种群的多样性，常采用单点变异，也就是随机翻转某一位基因，更好实现全局寻优这一过程。

5. 终止条件

最大迭代次数：达到预设代数 $(i)_{\max}$ 。

适应度收敛：种群适应度标准差 $S_f \leq (S_f)_{\max}$ 。

最优解稳定：连续多代最优解无显著改进。

三. 优化框架和过程描述

3.1 翼型优化框架

该翼型优化使用遗传算法，通过迭代优化翼型的 CST 参数，以最小化翼型的阻力系数 Cd 为目标。整个优化框架主要由以下几个部分组成：

初始化部分：设置初始翼型的 CST 参数，生成目标翼型的坐标数据，为遗传算法设置参数，包括种群大小、最大迭代次数、交叉概率、变异概率等。

遗传算法部分：作为核心优化算法，负责在解空间中搜索最优的 CST 参数

组合。通过选择、交叉、变异等操作不断进化种群，逐步逼近最优解。

目标函数部分：首先依据当前的 CST 参数，通过特定的数学模型生成全新的翼型。随后，将生成的翼型数据以规定格式写入指定文件，为后续计算提供基础。紧接着，调用外部程序 `main.exe` 对写入的翼型数据进行解析与计算，从而获取该翼型对应的阻力系数 C_d 。在此过程中，代码内置了完善的文件读取机制，能够有效处理文件缺失、数据损坏等异常情况，确保计算流程的稳定性。为便于直观观察优化进程，目标函数还集成了可视化功能，可实时呈现优化过程中的关键数据变化。在整个优化迭代过程中，传输给 `main` 应用程序的翼型几何外形参数会随着每次迭代而实时更新。每次迭代完成后，新计算得到的翼型参数都会被及时保存至特定的文本框中，有助于更好地分析和调控优化过程。

结果处理部分：绘制遗传算法的收敛曲线，展示优化过程中阻力系数的变化；绘制优化后的翼型，直观呈现优化效果。

翼型优化框架图如图 1 所示：

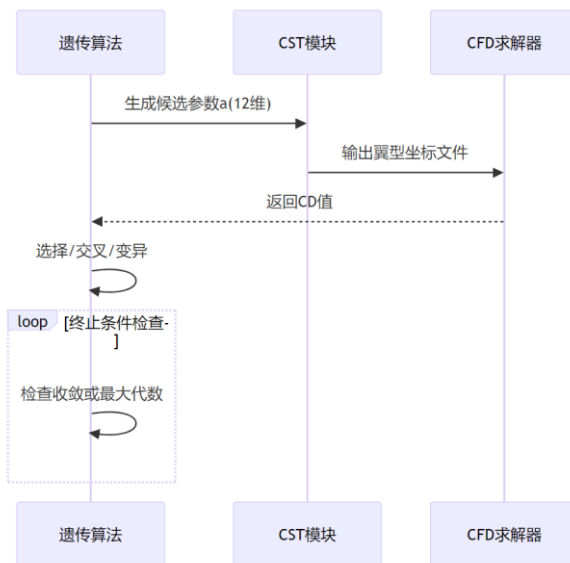


图 1 翼型优化框架图

3.2 翼型优化过程描述

1.翼型参数初始化：

定义初始翼型的 CST 参数 a_{up} 和 a_{low} ，将它们组合成初始参数向量 a_0 。通过 `cst_airfoil` 函数计算初始翼型的上、下表面 y 坐标，得到初始翼型的完整坐标数据。将初始翼型的坐标数据作为目标翼型，定义目标函数 f ，该函数接受 CST

参数向量 \mathbf{a} 作为输入，调用 `Target` 函数计算该参数下翼型的阻力系数 C_d 。

2. 运行遗传算法

种群初始化：根据设置的参数，随机生成初始种群，每个个体代表一组 CST 参数。在每一代迭代中，执行以下操作：

- 1) 选择操作：根据个体的适应度（即目标函数值，这里为阻力系数 C_d ），使用锦标赛选择方法从当前种群中选择适应度高的个体进入交配池。
- 2) 交叉操作：以一定的交叉概率对交配池中的个体进行单点交叉，交换部分基因，生成新的个体。
- 3) 变异操作：以一定的变异概率对新生成的个体进行单点变异，随机翻转某一位基因，增加种群的多样性。
- 4) 评估适应度：对新生成的种群中的每个个体，调用目标函数 `Target` 计算其阻力系数 C_d ，作为适应度值。
- 5) 终止判断：检查是否满足终止条件，如达到最大迭代次数、种群适应度收敛或最优解稳定。如果满足终止条件，则停止迭代；否则，继续下一轮迭代。

3. 目标函数计算

1) 生成新翼型数据：在目标函数 `Target` 中，对于输入的每个个体的 CST 参数矩阵，通过循环遍历每个个体。使用 `cst_airfoil` 函数根据当前个体的 CST 参数 \mathbf{a} 计算翼型的上下表面坐标 y_{up} 和 y_{low} 。同时，调用 `calculate_thickness` 函数计算每个翼型的厚度。

2) 写入翼型数据文件：将所有个体的翼型数据整理后，将这些数据批量写入文件 `geo_example.txt`，以便外部程序读取。

3) 调用外部程序计算：调用外部程序 `main.exe`，该程序会读取 `geo_example.txt` 中的翼型数据，并计算每个翼型的升力系数 C_l 和阻力系数 C_d ，将结果写入相应的结果文件中。

4) 读取计算结果：通过循环遍历每个个体，从结果文件中读取每个翼型的升力系数 C_l 和阻力系数 C_d 。

5) 约束检查与目标函数计算：检查每个翼型的升力系数 C_l 和厚度 C_m 是否满足约束条件。如果不满足约束条件，则在阻力系数 C_d 的基础上施加惩罚项；如果满足约束条件，则目标函数值 $f(i)$ 直接取阻力系数 C_d 。

6) 返回目标函数值：最终，目标函数 `Target` 返回一个包含所有个体目标函数值的向量 \mathbf{f} ，这些值将用于遗传算法中的适应度评估，以指导算法的进化方向，逐步找到满足约束条件且阻力系数最小的最优翼型。

4. 结果处理

- 1) 收敛曲线绘制：绘制遗传算法的收敛曲线，展示迭代过程中阻力系数 C_d

的变化情况，帮助分析算法的收敛性能。

2) 翼型绘制：绘制优化后的翼型，直观展示优化效果。

四、遗传算法的选择原因及参数调整

4.1 选择原因

1.遗传算法模拟生物进化过程，通过选择、交叉和变异等操作在解空间中进行搜索。它不依赖于目标函数的梯度信息，能够在复杂的、多峰的解空间中探索不同区域，具有很强的全局搜索能力，相比基于梯度的方法，更有可能找到全局最优解，而不是陷入局部最优陷阱。

2.对于目标函数的连续性、可导性等没有严格要求，无论是连续函数、离散函数，还是具有噪声、不连续的函数，遗传算法都能适用。在处理一些传统优化方法难以解决的复杂问题，如组合优化问题、多目标优化问题时，遗传算法展现出良好的适应性和鲁棒性。

3.遗传算法每次迭代操作是对一个种群进行处理，种群中的多个个体可以并行计算，这种天然的并行性使得遗传算法可以充分利用多核处理器或分布式计算环境，提高计算效率，尤其适用于大规模优化问题。

4.2 参数调整

1. 种群规模决定了遗传算法在每一代中包含的个体数量。较大的种群规模意味着算法能够在更广泛的解空间中进行搜索，增加找到全局最优解的可能性，但同时也会增加计算量和计算时间；较小的种群规模计算效率较高，但可能导致算法过早收敛，陷入局部最优。因此，在综合考虑问题的规模和复杂度，种群数量设置为 500。

2. 交叉操作是遗传算法中产生新个体的主要方式，交叉概率决定了种群中个体进行交叉操作的比例。较高的交叉概率可以增加种群的多样性，促进算法在解空间中进行更广泛的搜索，但过高的交叉概率可能导致优良个体的基因被破坏；较低的交叉概率则会使算法的搜索能力下降，收敛速度变慢，在本算法中将交叉概率设置为 0.85。

3. 变异操作是遗传算法中维持种群多样性的重要手段，适当的变异概率可以防止算法过早收敛，帮助算法跳出局部最优解；但变异概率过高会使算法退化为随机搜索，降低收敛速度。为了使得算法形成稳定的进化方向，增加对种群多样性的贡献程度，在本算法中变异概率为 0.01。

4.最大迭代次数分析同准牛顿迭代法，经过综合考虑，最大迭代次数取 100 次。

五、优化结果分析

当最大迭代次数设置为 100 次时，优化后翼型和遗传算法收敛曲线见图 2 和图 3：

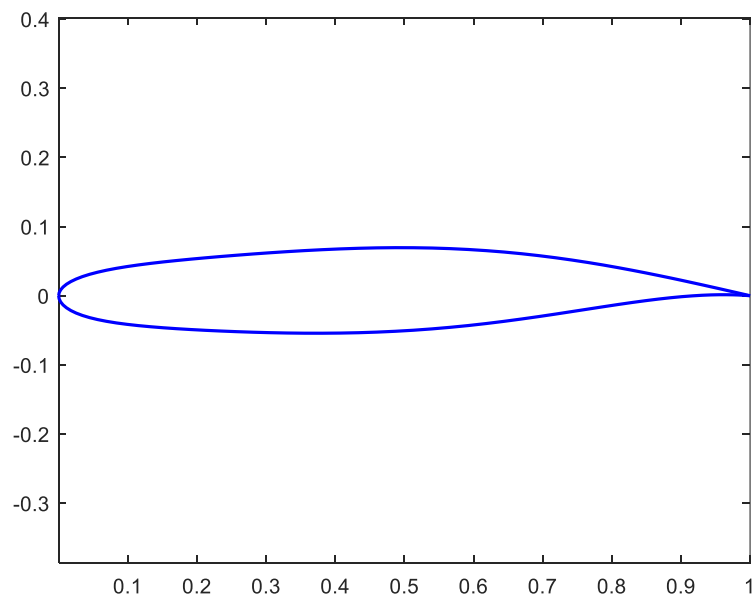


图 2 优化后翼型图

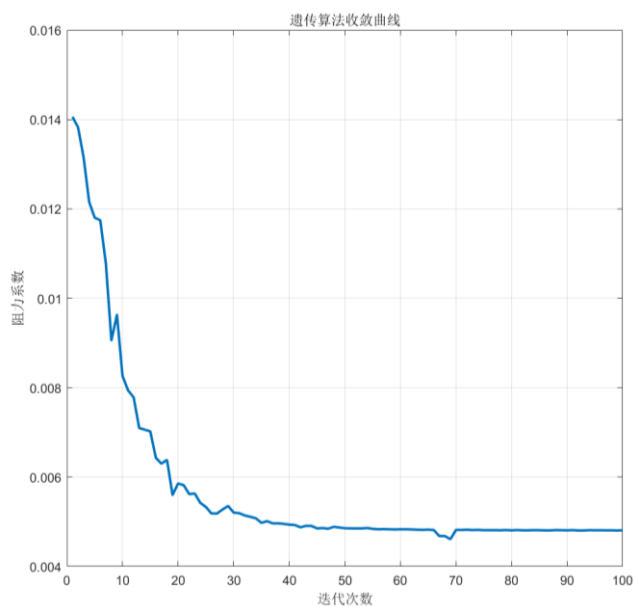


图 3 遗传算法收敛曲线

《飞行器优化设计》课程报告

从遗传算法收敛曲线特征分析，在迭代初始阶段，阻力系数 C_d 以较快速度下降，随着优化进程推进，曲线出现轻微波动，最终收敛至稳定数值区间。曲线会存在一些波动是因为遗传算法在搜索过程中，通过选择、交叉和变异等操作来更新种群，每次操作都可能引入一定的随机性，导致目标函数值在短期内出现波动；或是因为目标函数可能存在多个局部最优解，随着迭代次数增加，算法在搜索过程中可能会在不同的局部最优解之间切换。当从一个局部最优解附近跳到另一个局部最优解附近时，目标函数值会发生较大变化，导致收敛曲线出现波动。

由于翼型形状的特殊性，其头部和尾部的曲率变化较大，而中间部分的曲率变化很小，因此如果均匀分配坐标点，会导致翼型头部无法被准确描述，进而出现“尖头”现象。为了解决这一问题，需要改变坐标分配方式，代码中引入了与余弦分布，增加翼型头尾部的坐标点，从而使参数化翼型描述更加精准。

另外，为了提高算法的优化效率，优化过程中使用的并行计算。在遗传算法中，每一代个体都需要计算种群个数次适应度，若每一次计算都单独调用一次目标函数，就会导致总优化时间非常长。为了解决该问题，将算法模式改为同时将一个种群的所有数据传入 `main` 程序，这样只需要调用一次 `main` 程序，就可以一次性计算所有的个体适应度，从而大幅提升优化效率。

为了保证优化过程满足约束条件，算法采用了罚函数法，当优化结果导致新翼型厚度或升力系数低于基准翼型时，就会加一个惩罚项，从而剔除该数据。这样的操作保证了优化过程始终在约束条件下进行。

通过 100 次迭代过程，最终得到最优参数化系数为 0.1600、0.1226、0.1637、0.2400、0.2500、0.2300、-0.1700、-0.1000、-0.1550、-0.1700、-0.1500、0.0800，

最优阻力系数为 0.004817，优化后的翼型通过调整 CST 参数，优化了上、下翼面轮廓，从而改善气动性能，翼型在保持基本外形的同时，对局部曲率进行调整，减少流动分离或摩擦阻力，最终降低 C_d 。同时，与原阻力系数数值进行对比，实现了翼型减阻优化这一过程。