

作业 1 第三题报告

我实现的计算 Griewank 函数的梯度矩阵程序的输入格式为 (n 行 m 列), 具体实现见附件代码。输入矩阵为 n 行, 表示每个点 (计算实例) 包含 n 个维度, 或者说包含 n 个自变量; 而输入矩阵 m 列, 表示总共有 m 个点 (实例) 需要计算, 也就是说该程序可以实现批量计算梯度的效果。由于对于一个 n 自变量的点, 其梯度为一包含 n 个元素的向量, 所以对于总共 m 个点, 最终输出的矩阵维度同样为 (n 行 m 列), 每一列即表示对应点的梯度向量。

同理对于 Griewank 函数的 Hessian 矩阵计算程序, 为了实现批量计算我们同样设置输入矩阵形状为 (n 行 m 列), 即一次计算 m 个点, 每个点包含 n 个维度。但是不同于梯度是一个向量, 维度为 (n 行 1 列), 每个点的 Hessian 矩阵维度为 (n 行 n 列), 所以对于批量计算 Hessian 矩阵的程序, 最终输出结果中所有元素的数目之和应该为 $n*n*m$ 。为了更直观的理解输出 (每个点处的 Hessian 矩阵维度为 (n 行 n 列)), 所以我在程序中将输出形状设置为 (n 行 n 列 m 页), 也就是说输出的每一页 $H(:, :, i)$ 即代表输入中第 i 个点 (即输入矩阵的第 i 列) 处的 Hessian 矩阵。

对于上述梯度和 Hessian 矩阵的计算, 我在附件 (main.mlx) 这个实时脚本中均进行了批量点的计算测试, 并且与 Matlab 自带的相应函数进行了对比测试, 结果表明我自己实现的计算程序在计算结果方面的正确性, 并且对于批量计算的支持使得其使用更加方便。

附件中 main.mlx 为本次作业主程序, Griewank.mlx 为批量计算 Griewank 值的函数, Gradient_Griewank.mlx 为梯度计算函数, Hessian_Griewank.mlx 为 Hessian 矩阵计算函数, surf_func.mlx 为函数曲面绘图函数。