

优化理论与方法实验报告

实验二：约束优化算法实践

周旸

2023302011158

2023 级信息与计算科学

日期：2025 年 12 月 31 日

摘要

本文围绕典型非线性约束优化问题，基于 Matlab 实现并比较了增广拉格朗日方法（Augmented Lagrangian Method, ALM）与序列二次规划方法（Sequential Quadratic Programming, SQP）的数值表现。实验选取两个标准带约束 Rosenbrock 类测试算例：二维强非线性等式/不等式约束的 Constrained_Modified_Rosenbrock_2d，以及四维线性等式与混合不等式约束的 Constrained_Rosenbrock_4d；在统一收敛判据（以 KKT 残差为准，阈值 10^{-6} ）和多组初始点设置下，分别采用黄金分割（golden）与 Armijo–Wolfe 强 Wolfe 准则两种线搜索策略进行全局化，并记录外层迭代次数、目标函数值、可行性误差与 KKT 误差等指标，结合优化路径与收敛曲线进行分析。

数值结果表明：两种算法在所有初值与两类线搜索组合下均能稳定收敛到满足阈值的可行 KKT 点，显示出较好的数值鲁棒性；SQP 在接近解时通常呈现更快的局部误差下降，且迭代点更倾向于快速贴近并沿可行集推进，而 ALM 往往表现为“先可行后最优”的阶段性收敛过程。对于二维非凸算例，由于可行曲线上存在多个局部极小点，不同初值与线搜索策略会导致算法落入不同吸引域，从而出现收敛到不同极小点的现象；而四维算例在最优点处不等式约束多为严格可行，使得两种算法最终收敛终点更一致。本文的对比实验直观揭示了 ALM 与 SQP 在收敛速度、鲁棒性、参数敏感性与计算代价上的差异，为约束优化算法的实现细节与选型提供了参考。

关键词：约束优化；增广拉格朗日法；序列二次规划；KKT 条件；线搜索；Rosenbrock 测试函数

目录

1 引入	3
1.1 实验内容	3
1.2 实验目的	3
2 实验的理论基础	4
2.1 算例	4
2.1.1 2D 算例 Constrained_Modified_Rosenbrock_2d	4
2.1.2 4D 算例 Constrained_Rosenbrock_4d	6
2.2 线搜索算法	6
2.3 优化算法	6
2.3.1 增广 Lagrange 方法 (ALM)	7
2.3.2 序列二次规划方法 (SQP)	8
3 数值实验	8
3.1 实验设置	8
3.2 Constrained Modified Rosenbrock 2D	9
3.2.1 初值点选取与可行域结构	9
3.2.2 数值结果汇总 (含两种线搜索)	10
3.2.3 优化路径对比	10
3.2.4 收敛曲线与现象解释	12
3.3 Constrained Rosenbrock 4D	14
3.3.1 初值点选取	14
3.3.2 数值结果与收敛行为	14
3.4 小结: ALM 与 SQP 的优劣对比	17
4 思考题	18
5 总结与反思	20
5.1 总结	20
5.2 反思	20
参考文献	21

1 引入

1.1 实验内容

本实验围绕典型约束优化算法在标准带约束测试函数上的数值实现与性能比较展开。

1. 约束优化算法实现：基于课程内容，在 Matlab 中编程实现以下两类核心算法，并给出必要的线搜索/参数调整策略（如罚参数更新、乘子更新、子问题求解与步长控制等）：

- (a) 增广拉格朗日函数方法 (Augmented Lagrangian Method, ALM)
- (b) 序列二次规划方法 (Sequential Quadratic Programming, SQP)

实现中需包含目标函数值与梯度计算、约束函数与约束雅可比矩阵计算等基础模块，并保证代码结构清晰、注释完整。

2. 对比实验：按照实验说明统一设置初始点（可自行选择多组以测试鲁棒性）、收敛判据与最大迭代次数。收敛判据采用

$$\|\nabla f(x_k)\| \leq 10^{-6}, \quad \|c(x_k)\| \leq 10^{-6},$$

其中等式约束直接计入，不等式约束按 $\max(0, c(x_k))$ 计入；最大迭代步数取 $k_{\max} = 1000$ 。在两类测试函数上分别运行 ALM 与 SQP：

- (a) Constrained_Modified_Rosenbrock_2d (2D, 含强非线性等式约束与非线性不等式约束，并给定变量可行域范围)；
- (b) Constrained_Rosenbrock_4d (4D, 含线性等式/不等式约束与弱非线性不等式约束的混合约束)。

记录并比较各算法的迭代次数、收敛点、目标函数值、梯度范数、约束违反度（可行性误差）以及计算效率等指标，并整理成表格进行定量分析。

3. 可视化与分析：在数值结果基础上进行定性对比与理论解释：

- (a) 绘制两种测试函数上的收敛曲线（纵轴： $\log(\|\nabla f(x_k)\| + \|c(x_k)\|)$ ，横轴：迭代次数），比较不同算法的收敛速度与稳定性；
- (b) 对 2D 测试函数绘制优化路径与约束边界图：在等高线图上叠加等式/不等式约束边界，标注可行域，并给出从初始点到最优点的搜索轨迹；

4. 思考题与理论讨论：结合实验现象回答并解释：

- (a) 增广拉格朗日法将约束融入目标函数后，为何通常比单纯罚函数法具有更高的收敛精度？罚参数选择对算法稳定性与收敛行为有何影响？
- (b) SQP 在处理强非线性约束时往往表现更优，其“二次规划子问题逼近局部约束优化特性”的优势如何在 2D 测试函数（强非线性等式约束与峡谷结构交叉）上体现？

1.2 实验目的

1. 掌握约束优化中的典型数值算法：增广拉格朗日函数方法 (ALM) 与序列二次规划方法 (SQP)，理解其约束处理机制与实现要点（如罚参数转化、拉格朗日乘子更新、二次规划子问题近似以及与线搜索/步长控制的配合等）。
2. 通过编程实现 ALM 与 SQP，在带约束测试函数（2D 强非线性约束与 4D 混合约束）上开

展对比实验，比较两种算法在收敛速度、迭代次数、收敛精度、约束满足程度与计算效率等方面差异，并通过表格与可视化结果进行总结。

3. 理解并分析约束优化问题的关键影响因素：约束类型（等式/不等式、线性/非线性）、约束强度与可行域形状对算法性能的作用；进一步阐明 ALM 与 SQP 在约束处理策略、子问题构造与数值稳定性方面的本质差别，明确不同约束优化算法的适用场景。

2 实验的理论基础

2.1 算例

本次实验采用两个带约束算例：`Constrained_Modified_Rosenbrock_2d` 与 `Constrained_Rosenbrock_4d`。选择这两个算例的原因包括：

1. 两个算例的目标函数均基于 Rosenbrock 类构造，具有弯曲且狭窄的山谷结构，能够有效检验约束优化算法在强各向异性、非凸地形下的数值稳定性；
2. `Constrained_Modified_Rosenbrock_2d` 维度低，约束为等式曲线叠加不等式裁剪，可直观绘制等高线 + 约束边界 + 可行域，便于解释 ALM/SQP 的搜索行为；
3. `Constrained_Rosenbrock_4d` 同时包含线性等式/线性不等式/弱非线性不等式的混合约束，可用于比较算法在更高维与更复杂约束组合下的性能。

为在数值实验前大致了解约束优化算例的极小点情况，以及在数值实验中提供极小点的分类参考，除了 `Constrained_Modified_Rosenbrock_2d` 算例可以直接证明全局极小点为 $x^* = [1, 1]^\top$ 外，我们将通过可行域网格搜索得到若干网格意义下的局部极小参考点，并识别全局极小参考点。需要强调的是：这些坐标与函数值为数值近似，随网格精度可略有变化。

2.1.1 2D 算例 `Constrained_Modified_Rosenbrock_2d`

目标函数为

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2 + 5 \sin(2\pi x_1) \sin(2\pi x_2),$$

约束为

$$\begin{cases} c_1(x) = x_1^2 + x_2 - 2 = 0, \\ c_2(x) = x_1 + x_2^2 - 3 \leq 0, \\ x_1 \in [-1.5, 2], x_2 \in [-0.5, 3]. \end{cases}$$

真实全局极小点的理论判别 我们将证明 $x^* = [1, 1]^\top$ 一定是 `Constrained_Modified_Rosenbrock_2d` 约束优化算例的全局极小点。由等式约束 $c_1(x) = 0$ 可消元得 $x_2 = 2 - x_1^2$ ，从而将问题限制在一维可行曲线上。记 $u = x_1$ ，并定义

$$\phi(u) = f(u, 2 - u^2) = R(u) + S(u),$$

其中

$$R(u) = (1 - u)^2 + 100((2 - u^2) - u^2)^2 = (1 - u)^2 + 400(1 - u^2)^2 \geq 0,$$

$$S(u) = 5 \sin(2\pi u) \sin(2\pi(2-u^2)), \quad |S(u)| \leq 5.$$

显然 $\phi(1) = 0$, 即 $x = [1, 1]^\top$ 为可行点且 $f(1, 1) = 0$ 。

下面说明 $x = [1, 1]^\top$ 不仅是可行点, 而且在可行曲线上为严格局部极小点, 并进一步推出其为全局极小点。令 $u = 1 + t$ (t 充分小), 则 $x_2 = 2 - u^2 = 1 - 2t - t^2$ 。对主体项可得

$$R(1+t) = t^2 + 100(-4t - 2t^2)^2 = 1601t^2 + O(t^3).$$

另一方面, 由基本不等式 $|\sin z| \leq |z|$, 注意到

$$\sin(2\pi u) = \sin(2\pi(1+t)) = \sin(2\pi t), \quad \sin(2\pi x_2) = \sin(2\pi(x_2 - 1)),$$

从而当 $|t| \leq 1$ 时有

$$|\sin(2\pi u)| \leq 2\pi|t|, \quad |\sin(2\pi x_2)| \leq 2\pi|x_2 - 1| = 2\pi|2t + t^2| \leq 6\pi|t|.$$

于是正弦项满足二次上界

$$|S(1+t)| \leq 5 \cdot (2\pi|t|) \cdot (6\pi|t|) = 60\pi^2 t^2.$$

综合上述估计, 得到当 $|t|$ 足够小时

$$\phi(1+t) = R(1+t) + S(1+t) \geq (1601 - 60\pi^2)t^2 + O(t^3) > 0 \quad (t \neq 0),$$

因此 $u = 1$ (即 $x = [1, 1]^\top$) 在可行曲线上为严格局部极小点。

接着证明其为全局极小点。若存在可行点使 $\phi(u) < 0$, 由于 $S(u) \geq -5$, 必有 $R(u) < 5$ 。而 $R(u) = (1-u)^2 + 400(1-u^2)^2$ 在 $u = 1$ 附近按 t^2 迅速增长, 从而 $R(u) < 5$ 只能发生在 u 极接近 1 的邻域内; 但在该邻域内上式已证明 $\phi(u) \geq 0$, 与 $\phi(u) < 0$ 矛盾。因此在可行集合上有 $f(x) \geq 0$, 并且 $f(1, 1) = 0$, 故 $x^* = [1, 1]^\top$ 为真实全局极小点, 最优值为 $f^* = 0$ 。

可行域内的局部极小参考点与网格意义下的全局极小参考点 在可行域网格上检测到两个极小参考点:

$$x^{(1)} \approx [-0.998993, 1.002014]^\top, \quad f(x^{(1)}) \approx 3.997994,$$

$$x^{(2)} \approx [0.999998, 1.000005]^\top, \quad f(x^{(2)}) \approx 7.54 \times 10^{-9}.$$

其中 $x^{(2)}$ 位于真实全局极小点 $[1, 1]^\top$ 的极近邻, 是网格未恰好取到 $x_1 = 1$ 时的离散近似; 因此在后续数值实验中, 我们将 $x^* = [1, 1]^\top$ 视为该算例的真实全局极小点, 同时将 $x^{(2)}$ 作为网格意义下的全局极小参考点用于对比与可视化展示。 $x^{(1)}$ 为非全局的局部极小参考点。

积极集判别 本算例中等式约束 $c_1(x) = 0$ 对所有可行点恒成立, 因此在极小点处必为活跃约束。对不等式约束代入计算得

$$c_2(x^*) = c_2([1, 1]^\top) = 1 + 1 - 3 = -1 < 0,$$

并且

$$c_2(x^{(1)}) \approx -2.994960 < 0.$$

因此真实全局极小点 $x^* = [1, 1]^\top$ 以及局部极小参考点 $x^{(1)}$ 均不在 $c_2(x) = 0$ 边界上, 不等式约束不活跃, 积极集分别为

$$\mathcal{A}(x^*) = \{c_1\}, \quad \mathcal{A}(x^{(1)}) = \{c_1\}.$$

2.1.2 4D 算例 Constrained_Rosenbrock_4d

目标函数为 ($x = [x_1, x_2, x_3, x_4]^\top$)

$$f(x_1, x_2, x_3, x_4) = \sum_{k=1}^3 \left((1 - x_k)^2 + 100(x_{k+1} - x_k^2)^2 \right),$$

约束为

$$\begin{cases} c_1(x) = x_1 + x_2 + x_3 + x_4 - 2 = 0, \\ c_2(x) = x_1 - x_2 + 1 \geq 0, \\ c_3(x) = 2x_3 - x_4 - 3 \leq 0, \\ c_4(x) = x_1^2 + x_4^2 - 4 \leq 0. \end{cases}$$

可行域内的局部极小点与全局极小点 通过可行域网格搜索并做局部加密验证后，得到两个互不相同的极小参考点：

$$x^* \approx [0.797325, 0.636425, 0.405225, 0.161025]^\top, \quad f(x^*) \approx 0.528086,$$

$$x^{(1)} \approx [-0.355875, 0.644125, 0.842775, 0.868975]^\top, \quad f(x^{(1)}) \approx 49.594804.$$

其中 x^* 为网格意义下的全局极小参考点； $x^{(1)}$ 为非全局的局部极小参考点。

积极集判别 同样地，等式约束 $c_1(x) = 0$ 在所有可行点处恒成立，必为活跃约束。对不等式约束代入计算：

$$c_2(x^*) \approx 1.160900 > 0, \quad c_3(x^*) \approx -2.350575 < 0, \quad c_4(x^*) \approx -3.338344 < 0,$$

$$c_2(x^{(1)}) = 0, \quad c_3(x^{(1)}) \approx -2.183425 < 0, \quad c_4(x^{(1)}) \approx -3.118235 < 0.$$

因此 x^* 不位于任何不等式边界上，不等式均不活跃，

$$\mathcal{A}(x^*) = \{c_1\};$$

而 $x^{(1)}$ 位于边界 $c_2(x) = 0$ 上（其余不等式仍为严格不等式），

$$\mathcal{A}(x^{(1)}) = \{c_1, c_2\}.$$

2.2 线搜索算法

在实验二中，我们将使用与实验一相同的线搜索算法：0.618 方法实现（近似的）精确线搜索，以及 Armijo–Wolfe 准则实现非精确线搜索。在此不再赘述线搜索算法的细节。

2.3 优化算法

本实验考虑一般非线性约束优化问题

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{s.t. } c_E(x) = 0, \quad c_I(x) \leq 0, \quad (1)$$

其中 $c_E : \mathbb{R}^n \rightarrow \mathbb{R}^{m_E}$ 为等式约束， $c_I : \mathbb{R}^n \rightarrow \mathbb{R}^{m_I}$ 为不等式约束。记

$$c(x) \triangleq \begin{bmatrix} c_E(x) \\ [c_I(x)]_+ \end{bmatrix}, \quad [c_I(x)]_+ \triangleq \max\{c_I(x), 0\} \text{ (逐分量)}, \quad (2)$$

则 $\|c(x)\|_2$ 可用于衡量可行性误差。

KKT 条件与收敛判据 令拉格朗日函数

$$L(x, \lambda, \mu) = f(x) + \lambda^\top c_E(x) + \mu^\top c_I(x), \quad (3)$$

其中 $\lambda \in \mathbb{R}^{m_E}$, $\mu \in \mathbb{R}^{m_I}$ 且 $\mu \geq 0$ (逐分量)。KKT 条件为

$$\begin{cases} \nabla_x L(x, \lambda, \mu) = \nabla f(x) + J_E(x)^\top \lambda + J_I(x)^\top \mu = 0, \\ c_E(x) = 0, \quad c_I(x) \leq 0, \quad \mu \geq 0, \\ \mu_i c_{I,i}(x) = 0, \quad i = 1, \dots, m_I, \end{cases} \quad (4)$$

其中 $J_E(x) = \nabla c_E(x)$, $J_I(x) = \nabla c_I(x)$ 。由于约束极小点一般不满足 $\|\nabla f(x^*)\| = 0$, 因此本实验的停机准则采用 KKT 残差:

$$r_s(x, \lambda, \mu) \triangleq \|\nabla_x L(x, \lambda, \mu)\|_2, \quad r_c(x, \mu) \triangleq \|\min(\mu, -c_I(x))\|_2, \quad (5)$$

并以

$$\max\{r_s(x_k, \lambda_k, \mu_k), \|c(x_k)\|_2, r_c(x_k, \mu_k)\} \leq \varepsilon, \quad \varepsilon = 10^{-6} \quad (6)$$

作为收敛判据。

2.3.1 增广 Lagrange 方法 (ALM)

增广 Lagrange 方法的核心思想是: 在不把罚参数推到无穷大的前提下, 通过“乘子迭代 + 适度罚项”来逼近 KKT 点。对(1), 采用 Powell–Hestenes–Rockafellar 形式的增广拉格朗日函数

$$\Phi_\sigma(x, \lambda, \mu) = f(x) + \lambda^\top c_E(x) + \frac{\sigma}{2} \|c_E(x)\|_2^2 + \frac{1}{2\sigma} (\|[\mu + \sigma c_I(x)]_+\|_2^2 - \|\mu\|_2^2), \quad (7)$$

其中 $\sigma > 0$ 为罚参数, $[\cdot]_+$ 为逐分量取正部。

外层迭代框架:

1. 给定初值 x_0 、 λ_0 、 $\mu_0 \geq 0$ 、 $\sigma_0 > 0$;
2. 对 $k = 0, 1, 2, \dots$:
 - (a) (内层) 近似求解无约束子问题

$$x_{k+1} \approx \arg \min_x \Phi_{\sigma_k}(x, \lambda_k, \mu_k).$$

本实验中内层使用 BFGS (以逆 Hessian 近似 H_k) 生成方向 $d = -H_k \nabla_x \Phi_{\sigma_k}$, 再配合线搜索得到步长 α , 更新 $x \leftarrow x + \alpha d$ 。

- (b) 更新乘子

$$\lambda_{k+1} = \lambda_k + \sigma_k c_E(x_{k+1}), \quad \mu_{k+1} = [\mu_k + \sigma_k c_I(x_{k+1})]_+.$$

- (c) 依据可行性误差是否下降来更新罚参数 (误差下降不足则增大 σ_k ; 代码中在 σ 增大时重置 BFGS 近似以避免曲率信息失效)。

- (d) 若满足(6)则停止。

2.3.2 序列二次规划方法 (SQP)

SQP 的核心思想是：在当前迭代点处对约束进行线性化、对拉格朗日函数进行二次近似，从而构造二次规划 (QP) 子问题来得到搜索方向。其形式为

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^\top B_k d + \nabla f(x_k)^\top d, \\ \text{s.t.} \quad & c_E(x_k) + J_E(x_k)d = 0, \\ & c_I(x_k) + J_I(x_k)d \leq 0, \end{aligned} \tag{8}$$

其中 B_k 近似 $\nabla_{xx}^2 L(x_k, \lambda_k, \mu_k)$ 。

外层迭代框架：

1. 给定初值 x_0 与 B_0 (本实验取 $B_0 = I$)；
2. 对 $k = 0, 1, 2, \dots$ ：
 - (a) 求解 QP 子问题 (8) 得到方向 d_k (以及子问题乘子作为乘子估计)；
 - (b) 构造罚函数型的优值函数 (merit function)

$$\phi_\rho(x) = f(x) + \frac{\rho}{2} \|c_E(x)\|_2^2 + \frac{\rho}{2} \|c_I(x)\|_2^2, \tag{9}$$

其中 $\rho > 0$ 为罚参数 (代码中取固定值，用于全局化)；

- (c) 对 $\phi_\rho(x_k + \alpha d_k)$ 做线搜索，得到步长 α_k ，更新 $x_{k+1} = x_k + \alpha_k d_k$ ；
- (d) 用 (阻尼) BFGS 更新 B_k (以保证 B_k 的正定性，便于 QP 求解与数值稳定)；
- (e) 若满足 (6) 则停止。

在具体实现中，QP 子问题由 quadprog 求解。为避免数值问题，代码中对 B_k 进行正则化与 (必要时) 修正为正定；对不等式约束，代码采用“**近似活跃集**”策略：将 $c_I(x_k)$ 中接近 0 的分量 (如 $c_{I,i}(x_k) \geq -\tau$) 并入等式集合参与子问题求解，再将乘子投影到非负正交锥以满足 $\mu \geq 0$ 。

3 数值实验

3.1 实验设置

本文对两种约束优化算法——增广拉格朗日法 (ALM) 与序列二次规划 (SQP) ——在两个测试算例上进行对比：

- **Constrained Modified Rosenbrock 2D**: 1 个等式约束 $h(x) = 0$ ，以及若干不等式约束 $g(x) \leq 0$ (包含盒约束)。
- **Constrained Rosenbrock 4D**: 1 个线性等式约束与 3 个不等式约束。

每个算例分别在两种线搜索策略下运行：golden (黄金分割线搜索) 与 armijo_wolfe (Armijo-Wolfe 强 Wolfe 非精确线搜索)。所有实验均从给定的 4 个初值出发，并记录外层迭代信息 (ALM 外层为罚参数/乘子更新层；SQP 外层为 QP 子问题迭代层)。

收敛判据 (KKT 误差): 为了统一比较，本文以 KKT 误差作为收敛准则。记等式约束为 $h(x) = 0$ ，不等式约束为 $g(x) \leq 0$ ，其雅可比分别为 $J_h(x), J_g(x)$ 。对给定点 x ，用最小二乘估计乘子 (λ, μ)

(并对不等式乘子施加 $\mu \geq 0$), 定义

$$r_{\text{stat}}(x) = \|\nabla f(x) + J_h(x)^\top \lambda + J_g(x)^\top \mu\|_2, \quad (10)$$

$$r_{\text{fea}}(x) = \left\| \begin{bmatrix} h(x) \\ \max(g(x), 0) \end{bmatrix} \right\|_2, \quad r_{\text{comp}}(x) = \|\min(\mu, -g(x))\|_2, \quad (11)$$

$$r_{\text{KKT}}(x) = \max\{r_{\text{stat}}(x), r_{\text{fea}}(x), r_{\text{comp}}(x)\}. \quad (12)$$

当 $r_{\text{KKT}}(x^k) \leq 10^{-6}$ 时认为收敛。除此之外, 为了直观展示数值收敛过程, 我们还绘制

$$\|\nabla f(x^k)\|_2 + \|c(x^k)\|_2, \quad \|c(x)\|_2 := \left\| \begin{bmatrix} h(x) \\ \max(g(x), 0) \end{bmatrix} \right\|_2 \quad (13)$$

关于外层迭代次数 k 的曲线 (纵轴采用对数尺度, 与参考代码一致)。

3.2 Constrained Modified Rosenbrock 2D

3.2.1 初值点选取与可行域结构

本算例的 4 个初始点取为

$$x_1^{(0)} = (0.0, 0.0), \quad x_2^{(0)} = (0.5, 0.5), \quad x_3^{(0)} = (-1.2, 0.56), \quad x_4^{(0)} = (1.6, 2.0).$$

为了便于观察可行域与极小点位置, 图 1 给出了参考背景图 (仅含约束曲线、可行域与极小点图例, 不含迭代路径)。从该图可以看出: 可行集主要由等式约束曲线与盒约束共同确定; 同时沿可行曲线存在两个显著的局部驻点, 其中全局极小点在 $(1, 1)$ 附近 ($f \approx 0$), 另一个局部极小点在 $(-1, 1)$ 附近 ($f \approx 4$)。本文在表格中对“全局/局部”的判定采用与参考图一致的两点: 若收敛点落在 $(1, 1)$ 的邻域内则记为全局, 否则若落在 $(-1, 1)$ 的邻域内则记为局部。

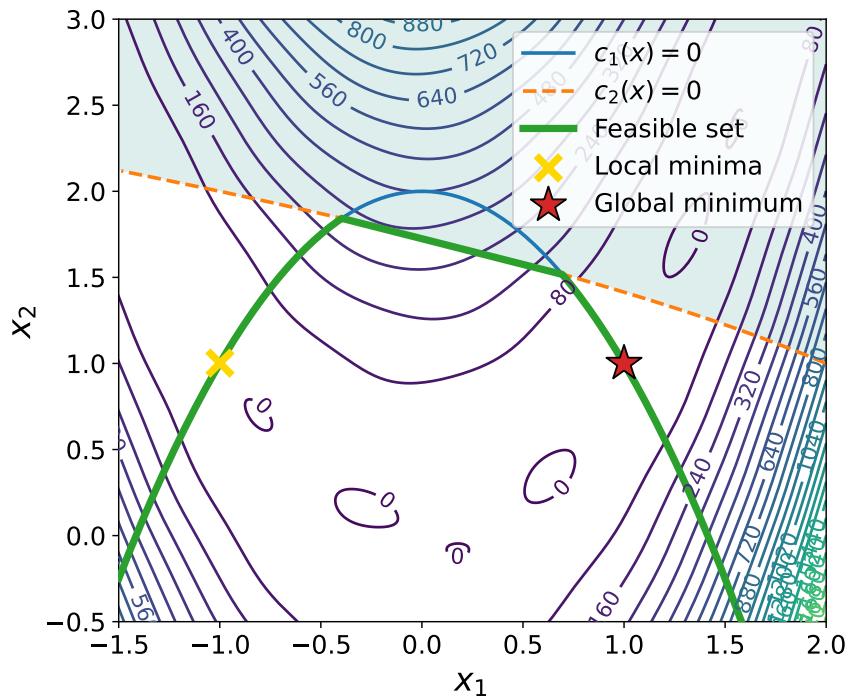


图 1: 2D 算例的可行域背景图 (不含迭代路径)。

3.2.2 数值结果汇总（含两种线搜索）

表 1 汇总了两种线搜索下 ALM 与 SQP 的收敛迭代次数（外层迭代）、最终目标函数值与 KKT 误差，并给出全局/局部极小点的判断。可以看到：16 组实验全部收敛，且 KKT 误差均达到 10^{-6} 量级。

表 1: 2D 算例结果汇总（包含 $\|f(x^*)\|$ 与 $\|c(x^*)\|$ ；类型：全局/局部）。

(a) golden (黄金分割线搜索)

$x^{(0)}$	ALM					SQP				
	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型
(0., 0.)	8	3.998	9.13×10^{-7}	9.13×10^{-7}	局部	13	3.998	4.44×10^{-16}	6.12×10^{-7}	局部
(0.5, 0.5)	1	4.69×10^{-18}	2.27×10^{-10}	5.74×10^{-8}	全局	5	3.87×10^{-17}	2.58×10^{-10}	1.89×10^{-7}	全局
(-1.2, 0.56)	1	1.74×10^{-21}	6.85×10^{-12}	7.51×10^{-10}	全局	11	3.998	0	4.88×10^{-7}	局部
(1.6, 2.)	8	3.998	9.13×10^{-7}	9.13×10^{-7}	局部	4	8.8×10^{-16}	1.39×10^{-9}	8.96×10^{-7}	全局

(b) armijo_wolfe (Armijo-Wolfe 非精确线搜索)

$x^{(0)}$	ALM					SQP				
	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型
(0., 0.)	8	3.998	9.13×10^{-7}	9.13×10^{-7}	局部	13	3.998	0	9.11×10^{-7}	局部
(0.5, 0.5)	13	3.42×10^{-12}	3.72×10^{-7}	3.72×10^{-7}	全局	5	2.31×10^{-17}	2.43×10^{-10}	1.44×10^{-7}	全局
(-1.2, 0.56)	8	3.998	5.72×10^{-8}	5.69×10^{-7}	局部	11	3.998	4.44×10^{-16}	2.76×10^{-7}	局部
(1.6, 2.)	14	1.32×10^{-11}	7.29×10^{-7}	7.29×10^{-7}	全局	4	3.72×10^{-16}	9.69×10^{-10}	5.8×10^{-7}	全局

3.2.3 优化路径对比

对每个（初值点 \times 线搜索）绘制一张同图叠加的优化路径图：一张图上同时包含 ALM 与 SQP 的迭代轨迹（图例仅含 ALM 与 SQP）。图 2-3 给出了两种线搜索下的 4 组路径对比。

从路径上可以观察到一个共性现象：**SQP 的迭代点更快贴近并沿着可行曲线前进**。这是因为 SQP 每步通过线性化约束并解二次规划子问题来产生方向，使得等式约束的线性可行性在局部得到较好保持；相对地，ALM 通过罚参数与乘子逐步强化可行性，其迭代点可能在早期出现更明显的“先修正可行性、再降低目标”的两阶段行为。

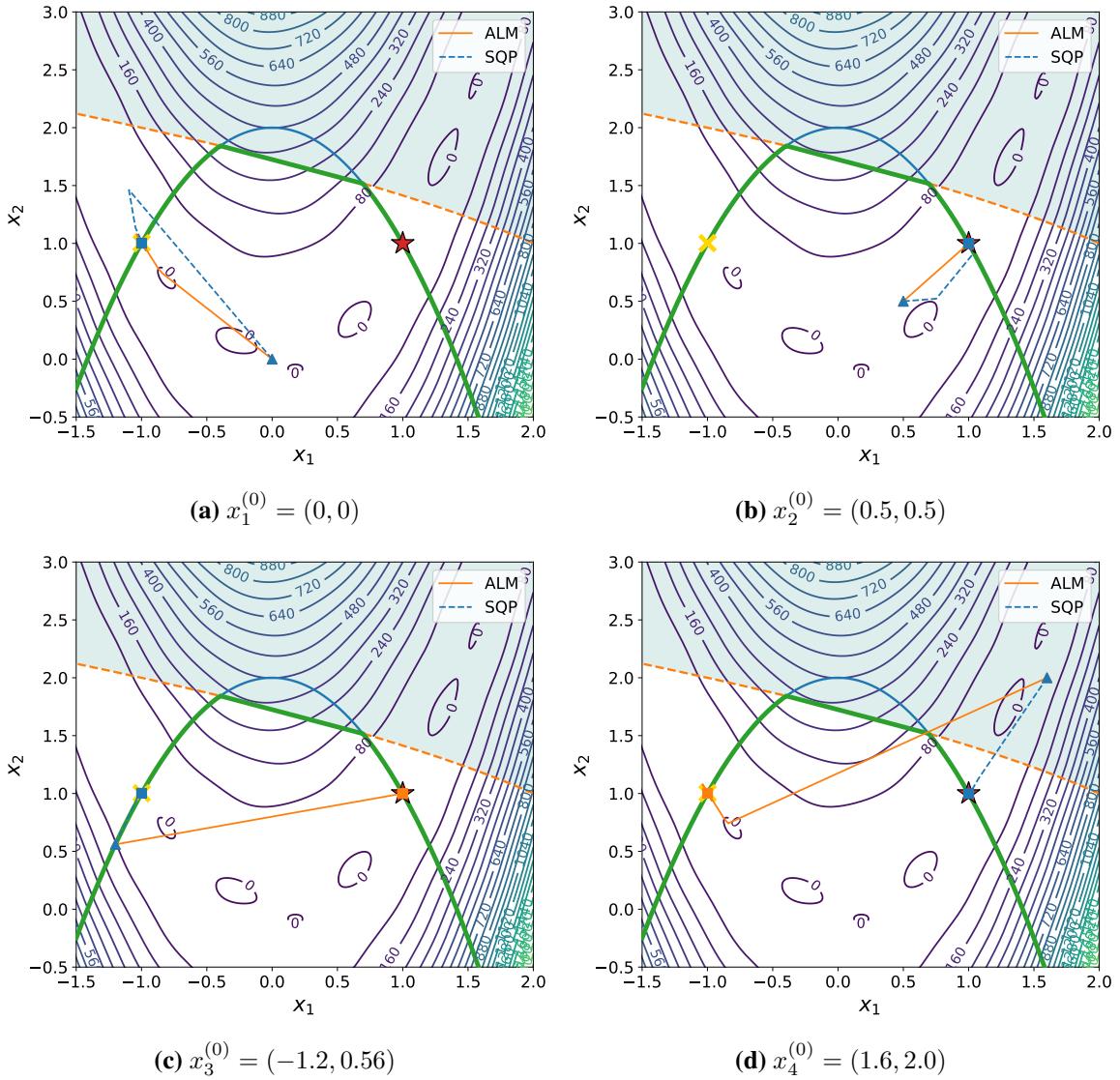


图 2: 2D: golden 下 ALM 与 SQP 的优化路径对比 (每幅图图例仅含 ALM/SQP)。

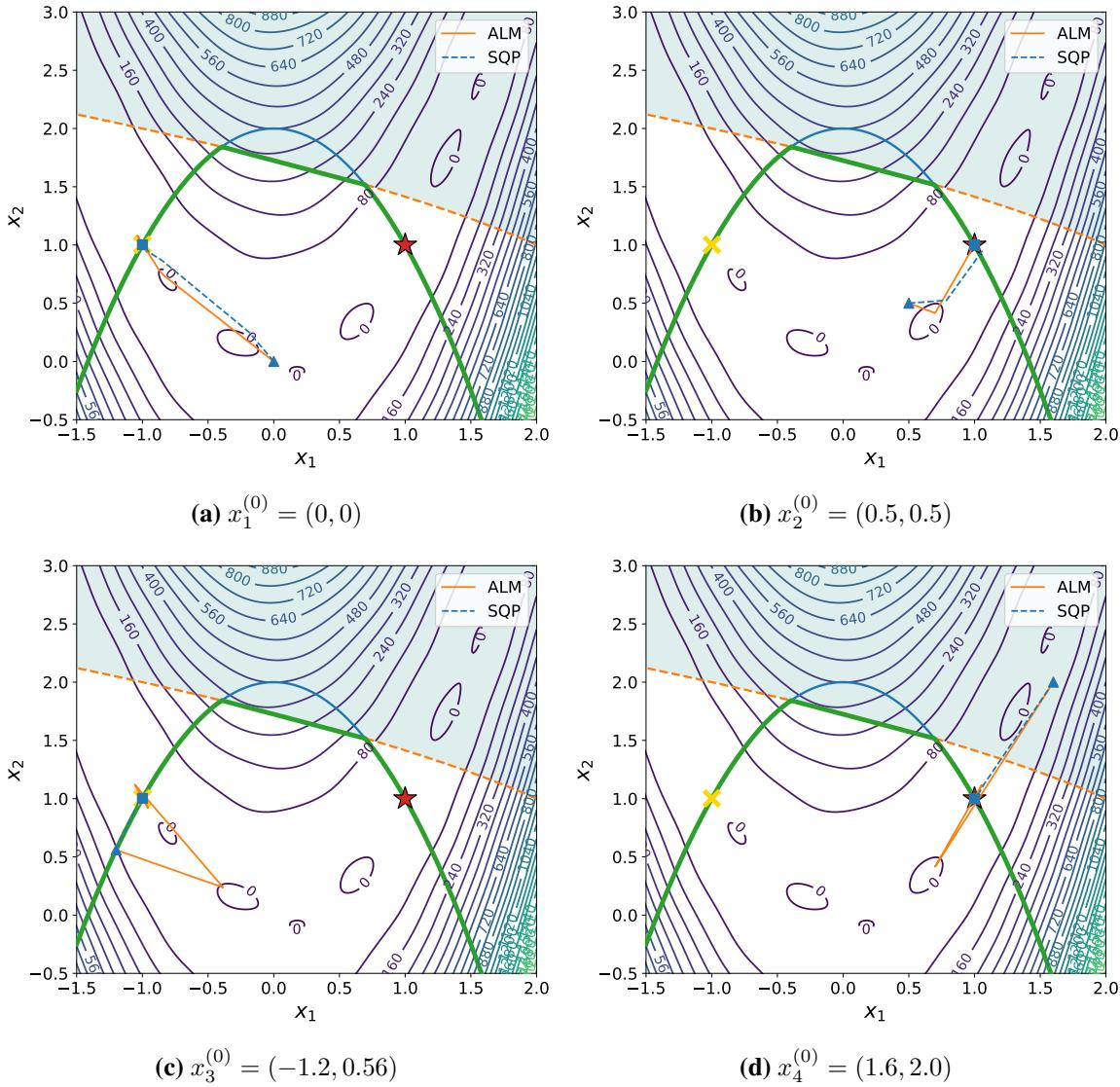


图 3: 2D: armijo_wolfe 下 ALM 与 SQP 的优化路径对比 (每幅图图例仅含 ALM/SQP)。

3.2.4 收敛曲线与现象解释

图 4-5 给出了 $\|\nabla f\| + \|c\|$ 随外层迭代的变化 (纵轴对数尺度)。整体而言两种算法均能在有限步内将误差压到 10^{-6} 附近，但收敛形态存在差异：

- **SQP:** 在进入局部收敛区域后，误差通常下降更快（体现出 SQP 的典型快速局部收敛特征），但其收敛到“全局/局部”更多取决于初值在可行曲线上的吸引域。
- **ALM:** 常见“先可行后最优”的两阶段过程；在罚参数更新或乘子校正时，误差曲线可能出现阶段性拐点。与 SQP 相比，ALM 对可行性修正更显式，但需要合适的罚参数更新策略与内层求解精度配合。

此外，由于目标函数沿可行曲线具有多个局部极小点，不同算法/线搜索在相同初值下可能收敛到不同极小点（见表 1 中 $x_3^{(0)}, x_4^{(0)}$ 的对比）。这属于非凸问题的正常现象：线搜索影响每步步长大小，从而改变迭代轨迹是否跨越“势垒”进入另一吸引域；而 ALM 与 SQP 的方向生成机制不同（ALM 以增广拉格朗日为目标、SQP 以二次规划近似为目标），也会进一步放大这种

差异。

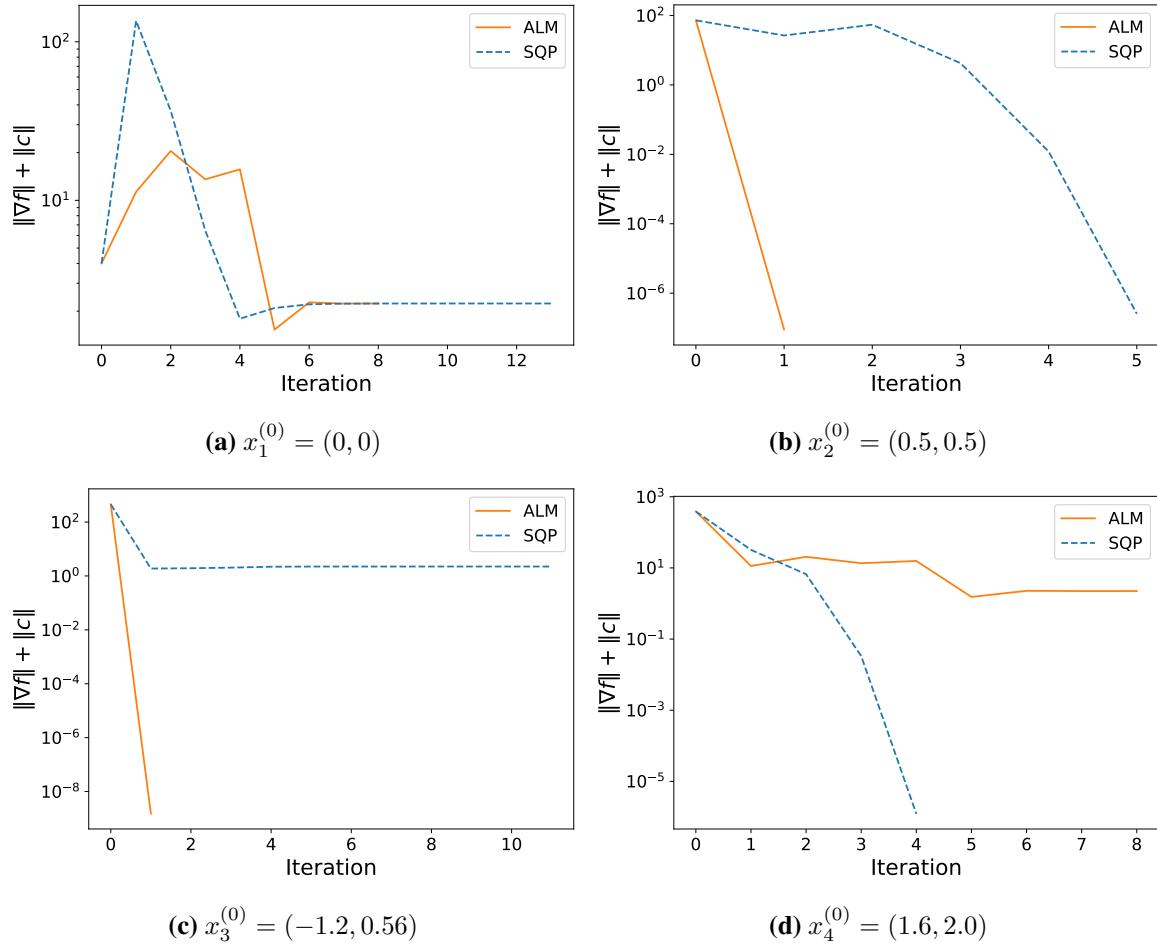


图 4: 2D: golden 下 $\|\nabla f\| + \|c\|$ 的收敛曲线 (纵轴对数尺度)。

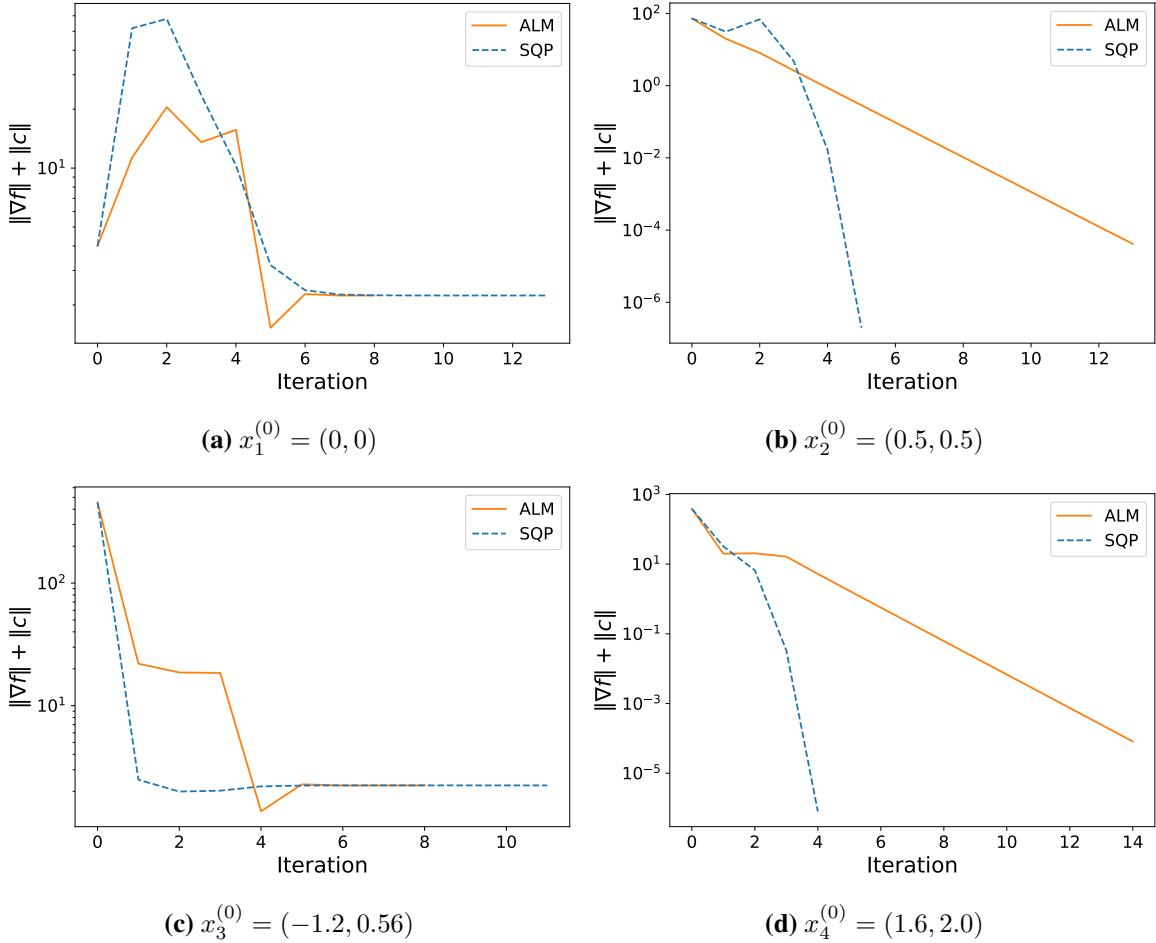


图 5: 2D: `armijo_wolfe` 下 $\|\nabla f\| + \|c\|$ 的收敛曲线 (纵轴对数尺度)。

3.3 Constrained Rosenbrock 4D

3.3.1 初值点选取

本算例的 4 个初始点取为

$$x_1^{(0)} = (0.5, 0.0, 0.3, 1.2), \quad x_2^{(0)} = (0.0, 0.0, 0.0, 0.0), \quad x_3^{(0)} = (0.0, 2.0, 0.0, 0.0), \quad x_4^{(0)} = (3.0, -1.0, 0.0, 0.0).$$

3.3.2 数值结果与收敛行为

表 2 汇总了 4D 算例在两种线搜索下的结果。可以看到：

- **16 组实验全部收敛**，并且最终都收敛到同一个可行 KKT 点（表中类型均判为全局）。
- 与 2D 相比，本算例在最优点处不等式约束均为严格可行（不活跃），因此两种算法在收敛后期更接近“等式约束问题”的行为，表现为稳定且一致的收敛终点。

图 6–7 给出了 $\|\nabla f\| + \|c\|$ 曲线：两种算法均能在较少的外层迭代内将误差下降到阈值附近，其中 ALM 的外层迭代次数整体略少；但需要强调的是，ALM 的每次外层迭代包含一个内层子问题求解，其单次迭代成本通常高于 SQP 的“一次 QP + 线搜索”步（本文主要比较外层收敛行为）。

表 2: 4D 算例结果汇总 (包含 $\|f(x^*)\|$ 与 $\|c(x^*)\|$; 类型: 全局/局部)。

(a) **golden** (黄金分割线搜索)

$x^{(0)}$	ALM				SQP					
	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型
(0.5, 0., 0.3, 1.2)	4	0.5281	4.41×10^{-9}	6×10^{-7}	全局	6	0.5281	4.44×10^{-16}	4.63×10^{-8}	全局
(0., 0., 0., 0.)	5	0.5281	3.54×10^{-7}	7.59×10^{-7}	全局	6	0.5281	8.88×10^{-16}	5.5×10^{-8}	全局
(0., 2., 0., 0.)	5	0.5281	3.66×10^{-7}	3.66×10^{-7}	全局	7	0.5281	2.22×10^{-16}	1.14×10^{-7}	全局
(3., -1., 0., 0.)	5	0.5281	3.6×10^{-7}	6.98×10^{-7}	全局	8	0.5281	2.22×10^{-16}	8.36×10^{-8}	全局

(b) **armijo_wolfe** (Armijo–Wolfe 非精确线搜索)

$x^{(0)}$	ALM				SQP					
	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型	iter	$\ f(x^*)\ $	$\ c(x^*)\ $	KKT 误差	类型
(0.5, 0., 0.3, 1.2)	4	0.5281	7.5×10^{-9}	8.94×10^{-7}	全局	6	0.5281	0	8.11×10^{-7}	全局
(0., 0., 0., 0.)	6	0.5281	1.63×10^{-8}	9.1×10^{-7}	全局	7	0.5281	0	1.6×10^{-13}	全局
(0., 2., 0., 0.)	6	0.5281	1.63×10^{-8}	9.1×10^{-7}	全局	7	0.5281	2.22×10^{-16}	5.13×10^{-11}	全局
(3., -1., 0., 0.)	7	0.5281	2.74×10^{-7}	4.72×10^{-7}	全局	8	0.5281	4.44×10^{-16}	1.39×10^{-11}	全局

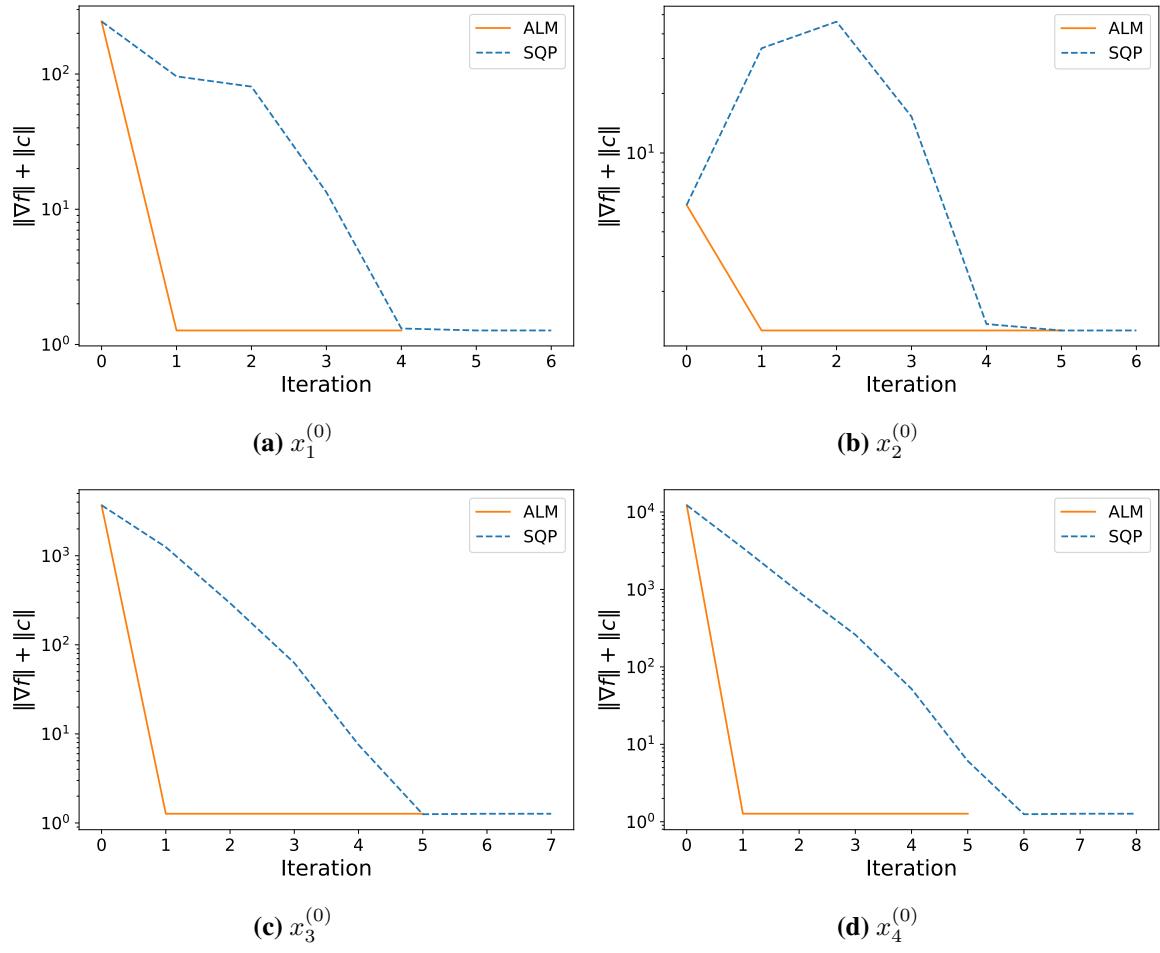


图 6: 4D: golden 下 $\|\nabla f\| + \|c\|$ 的收敛曲线 (纵轴对数尺度)。

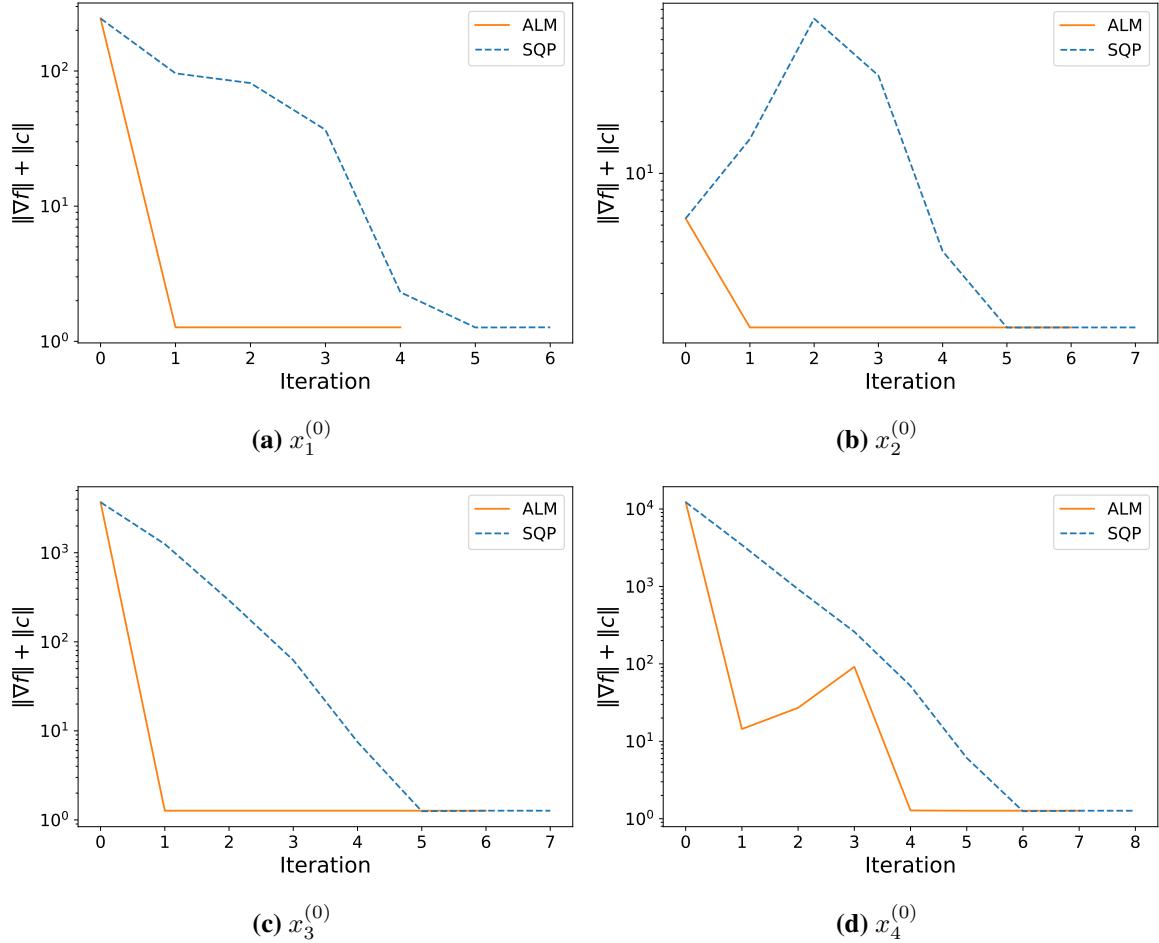


图 7: 4D: armijo_wolfe 下 $\|\nabla f\| + \|c\|$ 的收敛曲线 (纵轴对数尺度)。

3.4 小结：ALM 与 SQP 的优劣对比

综合两个算例的实验现象，可以得到以下结论（重点放在算法本身）：

- **收敛稳定性与鲁棒性:** 两种算法在本实验的所有初值与两种线搜索组合下均能收敛到满足 KKT 误差阈值的解, 说明在合适的正定化/正则化与线搜索配合下, 二者都具有较好的数值鲁棒性。
 - **局部收敛速度:** SQP 往往在接近解时表现出更快的误差下降 (对数尺度下下降更陡), 这是由其“二次模型 + 线性化约束”的牛顿型结构决定的; 当 Hessian/拉格朗日 Hessian 近似质量较好且活跃集识别稳定时, SQP 的优势更明显。
 - **对非凸多极小点的行为:** 在 2D 非凸算例中, ALM 与 SQP 都可能收敛到不同的局部极小点; 这种差异主要来源于两者的方向生成机制不同, 从而导致吸引域不同。实际应用中, 若更关注找到全局解, 通常需要配合多起点策略或全局化技巧。
 - **每步计算代价与参数敏感性:** ALM 的外层迭代次数在部分设置下较少, 但每次外层迭代包含一个内层子问题求解, 且对罚参数更新策略较敏感; SQP 每步需要解一个 KKT 系统/二次规划子问题, 单步代价也较高, 但参数更少、收敛行为更“牛顿化”。

因此，在中小规模、需要高精度 KKT 解且局部模型可靠的场景下，**SQP** 通常更具优势；而在起点不可行、约束结构复杂或希望通过罚函数逐步逼近可行集的场景下，**ALM** 往往更稳健且实现

灵活。

4 思考题

1. 增广拉格朗日法通过罚参数将约束融入目标函数，为何比单纯罚函数法的收敛精度更高？罚参数的选择对算法稳定性有何影响？

为何 ALM 比单纯罚函数法精度更高。 单纯罚函数法通常构造

$$\min_x f(x) + \frac{\rho}{2} \|h(x)\|^2 \quad (\text{或 } f(x) + \rho \|\max(g(x), 0)\|^2), \quad (14)$$

并依赖让 $\rho \rightarrow \infty$ 来强迫可行性。其主要困难在于：当 ρ 足够大以保证较高可行精度时，目标的 Hessian（或近似 Hessian）会被罚项主导，导致子问题 **严重病态**，线搜索/方向求解变得困难，从而在数值上很难同时做到“可行性很高 + 最优性很高”。

增广拉格朗日法（ALM）通过在罚项外加入拉格朗日乘子项并迭代更新乘子，以等式约束为例，其增广拉格朗日函数为

$$\mathcal{L}_\rho(x, \lambda) = f(x) + \lambda^\top h(x) + \frac{\rho}{2} \|h(x)\|^2, \quad (15)$$

不等式约束则可用带投影的形式等价地写入（实验中采用 $\mu \leftarrow \max(0, \mu + \rho g(x))$ 的更新方式）。ALM 的关键点是：**乘子更新抵消了约束误差的一阶影响**。在 x 附近若 $h(x) \neq 0$ ，罚函数法只能依靠巨大 ρ 产生足够的修正梯度；而 ALM 中 λ 的迭代会逐步逼近真实乘子 λ^* ，使得即便 ρ 不必无限大， $\nabla_x \mathcal{L}_\rho(x, \lambda)$ 仍能更准确地逼近 KKT 的站立性条件

$$\nabla f(x^*) + J_h(x^*)^\top \lambda^* + J_g(x^*)^\top \mu^* = 0. \quad (16)$$

因此 ALM 的典型性质是：**在保持中等罚参数的同时，也能获得更高的 KKT 精度**；反之罚函数法要想逼近 KKT（尤其是乘子相关的站立性），往往需要更大的罚参数从而导致病态。

结合数值实验的解释。 在本次数值实验中，两种算例均能达到 10^{-6} 量级的 KKT 误差，但 ALM 的外层收敛常呈现“先可行后最优”的阶段性特征：早期通过罚项显著降低 $\|c(x)\|$ ，随后乘子逐步稳定，最终站立性残差下降到阈值。尤其在 2D 非凸算例中，ALM 在不同初值下既能收敛到全局点也能收敛到局部点，但无论落在哪个吸引域，最终可行性残差与 KKT 残差都能稳定压到阈值附近，这体现了“乘子 + 罚项”的组合在数值上较罚函数更容易取得高精度 KKT 可行解。

罚参数对稳定性的影响。 罚参数（或记作 ρ ）对 ALM 的稳定性有“过小/过大”两类风险：

- **ρ 过小：** 罚项不足以推动可行性改进，导致外层迭代中 $\|c(x)\|$ 下降缓慢，乘子更新可能“追不上”可行性的改进，表现为收敛变慢甚至在非凸问题中出现震荡。
- **ρ 过大：** 增广拉格朗日子问题变得更接近罚函数法，数值上出现病态：梯度/Hessian 被罚项主导，步长更容易被迫缩小，内层求解（例如拟牛顿步）可能更不稳定，导致外层迭代效率下降。

因此实际实现中常采用**自适应更新**：当 $\|c(x)\|$ 改善不明显时增大 ρ ，否则保持或缓慢调整，从而在“可行性推动力”和“数值条件良好”之间折中。实验代码中也采用了类似策略（当约束违背下降不足时才增大罚参数），从而保证绝大多数实验都能稳定收敛。

2. 序列二次规划 (SQP) 在处理强非线性约束时表现更优，其核心原因是“二次规划子问题逼近约束优化局部特性”，请结合 2 维测试函数的约束特点解释这一优势？

SQP 的核心机制：局部 KKT 结构的一步逼近。 SQP 在当前迭代点 x^k 处，构造二次规划子问题

$$\min_d \frac{1}{2} d^\top B_k d + \nabla f(x^k)^\top d \quad \text{s.t.} \quad h(x^k) + J_h(x^k)d = 0, \quad g(x^k) + J_g(x^k)d \leq 0, \quad (17)$$

其中 B_k 近似拉格朗日函数 Hessian。该 QP 的 KKT 系统在形式上正是原问题 KKT 系统在 (x^k, λ^k, μ^k) 的一阶/二阶近似，因此当 x^k 接近解时，解出的方向 d 往往能同时兼顾：

- 通过线性化约束直接控制“朝可行集的方向”；
- 通过二次模型体现曲率信息，从而获得更快的局部收敛。

这就是 SQP 在强非线性约束问题上常表现优良的根本原因：**它不是把约束简单地“罚进目标”，而是把约束几何结构显式地写入子问题。**

结合 2D 测试算例的约束特点解释优势。 2D 算例的等式约束为 $h(x) = x_1^2 + x_2 - 2 = 0$ ，其可行集是一条弯曲的抛物线（再与盒约束截断形成可行弧段）；不等式约束包含 $g_1(x) = x_1 + x_2^2 - 3 \leq 0$ ，边界同样是强非线性曲线。因此在该算例上，“可行集的几何曲率”很强：若仅用罚函数法或仅靠增广项，迭代点可能在早期偏离可行曲线较远，随后需要通过较强罚项慢慢拉回。

SQP 的优势在于：在每一步 QP 子问题中，

$$h(x^k) + J_h(x^k)d = 0 \quad (18)$$

直接强制搜索方向 d 沿着等式约束的切空间/线性化可行集前进，这相当于在局部把曲线可行集“拉直”成一条线（或一个仿射子空间），从而更容易做出“沿可行曲线降低目标”的一步。这在 2D 路径图中体现为：SQP 的迭代点通常更快贴近并沿着可行曲线前进，而不是在可行集外大幅兜圈（这一点在多个初值与两种线搜索下都较一致）。

进一步解释：非凸 + 多吸引域下的表现。 2D 目标函数本身带有非凸的正弦扰动，沿可行曲线存在多个局部极小点（全局点 $(1, 1)$ 附近以及另一局部点 $(-1, 1)$ 附近）。SQP 的局部模型更忠实于“约束几何 + 目标局部曲率”，因此一旦进入某个吸引域，其后期收敛往往更快；但同时，吸引域的选择仍会受到初值与步长策略影响，因此在某些初值下不同线搜索会导致 SQP 最终落在不同极小点（全局/局部），这属于非凸问题的固有现象，而非算法失效。

小结。 在强非线性约束（如 2D 中弯曲等式约束与非线性不等式边界）下，SQP 通过“线性化约束 + 二次模型”在每步显式刻画可行集局部几何，因而更容易产生高质量的可行下降方向，表现为更快贴近可行集与更快的局部收敛；这正是其相对罚函数/增广罚项方法的结构性优势。

5 总结与反思

5.1 总结

围绕二维强非线性约束算例与四维混合约束算例，本文实现并比较了 ALM 与 SQP 在 golden 与 Armijo–Wolfe 两种线搜索下的收敛行为。综合实验现象，可归纳如下结论：

- **收敛性与鲁棒性：**在本文设置的初值与线搜索组合下，两种方法均能将 KKT 误差压到 10^{-6} 量级，说明在合理的正则化/正定化处理及线搜索全局化配合下，ALM 与 SQP 均具有较好的数值稳定性。
- **局部收敛速度差异：**SQP 由于“二次模型 + 约束线性化”的牛顿型结构，在进入局部收敛区域后往往下降更快；ALM 则常出现“先降低约束违背、再降低目标函数”的两阶段特征，误差曲线可能在罚参数更新或乘子校正处出现阶段性拐点。
- **非凸多极小点影响：**二维算例沿可行曲线存在多个局部极小点，因此不同初始点与步长策略会改变迭代轨迹是否跨越势垒进入另一吸引域，导致算法可能收敛到不同极小点；这属于非凸约束优化的固有现象，而非算法失效。
- **代价与参数敏感性：**ALM 的外层迭代次数在部分设置下较少，但每次外层迭代包含内层无约束子问题求解，且对罚参数更新策略较敏感；SQP 每步需要求解 QP 子问题并配合优值函数线搜索，单步代价也较高，但参数相对更少、收敛行为更“牛顿化”。

总体而言，在中小规模且希望获得高精度 KKT 解、局部模型可靠的场景下，SQP 通常更具优势；而在起点不可行、约束结构复杂或希望通过逐步强化可行性逼近可行集的场景下，ALM 往往更稳健且实现更灵活。

5.2 反思

尽管实验较系统地比较了两类典型约束优化算法，但仍存在一些局限与可改进之处：

- **比较维度仍偏“外层迭代”：**本文主要以外层迭代次数与误差曲线比较收敛形态，但 ALM 的外层包含内层求解，SQP 的单步包含 QP 求解，二者单步代价不可直接等同。后续可进一步统计 CPU 时间、函数/梯度调用次数等更公平的成本指标。
- **参数与策略的系统性不足：**ALM 的罚参数更新、内层求解精度与 BFGS 重置策略会显著影响效率；SQP 的活跃集判别阈值、Hessian 正定化与 merit function 参数也会影响全局化效果。后续可做更系统的敏感性实验，并尝试自适应策略（如动态罚参数或滤子方法）。
- **非凸问题的全局性不足：**二维算例的多吸引域现象提示：仅靠单一起点与常规线搜索难以保证获得全局解。后续可引入多起点策略、随机扰动重启、信赖域/多段线搜索或全局化的 SQP（如 trust-region SQP）以提高找到更优解的概率。

通过本次实现与对比，我对“约束处理方式（罚项/乘子/线性化）如何影响数值稳定性与收敛形态”有了更直观的认识：SQP 更像对局部 KKT 结构的直接逼近，而 ALM 更像通过“乘子迭代 + 适度罚项”逐步逼近可行 KKT 点。两者各有优势，算法选型与工程实现应结合问题规模、约束结构、初值可行性以及对精度与成本的侧重综合权衡。

参考文献

- 刘浩洋 et al. (2020). 最优化：建模、算法与理论. 北京: 高等教育出版社. ISBN: 9787040550351.
- 高立 (2014). 数值最优化方法. 北京大学数学教学系列丛书. 北京: 北京大学出版社. ISBN: 9787301246450.