

# 优化理论与方法实验报告

## 实验一：无约束优化算法性能对比实验

周旻

2023302011158

2023 级信息与计算科学

日期：2025 年 12 月 17 日

### 摘 要

本文围绕无约束优化问题，选取 `modified_rosenbrock_2d` 与 `rosenbrock_4d` 两个经典的测试函数，系统比较了最速下降法、Newton 法、BFGS 拟 Newton 法、共轭梯度法 (FR/PRP) 以及信赖域方法在不同初始点、不同线搜索策略下的数值表现。实验中分别采用 0.618 方法实现 (近似) 精确线搜索，并使用 Armijo-Wolfe 准则实现非精确线搜索，记录迭代次数、函数值、梯度范数等指标，并通过优化路径图和收敛曲线进行可视化分析。结果表明：在两个算例上，Newton 法、BFGS 法和信赖域方法在收敛速度与数值稳定性方面明显优于最速下降法和共轭梯度法，但所有算法都缺乏保证收敛到全局极小点的能力，对初始点和目标函数非凸性的敏感性较高。尤其是在 `modified_rosenbrock_2d` 上，共轭梯度法在没有重启策略且配合 0.618 方法的近似精确线搜索时会因“共轭方向”约束而失稳甚至发散，而在 `rosenbrock_4d` 上，高维性进一步放大了不同算法在效率和稳定性上的差异。本文通过数值对比和理论分析，直观展示了各类优化算法的优势与局限，为后续算法改进与实际应用中的算法选型提供了参考。

**关键词：**Rosenbrock 函数；无约束优化；线搜索；BFGS 拟 Newton 法；信赖域方法

# 目录

<b>1 引入</b>	<b>3</b>
1.1 实验内容	3
1.2 实验目的	3
<b>2 实验的理论基础</b>	<b>3</b>
2.1 算例	3
2.1.1 modified_rosenbrock_2d	4
2.1.2 rosenbrock_4d	5
2.2 线搜索算法	5
2.2.1 0.618 方法	6
2.2.2 Armijo–Wolfe 非精确线搜索	6
2.3 优化算法	7
2.3.1 最速下降法	7
2.3.2 Newton 法	7
2.3.3 BFGS 拟 Newton 法	7
2.3.4 共轭梯度法	8
2.3.5 信赖域方法	9
<b>3 数值实验</b>	<b>9</b>
3.1 实验结果和分析	10
3.2 代码使用说明	16
<b>4 思考题</b>	<b>17</b>
4.1 为什么 Newton 法在二次函数上能“一步收敛”，而最速下降法需要多次迭代？	17
4.2 BFGS 拟 Newton 法无需直接计算 Hessian 矩阵，却能接近 Newton 法的收敛速度，其核心思想是什么？	17
4.3 共轭梯度法在非二次函数上的收敛性能为何不如 BFGS？这与“共轭方向”的性质有何关系？	18
<b>5 总结与反思</b>	<b>18</b>
<b>参考文献</b>	<b>19</b>
<b>附录</b>	<b>20</b>
<b>A 精确线搜索的图表数据</b>	<b>20</b>

# 1 引入

## 1.1 实验内容

本实验围绕典型无约束优化算法在标准测试函数上的数值实现与性能比较展开。

### 1. 优化算法实现：在 Matlab 中分别实现

- (a) 最速下降法
- (b) Newton 法
- (c) BFGS 拟 Newton 法
- (d) 共轭梯度法 (FR/PRP 公式)
- (e) 信赖域算法

并配合 0.618 方法的近似精确线搜索与 Armijo–Wolfe 非精确线搜索确定步长。

### 2. 对比实验：按照实验说明统一选取初始点、收敛判据 $\|\nabla f(x_k)\| \leq 10^{-6}$ 及最大迭代步数 $k_{\max} = 1000$ ，在两类测试函数 (modified\_rosenbrock\_2d 和 rosenbrock\_4d) 上运行各种优化算法，记录迭代次数、函数值、梯度范数等指标，并通过表格、收敛曲线以及优化路径图进行比较分析。

### 3. 思考题与理论分析：在上述数值结果基础上，结合课上理论知识，对下列问题进行讨论与回答，同时引用实验中的具体现象进行解释，

- (a) 为什么 Newton 法在二次函数上能够“一步收敛”，而最速下降法需要多次迭代？
- (b) BFGS 拟 Newton 法在不显式计算 Hessian 矩阵的情况下，为什么能达到接近 Newton 法的收敛速度？其本质思想是什么？
- (c) 共轭梯度法在非二次函数上的收敛性能为何仍不如 BFGS？这与“共轭方向”的性质有何关系？

## 1.2 实验目的

- 1. 掌握无约束优化中的典型数值算法：最速下降法、Newton 法、BFGS 拟 Newton 法、共轭梯度法以及信赖域方法，理解各算法的基本思想和实现要点。
- 2. 通过编程实现上述算法，在测试函数 (modified\_rosenbrock\_2d 和 rosenbrock\_4d) 上，对比不同算法在收敛速度、迭代次数、收敛精度以及数值稳定性方面的差异。
- 3. 理解并分析不同算法在原理上的差别，如梯度信息的利用方式、对 Hessian 矩阵及其近似的依赖程度、搜索方向更新策略，以及精确线搜索与 Armijo–Wolfe 非精确线搜索在步长选取上的不同，阐明这些因素对优化效果和算法性能的影响。

# 2 实验的理论基础

## 2.1 算例

本次实验采用两个算例：modified\_rosenbrock\_2d 以及 rosenbrock\_4d。选择这两个算例的原因包括：

1. Rosenbrock 函数是无约束优化的**经典非凸算例**；
2. **算例的结构简单**，可以写出显式的梯度和 Hessian 矩阵，适用于各种一阶和二阶算法；同时算例的局部极小点和全局极小点已知，便于分析结果；
3. **算例的难度适中**，从图像看，是一条弯曲且非常窄的山谷，各向异性很强，谷底方向上函数变化平缓，垂直于山谷的方向又非常陡峭，但至少明确的**全局极小点**可以求解得到。

### 2.1.1 modified\_rozenbrock\_2d

modified\_rozenbrock\_2d 的函数表达式为

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

其梯度为

$$g(x_1, x_2) = \begin{bmatrix} -2(1 - x_1) - 400x_1(x_2 - x_1^2) + 10\pi \cos(2\pi x_1) \sin(2\pi x_2) \\ 200(x_2 - x_1^2) + 10\pi \sin(2\pi x_1) \cos(2\pi x_2) \end{bmatrix},$$

以及 Hessian 矩阵为

$$G(x_1, x_2) = \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 - 20\pi^2 \sin(2\pi x_1) \sin(2\pi x_2) & -400x_1 + 20\pi^2 \cos(2\pi x_1) \cos(2\pi x_2) \\ -400x_1 + 20\pi^2 \cos(2\pi x_1) \cos(2\pi x_2) & 200 - 20\pi^2 \sin(2\pi x_1) \sin(2\pi x_2) \end{bmatrix}.$$

modified\_rozenbrock\_2d 是在标准的二维 Rosenbrock 函数上加上正弦项，使得其在山谷区域出现周期性的起伏，形状如图 1 所示。对于二维 Rosenbrock 函数，极小点是唯一的；而 modified\_rozenbrock\_2d 在山谷区域有一系列的局部极小点。这个算例的好处在于，

1. 能够测试不同优化算法是否容易被局部极小点“吸引”，是否具有跳出局部极小点寻找全局极小点的能力；
2. 可以在平面内画出优化路径图，方便展示和分析不同优化算法的性质。

图 1 展示了 modified\_rozenbrock\_2d 的局部极小点和全局极小点的位置。其中全局极小点的坐标和目标函数值大致为

$$x^* \approx [1.30577, 1.72662]^\top, \quad f(x^*) \approx -4.50549.$$

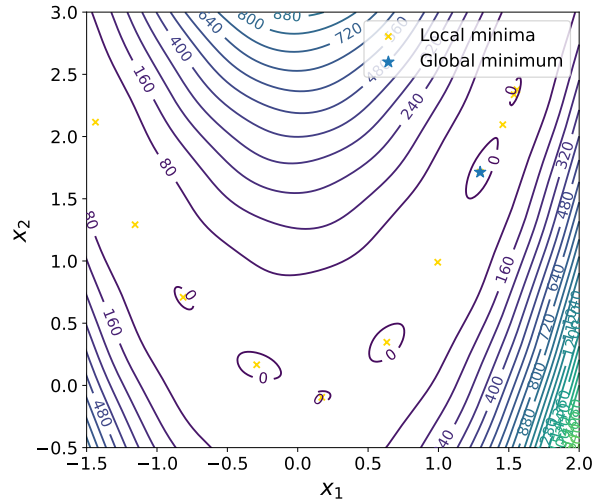


图 1: modified\_rozenbrock\_2d 的局部极小点和全局极小点

### 2.1.2 rosenbrock\_4d

rosenbrock\_4d 的函数表达式为

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

其梯度为

$$g(x_1, x_2, x_3, x_4) = \begin{bmatrix} 2(x_1 - 1) - 400x_1(x_2 - x_1^2) \\ 2(x_2 - 1) + 200(x_2 - x_1^2) - 400x_2(x_3 - x_2^2) \\ 2(x_3 - 1) + 200(x_3 - x_2^2) - 400x_3(x_4 - x_3^2) \\ 200(x_4 - x_3^2) \end{bmatrix},$$

以及 Hessian 矩阵为

$$G(x_1, x_2, x_3, x_4) = \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 & 0 & 0 \\ -400x_1 & 1200x_2^2 - 400x_3 + 202 & -400x_2 & 0 \\ 0 & -400x_2 & 1200x_3^2 - 400x_4 + 202 & -400x_3 \\ 0 & 0 & -400x_3 & 200 \end{bmatrix}.$$

rosenbrock\_4d 同样具有类似二维 Rosenbrock 函数的形状，即一条弯曲且非常窄的山谷，但将函数的维数拓展至四维，能够测试不同优化算法在较高维目标函数上的性能表现。显然，rosenbrock\_4d 的全局极小点的坐标和目标函数值为

$$x^* = [1, 1, 1, 1]^\top, \quad f(x^*) = 0.$$

事实上，根据 Kok and Sandrock (2009) 的工作，一阶最优性条件  $g(x_1, x_2, x_3, x_4) = 0$  中的  $x_1$  满足多项式方程

$$200(x_1 - 1)p(x_1) = 0,$$

其中

$$p(x_1) = 8000000x_1^8 + 8000000x_1^7 + 8120000x_1^6 + 8000000x_1^5 \\ + 4040600x_1^4 + 4039400x_1^3 + 2000201x_1^2 - 2x_1 + 1.$$

$x_1 = 1$  对应全局极小点，而  $p(x_1) = 0$  的两个实根对应 rosenbrock\_4d 另外两个驻点

$$x^\dagger \approx [-0.77566, 0.61309, 0.38206, 0.14597]^\top,$$

$$x^{\dagger\dagger} \approx [-0.65612, 0.44312, 0.20431, 0.04174]^\top.$$

利用二阶最优性条件，验证得  $x^\dagger, x^{\dagger\dagger}$  分别是 rosenbrock\_4d 的局部极小点和鞍点，对应目标函数值

$$f(x^\dagger) \approx 3.70142, \quad f(x^{\dagger\dagger}) \approx 3.70824.$$

## 2.2 线搜索算法

对于线搜索型优化方法，其给定下降方向后，需要由线搜索算法来决定步长。本实验主要采用两种线搜索算法：0.618 方法实现（近似的）精确线搜索，以及 Armijo-（强）Wolfe 准则实现非精确线搜索。

### 2.2.1 0.618 方法

0.618 方法由给定初始的步长区间，通过比较黄金分割点上目标函数值的大小，不断收缩步长区间，直至这个区间足够小来确定精确的步长。具体的算法介绍和算法步骤参见高立 (2014) 的 2.4 节。

值得强调的是，0.618 方法不是严格的“精确线搜索”，只能作为精确线搜索的近似手段。由于 0.618 方法只考虑步长区间上黄金分割点处的函数值比较，而忽略了区间上其他点以及步长区间之外点的函数值，导致在最优步长不一定落在 0.618 方法的步长区间内，甚至出现在下降方向上函数值不降反升的现象（因为函数在下降方向上并不永远下降，也不能保证没有多段上升和下降的区间）。只有当  $\varphi(\alpha) = f(x + \alpha d)$  是单峰函数时才能保证 0.618 方法是精确线搜索。为了方便起见，下文依然称 0.618 方法是精确线搜索方法。

### 2.2.2 Armijo–Wolfe 非精确线搜索

Armijo–（强）Wolfe 准则为

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho g_k^\top d_k \alpha, \quad (\text{Armijo})$$

$$|g(x_k + \alpha d_k)^\top d_k| \leq -\sigma g_k^\top d_k. \quad (\text{Wolfe})$$

其中  $0 < \rho < \sigma < 1$ ，通常取  $\rho = 10^{-4}$ ， $\sigma = 0.9$ 。

在具体实践中，Armijo–Wolfe 非精确线搜索将通过如下方式进行。

1. 给定当前点  $x$  与搜索方向  $d$ ，定义

$$\phi(\alpha) = f(x + \alpha d), \quad \phi'(\alpha) = \nabla f(x + \alpha d)^\top d.$$

设  $\phi_0 = \phi(0)$ ， $\phi'_0 = \phi'(0)$ ；若  $\phi'_0 \geq 0$ ，则搜索方向不是下降方向，算法报错退出。

2. 设初始步长  $\alpha^1$ （上标与实际步长区分），并令

$$\alpha^0 = 0, \quad \phi(\alpha^0) = \phi_0.$$

进入循环来确定最优步长所在的区间。

(a) 计算  $\phi(\alpha^k)$ 。

(b) 若

$$\phi(\alpha^k) > \phi_0 + \rho \alpha^k \phi'_0 \quad \text{或} \quad (k > 1 \text{ 且 } \phi(\alpha^k) \geq \phi(\alpha^{k-1})),$$

则说明步长过大，在区间  $[\alpha^{k-1}, \alpha^k]$  内二分搜索满足强 Wolfe 条件的步长并返回。

(c) 否则计算  $\phi'(\alpha^k)$ 。若

$$|\phi'(\alpha^k)| \leq -\sigma \phi'_0,$$

则  $\alpha^k$  同时满足 Armijo 条件和曲率条件，接受该步长并返回。

(d) 若  $\phi'(\alpha^k) \geq 0$ ，说明已经跨过最小值，在区间  $[\alpha^{k-1}, \alpha^k]$  内二分搜索满足强 Wolfe 条件的步长并返回。

(e) 否则当 Armijo 成立、曲率条件不满足且  $\phi'(\alpha^k) < 0$  时，将区间右端点外扩。令

$$\alpha^{k+1} = \min(\beta \cdot \alpha^k, \alpha_{\max}),$$

其中  $\beta$  是扩张系数，并令  $k \leftarrow k + 1$  继续循环。

## 2.3 优化算法

### 2.3.1 最速下降法

最速下降法选取负梯度为迭代方向，

$$d_k = -g_k,$$

显然当  $x_k \neq x^*$ ，即  $g_k \neq 0$  时，有

$$g_k^\top d_k = -g_k^\top g_k < 0,$$

即  $d_k$  一定是下降方向。

由于最速下降法只利用了梯度信息，所以不具有二次终止性。

### 2.3.2 Newton 法

Newton 法选取

$$d_k = -G_k^{-1} g_k,$$

其中  $G_k = \nabla^2 f(x_k)$  是  $x_k$  处的 Hessian 矩阵。下面定理说明 Newton 法具有很好的局部收敛速度。

**定理 2.1** (Newton 法的收敛速度). 假设目标函数  $f$  是二阶连续可微的函数，且 Hessian 矩阵在最优值点  $x^*$  的一个邻域  $N_\delta(x^*)$  内是 Lipschitz 连续的。如果函数  $f(x)$  在点  $x^*$  处满足  $\nabla f(x^*) = 0$  且 Hessian 矩阵正定，则 Newton 法具有二次收敛性，即  $\{x_k\}$   $Q$ -二次收敛到  $x^*$ ，同时  $\{\|g_k\|\}$   $Q$ -二次收敛到 0。

然而，对于一般的非凸函数且非极小点的充分小邻域内，不能保证 Hessian 矩阵的非奇异性 and 正定性，也无法保证  $d_k$  是下降方向。通常可以用修正 Newton 法

$$d_k = -(G_k + \lambda_k I)^{-1} g_k$$

确保  $d_k$  是下降方向。在本实验中，为方便起见，当  $G_k$  奇异或  $d_k = -G_k^{-1} g_k$  不是下降方向时，直接取负梯度方向  $d_k = -g_k$ 。

### 2.3.3 BFGS 拟 Newton 法

拟 Newton 法通过利用迭代点和梯度信息，维护近似 Hessian 矩阵的逆矩阵  $H_k$ ，并利用 Newton 法的算法框架进行迭代。BFGS 拟 Newton 法通过以下公式更新  $H_k$ ，

$$H_{k+1} = H_k + \left(1 + \frac{y_k^\top H_k y_k}{y_k^\top s_k}\right) \frac{s_k s_k^\top}{y_k^\top s_k} - \left(\frac{s_k y_k^\top H_k + H_k y_k s_k^\top}{y_k^\top s_k}\right). \quad (\text{BFGS})$$

下面定理说明了 BFGS 拟 Newton 法在一定条件下具有接近 Newton 法的收敛速度。

**定理 2.2** (BFGS 的收敛速度). 设  $f(x)$  二阶连续可微，在最优点  $x^*$  的一个邻域内 Hessian 矩阵 Lipschitz 连续，且使用 BFGS 迭代格式收敛到  $f$  的最优值点  $x^*$ 。若迭代点列  $\{x_k\}$  满足

$$\sum_{k=1}^{\infty} \|x_k - x^*\| < +\infty, \quad (1)$$

则  $\{x_k\}$  以  $Q$ -超线性收敛到  $x^*$ 。

同时, BFGS 还具有保下降方向的性质, 由下面定理阐述。

**定理 2.3** (BFGS 保下降方向). 设近似 *Hessian* 逆矩阵  $H_k$  对称正定, 且  $s_k^\top y_k > 0$ , 则 *BFGS* 公式下的  $H_{k+1}$  对称正定; 对于使用精确线搜索或 (强) *Wolfe* 准则的 *BFGS* 方法, 有  $s_k^\top y_k > 0$ 。

在实践中, 当  $y_k^\top s_k$  很小时会带来数值不稳定性。通常会用不更新近似 *Hessian* 逆矩阵的方式, 即令  $H_{k+1} = H_k$  来处理。

### 2.3.4 共轭梯度法

共轭梯度法的迭代方向为

$$d_k = -g_k + \beta_{k-1}d_{k-1},$$

其中

$$\beta_{k-1} = \frac{g_k^\top g_k}{g_{k-1}^\top g_{k-1}}, \quad (\text{FR})$$

$$\beta_{k-1} = \frac{g_k^\top (g_k - g_{k-1})}{g_{k-1}^\top g_{k-1}}, \quad (\text{PRP})$$

分别称为 **FR** 公式和 **PRP** 公式。对  $n$  维正定二次函数, 若不考虑舍入误差, 则共轭梯度法具有  $n$  次收敛的性质。这个性质得益于共轭梯度法迭代方向的共轭性质。

然而对于一般目标函数, 目标函数的 *Hessian* 矩阵不再是常数矩阵, 多步迭代方向的共轭性似乎就不具有实际意义。只有在精确线搜索 (或满足一定条件的非精确线搜索) 下, 才能保证迭代方向是下降方向。

**定理 2.4** (共轭梯度法保下降方向). 在非线性共轭梯度法中, 使用精确线搜索可以保证迭代方向的下降性。

**证明.** 对精确线搜索, 有  $g_k^\top d_{k-1} = 0$ , 从而

$$g_k^\top d_k = -g_k^\top g_k + \beta_{k-1}g_k^\top d_{k-1} < 0.$$

□



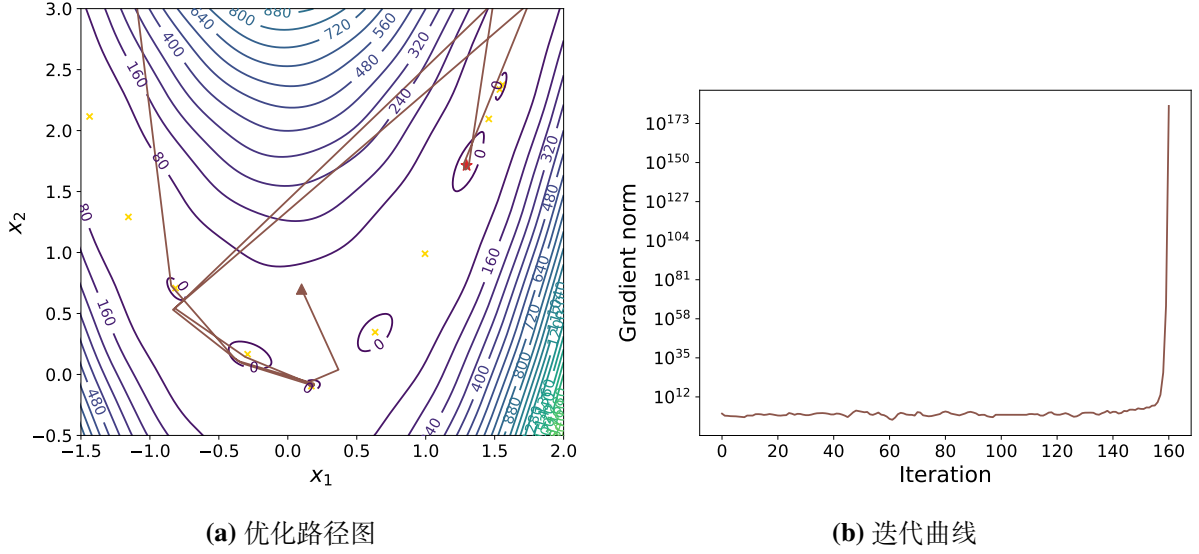


图 2: modified\_rosenbrock\_2d 算例, CG-FR (无重新开始策略), 精确线搜索, 初始点  $[0.1, 0.7]^T$

如图 2 所示, 采用 0.618 方法的精确线搜索, 多步之后共轭梯度法的优化路径变得凌乱, 出现了在多个极小点间大步长“横跳”的现象; 当迭代点出现在全局极小点附近, 也未能落入其中; 最后迭代发散。这是否与定理 2.4 的结论相违背呢? 事实上, 我们在 2.2.1 中已经强调, 0.618 方法不是严格的精确线搜索, 并不保证下一迭代点的函数值下降, 自然也不保证定理 2.4 保下降方向的性质。所以出现图 2 的现象是可能的。

因此, 通常共轭梯度法需要采用  $n$  步重新开始策略, 即每隔  $n$  步, 重置迭代方向为负梯度方向  $-g_k$  来保证方向的下降性。

### 2.3.5 信赖域方法

信赖域方法利用二次函数刻画目标函数在局部邻域内的形状, 通过近似二次函数在邻域内的极小点来更新迭代点的位置, 并根据下降量动态调整邻域的半径。其中, 近似二次函数可由梯度和 Hessian 矩阵的信息给出。信赖域方法在每一个迭代步都要求解一个信赖域子问题,

$$\min m(d) = f + g^\top d + \frac{1}{2} d^\top B d, \quad \text{s.t. } \|d\| \leq \Delta.$$

有关求解信赖域子问题的方法, 参见刘浩洋 et al. (2020) 的 6.6.2 节。本实验中采取迭代法求解信赖域子问题。

## 3 数值实验

经过数值实验, 下面将展示实验结果, 并对实验结果进行分析。

下面只展示采用 Armijo-Wolfe 非精确线搜索的结果。采用 0.618 方法做近似精确线搜索可以得到大致相同的现象, 图表见附录 A。事实上, 从实验结果可以看到, 在两个算例上 0.618 方法的近似精确线搜索不如 Armijo-Wolfe 非精确线搜索稳定。

### 3.1 实验结果和分析

对算例 `modified_rosenbrock_2d`, 选取初始点  $[-1.2, 1]^\top$ , 采用 Armijo-Wolfe 非精确线搜索, 得到的结果如表 1 和图 3, 4 所示。

表 1: `modified_rosenbrock_2d` 算例, 非精确线搜索, 初始点  $[-1.2, 1]^\top$  的结果数据

算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
Gradient Descent	65	0.809	$9.348 \times 10^{-7}$	局部极小点
Newton	37	-2.640	$3.236 \times 10^{-7}$	局部极小点
BFGS	9	0.809	$4.004 \times 10^{-8}$	局部极小点
CG-FR	35	0.809	$5.411 \times 10^{-7}$	局部极小点
CG-PRP	33	0.809	$6.032 \times 10^{-7}$	局部极小点
Trust Region	5	4.907	$5.074 \times 10^{-7}$	局部极小点

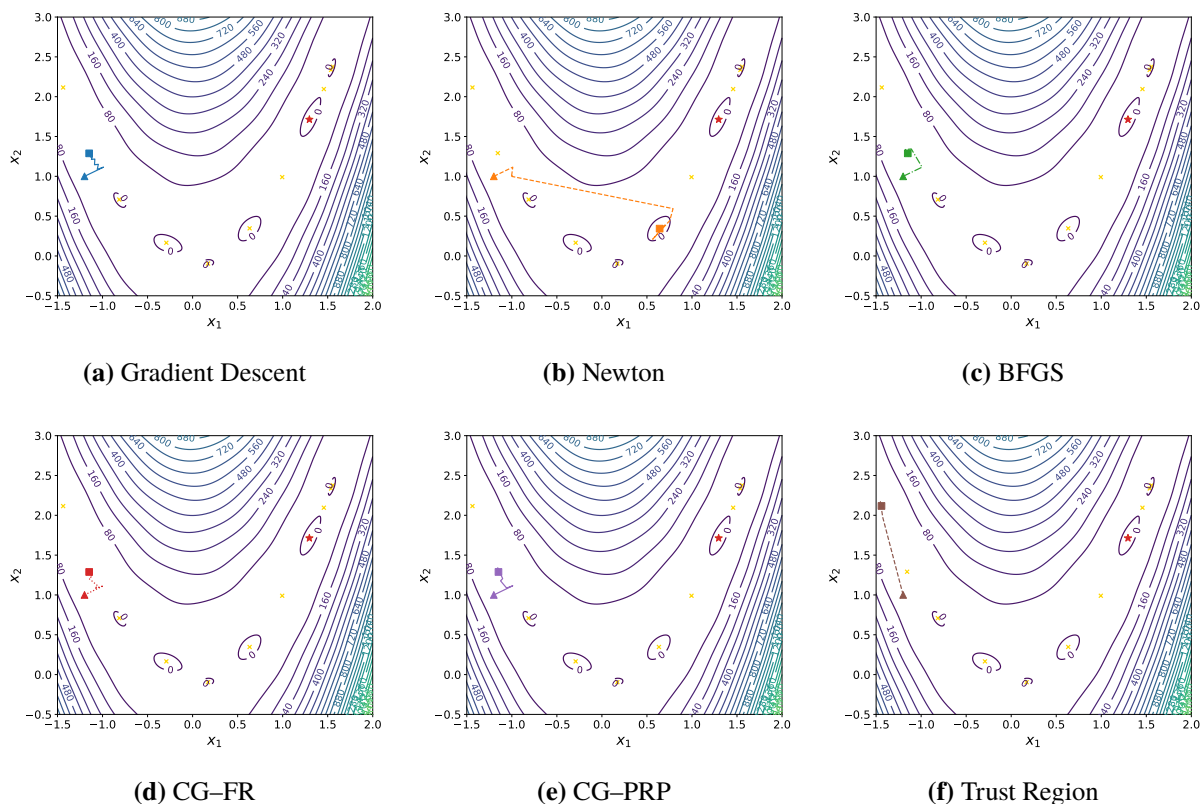


图 3: `modified_rosenbrock_2d` 算例, 非精确线搜索, 初始点  $[-1.2, 1]^\top$  的优化路径图

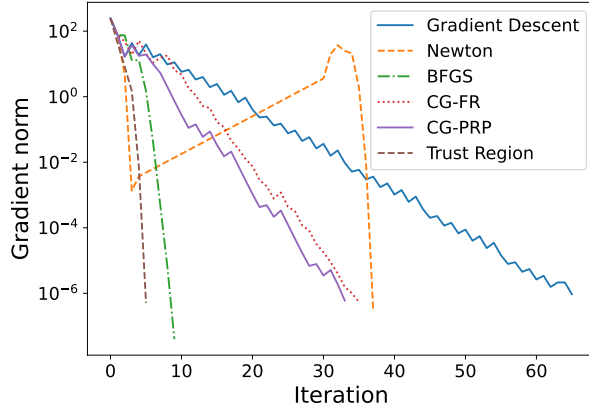


图 4: modified\_rosenbrock\_2d 算例, 非精确线搜索, 初始点  $[-1.2, 1]^T$  的迭代曲线

从表 1 和图 3 来看, 6 种算法均没有找到全局极小点。且除了 Newton 法和信赖域方法, 其他 4 种算法收敛到了离初始点最近的局部极小点。Newton 法找到了函数值更小的局部极小点, 信赖域则收敛到了函数值更大的局部极小点。

从表 1 和图 4 中, 我们可以观察到 6 种算法收敛速度的不同。从迭代初期看, Newton 法收敛最快, 之后由于找到了更好的下降方向和更好的极小点而有一段梯度范数的上升, 迭代的最后又迅速地收敛。信赖域方法和 BFGS 拟 Newton 法的收敛速度次之, 共轭梯度法的收敛速度较慢, 最速下降法的收敛速度最慢。这个结果符合我们的预期。

选取其他初始点

$$[0, 1.5]^T, \quad [0.6, 2.5], \quad [-1.5, 1]^T, \quad [0.1, 0.7]^T$$

的结果如表 2 和图 5, 6 所示, 现象与上述算法收敛速度的定性比较是一致的。

表 2: modified\_rosenbrock\_2d 算例, 非精确线搜索, 其他初始点的结果数据

初始点	算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
$[0, 1.5]^\top$	Gradient Descent	35	-1.851	$6.588 \times 10^{-7}$	局部极小点
	Newton	7	-0.254	$9.894 \times 10^{-14}$	局部极小点
	BFGS	11	-1.851	$6.057 \times 10^{-8}$	局部极小点
	CG-FR	68	6.140	$6.432 \times 10^{-7}$	局部极小点
	CG-PRP	35	-1.851	$6.588 \times 10^{-7}$	局部极小点
	Trust Region	6	-0.953	$1.867 \times 10^{-10}$	局部极小点
$[0.6, 2.5]^\top$	Gradient Descent	77	-4.505	$7.079 \times 10^{-7}$	全局极小点
	Newton	4	1.010	$8.715 \times 10^{-9}$	鞍点
	BFGS	11	0.000	$1.332 \times 10^{-8}$	局部极小点
	CG-FR	22	-1.851	$8.377 \times 10^{-7}$	局部极小点
	CG-PRP	88	-4.505	$7.110 \times 10^{-7}$	全局极小点
	Trust Region	5	-0.718	$5.649 \times 10^{-8}$	局部极小点
$[-1.5, 1]^\top$	Gradient Descent	88	-4.505	$5.874 \times 10^{-7}$	全局极小点
	Newton	6	2.815	$8.883 \times 10^{-8}$	局部极小点
	BFGS	10	-4.505	$2.968 \times 10^{-9}$	全局极小点
	CG-FR	26	-4.505	$9.728 \times 10^{-7}$	全局极小点
	CG-PRP	21	-2.640	$8.076 \times 10^{-7}$	局部极小点
	Trust Region	6	4.907	$1.452 \times 10^{-11}$	局部极小点
$[0.1, 0.7]^\top$	Gradient Descent	36	-1.851	$5.920 \times 10^{-7}$	局部极小点
	Newton	6	-0.254	$4.840 \times 10^{-11}$	局部极小点
	BFGS	9	-2.640	$7.924 \times 10^{-9}$	局部极小点
	CG-FR	30	-1.851	$1.372 \times 10^{-7}$	局部极小点
	CG-PRP	23	-2.640	$5.149 \times 10^{-7}$	局部极小点
	Trust Region	8	-2.640	$3.992 \times 10^{-10}$	局部极小点

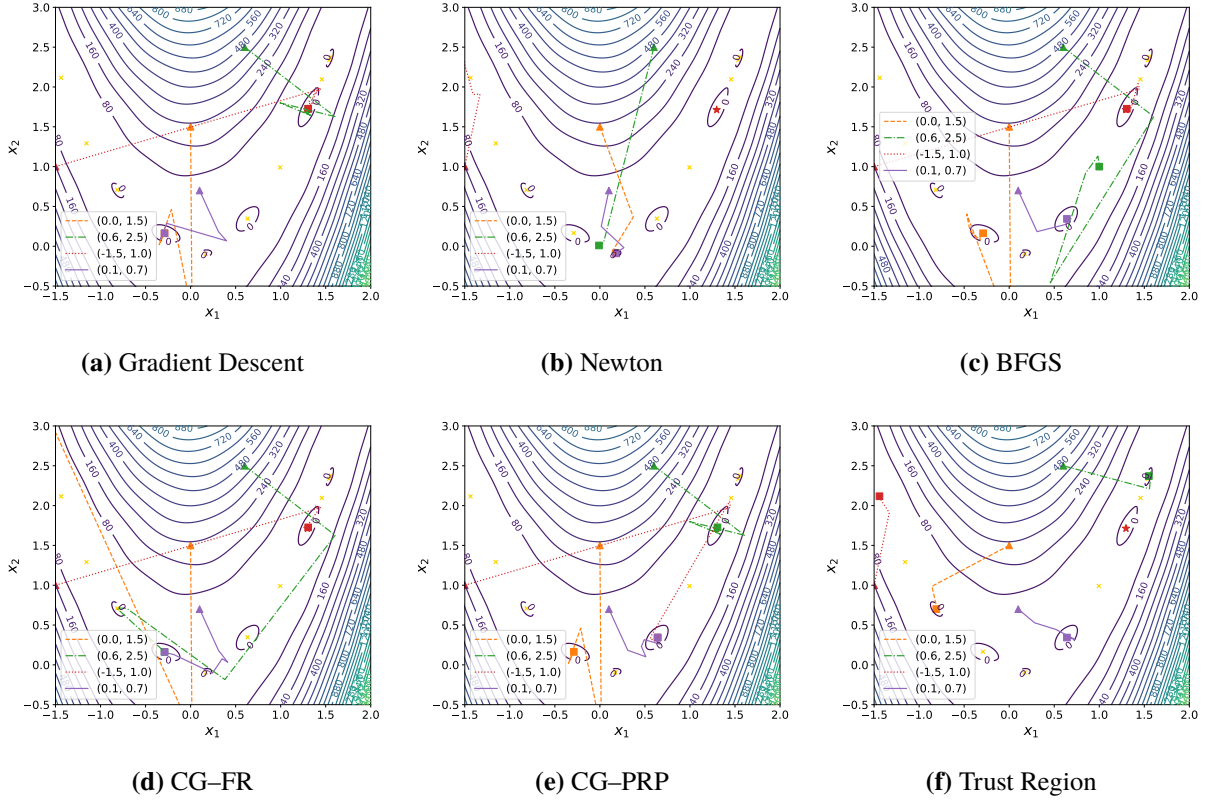


图 5: modified\_rosenbrock\_2d 算例, 非精确线搜索, 其他初始点的优化路径图

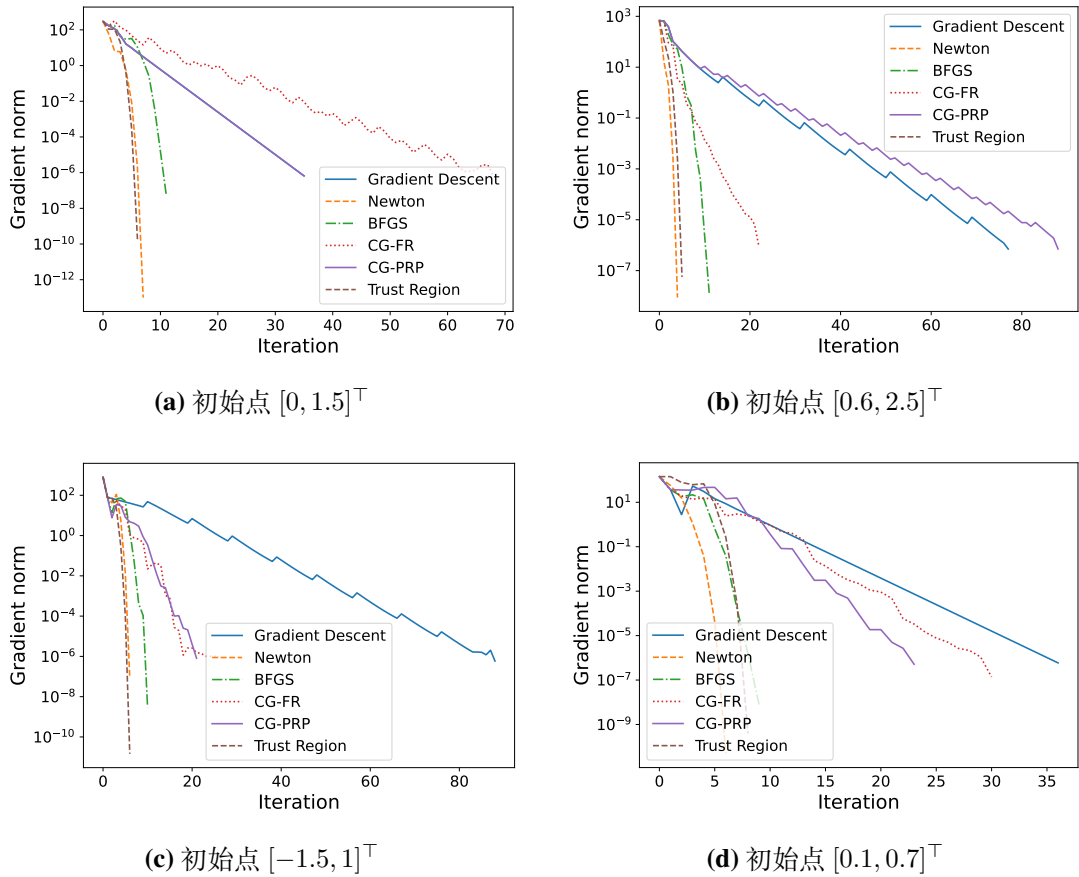


图 6: modified\_rosenbrock\_2d 算例, 非精确线搜索, 其他初始点的迭代曲线

从不同初始点的实验来看，

1. 6 种算法虽然都有能力找到目标函数的极小点，但也均没有跳出局部极小点寻找到全局极小点的能力。只有当初始点足够接近全局极小点时，算法才更有可能找到全局极小点，但这不是本质的；
2. 收敛速度上，Newton 法、BFGS 拟 Newton 法以及信赖域方法是最快的，其次是共轭梯度法，最速下降法最慢；
3. 值得强调的是，从优化路径上看，信赖域方法最保守，常常落入离初始点最近的极小点；而共轭梯度法相对更加“激进”，这很可能与“共轭方向”的性质有关；
4. 从迭代曲线上看，通常 Newton 法、BFGS 拟 Newton 法以及信赖域方法的曲线几乎不震荡，说明它们的演化会更稳健；相反，梯度下降法以及共轭梯度法的稳定性较差。

对算例 `rosenbrock_4d`，选取初始点  $[-1.2, 1, -0.8, 0.6]^T$ ，采用 Armijo-Wolfe 非精确线搜索，得到的结果如表 3 和图 7 所示。

表 3: `rosenbrock_4d` 算例，非精确线搜索，初始点  $[-1.2, 1, -0.8, 0.6]^T$  的结果数据

算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
Gradient Descent	1000	$1.778 \times 10^{-2}$	$3.348 \times 10^{-1}$	未收敛
Newton	10	$3.701 \times 10^0$	$1.112 \times 10^{-12}$	局部极小点
BFGS	40	$7.454 \times 10^{-17}$	$3.361 \times 10^{-7}$	全局极小点
CG-FR	579	$2.248 \times 10^{-13}$	$9.318 \times 10^{-7}$	全局极小点
CG-PRP	309	$2.259 \times 10^{-13}$	$8.061 \times 10^{-7}$	全局极小点
Trust Region	11	$3.701 \times 10^0$	$8.163 \times 10^{-8}$	局部极小点

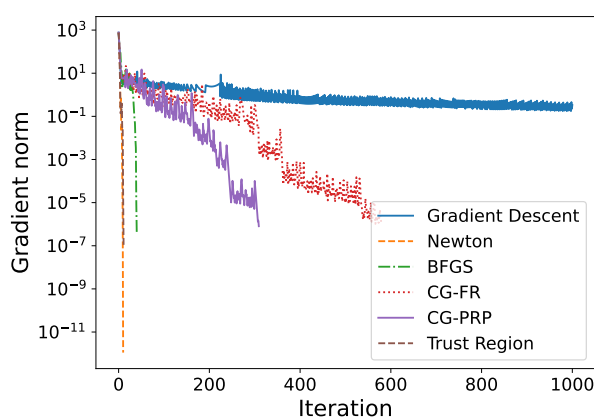


图 7: `rosenbrock_4d` 算例，非精确线搜索，初始点  $[-1.2, 1, -0.8, 0.6]^T$  的迭代曲线

选取其他初始点

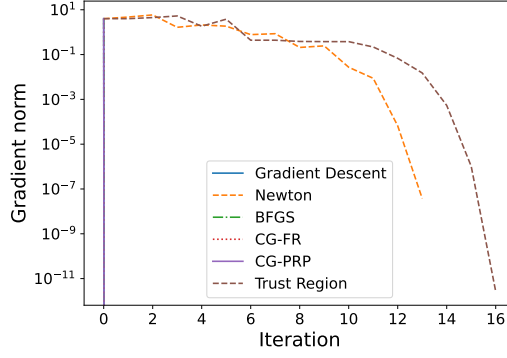
$$[-1, 1, 1, 1]^T, \quad [-1.2, 1, 0.5, 0.5]^T, \quad [-1.5, 1, 1, 1]$$

的结果如表 4 和图 8 所示。

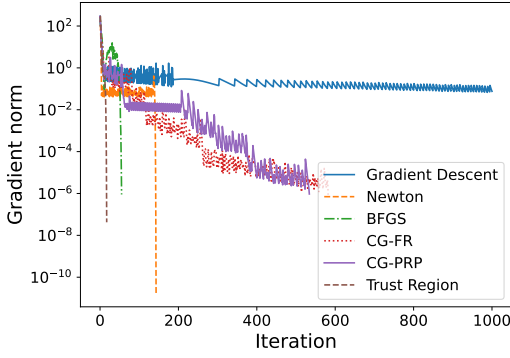
表 4: rosenbrock\_4d 算例, 非精确线搜索, 其他初始点的结果数据

初始点	算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
$[-1, 1, 1, 1]^\top$	Gradient Descent	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	Newton	13	$3.701 \times 10^0$	$3.791 \times 10^{-8}$	局部极小点
	BFGS	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	CG-FR	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	CG-PRP	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	Trust Region	16	$3.701 \times 10^0$	$2.674 \times 10^{-12}$	局部极小点
$[-1.2, 1, 0.5, 0.5]^\top$	Gradient Descent	1000	$3.704 \times 10^0$	$7.573 \times 10^{-2}$	未收敛
	Newton	143	$3.701 \times 10^0$	$1.765 \times 10^{-11}$	局部极小点
	BFGS	55	$6.361 \times 10^{-16}$	$8.968 \times 10^{-7}$	全局极小点
	CG-FR	582	$3.701 \times 10^0$	$7.750 \times 10^{-7}$	局部极小点
	CG-PRP	534	$3.701 \times 10^0$	$9.439 \times 10^{-7}$	局部极小点
	Trust Region	17	$3.701 \times 10^0$	$4.205 \times 10^{-8}$	局部极小点
$[-1.5, 1, 1, 1]^\top$	Gradient Descent	1000	$3.752 \times 10^0$	$8.403 \times 10^{-1}$	未收敛
	Newton	20	$3.701 \times 10^0$	$2.051 \times 10^{-7}$	局部极小点
	BFGS	59	$6.625 \times 10^{-20}$	$2.843 \times 10^{-9}$	全局极小点
	CG-FR	783	$3.701 \times 10^0$	$5.956 \times 10^{-7}$	局部极小点
	CG-PRP	445	$3.701 \times 10^0$	$9.811 \times 10^{-7}$	局部极小点
	Trust Region	21	$3.701 \times 10^0$	$1.147 \times 10^{-12}$	局部极小点

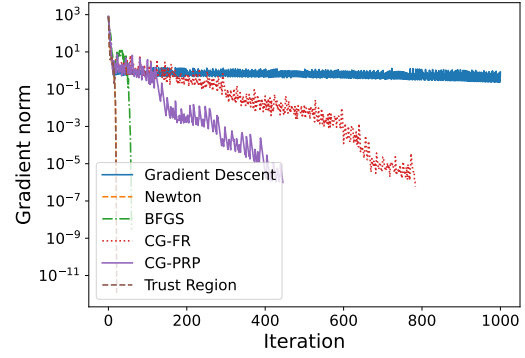




(a) 初始点  $[-1, 1, 1, 1]^T$



(b) 初始点  $[-1.2, 1, 0.5, 0.5]^T$



(c) 初始点  $[-1.5, 1, 1, 1]^T$

图 8: rosenbrock\_4d 算例, 非精确线搜索, 其他初始点的迭代曲线

在算例 `rosenbrock_4d` 上的结论与 `modified_rosenbrock_2d` 没有区别。同样, 各个算法均没有保证收敛到全局极小点的能力 (即使 BFGS 在表 3, 4 中均收敛到全局极小点, 但事实上是有可能收敛到局部极小点的, 见附录 A 的表 8)。同时, 更高维的算例使得算法收敛速度的差距更大。Newton 法、BFGS 拟 Newton 法以及信赖域方法优势显著, 共轭梯度法次之, 而算例 `rosenbrock_4d` 对于最速下降法是困难的, 虽然已经落在了极小点附近, 但由于问题的维度高, 对一般初始点 1000 次迭代还远远不能满足收敛准则。

### 3.2 代码使用说明

代码可以由附件或者 [GitHub](#) 获取。优化算法部分用 Matlab 编写, 绘图模块由 Python 编写, 并利用 Python 脚本驱动运行。支持以下的运行方式,

1. 运行 `run/run_modified_rosenbrock_single.py` 执行 `modified_rosenbrock_2d` 算例的单组参数实验, 参数由 `configs/modified_rosenbrock_single_config.json` 配置;
  2. 运行 `run/run_modified_rosenbrock_pipeline.py` 一次性执行 `modified_rosenbrock_2d` 算例的全部实验;
  3. 运行 `run/run_rosenbrock4_single.py` 执行 `rosenbrock_4d` 算例的单组参数实验, 参数由 `configs/rosenbrock4_single_config.json` 配置;
  4. 运行 `run/run_rosenbrock4_pipeline.py` 一次性执行 `rosenbrock_4d` 算例的全部实验。
- 实验的所有数据见 `logs/` 和 `outputs/`。



## 4 思考题

### 4.1 为什么 Newton 法在二次函数上能“一步收敛”，而最速下降法需要多次迭代？

设目标函数是严格凸二次函数，

$$f(x) = \frac{1}{2}x^\top Qx + b^\top x + c,$$

其中  $Q$  是正定矩阵。严格凸二次函数具有唯一全局极小点

$$x^* = -Q^{-1}b.$$

由严格凸二次函数的梯度和 Hessian 矩阵

$$g(x) = Qx + b, \quad G(x) = Q.$$

可以得到 Newton 步

$$x_{k+1} = x_k - G(x_k)^{-1}g(x_k) = x_k - Q^{-1}(Qx_k + b) = x^*.$$

即 Newton 法在严格凸二次函数上能“一步收敛”。

而对于最速下降法，由于只利用到目标函数的梯度信息，对于二次函数，对一般初始点不能“一步收敛”。从优化路径图（如图 3）上也能看出，其优化路径一般成锯齿状，对二次函数上的一般初始点也是如此。

### 4.2 BFGS 拟 Newton 法无需直接计算 Hessian 矩阵，却能接近 Newton 法的收敛速度，其核心思想是什么？

BFGS 拟 Newton 法通过维护近似 Hessian 逆矩阵  $H_k \approx G(x_k)^{-1}$ ，实现与 Newton 法类似的效果。

近似 Hessian 逆矩阵满足割线方程

$$H_{k+1}y_k = s_k,$$

其中  $s_k = x_{k+1} - x_k$ ， $y_k = g(x_{k+1}) - g(x_k)$ 。在满足割线方程的前提下，使得  $H_{k+1} - H_k$  在矩阵范数的意义下最小，即

$$\min_H \|H - H_k\|_W,$$

$$\text{s.t. } H = H^\top,$$

$$Hy_k = s_k,$$

其中  $W$  可以是任意满足割线方程的矩阵，

$$\|H\|_W = \|W^{1/2}HW^{1/2}\|_F,$$

这样就生成了 BFGS 公式。

BFGS 逐步累积曲率信息，得到越来越好的二阶近似，因此在极小值附近通常具有超线性收敛速度，效果接近 Newton 法。定理 2.2 也保证了在一定条件下 BFGS 有局部超线性的收敛速度。

从实验结果的迭代曲线（如图 7）上也可以看出，BFGS 拟 Newton 法具有超线性的收敛速度，且收敛速度接近 Newton 法。

### 4.3 共轭梯度法在非二次函数上的收敛性能为何不如 BFGS？这与“共轭方向”的性质有何关系？

对于严格二次函数，共轭梯度法通过构造一系列共轭方向，在精确线搜索和精确算术（没有浮点误差）下，共轭梯度法至多  $n$  步就能到达极小点，非常高效。

但对一般非二次函数，**Hessian 矩阵不再是固定的常数矩阵**。共轭梯度法在之前迭代步生成的下降方向在新的 Hessian 矩阵下不再是共轭方向，共轭性被破坏，方向也不能像二次情形那样张成理想的搜索子空间。所以对于非二次函数，共轭梯度法的性能相当有限。

相比之下，BFGS 每一步都在更新近似 Hessian 矩阵，比共轭梯度法更稳定也更快。

从实验结果的图表（如图 7）中也可以看出，共轭梯度法的迭代曲线更加震荡，同时收敛速度不如 Newton 型算法（Newton 法、BFGS 拟 Newton 法）。若不采取  $n$  步重新开始策略，共轭梯度法甚至会出现图 2 中的失稳现象，这也与共轭方向被破坏有关；而在实验的其他算法上并没有发现这种现象。

## 5 总结与反思

本实验以 `modified_rosenbrock_2d` 和 `rosenbrock_4d` 两个算例为平台，对最速下降法、Newton 法、BFGS 拟 Newton 法、共轭梯度法（FR/PRP）以及信赖域方法进行了系统的数值实验，结合 0.618 精确线搜索和 Armijo-Wolfe 非精确线搜索，对比了不同算法在收敛速度、数值稳定性以及对初始点敏感性等方面的差异。

整体来看，实验结果与课本中的理论结论基本一致。Newton 法在极小点附近具有最快的局部收敛速度，BFGS 拟 Newton 法在不显式构造 Hessian 的前提下，通过累积割线信息构造 Hessian 近似，数值表现非常接近 Newton 法；信赖域方法借助模型函数与信赖域半径的双重控制，表现出较强的稳健性和对非凸结构的适应能力；相比之下，最速下降法虽然实现最为简单，但收敛速度显著偏慢，对目标函数的各向异性非常敏感，共轭梯度法在非二次情形下的优势也远不如二次情形的  $n$  步收敛。

在 `modified_rosenbrock_2d` 上，所有算法在一般初始点下都更容易收敛到附近的局部极小点，而缺乏系统跳出局部极小点、寻找全局极小点的机制；只有当初始点已经落在全局极小点附近的山谷区域时，部分算法才有机会到达全局极小点，这说明经典确定性梯度法**并不擅长应对多极值非凸问题**。在 `rosenbrock_4d` 算例中，随着维数升高，算例的弯谷结构更加狭长，最速下降法在给定最大迭代步数内往往难以达到收敛判据；Newton 法、BFGS 法和信赖域方法仍能在较少步数内收敛，但是否到达全局极小点依旧强烈依赖初始点位置。BFGS 和共轭梯度法在部分初始点下成功找到全局极小点，而在另一些初始点下则停在局部极小点，这进一步说明算法、线搜索策略以及初始点的联合作用，远比单独讨论算法框架更为关键。综合本次实验，可以得到以下几点体会，

1. Newton 型方法（包括 BFGS 和信赖域方法）在工程实践中通常是更有竞争力的选择，它们通过利用或近似 Hessian 信息，大幅改善了搜索方向的质量；
2. 线搜索策略对整体性能影响显著，Armijo-Wolfe 非精确线搜索在稳定性上往往优于 0.618 方法的近似精确线搜索；

3. 经典算法对初始点的敏感性和对局部极值的困陷，在数值实验中表现得非常直观，这提醒我们在实际应用中往往需要辅以多起点策略、随机扰动或全局启发式方法。

本文的实验设计和分析仍存在不少不足，有待后续改进和完善，主要包括，

1. 测试函数仅选取了 `modified_rosenbrock_2d` 与 `rosenbrock_4d` 两个算例，维度较低、结构相对简单，尚未在更高维、更复杂或真实应用问题上验证结论的普适性；
2. 对线搜索参数（如 Armijo–Wolfe 中的  $\rho$ 、 $\sigma$ 、初始步长与扩张因子等）的敏感性分析不够系统，精确线搜索与非精确线搜索在效率与稳健性上的差异也可以通过更多定量指标（如函数调用次数）进行深入比较；
3. 信赖域子问题采用的数值求解方法较为基础，没有对比 Cauchy 点、Dogleg 法等不同子问题算法对总体表现的影响；
4. 对于共轭梯度法的失稳与发散现象，本文主要给出了数值现象和定性解释，尚未结合具体的迭代轨迹和 Hessian 谱性质做更细致的理论分析，例如系统讨论重启策略、阻尼策略对避免发散的作用；

总之，本实验通过实现和对比多种无约束优化算法，加深了对算法框架、线搜索策略、目标函数结构以及初始点选择之间相互作用的理解，也暴露了经典确定性梯度法在非凸、多极值问题上的局限性。未来若能在更多算例和实际应用场景中扩展这些实验，并结合随机优化、全局优化或机器学习方法进行改进，将有望得到更具实用价值的算法和经验。

希望老师批评指正！

## 参考文献

- Kok, Schalk and Carl Sandrock (2009). “Locating and characterizing the stationary points of the extended rosenbrock function”. In: *Evolutionary computation* 17.3, pp. 437–453.
- 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.

# 附录

## A 精确线搜索的图表数据

以下的图表数据与 3.1 中的是平行的，只是将线搜索方法换成了 0.618 方法。

表 5: modified\_rosenbrock\_2d 算例，精确线搜索，初始点  $[-1.2, 1]^T$  的结果数据

算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
Gradient Descent	21	0.8091	$8.243 \times 10^{-7}$	局部极小点
Newton	4	-0.7183	$3.205 \times 10^{-7}$	局部极小点
BFGS	7	-0.7183	$9.193 \times 10^{-7}$	局部极小点
CG-FR	78	30.86	$5.293 \times 10^{-8}$	局部极小点
CG-PRP	40	-2.640	$2.103 \times 10^{-7}$	局部极小点
Trust Region	5	4.907	$5.074 \times 10^{-7}$	局部极小点

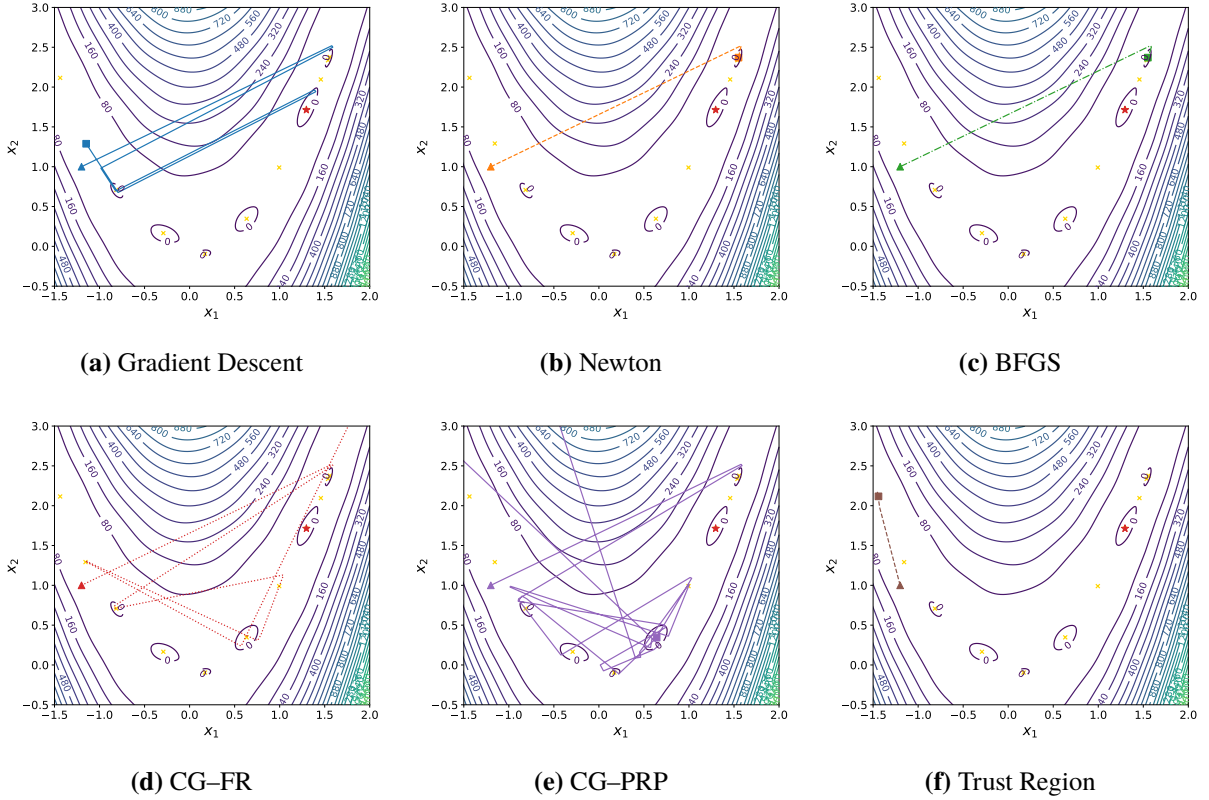


图 9: modified\_rosenbrock\_2d 算例，精确线搜索，初始点  $[-1.2, 1]^T$  的优化路径图

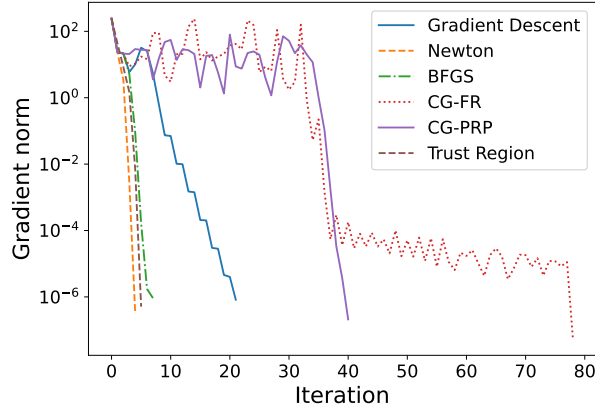


图 10: modified\_rosenbrock\_2d 算例, 精确线搜索, 初始点  $[-1.2, 1]^\top$  的迭代曲线

表 6: modified\_rosenbrock\_2d 算例, 精确线搜索, 其他初始点的结果数据

初始点	算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
$[0, 1.5]^\top$	Gradient Descent	8	-0.254	$2.230 \times 10^{-7}$	局部极小点
	Newton	5	-0.254	$3.956 \times 10^{-7}$	局部极小点
	BFGS	8	-0.254	$5.799 \times 10^{-7}$	局部极小点
	CG-FR	8	-0.254	$2.396 \times 10^{-7}$	局部极小点
	CG-PRP	7	-0.254	$3.391 \times 10^{-7}$	局部极小点
	Trust Region	6	-0.953	$1.867 \times 10^{-10}$	局部极小点
$[0.6, 2.5]^\top$	Gradient Descent	311	0.262	$3.283 \times 10^{-7}$	局部极小点
	Newton	7	-0.254	$1.359 \times 10^{-7}$	局部极小点
	BFGS	10	-1.851	$1.987 \times 10^{-7}$	局部极小点
	CG-FR	16	0.000	$2.545 \times 10^{-7}$	局部极小点
	CG-PRP	23	-2.640	$4.052 \times 10^{-7}$	局部极小点
	Trust Region	5	-0.718	$5.649 \times 10^{-8}$	局部极小点
$[-1.5, 1]^\top$	Gradient Descent	241	23.400	$7.732 \times 10^{-7}$	局部极小点
	Newton	9	4.907	$8.411 \times 10^{-7}$	局部极小点
	BFGS	8	-4.505	$2.259 \times 10^{-7}$	全局极小点
	CG-FR	24	-0.254	$3.083 \times 10^{-7}$	局部极小点
	CG-PRP	9	0.809	$2.245 \times 10^{-7}$	局部极小点
	Trust Region	6	4.907	$1.452 \times 10^{-11}$	局部极小点
$[0.1, 0.7]^\top$	Gradient Descent	13	-0.254	$4.511 \times 10^{-7}$	局部极小点
	Newton	5	-0.254	$4.316 \times 10^{-7}$	局部极小点
	BFGS	7	-0.254	$7.350 \times 10^{-7}$	局部极小点
	CG-FR	8	-0.254	$2.117 \times 10^{-7}$	局部极小点
	CG-PRP	8	-0.254	$2.722 \times 10^{-7}$	局部极小点
	Trust Region	8	-2.640	$3.992 \times 10^{-10}$	局部极小点

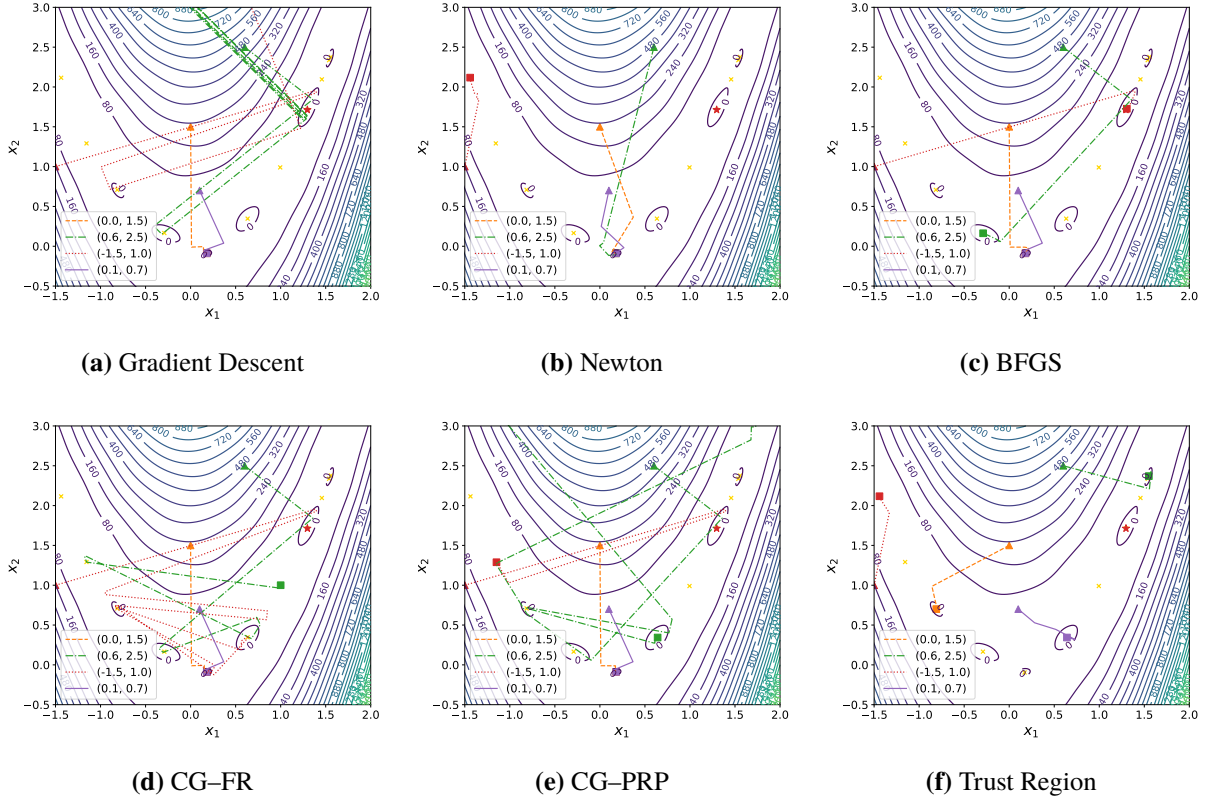


图 11: modified\_rosenbrock\_2d 算例, 精确线搜索, 其他初始点的优化路径图

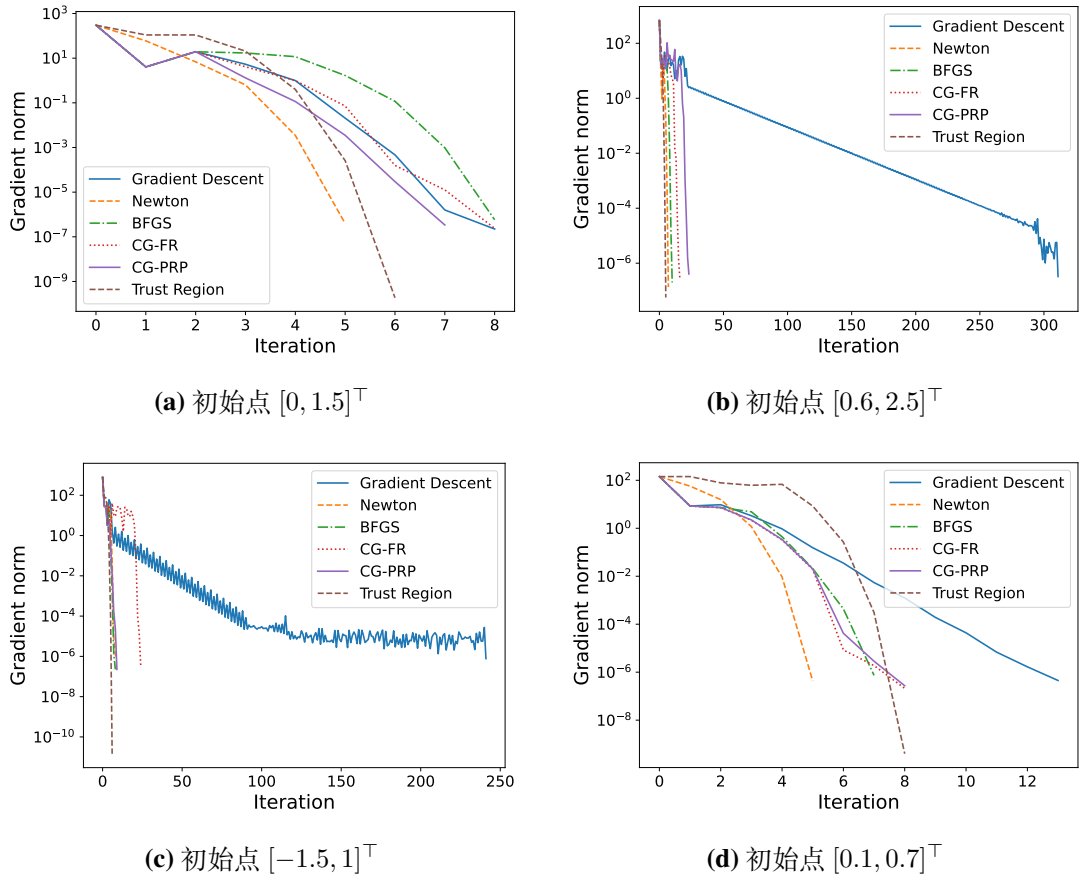


图 12: modified\_rosenbrock\_2d 算例, 精确线搜索, 其他初始点的迭代曲线

表 7: rosenbrock\_4d 算例，精确线搜索，初始点  $[-1.2, 1, -0.8, 0.6]^\top$  的结果数据

算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
Gradient Descent	1000	$2.543 \times 10^{-2}$	$3.122 \times 10^{-1}$	未收敛
Newton	10	$3.701 \times 10^0$	$1.620 \times 10^{-7}$	局部极小点
BFGS	29	$1.950 \times 10^{-18}$	$4.642 \times 10^{-8}$	全局极小点
CG-FR	81	$5.891 \times 10^{-16}$	$6.935 \times 10^{-7}$	全局极小点
CG-PRP	59	$2.356 \times 10^{-13}$	$9.302 \times 10^{-7}$	全局极小点
Trust Region	11	$3.701 \times 10^0$	$8.163 \times 10^{-8}$	局部极小点

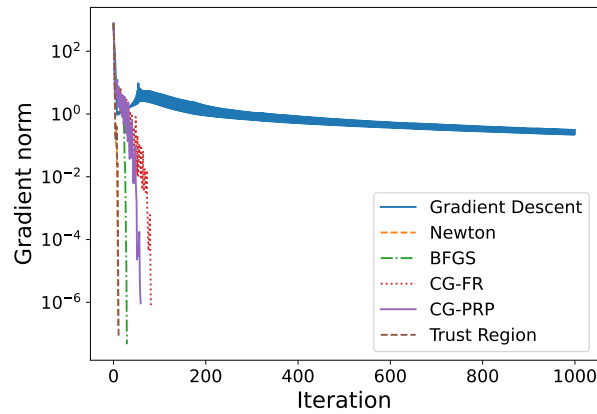


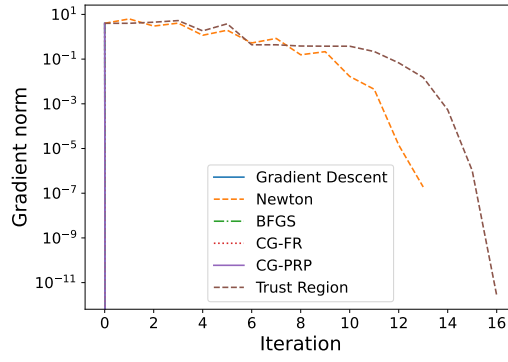
图 13: rosenbrock\_4d 算例，精确线搜索，初始点  $[-1.2, 1, -0.8, 0.6]^\top$  的迭代曲线



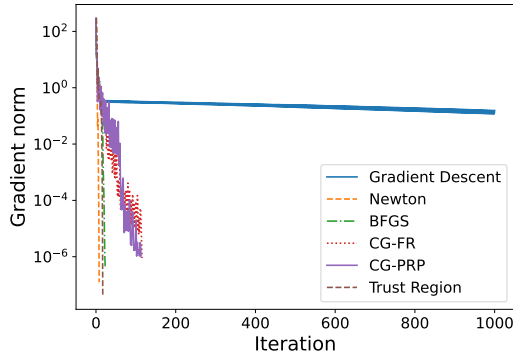
表 8: rosenbrock\_4d 算例，精确线搜索，其他初始点的结果数据

初始点	算法	迭代次数	终点 $f$ 值	终点 $\ \nabla f\ _2$	终点类型
$[-1, 1, 1, 1]^\top$	Gradient Descent	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	Newton	13	$3.701 \times 10^0$	$1.902 \times 10^{-7}$	局部极小点
	BFGS	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	CG-FR	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	CG-PRP	1	$0.000 \times 10^0$	$0.000 \times 10^0$	全局极小点
	Trust Region	16	$3.701 \times 10^0$	$2.674 \times 10^{-12}$	局部极小点
$[-1.2, 1, 0.5, 0.5]^\top$	Gradient Descent	1000	$3.711 \times 10^0$	$1.553 \times 10^{-1}$	未收敛
	Newton	8	$3.701 \times 10^0$	$1.230 \times 10^{-7}$	局部极小点
	BFGS	23	$3.701 \times 10^0$	$4.559 \times 10^{-7}$	局部极小点
	CG-FR	116	$3.701 \times 10^0$	$8.964 \times 10^{-7}$	局部极小点
	CG-PRP	113	$3.701 \times 10^0$	$9.967 \times 10^{-7}$	局部极小点
	Trust Region	17	$3.701 \times 10^0$	$4.205 \times 10^{-8}$	局部极小点
$[-1.5, 1, 1, 1]^\top$	Gradient Descent	1000	$3.804 \times 10^0$	$3.996 \times 10^{-1}$	未收敛
	Newton	20	$3.701 \times 10^0$	$9.797 \times 10^{-8}$	局部极小点
	BFGS	31	$3.701 \times 10^0$	$6.306 \times 10^{-7}$	局部极小点
	CG-FR	171	$3.701 \times 10^0$	$8.538 \times 10^{-7}$	局部极小点
	CG-PRP	135	$3.701 \times 10^0$	$8.051 \times 10^{-7}$	局部极小点
	Trust Region	21	$3.701 \times 10^0$	$1.147 \times 10^{-12}$	局部极小点

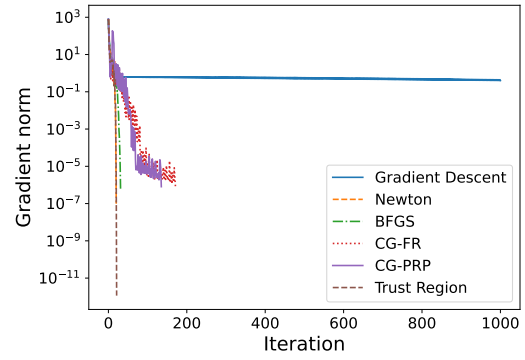




(a) 初始点  $[-1, 1, 1, 1]^T$



(b) 初始点  $[-1.2, 1, 0.5, 0.5]^T$



(c) 初始点  $[-1.5, 1, 1, 1]^T$

图 14: rosenbrock\_4d 算例, 精确线搜索, 其他初始点的迭代曲线