本文主要讨论了关于经典、“贪心”的、随机的BFGS算法在解正定对称系统中的显式超线性收敛率。

Introduction

对于非线性方程组的求解，也即考虑求解问题 $f(x)=0$, Newton方法具有局部二次收敛性质。在优化问题中，对应的Newton方法为，

$$x_{k+1}=x_k-\nabla^2 f(x_k)^{-1}\nabla f(x_k)$$

但其中涉及到Hessian矩阵的求逆公式，而在很多场景这个开销是花不起的，因此人们提出了Quasi-Newton方法。在优化中常用的Quasi-Newton方法包括DFP、BFGS、SR1三种算法，其均可以被概括为Broyden类算法，本质都是使用一个矩阵的迭代序列 G_k 不断逼近Hessian矩阵 $\nabla^2 f(x_k)$ 。与非线性方程求解中的Good Broyden和Bad Broyden方法不同的是，优化中需要保证 G_k 为对称正定矩阵，因此需要迭代格式可以满足结果也为对称正定矩阵。

对于Quasi-Newton算法，虽然不能证明其和Newton方法一样具有二次收敛性。但是对于一般的 L-光滑， μ- 强凸，并且Hessian矩阵满足Lipschitz的前提下，可以证明其具有超线性收敛性。但是在2021年之前的工作对于Broyden类的Quasi-Newton算法的超线性收敛的分析都是渐进意义下的结果，而人们仍然未知的是此类算法的收敛率的显式表达式。这个问题应该由Anton Rodomanov和Yurii Nesterov在2021年解决，这个工作可能对于Quasi-Newton类算法的理解是具有里程碑意义的，因此本文决定研究相关内容。

首先，实际上该问题可以先被简化为矩阵近似的问题。我们应当关心，给定矩阵 A, Quasi-Newton给出的矩阵迭代序列能以何种收敛率逼近矩阵 A . 如果我们找到了一个显式的收敛率，那么我们有理由相信，对于如下的二次函数的最小化问题，

$$\min_x f(x)=\frac{1}{2}x^\top Ax-b^\top x$$

Quasi-Newton算法应该也可以以一个显式的收敛率找到该问题的最优解。在此基础上，由于一个连续函数 $g(x)$ 的局部可以被二次函数所近似，那么我们也有理由相信，对于一般的 $g(x)$ 的最优化问题，在进入某个局部后，Quasi-Newton算法也将具有某一个显式的超线性收敛率。该局部就是Quasi-Newton算法的超线性收敛域。

对于上述问题，最近代表性的工作如下：Anton Rodomanov 与 Yurii Nesterov在 2021年的文章 [1] 提出了一种称为“贪心”的Quasi-Newton方法，其思想是选取一组标准正交基作为 G_k 的的迭代格式中的方向来满足 Secant方程，而非经典的Quasi-Newton算法中的前进方向 $x_{k+1}-x_k$。如果每次选取一个最好的基，使得矩阵近似的某个指标下降最快（也即 “贪心” 的含义“），那么可以得到对应算法的显式超线性收敛率。当然，相似的证明策略也可以用于经典的Quasi-Newton方法的证明中，在这两位作者同年的文章 [2] 就基于类似的证明手段给出了经典的Quasi-Newton方法的显式收敛率。 但由于经典的Quasi-Newton方法并非选取 ”贪心"的方向，此时的显式的超线性收敛率会慢于 “贪心” 的算法。后续 Dachao Lin，Haishan Ye，Zhihua Zhang 在文章 [3] 中紧接着类似的思想，证明了如果在正交基中随机选取一个方向，而非 “贪心”地选择方向，算法也具有相同的显式的超线性收敛率。相比于 “贪心”的算法，随机的Quasi-Newton算法，计算代价更低。因为 “贪心” 方向的选取涉及到特征值分解等计算，而随机的Quasi-Newton算法避免了上述计算开销。

上述文章中的分析适用于整个Broyden类算法，并且将对应的超线性收敛率从矩阵近似问题推广到了一大类比光滑强凸函数更广的函数：自和谐函数。但在本次Project中，我将仅仅关注于某一个具体的Broyden类算法。由于在实际中BFGS常被认为是收敛更快的，因此我选取BFGS算法作为研究对象。关于BFGS迭代公式的解释，以及使用线搜索算法下对应的全局收敛率以及局部渐进超线性收敛率的证明，由于其并非本次Project所关注的重点，我也将其放在如下链接中 [4] .

为了简便起见，我也仅仅关于于二次函数的最小化问题。但是已经足以展现相关工作的核心。

在给定一个方向 u 后，对于二次函数极小化问题，BFGS的矩阵迭代格式如下给出，

$$G_{k+1} = G_k - \frac{G_k u u^\top G_k}{u^\top G_k u} + \frac{A u u^\top A}{u^\top A u}.$$

可以证明该迭代格式具有如下的性质：对于某一正数 η , 若 $A \preceq G_k \preceq \eta A$， 那么则有 $A \preceq G_{k+1} \preceq \eta A$. 该性质告诉我们，如果我们对想要近似的矩阵 A 有一个比较好的初始估计范围的话，算法给出的矩阵迭代序列都将在这个范围内。实际上该性质对于整一类Broyden算法都是成立的，我将相关证明其放在如下链接中 [5].

在算法中，如果函数的 μ, L 已知或者对应的下界或者上界已知，我们可以得到关于 A 的范围的初始猜测，那么后续迭代都将在我们控制的界中。更具体地，选定 $G_0 = LI_n$, 那么对产生的序列 G_k 都成立 $A \preceq G_k \preceq \frac{L}{\mu} A$

对于极小化二次函数的收敛率，考虑采用如下的残差衡量，

$$\lambda(x) = \|x - x^*\|_A^2 = \nabla f(x)^\top A^{-1} \nabla f(x).$$

根据

$$\nabla f(x_{k+1}) = \nabla f(x_k) + A(x_{k+1} - x_k) = -(A^{-1} - G_k^{-1}) A \nabla f(x_k)$$

以及

$$(A^{-1} - G_k^{-1}) A (A^{-1} - G_k^{-1}) \leq \left(1 - \frac{\mu}{L}\right)^2 A^{-1}$$

可以看到残差单步下降

$$\lambda(x_{k+1}) = \nabla f(x_k)^\top (A^{-1} - G_k^{-1}) A (A^{-1} - G_k^{-1}) \nabla f(x_k) \leq \left(1 - \frac{\mu}{L}\right)^2 \lambda(x_k).$$

因此算法首先是全局线性收敛的，

$$\lambda(x_{k+1}) \leq \left(1 - \frac{\mu}{L}\right)^{2k} \lambda(x_k).$$

更进一步，为了衡量矩阵近似的超线性收敛率，选取如下指标衡量矩阵近似的程度，

$$\sigma_A(G) = tr(A^{-1}(G - A)).$$

对于 $A \preceq G_k$, 指标 $\sigma_A(G_k)$ 一定是非负的，且当且仅当 $G_k = A$ 时取零。对于BFGS算法可以看到上述指标单步下降，

$$
\begin{aligned}
\sigma_A(G_{k+1}) &= tr(A^{-1}(G_{k+1}-A)) \\
&= tr\left(A^{-1}\left(G_k - \frac{G_k u u^\top G_k}{u^\top G_k u} + \frac{A u u^\top A}{u^\top A u} - A\right)\right) \\
&= tr(A^{-1}(G-A)) - \frac{u^\top G_k A^{-1} G_k u}{u^\top G_k u} + 1 \\
&= \sigma_A(G_k) - \frac{u^\top G_k A^{-1} G_k u}{u^\top G_k u} + 1.
\end{aligned}
$$

Classcical BFGS

本节考虑经典的BFGS算法，迭代格式如下，

$$
\begin{aligned}
x_{k+1} &= x_k - G_k^{-1}\nabla f(x_k) \\
u &= x_{k+1} - x_k \\
G_{k+1} &= G_k - \frac{G_k u u^\top G_k}{u^\top G_k u} + \frac{A u u^\top A}{u^\top A u}.
\end{aligned}
$$

根据

$$
\nabla f(x_{k+1}) = \nabla f(x_k) + A(x_{k+1}-x_k) = -(G_k - A)u.
$$

可以得到残差的递推关系式，可以看到残差每次衰减的系数为如下定义的 θ_k ，

$$
\lambda(x_{k+1}) = u^\top (G_k - A)A^{-1}(G_k - A)u = \frac{u^\top (G_k - A)A^{-1}(G_k - A)u}{u^\top G_k A^{-1} G_k u}\lambda(x_k) \triangleq \theta_k \lambda(x_k).
$$

当 $A \preceq G_k$ ，回顾关于 $\sigma_A(G)$ 的下降量，可以巧妙地建立其和 θ_k 的联系，

$$
\begin{aligned}
\sigma_A(G_{k+1}) &= \sigma_A(G_k) - \frac{u^\top G_k A^{-1} G_k u}{u^\top G_k u} + 1 \\
&= \sigma_A(G_k) - \frac{u^\top G_k (A^{-1} - G_k^{-1}) G_k u}{u^\top G_k u} \\
&\le \sigma_A(G_k) - \frac{u^\top G_k (A^{-1} - G_k^{-1}) G_k u}{u^\top G_k A^{-1} G_k u - u^\top G_k (A^{-1} - G_k^{-1}) G_k u} \\
&\le \sigma_A(G_k) - \frac{u^\top G_k (A^{-1} - G_k^{-1}) G_k u}{u^\top G_k A^{-1} G_k u} \\
&= \sigma_A(G_k) - \frac{u^\top (G_k A^{-1} G_k - G_k) u}{u^\top G_k A^{-1} G_k u} \\
&\le \sigma_A(G_k) - \frac{u^\top (G_k A^{-1} G_k - 2G_k + A) u}{u^\top G_k A^{-1} G_k u} \\
&= \sigma_A(G_k) - \frac{u^\top (G_k - A) A^{-1} (G_k - A) u}{u^\top G_k A^{-1} G_k u} \\
&= \sigma_A(G_k) - \theta_k
\end{aligned}
$$

注意到 $\sigma_A(G_k)$ 关于 θ_k 单步下降，而 $\lambda(x_k)$ 关于 θ_k 单步衰减，利用均值不等式可以得到显式的收敛率，

$$
\lambda(x_{k+1}) \le \prod_{i=0}^{k-1}\theta_i \lambda(x_0) \le \left(\frac{\sum_{i=0}^{k-1}\theta_i}{k}\right)^k \lambda(x_0) \le \left(\frac{\sigma_A(G_0)}{k}\right)^k \lambda(x_0) \le \left(\frac{nL}{k\mu}\right)^k \lambda(x_0)
$$

最后一步代入了 G_0 的选择保证了 $\sigma_A(G_0) \le nL/\mu$.

结合算法的全局线性收敛性，可以导出当 x_{k_0} 进入某个局部使得 $\sigma_A(x_{k_0}) \le 1/2$ 之后，成立

$$\lambda(x_{k_0+k}) \leq \left(\frac{\sigma_A(G_{k_0})}{k}\right)^k \lambda(x_{k_0}) \leq \left(\frac{1}{k}\right)^k \left(\frac{1}{2}\right)^k \left(1-\frac{\mu}{L}\right)^{2k_0} \lambda(x_0).$$

Greedy BFGS

上述的分析中， $\sigma_A(G)$ 的下降速率影响了最终的收敛速率，这启发我们，采用 “贪心”的策略，选择更优的方向 u 使得 $\sigma_A(G_k)$ 下降得更快。对应的"最速"方向可以通过特征值分解得到。考虑如下的 “贪心” BFGS算法。

$$\begin{aligned} x_{k+1} &= x_k - G_k^{-1}\nabla f(x_k) \\ \tilde{u} &= \mathrm{agrmax}_{\mathrm{e_i}}\{\mathrm{e_i^\top G_k^{1/2} A^{-1} G_k^{1/2} e_i}\} \\ u &= G_k^{-1/2}\tilde{u} \\ G_{k+1} &= G_k - \frac{G_k u u^\top G_k}{u^\top G_k u} + \frac{A u u^\top A}{u^\top A u}. \end{aligned}$$

其中 $\mathrm{e_i}$ 表示第 i 个位置为1 ,其他位置为 0 的标准正交基。 经过如上的选取，

$$\begin{aligned} \sigma_A(G_{k+1}) &= \sigma_A(G_k) - \frac{u^\top G_k A^{-1} G_k u}{u^\top G_k u} + 1 \\ &= \sigma_A(G_k) - \frac{\tilde{u}^\top G_k^{1/2} A^{-1} G_k^{1/2} \tilde{u}}{\tilde{u}^\top \tilde{u}} + 1 \\ &= \sigma_A(G_k) - \max_i \left(G_k^{1/2} A^{-1} G_k^{1/2}\right)_{ii} + 1 \\ &\leq \sigma_A(G_k) - \frac{1}{n} tr(G_k^{1/2} A^{-1} G_k^{1/2}) + 1 \\ &= \sigma_A(G_k) - \frac{1}{n} tr(A^{-1} G_k) + 1 \\ &= \left(1 - \frac{1}{n}\right) \sigma_A(G_k) \end{aligned}$$

可以看到此时可以保证 $\sigma_A(G_k)$ 是线性收敛的。进一步利用 $\theta_k \leq \sigma_A(G_k)$ 可以给出 $\lambda(x_k)$ 的收敛估计，

$$\lambda(x_{k+1}) \leq \prod_{i=0}^{k-1} \theta_i \lambda(x_0) \leq \prod_{i=0}^{k-1} \sigma_A(G_i) \lambda(x_0) \leq \left(1-\frac{1}{n}\right)^{k(k-1)/2} \left(\frac{nL}{\mu}\right)^k \lambda(x_0)$$

结合算法的全局线性收敛性，可以导出当 x_{k_0} 进入某个局部使得 $\sigma_A(x_{k_0}) \leq 1/2$ 之后，成立

$$\lambda(x_{k_0+k}) \leq \left(1-\frac{1}{n}\right)^{k(k-1)/2} \left(\frac{1}{2}\right)^k \left(1-\frac{\mu}{L}\right)^{2k_0} \lambda(x_0).$$

如果对比与经典BFGS给出的数列估计可以发现 “贪心”的BFGS算法的收敛速度更快。

Random BFGS

由于 “贪心” 的BFGS每一步迭代需要计算特征值分解，考虑如下代价更低的随机 BFGS算法，

$$
\begin{aligned}
x_{k+1} &= x_k - G_k^{-1}\nabla f(x_k) \\
\tilde{u} &= \mathcal{N}(0, I_n) \\
u &= G_k^{-1/2}\tilde{u} \\
G_{k+1} &= G_k - \frac{G_k u u^\top G_k}{u^\top G_k u} + \frac{A u u^\top A}{u^\top A u}.
\end{aligned}
$$

算法每次只需要从一个标准正态分布中采样作为随机向量，替代了原本“贪心”的方向选择过程。

选取的随机向量其指向任意一个方向的概率相等，因此将其归一化之后服从 n 维单位超球面上的均匀分布，因此

$$
\mathbb{E}\left[\frac{\tilde{u}\tilde{u}^\top}{\tilde{u}^\top\tilde{u}}\right] = \frac{1}{n} I_n
$$

据此可以证明在期望意义下成立如下等式，

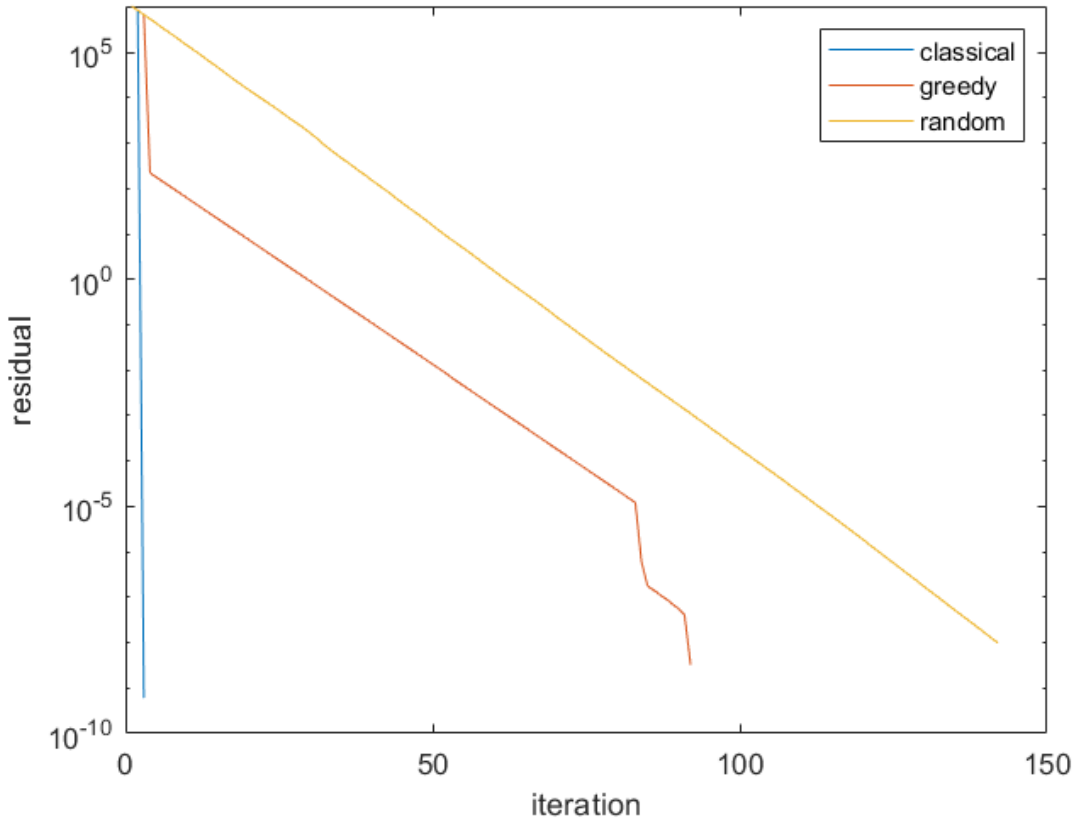
$$
\begin{aligned}
\mathbb{E}[\sigma_A(G_{k+1})] &= \mathbb{E}\left[\sigma_A(G_k) - \frac{u^\top G_k A^{-1} G_k u}{u^\top G_k u} + 1\right] \\
&= \mathbb{E}\left[\sigma_A(G_k) - \frac{\tilde{u}^\top G_k^{1/2} A^{-1} G_k^{1/2}\tilde{u}}{\tilde{u}^\top\tilde{u}} + 1\right] \\
&= \mathbb{E}\left[\sigma_A(G_k) - \frac{1}{n} tr(G_k^{1/2} A^{-1} G_k^{1/2}) + 1\right] \\
&= \sigma_A(G_k) - \frac{1}{n} tr(A^{-1} G_k) + 1 \\
&= \left(1 - \frac{1}{n}\right)\sigma_A(G_k)
\end{aligned}
$$

可以发现随机的BFGS算法得到的收敛率估计与 “贪心”的BFGS算法得到的估计相同，只是不等号变成了等号。因此，尽管随机的BFGS算法可能实际会稍慢一些，但是其期望意义下超线性收敛率的阶却仍然是相同的.

Numerical Experiment

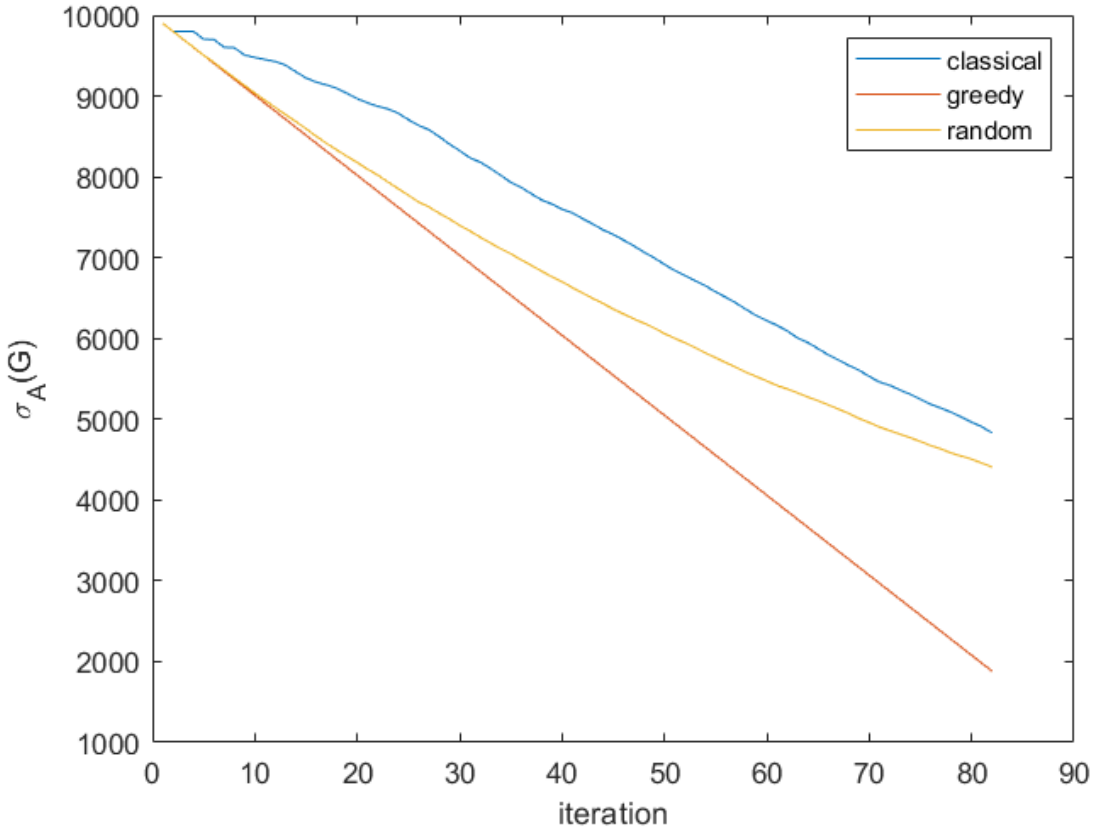
随机生成 $A^{100\times 100}, b^{100\times 1}$， 求解对应的二次函数的极小值，对比三种方法，结果如下

下图中横坐标为迭代次数，纵坐标为残差： $\|x - x^*\|_A$.



但是出乎意料的是，尽管在理论上经典的BFGS方法收敛不如新型的 “贪心”的BFGS方法，但在实际上还是经典的BFGS方法收敛迅速，只能说经典终究是经典。“贪心”的方法由于每次选取一个方向下降，残差会呈现出 “突降”的现象。而随机的BFGS方法收敛不如“贪心”的方法，这与理论结果相吻合。

但是如果单纯从指标 $\sigma_A(G)$ 上对比矩阵近似的程度，“贪心”和随机的方法都会优于BFGS算法。



以上的数值实验结果说明了，如果单纯从矩阵近似的角度看，“贪心”和随机的方法会给出Hessian矩阵更准确的估计。因此从理论上看，“贪心”和随机的BFGS应该更接近于Newton方法，从而有更快的收敛。但为什么实际中残差的下降还是经典的BFGS算法快呢？原因可能是因为选取的 $\sigma_A(G)$ 仅仅是一种度量指标，而非完整反映矩阵近似的程度。其次是经典的BFGS方法中考虑了Secant方程，从而在矩阵的更新过程中考虑了前进方向，对于实际优化问题可能会带来更快的收敛率。因此，我认为值得持续探索的问题包括：是否存在其他的读量矩阵近似程度的指标，并且基于该指标可以设计出新的Quasi-Newton算法？如果不借助于这些工作中采用的指标 $\sigma_A(G)$, 是否可以通过其他的一些手段给出BFGS算法的显式收敛率?

Summary

本文主要讨论了关于经典、“贪心”的、随机的BFGS算法在解正定对称系统中的显式超线性收敛率。从证明中可以发现，指标 $\sigma_A(G)$ 的选取是建立上述超线性收敛率分析的关键。该证明的美中不足之处是需要先获得 A 的范围的的初始估计，这可能在很多问题中未必可以直接获得。或许该范围估计也可以看成某种意义上的 “局部”收敛，当然该 “局部”的范围很宽，因此相应的工作给出的结果仍然是很有意义的。相关的证明思想可能可以启发对应的Quasi-Newton算法的研究，例如在上述工作之后，也有研究者分析了增量式的Quasi-Newton算法的超线性收敛率 [6]，以及将相应的算法推广到Minimax问题的鞍点求解上 [7]

Reference

[1] Greedy quasi-Newton methods with explicit superlinear convergence

[2] Rates of superlinear convergence for classical quasi-Newton methods

[3] Greedy and Random Quasi-Newton Methods with Faster Explicit Superlinear Convergence

[4] https://truenobility303.github.io/BFGS

[5] https://truenobility303.github.io/Greedy-and-Random-Newton

[6] Incremental Greedy BFGS: An Incremental Quasi-Newton Method with Explicit Superlinear Rate

[7] Quasi-Newton Methods for Saddle Point Problems and Beyond