无导数无约束优化软件 NEWUOA 参注

M.J.D. Powell

April 24, 2021

谢鹏程 顾卓然

Contents

随机拐点　初始的设置也要改变　选点重要　ρ_{beg} 前 2n+1 个点要变啦

摘要

本文说明使用 NEWUOA 软件寻找函数 $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$ 的最小值的细节。其中 $F(\underline{x})$ 值在任意 x 值处均可计算得出。NEWUOA 算法是迭代算法，在每次迭代的开始，我们都需要一个二次模型 $Q \approx F$，目的是在信赖域算法中被用来调整变量。当模型 Q 被修正时，新的模型 Q 继续在 m 个点处插值拟合目标函数 F. 文章建议插值条件的数目 m 取 $2n+1$ 比较合适, 剩余的自由度将通过极小化 $\nabla^2 Q$ 的变化量得到控制。在本算法的每次迭代中仅有一个插值点被轮换舍弃。因此，除了少数情况下的原点移动，每次迭代的工作量仅为 $\mathcal{O}(m+n)^2$，这允许 n 值取得很大。考虑到精确性和鲁棒性，在 NEWUOA 的发展中伴随着很多问题，它们包括：二次模型的初始选择；在计算机舍入误差存在的情况下插值条件需要保持的充分线性独立性；以及在更新允许模型 Q 得到快速修正的矩阵时涉及的的稳定性问题。文中也给出了这些问题的细节回答，同时使用该软件针对一些测试问题进行了尝试，为了刻画本软件处理 160 维变量问题的表现和效果，文中对其中 9 个测试问题的数值结果进行了分析和展示。

1 引言

在对目标函数进行优化时，我们需要关注目标函数的曲率，也即二阶导数信息，因此对函数进行二次逼近可以有效获得无约束优化问题迭代算法的快速收敛注：有观点 (孙德锋教授) 认为: 越是高维问题，越是应该使用二阶法。同时，我们知道每个二次模型有 $\frac{1}{2}(n+1)(n+2)$ 个独立参数，而在解决 n 值较大的实际应用问题时，我们需要非常昂贵的计算数量来计算目标函数值。因此本文详细介绍的新算法将尝试利用更少的数据构造满足需要的二次模型。在迭代的开始，模型 $Q(\underline{x}), \underline{x} \in \mathcal{R}^n$, 必须满足个如下的$m$个插值条件

$$Q(\underline{x}_i) = F(\underline{x}_i), i = 1, 2, \cdots, m \tag{1.1}$$

其中 $F(\underline{x}), \underline{x} \in \mathcal{R}^n$ 是目标函数，m 值由用户预先设定，m 个不同的插值点 $\underline{x}_i, i = 1, 2, \cdots, m$ 的位置是计算机自动选取的。为了使方程 (??) 总能为二阶导数矩阵 $\nabla^2 Q$ 提供条件，我们要求：$n + 2 \leq m \leq \frac{(n+1)(n+2)}{2}$。若 $m > \frac{(n+1)(n+2)}{2}$，满足方程 (1.1) 的二次模型是不存在的。本文最后一章将详细说明实际计算时选取 $m = 2n + 1$ 是最明智的。注：绝对明智吗？最明智吗？

文中想介绍的新算法之所以能成功，主要归功于基于“当 F 的一阶导数有效时更新 $\nabla^2 Q$ 的对称 Broyden 方法”的技术有无基于其他思想的方法？([?],195-198 页Find this paper/book?)。设目前有一个旧模型 Q_{old}，同时 Q_{new} 为满足兼容性且预留了一定参数自由度的新模型。本算法通过极小化 $\|\nabla^2 Q_{new} - \nabla^2 Q_{\text{old}}\|_F$ 来填补剩余的自由度，式中脚标 F 代表 Frobenius 范数

$$\|A\|_F = \{\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^2\}^{1/2}, A \in \mathcal{R}^{n\times n} \tag{1.2}$$

新模型 $Q = Q_{new}$ 的条件即是插值条件 (??)。因此 $\nabla^2 Q_{new}$ 是唯一确定的，且 Q_{new} 本身也是唯一的，这是因为点 $\underline{x}_i$ 在进行选择时排除了非零线性多项式 $p(\underline{x}), \underline{x} \in \mathcal{R}^n$ 满足 $p(\underline{x}_i) = 0, i = 1, 2, \cdots, m$ 的可能性。换句话说，算法保证了 $(n+1) \times m$ 矩阵

$$X = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \underline{x}_1 - \underline{x}_0 & \underline{x}_2 - \underline{x}_0 & \cdots & \underline{x}_m - \underline{x}_0 \end{pmatrix} \tag{1.3}$$

的行向量是线性独立的，表达式中 $\underline{x}_0$ 是任意给定向量。

通过考查待优化的目标函数 F 本身就是二次函数的情形，我们可以更直观地了解本文所介绍算法的更新技术部分的优势。根据在当前最新迭代开始时的条件：模型 $Q = Q_{\text{old}}$，算法提出选择一个新的变量向量 $\underline{x}_{new} = \underline{x}_{\text{opt}} + d$，其中 $\underline{x}_{\text{opt}}$ 使得 $F(\underline{x}_{\text{opt}})$ 取到目前所得 F 的计算值中的最小值。

如果差值 $\|F(\underline{x}_{new}) - Q_{\text{old}}(\underline{x}_{new})\|$ 相对较小，那么即使 $\nabla^2 Q \approx \nabla^2 F$ 的近似误差很明显，我们也认为模型 Q 很好地预测了 F 的新值；与此同时，如果差值 $\|F(\underline{x}_{new}) - Q_{\text{old}}(\underline{x}_{new})\|$ 相对较大，那么更新技术通过满足 $Q_{new}(\underline{x}_{new}) = F(\underline{x}_{new})$ 较为明显地提升了模型的准确性，由此可知这是一个双赢情形也可以理解为双保险方案。数值结果显示这些选择保证了由算法生成的变量向量具有非常好的收敛性，虽然二阶导数误差 $\|\nabla^2 Q - \nabla^2 F\|_F$ 通常来说相较于每次出现的 Q 都很大。因此算法似乎自动达到了，在减少对 $Q \approx F$ 做近似时其他特征的关注的同时，保持对所给二次模型变量对应变换有用特征的关注。后来，我们又发现：在实际计算中，计算 F 的运算量仅为n 的一阶量级，这就允许 n 取更大的值。更多关于本算法更新技术的细节分析请见 [?]。

这类结果在 2002 年初被首次发现，并在 [?] 一文进行了介绍。特别地，通过使用最小 Frobenius 范数更新算法，我们可以对具有 160 维变量的无约束无导数优化问题进行高精度求解，值得注意的是：虽然 $n = 160$ 时有 13122 个独立参数，而在求解此问题的实际计算中只需用到 9688 个参数。此后，作者开始研究可被广泛使用的 Fortran 软件。但这一任务一直拖到 2003 年 11 月，而造成拖延的主要原因是计算机计算舍入误差在进行一些复杂的测试问题时，导致了不可接受的精确度损失。经过 18 个月的努力，我们在中国杭州举办的优化方法和软件 10 周年纪念会上的相关进展报告中给出了更加可信的数值结果 [?]。由于随机数值不稳定，作者顶住压力，在论文中详细介绍了得到所给结果的算法细节。当 Q 的变化量由 $(m+n+1) \times (m+n+1)$ 的线性方程系统定义时，精确度的损失发生在运算量在 $\mathscr{O}(m^2)$ 内从 Q_{old} 推导得到 Q_{new} 的 Fortran 程序部分Q1: 为何 H 的 $m \times m$ 子矩阵 ω 的秩在理论上为 $m-n-1$? 。设 W 是这个系统的矩阵，W 的逆矩阵 $H = W^{-1}$ 被直接存储和更新。理论上，矩阵 H 的 $m \times m$ 子矩阵 Ω 的秩仅为 $m-n-1$；然而在实际操作中，这一性质不存在了。现在我们转而通过存储 Ω 的分解式来代替对 Ω 本身的存储。这样一来可以用不被计算机舍入误差干扰的方法计算出矩阵真实的秩以上如何对矩阵秩的数值计算产生了影响。该策略纠正了精确度的不可接受误差 [?], 继而 NEWUOA 算法的新版本可以继续发展和提升。本文的目的就是提供新算法的具体细节和数值结果。

第 2 章给出了 NEWUOA 算法的提纲，但是 m (插值条件的数目) 和模型 Q 的更新方法没有在这一章给出，因此该提纲的大部分内容也可以应用于作者于 2002 年提出的 UOBYQA 方法。UOBYQA 方法的每个二次模型由对目标函数 F 的 $\frac{1}{2}(n+1)(n+2)$ 个插值进行定义。算法中初始插值点的选取和首个二次模型的构造将在本文第 3 章给出。如前文所述，第三章还会给出初始矩阵 H 和子矩阵 Ω 分解式的公式。第 4 章主要介绍了在对插值点的位置进行修正时，模型 Q、矩阵 H 以及子矩阵 Ω 分解式的更新办法。在大部分迭代中，变量的变化量 d 是信赖域子问题

$$\text{Minimize} \quad Q(\underline{x}_{\text{opt}} + \underline{d}), \quad \text{subject to} \quad \|\underline{d}\| \leq \Delta \tag{1.4}$$

的近似解，其中参数 $\Delta > 0$ 可由 Q 得到，第 5 章将主要关注这一问题。

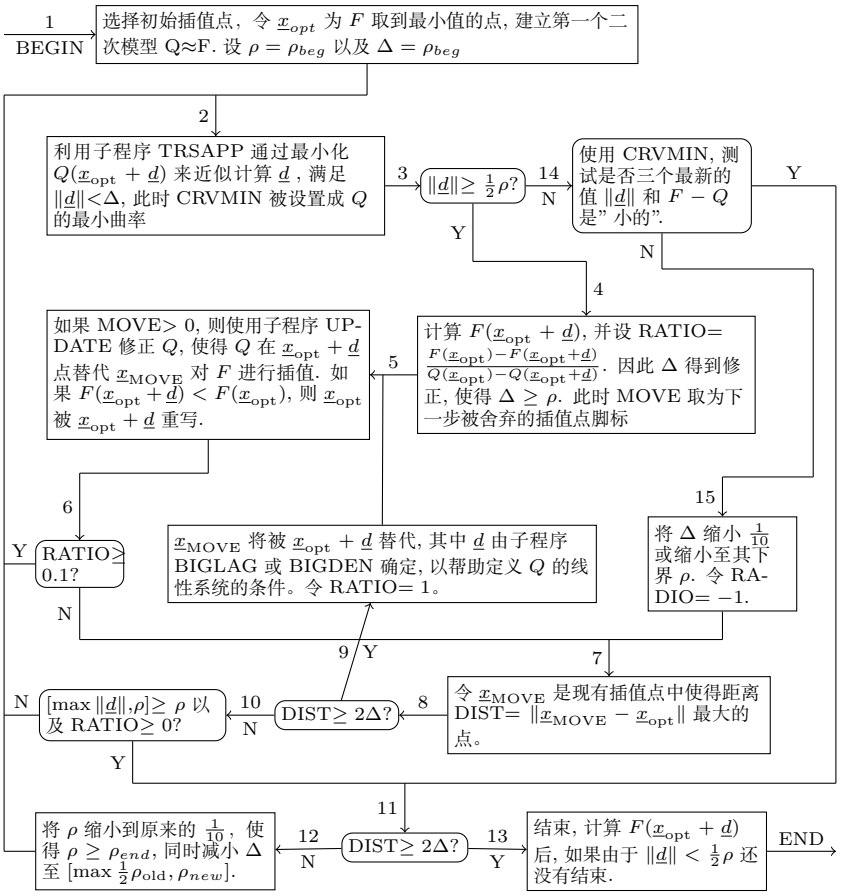
第 6 章的主要内容是选择 $\underline{d}$ 时的备用方法，该方法会在信赖域步没有实现目标函数 F 值的下降时启用。第 7 章将考虑 NEWUOA 算法的其他技术细节，其中包括 $\mathcal{R}^n$ 空间上的基点移动等，该技术对于在修正 H 时避免较大计算误差十分必要。一些数值实验结果和分析将在第 8 章给出。根据第一个实验对二次模型的更新处理，我们受到启发并对 NEWUOA 算法作出相应调整，这些调整将在对其它问题结果进行计算前给出。综合而看，本文所介绍的新算法适合多数无约束优化问题的计算。此外，第 3 章部分所给出的结论的证明将在附录中给出。

2 算法提纲

NEWUOA 软件的用户必须通过使用 Fortran 子程序计算函数 $F(\underline{x})$ 在任意向量变量 $\underline{x} \in \mathcal{R}^n$ 处的函数值这一方式，来对目标函数给出定义。与此同时，我们还需要：一个初始向量 $x_0 \in \mathcal{R}^n$Present a guidance about how to choose the initial vectorx_0，插值条件 (??) 的数目 m，以及信赖域半径的初始值 ρ_{beg} 和终止值 ρ_{end}。如第 1 章所述，m 是取值于如下区间内的一个固定整数

$$n+2 \leq m \leq \frac{(n+1)(n+2)}{2} \tag{2.1}$$

通常来讲，我们认为算法在 $m = 2n+1$ 时求解效率最高Why?。初始插值点 $\underline{x}_i, i = 1, 2, \cdots, m$ 和 $\underline{x}_0$ 满足性质 $\|\underline{x}_i - \underline{x}_0\|_\infty = \rho_{beg}$，这些我们将会在第 3 章具体阐述。$\rho_{beg}$ 值的选择应使得目标函数 F 在这些插值点处的计算所得值可以提供关于点 $\underline{x}_0$ 附近真实目标函数表现的有用信息。尤其是当计算可能包括超出舍入计算误差的假信息时，参数 ρ_{end} 需满足 $\rho_{end} \leq \rho_{beg}$ 且至少与变量最终值所要求精度达



到相同量级。

图 1 即为 NEWUOA 算法导图。模块 1 的具体操作在第 3 章给出，参数 ρ 是被限制在区间 $[\rho_{end},\rho_{beg}]$ 中的信赖域半径 Δ 的下界。Δ 的值在多数迭代中会得到修正，而设立参数 ρ 的目的即是：在对于目标函数 F 的每次计算有明显误差时，保持插值点 $\underline{x}_i, i=1,2,\cdots,m$ 之间的距离充分大，来限制模型 Q 受插值条件影响而产生的计算损失。因此参数 ρ 只在当约束条件 $\Delta\geq\rho$ 将要阻碍目标函数值下降时进行调整。参数 ρ 的每次改变，都是缩小到原来的 $\frac{1}{10}$，直到 $\rho=\rho_{end}$ 时迭代终止，具体可见图 1 中的模块 $11-13$。

图 1 中的模块 $2-6$ 在算法进行了一次信赖域迭代来计算目标函数 F 的新的值后逐步进行，从点 $\underline{x}_{\rm opt}$ 出发的步 $\underline{d}$ 在模块 2 中使用本文第 5 章所介绍的截断共轭梯度法, 求解子问题 (1.4) 得到。若此时 $\|\underline{d}\|\leq\Delta$，则模型 Q 在该程序中沿着每一个搜索方向一定具有正曲率，为了便于模块 14 的使用，CRVMIN 被设置成为这些曲率的最小值，之后将进行详细讨论。

现在算法进行至模块 4，该过程中 Δ 的修正取决于比率

$$\mathrm{RATIO}=\frac{\mathrm{F}(\underline{\mathrm{x}}_{\rm opt})-\mathrm{F}(\underline{\mathrm{x}}_{\rm opt}+\mathrm{d})}{\mathrm{Q}(\underline{\mathrm{x}}_{\rm opt})-\mathrm{Q}(\underline{\mathrm{x}}_{\rm opt}+\mathrm{d})} \tag{2.2}$$

细节详见本文第 7 章。此外，模块 4 的另一任务是选取下一个二次模型 Q 的 m 个插值点。通常情况下，现有点 $\underline{x}_i, i = 1, 2, \cdots, m$ 中会有一个点被替代，同时保留其他点。在这种情况下，模块 4 中的指标MOVE 将被设置为算法舍弃的插值点的脚标。除此之外，只有另外一种可能性，即不改变插值方程，这时 MOVE 值设为 0。MOVE 值的选取细节也将在第 4 章中给出。当实现了严格下降 $F(\underline{x}_{\text{opt}} + d) < F(\underline{x}_{\text{opt}})$ 时，算法强制令 MOVE > 0，其目的是使目前所计算的目标函数 F 的最优值能够被新的插值点集所包括。第 4 章将详细介绍模块 5 的更新操作，若下降率 (2.2) 足够大，则模块 6 直接回到模块 2 进入新的一次信赖域迭代。

当模块 4 所提供的函数变化量 $F(\underline{x}_{\text{opt}} + \underline{d}) - F(\underline{x}_{\text{opt}})$ 同模型变化量 $Q(\underline{x}_{\text{opt}} + \underline{d}) - Q(\underline{x}_{\text{opt}})$ 相比不理想时，进入模块 6 的 N 分支。出现这种情况通常是因为插值条件 (1.1) 中选取的点的位置不利于提供和保持一个好的二次模型，特别是当信赖域迭代造成距离 $\|\underline{x}_i - \underline{x}_{\text{opt}}\|, i = 1, 2, \cdots, m$ 比 Δ 值大很多时。因此模块 7 的作用是将当前插值点 x_{MOVE}(待舍弃) 确定为距离点 $\underline{x}_{\text{opt}}$ 最远的一个插值点。我们认为，若 $\|\underline{x}_{\text{MOVE}} - \underline{x}_{\text{opt}}\| \geq 2\Delta$ 成立，那么通过用 $Q(\underline{x}_{\text{opt}} + \underline{d}) = F(\underline{x}_{\text{opt}} + \underline{d})$ 来替代 $Q(\underline{x}_{\text{MOVE}}) = F(\underline{x}_{\text{MOVE}})$($\underline{d}$ 满足 $\|\underline{d}\| \leq \Delta$)，构建得到的模型 Q 会有显著提升。从图 1 可知 $\underline{d}$ 的实际选择在模块 9 中进行，由于这些选择取决于第 4 章的更新算法，故这方面细节将在随后的第 6 章给出。在计算了 $F(\underline{x}_{\text{opt}} + \underline{d})$ 的最新值之后从模块 9 接到模块 5，目的是同之前一样更新 Q。此时，由于在模块 9 中，人为地将 RATIO 值设为 1，算法将很快接入到模块 6 和模块 2。因此，我们的算法很快将二次模型携带的新信息派上用场。

在当前点的位置有如下性质时，模块 8 接出 N 分支。

$$\|\underline{x}_i - \underline{x}_{\text{opt}}\| < 2\Delta, \quad i = 1, 2, \cdots, m \tag{2.3}$$

继而，模块 10 中的比较测试对选取参数 ρ 当前值的工作是否完成进行了判断。我们看到，该工作完成的唯一标志是 $\|\underline{d}\| > \rho, \Delta > \rho$ 或 RATIO > 0 这三个条件中均有成立。如果前 2 个判断条件成立，那么新一轮的信赖域迭代将仍然使用与之前相同的 ρ，原因是参数 ρ 没有限制住 $\underline{d}$ 的多数选择；在第 3 个条件 RATIO > 0 成立的情况下，可知模块 4 中 $F(\underline{x}_{\text{opt}} + \underline{d}) < F(\underline{x}_{\text{opt}})$，我们倾向于保留旧的参数 ρ。此时，可获得目标函数 F 值的严格下降。因此，理论上算法在固定 ρ 时会产生一个无限循环。然而实际上，计算机计算时的有限精度为函数 F 可能出现的不同值的数量设置了天花板。

最后，我们考虑当模块 2 满足 $\|\underline{d}\| < \frac{1}{2}\rho$ 时图 1 中的操作。这时我们从模块 3

接到模块 14，通常情况下函数 $F(\underline{x}_{\text{opt}}+\underline{d})$ 值不会被计算出来，原因前面提及过，当 $\|\underline{d}\|$ 变小时，目标函数的计算差 $F(\underline{x}_{\text{opt}})-F(\underline{x}_{\text{opt}}+\underline{d})$ 趋向于给出关于真实目标函数的错误信息。

如果模块 14 分流至模块 15，在 $\Delta\geq\rho$ 时，会产生一个巨大的下降。接下来，从模块 7 开始，进行与之前相同的选择：替换掉插值点 $\underline{x}_{\text{MOVE}}$，或用新的参数 Δ 进行一次信赖域迭代，或因现有 ρ 的工作已完成而直接分流至模块 11。与此同时，我们看到模块 14 可以直接分流至模块 11，原因如下：设 $\hat{\underline{x}}_{\text{opt}}$ 和 $\breve{\underline{x}}_{\text{opt}}$ 分别为在当前 ρ 执行期间 $\underline{x}_{\text{opt}}$ 的最初值和最终值。并设 $\hat{\underline{x}}_i, i=1,2,\cdots,m$ 为这部分迭代开始时的插值点。当参数 ρ 比 ρ_{beg} 值小时，当前 ρ 在模块 12 中得到调整，同时，因为它比之前值小很多，我们预期点 $\hat{\underline{x}}_i$ 满足 $\|\hat{\underline{x}}_i-\hat{\underline{x}}_{\text{opt}}\|\geq 2\rho, i\neq\text{opt}$。另一方面，由于图 1 中的模块 7 和模块 8，模块 11 仅在 $\|\hat{\underline{x}}_i-\hat{\underline{x}}_{\text{opt}}\|<2\rho, i=1,2,\cdots,m$ 的情形下，可以从模块 10 分流进入。根据以上这些特点，我们认为至少需要有 $m-1$ 个新的目标函数值可以基于当前的 ρ 值计算而来。然而，让通向模块 11 的流程减少一些工作量，在提高效率方面有着重要意义，尤其是在 m 很大且 ρ_{end} 很小的时候。第 7 章给出了自模块 14 接出的 Y 分支测试细节。这些是基于如下假设：

如果最新迭代的差值 $F(\underline{x}_{\text{opt}}+\underline{d})-Q(\underline{x}_{\text{opt}}+\underline{d})$ 小于当前二阶项 $\frac{1}{8}\rho^2\text{CRVMIN}$，那么没有必要继续改进模型 Q 了。当算法进行到模块 14 的 Y 分支时，我们设向量 $\underline{d}_{\text{old}}$ 为在当前迭代中满足模块 3 中 $\|\underline{d}\|<\frac{1}{2}\rho$ 的向量 d。通常情况下，在变量空间中，$\underline{d}_{\text{old}}$ 是从 $\underline{x}_{\text{opt}}$ 出发所走的试探步中相当好的一个。所以，我们希望离开模块 11 后继续发挥向量 $\underline{d}$ 的作用。如果算法从模块 11 经过模块 12 进行到模块 2，那么由于二次模型和之前相同，我们将重新生成 $\underline{d}=\underline{d}_{\text{old}}$。同时，模块 12 中 Δ 的变化保证了 $\Delta\geq\frac{1}{2}\rho_{end}>\|\underline{d}_{\text{old}}\|$ 的性质。另一个选择是：如果算法进行到模块 14 和模块 11 的 Y 分支，可以看到 $F(\underline{x}_{\text{opt}}+\underline{d}_{\text{old}})$ 在模块 13 中得到计算。同时 NEWUOA 软件将会返回给用户使目标函数达到最小值的首个变量向量。这就完成了整个计算流程。

3 初始计算

我们将在本章详细介绍算法初期的相关计算。我们把首次迭代使用的的二次模型写成如下形式：

$$Q(\underline{x}_0+\underline{d})=Q(\underline{x}_0)+\underline{d}^T\nabla Q(\underline{x}_0)+\frac{1}{2}\underline{d}^T\nabla^2 Q\underline{d},\qquad \underline{d}\in\mathcal{R}^n \tag{3.1}$$

其中 $\underline{x}_0$ 是由 NEWUOA 软件用户提供的初始向量变量为随机——Courant 方法提供支撑。

(1) 当插值条件 (1.1) 的数目满足 $m > 2n+1$ 时，插值点 $x_i, i = 1, 2, \cdots, m$ 中的前 $2n+1$ 个点将按照如下方式进行选取：

$$\underline{x}_1 = \underline{x}_0 \quad \text{以及} \quad \left.\begin{aligned} \underline{x}_{i+1} &= \underline{x}_0 + \rho_{beg}\underline{e}_j \\ \underline{x}_{i+n+1} &= \underline{x}_0 - \rho_{beg}\underline{e}_j \end{aligned}\right\}, \quad i = 1, 2, ..., n \tag{3.2}$$

正如我们在前文所讲过的，其中使用的参数 ρ_{beg} 也是由软件用户提供，另：向量 $\underline{e}_i$ 是空间 $\mathcal{R}^n$ 中的第 i 个单位坐标向量。因此 $Q(x_0)$, $\nabla Q(x_0)$ 以及矩阵 $\nabla^2 Q(x_0)$ 的对角元都可由方程组 (1.1) 的前 $2n+1$ 个方程唯一给定。

(2) 当插值条件的数目 m 满足 $n+2 \le m \le 2n$ 时，初始插值点为向量 (3.2) 的前 m 个向量，于是此时有结论：$Q(\underline{x}_0)$，$\nabla Q(\underline{x}_0)$ 的前 $m-n-1$ 个元素，以及 $(\nabla^2 Q)_{ii}$ 被如前一样地定义。$\nabla^2 Q$ 的其余对角元均设为 0，基于此，$\nabla Q(\underline{x}_0)$ 的其他元素取值为：$\frac{F(\underline{x}_0+\rho_{beg}\underline{e}_i)-F(\underline{x}_0)}{\rho_{beg}}, m-n \le i \le n$。

在 $m > 2n+1$ 的情形，由于矩阵 $\nabla^2 Q$ 是对称的，初始点 $\underline{x}_i, i = 1, 2, \cdots, m$ 的选取使得条件 (1.1) 还提供了 $2(m-2n-1)$ 个 $\nabla^2 Q$ 矩阵的非对角元。特别地，对于 $i \in [2n+2, m]$，点 $\underline{x}_i$ 有如下形式：

$$\underline{x}_i = \underline{x}_0 + \sigma_p \rho_{beg}\underline{e}_p + \sigma_q \rho_{beg}\underline{e}_q \tag{3.3}$$

其中脚标 p 和 q 是在区间 $[1, n]$ 内的不同整数，且 σ_p 和 σ_q 包含在如下定义中：

$$\sigma_j = \begin{cases} -1, & F(\underline{x}_0 - \rho_{beg}\underline{e}_j) < F(\underline{x}_0 + \rho_{beg}\underline{e}_j) \\ +1, & F(\underline{x}_0 - \rho_{beg}\underline{e}_j) \ge F(\underline{x}_0 + \rho_{beg}\underline{e}_j), \end{cases} \tag{3.4}$$

这使得 (3) 的选取向目标函数 F 的更小值偏靠。因此方程 (1.1) 可以确定元素 $\nabla^2 Q_{pq} = \nabla^2 Q_{qp}$，理由是每个二次函数 $Q(\underline{x}), \underline{x} \in \mathcal{R}^n$ 有如下性质：

$$\begin{aligned} \rho_{beg}^{-2}\{Q(\underline{x}_0) - Q(\underline{x}_0 + \sigma_p\rho_{beg}\underline{e}_p) - Q(\underline{x}_0 + \sigma_q\rho_{beg}\underline{e}_q) \\ + Q(\underline{x}_0 + \sigma_p\rho_{beg}\underline{e}_p + \sigma_q\rho_{beg}\underline{e}_q)\} = \sigma_p\sigma_q(\nabla^2 Q)_{pq} \end{aligned} \tag{3.5}$$

简单起见，我们依照如下方法选取公式 (??) 中的脚标 p 和 q。我们设 j 为商 $\frac{i-n-2}{n}$ 的整数部分，由于 $i \ge 2n+2$，故知 $j \ge 1$，我们设 $p = i-n-1-jn$，这样一来 p 在区间 $[1, n]$ 中，同时我们令 q 取值 $p+j$ 或 $p+j-n$，在 $p+j > n$ 时选择

后者对 q 进行赋值。因此，举例而言，当 $n=5, m=20$ 时，共有 9 对 (p,q)，依序为: $\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,1\},\{1,3\},\{2,4\},\{3,5\},\{4,1\}$是否可以动手脚? 。

对于所有没有依照本段所述方法得到赋值的矩阵 $\nabla^2 Q$ 的非对角元，统一赋值为零，这就完成了初始二次模型 (3.1) 的具体赋值工作。

NEWUOA 软件的预备工作还包括对初始矩阵 $H=W^{-1}$ 的设置，其中矩阵 W 在定义二次模型改变量所用方程的线性系统中出现。回顾本文第 1 章所述可知，当二次模型 Q 从 Q_{old} 更新为 $Q_{new}=Q_{\text{old}}+D$ 时，我们构造如下的二次函数 D 使得 $\|\nabla^2 D\|_F$ 在以下约束下最小:

$$D(\underline{x}_i)=F(\underline{x}_i)-Q_{\text{old}}(\underline{x}_i) \tag{3.6}$$

以上约束等价于 $Q_{new}(\underline{x}_i)=F(\underline{x}_i), i=1,2,\cdots,m$，我们看到对于函数 D 的计算是一个二次规划问题，同时我们令 $\lambda_j, j=1,2,\cdots,m$ 作为该问题 KKT 条件的 Lagrange 乘子证明见 Least Frobenius norm updating of quadratic models that satisfy interpolation conditions in section 2(The solution of a variational problem) 。它们具有如下性质:

$$\sum_{j=1}^{m}\lambda_j=0 \quad , \quad \sum_{j=1}^{m}\lambda_j(\underline{x}_j-\underline{x}_0)=0 \tag{3.7}$$

且二次多项式 D 的二阶导数矩阵具备如下形式:

$$\nabla^2 D=\sum_{j=1}^{m}\lambda_j\underline{x}_j\underline{x}_j^T=\sum_{j=1}^{m}\lambda_j(\underline{x}_j-\underline{x}_0)(\underline{x}_j-\underline{x}_0)^T \tag{3.8}$$

[?]，表达 (3.8) 的最后一部分由方程 (3.7) 得到。$\nabla^2 D$ 的这一形式使得 D 可以写为如下函数形式:

$$D(\underline{x})=c+(\underline{x}-\underline{x}_0)^T\underline{g}+\frac{1}{2}\sum_{j=1}^{m}\lambda_j\{(\underline{x}-\underline{x}_0)^T(\underline{x}_j-\underline{x}_0)\}^2, \quad \underline{x}\in\mathcal{R}^n \tag{3.9}$$

同时我们探寻参数 $c\in\mathcal{R}^n$,$\underline{g}\in\mathcal{R}^n$ 以及 $\underline{\lambda}\in\mathcal{R}^m$ 的值。条件 (3) 和 (3.7) 给出下列线性方程组:

$$\left(\begin{array}{c|c} A & X^T \\ \hline X & 0 \end{array}\right)\left(\begin{array}{c} \underline{\lambda} \\ \hline c \end{array}\right)=\left(\begin{array}{c} \underline{r} \\ \hline 0 \end{array}\right) \quad \begin{array}{l} \updownarrow m \\ \updownarrow n+1, \end{array} \tag{3.10}$$

方程中矩阵 A 的元素为

$$A_{ij} = \frac{1}{2}\{(\underline{x}_i - \underline{x}_0)^T(\underline{x}_j - \underline{x}_0)\}^2, \quad 1 \le i, j \le m \tag{3.11}$$

其中 X 是矩阵 (1.3)，且向量 $\underline{r}$ 的元素为 $F(\underline{x}_i) - Q_{\text{old}}(\underline{x}_i), i = 1, 2, \cdots, m$。因此 W 和 H 分别为如下矩阵：

$$W = \left(\begin{array}{c|c} A & X^T \\ \hline X & 0 \end{array}\right), H = W^{-1} = \left(\begin{array}{c|c} \Omega & \Xi^T \\ \hline \Xi & \Upsilon \end{array}\right) \tag{3.12}$$

可以直接从向量 $\underline{x}_i, i = 1, 2, \cdots, m$ 推导出矩阵 W 的元素，但是我们要求 Ξ 和 Υ 的元素同 Ω 的分解式分离开来。幸运的是，已选定的初始插值点的位置为待求项提供了简单方便的公式，下面将给出介绍。这些公式的证明详见本文附录。初始的 $(n+1) \times m$ 矩阵 Ξ 的首行具有非常简单的形式：

$$\Xi_{1j} = \delta_{1j}, \quad j = 1, 2, ..., m. \tag{3.13}$$

除此之外，对于满足 $2 \le i \le \min[n+1, m-n]$ 的 i，矩阵 Ξ 的第 i 行有非零元素

$$\Xi_{ii} = (2\rho_{beg})^{-1} \quad , \quad \Xi_{ii+n} = -(2\rho_{beg})^{-1} \tag{3.14}$$

其余所有元素为零，这样的话就定义好了在 $m \ge 2n+1$ 情形下的初始矩阵 Ξ。否则，当 $m-n+1 \le i \le n+1$ 成立时，初始矩阵 Ξ 的第 i 行也仅含有 2 个非零元素，分别如下：

$$\Xi_{i1} = -(\rho_{beg})^{-1} \quad , \quad \Xi_{ii} = (\rho_{beg})^{-1} \tag{3.15}$$

这就基于给定的这些插值点完成了对矩阵 Ξ 的定义。除此之外，初始的 $(n+1) \times (n+1)$ 矩阵 Υ 相当稀疏，它在 $m \ge 2n+1$ 的情况下恒为零矩阵。如果 $m < 2n+1$，那么它只有最后的 $2n-m+1$ 个对角元非零，这些元素的取值如下：

$$\Upsilon_{ii} = -\frac{1}{2}\rho_{beg}^2, \quad m-n+1 \le i \le n+1. \tag{3.16}$$

如第 1 章所提及的，矩阵 Ω 的分解，保证了矩阵 Ω 的秩最多为 $m-n-1$，其分解形式如下：

$$\Omega = \sum_{k=1}^{m-n-1} s_k \underline{z}_k \underline{z}_k^T = \sum_{k=1}^{m-n-1} \underline{z}_k \underline{z}_k^T = ZZ^T \tag{3.17}$$

上面第 2 个等号成立的原因是每个 s_k 在初始时都被设置为 1。当 $1 \le k \le \min[n, m-n-1]$ 时，初始向量 $\underline{z}_k \in \mathcal{R}^m$ 的元素，也就是矩阵 Z 的第 k 列，被赋值如下：

$$\left.\begin{aligned} Z_{1k} &= -\sqrt{2}\rho_{beg}^{-2}, & Z_{k+1k} &= \frac{1}{2}\sqrt{2}\rho_{beg}^{-2}, \\ Z_{k+n+1k} &= \frac{1}{2}\sqrt{2}\rho_{beg}^{-2}, & Z_{jk} &= 0 \text{ 其他情况} \end{aligned}\right\} \tag{3.18}$$

所以，这里的每一列都只有 3 个非零元素。与此同时，当 $m > 2n+1$ 以及 $n+1 \le k \le m-n-1$ 时，初始的 $\underline{z}_k$ 取决于 (3.3) 所选取的 $\underline{x}_i$，此处 $i = k+n+1$。我们令 p, q, σ_p, σ_q 和之前的选取相同，同时我们定义 $\hat{p}$ 和 $\hat{q}$ 如下：

$$\underline{x}_{\hat{p}} = \underline{x}_0 + \sigma_p \rho_{beg} \underline{e}_p \quad \text{以及} \quad \underline{x}_{\hat{q}} = \underline{x}_0 + \sigma_q \rho_{beg} \underline{e}_q \tag{3.19}$$

由插值点的位置可知脚标 $\hat{p}$ 为 $p+1$ 或者 $p+n+1$，脚标 $\hat{q}$ 为 $q+1$ 或 $q+n+1$。当前情况下，矩阵 Z 的第 k 列有 4 个非零元，$\underline{z}_k$ 初始值赋值如下：

$$\left.\begin{aligned} Z_{1k} &= \rho_{beg}^{-2}, & Z_{\hat{p}k} &= Z_{\hat{q}k} = -\rho_{beg}^{-2}, \\ Z_{k+n+1k} &= \rho_{beg}^{-2}, & Z_{jk} &= 0 \text{ 其他情况} \end{aligned}\right\} \tag{3.20}$$

所有用来计算矩阵 $H = W^{-1}$ 非零元素的公式都可以在 $\mathcal{O}(m)$ 数量级的操作下完成，这得益于初始插值点的简便选择。但是美中不足的是初始阶段设置矩阵 Ξ，Υ 和 Z 零元素的工作量为 $\mathcal{O}(m^2)$。至此，我们就将 NEWUOA 算法的准备工作介绍完毕了。

4 更新程序

在本章，我们关注 NEWUOA 算法在选择插值点集时进行的每次迭代中对二次模型 Q 所做的变换。我们设新选取的插值点的位置如下：

$$\left.\begin{aligned} \underline{x}_t^+ &= \underline{x}_{\text{opt}} + \underline{d} = \underline{x}^+, \\ \underline{x}_i^+ &= \underline{x}_i, i \in \{1, 2, ..., m\} \setminus \{t\}, \end{aligned}\right\} \tag{4.1}$$

在书写时我们用脚标 t 来替代前文的 MOVE, 则这与图 1 所示算法的提纲相吻合，对于新的插值点来说，模型的改变量 $D = Q_{new} - Q_{\text{old}}$ 必须满足条件 (3.6) 的类似条件，且旧模型 Q_{old} 仍然在旧插值点处对目标函数 F 进行插值拟合。因此模型

的改变量 D 是在以下条件的约束下极小化 $\|\nabla^2 D\|_F$ 得到的二次函数。

$$D(\underline{x}_i^+) = \{F(\underline{x}^+) - Q_{\text{old}}(\underline{x}^+)\}\delta_{it}, \quad i = 1, 2, ..., m. \tag{4.2}$$

我们令 W^+ 和 H^+ 分别为下列矩阵

$$W^+ = \left(\begin{array}{c|c} A^+ & (X^+)^T \\ \hline X^+ & 0 \end{array}\right), \quad H^+ = \left(W^+\right)^{-1} = \left(\begin{array}{c|c} \Omega^+ & (\Xi^+)^T \\ \hline \Xi^+ & \Upsilon^+ \end{array}\right) \tag{4.3}$$

其中新矩阵 A^+ 和 X^+ 通过使用将新插值点替换掉旧插值点的方程 (1) 和 (3) 来得到定义。由系统 (3) 的推导和条件 (4) 可知模型的改变量 D 现在是函数

$$D(\underline{x}) = c^+ + (\underline{x} - x_0)^T g^+ + \frac{1}{2}\sum_{j=1}^{m} \lambda_j^+ \{(\underline{x} - \underline{x}_0)^T (\underline{x}_j^+ - \underline{x}_0)\}^2, \qquad \underline{x} \in \mathcal{R}^n \tag{4.4}$$

其参数是向量

$$\begin{pmatrix} \underline{\lambda}^+ \\ c^+ \\ g^+ \end{pmatrix} = \{F(\underline{x}^+) - Q_{\text{old}}(\underline{x}^+)\} H^+ \underline{e}_t, \tag{4.5}$$

其中向量 e_t 现在属于空间 $\mathcal{R}^{m+n+1}$，表达式 (4) 和 (4) 被 NEWUOA 软件用来生成在以下更新公式中使用的函数 D

$$Q_{new}(\underline{x}) = Q_{\text{old}}(\underline{x}) + D(\underline{x}), \quad \underline{x} \in \mathcal{R}^n \tag{4.6}$$

矩阵 $H = W^{-1}$ 在当前迭代中是有效的，子矩阵 Ξ 和 Υ 分别被存储，矩阵 Ω 的分解式 $\sum_{k=1}^{m-n-1} s_k z_k z_k^T$ 也被存储，但是矩阵 H^+ 在方程 (4) 中出现。因此矩阵 Ξ 和 Υ 被表达式 (4) 重写，同时新的分解式

$$\Omega^+ = \sum_{k=1}^{m-n-1} s_k^+ \underline{z}_k^+ (\underline{z}_k^+)^T \tag{4.7}$$

也是要求的。幸运的是，通过从 (4) 式对插值点进行的简单变换获得优势，以上这些任务的工作量仅需 $\mathcal{O}(m^2)$ 次操作。实际上，我们从方程 (4)、(1)、(3) 和 (4) 可推导得：矩阵 W 和 W^+ 的所有不同元素被限制在矩阵的第 t 行和第 t 列。因此矩阵差 $W^+ - W$ 是一个秩为 2 的矩阵，这表明矩阵差 $H^+ - H$ 的秩也为 2。因此，矩阵 Ξ^+、Υ^+ 和分解式 (4) 是通过 Sherman-Morrison 公式的拓展形式构造的，这些细节和相关分析在 [?] 一文中给出。因此接下来在考虑公式 (4) 的应用和实施之

前所展示的过程中只列出主要提纲。鉴于一个从头开始的针对二次模型的变换 (4) 需要 $\mathscr{O}(m^3)$ 数量级的计算机操作量，在 $\mathscr{O}(m^2)$ 数量级操作内完成更新矩阵 H 的任务对于 NEWUOA 软件的求解效率来说十分重要。

理论上，矩阵 H^+ 是有着如下元素的矩阵 W^+ 的逆矩阵：

$$\left.\begin{aligned} W_{it}^+ = W_{ti}^+ = (W^+\underline{e}_t) - i, \quad i = 1, 2, ..., m+n+1 \\ W_{ij}^+ = W_{ij} = Hij^{-1}, \text{其他情况}, \quad 1 \le i, j \le m+n+1. \end{aligned}\right\} \tag{4.8}$$

由这个表达式的右边项可以得到矩阵 H 和 W^+ 的第 t 列向量为新矩阵 H^+ 的推导提供了足够的信息。定义 (1) 和 (3) 表明 $W^+\underline{e}_t$ 元素如下：

$$\left.\begin{aligned} W_{it}^+ = \frac{1}{2}\{(\underline{x}_i^+ - \underline{x}_0)^T(\underline{x}^+ - \underline{x}_0)\}^2, \qquad i = 1, 2, ..., m \\ W_{m+1t}^+ = 1 \quad \text{以及} \quad W_{i+m+1t}^+ = (\underline{x}^+ - \underline{x}_0)_i, \quad i = 1, 2, ..., n \end{aligned}\right\}, \tag{4.9}$$

我们使用记号 $\underline{x}^+$ 替代 $\underline{x}_t^+$，原因是 $\underline{x}^+ = \underline{x}_{\text{opt}} + \underline{d}$ 在脚标 $t = \text{MOVE}$ 在图 1 模块 4 中被选定前就已经有效了。当然，脚标 t 必须使得矩阵 W^+ 非奇异，这在计算矩阵 H^+ 时被除数不为 0 的情况下成立。因此我们应用一个脚标 t 上依赖于新矩阵 H^+ 的计算矩阵 H^+ 的公式。设向量 $\underline{\omega} \in \mathcal{R}^{m+n+1}$ 的元素取值为：

$$\left.\begin{aligned} w_i = \frac{1}{2}\{(\underline{x}_i - \underline{x}_0)^T(\underline{x}^+ - \underline{x}_0)\}^2, \qquad i = 1, 2, ..., m \\ w_{m+1} = 1 \quad \text{以及} \quad w_{i+m+1} = (\underline{x}^+ - \underline{x}_0)_i \quad i = 1, 2, ..., n \end{aligned}\right\} \tag{4.10}$$

因此 $\underline{w}$ 同脚标 t 独立。方程 (4),(4) 和 (4) 表明向量 $W^+\underline{e}_t$ 只有第 t 个元素和向量 $\underline{w}$ 不相同，这允许我们将矩阵 H^+ 写成由矩阵 H、$\underline{w}$ 和向量 $\underline{e}_t$ 组合的形式。特别地，[?] 中推导了公式

$$\begin{aligned} H^+ = H + \sigma^{-1}[&\alpha(\underline{e}_t - H\underline{w})(\underline{e}_t - H\underline{w})^T - \beta H\underline{e}_t\underline{e}_t^T H \\ &+ \tau\{H\underline{e}_t(\underline{e}_t - H\underline{w})^T + (\underline{e}_t - H\underline{w})\underline{e}_t^T H\}] \end{aligned} \tag{4.11}$$

其中参数的表达式为

$$\left.\begin{aligned} \alpha = \underline{e}_t^T H\underline{e}_t, \quad \beta = \frac{1}{2}\|\underline{x}^+ - \underline{x}_0\|^4 - \underline{w}^T H\underline{w}, \\ \tau = \underline{e}_t^T H\underline{w} \quad \text{以及} \quad \sigma = \alpha\beta + \tau^2. \end{aligned}\right\} \tag{4.12}$$

注意，需要 x^+ 和需要 x_t 是不一样的 我们看到向量 $H\underline{w}$ 和 β 可以在脚标 t 被选定之前被计算出来，因此在实际计算中搞清楚记号 σ 对脚标 t 的依赖性成本并不

大，而这是出于确保 $\|\sigma\|$ 充分大的目的。第 7 章将详细介绍脚标 t 的实际选择。

在引入矩阵 Ω 的分解式之前，公式 (4) 在 NEWUOA 软件的早期版本中应用。在矩阵 Ω 的分解式 (3) 中的项 s_k 和 $\underline{z}_k, k = 1, 2, \cdots, m-n-1$，代替原矩阵 Ω 进行存储时，公式中矩阵 H 底部左部的 $(n+1) \times m$ 子矩阵和底部右部的 $(n+1) \times (n+1)$ 子矩阵仍被算法用来分别从矩阵 Ξ 和 Υ 构造新矩阵 Ξ^+ 和 Υ^+，且算法的新版本直接对向量 $H\underline{w}$ 和 $H\underline{e}_t$ 进行计算。

分解子矩阵 Ω 的目的是减少计算矩阵 $W = H^{-1}$ 时的舍入误差。理论上，这在每次迭代的开始阶段存在。然而在数值实验中，数值实验中的巨大误差会影响到矩阵 H 的计算，还会发生一些 H_{ii}，$1 \le i \le m$ 出现负值的情况，尽管我们知道矩阵 Ω 本应该是半正定的。因此，我们思考在矩阵 H 和 W^{-1} 差别很大时如何更新矩阵 H，假设当前迭代的计算是准确的，那么矩阵 H^+ 是元素为表达式 (4) 右端的矩阵的逆矩阵，其具有如下性质：

$$\left.\begin{array}{c} (H^+)^{-1}_{it} = W^+_{it} \quad \text{以及} \quad (H^+)^{-1}_{ti} = W^+_{ti}, \quad i = 1, 2, ..., m+n+1, \\ W^+_{ij} - (H^+)^{-1}_{ij} = W^+_{ij} - H^{-1}_{ij}, \quad \text{其他情况}, \quad 1 \le i, j \le m+n+1. \end{array}\right\} \tag{4.13}$$

换句话说，用矩阵 W^+ 和 H^+ 对矩阵 W 和 H 进行的重写除了让差矩阵 $W - H^{-1}$ 的第 t 行和第 t 列变为 0，其他项没有变化。由此知，当所有当前的插值点都被将来的迭代抛弃时，矩阵 $W - H^{-1}$ 的前 m 行和前 m 列的当前误差会消失。然而，方程 (4) 表明，矩阵 H^{-1} 的右下部分的 $(n+1) \times (n+1)$ 子矩阵的误差仍保存。分解式 (3) 为这种情况提供了完美的补救方法。实际上，如果 H 是任意一个非奇异的形如 (3) 的 $(m+n+1) \times (m+n+1)$ 的矩阵，同时若原矩阵左上方 $m \times m$ 个元素构成的子矩阵 Ω 的秩是 $m-n-1$，那么矩阵 H^{-1} 的右下部 $(n+1) \times (n+1)$ 子矩阵是零矩阵，这一结论可通过将逆矩阵 H^{-1} 的元素用矩阵 H 的余子式除以它的行列式 $detH$ 得到 (见 [?])。因此，分解式 (4) 保证了一个非常好的属性

$$(H^+)^{-1}_{ij} = W^+_{ij} = 0, \quad m+1 \le i, j \le m+n+1 \tag{4.14}$$

即使是在计算机舍入误差存在的情况下，该性质依然得到保证。在 NEWUOA 软件的最新版本中，对矩阵 H 的子矩阵 Ω 分解式的更新依赖于以下事实：

$$s^+_k = s_k \quad \text{且} \quad \underline{z}^+_k = \underline{z}_k, \quad k \in \mathcal{K} \tag{4.15}$$

上面的值满足式 (4)，其中 k 在数域 $\mathcal{K}$ 中当且仅当向量 $\underline{z}_k$ 的第 t 个元素是零。在利用上述事实之前，必要时我们还需针对以下求和项做一个基本变换。

$$\Omega^+ = \sum_{k=1}^{m-n-1} s_k \underline{z}_k \underline{z}_k^T \tag{4.16}$$

这意味着 $\mathcal{K}$ 集中的整数的个数最少为 $m-n-3$。特别地，NEWUOA 软件中有如下标注：当 (4) 中的 $s_i = s_j$ 成立时，方程在 $\underline{z}_i$ 和 $\underline{z}_j$ 分别被以下向量替代时仍然成立。

$$\cos\theta \underline{z}_i + \sin\theta \underline{z}_j \quad \text{和} \quad -\sin\theta \underline{z}_i + \cos\theta \underline{z}_j \tag{4.17}$$

其中 $\theta \in [0, 2\pi]$。θ 的选择允许在脚标 i 和 j 原先都不在集合 $\mathcal{K}$ 中时，将脚标 i 原因在上方或 j 加在 $\mathcal{K}$ 中。因此，因为 $s_k = \pm 1$ 对于每一个 k 均成立，只有一或两个新向量 $z_k^+, k = 1, 2, 3 \cdots m-n-1$ 必须要在保留了值 (4) 后计算。当 $\|\mathcal{K}\| = m-n-2$ 时，我们令 $\underline{z}_{m-n-1}^+$ 作为要求的新向量，这是常见的情形，因为理论上正定的 Ω 应排除掉 s_k 的负值。此时新的分解式 (4) 中最后一项由下式定义：

$$\left.\begin{aligned} s_{m-n-1}^+ &= \mathrm{sign}(\sigma) s_{m-n-1} \\ \underline{z}_{m-n-1}^+ &= \|\sigma\|^{-\frac{1}{2}} \{\tau \underline{z}_{m-n-1} + Z_{tm-n-1} \mathrm{chop}(\underline{e}_t - H\underline{w})\} \end{aligned}\right\}, \tag{4.18}$$

其中参数 τ, σ 和向量 $\underline{e}_t - H\underline{w}$ 仍然根据更新公式 (4.11) 进行赋值，其中 Z_{tm-n-1} 是向量 z_{m-n-1} 的第 t 个元素，且 $\mathrm{chop}(\underline{e}_t - H\underline{w})$ 是空间 $\mathcal{R}^m$ 中的向量，它的元素是向量 $\underline{e}_t - H\underline{w}$ 的前 m 个元素。该部分和下一段的假设、推导在[?]中给出。

另一种情况是 $\|\mathcal{K}\| = m-n-3$, 我们假设此时只有 $\underline{z}_1^+$ 和 $\underline{z}_2^+$ 没有由等式 (4) 提供，这时我们可以对记号进行简化，同时有符号 $s_1 = +1$ 和 $s_2 = -1$。继而我们知道分别被记为 Z_{t1} 和 Z_{t2} 的向量 $\underline{z}_1$ 和 $\underline{z}_2$ 的第 t 个元素是非零的。对于要求的新向量 $\underline{z}_1^+$ 和 $\underline{z}_2^+$，我们有多种供选择的可能，这是受与正交旋转 (4) 相关的自由度的影响，我们对其中的 2 个向量执行有效算法以避免它们被舍去。特别地，如果表达式 (4) 中的参数 β 是非负的，我们定义 $\zeta = \tau^2 + \beta Z_{t1}^2$ 且 NEWUOA 应用以下公式

$$\left.\begin{aligned} s_1^+ &= s_1 = +1, \quad s_2^+ = \mathrm{sign}(\sigma) s_2 = -\mathrm{sign}(\sigma), \\ \underline{z}_1^+ &= \|\zeta\|^{-\frac{1}{2}} \{\tau \underline{z}_1 + Z_{t1} \mathrm{chop}(\underline{e}_t - H\underline{w})\}, \\ \underline{z}_2^+ &= \|\zeta\sigma\|^{-\frac{1}{2}} \{-\beta Z_{t1} Z_{t2} \underline{z}_1 + \zeta \underline{z}_2 + \tau Z_{t2} \mathrm{chop}(\underline{e}_t - H\underline{w})\}. \end{aligned}\right\} \tag{4.19}$$

否则，当 $\beta<0$ 时，我们定义 $\zeta=\tau^2-\beta Z_{t2}^2$，且在 NEWUOA 中设定如下

$$\left.\begin{aligned} &s_1^+=\mathrm{sign}(\sigma)s_1=\mathrm{sign}(\sigma),\quad s_2^+=s_2=-1\\ &\underline{z}_1^+=\|\zeta\sigma\|^{-\frac{1}{2}}\{\zeta\underline{z}_1+\beta Z_{t1}Z_{t2}\underline{z}_1+\tau Z_{t1}\mathrm{chop}(\underline{e}_t-H\underline{w})\},\\ &\underline{z}_2^+=\|\zeta\|^{-\frac{1}{2}}\{\tau\underline{z}_2+Z_{t2}\mathrm{chop}(\underline{e}_t-H\underline{w})\}. \end{aligned}\right\} \tag{4.20}$$

前一段所描述的技术仅在至少有一个记号 $s_k,k=1,2,\cdots,m-n-1$ 取负值时使用，此时在方程 (4) 中，一定会在之前的某一步迭代中，出现参数 $\sigma<0$，原因是每个参数 s_k 在初始时都设为 $+1$。除此之外，在处理条件 $\alpha\geq0$ 和 $\beta\geq0$ 中时的的任何失败都是由于计算机计算舍入误差所引起。因此 [?] 提出可以对公式 (4) 中的参数 σ 做如下赋值

$$\sigma_{new}=\max[0,\alpha]\max[0,\beta]+\tau^2 \tag{4.21}$$

来替代先前的 $\alpha\beta+\tau^2$。然而，若新值同先前的值不同，则新矩阵 (4.11) 可能不满足 (4.8) 中的任一条件，除非分解式 (4.16) 和 (4.7) 保证矩阵 H^{-1} 和 $(H^+)^{-1}$ 的右下子矩阵为零。另一种保持 σ 值为正值的方法是保持 $\alpha=\underline{e}_t^TH\underline{e}_t,\tau=\underline{e}_t^TH\underline{w}$ 从表达式 (4.12)中定义的$\sigma=\alpha\beta+\tau^2$ 并用下式定义 β:

$$\beta_{\mathrm{new}}=\max\left[0,\frac{1}{2}\left\|\underline{x}^+-\underline{x}_0\right\|^4-\underline{w}^TH\underline{w}\right]. \tag{4.22}$$

在这种情况下对于 β 的任何变化都改变了元素 $(H^+)_{tt}^{-1}$，但是所有其它的稳定性性质 (4.13) 都得到了保留，正如在 [?] 的引理 2.3 证明中所述。因此方程 (4.21) 被舍弃，且我们通过数值实验考察了用 (4.22) 所述的赋值来代替 (4.12) 的定义。在数值结果中，仅仅在舍入误差损失巨大时才可发现关键差错，此时复原工作由条件 (4.13) 来提供，该工作十分重要。因此，之前所描述的用来更新矩阵 Ξ，Υ 和 Ω 的分解式的过程是提倡的，虽然实际情况中 α,β,σ 和一些符号 s_k 可能随机地变为负值。这些错误通常可以由 NEWUOA 软件通过几步迭代自动纠正。

除此之外，NEWUOA 在存储和更新矩阵 H 时还具有以下优点：当步 $\underline{d}$ 在图表 1 中由模块 2 得到计算时，模型 Q 的常数项是不相关的。除此之外，旧模型 Q_{old} 的常数项在公式 (4.5) 中不作要求，因为性质 $Q_{\mathrm{old}}(\underline{x}_{\mathrm{opt}})=F(\underline{x}_{\mathrm{opt}})$ 和 $\underline{x}^+=\underline{x}_{\mathrm{opt}}+\underline{d}$

允许方程被写为如下形式:

$$\begin{pmatrix} \underline{\lambda}^+ \\ c^+ \\ \underline{g}^+ \end{pmatrix} = \{[F(\underline{x}_{\text{opt}} + \underline{d}) - F(\underline{x}_{\text{opt}})] - [Q_{\text{old}}(\underline{x}_{\text{opt}} + \underline{d}) - Q_{\text{old}}(\underline{x}_{\text{opt}})]\} H^+ \underline{e}_t. \quad (4.23)$$

因此，NEWUOA 没有存储任何二次模型的常数项。由此可知表达式 (4.23) 中的 c^+ 被忽视了，这导致矩阵 H^+ 的第 $m+1$ 行对于通过公式 (4.6) 来修正 Q 变得没有必要。方程 (4.23) 说明在脚标 t 在区间 $[1, m]$ 中时，矩阵 H^+ 的第 $m+1$ 列也是不必要的。实际上，每个矩阵 H 的第 $m+1$ 行和第 $m+1$ 列的值都被 NEWUOA 的新版本省略，这等价于移除掉每个子矩阵 Ξ 的第一行和每个子矩阵 Υ 的第一行和第一列，但是这些子矩阵的其他元素保留下来了。通常情况下该技术通过将在目标函数 F 在 $x \in \mathcal{R}^n$ 上实际值上的注意转移到发生在目标函数和变量的变化上来获得准确度，正如在例子中方程 (4.23) 右手边所展示的那样。接下来的程序被 NEWUOA 用来在不使用矩阵 H 的第 $m+1$ 行的情况下更新矩阵 H。

设“opt”是在区间 $[1, m]$ 中的整数, 它使得 $i = \text{opt}$ 给出插值点 $\underline{x}_i, i = 1, 2, \cdots, m$ 中的最佳点，这就与第 1 章、第 2 章的记号吻合，同时令向量 $\underline{v} \in \mathcal{R}^{m+n+1}$ 具有以下元素:

$$\left.\begin{aligned} v_i &= \frac{1}{2}\{(\underline{x}_i - \underline{x}_0)^T(\underline{x}_{\text{opt}} - \underline{x}_0)\}^2, && i = 1, 2, ..., m \\ v_{m+1} &= 1 \quad \text{且} \quad v_{i+m+1} = (\underline{x}_{\text{opt}} - \underline{x}_0)_i && i = 1, 2, ..., n \end{aligned}\right\}. \quad (4.24)$$

根据 $H = W^{-1}$ 因此向量 $\underline{v}$ 是矩阵 W 的第 opt 列，故表达式 (3.12) 在理论上表明 $H\underline{v} = \underline{e}_{\text{opt}}$，其中 $_{\text{opt}}$ 是在空间 $\mathcal{R}^{m+n+1}$ 中的第 opt 个坐标向量。因此方程 (4.11) 和 (4.12) 中的 $H\underline{w}$ 和 $\underline{w}^T H \underline{w}$ 按照如下取值:

$$H\underline{w} = H(\underline{w} - \underline{v}) + \underline{e}_{\text{opt}} \quad (4.25)$$

以及

$$\underline{w}^T H \underline{w} = (\underline{w} - \underline{v})^T H (\underline{w} - \underline{v}) + 2\underline{w}^T \underline{e}_{\text{opt}} - \underline{v}^T \underline{e}_{\text{opt}}. \quad (4.26)$$

以上公式允许 (4.12) 中的参数在不需要知道矩阵 H 的第 $m+1$ 行和第 $m+1$ 列的情况下得到计算。原因是 $\underline{w} - \underline{v}$ 的第 $m+1$ 个元素是零。类似地，向量 $H\underline{w}$ 的前 m 个元素和后 n 个元素都可由公式 (4.25) 给出，且向量 $H\underline{e}_t$ 中的的这些元素是已知的。因此表达式 (4.11) 的所有项对于生成矩阵 Ξ^+ 和 Υ^+ 的待求部分是有

效的。除此之外，在构造 $\mathrm{chop}(\underline{e}_t - H\underline{w})$ 之后，就不再改变 Ω 分解式的更新了。[?] 中的引理 3 证明了：当该更新程序的新版本投入应用且矩阵 H 已被计算舍入误差损毁时，此时新的 H^+ 具备类似于条件 (4.13) 的稳定性。

我们看到所给的用于更新矩阵 H 的程序仅需要 $\mathscr{O}(m^2)$ 次计算机操作，这在所建议的 $m = 2n + 1$ 情形下是非常好的。另一方面，多项式函数 (4.4) 具有二阶导数矩阵。

$$\nabla^2 D = \sum_{j=1}^{m} \lambda_j^+ (\underline{x}_j^+ - \underline{x}_0)(\underline{x}_j^+ - \underline{x}_0)^T, \tag{4.27}$$

因此其元素的计算需要花费 $\mathscr{O}(mn^2)$ 次操作。因此 $\nabla^2 Q_{new}$ 是不能直接由公式 (4.6) 推导出来的。反而，正如在 [?] 的第 3 章的最后所建议的那样，NEWUOA 软件使用如下形式：

$$\left.\begin{aligned} \nabla^2 Q_{\mathrm{old}} &= \Gamma + \sum_{j=1}^{m} \gamma_j (\underline{x}_j - \underline{x}_0)(\underline{x}_j - \underline{x}_0)^T \\ \nabla^2 Q_{new} &= \Gamma^+ + \sum_{j=1}^{m} \gamma_j^+ (\underline{x}_j^+ - \underline{x}_0)(\underline{x}_j^+ - \underline{x}_0)^T \end{aligned}\right\}, \tag{4.28}$$

分别用 Γ^+ 和 $\gamma_j^+, j = 1, 2, \cdots m$ 来重写对称矩阵 Γ 和完全系数 $\gamma_j, j = 1, 2, \cdots, m$，在第一次迭代的开始。每个 γ_j 被设置为零，我们设矩阵 Γ 为初始二次模型的二阶导数矩阵，它的元素在第 3 章的前两段被定义。当 (4.6) 所示改变发生在二次模型上时，条件 (4.1)，(4.27) 和 (4.28) 允许选择：

$$\left.\begin{aligned} &\Gamma^+ = \Gamma + \gamma_t (\underline{x}_t - \underline{x}_0)(\underline{x}_t - \underline{x}_0)^T, \quad \Upsilon_t^+ = \lambda_t^+ \\ &\text{以及} \quad \gamma_j^+ = \gamma_j + \lambda_j^+, j \in \{1, 2, ..., m\} \setminus \{t\} \end{aligned}\right\} \tag{4.29}$$

这包括在 NEWUOA 中，原因是以上过程能在仅仅在 $\mathcal{O}(n^2)$ 次操作内完成。最后，二次模型 (3.1) 的梯度通过以下公式得到修正

$$\underline{\nabla} Q_{new}(\underline{x}_0) = \underline{\nabla} Q_{\mathrm{old}}(\underline{x}_0) + \underline{g}^+ \tag{4.30}$$

在具有表达式 (4.4) 和 (4.6) 的坐标中，向量 $\underline{g}^+$ 由方程 (4.23) 决定。除了在第 8 章描述的一些数值实验建议了一个最近的修正，在不需要非必要的常数项 $Q(\underline{x}_0)$ 情况下对于更新模型 Q 的描述是完整的。

5 信赖域子问题

我们从流程图图 1 中可以回忆得知子程序 TRSAPP 所生成的从 $\underline{x}_{\text{opt}}$ 出发的步 d 是如下子问题的近似解。

$$\text{Minimize} \quad Q(\underline{x}_{\text{opt}} + \underline{d}), \quad \text{subject to } \|\underline{d}\| \leq \Delta \tag{5.1}$$

接下来我们将详细解释求解该子程序所用的方法。图 1 表明信赖域半径 Δ 和二次模型 Q 是有效的，当信赖域求解子程序被启用时，正如第 4 章最后所提及的那样，二阶导数矩阵 $\nabla^2 Q$ 按照如下形式存储:

$$\nabla^2 Q = \Gamma + \sum_{k=1}^{m} \gamma_k (\underline{x}_k - \underline{x}_0)(\underline{x}_k - \underline{x}_0)^T, \tag{5.2}$$

原因是分别处理 $\nabla^2 Q$ 的所有元素可能会太繁重，尤其是在 n 值非常大时。

$$\nabla^2 Q \underline{u} = \Gamma \underline{u} + \sum_{k=1}^{m} \eta_k (\underline{x}_k - \underline{x}_0), \tag{5.3}$$

其中 $\eta_k = \gamma_k (\underline{x}_k - \underline{x}_0)^T \underline{u}, k = 1, 2, \cdots, m$, 且向量 $\underline{u}$ 是 $\mathcal{R}^n$ 空间中的一个生成向量。因此乘积 $\nabla^2 Q \underline{u}$ 对于任意的向量 $\underline{u}$ 都可以在 $\mathscr{O}(mn)$ 次操作内得到计算，此时用截断共轭梯度法的某一版本来生成 $\underline{d}$ 是合适的。(见 [**?**])。

该方法产生一个在 $\mathcal{R}^n$ 空间中的分段线性路径，起始于 $\underline{x}_{\text{opt}} = \underline{x}_{\text{opt}} + \underline{d}_0$, 其中 $\underline{d}_0 = 0$ 对于 $j \geq 1$, 我们设 $\underline{x}_{\text{opt}} + \underline{d}_j$ 为第 j 条分段的末段的终点，它具有如下形式:

$$\underline{x}_{\text{opt}} + \underline{d}_j = \underline{x}_{\text{opt}} + \underline{d}_{j-1} + \alpha_j \underline{s}_j, \quad j \geq 1 \tag{5.4}$$

其中 $\underline{s}_j$ 是分段的方向，α_j 目前是步长。我们没有给出任何预条件，原因是表达式 (5.1) 边界的范数 $\|\underline{d}\| \leq \Delta$ 是欧氏的。除此之外，在以下情况路径会在 $\underline{x}_{\text{opt}} + \underline{d}_{j-1}$ 处被截断:

1. $\|\underline{\nabla} Q(\underline{x}_{\text{opt}} + \underline{d}_{j-1})\|$ 充分小
2. $\|\underline{d}_{j-1}\| = \Delta$ 成立
3. 达到一些其它测试要求 (接下来将具体阐述)。

完整的路径具有如下性质：如果某点从点 $\underline{x}_{\text{opt}}$ 沿该路径移动，则自空间 $\mathcal{R}^n$ 中的点 $\underline{x}_{\text{opt}}$ 出发的欧氏距离单调增加。当路径的第 j 个线性分割段构造完成后，其

方向由以下公式定义：

$$\underline{s}_j = \begin{cases} -\underline{\nabla} Q(\underline{x}_{\rm opt}), & j = 1, \\ -\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1}) + \beta_j \underline{s}_{j-1}, & j \geq 2, \end{cases} \tag{5.5}$$

其中 β_j 是比率 $\|\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1})\|^2)/\|\underline{\nabla} Q\underline{x}_{\rm opt} + \underline{d}_{j-2}\|^2$，这一简便值是我们直接从 Fletcher 和 Reeves([?]) 那里拿来用的。继而，方程 (5.4) 的步长 α_j 被选中来。在 $\alpha_j \geq 0$ 及 $\|\underline{d}_j\| \leq \Delta$(对于每个 j) 的条件下最小化 $Q(\underline{x}_{\rm opt} + \underline{d}_j)$。公式 (5.5) 提供了著名的下降条件：

$$\underline{s}_j^T \underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1}) = -\|\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1})\|^2 < 0, \quad j \geq 1 \tag{5.6}$$

它取决于 α_{j-1} 在 $j \geq 2$ 时的选择。可以由 $\|\underline{d}_{j-1}\| < \Delta$ 得出 α_j 是正值的。乘积 $\nabla^2 Q\underline{u}$ 的形式 (5.3) 促进了梯度 $\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_j)$，$j \geq 0$ 和步长 $\alpha_j, j \geq 1$ 的计算。初始变量 $\underline{u}$ 是差 $\underline{x}_{\rm opt} - \underline{x}_0$, 为了从表达式 (3.1) 中获得梯度。

$$\underline{\nabla} Q(\underline{x}_{\rm opt}) = \nabla Q(\underline{x}_0) + \nabla^2 Q\{\underline{x}_{\rm opt} - \underline{x}_0\} \tag{5.7}$$

向量 $\underline{u}$ 的另一种选择就是出现的整个向量 (5.5)。$\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1})$ 和 $\nabla^2 Q\underline{s}_j$ 的有效性允许 α_j 轻松地被找到，因为正是 α 在区间 $[o, \hat{\alpha}_j]$ 中的值使得下面函数取到极小值。

$$Q(\underline{x}_{\rm opt} + \underline{d}_{j-1} + \alpha\underline{s}_j) = Q(\underline{x}_{\rm opt} + \underline{d}_{j-1}) + \alpha\underline{s}_j^T \underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1}) + \frac{1}{2}\alpha^2 \underline{s}_j^T \nabla^2 Q\underline{s}_j \tag{5.8}$$

其中 $\hat{\alpha}_j$ 是方程 $\|\underline{x}_{\rm opt} + \underline{d}_{j-1} + \hat{\alpha}_j\underline{s}_j\| = \Delta$ 的正根程序中有体现，因此我们尝试探寻 $Q(\underline{x}_{\rm opt} + \underline{d}_{j-1} + \alpha_j\underline{s}_j), 0 \leq \alpha_j \leq \hat{\alpha}_j$ 是否单调减少。方程 (5.6)、(5.8) 表明上面问题的答案在以下情形中是肯定的。

$$-\|\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1})\|^2 + \hat{\alpha}_j \underline{s}_j^T \nabla^2 Q\underline{s}_j \leq 0 \tag{5.9}$$

求导 且继而 $\alpha_j = \hat{\alpha}_j$ 被选中。否则，$\underline{s}_j^T \nabla^2 Q\underline{s}_j$ 是正值，且子程序选择如下值：

$$\alpha_j = \|\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1})\|^2 / \underline{s}_j^T \nabla^2 Q\underline{s}_j < \hat{\alpha}_j \tag{5.10}$$

在寻找到 α_j 之后，梯度 $\nabla Q(\underline{x}_{\rm opt}+\underline{d}_j)$ 由一下公式构造

$$\nabla Q(\underline{x}_{\rm opt}+\underline{d}_j)=\nabla Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})+a_j\nabla^2 Q\underline{s}_j \tag{5.11}$$

该公式由关系式 (5.4) 推导而来，乘积 $\nabla^2 Q\underline{s}_j$ 再次得到使用。本段所述的技术由路径的每个线分割段应用。

在 $\alpha_j=\hat{\alpha}_j$ 的的情形下，路径在 $\underline{x}_{\rm opt}+\underline{d}$ 处截断，原因是那时步 $\underline{d}=\underline{d}_j$ 是在信赖域 $\|\underline{d}\|\le\Delta$ 的边界上。除此之外, 当初始梯度 $\nabla Q(\underline{x}_{\rm opt})$ 等于 0 时（此为特殊情况），路径在起始点 $\underline{x}_{\rm opt}+\underline{d}_0=\underline{x}_{opt}$ 处被截断。否则，我们在当比率

$$[Q(\underline{x}_{\rm opt})-Q(\underline{x}_{\rm opt}+\underline{d}_j)]/[Q(\underline{x}_{\rm opt})-\min\{Q(\underline{x}_{\rm opt}+\underline{d}):\|\underline{d}\|\le\nabla\}] \tag{5.12}$$

充分接近于 1 时截断路径，目的是为了避免共轭梯度迭代仅仅轻微地增进了目标函数的减少，且这一减少还在二次模型的预测之中。而这一目标的实现办法完全是根据经验的如何根据经验。特别地，迭代在至少出现以下一种情形后终止:

$$\left.\begin{aligned}\|\nabla Q(\underline{x}_{\rm opt}+\underline{d}_j)\|&\le 10^{-2}\|\nabla Q(\underline{x}_{\rm opt})\|\\ [Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})-Q(\underline{x}_{\rm opt}+\underline{d}_j)]&\le 10^{-2}[Q(\underline{x}_{\rm opt]}-Q(\underline{x}_{\rm opt}+\underline{d}_j)]\end{aligned}\right\} \tag{5.13}$$

二次模型 Q 对于每个线分割段的变化由表达式 (5.8) 给出，且 $Q(\underline{x}_{\rm opt})-Q(\underline{x}_{\rm opt}+\underline{d}_j)$ 是到目前为止所有改变量的总和。同样，当指标 j 达到迭代次数的理论上界时路径也会被截断。我们设迭代次数为 n, 但是我们期望这个测试对于 $n\ge 10$ 的情形是多余的。

设 $\underline{x}_{\rm opt}+\underline{d}$ 为路径的终点。步 $\underline{d}=\underline{d}_j$ 由子程序 TRSAPP 返回得到，原因是那时共轭梯度迭代就不受信赖域边界的干扰了。除此之外，在图 1 模块 2 所介绍过的参数 CRVMIN 将被赋值为

$$\mathrm{CRVMIN}=\min\{\underline{\mathrm{s}}_\mathrm{i}^{\mathrm{T}}\nabla^2\mathrm{Q}\underline{\mathrm{s}}_\mathrm{i}/\|\underline{\mathrm{s}}_\mathrm{i}\|^2:\mathrm{i}=1,2,\cdots,\mathrm{j}\}. \tag{5.14}$$

否则，CRVMIN 将被设置为 0，且由于比率 (5.12) 可能明显比 1 小，下面的迭代程序将会得到应用。该程序同样由 $\underline{d}_{j-1}$ 计算 $\underline{d}_j$ 且初始点 $\underline{x}_{\rm opt}+\underline{d}_{j-1}$ 是一段截断分段线性路径的终点，因此 $\nabla Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ 是有效的。条件 $\|\underline{d}_j\|=\|\underline{d}_{j-1}\|=\Delta$ 将会在附加程序的每一次迭代中成立。我们决定在每一次迭代的开始仅仅使用步 $\underline{d}_{j-1}$ 和 $\nabla Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})$，不论 $\underline{d}=\underline{d}_{j-1}$ 是否可作为子问题 (5.1) 的一个近似解而被接受。若 $\underline{d}_{j-1}$ 是真的解，那么，根据子问题的 KKT 条件，$\nabla Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ 会

是 $\underline{d}_{j-1}$ 的非正数倍，且我们同样将注意力放在条件 (5.13) 的第 1 行中。实际上，在如下不等式中的一个或全部都达到时，子程序 TRSAPP 选取步 $\underline{d}=\underline{d}_j$

$$\left.\begin{aligned}\|\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})\| &\le 10^{-2}\|\underline{\nabla}Q(\underline{x}_{\rm opt})\|\\ \underline{d}_{j-1}^T\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1}) &\le -0.99\|\underline{d}_{j-1}\|\|\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})\|\end{aligned}\right\} \tag{5.15}$$

否则，$\underline{d}_{j-1}$ 和 $\nabla Q(\underline{x}_{\rm opt}+\underline{d})$ 跨占 $\mathcal{R}^n$ 的二维子空间，且 $\underline{d}_j$ 将被计算成为使得 $Q(x_{\rm opt}+d_{j-1})$ 在 $\|\underline{d}_j\|=\Delta$ 的限制下达到极小值的向量。因此 $\underline{d}_j$ 有如下形式：

$$\underline{d}_j=\underline{d}(\theta)=\cos\theta\underline{d}_{j-1}+\sin\theta\underline{s}_j,\qquad \theta\in[0,2\pi], \tag{5.16}$$

其中现在的搜索方向 $\underline{s}_j$ 被选定为具有如下性质的二维子空间中的一个向量。

$$\underline{s}_j^T\underline{d}_{j-1}=0\qquad,\qquad \|\underline{s}_j\|=\Delta \tag{5.17}$$

方程 (5.16) 说明 $Q(\underline{x}_{\rm opt}+\underline{d}(\theta))$ 可以表示为：

$$\begin{aligned}&Q(\underline{x}_{\rm opt})+(\cos\theta\underline{d}_{j-1}+\sin\theta\underline{s}_j)^T\underline{\nabla}Q(\underline{x}_{\rm opt}+(\frac{1}{2}\cos^2\theta\underline{d}_{j-1}+\cos\theta\sin\theta\underline{s}_j)^T\\ &\{\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})-\underline{\nabla}Q(\underline{x}_{\rm opt})\}+\frac{1}{2}\sin^2\theta\underline{s}_j^t\nabla^2Q\underline{s}_j,\qquad 0\le\theta\le 2\pi,\end{aligned} \tag{5.18}$$

原因是大括号中所表示的是乘积 $\nabla^2Q\underline{d}_{j-1}$。$\nabla^2Q\underline{s}_j$ 又是由公式 (5.3) 构造出来的，在这之后，最小化函数 (5.18) 仅需用 $\mathscr{O}(n)$ 次操作Why?。因此步 $\underline{d}_j$ 被确定了，且子程序给出返回值 $\underline{d}=\underline{d}_j$ 在条件 (5.13) 的第 2 项成立时，或者当脚标 j 至少为 n 时。另一种选择是 $\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_j)$ 通过应用关于方程 (5.16) 给出如下梯度的相关接邻来使用下一次迭代得到计算。

$$\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_j)=(1-\cos\theta)\underline{\nabla}Q(\underline{x}_{\rm opt})+\cos\theta\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})+\sin\theta\nabla^2Q\underline{s}_j \tag{5.19}$$

此时脚标 j 的值将逐个增加以使本段程序继续迭代。

6 子程序 BIGLAG 和 BIGDEN

我们回忆第 2 章可知：若达到了图 1 的模块 9，则脚标条件为 $i=$ MOVE 的条件 (1.1)，将由插值条件 $Q(\underline{x}_{\rm opt}+\underline{d})=F(\underline{x}_{\rm opt}+\underline{d})$ 所代替，其中，$\underline{d}$ 由本章提出算法进行计算。理论上，给定脚标 MOVE，d 的选择可由目前插值点 $\underline{x}_i, i>m$ 的位置推导得来，但是在实际操作中，它还取决于被存储和更新的矩阵，即表达

式 (3.12) 中子矩阵 Ξ 和 Υ 和分解式 (4.16)。在书写时，我们使用脚标 t 来代替 MOVE, 这样做的目的是保留第 4 章中的记号，特别地，方程 (4.1) 展示了插值点的新位置。

现有插值点的第 t 个 Lagrange 方程是重要的。满足以下 Lagrange 条件的是二次多项式 $l_t(\underline{x}), \underline{x} \in \mathcal{R}^n$

$$l_t(\underline{x}_i) = \delta_{it} \qquad i = 1, 2, \cdots, m \tag{6.1}$$

在通常出现的 $m < \frac{1}{2}(n+1)(n+2)$ 情况下，剩余的自由度通过极小化 Frobenius 范数 $\|\nabla^2 l_t\|_F$ 来占位，因此，l_t 是如下函数

$$l_t(\underline{x}) = c + (\underline{x} - \underline{x}_0)^T \underline{g} + \frac{1}{2}\sum_{k=1}^{m} \lambda_k \{(\underline{x} - \underline{x}_0)^T (\underline{x}_k - \underline{x}_0)\}^2, \qquad \underline{x} \in \mathcal{R}^n, \tag{6.2}$$

参数 $c, \underline{g}$ 和 λ_k 是由线性系统方程 (3.10) 定义的，其中等式右边现在是坐标向量 $\underline{e}_t \in \mathcal{R}^{m+n+1}$, 因此，这些参数是表达式 (3.12) 表示的矩阵 H 的第 t 列的元素。对于任意 $\underline{x} \in \mathcal{R}^n$, 我们设 $\underline{w}(\underline{x})$ 是在空间 $\mathcal{R}^{m+n+1}$ 中的具有下面元素的向量:

$$\left.\begin{array}{ll} w(\underline{x})_k = \frac{1}{2}\{(\underline{x} - \underline{x}_0)^T (\underline{x}_k - \underline{x}_0)\}^2, & k = 1, 2, \cdots, m \\ w(\underline{x})_{m+1} = 1 \quad \text{以及} \quad w(\underline{x})_{i+m+1} = (\underline{x} - \underline{x}_0)_i, & i = 1, 2, \cdots, m \end{array}\right\}. \tag{6.3}$$

由此可知表达式 (6.2) 可写为如下形式

$$l_t(\underline{x}) = \sum_{k=1}^{m} \lambda_k w(\underline{x})_k + c w(\underline{x})_{m+1} + \sum_{i=1}^{n} g_i w(\underline{x})_{i+m+1} = (H\underline{e}_t)^t \underline{w}(\underline{x}) \tag{6.4}$$

因此当对称矩阵 H 由公式 (4.11) 更新时，针对插值点的变换 (4.1)，表达式 (4.12) 包括了下面值

$$\tau = \underline{e}_t^T H \underline{w} = \underline{e}_t^T H \underline{w}(\underline{x}^+) = (H\underline{e}_t)^T \underline{w}(\underline{x}_{\text{opt}} + \underline{d}) = l_t(\underline{x}_{\text{opt}} + \underline{d}). \tag{6.5}$$

因此 Lagrange 函数 (6.2) 赋予了 τ 关于 $\underline{d}$ 的选择的依赖性，如第 4 章所述，我们期待在应用公式 (4.11) 时，一个相对较大的 $\sigma = \alpha\beta + \tau^2$ 的系数是我们所期望的，通常情况下，$\sigma > \tau^2$ 在实际中是成立的，原因是在理论上 α 和 β 都是正值的。因此我们由前段叙述中可以推知令 $\underline{d}$ 是以下子问题的近似解是有帮助的。这里是否

有可能细化

$$\text{Maximize} \qquad \|l_t(\underline{x}_{\text{opt}}+\underline{d})\| \qquad \text{subject to} \qquad \|\underline{d}\| \le \bar{\Delta}, \tag{6.6}$$

其中 $\Delta > 0$，该计算由子程序 BIGLAG 进行，具体细节将在下一段中给出。在 $m = \frac{1}{2}(n+1)(n+2)$ 的情况下，有一个对于较大值 $|l_t(\bar{\Delta}x_{\text{opt}} + \bar{\Delta}d)|$ 的极其好的理由。特别地，操作者为二次多项式空间选择一个便利的基，目的是使由插值条件 (1.1) 得来的二次模型 Q 的结构趋向于线性方程 $m \times m$ 系统的解。设 B 和 B^+ 分别是插值点发生了变换 (4.1) 的系统的旧矩阵和新矩阵。那时，正如 Powell 在 2001 年所写文章所展现得那样，在 $\bar{\Delta}d \in \mathcal{R}^n$ 上的变化率 $\det B^+/\det B$关于 $d \in \mathcal{R}^n$的仅仅是二次多项式，它正是 $l_t(\underline{x}_{\text{opt}}+\underline{d}), \underline{d} \in \mathcal{R}^n$, 原因正是因为 Lagrange 插值条件，因此(6)，在这种情况下，子问题 (6.6) 对于非奇异化 B^+ 的过程是高度适合的。

BIGLAG 方法是迭代方法而且它就像第 5 章最后一段所述的过程。正如在方程 (5.16) 中所述的那样，第 j 次迭代生成了如下的向量:

$$\underline{d}_j = \underline{d}(\theta) = \cos\theta \underline{d}_{j-1} + \sin\theta \underline{s}_j, \tag{6.7}$$

其中 $\underline{d}_{j-1}$ 是在当前迭代开始时的最佳估计。其中的 $\underline{d}_{j-1}$ 和 $\underline{s}_j$ 有如下性质:

$$\|\underline{d}_{j-1}\| = \|\underline{s}_j\| = \bar{\Delta} \qquad \text{以及} \qquad \underline{s}_t^T \underline{d}_{j-1} = 0 \tag{6.8}$$

因为若 $\|\underline{d}_{j-1}\| = \bar{\Delta}$，则 $\|\underline{s}_j\|$ 一定为 $\bar{\Delta}$(算下平方) 且其中方程 (6.7) 的角度 θ 在求解 $|l_t(\underline{x}_{\text{opt}}+\underline{d}_j)|$ 的极大值的过程得到计算。以下选择

$$\underline{d}_0 = \pm\bar{\Delta}(\underline{x}_t - \underline{x}_{\text{opt}})/\|\underline{x}_t - \underline{x}_{\text{opt}}\| \tag{6.9}$$

由第一次迭代创造，同时还提供了 $|l_t(\underline{x}_{\text{opt}}+\underline{d}_0)|$ 的更大值。也可以知道 $\nabla l_t(\underline{x}_{\text{opt}}+\underline{d}_0)$，原因是 l_t 是满足 Lagrange 条件，$l_t(\underline{x}_{\text{opt}}) = 0$ 且 $l_t(\underline{x}_t) = 1$ 的二次多项式。第一次迭代中的向量 $\underline{s}_1$ 在 $\underline{d}_0$ 和 $\nabla l_t(\underline{x}_{\text{opt}})$ 张成的二维子空间中选取，由此下面两个不等式成立

$$\left.\begin{aligned} \|\underline{d}_0^T \nabla l_t(\underline{x}_{\text{opt}})\|^2 &\le 0.99\bar{\Delta}^2 \|\nabla l_t(\underline{x}_{\text{opt}})\|^2 \\ \text{以及} \qquad \|\nabla l_t(\underline{x}_{\text{opt}})\| &\ge 0.1\|l_t(\underline{x}_{\text{opt}}+\underline{d}_0)\|/\bar{\Delta} \end{aligned}\right\} \tag{6.10}$$

原因是在子空间退化或者在如下等式的一阶项的边界, $\bar{\Delta}\|\underline{\nabla} l_t(\underline{x}_{\rm opt})\|$ 是比不过 $\|l_t(\underline{x}_{\rm opt}+\underline{d}_0)\|$ 的$\underline{\nabla} l_t(\underline{x}_{\rm opt})$的这个用法是没有吸引力的。

$$l_t(\underline{x}_{\rm opt}+\underline{d})=\underline{d}^T\underline{\nabla} l_t(\underline{x}_{\rm opt})+\frac{1}{2}\underline{d}^T\underline{\nabla}^2 l_t\underline{d},\qquad \|\underline{d}\|\le\bar{\Delta} \tag{6.11}$$

或者说，若至少6中的一种情况失败了, 则 $\underline{s}_1$ 可由 $\underline{s}_j, j\ge 2$的计算方法给出定义。特别地，$\underline{s}_j$ 是 $\underline{d}_{j-1}$ 和 $\underline{\nabla} l_t(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ 的一个线性组合，且具有性质 (6.8)，除了在如下不太可能的的情形下子程序返回值为向量 $\underline{d}=\underline{d}_{j-1}$

$$|\underline{d}_{j-1}^T\underline{\nabla} l_t(\underline{x}_{\rm opt}+\underline{d}_{j-1})|^2\ge(1-10^{-8})\,\bar{\Delta}^2\|\underline{\nabla} l_t(\underline{x}_{\rm opt}+\underline{d}_{j-1})\|^2. \tag{6.12}$$

通常情况下，在达到以下条件后终止

$$\|l_t(\underline{x}_{\rm opt}+\underline{d}_j)\|\le 1.1\|l_t(\underline{x}_{\rm opt}+\underline{d}_{j-1})\| \tag{6.13}$$

原因是此时迭代并没有改进子问题 (6.6) 的目标函数太多的问题。那么得到了返回值 $\underline{d}=\underline{d}_j$, 这在 j 达到值 n 的条件下也会发生。否则，如在方程 (5.19) 中所述，我们计算梯度

$$\underline{\nabla} l_t(\underline{x}_{\rm opt}+\underline{d}_j)=(1-\cos\theta)\underline{\nabla} l_t(\underline{x}_{\rm opt})+\cos\theta\nabla l_t(\underline{x}_{\rm opt}+\underline{d}_{j-1})+\sin\theta\nabla^2 l_t\underline{s}_j \tag{6.14}$$

且此时在接下来的迭代中 j 值增加了。因此函数 (6.2) 的二阶矩阵不是独立建造的。在第 5 章中的关于二阶导数矩阵 $\nabla^2 Q$ 的评述也适用于 $\nabla^2 l_t$, 包括下面公式:

$$\nabla^2 l_t\underline{u}=\{\sum_{k=1}^{m}\lambda_k(\underline{x}_k-\underline{x}_0)(\underline{x}_k-\underline{x}_0)^T\}\underline{u}=\sum_{k=1}^{m}\eta k(\underline{x}_k-\underline{x}_0), \tag{6.15}$$

其中 $\eta_k=\lambda_k(\underline{x}_k-\underline{x}_0)^T\underline{u}, k=1,2,\cdots,m$ 目前出现的向量 $\underline{u}$ 仅为 $\underline{x}_{\rm opt}+\underline{x}_0, \underline{d}_0$ 和每个 $\underline{s}_j$. 因此，BIGLAG 的工作量同子程序 TRSAPP 的工作量相近。子问题 (6.6) 的参数 $\bar{\Delta}$ 综合以下 3 个考虑被自动设置为某一值。

(1) 由于 ρ 的目的，如第 2 章的第 2 段所述，边界 $\bar{\Delta}\ge\rho$ 被利用。

(2)Υ 分支在图 1 的模块 8 引出时被采用，原因是 DIST $=\|\underline{\rm x}_{\rm t}-\underline{\rm x}_{\rm opt}\|$ 是无法接受的大，故条件 $\bar{\Delta}\le 0.1$DIST 是合理的。

(3)$\bar{\Delta}$ 应该小于第五章所讲信赖域子问题中的当前 Δ，同时我们期望 Δ 或许可

以减半。这些想法和细节提供了以下选择：

$$\bar{\Delta} = \max[\min\{0.1\text{DIST}, 0.5\}, \rho], \tag{6.16}$$

这在实际操作中看起来很现实，即实际所给定的 ρ_{beg} 导致 ρ 比所要求的针对变量的改变更小。在通过子程序 BIGLAG 构建完步 $\underline{d}$ 之后，参数 (4.12) 得到计算，且 $\underline{x}^+$ 是等于向量 $\underline{x}_{\rm opt} + \underline{d}$. 我们已经陈述过在理论上参数 α 和 β 都是正的这一事实，但是 $\sigma = \alpha\beta + \tau^2$ 可能随机地取到负值，这是由于计算机舍入误差。我们同样回顾可知公式 (4.11) 即便在 σ 是负值时也可应用，但是如果 σ 特别接近于零，则更新将会没有帮助。因此来自 BIGLAG 的 $\underline{d}$ 在且仅在当前参数 σ 小到具有以下性质时不再继续使用。

$$|\sigma| = |\alpha\beta + \tau^2| \leq 0.8\tau^2. \tag{6.17}$$

步 $\underline{d}$ 的选择由我们称为 BIGDEN 的子程序决定，该程序通过寻找一个记号 $|\sigma|$ 的较大值来替代 $|\tau|$ 的一个较大值。使用定义 (6.3) 可知 σ 关于 $\underline{x} = \underline{x}_{\rm opt} + \underline{d}$ 的依赖性，通过将 $\underline{x}^+ = \underline{x}$ 和 $\underline{w} = \underline{w}(x)$ 代入到表达式 (4.12) 来获得。那么 BIGDEN 将步 $\underline{d}$ 设置为下列子问题的一个近似解。

$$\text{Maximize} \quad |\sigma(\underline{x}_{\rm opt}) + \underline{d}\| \qquad \text{subject to} \qquad \|\underline{d}\| \leq \bar{\Delta} \tag{6.18}$$

其中 $\bar{\Delta}$ 仍然具有值 (6.16). 这个任务比 BIGLAG 的计算艰难多了，原因是 $\sigma(\underline{x}), \underline{x} \in \mathcal{R}^n$, 是二次多项式。幸运的是，数值实验说明情形 (6.17) 在实际中非常罕见。

子程序 BIGLAG 和 BIGDEN 的方法是类似的，除了由于它们目标函数的不同而导致的较明显的区别。实际上，BIGDEN 同样也挑选满足方程 (6.8) 的初始向量 $\underline{d}_0$ 和 $\underline{s}_1$, 目的是开始一个迭代过程。同样地，第 j 次迭代让 $\underline{d}_j$ 拥有形式 (6.7) 但是目前 θ 是通过极大化 $|\sigma(\underline{x}_{\rm opt} + \underline{d}_j)|$ 来计算的，当 $j \geq 2$ 时，向量 $\underline{d} = \underline{d}_j$ 在具有以下性质时，我们使用 BIGDEN 来获取返回值：

$$|\sigma(\underline{x}_{\rm opt} + \underline{d}_j)| \leq 1.1\|\sigma(\underline{x}_{\rm opt} + \underline{d}_{j-1})\|, \tag{6.19}$$

或者在 j 已经达到值 n 的条件下。测试 (6.19) 与条件 (6.13) 类似。否则，在 j 值增加 1 后，梯度 $\nabla\sigma(\underline{x}_{\rm opt}) + \underline{d}_{j-1}$ 得到构建，使用了一些已知的数据，正如本章最

后所描述的那样。如果不等式

$$\|\underline{d}_{j-1}^T\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_{j-1})\|^2<(1-10^{-8})\bar{\Delta}^2\|\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_{j-1})\|^2 \tag{6.20}$$

成立，则 $\mathcal{S}_j=\text{span}\{\underline{d}_{j-1},\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_{j-1})\}$ 是良定义的 $\mathcal{R}^n$ 的二维子空间。这时另一个迭代开始，$\underline{s}_j$ 被设置为 $\mathcal{S}_j$ 中具有性质 (6.8) 的向量。然而，若测试 (6.20) 失败，求解子问题 (6.18) 的一阶条件会接近实现。因此，BIGDEN 返回值为 $\underline{d}=\underline{d}_{j-1}$。在 BIGDEN 中 $\underline{d}_0$ 的选择是被 BIGLAG 挑选的 $\underline{d}$。原因是我们期望在 $|l_t(\underline{x}_{\rm opt}+\underline{d})|$ 大时，$|\sigma(\underline{x}_{\rm opt}+\underline{d})|$ 也很大，尽管舍入误差已经造成了不受欢迎的情形 (6.17)。方向 $\underline{s}_1$ 从 $\mathcal{S}_1=\text{span}\{\underline{d}_0,\underline{u}\}$ 取得，其中 $\underline{u}$ 为从点 $\underline{x}_{\rm opt}$ 到其它某一插值点的步。k 值的选取取决于率

$$w_i=\frac{|(\underline{x}_i-\underline{x}_{\rm opt})^T\underline{d}_0|^2}{\|\underline{x}_i-\underline{x}_{\rm opt})\|^2\|\underline{d}_0\|^2},\qquad i\in\{1,2,\cdots,,m\}\backslash\{\text{opt}\}. \tag{6.21}$$

我们先令 $k=t$。该选择在 $w_t\le 0.99$ 的情况下所出。否则，k 值选取为使得 w_k 是率 (6.21) 的最小值的值。对于该程序的一些批评主要是说它忽略了目标函数 σ，这就是为什么对于终止的测试 (6.19) 没有在第一轮迭代中进行尝试。可能性 $\underline{u}=\underline{\nabla}\sigma(\underline{x}_{\rm opt})$ 是不吸引人的，原因是 $\underline{\nabla}\sigma(\underline{x}_{\rm opt})$ 在算术中正好为零，且挑选 $\underline{u}=\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_0)$ 可能并不方便，原因是前文所述用来支持梯度的数量还不够。这里的细节是否可修改

设 $\hat{\sigma})(\theta),\theta\in\mathcal{R}$ 在 $\underline{d}=\underline{d}_\theta$ 是向量 (6.7) 时，是 $\sigma(\underline{x}_{\rm opt}+\underline{d})$ 的值。BIGDEN 迭代的主要任务是聚集系数 $\check{\sigma}_l,l=1,2,\cdots 9$, 使得 $\hat{\sigma}$ 由以下方程确定

$$\hat{\sigma}(\theta)=\check{\sigma}_1+\sum_{k=1}^{4}\{\check{\sigma}_{2k}\cos(k\theta)+\check{\sigma}_{2k+1}\sin(k\theta)\},\qquad\theta\in\mathcal{R}. \tag{6.22}$$

由于方程 (4.25) 的右手边被用于计算 σ，维度为 $(m+n)\times 5$ 的矩阵 U 和 V 构建出来，分别提供 $\underline{w}-\underline{v}$ 和 $H(\underline{w}-\underline{v})$ 有关部分对于角度 θ 的依赖性。我们通过将下面的向量放入表达式 (6.3) 来定义 $\underline{w}$

$$\underline{x}=\underline{x}_{\rm opt}+\underline{d}(\theta)=\underline{x}_{\rm opt}+\cos\theta\underline{d}_{j-1}+\sin\theta\underline{s}_j,\quad\theta\in\mathcal{R}, \tag{6.23}$$

但是 $\underline{v}$ 的定义 (4.24) 同 θ 是独立的。因此，我们找到元素

$$\begin{aligned}(\underline{w}-\underline{v})_i &= \frac{1}{2}\{(\underline{x}-\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)\}^2 - \frac{1}{2}\{(\underline{x}_{\rm opt}-\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)\}^2 \\ &= \frac{1}{2}\{(\underline{x}-\underline{x}_{\rm opt})^T(\underline{x}_i-\underline{x}_0)\}\{(\underline{x}-\underline{x}_{\rm opt}-2\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)\} \\ &= \{\hat{v}_i\cos\theta+\hat{w}_i\sin\theta\}\{\hat{v}_i+\frac{1}{2}\hat{v}_i\cos\theta+\frac{1}{2}\hat{w}_i\sin\theta\}, \qquad 1\le i\le m,\end{aligned} \tag{6.24}$$

以及

$$(\underline{w}-\underline{v})_{i+m+1} = \cos\theta(\underline{d}_{j-1})_i + \sin\theta(\underline{s}_j)_i \qquad i=1,2,\cdots,n, \tag{6.25}$$

其中 $\hat{u}_i,\hat{v}_i$ 和 $\hat{w}_i$ 分别标量积 $(\underline{x}_{\rm opt}-\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)$,$\underline{d}_{j-1}^T(\underline{x}_i-\underline{x}_0)$, $\underline{s}_j^T(\underline{x}_i-\underline{x}_0)$, 我们通过将这些 $\underline{w}-\underline{v}$ 的元素视为 θ 的函数来构建 U 的行，将它们逐一写为如下形式：

$$U_{i1}+U_{i2}\cos\theta+U_{i3}\sin\theta+U_{i4}\cos(2\theta)+U_{i5}\sin(2\theta), \qquad i=1,2,\cdots,m+n \tag{6.26}$$

接下来，我们依据下面的性质来定义 V:

$$V_{i1}+V_{i2}\cos\theta+V_{i3}\sin\theta+V_{i4}\cos(2\theta)+V_{i5}\sin(2\theta), \qquad i=1,2,\cdots,m+n, \tag{6.27}$$

上面这些项是 $H(\underline{w}-\underline{v})$ 的前 m 个和后 n 项。换句话说，由于 $(\underline{w}-\underline{v})_{m+1}$ 是零，V 是乘积 $H_{red}U$, 其中 H_{red} 是除去了第 $(m+1)$ 行和列的矩阵 H，其得到了描述了方程 (4.23) 一段的注意。两个表达 (6.26) (6.27) 的乘积展现了其为一个约束加上 $\cos(k\theta)$ 和 $\sin(k\theta), k=1,2,3,4$ 的线性组合，且结果对 i 求和，因此我们找到以下方程的系数 $\check{\beta}_l, l=1,2,\cdots,9$:

$$(\underline{w}-\underline{v})^T H(\underline{w}-\underline{v}) = \check{\beta}_1 + \sum_{k=1}^{4}\{\check{\beta}_{2k}\cos(k\theta)+\check{\beta}_{2k}\sin(k\theta)\}, \qquad \theta\in\mathcal{R}. \tag{6.28}$$

接下来介绍这些系数对表达式 (6.22) 的贡献：定义 (6.3) 和 (6.24) 提供了公式 (4.26) 的 $\underline{w}^T\underline{e}_{\rm opt} = \frac{1}{2}\{(\underline{x}_{\rm opt}-\underline{x}_0)^T(\underline{x}-\underline{x}_0)\}^2$ 和 $\underline{v}^T\underline{e}_{\rm opt} = \frac{1}{2}\|\underline{x}_{\rm opt}-\underline{x}_0\|^4$。由方程 (4.12)，(4.26) 和 (4.25)，可知允许 $\hat{\sigma}$能被写成如下形式：

$$\begin{aligned}\hat{\sigma}(\theta) = \alpha[&\frac{1}{2}\|\underline{x}-\underline{x}_0\|^4 - \{(\underline{x}_{\rm opt}-\underline{x}_0)^T(\underline{x}-\underline{x}_0)\}^2 \\ &+\frac{1}{2}\|\underline{x}_{\rm opt}-\underline{x}_0\|^4] - \alpha(\underline{w}-\underline{v})^T H(\underline{w}-\underline{v}) + [\underline{e}_t^T H(\underline{w}-\underline{v})]^2\end{aligned} \tag{6.29}$$

由于 $\alpha=\underline{e}_t^T H\underline{e}_t$ 是独立于 $\underline{x}=\underline{x}_{\rm opt}+d(\theta)$ 的，子程序 BIGDEN 将方程 (6.22) 所要求的系数在初始时设置为 $\check{\sigma}_l=-\alpha\check{\beta}_l, l=1,2,\cdots,9$, 的平方项来进行调整。对

于该方程最后一项的调整是基于在 $i = t$ 情形下，关于 θ 的函数 $e_t^T H(w - v)$ 开始的，因此 BIGDEN 将该方程的平方项，表达如常数加一个 $cos(k\theta)$ 和 $sin(k\theta)$ k=1,2,3,4 的线性组合，且它将合成系数加到了 $\check{\sigma}_l$ 的相关值上,$l = 1, 2, \cdots 9$. 除此之外，我们可以由条件 (6.23) 和 (6.8) 推知方程 (6.29) 的第一个平方值包含下式：

$$(\bar{\Delta}^2 + \bar{v}_{\text{opt}} \cos\theta + \hat{w}_{\text{opt}} \sin\theta)^2 + \bar{\Delta}^2 + (\bar{u}_{\text{opt}} - \frac{1}{2}\bar{\Delta}^2), \qquad \theta \in \mathcal{R}, \tag{6.30}$$

$\bar{\Delta}$ 从何来？包含指什么？ 其中 $\hat{u}_{\text{opt}}, \hat{v}_{\text{opt}}$ 和 $\hat{w}_{\text{opt}}$ 取自表达式 (6.24) 并由此可知，$\check{\sigma}_l$ 系数的最终调整是基于元素的。接下来并确定了最大化 $|\hat{\sigma}(2\pi\hat{i}/50)|$ 的指数，$\hat{i} \in [0, 49]$。BIGDEN 直接从方程 (6.22) 计算了值 $\hat{\alpha}(2\pi/50), i = 0, 1, \cdots 49$ 接下来二次多项式 $\hat{q}(\theta), \theta \in \mathcal{R}$, 由对 $\hat{\sigma}$ 进行的在点 $\theta = 2\pi i/50, i = \hat{i} - 1, \hat{i}, \hat{i} + 1$ 的处的插值来构建，$\underline{d}_j$ 的定义 (6.7) 的 θ 的选择，通过将其赋值为使得 $|\hat{q}(\theta)|$ 在其插值点范围中极大来完成。

在计算完 $\underline{d}_j$ 且测试 (6.19) 失败时，我们需要梯度 $\underline{\nabla}\sigma(\underline{x}_{\text{opt}} + \underline{d}_{j-1})$。我们准备从表达式 (6.29) 来推导它，右手边是方程 $\sigma(\underline{x}), \underline{x} \in \mathcal{R}$ 其中 $\underline{w}$ 通过方程 (6.3) 取决于 $\underline{x}$. 我们考虑寻找在 j 的旧值下寻找 $\underline{\nabla}\sigma(\underline{x}_{\text{opt}} + \underline{d}_j)$ 的等价任务，以此来保留前三段的记号。方程 (6.29) 的第一行在 $\underline{x} = \underline{x}_{\text{opt}} + \underline{d}$ 处的梯度是向量

$$\begin{aligned} &2\alpha[\|\underline{x} - \underline{x}_0\|^2(\underline{x} - \underline{x}_0) - \{(\underline{x}_{\text{opt}} - \underline{x}_0)^T(\underline{x} - \underline{x}_0)\}(\underline{x}_{\text{opt}} - \underline{x}_0)] \\ &\qquad = 2\alpha[\|\underline{x} - \underline{x}_0\|^2\underline{d}_j + -\{\underline{d}_t^T(\underline{x} - \underline{x}_0)\}(\underline{x}_{\text{opt}} - \underline{x}_0)] \end{aligned} \tag{6.31}$$

其右手边由关系式 $(\underline{x} - \underline{x}_0) = \underline{d}_j + (\underline{x}_{\text{opt}} - \underline{x}_0)$ 来得到，其通过使用 BIGDEN 得到，目的是在 $\|\underline{d}_j\|$ 相对小的时候避免抵消。方程 (6.29) 的梯度的余项是以下求和式：

$$-2\alpha \sum_{i=1}^{m+n+1} \{H(\underline{w} - \underline{v})\}_i \underline{\nabla}\{w(\underline{x})_i\} + 2\{\underline{e}_t^T H(\underline{w} - \underline{v})\} \sum_{i=1}^{m+n+1} H_{ti} \underline{\nabla}\{w(\underline{x})_i\}. \tag{6.32}$$

目前，一项该工作的优势在于用来选择 θ 的向量 $H(\underline{w} - \underline{v})$ 的前 m 和后 n 个元素，因此表达式 (6.27) 提供了数 $\hat{\eta}_i = \{H(\underline{w} - \underline{v})\}_i, i = 1, 2, \cdots, m$ 和 $\check{\eta}_i = \{H(\underline{w} - \underline{v})\}_{i+m+1}, i = 1, 2, \cdots, n$ 回忆方程 (4.12)(4.25),$t \neq \text{opt}$, 知 $\underline{e}_t^T H(\underline{w} - \underline{v})$ 是 τ 的当前值。因此由于定义 (6.3) 显示 $w(x)_{m+1}$ 是常值，和式 (6.32) 可以被写为如下形式：

$$2\sum_{i=1}^{m}(\tau H_{ti} - \alpha\bar{\eta}_i)\underline{\nabla}\{w(\underline{x})_i\} + 2\sum_{i=1}^{n}(\tau H_{ti+m+1} - \alpha\bar{\eta}_i)\underline{\nabla}\{w(\underline{x})_{i+m+1}\} \tag{6.33}$$

等式 (6.3) 给出: $\underline{\nabla}\{w(\underline{x})_i = \{(\underline{x}-\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)\}(\underline{x}_i-\underline{x}_0), i = 1,2,\cdots m$ 以及 $\underline{\nabla}\{w(\underline{x})_{i+m+1}\} = \underline{e}_i, i = 1,2,\cdots n$。

由此知所要求的 $\sigma(\underline{x})$ 的梯度是命名为 (6.31) 和

$$2\sum_{i=1}^{m}\{(\tau H_{ti} - \alpha\bar{\eta}_i)(\underline{x}-\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)\} \quad (\underline{x}_i-\underline{x}_0) \tag{6.34}$$

以及 $\mathcal{R}^n$ 中的元素为 $2(\tau H_{ti+m+1} - \alpha\check{\eta}_i), i = 1,2,\cdots n,$ 的向量之和。以上即为 BIGDEN 方法的完整描述。

7 NEWUOA 软件的其它细节

我们从第 2 章的图 1 可知 Δ 的修正和 MOVE 设置在子模块 4中进行，ρ 的减小在模块 12 进行，同时，模块 14 给出了一个测试。此外从第 1 章的末尾可回忆得知原点的移动对于 H 矩阵的准确性十分重要，以上这些操作的细节将在本章列出:

设 Δ_{old} 和 Δ_{new} 分别是 Δ 在模块 4 中产生的旧值和新值，如前所述，Δ_{new} 的选择取决于率 (2.2)，同时步 $\underline{d}$ 的欧氏长度也受到了关注。

在 RATIO $\le 0.1, 0.1 <$ RATIO ≤ 0.7 和 RATIO ≥ 0.7 的三种情形下，Δ_{new} 的可能值分别是 $\frac{1}{2}\|\underline{d}\|, \|\underline{d}\|$ 和 $2\|\underline{d}\|$，但是我们认为，如果 RATIO > 0.1, 则对于 Δ 在下一迭代中，一个巨大的减少是过于受限的。除此之外，我们观察界限 $\Delta \ge \rho$, 我们试图通过避免信赖域半径离 ρ 太近而锐化模块 10 的测试，因此，在 $\Delta_{\text{int}} \le 1.5\rho$ 或 $\Delta_{\text{int}} > 1.5\rho$ 时，NEWUOA 软件分别将 Δ_{new} 设置为 ρ 或 Δ_{int}，其中 Δ_{int} 是中间值。

$$\Delta_{\text{int}} = \begin{cases} \dfrac{1}{2}\|\underline{d}\|, & \text{RATIO} \in 0.1, \\ \max\{\|\underline{d}\|, \dfrac{1}{2}\Delta_{\text{old}}\}, & 0.1 < \text{RATIO} \le 0.7, \\ \max\{2\|\underline{d}\|, \dfrac{1}{2}\Delta_{\text{old}}\}, & \text{RATIO} > 0.7. \end{cases} \tag{7.1}$$

正如表达式 (4.12) 所说模块 4 中 MOVE 的选择为更新公式 (4.11) 提供了一个相对大的分母。我们回忆可知在这个表达式中的 $H\underline{w}$ 和 β 是关于脚标 t 独立的。设 $\mathcal{T}$ 是集合 $\{1,2,\cdots m\}$，除了指标 opt 在 $F(\underline{x}_{\text{opt}} + \underline{d}) \ge F(\underline{x}_{\text{opt}})$ 时被排除在 $\underline{\mathcal{T}}$ 之外，目的是阻止 $\underline{x}_{\text{opt}}$ 从插值点集移走。数值

$$\sigma_t = (\underline{e}_t^T H \underline{e}_t)\beta + (\underline{e}_t^T H \underline{w})^2, \qquad t \in \underline{\mathcal{T}}, \tag{7.2}$$

被计算出来了，此时 σ_t 是可能从选择 MOVE = t 中得出的记号，然而，因为我们想保持插值点靠近 x_{opt} 故在使 $|\sigma_{\text{MOVE}}|$ 尽可能大的时候有一个较大的劣势。比如这个劣势在以下情况中发生：当 $\underline{x}_i, i = 1, 2, \cdots m$ 中至少有 $n+1$ 个点同 $\underline{x}_{\text{opt}}$ 的距离在 Δ 内，但是 $\underline{x}_t$ 却远得多。接下来 Lagrange 条件 (6.1) 建议 l_t 可能分母不像方程 $\frac{\|\underline{x}-\underline{x}_{\text{opt}}\|^2}{\|\underline{x}-\underline{x}_{\text{opt}}\|^2}$ $\underline{x} \in \mathcal{R}^n$，受界 $\|\underline{d}\| \leq \Delta$ 的影响，可能会有如下性质

$$\|l_t(\underline{x}_{\text{opt}} + \underline{d})\| = \mathscr{O}(\Delta^2/\|\underline{x}_t - \underline{x}_{\text{opt}}\|^2.) \tag{7.3}$$

现在方程 (6.5) 包括 $\underline{e}_t^T H \underline{w} = l_t(\underline{x}_{\text{opt}} + \underline{d})$, 且其常用于帮助 $(e_t^T H e_t)\beta$ 和 $(e_t^T H w)^2$ 成为表达式 (7.2) 的类似量级的正数值。因此，对于通常的 $t \in I$, 在 $\|x_t - x_{\text{opt}}\| \leq \Delta$ 或 $\|x_t - x_{\text{opt}}\| \geq \Delta$ 的情形下，$|\sigma_{t^*}|$ 分别是 $\mathscr{O}(1)$ 或 $\mathscr{O}(\Delta^4/\|x_t - x_{\text{opt}}\|^4)$. 因此，NEWUOA 将 MOVE 设置为零或者是满足下列方程的指标 $t^* \in \mathcal{T}$:

$$w_{t^*}|\sigma_{t^*}| = \max\{w_t|\sigma_t\| : t \in \mathcal{T}\}, \tag{7.4}$$

其中 w_t 是一个权重向量，它对于距离 $\underline{x}_{\text{opt}}$ 较远的插值点 $\underline{x}$ 的自动移动是必要的。这个移动是受使用 $\|\underline{x}_t - \underline{x}_{\text{opt}}\|$ 的一个六次幂激励得出的，而非是前文所示的四次幂。另一个考虑是让插值点仅在半径 Δ 减少抑或已经达到其最低界 ρ 时，在 $\underline{x}_{\text{opt}}$ 附近聚合，因此权重赋值如下:

$$w_t = \max[1, \{\|\underline{x}_t - \underline{x}^*\|/\max[0.1\Delta, \rho]\}^6] \quad , t \in \mathcal{T}, \tag{7.5}$$

其中 $\underline{x}^*$ 是准备在图 1 中的模块 5 接受挑选的 $\underline{x}_{\text{opt}}$。脚标 MOVE = 0 选择坚持旧的插值点，因此其仅在情形 $F(\underline{x}_{\text{opt}} + \underline{d}) \geq F(\underline{x}_{\text{opt}}$ 下是有效的。我们希望避免导致 H 元素的非正常增长。这使得我们开始考虑在一个遥远的插值点被舍弃时，一些增长是正常的事情，因此，脚标 MOVE 被设置为零而非 t^* 当且仅当条件 $F(\underline{x}_{\text{opt}} + \underline{d}) \geq F(\underline{x}_{\text{opt}}$ 和 $w_{t^*}\|\sigma_{t^*}\| \leq 1$ 均成立。

在图 1 模块 12 中 ρ 的值从 ρ_{old} 减少到 ρ_{new}，减少量为 $\frac{1}{10}$，除非只有一或两个发生在 ρ 上的变化试图获得值 $\rho = \rho_{end}$. 方程 $\rho_{\text{old}}/\rho_{new} = \rho_{new}/\rho_{end}$ 在后者情形的两个减少。这些分析和参数的选择为使用 NEWUOA 软件来调整 ρ 提供了一下公式

$$\rho_{new} = \begin{cases} \rho_{end}, & \rho_{\text{old}} \leq 16\rho_{end}, \\ (\rho_{\text{old}}\rho_{end})^{\frac{1}{2}}, & 16\rho_{end} < \rho_{\text{old}} \leq 250\rho_{end}, \\ 0.1\rho_{\text{old}}, & \rho_{\text{old}} > 250\rho_{end}. \end{cases} \tag{7.6}$$

图 1 中模块 14 的原理在第 2 章倒数第 2 段中进行了阐释，基于当前 ρ 值的计算在 Y 分支得到选取到此是完整的。

我们可以看到模块 14 在模块 2 中信赖域子问题在进行了一个具有性质 $\|\underline{d}\| < \frac{1}{2}\rho$ 的迭代步时得到使用，这表明当前的二次模型是凸的。因此，假设 CRVMIN 是 $\nabla^2 Q$ 最小特征根的有效估计，我们希望在对于目标函数 F 的预测减小值，记作 $Q(\underline{x}_{\rm opt}) - Q(\underline{x}_{\rm opt} + \underline{d})$ 比 $\frac{1}{8}\rho^2$(CRVMIN) 小时，不计算 $Q(\underline{x}_{\rm opt} + \underline{d})$。除此之外，如果最新迭代的误差值也比这个值要小时，我们以为这时尝试提升模型的准确度是对的。特别地，在具备以下条件时，模块 14 的测试得到满足，至少 F 的 3 个新值基于当前的 ρ 值得到了计算以及所有条件

$$\|\underline{d}^{(j)}\| \le \rho \quad 以及 \quad |Q_j(\underline{x}_{\rm opt}^{(j)} + \underline{d}^{(j)}) - F(\underline{x}_{\rm opt}^{(j)} + \underline{d}^{(j)})| \le \frac{1}{8}\rho^2 {\rm CRVMIN}, {\rm j} \in \mathscr{I} \quad (7.7)$$

成立。其中 Q_j, $\underline{d}^{(j)}$ 和 $\underline{x}_{\rm opt}^{j}$ 分别是第 j 项迭代中在模块 5 开始的 Q, $\underline{d}$, 和 $\underline{x}_{\rm opt}$, 其中 CRVMIN, 在当前迭代生成，且 τ, 包括 3 个整数，记为在当前迭代之前的 3 个对于模块 5 的新的访问。因此虽然有距离 $\|\underline{x}_i - \underline{x}_{\rm opt}\|, i = 1, 2, \cdots, m$ 可能会超过 2ρ 具有当前 ρ 值的 NEWUOA 常常终止。CRVMIN 为 $\nabla^2 Q$ 的最小特征根

为了在实际中展现 $\underline{x}_0$ 对于更新公式 (4.11) 到产生的舍入误差的重要性。我们假设所有插值点之间距离 $\|\underline{x}_i - \underline{x}_j\|, 1 \le i < j \le m$ 都是一倍的，以及 $\|\underline{d}\| = \|\underline{x}^+ - \underline{x}_{\rm opt}\|$ 也是一倍的，但是我们让 $\|\underline{x}_{\rm opt} - \underline{x}_0\| = M$ 是很大的。理论上表达式 (4.12) 的参数 α, β, τ 和 σ 也是矩阵 H 的头 $m \times m$ 子矩阵，是关于 $\underline{x}_0$ 独立的 [?], 但是定义 (4.10) 说明每个 $\underline{w}$ 的前 m 个元素都大约为 $\frac{1}{2}M^4$. 因此在下面公式中发生了很多抵消。

$$\beta = \frac{1}{2}\|\underline{x}^+ - \underline{x}_0\|^4 - \underline{w}^T H \underline{w}. \quad (7.8)$$

除此之外，如果在 H_{11} 中有 ε 的误差，且如果在方程 (7.8) 的右手边没有其它误差，那么 β 可以包括一个尺度为 $M^8\varepsilon$ 的误差，这个 M 的幂非常大以致于 $M > 100$ 可能会分离。将表达式 (4.26) 用公式 (7.8) 代替，不适合度更弱，原因是 H_{11} 是与 $-(w_1 - v_1)^2$ 相乘得到，且方程 (6.24) 的中间行提供了值：

$$w_1 - v_1 = \frac{1}{2}\{(\underline{x}^+ - \underline{x}_{\rm opt})^T(\underline{x}_1 - \underline{x}_0)\}\{(\underline{x}^+ + \underline{x}_{\rm opt} - 2\underline{x}_0)^T(\underline{x}_1 - \underline{x}_0)\}. \quad (7.9)$$

这里误差分析的 8 次幂和 6 次幂再看清楚些

因此，β 的误差现在是尺度 $M^6\varepsilon\cos^2\theta$ 其中 θ 是在 $\underline{x}_1 - \underline{x}_0$ 和 $\underline{d} = \underline{x}^+ - \underline{x}_{\rm opt}$

之间的角度。分解式 (4.16) 同样也帮助当前足够效率的获得，除此之外，我们从 REAL*8 的数值实验中发现，使用一些复杂的目标函数时，一系列迭代可能在 $\|\underline{x}_{\rm opt}-\underline{x}_0\|\geq 10^{2.5}\|\underline{d}\|$ 被允许出现在第 4 章更新计算中时导致不可接受的误差。因此 NEWUOA 测试了条件：

$$\|\underline{d}\|^2\leq 10^{-3}\|\underline{x}_{\rm opt}-x_0\|^2 \tag{7.10}$$

在如图 1 模块 5 所示用 $\underline{x}_{\rm opt}+\underline{d}$ 来替代 $\underline{x}_{\rm MOVE}$ 之前。如果条件成立，那么 $\underline{x}_0$ 被 $\underline{x}_{\rm opt}$ 重写，这发生在模块 5 的开始。该操作选择了矩阵 (1.3) 的后 n 行和所有元素 (3.11)，然而实际上，表达式所示的矩阵 (1.3) 替代 W 被存储，因此 H 在使用时受发生在 W 的变化影响的方式得到修正，除了矩阵 H 的第 $m+1$ 行和第 $m+1$ 列不作要求，该任务的细节在 ([?]) 的第 8 章作了详细分析，因此下面仅给出当 $\underline{x}_0$ 移位时发生在 H 上的变化的大致纲要。

设在 $\underline{x}_0$ 被 $\underline{x}_{\rm opt}$ 重写之前 $\underline{x}_{av}$ 和 s 分别是向量 $\frac{1}{2}(\underline{x}_0+\underline{x}_{\rm opt})$ 和 $(\underline{x}_{\rm opt}-\underline{x}_0)$。设矩阵 Y 是拥有下面列向量的 $n\times m$ 矩阵：

$$\underline{y}_j=\{\underline{s}^T(\underline{x}_j-\underline{x}_{av})\}(\underline{x}_j-\underline{x}_{av})+\frac{1}{4}\|\underline{s}\|^2\underline{s},\qquad j=1,2,\cdots,m \tag{7.11}$$

并令参数 $\Theta_{\rm old}$ 和 Θ_{new} 分别为旧的和新的矩阵 H 除去它们 ([?]) 的第 $m+1$ 行和列后得矩阵。那么，根据方程 (5.11) 和 (5.12) 知 θ_{new} 通过以下公式得到定位

$$\Theta_{\rm new}=\left(\begin{array}{c|c} I & 0\\ \hline Y & I\end{array}\right)\Theta_{\rm old}\left(\begin{array}{c|c} I & Y^T\\ \hline 0 & I\end{array}\right) \tag{7.12}$$

因此，如前所述，表达式 (3.12) 的子矩阵没有受到干扰，但我们保持其分解式 (4.16)。可以从表达式 (3.12) 和 (7.12) 知乘积 $Y\Omega$ 以及求和式 $Y\Xi_{red}^T+\Xi_{red}Y^T+Y\Omega Y^T$ 分别被加在 Ξ 的最后 n 行和 Υ 尾部的 $n\times n$ 子矩阵。其中 Ξ_{red} 是原始矩阵 Ξ 去除掉第一行。当 $\underline{x}_0$ 被 $\underline{x}_{\rm opt}$ 重写时，梯度 $\nabla Q(\underline{x}_0)$ 也必须随之得到修正。特别地由于函数 (3.1) 可以被写为如下形式：

$$Q(\underline{x}_{\rm opt}+\underline{d})=Q(\underline{x}_{\rm opt})+\underline{d}^T\nabla Q(\underline{x}_{\rm opt})+\frac{1}{2}\underline{d}^T\nabla^2Q\underline{d},\qquad \underline{d}\in\mathcal{R}^n \tag{7.13}$$

且由于 $\nabla Q(\underline{x}_{\rm opt})=\nabla Q(\underline{x}_0)+\nabla^2Q\underline{s}$ 可从 $\underline{s}=\underline{x}_{\rm opt}-\underline{x}_0$ 得到，向量 ∇^2Q 被添加进 $\nabla Q(\underline{x}_0)$,$Q$ 的常数项是不必要的，这一点正如第 4 章最后所说，且 ∇^2Q 是独立于

$\underline{x}_0$，这一点除了在方程 (4.28) 所述，它可以被表达成求和式：

$$\begin{aligned}\nabla^2 Q &= \Gamma + \sum_{j=1}^{m} \gamma_j (\underline{x}_j - \underline{x}_0)(\underline{x}_j - \underline{x}_0)^T \\ &= \Gamma + \sum_{j=1}^{m} \gamma_j (\underline{x}_j - \underline{x}_0 + \underline{s})(\underline{x}_j - \underline{x}_0 + \underline{s})^T \\ &= \Gamma + \underline{v}\underline{s}^T + \underline{v}\underline{s}^T + \sum_{j=1}^{m} \gamma_j (\underline{x}_j - \underline{x}_{\text{opt}})(\underline{x}_j - \underline{x}_{\text{opt}})^T\end{aligned} \tag{7.14}$$

其中 $\underline{v} = \sum_{j=1}^{m} \gamma_j (\underline{x}_j - \underline{x}_{\text{opt}} + \frac{1}{2}\underline{s}) = \sum_{j=1}^{m} \gamma_j (\underline{x}_j - \underline{x}_{av})$ 因此，尽管参数 $\gamma_j j = 1, 2, \cdots m$ 是不变的，在 $\underline{x}_0$ 中的移动要求 $\underline{v}\underline{s}^T + \underline{s}\underline{v}^T$ 添加到 T 中。

前段的工作量仅仅是 $\mathscr{O}(mn)$，但是乘积 (7.12) 的补充占用了 $O(m^2 n)$, 的操作。因此我们希望条件 (7.10) 仅在迭代总数的小部分分解中成立，特别是在 n 较大时。下一章中数值实验的运行次数粗略回答了此问题，受此启发，软件 NEWUOA 每次迭代的平均工作通常离 $\mathscr{O}(mn)$ 很近。

8 数值结果

在 2003 年 11 月，作者将前文描述的已经测试过一系列多至 200 维问题的 NEWUOA 算法的 Fortran 发行公开。之后，在 Erlce 的一系列会议中，他和 Nick Gould 讨论了一些可以进行尝试的其它问题，这引出了更多实验。

从这些实验可以搞清楚一件事，即它那更多的修正可能会具有随机地优势，当前已经做了，而且是本章的第一个目标，原因是与其一起的数值结果被 NEWUOA 的最新版本计算。启发了我们进行修正的实验是 Buekley(1989) 第 98 页的 VARDIM 测试问题。其目标函数是二次多项式

$$F(\underline{x}) = \sum_{l=1}^{n} (x_l - 1)^2 + \{\sum_{l=1}^{n} (x_l - 1)\}^2 + \{\sum_{l=1}^{n} (x_l - 1)\}^4, \qquad \underline{x} \in \mathcal{R}^n \tag{8.1}$$

其在 $\underline{x} = \underline{e}$ 由 1 组成的向量取到函数的最小值零。解析差给出二阶导数矩阵为

$$\nabla^2 F(\underline{x}) = 2I + [2 + 12\{\sum_{l=1}^{n} l(x_l - 1)\}^2]\Theta \qquad \underline{x} \in \mathcal{R}^n \tag{8.2}$$

其中 I 是 $n \times n$ 单位矩阵，且 Θ 是秩一矩阵，且其元素为 $\Theta_{ij} = ij, 1 \le i, j \le n$. 因此，$\nabla^2 F(\underline{x})$ 有 $n-1$ 个特征值为 2,1 个特征值为 $2 + [\frac{1}{3} + 2\{\sum_{l=1}^{n} l(x_l - 1)\}^2] n(n+1)(2n+1)$ 然而，当 NEWUOA 在 $m = 2n + 1$ 的情形下使用时。初始的二次模

型有一个对角的二阶导数矩阵，$\nabla^2 Q$ 的对角元素近似等于 $\nabla^2 F(\underline{x}_0)$ 的对角元素，其中 $\underline{x}_0$ 是给定的变量的起始变量，在 VARDIM 测试问题中。我们有元素 $1-i/n, i=1,2,\cdots n$, 因此 $\nabla^2 Q$ 的特征值在初始阶段大于 $2+[2+12\{\sum_{l=1}^{n} l(x_l-1)\}^2]i^2, i=1,2,\cdots n$, 平方项是 $2+\frac{1}{3}(n+1)^2(2n+1)^2$. 由此可知；在计算的初始阶段，$\nabla^2 Q$ 是 $\nabla^2 F$ 一个很差的估计。此外，若 $n=80$，$\nabla^2 Q$ 的特征值会从 5.7×10^7 到 3.6×10^{11}，但是 $\nabla^2 F$ 在 $\underline{x}=\underline{e}$ 最大特征值仅仅是 347762，因此 NEWUOA 不能表现得令人满意，除非通过一个系列迭代的更新公式，可以作出对 $\nabla^2 Q$ 的巨大提升。

然而，不幸的是，最小 Frobenius 范数更新方法对 $\nabla^2 Q$ 作出了被新的插值条件允许的最小改变，故 NEWUOA 的基本方法不适用于 VARDIM 测试问题，因此最新修正尝试识别何时 $\nabla^2 Q$ 的元素值会过于大，以及是否有较强的依据来证明这一可能性，此时，用 $Q_{\rm int}$ 来替代 Q，其为使得 $\|\nabla^2 Q_{\rm int}\|_F$ 极小化的二次模型，而非 $\nabla^2 Q$ 的变量的 Frobenius 范数，这些的约束条件为 $Q_{\rm int}(\underline{x}_i)=F(\underline{x}_i), i=1,2,\cdots m$, 插值点 $\underline{x}_i$ 是从图 1 模块 5 出来的更新的情形。当 $Q_{\rm int}$ 受到青睐时，梯度 $\underline{\nabla} Q_{\rm int}(\underline{x}_0)$ 和一下表达式的参数 $\gamma_j, j=1,2,\cdots m$ 是被要求的。

$$\nabla^2 Q_{\rm int}=\sum_{j=1}^{m}\gamma_j(\underline{x}_j-\underline{x}_0)(\underline{x}_j-\underline{x}_0)^T \tag{8.3}$$

由 $Q_{\rm int}$ 的定义可以它们是向量 $\underline{g}$ 和系统 (3.10) 中 $\underline{\lambda}$ 的元素，其中 γ 由元素 $\gamma_i=F(\underline{x}_i), i=1,2,\cdots m, \phi\in\mathcal{R}$ 任意选取。一些有计算机舍入误差导致的损失通过选择 $\phi=F(\underline{x}_{\rm opt})$ 来避免了。我们从记号 (3.12) 可以推知：$\underline{g}$ 和 $\underline{\lambda}$ 分别是乘积 $\Xi_{\rm red}\underline{r}$ 和 $\Omega\underline{r}$, 其中 $\Xi_{\rm red}$ 仍然是矩阵 Ξ 除掉它的第一行。因此 NEWUOA 的算法在 $\mathscr{O}(m^2)$ 以操作内构造了一个 $Q_{\rm int}$ 的有用形式。

当 $\nabla^2 Q$ 的元素太大时，插值方程 (1.1) 指出 $\|\underline{\nabla} Q(\underline{x})\|$ 对于多数变量所构成的向量也过于大。通常情况下一个 $\|\underline{\nabla} Q(\underline{x}_{\rm opt})\|$ 的大值会导致率 (2.2) 变小。此外，由于 $\underline{\nabla} Q(\underline{x}_0)$ 是有效的，且我们已经发现 $\underline{\nabla} Q_{\rm int}(\underline{x}_0)$ 是乘积 $\Xi_{\rm red} r$, 比较 $\|\underline{\nabla} Q_{int}(\underline{x}_0)\|$ 和 $\|\underline{\nabla} Q(\underline{x}_0)\|$ 也是简单的。

在 NEWUOA 新版本中从图 1 的模块 4 到这模块 5 的迭代中，一个 flag 被设置为 YES 或者 NO，当模块 5 末尾的以下条件成立时选择 YES

$$\text{RATIO}\le 0.01 \qquad \text{以及} \qquad \|\underline{\nabla}\text{Q}_{\rm int}(\underline{\text{x}}_0)\|\le 0.1\|\underline{\nabla}\text{Q}(\underline{\text{x}}_0)\|. \tag{8.4}$$

接着，模型 Q 被 $Q_{\rm int}$ 替代当且仅当三个标志连续被设置成了 YES。具有 80 维

变量的 NEWUOA 的一个较老版本求解得到，而无需顾及曾提及的 $\nabla^2 Q$ 的缺点。未修正和已修正版本在使用 $\rho_{beg} = (2n)^{-1}$ 和 $\rho_{beg} = 10^{-6}$ 时的结果分别展示在表 1 的左手边和右手边。标记 $\#F$ 代表对目标函数的计算的总次数，另外 $\underline{x}_{\rm fin}$ 是由 NEWUOA 算法反馈来的变量向量，原因是其给出了 F 的最小计算值。理论上，变量重新排序对于不会造成结果的区别，插值点的初始集合保持 $m = 2n+1$ 不变，故该机制可以被用来探索计算机舍入误差导致的一些影响。表 1 中冒号左右两边

n	Original NEWUOA		Modified NEWUOA	
	$\#F$	$F\,(\underline{x}_{\rm fin}\,)$	$\#F$	$F\,(\underline{x}_{\rm fin}\,)$
20	12018: 11517	$2\times10^{-11} : 8\times10^{-11}$	5447: 4610	$4\times10^{-11} : 3\times10^{-11}$
40	45510: 56698	$7\times10^{-10} : 3\times10^{-10}$	17106: 17853	$1\times10^{-10} : 8\times10^{-11}$
80	196135: 234804	$7\times10^{-9} : 3\times10^{-9}$	60305: 55051	$1\times10^{-10} : 3\times10^{-10}$

表 1：在 $m = 2n+1$ 时使用两个版本的 NEWUOA 方法求解 VARDIM 问题

的条目是在不同排序下得到的。我们可以从中看出舍入误差对数值实验结果有很高的影响力，以及 $F(\underline{x}_{\rm fin})$ 的值是符合要求的，同时还可以发现修正成功地达到了减少 $\#F$ 的目的。

在 NEWUOA 软件的发展过程中，被使用最多的目标函数是三角平方求和函数

$$F(\underline{x}) = \sum_{i=1}^{2n}\{b_i - \sum_{j=1}^{n}(S_{ij}\sin(\theta_j x_j + C_{ij}\cos(\theta_j x_j))\}^2, \qquad \underline{x} \in \mathcal{R}^n \tag{8.5}$$

我们将其记作 TRIGSSQS。矩阵 S 和 C 的元素是由区间 [-100 ,100] 内的随机整数，每个标量因子 Q_j 都是由 [0.1,1] 上的对数分布得到的，且参数 $b_i, i = 1, 2, \cdots, 2n$，是通过 $F(\underline{x}^*) = 0$ 来进行定义的，其中 $\underline{x}^*$ 由元素；$x_j^* = \hat{x}_j^*/\theta_j, j = 1, 2, \cdots n$，每个 $\underline{\hat{x}}_j^*$ 是由 $[-\pi, \pi]$ 上的均匀分布选取得。

初始向量 $\underline{x}_0$ 有元素：$(\hat{x}_j^* + 0.1\hat{y}_j^*)/\theta_j, j = 1, 2, \cdots, n$ 其中每个 $\hat{y}_j^*$ 也同样是从 $[-\pi, \pi]$ 上随机选取的。由于周期性函数 (8.5) 具有鞍点和极大值，同时缩放系数 θ_j 的取值提出了一个比 $\theta_j = 1, j = 1, 2, \cdots, n$ 更难的问题。对于任意 n，我们通过选择不同的随机数来生成 5 个不同的目标函数和起始点。我们让插值条件的个数，记为 m，为 $2n+1, m^{(\rm av)}$ 或 $\frac{1}{2}(n+1)(n+2)$，其中 $m^{(\rm av)}$ 是距离 $\{(n+\frac{1}{2})(n+1)(n+2)\}^{\frac{1}{2}}$ 最近的一个整数，NEWUOA 软件在这些情况下的结果，(在有 n 的 4 个值和参数 $\rho_{\rm beg} = 10^{-1}, \rho_{\rm end} = 10^{-6}$ 的情况下) 展示在表 2 中，表中主要部分的条目是 5 个所提到在表 2 中，测试问题的平均 $\#F$ 值和 $\underline{x}_{\rm fin}$ 值都已经被定义好了。同样，这些结果对于计算机舍入误差十分敏感。考虑到 Sun Ultra 10 工作站上的运行时间，表中的破折号表示对相应问题没有进行尝试。

在表中，$m = 2n+1$ 部分 $\#F$ 的值明显比作者一开始期望得要小，原因是当 n

很大时，它们变得比二次模型的自由度的数目要小。这个受到高度欢迎的情形为使用最小 Frobenius 范数的更新技术提供了特别棒的支持。对比表中 $\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$ 条目和 ρ_{bed} 可知该计算的精确度是符合要求的。

NEWUOA 算法，尤其是表 1 中界 $\Delta \geq \rho$ 的使用，确定为适合于极小化具有不连续一阶导数的方程。

n	$m = 2n+1$		$m = m^{(\text{av})}$		$m = \frac{1}{2}(n+1)(n+2)$	
	$\#F$	$\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$
20	931	1.4×10^{-6}	833	6.9×10^{-7}	649	2.0×10^{-7}
40	1809	4.2×10^{-6}	1716	1.3×10^{-6}	2061	5.5×10^{-7}
80	3159	3.8×10^{-6}	3471	2.1×10^{-6}	–	–
160	6013	5.8×10^{-6}	–	–	–	–

表 2：使用 NEWUOA 方法求解 5 个版本的 TRIGSSQS 问题均值

因此表 3 给了对于具有如下形式的目标函数 TRIGSABS 使用此方法求解的结果。

$$F(\underline{x}) = \sum_{i=1}^{2n} |b_i - \sum_{j=1}^{n} (S_{ij} \sin x_j + C_{ij} \cos x_j)|, \qquad \underline{x} \in \mathcal{R}^n \tag{8.6}$$

参数 b_i, S_{ij} 和 C_{ij} 以及初始向量 $\underline{x}_0$ 是按照前面所述随机生成的，除了我们令缩放系数 $\theta_j = 1, j = 1, 2, \cdots, n$, 不同的随机数，如前一样对每个 n 提供 5 个不同的测试问题。我们保留 $\rho_{beg} = a1$。

n	$m = 2n+1$		$m = m^{(av)}$		$m =$	$\frac{1}{2}(n+1)(n+2)$
	$\#F$	$\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$
20	1454	1.0×10^{-8}	2172	6.6×10^{-9}	4947	4.8×10^{-9}
40	3447	1.6×10^{-8}	6232	7.7×10^{-9}	24039	5.9×10^{-9}
80	7626	1.2×10^{-8}	16504	7.2×10^{-9}	–	–
160	16496	2.2×10^{-8}	–	–	–	–

表 3：使用 NEWUOA 方法求解 5 个版本的 TRIGSABS 问题均值

但是我们设置 $\rho_{bed} = 10^{-8}$ 目的是使目标函数 F 在取得极小值的 $\underline{x} = \underline{x}^*$ 处锐利的优势。表 3 的条目和表 2 的条目是类似的。我们发现，对于每一个 n,$\#F$ 的最小值出现在 $m = 2n + 1$ 这一列，这一结果令人振奋。如果 ρ_{end} 减小至 $10^{-6}, m = 2n + 1$ 和 $n = 160$ 的图变成 $\#F = 12007$ 和 $\|\underline{x}_{\text{fin}} - \underline{x}^* \| b = 1.6 \times 10^{-6}$。故 $\#F$ 又比二次模型的自由度的数目少。

我们接下来考虑一个作者近期创造出的测试问题，叫作 SPHRPTS 这里 n 是偶数，且 $\frac{n}{2}$ 个点必须安置在三维空间中的单位球的表面上，位置还要离得很远。我

们让第 k 个点 $\underline{p}_k \in \mathcal{R}^3$ 有坐标。

$$\underline{p}_k = \begin{pmatrix} \cos x_{2k-1} \cos x_{2k} \\ \sin x_{2k-1} \cos x_{2k} \\ \sin x_{2k} \end{pmatrix}, \quad k = 1, 2, \ldots, n/2, \tag{8.7}$$

其中 $\underline{x} \in \mathcal{R}^n$ 仍然是变量向量，问题为极小化函数

$$F(\underline{x}) = \sum_{k=2}^{n/2} \sum_{l=1}^{k-1} \|\underline{p}_l - \underline{p}_k\|^{-2} \qquad \underline{x} \in \mathcal{R}^n, \tag{8.8}$$

其中点 $\underline{p}_k$ 在初始时均匀点置放在球的赤道上，向量 $\underline{x}_0$ 具有元素 $(\underline{x}_0)_{2k-1} = 4\pi k/n$ 和 $(\underline{x}_0)_{2k} = 0, k = 1, 2, \cdots n/2$, NEWUOA 软件被应用于此问题，将 m 和 n 赋值如从表 2 和表 3，另 $\rho_{\text{beg}} = n^{-1}$，$\rho_{\text{end}} = 10^{-6}$，$\#F$ 的结果值在表 4 中冒号的左侧展示。

n	$m = 2n + 1$	$m = m^{(\text{av})}$	$m = \frac{1}{2}(n + 1)(n + 2)$
20	2077: 351	1285: 513	1161: 627
40	7245: 1620	4775: 2884	6636: 2924
80	9043: 3644	18679: 13898	
160	24031: 8193	–	–

表 4: 求解 SPHRPTS 问题所用的 $\#F$ 数目

我们还发现 $F(\underline{x}_{\text{fin}})$ 和 $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$ 的极小值保持一致，对于超出 10 位小数的空间，尽管最佳变量向量有很多的自由度，原因是单位球的点和旋转的置换不会选择双倍和 (8.8)，因此在实际中很多变量的调整仅仅导致目标函数的一点减少。实际上，在计算了每个 $F(\underline{x}_{\text{fin}})$ 之后，我们检查，通过 NEWUOA 计算得到的 F 的序列，目的是标记满足 $F(\underline{x}) \leq 1.001F(\underline{x}_{\text{fin}})$ 的首个值所构成序列的位置。这些位置在表 4 中冒号的右手边给出。我们发现对于 SPHRPTS 问题，太多数工作花费在对于 F 的临界增长，特别是在计算 $m = 2n + 1$ 列时。NEWUOA 软件被用来测试了 n 个其他作者提出的问题。最后的表展示了在下面 5 种情况下使用 $m = 2n + 1$ 的结果。ARWHEAD 问题 (见 [?] 的附录) 有目标函数

$$F(\underline{x}) = \sum_{i=1}^{n-1} \{(x_i^2 + x_n^2)^2 - 4x_i + 3\}, \qquad \underline{x} \in \mathcal{R}^n, \tag{8.9}$$

且起始点 $\underline{x}_0$ 是 $\underline{e} \in \mathcal{R}^n$, 其仍然是元素为 1 的向量在 CHROSEN 问题 (见 [?] 的第

45 页)，我们设目标函数 F 为函数

$$F(\underline{x}) = \sum_{i=1}^{n-1}\{4(x_i - x_{i+1}^2)^2 + (1 - x_{i+1})^2\}, \qquad \underline{x} \in \mathcal{R}^n, \tag{8.10}$$

且初始点 x_0 为 $-e \in \mathcal{R}^n$。PENALTY1 问题 (见 [?] 的第 79 页)，包括两个参数

$$F(\underline{x}) = 10^{-5}\sum_{i=1}^{n}(x_i - 1)^2 + (\frac{1}{4} - \sum_{i=1}^{n} x_i^2)^2, \qquad \underline{x} \in \mathcal{R}^n, \tag{8.11}$$

起始点为 $(\underline{x}_0)_i = i, i = 1, 2, \cdots n.$ 我们对于 PENALTY2 问题 (见 [?] 的第 80 页) 的参数选择。给出方程

$$\begin{aligned} F(\underline{x}) = &\sum_{i=2}^{n}\{(e^{x_{i-1}/10} + e^{x_i/10} - e^{(i-1)/10} - e^{i/10})^2 + (e^{x_i/10} - e^{-1/10})^2\} \\ &+ \{1 - \sum_{i=1}^{n}(n - i + 1)x_i^2\}^2 + (x_1 - \frac{1}{5})^2, \qquad \underline{x} \in \mathcal{R}^n, \end{aligned} \tag{8.12}$$

且起始点 $\underline{x}_0$ 是 $\frac{1}{2}\underline{e} \in \mathcal{R}^n$。PENALTY3 问题 (见 [?] 的第 81 页，1989) 有目标函数

$$\begin{aligned} F(\underline{x}) = &10^{-3}(1 + Re^{x_n} + Se^{x_{n-1}} + RS) \\ &+ \{\sum_{i=1}^{n}(x_i^2 - n)\}^2 + \sum_{i=1}^{n/2}(x_i - 1)^2, \qquad \underline{x} \in \mathcal{R}^n, \end{aligned} \tag{8.13}$$

其中 R 和 S 是以下求和：

$$R = \sum_{i=1}^{n-2}(x_i + 2x_{i+1} + 10x_{i+2} - 1)^2, \qquad \text{以及} \qquad S = \sum_{i=1}^{n-2}(2x_i + x_{i+1} - 3)^2, \tag{8.14}$$

且我们令起始点 $\underline{x}_0 \in \mathcal{R}^n$ 为零向量。我们在每个情形中 $\rho_{end} = 10^{-6}$，同时，ρ_{beg} 在 ARWHEAD，CHROEN，PENALTY1，PENALTY2 和 PENALTY3 中分别被赋值为 $0.5, 0.5, 1.0$ 1.0 和 1.0。表 5 展示了将在我们设置 n 为 (除了 $*$ 表示 $\#F$) 超过 500,000 的常用值时，将 NEWUOA 应用于这些时产生的函数估计的数目。

n	ARWHEAD	CHROSEN	PENALTY1	PENALTY2	PENALTY3
20	404	845	7476	2443	3219
40	1497	1876	14370	2455	16589
80	3287	4314	32390	5703	136902
160	8504	9875	72519	$\star$	$\star$

表 5：在 $m = 2n + 1$ 时求解 5 个问题的所用 $\#F$ 数目

所有 ARWHEAD, CHROEN, PENALTY1 的计算都成功完成了，最大的距离 $\|\underline{x}_{\text{fin}} - \underline{x}^*\|_\infty$ 是 6.1×10^{-6}，其中 $\underline{x}_{\text{fin}}$ 和 $\underline{x}^*$ 仍然是最后和最优的变量向量。在

$n \leq 80$ 时的对于 PENALTY2 的计算中，$F(\underline{x}_{\text{fin}})$ 的值与 13 小数位保持一致，同时有从变量和 m 的其它选择的置换得到的其它值。然而，当 $n = 160$ 被选中时，常数 $e^{i/10}, i = 1, 2, \cdots, n$ 从 1.1 遍布到 9×10^{-6}，故在表达式 (8.12) 中的第一个求和 τ 号下的尺度，从 1 跨到 10^{13}，这导致 PENALTY2 问题在 REAL*8 代数中过于困难。我们将 PENALTY3 中计算所给出的值同交换了变量后得这些值进行了对比。$\#F$ 条目在 $n = 20, n = 40$ 和 $n = 80$ 的情况下分别变为 $4336, 18209$ 和 125884，这同表 5 的最后一列相吻合。此外，对于每个 n，$\underline{x}_{\text{fin}}$ 的两个值都比 n^2 略小，且它们与 11 小数位保持一致。然而，PENALTY3 的一个性质是目标函数的极小值离 10^{-3} 非常近且很难被找到，这个尺度通过挑选变量 $x_i = 1, i = 1, 2, \cdots, n-1$ 和 $x_n = -(n^2 - n + 1)^{1/2}$ 被显示出来，原因是 e^{x_n} 非常小而且 S 和表达式的第 2 行都是零，这提供了 $F(\underline{x}) = 10^{-3}(1 + Re^{x_n}) \approx 10^{-3}$ 当 NEWUOA 被应用到 $n = 160$ 的 PENALTY3 时，变量的原始顺序产生 $\#F = 629582$ 和 $F(\underline{x}_{\text{fin}}) = 0.001002$. 我们不曾期望新顺序应用起来是仍然很棒的，原因是结果的差别完全由计算机舍入误差导致。

每个迭代的平均工作量在第 7 章最后提及，在 $m = 2n+1$ 的情形最佳为 $\mathscr{O}(n^2)$. 我们在表 5 中的 ARWHEAD 和 PENALTY1 实验中测试该可能性。在 Sun Ultra 10 工作台上每次计算的总时间 (以秒为单位) 都除以了 n^2 和 $\#F$ 的乘积。在 $n = 20, n = 40, n = 80$ 和 $n = 160$ 情形下求解 ARWHEAD 问题的最终商分别是 $8.4\times10^{-6}, 8.0\times10^{-6}, 8.4\times10^{-6}, 8.5\times10^{-6}$, 和 8.8×10^{-6}, 且 PENALTY1 问题的相对商分别是 $9.2\times10^{-6}, 8.5\times10^{-6}, 8.6\times10^{-6}$ 和 9.3×10^{-6}。

在最后一种情形下的运行时间接近 5 个小时，与此同时，在 $n = 20$ 时，求解 ARWHEAD 问题仅仅需要 1.36 秒。这些发现告诉我们，每次迭代的平均复杂度同 n^2 成比例，这也是最受欢迎的。

软件 NEWUOA 的发展进行了将近 3 年，这项工作是让人非常沮丧的，原因是在矩阵 Ω 的分解式被引进之前，在复杂情形中计算机舍入误差会导致剧烈的损害，因此作者曾怀疑使用显性 (直接) 逆矩阵 $H = W^{-1}$ 来替代使用允许系统 (3.10) 在 $\mathscr{O}(m^2)$ 次操作内被求解的 W 的分解形式。数值结果仍然对于计算机舍入误差十分敏感，但是本章的实验表明最终达到了一个好的精确度，其证实了所给技术的稳定性，因此我们总结可知为了更新二次模型所使用的最小 Frobenius 范数方法在求解无约束无导数优化问题时是相当成功的。

欢迎读者们通过发送电子邮件至 *mjdp@com.ac.uk* 来得到 NEWUOA 的 Fortran 软件的免费版本。

附录：第 3 章中的证明

第 3 章最后两段所提出的论断在下面作为带有证明的引论给出。在第 3 章开头，我们在矩阵 (3.12) 之后描述了相关插值点的位置。

引理 1: 初始矩阵 Ξ 的首行有元素 (3.13)，且对于每一个满足 $2 \leq i \leq \min[n+1, m-n]$ 的整数 i, 第 i 行包括元素 (3.14). 当 $m \leq 2n$ 时，矩阵 Ξ 剩余行的非元素取值 (3.15)，其中 i 是区间 $[m-n+1, n+1]$ 内的任意整数。初始矩阵的所有其它元素都是零。

证明：对于区间 $[1, m]$ 内的每一个整数 j, 我们设二次多项式：

$$l_j(x) = l_j(x_0) + (x+x_0)^T \nabla l_j(x_0) + \frac{1}{2}(x+x_0)^T \nabla^2 l_j(x_0) \qquad , x \in \mathcal{R}^n \qquad (A.1)$$

为初始插值点处的第 j 个 Lagrange 函数，这意味着 $\|\nabla^2 l_j\|_F$ 在下面条件的约束下尽可能地小。

$$l_j(x) = \delta_{ij}, \qquad i = 1, 2, \cdots, m \qquad (A.2)$$

在这第 6 章的第 2 段有所陈述。l_j 的结构和第 3 章中二次多项式 D 的结构是一样的，约束 (3.6) 有性质 $F(x_i) - Q_{\mathrm{old}(x_i)} = \delta_{ij}, i = 1, 2, \cdots, m$。因此当 γ 是坐标向量 $e_j \in \mathcal{R}^m$ 时 $l_j(x_0)$ 和 $\nabla l_j(x_0)$ 分别同系统 (3.10) 中的 c 和 g 相同。在这种情况下，方程 (3.10) 左手边的分别向量是 W^{-1} 的第 j 列。从标记 (3.12) 可以得出 $l_j(x_0)$ 和 $\nabla l_j(x_0)$ 提供了矩阵 Ξ 的第 j 列，如下面表达式所示：

$$\Xi = \begin{pmatrix} l_1(x_0) & l_2(x_0) & \cdots & l_m(x_0) \\ \nabla l_1(x_0) & \nabla l_2(x_0) & \cdots & \nabla l_m(x_0) \end{pmatrix} \qquad (A.3)$$

剩余的证明部分依赖于初始插值点的位置。特别地，由于在条件 (A.2) 中每个 j 的首项的选择 $x_1 = x_0$，矩阵 (A.3) 的第一行有元素 (3.13)。除此之外，当 k 满足 $1 \leq k \leq \min[n, m-n-1]$ 时，点 $x_{k+1} = x_0 + \rho_{beg} e_k$ 和 $x_{k+n+1} = x_0 - \rho_{beg} e_k$ 被选择了，故 $\nabla l_j(x_0) l_j(x_0)$ 的第 k 个元素是差商：

$$\begin{aligned} (\nabla l_j(x_0))_k &= (2\rho_{beg})^{-1} (\mathscr{L}_j(x_{k+1}) - \mathscr{L}_j(x_{k+n+1})) \\ &= (2\rho_{beg})^{-1} (\delta_{k+1j} - \delta_{k+n+1j}), \qquad j = 1, 2, \cdots, m \qquad (A.4) \end{aligned}$$

原因是 l_j 是取值为 (A.2) 的二次函数。我们用脚标 i 来替代 $k+1$, 且表达式 (A.3) 给出 $\nabla l_j(x_0)_k = \Xi_{k+1j} = \Xi_{ij}$ 从方程 (A.4) 可知公式 (3.14) 确实针对 $2 \le i \le \min[n+1, m-n]$ 提供了矩阵 Ξ 第 j 行的所有非零向量。最终, 若 k 满足 $m-n \le k \le n$, 则只有向量 $x_1 = x_0, x_{k+1} = x_0 + \rho_{beg} e_k$ 和 $x_0 - \rho_{beg} e_k$ 的前两个是插值点。此外, 由条件 (A.2) 约束下极小化 $\|\nabla^2 l_j\|_F$ 得出 $(\nabla^2 l_j)_{kk} = 0, j = 1, 2, \cdots, m$, 故单变量函数 $l_j(x_0 + \alpha e_k), \alpha \in R$ 对于每个 j 都是一个线性多项式, 因此矩阵 (A.3) 的第 $k+1$ 行包括差商:

$$\begin{aligned} \Xi_{k+1j} = (\nabla_j(x_0))_k &= (\rho_{beg})^{-1}(\mathscr{L}_j(x_{k+1}) - \mathscr{L}_j(x_1)) \\ &= (\rho_{beg})^{-1}(\delta_{k+1j} - \delta_{1j}), \qquad j = 1, 2, \cdots, m \end{aligned} \qquad (A.5)$$

同样, 我们再次用 i 来替换 $k+1$, 故方程 (A.5) 给出; 矩阵 Ξ 第 i 行的非零向量, 在 $m-n+1 \le i \le n+1$ 时有值 (3.15)。引理的证明完成。

引理 2: 当 $m \ge 2n+1$ 成立时, 初始矩阵 Y 是零矩阵。否则, Y 是对角矩阵, 且表达式 (3.16) 给出 Y 的所有非零元。

证明: $\widetilde{m}$ 是整数 $\min[m, 2n+1]$, 并设 $\widetilde{\Xi}, \widetilde{A}$ 和 $\widetilde{x}$ 分别为 Ξ, A 和 x 的头 $(n+1)\times\widetilde{m}, \widetilde{m}\times(n+1)$ 和 $(n+1)\times(n+1)$ 子矩阵。定义 (3.12) 提出矩阵方程 $\Xi A + \gamma x = 0$ 且其前 $n+1$ 列给出性质 $\widetilde{\Xi}\widetilde{A} + \gamma\widetilde{x} = 0$。这取决于引理 1 中的性质, 那若 $m > 2n+1$, 则 Ξ 的后 $m-2n-1$ 列为零。我们从方程 (3.2) 和 (3.11) 推导出 $\widetilde{A}$ 具有元素:

$$\left.\begin{aligned} \widetilde{A}_{ii} = A_{ii} = \frac{1}{2}\rho_{beg}^4, i = 2, 3, \cdots n+1 \\ \widetilde{A}_{i+ni} = A_{i+ni} = \frac{1}{2}\rho_{beg}^4, i = 2, 3, \cdots, \widetilde{m} - n \\ \widetilde{A}_{ij} = A_{ij} = 0, \end{aligned}\right\} \begin{aligned} &i = 1, 2, \cdots, , \widetilde{m}. \\ &j = 1, 2, \cdots, n+1. \end{aligned} \qquad (A.6)$$

我们寻找乘积 $\widetilde{\Xi}, \widetilde{A}$ 的元素, 它是一个方阵, 对于每一个区间 $[1, n+1]$ 内的 j, 方程 (3.13), (3.14) 和 (3.15) 给出公式:

$$(\widetilde{\Xi}\widetilde{A})_{ij} = \begin{cases} \widetilde{A}_{1j}, & i = 1 \\ (2\rho_{beg})^{-1}(\widetilde{A}_{ij} - \widetilde{A}_{i+nj}), & 2 \le i \le \min[n+1, m-n], \\ \rho_{beg})^{-1}(\widetilde{A}_{ij} - \widetilde{A}_{1j}), & m-n+1 \le i \le n+1, \end{cases} \qquad (A.7)$$

最后一行在 $m \ge 2n+1$ 时是无效的, 从方程 (A.6) 可知 $\widetilde{\Xi}, \widetilde{A}$ 是一个对角矩阵, 且它的第一行和第一列都是零。此外, 由于 $\min[n+1, m-n]$ 和 $\widetilde{m} - n$ 相同, 我们

找到对角元素。

$$\left.\begin{aligned}(\widetilde{\Xi}\widetilde{A})_{ii} &= 0, && 1 \le i \le \widetilde{m} - n \\ (\widetilde{\Xi}\widetilde{A})_{ii} &= \frac{1}{2}\rho_{beg})^3, && m - n + 1 \le i \le n + 1\end{aligned}\right\} \qquad (A.8)$$

现在我们考虑性质 $\widetilde{\Xi}\widetilde{A} + \gamma\widetilde{x} = 0$。$X$ 在 $x_1 = x_0$ 下的定义 (1.3) 指出 $\widetilde{\Xi}e_1 = e_1$, 其中 e_1 是 $\mathcal{R}^{n+1}$ 中的第一个坐标向量，同时我们从之前可知 $\widetilde{\Xi}\widetilde{A}e_1 = 0$，由 $(\widetilde{\Xi}\widetilde{A} + \gamma\widetilde{x})e_1 = 0$ 可知 γ 的第 1 列也是零, 因此即使在 $\widetilde{x}$ 的第 1 行发生了某些变化的情况下 $\widetilde{\Xi}\widetilde{A} + \gamma\widetilde{x} = 0$ 仍然正确。表达式 (1.3) 和 (3.2) 令新的 $\widetilde{x}$ 为 ρ_{beg} 乘积 $(n+1)\times(n+1)$ 单位矩阵，因此 γ 是矩阵 $-\rho_{beg}^{-1}\widetilde{\Xi}\widetilde{A}$ 我们通知它是对角的，此外，可以从方程 (A.8) 推得 γ 在 $m \ge 2n+1$ 的情形下是零矩阵，以及，否则 γ 的非零元素取值 (3.16). 因此引理得证。

引理 3, 初始矩阵 Ω, 具有以下分解式;

$$\Omega = \sum_{k=1}^{m-n-1} z_k z_k^T = ZZ^T \qquad (A.9)$$

其中向量 $Z_k \in \mathcal{R}^m, k = 1, 2, \cdots, m-n-1$, 是 Z 矩阵的列向量，此外，这些向量的首个 $\min[n, m-n-1]$ 具有元素 (3.18)，同时，若 $m > 2n+1$, 则剩余的向量有元素 (3.20)，脚标 $\hat{p}$ 和 $\hat{q}$ 在第 3 章的最后一段有所介绍。

证明：设 $Q(x), x \in \mathcal{R}^n$ 是第 3 章开头所给的初始的二次模型。二阶导数矩阵 $\nabla^2 Q$ 的每一个元素要么有方程 (1.1) 定义，要么设置为零。因此，在插值条件的约束下模型 Q 的选择使得 $\|\nabla^2 Q\|_F$ 极小。由系统 (3.10) 的推导可知，如果我们让 r 具有元素 $r_i = F(x_i), i = 1, 2, \cdots, m$, 且如果我们设 $\lambda = \Omega r$, 其中 Ω 是表达式 (3.12) 中选取的，则 λ 是满足约束 (3.7) 且使得 $\nabla^2 Q$ 是矩阵 (3.8) 的唯一向量。

这些评述唯一地刻画了矩阵 Ω，原因是它们对于所有右手边 $r_i = F(x_i), i = 1, 2, \cdots, m$ 都有效。因此我们可充分地核查知如果我们将矩阵 (A.9) 对于一般的 r 都代入 λ 方程 $\lambda = \Omega r$，则 λ 就具有之前所提的性质。

约束 (3.7) 的第一条是 $\lambda^T e = 0$, 其中 $e \in \mathcal{R}^m$ 是由 1 构成的向量。将其用 $\lambda = \Omega r$, 和 $\Omega = zz^T$ 替换，则条件变成 $r^T zz^T e = 0$, 它是达到的，原因是矩阵 z 的每一列 z_k 有元素 (3.18) 或 (3.20)，且两个元素集都提出 $z_k^T e = 0$. 类似地，关系 $\lambda = zz^T r$ 指出其它约束 (3.7) 在 Z 满足方程时也成立。

$$\sum_{i=1}^{m} Z_{ik}(x_i - x_0) = 0, \qquad k = 1, 2, \cdots m-n-1 \qquad (A.10)$$

对于 $1 \leq k \leq \min[n, m-n-1]$, 值 (3.18) 和 (3.2) 表明表正式的左手边是差 $\rho_{beg}e_k - \rho_{beg}e_k = 0$ 的一个倍数，或者对于 $n+1 \leq k \leq m-n-1$, 值 (3.20)，(3.19) 和 (3.3) 在 $i = k+n+1$, 和 $x_1 = x_0$, 时给出如下条件。

$$\sum_{i=1}^{m} Z_{ik}(x_i - x_0) = 2\rho_{beg})^{-2}(-\sigma_p\rho_{beg}e_p - \sigma_q\rho_{beg}e_q + \sigma_p\rho_{beg}e_p + \sigma_q\rho_{beg}e_q) = 0. \qquad (A.11)$$

因此，对于一般 $r =\in \mathcal{R}^m$, 向量 $\lambda = \Omega r$ 的确服从约束 (3.7) 通过替换 $\lambda = zz^T r$, 我们将矩阵 (3.8) 写为如下形式:

$$\nabla^2 D = \sum_{k=1}^{m-n-1} (Z_k^T r)\{\sum_{j=1}^{m} Z_{jk}(x_j - x_0)(x_j - x_0)^T\} \qquad (A.12)$$

同时我们通过建立 $\nabla^2 Q = \nabla^2 D$ 来完成证明。对于 $1 \leq k \leq \min[n, m-n-1]$, 元素 (3.18) 提供方程

$$\left.\begin{aligned} Z_k^T r = \sqrt{2}\rho_{beg}^{-2}\{-F(x_0) + \frac{1}{2}F(x_0 + \rho_{beg}e_k) + \frac{1}{2}F(x_0 - \rho_{beg}e_k) \\ \sum_{j=1}^{m} Z_{jk}(x_j - x_0)(x_j - x_0)^T = \sqrt{2}\rho_{beg}^{-2}\{\rho_{beg}^2 e_k e_k^T\} = \sqrt{2}e_k e_k^T \end{aligned}\right\} \qquad (A.13)$$

除此之外，本章第一段中的构造使得差商

$$(\nabla^2 Q)_{kk} = \rho_{beg}^{-2}\{F(x_0 - \rho_{beg}e_k) - 2F(x_0) + F(x_0 + \rho_{beg}e_k)\}, \qquad (A.14)$$

由此可知表达式 (A.12) 的对 k 求和的前 $\min[n, m-n-1]$ 项提供了一个对角矩阵。其对角元同 $\nabla^2 Q$ 的对角元相同。因此 $\nabla^2 Q = \nabla^2 D$ 在 $m \leq 2n+1$ 的情形达到。其保留下来展示出若 $m > 2n+1$, 则表达式 (A.12) 中的最后 $m-2n+1$ 个 k 值，在不干扰对角元的情况下生成了 $\nabla^2 Q$ 的下三角矩阵。

对于区间 $[n+1, m-n-1]$ 中的每个 k 插值点 (3.3) 和 (3.19) 在 $i = k+n+1$ 情形的相关的。实际上，元素 (3.20) 指出 $z_k^T r$ 仅是方程 (3.5) 的左手边，与此同时，表达式 (A.12) 中的大括号里的项是矩阵

$$-e_p e_p^T - e_p e_p^T + (\sigma_p e_p + \sigma_q e_q)(\sigma_p e_p + \sigma_q e_q)^T = \sigma_p\sigma_q(e_p e_q^T + e_q e_p^T) \qquad (A.15)$$

因此，求和式 (A.12) 的第 k 项有助于数值由 (3.5) 确定的 $(\nabla^2 Q)_{pq}, (\nabla^2 D)_{qp}$，同时它也没有改变 $\nabla^2 D$ 的任何其他元素。因此 $(\nabla^2 Q)$ 的所有可能非零的不同元素有表达式 (A.12) 的不同 k 值提供，矩阵 Z 的初始选择的调整完成了。