DAMTP 2004/NA08

The NEWUOA software for unconstrained optimization without derivatives[1]

M.J.D. Powell

Abstract: The NEWUOA software seeks the least value of a function $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, when $F(\underline{x})$ can be calculated for any vector of variables $\underline{x}$. The algorithm is iterative, a quadratic model $Q\approx F$ being required at the beginning of each iteration, which is used in a trust region procedure for adjusting the variables. When Q is revised, the new Q interpolates F at m points, the value $m=2n+1$ being recommended. The remaining freedom in the new Q is taken up by minimizing the Frobenius norm of the change to $\nabla^2 Q$. Only one interpolation point is altered on each iteration. Thus, except for occasional origin shifts, the amount of work per iteration is only of order $(m+n)^2$, which allows n to be quite large. Many questions were addressed during the development of NEWUOA, for the achievement of good accuracy and robustness. They include the choice of the initial quadratic model, the need to maintain enough linear independence in the interpolation conditions in the presence of computer rounding errors, and the stability of the updating of certain matrices that allow the fast revision of Q. Details are given of the techniques that answer all the questions that occurred. The software was tried on several test problems. Numerical results for nine of them are reported and discussed, in order to demonstrate the performance of the software for up to 160 variables.

Department of Applied Mathematics and Theoretical Physics,
Centre for Mathematical Sciences,
Wilberforce Road,
Cambridge CB3 0WA,
England.

November, 2004.

[1]Presented at The 40th Workshop on Large Scale Nonlinear Optimization (Erice, Italy, 2004).

1. Introduction

Quadratic approximations to the objective function are highly useful for obtaining a fast rate of convergence in iterative algorithms for unconstrained optimization, because usually some attention has to be given to the curvature of the objective function. On the other hand, each quadratic model has $\frac{1}{2}(n+1)(n+2)$ independent parameters, and this number of calculations of values of the objective function is prohibitively expensive in many applications with large n. Therefore the new algorithm tries to construct suitable quadratic models from fewer data. The model $Q(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, at the beginning of a typical iteration, has to satisfy only m interpolation conditions

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

where $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, is the objective function, where the number m is prescribed by the user, and where the positions of the different points $\underline{x}_i$, $i=1,2,\ldots,m$, are generated automatically. We require $m \geq n+2$, in order that the equations (1.1) always provide some conditions on the second derivative matrix $\nabla^2 Q$, and we require $m \leq \frac{1}{2}(n+1)(n+2)$, because otherwise no quadratic model Q can satisfy all the equations (1.1) for general right hand sides. The numerical results in the last section of this paper give excellent support for the choice $m=2n+1$.

The success of the new algorithm is due to a technique that is suggested by the symmetric Broyden method for updating $\nabla^2 Q$ when first derivatives of F are available (see pages 195–198 of Dennis and Schnabel, 1983, for instance). Let an old model $Q_{\rm old}$ be present, and let the new model $Q_{\rm new}$ be required to satisfy some conditions that are compatible and that leave some freedom in the parameters of $Q_{\rm new}$. The technique takes up this freedom by minimizing $\|\nabla^2 Q_{\rm new}-\nabla^2 Q_{\rm old}\|_F$, where the subscript "$F$" denotes the Frobenius norm

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

Our conditions on the new model $Q=Q_{\rm new}$ are the interpolation equations (1.1). Thus $\nabla^2 Q_{\rm new}$ is defined uniquely, and $Q_{\rm new}$ itself is also unique, because the automatic choice of the points $\underline{x}_i$ excludes the possibility that a nonzero linear polynomial $p(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, has the property $p(\underline{x}_i)=0$, $i=1,2,\ldots,m$. In other words, the algorithm ensures that the rows of the $(n+1)\times m$ matrix

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

are linearly independent, where $\underline{x}_0$ is any fixed vector.

The strength of this updating technique can be explained by considering the case when the objective function F is quadratic. Guided by the model $Q=Q_{\rm old}$ at the beginning of the current iteration, a new vector of variables $\underline{x}_{\rm new}=\underline{x}_{\rm opt}+\underline{d}$ is chosen, where $\underline{x}_{\rm opt}$ is such that $F(\underline{x}_{\rm opt})$ is the least calculated value of F so far. If

the error $|F(\underline{x}_{\rm new})-Q_{\rm old}(\underline{x}_{\rm new})|$ is relatively small, then the model has done well in predicting the new value of F, even if the errors of the approximation $\nabla^2 Q\approx\nabla^2 F$ are substantial. On the other hand, if $|F(\underline{x}_{\rm new})-Q_{\rm old}(\underline{x}_{\rm new})|$ is relatively large, then, by satisfying $Q_{\rm new}(\underline{x}_{\rm new})=F(\underline{x}_{\rm new})$, the updating technique should improve the accuracy of the model significantly, which is a win/win situation. Numerical results show that these welcome alternatives provide excellent convergence in the vectors of variables that are generated by the algorithm, although usually the second derivative error $\|\nabla^2 Q-\nabla^2 F\|_F$ is big for every Q that occurs. Thus the algorithm seems to achieve automatically the features of the quadratic model that give suitable changes to the variables, without paying much attention to other features of the approximation $Q\approx F$. This suggestion is made with hindsight, after discovering experimentally that the number of calculations of F is only $\mathcal{O}(n)$ in many cases that allow n to be varied. Further discussion of the efficiency of the updating technique can be found in Powell (2004b).

The first discovery of this kind, made in January 2002, is mentioned in Powell (2003). Specifically, by employing the least Frobenius norm updating method, an unconstrained minimization problem with 160 variables was solved to high accuracy, using only 9688 values of F, although quadratic models have 13122 independent parameters in the case $n=160$. Then the author began to develop a Fortran implementation of the new procedure for general use, but that task was not completed until December, 2003, because, throughout the first 18 months of the development, computer rounding errors caused unacceptable loss of accuracy in a few difficult test problems. A progress report on that work, with some highly promising numerical results, was presented at the conference in Hangzhou, China, that celebrated the tenth anniversary of the journal *Optimization Methods and Software* (Powell, 2004b). The author resisted pressure from the editor and referees of that paper to include a detailed description of the algorithm that calculated the given results, because of the occasional numerical instabilities. The loss of accuracy occurred in the part of the Fortran software that derives $Q_{\rm new}$ from $Q_{\rm old}$ in only $\mathcal{O}(m^2)$ computer operations, the change to Q being defined by an $(m+n+1)\times(m+n+1)$ system of linear equations. Let W be the matrix of this system. The inverse matrix $H=W^{-1}$ was stored and updated explicitly. In theory the rank of Ω, which is the leading $m\times m$ submatrix of H, is only $m-n-1$, but this property was lost in practice. Now, however, a factorization of Ω is stored instead of Ω itself, which gives the correct rank in a way that is not damaged by computer rounding errors. This device corrected the unacceptable loss of accuracy (Powell, 2004c), and then the remaining development of the final version of NEWUOA became straightforward. The purpose of the present paper is to provide details and some numerical results of the new algorithm.

An outline of the method of NEWUOA is given in Section 2, but m (the number of interpolation conditions) and the way of updating Q are not mentioned, so most of the outline applies also to the UOBYQA software of the author (Powell, 2002), where each quadratic model is defined by interpolation to $\frac{1}{2}(n+1)(n+2)$ values of F. The selection of the initial interpolation points and the construction

of the first quadratic model are described in Section 3, with formulae for the initial matrix H and the factorization of Ω, as introduced in the previous paragraph. Not only Q but also H and the factorization of Ω are updated when the positions of the interpolation points are revised, which is the subject of Section 4. On most iterations, the change in variables $\underline{d}$ is an approximate solution to the trust region subproblem

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

which receives attention in Section 5, the parameter $\Delta>0$ being available with Q. Section 6 addresses an alternative way of choosing $\underline{d}$, which may be invoked when trust region steps fail to yield good reductions in F. Other details of the algorithm are considered in Section 7, including shifts of the origin of $\mathcal{R}^n$, which are necessary to avoid huge losses of accuracy when H is revised. Several numerical results are presented and discussed in Section 8. The first of these experiments suggests a modification to the procedure for updating the quadratic model, which was made to NEWUOA before the calculation of the other results. It seems that the new algorithm is suitable for a wide range of unconstrained minimization calculations. Proofs of some of the assertions of Section 3 are given in an appendix.

2. An outline of the method

The user of the NEWUOA software has to define the objective function by a Fortran subroutine that computes $F(\underline{x})$ for any vector of variables $\underline{x}\in\mathcal{R}^n$. An initial vector $\underline{x}_0\in\mathcal{R}^n$, the number m of interpolation conditions (1.1), and the initial and final values of a trust region radius, namely $\rho_{\rm beg}$ and $\rho_{\rm end}$, are required too. It is mentioned in Section 1 that m is a fixed integer from the interval

$$n+2 \le m \le \tfrac{1}{2}(n+1)(n+2), \tag{2.1}$$

and that often the choice $m=2n+1$ is good for efficiency. The initial interpolation points $\underline{x}_i$, $i=1,2,\ldots,m$, include $\underline{x}_0$, while the other points have the property $\|\underline{x}_i-\underline{x}_0\|_\infty=\rho_{\rm beg}$, as specified in Section 3. The choice of $\rho_{\rm beg}$ should be such that the computed values of F at these points provide useful information about the behaviour of the true objective function near $\underline{x}_0$, especially when the computations may include some spurious contributions that are larger than rounding errors. The parameter $\rho_{\rm end}$, which has to satisfy $\rho_{\rm end}\le\rho_{\rm beg}$, should have the magnitude of the required accuracy in the final values of the variables.

An outline of the method is given in Figure 1. The details of the operations of Box 1 are addressed in Section 3. The parameter ρ is a lower bound on the trust region radius Δ from the interval $[\rho_{\rm end},\rho_{\rm beg}]$. The value of Δ is revised on most iterations, but the purpose of ρ is to maintain enough distance between the interpolation points $\underline{x}_i$, $i=1,2,\ldots,m$, in order to restrict the damage to Q from the interpolation conditions (1.1) when there are substantial errors in each computation of F. Therefore ρ is altered only when the constraint $\Delta\ge\rho$ seems to be preventing futher reductions in the objective function. Each change to ρ is

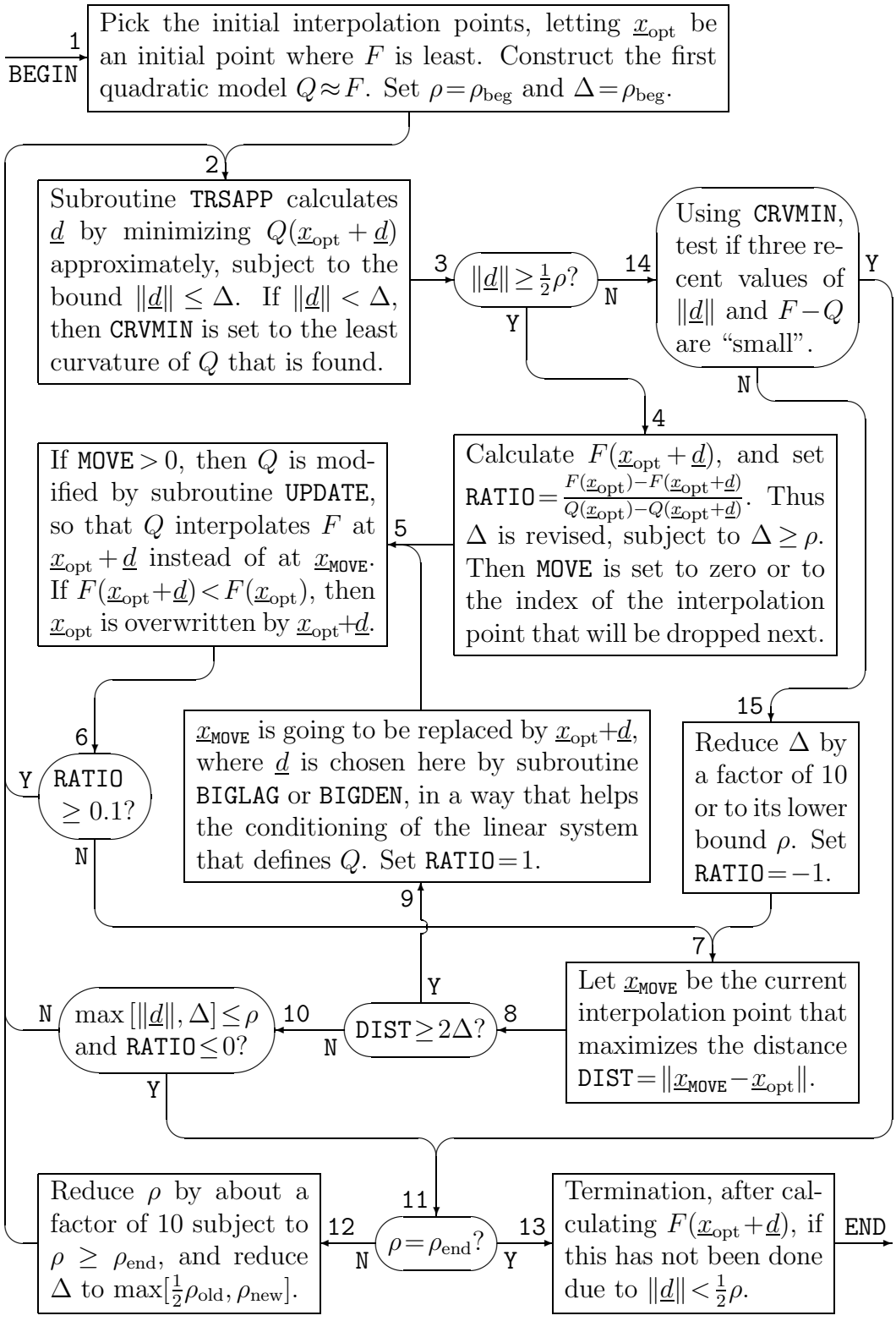

Figure 1: An outline of the method, where Y = Yes and N = No

a decrease by about a factor of ten, except that the iterations are terminated in the case $\rho=\rho_{\rm end}$, as shown in Boxes 11-13 of the figure.

Boxes 2-6 of Figure 1 are followed in sequence when the algorithm performs a trust region iteration that calculates a new value of F. The step $\underline{d}$ from $\underline{x}_{\rm opt}$ is derived from the subproblem (1.4) in Box 2 by the truncated conjugate gradient procedure of Section 5. If $\|\underline{d}\|<\Delta$ occurs here, then Q has positive curvature along every search direction of that procedure, and CRVMIN is set to the least of those curvatures, for use in Box 14 when the N branch is taken from Box 3, which receives attention later. Box 4 is reached in the present case, however, where Δ is revised in a way that depends on the ratio

$$\texttt{RATIO} \;=\; \{F(\underline{x}_{\rm opt})-F(\underline{x}_{\rm opt}+\underline{d})\}\,/\,\{Q(\underline{x}_{\rm opt})-Q(\underline{x}_{\rm opt}+\underline{d})\}, \qquad (2.2)$$

as described in Section 7. The other task of Box 4 is to pick the m interpolation points of the next quadratic model. Usually one of the current points $\underline{x}_i$, $i=1,2,\ldots,m$, is replaced by $\underline{x}_{\rm opt}+\underline{d}$, and all the other points are retained. In this case the integer MOVE is set in Box 4 to the index of the interpolation point that is dropped. The only other possibility is no change to the interpolation equations, and then MOVE is set to zero. Details of the choice of MOVE are also given in Section 7, the case $\texttt{MOVE}>0$ being mandatory when the strict reduction $F(\underline{x}_{\rm opt}+\underline{d})<F(\underline{x}_{\rm opt})$ is achieved, in order that the best calculated value of F so far is among the new interpolation conditions. The updating operations of Box 5 are the subject of Section 4. Box 6 branches back to Box 2 for another trust region iteration if the ratio (2.2) is sufficiently large.

The N branch is taken from Box 6 of the figure when Box 4 has provided a change $F(\underline{x}_{\rm opt})-F(\underline{x}_{\rm opt}+\underline{d})$ in the objective function that compares unfavourably with the predicted reduction $Q(\underline{x}_{\rm opt})-Q(\underline{x}_{\rm opt}+\underline{d})$. Usually this happens because the positions of the points $\underline{x}_i$ in the interpolation equations (1.1) are unsuitable for maintaining a good quadratic model, especially when the trust region iterations have caused some of the distances $\|\underline{x}_i-\underline{x}_{\rm opt}\|$, $i=1,2,\ldots,m$, to be much greater than Δ. Therefore the purpose of Box 7 is to identify the current interpolation point, $\underline{x}_{\texttt{MOVE}}$ say, that is furthest from $\underline{x}_{\rm opt}$. We take the view that, if $\|\underline{x}_{\texttt{MOVE}}-\underline{x}_{\rm opt}\|\geq 2\Delta$ holds, then Q can be improved substantially by replacing the interpolation condition $Q(\underline{x}_{\texttt{MOVE}})=F(\underline{x}_{\texttt{MOVE}})$ by $Q(\underline{x}_{\rm opt}+\underline{d})=F(\underline{x}_{\rm opt}+\underline{d})$, for some step $\underline{d}$ that satisfies $\|\underline{d}\|\leq\Delta$. We see in the figure that the actual choice of $\underline{d}$ is made in Box 9, details being given in Section 6, because they depend on the updating formulae of Section 4. Then Box 5 is reached from Box 9, in order to update Q as before, after the calculation of the new function value $F(\underline{x}_{\rm opt}+\underline{d})$. In this case the branch from Box 6 to Box 2 is always followed, due to the setting of the artificial value $\texttt{RATIO}=1$ at the end of Box 9. Thus the algorithm makes use immediately of the new information in the quadratic model.

The N branch is taken from Box 8 when the positions of the current points $\underline{x}_i$, $i=1,2,\ldots,m$, are under consideration, and when they have the property

$$\|\underline{x}_i-\underline{x}_{\rm opt}\| \;<\; 2\Delta, \qquad i=1,2,\ldots,m. \qquad (2.3)$$

Then the tests in Box 10 determine whether the work with the current value of ρ is complete. We see that the work continues if and only if one or more of the conditions $\|\underline{d}\| > \rho$, $\Delta > \rho$ or RATIO > 0 holds. Another trust region iteration is performed with the same ρ in the first two cases, because ρ has not restricted the most recent choice of $\underline{d}$. In the third case, RATIO > 0 implies $F(\underline{x}_{\rm opt}+\underline{d}) < F(\underline{x}_{\rm opt})$ in Box 4, and we prefer to retain the old ρ while strict reductions in the objective function are being obtained. Thus an infinite loop with ρ fixed may happen in theory. In practice, however, the finite precision of the computer arithmetic provides an upper bound on the number of different values of F that can occur.

Finally, we consider the operations of Figure 1 when the step $\underline{d}$ of Box 2 satisfies $\|\underline{d}\| < \frac{1}{2}\rho$. Then Box 14 is reached from Box 3, and often $F(\underline{x}_{\rm opt}+\underline{d})$ is not going to be calculated, because, as mentioned already, the computed difference $F(\underline{x}_{\rm opt})-F(\underline{x}_{\rm opt}+\underline{d})$ tends to give misleading information about the true objective function when $\|\underline{d}\|$ becomes small. If Box 14 branches to Box 15, a big reduction is made in Δ if allowed by $\Delta \geq \rho$, and then, beginning at Box 7, there is a choice as before between replacing the interpolation point $\underline{x}_{\rm MOVE}$, or performing a trust region iteration with the new Δ, or going to Box 11 because the work with the current ρ is complete. Alternatively, we see that Box 14 can branch directly to Box 11, the reason being as follows.

Let $\underline{\hat{x}}_{\rm opt}$ and $\underline{\check{x}}_{\rm opt}$ be the first and last values of $\underline{x}_{\rm opt}$ during all the work with the current ρ, and let $\underline{\hat{x}}_i$, $i=1,2,\ldots,m$, be the interpolation points at the start of this part of the computation. When ρ is less than $\rho_{\rm beg}$, the current ρ was selected in Box 12, and, because it is much smaller than its previous value, we expect the points $\underline{\hat{x}}_i$ to satisfy $\|\underline{\hat{x}}_i - \underline{\hat{x}}_{\rm opt}\| \geq 2\rho$, $i \neq {\rm opt}$. On the other hand, because of Boxes 7 and 8 in the figure, Box 11 can be reached from Box 10 only in the case $\|\underline{x}_i - \underline{\check{x}}_{\rm opt}\| < 2\rho$, $i=1,2,\ldots,m$. These remarks suggest that at least $m-1$ new values of the objective function may be calculated for the current ρ. It is important to efficiency, however, to include a less laborious route to Box 11, especially when m is large and $\rho_{\rm end}$ is tiny. Details of the tests that pick the Y branch from Box 14 are given in Section 7. They are based on the assumption that there is no need for further improvements to the model Q, if the differences $|F(\underline{x}_{\rm opt}+\underline{d})-Q(\underline{x}_{\rm opt}+\underline{d})|$ of recent iterations compare favourably with the current second derivative term $\frac{1}{8}\rho^2$CRVMIN.

When the Y branch is taken from Box 14, we let $\underline{d}_{\rm old}$ be the vector $\underline{d}$ that has satisfied $\|\underline{d}\| < \frac{1}{2}\rho$ in Box 3 of the current iteration. Often $\underline{d}_{\rm old}$ is an excellent step to take from $\underline{x}_{\rm opt}$ in the space of the variables, so we wish to allow its use after leaving Box 11. If Box 2 is reached from Box 11 via Box 12, then $\underline{d} = \underline{d}_{\rm old}$ is generated again, because the quadratic model is the same as before, and the change to Δ in Box 12 preserves the property $\Delta \geq \frac{1}{2}\rho_{\rm old} > \|\underline{d}_{\rm old}\|$. Alternatively, if the Y branches are taken from Boxes 14 and 11, we see in Box 13 that $F(\underline{x}_{\rm opt}+\underline{d}_{\rm old})$ is computed. The NEWUOA software returns to the user the first vector of variables that gives the least of the calculated values of the objective function.

3. The initial calculations

We write the quadratic model of the first iteration in the form

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

$\underline{x}_0$ being the initial vector of variables that is provided by the user. When the number of interpolation conditions (1.1) satisfies $m\geq 2n+1$, the first $2n+1$ of the points $\underline{x}_i$, $i=1,2,\ldots,m$, are chosen to be the vectors

$$\underline{x}_1=\underline{x}_0 \qquad \text{and} \qquad \left.\begin{array}{r}\underline{x}_{i+1}=\underline{x}_0+\rho_{\rm beg}\,\underline{e}_i\\ \underline{x}_{i+n+1}=\underline{x}_0-\rho_{\rm beg}\,\underline{e}_i\end{array}\right\}, \quad i=1,2,\ldots,n, \tag{3.2}$$

where $\rho_{\rm beg}$ is also provided by the user as mentioned already, and where $\underline{e}_i$ is the i-th coordinate vector in $\mathcal{R}^n$. Thus $Q(\underline{x}_0)$, $\underline{\nabla}Q(\underline{x}_0)$ and the diagonal elements $(\nabla^2Q)_{ii}$, $i=1,2,\ldots,n$, are given uniquely by the first $2n+1$ of the equations (1.1). Alternatively, when m satisfies $n+2\leq m\leq 2n$, the initial interpolation points are the first m of the vectors (3.2). It follows that $Q(\underline{x}_0)$, the first $m-n-1$ components of $\underline{\nabla}Q(\underline{x}_0)$ and $(\nabla^2Q)_{ii}$, $i=1,2,\ldots,m-n-1$, are defined as before. The other diagonal elements of ∇^2Q are set to zero, so the other components of $\underline{\nabla}Q(\underline{x}_0)$ take the values $\{F(\underline{x}_0+\rho_{\rm beg}\,\underline{e}_i)-F(\underline{x}_0)\}/\rho_{\rm beg}$, $m-n\leq i\leq n$.

In the case $m>2n+1$, the initial points $\underline{x}_i$, $i=1,2,\ldots,m$, are chosen so that the conditions (1.1) also provide $2(m-2n-1)$ off-diagonal elements of ∇^2Q, the factor 2 being due to symmetry. Specifically, for $i\in[2n+2,m]$, the point $\underline{x}_i$ has the form

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

where p and q are different integers from $[1,n]$, and where σ_p and σ_q are included in the definitions

$$\sigma_j \;=\; \left\{\begin{array}{ll}-1, & F(\underline{x}_0-\rho_{\rm beg}\,\underline{e}_j)<F(\underline{x}_0+\rho_{\rm beg}\,\underline{e}_j)\\ +1, & F(\underline{x}_0-\rho_{\rm beg}\,\underline{e}_j)\geq F(\underline{x}_0+\rho_{\rm beg}\,\underline{e}_j),\end{array}\right. \qquad j=1,2,\ldots,n, \tag{3.4}$$

which biases the choice (3.3) towards smaller values of the objective function. Thus the element $(\nabla^2Q)_{pq}=(\nabla^2Q)_{qp}$ is given by the equations (1.1), since every quadratic function $Q(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, has the property

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

For simplicity, we pick p and q in formula (3.3) in the following way. We let j be the integer part of the quotient $(i-n-2)/n$, which satisfies $j\geq 1$ due to $i\geq 2n+2$, we set $p=i-n-1-jn$, which is in the interval $[1,n]$, and we let q have the value $p+j$ or $p+j-n$, the latter choice being made in the case $p+j>n$. Hence, if $n=5$ and $m=20$, for example, there are 9 pairs $\{p,q\}$, generated in the order $\{1,2\}$, $\{2,3\}$, $\{3,4\}$, $\{4,5\}$, $\{5,1\}$, $\{1,3\}$, $\{2,4\}$, $\{3,5\}$ and $\{4,1\}$. All the off-diagonal

elements of $\nabla^2 Q$ that are not provided by the method of this paragraph are set to zero, which completes the specification of the initial quadratic model (3.1).

The preliminary work of NEWUOA includes also the setting of the initial matrix $H = W^{-1}$, where W occurs in the linear system of equations that defines the change to the quadratic model. We recall from Section 1 that, when Q is updated from $Q_{\rm old}$ to $Q_{\rm new} = Q_{\rm old} + D$, say, the quadratic function D is constructed so that $\|\nabla^2 D\|_F^2$ is least subject to the constraints

$$D(\underline{x}_i) \;=\; F(\underline{x}_i) - Q_{\rm old}(\underline{x}_i), \qquad i = 1, 2, \ldots, m, \tag{3.6}$$

these constraints being equivalent to $Q_{\rm new}(\underline{x}_i) = F(\underline{x}_i)$, $i = 1, 2, \ldots, m$. We see that the calculation of D is a quadratic programming problem, and we let λ_j, $j = 1, 2, \ldots, m$, be the Lagrange multipliers of its KKT conditions. They have the properties

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

and the second derivative matrix of D takes the form

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

(Powell, 2004a), the last part of expression (3.8) being a consequence of the equations (3.7). This form of $\nabla^2 D$ allows D to be the function

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

and we seek the values of the parameters $c \in \mathcal{R}$, $\underline{g} \in \mathcal{R}^n$ and $\underline{\lambda} \in \mathcal{R}^m$. The conditions (3.6) and (3.7) give the square system of linear equations

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

where A has the elements

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

where X is the matrix (1.3), and where $\underline{r}$ has the components $F(\underline{x}_i) - Q_{\rm old}(\underline{x}_i)$, $i = 1, 2, \ldots, m$. Therefore W and H are the matrices

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

say. It is straightforward to derive the elements of W from the vectors $\underline{x}_i$, $i = 1, 2, \ldots, m$, but we require the elements of Ξ and Υ explicitly, with a factorization of Ω. Fortunately, the chosen positions of the initial interpolation points

provide convenient formulae for all of these terms, as stated below. Proofs of the correctness of the formulae are given in the appendix.

The first row of the initial $(n+1)\times m$ matrix Ξ has the very simple form

$$\Xi_{1j} \;=\; \delta_{1j}, \qquad j=1,2,\ldots,m. \tag{3.13}$$

Further, for integers i that satisfy $2\le i\le \min[n+1, m-n]$, the i-th row of Ξ has the nonzero elements

$$\Xi_{ii} \;=\; (2\,\rho_{\rm beg})^{-1} \qquad \text{and} \qquad \Xi_{i\,i+n} \;=\; -(2\,\rho_{\rm beg})^{-1}, \tag{3.14}$$

all the other entries being zero, which defines the initial Ξ in the cases $m\ge 2n+1$. Otherwise, when $m-n+1\le i\le n+1$ holds, the i-th row of the initial Ξ also has just two nonzero elements, namely the values

$$\Xi_{i1} \;=\; -(\rho_{\rm beg})^{-1} \qquad \text{and} \qquad \Xi_{ii} \;=\; (\rho_{\rm beg})^{-1}, \tag{3.15}$$

which completes the definition of Ξ for the given interpolation points. Moreover, the initial $(n+1)\times(n+1)$ matrix Υ is amazingly sparse, being identically zero in the cases $m\ge 2n+1$. Otherwise, its only nonzero elements are the last $2n-m+1$ diagonal entries, which take the values

$$\Upsilon_{ii} \;=\; -\tfrac{1}{2}\,\rho_{\rm beg}^2, \qquad m-n+1\le i\le n+1. \tag{3.16}$$

The factorization of Ω, mentioned in Section 1, guarantees that the rank of Ω is at most $m-n-1$, by having the form

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

the second equation being valid because each s_k is set to one initially. When $1\le k\le \min[n, m-n-1]$, the components of the initial vector $\underline{z}_k\in\mathcal{R}^m$, which is the k-th column of Z, are given the values

$$\left.\begin{array}{ll} Z_{1k}=-\sqrt{2}\,\rho_{\rm beg}^{-2}, & Z_{k+1\,k}=\frac{1}{2}\sqrt{2}\,\rho_{\rm beg}^{-2}, \\ Z_{k+n+1\,k}=\frac{1}{2}\sqrt{2}\,\rho_{\rm beg}^{-2}, & Z_{jk}=0 \;\;\text{otherwise,} \end{array}\right\} \tag{3.18}$$

so each of these columns has just three nonzero elements. Alternatively, when $m>2n+1$ and $n+1\le k\le m-n-1$, the initial $\underline{z}_k$ depends on the choice (3.3) of $\underline{x}_i$ in the case $i=k+n+1$. We let p, q, σ_p and σ_q be as before, and we define $\hat{p}$ and $\hat{q}$ by the equations

$$\underline{x}_{\hat{p}} \;=\; \underline{x}_0 + \sigma_p\,\rho_{\rm beg}\,\underline{e}_p \qquad \text{and} \qquad \underline{x}_{\hat{q}} \;=\; \underline{x}_0 + \sigma_q\,\rho_{\rm beg}\,\underline{e}_q. \tag{3.19}$$

It follows from the positions of the interpolation points that $\hat{p}$ is either $p+1$ or $p+n+1$, while $\hat{q}$ is either $q+1$ or $q+n+1$. Now there are four nonzero elements in the k-th column of Z, the initial $\underline{z}_k$ being given the components

$$\left.\begin{array}{ll} Z_{1k}=\rho_{\rm beg}^{-2}, & Z_{\hat{p}k}=Z_{\hat{q}k}=-\rho_{\rm beg}^{-2}, \\ Z_{k+n+1\,k}=\rho_{\rm beg}^{-2}, & Z_{jk}=0 \;\;\text{otherwise.} \end{array}\right\} \tag{3.20}$$

All the given formulae for the nonzero elements of $H=W^{-1}$ are applied in only $\mathcal{O}(m)$ operations, due to the convenient choice of the initial interpolation points, but the work of setting the zero elements of Ξ, Υ and Z is $\mathcal{O}(m^2)$. The description of the preliminary work of NEWUOA is complete.

4. The updating procedures

In this section we consider the change that is made to the quadratic model Q on each iteration of NEWUOA that alters the set of interpolation points. We let the new points have the positions

$$\left.\begin{array}{rcll} \underline{x}_t^+ &=& \underline{x}_{\rm opt}+\underline{d} \,=\, \underline{x}^+, & \text{say,} \\ \underline{x}_i^+ &=& \underline{x}_i, & i\in\{1,2,\ldots,m\}\backslash\{t\}, \end{array}\right\} \tag{4.1}$$

which agrees with the outline of the method in Figure 1, because now we write t instead of `MOVE`. The change $D=Q_{\rm new}-Q_{\rm old}$ has to satisfy the analogue of the conditions (3.6) for the new points, and $Q_{\rm old}$ interpolates F at the old interpolation points. Thus D is the quadratic function that minimizes $\|\nabla^2 D\|_F$ subject to the constraints

$$D(\underline{x}_i^+) \,=\, \left\{F(\underline{x}^+)-Q_{\rm old}(\underline{x}^+)\right\}\delta_{it}, \qquad i=1,2,\ldots,m. \tag{4.2}$$

Let W^+ and H^+ be the matrices

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

where A^+ and X^+ are defined by replacing the old interpolation points by the new ones in equations (1.3) and (3.11). It follows from the derivation of the system (3.10) and from the conditions (4.2) that D is now the function

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

the parameters being the components of the vector

$$\left(\begin{array}{c} \underline{\lambda}^+ \\ \hline c^+ \\ \underline{g}^+ \end{array}\right) \,=\, \left\{F(\underline{x}^+)-Q_{\rm old}(\underline{x}^+)\right\}H^+\underline{e}_t, \tag{4.5}$$

where $\underline{e}_t$ is now in $\mathcal{R}^{m+n+1}$. Expressions (4.5) and (4.4) are used by the NEWUOA software to generate the function D for the updating formula

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

The matrix $H=W^{-1}$ is available at the beginning of the current iteration, the submatrices Ξ and Υ being stored explicitly, with the factorization $\sum_{k=1}^{m-n-1} s_k\underline{z}_k\underline{z}_k^T$

of Ω that has been mentioned, but H^+ occurs in equation (4.5). Therefore Ξ and Υ are overwritten by the submatrices Ξ^+ and Υ^+ of expression (4.3), and also the new factorization

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

is required. Fortunately, the amount of work of these tasks is only $\mathcal{O}(m^2)$ operations, by taking advantage of the simple change (4.1) to the interpolation points. Indeed, we deduce from equations (4.1), (1.3), (3.11), (3.12) and (4.3) that all differences between the elements of W and W^+ are confined to the t-th row and column. Thus $W^+ - W$ is a matrix of rank two, which implies that the rank of $H^+ - H$ is also two. Therefore Ξ^+, Υ^+ and the factorization (4.7) are constructed from H by an extension of the Sherman–Morrison formula. Details and some relevant analysis are given in Powell (2004c), so only a brief outline of these calculations is presented below, before considering the implementation of formula (4.6). The updating of H in $\mathcal{O}(m^2)$ operations is highly important to the efficiency of the NEWUOA software, since an *ab initio* calculation of the change (4.4) to the quadratic model would require $\mathcal{O}(m^3)$ computer operations.

In theory, H^+ is the inverse of the matrix W^+ that has the elements

$$\left.\begin{array}{l} W_{it}^+ = W_{ti}^+ = (W^+ \underline{e}_t)_i, \quad i = 1, 2, \ldots, m+n+1, \\ W_{ij}^+ = W_{ij} = H_{ij}^{-1}, \quad \text{otherwise}, \quad 1 \le i, j \le m+n+1. \end{array}\right\} \tag{4.8}$$

It follows from the right hand sides of this expression that H and the t-th column of W^+ provide enough information for the derivation of H^+. The definitions (1.3) and (3.11) show that $W^+ \underline{e}_t$ has the components

$$\left.\begin{array}{ll} W_{it}^+ \;=\; \frac{1}{2} \{(\underline{x}_i^+ - \underline{x}_0)^T (\underline{x}^+ - \underline{x}_0)\}^2, & i = 1, 2, \ldots, m \\ W_{m+1\,t}^+ = 1 \quad \text{and} \quad W_{i+m+1\,t}^+ = (\underline{x}^+ - \underline{x}_0)_i, & i = 1, 2, \ldots, n \end{array}\right\}, \tag{4.9}$$

the notation $\underline{x}^+$ being used instead of $\underline{x}_t^+$, because $\underline{x}^+ = \underline{x}_{\rm opt} + \underline{d}$ is available before $t = \texttt{MOVE}$ is picked in Box 4 of Figure 1. Of course t must have the property that W^+ is nonsingular, which holds if no divisions by zero occur when H^+ is calculated. Therefore we employ a formula for H^+ that gives conveniently the dependence of H^+ on t. Let the components of $\underline{w} \in \mathcal{R}^{m+n+1}$ take the values

$$\left.\begin{array}{ll} w_i \;=\; \frac{1}{2} \{(\underline{x}_i - \underline{x}_0)^T (\underline{x}^+ - \underline{x}_0)\}^2, & i = 1, 2, \ldots, m \\ w_{m+1} = 1 \quad \text{and} \quad w_{i+m+1} = (\underline{x}^+ - \underline{x}_0)_i, & i = 1, 2, \ldots, n \end{array}\right\}, \tag{4.10}$$

so $\underline{w}$ is independent of t. Equations (4.1), (4.9) and (4.10) imply that $W^+ \underline{e}_t$ differs from $\underline{w}$ only in its t-th component, which allows H^+ to be written in terms of H, $\underline{w}$ and $\underline{e}_t$. Specifically, Powell (2004a) derives the formula

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

the parameters being the expressions

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

We see that $H\underline{w}$ and β can be calculated before t is chosen, so it is inexpensive in practice to investigate the dependence of the denominator σ on t, in order to ensure that $|\sigma|$ is sufficiently large. The actual selection of t is addressed in Section 7.

Formula (4.11) was applied by an early version of NEWUOA, before the introduction of the factorization of Ω. The bottom left $(n+1)\times m$ and bottom right $(n+1)\times(n+1)$ submatrices of this formula are still used to construct Ξ^+ and Υ^+ from Ξ and Υ, respectively, the calculation of $H\underline{w}$ and $H\underline{e}_t$ being straightforward when the terms s_k and $\underline{z}_k$, $k=1,2,\ldots,m-n-1$, of the factorization (3.17) are stored instead of Ω.

The purpose of the factorization is to reduce the damage from rounding errors to the identity $W=H^{-1}$, which holds in theory at the beginning of each iteration. It became obvious from numerical experiments, however, that huge errors may occur in H in practice, including a few negative values of H_{ii}, $1\le i\le m$, although Ω should be positive semi-definite. Therefore we consider the updating of H when H is very different from W^{-1}, assuming that the calculations of the current iteration are exact. Then H^+ is the inverse of the matrix that has the elements on the right hand side of expression (4.8), which gives the identities

$$\left.\begin{array}{l} (H^+)^{-1}_{it} = W^+_{it} \quad \text{and} \quad (H^+)^{-1}_{ti} = W^+_{ti}, \quad i=1,2,\ldots,m+n+1, \\ W^+_{ij} - (H^+)^{-1}_{ij} = W_{ij} - H^{-1}_{ij}, \quad \text{otherwise}, \quad 1\le i,j\le m+n+1. \end{array}\right\} \tag{4.13}$$

In other words, the overwriting of W and H by W^+ and H^+ makes no difference to the elements of $W-H^{-1}$, except that the t-th row and column of this error matrix become zero. It follows that, when all the current interpolation points have been discarded by future iterations, then all the current errors in the first m rows and columns of $W-H^{-1}$ will have been annihilated. Equation (4.13) suggests, however, that any errors in the bottom right $(n+1)\times(n+1)$ submatrix of H^{-1} are retained. The factorization (3.17) provides the perfect remedy to this situation. Indeed, if H is any nonsingular $(m+n+1)\times(m+n+1)$ matrix of the form (3.12), and if the rank of the leading $m\times m$ submatrix Ω is $m-n-1$, then the bottom right $(n+1)\times(n+1)$ submatrix of H^{-1} is zero, which can be proved by expressing the elements of H^{-1} as cofactors of H divided by $\det H$ (Powell, 2004c). Thus the very welcome property

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

is guaranteed by the factorization (4.7), even in the presence of computer rounding errors.

The updating of the factorization of Ω by NEWUOA depends on the fact that the values

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

are suitable in expression (4.7), where k is in $\mathcal{K}$ if and only if the t-th component of $\underline{z}_k$ is zero. Before taking advantage of this fact, an elementary change is made if necessary to the terms of the sum

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

which forces the number of integers in $\mathcal{K}$ to be at least $m-n-3$. Specifically, NEWUOA employs the remark that, when $s_i = s_j$ holds in equation (4.16), then the equation remains true if $\underline{z}_i$ and $\underline{z}_j$ are replaced by the vectors

$$\cos\theta \, \underline{z}_i + \sin\theta \, \underline{z}_j \qquad \text{and} \qquad -\sin\theta \, \underline{z}_i + \cos\theta \, \underline{z}_j, \tag{4.17}$$

respectively, for any $\theta \in [0, 2\pi]$. The choice of θ allows either i or j to be added to $\mathcal{K}$ if both i and j were not in $\mathcal{K}$ previously. Thus, because $s_k = \pm 1$ holds for each k, only one or two of the new vectors $\underline{z}_k^+$, $k = 1, 2, \ldots, m-n-1$, have to be calculated after retaining the values (4.15). When $|\mathcal{K}| = m-n-2$, we let $\underline{z}_{m-n-1}^+$ be the required new vector, which is the usual situation as the theoretical positive definiteness of Ω should exclude negative values of s_k. Then the last term of the new factorization (4.7) is defined by the equations

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

where τ, σ and $\underline{e}_t - H \, \underline{w}$ are taken from the updating formula (4.11), where $Z_{t\,m-n-1}$ is the t-th component of $\underline{z}_{m-n-1}$, and where $\text{chop}\,(\underline{e}_t - H \, \underline{w})$ is the vector in $\mathcal{R}^m$ whose components are the first m components of $\underline{e}_t - H \, \underline{w}$. These assertions and those of the next paragraph are justified in Powell (2003c).

In the alternative case $|\mathcal{K}| = m–n–3$, we simplify the notation by assuming that only $\underline{z}_1^+$ and $\underline{z}_2^+$ are not provided by equation (4.15), and that the signs $s_1 = +1$ and $s_2 = -1$ occur. Then the t-th components of $\underline{z}_1$ and $\underline{z}_2$, namely Z_{t1} and Z_{t2}, are nonzero. Many choices of the required new vectors $\underline{z}_1^+$ and $\underline{z}_2^+$ are possible, because of the freedom that corresponds to the orthogonal rotation (4.17). We make two of them available to NEWUOA, in order to avoid cancellation. Specifically, if the parameter β of expression (4.12) is nonnegative, we define $\zeta = \tau^2 + \beta \, Z_{t1}^2$, and NEWUOA applies the formulae

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

Otherwise, when $\beta<0$, we define $\zeta=\tau^2-\beta\,Z_{t2}^2$, and NEWUOA sets the values

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

The technique of the previous paragraph is employed only if at least one of the signs s_k, $k=1,2,\ldots,m-n-1$, is negative, and then $\sigma<0$ must have occurred in equation (4.18) on an earlier iteration, because every s_k is set to $+1$ initially. Moreover, any failure in the conditions $\alpha\geq 0$ and $\beta\geq 0$ is due to computer rounding errors. Therefore Powell (2004a) suggests that the parameter σ in formula (4.11) be given the value

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

instead of $\alpha\beta+\tau^2$. If the new value is different from before, however, then the new matrix (4.11) may not satisfy any of the conditions (4.8), except that the factorizations (4.16) and (4.7) ensure that the bottom right $(n+1)\times(n+1)$ submatrices of H^{-1} and $(H^+)^{-1}$ are zero. Another way of keeping σ positive is to retain $\alpha=\underline{e}_t^T H\underline{e}_t$, $\tau=\underline{e}_t^T H\underline{w}$ and $\sigma=\alpha\beta+\tau^2$ from expression (4.12), and to define β by the formula

$$\beta_{\mathrm{new}} \;=\; \max\Big[\,0,\;\tfrac{1}{2}\,\|\underline{x}^+ - \underline{x}_0\|^4 - \underline{w}^T H\,\underline{w}\,\Big]. \tag{4.22}$$

In this case any change to β alters the element $(H^+)^{-1}_{tt}$, but every other stability property (4.13) is preserved, as proved in Lemma 2.3 of Powell (2004c). Therefore equation (4.21) was abandoned, and the usefulness of the value (4.22) instead of the definition (4.12) of β was investigated experimentally. Substantial differences in the numerical results were found only when the damage from rounding errors was huge, and then the recovery that is provided by *all* of the conditions (4.13) is important to efficiency. Therefore the procedures that have been described already for updating Ξ, Υ and the factorization of Ω are preferred, although in practice α, β, σ and some of the signs s_k may become negative occasionally. Such errors are usually corrected automatically by a few more iterations of NEWUOA.

Another feature of the storage and updating of H by NEWUOA takes advantage of the remark that, when $\underline{d}$ is calculated in Box 2 of Figure 1, the constant term of Q is irrelevant. Moreover, the constant term of Q_{old} is not required in equation (4.5), because the identities $Q_{\mathrm{old}}(\underline{x}_{\mathrm{opt}})=F(\underline{x}_{\mathrm{opt}})$ and $\underline{x}^+=\underline{x}_{\mathrm{opt}}+\underline{d}$ allow this equation to be written in the form

$$\left(\begin{array}{c} \underline{\lambda}^+ \\ \hline c^+ \\ \underline{g}^+ \end{array}\right) \;=\; \Big\{[\,F(\underline{x}_{\mathrm{opt}}+\underline{d}) - F(\underline{x}_{\mathrm{opt}})\,] - [\,Q_{\mathrm{old}}(\underline{x}_{\mathrm{opt}}+\underline{d}) - Q_{\mathrm{old}}(\underline{x}_{\mathrm{opt}})\,]\Big\}\,H^+\underline{e}_t. \tag{4.23}$$

Therefore NEWUOA does not store the constant term of any quadratic model. It follows that c^+ in expression (4.23) is ignored, which makes the $(m+1)$-th row

of H^+ unnecessary for the revision of Q by formula (4.6). Equation (4.23) shows that the $(m+1)$-th column of H^+ is also unnecessary, t being in the interval $[1,m]$. Actually, the $(m+1)$-th row and column of every H matrix are suppressed by NEWUOA, which is equivalent to removing the first row of every submatrix Ξ and the first row and column of every submatrix Υ, but the other elements of these submatrices are retained. Usually this device gains some accuracy by diverting attention from actual values of F and $\underline{x}\in\mathcal{R}^n$ to the changes that occur in the objective function and the variables, as shown on the right hand side of equation (4.23) for example. The following procedure is used by NEWUOA to update H without its $(m+1)$-th row and column.

Let "opt" be the integer in $[1,m]$ such that $i=\text{opt}$ gives the best of the interpolation points $\underline{x}_i$, $i=1,2,\ldots,m$, which agrees with the notation in Sections 1 and 2, and let $\underline{v}\in\mathcal{R}^{m+n+1}$ have the components

$$\left.\begin{array}{ll} v_i \;=\; \frac{1}{2}\,\{(\underline{x}_i-\underline{x}_0)^T(\underline{x}_{\rm opt}-\underline{x}_0)\}^2, & i=1,2,\ldots,m \\ v_{m+1}=1 \quad\text{and}\quad v_{i+m+1}=(\underline{x}_{\rm opt}-\underline{x}_0)_i, & i=1,2,\ldots,n \end{array}\right\}. \tag{4.24}$$

Therefore $\underline{v}$ is the opt-th column of the matrix W, so expression (3.12) implies $H\underline{v}=\underline{e}_{\rm opt}$ in theory, where $\underline{e}_{\rm opt}$ is the opt-th coordinate vector in $\mathcal{R}^{m+n+1}$. Thus the terms $H\underline{w}$ and $\underline{w}^TH\underline{w}$ of equations (4.11) and (4.12) take the values

$$H\,\underline{w} \;=\; H\,(\underline{w}-\underline{v})+\underline{e}_{\rm opt} \tag{4.25}$$

and

$$\underline{w}^T\!H\,\underline{w} \;=\; (\underline{w}-\underline{v})^TH\,(\underline{w}-\underline{v})+2\,\underline{w}^T\underline{e}_{\rm opt}-\underline{v}^T\underline{e}_{\rm opt}. \tag{4.26}$$

These formulae allow the parameters (4.12) to be calculated without the $(m+1)$-th row and column of H, because the $(m+1)$-th component of $\underline{w}-\underline{v}$ is zero. Similarly, the first m and last n components of $H\underline{w}$ are given by formula (4.25), and these components of $H\underline{e}_t$ are known. Thus all the terms of expression (4.11) are available for generating the required parts of Ξ^+ and Υ^+. Moreover, after constructing $\text{chop}\,(\underline{e}_t-H\underline{w})$, the updating of the factorization of Ω is unchanged. It is proved in Lemma 3 of Powell (2004a) that, when this version of the updating procedure is applied, and when H has been damaged by rounding errors, then the new H^+ enjoys stability properties that are analogous to the conditions (4.13).

We see that the given procedures for updating H require only $\mathcal{O}(m^2)$ computer operations, which is highly favourable in the recommended case $m=2n+1$. On the other hand, the function (4.4) has the second derivative matrix

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

so the calculation of its elements would take $\mathcal{O}(mn^2)$ operations. Therefore $\nabla^2Q_{\rm new}$ is not derived explicitly from formula (4.6). Instead, as suggested at the end of Section 3 of Powell (2004a), the NEWUOA software employs the forms

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

overwriting the symmetric matrix Γ and the real coefficients γ_j, $j=1,2,\ldots,m$, by Γ^+ and γ_j^+, $j=1,2,\ldots,m$, respectively. At the beginning of the first iteration, each γ_j is set to zero, and we let Γ be the second derivative matrix of the initial quadratic model, its elements being specified in the first two paragraphs of Section 3. When the change (4.6) is made to the quadratic model, conditions (4.1), (4.27) and (4.28) allow the choices

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

which are included in NEWUOA, because they can be implemented in only $\mathcal{O}(n^2)$ operations. Finally, the gradient of the quadratic model (3.1) is revised by the formula

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

in accordance with expressions (4.4) and (4.6), where $\underline{g}^+$ is taken from equation (4.23). The description of the updating of Q, without the unnecessary constant term $Q(\underline{x}_0)$, is complete, except that some of the numerical results of Section 8 suggested a recent modification that is described there.

5. The trust region subproblem

We recall from Box 2 of Figure 1 that subroutine `TRSAPP` generates a step $\underline{d}$ from $\underline{x}_{\rm opt}$ that is an approximate solution of the subproblem

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

The method of the subroutine is explained below. Figure 1 shows that the trust region radius Δ and the quadratic model Q are available when the subroutine is called, but, as mentioned at the end of Section 4, the matrix $\nabla^2 Q$ is stored in the form

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

because it would be too onerous to work with all the elements of $\nabla^2 Q$ explicitly when n is large. Expression (5.2) implies the identity

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

where $\eta_k=\gamma_k\,(\underline{x}_k-\underline{x}_0)^T\underline{u}$, $k=1,2,\ldots,m$, and where $\underline{u}$ is a general vector in $\mathcal{R}^n$. Thus the product $\nabla^2 Q\,\underline{u}$ can be calculated in $\mathcal{O}(mn)$ operations for any choice of $\underline{u}$. Therefore it is suitable to generate $\underline{d}$ by a version of the truncated conjugate gradient method (see Conn, Gould and Toint, 2000, for instance).

This method produces a piecewise linear path in $\mathcal{R}^n$, starting at $\underline{x}_{\rm opt}=\underline{x}_{\rm opt}+\underline{d}_0$, where $\underline{d}_0=0$. For $j\geq 1$, we let $\underline{x}_{\rm opt}+\underline{d}_j$ be the point on the path at the end of the j-th line segment. It has the form

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

where $\underline{s}_j$ is the direction of the line segment and α_j is now a steplength. We do not include any preconditioning, because the norm of the bound $\|\underline{d}\|\leq\Delta$ in expression (5.1) is Euclidean. Moreover, the path is truncated at $\underline{x}_{\rm opt}+\underline{d}_{j-1}$ if $\|\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})\|$ is sufficiently small, if $\|\underline{d}_{j-1}\|=\Delta$ holds, or if some other test is satisfied, as specified later. The complete path has the property that, if one moves along it from $\underline{x}_{\rm opt}$, then the Euclidean distance from $\underline{x}_{\rm opt}$ in $\mathcal{R}^n$ increases monotonically.

When the j-th line segment of the path is constructed, its direction is defined by the formula

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

where β_j is the ratio $\|\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})\|^2/\|\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-2})\|^2$, this convenient value being taken from Fletcher and Reeves (1964). Then the steplength α_j of equation (5.4) is chosen to minimize $Q(\underline{x}_{\rm opt}+\underline{d}_j)$ subject to $\alpha_j\geq 0$ and $\|\underline{d}_j\|\leq\Delta$ for each j. Formula (5.5) provides the well-known descent condition

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

which depends on the choice of α_{j-1} when $j\geq 2$. It follows from $\|\underline{d}_{j-1}\|<\Delta$ that α_j is positive.

The form (5.3) of the product $\nabla^2 Q\,\underline{u}$ assists the calculation of the gradients $\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_j)$, $j\geq 0$, and the steplengths α_j, $j\geq 1$. The initial vector $\underline{u}$ is the difference $\underline{x}_{\rm opt}-\underline{x}_0$, in order to obtain from expression (3.1) the gradient

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

The other choices of $\underline{u}$ are just all the vectors (5.5) that occur. The availability of $\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ and $\nabla^2 Q\,\underline{s}_j$ allows α_j to be found cheaply, because it is the value of α in the interval $[0,\hat{\alpha}_j]$ that minimizes the function

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

where $\hat{\alpha}_j$ is the positive root of the equation $\|\underline{x}_{\rm opt}+\underline{d}_{j-1}+\hat{\alpha}_j\underline{s}_j\|=\Delta$. Therefore we ask whether $Q(\underline{x}_{\rm opt}+\underline{d}_{j-1}+\alpha\underline{s}_j)$, $0\leq\alpha\leq\hat{\alpha}_j$, decreases monotonically. Equations (5.6) and (5.8) imply that the answer is affirmative in the case

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

and then $\alpha_j = \hat{\alpha}_j$ is selected. Otherwise, $\underline{s}_j^T \nabla^2 Q \, \underline{s}_j$ is positive, and the subroutine picks the value

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

After finding α_j, the gradient $\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_j)$ is constructed by the formula

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

which is derived from the relation (5.4), the product $\nabla^2 Q \, \underline{s}_j$ being employed again. The techniques of this paragraph are applied for each line segment of the path.

The path is truncated at $\underline{x}_{\rm opt} + \underline{d}_j$ in the case $\alpha_j = \hat{\alpha}_j$, because then $\underline{d} = \underline{d}_j$ is on the boundary of the trust region $\|\underline{d}\| \le \Delta$. Moreover, it is truncated at its starting point $\underline{x}_{\rm opt} + \underline{d}_0 = \underline{x}_{\rm opt}$ in the unusual case when the initial gradient $\underline{\nabla} Q(\underline{x}_{\rm opt})$ is identically zero. Otherwise, we try to truncate the path when the ratio

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

is sufficiently close to one, in order to avoid conjugate gradient iterations that improve only slightly the reduction in the objective function that is predicted by the quadratic model. The implementation of this aim is empirical. Specifically, the iterations are terminated if at least one of the conditions

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

is satisfied, the change in Q for each line segment being derived from expression (5.8), and $Q(\underline{x}_{\rm opt}) - Q(\underline{x}_{\rm opt} + \underline{d}_j)$ is the sum of the changes so far. The path is also truncated if j reaches the theoretical upper bound on the number of iterations, namely n, but we expect this test to be redundant for $n \ge 10$.

Let $\underline{x}_{\rm opt} + \underline{d}_j$ be the final point of the path. The step $\underline{d} = \underline{d}_j$ is returned by subroutine `TRSAPP` in the case $\|\underline{d}_j\| < \Delta$, because then there is no interference with the conjugate gradient iterations from the trust region boundary. Further, the parameter `CRVMIN`, introduced in Box 2 of Figure 1, is given the value

$$\texttt{CRVMIN} \;=\; \min \Big\{ \, \underline{s}_i^T \nabla^2 Q \, \underline{s}_i \,/\, \|\underline{s}_i\|^2 : i = 1, 2, \ldots, j \, \Big\}. \tag{5.14}$$

Otherwise, `CRVMIN` is set to zero, and, because of the possibility that the ratio (5.12) may be substantially less than one, the following iterative procedure is applied. It also calculates $\underline{d}_j$ from $\underline{d}_{j-1}$, the initial point $\underline{x}_{\rm opt} + \underline{d}_{j-1}$ being the final point of the truncated piecewise linear path, so $\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1})$ is available. The conditions $\|\underline{d}_j\| = \|\underline{d}_{j-1}\| = \Delta$ are going to hold on every iteration of the additional procedure.

At the beginning of an iteration, we decide, using only $\underline{d}_{j-1}$ and $\underline{\nabla} Q(\underline{x}_{\rm opt} + \underline{d}_{j-1})$, whether $\underline{d} = \underline{d}_{j-1}$ is acceptable as an approximate solution of the subproblem (5.1). If $\underline{d}_{j-1}$ were the true solution, then, by the KKT conditions of the subproblem,

$\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ would be a nonpositive multiple of $\underline{d}_{j-1}$, and we also give attention to the first of the conditions (5.13). Indeed, subroutine TRSAPP picks $\underline{d}=\underline{d}_{j-1}$ if one or both of the inequalities

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

is achieved. Otherwise, $\underline{d}_{j-1}$ and $\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ span a two dimensional subspace of $\mathcal{R}^n$, and $\underline{d}_j$ is calculated to be the vector in this subspace that minimizes $Q(\underline{x}_{\rm opt}+\underline{d}_j)$ subject to $\|\underline{d}_j\|=\Delta$. Therefore $\underline{d}_j$ has the form

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

where now the search direction $\underline{s}_j$ is chosen to be a vector in the two dimensional subspace that has the properties

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

Equation (5.16) implies that $Q(\underline{x}_{\rm opt}+\underline{d}(\theta))$ is the expression

$$\begin{aligned}Q(\underline{x}_{\rm opt})+\Big(\cos\theta\,\underline{d}_{j-1}+\sin\theta\,\underline{s}_j\Big)^T\underline{\nabla}Q(\underline{x}_{\rm opt})+\Big(\tfrac{1}{2}\cos^2\theta\,\underline{d}_{j-1}+\cos\theta\,\sin\theta\,\underline{s}_j\Big)^T \\ \Big\{\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_{j-1})-\underline{\nabla}Q(\underline{x}_{\rm opt})\Big\}+\tfrac{1}{2}\sin^2\theta\,\underline{s}_j^T\nabla^2Q\,\underline{s}_j, \quad 0\leq\theta\leq 2\pi,\end{aligned} \tag{5.18}$$

because the last term in braces is the product $\nabla^2Q\,\underline{d}_{j-1}$. Again $\nabla^2Q\,\underline{s}_j$ is constructed by formula (5.3), after which the minimization of the function (5.18) takes only $\mathcal{O}(n)$ operations. Thus $\underline{d}_j$ is determined, and the subroutine returns $\underline{d}=\underline{d}_j$ if the second of the conditions (5.13) holds, or if j is at least n. Alternatively, $\underline{\nabla}Q(\underline{x}_{\rm opt}+\underline{d}_j)$ is calculated for the next iteration, by applying the remark that equation (5.16) gives the gradient

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

Then j is increased by one, in order that the procedure of this paragraph can be applied recursively until termination occurs.

6. Subroutines BIGLAG and BIGDEN

We recall from Section 2 that, if Box 9 of Figure 1 is reached, then the condition (1.1) with index $i=$ MOVE is going to be replaced by the interpolation condition $Q(\underline{x}_{\rm opt}+\underline{d})=F(\underline{x}_{\rm opt}+\underline{d})$, where $\underline{d}$ is calculated by the procedure of this section. In theory, given the index MOVE, the choice of $\underline{d}$ is derived from the positions $\underline{x}_i$, $i=1,2,\ldots,m$, of the current interpolation points, but in practice it depends also on the errors that occur in the matrices that are stored and updated, namely the submatrices Ξ and Υ of expression (3.12) and the factorization (4.16). We

write t instead of MOVE, in order to retain the notation of Section 4. In particular, equation (4.1) shows the new positions of the interpolation points.

The t-th Lagrange function of the current interpolation points is important. It is the quadratic polynomial $\ell_t(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, that satisfies the Lagrange conditions

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

where the remaining freedom in the usual case $m<\frac{1}{2}(n+1)(n+2)$ is taken up by minimizing the Frobenius norm $\|\nabla^2\ell_t\|_F$. Therefore ℓ_t is the function

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

the parameters c, $\underline{g}$ and λ_k, $k=1,2,\ldots,m$, being defined by the linear system of equations (3.10), where the right hand side is now the coordinate vector $\underline{e}_t\in\mathcal{R}^{m+n+1}$. Thus the parameters are the elements of the t-th column of the matrix H of expression (3.12). For each $\underline{x}\in\mathcal{R}^n$, we let $\underline{w}(\underline{x})$ be the vector in $\mathcal{R}^{m+n+1}$ that has the components

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

It follows that expression (6.2) can be written in the form

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

Therefore, when the symmetric matrix H is updated by formula (4.11), because of the change (4.1) to the interpolation points, expression (4.12) includes the value

$$\tau\;=\;\underline{e}_t^T H\,\underline{w}\;=\;\underline{e}_t^T H\,\underline{w}(\underline{x}^+)\;=\;(H\,\underline{e}_t)^T\underline{w}(\underline{x}_{\rm opt}+\underline{d})\;=\;\ell_t(\underline{x}_{\rm opt}+\underline{d}). \tag{6.5}$$

Thus the Lagrange function (6.2) gives the dependence of τ on the choice of $\underline{d}$.

As mentioned in Section 4, we expect a relatively large modulus of the denominator $\sigma=\alpha\beta+\tau^2$ to be beneficial when formula (4.11) is applied. Usually $\sigma>\tau^2$ holds in practice, because in theory both α and β are positive. Thus we deduce from the previous paragraph that it may be advantageous to let $\underline{d}$ be an approximate solution of the subproblem

$$\text{Maximize}\quad|\ell_t(\underline{x}_{\rm opt}+\underline{d})|\quad\text{subject to}\quad\|\underline{d}\|\leq\overline{\Delta}, \tag{6.6}$$

where $\overline{\Delta}>0$ is prescribed. This calculation is performed by subroutine BIGLAG, details being given in the next paragraph. There is an excellent reason for a large value of $|\ell_t(\underline{x}_{\rm opt}+\underline{d})|$ in the case $m=\frac{1}{2}(n+1)(n+2)$. Specifically, one picks a convenient basis of the space of quadratic polynomials, in order that the construction of Q from the interpolation conditions (1.1) reduces to the solution of an $m\times m$ system of linear equations. Let B and B^+ be the old and new matrices

of the system when the change (4.1) is made to the interpolation points. Then, as shown in Powell (2001), the dependence of the ratio $\det B^+/\det B$ on $\underline{d}\in\mathcal{R}^n$ is just a quadratic polynomial, which is exactly $\ell_t(\underline{x}_{\rm opt}+\underline{d})$, $\underline{d}\in\mathcal{R}^n$, because of the Lagrange conditions (6.1). In this case, therefore, the subproblem (6.6) is highly suitable for promoting the nonsingularity of B^+.

The method of BIGLAG is iterative, and is like the procedure of the last paragraph of Section 5. As in equation (5.16), the j-th iteration generates the vector

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

where $\underline{d}_{j-1}$ is the best estimate of the required $\underline{d}$ at the beginning of the current iteration, where $\underline{d}_{j-1}$ and $\underline{s}_j$ have the properties

$$\|\underline{d}_{j-1}\|=\|\underline{s}_j\|=\overline{\Delta}\qquad\text{and}\qquad \underline{s}_j^T\underline{d}_{j-1}=0, \tag{6.8}$$

and where the angle θ of equation (6.7) is calculated to maximize $|\ell_t(\underline{x}_{\rm opt}+\underline{d}_j)|$. The choice

$$\underline{d}_0 \;=\; \pm\,\overline{\Delta}\,(\underline{x}_t-\underline{x}_{\rm opt})\,/\,\|\underline{x}_t-\underline{x}_{\rm opt}\| \tag{6.9}$$

is made for the first iteration, with the sign that provides the larger value of $|\ell_t(\underline{x}_{\rm opt}+\underline{d}_0)|$, which implies $\underline{\nabla}\ell_t(\underline{x}_{\rm opt}+\underline{d}_0)\neq 0$, because ℓ_t is a quadratic polynomial that satisfies the Lagrange conditions $\ell_t(\underline{x}_{\rm opt})=0$ and $\ell_t(\underline{x}_t)=1$. The vector $\underline{s}_1$ of the first iteration is taken from the two dimensional subspace that is spanned by $\underline{d}_0$ and $\underline{\nabla}\ell_t(\underline{x}_{\rm opt})$, provided that both the inequalities

$$\left.\begin{aligned} |\underline{d}_0^T\underline{\nabla}\ell_t(\underline{x}_{\rm opt})|^2 \;&\leq\; 0.99\,\overline{\Delta}^{\,2}\,\|\underline{\nabla}\ell_t(\underline{x}_{\rm opt})\|^2 \\ \text{and}\quad \|\underline{\nabla}\ell_t(\underline{x}_{\rm opt})\| \;&\geq\; 0.1\,|\ell_t(\underline{x}_{\rm opt}+\underline{d}_0)|\,/\,\overline{\Delta} \end{aligned}\right\} \tag{6.10}$$

hold, because this use of $\underline{\nabla}\ell_t(\underline{x}_{\rm opt})$ is unattractive if the subspace is nearly degenerate, or if the bound $\overline{\Delta}\,\|\underline{\nabla}\ell_t(\underline{x}_{\rm opt})\|$ on the first order term of the identity

$$\ell_t(\underline{x}_{\rm opt}+\underline{d}) \;=\; \underline{d}^T\underline{\nabla}\ell_t(\underline{x}_{\rm opt})+\tfrac{1}{2}\,\underline{d}^T\,\underline{\nabla}^2\ell_t\,\underline{d},\qquad \|\underline{d}\|\leq\overline{\Delta}, \tag{6.11}$$

compares unfavourably with $|\ell_t(\underline{x}_{\rm opt}+\underline{d}_0)|$. Alternatively, if at least one of the conditions (6.10) fails, then $\underline{s}_1$ is defined by the technique that gives $\underline{s}_j$ for $j\geq 2$. Specifically, $\underline{s}_j$ is a linear combination of $\underline{d}_{j-1}$ and $\underline{\nabla}\ell_t(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ that has the properties (6.8), except that the subroutine returns the vector $\underline{d}=\underline{d}_{j-1}$ in the unlikely situation

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

The usual test for termination is the condition

$$|\ell_t(\underline{x}_{\rm opt}+\underline{d}_j)| \;\leq\; 1.1\,|\ell_t(\underline{x}_{\rm opt}+\underline{d}_{j-1})|, \tag{6.13}$$

because the iteration has not improved very much the objective function of the subproblem (6.6). Then $\underline{d}=\underline{d}_j$ is returned, which happens too if j reaches the value n. Otherwise, as in equation (5.19), the gradient

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

is calculated, and then j is increased by one for the next iteration. Because the second derivative matrix of the function (6.2) is not constructed explicitly, the remarks on $\nabla^2 Q$ in Section 5 apply also to $\nabla^2 \ell_t$, including the use of the formula

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

where $\eta_k = \lambda_k (\underline{x}_k - \underline{x}_0)^T \underline{u}$, $k = 1, 2, \ldots, m$. Now the vectors $\underline{u}$ that occur are just $\underline{x}_{\rm opt} - \underline{x}_0$, $\underline{d}_0$ and each $\underline{s}_j$, so the amount of work of BIGLAG is similar to that of subroutine TRSAPP.

The parameter $\overline{\Delta}$ of the subproblem (6.6) is set automatically to a value that depends on three considerations. Firstly, because of the purpose of ρ, as described in the second paragraph of Section 2, the bound $\overline{\Delta} \geq \rho$ is imposed. Secondly, the Y-branch has been taken from Box 8 of Figure 1 because DIST $= \|\underline{x}_t - \underline{x}_{\rm opt}\|$ is unacceptably large, so the condition $\overline{\Delta} \leq 0.1 \, {\tt DIST}$ is reasonable. Thirdly, $\overline{\Delta}$ should be no greater than the current Δ of the trust region subproblem of Section 5, and we anticipate that Δ may be halved. These remarks provide the choice

$$\overline{\Delta} \;=\; \max \left[\, \min\{0.1 \, {\tt DIST}, \, 0.5 \, \Delta\}, \, \rho \, \right], \tag{6.16}$$

which seems to be suitable in practice, even if the given $\rho_{\rm beg}$ causes ρ to be much less than the required changes to the variables.

After the construction of $\underline{d}$ by subroutine BIGLAG, the parameters (4.12) are calculated, $\underline{x}^+$ being the vector $\underline{x}_{\rm opt} + \underline{d}$. It has been mentioned already that in theory α and β are positive, but that negative values of $\sigma = \alpha\beta + \tau^2$ may occur occasionally, due to computer rounding errors. We recall also that formula (4.11) is applied even if σ is negative, but the updating would be unhelpful if σ were too close to zero. Therefore the $\underline{d}$ from BIGLAG is rejected if and only if the current parameters have the property

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

The alternative choice of $\underline{d}$ is made by calling subroutine BIGDEN, which seeks a big value of the denominator $|\sigma|$ instead of a big value of $|\tau|$. The dependence of σ on $\underline{x} = \underline{x}_{\rm opt} + \underline{d}$ is obtained by substituting $\underline{x}^+ = \underline{x}$ and $\underline{w} = \underline{w}(\underline{x})$ into expression (4.12), using the definition (6.3). Then BIGDEN sets $\underline{d}$ to an approximation to the solution of the subproblem

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

where $\overline{\Delta}$ still has the value (6.16). This task is much more laborious than the calculation of BIGLAG, because $\sigma(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, is a quartic polynomial. Fortunately, numerical experiments show that the situation (6.17) is very rare in practice.

The methods of subroutines BIGLAG and BIGDEN are similar, except for obvious changes due to the differences between their objective functions. Indeed, BIGDEN also picks initial vectors $\underline{d}_0$ and $\underline{s}_1$ that satisfy the equations (6.8), in order to

begin an iterative procedure. Again the j-th iteration lets $\underline{d}_j$ have the form (6.7), but now θ is calculated to maximize $|\sigma(\underline{x}_{\rm opt}+\underline{d}_j)|$. When $j\geq 2$, the vector $\underline{d}=\underline{d}_j$ is returned by BIGDEN if it has the property

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

or if j has reached the value n, the test (6.19) being analogous to condition (6.13). Otherwise, after increasing j by one, the gradient $\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ is constructed, using some numbers that are known already, as described at the end of this section. If the inequality

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

holds, then $\mathcal{S}_j={\rm span}\{\underline{d}_{j-1},\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_{j-1})\}$ is a well-defined two dimensional subspace of $\mathcal{R}^n$. Then another iteration is performed, $\underline{s}_j$ being set to a vector in $\mathcal{S}_j$ with the properties (6.8). If the test (6.20) fails, however, the first order conditions for the solution of the subproblem (6.18) are nearly achieved, so BIGDEN returns the vector $\underline{d}=\underline{d}_{j-1}$.

The choice of $\underline{d}_0$ in BIGDEN is the $\underline{d}$ that has just been picked by BIGLAG, because we expect $|\sigma(\underline{x}_{\rm opt}+\underline{d})|$ to be large when $|\ell_t(\underline{x}_{\rm opt}+\underline{d})|$ is large, although rounding errors have caused the unwelcome situation (6.17). The direction $\underline{s}_1$ is taken from the space $\mathcal{S}_1={\rm span}\{\underline{d}_0,\underline{u}\}$, where $\underline{u}$ is the step $\underline{x}_k-\underline{x}_{\rm opt}$ from $\underline{x}_{\rm opt}$ to one of the other interpolation points. The value of k depends on the ratios

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

Priority is given to $k=t$, this selection being made in the case $\omega_t\leq 0.99$. Otherwise, k is such that ω_k is the least of the ratios (6.21). A criticism of this procedure is that it ignores the objective function σ, which is why the test (6.19) for termination is not tried on the first iteration. The possibility $\underline{u}=\underline{\nabla}\sigma(\underline{x}_{\rm opt})$ is unattractive, because $\underline{\nabla}\sigma(\underline{x}_{\rm opt})$ is zero in exact arithmetic, and it would be inconvenient to pick $\underline{u}=\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_0)$, because the numbers that assist the construction of this gradient, mentioned in the previous paragraph, are not yet available.

Let $\hat{\sigma}(\theta)$, $\theta\in\mathcal{R}$, be the value of $\sigma(\underline{x}_{\rm opt}+\underline{d})$, when $\underline{d}=\underline{d}(\theta)$ is the vector (6.7). The main task of an iteration of BIGDEN is to assemble the coefficients $\check{\sigma}_\ell$, $\ell=1,2,\ldots,9$, such that $\hat{\sigma}$ is the function

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

Because the right hand side of equation (4.25) is used in the calculation of σ, matrices U and V of dimension $(m+n)\times 5$ are constructed, that provide the dependence of the relevant parts of $\underline{w}-\underline{v}$ and $H(\underline{w}-\underline{v})$, respectively, on θ. We define $\underline{w}$ by putting the vector

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

into expression (6.3), but the definition (4.24) of $\underline{v}$ is independent of θ. Thus we find the components

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

and

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

where $\hat{u}_i$, $\hat{v}_i$ and $\hat{w}_i$ are the scalar products $(\underline{x}_{\rm opt}-\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)$, $\underline{d}_{j-1}^T(\underline{x}_i-\underline{x}_0)$ and $\underline{s}_j^T(\underline{x}_i-\underline{x}_0)$, respectively. We construct the rows of U by regarding these components of $\underline{w}-\underline{v}$ as functions of θ, writing them in sequence in the form

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

Then we define V by the property that the terms

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

are the first m and last n components of $H(\underline{w}-\underline{v})$. In other words, because $(\underline{w}-\underline{v})_{m+1}$ is zero, V is the product $H_{\rm red}U$, where $H_{\rm red}$ is the matrix H without its $(m+1)$-th row and column, which receives attention in the paragraph that includes equation (4.23). The product of the displays (6.26) and (6.27) is expressed as a constant plus a linear combination of $\cos(k\theta)$ and $\sin(k\theta)$, $k=1,2,3,4$, and the results are summed over i. Thus we find the coefficients $\check{\beta}_\ell$, $\ell=1,2,\ldots,9$, of the function

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

The contribution from these coefficients to expression (6.22) is explained below.

The definitions (6.3) and (4.24) provide $\underline{w}^T\underline{e}_{\rm opt}=\frac{1}{2}\{(\underline{x}_{\rm opt}-\underline{x}_0)^T(\underline{x}-\underline{x}_0)\}^2$ and $\underline{v}^T\underline{e}_{\rm opt}=\frac{1}{2}\|\underline{x}_{\rm opt}-\underline{x}_0\|^4$ in formula (4.26). Hence equations (4.12), (4.26) and (4.25), with $t\ne{\rm opt}$, allow $\hat{\sigma}$ to be written in the form

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

Therefore, because $\alpha=\underline{e}_t^TH\,\underline{e}_t$ is independent of $\underline{x}=\underline{x}_{\rm opt}+\underline{d}(\theta)$, subroutine `BIGDEN` sets the required coefficients of the function (6.22) to $\check{\sigma}_\ell=-\alpha\check{\beta}_\ell$, $\ell=1,2,\ldots,9$, initially, and then it makes the adjustments that provide the square bracket terms of equation (6.29).

The adjustment for the last term of this equation begins with the remark that $\underline{e}_t^T H(\underline{w}-\underline{v})$ is the function (6.27) of θ in the case $i=t$. Therefore BIGDEN expresses the square of this function as a constant plus a linear combination of $\cos(k\theta)$ and $\sin(k\theta)$, $k=1,2,3,4$, and it adds the resultant coefficients to the corresponding values of $\check{\sigma}_\ell$, $\ell=1,2,\ldots,9$. Moreover, one can deduce from the conditions (6.23) and (6.8) that the first square brackets of equation (6.29) contain the function

$$\left(\overline{\Delta}^2+\hat{v}_{\rm opt}\cos\theta+\hat{w}_{\rm opt}\sin\theta\right)^2+\overline{\Delta}^2\left(\hat{u}_{\rm opt}-\tfrac{1}{2}\overline{\Delta}^2\right),\qquad\theta\in\mathcal{R},\tag{6.30}$$

where $\hat{u}_{\rm opt}$, $\hat{v}_{\rm opt}$ and $\hat{w}_{\rm opt}$ are taken from expression (6.24). It follows that the final adjustment of the $\check{\sigma}_\ell$ coefficients is elementary. Next, BIGDEN computes the values $\hat{\sigma}(2\pi i/50)$, $i=0,1,\ldots,49$, directly from equation (6.22), identifying the integer $\hat{i}\in[0,49]$ that maximizes $|\hat{\sigma}(2\pi\hat{i}/50)|$. Then the quadratic polynomial $\hat{q}(\theta)$, $\theta\in\mathcal{R}$, is constructed by interpolation to $\hat{\sigma}$ at the points $\theta=2\pi i/50$, $i=\hat{i}-1,\hat{i},\hat{i}+1$. The choice of θ for the definition (6.7) of $\underline{d}_j$ is completed by giving it the value that maximizes $|\hat{q}(\theta)|$ within the range of its interpolation points.

After calculating $\underline{d}_j$, and then increasing j by one if the test (6.19) fails, the gradient $\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_{j-1})$ is required, as mentioned already. We are going to derive it from expression (6.29), the right hand side being the function $\sigma(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, where $\underline{w}$ depends on $\underline{x}$ through equation (6.3). We consider the equivalent task of finding $\underline{\nabla}\sigma(\underline{x}_{\rm opt}+\underline{d}_j)$ for the old value of j, in order to retain the notation of the previous three paragraphs.

The gradient of the first line of the function (6.29) at $\underline{x}=\underline{x}_{\rm opt}+\underline{d}_j$ is the vector

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

the right hand side being given by the relation $(\underline{x}-\underline{x}_0)=\underline{d}_j+(\underline{x}_{\rm opt}-\underline{x}_0)$. It is employed by BIGDEN, in order to avoid some cancellation when $\|\underline{d}_j\|$ is relatively small. The remainder of the gradient of the function (6.29) is the sum

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

An advantage of the work so far is that the terms (6.27) for the chosen θ are the first m and last n components of $H(\underline{w}-\underline{v})$. Thus expression (6.27) provides the numbers $\hat{\eta}_i=\{H(\underline{w}-\underline{v})\}_i$, $i=1,2,\ldots,m$, and $\check{\eta}_i=\{H(\underline{w}-\underline{v})\}_{i+m+1}$, $i=1,2,\ldots,n$. We recall from equations (4.12) and (4.25), with $t\neq{\rm opt}$, that $\underline{e}_t^TH(\underline{w}-\underline{v})$ is the current value of τ. Therefore, because the definition (6.3) shows that $w(\underline{x})_{m+1}$ is constant, the sum (6.32) can be written in the form

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

Equation (6.3) gives $\underline{\nabla}\{w(\underline{x})_i\}=\{(\underline{x}-\underline{x}_0)^T(\underline{x}_i-\underline{x}_0)\}(\underline{x}_i-\underline{x}_0)$, $i=1,2,\ldots,m$, and $\underline{\nabla}\{w(\underline{x})_{i+m+1}\}=\underline{e}_i$, $i=1,2,\ldots,n$. It follows that the required gradient of $\sigma(\underline{x})$ is the sum of three vectors, namely expression (6.31), the sum

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

and the vector in $\mathcal{R}^n$ with the components $2\,(\tau H_{t\,i+m+1}-\alpha\,\check{\eta}_i)$, $i=1,2,\ldots,n$. The description of the method of BIGDEN is complete.

7. Other details of NEWUOA

We see in Figure 1 of Section 2 that Δ is revised and MOVE is set in Box 4, that ρ is reduced in Box 12, and that a test is made in Box 14. We recall also from the end of Section 1 that shifts of origin are important to the accuracy of the H matrix. The details of these operations are addressed in this section.

Let $\Delta_{\rm old}$ and $\Delta_{\rm new}$ be the old and new values of Δ that occur in Box 4. As mentioned already, the choice of $\Delta_{\rm new}$ depends on the ratio (2.2), and also the Euclidean length of the step $\underline{d}$ receives attention. Possible values of $\Delta_{\rm new}$ are $\frac{1}{2}\|\underline{d}\|$, $\|\underline{d}\|$ and $2\|\underline{d}\|$ in the cases RATIO≤ 0.1, $0.1<$RATIO≤ 0.7 and RATIO>0.7, respectively, but we take the view that, if RATIO>0.1, then a large reduction in Δ may be too restrictive on the next iteration. Moreover, we observe the bound $\Delta\ge\rho$, and we prefer to sharpen the test in Box 10 by avoiding trust region radii that are close to ρ. Therefore NEWUOA sets $\Delta_{\rm new}$ to ρ or to $\Delta_{\rm int}$ in the cases $\Delta_{\rm int}\le 1.5\rho$ or $\Delta_{\rm int}>1.5\rho$, respectively, where $\Delta_{\rm int}$ is the intermediate value

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

The selection of MOVE in Box 4 provides a relatively large denominator for the updating formula (4.11), as stated after expression (4.12). We recall that $H\underline{w}$ and β in this expression are independent of t. Let $\mathcal{T}$ be the set $\{1,2,\ldots,m\}$, except that the integer "opt" is excluded from $\mathcal{T}$ in the case $F(\underline{x}_{\rm opt}+\underline{d})\ge F(\underline{x}_{\rm opt})$, in order to prevent the removal of $\underline{x}_{\rm opt}$ from the set of interpolation points. The numbers

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

are calculated, σ_t being the denominator that would result from choosing MOVE$=t$. There is a strong disadvantage in making $|\sigma_{\texttt{MOVE}}|$ as large as possible, however, as we prefer to retain interpolation points that are close to $\underline{x}_{\rm opt}$. The disadvantage occurs, for instance, when at least $n+1$ of the points $\underline{x}_i$, $i=1,2,\ldots,m$, are within distance Δ of $\underline{x}_{\rm opt}$, but $\underline{x}_t$ is much further away. Then the Lagrange conditions (6.1) suggest that ℓ_t may be not unlike the function $\|\underline{x}-\underline{x}_{\rm opt}\|^2/\|\underline{x}_t-\underline{x}_{\rm opt}\|^2$, $\underline{x}\in\mathcal{R}^n$, which, because of the bound $\|\underline{d}\|\le\Delta$, would imply the property

$$|\ell_t(\underline{x}_{\rm opt}+\underline{d})|\;=\;\mathcal{O}\left(\Delta^2/\,\|\underline{x}_t-\underline{x}_{\rm opt}\|^2\right). \tag{7.3}$$

Now the equations (6.5) include $\underline{e}_t^T H\underline{w}=\ell_t(\underline{x}_{\rm opt}+\underline{d})$, and it is usual for $(\underline{e}_t^T H\underline{e}_t)\beta$ and $(\underline{e}_t^T H\underline{w})^2$ to be positive numbers of similar magnitudes in expression (7.2). Thus, for general $t\in\mathcal{T}$, it may happen that $|\sigma_t|$ is $\mathcal{O}(1)$ or $\mathcal{O}(\Delta^4/\|\underline{x}_t-\underline{x}_{\rm opt}\|^4)$ in the case $\|\underline{x}_t-\underline{x}_{\rm opt}\|\leq\Delta$ or $\|\underline{x}_t-\underline{x}_{\rm opt}\|>\Delta$, respectively. Therefore NEWUOA sets `MOVE` either to zero or to the integer $t^*\in\mathcal{T}$ that satisfies the equation

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

where w_t is a weighting factor that is necessary for the automatic removal of interpolation points that are far from $\underline{x}_{\rm opt}$. This removal is encouraged by using a sixth power of $\|\underline{x}_t-\underline{x}_{\rm opt}\|$ instead of the fourth power that is indicated above. Another consideration is that interpolation points tend to cluster near $\underline{x}_{\rm opt}$ only when Δ is either being reduced or is at its lower bound ρ, so the weights are given the values

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

where $\underline{x}^*$ is the $\underline{x}_{\rm opt}$ that is going to be selected in Box `5` of Figure 1. The `MOVE`$=0$ alternative preserves the old interpolation points, so it is available only in the case $F(\underline{x}_{\rm opt}+\underline{d})\geq F(\underline{x}_{\rm opt})$. We wish to avoid applications of formula (4.11) that cause abnormal growth in the elements of H, taking into consideration that some growth is usual when a remote interpolation point is dropped. Therefore `MOVE` is set to zero instead of to t^* if and only if both the conditions $F(\underline{x}_{\rm opt}+\underline{d})\geq F(\underline{x}_{\rm opt})$ and $w_{t^*}|\sigma_{t^*}|\leq 1$ hold.

The value of ρ is decreased from $\rho_{\rm old}$ to $\rho_{\rm new}$ in Box `12` of Figure 1. The reduction is by a factor of 10, unless only one or two changes to ρ are going to attain the final value $\rho=\rho_{\rm end}$. The equation $\rho_{\rm old}/\rho_{\rm new}=\rho_{\rm new}/\rho_{\rm end}$ gives a balance between the two reductions in the latter case. These remarks and some choices of parameters provide the formula

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

for the adjustment of ρ by NEWUOA.

The reason for Box `14` in Figure 1 is explained in the penultimate paragraph of Section 2, the calculations with the current value of ρ being complete if the "`Y`" branch is taken. We see that Box `14` is reached when the trust region subproblem of Box `2` yields a step $\underline{d}$ that has the property $\|\underline{d}\|<\frac{1}{2}\rho$, which suggests that the current quadratic model Q is convex. Therefore, assuming that `CRVMIN` is a useful estimate of the least eigenvalue of $\nabla^2 Q$, we prefer not to calculate $F(\underline{x}_{\rm opt}+\underline{d})$ when the predicted reduction in F, namely $Q(\underline{x}_{\rm opt})-Q(\underline{x}_{\rm opt}+\underline{d})$, is less than $\frac{1}{8}\rho^2$`CRVMIN`. Further, if the values of the error $|Q(\underline{x}_{\rm opt}+\underline{d})-F(\underline{x}_{\rm opt}+\underline{d})|$ on recent iterations are also less than this amount, then we take the view that trying to improve the accuracy of the model would be a waste of effort. Specifically, the test in Box `14`

is satisfied if at least 3 new values of F have been computed for the current ρ, and if all the conditions

$$\|\underline{d}^{(j)}\| \leq \rho \quad \text{and} \quad |Q_j(\underline{x}_{\rm opt}^{(j)}+\underline{d}^{(j)}) - F(\underline{x}_{\rm opt}^{(j)}+\underline{d}^{(j)})| \,\leq\, \tfrac{1}{8}\rho^2 \texttt{CRVMIN}, \quad j\in\mathcal{J}, \tag{7.7}$$

hold, where Q_j, $\underline{d}^{(j)}$ and $\underline{x}_{\rm opt}^{(j)}$ are Q, $\underline{d}$ and $\underline{x}_{\rm opt}$ at the beginning of Box 5 on the j-th iteration, where CRVMIN is generated on the current iteration, and where $\mathcal{J}$ contains 3 integers, namely the iteration numbers of the 3 most recent visits to Box 5 before the current iteration. Thus the work of NEWUOA with the current ρ is terminated often, although some of the distances $\|\underline{x}_i-\underline{x}_{\rm opt}\|$, $i=1,2,\ldots,m$, may exceed 2ρ.

In order to show the importance of $\underline{x}_0$ in practice to the rounding errors of the updating formula (4.11), we assume that all the distances $\|\underline{x}_i-\underline{x}_j\|$, $1\leq i<j\leq m$, between interpolation points are of magnitude one, that $\|\underline{d}\|=\|\underline{x}^+-\underline{x}_{\rm opt}\|$ is also of magnitude one, but that $\|\underline{x}_{\rm opt}-\underline{x}_0\|=M$, say, is large. In theory, the parameters α, β, τ and σ of expression (4.12), and also the leading $m\times m$ submatrix of H, are independent of $\underline{x}_0$ (Powell, 2004a), but the definition (4.10) implies that each of the first m components of $\underline{w}$ is approximately $\frac{1}{2}M^4$. Thus much cancellation occurs in the formula

$$\beta \,=\, \tfrac{1}{2}\,\|\underline{x}^+-\underline{x}_0\|^4 - \underline{w}^T H\,\underline{w}. \tag{7.8}$$

Further, if there were an error of ε in H_{11}, and if there were no other errors on the right hand side of equation (7.8), then β would include an error of magnitude $M^8\varepsilon$, this power of M being so large that $M>100$ could be disastrous. The substitution of expression (4.26) into formula (7.8) is less unfavourable, because H_{11} is multiplied by $-(w_1-v_1)^2$, and the middle line of equation (6.24) provides the value

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

Thus the error in β is now of magnitude $M^6\varepsilon\cos^2\theta$, where θ is the angle between $\underline{x}_1-\underline{x}_0$ and $\underline{d}=\underline{x}^+-\underline{x}_{\rm opt}$. The factorization (4.16) also helps the attainment of adequate accuracy. Nevertheless, we found from numerical experiments in REAL*8 arithmetic, using some difficult objective functions, that sequences of iterations may cause unacceptable errors if $\|\underline{x}_{\rm opt}-\underline{x}_0\|\geq 10^{2.5}\|\underline{d}\|$ is allowed in the updating calculations of Section 4. Therefore NEWUOA tests the condition

$$\|\underline{d}\|^2 \leq 10^{-3}\,\|\underline{x}_{\rm opt}-\underline{x}_0\|^2 \tag{7.10}$$

before replacing $\underline{x}_{\texttt{MOVE}}$ by $\underline{x}_{\rm opt}+\underline{d}$ in Box 5 of Figure 1. If this condition holds, then $\underline{x}_0$ is overwritten by the $\underline{x}_{\rm opt}$ that occurs at the beginning of Box 5, which alters the last n rows of the matrix (1.3) and all the elements (3.11). In practice, however, the matrix $H=W^{-1}$ of expression (3.12) is stored instead of W. Therefore H is revised in the way that is implied by the change to W, except that the $(m+1)$-th row and column of H are not required. Details of this task are considered in Section 5 of Powell (2004a), so only a brief outline is given below of the changes that are made to H when $\underline{x}_0$ is shifted.

Let $\underline{x}_{\rm av}$ and $\underline{s}$ be the vectors $\frac{1}{2}(\underline{x}_0+\underline{x}_{\rm opt})$ and $\underline{x}_{\rm opt}-\underline{x}_0$, respectively, before $\underline{x}_0$ is overwritten by $\underline{x}_{\rm opt}$, let Y be the $n\times m$ matrix that has the columns

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

and let $\Theta_{\rm old}$ and $\Theta_{\rm new}$ be the old and new H matrices without their $(m+1)$-th rows and columns. Then, according to equations (5.11) and (5.12) of Powell (2004a), $\Theta_{\rm new}$ is defined by the formula

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

Thus, as mentioned already, the submatrix Ω of expression (3.12) is undisturbed, and we keep its factorization (4.16). It follows also from expressions (3.12) and (7.12) that the product $Y\Omega$ and the sum $Y\Xi_{\rm red}^T+\Xi_{\rm red}Y^T+Y\Omega\,Y^T$ are added to the last n rows of Ξ and to the trailing $n\times n$ submatrix of Υ, respectively, where $\Xi_{\rm red}$ is the original matrix Ξ without its first row.

When $\underline{x}_0$ is overwritten by $\underline{x}_{\rm opt}$, the gradient $\underline{\nabla}Q(\underline{x}_0)$ has to be revised too. Specifically, because the function (3.1) can be written in the form

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

and because $\underline{\nabla}Q(\underline{x}_{\rm opt})=\underline{\nabla}Q(\underline{x}_0)+\nabla^2Q\,\underline{s}$ follows from $\underline{s}=\underline{x}_{\rm opt}-\underline{x}_0$, the vector $\nabla^2Q\,\underline{s}$ is added to $\underline{\nabla}Q(\underline{x}_0)$. The constant term of Q is unnecessary, as stated at the end of Section 4, and ∇^2Q is independent of $\underline{x}_0$, except that, as in equation (4.28), it is expressed as the sum

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

where $\underline{v}=\sum_{j=1}^m\gamma_j\,(\underline{x}_j-\underline{x}_{\rm opt}+\frac{1}{2}\underline{s})=\sum_{j=1}^m\gamma_j\,(\underline{x}_j-\underline{x}_{\rm av})$. Therefore the shift in $\underline{x}_0$ requires $\underline{v}\,\underline{s}^T+\underline{s}\,\underline{v}^T$ to be added to Γ, although the parameters γ_j, $j=1,2,\ldots,m$, are unchanged.

The amount of work in the previous paragraph is only $\mathcal{O}(mn)$, but the implementation of the product (7.12) takes $\mathcal{O}(m^2n)$ operations. Therefore we hope that condition (7.10) holds on only a small fraction of the total number of iterations, especially when n is large. Rough answers to this question are provided by the running times of the numerical experiments of the next section. They suggest that usually the average work per iteration of NEWUOA is close to $\mathcal{O}(mn)$.

8. Numerical results

In December, 2003, the author released the Fortran software of the version of NEWUOA that has been described, having tested it on a range of problems with

up to 200 variables. Then, at the conference in Erice of these proceedings, he discussed with Nick Gould some other problems that might be tried, which led to more experiments. It became clear from one of them that a further modification would be advantageous occasionally. It has now been made, and is the first subject of this section, because the numerical results that follow were calculated by the new version of NEWUOA.

The experiment that suggested the modification is the VARDIM test problem on page 98 of Buckley (1989). The objective function is the quartic polynomial

$$F(\underline{x}) \;=\; \sum_{\ell=1}^{n}(x_\ell-1)^2+\Big\{\sum_{\ell=1}^{n}\ell\,(x_\ell-1)\Big\}^2+\Big\{\sum_{\ell=1}^{n}\ell\,(x_\ell-1)\Big\}^4, \qquad \underline{x}\in\mathcal{R}^n, \quad (8.1)$$

which takes its least value of zero at $\underline{x}=\underline{e}$, the vector of ones. Analytic differentiation gives the second derivative matrix

$$\nabla^2F(\underline{x}) \;=\; 2\,I+\Big[\,2+12\,\{\,\textstyle\sum_{\ell=1}^{n}\ell\,(x_\ell-1)\}^2\Big]\,\Theta, \qquad \underline{x}\in\mathcal{R}^n, \quad (8.2)$$

where I is the $n\times n$ unit matrix and where Θ is the rank one matrix that has the elements $\theta_{ij}=ij$, $1\le i,j\le n$. Thus $\nabla^2F(\underline{x})$ has $n-1$ eigenvalues of 2 and one of $2+[\,\frac{1}{3}+2\{\sum_{\ell=1}^{n}\ell(x_\ell-1)\}^2]\,n(n+1)(2n+1)$. When NEWUOA is employed with $m=2n+1$, however, the initial quadratic model has a diagonal second derivative matrix, the diagonal elements of ∇^2Q being approximately those of $\nabla^2F(\underline{x}_0)$, where $\underline{x}_0$ is the given starting vector of variables, which has the components $1\text{–}i/n$, $i=1,2,\ldots,n$, in the VARDIM test problem. Thus initially the eigenvalues of ∇^2Q are about $2+[\,2+12\{\sum_{\ell=1}^{n}\ell(x_\ell-1)\}^2]\,i^2$, $i=1,2,\ldots,n$, the term in square brackets being $2+\frac{1}{3}(n+1)^2(2n+1)^2$. It follows that, at the start of the calculation, ∇^2Q is a very bad estimate of ∇^2F. Further, if $n=80$ for instance, the range of eigenvalues of ∇^2Q initially is from about 5.7×10^7 to 3.6×10^{11}, but the large eigenvalue of ∇^2F at the solution $\underline{x}=\underline{e}$ is only 347762. Therefore NEWUOA cannot perform satisfactorily unless huge improvements are made to ∇^2Q by the updating formulae of the sequence of iterations.

Unfortunately, however, each application of the least Frobenius norm updating method makes the smallest change to ∇^2Q that is allowed by the new interpolation conditions, so the basic method of NEWUOA is not suitable for the VARDIM test problem. Therefore the recent modification tries to recognise when the elements of ∇^2Q are much too large, and, if there is strong evidence for this possibility, then Q is replaced by $Q_{\rm int}$, which is the quadratic model that minimizes $\|\nabla^2Q_{\rm int}\|_F$, instead of the Frobenius norm of the change to ∇^2Q, subject to the conditions $Q_{\rm int}(\underline{x}_i)=F(\underline{x}_i)$, $i=1,2,\ldots,m$, the interpolation points $\underline{x}_i$ being the updated ones at the exit from Box 5 of Figure 1. When $Q_{\rm int}$ is preferred, the gradient $\underline{\nabla}Q_{\rm int}(\underline{x}_0)$ and the parameters γ_j, $j=1,2,\ldots,m$, of the expression

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

are required. It follows from the definition of $Q_{\rm int}$ that they are the vector $\underline{g}$ and the components of $\underline{\lambda}$ in the system (3.10), where $\underline{r}$ has the components $r_i=F(\underline{x}_i)-\phi$,

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

Table 1: Two versions of NEWUOA applied to VARDIM with $m=2n+1$

$i=1,2,\ldots,m$, for any $\phi\in\mathcal{R}$. Some damage from rounding errors is avoided by the choice $\phi=F(\underline{x}_{\rm opt})$. We deduce from the notation (3.12) that $\underline{g}$ and $\underline{\lambda}$ are the products $\Xi_{\rm red}\,\underline{r}$ and $\Omega\,\underline{r}$, respectively, where $\Xi_{\rm red}$ is still the matrix Ξ without its first row. Thus NEWUOA constructs a useful form of $Q_{\rm int}$ in $\mathcal{O}(m^2)$ operations.

When the elements of ∇^2Q are much too large, the interpolation equations (1.1) imply that $\|\underline{\nabla}Q(\underline{x})\|$ is also much too large for most vectors of variables. Usually a huge value of $\|\underline{\nabla}Q(\underline{x}_{\rm opt})\|$ causes the ratio (2.2) to be tiny. Moreover, because $\underline{\nabla}Q(\underline{x}_0)$ is available, and because we have found that $\underline{\nabla}Q_{\rm int}(\underline{x}_0)$ is the product $\Xi_{\rm red}\,\underline{r}$, it is easy to compare $\|\underline{\nabla}Q_{\rm int}(\underline{x}_0)\|$ with $\|\underline{\nabla}Q(\underline{x}_0)\|$. On the iterations of the new version of NEWUOA that reach Box `5` of Figure 1 from Box `4`, a flag is set to `YES` or `NO`, the `YES` being chosen when the conditions

$$\texttt{RATIO}\leq 0.01 \qquad \text{and} \qquad \|\underline{\nabla}Q_{\rm int}(\underline{x}_0)\|\leq 0.1\,\|\underline{\nabla}Q(\underline{x}_0)\| \tag{8.4}$$

hold at the end of Box `5`. Then Q is replaced by $Q_{\rm int}$ if and only if three consecutive settings of the flag are all `YES`.

The VARDIM test problem with 80 variables can be solved by the older version of NEWUOA, in spite of the deficiencies in ∇^2Q that have been noted. Results for the unmodified and modified versions, using $\rho_{\rm beg}=(2n)^{-1}$ and $\rho_{\rm end}=10^{-6}$, are displayed on the left and right hand sides, respectively, of Table 1. The heading $\#F$ denotes the total number of calculations of the objective function, and $\underline{x}_{\rm fin}$ is the vector of variables that is returned by NEWUOA, because it gives the least calculated value of F. In theory, a reordering of the variables makes no difference, the initial set of interpolation points being unchanged for $m=2n+1$, so this device can be used to investigate some effects of computer rounding errors. The entries to the left and right of the colons in Table 1 were obtained with different orderings. We see that rounding errors are highly influential, that the values of $F(\underline{x}_{\rm fin})$ are satisfactory, and that the modification is successful in reducing $\#F$.

During the development of NEWUOA, the objective function that was used most is the trigonometric sum of squares

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

n	$m=2n+1$		$m=m^{(\mathrm{av})}$		$m=\frac{1}{2}(n+1)(n+2)$	
	$\#F$	$\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty$
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}	—	—	—	—

Table 2: Averages for NEWUOA applied to 5 versions of TRIGSSQS

namely TRIGSSQS. The elements of the matrices S and C are random integers from $[-100, 100]$, each scaling factor θ_j is sampled from the logarithmic distribution on $[0.1, 1]$, and the parameters b_i, $i=1,2,\ldots,2n$, are defined by $F(\underline{x}^*)=0$, where $\underline{x}^*$ has the components $x_j^*=\hat{x}_j^*/\theta_j$, $j=1,2,\ldots,n$, each $\hat{x}_j^*$ being picked from the uniform distribution on $[-\pi,\pi]$. The initial vector $\underline{x}_0$ has the components $(\hat{x}_j^*+0.1\hat{y}_j^*)/\theta_j$, $j=1,2,\ldots,n$, where every $\hat{y}_j^*$ is also taken at random from $[-\pi,\pi]$. The function (8.5) has saddle points and maxima, due to periodicity, and the values of the scaling factors θ_j provide a tougher problem than the case $\theta_j=1$, $j=1,2,\ldots,n$. For each n, we generate five different objective functions and starting points by choosing different random numbers. We let the number of interpolation conditions, namely m, be $2n+1$, $m^{(\mathrm{av})}$ or $\frac{1}{2}(n+1)(n+2)$, where $m^{(\mathrm{av})}$ is the integer that is nearest to $\{(n+\frac{1}{2})(n+1)(n+2)\}^{1/2}$. Results of the NEWUOA software for some of these cases, with four values of n and the parameters $\rho_{\mathrm{beg}}=10^{-1}$ and $\rho_{\mathrm{end}}=10^{-6}$, are reported in Table 2, the entries in the main part of the table being averages for the five different test problems that have been mentioned. Both $\#F$ and $\underline{x}_{\mathrm{fin}}$ have been defined already. Again the results are sensitive to the effects of computer rounding errors. The dashes in the table indicate that the problems were not tried, because of the running times that would be required on a Sun Ultra 10 workstation. The values of $\#F$ in the $m=2n+1$ part of the table are much smaller than the author had expected originally, because they become less than the number of degrees of freedom in a quadratic model when n is large. This highly welcome situation provides excellent support for the least Frobenius norm updating technique. The accuracy of the calculations is satisfactory, the $\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty$ entries in the table being comparable to ρ_{end}.

The method of NEWUOA, in particular the use of the bound $\Delta\geq\rho$ in Figure 1, is intended to be suitable for the minimization of functions that have first derivative discontinuities. Therefore Table 3 gives some results for the objective function TRIGSABS, which has the form

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

The parameters b_i, S_{ij} and C_{ij}, and the initial vector $\underline{x}_0$, are generated randomly

n	$m=2n+1$		$m=m^{(\mathrm{av})}$		$m=\frac{1}{2}(n+1)(n+2)$	
	$\#F$	$\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty$	$\#F$	$\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty$
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}	—	—	—	—

Table 3: Averages for NEWUOA applied to 5 versions of TRIGSABS

as in the previous paragraph, except that we employ the scaling factors $\theta_j=1$, $j=1,2,\ldots,n$. Different random numbers provide five test problems for each n as before. We retain $\rho_{\mathrm{beg}}=0.1$, but we set $\rho_{\mathrm{end}}=10^{-8}$, in order to take advantage of the sharpness of the minimum of F at $\underline{x}=\underline{x}^*$. The entries in Table 3 are analogous to those of Table 2. We see that, for each n, the least value of $\#F$ occurs in the $m=2n+1$ column, those results being very encouraging. If ρ_{end} is reduced to 10^{-6}, the figures for $m=2n+1$ and $n=160$ become $\#F=12007$ and $\|\underline{x}_{\mathrm{fin}}-\underline{x}^*\|_\infty=1.6\times10^{-6}$, so again $\#F$ is less than the number of degrees of freedom in a quadratic model.

We consider next a test problem that was invented by the author recently, namely SPHRPTS. Here n is even, and $n/2$ points have to be placed on the surface of the unit sphere in three dimensions at positions that are far apart. We let the k-th point $\underline{p}_k\in\mathcal{R}^3$ have the coordinates

$$\underline{p}_k \;=\; \left(\begin{array}{c}\cos x_{2k-1}\,\cos x_{2k}\\ \sin x_{2k-1}\,\cos x_{2k}\\ \sin x_{2k}\end{array}\right), \qquad k=1,2,\ldots,n/2, \tag{8.7}$$

where $\underline{x}\in\mathcal{R}^n$ is still the vector of variables. The problem is to minimize the function

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

where initially the points $\underline{p}_k$ are equally spaced on the equator of the sphere, the vector $\underline{x}_0$ having the components $(\underline{x}_0)_{2k-1}=4\pi k/n$ and $(\underline{x}_0)_{2k}=0$, $k=1,2,\ldots,n/2$. The NEWUOA software was applied to this problem, taking m and n from Tables 2 and 3, with $\rho_{\mathrm{beg}}=n^{-1}$ and $\rho_{\mathrm{end}}=10^{-6}$. The resultant values of $\#F$ are shown to the left of the colons in Table 4. We found also that $F(\underline{x}_{\mathrm{fin}})$ agrees with the minimum value of $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, to more than 10 decimal places, although there is much freedom in the optimal vector of variables, because permutations of the points and rotations of the unit sphere do not alter the value of the double sum (8.8). Therefore many adjustments of the variables in practice cause only a tiny reduction in the objective function. Indeed, after computing each $F(\underline{x}_{\mathrm{fin}})$, we inspected the sequence of values of F calculated by NEWUOA, in order to note

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

Table 4: Values of $\#F$ for the SPHRPTS problem

the position in the sequence of the first value that satisfies $F(\underline{x}) \leq 1.001\,F(\underline{x}_{\mathrm{fin}})$. These positions are given to the right of the colons in Table 4. We see that, for the SPHRPTS problem, most of the work is spent on marginal improvements to F, especially during the calculations of the $m=2n+1$ column.

The NEWUOA software has also been tested on several problems that have been proposed by other authors. The final table presents results in the following five cases using $m=2n+1$. The ARWHEAD problem (see the Appendix of Conn *et al*, 1994) has the objective function

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

and the starting point $\underline{x}_0$ is $\underline{e}\in\mathcal{R}^n$, which is still the vector of ones. In the CHROSEN problem (see page 45 of Buckley, 1989), we let F be the function

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

and the starting point $\underline{x}_0$ is $-\underline{e}\in\mathcal{R}^n$. The PENALTY1 problem (see page 79 of Buckley, 1989) includes two parameters, and we pick the objective function

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

with the starting point $(\underline{x}_0)_i=i$, $i=1,2,\ldots,n$. Our choice of parameters for the PENALTY2 problem (see page 80 of Buckley, 1989) gives the function

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

and the starting point $\underline{x}_0$ is $\frac{1}{2}\underline{e}\in\mathcal{R}^n$. The PENALTY3 problem (see page 81 of Buckley, 1989) has the objective function

$$\begin{aligned} F(\underline{x}) \;=\; & 10^{-3}\left(1 + R\,e^{x_n} + S\,e^{x_{n-1}} + R\,S\right) \\ & + \Big\{\sum_{i=1}^{n}(x_i^2-n)\Big\}^2 + \sum_{i=1}^{n/2}(x_i-1)^2, \qquad \underline{x}\in\mathcal{R}^n, \end{aligned} \tag{8.13}$$

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	$\star$	$\star$

Table 5: Values of $\#F$ for 5 problems with $m=2n+1$

where R and S are the sums

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

and we let the starting point $\underline{x}_0 \in \mathcal{R}^n$ be the zero vector. We set $\rho_{\rm end} = 10^{-6}$ in every case, while $\rho_{\rm beg}$ is given the value 0.5, 0.5, 1.0, 0.1 and 0.1 for ARWHEAD, CHROSEN, PENALTY1, PENALTY2 and PENALTY3, respectively. Table 5 shows the numbers of function evaluations that occurred when NEWUOA was applied to these problems with our usual choices of n, except that $\star$ indicates that $\#F$ exceeded 500,000.

All the ARWHEAD, CHROSEN and PENALTY1 calculations were completed successfully, the greatest distance $\|\underline{x}_{\rm fin} - \underline{x}^*\|_\infty$ being 6.1×10^{-6}, where $\underline{x}_{\rm fin}$ and $\underline{x}^*$ are still the final and optimal vectors of variables. Good accuracy was also achieved in the PENALTY2 calculations with $n \le 80$, the values of $F(\underline{x}_{\rm fin})$ agreeing to 13 decimal places with other values that were obtained for permutations of the variables and other choices of m. When $n=160$ is selected, however, the constants $e^{i/10}$, $i=1,2,\ldots,n$, vary from 1.1 to 9×10^6, so the magnitudes of the terms under the first summation sign of expression (8.12) vary from 1 to 10^{13}, which causes the PENALTY2 problem to be too difficult in REAL*8 arithmetic. We compared the given results of the PENALTY3 calculations with those that occurred after permuting the variables. The $\#F$ entries became 4336, 18209 and 125884 for $n=20$, $n=40$ and $n=80$, respectively, which agrees well with the last column of Table 5. Further, for each n, the two values of $F(\underline{x}_{\rm fin})$ were slightly less than n^2, and they agreed to about 11 decimal places. A feature of PENALTY3, however, is that the minimum value of the objective function is close to 10^{-3} and is hard to find. This magnitude is exposed by picking the variables $x_i=1$, $i=1,2,\ldots,n-1$, and $x_n = -(n^2-n+1)^{1/2}$, because then e^{x_n} is tiny and both S and the second line of expression (8.13) are zero, which provides $F(\underline{x}) = 10^{-3}(1+R\,e^{x_n}) \approx 10^{-3}$. When NEWUOA was applied to PENALTY3 with $n=160$, the original ordering of the variables yielded $\#F = 629582$ and $F(\underline{x}_{\rm fin}) = 25447.688$, while the new ordering yielded $\#F = 16844$ and $F(\underline{x}_{\rm fin}) = 0.001002$. We had not expected the new ordering to be so favourable, because the differences in the results are due entirely to computer rounding errors.

The average amount of work per iteration is mentioned at the end of Section 7, being at best $\mathcal{O}(n^2)$ in the case $m=2n+1$. We tested this possibility in the ARWHEAD and PENALTY1 experiments of Table 5. The total time in seconds of each calculation on a Sun Ultra 10 workstation was divided by the product of n^2 and $\#F$. The resultant quotients for ARWHEAD are 8.4×10^{-6}, 8.0×10^{-6}, 8.5×10^{-6} and 8.8×10^{-6} in the cases $n=20$, $n=40$, $n=80$ and $n=160$, respectively, and the corresponding quotients for PENALTY1 are 9.2×10^{-6}, 8.5×10^{-6}, 8.6×10^{-6} and 9.3×10^{-6}, the running time in the last case being nearly 5 hours, while ARWHEAD with $n=20$ was solved in only 1.36 seconds. These findings suggest that the average complexity of each iteration is proportional to n^2, which is most welcome.

The development of NEWUOA has taken nearly three years. The work was very frustrating, due to severe damage from computer rounding errors in difficult cases, before the factorization (4.16) of Ω was introduced. Therefore the author has had doubts about the use of the explicit inverse matrix $H=W^{-1}$, instead of using a factored form of W that allows the system (3.10) to be solved in $\mathcal{O}(m^2)$ operations. The numerical results are still highly sensitive to computer rounding errors, but the experiments of this section show that good accuracy is achieved eventually, which confirms the stability of the given techniques. Thus we conclude that the least Frobenius norm method for updating quadratic models is highly successful in unconstrained minimization calculations without derivatives. Readers are invited to request a free copy of the NEWUOA Fortran software by sending an e-mail to mjdp@cam.ac.uk.

Appendix: Proofs for Section 3

The assertions of the last two paragraphs of Section 3 are presented below as lemmas with proofs. The positions of the relevant interpolation points are described at the beginning of Section 3, followed by the definitions of the matrices (3.12).

Lemma 1: The first row of the initial matrix Ξ has the elements (3.13), and, for every integer i that satisfies $2\le i\le\min[n+1,m-n]$, the i-th row includes the elements (3.14). When $m\le2n$, the nonzero elements of the remaining rows of Ξ take the values (3.15), where i is any integer from the interval $[m-n+1,n+1]$. All other elements of the initial matrix Ξ are zero.

Proof: For each integer j in $[1,m]$, we let the quadratic polynomial

$$\ell_j(\underline{x}) \;=\; \ell_j(\underline{x}_0)+(\underline{x}-\underline{x}_0)^T\underline{\nabla}\ell_j(\underline{x}_0)+\tfrac{1}{2}\,(\underline{x}-\underline{x}_0)^T\,\nabla^2\ell_j\,(\underline{x}-\underline{x}_0),\quad \underline{x}\in\mathcal{R}^n, \qquad (A.1)$$

be the j-th Lagrange function of the initial interpolation points, which means that $\|\nabla^2\ell_j\|_F$ is as small as possible subject to the conditions

$$\ell_j(\underline{x}_i) \;=\; \delta_{ij},\qquad i=1,2,\ldots,m, \qquad (A.2)$$

as stated in the second paragraph of Section 6. The construction of ℓ_j is the same as the construction of D in Section 3, if the constraints (3.6) have the right hand

sides $F(\underline{x}_i)-Q_{\rm old}(\underline{x}_i)=\delta_{ij}$, $i=1,2,\ldots,m$. Therefore $\ell_j(\underline{x}_0)$ and $\underline{\nabla}\ell_j(\underline{x}_0)$ are the same as c and $\underline{g}$, respectively, in the system (3.10), when $\underline{r}$ is the coordinate vector $\underline{e}_j\in\mathcal{R}^m$. In this case, the partitioned vector on the left hand side of equation (3.10) is the j-th column of W^{-1}. It follows from the notation (3.12) that $\ell_j(\underline{x}_0)$ and $\underline{\nabla}\ell_j(\underline{x}_0)$ provide the j-th column of Ξ, as shown in the expression

$$\Xi \;=\; \left(\begin{array}{cccc} \ell_1(\underline{x}_0) & \ell_2(\underline{x}_0) & \cdots & \ell_m(\underline{x}_0) \\ \underline{\nabla}\ell_1(\underline{x}_0) & \underline{\nabla}\ell_2(\underline{x}_0) & \cdots & \underline{\nabla}\ell_m(\underline{x}_0) \end{array}\right). \tag{A.3}$$

The remainder of the proof depends on the positions of the initial interpolation points. In particular, because of the choice $\underline{x}_1=\underline{x}_0$ with the first of the conditions (A.2) for each j, the first row of the matrix (A.3) has the elements (3.13). Moreover, when k satisfies $1\le k\le\min[n,m-n-1]$, the points $\underline{x}_{k+1}=\underline{x}_0+\rho_{\rm beg}\underline{e}_k$ and $\underline{x}_{k+n+1}=\underline{x}_0-\rho_{\rm beg}\underline{e}_k$ have been chosen, so the k-th component of $\underline{\nabla}\ell_j(\underline{x}_0)$ is the divided difference

$$\begin{aligned} \Big(\underline{\nabla}\ell_j(\underline{x}_0)\Big)_k &= (2\,\rho_{\rm beg})^{-1}\Big(\ell_j(\underline{x}_{k+1})-\ell_j(\underline{x}_{k+n+1})\Big) \\ &= (2\,\rho_{\rm beg})^{-1}\,(\delta_{k+1\,j}-\delta_{k+n+1\,j}), \quad j=1,2,\ldots,m, \end{aligned} \tag{A.4}$$

because ℓ_j is a quadratic that takes the values (A.2). We replace $k+1$ by i, and then expression (A.3) gives $(\underline{\nabla}\ell_j(\underline{x}_0))_k=\Xi_{k+1\,j}=\Xi_{ij}$. It follows from equation (A.4) that formula (3.14) does provide all the nonzero elements of the i-th row of Ξ for $2\le i\le\min[n+1,m-n]$. Finally, if k satisfies $m-n\le k\le n$, then only the first two of the vectors $\underline{x}_1=\underline{x}_0$, $\underline{x}_{k+1}=\underline{x}_0+\rho_{\rm beg}\underline{e}_k$ and $\underline{x}_0-\rho_{\rm beg}\underline{e}_k$ are interpolation points. Further, the minimization of $\|\nabla^2\ell_j\|_F$ subject to the conditions (A.2) yields $(\nabla^2\ell_j)_{kk}=0$, $j=1,2,\ldots,m$, so the univariate function $\ell_j(\underline{x}_0+\alpha\underline{e}_k)$, $\alpha\in\mathcal{R}$, is a linear polynomial for each j. Therefore the $(k+1)$-th row of the matrix (A.3) contains the divided differences

$$\begin{aligned} \Xi_{k+1\,j} \;=\; \Big(\underline{\nabla}\ell_j(\underline{x}_0)\Big)_k &= (\rho_{\rm beg})^{-1}\Big(\ell_j(\underline{x}_{k+1})-\ell_j(\underline{x}_1)\Big) \\ &= (\rho_{\rm beg})^{-1}\,(\delta_{k+1\,j}-\delta_{1j}), \quad j=1,2,\ldots,m. \end{aligned} \tag{A.5}$$

Again we replace $k+1$ by i, so equation (A.5) establishes that the nonzero elements of the i-th row of Ξ have the values (3.15) when i satisfies $m-n+1\le i\le n+1$. The proof of the lemma is complete. $\Box$

Lemma 2: When $m\ge 2n+1$ holds, the initial matrix Υ is identically zero. Otherwise, Υ is a diagonal matrix, and expression (3.16) gives all the elements of Υ that are nonzero.

Proof: Let $\tilde{m}$ be the integer $\min[m,2n+1]$, and let $\tilde{\Xi}$, $\tilde{A}$ and $\tilde{X}$ be the leading $(n+1)\times\tilde{m}$, $\tilde{m}\times(n+1)$ and $(n+1)\times(n+1)$ submatrices of Ξ, A and X, respectively. The definitions (3.12) provide the matrix equation $\Xi A+\Upsilon X=0$, and its first $n+1$ columns give the identity $\tilde{\Xi}\tilde{A}+\Upsilon\tilde{X}=0$, which depends on the property in Lemma 1 that, if $m>2n+1$, then the last $m-2n-1$ columns of Ξ are zero. We deduce

from equations (3.2) and (3.11) that $\tilde{A}$ has the elements

$$\left.\begin{array}{cc} \tilde{A}_{ii}=A_{ii}=\frac{1}{2}\,\rho_{\rm beg}^4, & i=2,3,\ldots,n+1 \\ \tilde{A}_{i+n\,i}=A_{i+n\,i}=\frac{1}{2}\,\rho_{\rm beg}^4, & i=2,3,\ldots,\tilde{m}-n \\ \tilde{A}_{ij}=A_{ij}=0, & \text{otherwise} \end{array}\right\}, \quad \begin{array}{c} i=1,2,\ldots,\tilde{m}, \\ j=1,2,\ldots,n+1. \end{array} \tag{A.6}$$

We seek the elements of the product $\tilde{\Xi}\tilde{A}$, which is a square matrix. For each integer j in $[1,n+1]$, equations (3.13), (3.14) and (3.15) give the formula

$$(\tilde{\Xi}\tilde{A})_{ij} \;=\; \left\{\begin{array}{cc} \tilde{A}_{1j} & i=1, \\ (2\,\rho_{\rm beg})^{-1}\,(\tilde{A}_{ij}-\tilde{A}_{i+n\,j}), & 2\le i\le\min[n+1,m-n], \\ (\rho_{\rm beg})^{-1}\,(\tilde{A}_{ij}-\tilde{A}_{1j}), & m-n+1\le i\le n+1, \end{array}\right. \tag{A.7}$$

the last line being void in the case $m\ge 2n+1$. It follows from equation (A.6) that $\tilde{\Xi}\tilde{A}$ is a diagonal matrix, and that its first row and column are zero. Further, because $\min[n+1,m-n]$ is the same as $\tilde{m}-n$, we find the diagonal elements

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

We now consider the identity $\tilde{\Xi}\tilde{A}+\Upsilon\tilde{X}=0$. The definition (1.3) of X with $\underline{x}_1=\underline{x}_0$ imply $\tilde{\Xi}\underline{e}_1=\underline{e}_1$, where $\underline{e}_1$ is the first coordinate vector in $\mathcal{R}^{n+1}$, and we recall $\tilde{\Xi}\tilde{A}\underline{e}_1=0$. It follows from $(\tilde{\Xi}\tilde{A}+\Upsilon\tilde{X})\underline{e}_1=0$ that the first column of Υ is also zero. Thus $\tilde{\Xi}\tilde{A}+\Upsilon\tilde{X}=0$ remains true if any change is made to the first row of $\tilde{X}$. Expressions (1.3) and (3.2) allow the new $\tilde{X}$ to be $\rho_{\rm beg}$ times the $(n+1)\times(n+1)$ unit matrix. Hence Υ is the matrix $-\rho_{\rm beg}^{-1}\,\tilde{\Xi}\tilde{A}$, which we know is diagonal. Further, we deduce from equation (A.8) that Υ is the zero matrix in the cases $m\ge 2n+1$, and that otherwise the nonzero elements of Υ take the values (3.16). Therefore the lemma is true. □

Lemma 3: The initial matrix Ω has the factorization

$$\Omega \;=\; \sum_{k=1}^{m-n-1}\underline{z}_k\,\underline{z}_k^T \;=\; Z\,Z^T, \tag{A.9}$$

where the vectors $\underline{z}_k\in\mathcal{R}^m$, $k=1,2,\ldots,m-n-1$, are the columns of Z. Further, the first $\min[n,m-n-1]$ of these vectors have the components (3.18), and, if $m>2n+1$, the remaining vectors have the components (3.20), the subscripts $\hat{p}$ and $\hat{q}$ being introduced in the last paragraph of Section 3.

Proof: Let $Q(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, be the initial quadratic model, given at the beginning of Section 3. Each element of $\nabla^2 Q$ is either defined by the equations (1.1) or is set to zero. Therefore the choice of Q minimizes $\|\nabla^2 Q\|_F$ subject to the interpolation conditions. It follows from the derivation of the system (3.10) that, if we let $\underline{r}$ have the components $r_i=F(\underline{x}_i)$, $i=1,2,\ldots,m$, and if we set $\underline{\lambda}=\Omega\underline{r}$, where Ω is

taken from expression (3.12), then $\underline{\lambda}$ is the unique vector satisfying the constraints (3.7), such that $\nabla^2 Q$ is the matrix (3.8). These remarks characterise Ω uniquely, because they are valid for all right hand sides $r_i = F(\underline{x}_i)$, $i = 1, 2, \ldots, m$. Hence it is sufficient to verify that, if we put the matrix (A.9) into the equation $\underline{\lambda} = \Omega \underline{r}$ for general $\underline{r}$, then $\underline{\lambda}$ has the properties that have been mentioned.

The first of the constraints (3.7) is $\underline{\lambda}^T \underline{e} = 0$, where $\underline{e} \in \mathcal{R}^m$ is the vector of ones. Substituting $\underline{\lambda} = \Omega \underline{r}$ and $\Omega = Z Z^T$, this condition becomes $\underline{r}^T Z Z^T \underline{e} = 0$, which is achieved, because each column $\underline{z}_k$ of Z has the components (3.18) or (3.20), and both sets of components provide $\underline{z}_k^T \underline{e} = 0$. Similarly, the relation $\underline{\lambda} = Z Z^T \underline{r}$ implies that the other constraint (3.7) also holds if Z satisfies the equations

$$\sum_{i=1}^{m} Z_{ik} \, (\underline{x}_i - \underline{x}_0) \; = \; 0, \qquad k = 1, 2, \ldots, m-n-1. \tag{A.10}$$

For $1 \le k \le \min[n, m–n–1]$, the values (3.18) and (3.2) imply that the left hand side of this expression is a multiple of the difference $\rho_{\rm beg} \underline{e}_k - \rho_{\rm beg} \underline{e}_k = 0$. Alternatively, for $n+1 \le k \le m-n-1$, the values (3.20), (3.19) and (3.3), with $i = k+n+1$ and $\underline{x}_1 = \underline{x}_0$, give the condition

$$\sum_{i=1}^{m} Z_{ik} \, (\underline{x}_i - \underline{x}_0) = \rho_{\rm beg}^{-2} \, (-\sigma_p \, \rho_{\rm beg} \, \underline{e}_p - \sigma_q \, \rho_{\rm beg} \, \underline{e}_q + \sigma_p \, \rho_{\rm beg} \, \underline{e}_p + \sigma_q \, \rho_{\rm beg} \, \underline{e}_q) = 0. \tag{A.11}$$

Thus, for general $\underline{r} \in \mathcal{R}^m$, the vector $\underline{\lambda} = \Omega \underline{r}$ does obey the constraints (3.7).

By substituting $\underline{\lambda} = Z Z^T \underline{r}$, we write the matrix (3.8) in the form

$$\nabla^2 D \; = \; \sum_{k=1}^{m-n-1} \, (\underline{z}_k^T \underline{r}) \, \Big\{ \sum_{j=1}^{m} \, Z_{jk} \, (\underline{x}_j - \underline{x}_0) \, (\underline{x}_j - \underline{x}_0)^T \Big\}, \tag{A.12}$$

and we complete the proof by establishing $\nabla^2 Q = \nabla^2 D$. For $1 \le k \le \min[n, m–n–1]$, the components (3.18) provide the equations

$$\left. \begin{array}{l} \underline{z}_k^T \underline{r} \; = \; \sqrt{2} \, \rho_{\rm beg}^{-2} \, \Big\{ -F(\underline{x}_0) + \frac{1}{2} \, F(\underline{x}_0 + \rho_{\rm beg} \, \underline{e}_k) + \frac{1}{2} \, F(\underline{x}_0 - \rho_{\rm beg} \, \underline{e}_k) \Big\} \\ \sum_{j=1}^{m} \, Z_{jk} \, (\underline{x}_j - \underline{x}_0) \, (\underline{x}_j - \underline{x}_0)^T \; = \; \sqrt{2} \, \rho_{\rm beg}^{-2} \, \Big\{ \rho_{\rm beg}^2 \, \underline{e}_k \, \underline{e}_k^T \Big\} \; = \; \sqrt{2} \, \underline{e}_k \, \underline{e}_k^T \end{array} \right\}. \tag{A.13}$$

Moreover, the construction in the first paragraph of this section employs the divided difference

$$(\nabla^2 Q)_{kk} \; = \; \rho_{\rm beg}^{-2} \, \Big\{ F(\underline{x}_0 - \rho_{\rm beg} \, \underline{e}_k) - 2 \, F(\underline{x}_0) + F(\underline{x}_0 + \rho_{\rm beg} \, \underline{e}_k) \Big\}. \tag{A.14}$$

It follows that the first $\min[n, m-n-1]$ terms of the sum over k in expression (A.12) provide a diagonal matrix, whose diagonal elements are the same as those of $\nabla^2 Q$. Thus $\nabla^2 Q = \nabla^2 D$ is achieved in the cases $m \le 2n+1$. It remains to show that, if $m > 2n+1$, then the last $m–2n–1$ values of k in expression (A.12) generate the off-diagonal elements of $\nabla^2 Q$ without disturbing the diagonal elements.

For each k in the interval $[n+1, m-n-1]$, the interpolation points (3.3) and (3.19) are relevant with $i = k+n+1$. Indeed, the components (3.20) imply that

$\underline{z}_k^T \underline{r}$ is just the left hand side of equation (3.5), while the term in the braces of expression (A.12) is the matrix

$$-\underline{e}_p \underline{e}_p^T - \underline{e}_q \underline{e}_q^T + (\sigma_p \underline{e}_p + \sigma_q \underline{e}_q)(\sigma_p \underline{e}_p + \sigma_q \underline{e}_q)^T \;=\; \sigma_p \sigma_q (\underline{e}_p \underline{e}_q^T + \underline{e}_q \underline{e}_p^T). \qquad (A.15)$$

Therefore the k-th term of the sum (A.12) contributes to $(\nabla^2 D)_{pq}$ and $(\nabla^2 D)_{qp}$ the amount that is required by equation (3.5), and it does not alter any other element of $\nabla^2 D$. Thus all the different elements of $\nabla^2 Q$ that can be nonzero are provided by the different values of k in expression (A.12). The justification of the initial choice of Z is complete. □

Acknowledgements

The author is very grateful for the facilities he has enjoyed, throughout the development of NEWUOA, as an Emeritus Professor at the Centre for Mathematical Sciences of the University of Cambridge. He has also received excellent support for this research from the City University of Hong Kong and from the University of Minnesota. The first numerical experiments on the given method for updating Q were run during a two month stay in Hong Kong, and the investigations of several auxiliary techniques were helped greatly by discussions with other visitors during the IMA Program on Optimization in Minneapolis.

References

A.G. Buckley (1989), "Test functions for unconstrained minimization", Technical Report 1989 CS-3, Dalhousie University, Canada.

A.R. Conn, N.I.M. Gould, M. Lescrenier and Ph.L. Toint (1994), "Performance of a multifrontal scheme for partially separable optimization", in *Advances in Optimization and Numerical Analysis*, eds. Susana Gomez and Jean-Pierre Hennart, Kluwer Academic (Dordrecht), pp. 79–96.

A.R. Conn, N.I.M. Gould and Ph.L. Toint (2000), *Trust-Region Methods*, MPS–SIAM Series on Optimization (Philadelphia).

J.E. Dennis and R.B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall (Englewood Cliffs).

R. Fletcher and C.M. Reeves (1964), "Function minimization by conjugate gradients", *Computer J.*, Vol. 7, pp. 149–154.

M.J.D. Powell (2001), "On the Lagrange functions of quadratic models that are defined by interpolation", *Optim. Meth. Software*, Vol. 16, pp. 289–309.

M.J.D. Powell (2002), "UOBYQA: unconstrained optimization by quadratic approximation", *Math. Programming*, Vol. 92, pp. 555–582.

M.J.D. Powell (2003), "On trust region methods for unconstrained minimization without derivatives", *Math. Programming*, Vol. 97, pp. 605–623.

M.J.D. Powell (2004a), "Least Frobenius norm updating of quadratic models that satisfy interpolation conditions", *Math. Programming*, Vol. 100, pp. 183–215.

M.J.D. Powell (2004b), "On the use of quadratic models in unconstrained minimization without derivatives", *Optim. Meth. Software*, Vol. 19, pp. 399–411.

M.J.D. Powell (2004c), "On updating the inverse of a KKT matrix", in *Numerical Linear Algebra and Optimization*, ed. Ya-xiang Yuan, Science Press (Beijing), pp. 56–78.