DAMTP 2007/NA05

Developments of NEWUOA for unconstrained minimization without derivatives

M.J.D. Powell

Abstract: The NEWUOA software is described briefly, with some numerical results that show good efficiency and accuracy in the unconstrained minimization without derivatives of functions of up to 320 variables. Some preliminary work on an extension of NEWUOA that allows simple bounds on the variables is also described. It suggests a variation of a technique in NEWUOA for maintaining adequate linear independence in the interpolation conditions that are used, which leads to five versions of the technique including the original one. Numerical experiments suggest that the new versions have some merit, but the details of the calculations are influenced strongly by computer rounding errors. The dependence of the number of iterations on the number of interpolation conditions is also investigated numerically. A surprising case with $n=160$ is found, n being the number of variables, where the number of iterations is reduced when the number of conditions is decreased from $2n+1$ to $n+6$. The given conclusions may assist the development of some new software for unconstrained optimization.

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

June, 2007.

1. A success of NEWUOA

The NEWUOA Fortran software (Powell, 2006) seeks the minimum of a function $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, which is specified by a subroutine, provided by the user, that calculates the value of $F(\underline{x})$ for any given vector of variables $\underline{x} \in \mathcal{R}^n$. There are no constraints on the variables and no derivatives of F are required. The user has to provide too a starting vector $\underline{x}^{(\mathrm{o})} \in \mathcal{R}^n$, values of the parameters ρ_{beg} and ρ_{end}, and an integer m from the interval $[n+2, \frac{1}{2}(n+1)(n+2)]$. Here ρ_{beg} and ρ_{end} are the Euclidean lengths of changes that are made to the variables initially and at the end of the calculation, respectively, so $\rho_{\mathrm{beg}} \geq \rho_{\mathrm{end}}$ must hold, and ρ_{end} can be used to control the final accuracy. The purpose of m is that each iteration of NEWUOA employs a quadratic model (quadratic polynomial approximation to F) that interpolates just m values of the objective function. Each iteration changes only one (or very occasionally none) of the interpolation points, keeping m fixed. The value $m = 2n+1$ is recommended. Copies of the software are available free of charge from the author at the e-mail address mjdp@cam.ac.uk.

Let $\underline{x}^{(k)}$, $k = 1, 2, 3, \ldots$, be the best vector of variables at the beginning of the k-th iteration, which means that $F(\underline{x}^{(k)})$ is the least calculated value of F so far. Let the quadratic model at the beginning of the k-th iteration be the function

$$Q^{(k)}(\underline{x}^{(k)} + \underline{d}) \;=\; F(\underline{x}^{(k)}) + \underline{d}^T \underline{g}^{(k)} + \tfrac{1}{2}\, \underline{d}^T \nabla^2 Q^{(k)} \underline{d}, \qquad \underline{d} \in \mathcal{R}^n, \tag{1.1}$$

its parameters being the vector $\underline{g}^{(k)} \in \mathcal{R}^n$ and the $n \times n$ symmetric matrix $\nabla^2 Q^{(k)}$. Let the interpolation conditions of $Q^{(k)}$ be the equations

$$Q^{(k)}(\underline{y}_j^{(k)}) \;=\; F(\underline{y}_j^{(k)}), \qquad j = 1, 2, \ldots, m, \tag{1.2}$$

where the points $\underline{y}_j^{(k)} \in \mathcal{R}^n$, $j = 1, 2, \ldots, m$, have been chosen automatically, one of them being $\underline{x}^{(k)}$ with the property

$$F(\underline{x}^{(k)}) \;=\; \min\{F(\underline{y}_j^{(k)}) : j = 1, 2, \ldots, m\}. \tag{1.3}$$

The parameters $\rho^{(k)}$ and $\Delta^{(k)}$ are also required by the k-th iteration, $\Delta^{(k)}$ being a trust region radius that satisfies the bound

$$\Delta^{(k)} \;\geq\; \rho^{(k)}, \qquad k = 1, 2, 3, \ldots, \tag{1.4}$$

and $\rho^{(k)}$ being a positive number that is decreased automatically from ρ_{beg} to ρ_{end}. The choice $\rho^{(k+1)} = \rho^{(k)}$ is made on most iterations, because the alternative $\rho^{(k+1)} < \rho^{(k)}$ is preferred only if the bound (1.4) seems to be preventing further progress. Thus NEWUOA is suitable for the minimization of noisy objective functions.

There are two types of iteration, namely "trust region" and "alternative". On each "trust region" iteration, the step $\underline{d}^{(k)}$ from $\underline{x}^{(k)}$ is a vector $\underline{d}$ that is calculated by applying an extension of the truncated conjugate gradient method to the subproblem

$$\text{Minimize} \quad Q^{(k)}(\underline{x}^{(k)} + \underline{d}) \quad \text{subject to} \quad \|\underline{d}\| \leq \Delta^{(k)}. \tag{1.5}$$

The extension may improve $\underline{d}$ by some two dimensional searches if it reaches the trust region boundary, details being given in Powell (2006). If $\|\underline{d}^{(k)}\| < \frac{1}{2}\rho^{(k)}$ occurs, the view is taken that $\underline{x}^{(k)}+\underline{d}^{(k)}$ is too close to $\underline{x}^{(k)}$, so $\underline{d}^{(k)}$ is abandoned, and the current iteration is switched to one of "alternative" type, or a test may decide that the work with the current $\rho^{(k)}$ is complete. It is usual, however, for $\|\underline{d}^{(k)}\| \geq \frac{1}{2}\rho^{(k)}$ to hold, and then the new function value $F(\underline{x}^{(k)}+\underline{d}^{(k)})$ is calculated. Further, $\rho^{(k+1)}=\rho^{(k)}$ is set, the new trust region radius $\Delta^{(k+1)} \geq \rho^{(k+1)}$ is chosen in a usual way that depends on the ratio

$$\left\{F(\underline{x}^{(k)}) - F(\underline{x}^{(k)}+\underline{d}^{(k)})\right\} \Big/ \left\{Q^{(k)}(\underline{x}^{(k)}) - Q^{(k)}(\underline{x}^{(k)}+\underline{d}^{(k)})\right\}, \tag{1.6}$$

and $\underline{x}^{(k+1)}$ is given by the formula

$$\underline{x}^{(k+1)} = \begin{cases} \underline{x}^{(k)}+\underline{d}^{(k)}, & F(\underline{x}^{(k)}+\underline{d}^{(k)}) < F(\underline{x}^{(k)}), \\ \underline{x}^{(k)}, & F(\underline{x}^{(k)}+\underline{d}^{(k)}) \geq F(\underline{x}^{(k)}). \end{cases} \tag{1.7}$$

The new model $Q^{(k+1)}$ has to satisfy the conditions (1.2) with k replaced by $k+1$, where the new set of interpolation points has the form

$$\underline{y}_j^{(k+1)} = \begin{cases} \underline{x}^{(k)}+\underline{d}^{(k)}, & j=t, \\ \underline{y}_j^{(k)}, & j\neq t, \end{cases} \qquad j=1,2,\ldots,m, \tag{1.8}$$

for some integer t in $[1,m]$. The selection of t and the definition of $Q^{(k+1)}$ are addressed later. A "trust region" iteration is followed by another "trust region" iteration if the ratio (1.6) is at least 0.1, but otherwise the next iteration is of "alternative" type.

Usually an "alternative" iteration tries to improve the quadratic model by moving the interpolation point that is furthest from $\underline{x}^{(k)}$, where k is still the iteration number. It begins by calculating an integer t from $[1,m]$ that satisfies the equation

$$\|\underline{y}_t^{(k)}-\underline{x}^{(k)}\| = \max\{\|\underline{y}_j^{(k)}-\underline{x}^{(k)}\| : j=1,2,\ldots,m\}. \tag{1.9}$$

If $\|\underline{y}_t^{(k)}\| \leq 2\Delta^{(k)}$ holds, however, then it is assumed that there is no need to move $\underline{y}_t^{(k)}$. Instead, the iteration is switched to one of "trust region" type or the sequence of iterations with the current $\rho^{(k)}$ is terminated. A trust region iteration is preferred without reducing $\rho^{(k)}$ if one or both of the conditions $\Delta^{(k)}>\rho^{(k)}$ and $F(\underline{x}^{(k)}) < F(\underline{x}^{(k-1)})$ is satisfied. If both these conditions fail, however, the next action depends on whether or not $\rho^{(k)}$ has reached its lower bound. Specifically, termination occurs in the case $\rho^{(k)}=\rho_{\rm end}$, and otherwise both $\rho^{(k)}$ and $\Delta^{(k)}$ are reduced before the switch to an iteration of "trust region" type.

When the point $\underline{y}_t^{(k)}$ of equation (1.9) satisfies $\|\underline{y}_t^{(k)}\|>2\Delta^{(k)}$ on an "alternative" iteration, the new interpolation points have the form (1.8), so $\underline{d}^{(k)}$ is selected for the definition of $\underline{y}_t^{(k+1)}$, and then $F(\underline{x}^{(k)}+\underline{d}^{(k)})$ is calculated. The choice of $\underline{d}^{(k)}$

depends on the Lagrange function $\Lambda_t^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, which is a quadratic polynomial that takes the values

$$\Lambda_t^{(k)}(\underline{y}_j^{(k)}) \;=\; \delta_{jt}, \qquad j=1,2,\ldots,m, \tag{1.10}$$

where δ_{jt} is the Kronecker delta. These equations define all the coefficients of $\Lambda_t^{(k)}$ in the case $m=\frac{1}{2}(n+1)(n+2)$, but otherwise the freedom is taken up by the method of the updating of the quadratic model, which is the subject of the next two paragraphs. A large value of $|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d}^{(k)})|$ is helpful to the linear independence of the constraints on $Q^{(k+1)}$, namely the equations (1.2) with k increased by one. Therefore the $\underline{d}^{(k)}$ of an "alternative" iteration is obtained by applying to the subproblem

$$\text{Maximize} \quad |\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d})| \quad \text{subject to} \quad \|\underline{d}\|\leq\Delta^{(k)} \tag{1.11}$$

the extension to the truncated conjugate gradient method that has been mentioned for subproblem (1.5), except that the calculation does not begin at $\underline{d}=0$. After choosing t and $\underline{d}^{(k)}$, the updating of the model is the same on "alternative" and "trust region" iterations. Formula (1.7) is also applied in both cases. The "alternative" iteration of this paragraph sets $\Delta^{(k+1)}=\Delta^{(k)}$ and $\rho^{(k+1)}=\rho^{(k)}$, and is followed always by an iteration of "trust region" type.

Let $D^{(k)}(\underline{x})=Q^{(k+1)}(\underline{x})-Q^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, be the change to the model that is made on the k-th iteration. We write it as the quadratic function

$$D^{(k)}(\underline{x}) \;=\; D^{(k)}(\hat{\underline{x}})+(\underline{x}-\hat{\underline{x}})^T\underline{h}^{(k)}+\tfrac{1}{2}\,(\underline{x}-\hat{\underline{x}})^T\,\nabla^2D^{(k)}\,(\underline{x}-\hat{\underline{x}}), \qquad \underline{x}\in\mathcal{R}^n, \tag{1.12}$$

where $\hat{\underline{x}}$ is a fixed vector that has been chosen already, which is useful because shifts of origin avoid some losses of accuracy. Further, for a reason given below, we let $\nabla^2D^{(k)}$ have the symmetric form

$$\nabla^2D^{(k)} \;=\; \textstyle\sum_{j=1}^m \mu_j\,(\underline{y}_j^{(k+1)}-\hat{\underline{x}})\,(\underline{y}_j^{(k+1)}-\hat{\underline{x}})^T, \tag{1.13}$$

for certain parameters μ_j, $j=1,2,\ldots,m$, that have the properties

$$\textstyle\sum_{j=1}^m \mu_j \;=\; 0 \qquad \text{and} \qquad \sum_{j=1}^m \mu_j\,\underline{y}_j^{(k+1)} \;=\; 0. \tag{1.14}$$

Thus the construction of $D^{(k)}$ requires $m+n+1$ numbers to be found, namely $D^{(k)}(\hat{\underline{x}})$, the components of $\underline{h}^{(k)}\in\mathcal{R}^n$, and the parameters μ_j, $j=1,2,\ldots,m$. Besides the $n+1$ constraints (1.14), these numbers are required to satisfy the interpolation conditions

$$\begin{aligned} D^{(k)}(\underline{y}_j^{(k+1)}) \;&=\; Q^{(k+1)}(\underline{y}_j^{(k+1)})-Q^{(k)}(\underline{y}_j^{(k+1)}) \;=\; F(\underline{y}_j^{(k+1)})-Q^{(k)}(\underline{y}_j^{(k+1)}) \\ &=\; \left\{F(\underline{y}_t^{(k+1)})-Q^{(k)}(\underline{y}_t^{(k+1)})\right\}\,\delta_{jt}, \qquad j=1,2,\ldots,m, \end{aligned} \tag{1.15}$$

the last line being derived from equations (1.2) and (1.8). Thus the change to the model is defined by a square system of linear equations of dimension $m+n+1$,

although a general quadratic polynomial in n variables has $\frac{1}{2}(n+1)(n+2)$ degrees of freedom. This feature of NEWUOA is very welcome, because it allows the routine work of each iteration to be only $\mathcal{O}(n^2)$ when m is of magnitude n.

The given method for updating the quadratic model is the solution of a variational problem. Specifically, after satisfying the interpolation conditions (1.15), the freedom in $Q^{(k+1)}$ is taken up by minimizing $\|\nabla^2 Q^{(k+1)}-\nabla^2 Q^{(k)}\|_F$ subject to the symmetry of the second derivative matrices. The technique of minimizing the Frobenius norm of the change to $\nabla^2 Q$ is well known when first derivatives are available, and is considered in Section 9.1 of Dennis and Schnabel (1983), for instance. One can regard $\nabla^2 Q^{(k+1)}$ as a least squares projection of $\nabla^2 Q^{(k)}$ into the set of second derivative matrices that are compatible with the new interpolation conditions. It follows that, if $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, itself is quadratic, then the errors $\|\nabla^2 Q^{(k)}-\nabla^2 F\|_F$, $k=1,2,3,\ldots$, decrease monotonically, and the changes $\|\nabla^2 Q^{(k+1)}-\nabla^2 Q^{(k)}\|_F$, $k=1,2,3,\ldots$, tend to zero, which may be the reason for the good convergence properties of NEWUOA.

The task of deriving $Q^{(k+1)}$ from the equations (1.12)–(1.15) requires only of magnitude $(m+n)^2$ operations in the NEWUOA software, because the inverse of the matrix of this $(m+n+1)$ by $(m+n+1)$ linear system is stored and updated explicitly, except that some loss of accuracy is avoided by working with a factorization of the leading m by m submatrix of the inverse matrix. The updating of the inverse matrix is cheap, because only the t-th row and column of the matrix of the linear system are altered when the change (1.8) is made to the interpolation points. If this change caused the new system to be singular, then a division by zero would occur in the updating of the inverse matrix. Therefore singularity is avoided by keeping the divisors away from zero, which assists the choice of t on "trust region" iterations, and which justifies the choice of $\underline{d}^{(k)}$ on "alternative" iterations after obtaining t from equation (1.9). The Lagrange function $\Lambda_t^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, of the subproblem (1.11) is defined to be the quadratic such that $\|\nabla^2\Lambda_t^{(k)}\|_F$ is least subject to the conditions (1.10) and the symmetry of $\nabla^2\Lambda_t^{(k)}$, which is a special case of the variational problem that is behind the updating of Q. Therefore the coefficients of $\Lambda_t^{(k)}$ are available in the t-th column of the inverse matrix that is stored at the beginning of the k-th iteration.

A full description of NEWUOA is given in Powell (2006). It includes the choice of the points $\underline{y}_j^{(1)}$, $j=1,2,\ldots,m$, and the model $Q^{(1)}(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, for the first iteration. If m is set to $2n+1$, then two more points are placed on each coordinate direction through $\underline{x}^{(\mathrm{o})}$ (the starting vector provided by the user), and $\nabla^2 Q^{(1)}$ is diagonal. Another topic is the changes that are made occasionally to the vector $\hat{\underline{x}}$ of equation (1.12). It is explained also that the storage of $\nabla^2 Q^{(k+1)}$ includes a sum of the form (1.13), because the explicit calculation of all the elements of $\nabla^2 Q^{(k+1)}$ would require $\mathcal{O}(mn^2)$ operations.

An objective function that shows the efficiency of NEWUOA in favourable

n	Numbers of calculations of F ($\#F$)				
	Case 1	Case 2	Case 3	Case 4	Case 5
10	370	340	348	446	319
20	999	928	921	960	780
40	1951	1776	1629	1916	2114
80	3262	3497	3390	3183	3172
160	5589	5994	6492	6427	6124
320	11593	11391	12042	11780	11887

Table 1: NEWUOA applied to the test problem (1.16)

circumstances is the sum of squares

$$F(\underline{x}) \;=\; \sum_{i=1}^{2n} \left\{ b_i - \sum_{j=1}^{n} \left[S_{ij} \sin(x_j/\sigma_j) + C_{ij} \cos(x_j/\sigma_j) \right] \right\}^2, \qquad \underline{x} \in \mathcal{R}^n. \quad (1.16)$$

The elements S_{ij} and C_{ij} are random integers from $[-100, 100]$, each σ_j is chosen randomly from $[1, 10]$, and each b_i is defined by $F(\underline{x}^*) = 0$, for a vector $\underline{x}^* \in \mathcal{R}^n$ that is also chosen randomly. Thus F is periodic, with local maxima and saddle points and with a global minimum at $\underline{x} = \underline{x}^*$. The starting vector $\underline{x}^{(\mathrm{o})}$ is picked by letting the weighted differences $(x_j^{(\mathrm{o})} - x_j^*)/\sigma_j$, $j = 1, 2, \ldots, n$, be random numbers from $[-\pi/10, \pi/10]$, and the values $\rho_{\mathrm{beg}} = 0.1$ and $\rho_{\mathrm{end}} = 10^{-6}$ are set. For each choice of n, five test problems were generated randomly. The total number of calculations of F in each case is shown in Table 1. The accuracy of these tests is good, all the final values of the error $\|\underline{x}^{(k)} - \underline{x}^*\|_\infty$ being less than 1.5×10^{-5}.

The most striking features of the table are that the growth of $\#F$ as n increases seems to be no faster than linear, and that some problems with 320 variables and no sparsity have been solved. The results of several other calculations are reported in Powell (2006), including some with discontinuities in the first derivatives of F.

These successes of NEWUOA encouraged the author to seek extensions of the software that allow constraints on the variables. In particular, changes were made to the Fortran subroutines that forced the variables to maintain the bounds

$$a_i \leq x_i \leq b_i, \qquad i = 1, 2, \ldots, n, \quad (1.17)$$

throughout the calculation, for given values of a_i and b_i that satisfy $b_i \geq a_i + 2\rho_{\mathrm{beg}}$, $i = 1, 2, \ldots, n$. That work is addressed in Section 2. The bounds on the variables suggested the replacement of subproblem (1.11) by an easier calculation, and the modification was found to be very helpful in reducing the total number of function evaluations. Therefore the idea of including a similar modification in the "alternative" iterations of NEWUOA was investigated, which gave the results that are reported and discussed in Section 3. Finally, the subject of Section 4 is some numerical experiments that explore the dependence of the number of iterations on the choice of m.

2. Extensions to simple bounds

I began to develop the BOBYQA Fortran software in 2006 for minimizing $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, subject to the simple bounds (1.17). The name denotes Bound Optimization BY Quadratic Approximation, which seemed to cause amusement when I presented an invited paper on the work so far at the International Conference on Numerical Analysis and Optimization, held in Beijing to celebrate my 70th birthday in September, 2006. Whenever BOBYQA calls the subroutine that returns the function value $F(\underline{x})$, the components of $\underline{x}$ satisfy the constraints. Therefore the subproblems (1.5) and (1.11) are replaced by the calculations

$$\left.\begin{array}{rl}\text{Minimize} & Q^{(k)}(\underline{x}^{(k)}+\underline{d}) \quad \text{subject to} \quad \|\underline{d}\|\le\Delta^{(k)},\\ \text{and} & a_i\le x_i^{(k)}+d_i\le b_i, \qquad i=1,2,\ldots,n,\end{array}\right\} \tag{2.1}$$

and

$$\left.\begin{array}{rl}\text{Maximize} & |\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d})| \quad \text{subject to} \quad \|\underline{d}\|\le\Delta^{(k)},\\ \text{and} & a_i\le x_i^{(k)}+d_i\le b_i, \qquad i=1,2,\ldots,n,\end{array}\right\} \tag{2.2}$$

respectively. Most features of BOBYQA, however, are copied from NEWUOA, including the use of "trust region" and "alternative" iterations, with the same procedures for adjusting the trust region radii $\Delta^{(k)}$ and their lower bounds $\rho^{(k)}$. The values of $\rho_{\rm beg}$ and $\rho_{\rm end}$, with m and the starting vector $\underline{x}^{(\rm o)}$, are given as before. The ways of storing and updating the quadratic models and the inverse of the matrix of the system (1.12)–(1.15) are also the same, as are the occasional adjustments of the vector $\hat{\underline{x}}$ of expression (1.12). Further, the same techniques are used to pick the integer t of formula (1.8) on each iteration. An extra feature of BOBYQA is a shift of $\underline{x}^{(\rm o)}$ if necessary, so that the initial interpolation points $\underline{y}_j^{(1)}$, $j=1,2,\ldots,m$, are all feasible, which is straightforward, because the conditions $b_i\ge a_i+2\rho_{\rm beg}$, $i=1,2,\ldots,n$, are imposed on the given bounds, as mentioned already.

The method of BOBYQA for finding a rough solution of the subproblem (2.1) is considered next, because it has not been published yet. Let $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, be differentiable, let $\underline{x}^*$ be the vector of variables that minimizes F, and let $\underline{g}^*$ be the gradient $\underline{\nabla}F(\underline{x}^*)$. For each integer i in $[1,n]$, the condition $g_i^*>0$ or $g_i^*<0$ implies $x_i^*=a_i$ or $x_i^*=b_i$, respectively, which makes a case for setting $d_i=0$ in the subproblem (2.1) if both $g_i^{(k)}>0$ and $x_i^{(k)}=a_i$ or both $g_i^{(k)}<0$ and $x_i^{(k)}=b_i$ hold, where $\underline{g}^{(k)}=\underline{\nabla}Q^{(k)}(\underline{x}^{(k)})$ is introduced in equation (1.1). This test is applied by BOBYQA to identify components of $\underline{d}$ that will remain at zero during the approximate solution of the subproblem (2.1), the other components being "free". Only the free components are adjusted by the truncated conjugate gradient procedure of NEWUOA, starting at $\underline{d}=0$, but a free component may try to violate one of the bounds $a_i\le x_i^{(k)}+d_i\le b_i$ during the calculation. If this happens, then a free component that offends is transferred to the set of components that are fixed, and the truncated conjugate gradient method is restarted at the current point with one fewer free variables than before. Usually the value of the new

fixed component of $\underline{d}$ is nonzero, because it is defined by putting $\underline{x}^{(k)}+\underline{d}$ on the boundary of the linear constraint that was about to be violated. If this procedure ends on the trust region boundary, then BOBYQA may make further changes to $\underline{d}$ that are similar to those of NEWUOA, except that the bound constraints may require one or more additional components of $\underline{d}$ to be fixed. In this case any further adjustments to $\underline{d}$, with fewer free variables, continue from the current point on the trust region boundary. The amount of work of each change to $\underline{d}$ is $\mathcal{O}(mn)$, because the change has to be multiplied by the matrix $\nabla^2 Q^{(k)}$. The number of changes hardly ever exceeds ten for each of the subproblems (2.1), due to the stopping conditions that are used.

We consider next the calculation (2.2). It is stated in Section 1 that NEWUOA solves the subproblem (1.11) approximately by the extension to the truncated conjugate gradient method that makes two dimensional searches on the trust region boundary. The starting point $\underline{d}$ of that procedure satisfies $\|\underline{d}\| = \Delta^{(k)}$, which is suitable because there is no need to look inside the trust region for the greatest value of the modulus of a quadratic function. The bound constraints of the subproblem (2.2), however, may make it necessary for the choice of $\underline{d}$ by BOBYQA to satisfy $\|\underline{d}\| < \Delta^{(k)}$. We are going to compare two versions of the approximate solution of the subproblem (2.2), the first one being an extension of the truncated conjugate gradient method again. An important difference from the procedure of the previous paragraph is that the starting point is never at $\underline{d}=0$, which is the centre of the trust region. Instead, a preliminary calculation, described in the next two paragraphs, picks a $\underline{d}$ that is feasible for the subproblem (2.2), say $\underline{d}_{\rm cd}^{(k)}$, such that $|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d}_{\rm cd}^{(k)})|$ is quite large. Then the rough solution $\underline{d}$ to the subproblem (2.2) is found by the method of the previous paragraph with two modifications. Firstly, the quadratic $\pm\Lambda_t^{(k)}$ is minimized instead of $Q^{(k)}$, where the $\pm$ sign is chosen so that $\pm\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d}_{\rm cd}^{(k)})$ is negative, and secondly the starting point $\underline{d}=0$ with the use of the gradient $\underline{g}^{(k)}=\underline{\nabla} Q^{(k)}(\underline{x}^{(k)})$ are replaced by $\underline{d}=\underline{d}_{\rm cd}^{(k)}$ with the corresponding use of the gradient $\underline{\nabla} Q^{(k)}(\underline{x}^{(k)}+\underline{d}_{\rm cd}^{(k)})$ for the initial assignment of the "free" variables. Then one or more conjugate gradient steps are taken only if $\underline{d}_{\rm cd}^{(k)}$ is strictly inside the trust region. As before, all other changes to $\underline{d}$ are made by two dimensional searches in the intersection of the trust region boundary with the linear space that is spanned by the current "free" variables. We introduce the name $\underline{d}_{\rm full}^{(k)}$ for the final $\underline{d}$.

The point $\underline{d}_{\rm cd}^{(k)}$ has the subscript "cd", because its construction depends on line searches along coordinate directions and along one other line. We let $\mathcal{L}_{\rm cd}\subset\mathcal{R}^n$ be the union of these $n+1$ lines, a point being in $\mathcal{L}_{\rm cd}$ if and only if it has one of the forms $\underline{x}^{(k)}+\alpha\underline{e}_i$, $i=1,2,\ldots,n$, and $\underline{x}^{(k)}+\alpha(\underline{y}_t^{(k)}-\underline{x}^{(k)})$, for some real α, where $\underline{e}_i$ is the i-th coordinate direction in $\mathcal{R}^n$. The starting point $\underline{d}_{\rm cd}^{(k)}$ is defined to be the solution of the subproblem

$$\left.\begin{array}{c}\text{Maximize}\quad |\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d})|\quad\text{subject to}\quad \|\underline{d}\|\le\Delta^{(k)},\\ a_i\le x_i^{(k)}+d_i\le b_i,\quad i=1,2,\ldots,n,\quad\text{and}\quad \underline{x}^{(k)}+\underline{d}\in\mathcal{L}_{\rm cd}.\end{array}\right\}\qquad(2.3)$$

We see that $\underline{d}_{\rm cd}^{(k)}$ is chosen from a subset of the feasible vectors of the calculation (2.2), its freedom being taken up by maximizing the same objective function $|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d})|$, $\underline{d}\in\mathcal{R}^n$, as before. The coordinate directions in $\mathcal{L}_{\rm cd}$ allow much of the original feasible region to be explored, and the straight line through $\underline{x}^{(k)}$ and $\underline{y}_t^{(k)}$ guarantees that $|\Lambda_t^{(k)}(\underline{x}_k+\underline{d}_{\rm cd}^{(k)})|$ is positive, due to the Lagrange condition $\Lambda_t^{(k)}(\underline{y}_t^{(k)})=1$.

One reason for this choice of $\underline{d}_{\rm cd}^{(k)}$ is that it is easy to apply. Searches along $n+1$ lines through $\underline{x}^{(k)}$ are required, so the Lagrange condition $\Lambda_t^{(k)}(\underline{x}^{(k)})=0$ is helpful, and the gradient $\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)})$ is computed for use in all the line searches. It is sufficent to pick only one more linear functional of $\Lambda_t^{(k)}$ on each of the $n+1$ lines, because $\Lambda_t^{(k)}$ is quadratic. Specifically, we employ the diagonal second derivatives $(\nabla^2\Lambda_t^{(k)})_{ii}$, $i=1,2,\ldots,n$, in the searches along coordinate directions, and we employ the value $\Lambda_t^{(k)}(\underline{y}_t^{(k)})=1$ in the other line search. It has been noted already that the parameters of $\Lambda_t^{(k)}$ are available in the t-th column of the inverse matrix that is stored, and that they include the gradient $\underline{\nabla}\Lambda_t^{(k)}(\underline{\hat{x}})$ and the parameters $\mu_{tj}^{(k)}$ of the expression

$$\nabla^2\Lambda_t^{(k)} \;=\; \textstyle\sum_{j=1}^m \mu_{tj}^{(k)}\,(\underline{y}_j^{(k)}-\underline{\hat{x}})\,(\underline{y}_j^{(k)}-\underline{\hat{x}})^T. \qquad (2.4)$$

Therefore calculating $\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)})$ and all the elements $(\nabla^2\Lambda_t^{(k)})_{ii}$, $i=1,2,\ldots,n$, takes only $\mathcal{O}(mn)$ operations. The remaining work of searching along all the lines in $\mathcal{L}_{\rm cd}$ is only of magnitude n. Thus the solution $\underline{d}=\underline{d}_{\rm cd}^{(k)}$ of the subproblem (2.3) is found cheaply and exactly.

The above version of BOBYQA was applied to the following test problem. Let n be even and let the variables x_i, $i=1,2,\ldots,n$, be the coordinates of $n/2$ points in two dimensions, namely the vectors

$$\underline{p}_j \;=\; \left(\begin{array}{c} x_{2j-1} \\ x_{2j} \end{array}\right) \in \mathcal{R}^2, \qquad j=1,2,\ldots,n/2. \qquad (2.5)$$

The problem is to place these points on and within the unit square in a way that keeps the points apart. Specifically, the objective function

$$F(\underline{x}) \;=\; \sum_{i=2}^{n/2}\sum_{j=1}^{i-1} \min\left[\|\underline{p}_i-\underline{p}_j\|^{-1},\,10^6\right], \qquad \underline{x}\in\mathcal{R}^n, \qquad (2.6)$$

is minimized subject to the bounds

$$0\leq x_i\leq 1, \qquad i=1,2,\ldots,n. \qquad (2.7)$$

The term 10^6 in expression (2.6) is hardly ever relevant. The components of the starting vector $\underline{x}^{(\rm o)}$ are picked randomly and independently from the uniform distribution on $[0,1]$, except that this procedure is repeated if necessary until the condition

$$\min\left\{\|\underline{p}_i^{(\rm o)}-\underline{p}_j^{(\rm o)}\| : j=1,2,\ldots,i-1;\; i=2,3,\ldots,n/2\right\} \;>\; 0.2\,(n/2)^{-1/2} \qquad (2.8)$$

n	Numbers of calculations of F ($\#F$)				
	Case 1	Case 2	Case 3	Case 4	Case 5
20	715	842	752	626	852
40	3822	1988	3880	3719	2991
80	13766	12594	24399	13767	13893
160	36576	48185	50339	50301	58124
320	142263	95285	101546	181363	154182

Table 2: BOBYQA (Version 1) applied to the problem (2.6)–(2.7)

is achieved, where $\underline{p}_j^{(\rm o)}$, $j=1,2,\ldots,n/2$, are the points (2.5) in the case $\underline{x}=\underline{x}^{(\rm o)}$. The values $\rho_{\rm beg}=10^{-2}$ and $\rho_{\rm end}=10^{-6}$ are set. The choices of n are shown in the tables of results, and the number of interpolation conditions is always $m=2n+1$. Five cases for each n are generated by different random choices of $\underline{x}^{(\rm o)}$ that satisfy condition (2.8). We ignore the fact that the problem has many local and global minima. Instead, the accuracy of each final vector of variables is indicated by recording the final residuals of the KKT conditions.

Table 2 gives the total number of function values that occur when BOBYQA (Version 1) is applied to each of the test problems of the previous paragraph. It seems that these calculations are more difficult than those of Table 1. The greatest final value of the ∞-norm of the KKT residual vector in the experiments of Table 2 is 8.6×10^{-5}.

Version 2 of BOBYQA is a simplification of Version 1. Specifically, $\underline{d}_{\rm cd}^{(k)}$ is chosen instead of $\underline{d}_{\rm full}^{(k)}$ as the approximate solution of the subproblem (2.2), which avoids all the conjugate gradient steps and two dimensional searches that occur when $\underline{d}_{\rm full}^{(k)}$ is calculated from $\underline{d}_{\rm cd}^{(k)}$. A disadvantage of Version 2, however, is that the divisors tend to be smaller when updating the inverse matrix of Section 1, which provides the parameters of the Lagrange functions and of the change $D^{(k)}=Q^{(k+1)}-Q^{(k)}$ to the quadratic model. On the other hand, it happens often that $\underline{d}_{\rm cd}^{(k)}$ is a multiple of a coordinate direction, and then the move from $\underline{x}^{(k)}$ to $\underline{x}^{(k)}+\underline{d}_{\rm cd}^{(k)}$ stays on the boundary of all or all but one of the constraints (1.17) that are active at $\underline{x}^{(k)}$. Thus the new function value $F(\underline{x}^{(k)}+\underline{d}_{\rm cd}^{(k)})$ of Version 2 of BOBYQA may be more useful than the value $F(\underline{x}^{(k)}+\underline{d}_{\rm full}^{(k)})$ of Version 1.

The last remark shows wisdom after the event that was accumulated from several numerical experiments. In particular, when Version 2 is applied instead of Version 1 to the test problems of Table 2, without changing the data that are supplied by the user, the results in Table 3 are obtained. We see in these tests that the simplification to BOBYQA provides major gains in efficiency, except in some of the $n=20$ cases. The final values of the ∞-norms of the KKT residual vectors in the Table 3 calculations are all less than 1.3×10^{-5}, which is another improvement over the experiments of Table 2.

n	Numbers of calculations of F ($\#F$)				
	Case 1	Case 2	Case 3	Case 4	Case 5
20	723	592	935	625	723
40	3380	1291	2847	3220	2291
80	12336	11020	7963	9458	11157
160	29834	21806	23746	24990	21212
320	66682	48242	53189	64121	44775

Table 3: BOBYQA (Version 2) applied to the problem (2.6)–(2.7)

The unexpected superiority of Version 2 over Version 1 of BOBYQA encouraged me to investigate a similar modification to NEWUOA. The progress so far of this new work is reported in the next section. It has taken so much of my time that the development of BOBYQA has not advanced during the last six months.

3. On the "alternative" iterations of NEWUOA

It is suggested in the last section that Version 2 of BOBYQA may be more efficient than Version 1, because the step from $\underline{x}^{(k)}$ to $\underline{x}^{(k)}+\underline{d}_{\rm cd}^{(k)}$ stays on the boundaries of all, or all but one, of the constraints that are active at $\underline{x}^{(k)}$. Nevertheless, in this section we consider the possibility that Version 2 may also be better when there are no constraints on the variables. For convenience, we continue to apply the BOBYQA software, and we remove the influence of the constraints by setting the values $a_i=-10^{10}$ and $b_i=10^{10}$, $i=1,2,\ldots,n$, in expression (1.17). Thus the vectors $\underline{d}_{\rm full}^{(k)}$ and $\underline{d}_{\rm cd}^{(k)}$, derived from subproblems (2.2) and (2.3) respectively, become independent of the simple bounds, but the other features of these vectors are retained. In particular, $\underline{d}_{\rm cd}^{(k)}$ remains the exact solution of subproblem (2.3), because the set $\mathcal{L}_{\rm cd}$ still contains only n+1 straight lines, namely the lines through $\underline{x}^{(k)}$ that are parallel to the coordinate directions, and the single line through $\underline{x}^{(k)}$ and the interpolation point $\underline{y}_t^{(k)}$ that is going to be dropped. Further, $\underline{d}_{\rm cd}^{(k)}$ is still used as a starting point of the iterative procedure that calculates $\underline{d}_{\rm full}^{(k)}$ by applying conjugate gradient steps and two dimensional searches on the boundary of the trust region. Therefore, in the usual case when $\underline{d}_{\rm full}^{(k)}$ is different from $\underline{d}_{\rm cd}^{(k)}$, the extra work of Version 1 improves the value of the objective function $|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d})|$, $\underline{d}\in\mathcal{R}^n$, that occurs in the two subproblems.

At present there is only one version of NEWUOA, but in this section we are going to compare algorithms for unconstrained optimization that differ in the calculation of $\underline{d}^{(k)}$ by the "alternative" iterations. Therefore we take the view that the results of Table 1 are given by "The $\underline{d}_{\rm full}^{(k)}$ version applied to problem (1.16)". Indeed, most of the captions of the tables in this section have the form "The $\underline{d}_*^{(k)}$

n	Numbers of calculations of F ($\#F$)				
	Case 1	Case 2	Case 3	Case 4	Case 5
10	327	303	349	412	235
20	837	917	826	725	766
40	1550	1623	1747	1732	1823
80	3645	3488	4079	3357	3578
160	6573	7084	7012	6733	6761
320	12506	12935	12839	12407	12291

Table 4: The $\underline{d}_{\rm cd}^{(k)}$ version applied to problem (1.16)

version applied to problem (u.v)", where the subscript "$*$" may be "full" or "cd" for instance, $\underline{d}_*^{(k)}$ being the choice of $\underline{d}^{(k)}$ on the alternative iterations, while "(u.v)" is the number of the display in the text that specifies the objective function of the unconstrained calculation. All of the numerical results were produced by versions of BOBYQA, as mentioned already, with a_i and b_i, $i=1,2,\ldots,n$, being set to values that cause the bounds (1.17) to be irrelevant.

In order to make comparisons with Table 1, the $\underline{d}_{\rm cd}^{(k)}$ version of the software was applied to the sum of squares objective function (1.16), using exactly the same data and random numbers as before. The results are given in Table 4. We see that all but two of the entries with $n\leq 40$ are better than the corresponding entries in Table 1, but, unfortunately, all the entries in the last two rows of Table 4 are worse. These figures suggest that, besides being easier to implement, the replacement of $\underline{d}_{\rm full}^{(k)}$ by $\underline{d}_{\rm cd}^{(k)}$ may reduce the number of iterations in many unconstrained calculations with fewer than 50 variables. The final $\underline{x}^{(k)}$ achieves the property $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty<1.5\times10^{-5}$ in all the tests of Table 4.

The relatively poor figures in the last two rows of Table 4 may be due to the hugeness of spaces of more than 100 variables. Indeed, if $\underline{d}$ is restricted to the $n+1$ straight lines of $\mathcal{L}_{\rm cd}$, then it is highly unlikely that a good approximate solution to the subproblem (1.11) can be found. Nevertheless, the results in the $n=160$ and $n=320$ rows of Table 4 are not bad. Therefore more research is needed into the choice of $\underline{d}^{(k)}$ by the alternative iterations.

We are going to investigate another choice experimentally, partly because it has the advantage of being convenient if there are general linear constraints on the variables. In order to describe it, we let $\underline{d}_{\rm ip}^{(k)}$ be the exact solution to the subproblem

$$\text{Maximize}\quad |\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d})|\quad\text{subject to}\quad \|\underline{d}\|\leq\Delta^{(k)}\quad\text{and}\quad \underline{x}^{(k)}+\underline{d}\in\mathcal{L}_{\rm ip},\quad (3.1)$$

where $\mathcal{L}_{\rm ip}\subset\mathcal{R}^n$ is the union of $m-1$ straight lines, each of these lines being through $\underline{x}^{(k)}$ and another of the interpolation points $\underline{y}_j^{(k)}$, $j=1,2,\ldots,m$. Again the value $\Lambda_t^{(k)}(\underline{x}^{(k)})=0$ is helpful, and again the gradient $\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)})$ is computed

n	Numbers of calculations of F ($\#F$)				
	Case 1	Case 2	Case 3	Case 4	Case 5
10	298	302	312	362	300
20	1021	870	711	850	776
40	1906	1674	1451	1867	1878
80	3387	2933	3288	3236	3061
160	5983	5853	5911	6239	5802
320	11104	11599	11503	11427	11559

Table 5: The $\underline{d}_{\rm ip+}^{(k)}$ version applied to problem (1.16)

for use in all the line searches. Further, the remaining freedom in the quadratic function $\Lambda_t^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, on each of the straight lines is taken up by one of the Lagrange conditions (1.10). Thus the calculation of $\underline{d}_{\rm ip}^{(k)}$ is even easier than that of $\underline{d}_{\rm cd}^{(k)}$. Numerical results with the choice $\underline{d}^{(k)}=\underline{d}_{\rm ip}^{(k)}$ on the alternative iterations, however, tend to be unfavourable when n is large. In particular, the values of $\#F$ that correspond to the entries in the last row of Table 1 or 4 are 15043, 16391, 19859, 21582 and 15069, while the final errors $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ in these cases range from 2.5×10^{-5} to 3.8×10^{-5}. General linear constraints have been mentioned because, if the points $\underline{y}_j^{(k)}$, $j=1,2,\ldots,m$, are feasible, then all the line segments between them are also feasible, and they include much of the set $\mathcal{L}_{\rm ip}$.

A weakness of picking $\underline{d}^{(k)}=\underline{d}_{\rm ip}^{(k)}$ is that there is no resistance to any tendencies for the points $\underline{y}_j^{(k)}$, $j=1,2,\ldots,m$, to lie in an affine subset of dimension less than n. Indeed, if the points $\underline{y}_j^{(k)}$, $j=1,2,\ldots,m$, had this property, then it would be inherited by the new interpolation points when formula (1.8) is applied. Although this degenerate situation is hypothetical, it provides some support for giving more attention to the gradient $\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)})$. Therefore we compare $|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d}_{\rm ip}^{(k)})|$ with the greatest value of $|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d})|$ that can be achieved by letting $\underline{d}$ be a multiple of $\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)})$ that satisfies $\|\underline{d}\|\le\Delta^{(k)}$. This value is at least the product $\|\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)})\|\,\Delta^{(k)}$, because $\Lambda_t^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, is a quadratic function that is zero at $\underline{x}^{(k)}$. Our comparison is the easy test

$$|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d}_{\rm ip}^{(k)})|\ \ge\ c_{\rm ip}\,\|\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)})\|\,\Delta^{(k)},\tag{3.2}$$

where $c_{\rm ip}$ is a prescribed nonnegative constant. If this condition holds, we retain $\underline{d}^{(k)}=\underline{d}_{\rm ip}^{(k)}$, but otherwise we prefer the step

$$\underline{d}^{(k)}\ =\ \pm\Delta^{(k)}\,\|\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)}\|^{-1}\,\underline{\nabla}\Lambda_t^{(k)}(\underline{x}^{(k)}),\tag{3.3}$$

where the $\pm$ sign is calculated to give the greater value of $|\Lambda_t^{(k)}(\underline{x}^{(k)}+\underline{d}^{(k)})|$. We let $\underline{d}_{\rm ip+}^{(k)}$ denote this choice of $\underline{d}^{(k)}$ on the alternative iterations.

Version	Numbers of calculations of F ($\#F$)				
	Case 1	Case 2	Case 3	Case 4	Case 5
$\underline{d}^{(k)}_{\rm full}$	25281	26303	24615	26498	26297
$\underline{d}^{(k)}_{\rm cd}$	34488	30192	27713	31164	37343
$\underline{d}^{(k)}_{\rm ip}$	47685	201260	87080	34233	187714
$\underline{d}^{(k)}_{\rm ip+}$	10093	10721	10905	10305	10611
$\underline{d}^{(k)}_{\rm ip++}$	24810	10014	26351	11651	26831

Table 6: All versions applied to problem (3.4) with $n=320$

Table 5 gives results for the trigonometric sum of squares when $\underline{d}^{(k)}_{\rm ip+}$ is tried with $c_{\rm ip}=0.1$, all the other data and random numbers of the experiments being as before. The final values of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ are now at most 1.1×10^{-5}. A comparison with Table 1 shows that the entries in Table 5 are less than the corresponding values of $\#F$ in Table 1 in 25 of the 30 cases. Moreover, all the entries in the last 3 rows of Table 5 are better than the corresponding entries in those rows of Table 4. Thus we find that the choice $\underline{d}^{(k)}=\underline{d}^{(k)}_{\rm ip+}$ brings some advantages. The $\underline{d}^{(k)}_{\rm ip+}$ calculations were repeated with $c_{\rm ip}=0.5$ instead of $c_{\rm ip}=0.1$. Improvements in $\#F$ over the results in Table 5 occurred in only 7 of the 30 trials, and the greatest final value of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ increased a little to 1.4×10^{-5}. For convenience later, we let $\underline{d}^{(k)}_{\rm ip+}$ and $\underline{d}^{(k)}_{\rm ip++}$ denote the $\underline{d}^{(k)}_{\rm ip+}$ choices of $\underline{d}^{(k)}$ in the cases $c_{\rm ip}=0.1$ and $c_{\rm ip}=0.5$, respectively.

The author has applied the $\underline{d}^{(k)}_{\rm full}$, $\underline{d}^{(k)}_{\rm cd}$, $\underline{d}^{(k)}_{\rm ip}$, $\underline{d}^{(k)}_{\rm ip+}$ and $\underline{d}^{(k)}_{\rm ip++}$ versions of BOBYQA to the ARWHEAD test problem in Powell (2006), which has the objective function

$$F(\underline{x}) \;=\; \textstyle\sum_{j=1}^{n-1}\,\{(x_j^2+x_n^2)^2-4\,x_j+3\}, \qquad \underline{x}\in\mathcal{R}^n. \tag{3.4}$$

The least value of F is zero, which occurs when the variables take the values $x_j=x_j^*=1,\ j=1,2,\ldots,n-1$ and $x_n=x_n^*=0$. The starting vector $\underline{x}^{(\rm o)}$ is given the components $\underline{x}^{(\rm o)}_j=1,\ j=1,2,\ldots,n$, and the other parameters are set to the values $m=2n+1$, $\rho_{\rm beg}=0.5$ and $\rho_{\rm end}=10^{-6}$. Again the choices of n are 10, 20, 40, 80, 160 and 320, and again there are five cases for each n, distinguished now by the ordering of the variables, which is done by a random permutation. The permutations would be irrelevant in exact arithmetic, but in practice they can cause the calculations to proceed very differently. The final values of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ are a triumph for the ability of BOBYQA to recover from damage by computer rounding errors, the greatest final value of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ in all of the ARWHEAD tests being only 8.0×10^{-6}. The relative changes in $\#F$ due to the permutations are greatest when there are 320 variables, these results being shown in Table 6. The tests with smaller values of n confirm that the $\underline{d}^{(k)}_{\rm ip}$ version of BOBYQA requires the most iterations. The superiority of the $\underline{d}^{(k)}_{\rm ip+}$ version in Table 6, however, fails

Version	Numbers of calculations of F ($\#F$)				
	Case 1	Case 2	Case 3	Case 4	Case 5
$\underline{d}_{\rm full}^{(k)}$	21739^{ℓ}	23384	21727	24735^{ℓ}	27177
$\underline{d}_{\rm cd}^{(k)}$	18126	20255	18783	19445	18794
$\underline{d}_{\rm ip}^{(k)}$	91629^{ℓ}	13746	14960	13764	14374
$\underline{d}_{\rm ip+}^{(k)}$	26932^{ℓ}	23614	27673	25006	27475
$\underline{d}_{\rm ip++}^{(k)}$	24009^{ℓ}	26026^{ℓ}	20372	23319	24895

Table 7: All versions applied to problem (3.5) with $n=320$

spectacularly when n is reduced. In particular, when n is 80, the five values of $\#F$ for $\underline{d}_{\rm ip+}^{(k)}$ are all in the range $[7466, 7995]$, while the corresponding ranges for $\underline{d}_{\rm full}^{(k)}$, $\underline{d}_{\rm cd}^{(k)}$ and $\underline{d}_{\rm ip++}^{(k)}$ are $[1990, 2406]$, $[3690, 4469]$ and $[3170, 4258]$, respectively. Furthermore, the $\underline{d}_{\rm cd}^{(k)}$ version is the clear winner in all the $n=40$ tests, but $\underline{d}_{\rm ip++}^{(k)}$ takes first place easily with $\#F=6974$ in one of the $n=160$ cases. These major differences seem to be unreasonable.

More confusion occurs when the five versions of BOBYQA are applied to the CHROSEN test problem in Powell (2006). It has the objective function

$$F(\underline{x}) \;=\; \textstyle\sum_{j=1}^{n-1} \{4\,(x_j-x_{j+1}^2)^2+(1-x_{j+1})^2\}, \qquad \underline{x}\in\mathcal{R}^n, \tag{3.5}$$

which is least at $x_j=x_j^*=1$, $j=1,2,\ldots,n$, but convergence to a local minimum is also possible, and then the final values of $F(\underline{x}^{(k)})$ and $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ are about 3.628 and 1.784, respectively. As in the ARWHEAD experiments, the data are $m=2n+1$, $\rho_{\rm beg}=0.5$ and $\rho_{\rm end}=10^{-6}$, and permutations of the variables provide five cases for each n. The values of n are also as before, but the starting vector $\underline{x}^{(\rm o)}$ is given the components $x_j^{(\rm o)}=-1$, $j=1,2,\ldots,n$. The $n=320$ results are shown in Table 7 for comparison with Table 6. The superscript "ℓ" in Table 7 denotes convergence to the local minimum that has been mentioned, while in the other cases the greatest final value of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ is 8.1×10^{-5}. Ignoring the $\underline{d}_{\rm ip}^{(k)}$ rows for the moment, we see that $\underline{d}_{\rm cd}^{(k)}$ is a clear winner in Table 7, but it is in last place in Table 6. Another difference is that the superiority of $\underline{d}_{\rm ip+}^{(k)}$ over $\underline{d}_{\rm ip++}^{(k)}$ in Table 6 is not achieved in Table 7. Further, in each of the 25 CHROSEN tests with smaller values of n, the $\underline{d}_{\rm ip+}^{(k)}$ value of $\#F$ is greater than 1.2 times the $\underline{d}_{\rm ip++}^{(k)}$ value. The huge success of $\underline{d}_{\rm ip}^{(k)}$ in Cases 2–5 of Table 7 is an unexplained mystery, its severe inefficiency in Case 1 being typical, as confirmed in Table 6. Another huge success and four severe inefficiencies are given by $\underline{d}_{\rm ip}^{(k)}$ in the $n=160$ experiments of this paragraph, the values of $\#F$ being 75049, 6708, 27575^{ℓ}, 32820^{ℓ} and 32327^{ℓ}.

By changing the starting vector $\underline{x}^{(\rm o)}$, we obtain some results for the CHROSEN function (3.5) that are not chaotic for large n. Specifically, each component of

n	Average values of $\#F$				
	$\underline{d}_{\rm full}^{(k)}$	$\underline{d}_{\rm cd}^{(k)}$	$\underline{d}_{\rm ip}^{(k)}$	$\underline{d}_{\rm ip+}^{(k)}$	$\underline{d}_{\rm ip++}^{(k)}$
10	326	329	375	367	325
20	772	708	822	745	727
40	1744	1666	2749	1786	1801
80	3810	3582	7965	3566	3523
160	8291	7697	33010	7837	7475
320	15210	14320	73852	15789	13075

Table 8: All versions applied to problem (3.5) with random $\underline{x}^{(\rm o)}$

$\underline{x}^{(\rm o)}$ is picked randomly and independently from the logarithmic distribution on $[0.5, 2.0]$, we retain $m = 2n+1$ and $\rho_{\rm end} = 10^{-6}$, but we set $\rho_{\rm beg} = 0.1$. Again five test problems are generated for each of the usual values of n, by different choices of the random $\underline{x}^{(\rm o)}$ instead of by permutations of the variables. These data avoid convergence to the local minimum, the greatest final value of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ throughout the experiments being 7.0×10^{-4}, which is due to the poor accuracy of the $\underline{d}_{\rm ip}^{(k)}$ calculations with 320 variables. The results are so consistent that, for each n and each choice of $\underline{d}^{(k)}$ on the "alternative" iterations, it is suitable to present averages of $\#F$ over the five test problems. These results are reported in Table 8, the averages being rounded to the nearest integer. We see that the use of $\underline{d}_{\rm ip}^{(k)}$ can be very expensive, as noted earlier, but that there are not large differences between the other columns of Table 8. The $\underline{d}_{\rm cd}^{(k)}$ and $\underline{d}_{\rm ip++}^{(k)}$ versions seem to be best for the smaller and larger values of n, respectively. Thus there is good support for the idea of replacing $\underline{d}_{\rm full}^{(k)}$ by a choice of $\underline{d}^{(k)}$ that is easier to calculate.

4. On the number of interpolation points

The number of interpolation points throughout the experiments so far is $m=2n+1$. We investigate briefly in this section whether some other choices of m may be more efficient. Reductions in m decrease the routine work of each iteration, because one of the main tasks is updating the inverse of the matrix of an $(m+n+1)$ by $(m+n+1)$ linear system, as mentioned in Section 1. On the other hand, decreases in m also reduce the amount of information that is given to quadratic models by interpolation conditions, so the total number of iterations may increase. We continue to take n from the set $\{10, 20, 40, 80, 160, 320\}$ for our numerical tests. We compare the choices $m=n+6$, $m=1.5n+1$ and $m=2n+1$, because interpolation takes up $m-n-1$ degrees of freedom in the second derivative matrix of each quadratic model, and we prefer m to be of magnitude n.

n	Range of $\#F$		
	$m=2n+1$	$m=1.5n+1$	$m=n+6$
10	235–446	327–614	327–614
20	711–1021	885–1640	1025–1830
40	1451–2284	2049–3289	2747–4749
80	2933–4079	5825–7346	6263–12246
160	5589–9510	11941–14719	15762–27477
320	11018–21582	24025–29527	34854–61109

Table 9: All versions applied to problem (1.16) for 3 values of m

n	Range of $\#F$		
	$m=2n+1$	$m=1.5n+1$	$m=n+6$
10	248–474	252–496	252–496
20	603–962	611–968	718–1139
40	1459–2021*	1548–2303*	1502–2452
80	3096–4110*	3091–4809*	3763–4903
160	7139–8929*	7267–10022*	8682–12720
320	12241–18465*	14064–20433*	18719–30007

Table 10: The CHROSEN tests with random $\underline{x}^{(\mathrm{o})}$ for 3 values of m

Our first experiments with the smaller values of m employ the sum of squares objective function (1.16), without changing any of the other values of the parameters and data that have been specified already. All five versions of the choice of $\underline{d}^{(k)}$ on the "alternative" iterations are tried. Thus 25 values of $\#F$ are obtained for each m and n, the least and greatest of them being reported in Table 9. We see that the number of iterations increases when m is decreased, which is alleviated slightly by the reduction in the work of each iteration. All the final values of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ in the tests of Table 9 are less than 1.3×10^{-4}. Another interesting point that receives attention later is that, although it is stated near the end of the paragraph that includes expression (3.1) that the choice $\underline{d}^{(k)}=\underline{d}_{\mathrm{ip}}^{(k)}$ is unfavourable, this does not happen in the present experiments with the smaller values of m.

The Table 8 tests with the CHROSEN objective function (3.5) and the random choices of $\underline{x}^{(\mathrm{o})}$ have also been run for $m=1.5n+1$ and $m=n+6$. The range of $\#F$ that occurs for each m and n is shown in Table 10, except that the superscript "$*$" indicates that all the $\underline{d}_{\mathrm{ip}}^{(k)}$ calculations are excluded from the results. The reason for doing so is to avoid counts that are unfair to the other choices of $\underline{d}^{(k)}$, as indicated by the very large numbers in the $\underline{d}_{\mathrm{ip}}^{(k)}$ column of Table 8. The $\underline{d}_{\mathrm{ip}}^{(k)}$ values of $\#F$ are particularly remarkable when m is $1.5n+1$ and n is 320. Indeed,

Version	Range of $\#F$ for the 5 cases		
	$m=2n+1$	$m=1.5n+1$	$m=n+6$
$\underline{d}^{(k)}_{\rm full}$	8658–9695	7967–9538	5673–7015
$\underline{d}^{(k)}_{\rm cd}$	9357–11592	7906–10102	7999–14719
$\underline{d}^{(k)}_{\rm ip}$	53860–59765	20495–37102	7439–10698
$\underline{d}^{(k)}_{\rm ip+}$	16531–19105	4646–14549	9361–9749
$\underline{d}^{(k)}_{\rm ip++}$	6974–14404	4885–9270	5200–6000

Table 11: The ARWHEAD tests with $n=160$ for 3 values of m

in the five cases that are provided by the random choices of $\underline{x}^{(\rm o)}$, the values of $\#F$ are 132095, 92833, 49485, 11405 and 11841. It follows from Table 10 that, although $\underline{d}^{(k)}_{\rm ip}$ is a clear loser on the first three of these occasions, it is a clear winner in the other two cases. There are no superscripts in the last column of the table, because, for $m=n+6$, the performance of $\underline{d}^{(k)}_{\rm ip}$ is similar to the performance of the other choices of $\underline{d}^{(k)}$. Another noteworthy feature of Table 10 is that the increases in $\#F$ are much smaller than in Table 9 when m is reduced. Thus, for each n, the choice $m=n+6$ provides the fastest of the calculations in Table 10.

It happens sometimes that the number of iterations is reduced when m is decreased from $2n+1$ to $1.5n+1$ and then from $1.5n+1$ to $n+6$. This unexpected behaviour is clear in many of the tests for large n when F is the ARWHEAD objective function (3.4). As in the experiments of Table 6, we set $\rho_{\rm beg}=0.5$, $\rho_{\rm end}=10^{-6}$ and $x_j^{(\rm o)}=1$, $j=1,2,\ldots,n$, and again five cases are constructed for each n by random perturbations of the variables. The results for $n=160$ are reported in Table 11, including a comparison of the five versions of $\underline{d}^{(k)}$ on the "alternative" iterations, the entries being the least and greatest values of $\#F$ that occurred among the five cases for each m and $\underline{d}^{(k)}$. We see that, when m is reduced, the upper limits of the ranges become smaller in every row of the table, except for the last entry in the $\underline{d}^{(k)}_{\rm cd}$ row. All the other usual choices of n were tried too, the greatest final value of $\|\underline{x}^{(k)}-\underline{x}^*\|_\infty$ in all these experiments being only 1.02×10^{-5}. For $n=320$ and $m=n+6$, the ranges of $\#F$ that correspond to the last column of Table 11 are 14826–20885, 21610–28663, 13198–15177, 13396–18950 and 15425–19557 for $\underline{d}^{(k)}_{\rm full}$, $\underline{d}^{(k)}_{\rm cd}$, $\underline{d}^{(k)}_{\rm ip}$, $\underline{d}^{(k)}_{\rm ip+}$ and $\underline{d}^{(k)}_{\rm ip++}$, respectively. Thus $\underline{d}^{(k)}_{\rm ip}$ has become a hero, although it is highly inefficient in Tables 6 and 8.

One reason for preferring $n=160$ to $n=320$ in Table 11 is to avoid the very wide range of $\#F$ that is given by the five permutations of the variables when the choices $n=320$, $m=1.5n+1$ and $\underline{d}^{(k)}=\underline{d}^{(k)}_{\rm ip}$ are made. Indeed, $\#F$ takes the values 16616, 12941, 11692, 10288 and 106295. The corresponding entry for $n=160$ in Table 11 is the range 20495–37102. By comparing these figures, we find that substantial reductions in $\#F$ are possible when n is increased if the consequences

of the permutations of the variables are favourable. Such conclusions have to be drawn from numerical experiments and not from theoretical analysis, because the permutations are irrelevant in exact arithmetic. These huge differences in $\#F$ in practice do not impair the accuracy of the final $\underline{x}^{(k)}$, which confirms that some of the techniques in the software provide good automatic recovery from damage by computer rounding errors.

I have to decide on one choice of $\underline{d}^{(k)}$ by the "alternative" iterations before releasing the BOBYQA software. The present paper has been written in advance of that selection, in order to provide a contribution on recent research to the Special Issue of IMAJNA that will celebrate my 70th birthday. I hope the reader finds that the information in the given numerical results is sufficiently interesting to compensate for the lack of completeness in the work that is presented. I would like to pick a $\underline{d}^{(k)}$ that can be adapted conveniently to linear constraints on the variables, that can be calculated as easily as the exact solution of subproblem (2.3) or (3.1), that is not influenced much by orthogonal rotations of the variables, and that combines good final accuracy with relatively low values of $\#F$. This task is very challenging because BOBYQA is intended to be suitable for the minimzation of a wide range of objective functions.

The choice of the number m of interpolation points also deserves further investigation. A comparison of Tables 9 and 11 suggests that the best choice may depend strongly on $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$. Another consideration is that the number and positions of the points have to be suitable for defining all the linear polynomial terms in the updating of the quadratic model, because the method that takes up the freedom in the updating gives attention only to second derivatives. Thus the updating has the welcome property of being independent of shifts of the origin of $\mathcal{R}^n$. I am going to ask the users of BOBYQA to pick their values of m. Therefore my decisions need to provide software that is without the kinds of inefficiencies that are shown in the $\underline{d}_{\rm ip}^{(k)}$ entries of Tables 6 and 8.

References

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

M.J.D. Powell (2006), "The NEWUOA software for unconstrained optimization without derivatives", in *Large-Scale Nonlinear Optimization*, eds. G. Di Pillo and M. Roma, Springer (New York), pp. 255–297.