DAMTP 1998/NA04

Direct search algorithms for optimization calculations[1]

M.J.D. Powell

Abstract: Many different procedures have been proposed for optimization calculations when first derivatives are not available. Further, several researchers have contributed to the subject, including some who wish to prove convergence theorems, and some who wish to make any reduction in the least calculated value of the objective function. There is not even a key idea that can be used as a foundation of a review, except for the problem itself, which is the adjustment of variables so that a function becomes least, where each value of the function is returned by a subroutine for each trial vector of variables. Therefore the paper is a collection of essays on particular strategies and algorithms, in order to consider the advantages, limitations and theory of several techniques. The subjects that are addressed are line search methods, the restriction of vectors of variables to discrete grids, the use of geometric simplices, conjugate direction procedures, trust region algorithms that form linear or quadratic approximations to the objective function, and simulated annealing. We study the main features of the methods themselves, instead of providing a catalogue of references to published work, because an understanding of these features may be very helpful to future research.

Department of Applied Mathematics and Theoretical Physics,
University of Cambridge,
Silver Street,
Cambridge CB3 9EW,
England.

March, 1998.

[1]To be published in *Acta Numerica*, Vol. 7, Cambridge University Press (1998).

1. Introduction

The author has contributed several Fortran subroutines for optimization calculations to IMSL (International Mathematical and Statistical Libraries), assuming that the gradient of the objective function is available. Then, in all cases, IMSL produced versions of them that make difference approximations to derivatives automatically, as many of its customers would have gone elsewhere if they had been asked to specify the first derivatives. The author did not include the difference approximations himself, because he was aware that they could cause disastrous loss of accuracy in pathological cases, nor did he object strongly to the expediency of IMSL, because he wanted his software to be employed for a wide range of applications. Thus, about ten years ago, the demand from many computer users for numerical methods for optimization was met in an imperfect way that was usually adequate. The main reason for our work is to show that there are still many opportunities for improvements to optimization algorithms that do not require derivatives.

Another fundamental objection to difference approximations, in addition to loss of accuracy due to cancellation and division by small numbers, is that each gradient vector is replaced by a tight cluster of at least $n+1$ function values, where n is the number of variables. Instead, it seems more suitable, intuitively, to spread out the points at which the objective function is calculated, especially if sample values have to be taken from many parts of the space of the variables. Furthermore, when the values include some random noise, then the contribution from the noise to predicted rates of change is less if the evaluation points are widely spaced. Therefore this paper addresses algorithms for optimization that are basically different from the ones that employ gradients. On the other hand, the computer user who requires the best algorithm for his or her application should keep in mind the possibilities of difference approximations and automatic differentiation.

We let $\underline{x} \in \mathcal{R}^n$ denote a typical vector of variables, and we reserve the notation

$$F(\underline{x}), \qquad \underline{x} \in \mathcal{S} \subseteq \mathcal{R}^n, \tag{1.1}$$

for the objective function, where $\mathcal{S}$ may be a subset of $\mathcal{R}^n$ if constraints are present. Further, letting $\mathcal{X}$ be the set of points in $\mathcal{R}^n$ at which the constraints are satisfied, we assume $\mathcal{X} \subseteq \mathcal{S}$. The optimization problem under consideration is to seek a vector $\underline{x}^* \in \mathcal{X}$ that has the property

$$F(\underline{x}^*) \leq F(\underline{x}), \qquad \underline{x} \in \mathcal{X}, \tag{1.2}$$

using calculated values of $F(\underline{x})$ and any constraint functions at points $\underline{x}$ that are chosen automatically. Most of the available algorithms, however, are for the case when the variables are unconstrained. Therefore we follow the usual recourse of mentioning that penalty and barrier function techniques have been developed that replace constrained calculations by unconstrained ones, good descriptions of them being given in the books by Fletcher (1987) and Gill, Murray and Wright (1981). This approach, unfortunately, usually reduces the precision of the information on the final values of the variables that is provided by explicit constraints. Therefore research on new algorithms for nonlinear constraints without the calculation of derivatives could be important to a wide range of applications. We note also that there are some serious deficiencies in the current methods for unconstrained optimization.

Three main techniques are employed for trying to achieve global convergence, namely line searches, trust regions and discrete grids. The suitability of line searches and discrete grids for this purpose is addressed in Sections 2 and 3, respectively, but trust region algorithms are not studied until Sections 6 and 7, as they require approximations to $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$. Section 4 considers simplex methods, including the procedure of Nelder and Mead (1965), because citation indexes show that it is the method of choice for many applications. Fast convergence is often achieved by algorithms that are designed to be highly efficient when the objective function is quadratic. Therefore the use of conjugate directions is the subject of Section 5. In Section 6, linear approximations are made to $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, and to any constraint functions, which give a highly convenient and slow procedure for small numbers of variables. Some of the inefficiencies can be removed by quadratic approximations, as described in Section 7, but current procedures of this kind do not allow nonlinear constraints explicitly. Moreover, some methods make random changes to the variables, including simulated annealing, which is considered in Section 8. Finally, in Section 9, we discuss the convergence properties, the limitations, and some possible developments of the given techniques for minimization without derivatives.

2. Line search methods

Line search methods for unconstrained optimization are iterative, a starting vector of variables $\underline{x}_1 \in \mathcal{R}^n$ has to be given, and, for $k=1,2,3,\ldots$, the k-th iteration derives $\underline{x}_{k+1}$ from $\underline{x}_k$ in the following way. A nonzero search direction $\underline{d}_k \in \mathcal{R}^n$ is chosen. Then the function of one variable $\phi(\alpha) = F(\underline{x}_k + \alpha\,\underline{d}_k)$, $\alpha \in \mathcal{R}$, receives attention, in order to pick a new vector of variables of the form

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k\,\underline{d}_k. \tag{2.1}$$

For example, an 'exact line search' would set the step-length α_k to an α that minimizes $\phi(\alpha)$. In practice, however, one tries to choose α_k in a way that requires very few values of $F(\underline{x}_k + \alpha\,\underline{d}_k)$, $\alpha \in \mathcal{R}$, on each iteration, and it is usual to satisfy the condition

$$F(\underline{x}_{k+1}) \le F(\underline{x}_k), \qquad k=1,2,3,\ldots\,. \tag{2.2}$$

Of course the search directions should be able to explore the full space of the variables. Therefore line search methods should have the property that, for some integer $\ell \geq n$, any ℓ consecutive search directions span $\mathcal{R}^n$ in a strict sense. If this condition failed, then a nonzero $\underline{v} \in \mathcal{R}^n$ would be (nearly) orthogonal to the directions. Therefore a convenient form of the strict sense is that the bound

$$\max\{|\underline{v}^T\underline{d}_j| \,/\, \|\underline{d}_j\|_2 : j = k-\ell+1, k-\ell+2, \ldots, k\} \geq c\,\|\underline{v}\|_2, \qquad \underline{v} \in \mathcal{R}^n, \tag{2.3}$$

is satisfied for $k \geq \ell$, where c is a positive constant. For example, a way of achieving this condition, which gives $\ell = n$ and $c = n^{-1/2}$, is to let each $\underline{d}_k$ be a coordinate direction in $\mathcal{R}^n$, and to cycle round the n coordinate directions recursively as k increases. Rosenbrock (1960) provides an extension of this technique that is sometimes useful. His first n directions are also the coordinate directions, but, when k is any positive integer multiple of n, then, before starting the $(k+1)$-th iteration, he generates $\underline{d}_{k+1}, \underline{d}_{k+2}, \ldots, \underline{d}_{k+n}$ in sequence, by applying the Gram–Schmidt procedure to the differences $\underline{x}_{k+1} - \underline{x}_{k-n+j}$, $j = 1, 2, \ldots, n$. Further, he ensures that every step-length is nonzero, although condition (2.2) may have to fail. Some other choices of search directions are given in Section 5.

Unfortunately, condition (2.3) and exact line searches do not guarantee that limit points of the sequence $\underline{x}_k$, $k = 1, 2, 3, \ldots$, are good estimates of optimal vectors of variables, even if the objective function is continuously differentiable, and the level set $\{\underline{x} : F(\underline{x}) \leq F(\underline{x}_1)\}$ is bounded. Indeed, Powell (1973) gives an example of bad behaviour, with $n = 3$ and exact line searches, where the sequence $\underline{d}_k$, $k = 1, 2, 3, \ldots$, is generated by cycling round the coordinate directions, as mentioned in the previous paragraph. Here, for each integer i in $[1, 6]$, the infinite sequence $\underline{x}_{6j+i}$, $j = 1, 2, 3, \ldots$, tends to one vertex of a cube, and the path from $\underline{x}_k$ to $\underline{x}_{k+6}$ tends to be a cycle along six edges of the cube. Further, the objective function is constant on each of these edges, which implies that two components of the gradient $\underline{\nabla} F$ are zero at each limiting vertex. The other component of $\underline{\nabla} F(\underline{x}_k)$, however, is bounded away from zero for each k. It follows that the calculated vectors of variables do not approach a stationary point of F. Therefore it is easy to modify the algorithm so that the objective function becomes less than the actual limit of the decreasing sequence $F(\underline{x}_k)$, $k \to \infty$. Specifically, we replace $\underline{d}_j$ by a difference approximation to $-\underline{\nabla} F(\underline{x}_j)$ for any integer j that is sufficiently large. Furthermore, we find in the remainder of this section that there is another remedy that does not require an estimate of $\underline{\nabla} F$.

The kind of ingredient that avoids the bad behaviour above is imposing the condition that, if $\|\underline{x}_{k+1} - \underline{x}_k\|$ is bounded away from zero, then $F(\underline{x}_k) - F(\underline{x}_{k+1})$ is bounded away from zero too. Hence, in the usual case when $F(\underline{x}_k)$, $k = 1, 2, 3, \ldots$, converges monotonically, we have the limit

$$\|\underline{x}_{k+1} - \underline{x}_k\| \to 0 \quad \text{as} \quad k \to \infty, \tag{2.4}$$

which prevents the cycling round the edges of the cube in the example of the previous paragraph. Further, if the directions $\underline{d}_j$, $j=1,2,3,\ldots$, satisfy inequality (2.3), and if $\underline{x}^*$ is any limit point of the infinite sequence $\underline{x}_k$, $k=1,2,3,\ldots$, then $\underline{\nabla}F(\underline{x}^*)=0$ can be obtained by a suitable line search, provided that F is continuously differentiable and bounded below. We are going to prove this assertion, not only because the restriction on $\|\underline{x}_{k+1}-\underline{x}_k\|$ may be valuable to future algorithms, but also because the method of proof provides a demonstration of the kind of analysis that can establish convergence properties. A way of achieving the restriction, due to Lucidi and Sciandrone (1997), will be given after the proof.

We aim to deduce a contradiction from the assumption $\|\underline{\nabla}F(\underline{x}^*)\|_2=\eta$, where η is a positive constant and where $\underline{x}^*$ is a limit point of the sequence $\underline{x}_k$, $k=1,2,3,\ldots$, as stated already. We seek some integers j such that $\underline{\nabla}F(\underline{x}_j)^T\underline{d}_j/\|\underline{d}_j\|_2$ is bounded away from zero, because then the step-length α_j of the equation $\underline{x}_{j+1}=\underline{x}_j+\alpha_j\,\underline{d}_j$ can be chosen so that $F(\underline{x}_j)-F(\underline{x}_{j+1})$ is also bounded away from zero, which gives the required contradiction if this happens an infinite number of times. Now, by setting $\underline{v}=\underline{\nabla}F(\underline{x}^*)$ in expression (2.3), we deduce that the inequality

$$|\underline{\nabla}F(\underline{x}^*)^T\underline{d}_j|\,/\,\|\underline{d}_j\|_2 \ge c\,\|\underline{\nabla}F(\underline{x}^*)\|_2 = c\eta \tag{2.5}$$

is achieved at least once for every ℓ consecutive positive integers j. Further, because $\underline{\nabla}F$ is continuous, this inequality implies $|\underline{\nabla}F(\underline{x}_j)^T\underline{d}_j|/\|\underline{d}_j\|_2\ge\frac{1}{2}c\eta$, provided that $\underline{x}_j$ is sufficiently close to $\underline{x}^*$. Specifically, $\underline{x}_j$ is close enough to $\underline{x}^*$ if it satisfies $\|\underline{x}_j-\underline{x}^*\|_2\le\varepsilon$, where ε is a positive constant that provides the property

$$\|\underline{\nabla}F(\underline{x})-\underline{\nabla}F(\underline{x}^*)\|_2 \le \tfrac{1}{2}c\eta \quad \text{if} \quad \|\underline{x}-\underline{x}^*\|_2\le\varepsilon, \tag{2.6}$$

because then the Cauchy–Schwarz inequality and condition (2.5) give the bound

$$\frac{|\underline{\nabla}F(\underline{x}_j)^T\underline{d}_j|}{\|\underline{d}_j\|_2} \ge \frac{|\underline{\nabla}F(\underline{x}^*)^T\underline{d}_j| - |\{\underline{\nabla}F(\underline{x}_j)-\underline{\nabla}F(\underline{x}^*)\}^T\underline{d}_j|}{\|\underline{d}_j\|_2} \ge c\eta-\tfrac{1}{2}c\eta=\tfrac{1}{2}c\eta. \tag{2.7}$$

Therefore it remains to show that, on an infinite number of occasions, ℓ consecutive positive integers j satisfy $\|\underline{x}_j-\underline{x}^*\|_2\le\varepsilon$. The limit (2.4) is helpful, because it admits an integer $j_0>0$ such that $\|\underline{x}_{j+1}-\underline{x}_j\|_2\le\frac{1}{2}\varepsilon/(\ell-1)$ holds for all $j\ge j_0$. Hence, if $\|\underline{x}_k-\underline{x}^*\|_2\le\frac{1}{2}\varepsilon$ occurs for some integer $k\ge j_0$, then $\|\underline{x}_j-\underline{x}^*\|_2\le\varepsilon$ is obtained by every integer j in $[k,k+\ell-1]$. This does happen an infinite number of times, because $\underline{x}^*$ is a limit point of $\underline{x}_k$, $k=1,2,3,\ldots$, even if we require the differences between the chosen integers k to be at least ℓ. The proof is complete.

The line search procedure in Section 5 of Lucidi and Sciandrone (1997) is suitable for the above analysis, although some parameters are required that may be difficult to choose well in practice. They are numbers γ and δ that satisfy $\gamma>0$ and $0<\delta<1$, and a positive sequence $\{\beta_k : k=1,2,3,\ldots\}$ that tends to zero as $k\to\infty$. Then, on each iteration, there is a search for a step-length $\alpha_k=\alpha$ that has the properties

$$\left.\begin{array}{r}F(\underline{x}_k+\alpha\,\underline{d}_k)\leq F(\underline{x}_k)-\gamma\alpha^2\,\|\underline{d}_k\|_2^2\quad\text{and}\\ \min\,[F(\underline{x}_k+\hat{\alpha}\,\underline{d}_k),F(\underline{x}_k-\hat{\alpha}\underline{d}_k)]\geq F(\underline{x}_k)-\gamma\hat{\alpha}^2\,\|\underline{d}_k\|_2^2\end{array}\right\}, \tag{2.8}$$

where $\hat{\alpha}=\alpha/\delta$. If the first line of expression (2.8) holds for a trial $\alpha>0$, then either α is acceptable or the second line shows that a step-length of larger modulus is allowed by the first line, namely α/δ or $-\alpha/\delta$. Thus the modulus of α is increased if necessary, and the second line is tested for the new α. This procedure is continued recursively until α is acceptable, which happens eventually because we are assuming that F is bounded below. Alternatively, if the first line of expression (2.8) fails, not only for the initial α but also for $-\alpha$, then α is replaced by $\alpha\delta$ and these tests are tried again. Thus the second inequality of expression (2.8) is achieved by the new α. Again recursion is applied, either until an acceptable step-length is found or until $\|\alpha\,\underline{d}_k\|_2<\beta_k$ occurs, the choice $\alpha_k=0$ being made in the latter case. Moreover, the search directions $\underline{d}_k$, $k=1,2,3,\ldots$, have to satisfy the strict linear independence condition (2.3). These constructions provide the conclusion $\underline{\nabla}F(\underline{x}^*)=0$ of the previous paragraph, as shown below.

The first line of expression (2.8) and equation (2.1) imply the bound

$$F(\underline{x}_k)-F(\underline{x}_{k+1})\geq\gamma\,\|\underline{x}_k-\underline{x}_{k+1}\|_2^2,\qquad k=1,2,3,\ldots, \tag{2.9}$$

when α_k is positive, and the bound is trivial when α_k is zero. Therefore the limit (2.4) at the beginning of the given analysis is valid, and the conclusion $\underline{\nabla}F(\underline{x}^*)=0$ of the analysis holds, provided that inequality (2.7) causes $F(\underline{x}_j)-F(\underline{x}_{j+1})$ to be bounded away from zero. Further, the method of analysis allows us to restrict attention to values of j that satisfy two more conditions. Firstly, we assume $j\geq j_0$, where j_0 is any fixed positive integer, which may be larger than the j_0 introduced earlier. Thus we allow for the zero step-lengths in the line search procedure under consideration. Secondly, we assume $\|\underline{x}_j-\underline{x}^*\|_2\leq\varepsilon$, although our previous use of this bound was only to establish the existence of integers j that have the property (2.7). Thus the uniform continuity of $\underline{\nabla}F$ in any neighbourhood of $\underline{x}^*$ provides the condition

$$\|\underline{\nabla}F(\underline{x})-\underline{\nabla}F(\underline{x}_j)\|_2\leq\tfrac{1}{4}c\eta\quad\text{if}\quad\|\underline{x}-\underline{x}_j\|_2\leq\hat{\varepsilon}, \tag{2.10}$$

for all of the values of j that are retained, where $\hat{\varepsilon}$ is a positive number that is independent of j.

Now it follows from expressions (2.7) and (2.10) that the gradient of the line search function $\phi(\alpha)=F(\underline{x}_j+\alpha\,\underline{d}_j)$, $\alpha\in\mathcal{R}$, is bounded by the inequality

$$\begin{aligned}|\phi'(\alpha)|=|\underline{d}_j^T\underline{\nabla}F(\underline{x}_j+\alpha\,\underline{d}_j)|\ &\geq\ |\underline{d}_j^T\underline{\nabla}F(\underline{x}_j)|-\|\underline{d}_j\|_2\,\|\underline{\nabla}F(\underline{x}_j+\alpha\,\underline{d}_j)-\underline{\nabla}F(\underline{x}_j)\|_2\\ &\geq\ \tfrac{1}{4}c\eta\,\|\underline{d}_j\|_2\quad\text{if}\quad\|\alpha\,\underline{d}_j\|_2\leq\hat{\varepsilon}.\end{aligned} \tag{2.11}$$

Therefore, by choosing the sign of α to be opposite to the sign of $\phi'(0)$ and by applying $\phi(\alpha) = \int_0^\alpha \phi'(\theta)\, d\theta$, we find the relation

$$F(\underline{x}_j + \alpha\, \underline{d}_j) \leq F(\underline{x}_j) - \tfrac{1}{4} c\eta\, \|\alpha\, \underline{d}_j\|_2 \quad \text{if} \quad \|\alpha\, \underline{d}_j\|_2 \leq \hat{\varepsilon}. \tag{2.12}$$

Thus the first line of expression (2.8) is achieved by every α of the appropriate sign that satisfies $\|\alpha\, \underline{d}_j\|_2 \leq \hat{\varepsilon}$ and $\gamma\, \|\alpha\, \underline{d}_j\|_2 \leq \frac{1}{4} c\eta$. It follows that, if the parameter β_j of the line search procedure is at most $\delta \min[\hat{\varepsilon}, \frac{1}{4} c\eta/\gamma]$, and if the first trial value of α on the j-th iteration is at least β_j, then the procedure provides a step-length α_j that is positive. The first of these conditions is irrelevant if j_0 is sufficiently large, as assumed in the previous paragraph, and any sensible implementation observes the second condition. Therefore both of the inequalities (2.8) hold for $k = j$ with $\alpha = \alpha_j > 0$. We deduce from the second one and from the property (2.12) that $\|\hat{\alpha}\, \underline{d}_j\|_2 = \|\alpha_j\, \underline{d}_j\|_2 / \delta$ is no less than $\min[\hat{\varepsilon}, \frac{1}{4} c\eta/\gamma]$, which gives the inequality

$$\|\underline{x}_{j+1} - \underline{x}_j\|_2 = \|\alpha_j\, \underline{d}_j\|_2 \geq \delta \min[\hat{\varepsilon}, \tfrac{1}{4} c\eta/\gamma]. \tag{2.13}$$

Thus condition (2.9) provides a positive lower bound on $F(\underline{x}_j) - F(\underline{x}_{j+1})$ as required. Therefore line search methods without derivatives can provide convergence properties of the kind that are acclaimed by theoreticians when $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, need not be convex. Some further work on these questions, including new line search procedures, can be found in Grippo, Lampariello and Lucidi (1988) and in Lucidi and Sciandrone (1997).

3. Discrete grid methods

We introduce discrete grid methods by considering a simple procedure in the case when the components of $\underline{x} \in \mathcal{R}^n$ are bounded by the constraints

$$a_i \leq x_i \leq b_i, \qquad i = 1, 2, \ldots, n, \tag{3.1}$$

for given values of a_i and b_i that satisfy $a_i < b_i$, $i = 1, 2, \ldots, n$. For each i, a mesh size $h_i = (b_i - a_i)/\nu_i$ is chosen for some positive integer ν_i, and we let $\mathcal{G}$ be the finite rectangular grid

$$\mathcal{G} = \{\underline{x} : h_i^{-1}(x_i - a_i) \in \mathcal{Z} \cap [0, \nu_i],\ i = 1, 2, \ldots, n\}, \tag{3.2}$$

where $\mathcal{Z}$ denotes the set of integers. Let an iterative algorithm that seeks the least value of $F(\underline{x})$, $\underline{x} \in \mathcal{G}$, be given a starting point $\underline{x}_1 \in \mathcal{G}$, and let the k-th iteration for each k calculate $\underline{x}_{k+1} \in \mathcal{G}$ in a way that satisfies the condition

$$F(\underline{x}_{k+1}) \leq F(\underline{x}_k), \qquad k = 1, 2, 3, \ldots\,. \tag{3.3}$$

Further, let the algorithm terminate if any ℓ consecutive iterations fail to reduce the objective function, where ℓ is a prescribed positive integer. Then the finiteness of $\mathcal{G}$ guarantees that termination occurs. A reason for making this remark is to provide a contrast with the details of the analysis of convergence in the previous section.

An obvious algorithm of this type generates $\underline{x}_{k+1}$ by trying to improve only one component of $\underline{x}_k$ on the k-th iteration, so it makes searches along the coordinate directions $\underline{e}_i$, $i=1,2,\ldots,n$, where the i-th component of $\underline{e}_i$ is one and all the other components are zero. It is usual to cycle round these directions recursively as before, letting the search direction of the k-th iteration be $\underline{e}_i$, where $i\in[1,n]$ is defined by the condition that $(k-i)/n$ is an integer. Further, let the step-length of the k-th iteration be chosen so that the strict reduction $F(\underline{x}_{k+1})<F(\underline{x}_k)$ occurs if $\underline{x}_{k+1}\neq\underline{x}_k$, and so that $\underline{x}_{k+1}\in\mathcal{G}$ has the properties

$$F(\underline{x}_{k+1}) \le F(\underline{x}_{k+1}-h_i\,\underline{e}_i) \qquad \text{and} \qquad F(\underline{x}_{k+1}) \le F(\underline{x}_{k+1}+h_i\,\underline{e}_i), \tag{3.4}$$

except that the first or second of these conditions is dropped if $\underline{x}_{k+1}-h_i\,\underline{e}_i$ or $\underline{x}_{k+1}+h_i\,\underline{e}_i$, respectively, is outside the feasible region, due to the i-th component of $\underline{x}_{k+1}$ being a_i or b_i. Thus, if $\underline{x}^*\in\mathcal{G}$ is the vector of variables that is provided by the algorithm at termination, because $\ell=n$ iterations have not reduced F, then the inequality

$$F(\underline{x}^*) \le F(\underline{x}), \qquad \underline{x}\in\mathcal{N}(\underline{x}^*), \tag{3.5}$$

is satisfied, where $\mathcal{N}(\underline{x}^*)$ is the set of points of $\mathcal{G}$ that are neighbours of $\underline{x}^*$ along coordinate directions, and we include $\underline{x}^*$ in $\mathcal{N}(\underline{x}^*)$ too.

Another grid search algorithm that achieves condition (3.5) is proposed by Hooke and Jeeves (1961). It is usually more efficient than the method of the previous paragraph, because it can generate helpful changes to the variables that are not along coordinate directions. Again $\underline{x}_1\in\mathcal{G}$ is given, but the k-th iteration derives $\underline{x}_{k+1}$ from $\underline{x}_k$ in the following way. If $k\ge 2$ and if the vector $\underline{y}_k=2\underline{x}_k-\underline{x}_{k-1}$ satisfies the constraints (3.1), which implies $\underline{y}_k\in\mathcal{G}$, then the algorithm sets $\underline{y}_k^*$ to a point in $\mathcal{N}(\underline{y}_k)$ that provides the least value of $F(\underline{x})$, $\underline{x}\in\mathcal{N}(\underline{y}_k)$, where $\mathcal{N}(\cdot)$ has been defined already. Further, if $F(\underline{y}_k^*)$ is strictly less than $F(\underline{x}_k)$, then the choice $\underline{x}_{k+1}=\underline{y}_k^*$ is made. In all other cases, however, the algorithm sets $\underline{x}_k^*$ to a point in $\mathcal{N}(\underline{x}_k)$ that minimizes $F(\underline{x})$, $\underline{x}\in\mathcal{N}(\underline{x}_k)$, and $\underline{x}_{k+1}=\underline{x}_k^*$ is chosen if the strict inequality $F(\underline{x}_k^*)<F(\underline{x}_k)$ holds. Otherwise, termination occurs with $\underline{x}^*$ equal to $\underline{x}_k$. Thus the final vector of variables has the property (3.5) as claimed. We note that the method is based on the hypothesis that, because the step from $\underline{x}_{k-1}$ to $\underline{x}_k$ reduces the objective function, the displacement $\underline{x}_k-\underline{x}_{k-1}$ from $\underline{x}_k$ is likely to provide a further reduction, especially if the new point is chosen to be the best one in the set $\mathcal{N}(2\underline{x}_k-\underline{x}_{k-1})=\mathcal{N}(\underline{y}_k)$.

When the objective function has continuous first derivatives, the property (3.5) allows $\underline{x}^*$ to be related to the first order conditions for a minimum of $F(\underline{x})$, $\underline{x}\in\mathcal{X}$, where $\mathcal{X}$ is the set of points in $\mathcal{R}^n$ that satisfy the constraints (3.1). These conditions are that, for each integer i in $[1,n]$, the derivative $\partial F(\underline{x})/\partial x_i$ is zero, nonnegative or nonpositive in the case $a_i < x_i < b_i$, $x_i = a_i$ or $x_i = b_i$, respectively. Now, if $\underline{x}^*{+}h_i\,\underline{e}_i$ is in the set $\mathcal{N}(\underline{x}^*)$, then the property (3.5) includes the inequality $F(\underline{x}^*) \le F(\underline{x}^*{+}h_i\,\underline{e}_i)$. Moreover, the mean value theorem gives the equation

$$F(\underline{x}^*{+}h_i\,\underline{e}_i) = F(\underline{x}^*) + h_i\,\partial F(\underline{x}^*{+}\zeta_i\,\underline{e}_i)/\partial x_i, \tag{3.6}$$

for some ζ_i in $[0, h_i]$. Thus the derivative $\partial F(\underline{x}^*{+}\zeta_i\,\underline{e}_i)/\partial x_i$ is nonnegative, which implies the condition

$$\partial F(\underline{x}^*)/\partial x_i \ge \partial F(\underline{x}^*)/\partial x_i - \partial F(\underline{x}^*{+}\zeta_i\,\underline{e}_i)/\partial x_i \ge -\omega_i(h_i), \tag{3.7}$$

where ω_i is the modulus of continuity of $\partial F(\underline{x})/\partial x_i$, $\underline{x} \in \mathcal{X}$. Similarly, if $\underline{x}^*{-}h_i\,\underline{e}_i$ is in $\mathcal{N}(\underline{x}^*)$, we deduce the bound $\partial F(\underline{x}^*)/\partial x_i \le \omega_i(h_i)$. It follows that the modulus of the difference between the i-th component of $\nabla F(\underline{x}^*)$ and the i-th component of a hypothetical gradient vector that satisfies the first order conditions for optimality at $\underline{x}^*$ is at most $\omega_i(h_i)$ for $i = 1, 2, \ldots, n$.

This result suggests a useful extension of the grid methods described already. It is to apply one of them until termination, and to repeat this procedure recursively, where the mesh sizes are reduced before each new step of the recursion. Specifically, after calculating the vector $\underline{x}^* = \underline{x}^*(h_1, h_2, \ldots, h_n)$ that achieves the property (3.5) for the original mesh sizes, we might halve all the mesh sizes, and then employ the algorithm to generate a new vector $\underline{x}^*$, using the old $\underline{x}^*$ as the starting point for the new calculation. If the number of recursions is infinite, and if $\hat{\underline{x}}^*$ is any limit point of the sequence $\underline{x}^*(h_1, h_2, \ldots, h_n)$ as $\max\{h_i : i = 1, 2, \ldots, n\}$ tends to zero, then it follows from the previous paragraph that $\hat{\underline{x}}^*$ satisfies the first order conditions for the least value of $F(\underline{x})$, $\underline{x} \in \mathcal{X}$.

Of course there are analogous procedures for unconstrained calculations and for the case when there are bounds on some but not all of the variables. We continue to employ a rectangular grid $\mathcal{G}$ in these cases, and to let h_i denote the mesh size in the i-th coordinate direction for each integer i in $[1, n]$. Further, the starting point $\underline{x}_1$ is still required to be a grid point, and we let its components be ξ_i, $i = 1, 2, \ldots, n$. Therefore $\underline{x}$ is a point of $\mathcal{G}$ if and only if it satisfies any constraints and if all the ratios $(x_i - \xi_i)/h_i$, $i = 1, 2, \ldots, n$, are integers. These parameters should be chosen so that any bound on $\underline{x}$ holds as an equation at some of the grid points. Then condition (3.5) can be achieved as before when the level set $\{\underline{x} : F(\underline{x}) \le F(\underline{x}_1),\ \underline{x} \in \mathcal{X}\}$ is bounded.

On the other hand, rectangular grid methods are hardly ever suitable for constraints that are more general than the simple bounds (3.1). For example, let $n = 2$, let $F(\underline{x}) = x_1 + 2x_2$, $\underline{x} \in \mathcal{R}^2$, let the constraints be $-10 \le x_1 \le 10$, $x_2 \le 10$ and $x_1 + x_2 + 10 \ge 0$, let $h_1 = h_2 = 1$, and let the starting point $\underline{x}_1$ be at the origin. Then $\mathcal{G}$ includes many points on the boundary of $\mathcal{X}$, and the objective function can be reduced by decreases in x_1 and x_2. Thus it is likely that a point with the coordinates $(x_1, -x_1 - 10)$ will be calculated for some integer x_1 in $[-10, 0]$, because this point is on the boundary of the last constraint. Here a decrease in x_1 or x_2 would violate the constraint, while an increase in x_1 or x_2 would increase the value of the objective function. Therefore termination is likely to occur with $\underline{x}_1 \leq 0$, although the optimal vector of variables is the grid point with the coordinates $(10, -20)$. The deficiency happens because further progress requires a search direction that is along or close to the boundary of the last constraint, but the direction of that boundary in $\mathcal{R}^2$ is very different from both of the coordinate directions.

If $x_1 + x_2 + 10 \geq 0$ were the only constraint on the variables in the above example, then a linear change of variables $\underline{y} = B\underline{x}$ could be made, where B is an $n \times n$ nonsingular matrix that provides $y_1 = x_1 + x_2$. Then any of the methods described already could be applied to seek the least value of $F(B^{-1}\underline{y})$ subject to $y_1 \geq -10$. The notation B for the matrix of the change of variables is taken from Torczon (1997), because she develops a general convergence theory for discrete grid methods.

That work allows both increases and decreases in mesh sizes, assuming that the variables are unconstrained and that the level set $\{\underline{x} : F(\underline{x}) \leq F(\underline{x}_1)\}$ is bounded. We consider the technique when B is the unit matrix and when the mesh sizes satisfy $h_1 = h_2 = \cdots = h_n = h$, say. The value of h can be adjusted automatically, and we let its initial value be $h = 1$. Then the technique includes the following four important features. Firstly, every mesh size is an integral power of a rational number $\tau = \beta/\alpha < 1$, where α and β are prescribed positive integers that are relatively prime. Secondly, if the k-th iteration gives $\underline{x}_{k+1} \neq \underline{x}_k$, then every component of the difference $\underline{x}_{k+1} - \underline{x}_k$ is an integral multiple of the current h and the strict inequality $F(\underline{x}_{k+1}) < F(\underline{x}_k)$ is achieved. Thirdly, $\underline{x}_{k+1} = \underline{x}_k$ occurs only if $\underline{x}_k$ has the property

$$F(\underline{x}_k) \leq F(\underline{x}), \qquad \underline{x} \in \mathcal{N}_h(\underline{x}_k), \tag{3.8}$$

where $\mathcal{N}_h$ is the $\mathcal{N}$ that has been defined already, the subscript denoting the mesh size. Fourthly, the k-th iteration does not reduce h if $F(\underline{x}_{k+1})$ is less than $F(\underline{x}_k)$, but otherwise the mesh size of the $(k+1)$-th iteration is the mesh size of the k-th iteration multiplied by τ.

The main conclusion of Torczon (1997) is that, if the objective function has continuous first derivatives and if the number of iterations is infinite, then the given conditions imply the convergence property

$$\liminf_{k \to \infty} \|\underline{\nabla} F(\underline{x}_k)\|_2 = 0. \tag{3.9}$$

It can be proved by letting $\mathcal{K}$ be the set of integers k such that the mesh size of the $(k+1)$-th iteration is less than all previous mesh sizes, and by deducing that the number of elements of $\mathcal{K}$ is infinite. Indeed, if we suppose that $|\mathcal{K}|$ is finite, then we can let ℓ and m be integers such that every mesh size is in the set $\{\tau^i : i = \ell, \ell+1, \ldots, m\}$, where $\ell \leq 0 \leq m$, the existence of ℓ being due to the boundedness of the level set $\{\underline{x} : F(\underline{x}) \leq F(\underline{x}_1)\}$. It follows from $\tau = \beta/\alpha$ that, for every iteration number k, the components of the difference $\underline{x}_{k+1} - \underline{x}_1$ are integral multiples of the positive constant $\beta^\ell \alpha^{-m}$. Thus, due to the boundedness of the level set, the monotonically decreasing sequence $\{F(\underline{x}_k) : k = 1, 2, 3, \ldots\}$ satisfies $F(\underline{x}_{k+1}) < F(\underline{x}_k)$ for only a finite number of values of k. Therefore there exists k_0 such that the k-th iteration multiplies h by τ for all $k \geq k_0$, which contradicts the hypothesis $|\mathcal{K}| < \infty$. Hence the sequence $\{\underline{x}_k : k \in \mathcal{K}\}$ is infinite. Further, the given conditions and choices imply that, for every positive integer j, the j-th element of this sequence has the property (3.8) with $h = \tau^{j-1}$. Thus the argument of the paragraph containing expressions (3.6) and (3.7) gives the bound

$$|\partial F(\underline{x}_k)/\partial x_i| \leq \omega_i(\tau^{j-1}), \qquad i = 1, 2, \ldots, n. \tag{3.10}$$

Therefore the infinite sequence $\{\|\underline{\nabla} F(\underline{x}_k)\|_2 : k \in \mathcal{K}\}$ converges to zero, which completes the proof of the assertion (3.9).

This method of proof does not show, however, that a small value of $\|\underline{\nabla} F(\underline{x}_k)\|_2$ is achieved by a number of iterations that is suitable for practical computation. For example, if $\tau = 2/3$, if $h \approx 10^{-6}$ is needed to provide adequate accuracy, and if $h \approx 100$ is the greatest mesh size that occurs, then the formulae $(2/3)^{34} = 1.03 \times 10^{-6}$ and $(3/2)^{11} = 86.5$ imply that the components of $\underline{x}_{k+1} - \underline{x}_1$ should be integral multiples of $(1/3)^{34}(1/2)^{11} = 2.93 \times 10^{-20} = \eta_{\min}$, say. Then it is crucial to the last paragraph that $\eta_{\min}$ is a positive constant, although its value is less than the usual relative precision of computer arithmetic. Therefore it is hoped that there is no close relation between performance in practice and the importance of the rationality of τ to the given analysis. A strong advantage of discrete grid methods is emphasised by Torczon (1997). It is that any trial step from $\underline{x}_k$ to another grid point is allowed by the convergence theory if it provides the strict inequality $F(\underline{x}_{k+1}) < F(\underline{x}_k)$. On the other hand, proofs of convergence of several other algorithms require the reduction in the objective function to be 'sufficiently large'.

4. Simplex methods

A simplex is the convex hull of $n+1$ points in $\mathcal{R}^n$, where the points satisfy the nondegeneracy condition that the volume of the hull is nonzero. This condition is equivalent to the linear independence of the vectors $\underline{v}_i - \underline{v}_1$, $i = 2, 3, \ldots, n+1$, where $\underline{v}_i \in \mathcal{R}^n$, $i = 1, 2, \ldots, n+1$, are the $n+1$ points. Simplex methods for unconstrained optimization are iterative. A simplex is available at the beginning of each iteration, and we let $\underline{v}_i$, $i = 1, 2, \ldots, n+1$, be its vertices. Further, the values $F(\underline{v}_i)$, $i = 1, 2, \ldots, n+1$, of the objective function are known. The iteration picks an integer m from $[1, n+1]$, that is usually defined by the property

$$F(\underline{v}_m) \geq F(\underline{v}_i), \qquad i=1,2,\ldots,n+1, \tag{4.1}$$

in order that $F(\underline{v}_m)$ is the worst of the function values. Further, a new vector of variables, $\hat{\underline{v}}_m$ say, is generated, a frequent choice being the point

$$\hat{\underline{v}}_m = (2/n) \sum_{\substack{i=1 \\ i \neq m}}^{n+1} \underline{v}_i - \underline{v}_m, \tag{4.2}$$

because $\hat{\underline{v}}_m$ is on the straight line from $\underline{v}_m$ through the centroid of the other vertices. Then $F(\hat{\underline{v}}_m)$ is calculated. Its value may allow the iteration to be completed by letting the simplex for the next iteration be the current one, except that $\underline{v}_m$ is replaced by $\hat{\underline{v}}_m$. Another possibility is that $\hat{\underline{v}}_m$ may be revised in a way that depends on the new function value, and then $F(\hat{\underline{v}}_m)$ is required at the revised point. Moreover, it happens occasionally that the new information causes the current simplex to shrink, in which case $\underline{v}_i$ is overwritten by $\frac{1}{2}(\underline{v}_i+\underline{v}_\ell)$ for $i=1,2,\ldots,n+1$, where ℓ is an integer from $[1, n+1]$ such that $F(\underline{v}_\ell)$ is the least of the numbers $F(\underline{v}_i)$, $i=1,2,\ldots,n+1$. Thus the new current simplex has n new vertices, and the objective function has to be calculated at all of them before the next iteration is begun.

The first algorithm of this kind was proposed by Spendley, Hext and Himsworth (1962). We consider a version of their method that employs formula (4.2) on every iteration. If $F(\hat{\underline{v}}_m) \geq F(\underline{v}_m)$ occurs, then the calculation is terminated if the simplex has become sufficiently small, and otherwise the iteration shrinks the simplex. Alternatively, when $F(\hat{\underline{v}}_m)$ is less than $F(\underline{v}_m)$, then the only modification to the current simplex for the next iteration is the replacement of $\underline{v}_m$ by $\hat{\underline{v}}_m$. A small departure from the procedure of the previous paragraph assists the achievement of the condition $F(\hat{\underline{v}}_m) < F(\underline{v}_m)$. Specifically, if the usual choice of m would cause the vector (4.2) to be the vertex that was deleted from the previous simplex by the previous iteration, then m is chosen to be an integer in $[1, n+1]$ such that $F(\underline{v}_m)$ is the second largest of the numbers $F(\underline{v}_i)$, $i=1,2,\ldots,n+1$. Thus the same value of m is not picked by two consecutive iterations, unless a shrink is applied by the earlier iteration. Furthermore, this algorithm has the property that every simplex is regular if the initial simplex is regular, where a simplex is regular if and only if all its edges have the same length.

The last remark is relevant to an analysis of convergence in the general case. Indeed, let the method of the previous paragraph be applied for several iterations to the function $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, where the simplex at the beginning of the calculation has the vertices $\underline{v}_i^{(0)}$, $i=1,2,\ldots,n+1$. Further, let $\underline{w}_i^{(0)}$, $i=1,2,\ldots,n+1$, be the vertices of any regular simplex in $\mathcal{R}^n$, and then define the $n \times n$ nonsingular matrix B by the equations

$$\underline{w}_i^{(0)} - \underline{w}_1^{(0)} = B\,(\underline{v}_i^{(0)} - \underline{v}_1^{(0)}), \qquad i=2,3,\ldots,n+1. \tag{4.3}$$

We now consider applying the method of the last paragraph to the function $G(\underline{y}) = F(\underline{v}_1^{(0)} + B^{-1}\{\underline{y} - \underline{w}_1^{(0)}\})$, $\underline{y} \in \mathcal{R}^n$, starting with the regular simplex that has just been mentioned. The identities $G(\underline{w}_i^{(0)}) = F(\underline{v}_i^{(0)})$, $i=1,2,\ldots,n{+}1$, imply that the initial choice of m is the same as before. Thus, for the new calculation, formula (4.2) provides the vector

$$\hat{\underline{w}}_m^{(0)} = (2/n) \sum_{\substack{i=1\\ i\neq m}}^{n+1} \underline{w}_i^{(0)} - \underline{w}_m^{(0)}. \tag{4.4}$$

Further, if $\hat{\underline{v}}_m^{(0)}$ is the vector (4.2) that occurs on the first iteration of the original calculation, then $\hat{\underline{w}}_m^{(0)}$ is related to $\hat{\underline{v}}_m^{(0)}$ by the identity

$$\begin{aligned} \underline{v}_1^{(0)} + B^{-1}\left\{\hat{\underline{w}}_m^{(0)} - \underline{w}_1^{(0)}\right\} \;&=\; \underline{v}_1^{(0)} + B^{-1}\Big\{ (2/n) \sum_{\substack{i=1\\ i\neq m}}^{n+1} (\underline{w}_i^{(0)} - \underline{w}_1^{(0)}) - (\underline{w}_m^{(0)} - \underline{w}_1^{(0)}) \Big\} \\ &=\; \underline{v}_1^{(0)} + (2/n) \sum_{\substack{i=1\\ i\neq m}}^{n+1} (\underline{v}_i^{(0)} - \underline{v}_1^{(0)}) - (\underline{v}_m^{(0)} - \underline{v}_1^{(0)}) \;=\; \hat{\underline{v}}_m^{(0)}. \end{aligned} \tag{4.5}$$

It follows that the function values $G(\hat{\underline{w}}_m^{(0)})$ and $F(\hat{\underline{v}}_m^{(0)})$ are the same. Hence the simplex at the beginning of the second iteration of the new calculation is related to the simplex at the beginning of the second iteration of the old calculation by the change of variables $\underline{y} = \underline{w}_1^{(0)} + B(\underline{x} - \underline{v}_1^{(0)})$, $\underline{x} \in \mathcal{R}^n$. Further, it can be shown by induction that this relation is inherited by the simplices of all iterations. Therefore, when analysing convergence, there is no loss of generality in assuming that every simplex is regular. It can be shown similarly that there is also no loss of generality in making any nondegenerate choice of the initial simplex.

One fundamental convergence question is whether the method of the second paragraph of this section can continue indefinitely without any shrinks, when the level set $\{\underline{x} : F(\underline{x}) \leq F_0\}$ is bounded, where F_0 is the largest of the function values at the vertices of the initial simplex. The grid argument of the previous section provides a negative answer when $n=2$. Indeed, in this case there is no loss of generality in letting the initial vertices have the components $\underline{v}_1^{(0)} = (0,0)$, $\underline{v}_2^{(0)} = (1,0)$ and $\underline{v}_3^{(0)} = (0,1)$. Thus it follows from formula (4.2) with $n=2$ that, until the first shrink occurs, the components of every vertex of every simplex are integers. Moreover, every iteration before a shrink provides a strict reduction in the sum $\sum_{i=1}^{n+1} F(\underline{v}_i)$, where the points $\underline{v}_i$, $i=1,2,\ldots,n{+}1$, are still the vertices of the current simplex. Therefore the shrink is guaranteed by the remark that the number of different values of this sum subject to the conditions $F(\underline{v}_i) \leq F_0$ and $\underline{v}_i \in \mathcal{Z}^n$ is finite, where $\mathcal{Z}^n$ is the set of vectors in $\mathcal{R}^n$ whose components are integers.

When $n \geq 3$, however, then the recursive use of formula (4.2) can generate an infinite number of different vertices of simplices in bounded regions of $\mathcal{R}^n$. For example, let $n=3$, let the initial simplex be a regular tetrahedron, and let two of its vertices have the components $\underline{v}_1=(0,0,0)$ and $\underline{v}_2=(1,0,0)$. We consider the replacement of $\underline{v}_m$ by $\hat{\underline{v}}_m$ recursively in the case when $\underline{v}_1$ and $\underline{v}_2$ remain as vertices of every simplex that occurs. We know that all these simplices are regular tetrahedra and that any two consecutive ones have a common face. Therefore each replacement is equivalent to rotating the current simplex about the x-axis through an angle θ that satisfies $\cos\theta = 1/3$ and $-\pi/2 < \theta < \pi/2$. We let every θ be positive. Hence all the simplices are different if and only if the ratio $k\theta/(2\pi)$ is not an integer for every positive integer k. This happens because such values of k are excluded by the conditions $\cos\theta = 1/3$ and $\cos(k\theta) = 1$. Indeed, if $\cos(k\theta)$ is expressed as a polynomial of degree k in $\cos\theta$, then the coefficient of $(\cos\theta)^k$ is 2^{k-1}, which is not divisible by three, and all the coefficients are integers. Therefore $\cos\theta = 1/3$ cannot be an exact solution of $\cos(k\theta) = 1$, which completes the analysis of the example. The author does not know the answer to the question whether, for $n \geq 3$, the simplex method under consideration always includes a shrink in exact arithmetic, provided that the objective function has bounded level sets and suitable smoothness properties. On the other hand, a shrink is guaranteed in practice, because the number of different values of $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, in computer arithmetic is finite.

Another simplex algorithm is proposed by Dennis and Torczon (1991). It has the property that, due to a grid argument of the kind given already, shrinks occur for any number of variables, provided that level sets of the objective function are bounded. Each iteration of this algorithm begins as before with a simplex whose vertices are $\underline{v}_i$, $i=1,2,\ldots,n+1$, and with the function values $F(\underline{v}_i)$, $i=1,2,\ldots,n+1$. Then the basic version of an iteration finds an integer ℓ in $[1,n+1]$ such that $F(\underline{v}_\ell)$ is the least of these function values. Now the simplex that has the vertices

$$\hat{\underline{v}}_i = 2\,\underline{v}_\ell - \underline{v}_i, \qquad i=1,2,\ldots,n+1, \tag{4.6}$$

is congruent to the current one, and a switch to this simplex is likely to reduce the least function value at a vertex, because, for each i, the inequality $F(\underline{v}_\ell) \leq F(\underline{v}_i)$ is satisfied, and $\underline{v}_\ell$ is the mid-point of the line segment from $\underline{v}_i$ to $\hat{\underline{v}}_i$. Therefore the algorithm calculates the objective function at the n vertices of the new simplex that are different from $\underline{v}_\ell = \hat{\underline{v}}_\ell$. It is possible, however, that the new function values fail to achieve the strict reduction

$$\min\{F(\hat{\underline{v}}_i) : i=1,2,\ldots,n+1\} < F(\underline{v}_\ell). \tag{4.7}$$

When this unsuccessful case occurs, then, as before, either the procedure is terminated because the current simplex is sufficiently small, or there is a shrink that overwrites $\underline{v}_i$ by $\frac{1}{2}(\underline{v}_i+\underline{v}_\ell)$ for $i=1,2,\ldots,n+1$. Alternatively, when condition (4.7) holds, then the basic iteration includes a test for doubling the size of the simplex by choosing the vertices $\check{\underline{v}}_i = 2\,\hat{\underline{v}}_i - \underline{v}_\ell$, $i=1,2,\ldots,n+1$, so the n function values $F(\check{\underline{v}}_i)$, $i \neq \ell$, are calculated too. Then the inequality

$$\min\{F(\check{\underline{v}}_i) : i=1,2,\ldots,n+1\} < \min\{F(\hat{\underline{v}}_i) : i=1,2,\ldots,n+1\} \tag{4.8}$$

is tried. If the conditions (4.7) and (4.8) are both satisfied, the points $\check{\underline{v}}_i$, $i = 1,2,\ldots,n+1$, become the vertices of the simplex of the next iteration, but, in the remaining case when inequalities (4.7) and (4.8) hold and fail, respectively, the vertices of the new simplex are $\hat{\underline{v}}_i$, $i=1,2,\ldots,n+1$. The fact that each iteration requires $2n$ evaluations of the objective function may be an advantage in practice, because the algorithm is designed to be suitable for parallel machines.

When analysing the convergence of this method, there is no loss of generality in selecting every vertex of the initial simplex from $\mathcal{Z}^n$. Hence, if σ_k and τ_k are the number of times that the size of the simplex is shrunk or doubled, respectively, during the first k iterations, and if μ_k is the greatest of the differences $\sigma_j - \tau_j$, $j=0,1,\ldots,k-1$, where $\sigma_0=\tau_0=0$, then, at the beginning of the k-th iteration, every component of every vertex of the current simplex is an integer multiple of $2^{-\mu_k}$. Thus, under the usual assumption that the set $\{\underline{x} : F(\underline{x}) \leq F_0\}$ is bounded, a finite upper bound on the increasing sequence of integers $\{\mu_k : k=1,2,3,\ldots\}$ would provide a contradiction if the number of iterations were infinite. Therefore a simplex that is small enough to give termination occurs eventually. When this happens, the algorithm provides the relations

$$F(\underline{v}_\ell) \leq F(\underline{v}_i), \quad F(\underline{v}_\ell) \leq F(\hat{\underline{v}}_i) \quad \text{and} \quad \underline{v}_\ell = \tfrac{1}{2}(\underline{v}_i + \hat{\underline{v}}_i), \quad i=1,2,\ldots,n+1. \tag{4.9}$$

Further, the shape of the current simplex does not degenerate during the calculation because the shapes of all the simplices are the same. Thus the conditions (4.9) are analogous to expression (3.8). It follows that, if the objective function is continuously differentiable, and if the property $\|\underline{\nabla} F(\underline{v}_\ell)\|_2 \leq \varepsilon$ is required in exact arithmetic, where ε is any prescribed positive number, then it is suitable to let the final simplex be sufficiently small in the test for termination.

The algorithm in the second paragraph of this section possesses a similar property, except that, for $n \geq 3$, it is an assumption that enough shrinks are made to achieve the termination condition for any choice of the final size of the simplex. The following analysis gives an explanation of this property in the case when every simplex is regular. Let termination occur because $F(\hat{\underline{v}}_m) \geq F(\underline{v}_m)$ holds, and because the lengths of the edges of the current simplex are at most δ, which is a prescribed positive number. We may assume that the inequalities (4.1) are satisfied, because, if the alternative choice of m were made, then a switch to the usual choice would cause the new $\hat{\underline{v}}_m$ to be the vertex of the previous simplex that was replaced by $\underline{v}_m$, so the condition $F(\hat{\underline{v}}_m) \geq F(\underline{v}_m)$ is retained. We let $\underline{v}_*$ and $\underline{g}_*$ be the vectors $n^{-1} \sum_{i=1,\, i\neq m}^{n+1} \underline{v}_i$ and $\underline{\nabla} F(\underline{v}_*)$, respectively. Therefore, because the objective function has a continuous gradient in a bounded level set, the formula

$$F(\underline{x}) = F(\underline{v}_*) + (\underline{x}-\underline{v}_*)^T \underline{g}_* + o(\|\underline{x}-\underline{v}_*\|_2), \tag{4.10}$$

is valid for all points $\underline{x}$ that are of interest. Hence, by letting $\underline{x}$ run through the vertices of the current simplex that are different from $\underline{v}_m$ and averaging, we deduce the equation

$$F(\underline{v}_*) = (1/n) \sum_{\substack{i=1 \\ i \neq m}}^{n+1} F(\underline{v}_i) + o(\delta). \tag{4.11}$$

It follows from expression (4.1) that $F(\underline{v}_m) - F(\underline{v}_*)$ is at least $o(\delta)$, so expression (4.10) implies $(\underline{v}_m - \underline{v}_*)^T \underline{g}_* \geq o(\delta)$. Similarly, $F(\hat{\underline{v}}_m) \geq F(\underline{v}_m)$ implies $(\hat{\underline{v}}_m - \underline{v}_*)^T \underline{g}_* \geq o(\delta)$. Therefore, by writing equation (4.2) in the form $\hat{\underline{v}}_m - \underline{v}_* = -(\underline{v}_m - \underline{v}_*)$, we establish $(\underline{v}_m - \underline{v}_*)^T \underline{g}_* = o(\delta)$. Furthermore, by combining this result with conditions (4.1) and (4.10), we find the bounds

$$\begin{aligned} (\underline{v}_i - \underline{v}_*)^T \underline{g}_* \;&=\; F(\underline{v}_i) - F(\underline{v}_*) + o(\delta) \;\leq\; F(\underline{v}_m) - F(\underline{v}_*) + o(\delta) \\ &=\; (\underline{v}_m - \underline{v}_*)^T \underline{g}_* + o(\delta) \;=\; o(\delta), \qquad i = 1, 2, \ldots, n+1. \end{aligned} \tag{4.12}$$

Let j be any integer in $[1, n+1]$ that is different from m, and let $\hat{\underline{v}}_j$ be the point where the straight line from $\underline{v}_j$ through $\underline{v}_*$ leaves the current simplex. Then $\|\hat{\underline{v}}_j - \underline{v}_*\|_2$ is of magnitude δ and, because $\hat{\underline{v}}_j$ is in the convex hull of the vertices, it has the form $\hat{\underline{v}}_j = \sum_{i=1}^{n+1} \theta_i \underline{v}_i$, where the multipliers θ_i, $i = 1, 2, \ldots, n+1$, are nonnegative and sum to one. Thus the bounds (4.12) give the inequality

$$(\hat{\underline{v}}_j - \underline{v}_*)^T \underline{g}_* = \sum_{i=1}^{n+1} \theta_i \, (\underline{v}_i - \underline{v}_*)^T \underline{g}_* \leq o(\delta). \tag{4.13}$$

Further, because the directions $\underline{v}_j - \underline{v}_*$ and $\hat{\underline{v}}_j - \underline{v}_*$ are exactly opposite to each other, expressions (4.12) and (4.13) imply the stronger condition $(\underline{v}_j - \underline{v}_*)^T \underline{g}_* = o(\delta)$, which holds for every integer j in $[1, n+1]$. Therefore $\underline{g}_*$ satisfies the equations

$$(\underline{v}_i - \underline{v}_1)^T \underline{g}_* = o(\delta), \qquad i = 2, 3, \ldots, n+1. \tag{4.14}$$

Thus all of the ratios $(\underline{v}_i - \underline{v}_1)^T \underline{g}_* \,/\, \|\underline{v}_i - \underline{v}_1\|_2$, $i = 2, 3, \ldots, n+1$, tend to zero if the lengths of the edges of the simplices are shrunk to zero. It follows from the uniform linear independence of the vectors $\underline{v}_i - \underline{v}_1$, $i = 2, 3, \ldots, n+1$, that $\|\underline{g}_*\|_2$ tends to zero too. Hence, for any positive number ε, the termination condition implies $\|\underline{g}_*\|_2 \leq \varepsilon$ in exact arithmetic, provided that δ is sufficiently small.

We consider also the algorithm of Nelder and Mead (1965), because it has become the most popular simplex method in practice for unconstrained optimization. The introduction to that paper states "In the method to be described the simplex adapts itself to the local landscape, elongating down long inclined planes, changing direction on encountering a valley at an angle, and contracting in the neighbourhood of a minimum". These properties are often, but not always, achieved in numerical experiments. An iteration of the algorithm, with typical parameter values, has all the features that are mentioned in the opening paragraph of this section, the other details being as follows. Let $F(\underline{v}_\ell)$ and $F(\underline{v}_{\hat{m}})$ be the least and second largest of the numbers $F(\underline{v}_i)$, $i=1,2,\ldots,n+1$. If the function value at the point (4.2) satisfies the conditions $F(\underline{v}_\ell) \leq F(\hat{\underline{v}}_m) < F(\underline{v}_{\hat{m}})$, then $\underline{v}_m$ is overwritten by $\hat{\underline{v}}_m$ and the next iteration is begun. Otherwise, if $F(\hat{\underline{v}}_m) < F(\underline{v}_\ell)$ occurs, then elongation is attempted by calculating $F(\check{\underline{v}}_m)$, where $\check{\underline{v}}_m$ is the point $\hat{\underline{v}}_m + \frac{1}{2}(\hat{\underline{v}}_m - \underline{v}_m)$. Further, $\underline{v}_m$ is replaced by $\check{\underline{v}}_m$ or $\hat{\underline{v}}_m$ in the case $F(\check{\underline{v}}_m) < F(\hat{\underline{v}}_m)$ or $F(\check{\underline{v}}_m) \geq F(\hat{\underline{v}}_m)$, respectively, which provides the simplex for the next iteration. In the remaining situation $F(\hat{\underline{v}}_m) \geq F(\underline{v}_{\hat{m}})$, the inequality $F(\hat{\underline{v}}_m) < F(\underline{v}_m)$ is tested. If it holds or fails then $\hat{\underline{v}}_m$ is overwritten by $\frac{3}{4}\hat{\underline{v}}_m + \frac{1}{4}\underline{v}_m$ or $\frac{1}{4}\hat{\underline{v}}_m + \frac{3}{4}\underline{v}_m$, respectively, and $F(\hat{\underline{v}}_m)$ is calculated at the new point. Then $\hat{\underline{v}}_m$ replaces $\underline{v}_m$ as before if and only if the new $F(\hat{\underline{v}}_m)$ is strictly less than $F(\underline{v}_m)$, the alternative being to shrink the simplex in the way that has been described already. The calculation may be terminated when the lengths of the edges of the current simplex become less than a prescribed positive number. Some of these details are taken from Wright (1996).

That paper includes an excellent discussion of the limitations, disadvantages, successes and developments of the Nelder and Mead algorithm. The fact that literature searches show that it is the most used method for unconstrained optimization in practice is remarkable, because some severe cases of failure have been found. One source of trouble is that a sequence of iterations can cause the volume of the current simplex to tend to zero, when the length of the longest edge of the simplex is bounded below by a positive constant. In particular, let $n = 2$, let the vertices of the current simplex have the coordinates $(\alpha, 0)$, $(0, -1)$ and $(0, 1)$, where α is nonzero, and let F be any smooth function with the following properties. The form of F on the x-axis, namely $F(x, 0)$, $x \in \mathcal{R}$, is strictly convex and is least at $x=0$. Further, the inequalities

$$F(0,-1) \leq F(0,0) \quad \text{and} \quad F(0,1) \leq F(0,0) \tag{4.15}$$

are satisfied. Then formula (4.2) provides $\hat{\underline{v}}_m = (-\alpha, 0)$, which gives the $F(\hat{\underline{v}}_m) \geq F(\underline{v}_{\hat{m}})$ situation, so the components of $\hat{\underline{v}}_m$ are altered to $(-\frac{1}{2}\alpha, 0)$ or $(\frac{1}{2}\alpha, 0)$. Now the choice between these alternatives by the algorithm, the convexity condition and $F(0,0) < F(\alpha, 0)$ imply that the new value of $F(\hat{\underline{v}}_m)$ is strictly less than $F(\underline{v}_m)$. Thus the outcome of the iteration is the replacement of α by $-\frac{1}{2}\alpha$ or $\frac{1}{2}\alpha$. Such iterations can continue indefinitely, and then the current simplex tends to have the vertices $(0,0)$, $(0,-1)$ and $(0,1)$. This result is particularly unfortunate, because the modulus of the second component of the gradient $\underline{\nabla} F(\underline{x})$ can be large at all the vertices of all the simplices that occur during the calculation.

The function $F(\underline{x})$, $\underline{x} \in \mathcal{R}^2$, can be convex in the example, provided that both of the conditions (4.15) hold as equations. Then the assumptions imply that the least value of the objective function occurs at the origin, so the performance of the algorithm is adequate. On the other hand, McKinnon (1997) constructs a more interesting case with $n=2$, where the objective function is strictly convex and has continuous second derivatives, where the gradient vector $\underline{\nabla} F(0,0)$ is nonzero, where the number of iterations of the Nelder and Mead (1965) algorithm is infinite, and where all the vertices of the current simplex tend to the origin. Further, this unacceptable behaviour occurs without any shrinks, which is possible because the directions of the edges of the current simplex tend to be orthogonal to $\underline{\nabla} F(0,0)$. These examples are disturbing, and they provide some very strong reasons for questioning the current use of simplex methods for unconstrained optimization calculations.

5. Conjugate direction methods

As mentioned already, conjugate direction methods apply line searches, and they are designed to be efficient for unconstrained minimization when $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, is a strictly convex quadratic function. Further, it is hoped that some of the efficiency of the quadratic case will be inherited when the objective function is general. We assume the quadratic form

$$F(\underline{x}) = F_0 + \underline{x}^T \underline{g}_0 + \tfrac{1}{2}\, \underline{x}^T A\, \underline{x}, \qquad \underline{x} \in \mathcal{R}^n, \tag{5.1}$$

when defining conjugate directions, where A is a constant positive definite symmetric matrix. Specifically, the nonzero vectors $\underline{d}_i \in \mathcal{R}^n$ and $\underline{d}_j \in \mathcal{R}^n$ are conjugate if and only if they satisfy the equation

$$\underline{d}_i^T A\, \underline{d}_j = 0. \tag{5.2}$$

The most useful consequence of conjugacy is that, if F is the quadratic function (5.1), if the k-th iteration of an algorithm is making a line search from $\underline{x}_k$ along the direction $\underline{d}_k$, and if $\underline{d}$ is any direction conjugate to $\underline{d}_k$, then the scalar product $\underline{d}^T \underline{\nabla} F(\underline{x}_k + \alpha\, \underline{d}_k)$ is independent of the step-length $\alpha \in \mathcal{R}$ of the line search, which can be verified easily. In particular, if $\underline{d}^T \underline{\nabla} F(\underline{x}_k) = 0$ holds, then it follows from formula (2.1) that $\underline{d}^T \underline{\nabla} F(\underline{x}_{k+1}) = 0$ is achieved too. Moreover, an exact line search along $\underline{d}_k$ provides $\underline{d}_k^T \underline{\nabla} F(\underline{x}_{k+1}) = 0$. Thus, if each iteration makes an exact line search, and if each search direction is conjugate to all the previous search directions, then $\underline{x}_{k+1}$ enjoys the property

$$\underline{d}_j^T \underline{\nabla} F(\underline{x}_{k+1}) = 0, \qquad j = 1, 2, \ldots, k, \tag{5.3}$$

for every positive integer k. Further, if $\underline{\nabla} F(\underline{x}_{k+1})$ is nonzero, then the possible choice $\underline{d}_{k+1} = -A^{-1} \underline{\nabla} F(\underline{x}_{k+1})$ shows that there exists a $\underline{d}_{k+1}$ that is conjugate to $\underline{d}_j$, $j = 1, 2, \ldots, k$. On the other hand, it follows from the positive definiteness of A that directions that are conjugate to each other are linearly independent. Therefore $\underline{\nabla} F(\underline{x}_{k+1})=0$ occurs in equation (5.3) if k attains the value n, and then $\underline{x}_{k+1}$ is the optimal vector of variables.

The remarks made so far are well known, and they are the basis of some very successful algorithms for unconstrained optimization when the gradient $\underline{\nabla} F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, can be calculated. We are assuming, however, that no derivatives are available. Therefore we turn our attention to the construction of search directions from function values in ways that provide some useful conjugacy properties if F happens to be quadratic.

The following technique is highly useful. If F is a strictly convex quadratic function, and if $\underline{x}_p$ and $\underline{x}_q$ are different points of $\mathcal{R}^n$ that satisfy $\underline{d}^T \underline{\nabla} F(\underline{x}_p)=0$ and $\underline{d}^T \underline{\nabla} F(\underline{x}_q)=0$, for some nonzero vector $\underline{d}$, then the choice $\underline{d}_q = \underline{x}_q - \underline{x}_p$ provides a search direction that is conjugate to $\underline{d}$. It is true because equation (5.1) implies the identity

$$\underline{d}^T \underline{\nabla} F(\underline{x}_q) - \underline{d}^T \underline{\nabla} F(\underline{x}_p) = \underline{d}^T (A\,\underline{x}_q + \underline{g}_0) - \underline{d}^T (A\,\underline{x}_p + \underline{g}_0) = \underline{d}^T A\,(\underline{x}_q - \underline{x}_p), \quad (5.4)$$

and it allows sequences of mutually conjugate directions to be formed. Indeed, let the directions $\underline{d}^{(j)}$, $j=1,2,\ldots,\ell$, be mutually conjugate, where ℓ is an integer from $[1, n-1]$, and suppose that we require a direction $\underline{d}^{(\ell+1)}$ that is conjugate to $\underline{d}^{(j)}$, $j=1,2,\ldots,\ell$. Then we pick integers p and q that satisfy $p \geq \ell+1$ and $q \geq p+\ell+1$, and we let the sequence of points $\underline{x}_k$, $k=1,2,3,\ldots$, be generated by an algorithm that makes exact line searches on every iteration. Further, we include the search directions

$$\underline{d}_{p-\ell-1+j} = \underline{d}_{q-\ell-1+j} = \underline{d}^{(j)}, \qquad j=1,2,\ldots,\ell. \quad (5.5)$$

Then, corresponding to equation (5.3), the line searches and conjugacy provide the identities

$$\underline{d}_{p-\ell-1+j}^T \underline{\nabla} F(\underline{x}_p)=0 \quad \text{and} \quad \underline{d}_{q-\ell-1+j}^T \underline{\nabla} F(\underline{x}_q)=0, \qquad j=1,2,\ldots,\ell, \quad (5.6)$$

in the quadratic case. Therefore the remark at the beginning of this paragraph is applicable if $\underline{d}$ is any of the vectors $\underline{d}^{(j)}$, $j=1,2,\ldots,\ell$. It follows that the direction $\underline{d}^{(\ell+1)} = \underline{x}_q - \underline{x}_p$ satisfies all the required conjugacy conditions, provided that $\underline{x}_q$ is different from $\underline{x}_p$. A generalization of this method helps to achieve $\underline{x}_q \neq \underline{x}_p$. It is that the step from $\underline{x}_p$ to $\underline{x}_{p+1}$ can be arbitrary, because $q \geq p+\ell+1$ implies that $\underline{d}_p$ is not included in expression (5.5).

This construction allows the least value of a strictly convex quadratic function to be calculated using $\frac{1}{2}n(n+1)$ line searches and $n-1$ displacements, as suggested by Smith (1962). A version of his method lets $\underline{s}^{(j)}$, $j=1,2,\ldots,n$, be any linearly independent vectors in $\mathcal{R}^n$, and it forms a sequence of mutually conjugate directions $\underline{d}^{(j)}$, $j=1,2,\ldots,n$, such that, for each integer ℓ in $[1,n]$, the direction $\underline{d}^{(\ell)}$ is in the linear space spanned by $\underline{s}^{(j)}$, $j=1,2,\ldots,\ell$. Therefore $\underline{d}^{(1)} = \underline{s}^{(1)}$ and $\ell=1$

are set initially, and $\underline{x}_1$ is any given point in $\mathcal{R}^n$. Then $\underline{x}_2$ is calculated by an exact line search from $\underline{x}_1$ along the direction $\underline{d}^{(1)}$. These starting conditions are a special case of the fact that, when $\underline{d}^{(j)}$, $j=1,2,\ldots,\ell$, are known for any ℓ in $[1,n]$, then a point $\underline{x}_p$ is available that satisfies the first part of expression (5.6) for the search directions $\underline{d}_{p-\ell-1+j}$, $j=1,2,\ldots,\ell$, the value of p being $\frac{1}{2}\ell(\ell+3)$, which includes $p=2$ when $\ell=1$. For each $\ell<n$, the method generates a point $\underline{x}_q$, different from $\underline{x}_p$, that achieves the second part of expression (5.6), in order that the choice $\underline{d}^{(\ell+1)}=\underline{x}_q-\underline{x}_p$ has the required conjugacy properties. Specifically, $\underline{x}_{p+1}$ is any point of the form $\underline{x}_p+\alpha_p\,\underline{s}^{(\ell+1)}$, where α_p is nonzero, and then, for $k=p+1,p+2,\ldots,p+\ell$, the new vector of variables $\underline{x}_{k+1}$ is obtained by an exact line search from $\underline{x}_k$ in the direction $\underline{d}_k=\underline{d}^{(k-p)}$, which provides $q=p+\ell+1$. There is also an exact line search from $\underline{x}_q$ in the direction $\underline{d}^{(\ell+1)}=\underline{x}_q-\underline{x}_p$, giving the point $\underline{x}_{q+1}=\underline{x}_{p+\ell+2}$. The conjugacy conditions allow this point to be the new $\underline{x}_p$ when ℓ is increased by one, and the identity

$$\tfrac{1}{2}\,\ell\,(\ell+3)+\ell+2=\tfrac{1}{2}\,(\ell+1)\,(\ell+4) \tag{5.7}$$

shows that $p=\frac{1}{2}\ell(\ell+3)$ is preserved. The method is applied recursively until $\ell=n$ is attained. Indeed, $\underline{x}_p$ is optimal when $\ell=n$, because it is the result of n consecutive line searches along mutually conjugate directions.

This method can be applied when F is a general function, using a practical line search method that would be exact if F were quadratic. Then we call the technique of the last paragraph a 'cycle', because it is usual to employ several cycles, where the final point $\underline{x}_p$ of each cycle becomes the starting point $\underline{x}_1$ of the next cycle. An unsatisfactory feature of this algorithm, however, is that, for each integer ℓ in $[1,n]$, a cycle makes just $n+1-\ell$ line searches along $\underline{d}^{(\ell)}$, so the search direction $\underline{d}^{(n)}$ occurs only once in the $\frac{1}{2}n(n+1)$ line searches. Therefore Powell (1964) suggests the following procedure, where each cycle includes only n or $n+1$ line searches, using directions that span the full space of the variables.

A cycle requires a starting point $\underline{x}_1$ and n linearly independent directions $\underline{d}_j$, $j=1,2,\ldots,n$, say, that are not expected to have any conjugacy properties initially in the quadratic case. For $k=1,2,\ldots,n$, the point $\underline{x}_{k+1}$ is generated by a line search from $\underline{x}_k$ along $\underline{d}_k$. Then the function value $F(\underline{x}_{n+1}+\underline{d}_{n+1})$, where $\underline{d}_{n+1}=\underline{x}_{n+1}-\underline{x}_1$, is calculated for a test that is specified in the next paragraph. If the test fails, the next cycle is given the current search directions, and its starting point is the current $\underline{x}_{n+1}$. Otherwise, if the test succeeds, then $\underline{x}_{n+2}$ is generated by a line search from $\underline{x}_{n+1}$ along $\underline{d}_{n+1}$, and $\underline{x}_{n+2}$ becomes the starting point of the next cycle. Further, the search directions of the next cycle are obtained by deleting one of the first n vectors from the sequence $\underline{d}_1,\underline{d}_2,\ldots,\underline{d}_{n+1}$, without changing the order of the sequence. A reasonable way of picking the vector that is deleted is also given in the next paragraph. For the moment, however, we make the ideal assumptions that F is a strictly convex quadratic function, that all the line searches are exact, that the test that has been mentioned never fails, that the deleted vector is always $\underline{d}_1$, that the search directions $\underline{d}_j$, $j=2,3,\ldots,n+1$, of every cycle are linearly independent, and that the number of cycles is n. Then, by considering the construction of $\underline{d}^{(\ell+1)}$ in the paragraph that includes expressions (5.4)–(5.6), we deduce that, at the end of the j-th cycle, the last j of the directions $\underline{d}_1,\underline{d}_2,\ldots,\underline{d}_{n+1}$ are mutually conjugate, where j is any integer in $[1,n]$. This statement is trivial for $j=1$ and, using induction, we suppose that it is true for $j\in[1,n-1]$. In this case, corresponding to equation (5.6), the points $\underline{x}_1$ and $\underline{x}_{n+1}$ of the $(j+1)$-th cycle have the properties

$$\underline{d}_k^T \underline{\nabla} F(\underline{x}_1) = 0 \quad \text{and} \quad \underline{d}_k^T \underline{\nabla} F(\underline{x}_{n+1}) = 0, \qquad k=n,n-1,\ldots,n-j+1, \tag{5.8}$$

where $\underline{d}_k$ is now the k-th search direction of the $(j+1)$-th cycle. Thus the vector $\underline{d}_{n+1} = \underline{x}_{n+1} - \underline{x}_1$, which does not vanish because of the linear independence assumption, is conjugate to $\underline{d}_k$, $k=n,n-1,\ldots,n-j+1$, which establishes the inductive hypothesis. It follows that the last n search directions of the n-th cycle are mutually conjugate. Therefore the optimal vector of variables is calculated, the total number of line searches of all the cycles being $n(n+1)$.

Let m be the integer in $[1,n+1]$ such that the cycle of the previous paragraph removes $\underline{d}_m$ from the sequence $\underline{d}_1,\underline{d}_2,\ldots,\underline{d}_{n+1}$, in order to provide the search directions for the next cycle. Therefore $m=n+1$ occurs if and only if the test to be described fails. The purpose of the test is to make the new directions as linearly independent as possible in the quadratic case. We define the 'Euclidean measure' of linear independence of the vectors $\underline{s}_j \in \mathcal{R}^n$, $j=1,2,\ldots,n$, to be the modulus of the determinant of the matrix with the columns $\underline{s}_j/\|\underline{s}_j\|_2$, $j=1,2,\ldots,n$. Moreover, when F is the quadratic function (5.1) and when its second derivative matrix A is positive definite, we define the 'natural measure' to be the modulus of the determinant of the matrix with the columns $\underline{s}_j/(\underline{s}_j^T A\, \underline{s}_j)^{1/2}$, $j=1,2,\ldots,n$. The method of Powell (1964) chooses m in a way that maximizes the 'natural measure' of the directions that are retained. In order to specify this choice, we deduce from the identity

$$\underline{d}_{n+1} = \underline{x}_{n+1} - \underline{x}_1 = \sum_{k=1}^{n} \alpha_k\, \underline{d}_k \tag{5.9}$$

that, if $m \le n$ occurs in the quadratic case, then the natural measure of the retained directions is the natural measure of $\underline{d}_j$, $j=1,2,\ldots,n$, multiplied by the number

$$\beta_m = \alpha_m\, (\underline{d}_m^T A\, \underline{d}_m)^{1/2} / \, (\underline{d}_{n+1}^T A\, \underline{d}_{n+1})^{1/2}, \tag{5.10}$$

but we require an expression for this quantity in terms of the available function values. Now, for each integer k in $[1,n]$, the point $\underline{x}_{k+1}$ is calculated by an exact line search along $\underline{d}_k$, which provides the formula

$$F(\underline{x}_k) - F(\underline{x}_{k+1}) = \tfrac{1}{2}\, (\underline{x}_k - \underline{x}_{k+1})^T A\, (\underline{x}_k - \underline{x}_{k+1}) = \tfrac{1}{2}\, \alpha_k^2\, \underline{d}_k^T A\, \underline{d}_k \tag{5.11}$$

in the quadratic case. We also make use of the fact that the function (5.1) has the property

$$F(\underline{x}_{n+1}-\underline{d}_{n+1}) - 2\,F(\underline{x}_{n+1}) + F(\underline{x}_{n+1}+\underline{d}_{n+1}) = \underline{d}_{n+1}^T A\,\underline{d}_{n+1}. \qquad (5.12)$$

Therefore m is picked by the following procedure for all functions $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$. We let $\hat{m}$ be a value of k in $[1,n]$ that maximizes the difference $F(\underline{x}_k) - F(\underline{x}_{k+1})$, $k=1,2,\ldots,n$. The choice $m=\hat{m}$ is made if the test

$$F(\underline{x}_{\hat{m}}) - F(\underline{x}_{\hat{m}+1}) > \tfrac{1}{2}\,F(\underline{x}_1) - F(\underline{x}_{n+1}) + \tfrac{1}{2}\,F(\underline{x}_{n+1}+\underline{d}_{n+1}) \qquad (5.13)$$

is satisfied, which provides $\beta_m > 1$ in the quadratic case, but $m=n+1$ is selected otherwise. It can be deduced from the condition $F(\underline{x}_{\hat{m}}) - F(\underline{x}_{\hat{m}+1}) \le F(\underline{x}_1) - F(\underline{x}_{n+1})$ that the test (5.13) is equivalent to the one that is used in Powell (1964).

A property of the 'natural measure' makes it appropriate for the present application. The property is that, if F is the strictly convex quadratic function (5.1), then the natural measure of the directions $\underline{d}_k \in \mathcal{R}^n$, $k=1,2,\ldots,n$, is greatest if and only if the directions are mutually conjugate. In order to prove this assertion, we assume without loss of generality that the lengths of the directions satisfy $\underline{d}_k^T A\,\underline{d}_k = 1$, $k=1,2,\ldots,n$. Therefore the natural measure is $|\det D|$, where D is the $n \times n$ matrix with the columns $\underline{d}_k$, $k=1,2,\ldots,n$. Further, if the directions are mutually conjugate, then D satisfies $D^T A\,D = I$, which implies $|\det D| = (\det A)^{-1/2}$. Thus the natural measure is the same for all sets of mutually conjugate directions. It follows that the assertion is true, provided that lack of conjugacy allows the natural measure to be increased. Therefore we assume $\underline{d}_1^T A\,\underline{d}_2 \neq 0$. If $\underline{d}_1$ and $\underline{d}_2$ are overwritten by $\underline{s}_1 = \cos\theta\,\underline{d}_1 - \sin\theta\,\underline{d}_2$ and $\underline{s}_2 = \sin\theta\,\underline{d}_1 + \cos\theta\,\underline{d}_2$, respectively, for any real θ, then $\det D$ does not change, but the natural measure of the directions is multiplied by $\{(\underline{s}_1^T A\,\underline{s}_1)\,(\underline{s}_2^T A\,\underline{s}_2)\}^{-1/2}$. This factor is greater than one when $\theta=\pi/4$, say, due to the elementary relation

$$\begin{aligned}(\underline{s}_1^T A\,\underline{s}_1)\,(\underline{s}_2^T A\,\underline{s}_2) \;&=\; \tfrac{1}{4}\left\{(\underline{d}_1-\underline{d}_2)^T A\,(\underline{d}_1-\underline{d}_2)\right\}\left\{(\underline{d}_1+\underline{d}_2)^T A\,(\underline{d}_1+\underline{d}_2)\right\} \\ &=\; (1-\underline{d}_1^T A\,\underline{d}_2)\,(1+\underline{d}_1^T A\,\underline{d}_2) = 1-(\underline{d}_1^T A\,\underline{d}_2)^2 < 1, \qquad (5.14)\end{aligned}$$

which completes the proof.

On the other hand, the 'Euclidean measure' of linear independence of the directions $\underline{d}_k$, $k=1,2,\ldots,n$, is greatest if and only if the directions are mutually orthogonal. It has the advantage over the 'natural measure' that it cannot become misleading in general calculations due to pathological features of the objective function, but it is less suitable as an aid for achieving conjugacy. It is possible, however, to maximize both measures simultaneously when F is the function (5.1), by letting the search directions be mutually orthogonal eigenvectors of the symmetric matrix A. The 'Praxis' algorithm of Brent (1973) takes advantage of this remark in the following way. The natural measure of linear independence is employed throughout, and also, after about every n^2 line searches, the matrix A is defined by the equation $D^T A\, D = I$, where the columns of D are still the vectors $\underline{d}_k$, $k=1,2,\ldots,n$. Further, these vectors are normalized so that the second derivative of the line search function $F(\underline{x}_k + \alpha\, \underline{d}_k)$, $\alpha \in \mathcal{R}$, is approximately one at $\alpha=\alpha_k$ for each k. Then the current search directions are replaced by mutually orthogonal eigenvectors of A. Another interesting use of eigenvectors occurs in the algorithm of Brodlie (1975). Here the technique for achieving the conjugacy condition $\underline{d}_p^T A\, \underline{d}_q = 0$, where p and q are any integers that satisfy $1 \leq p < q \leq n$, is analogous to the annihilation of the (p,q) off-diagonal matrix element in Jacobi's method for calculating the eigenvalues of an $n \times n$ symmetric matrix. Thus the search directions can remain mutually orthogonal, but, when F is quadratic and $n \geq 3$, it is usually not possible to calculate the least value of F in a finite number of iterations.

There is some convergence theory for the algorithms of this section when $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, is a convex function that need not be quadratic. In particular, the analysis of Toint and Callier (1977) allows some freedom in the step-lengths of the line searches.

6. Linear approximation methods

The changes to the variables in the simplex methods of Section 4 depend on the positions $\underline{v}_i$, $i=1,2,\ldots,n+1$, of the vertices of the current simplex, and on an integer m in $[1, n+1]$ that is usually defined by the conditions $F(\underline{v}_m) \geq F(\underline{v}_i)$, $i=1,2,\ldots,n+1$. These methods make no other use of $F(\underline{v}_i)$, $i=1,2,\ldots,n+1$, however, when choosing the next vector of variables for the calculation of the objective function, although the function values at the vertices can provide highly useful information when F is smooth. In particular, there is a unique linear polynomial from $\mathcal{R}^n$ to $\mathcal{R}$, Φ say, that satisfies the interpolation conditions

$$\Phi(\underline{v}_i) = F(\underline{v}_i), \qquad i=1,2,\ldots,n+1, \tag{6.1}$$

and often $\underline{\nabla}\Phi$ is very helpful for reducing the least calculated value of F. Therefore we will consider changes to the variables that are derived from Φ. The given procedures also allow constraints on the variables of the form

$$c_p(\underline{x}) \geq 0, \qquad p=1,2,\ldots,m, \tag{6.2}$$

where m denotes the number of constraints from now until the end of the section. The constraint functions have to be specified by a subroutine that calculates $c_p(\underline{x})$, $p=1,2,\ldots,m$, at points $\underline{x} \in \mathcal{R}^n$ that are generated automatically. These points include the vertices of the current simplex, in order that, for each p, we can let γ_p be the linear polynomial from $\mathcal{R}^n$ to $\mathcal{R}$ whose coefficients are defined by the equations $\gamma_p(\underline{v}_i) = c_p(\underline{v}_i)$, $i = 1, 2, \ldots, n+1$. Then the inequalities (6.2) are approximated by the linear conditions

$$\gamma_p(\underline{x}) \geq 0, \qquad p=1, 2, \ldots, m. \tag{6.3}$$

Most of the techniques of this section are taken from Powell (1994).

We restrict attention to iterative algorithms, where, for $k = 1, 2, 3, \ldots$, the k-th iteration employs the current simplex and the linear approximations of the previous paragraph. The initial simplex is constructed from data that include a recommended distance between vertices, namely $\Delta_1 > 0$. For each k, we let $\underline{x}_k$ be the best of the vertices $\underline{v}_i$, $i=1, 2, \ldots, n+1$, which implies the conditions

$$\underline{x}_k \in \{\underline{v}_i : i=1, 2, \ldots, n+1\} \quad \text{and} \quad F(\underline{x}_k) \leq F(\underline{v}_i), \quad i=1, 2, \ldots, n+1, \tag{6.4}$$

in the unconstrained case. A suitable extension of the last inequality for $m \geq 1$ is given in the final paragraph of this section. Each iteration until termination generates a new vector of variables, $\underline{v}_{n+2}$ say, where the difference $\underline{v}_{n+2} - \underline{x}_k$ is either a 'minimization step' or a 'simplex step'. The values of F and any constraint functions are calculated at $\underline{v}_{n+2}$. Then the $n+1$ vertices of the simplex of the next iteration are chosen by deleting one point from the set $\{\underline{v}_i : i = 1, 2, \ldots, n+2\}$. Further, $\underline{x}_{k+1}$ is defined in the way mentioned earlier, any ties being broken by retaining $\underline{x}_{k+1} = \underline{x}_k$, unless a change provides a strict improvement according to the criterion for the best vertex. An iteration also sets the parameters Δ_{k+1} and ρ_{k+1} before increasing k, where $\rho_1 = \Delta_1$. All of these operations receive further consideration below.

The minimization of $\Phi(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, subject to the constraints (6.3), is a linear programming problem that usually fails to have a finite solution in the case $m < n$. Further, it is likely that the linear approximations are too inaccurate to be useful when $\underline{x}$ is far from the current simplex. Therefore we consider algorithms that employ trust region bounds. Specifically, the vector $\underline{v}_{n+2}$ of the k-th iteration has to satisfy the inequality

$$\|\underline{v}_{n+2} - \underline{x}_k\| \leq \rho_k, \tag{6.5}$$

where ρ_k is a positive number that is available at the beginning of the iteration, but it may be reduced occasionally. On most iterations, $\underline{v}_{n+2}$ is the vector $\underline{x}$ that minimizes $\Phi(\underline{x})$ subject to $\|\underline{x} - \underline{x}_k\| \leq \rho_k$ and the conditions (6.3), and then $\underline{v}_{n+2} - \underline{x}_k$ is the 'minimization step' of the previous paragraph, provided that $\|\underline{v}_{n+2} - \underline{x}_k\|$ is as small as possible if the solution to this subproblem is not unique. It can happen, however, that the constraints of the subproblem are inconsistent, and then the 'minimization step' is defined by minimizing the greatest violation of a linear constraint, namely $\max\{-\gamma_p(\underline{v}_{n+2}) : p=1, 2, \ldots, m\}$, subject to inequality (6.5), where again any nonuniqueness is taken up by reducing $\|\underline{v}_{n+2} - \underline{x}_k\|$. Powell (1994) addresses these calculations when the vector norm is Euclidean, and recommends a procedure that generates the path $\underline{v}_{n+2}(\alpha)$, $0 \leq \alpha \leq \rho_k$, in $\mathcal{R}^n$, where $\underline{v}_{n+2}(\alpha)$ is the $\underline{v}_{n+2}$ that would be required if ρ_k were equal to α. This path begins at the point $\underline{v}_{n+2}(0)=\underline{x}_k$, and is continuous and piecewise linear. Further, the different pieces of the path correspond to different indices of critical constraints, the q-th constraint being critical if and only if the conditions $\gamma_q(\underline{x})\leq 0$ and $\gamma_q(\underline{x})\leq\gamma_p(\underline{x})$, $p=1,2,\ldots,m$, hold. Sometimes the length $\|\underline{v}_{n+2}-\underline{x}_k\|$ of the 'minimization step' is 'too small', and then it is usual to replace $\underline{v}_{n+2}-\underline{x}_k$ by a 'simplex step'. There are also iterations that calculate only a 'simplex step'. The reasons for these alternatives are as follows.

We consider the case when there are no given constraints on the variables, when $\underline{v}_{n+2}-\underline{x}_k$ is a 'minimization step', when $\|\underline{v}_{n+2}-\underline{x}_k\|$ is large enough for $F(\underline{v}_{n+2})$ to be calculated, and when the new function value has the property

$$F(\underline{v}_{n+2}) \geq F(\underline{x}_k). \tag{6.6}$$

Then, because the definition of the minimization step implies $\Phi(\underline{v}_{n+2})<F(\underline{x}_k)$, the approximation $\Phi(\underline{v}_{n+2})\approx F(\underline{v}_{n+2})$ is inadequate. There are two main causes of the inadequacy, and it is important to distinguish between them. Firstly, $\underline{v}_{n+2}$ may be so far from $\underline{x}_k$ that very good linear approximations to F in a neighbourhood of $\underline{x}_k$ may be unsuitable at $\underline{v}_{n+2}$, due to second and higher order terms or lack of smoothness of the objective function. Secondly, although the bound (6.5) may ensure that any one of these very good approximations provides a minimization step that is successful at reducing the least calculated value of F, the interpolation conditions (6.1) may define a linear polynomial Φ that is unhelpful. This can happen if one or more of the distances $\|\underline{v}_i-\underline{x}_k\|$, $i=1,2,\ldots,n{+}1$, is much greater than ρ_k or if the current simplex is nearly degenerate. The appropriate remedy in the first case is to shorten the length of the minimization step on the next iteration by choosing $\rho_{k+1}<\rho_k$, which is a standard technique in trust region algorithms. In the second case, however, the remedy is to choose a better simplex. When $\underline{v}_{n+2}$ is calculated for the latter purpose, we call $\underline{v}_{n+2}-\underline{x}_k$ a 'simplex step', the actual choice of $\underline{v}_{n+2}$ being the subject of the following paragraph. The choice is independent of Φ, except for a plus or minus sign, and $\underline{v}_{n+2}$ becomes one of the vertices of the simplex of the next iteration. The need for such steps is clear if a given constraint on the variables is linear, and if $\underline{v}_{n+2}$ satisfies the constraint as an equation for all minimization steps. Indeed, if there were no simplex steps in this case, and if all of the vertices of the initial simplex have been removed from the current simplex by earlier iterations, then all of the current vertices $\underline{v}_i$, $i=1,2,\ldots,n{+}1$, are on the boundary of the linear constraint. Thus the equations (6.1) fail to define the coefficients of Φ, because the matrix of the equations is singular. Therefore a reason for the 'simplex steps' is to oppose any tendencies for the current simplex to become degenerate.

It has been mentioned that $\Delta_1>0$ is a prescribed parameter that controls the size of the initial simplex. Most of the later iterations employ $\Delta_k=\Delta_{k-1}$, and each Δ_k is an acceptable length for the edges of the current simplex, the length being relevant to the suitability of the linear polynomial Φ defined by the equations (6.1). Specifically, it is assumed that the nonlinearities of the objective function may damage the usefulness of the approximation $\Phi \approx F$, if any of the distances $\|\underline{v}_i - \underline{x}_k\|$, $i = 1, 2, \ldots, n+1$, is much greater than Δ_k. On the other hand, when 'minimization steps' are successful at improving the best vector of variables so far, then there is no need for any 'simplex steps'. Thus 20 consecutive iterations, say, may make changes to the variables that are minimization steps, and all of the changes may be in roughly the same direction in $\mathcal{R}^n$, which causes $\max\{\|\underline{v}_i - \underline{x}_k\| : i = 1, 2, \ldots, n+1\}$ to become large. Eventually, however, we expect the sequence of successful iterations to be interrupted by a minimization step that makes $\underline{v}_{n+2}$ no better than $\underline{x}_k$, which means that inequality (6.6) occurs in the unconstrained case. Then the next iteration employs a 'simplex step'. When the k-th iteration tries to take a simplex step, an integer ℓ in $[1, n]$ is calculated that has the property

$$\|\underline{v}_\ell - \underline{x}_k\| = \max\{\|\underline{v}_i - \underline{x}_k\| : i = 1, 2, \ldots, n+1\}. \tag{6.7}$$

Further, the condition $\|\underline{v}_\ell - \underline{x}_k\| \leq \beta\,\Delta_k$ is tested, where $\beta > 1$ is a prescribed constant that has the value $\beta = 2.1$ in the work of Powell (1994). If the test fails, then $\underline{v}_{n+2}$ is chosen in a way that makes it suitable to delete $\underline{v}_\ell$ from the set $\{\underline{v}_i : i = 1, 2, \ldots, n+2\}$, when generating the vertices of the simplex of the next iteration. Specifically, letting $\underline{\omega}_\ell \in \mathcal{R}^n$ be a vector of unit length that is orthogonal to the face of the current simplex that is without $\underline{v}_\ell$, we let $\underline{v}_{n+2}$ be the point

$$\underline{v}_{n+2} = \underline{x}_k \pm \Delta_k\,\underline{\omega}_\ell, \tag{6.8}$$

where the $\pm$ sign is negative if and only if $\underline{x}_k - \Delta_k\,\underline{\omega}_\ell$ is better than $\underline{x}_k + \Delta_k\,\underline{\omega}_\ell$, according to a criterion in the last paragraph of this section. Otherwise, if $\|\underline{v}_\ell - \underline{x}_k\| \leq \beta\,\Delta_k$ is achieved, the algorithm seeks a different integer ℓ in $[1, n+1]$. Indeed, letting σ_i be the distance from $\underline{v}_i$ to the plane in $\mathcal{R}^n$ that contains the vertices of the current simplex that are different from $\underline{v}_i$, the new ℓ minimizes σ_ℓ subject to $\underline{v}_\ell \neq \underline{x}_k$. Therefore a very small value of σ_ℓ indicates that the simplex is nearly degenerate. The inequality $\sigma_\ell \geq \alpha\,\Delta_k$ is tried, where $\alpha < 1$ is another positive constant, for instance $\alpha = 1/4$. If the inequality fails, then $\underline{v}_{n+2}$ is defined by formula (6.8) for the new ℓ, where the $\pm$ sign and $\underline{\omega}_\ell$ are as before. Further, the new simplex is generated by replacing $\underline{v}_\ell$ by $\underline{v}_{n+2}$ in the list of vertices, which increases the volume of the current simplex by the factor Δ_k / σ_ℓ. If $\sigma_\ell \geq \alpha\,\Delta_k$ holds, however, then the positions of the vertices $\underline{v}_i$, $i = 1, 2, \ldots, n+1$, are assumed to be adequate for the equations (6.1) that define Φ, and we say that the simplex is 'acceptable'. Then the iteration tries to generate $\underline{v}_{n+2}$ by a 'minimization step' instead of by a 'simplex step'.

We are now ready to consider the choices between the minimization and simplex step alternatives, the values of Δ_k and ρ_k, $k = 1, 2, 3, \ldots$, and a condition for terminating the calculation. Simple rules are recommended for adjusting Δ_k and for termination. Specifically, Δ_1 is given, and, until termination, the k-th iteration sets $\Delta_{k+1} = \Delta_k$, where k is still the iteration number. The value of Δ_k at the start of the k-th iteration is provisional, however, in order that a few iterations can reduce Δ_k, although no increases are allowed. A positive parameter, Δ_* say, has to be prescribed that satisfies $\Delta_* \le \Delta_1$, because it is a lower bound on every Δ_k. The changes in Δ_k have to be such that $\Delta_k = \Delta_*$ occurs after a finite number of reductions. The situation that causes a reduction is described in the paragraph after next. The calculation terminates when this situation occurs and Δ_k has already reached the value Δ_*. These rules afford the following useful properties. Every iteration until termination picks a vector of variables $\underline{v}_{n+2}$ that satisfies the inequality

$$\|\underline{v}_{n+2} - \underline{x}_k\| \ge \Delta_k, \tag{6.9}$$

and Δ_k is not reduced until this condition seems to prevent further improvements to the variables. The user can pick a value of Δ_1 that causes substantial adjustments to the variables to be tried at the beginning of the calculation, which can alleviate the damage from any random noise in the function values. Then the bound (6.9) can be refined gradually by the decreases in Δ_k. Further, when the given functions are smooth, good accuracy can usually be achieved at termination by letting Δ_* be sufficiently small.

Powell (1994) sets $\rho_k = \Delta_k$ throughout the calculation, but changes to the variables that are much greater than Δ_k are sometimes necessary for efficiency. Indeed, there are unconstrained calculations with quadratic objective functions such that, when Δ_k is reduced, the distance from $\underline{x}_k$ to the optimal vector of variables is of magnitude $M\Delta_k$, where M is the condition number of the second derivative matrix $\nabla^2 F$. Therefore it may be helpful to allow ρ_k to be much larger than Δ_k. Moreover, the initial choice $\rho_1 = \Delta_1$ has been mentioned already, and it is reasonable to set ρ_k to the new value of Δ_k when Δ_k is decreased, because ρ_k should become less than the old value of Δ_k for the moment, but condition (6.9) excludes $\rho_k < \Delta_k$. These remarks suggest the following guidelines for the choice of ρ_k, $k = 1, 2, 3, \ldots$. We pick $\rho_k = \Delta_k$ for each new Δ_k, which causes ρ_k to be less than its value at the beginning of any iteration that decreases Δ_k, but there are no other changes to ρ_k during an iteration. The value $\rho_{k+1} = \rho_k$ is often set at the end of the k-th iteration, and it always occurs when $\underline{v}_{n+2} - \underline{x}_k$ is a 'simplex step'. On the other hand, $\rho_{k+1} > \rho_k$ is allowed when $\underline{v}_{n+2} - \underline{x}_k$ is a 'minimization step' that provides $\underline{x}_{k+1} \ne \underline{x}_k$. If a minimization step fails to improve the best vector of variables so far, however, then the next minimization step is required to be substantially shorter than the present one, except that the bound (6.9) is preserved. Therefore the value

$$\rho_{k+1} = \max\,[\,\Delta_k,\ \tfrac{1}{2}\,\|\underline{v}_{n+2} - \underline{x}_k\|\,], \tag{6.10}$$

for example, may be suitable. Some choices of ρ_k after successful minimization steps can be found in Chapter 5 of Fletcher (1987), for instance, and in other published descriptions of trust region methods.

Each iteration until termination has to choose a 'minimization step' or a 'simplex step'. The method that fixes the choice is specified below, using the nomenclature that a minimization step is 'long enough' if it satisfies inequality (6.9), and is 'questionable' unless its length is exactly Δ_k and the current simplex is 'acceptable'. When the iteration does not reduce Δ_k, then the choice between the alternatives is determined by the following four rules, which are given in order of priority. (1) A minimization step is preferred if it is long enough and if either $k=1$ or the previous iteration improved the best vector of variables so far. (2) A minimization step is preferred if it is long enough and if the previous iteration applied a simplex step. (3) A simplex step is preferred if neither (1) nor (2) apply and if the current simplex is not acceptable. (4) A minimization step is preferred if it is long enough, if the current simplex is acceptable and if the previous iteration employed a minimization step that is questionable. Thus the remaining possibilities are the following two situations. (5) The current simplex is acceptable and the minimization step is not long enough. (6) The current simplex is acceptable, the minimization step is long enough, the previous iteration applied a minimization step that is not questionable, but that iteration did not improve the best vector of variables. In these cases the time has come to reduce Δ_k and ρ_k. Therefore termination occurs if Δ_k has attained the value Δ_*. Otherwise, after reducing Δ_k and ρ_k, the required choice is determined by three more rules. (7) The minimization step is preferred if it is long enough for the new Δ_k. (8) The simplex step is chosen if (7) fails and if the current simplex is not acceptable for the new Δ_k. (9) In all other cases, Δ_k is still too large, so we introduce a recursion by branching back to the part of the algorithm that either causes termination or reduces Δ_k. Thus each iteration before termination picks just one vector $\underline{v}_{n+2}$ at which the values of the given functions from $\mathcal{R}^n$ to $\mathcal{R}$ are calculated.

Another question that requires an answer is the choice of the $n+1$ vertices of the simplex for the next iteration from $\{\underline{v}_i : i=1,2,\ldots,n+2\}$. We let $\underline{v}_\ell$ be the point that is not retained, which agrees with equation (6.8) when $\underline{v}_{n+2}-\underline{x}_k$ is a 'simplex step'. We propose a new choice of ℓ when $\underline{v}_{n+2}-\underline{x}_k$ is a 'minimization step', however, because the technique in Powell (1994) assumes $\rho_k=\Delta_k$ for every k. Let $\underline{x}_{k+1}\in\{\underline{v}_i : i=1,2,\ldots,n+2\}$ be determined before ℓ is selected, which is possible because we require the best vector of variables so far. Further, let the real multipliers θ_i, $i=1,2,\ldots,n+2$, satisfy the equation

$$\sum_{i=1}^{n+2} \theta_i\,(\underline{v}_i-\underline{x}_{k+1}) = 0, \tag{6.11}$$

where θ_{i_*} is zero for the integer i_* that is defined by $\underline{x}_{k+1}=\underline{v}_{i_*}$, but some of the other multipliers are nonzero. It follows from the nondegeneracy of the current simplex that the values of the multipliers are determined uniquely except for a scaling factor. Now, if i and j are different integers in $[1,n{+}2]$ such that θ_i and θ_j are nonzero, and if $\mathcal{S}_i$ and $\mathcal{S}_j$ are the new simplices for $\ell{=}i$ and $\ell{=}j$, respectively, then equation (6.11) implies the property

$$|\,\mathrm{Vol}\,\mathcal{S}_i\,/\,\mathrm{Vol}\,\mathcal{S}_j\,| = |\,\theta_i\,/\,\theta_j\,|\,. \tag{6.12}$$

Therefore it may be suitable to pick ℓ by satisfying the condition $|\theta_\ell|{=}\max\{|\theta_i| : i{=}1,2,\ldots,n{+}2\}$. This method, however, would favour the retention of any points $\underline{v}_i$ that are far from $\underline{x}_{k+1}$, and we do not want the new simplex to have a large volume because the lengths of some of its sides are much greater than Δ_k. Instead we take the view for the moment that, for every i in $[1,n{+}2]$ such that $\|\underline{v}_i{-}\underline{x}_{k+1}\|$ exceeds Δ_k, the point $\underline{v}_i$ is replaced by the point on the line segment from $\underline{x}_{k+1}$ to $\underline{v}_i$ that is distance Δ_k from $\underline{x}_{k+1}$, but $\underline{v}_i$ is unchanged for the other values of i. Then we choose ℓ by applying the procedure just described to these new points. Specifically, for each integer i in $[1,n{+}2]$, we find that θ_i in equation (6.11) has to be scaled by $\max[1,\|\underline{v}_i-\underline{x}_{k+1}\|/\Delta_k]$, because of the temporary change to $\underline{v}_i$. Therefore we let ℓ be an integer in $[1,n{+}2]$ that has the property

$$|\theta_\ell|\;\max\left[\Delta_k,\|\underline{v}_\ell{-}\underline{x}_{k+1}\|\right]\;\geq\;|\theta_i|\;\max\left[\Delta_k,\|\underline{v}_i{-}\underline{x}_{k+1}\|\right], \quad i{=}1,2,\ldots,n{+}2. \tag{6.13}$$

Thus, if the current simplex has a vertex that is far from the best vector of variables, there is a tendency to exclude it from the simplex of the next iteration.

The merit function, Ψ say, of the calculation provides a balance between the value of the objective function and any constraint violations, in order to determine the best vertex of the current simplex. Specifically, Ψ is the same as F when there are no constraints, and, for $m{\geq}1$, the form

$$\Psi(\underline{x}) = F(\underline{x}) + \mu\,[\,\max\{-c_p(\underline{x}) : p{=}1,2,\ldots,m\}]_+, \qquad \underline{x}{\in}\mathcal{R}^n, \tag{6.14}$$

is taken from Powell (1994). Here μ is a parameter that is zero initially and that may be increased automatically as described below. Further, the subscript '+' indicates that the expression in square brackets is replaced by zero if and only if its value is negative. Thus $\Psi(\underline{x}) = F(\underline{x})$ occurs whenever $\underline{x}$ is feasible, and it is helpful to scale the constraint functions so that the values $c_p(\underline{x})$, $p{=}1,2,\ldots,m$, have similar magnitudes for typical vectors $\underline{x}$. Expression (6.4) is extended to $m{\geq}0$ by requiring the best vertex $\underline{x}_k$ to satisfy the conditions

$$\underline{x}_k \in \{\underline{v}_i : i{=}1,2,\ldots,n{+}1\} \quad \text{and} \quad \Psi(\underline{x}_k){\leq}\Psi(\underline{v}_i), \quad i{=}1,2,\ldots,n{+}1. \tag{6.15}$$

After choosing $\underline{x}_k$, both the minimization and the simplex steps are independent of Ψ and μ, but Ψ is usually important to what happens next. Indeed, if the new vector of variables of the k-th iteration, namely $\underline{v}_{n+2}$, is generated by a minimization step, then usually another minimization step is chosen by the $(k{+}1)$-th iteration if and only if the strict inequality $\Psi(\underline{v}_{n+2}){<}\Psi(\underline{x}_k)$ holds. Further, this inequality should be achieved if all the linear approximations of the first paragraph of this section are exact. Therefore we require the value of μ to provide the property

$$\Upsilon(\underline{v}_{n+2}) < \Upsilon(\underline{x}_k), \qquad \text{if } \underline{v}_{n+2}-\underline{x}_k \text{ is a minimization step,} \tag{6.16}$$

where Υ is the piecewise linear approximation

$$\Upsilon(\underline{x}) = \Phi(\underline{x}) + \mu\,[\,\max\{-\gamma_p(\underline{x}) : p=1,2,\ldots,m\}\,]_+, \qquad \underline{x}\in\mathcal{R}^n, \tag{6.17}$$

to the merit function (6.14). Now a minimization step either reduces the contribution from the constraints to expression (6.17), or the contribution is zero and $\Phi(\underline{v}_{n+2})<\Phi(\underline{x}_k)$ occurs, where we are excluding steps that are zero, because they are abandoned automatically, due to the failure of inequality (6.9). It follows that condition (6.16) can be achieved whenever it is required by choosing a sufficiently large value of μ. Therefore Powell (1994) proposes the following technique for increasing μ. Whenever a minimization step is calculated that has the property (6.9), we let $\bar{\mu}$ be the least nonnegative value of μ that provides $\Upsilon(\underline{v}_{n+2})\leq\Upsilon(\underline{x}_k)$. Further, μ is unchanged in the case $\mu\geq\frac{3}{2}\bar{\mu}$, but otherwise it is increased to $2\bar{\mu}$. A possible consequence of an increase in μ is that $\underline{x}_k$ is no longer the optimal vertex, and then the calculated minimization step would be incorrect. Therefore $\underline{x}_k$ is changed if necessary to another vertex that satisfies the conditions (6.15). Then the minimization step is recalculated, so μ may have to be increased again, which may cause a further change to the optimal vertex. Fortunately, this procedure does not cycle, because each change to $\underline{x}_k$ causes a strict reduction in $\max\{-\gamma_p(\underline{x}_k) : p=1,2,\ldots,m\}$. Another use of Υ is that the $\pm$ sign of expression (6.8) is negative if and only if $\Upsilon(\underline{x}_k-\Delta_k\,\underline{\omega}_\ell)$ is less than $\Upsilon(\underline{x}_k+\Delta_k\,\underline{\omega}_\ell)$. Some remarks on the convergence properties of the algorithm of this section are made in Section 9.

7. Quadratic approximation methods

Many of the techniques of this section are similar to those of Section 6, but now we let the approximation $\Phi(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, to the objective function $F(\underline{x})$, $\underline{x}\in\mathcal{R}^n$, be a quadratic polynomial instead of a linear polynomial. Therefore Φ has $\frac{1}{2}(n+1)\,(n+2)=\hat{n}$, say, independent coefficients, that may be defined by the interpolation conditions

$$\Phi(\underline{v}_i) = F(\underline{v}_i), \qquad i=1,2,\ldots,\hat{n}, \tag{7.1}$$

where the vectors $\underline{v}_i$, $i=1,2,\ldots,\hat{n}$, are points in $\mathcal{R}^n$. These points should have the property that, if expression (7.1) is written as a system of linear equations, the unknowns being the coefficients, then the matrix of the system is nonsingular. The

Lagrange functions of the interpolation problem will be useful later. Therefore we reserve the notation χ_i, $i=1,2,\ldots,\hat{n}$, for the quadratic polynomials from $\mathcal{R}^n$ to $\mathcal{R}$ that satisfy the equations

$$\chi_i(\underline{v}_j) = \delta_{ij}, \qquad 1 \le i, j \le \hat{n}, \tag{7.2}$$

where δ_{ij} is the Kronecker delta. It follows that Φ is the function

$$\Phi(\underline{x}) = \sum_{i=1}^{n} F(\underline{v}_i)\,\chi_i(\underline{x}), \qquad \underline{x} \in \mathcal{R}^n. \tag{7.3}$$

The main advantage of quadratic over linear polynomials is that quadratics include some second derivative information, which allows the development of algorithms that have useful superlinear convergence properties. We are going to consider some of the ideas that have been proposed for constructing and applying quadratic approximations to F when there are no constraints on the variables.

The algorithm of Winfield (1973) not only employs the interpolation equations (7.1) to define Φ, but also it includes some of the earliest work on trust regions. The k-th iteration of that method is given all the values of the objective function calculated so far, $\hat{n}$ of them being obtained before the first iteration. Let these values be $F(\underline{v}_i)$, $i=1,2,\ldots,\check{n}$, where $\check{n} \ge \hat{n}$, let $\underline{x}_k$ be a best vector of variables, which means that it satisfies the conditions

$$\underline{x}_k \in \{\underline{v}_i : i=1,2,\ldots,\check{n}\} \quad \text{and} \quad F(\underline{x}_k) \le F(\underline{v}_i), \quad i=1,2,\ldots,\check{n}, \tag{7.4}$$

and let the current data be ordered so that the sequence of distances $\|\underline{v}_i - \underline{x}_k\|$, $i=1,2,\ldots,\check{n}$, increases monotonically. Then the k-th iteration generates the quadratic polynomial Φ by trying to interpolate the function values of only the first $\hat{n}$ terms of the sequence, in accordance with the notation (7.1). Further, the iteration calculates the vector $\underline{x} \in \mathcal{R}^n$ that minimizes $\Phi(\underline{x})$ subject to the bound $\|\underline{x} - \underline{x}_k\| \le \rho_k$, where the trust region radius is chosen automatically and satisfies $\rho_k \le 0.99\,\|\underline{v}_{\hat{n}} - \underline{x}_k\|$, in order that the value of F at the new point will be included in the interpolation conditions of the $(k+1)$-th iteration. One reason for mentioning the algorithm is that it acts in an enterprising way when the system (7.1) is degenerate. Specifically, the degeneracy is ignored, it is assumed that the calculation of Φ is sufficiently robust to provide a quadratic function that allows the trust region subproblem to be solved, and the resultant $\underline{x}$ receives no special treatment. Thus some unpredictable changes to the variables occur that may remove the degeneracy after a few iterations. Indeed, Winfield (1973) states that "This natural cure of ill-conditioning is more efficient than restarting the algorithm by evaluating $F(\underline{x})$ at the points of a grid". The other methods that we study, however, ensure that each Φ is well defined.

The Lagrange functions that have been mentioned provide a convenient way of avoiding singularity in the equations (7.1). The technique suggests itself if one tries to modify the algorithm of Powell (1994) for unconstrained optimization, described in the previous section, so that the linear polynomial Φ of expression (6.1) is replaced by the quadratic polynomial that is defined by the equations (7.1). We retain from Section 6 the parameters Δ_k and ρ_k, $k=1,2,3,\ldots$, and the rules that give their values. Moreover, in the quadratic case, the points $\underline{v}_i$, $i=1,2,\ldots,\hat{n}$, for the first iteration can be the vertices and the mid-points of the edges of a nondegenerate simplex in $\mathcal{R}^n$, where the lengths of the edges are still of magnitude Δ_1. Otherwise, for $k\geq 2$, these points are chosen by the previous iteration, and $\underline{x}_k$ satisfies the conditions

$$\underline{x}_k \in \{\underline{v}_i : i=1,2,\ldots,\hat{n}\} \quad \text{and} \quad F(\underline{x}_k) \leq F(\underline{v}_i), \quad i=1,2,\ldots,\hat{n}. \tag{7.5}$$

Further, $\underline{v}_{\hat{n}+1}-\underline{x}_k$ is still a 'minimization step' if $\underline{v}_{\hat{n}+1}$ is the vector $\underline{x}\in\mathcal{R}^n$ that minimizes $\Phi(\underline{x})$ subject to $\|\underline{x}-\underline{x}_k\|\leq\rho_k$, which is the trust region subproblem of the previous paragraph. On the other hand, a 'simplex step' is usually required if the previous iteration generated a minimization step that failed to reduce the least calculated value of F. In this case, guided by equation (6.7), we let ℓ be an integer in $[1,\hat{n}]$ that maximizes $\|\underline{v}_\ell-\underline{x}_k\|$. If this distance is unacceptably large, then we have to pick a point $\underline{v}_{\hat{n}+1}$ that will replace $\underline{v}_\ell$ in the system (7.1) on the next iteration. Therefore we require a formula that is analogous to expression (6.8), and that is suitable when Φ is a quadratic polynomial.

Now the main property of the point (6.8) is that it maximizes the volume of the simplex of the next iteration subject to $\|\underline{v}_{n+2}-\underline{x}_k\|\leq\Delta_k$. Further, the volume of the simplex is a constant multiple of the modulus of the determinant of the matrix of the system (6.1), when the usual basis of the space of linear polynomials is employed. Therefore an analogous choice of the 'simplex step' when Φ is quadratic would maximize the modulus of the determinant of the $\hat{n}\times\hat{n}$ system (7.1), after $\underline{v}_\ell$ is replaced by $\underline{v}_{\hat{n}+1}$, where $\underline{v}_{\hat{n}+1}$ has to satisfy $\|\underline{v}_{\hat{n}+1}-\underline{x}_k\|\leq\Delta_k$. We write $\underline{x}=\underline{v}_{\hat{n}+1}$ for the moment, we regard the new determinant as a function of $\underline{x}\in\mathcal{R}^n$, and we find that it is a quadratic polynomial in $\underline{x}$. Further, the determinant must vanish if $\underline{x}$ is any point of the set $\{\underline{v}_i : i=1,2,\ldots,\hat{n}\}$ that is different from $\underline{v}_\ell$. Thus an elementary normalization provides the identity

$$\text{New determinant / Old determinant} \;=\; \chi_\ell(\underline{x}), \qquad \underline{x}\in\mathcal{R}^n. \tag{7.6}$$

Therefore we define $\underline{v}_{\hat{n}+1}-\underline{x}_k$ to be a 'simplex step' for the chosen integer $\ell\in[1,\hat{n}]$ if and only if $\underline{v}_{\hat{n}+1}$ is a vector of variables $\underline{x}$ that maximizes $|\chi_\ell(\underline{x})|$ subject to $\|\underline{x}-\underline{x}_k\|\leq\Delta_k$. This definition has the advantage of being independent of the choice of basis of the space of quadratic polynomials. Further, the simplex step can be calculated by solving two trust region subproblems of the type that has been encountered already. Indeed, if two vectors of variables are generated by minimizing the quadratic functions χ_ℓ and $-\chi_\ell$ subject to the trust region bound, then the required $\underline{v}_{\hat{n}+1}$ is the vector that gives the larger value of $|\chi_\ell|$.

When the k-th iteration tries to take a 'simplex step', the algorithm may find that all of the points $\underline{v}_i$, $i=1,2,\ldots,\hat{n}$, are sufficiently close to $\underline{x}_k$, which corresponds to the condition $\|\underline{v}_\ell - \underline{x}_k\| \leq \beta\,\Delta_k$ in Section 6. Then a test for near-degeneracy of the system (7.1) is required, that is analogous to the use of σ_ℓ and α in the paragraph that includes equations (6.7) and (6.8). There the replacement of $\underline{v}_\ell$ by the point (6.8) increases the modulus of the determinant of the system (6.1) by the factor Δ_k/σ_ℓ. Therefore we continue to let $\alpha < 1$ be a positive constant, for instance $\alpha = 1/4$, and we seek an integer ℓ in $[1,\hat{n}]$ such that the replacement of $\underline{v}_\ell$ by $\underline{v}_{\hat{n}+1}$ increases the modulus of the determinant of the system (7.1) by a factor of more than $1/\alpha$, where $\underline{v}_{\hat{n}+1}$ is defined at the end of the previous paragraph, because this choice maximizes the modulus of the new determinant. Specifically, the test for near-degeneracy in the quadratic case is as follows. The integer ℓ runs through the set $\{1,2,\ldots,\hat{n}\}$, but similar tests on recent iterations may make it advantageous not to begin with $\ell = 1$. For each ℓ, the maximum value of $|\chi_\ell(\underline{x})|$, $\|\underline{x} - \underline{x}_k\| \leq \Delta_k$, is calculated. If $|\chi_\ell(\underline{x})| > 1/\alpha$ occurs, the task of searching for a suitable ℓ is complete, because the replacement of $\underline{v}_\ell$ by the vector $\underline{v}_{\hat{n}+1}$ that has been mentioned provides a substantial improvement to the positions of the interpolation points. Then $\underline{v}_{\hat{n}+1} - \underline{x}_k$ is a 'simplex step', and the function value $F(\underline{v}_{\hat{n}+1})$ is required for the system (7.1) of the next iteration. Otherwise, if no integer ℓ in $[1,\hat{n}]$ provides $|\chi_\ell(\underline{v}_{\hat{n}+1})| > 1/\alpha$, then, as before, we say that the current interpolation points are 'acceptable', and the iteration may generate $\underline{v}_{\hat{n}+1}$ by a 'minimization step'. We also retain the rule that the minimization step is abandoned if it fails to satisfy $\|\underline{v}_{\hat{n}+1} - \underline{x}_k\| \geq \Delta_k$, which is important to the criteria for reducing Δ_k and for termination, as described in Section 6.

We let each choice between a 'minimization' and a 'simplex' step in the quadratic case be the same as in Section 6, the rules in the paragraph after equation (6.10) being applied as before. A modification is needed, however, to the technique that selects the interpolation points for the $(k+1)$-th iteration, after $F(\underline{v}_{\hat{n}+1})$ has been calculated and $\underline{v}_{\hat{n}+1} - \underline{x}_k$ is a minimization step. These points are all but one of the vectors $\underline{v}_i$, $i=1,2,\ldots,\hat{n}+1$, and again we let $\underline{v}_\ell$ denote the point that is rejected. Here it is important to note that, in contrast to the previous two paragraphs, $\underline{v}_{\hat{n}+1}$ is now independent of ℓ, because it is generated by the minimization step before ℓ is chosen. In order to retain a best vector of variables so far, we let i_* be an integer in $[1,\hat{n}+1]$ such that $F(\underline{v}_{i_*})$ is the least of the function values $F(\underline{v}_i)$, $i=1,2,\ldots,\hat{n}+1$. Then the value $\ell = i_*$ is prohibited, because $\underline{x}_{k+1}$ is going to be the point $\underline{v}_{i_*}$. It would be straightforward to pick the ℓ that maximizes the modulus of the determinant of the system (7.1) on the next iteration if we wished to do so. Indeed, if $\ell \in [1,\hat{n}]$, then it follows from the identity (7.6) that the determinant of the new system is the determinant of the present one multiplied by $\chi_\ell(\underline{v}_{\hat{n}+1})$. Therefore, after defining $\theta_{\hat{n}+1} = 1$ and $\theta_i = \chi_i(\underline{v}_{\hat{n}+1})$, $i=1,2,\ldots,\hat{n}$, and then replacing θ_{i_*} by zero, we would let ℓ satisfy the equation $|\theta_\ell| = \max\{|\theta_i| : i=1,2,\ldots,\hat{n}+1\}$. This notation provides an analogy with the statement made immediately after expression (6.12). Again, however, we prefer to take the distances $\|\underline{v}_i-\underline{x}_{k+1}\|$, $i=1,2,\ldots,\hat{n}$, into account, so our choice of ℓ is derived from a condition that is analogous to inequality (6.13).

Specifically, if i is any integer in $[1,\hat{n}]$ such that $\|\underline{v}_i-\underline{x}_{k+1}\|>\Delta_k$ occurs, we make a notional shift of $\underline{v}_i$ to $\underline{\hat{v}}_i$, say, which is a point on the line segment from $\underline{x}_{k+1}$ to $\underline{v}_i$ that is within distance Δ_k of $\underline{x}_{k+1}$. Further, we let $\hat{\chi}_i$ be the quadratic polynomial that satisfies the Lagrange conditions $\hat{\chi}_i(\underline{\hat{v}}_i)=1$ and $\hat{\chi}_i(\underline{v}_j)=0$, for every integer j in $[1,\hat{n}]$ that is different from i. Hence $\hat{\chi}_i$ is the function

$$\hat{\chi}_i(\underline{x}) = \chi_i(\underline{x}) \,/\, \chi_i(\underline{\hat{v}}_i), \qquad \underline{x}\in\mathcal{R}^n. \tag{7.7}$$

Now, because of the inequality $\|\underline{v}_i-\underline{x}_{k+1}\|>\Delta_k$, we assume that the temporary replacement of $\underline{v}_i$ by $\underline{\hat{v}}_i$ would make the determinant of the system (7.1) more relevant to our consideration of possible near-degeneracy. Therefore we change the value of θ_i, given in the previous paragraph, to the number $\hat{\chi}_i(\underline{v}_{\hat{n}+1})=\chi_i(\underline{v}_{\hat{n}+1})/\chi_i(\underline{\hat{v}}_i)$, but there is still some freedom in the position of $\underline{\hat{v}}_i$. We have to avoid positions that are too close to other interpolation points, and it is easy to make $|\chi_i(\underline{\hat{v}}_i)|$ as large as possible, because χ_i is a quadratic function of one variable on the line segment from $\underline{x}_{k+1}$ to $\underline{v}_i$. On the other hand, it would be unsuitable to allow $|\chi_i(\underline{\hat{v}}_i)|$ to exceed one, because then $\|\underline{v}_i-\underline{x}_{k+1}\|>\Delta_k$ would assist the retention of $\underline{v}_i$ in the set of interpolation points. These remarks lead to the formula

$$\theta_i = \chi_i(\underline{v}_{\hat{n}+1}) \,/\, \min\left[1, \max\{|\chi_i(\underline{x}_{k+1}+\alpha\,[\underline{v}_i-\underline{x}_{k+1}])| : 0\le\alpha\le\bar{\alpha}\}\right], \tag{7.8}$$

where $\bar{\alpha}=\Delta_k/\|\underline{v}_i-\underline{x}_{k+1}\|$. Further, this choice is just $\theta_i=\chi_i(\underline{v}_{\hat{n}+1})$, as before, when i is any integer in $[1,\hat{n}]$ that satisfies $i\neq i_*$ and $\|\underline{v}_i-\underline{x}_{k+1}\|\le\Delta_k$. Moreover, $\theta_{\hat{n}+1}=1$ is the most reasonable scaling factor to apply to the determinant when there is no change to the interpolation points. Therefore we recommend these values of θ_i, and, after replacing θ_{i_*} by zero, we let ℓ be an integer in $[1,\hat{n}{+}1]$ that maximizes $|\theta_\ell|$.

We have found that, due to the identity (7.6), Lagrange functions are highly useful for selecting points $\underline{v}_i$, $i=1,2,\ldots,\hat{n}$, such that the quadratic polynomial Φ is well defined by the equations (7.1). I presented a paper on this technique at the '5th Stockholm Optimization Days' in 1994, but I did not write a report on it then, because I had hoped that the technique would be developed further by a research student. Later I encouraged another research student, namely Evan Jones, to study the subject, and he proposed the two trust region radii, namely Δ_k and ρ_k, that are introduced in Section 6, although there is only one trust region radius in the algorithm of Powell (1994). He investigated numerically whether or not the number of iterations is reduced by including a second trust region radius, and usually some improvements occur. In any case, the idea is attractive, because it allows some large changes to the variables to co-exist with the security of never increasing Δ_k as k advances. Therefore I described the method, and showed some preliminary numerical results, at the '5th SIAM Conference on Optimization' in 1996. I had expected Jones to write a paper on his work, but unfortunately that has not happened either. Therefore one of the reasons for the present contribution to *Acta Numerica* is to catch up on the recording of this work.

Another description of the use of Lagrange functions is given in Section 4 of Conn, Scheinberg and Toint (1997b), which includes a generous acknowledgement to the conference talks mentioned in the previous paragraph. That paper also addresses the idea of employing 'Newton fundamental polynomials' instead of Lagrange functions, where these polynomials in the quadratic case are a constant Lagrange polynomial, n linear Lagrange polynomials, and $\frac{1}{2}n(n+1)$ quadratic Lagrange polynomials that are derived from one, $n+1$ and all of the interpolation points $\underline{v}_i$, $i=1,2,\ldots,\hat{n}$, respectively. They provide a different basis of the $\hat{n}=\frac{1}{2}(n+1)\,(n+2)$ dimensional space of quadratic polynomials, which is helpful when fewer than $\hat{n}$ values of F are available to determine Φ. An outline of a trust region algorithm for unconstrained minimization without derivatives is given too. A major departure from the work of this section is that, if the k-th iteration takes a 'minimization step' that reduces F by an amount that compares favourably with the corresponding reduction in Φ, then Δ_{k+1} is allowed to be larger than Δ_k. Nevertheless, this trust region radius is reduced only when the positions of the interpolation points satisfy acceptability conditions that are similar to the ones specified in the complete paragraph that follows equation (7.6). Therefore, in comparison with the technique of Jones that employs both Δ_k and ρ_k, several extra function values may occur if the larger trust region radius is successful for only a small number of iterations. An earlier paper by Conn, Scheinberg and Toint (1997a) also considers Newton fundamental polynomials and presents an outline of a similar trust region algorithm. Further, the convergence of the algorithm is studied under certain assumptions, including the uniform boundedness of the second derivative matrices $\nabla^2\Phi$. It is proved that, if the number of iterations is infinite, then the property $\liminf_{k\to\infty}\|\underline{\nabla}F(\underline{x}_k)\|=0$ is achieved.

The last topic of this section is the algorithm of Elster and Neumaier (1995), which is designed for optimization calculations subject to the simple bounds (3.1) on the variables. The algorithm is remarkable, because it combines quadratic approximations to F and trust regions with some of the properties of discrete grids that are considered in Section 3. Thus termination is achieved, even if the values of the objective function are distorted by noise. There is a close analogy with the two trust region idea of Evan Jones, because it is appropriate to let ρ_k be the trust region radius and Δ_k be the grid size. The algorithm retains all the calculated values of F. Then each quadratic approximation Φ is formed by least squares fitting to some of them, using a technique that is interesting, because it begins by generating a Hessian approximation G, and then it restricts attention to only about $2n+2$ function values, in order to fit the parameters $a\in\mathcal{R}$, $\underline{g}\in\mathcal{R}^n$

and $\kappa \in \mathcal{R}$ of the approximation

$$\Phi(\underline{x}) = a + \underline{g}^T(\underline{x}-\underline{x}_k) + \tfrac{1}{2}\kappa\,(\underline{x}-\underline{x}_k)^T G\,(\underline{x}-\underline{x}_k), \qquad \underline{x} \in \mathcal{R}^n, \tag{7.9}$$

where $\underline{x}_k$ is still the vector of variables that provides the least value of F so far. The algorithm requires Φ, ρ_k and $\underline{x}_k$ for the calculation of a 'minimization step'. Then the new vector of variables at the end of this step is shifted to the nearest grid point, $\underline{x}_+$ say. The use of grids ensures that, after only a finite number of iterations, the function value $F(\underline{x}_+)$ will have been found by an earlier iteration. When this happens, or when three consecutive minimization steps fail to achieve $F(\underline{x}_+) < F(\underline{x}_k)$, a procedure is invoked that is similar to a 'simplex step'. The procedure derives and may apply a linear polynomial approximation to F, using values of the objective function at grid points that have to be neighbours of $\underline{x}_k$. Thus the decision is taken whether or not to reduce Δ_k before resuming the minimization steps. Alternatively, as in Section 6, termination occurs if a reduction in Δ_k is required but Δ_k is already at a prescribed lower bound. Several numerical experiments in Elster and Neumaier (1995) show that this algorithm compares favourably with the method of Nelder and Mead (1965), and with a finite difference implementation of a quasi-Newton algorithm.

8. Simulated annealing

The algorithms that we have studied so far are designed to converge to a local minimum of the objective function $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, subject to any constraints on the variables. Many practical optimization calculations, however, have several local minima that are not optimal. Therefore some methods that select vectors of variables using random number generators have become very popular for a wide range of applications. Usually they possess the property that, if there are no constraints, if the objective function is continuous and has bounded level sets, if its least value is $F(\underline{x}^*)$, and if ε is any positive number, then, as the number of random vectors tends to infinity, there is probability one of choosing an $\underline{x}$ that satisfies $F(\underline{x}) \le F(\underline{x}^*) + \varepsilon$. Two approaches of this kind, namely 'simulated annealing' and 'genetic algorithms' are highly active fields of research, and the procedures that have been developed are employed often in practice. Indeed, the books by van Laarhoven and Aarts (1987) and by Goldberg (1989), respectively, each contain more than 200 references.

Genetic algorithms require a one-to-one correspondence between strings of binary digits and vectors of variables in finite precision arithmetic. Thus there is a value of $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, for each string. An iteration constructs a new set of m strings from a given set of m strings, where m is a parameter, the value $m = 100$ being typical. New strings are generated in pairs, where each pair is derived from two 'parents' that are picked randomly from the given set. The random choice is biased towards better values of the objective function, so a string can be a parent more than once during the iteration, which provides some 'natural selection'. Further, some randomness also occurs in the procedure that breeds two children from the binary digits of their parents. Hence it is difficult to relate the actual changes of variables to the original optimization calculation. On the other hand, several properties of the simulated annealing method have interesting explanations in terms of the objective function and the sequence of iterations. Therefore that method will be considered in the remainder of this section.

The simulated annealing procedure is analogous to the cooling of a liquid to a solid state, starting at a high temperature. When the liquid is hot, very many changes of state can occur, which correspond to many changes of the variables of an optimization calculation, that are allowed to make the objective function worse. Then the energy of the system is lowered, and, if the rate of cooling is sufficiently slow, the liquid should become frozen in the state of least energy, the analogy in the optimization calculation being the required global minimum. An algorithm requires a weighted list of possible changes from one state to another. Moreover, the probability that a change is made depends on the objective functions of the states and on a parameter that corresponds to temperature. The following paragraph gives a method that applies these ideas in the discrete case when $\underline{x}$ is restricted to a finite set $\mathcal{X} \subset \mathcal{R}^n$ that has m elements, say. In practice, m may be the number of different routes that can occur in the travelling salesman problem, but it is instructive to consider much smaller values of m.

Let $\mathcal{X}$ be the set $\{\underline{x}^{(i)} : i=1,2,\ldots,m\}$, and let W be a very sparse $m \times m$ matrix of nonnegative real numbers, such that W_{ij} is positive if and only if a transition from state $\underline{x}^{(i)}$ to $\underline{x}^{(j)}$ is allowed for high enough temperatures. The transition is more likely if W_{ij} is larger, and we assume the normalization

$$\sum_{j=1}^{m} W_{ij} = 1, \qquad i=1,2,\ldots,m. \tag{8.1}$$

The method requires these weights, a starting point $\underline{x}_1 \in \mathcal{X}$, and a positive parameter T, which is the initial value of the temperature. For $k=1,2,3,\ldots$, the vector of variables $\underline{x}_k$ is revised to $\underline{x}_{k+1}$ in the following way. Let q be the integer in $[1,m]$ that satisfies $\underline{x}^{(q)} = \underline{x}_k$. An integer p is picked at random from $[1,m]$, where the probability of choosing p is W_{qp}. The new vector $\underline{x}_{k+1}$ is always $\underline{x}^{(p)}$ in the case $F(\underline{x}^{(p)}) \le F(\underline{x}^{(q)})$. Otherwise, the new vector is either $\underline{x}^{(p)}$ or $\underline{x}^{(q)}$, the selection being random with the probabilities

$$\exp\left\{-[F(\underline{x}^{(p)}) - F(\underline{x}^{(q)})]/T\right\} \quad \text{or} \quad 1 - \exp\left\{-[F(\underline{x}^{(p)}) - F(\underline{x}^{(q)})]/T\right\}, \tag{8.2}$$

respectively. Occasionally an iteration may reduce the value of T, by observing some rules that are considered later. These rules also provide a condition for terminating the sequence of iterations.

We consider the probability that $\underline{x}_{k+1}$ is equal to $\underline{x}^{(i)}$, when the positive number T is not altered in the algorithm that has just been described. For convenience of notation, we assume without loss of generality that the sequence $F(\underline{x}^{(i)})$, $i=1,2,\ldots,m$, increases monotonically. Therefore, if $\underline{x}_k=\underline{x}^{(j)}$, then $\underline{x}_{k+1}=\underline{x}^{(i)}$ occurs with probability P_{ij}, where P is the $m\times m$ matrix that has the elements

$$P_{ij} = \begin{cases} W_{ji}, & i<j, \\ W_{jj} + \sum_{\ell=j+1}^{m} W_{j\ell} \left(1 - \exp\left\{-[F(\underline{x}^{(\ell)})-F(\underline{x}^{(j)})]/T\right\}\right), & i=j, \\ W_{ji} \exp\left\{-[F(\underline{x}^{(i)})-F(\underline{x}^{(j)})]/T\right\}, & i>j, \end{cases} \tag{8.3}$$

except that the sum is suppressed in the case $i=j=m$. It follows that, if $\underline{v}_k$ is the vector in $\mathcal{R}^m$ whose j-th component is the probability that $\underline{x}_k=\underline{x}^{(j)}$ holds, where j is any integer in $[1,m]$, then the i-th component of the product $\underline{v}_{k+1}=P\underline{v}_k$ is the probability that $\underline{x}_{k+1}$ is equal to $\underline{x}^{(i)}$. Further, if s is the integer in $[1,m]$ such that $\underline{x}_1$ is $\underline{x}^{(s)}$, and if $\underline{e}_s$ is the s-th coordinate vector in $\mathcal{R}^m$, then we deduce the formula

$$\underline{v}_k = P^{k-1}\underline{e}_s, \qquad k=1,2,3,\ldots\,. \tag{8.4}$$

Now the property

$$\|P\|_1 = \max\{\textstyle\sum_{i=1}^m |P_{ij}| = \sum_{i=1}^m P_{ij} = 1 : j=1,2,\ldots,m\} = 1 \tag{8.5}$$

implies that the spectral radius of P is at most one, and in fact it equals one, because of the eigenvalue equation $\underline{e}^T P = \underline{e}^T$, where the components of $\underline{e}\in\mathcal{R}^m$ are all one. Moreover, we make a nondegeneracy assumption that is suitable for the present application, namely that there is only one eigenvalue of P with modulus equal to the spectral radius. Hence, if $\underline{v}_*\in\mathcal{R}^m$ is a nonzero solution of $P\underline{v}_*=\underline{v}_*$, then the sequence of vectors (8.4) converges to a multiple of $\underline{v}_*$ as $k\to\infty$. Further, because every vector in this sequence is nonnegative and has components that sum to one, the limit of the sequence has these properties too. It follows that we can normalize $\underline{v}_*$ so that its components also sum to one, and then formula (8.4) provides $\lim_{k\to\infty}\underline{v}_k=\underline{v}_*$, for every choice of the integer s. Therefore the limiting behaviour of the algorithm of the previous paragraph is that the states $\underline{x}^{(i)}$, $i=1,2,\ldots,m$, occur with probabilities that are equal to the corresponding components of the nonnegative vector $\underline{v}_*$.

Let $(\underline{v})_i$ denote the i-th component of $\underline{v}\in\mathcal{R}^n$. When T is small, it is usual for $(\underline{v}_*)_i$ to be of magnitude $\exp\{-[F(\underline{x}^{(i)})-F(\underline{x}^{(1)})]/T\}$ for $i=1,2,\ldots,n$. It follows from the previous paragraph that the simulated annealing algorithm is likely to pick the state $\underline{x}^{(1)}$, which is the optimal vector of variables. A way of making the assertion about magnitudes plausible depends on the $m\times m$ diagonal matrix D with the diagonal elements $\exp\{-F(\underline{x}^{(i)})/T\}$, $i=1,2,\ldots,m$. Specifically, we define $\hat{\underline{v}}_*\in\mathcal{R}^m$ to be the positive multiple of $D^{-1}\underline{v}_*$ that satisfies $\|\hat{\underline{v}}_*\|_2=1$, we deduce from the eigenvalue equation $P\underline{v}_*=\underline{v}_*$ that $\hat{\underline{v}}_*$ is an eigenvector of the matrix $Q = D^{-1}PD$ with eigenvalue one, and we find that expression (8.3) provides the elements

$$Q_{ij} = \begin{cases} W_{ji} \exp\left\{-[F(\underline{x}^{(j)})-F(\underline{x}^{(i)})]/T\right\}, & i<j, \\ W_{jj} + \sum_{\ell=j+1}^{m} W_{j\ell} \left(1 - \exp\left\{-[F(\underline{x}^{(\ell)})-F(\underline{x}^{(j)})]/T\right\}\right), & i=j, \\ W_{ji}, & i>j. \end{cases} \qquad (8.6)$$

Thus, if $T \to 0$, the matrix Q remains bounded, and each element Q_{ij} with $i<j$ tends to zero in the usual case when the inequality $F(\underline{x}^{(i)}) \le F(\underline{x}^{(j)})$ is strict. Further, the first row of Q tends to be a multiple of the coordinate vector $\underline{e}_1^T$ if $F(\underline{x}^{(1)})$ is less than $F(\underline{x}^{(2)})$. Now, because Q is similar to P, the nondegeneracy assumption of the previous paragraph implies that all but one of the eigenvalues of Q have modulus less than one, so, apart from scaling by a constant, $\hat{\underline{v}}_*$ is the only eigenvector of Q with eigenvalue one. These remarks suggest that $(\hat{\underline{v}}_*)_1$ may remain bounded away from zero as $T \to 0$. Further, the nearly upper triangular structure of Q and the equation $Q\,\hat{\underline{v}}_* = \hat{\underline{v}}_*$ make it possible that none of the components of $\hat{\underline{v}}_*$ tends to zero as $T \to 0$. Then the magnitudes of the components of $\underline{v}_*$ can be derived from the fact that $\underline{v}_*$ is the multiple of $D\,\hat{\underline{v}}_*$ whose components sum to one, which yields the assertion under consideration.

When F is a function of one variable, and when $\mathcal{X}$ contains m different real numbers $\underline{x}^{(i)}$, $i=1,2,\ldots,m$, then a typical choice of the matrix W lets W_{ij} be nonzero if and only if $i \neq j$ and there are no elements of $\mathcal{X}$ between the numbers $\underline{x}^{(i)}$ and $\underline{x}^{(j)}$. Further, it lets the nonzero elements in each row of W be the same. Hence, if α and β are the integers in $[1,m]$ such that $\underline{x}^{(\alpha)}$ and $\underline{x}^{(\beta)}$ are the least and greatest elements of $\mathcal{X}$, respectively, then the α-th and β-th rows of W each contain only one nonzero element, which has the value 1, while the other rows of W each contain two nonzero elements, and they have the value 1/2, in accordance with equation (8.1). The reason for mentioning this example is that it provides excellent corroboration for the suggestions in the previous paragraph. Indeed, it is straightforward to show that the vector $\underline{v} \in \mathcal{R}^m$ that has the positive components

$$v_i = (1 - \tfrac{1}{2}\delta_{i\alpha} - \tfrac{1}{2}\delta_{i\beta}) \exp\left\{-F(\underline{x}^{(i)})/T\right\}, \qquad i=1,2,\ldots,m, \qquad (8.7)$$

satisfies the eigenvalue equation $P\underline{v} = \underline{v}$. Therefore $\underline{v}_*$ is the multiple of $\underline{v}$ whose components sum to one, which agrees completely with the suggestions.

It can also happen that the limit (8.4) is unhelpful to the calculation of a global minimum. For example, we consider the case when $n=1$, $m=4$ and $\underline{x}^{(i)} = i$, $i=1,2,3,4$, but we depart from the previous ordering of points of $\mathcal{X}$ by picking the function values

$$F(\underline{x}^{(1)})=0, \quad F(\underline{x}^{(2)})=3, \quad F(\underline{x}^{(3)})=1 \quad \text{and} \quad F(\underline{x}^{(4)})=2. \qquad (8.8)$$

Further, we let each row of W have two elements of zero and two elements of 1/2, where $W_{ij}=1/2$ occurs if and only if $\underline{x}^{(j)}$ is one of the two nearest neighbours of

$\underline{x}^{(i)}$. Thus P is the matrix

$$P = \begin{pmatrix} 1-\frac{1}{2}\theta-\frac{1}{2}\theta^3 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2}\theta^3 & 0 & \frac{1}{2}\theta^2 & \frac{1}{2}\theta \\ \frac{1}{2}\theta & \frac{1}{2} & 1-\frac{1}{2}\theta-\frac{1}{2}\theta^2 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2}\theta & \frac{1}{2}-\frac{1}{2}\theta \end{pmatrix}, \tag{8.9}$$

where θ denotes $\exp(-1/T)$. Hence straightforward calculation gives the exact solution

$$\underline{v} = \begin{pmatrix} (2+\theta)\,\theta \\ (2+\theta)\,(1+\theta^2)\,\theta^2 \\ (2+\theta^2)\,(1+\theta) \\ (2+\theta^2)\,\theta \end{pmatrix} \tag{8.10}$$

of the eigenvalue equation $P\underline{v}=\underline{v}$. Therefore, when T is small, the given algorithm tends to provide the vector of variables $\underline{x}^{(3)}$, although it is clear that $F(\underline{x}^{(1)})$ is the least of the function values (8.8). This pathological behaviour occurs because a transition from the $\underline{x}^{(1)}$ state to the $\underline{x}^{(3)}$ state is more likely than a return from the $\underline{x}^{(3)}$ state to the $\underline{x}^{(1)}$ state, although $F(\underline{x}^{(1)})$ is less than $F(\underline{x}^{(3)})$. Indeed, the return from $\underline{x}^{(3)}$ to $\underline{x}^{(1)}$ has to be via $\underline{x}^{(2)}$, and the large value of $F(\underline{x}^{(2)})$ is obstructive. Therefore it is helpful if the sparsity structure of the matrix W is symmetric.

The rate of cooling is very important to efficiency, especially when the starting vector of variables $\underline{x}_1 \in \mathcal{X}$ is near or at a local minimum of the objective function that is not optimal. We address some of the questions that arise when $\mathcal{X}$ is the finite set $\{\underline{x}^{(i)} : i=1,2,\ldots,m\}$ as before. Further, we suppose for convenience that the function values $F(\underline{x}^{(i)})$, $i=1,2,\ldots,m$, are all different. Therefore it is suitable to define $\underline{x}^{(j)}$ to be a local minimum if, for every integer i in $[1,m]$ that satisfies $i \neq j$ and $W_{ij} > 0$, the strict inequality $F(\underline{x}^{(i)}) > F(\underline{x}^{(j)})$ holds. Because the simulated annealing method is likely to pick a vector of variables that is a local minimum, and because we can regard any $\underline{x}_{k+1}$ as a starting point for the subsequent iterations, we address the case when the initial vector $\underline{x}_1 = \underline{x}^{(s)}$ is a local minimum. Then the probability that $\underline{x}_2$ is different from $\underline{x}_1$ has the value $\sum_{i=1,\, i\neq s}^{m} W_{is} \exp\{-[F(\underline{x}^{(i)}) - F(\underline{x}^{(s)})]/T\}$, which tends to zero if $T \to 0$. Further, when $\underline{x}_2$ is different from $\underline{x}_1$, then the most likely choice of $\underline{x}_3$ may be $\underline{x}^{(s)}$ again. These remarks indicate why a fairly large value of T should be chosen initially.

A suitable way of making this choice automatically is as follows. For each iteration number k, let the integers p and q be taken from expression (8.2), so q is defined by $\underline{x}^{(q)} = \underline{x}_k$, while $\underline{x}^{(p)}$ is the new vector of variables if $\underline{x}_{k+1}$ is different from $\underline{x}_k$. We make use of the remark that $F(\underline{x}^{(p)})$–$F(\underline{x}^{(q)}) = \Delta_k$, say, is independent of T, and we aim for a probability of about 50% that an iteration will change the vector of variables when a change would increase the objective function. Therefore, if k is the index of the first iteration that gives $\Delta_k > 0$, we define a provisional value of T by the condition $\exp(-\Delta_k/T)=1/2$. Then, on subsequent iterations, the provisional T is revised to an approximate solution of the equation

$$\text{Average value of } \{\exp(-\Delta_j/T) : j\in[1,k],\ \Delta_j>0\} \;=\; 1/2, \tag{8.11}$$

until the number of values of j in the braces reaches a prescribed amount. For instance, we may fix the initial T by condition (8.11) when 50 of the numbers Δ_j, $j=1,2,\ldots,k$, are positive, and this T may be used for hundreds of iterations.

The analogy with the cooling of a liquid motivates some automatic reductions in T, but it has been mentioned that the temperature should not be decreased too rapidly. Specifically, we require the property that, if $\underline{x}_1=\underline{x}^{(s)}$ is a 'strict local minimum', then the probability of choosing $\underline{x}_{k+1}=\underline{x}_k$ on every iteration is zero for $k\rightarrow\infty$. Here 'strict local minimum' means that, if a single iteration can change the variables from $\underline{x}^{(s)}$ to $\underline{x}^{(\ell)}$, then $F(\underline{x}^{(\ell)}) > F(\underline{x}^{(s)})$ holds. The probability of $\underline{x}_{k+1} = \underline{x}_k$ when $\underline{x}_k = \underline{x}^{(s)}$ is the P_{ss} element of expression (8.3), assuming the usual ordering $F(\underline{x}^{(i)}) \leq F(\underline{x}^{(i+1)})$, $i=1,2,\ldots,m-1$. We also assume the strict inequality $F(\underline{x}^{(s)}) < F(\underline{x}^{(s+1)})$. Therefore, letting T_k be the current temperature, this probability has the lower bound

$$\begin{aligned} P_{ss} \;&\geq\; W_{ss}+\textstyle\sum_{\ell=s+1}^{m} W_{s\ell}\left(1-\exp\left\{-[F(\underline{x}^{(s+1)})-F(\underline{x}^{(s)})]/T_k\right\}\right) \\ &\geq\; 1-\exp\left\{-[F(\underline{x}^{(s+1)})-F(\underline{x}^{(s)})]/T_k\right\} \;=\; 1-\exp(-\Delta_*/T_k), \end{aligned} \tag{8.12}$$

say, the second inequality being valid because the local minimum property of $\underline{x}^{(s)}$ provides $\sum_{i=s}^{m} W_{si}=1$. Thus there is zero probability of no change to the variables only if the sequence of temperatures T_k, $k=1,2,3,\ldots$, satisfies the equation

$$\textstyle\prod_{k=1}^{\infty}\{1-\exp(-\Delta_*/T_k)\}=0, \tag{8.13}$$

which is equivalent to the condition

$$\textstyle\sum_{k=1}^{\infty}\exp(-\Delta_*/T_k)=\infty. \tag{8.14}$$

It follows that the cooling rate is too fast if we let each T_k be of magnitude $1/k$. Indeed, in this case the analytic formula for the sum of a geometric progression shows that the left hand side of expression (8.14) is finite.

On the other hand, if T_k is of magnitude $1/\log k$ for large k, then condition (8.14) can hold. Specifically, the choice $T_k=c\,/\log k$, where c is a positive constant, has the property (8.14) if and only if the integral

$$\int_1^{\infty}\exp(-\Delta_* c^{-1}\log\theta)\,d\theta \;=\; \int_0^{\infty}\exp\{(1-\Delta_* c^{-1})\,t\}\,dt \tag{8.15}$$

is infinite, where the second integral is due to change of variables $t=\log\theta$. Therefore we require c to be at least Δ_*, and a greater lower bound on c can allow for the inequalities of expression (8.12). Thus $T_k \sim 1/\log k$ is the cooling rate that is usually recommended for a simulated annealing algorithm, as mentioned in van Laarhoven and Aarts (1987).

In practice, however, it is easy to avoid $\underline{x}_k = \underline{x}^{(s)}$ for every k by not decreasing T until $\underline{x}_{k+1}$ is different from $\underline{x}_k$. Indeed, a typical 'cooling schedule' would begin by generating T by the method described earlier that satisfies equation (8.11) for a suitable choice of k. Then, whenever 50 iterations, say, have provided increases in the objective function for the current T, either the temperature is multiplied by 0.9 or the calculation is terminated, the last action being taken when T is sufficiently small. Moreover, in the usual case when the range of the variables $\underline{x} \in \mathcal{R}^n$ is a continuum instead of a discrete set $\mathcal{X}$, a technique is required for choosing the vector $\underline{x}^{(p)}$ of the method in the paragraph that includes equation (8.1). It is often suitable to pick $\underline{x}^{(p)}$ at random from the set $\{\underline{x} : \|\underline{x} - \underline{x}_k\|_2 \le \rho\}$, where ρ is a prescribed positive constant. Then the rules for deciding between the alternatives $\underline{x}_{k+1} = \underline{x}^{(q)}$ and $\underline{x}_{k+1} = \underline{x}^{(p)}$ are the same as before. There are several computer programs that apply the simulated annealing method. For example, a program for the solution of the travelling salesman problem is given in the numerical recipes book by Press, Flannery, Teukolsky and Vetterling (1986).

9. Discussion

Usually it is very difficult to discover general convergence theorems for algorithms for nonlinear optimization that perform well in practice, especially when first derivatives are not available. Therefore the author has always supported the view that new algorithms should be developed from ideas for techniques that may provide improvements to actual calculations. Further, as well as trying the techniques in numerical experiments, in order to find any advantages over other methods, it is important to study what can go wrong. Such research is hardly ever conclusive, and the outcome may be an algorithm that compares favourably with other procedures on a range of test problems, but that fails occasionally. Many papers on such work are published in journals and even more are rejected. On the other hand, the literature also includes several descriptions of algorithms that are designed so that convergence properties can be proved, which rules out methods that allow the construction of examples that show failure. In particular, both the procedure that makes exact line searches along coordinate directions recursively and the method of Nelder and Mead (1965) are excluded, because of the counter-examples that are mentioned in Sections 2 and 4, respectively. Indeed, there seems to be hardly any correlation between the algorithms that are in regular use for practical applications and the algorithms that enjoy guaranteed convergence in theory, although advances on the theoretical side should influence the development of software for optimization calculations. We will consider the methods of Sections 2–8 in the light of these comments.

The convergence theory of Section 2 presents a result that is not well-known. It is that the conditions (2.3) and (2.8) on the search directions and step-lengths, respectively, can ensure that $\liminf\{\|\underline{\nabla} F(\underline{x}_k)\| : k=1,2,3,\ldots\}$ is zero in exact arithmetic, provided that the objective function has continuous first derivatives and the sequence $\underline{x}_k$, $k=1,2,3,\ldots$, has a limit point. This property is achieved by a method that requires the positive parameters β_k, $k=1,2,3,\ldots$, that are introduced just before expression (2.8), but the need for these data is objectionable. Indeed, if the parameters are not superfluous, then they fix some important decisions about step-lengths before all or most of the values of the objective function are calculated. Instead it would be reasonable to generate the parameters automatically as the iterations proceed, which would also assist the use of the algorithm. We explain this remark by assuming that $F(\underline{x}_p)-F(\underline{x}^*)$ is less than $\gamma\beta_p^2$ at the beginning of the p-th iteration, where $F(\underline{x}^*)$ is the least value of the objective function. Then the first line of expression (2.8) implies $\|\alpha\,\underline{d}_p\|_2<\beta_p$, but the procedure in Section 2 sets the step-length α_p to zero in this case. Further, if q is the number of the next iteration that changes the variables, then $\underline{x}_q=\underline{x}_p$ and $\gamma\beta_q^2\le F(\underline{x}_p)-F(\underline{x}^*)$ must hold, while the iterations with numbers $k\in[p,q-1]$ each calculate at least one value of $F(\underline{x}_k+\alpha\,\underline{d}_k)$, although they cannot alter the current vector of variables until $\gamma\beta_k^2$ is reduced to at most $F(\underline{x}_p)-F(\underline{x}^*)$. Therefore it is suitable to pick $\beta_{k+1}<\beta_k$ during these iterations, and to set $\beta_{k+1}=\beta_k$ in the case $\underline{x}_{k+1}\ne\underline{x}_k$, but $\beta_{k+1}=\beta_k$ may be safer than $\beta_{k+1}<\beta_k$ if the reason for $\underline{x}_{k+1}=\underline{x}_k$ is that $\underline{d}_k$ is practically orthogonal to the unknown gradient $\underline{\nabla} F(\underline{x}_k)$. Thus the initial choice of β_k, $k=1,2,3,\ldots$, may be avoided, by letting β_{k+1} depend on β_k and available values of F, which is like the adjustment of trust region radii in trust region algorithms. Further, if the differences $\beta_k-\beta_{k+1}$ are sufficiently parsimonious, then it may be possible to replace both parts of expression (2.8) by the single condition

$$F(\underline{x}_k+\alpha\,\underline{d}_k)\le F(\underline{x}_k)-\gamma\beta_k^2 \quad \text{if} \quad \alpha\ne 0, \tag{9.1}$$

with the proviso that the step-length is set to zero only if inequality (9.1) fails in the case $\alpha=\beta_k/\|\underline{d}_k\|_2$. These remarks suggest that some useful convergence properties can be achieved in a line search method, by using search directions that satisfy expression (2.3), and by introducing one or two trust region ideas into the control of step-lengths.

The discrete grid methods, considered in Section 3, are both an inspiration and a target for adverse criticism. The analysis of convergence given there justifies the title "Direct search methods: once scorned, now respectable" of Wright (1996). Moreover, if the set $\{\underline{x} : F(\underline{x})\le F(\underline{x}_1)\}$ is bounded and if the mesh size does not change, then eventually a vector of variables is generated that has occurred already during the calculation, which is useful, because this situation can activate operations like reductions in the mesh size, as stated in Section 3 and in the last paragraph of Section 7. On the other hand, examples like the minimization of the function

$$F(\underline{x}) = F(x_1, x_2) = (x_1 - x_2 + 0.2)^2 + 10^{-4}(x_1 + x_2 + 77)^2, \qquad \underline{x} \in \mathcal{R}^2, \qquad (9.2)$$

cause some consternation. Specifically, if the points of the grid are all vectors whose components are integers, and if $F(0,0) = 0.6329$ has been calculated, then the best grid point so far has to satisfy $x_1 = x_2$. Therefore it cannot be one of the four neighbours of $(0,0)$. Further, the least value of the continuous function (9.2) occurs at $(x_1, x_2) = (-38.6, -38.4)$, and the nearest grid point is $(-39, -38)$, which gives a function value that exceeds $F(0,0)$. Thus the use of discrete grids can impair the efficiency of the algorithm of Elster and Neumaier (1995), for example, even when F is quadratic. These disadvantages of restricting the vectors of variables to discrete grids are well known, so 'respectability' has been achieved by the recent work on convergence theory.

I have never liked the simplex methods of Section 4, because it seems very wasteful to make changes to the variables in ways that depend only on the signs of differences between calculated function values, the magnitudes of the differences being ignored. Therefore I was surprised to be told more than ten years ago that the Nelder and Mead (1965) algorithm is the method for unconstrained optimization that is used most often. I was even more surprised to learn later about the cases of failure that are reported at the end of Section 4. These remarks provide a clear demonstration that there is a strong need for easy-to-use algorithms that are more reliable. New procedures are proposed occasionally. For example, Kelley (1997) has developed an 'oriented restart' technique that replaces the current simplex by a more suitable one automatically, when the current simplex is found to be nearly degenerate. Another possible way of avoiding collapse is to confine all the vertices of all the simplices to a discrete grid. Then it may be possible to generalise equation (4.2) to the formula

$$\hat{\underline{v}}_m = 2 \sum_{\substack{i=1 \\ i \neq m}}^{n+1} \theta_i \, \underline{v}_i - \underline{v}_m, \qquad (9.3)$$

where the multipliers θ_i are nonnegative and sum to one. Thus the next vector of variables can depend on the actual values of the objective function at the vertices of the current simplex, which may provide better efficiency in typical applications.

The conjugate direction method of Powell (1964), described in Section 5, is another algorithm that has been in regular use for more than 30 years. When I developed it, I was concerned about the choice of m in the paragraph that includes expressions (5.9)–(5.13), because the given procedure may not retain those search directions that have already been given the required conjugacy properties in the quadratic case. Then I took the view that it was possible to secure convergence by using search directions that span the full space of the variables. However, we know now, from the example in the third paragraph of Section 2, that even condition (2.3) on an infinite sequence of search directions does not guarantee $\liminf_{k\to\infty} \|\underline{\nabla} F(\underline{x}_k)\| = 0$ when the line searches are exact, although F can be very smooth with bounded level sets. Therefore, because sufficiently accurate line searches are vital to the given techniques that achieve conjugacy, the convergence properties of conjugate direction methods deserve further attention. It is possible that our remarks on inequality (9.1), that suggest a zero step-length if the inequality fails, may be helpful.

The first version of the method in Section 6 was developed for Westland Helicopters, in order to solve a range of optimization problems that each have about 4 variables and 10 constraints. It was anticipated correctly that, due to the tiny number of variables, the inefficiencies that arise from linear approximations to the objective and constraint functions would be tolerable. An easy-to-use Fortran program was written that implements the algorithm of Powell (1994), and it is available from the author. The program is very slow, as expected, when there are no constraints, because of the importance of the curvature of the objective function. On the other hand, linear approximations to constraints are usually excellent for reducing the freedom in the variables in a suitable way. We address one convergence question, namely whether the method of Section 6 terminates if only a finite number of iterations cause $\underline{x}_{k+1}$ to be different from $\underline{x}_k$. This assumption is reasonable, because of the limited precision of the computer arithmetic, and because $\underline{x}_{k+1} \neq \underline{x}_k$ requires the strict inequality $\Psi(\underline{x}_{k+1}) < \Psi(\underline{x}_k)$, where Ψ is the merit function (6.14). We give careful attention to the complete paragraph that follows equation (6.10), remembering that the number of reductions in Δ_k is also finite. Thus we find that termination does not occur if and only if the situations (5) and (6) are not reached during an infinite number of consecutive iterations, and we suppose that termination fails in this way. Then Δ_k is independent of k, and the above assumption allows $\underline{x}_k$ to be independent of k too, without loss of generality. In other words, no iteration improves the best vector of variables so far, which excludes rule (1) in the paragraph we are considering. The number of minimization steps is infinite, however, because otherwise every step would be a simplex step eventually, which would give a contradiction, due to the increases in the volume of the simplex after the inequality

$$\max\{\|\underline{v}_i - \underline{x}_k\| : i = 1, 2, \ldots, n+1\} \leq \beta\,\Delta_k \tag{9.4}$$

is achieved. Therefore the trust region radius ρ_k is reduced until $\rho_k = \Delta_k$ holds for all sufficiently large k. Then $\beta > 1$ implies that a minimization step does not increase the number of integers i in $[1, n+1]$ that satisfy $\|\underline{v}_i - \underline{x}_k\| > \beta\,\Delta_k$, nor does it diminish the volume of the simplex, although a nonoptimal vertex of the simplex may be changed. It follows that the volume of the simplex would become unbounded if the number of simplex steps were infinite, so rules (2) and (3) are also irrelevant for large k. Therefore every simplex becomes acceptable. Further, we deduce from $\rho_k = \Delta_k$ that there are no more 'questionable' minimization steps. Thus rule (4) is excluded too, which completes the proof of termination. One purpose of this analysis is to justify the given rules. In particular, lack of termination would be prevalent if the last proviso of rule (4) were deleted, because it is usual for the minimization step to satisfy $\|\underline{v}_{n+2} - \underline{x}_k\| = \rho_k$ when Φ is a linear function.

The analysis of termination in the previous paragraph applies also to the algorithm that is the main subject of Section 7. Further, we can generalize that algorithm by letting Φ be an approximation to F from any prescribed finite dimensional linear space of functions from $\mathcal{R}^n$ to $\mathcal{R}$, say $\mathcal{A}$. Then $\hat{n}$ is the dimension of $\mathcal{A}$, and each iteration begins with points $\underline{v}_i$, $i = 1, 2, \ldots, \hat{n}$, such that the equations (7.1) define Φ uniquely. Therefore $\mathcal{A}$ includes cardinal functions χ_i, $i = 1, 2, \ldots, \hat{n}$, that are derived from the conditions (7.2) as before. Further, all the uses of the cardinal functions in Section 7 are preserved, including the definition of the 'simplex step'. In particular, if $\mathcal{A}$ is the $(n+1)$-dimensional space of linear polynomials, then the generalization reduces to the method of Section 6 when there are no constraints. A choice of $\mathcal{A}$ that is between the linear polynomial case and the $\hat{n} = \frac{1}{2}(n+1)(n+2)$ quadratic polynomial case has been suggested by Philippe Toint (private communication), and I expect it to be highly useful. It is relevant when F has the form

$$F(\underline{x}) = \sum_{j=1}^{t} F_j(\underline{x}), \qquad \underline{x} \in \mathcal{R}^n, \tag{9.5}$$

where each of the functions F_j, $j = 1, 2, \ldots, t$, is independent of most of the components of $\underline{x}$. Then we let $\mathcal{S}$ be the set of pairs of integers $\{p, q\}$ from $[1, n]$, such that $\{p, q\}$ is in $\mathcal{S}$ if and only if no F_j depends on both x_p and x_q. Alternatively, if F is twice differentiable, then we pick the set

$$\mathcal{S} = \{\{p, q\} : \partial^2 F(\underline{x}) \,/\, \partial x_p \, \partial x_q \equiv 0\}, \tag{9.6}$$

which may be larger than before. Of course we let $\mathcal{A}$ be the linear space of quadratic polynomials with the zero second derivatives $\partial^2 \Phi(\underline{x}) \,/\, \partial x_p \, \partial x_q = 0$, $\{p, q\} \in \mathcal{S}$, for every Φ in $\mathcal{A}$. Thus the elements of $\mathcal{A}$ are given some of the sparsity properties of F, and $\hat{n}$ may be much less than $\frac{1}{2}(n+1)(n+2)$, which would provide large reductions in the amount of routine computation. Another question that is important to applications is whether the method of this paragraph can be extended to allow constraints on the variables. It is straightforward to approximate the constraints $c_p(\underline{x}) \geq 0$, $p = 1, 2, \ldots, m$, by the inequalities $\gamma_p(\underline{x}) \geq 0$, $p = 1, 2, \ldots, m$, where each γ_p is the element of $\mathcal{A}$ that is defined by the equations $\gamma_p(\underline{v}_i) = c_p(\underline{v}_i)$, $i = 1, 2, \ldots, \hat{n}$, at the beginning of the iteration. The calculation of a minimization step would be difficult, however, if we had to minimize a quadratic function subject to quadratic constraints. Therefore it may be best to give most attention to linear constraints for a while. The algorithm of Elster and Neumaier (1995), for instance, allows simple bounds on the variables.

The simulated annealing method of Section 8 is attractive to many computer users, because it is easy to apply, the amount of routine work of each iteration is only linear in n, and some theory suggests that it is possible to find the global solution of several optimization problems. Further, in many applications, any reduction in the value of the objective function may be financially rewarding, which does not favour algorithms that stop at a local minimum. Indeed, the failures of the method of Nelder and Mead (1965) may seem to be no worse than termination at a local minimum. Of course these remarks depend on the nature of the application, partly because high accuracy would be inappropriate for many models of real situations. On the other hand, there is also a wide range of precise optimization problems in science and engineering, so there are strong reasons for further developments of the techniques of Sections 2–7, and it would be good to have some efficient software for such problems that is attractive to computer users. My opinion of simulated annealing is that, for small and moderate values of n, it is highly inefficient to take decisions randomly. Therefore one may be able to construct procedures that provide similar results much more quickly, in cases when most of the computing time is spent on calculations of values of the objective function. Indeed, the importance of global optimization in practice makes an excellent case for much more work on general algorithms that need not be trapped by local minima.

References

R.P. Brent (1973), *Algorithms for Minimization without Derivatives*, Prentice–Hall (Englewood Cliffs).

K.W. Brodlie (1975), "A new direction set method for unconstrained minimization without evaluating derivatives", *J. Inst. Maths Applics*, Vol. 15, pp. 385–396.

A.R. Conn, K. Scheinberg & Ph.L. Toint (1997a), "On the convergence of derivative-free methods for unconstrained optimization", in *Approximation Theory and Optimization*, eds. M.D. Buhmann & A. Iserles, Cambridge University Press (Cambridge), pp. 83–108.

A.R. Conn, K. Scheinberg & Ph.L. Toint (1997b), "Recent progress in unconstrained nonlinear optimization without derivatives", *Math. Programming*, Vol. 79, pp. 397–414.

J.E. Dennis & V. Torczon (1991), "Direct search methods on parallel machines", *SIAM J. Optim.*, Vol. 1, pp. 448–474.

C. Elster & A. Neumaier (1995), "A grid algorithm for bound constrained optimization of noisy functions", *IMA J. Numer. Anal.*, Vol. 15, pp. 585–608.

R. Fletcher (1987), *Practical Methods of Optimization*, John Wiley & Sons (Chichester).

P.E. Gill, W. Murray & M.H. Wright (1981), *Practical Optimization*, Academic Press (London).

D. E. Goldberg (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley (Reading).

L. Grippo, F. Lampariello & S. Lucidi (1988), "Global convergence and stabilization of unconstrained minimization methods without derivatives", *J. Optim. Theory Appl.*, Vol. 56, pp. 385–406.

R. Hooke & T.A. Jeeves (1961), "Direct search solution of numerical and statistical problems", *J. Assoc. Comput. Mach.*, Vol. 8, pp. 212–229.

C.T. Kelley (1997), "Detection and remediation of stagnation in the Nelder–Mead algorithm using a sufficient decrease condition", North Carolina State University Report CRSC-TR97-2.

P.J.M. van Laarhoven & E.H.L. Aarts (1987), *Simulated Annealing: Theory and Applications*, Reidel Publishing Co. (Dordrecht).

S. Lucidi & M. Sciandrone (1997), "On the global convergence of derivative free methods for unconstrained optimization", preprint, Università di Roma 'La Sapienza', Italy.

K.I.M. McKinnon (1997), "Convergence of the Nelder–Mead simplex method to a nonstationary point", preprint (to be published in *SIAM J. Optim.*).

J.A. Nelder & R. Mead (1965), "A simplex method for function minimization", *Comput. J.*, Vol. 7, pp. 308–313.

M.J.D. Powell (1964), "An efficient method for finding the minimum of a function of several variables without calculating derivatives", *Comput. J.*, Vol. 7, pp. 155–162.

M.J.D. Powell (1973), "On search directions for minimization algorithms", *Math. Programming*, Vol. 4, pp. 193–201.

M.J.D. Powell (1994), "A direct search optimization method that models the objective and constraint functions by linear interpolation", in *Advances in Optimization and Numerical Analysis*, eds. S. Gomez & J-P. Hennart, Kluwer Academic (Dordrecht), pp. 51–67.

W.H. Press, B.P. Flannery, S.A. Teukolsky & W.T. Vetterling (1986), *Numerical Recipes: The Art of Scientific Computation*, Cambridge University Press (Cambridge).

H.H. Rosenbrock (1960), "An automatic method for finding the greatest or least value of a function", *Comput. J.*, Vol. 3, pp. 175–184.

C.S. Smith, (1962), "The automatic computation of maximum likelihood estimates", N.C.B. Sci. Dept. Report SC 846/MR/40.

W. Spendley, G.R. Hext & F.R. Himsworth (1962), "Sequential application of simplex designs in optimisation and evolutionary operation", *Technometrics*, Vol. 4, pp. 441–461.

Ph L. Toint & F.M. Callier (1977), "On the accelerating property of an algorithm for function minimization without calculating derivatives", *J. Optim. Theory Appl.*, Vol. 23, pp. 531–547.

V. Torczon (1997), "On the convergence of pattern search algorithms", *SIAM J. Optim.*, Vol. 7, pp. 1–25.

D. Winfield (1973), "Function minimization by interpolation in a data table", *J. Inst. Maths Applics*, Vol. 12, pp. 339–347.

M.H. Wright (1996), "Direct search methods: once scorned, now respectable", in *Numerical Analysis 1995*, eds. D.F. Griffiths & G.A. Watson, Addison Wesley Longman (Harlow), pp. 191–208.