A direct search conjugate directions algorithm for unconstrained minimization

I. D. Coope[*] C. J. Price[*]

(Received 7 August 2000)

Abstract

A direct search algorithm for unconstrained minimization of smooth functions is described. The algorithm minimizes the function over a sequence of successively finer grids. Each grid is defined by a set of basis vectors. From time to time these basis vectors are updated to include available second derivative information by making some basis

---

[*]Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand. `mailto:I.Coope@math.canterbury.ac.nz` and `mailto:C.Price@math.canterbury.ac.nz`

[0]See `http://anziamj.austms.org.au/V42/CTAC99/Coo2` for this article and ancillary services, © Austral. Mathematical Soc. 2000. Published 27 Nov 2000.

vectors mutually conjugate. Convergence to one or more stationary points is shown, and the finite termination property of conjugate direction methods on strictly convex quadratics is retained. Numerical results show that the algorithm is effective on a variety of problems including ill-conditioned problems.

Contents

1 Introduction

There has been much recent interest in derivative free methods for unconstrained optimization [1, 6, 9]. A variety of provably convergent methods have been described, including ones based on line searches, trust regions, and on grids. The algorithm presented here is in the last category, and uses the convergence theory developed in [2].

A minimizer of a given C^1 objective function $f : R^n \to R$ is sought, where the gradient ∇f of f is locally Lipschitz. The algorithm does not make explicit use of ∇f, but minimizes f by examining it on a sequence $\{\mathcal{G}^{(m)}\}_{m=1}^{\infty}$ of successively finer grids. Each grid $\mathcal{G}^{(m)}$ is defined by a set of n linearly independent basis vectors $\mathcal{V}^{(m)} = \left\{v_i^{(m)} : i \in 1, \ldots, n\right\}$. The points on the grid $\mathcal{G}^{(m)}$ are

$$\mathcal{G}^{(m)} = \left\{x \in R^n : x = x_{\text{o}}^{(m)} + h^{(m)} \sum_{i=1}^{n} \eta_i v_i^{(m)} \text{ for integer } \eta_i \ \forall i \in 1, \ldots, n\right\}$$

The parameter $h^{(m)}$ is referred to as the mesh size, and is adjusted as m is increased in order to ensure that the meshes become finer in a manner needed to establish convergence. The point $x_{\text{o}}^{(m)}$ is included to allow each grid to have a different origin to its predecessor. The grid points are referenced via

η rather than x to avoid the accumulation of round off errors from repeated movements on $\mathcal{G}^{(m)}$.

The algorithm seeks to minimize f over each grid $\mathcal{G}^{(m)}$ by generating a sequence of iterates $\{x^{(k)}\}_{k=1}^{\infty}$, where a minimiser of f over a grid is defined as follows:

Definition 1 (Grid local minimum) *A point x on the grid $\mathcal{G}^{(m)}$ is defined as a grid local minimum if and only if*

$$f(x + h^{(m)}v) \geq f(x) \quad \textit{and} \quad f(x - h^{(m)}v) \geq f(x) \quad \forall v \in \mathcal{V}^{(m)}$$

This definition is motivated by the observation that if

$$(\nabla f(x))^T v \geq 0 \quad \text{and} \quad (\nabla f(x))^T (-v) \geq 0 \quad \forall v \in \mathcal{V}^{(m)} \tag{1}$$

then x is a stationary point of f (see e.g. [2]). The conditions which define a grid local minimum are a finite difference approximation to this. In each main iteration of the algorithm, a grid $\mathcal{G}^{(m)}$ is selected using previous information, and a grid local minimiser of f over $\mathcal{G}^{(m)}$ is sought through a series of line searches along the directions in $\mathcal{V}^{(m)}$. In practice, a finite number of alterations to the grid are permitted during the line searches. An outline of the algorithm's form is as follows:

(i) Initialize all variables. $k = 1$.

(ii) Execute any finite process. If descent is obtained, let $x^{(k+1)}$ be the lowest known point and increment k. Search cyclically along the directions $v_1, \ldots, v_n$ for grid points which are lower than the current iterate. Each time a line search yields descent, $x^{(k+1)}$ is chosen as the lowest known point, and k is incremented. When a grid local minimum is found, proceed to the next step.

(iii) Execute any finite process. If descent is obtained, let $x^{(k+1)}$ be the lowest known point and increment k. Form a new grid with its origin at the current lowest iterate. If stopping criteria are not satisfied, go to step (ii).

It is shown in [2] that, under mild conditions, an algorithm with this framework generates a sequence of grid local minima which converge to one or more stationary points of f. For convenience this theorem is restated here, with a slight specialization to reflect the definition of a grid local minimum used herein.

Theorem 2 *Given*

(a) The sequence of iterates $\{x^{(k)}\}_{k=1}^{\infty}$ is bounded;

(b) $f(x)$ is continuously differentiable, and its gradient $\nabla f(x)$ is Lipschitz in any bounded region of R^n;

(c) There exist positive constants K and τ_{det} such that $|\det(v_1^{(m)} \dots v_n^{(m)})| \geq \tau_{det}$ and $\|v_i^{(m)}\| \leq K$ for all m and i; and

(d) $h^{(m)} \to 0$ as $m \to \infty$;

then each cluster point $\hat{x}^{(\infty)}$ of the subsequence $\{\hat{x}^{(m)}\} \subseteq \{x^{(k)}\}$ is a stationary point of $f(x)$. Here each $\hat{x}^{(m)}$ is the grid local minimum of $\mathcal{G}^{(m)}$ found by the algorithm.

Proof: See [2]. ♠

2 General Description of the Algorithm

The members of $\mathcal{V}^{(m)}$ are chosen to maintain any known second derivative information in the form of mutually conjugate directions. The set of directions $\mathcal{V}^{(m)}$ is divided into two subsets: $\mathcal{V}_c^{(m)} = \{v_1^{(m)}, \dots, v_c^{(m)}\}$ and $\mathcal{V}_{\mathrm{nc}}^{(m)} = \{v_{c+1}^{(m)}, \dots, v_n^{(m)}\}$. The members of $\mathcal{V}_c$ are regarded as mutually conjugate, whereas the members of $\mathcal{V}_{\mathrm{nc}}$ are not. These basis vectors form the columns of the matrix $V^{(m)} = \left[v_1^{(m)} \dots v_n^{(m)}\right]$. For convenience the matrices $V_c^{(m)}$ and $V_{\mathrm{nc}}^{(m)}$ will be used to refer to the first c and the last $n-c$ columns of $V^{(m)}$ respectively.

The algorithm repeatedly conducts line searches along the directions in $\mathcal{V}^{(m)}$ until a grid local minimum is found. Between grid local minima, existing members of $\mathcal{V}_c^{(m)}$ are not changed during these line searches. Each member of $\mathcal{V}_{\text{nc}}^{(m)}$ can be changed once between grid local minima. This occurs when $v \in \mathcal{V}_{\text{nc}}^{(m)}$ is removed from $\mathcal{V}_{\text{nc}}^{(m)}$, and replaced by a new conjugate direction which is then included in the set $\mathcal{V}_c^{(m)}$. These new conjugate directions are generated using the parallel subspace theorem (see e.g.. [3, 7, 8]). This process continues until a grid local minimum is found. The directions in $\mathcal{V}_c^{(m)}$ are then scaled so that they have unit estimated curvature along them. This ensures that, when $c = n$, $V_c V_c^T$ is the inverse Hessian on a strictly convex quadratic.

Each new conjugate direction changes the grid $\mathcal{G}^{(m)}$. Each such grid alteration removes a vector from $\mathcal{V}_{\text{nc}}$, hence only a finite number of such alterations can be made without locating a grid local minimum. These alterations are permitted as part of the finite process in step (ii) of the algorithm outline.

At each grid local minimum, if less than a full set of conjugate directions is known, then these are retained. Otherwise the members of $\mathcal{V}^{(m)}$ are reordered, the conjugate directions are no longer regarded as such, and the process begins again with $c = 1$. On each new grid the conjugate directions are searched first. This minimizes the number of line searches needed to generate new conjugate directions.

At each grid local minimizer, a second order estimate $\hat{g}_v^{(m)}$ of $V^T \nabla f$ is obtained. On noting VV^T approximates the inverse Hessian, the Newton step $p = -(\nabla^2 f)^{-1} \nabla f$ can be estimated. The algorithm conducts a brief search along p for a lower point before selecting the next grid. This search forms part of the finite process in step (iii).

2.1 The Line and Ray Searches

The form of the algorithm requires that a search from an iterate x along $v_i^{(m)}$ may be abandoned only after f has been calculated at $x + h_i^{(m)} v_i^{(m)}$ and $x - h^{(m)} v_i^{(m)}$. Hence if the algorithm searches along all n directions $v_1^{(m)}, \ldots, v_n^{(m)}$ from x without finding a point lower than x, then x is a grid local minimum. If a lower point than x is located, then the algorithm searches further along that direction. More precisely, if $f(x + h^{(m)} v_i^{(m)}) < f(x)$ then a ray search along the ray $x + \alpha h^{(m)} v_i^{(m)}$, $\alpha > 0$ is performed; otherwise if $f(x - h^{(m)} v_i^{(m)}) < f(x)$ a ray search along the ray $x - \alpha h^{(m)} v_i^{(m)}$, $\alpha > 0$ is performed; otherwise the line search is terminated unsuccessfully.

Each ray search from x along v_o calculates $f(x + \alpha h^{(m)} v_\mathrm{o})$ at successively larger integer values of α as long as a decreasing sequence of function values is obtained. When the last value is not lower than the second to last value, then the ray search is terminated, and the penultimate α value determines the new iterate. The first two values are $\alpha = 1$ and $\alpha = 2$, unless $v_o = -v_i^{(m)}$, in which case the second value is $\alpha = -1$. Each subsequent α value is calculated using the formula $\alpha = \max(\alpha + 1, \min(8\alpha, \lfloor \alpha_q + 0.5 \rfloor))$ Here $\lfloor \cdot \rfloor$ denotes the floor function, and α_q is defined as the minimizer of the one dimensional quadratic interpolating the last three points on the line $x + \alpha h^{(m)} v_\mathrm{o}$ at which

f was calculated. If the interpolating quadratic is not strictly convex, then $\alpha_q = 8\alpha$ is used.

3 The Main Algorithm

The basic structure of the algorithm is as follows

1. Initialize $m = k = c = 1$, $i = 0$, starting point $x^{(0)}$. Set $x_\mathrm{b} =$ 'unknown', $h^{(0)} = \infty$, $h^{(1)} = 1$, and $V^{(1)} = I_n$.
2. (a) Set $i = i + 1$. If $i > n$, set $i = 1$. If $i = 1$ set $x_\mathrm{old} = x^{(k)}$.
   (b) execute a line search along the direction $v_i^{(m)}$ from $x^{(k)}$.
   (c) if $i = c$, $c < n$, and $x_\mathrm{b} \neq$ 'unknown', then augment the set of conjugate directions as described in section 3.2.
   (d) if a grid local minimum has been found go to step 3, otherwise alter h as specified in section 3.1.
   (e) if $i = n$ do a ray search along $x^{(k)} + \alpha(x^{(k)} - x_\mathrm{old})$, $\alpha > 0$, α integer. Go to step 2(a).
3. Calculate $g_v^{(m)}$ and scale each member of $\mathcal{V}_c^{(m)}$ so the estimated curvature along each direction is unity.
4. Perform a 2 point line search along the quasi-Newton direction.

5. If $f(x_{\mathrm{e}}) < f(x^{(k+1)})$, then set $x^{(k+1)} = x_{\mathrm{e}}$.

6. Choose $h^{(m+1)} = h^{(m)}/s_r$ and updates_r.

7. If $c \geq n$ set $c = 1$, and $x_{\mathrm{b}} =$ 'unknown'. Set $v_1^{(m+1)} = v_n^{(m)}$, set $v_i^{(m+1)} = v_{i-1}^{(m)}$ for all $i = 2, \ldots, n$. Orthogonalize $V^{(m+1)}$.

8. Set $i = 0$, increment m, and go to step 2.

The variable i is the index of the direction being used in the line search.

Here $\hat{g}_v^{(m)} \approx \left(V^{(m)}\right)^T \nabla f(\hat{x}^{(m)})$ is the estimated gradient of $f(x+h^{(m)}V^{(m)}\eta)$ with respect to $h^{(m)}\eta$. At each grid local minimiser $\hat{x}^{(m)}$, the function value is known at each of the points $\hat{x}^{(m)} \pm h^{(m)}v_i^{(m)}$, $i = 1, \ldots, n$, and so central difference estimates along each $v_i^{(m)}$ directly yield each element of $\hat{g}_v^{(m)}$.

In step 7 $V^{(m)}$ is orthogonalized by post-multiplying it by an orthogonal matrix Q, where Q is chosen so that $Q^T \left(V^T V\right)^{(m)} Q$ is a diagonal matrix. Orthogonalizing $V^{(m)}$ in this way leaves the estimate $\left(VV^T\right)^{(m)}$ of the inverse Hessian unaltered.

3.1 Choosing the mesh size

Each time a new grid is selected in step 6, $h^{(m)}$ is divided by a factor s_r, and the scale down factor s_r is then updated via the following process: if the number of line searches on the previous grid is exceeds $4n + n^2/2$ then s_r is reduced according to the formula

$$s_r = \max\left(1 + [s_r - 1]/4, s_{\min}\right).$$

Otherwise, if the number of line searches on the previous grid is less than $2n$ then s_r is increased using the formula:

$$s_r = \min\left(1 + 2\left(s_r - 1\right), s_{\max}\right).$$

Here $s_{\max} \geq s_{\min} \geq 1$ is required. The values $s_{\min} = 1.01$ and $s_{\max} = 8$ were used to generate the numerical results presented herein. The reason for this adaptive strategy for reducing h is to allow grids to become fine quickly when grid local minima are being found quickly, but to avoid grids that are too fine. In the latter event, if the grid is poorly oriented then many line searches may be made before a grid local minimum is found, and until a grid local minimum is found there is only limited scope for re-orienting the grid. The ray search in step 2(e) is also used to speed up the location of a grid local minimum on each grid.

For the same reason, every time $n^2 + 8n$ consecutive line searches are executed without leaving step 2 the algorithm attempts to increase h at the end of step 2(d) according to the formula

$$h^{(m)} = \min\left(2h^{(m)}, h^{(m-1)}/s_{\min}\right).$$

The use of $h^{(0)} = \infty$ allows the algorithm to scale the initial grid up as much as is necessary to obtain a grid local minimum. These alterations are part of the finite process in step (ii) of the algorithm outline.

3.2 Generating the Set of Conjugate Directions

When f is a strictly convex quadratic, the searches along the directions in $\mathcal{V}_c^{(m)}$ allow the minimizer x_b of f over the manifold $\mathcal{M}$ to be calculated, where $\mathcal{M} = \{x_\mathrm{b} + V_c\zeta \ : \ \exists \zeta \in R^c\}$. Provided a non-zero step occurs in the following $n - c$ line searches along the directions in $\mathcal{V}_\mathrm{nc}^{(m)}$, the sequence of iterates is translated off $\mathcal{M}$. The next searches along the directions in $\mathcal{V}_c^{(m)}$ then allow the minimizer x_e of f on a manifold parallel to $\mathcal{M}$ to be calculated. The direction $x_\mathrm{e} - x_\mathrm{b}$ is conjugate to all members of $\mathcal{V}_c^{(m)}$ (see e.g. [7, 3, 8]).

Using $h^{(m)}V^{(m)}\eta_\mathrm{new} = x_\mathrm{e} - x_\mathrm{b}$, the new conjugate direction $x_\mathrm{e} - x_\mathrm{b}$ replaces the direction v_j in $\mathcal{V}_\mathrm{nc}^{(m)}$ for which the absolute value of the j^th component $(\eta_\mathrm{new})_j$ of η_new is maximal. The order of the remaining members of $\mathcal{V}_\mathrm{nc}^{(m)}$ is retained, the new conjugate direction is transferred from $\mathcal{V}_\mathrm{nc}^{(m)}$ to $\mathcal{V}_c^{(m)}$, and c is incremented.

If $(\eta_\mathrm{new})_j = 0$ for each $j = c + 1, \ldots, n$ then no displacement off the manifold $\mathcal{M}$ has occurred, in which case the updateis abandoned, and x_b is set to x_e. If the updateis successful, then x_b is reset to 'unknown.'

The ability to calculate the location of x_b stems from the fact that each line search provides function values at three or more points along the line in question. This allows the step to that line's exact minimizer to be calculated for a strictly convex quadratic, by minimizing the one dimensional quadratic interpolating the last three points at which f was calculated on the line. The form of the line search guarantees this interpolating quadratic is strictly convex except when all three interpolated function values are equal. In the latter case the middle interpolated point is taken as the line's minimiser. The contiguity of the searches along the members of $\mathcal{V}_c$, and conjugacy means that the sum of these steps to each line's minimiser is the step to the minimiser x_{b}.

It can be shown that each updateto V is via either by scaling of columns, or post-multiplication by a rank 1 matrix. Hence the determinant $|\det(V)|$ in condition (c) of theorem 2 can be updated from iteration to iteration.

3.3 Scaling the members of $\mathcal{V}_c$

At each grid local minimum, the directions in $\mathcal{V}_c^{(m)}$ are scaled to incorporate curvature information from the line searches along elements in $\mathcal{V}_c^{(m)}$. Let the estimate of the second derivative of f at $\hat{x}^{(m)}$ along the direction $v_i^{(m)}$ be $H_i^{(m)}$. Then

$$v_i^{(m+1)} = v_i^{(m)} \left[\max\left(\epsilon, H_i^{(m)}\right)\right]^{-\frac{1}{2}} \quad \forall i = 1, \ldots, c \tag{2}$$

so that the estimate of the second derivative of f at $\hat{x}^{(m)}$ along each new direction $v_i^{(m+1)}$ is 1, for $i = 1, \ldots, c$. Here ϵ is a small positive constant (10^{-8}) used to avoid divide by zero problems. Although the form of the line search means that $H_i < 0$ is impossible, $H_i = 0$ can occur when $f(x) = f(x + hv) = f(x - hv)$.

The scaling of $v_i^{(m+1)}$ in (2) may result in the violation of the bound $\|v\| \leq K$ in condition (c) of theorem 2, in which case $v_i^{(m+1)}$ is scaled so that $\|v_i^{(m+1)}\| = K$.

3.4 Stopping Conditions

The numerical results presented herein were generated using the simple test

$$\|\hat{g}_v^{(m)}\|_2 \leq \tau \tag{3}$$

where the stopping tolerance τ was set at 10^{-5}. The use of g_v in (3) is preferred because, given $VV^T \approx G_*^{-1}$,

$$\|\hat{g}_v\|_2^2 \approx g^T G_*^{-1} g \approx (\hat{x} - x_*)^T G_* (\hat{x} - x_*)$$

where the Taylor series approximation $g(x) = G_*(x - x_*)$ has been used, and where $G_* = \nabla^2 f(x^*)$. Clearly, (3) provides an estimate of the difference between the least known and optimal values of f.

More sophisticated tests [4] may be applied to the sequence of grid local minima, but the 'infrequent' nature of this sequence reduces the value of such tests.

4 Exact Termination on a Quadratic

It has been shown in theorem 2 that the subsequence of grid local minimizers converges to a stationary point. It is now shown that the algorithm possesses the property of finite termination on strictly convex quadratics.

Theorem 3 *Let f be a strictly convex quadratic of the form*

$$f(x) = \tfrac{1}{2}x^T Gx + a^T x \tag{4}$$

then the algorithm finds the exact minimiser x^* of $f(x)$ in a finite number of function evaluations.

Proof: First, it is shown that the algorithm generates a full set of conjugate directions unless it selects x^* as an iterate before this process is complete. Let $\mathcal{V}_c$ be the set of conjugate directions at the j^{th} iteration, where $x^{(j)}$ has been obtained from a search along v_c. Let $\mathcal{M} = \{x^{(j)} + V_c\zeta : \exists\zeta \in R^c\}$, and let $x_{\rm b}$ minimise f over $\mathcal{M}$. Although the searches along the members of $\mathcal{V}_c$ do not select $x_{\rm b}$ as an iterate, they do provide enough information to calculate $x_{\rm b}$ exactly when f is of the form (4). It is first shown that either (a) $x_{\rm b} = x^*$; or (b) a direction $v_{\rm new}$ conjugate to every member of $\mathcal{V}_c$ is generated.

To show (b) occurs it is sufficient to show that the algorithm performs a set of line searches along the directions in $\mathcal{V}_c$ from an iterate $x^{(k)} \notin \mathcal{M}$, for some $k > j$. Together with the parallel subspace theorem, the first such set of searches yields v_{new}.

If the algorithm takes a non-zero step along a direction in $\mathcal{V}_{\text{nc}}$, the linear independence of $\mathcal{V}$ ensures the subsequent set of searches along the directions in $\mathcal{V}_c$ are completed, and take place off $\mathcal{M}$. Otherwise the searches for $\mathcal{V}_{\text{nc}}$ make no movement, and conjugacy ensures that one set of searches along the directions in $\mathcal{V}_c$ will locate a grid local minimum. Steps 4 and 5 are then executed, ensuring the next iterate x satisfies $f(x) \leq f(x_{\text{b}})$. If this inequality is strict, then $x \notin \mathcal{M}$. Otherwise $x = x_{\text{b}}$, and the next n line searches will either return x_{b} as a grid local minimum or move to a lower iterate (necessarily not on $\mathcal{M}$). In the former case, the algorithm executes step 4 at x_{b}. If $x_{\text{b}} = x^*$, the solution has been found, otherwise $\nabla f(x_{\text{b}})$ is non-zero (because $x_{\text{b}} \neq x^*$), and orthogonal to $\mathcal{M}$. The use of central differences means that $\nabla f(x_{\text{b}})$ is known exactly. Now V is of full rank, and so $p = -VV^T\nabla f(x_{\text{b}})$ is a non-zero direction of descent. The line $\ell(\alpha) = x_{\text{b}} + \alpha p$, $\alpha \in R$ intersects $\mathcal{M}$ at x_{b} only. Step 4 of the algorithm looks at two points on the line. These are $x_{\text{b}} + p$ and $x_{\text{b}} + \alpha_p p$ where the latter is the minimizer of f over $\ell(\alpha)$. Now because p is a descent direction at x_{b} it follows that neither $x_{\text{b}} + p$ nor $x_{\text{b}} + \alpha_p p$ lie on $\mathcal{M}$. Hence step 4 moves the sequence of iterates off $\mathcal{M}$.

The above argument shows the algorithm either encounters x^*, or generates a full set of conjugate directions. In the latter case $g_v = V^T\nabla f$, and, when $c = n$, the inverse Hessian $(\nabla^2 f)^{-1} = VV^T$ because of the scaling in step 3. Hence $p = -Vg_v$ is the exact step to x^*, and step 4 of the algorithm ensures that this step will be taken. ♠

5 Numerical Results

The algorithm was tested on the first 19 test problems listed in [5]. The results for these problems are listed in Table 1, where '# fcn' denotes the number of function evaluations performed, and $f^\sharp$ is the function value at the final iterate. The second, starred, set of results for the helical valley problem use $h^{(1)} = 0.9$ rather than $h^{(1)} = 1$. With the latter choice the solution x^* is a grid local minimizer of the initial grid, and so the algorithm locates it artificially fast. The second set of results for Powell's badly scaled function (marked with a †) uses a stopping tolerance of $\tau = 10^{-8}$ rather than $\tau = 10^{-5}$, as this lower accuracy is achievable at points far from the solution. The results show that the algorithm is quite capable of achieving the higher accuracy of $\tau = 10^{-8}$.

The algorithm was also tested on a family of quadratics of the form

$$f(x) = (x - \mathbf{1})^T G_n (x - \mathbf{1})$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T$ and $x^{(0)} = \pi \left(1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n}\right)^T$. Here G_n is the $n \times n$ tridiagonal matrix with all diagonal elements equal to 2, and all super- and sub-diagonal elements equal to 1. Results are listed in Table 2, where the $\sharp$ superscript denotes the value of the quantity taken at the final iterate $x^\sharp$, and x^* is the solution.

The results show that the algorithm is effective on a wide variety of problems which includes ill-conditioned problems. The property of exact termination on strictly convex quadratics is verified by the numerical results. The stopping condition is satisfied when $f \approx 10^{-5}$, yet the final function values are many orders of magnitude smaller than this.

6 Conclusion

A provably convergent derivative free conjugate directions algorithm has been presented. Successive grids are chosen to incorporate known second derivative information generated by use of the parallel subspace theorem. Consequently the algorithm retains the property of exact termination on strictly convex quadratics. This property is verified by numerical results for the family of tridiagonal quadratics. Numerical results for general unconstrained problems show that the algorithm is effective in practice, even on problems which are ill-conditioned.

References

[1] A.R. Conn, K. Scheinberg, and P. Toint. On the convergence of derivative free methods for unconstrained optimization. In

Table 1: Numerical Results on 19 standard test functions.

Problem	n	# fcn	$f^\sharp$	$x^\sharp$
Rosenbrock	2	380	3.6e-11	(1.000006, 1.000012)
Freudenstein & Roth	2	75	48.98	(11.42, -0.8968)
Powell badly scaled	2	734	1.9e-7	(1.3e-5, 7.520)
Powell badly scaled†	2	1784	6.7e-18	(1.1e-5, 9.106)
Brown badly scaled	2	58	1.4e-20	(10^6, 2.0e-6)
Beale	2	87	5.6e-13	(3.000002, 0.5000005)
Jennrich & Sampson	2	154	124.4	(0.2578, 0.2578)
Helical valley	3	11	0	(1, 0, 0)
Helical valley∗	3	303	4.2e-11	(1, -1.2e-7, -7.1e-7)
Bard	3	200	17.43	(0.8407, 8.0e+13, 1)
Gaussian	3	47	1.1e-8	(0.398956, 1.00002, 0)
Meyer	3	9070	87.95	(0.0056, 6181, 345.2)
Gulf Research	3	655	1.8e-13	(50, 25, 1.5)
Box 3-dimensional	3	227	0.01409	(-1.071, 436.4, 1.944)
Powell singular	4	242	2.6e-11	(0.0013, -0.00013, 0.0008, 0.0008)
Wood	4	315	4.9e-12	(1, 1, 1, 1)
Kowalik and Osborne	4	317	3.1e-4	(0.1928, 0.1913, 0.1231, 0.1361)
Brown and Dennis	4	232	85822	(-11.59, 13.2, -0.4034, 0.2368)
Osborne 1	5	1413	5.5e-5	(0.375, 1.94, -1.46, 0.0129, 0.0221)
Biggs exp6	6	3403	1.9e-11	(3.999, 10, 2.999, 4.999, 0.9998, 0.9997)
Osborne 2	11	2341	0.04014	(1.31, 0.432, 0.634, 0.600, 0.754, 0.904, 1.37, 4.82, 2.40, 4.57, 5.67)

Table 2: Numerical Results on a family of quadratics.

n	# fcn	$f^\sharp$	$\|g_v^\sharp\|$	$\|x^\sharp - x^*\|$
2	19	0	0	0
4	67	2.5e-32	2.0e-16	1.0e-16
6	121	1.2e-31	5.8e-16	7.3e-16
8	235	2.8e-30	2.4e-15	2.1e-15
10	353	1.7e-30	1.9e-15	1.4e-15
20	1156	1.4e-20	1.7e-10	8.7e-11
30	2317	2.4e-20	2.2e-20	3.0e-10

M.D. Buhmann and A. Iserles, editors, *Approximation Theory and Optimization*, pages 83–108, Cambridge, 1997, Cambridge University Press.

[2] I.D. Coope and C.J. Price. On the convergence of grid based methods for unconstrained optimization. *Research Report 180*, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.

[3] R. Fletcher. *Practical Methods of Optimization*. Wiley, 1987.

[4] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, 1981.

[5] J.J. Moré, B.S. Garbow, and K.E. Hillstrom. Testing unconstrained optimization software. *ACM Trans. Math. Software*, 7:17–41, 1981.

[6] M.J.D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, 1998.

[7] M.J.D. Powell. An efficient method of finding the minimum of a function of several variables without calculating derivatives. *Computer J.*, 7:155–162, 1964.

[8] C.S. Smith. The automatic computation of maximum likelihood estimates. *N. C. B. Sci. Dept. Report*, SC846/MR/40, 1962.

[9] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optimization*, 7:1–25, 1997. C480 C480, C481, C482, C483 C484, C489 C491 C494 C480 C484, C489 C484, C489 C480