

Updating quadratic models in optimization calculations
without derivatives

Michael J.D. Powell

Berlin, July 20, 2006

The problem

①

Seek the least value of $F(\underline{x})$,
 $\underline{x} \in \mathbb{R}^n$, when $F(\underline{x})$ can be calculated
for any \underline{x} , but no derivatives of
 F are available.

There are no constraints on the
variables, except that later some
new work is described for simple
bounds $l_i \leq x_i \leq u_i$, $i=1,2,\dots,n$,
with $l_i < u_i$, $i=1,2,\dots,n$.

I have developed software in
Fortran that is usually adequate
for up to 100 variables.

The points where $F(\underline{x})$ is calculated
are kept apart, in order to restrict
damage from noise in the function
values.

Methods that use quadratic models ②

These methods are iterative. Usually the k -th iteration requires a starting point $\underline{x}_k \in \mathbb{R}^n$, a "trust region radius" $\Delta_k > 0$ and a quadratic function $Q_k(\underline{x}), \underline{x} \in \mathbb{R}^n$, that is an approximation to $F(\underline{x}), \underline{x} \in \mathbb{R}^n$.

Here \underline{x}_k is such that $F(\underline{x}_k)$ is the least calculated value of F so far, and Q_k satisfies $Q_k(\underline{x}_k) = F(\underline{x}_k)$.

Typically \underline{x}_{k+1} is chosen by seeking the least value of $Q_k(\underline{x})$ in the region $\{\underline{x} : \|\underline{x} - \underline{x}_k\| \leq \Delta_k\}$.

Let $\underline{x}_k + \underline{d}_k$ be the trial value of \underline{x}_{k+1} that is given by this trust region subproblem.

On the choice of $Q_k(\underline{x}), \underline{x} \in \mathbb{R}^n$ ③

Each quadratic function Q_k interpolates m values of $F(\underline{x}), \underline{x} \in \mathbb{R}^n$, where m is a prescribed integer from the interval $[n+2, \frac{1}{2}(n+1)(n+2)]$. Let these interpolation conditions be

$$Q_k(\underline{y}_j) = F(\underline{y}_j), \quad j=1, 2, \dots, m,$$

so \underline{x}_k is one of the points \underline{y}_j .

We require $m \geq n+2$, in order that the interpolation conditions provide some information about any second derivatives of F .

The upper bound $\frac{1}{2}(n+1)(n+2)$ on m is the number of degrees of freedom in the parameters of $Q_k(\underline{x}), \underline{x} \in \mathbb{R}^n$.

④

On the choice of m

I picked $m = \frac{1}{2}(n+1)(n+2)$ in my early work, and then each Q_k is defined by an $m \times m$ linear system, but $n \geq 50$ implies $m \geq 1326$.

Therefore now I much prefer $m = 2n + 1$, and I take up the freedom in Q_k by minimizing $\|\nabla^2 Q_k\|_F$ or $\|\nabla^2 Q_k - \nabla^2 Q_{k-1}\|_F$ in the cases $k = 1$ or $k \geq 2$, respectively.

If $F(x)$, $x \in \mathbb{R}^n$ is quadratic, this method for $k \geq 2$ provides the highly valuable property

$$\begin{aligned} \|\nabla^2 Q_k - \nabla^2 F\|_F^2 &= \|\nabla^2 Q_{k-1} - \nabla^2 F\|_F^2 \\ &\quad - \|\nabla^2 Q_k - \nabla^2 Q_{k-1}\|_F^2. \end{aligned}$$

⑤
On the interpolation points y_j , $j=1,2,\dots,m$

- They are kept apart (by an amount that is reduced occasionally), in order to help the accuracy of the approximation $Q_k \approx F$ when F is noisy.
- The possibility of degeneracies and inconsistencies in the interpolation conditions must be avoided.
- The condition $Q_k(\underline{y}_j) = F(\underline{y}_j)$ may be unhelpful when $\|\underline{y}_j - \underline{x}_k\|$ is large.
- When the set $\{\underline{y}_j : j=1,2,\dots,m\}$ is updated, we alter the position of only one of the interpolation points.

doc05.jp2

After the trust region subproblem ⑥

Let \underline{d}_k be found by seeking the least value of $Q_k(\underline{x}_k + \underline{d})$ subject to $\|\underline{d}\| \leq \Delta_k$.

Unless $\|\underline{d}_k\|$ is unacceptably small, $F(\underline{x}_k + \underline{d}_k)$ is calculated, and $\underline{x}_k + \underline{d}_k$ replaces one of the old interpolation points. The choice of the rejected point assists the construction of Q_{k+1} .

Further, we apply the formula

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

If $F(\underline{x}_k + \underline{d}_k)$ compares unfavourably with $F(\underline{x}_k)$, or if $\|\underline{d}_k\|$ was unacceptably small, then the next trial step \underline{d} is generated by the following "alternative subproblem".

⑦

The alternative subproblem for d_k

Let y_ϵ be the point in the set $\{y_j : j=1,2,\dots,m\}$ that is furthest from x_k , and let $\Lambda_\epsilon(x)$, $x \in \mathbb{R}^n$, be the quadratic polynomial that satisfies the Lagrange conditions

$$\Lambda_\epsilon(y_j) = \delta_{j\epsilon}, \quad j=1,2,\dots,m,$$

any freedom being taken up by minimizing $\|\nabla^2 \Lambda_\epsilon\|_F$.

The alternative choice of d_k should maximize $|\Lambda_\epsilon(x_k + d)|$ subject to $\|d\| \leq \Delta_k$. Then it is highly suitable to replace y_ϵ by the point $x_k + d_k$.

If tests suggest that $Q_k \approx F$ is adequate, however, then termination or a reduction in a lower bound on Δ_k is preferred instead.

doc07.jp2

The NEWUOA software

⑧

This Fortran software applies the given algorithm. I started to develop it in January, 2002, but did not complete it until December, 2003, because severe inefficiencies were caused occasionally by rounding errors.

The amount of routine work per iteration is only about $O(m^2)$ due to the use of updating techniques. The rounding error difficulties were solved by a matrix factorization that preserves exactly the low rank of a submatrix that occurs.

Some functions of 160 variables have been minimized to high accuracy using fewer than 10,000 values of F , although $\frac{1}{2} \times 161 \times 162 = 13,041$.

doc08.jp2

Bounds on the variables

⑨

Can a modification of **NEWUOA** allow the bounds $l_i \leq x_i \leq u_i$, $i=1,2,\dots,n$, on the variables, where $l_i < u_i$.

I replaced the subroutines for the "trust region" and "alternative" subproblems by ones that minimize $Q_k(\underline{x}_k + \underline{d})$ and maximize $|\Omega_k(\underline{x}_k + \underline{d})|$ again, but now the constraints on \underline{d} are $\|\underline{d}\| \leq \Delta_k$ and $l_i \leq x_k + d_i \leq u_i$.

I hoped that no changes would be required to the techniques of **NEWUOA** that perform updating, that keep the interpolation points apart, and that avoid tendencies towards degeneracies.

I investigated these questions experimentally.

10

A test function for BOBYQA

BOBYQA (Bound Optimization BY Quadratic Approximation) is my name for this experimental software.

Let n be even, and, for each $\underline{x} \in \mathbb{R}^n$, we define the points

$$\underline{p}_k = \begin{pmatrix} x_{2k-1} \\ x_{2k} \end{pmatrix} \in \mathbb{R}^2, \quad k=1, 2, \dots, n/2.$$

Further, we impose the bounds $0 \leq x_i \leq 1$, $i=1, 2, \dots, n$, and we employ the objective function

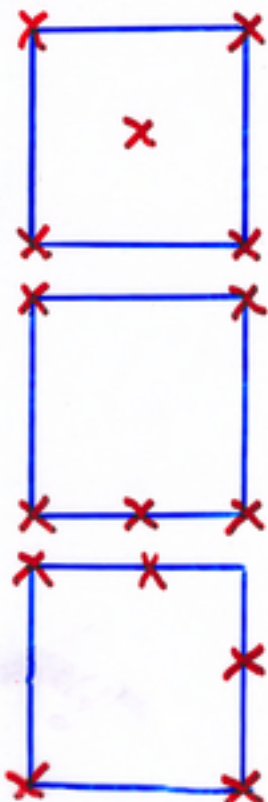
$$F(\underline{x}) = \sum_{k=2}^{n/2} \sum_{j=1}^{k-1} \frac{1}{\|\underline{p}_j - \underline{p}_k\|}, \quad \underline{x} \in \mathbb{R}^n.$$

Thus we try to place $n/2$ points on the unit square in a way that minimizes the sum of reciprocals of distances between pairs of points.

(11)

Local minima when $n=10$

For $n=10$, the initial vector of variables for **BOBYQA** was picked at random from the uniform distribution on $[0,1]^{10} \subset \mathbb{R}^{10}$. We tried 100 different starting points, and the final positions of $P_k \in \mathbb{R}^2$, $k=1,2,3,4,5$, had the following forms:



34 times,
Final F is 11.0711.

53 times
Final F is 11.2031.

13 times
Final F is 11.3607.

doc11.jp2

(12)

Larger values of n

For $n \geq 20$, the initial x was picked at random from $[0, 1]^n$, using recursion if necessary, until all the initial points $P_k \in \mathbb{R}^2$ satisfied

$$\|P_k - P_j\| \geq 0.2/\sqrt{n/2}, \quad k \neq j.$$

We tried 6 values of n and 5 different starting points for each one. The range of #F (number of calculations of F) for each n is as follows, local minima being found in all cases.

| n | Range of #F |
|-----|-----------------|
| 20 | 819 - 1,247 |
| 40 | 4,193 - 11,023 |
| 60 | 7,403 - 28,237 |
| 80 | 14,436 - 44,526 |
| 100 | 19,638 - 52,075 |
| 120 | 26,374 - 77,565 |

(13)

Further information

A description of the method of **NEWUDA** is given in "The **NEWUDA** software for unconstrained optimization without derivatives" by **M.J.D. Powell** in *Large-Scale Nonlinear Optimization*, editors **G. Di Pillo** and **M. Roma**, Springer Publishing Co (2006).

Please e-mail me at **mjdp@cam.ac.uk** if you would like to receive the **Fortran** listing of **NEWUDA**.

I intend to do much more work on the proposed technique for bounds on the variables before writing a paper on this research.

doc13.jp2