The modified absolute-value factorization norm for trust-region minimization *

Nicholas I. M. Gould
Department for Computation and Information, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England.

Jorge Nocedal
Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208-3118, USA

* The work of the second author was supported by a National Science Foundation grant CCR-9625613 and by a Department of Energy grant DE-FG02-87ER25047-A004.

Abstract

A trust-region method for unconstrained minimization, using a trust-region norm based upon a modified absolute-value factorization of the model Hessian, is proposed. It is shown that the resulting trust-region subproblem may be solved using a single factorization. In the convex case, the method reduces to a backtracking Newton linesearch procedure. The resulting software package is available as `HSL_VF06` within the Harwell Subroutine Library. Numerical evidence shows that the approach is effective in the nonconvex case.

1. Introduction

In this paper, we are concerned with trust-region methods for the unconstrained minimization of a function $f(x)$ of n real variables x. At the k-th iteration of such a method, a *model* $q_k(s)$ of $f(x_k + s)$ is approximately minimized within a *trust region* $\|s\| \leq \Delta_k$ with the aim of improving upon the current estimate of the minimizer x_k. The approximate solution s_k of this trust-region subproblem yields the improved estimate $x_{k+1} = x_k + s_k$ if the reduction in f predicted by this model translates into an significant actual reduction of $f(x_k + s_k)$. If such a reduction is not realized, the trust-region *radius* Δ_k is reduced, and the model resolved. If there is a good agreement between model and function, the radius may be increased. The method is blessed with a powerful convergence theory regardless of which norm defines the trust region, provided that the chosen norm is is uniformly related to the ℓ_2-norm. Little attention has been given to the appropriate choice of norm considering how strongly this choice affects the computation at every iteration of the algorithm. In this paper, we suggest that there is a particular norm which has computational advantages over the ℓ_2- or ℓ_∞-norms which are commonly considered.

2. The subproblem

We consider the quadratic model

$$q(s) = \langle g, s\rangle + \tfrac{1}{2}\langle s, Hs\rangle, \tag{2.1}$$

where g and H are approximations of the gradient and Hessian of $f(x)$, and $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product — for brevity, we have dropped the dependence of these quantities on k. We shall be concerned with elliptical trust regions of the form

$$\|s\|_N \leq \Delta, \quad \text{where} \quad \|s\|_N^2 = \langle s, Ns\rangle \tag{2.2}$$

and N is a real symmetric positive-definite matrix. A global solution to the trust-region subproblem is characterized by the following result.

THEOREM 2.1. ([12], [29]) Any global minimizer s_* of $q(s)$ subject to $\|s\|_N \leq \Delta$ satisfies the equation

$$H(\lambda_*)s_* = -g, \tag{2.3}$$

where $H(\lambda_*) \equiv H + \lambda N$ is positive semi-definite, $\lambda_* \geq 0$ and $\lambda_*(\|s_*\|_N - \Delta) = 0$. If $H(\lambda_*)$ is positive definite, s_* is unique.

2.1. Notation

Suppose that G is any real symmetric, possibly indefinite, matrix, and that G has a spectral decomposition

$$G = U\Lambda U^T,$$

where Λ is a diagonal matrix of eigenvalues, and U is an orthonormal matrix whose columns are the corresponding eigenvectors. Then we say that the *absolute value* of G is the matrix

$$|G| = U|\Lambda|U^T,$$

where $|\Lambda|$ is the diagonal matrix of absolute values of the eigenvalues of G.

We shall denote the (appropriately dimensioned) identity matrix by I. The square root $D^{\frac{1}{2}}$ of a diagonal matrix D is simply the diagonal matrix whose entries are $\sqrt{d_{ii}}$, while the generalized inverse D^+ is the diagonal matrix whose entries are $1/d_{ii}$ if $d_{ii} \neq 0$ and 0 if $d_{ii} = 0$.

3. The trust-region norm

We suppose, for now, that H is nonsingular. We will relax this assumption in Section 3.4.

3.1. The spectral trust region

We believe that the shape of an ideal trust region should reflect the geometry of the model, and not give undeserved weight to certain directions. Indeed, perhaps the ideal trust region would be in the $|H|$ norm, for which

$$\|s\|^2_{|H|} = \langle s, |H|s\rangle. \tag{3.1}$$

This norm reflects the scaling of the underlying problem — directions for which the model is changing fastest, and thus those for which the model may differ most from the true function, are restricted more than those directions for which the curvature is small. It has a further interesting property, namely, that a single matrix factorization

$$H = U\Lambda U^T \tag{3.2}$$

is needed to solve the problem. For, on writing

$$s_{\rm D} = U^T s \quad \text{and} \quad g_{\rm D} = U^T g,$$

and using the orthonormality of U, the solution of the trust-region subproblem may be expressed as $s = Us_{\rm D}$, where $s_{\rm D}$ solves the *diagonal* trust-region subproblem

$$\underset{s_{\rm D}\in\mathbb{R}^n}{\text{minimize}} \ \langle g_{\rm D}, s_{\rm D}\rangle + \tfrac{1}{2}\langle s_{\rm D}, \Lambda s_{\rm D}\rangle \quad \text{subject to} \quad \langle s_{\rm D}, |\Lambda| s_{\rm D}\rangle \leq \Delta^2. \tag{3.3}$$

The diagonal trust-region subproblem is, as we shall see, extremely inexpensive to solve. The major drawback of such an approach is, of course, the cost of the spectral factorization (3.2). For problems involving a large number of variables, this decomposition is likely out of the question.

The absolute-value factorization was originally proposed by Greenstadt [18] in conjunction with linesearch methods for unconstrained minimization.

3.2. The absolute-value trust region

With this in mind, we consider a symmetric, indefinite factorization of the form

$$H = PLBL^TP^T, \tag{3.4}$$

where P is a permutation matrix, L unit lower triangular and B block diagonal, with blocks of size at most two. We shall refer to the blocks as 1 by 1 and 2 by 2 pivots. Notice that the inertia of H — the numbers of positive, negative and zero eigenvalues of H — is trivially obtained by summing the inertia of the pivots. Such a factorization was first proposed by Bunch and Parlett [4] and later improved by Bunch and Kaufman [3] and Fletcher [11] in the dense case and Duff, Reid and co-workers [10], [8] in the sparse case. More recently, Ashcraft, Grimes and Lewis [1] and Higham [22] have exposed a potentially serious flaw in the approach in that the norm of the generated factor L may be unbounded relative to $\|H\|$. While, as Higham [22] has shown that this does not always lead to instability, a more restricted form of pivoting, as typified by the proposal in [1], may be required to ensure that $\|L\|$ stays bounded. Interestingly, the sparse method proposed by Duff and Reid [8] and implemented within the Harwell Subroutine Library [19] code `MA27` already provided a suitably bounded $\|L\|$ and will be suitable for our purposes.

We suggest that a good choice for the trust-region norm is

$$\|s\|_M^2 = \langle s, Ms\rangle, \tag{3.5}$$

where

$$M = PL|B|L^TP^T. \tag{3.6}$$

Observe that $|B|$ is simply computed by taking the absolute values of the 1 by 1 pivots, and by forming an independent spectral decomposition of each of the 2 by 2 pivots and reversing the signs of any resulting negative eigenvalues. By analogy with the Spectral method, writing

$$s_{\rm B} = L^TP^Ts \quad \text{and} \quad g_{\rm B} = L^{-1}P^Tg, \tag{3.7}$$

the solution of the trust-region subproblem may be expressed as $s = PL^{-T}s_{\rm B}$, where $s_{\rm B}$ solves the *block-diagonal* trust-region subproblem

$$\underset{s_{\rm B}\in\mathbb{R}^n}{\text{minimize}} \ \langle g_{\rm B}, s_{\rm B}\rangle + \tfrac{1}{2}\langle s_{\rm B}, Bs_{\rm B}\rangle \quad \text{subject to} \quad \langle s_{\rm B}, |B| s_{\rm B}\rangle \leq \Delta^2. \tag{3.8}$$

Once again, a single factorization suffices, but this time the factorization may be affordable even when n is large. Note that Gill, Murray, Ponceléon and Saunders [14] proposed this modified factorization as a preconditioner for iterative methods, while Chen and Higham [5] suggest it as an alternative to the modified Cholesky factorizations [13], [15], [28] used within linesearch-based methods.

We note, in passing, that others have used the factorization (3.4) to define trust-region norms. Goldfarb [16] suggests using (3.5), but where (3.6) is replaced by

$$M = PLL^T P^T. \tag{3.9}$$

Following the change of variables (3.7), the resulting block-diagonal trust-region is then of the form

$$\underset{s_{\rm B}\in\mathbb{R}^n}{\text{minimize}} \ \langle g_{\rm B}, s_{\rm B}\rangle + \tfrac{1}{2}\langle s_{\rm B}, Bs_{\rm B}\rangle \quad \text{subject to} \quad \|s_{\rm B}\| \leq \Delta$$

and its solution is again straightforward to obtain. This idea has recently been further explored by Xu and Zhang [31]. However, we believe that using (3.9) rather than (3.6) does not reflect the proper scaling of the underlying problem. Indeed, if H were a diagonal matrix, (3.5) remains as the ℓ_2 norm regardless of how ill-conditioned H might be.

3.3. Solving the diagonal and block-diagonal trust region subproblems

As the diagonal trust-region subproblem is a special (but not very special) case of the block-diagonal case, here we shall concentrate on the latter. One could simply apply a standard trust-region solver like `GQTPAR` [24] to (3.8), but we prefer not to do this as this would, to some extent, ignore the structure in hand.

As B and $|B|$ share eigenvectors, we may write

$$B = Q\Theta Q^T \quad \text{and} \quad |B| = Q|\Theta|Q^T,$$

where each column of Q is nonzero in at most two positions, with entries corresponding to the eigenvectors of the diagonal blocks, and the entries of the diagonal matrix Θ are the corresponding eigenvalues. On defining

$$s_{\rm S} = |\Theta|^{\frac{1}{2}} Q^T s_{\rm B} \quad \text{and} \quad g_{\rm S} = |\Theta|^{-\frac{1}{2}} Q^T g_{\rm B},$$

we may solve (3.8) by finding $s_{\rm S}$ to

$$\underset{s_{\rm S}\in\mathbb{R}^n}{\text{minimize}} \ \langle g_{\rm S}, s_{\rm S}\rangle + \tfrac{1}{2}\langle s_{\rm S}, Ds_{\rm S}\rangle \quad \text{subject to} \quad \|s_{\rm S}\|_2 \leq \Delta, \tag{3.10}$$

and then recover $s_{\rm B} = Q|\Theta|^{-\frac{1}{2}} s_{\rm S}$. Significantly, the matrix $D \equiv |\Theta|^{-\frac{1}{2}}\Theta|\Theta|^{-\frac{1}{2}}$ is diagonal with entries ± 1. The required solution must then satisfy

$$(D + \lambda I)s_{\rm S} = -g_{\rm S}, \tag{3.11}$$

where the nonnegative Lagrange multiplier λ is sufficiently large to ensure that $D + \lambda I$ is positive semi-definite, and is zero if $s_{\rm S}$ lies within the trust region $\|s_{\rm S}\|_2 \leq \Delta$.

There are two cases to consider. Firstly, if $D = I$, the solution to (3.11) is

$$s_{\rm S} = -\frac{1}{1+\lambda} g_{\rm S}.$$

If $\|g_{\rm S}\|_2 < \Delta$, the solution to (3.10) is given by $s_{\rm S} = -g_{\rm B}$ and $\lambda = 0$. This corresponds to the unconstrained minimizer of the model lying interior to the trust region. If, on the other hand, $\|g_{\rm S}\|_2 \geq \Delta$, the solution to (3.10) is obtained by finding the value of $\lambda \geq 0$ for which

$$\frac{1}{(1+\lambda)} \|g_{\rm S}\|_2 = \Delta.$$

This is a linear equation in λ and thus the solution is trivial to obtain; the required $s_{\rm S}$ is

$$s_{\rm S} = -\frac{\Delta}{\|g_{\rm S}\|_2} g_{\rm S}.$$

This corresponds to the case where the model is convex, but the trust region excludes the unconstrained minimizer of the model. Notice, also, in this case, a reduction in the trust region radius following an unsuccessful step merely reduces the length of the step in the direction $-g_{\rm B}$. Such a strategy is identical in its effect (if not in its motivation) to a backtracking linesearch along the quasi-Newton direction $-H^{-1}g$, and thus there is a strong similarity between trust-region and linesearch methods with this choice of trust region.

Secondly, if H has negative eigenvalues, D will have some diagonal entries of -1. Suppose $P_{\rm S}$ is a permutation matrix which arranges that all the positive diagonals $(+1)$ of D precede its negative diagonals (-1). Then it is easy to show that

$$s_{\rm S} = -\frac{1}{\lambda^2 - 1} P_{\rm S}^T \begin{pmatrix} (\lambda-1)I & 0 \\ 0 & (\lambda+1)I \end{pmatrix} P_{\rm S} g_{\rm S}. \tag{3.12}$$

As H is indefinite, the solution must lie on the trust-region boundary. Thus, we may obtain λ as the root larger than 1 of the quartic equation

$$\langle P_{\rm S} g_{\rm S}, \begin{pmatrix} (\lambda-1)^2 I & 0 \\ 0 & (\lambda+1)^2 I \end{pmatrix} P_{\rm S} g_{\rm S} \rangle = (\lambda^2 - 1)^2 \Delta^2.$$

Although in principle this root may be found explicitly by Ferrari's method (see, for instance, [30], [26]), Newton's method is equally suitable here. A slight complication may occur when all of the components of $P_{\rm S} g_{\rm S}$ corresponding to the negative diagonals of D are zero. For then (3.12) yields

$$s_{\rm S} = -\frac{1}{\lambda+1} P_{\rm S}^T \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} P_{\rm S} g_{\rm S},$$

and it may be that there is no root larger than 1 of the resulting feasibility equation

$$\langle P_{\rm S} g_{\rm S}, \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} P_{\rm S} g_{\rm S} \rangle = (\lambda+1)^2 \Delta^2.$$

This case corresponds to the "hard" case in [24], and here, as there, the solution includes a contribution from a suitable eigenvector. In our case, it is of the form

$$s_{\rm S}(\alpha) = -\tfrac{1}{2} P_{\rm S}^T \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} P_{\rm S} g_{\rm S} + \alpha P_{\rm S}^T \begin{pmatrix} 0 \\ u \end{pmatrix},$$

where u is any nonzero vector, and α is chosen as a root of the quadratic equation $\langle s_{\rm S}(\alpha), s_{\rm S}(\alpha)\rangle = \Delta^2$.

3.4. Coping with singularity

Clearly, it is important to deal with any matrix H including those which are, or are close to being, singular. Chen and Higham [5] suggest that it suffices to compute the factorization (3.6) and to replace each eigenvalue θ of the block diagonal B with the value

$$\gamma = \begin{cases} \theta & \text{if} \quad \theta \geq \delta \quad \text{or} \\ \delta & \text{otherwise} \end{cases} \tag{3.13}$$

for some small $\delta > 0$. An alternative, which is closer in spirit both to the absolute value perturbation and to Greenstadt [18]'s perturbation, is to replace each eigenvalue by

$$\gamma = \begin{cases} \theta & \text{if} \quad \theta \geq \delta \quad \text{or} \\ -\theta & \text{if} \quad \theta \leq -\delta \quad \text{or} \\ \delta & \text{otherwise.} \end{cases} \tag{3.14}$$

In any event, this does not significantly affect our previous discussion. For, if we let C denote the (possibly) modified block diagonal matrix B, we now use the trust-region norm (3.5) with M defined as

$$M = PLCL^T P^T. \tag{3.15}$$

We shall refer to (3.15) as the *modified absolute-value* factorization. If we make the change of variables (3.7), we must solve the block-diagonal trust-region subproblem

$$\underset{s_{\rm B}\in\mathbb{R}^n}{\text{minimize}} \ \langle g_{\rm B}, s_{\rm B}\rangle + \tfrac{1}{2}\langle s_{\rm B}, Bs_{\rm B}\rangle \quad \text{subject to} \quad \langle s_{\rm B}, Cs_{\rm B}\rangle \leq \Delta^2. \tag{3.16}$$

It is of little consequence that BC^{-1} no longer necessarily has eigenvalues ± 1, for, as we shall now see, solving the problem (3.16) is also straightforward.

As before, B and C share eigenvectors. We may thus write

$$B = Q\Theta Q^T \quad \text{and} \quad C = Q\Gamma Q^T,$$

where Q is as before, and the entries of the diagonal matrices Θ and Γ are, respectively, the values θ and γ considered in (3.13) or (3.14). Using the transformation

$$s_{\rm S} = \Gamma^{\frac{1}{2}} Q^T s_{\rm B} \quad \text{and} \quad g_{\rm S} = \Gamma^{-\frac{1}{2}} Q^T g_{\rm B},$$

we may recover the solution to (3.16) from $s_{\rm B} = Q\Gamma^{-\frac{1}{2}} s_{\rm S}$, where $s_{\rm S}$ is found to

$$\underset{s_{\rm S}\in\mathbb{R}^n}{\text{minimize}} \ q_{\rm S}(s_{\rm S}) \equiv \langle g_{\rm S}, s_{\rm S}\rangle + \tfrac{1}{2}\langle s_{\rm S}, Ds_{\rm S}\rangle \quad \text{subject to} \quad \|s_{\rm S}\|_2 \leq \Delta, \tag{3.17}$$

and where $D \equiv \Gamma^{-\frac{1}{2}}\Theta\Gamma^{-\frac{1}{2}}$ is diagonal. Once again, one could simply apply the Moré–Sorensen [24] algorithm to this problem, but this ignores the facts that the diagonal systems involved are trivial to solve, and that the leftmost eigenvalue of D and a corresponding eigenvector are trivial to obtain. We therefore prefer the following simplification.

If D merely has entries ± 1, the procedure outlined in Section 3.3 is appropriate. So, now suppose that D has a more complicated distribution of values. Then we may apply Algorithm 3.1.

Algorithm 3.1: Newton iteration to solve (3.17)

Let $\epsilon \in (0,1)$.

1. If D is positive definite, set $\lambda = 0$ and $s_{\rm S} = -D^{-1} g_{\rm S}$.

   1a. If $\|s_{\rm S}\|_2 \leq \Delta$, stop.

2. Otherwise, compute the leftmost eigenvalue, θ of D, set $\lambda = -\theta$ and define $g^{\rm n}_{\rm S}$ so that

$$(g^{\rm n}_{\rm S})_i = \begin{cases} (g_{\rm S})_i & \text{if} \quad (D)_{ii} + \lambda = 0 \\ 0 & \text{otherwise.} \end{cases}$$

   2a. If $g^{\rm n}_{\rm S} = 0$, set $s_{\rm S} = -(D + \lambda I)^{+} g_{\rm S}$.

      i. If $\|s_{\rm S}\|_2 \leq \Delta$, compute an eigenvector u corresponding to θ, find the root α of the equation $\|s_{\rm S} + \alpha u\|_2 = \Delta$ which makes the model $q_{\rm S}(s_{\rm S} + \alpha u)$ smallest, replace $s_{\rm S}$ by $s_{\rm S} + \alpha u$, and stop.

   2b. Otherwise, replace λ by $\lambda + \|g^{\rm n}_{\rm S}\|_2 / \Delta$, and set $s_{\rm S} = -(D + \lambda I)^{-1} g_{\rm S}$.

3. If

$$|\|s_{\rm S}\|_2 - \Delta| \leq \epsilon \Delta,$$

   stop.

4. Replace λ by $\lambda + \left(\dfrac{\|s_{\rm S}\|_2 - \Delta}{\Delta} \right) \left(\dfrac{\|s_{\rm S}\|_2^2}{\langle s_{\rm S}, (D + \lambda I)^{+} s_{\rm S} \rangle} \right)$.

5. Set $s_{\rm S} = -(D + \lambda I)^{+} g_{\rm S}$ and go to step 3.

Figure 3.0.

The iteration in Steps 3 to 5 is simply Newton's method to find the appropriate root of the secular equation

$$\frac{1}{\| - (D + \lambda I)^{+} g_{\rm S}\|_2} = \frac{1}{\Delta}$$

(see [21], [24], for details). Step 1 caters for the case where the model is strictly convex, while step 2 is for the more general case where the solution must lie on the trust-region boundary. The precaution in Step 1a is simply to detect the solution when it lies interior to the trust region, while that in Step 2a(i) is to compute the solution in the "hard case" in [24]. The iteration is globally linearly and asymptotically quadratically convergent from the starting values given in Steps 1 and 2. The tolerance ϵ should be set at the level of the machine precision, ϵ_M. We stress that, while this algorithm is appropriate even if D is simply a diagonal matrix with entries ± 1, the procedure outlined in Section 3.3 is more appropriate in this case.

3.5. The suitability of the norm

It remains for us to show that the norms defined by the modified absolute-value factorization (3.15) are uniformly related to the ℓ_2-norm, and thus are suitable within a trust-region method. Thus we need to show that there are constants $0 < \gamma_1 < \gamma_2$, independent of the iteration, for which

$$\gamma_1 \|s\|_2^2 \leq \langle s, Ms \rangle \leq \gamma_2 \|s\|_2^2.$$

Equivalently, we need to show that the smallest and largest eigenvalues, $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$, of M are bounded, and bounded away from zero. The analysis here is based upon that given by Higham [22].

Firstly, by construction, both of (3.13) and (3.14) satisfy the bounds

$$\delta \leq \|C\| = \max(\delta, \|D\|) \leq \max\left(\delta, \|H\| \|(LL^T)^{-1}\|\right).$$

Using Theorem 3.2 in [22], it then follows that

$$\lambda_{\min}(LL^T)\lambda_{\min}(C) \leq \lambda_{\min}(M) \leq \lambda_{\max}(M) \leq \lambda_{\max}(LL^T)\lambda_{\max}(C),$$

and hence that

$$\delta\lambda_{\min}(LL^T) \leq \lambda_{\min}(M) \leq \lambda_{\max}(M) \leq \lambda_{\max}(LL^T) \max\left(\delta, \|D\| \leq \|H\| \|(LL^T)^{-1}\|\right).$$

But, as Higham then points out, if the largest entry in L is bounded by some β, it is straightforward to bound

$$1 \leq \lambda_{\max}(LL^T) \leq n + \tfrac{1}{2}n(n-1)\beta^2 \quad \text{and} \quad (1+\beta)^{2-2n} \leq \lambda_{\min}(LL^T) \leq 1.$$

Thus so long as L and H are bounded, the norms defined by the modified absolute-value factorization (3.15) are uniformly related to the ℓ_2-norm.

The matrix H will be bounded if, for instance, a Newton (second-order Taylor series) model is used, and if the iterates stay in a bounded set. But now we see the importance of using a factorization which bounds the growth in the elements of L. Ashcraft, Grimes and Lewis [1] show that the original method of Bunch and Parlett [4] and that of Fletcher [11] both generate bounded L, as do the sparse methods of Duff and Reid [8], [9]). However, the more popular Bunch–Kaufman [3] method and the block version implemented in `LAPACK` may not, and thus must be viewed as untrustworthy for our application.

4. Numerical experiments

The algorithm sketched in Sections 3.3 and 3.4 has been implemented as a Fortran 90 module, `HSL_VF06`, within the [20] (HSL) ([20]). The factorization (3.4) is performed using the HSL code `MA27` [7]. A concise summary of `HSL_VF06` is given as Algorithm 4.1.

Algorithm 4.1: HSL_VF06

1. Factorize $H = PLBL^TP^T$, using subroutines MA27A and MA27B, and $B = Q\Theta Q^T$. Obtain the diagonal matrix Γ from (3.14) with $\delta = \sqrt{\epsilon_M}$, and set $C = Q\Gamma Q^T$ and $D = \Gamma^{-\frac{1}{2}}\Theta\Gamma^{-\frac{1}{2}}$.
2. Solve $PLP^T g_{\rm b} = g$ using MA27Q.
3. Obtain $g_{\rm s} = \Gamma^{-\frac{1}{2}}Q^TP^Tg_{\rm b}$
4. Find $s_{\rm S} = \arg\min \langle g_{\rm S}, s_{\rm S}\rangle + \frac{1}{2}\langle s_{\rm S}, Ds_{\rm S}\rangle \quad$ subject to $\quad \|s_{\rm S}\|_2 \leq \Delta$ using Algorithm 3.1, with stopping tolerance $\epsilon = 10n\epsilon_M$.
5. Recover $s_{\rm q} = PQ\Gamma^{\frac{1}{2}}s_{\rm S}$.
6. Solve $PCL^TP^Ts = s_{\rm q}$ using MA27R.

Figure 4.0.

In order to demonstrate the potential of our proposal, we have conducted a limited number of numerical tests using HSL_VF06. We consider the standard trust-region method for the minimization of an objective $f(x)$ of n real variables x presented as Algorithm 4.2.

We choose the specific values $\epsilon_g = 0.00001$, $\eta_1 = 0.01$, $\eta_2 = 0.95$, $\gamma_1 = 0.5$, and $\gamma_2 = 2$, and set an upper limit of $20n$ iterations. In all cases, the initial trust-region radius is set to $\|M_0\|_\infty$. The step s_k in step 2 is computed using either Algorithm 4.1, or using the algorithm proposed in [17] and implemented as the HSL fortran 90 module HSL_VF05 using default settings. The latter algorithm is appropriate for general trust-region norms, but is not as efficient as HSL_VF06 when the absolute-value norm (3.5)–(3.6) is used.

In our tests we compare three choices of norm, namely the ℓ_2 norm, the absolute-value norm, and the norm defined by forming the Schnabel-Eskow [28] modified Cholesky factorization of H. The latter also uses MA27, and is available as part of the LANCELOT nonlinear programming package (see Chapter 3 of [6]). Other norms have been compared by [17].

All our tests were performed on an IBM RISC System/6000 3BT workstation with 64 Megabytes of RAM; the codes are all double precision Fortran 90, compiled under xlf90 with -O optimization, and IBM library BLAS are used. The test examples we consider are the currently available larger examples from the CUTE test set [2] for which negative curvature is frequently encountered. Tests were terminated if more than thirty CPU minutes elapsed.

The results of our tests are given in Table 4.2. In these tables, in addition to the name and dimension of each example, we give the number of objective function ("#f") and derivative ("#g") values computed, and the total CPU time required in seconds. We indicate those cases where one or other method performs at least 10% better than its competitors by highlighting the relevant figure in bold. A † indicates that convergence to different local minimizers occurred.

The results may effectively be divided into three categories. Into the first category fall problems which appear to be relatively easy, that is those which require few evaluations without a

Algorithm 4.2: Standard Trust-Region Algorithm

0. An initial point x_0 and an initial trust-region radius Δ_0 are given, as are constants ϵ_g, η_1, η_2, γ_1, and γ_2, which are required to satisfy the conditions

$$0 < \eta_1 \leq \eta_2 < 1 \text{ and } 0 < \gamma_1 < 1 \leq \gamma_2.$$

Set $k = 0$.

1. Stop if $\|\nabla_x f(x_k)\|_2 \leq \epsilon_g$.

2. Define a second-order Taylor series model q_k and a positive-definite preconditioner M_k. Compute a step s_k to "sufficiently reduce the model" q_k within the trust-region $\|s\|_{M_k} \leq \Delta_k$.

3. Compute the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{q_k(x_k) - q_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$, let $x_{k+1} = x_k + s_k$; otherwise let $x_{k+1} = x_k$.

4. Set

$$\Delta_{k+1} = \begin{cases} \gamma_2 \Delta_k & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \gamma_1 \Delta_k & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment k by one and go to Step 1.

Figure 4.0.

sophisticated trust-region norm. For such problems, the ℓ_2 norm performs best, and the other norms add little while incurring the extra expense of the factorization. The excellent behaviour of the ℓ_2 norm on such problems has already been noted (see, [17]). The second category contains problems for which the Hessian or its factors are relatively dense, and the cost of forming the preconditioner dominates. This category includes problems `EIGENALS`, `MSQRTALS`, `NONCVXU2`, `SPARSINE` and `SPMSRTLS`. These indicate the limitations of our approach, and for these problems preconditioners which try to mimic the structure of the Hessian without incurring the cost of the fill-in — such as the limited-memory incomplete Cholesky factorization proposed in [23], and the references contained therein — are likely to be preferable. The third category contains the harder, highly nonlinear problems `CURLYxx`, `NONCVXUN`, `SBRYBND`, `SCOSINE` and `SCURLYxx`. For these problems, the ℓ_2 norm is ineffective, and some rescaling is necessary. Interestingly, the modified absolute-value preconditioner outperforms the other sophisticated preconditioner on all but one of these, often by a large margin.

It is interesting to note that the number of "wasted" function evaluations (the difference between $\#g$ and $\#f$ in in Table 4.2) is significantly higher for the new method than for its competitors. There appear to be two reasons for this. Firstly, the initial trust-region radius, $\|M_0\|_\infty$, is often far too large when using the factorization preconditioners, and many iterations

Table 4.1. A comparison of trust-region methods using the ℓ_2, modified Cholesky and modified absolute-value norms. See the text for a key to the data.

example		n	ℓ_2			modified Cholesky			modified abs-value		
			$\#f$	$\#g$	CPU	$\#f$	$\#g$	CPU	$\#f$	$\#g$	CPU
BROYDN7D	†	1000	110	103	7.4	50	35	**4.1**	126	81	10.0
BRYBND		1000	13	13	**0.9**	15	15	2.3	23	15	2.6
CHAINWOO	†	1000	915	626	81.8	176	115	**7.9**	175	103	9.5
COSINE		1000	11	11	**0.1**	41	25	1.3	20	14	1.0
CRAGGLVY	†	1000	19	19	0.9	23	23	1.4	18	16	1.3
CURLY10		1000	23	21	29.9	57	35	**7.6**	55	33	8.6
CURLY20		1000	21	20	35.8	57	36	27.7	8	8	**6.6**
CURLY30		1000	22	21	46.2	71	42	81.9	12	9	**18.3**
DIXMAANA		1500	13	13	**0.3**	35	23	1.4	8	8	0.7
DIXMAANE		1500	14	14	**1.8**	150	85	7.9	115	76	8.2
DQRTIC		1000	43	43	**0.3**	54	54	1.3	33	32	1.3
EIGENALS		930	66	53	**71.0**	63	47	85.4	$>$ 1800 secs.		
FREUROTH		1000	17	17	**0.4**	86	48	3.5	132	84	7.5
GENHUMPS		1500	14474	13964	1023.0	$>20n$ its.			10208	9521	**796.4**
GENROSE		1000	721	665	48.1	434	312	**19.3**	1109	776	58.7
MANCINO		100	24	23	**20.2**	77	58	275.9	19	10	37.5
MSQRTALS		1024	35	30	**394.2**	$>$ 1800 secs.			$>$ 1800 secs.		
NCB20B		1000	45	29	141.1	33	20	26.8	21	12	**10.8**
NONCVXUN		1000	$>$ 1800 secs.			$>20n$ its.			3020	2493	**490.7**
NONCVXU2		1000	272	227	**29.7**	$>20n$ its.			$>20n$ its.		
SBRYBND		1000	$>$ 1800 secs.			59	28	9.7	65	28	**5.5**
SCOSINE	†	1000	$>$ 1800 secs.			90	70	4.4	70	14	**1.7**
SCURLY10		1000	$>$ 1800 secs.			61	45	10.2	40	6	**2.2**
SCURLY20		1000	$>$ 1800 secs.			75	52	41.3	41	6	**6.3**
SCURLY30		1000	$>$ 1800 secs.			75	52	105.4	45	7	**16.3**
SENSORS	†	100	21	20	**8.1**	66	51	32.0	61	39	24.05
SINQUAD		5000	152	99	**21.7**	14	14	100.0	14	14	86.0
SPARSINE		1000	16	16	**36.5**	361	205	1047.5	$>$ 1800 secs.		
SPMSRTLS	†	1000	18	16	**2.05**	$>$ 1800 secs.			$>$ 1800 secs.		

are required to cut it to a value for which progress may be made. In our experience, it is usually beneficial to determine a good initial radius, and, given how inexpensive the wasted iterations are in our case — the functions are cheap to evaluate, and the solution of the block-diagonal trust-region problems are, by design, trivial — the cost is not especially high. However, as evaluation costs may be high in general, more sophisticated strategies, such as that in [27], may be preferred.

The second cause of wasted function evaluations happened far less frequently, but occurs following a change in the shape of the trust-region as one or more eigenvalues change sign. In some cases, — the example `SCOSINE` is a point in case — a significant number of radius reductions were required to find a value appropriate for the new geometry. We foresee this as a significant problem, and are currently investigating more sophisticated schemes for trust-region management.

5. Discussion and conclusions

We believe that our results indicate that the modified absolute-value factorization provides a useful norm for trust-region minimization so long as as the factorization is feasible. In particular, for ill-conditioned problems, the norm appears to be especially effective. We do not pretend that (3.15) is uniformly appropriate, but suggest that, at the very least, its use should be considered when a problem is know to be ill-conditioned.

We recognize some potential difficulties with our approach. The attendees at the 1981 NATO Advanced Research Institute on "Nonlinear Optimization" (see contributions 1.31–1.35 in [25]) had much to say about Goldfarb [16]'s proposal, and the comments made there are equally appropriate here. In particular Roger Fletcher (Dundee) expressed concern that the distortion induced by (3.5) and (3.9) may be substantial. We accept that (3.15) may not be as desirable as (3.1), but believe that while (3.1) is out of the question for most large-scale problems, (3.15) is practical, and often useful, for many of them. Fletcher also worried that changes in the pivot ordering during the factorization of a sequence of problems may make it difficult to derive effective methods for adjusting the trust-region radius. Whilst we have observed occasions where pivot-order changes have drastically altered the geometry, and while this sometimes requires a large number of wasted iterations in which the trust-region radius is reduced, for the vast majority of iterations the usual, naive trust-region management seems to be satisfactory. However, we recognize this as a possible defect, and are currently investigating more sophisticated trust-region adjustment strategies both in this and other contexts.

Acknowledgement

The authors are grateful to Nick Higham, whose comments at the 1997 Dundee Conference on Numerical Analysis were the inspiration for this work, and the curators of the Botanical Gardens, Dundee, for providing the tranquil setting in which the ideas presented here were allowed to blossom.

References

1. C. Ashcraft, R. G. Grimes, and J. G. Lewis. Accurate symmetric indefinite linear equation solvers. Technical report, Boeing Computer Services, Seattle, Washington, USA, 1995.
2. I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, 21(1):123–160, 1995.
3. J. R. Bunch and L. C. Kaufman. Some stable methods for calculating inertia and solving symmetric linear equations. *Mathematics of Computation*, 31:163–179, 1977.

4. J. R. Bunch and B. N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 8(4):639–655, 1971.
5. S. H. Cheng and N. J. Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. Numerical Analysis Report No. 289, Manchester Centre for Computational Mathematics, Manchester, England, 1996.
6. A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A)*. Number 17 in Springer Series in Computational Mathematics. Springer Verlag, Heidelberg, Berlin, New York, 1992.
7. I. S. Duff and J. K. Reid. MA27: A set of Fortran subroutines for solving sparse symmetric sets of linear equations. Report R-10533, AERE Harwell Laboratory, Harwell, UK, 1982.
8. I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.
9. I. S. Duff and J. K. Reid. Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems. *ACM Transactions on Mathematical Software*, 22(2):227–257, 1996.
10. I. S. Duff, J. K. Reid, N. Munksgaard, and H. B. Neilsen. Direct solution of sets of linear equations whose matrix is sparse, symmetric and indefinite. *Journal of the Institute of Mathematics and its Applications*, 23:235–250, 1979.
11. R. Fletcher. Factorizing symmetric indefinite matrices. *Linear Algebra and its Applications*, 14:257–272, 1976.
12. D. M. Gay. Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, 2(2):186–197, 1981.
13. P. E. Gill and W. Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, 7(3):311–350, 1974.
14. P. E. Gill, W. Murray, D. B. Ponceléon, and M. A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM Journal on Matrix Analysis and Applications*, 13(1):292–311, 1992.
15. P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London and New York, 1981.
16. D. Goldfarb. The use of negative curvature in minimization algorithms. Technical Report TR80-412, Department of Computer Sciences, Cornell University, Ithaca, New York, USA, 1980.
17. N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. Technical Report RAL-TR-97-028, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 1997.
18. J. Greenstadt. On the relative efficiencies of gradient methods. *Mathematics of Computation*, 21:360–367, 1967.
19. Harwell Subroutine Library. *A catalogue of subroutines (release 12)*. AEA Technology, Harwell, Oxfordshire, England, 1995.
20. Harwell Subroutine Library. *A catalogue of subroutines (release 13)*. AEA Technology, Harwell, Oxfordshire, England, 1998. To appear.
21. M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report T. P. 515, AERE Harwell Laboratory, Harwell, UK, 1973.
22. N. J. Higham. Stability of the diagonal pivoting method with partial pivoting. Numerical Analysis Report No. 265, Manchester Centre for Computational Mathematics, Manchester, England, 1995.
23. C.-J. Lin and J. J. Moré. Incomplete Cholesky factorizations with limited memory. Technical Report ANL/MCS-P682-0897, Argonne National Laboratory, Illinois, USA, 1997.
24. J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.
25. M. J. D. Powell, editor. *Nonlinear Optimization 1981*, London and New York, 1982. Academic Press.
26. H. E. Salzer. A note on the solution of quartic equations. *Mathematics of Computation*, 14(71):279–281, 1960.
27. A. Sartenaer. Automatic determination of an initial trust region in nonlinear programming. *SIAM Journal on Scientific Computing*, 18(6):1788–1804, 1997.

28. R. B. Schnabel and E. Eskow. A new modified Cholesky factorization. *SIAM Journal on Scientific Computing*, 11(6):1136–1158, 1991.
29. D. C. Sorensen. Newton's method with a model trust modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.
30. H. W. Turnbull. *Theory of equations*. Oliver and Boyd, Edinburgh and London, 1939.
31. C. Xu and J. Z. Zhang. A scaled optimal path trust region algorithm. Talk at the 16th International Symposium on Mathematical Programming in Lausanne, Switzerland, August 1997.