DAMTP 1997/NA12

The use of band matrices for second derivative approximations in trust region algorithms

M.J.D. Powell

In many trust region algorithms for optimization calculations, each iteration seeks a vector $\underline{d} \in \mathcal{R}^n$ that solves the linear system of equations $(B+\lambda I)\underline{d}=-\underline{g}$, where B is a symmetric estimate of a second derivative matrix, I is the unit matrix, $\underline{g}$ is a known gradient vector, and λ is a parameter that controls the length of $\underline{d}$. Several values of λ may be tried on each iteration, and, when there is no helpful sparsity in B, it is usual for each solution to require $\mathcal{O}(n^3)$ operations. However, if an orthogonal matrix Ω is available such that $M=\Omega^T B\,\Omega$ is an $n \times n$ matrix of bandwidth $2s+1$, then $\Omega^T d$ can be calculated in only $\mathcal{O}(ns^2)$ operations for each new λ, by writing the system in the form $(M+\lambda I)(\Omega^T \underline{d}) = -\Omega^T \underline{g}$. We find, unfortunately, that the construction of M and Ω from B is usually more expensive than the solution of the original system, but in variable metric and quasi-Newton algorithms for unconstrained optimization, each iteration changes B by a matrix whose rank is at most two, and then updating techniques can be applied to Ω. Thus it is possible to reduce the average work per iteration from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^{7/3})$ operations. Here the elements of each orthogonal matrix are calculated explicitly, but instead one can express the orthogonal matrix updates as products of Givens rotations, which allows the average work per iteration to be only $\mathcal{O}(n^{11/5})$ operations. Details of procedures that achieve these savings are described, and the $\mathcal{O}(n^{7/3})$ complexity is confirmed by numerical results.

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

July, 1997.

1. Introduction

Many algorithms for calculating the least value of a function $F(\underline{x})$, $\underline{x} \in \mathcal{R}^n$, are iterative, and, given a starting vector of variables $\underline{x}_1$, they generate a sequence $\underline{x}_{k+1}$, $k=1,2,3,\ldots$, in $\mathcal{R}^n$, such that $F(\underline{x}_k)$, $k=1,2,3,\ldots$, is intended to converge to the required least function value. Therefore we regard $\underline{x}_k$ as the current vector of variables at the beginning of the k-th iteration. Further, in quasi-Newton and Newton–Raphson algorithms, both $F(\underline{x}_k)$ and the gradient vector $\underline{\nabla} F(\underline{x}_k)$ are available, and also an $n \times n$ symmetric matrix B_k has been formed, that is the true second derivative matrix $\nabla^2 F(\underline{x}_k)$ in the Newton–Raphson case, but otherwise B_k tends to include some useful second derivative information as k increases. Indeed, most quasi-Newton algorithms choose B_k in a way that satisfies the equation

$$B_k\,(\underline{x}_k - \underline{x}_{k-1}) = \underline{g}_k - \underline{g}_{k-1}, \qquad k \geq 2, \tag{1.1}$$

because every function F with integrable second derivatives has the property

$$\left\{ \int_{\theta=0}^{1} \nabla^2 F(\underline{x}_{k-1} + \theta\,[\underline{x}_k - \underline{x}_{k-1}]\,)\, d\theta \right\} (\underline{x}_k - \underline{x}_{k-1}) = \underline{g}_k - \underline{g}_{k-1}, \tag{1.2}$$

and we see that the matrix inside the braces is the average value of $\nabla^2 F$ on the line segment between $\underline{x}_{k-1}$ and $\underline{x}_k$.

These data provide the approximation

$$F(\underline{x}_k + \underline{d}) \approx Q_k(\underline{d}) = F(\underline{x}_k) + \underline{d}^T \underline{g}_k + \tfrac{1}{2} \underline{d}^T B_k\, \underline{d}, \qquad \underline{d} \in \mathcal{R}^n, \tag{1.3}$$

which assists the choice of $\underline{x}_{k+1}$ by the k-th iteration. For example, in a "line search algorithm" every B_k is forced to be positive definite, in order that the value of $Q_k(\underline{d})$ is least when $\underline{d}$ satisfies $\underline{\nabla} Q_k(\underline{d}) = \underline{g}_k + B_k\, \underline{d} = 0$, which defines $\underline{d}$ uniquely. Termination may occur if $\|\underline{d}\|$ or $\|\underline{g}_k\|$ is sufficiently small, but otherwise $\underline{x}_{k+1}$ has the form $\underline{x}_{k+1} = \underline{x}_k + \alpha\, \underline{d}$, the step-length α being generated in a way that provides the reduction $F(\underline{x}_{k+1}) < F(\underline{x}_k)$ in the objective function. However, we are going to study some techniques for "trust region methods", that also employ the approximation (1.3), but that do not require B_k to be positive definite.

The term "trust region" means that there is a bound of the form

$$\|\underline{d}\| \leq \Delta_k \tag{1.4}$$

on the vectors $\underline{d}$ that are allowed in expression (1.3), where we specialise to the case of the Euclidean vector norm, and where the positive number Δ_k is generated automatically before the start of the k-th iteration for $k \geq 2$. In other words, the vectors $\underline{d}$ that satisfy the constraint (1.4) are in a *region* of $\mathcal{R}^n$ where we *trust* the approximation (1.3). The prototype of such an algorithm tries to set $\underline{x}_{k+1}$ to $\underline{x}_k + \underline{d}^*$, where $\underline{d}^*$ is the $\underline{d}$ that minimizes $Q_k(\underline{d})$ subject to condition (1.4). The criterion for accepting this trial value of $\underline{x}_{k+1}$ usually depends on a comparison of $F(\underline{x}_k)$, $Q_k(\underline{d}^*)$ and $F(\underline{x}_k+\underline{d}^*)$. If $\underline{x}_k+\underline{d}^*$ is found to be unacceptable, then a smaller trust region is selected for the next iteration. All of the procedures that have been mentioned so far are well-known, and are described in text-books, for example Fletcher (1987).

We address the calculation of $\underline{d}^*$, and we present a technique that can save much work in quasi-Newton algorithms when n is large and B_k has no useful sparsity. The technique can be included in those algorithms that derive $\underline{d}^*$ from the following theorem (see Moré, 1983, for instance), which is a reformulation of the Karush–Kuhn–Tucker conditions of the optimization problem that defines $\underline{d}^*$.

Theorem 1: Let μ_k be the least nonnegative number such that the matrix $B_k+\mu_k I$ has no negative eigenvalues. Then $\underline{d}^*$ satisfies $\|\underline{d}^*\| \leq \Delta_k$ and the equations

$$(B_k+\lambda\, I)\, \underline{d}^* = -\underline{g}_k \quad \textit{and} \quad \lambda\, (\Delta_k - \|\underline{d}^*\|) = 0 \tag{1.5}$$

for some $\lambda \geq \mu_k$. Conversely, if these conditions hold for some $\lambda \geq \mu_k$ and some $\underline{d}^* \in \mathcal{R}^n$, then $\underline{d}^*$ provides the least value of the approximation (1.3) subject to the trust region bound (1.4). □

The usual way of employing the theorem to calculate $\underline{d}^*$ is described by Moré and Sorensen (1983). One can take the following view of the calculation. Several nonnegative values of λ are tried, and the following questions are asked about the matrix $B_k+\lambda\, I$ for each one. (1) Does $B_k+\lambda\, I$ have any negative eigenvalues? (2) Are all the eigenvalues of this matrix positive? (3) Are all the eigenvalues nonnegative with at least one of them being zero? Exactly one of these questions has an affirmative answer for each λ. In the first case, the trial λ is less than the value that is required. In the second case, the equations

$$(B_k+\lambda\, I)\, \underline{d} = -\underline{g}_k \tag{1.6}$$

define $\underline{d} \in \mathcal{R}^n$ uniquely, and λ is too small if and only if this vector has the property $\|\underline{d}\| > \Delta_k$. Further, λ is too large if and only if both $\lambda > 0$ and $\|\underline{d}\| < \Delta_k$ hold. In the third case, when $B_k+\lambda\, I$ is singular, the equations (1.6) usually have no solutions, and then λ is too small. It is also too small when the system (1.6) is consistent and *every* solution $\underline{d}$ satisfies $\|\underline{d}\| > \Delta_k$. Otherwise, the required value of λ has been found.

Efficient ways of adjusting λ, and tolerances that make the number of trials finite in practical calculations, are also considered by Moré and Sorensen (1983). A typical iteration solves the equations (1.6) for only two different values of λ, but more values might be advantageous if this calculation were less expensive. The amount of work to solve the system is $\mathcal{O}(n^3)$ for each λ when B_k is a general symmetric matrix.

Therefore the author has had the following idea in mind for several years. The equations (1.6) can be written in the form

$$\Omega_k^T (B_k+\lambda\, I)\, \Omega_k\, \Omega_k^T \underline{d} = -\Omega_k^T \underline{g}_k \quad \Longleftrightarrow \quad (\Omega_k^T B_k\, \Omega_k+\lambda\, I)\, (\Omega_k^T \underline{d}) = -(\Omega_k^T \underline{g}_k), \tag{1.7}$$

where Ω_k is any $n \times n$ orthogonal matrix. Further, by letting Ω_k be the product of at most $\frac{1}{2}(n-2)(n-1)$ Givens rotations or $n-2$ Householder rotations, the matrix $\Omega_k^T B_k \Omega_k = M_k$, say, can be made tridiagonal (Wilkinson, 1965), and we set $\underline{\widehat{g}}_k = \Omega_k^T \underline{g}_k$. Thus expression (1.7) becomes the tridiagonal system

$$(M_k + \lambda I)(\Omega_k^T \underline{d}) = -\underline{\widehat{g}}_k. \tag{1.8}$$

Hence, for each $\lambda > \mu_k$, it provides the vector $\Omega_k^T \underline{d}$ in only $\mathcal{O}(n)$ computer operations. Further, due to the identity $\|\Omega_k^T \underline{d}\| = \|\underline{d}\|$, there is no need to derive $\underline{d}$ from $\Omega_k^T \underline{d}$ until the adjustment of λ is complete. It is not clear, however, that the idea will reduce the amount of computation when B_k is general, because the work of calculating Ω_k, M_k and $\underline{\widehat{g}}$ is $\mathcal{O}(n^3)$. Unfortunately, we find in Section 2 that usually these preliminary operations are more expensive than the savings that occur during the adjustment of λ.

On the other hand, B_k is not a general symmetric matrix in quasi-Newton algorithms for unconstrained optimization calculations. Indeed, in the very successful DFP, BFGS and symmetric Broyden methods, B_{k+1} is generated by adding a matrix of rank two to B_k. Moreover, the rank one method achieves equation (1.1) by applying the formula

$$B_{k+1} = B_k + \frac{\underline{r}_k \underline{r}_k^T}{\underline{r}_k^T (\underline{x}_k - \underline{x}_{k-1})}, \qquad k \geq 2, \tag{1.9}$$

where $\underline{r}_k$ is the vector $\underline{g}_k - \underline{g}_{k-1} - B_k(\underline{x}_k - \underline{x}_{k-1})$, except that there are safeguards in case the denominator of expression (1.9) is close to zero. Therefore we ask whether the idea of the previous paragraph can reduce the amount of computation when the rank of the difference matrix $B_{k+1} - B_k$ is only one or two for every k. An obvious question of this kind is the following one. If $B_{k+1} - B_k$ has rank one, and if an orthogonal matrix Ω_k is available such that $M_k = \Omega_k^T B_k \Omega_k$ is tridiagonal, can one generate matrices Ω_{k+1} and M_{k+1}, for the technique of the previous paragraph on the next iteration, in fewer than $\mathcal{O}(n^3)$ operations? This question is also addressed in Section 2, and again the answer seems to be unfavourable.

Therefore Section 3 considers the possibility of letting the band-widths of the matrices M_k, $k = 1, 2, 3, \ldots$, be greater than three. We find that, if the band-width is of magnitude $n^{2/3}$, then the average work per iteration of a quasi-Newton trust region algorithm can be reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^{7/3})$, assuming that the number of trial values of λ on each iteration is bounded by a constant that is independent of n. This assertion is justified by giving careful attention to the calculation of Ω_{k+1} and M_{k+1} from Ω_k and M_k when the difference $B_{k+1} - B_k$ is a matrix of rank one.

In Section 4, some numerical results are presented in order to investigate whether the savings of Section 3 are achieved for values of n that are useful in practice. They suggest that the new technique is advantageous when $n \geq 50$, so, for larger values of n, the routine work of a quasi-Newton trust region algorithm without sparsity is reduced by a factor of about $(n/50)^{2/3}$. In these experiments, the elements of the matrices Ω_k and M_k are calculated explicitly for each k. Further savings may be possible, however, by expressing Ω_k as a product of elementary rotations. This subject is also addressed in Section 4.

There are some other useful techniques for estimating the minimizer of $\{Q_k(\underline{d}) : \|\underline{d}\| \leq \Delta_k\}$, where Q_k is still the quadratic function (1.3). In particular, several trust region algorithms apply a version of the conjugate gradient method to the symmetric linear system $B_k \underline{d} = -\underline{g}_k$, beginning with $\underline{d}_1 = 0$, as suggested by Steihaug (1983). Thus, if B_k is positive definite, the iterations of the method generate a sequence $\underline{d}_{j+1}$, $j=1,2,3,\ldots$, say, such that $\underline{d}_{j+1}$ is the solution of this system after at most n iterations in exact arithmetic, unless the procedure is terminated sooner. Further, the method has the property that the sequence of norms $\|\underline{d}_{j+1}\|$, $j=1,2,3,\ldots$, increases monotonically. Therefore the iterations of the conjugate gradient method continue until $\|\underline{d}_{j+1}\| \geq \Delta_k$ occurs or until $\|B_k \underline{d}_{j+1} + \underline{g}_k\|$ is sufficiently small. In the former case, we approximate $\underline{d}^*$ by the vector on the line segment from $\underline{d}_j$ to $\underline{d}_{j+1}$ that is on the trust region boundary, and in the latter case we pick the approximation $\underline{d}_{j+1}$. Alternatively, when B_k is not positive definite, the stopping conditions that have been mentioned are retained, but, before one of them is achieved, the conjugate gradient method may choose a search direction $\underline{s}_j$, say, from $\underline{d}_j$, that satisfies $\underline{s}_j^T B_k \underline{s}_j \leq 0$. Termination also occurs in this case, the estimate of $\underline{d}^*$ being the vector $\underline{d}_j + \theta \underline{s}_j$, where θ is the positive number that is defined by $\|\underline{d}_j + \theta \underline{s}_j\| = \Delta_k$. This procedure has the advantage that there is no need for B_k to be positive definite. Further, some pre-conditioning can be included.

Therefore we ask whether there is a need for the new techniques that are the subject of Sections 3 and 4. An easy answer is that trust region algorithms are so useful that any idea deserves careful investigation. Moreoever, our procedure will require less work when the number of conjugate gradient iterations in the previous paragraph exceeds about $n^{1/3}$. More importantly, it can happen that the given version of the conjugate gradient method generates a poor approximation to $\underline{d}^*$. For example, if $\Delta_k = 1$ and if Q_k is the quadratic function

$$Q_k(\underline{d}) = -10^{-6} d_1 - 10^{-6} d_1^2 - d_2^2, \qquad \underline{d} \in \mathcal{R}^2, \tag{1.10}$$

then, in the notation above, $\underline{s}_1$ is a multiple of the first coordinate direction and $\underline{s}_1^T B_k \underline{s}_1 < 0$ occurs. Therefore $\underline{d}^*$ is approximated by the vector $(1,0)^T$, so the final value of $Q_k(\underline{d})$ is only -2×10^{-6}. This choice is unsuitable, because $\underline{d} = (0,1)^T$ also satisfies the trust region bound and it implies $Q_k(\underline{d}^*) \leq -1$. Instead, the chosen value of $\underline{d}$ should have the property that $Q_k(\underline{d})/Q_k(\underline{d}^*)$ is close to one, because $Q_k(\underline{d})$ is the predicted value of the reduction $F(\underline{x}_k + \underline{d}) - F(\underline{x}_k)$. It is shown by Moré and Sorensen (1983) that a good value of $Q_k(\underline{d})$ can always be achieved by a procedure for estimating $\underline{d}^*$ that is based on Theorem 1.

n	Full-solve	Transform	Band-solve
50	0.05	0.47	< 0.01
100	0.37	3.57	< 0.01
200	2.95	28.33	< 0.01
400	24.21	232.66	< 0.01
800	198.74	1909.89	< 0.02

Table 1: The timings in seconds for inequality (2.1)

2. Tridiagonal updating

The possible advantages of employing the system of equations (1.8) instead of expression (1.6), when B_k is a general symmetric matrix and M_k is tridiagonal, were tested in the following way. Three Fortran programs were written. One solves the equations (1.6) by Cholesky factorization, given $B_k+\lambda I$ and $\underline{g}_k$, where $B_k+\lambda I$ has no sparsity and is positive definite. The second program is given the general symmetric matrix B_k, and it forms the orthogonal matrix Ω_k explicitly, as a product of $\frac{1}{2}(n-2)(n-1)$ Givens rotations, such that $\Omega_k^T B_k \Omega_k = M_k$ is tridiagonal, the tridiagonal part of M_k being calculated too. The third program solves the system (1.8) using a tridiagonal Cholesky factorization, where M_k is available from the second program, where $\lambda \in \mathcal{R}$ and $\widehat{\underline{g}}_k \in \mathcal{R}^n$ are given, and where $M_k+\lambda I$ is positive definite. Let "Full-solve", "Transform" and "Band-solve" be the times that are taken by each of these calculations, respectively, and let Λ_k be the number of trial values of λ on the k-th iteration of the trust region method. Then it is advantageous to employ the tridiagonal form (1.8) of the equations (1.6) when Λ_k satisfies the condition

$$\text{Transform} + \Lambda_k \, \text{Band-solve} < \Lambda_k \, \text{Full-solve}. \tag{2.1}$$

Typical timings were found by running the programs on a Sun 4 workstation for several values of n, the results being shown in Table 1. We see that the orthogonal transformation to tridiagonal form saves work only when Λ_k is at least 10, but much smaller values of Λ_k occur in practice (Moré, 1983).

Therefore we turn to the other suggestion that is made in Section 1 for better efficiency, namely to try to improve on the "Transform" times of Table 1 when each matrix B_{k+1} is formed by adding a matrix of low rank to B_k. Specifically, we let B_{k+1} be defined by the equation

$$B_{k+1} = B_k + \phi_k \underline{r}_k \underline{r}_k^T, \tag{2.2}$$

where $\phi_k \in \mathcal{R}$ and $\underline{r}_k \in \mathcal{R}^n$ are known, which is analogous to formula (1.9). Further, we assume that not only B_k, but also an orthogonal matrix Ω_k and a tridiagonal matrix M_k are available, such that $M_k = \Omega_k^T B_k \Omega_k$. These assumptions provide the identity

$$\Omega_k^T B_{k+1} \Omega_k = M_k + \phi_k (\Omega_k^T \underline{r}_k)(\Omega_k^T \underline{r}_k)^T = M_k + \phi_k \widehat{\underline{r}}_k \widehat{\underline{r}}_k^T, \tag{2.3}$$

say. It follows that $\Omega_{k+1}^T B_{k+1} \Omega_{k+1}$ can be made tridiagonal by setting $\Omega_{k+1} = \Omega_k \Omega$, where Ω is any orthogonal matrix that forces $M_{k+1} = \Omega^T (M_k + \phi_k \widehat{\underline{r}}_k \widehat{\underline{r}}_k^T) \Omega$ to be tridiagonal. We consider the construction of Ω in this way, in order to try to take advantage of the sparsity of M_k, and we let Ω be a product of Givens rotations. Therefore, if the elements of $\Omega_{k+1} = \Omega_k \Omega$ are found explicitly and if Ω_k is a general orthogonal matrix, then the total work of the calculation of Ω_{k+1} and M_{k+1} is less than $\mathcal{O}(n^3)$ operations if and only if the number of Givens rotations is fewer than $\mathcal{O}(n^2)$. We ask whether this aim can be achieved.

The author has spent much time on trying to develop a method that constructs a suitable matrix Ω in $\mathcal{O}(n)$ rotations, and these attempts have led him to believe that the task is impossible when M_k is a general tridiagonal symmetric matrix, but he has not proved the conjecture. Therefore the remainder of this section will present some remarks that suggest that usually the number of rotations, m say, has to be of magnitude n^2. We let G_i, $i = 1, 2, \ldots, m$, be the Givens rotations that occur, so Ω^T is the product $G_m G_{m-1} \cdots G_1$. Further, we consider the sequence

$$A_0 = M_k + \phi_k \widehat{\underline{r}}_k \widehat{\underline{r}}_k^T \quad \text{and} \quad A_i = G_i A_{i-1} G_i^T, \quad i = 1, 2, \ldots, m, \tag{2.4}$$

because this notation provides the identity

$$A_m = G_m G_{m-1} \cdots G_1 A_0 G_1^T G_2^T \cdots G_m^T = \Omega^T (M_k + \phi_k \widehat{\underline{r}}_k \widehat{\underline{r}}_k^T) \Omega = M_{k+1}. \tag{2.5}$$

It is usual in practice to set $A = A_0$ initially, and then to overwrite A by $G_i A G_i^T$ for $i = 1, 2, \ldots, m$, in order that A becomes the required tridiagonal matrix M_{k+1}. We employ this point of view for our analysis, although it would be inefficient computationally to treat $\phi_k \widehat{\underline{r}}_k \widehat{\underline{r}}_k^T$ as a full $n \times n$ matrix.

Now the structure of a Givens rotation is such that, when A is overwritten by $G_i A G_i^T$, then only two rows and columns of A are changed. Further, the angle of rotation can be chosen so that any one of the nonzero off-diagonal elements in these rows and columns of the old A is overwritten by zero. For example, if A has the structure that is shown on the left hand side of the display

$$\begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 \\ 0 & \times & \times & \times & 0 \\ 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{pmatrix} \underset{(3,4)\text{ rotation}}{\Longrightarrow} \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{pmatrix}, \tag{2.6}$$

then a rotation that changes the third and fourth rows and columns of A can yield $A_{42}=A_{24}=0$. Further, because the new values of A_{3j} and A_{4j} are linear combinations of the old values of A_{3j} and A_{4j} for $j=1$, 2 and 5, and because the symmetry of A is preserved, the new matrix A has the structure on the right hand side of expression (2.6). We see that the rotation does not reduce the number of nonzero off-diagonal elements of A. Nevertheless, it is advantageous, because the new A can be made tridiagonal by applying a rotation that gives $A_{53}=A_{35}=0$ by combining the last two rows and columns of A.

The example (2.6) shows that, if the nonzero elements of A are close to the diagonal, and if one of them is made nonzero, then a zero element may become nonzero. In order that the new nonzeros are not unnecessarily far from the diagonal, we restrict attention to Givens rotations that alter adjacent rows and columns of A. Specifically, for $\ell=1,2,\ldots,n-1$, we let $G^{(\ell)}$ denote a rotation that mixes the ℓ-th and $(\ell+1)$-th rows and columns of A. Therefore, if the first $\ell-1$ (or last $n-\ell-1$) elements of $\underline{\widehat{r}}_k \in \mathcal{R}^n$ are zero, then we can pick $G^{(\ell)}$ so that the first ℓ (or last $n-\ell$) elements of $G^{(\ell)}\underline{\widehat{r}}_k$ are zero. Thus the rank one matrix $G^{(\ell)}(\phi_k \underline{\widehat{r}}_k \underline{\widehat{r}}_k^T)\, G^{(\ell)T}$ can have more zero rows and columns than $\phi_k \underline{\widehat{r}}_k \underline{\widehat{r}}_k^T$, although $G^{(\ell)}$ is a single Givens rotation. By applying this remark recursively, it should be possible for m in expression (2.5) to be much less than $\frac{1}{2}(n-2)\,(n-1)$, which is the number of Givens rotations that is mentioned at the beginning of this section. On the other hand, the overwriting of A by $G^{(\ell)} A\, G^{(\ell)T}$ tends to spread out the contribution from M_k to A. We take the view that any advantages from the rank one part of formula (2.2) do not apply to elements of A that have received a contribution from M_k. We also assume that no zeros occur in the tridiagonal part of each matrix A.

We say that 'stage q has been reached' when the current A has the property that the elements A_{ij}, $q+2 \le i \le n$, $1 \le j \le q$, are all zero and remain at zero for the rest of the calculation, where q is any integer from $[1, n-2]$. Therefore the final A is achieved when the q-th stage is reached for all of these values of q. For $q=1,2,\ldots,n-2$, we let $i(q)$ be the least integer in $[0, m]$ such that the q-th stage is reached when A is the matrix $A_{i(q)}$, where we are using the notation of formula (2.4). In general, the numbers $i(q)$, $q=1,2,\ldots,n-2$, are different positive integers. We address this case, and we let $\widehat{m}$ be the integer $i(\widehat{q})$, where $\widehat{q} \in [1, n-2]$ is defined by the equation

$$i(\widehat{q}) = \min\{i(q) : q=1,2,\ldots,n-2\}. \tag{2.7}$$

Therefore the replacement of A by $G_{\hat{m}} A\, G_{\hat{m}}^T$ in the calculation causes the bottom left $(n-\widehat{q}-1)\times\widehat{q}$ submatrix of A to become zero. Because the replacement reduces the rank of this submatrix, $G_{\hat{m}}$ must be a Givens rotation of type $G^{(\hat{q})}$ or $G^{(\hat{q}+1)}$. Furthermore, the element $A_{\hat{q}+2\,\hat{q}}$ retains the value zero for the remainder of the calculation, although we are assuming that every matrix A has the properties $A_{\hat{q}+2\,\hat{q}+1} \neq 0$ and $A_{\hat{q}+1\,\hat{q}} \neq 0$. It follows that none of the rotations G_i, $i=\widehat{m}+1, \widehat{m}+$

$2,\ldots,m$, is of type $G^{(\hat{q})}$ or $G^{(\hat{q}+1)}$. Now a rotation of type $G^{(p)}$ commutes with a rotation of type $G^{(\ell)}$ if p and ℓ are any integers in $[1,n-2]$ that satisfy $|p-\ell|\geq 2$. Therefore we may assume without loss of generality that there is an integer $\check{m}$ in $[\widehat{m},m]$ such that the rotations G_i, $i=\widehat{m}+1,\widehat{m}+2,\ldots,\check{m}$, are of type $G^{(\ell)}$ with $\ell\leq\widehat{q}-1$, while the rotations G_i, $i=\check{m}+1,\check{m}+2,\ldots,m$, are of type $G^{(\ell)}$ with $\ell\geq\widehat{q}+2$. We also assume $\widehat{q}\leq n-3$, which does not lose generality in our investigation of the magnitude of m, because the ordering of all rows and columns of matrices can be reversed. Thus A has the required form after $\check{m}$ rotations, except that the bottom right $(n-\widehat{q}+1)\times(n-\widehat{q}+1)$ submatrix of A has the structure

$$\begin{pmatrix} \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \times & ? & \cdots & ? \\ 0 & \times & \times & \times & \ddots & \vdots \\ 0 & ? & \times & \times & \ddots & ? \\ \vdots & \vdots & \ddots & \ddots & \ddots & \times \\ 0 & ? & \cdots & ? & \times & \times \end{pmatrix}. \tag{2.8}$$

The last $m-\check{m}$ rotations have to overwrite this submatrix by a tridiagonal one without disturbing the first row and column that are shown. We seek the least number of rotations that are necessary for this final part of the calculation.

We consider the number of rotations of type $G^{(\hat{q}+2)}$ that occur in the sequence G_i, $i=\check{m}+1,\check{m}+2,\ldots,m$. If there are two of them, then we try to reorder the sequence without altering the value of the product $G_m G_{m-1}\cdots G_{\check{m}+1}$, so that the two occurrences of $G^{(\hat{q}+2)}$ are as close together as possible in the sequence. Again we make use of the fact that $G^{(p)}$ commutes with $G^{(\ell)}$ if $|p-\ell|\geq 2$. If the reordering causes two Givens rotations of the same type, $G^{(\ell)}$ say, to be adjacent, then these rotations are replaced by their product, because it is another Givens rotation of type $G^{(\ell)}$. Further, the product is dropped from the sequence if it happens to be a diagonal matrix, which requires further modifications if there are diagonal elements of -1. Thus m is reduced by one or two. When both $G^{(\hat{q}+2)}$ rotations survive, then we let G_j be the first occurrence of a $G^{(\hat{q}+2)}$ rotation in the reordered sequence G_i, $i=\check{m}+1,\check{m}+2,\ldots,m$. It follows that G_{j+1} is of type $G^{(\hat{q}+3)}$ and that G_{j+2} is either the other occurrence of $G^{(\hat{q}+2)}$ or is of type $G^{(\hat{q}+4)}$. In the former case, the product $G_{j+2}G_{j+1}G_j$, which is a $G^{(\hat{q}+2)}$ times a $G^{(\hat{q}+3)}$ times a $G^{(\hat{q}+2)}$ rotation, can also be expressed as a $G^{(\hat{q}+3)}$ times a $G^{(\hat{q}+2)}$ times a $G^{(\hat{q}+3)}$ rotation. We prefer the new form, in order that there is only one $G^{(\hat{q}+2)}$ rotation for the current value of m. Otherwise, when G_j, G_{j+1} and G_{j+2} are of types $G^{(\hat{q}+2)}$, $G^{(\hat{q}+3)}$ and $G^{(\hat{q}+4)}$, respectively, we let p be the least integer such that G_{j+p} is not of type $G^{(\hat{q}+p+2)}$, so it is of type $G^{(\ell)}$ for some integer ℓ that satisfies $\widehat{q}+2\leq\ell\leq\widehat{q}+p$. Further, because the commutivity property allows G_{j+p-1} to be exchanged with G_{j+p} if $\ell\leq\widehat{q}+p-1$, which can be done recursively, we assume without loss of generality that $\ell = \widehat{q}+p$. Thus the product $G_{j+p}G_{j+p-1}G_{j+p-2}$ is a $G^{(\hat{q}+p)}$ times a $G^{(\hat{q}+p+1)}$ times a $G^{(\hat{q}+p)}$ rotation. Therefore we express it as a $G^{(\hat{q}+p+1)}$ times a $G^{(\hat{q}+p)}$ times a $G^{(\hat{q}+p+1)}$ rotation, which removes a $G^{(\hat{q}+2)}$ rotation as before if $p=2$. Otherwise, if $p>2$, then the new G_{j+p-2} is of type $G^{(\hat{q}+p+1)}$, so it commutes with G_i, $j \le i \le j+p-3$. It follows that the two occurrences of $G^{(\hat{q}+2)}$ can be moved closer together. By continuing in this way, the occurrences will become so close that they can always be replaced by a single $G^{(\hat{q}+2)}$ rotation. Similarly, any number of $G^{(\hat{q}+2)}$ rotations can be reduced to a single rotation without increasing m.

Having ensured that there is only one $G^{(\hat{q}+2)}$ rotation, we use the commutivity again, in order to position $G^{(\hat{q}+2)}$ as early as possible in the sequence G_i, $i = \check{m}+1, \check{m}+2, \ldots, m$, without increasing m. Again we may replace a $G^{(\ell)}$ times a $G^{(\ell+1)}$ times a $G^{(\ell)}$ rotation by a $G^{(\ell+1)}$ times a $G^{(\ell)}$ times a $G^{(\ell+1)}$ rotation. It follows that, if $G_{\check{m}+p}$ becomes the $G^{(\hat{q}+2)}$ rotation, then $G_{\check{m}+i}$ is a $G^{(\hat{q}+p-i+2)}$ rotation for $i = 1, 2, \ldots, p$. Now $\widehat{q}$ is the index of the first row and column of the submatrix (2.8), the sequence of rotations has to reduce the sum of squares $\sum_{i=\hat{q}+3}^{n} A_{i\,\hat{q}+1}^2$ in the $(\widehat{q}+1)$-th column of A to zero, but $G_{\check{m}+p}$ is the only rotation in the sequence G_i, $i = \check{m}+1, \check{m}+2, \ldots, m$, that alters the value of this sum of squares. Moreover, $A_{\hat{q}+2\,\hat{q}+1}$ and $A_{\hat{q}+3\,\hat{q}+1}$ are the only elements of this column that are altered by $G_{\check{m}+p}$. Therefore the angle of this rotation is determined by the need to achieve $A_{\hat{q}+3\,\hat{q}+1}=0$, the previous rotations have to provide $A_{i\,\hat{q}+1}=0$, $i = \widehat{q}+4, \widehat{q}+5, \ldots, n$, and the final value of $A_{\hat{q}+2\,\hat{q}+1}$ is nonzero. Furthermore, by working backwards through the rotations from $A_{\check{m}+p}$, which is the matrix A that is generated by the Givens rotation $G_{\check{m}+p}$, we deduce that the elements below the diagonal in the second column of the submatrix (2.8) are the last $n-\widehat{q}-1$ elements of the corresponding column of $(G_{\check{m}+p}G_{\check{m}+p-1}\cdots G_{\check{m}+1})^T A_{\check{m}+p}$. It follows from the suppression of diagonal rotation matrices that the element $A_{\hat{q}+p+2\,\hat{q}+1}$ of the submatrix (2.8) is nonzero, and that zeros occur below this element in the second column of the submatrix. Therefore the rotations G_i, $i = \check{m}+1, \check{m}+2, \ldots, \check{m}+p$, are *defined* by the need to provide the structure

$$\begin{pmatrix} \times & \times & 0 & 0 & 0 & \cdots & 0 \\ \times & \times & \times & 0 & 0 & \cdots & 0 \\ 0 & \times & \times & \times & ? & \cdots & ? \\ 0 & 0 & \times & \times & \times & \ddots & \vdots \\ 0 & 0 & ? & \times & \times & \ddots & ? \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \times \\ 0 & 0 & ? & \cdots & ? & \times & \times \end{pmatrix}, \tag{2.9}$$

in the bottom right $(n-\widehat{q}+1)\times(n-\widehat{q}+1)$ submatrix of $A_{\check{m}+p}$.

The remaining rotations G_i, $i = \check{m}+p+1, \check{m}+p+2, \ldots, m$, are of type $G^{(\ell)}$

with $\ell \geq \widehat{q}+3$, so they do not alter the first two rows and columns of the structure (2.9). It follows that stage $\widehat{q}+1$ is reached when this structure occurs. Further, because the numbers $i(q)$, $i=1,2,\ldots,n-2$, of expression (2.7) are assumed to be different, the integer p of the previous paragraph is positive, and $i(\widehat{q}+1)$ has the value $\check{m}+p$. These remarks confirm that one or more of the "?" elements in the second column of the submatrix (2.8) is nonzero. This submatrix is a part of $A_{\hat{m}} = G_{\hat{m}} A_{\hat{m}-1} G_{\hat{m}}^T$, and we recall that $G_{\hat{m}}$ is of type $G^{(\hat{q})}$ or $G^{(\hat{q}+1)}$, but the following argument shows that $G_{\hat{m}}$ is not of type $G^{(\hat{q})}$.

Let i be an integer from $[\widehat{q}+3, n]$ such that $A_{i\,\hat{q}+1}$ is a nonzero "?" element in the second column of expression (2.8). Then, if $G_{\hat{m}}$ were of type $G^{(\hat{q})}$, the replacement of $A = A_{\hat{m}-1}$ by $A = A_{\hat{m}} = G_{\hat{m}} A_{\hat{m}-1} G_{\hat{m}}^T$ would cause both $A_{\hat{q}+2\,\hat{q}}$ and $A_{i\hat{q}}$ to become zero. This possibility is excluded by our assumptions, however, because a rotation of type $G^{(\hat{q})}$ causes $A_{\hat{q}+2\,\hat{q}}$ to receive a contribution from the tridiagonal part of A. Therefore $G_{\hat{m}}$ is of type $G^{(\hat{q}+1)}$. On the other hand, if $\widehat{q} \geq 2$ occurred, then, by considering the rotations G_i, $i = \widehat{m}+1, \widehat{m}+2, \ldots, \check{m}$, which are of type $G^{(\ell)}$ with $\ell \leq \widehat{q}-1$, we would find similarly that $G_{\hat{m}}$ is not of type $G^{(\hat{q}+1)}$. Thus we have a contradiction unless $\widehat{q}=1$, the alternative $\widehat{q}=n-2$ being ruled out by our assumption $\widehat{q} \leq n-3$. It follows that expression (2.8) shows the complete $n \times n$ matrix A after $\check{m}$ rotations and that $\check{m}$ is equal to $\widehat{m}$.

Next we rearrange the rotations G_i, $i = \widehat{m}+p+1, \widehat{m}+p+2, \ldots, m$, if necessary, without increasing m, so that there is only one $G^{(4)}$ rotation that occurs as early as possible. Then we deduce from previous arguments that the leading rotations of the new sequence are *defined* by the zeros that have to be introduced into the third column of the submatrix (2.9). By applying this construction recursively, we reach the following conclusion. If an $n \times n$ symmetric matrix with the structure (2.8) is made tridiagonal by an orthogonal similarity transformation, and if the transformation is a product of Givens rotations of type $G^{(\ell)}$, $\ell \in [3, n-1]$, so no transformation disturbs the leading row and column of the matrix, then the following well-known procedure minimizes the number of Givens rotations. We work through the columns of the matrix, A say, in sequence. Within the j-th column, we consider the elements A_{ij}, $i = j+2, j+3, \ldots, n$, in reverse order. Whenever A_{ij} is nonzero, we overwrite A by GAG^T, where G is the Givens rotation of type $G^{(i-1)}$ that causes the new value of A_{ij} to be zero. Thus any zeros that were introduced deliberately by previous rotations are preserved, and A becomes tridiagonal as required. Our discovery that this procedure minimizes the number of rotations subject to the given assumptions may be new.

Therefore we seek a lower bound on the number of Givens rotations that occur when this procedure is applied to the symmetric matrix (2.8), so the first stage has been reached by a preliminary calculation. Then the $(q+1)$-th stage is achieved from the q-th stage for $q = 1, 2, \ldots, n-3$. Specifically, for each of these values of q, $A_{i(q+1)}$ is calculated from $A_{i(q)}$ by applying Givens similarity transformations to the current A, the number of transformations being the difference

$$i(q+1) - i(q) = \ell(q) - q-2, \tag{2.10}$$

where $i(q)$ is defined just before equation (2.7), and where $\ell(q)$ is the row number of the last nonzero element in the $(q+1)$-th column of $A_{i(q)}$. Our bound is derived from the claim that usually $\ell(q)$ has the property

$$\ell(q) \geq \min[2q+2, n], \qquad q=1,2,\ldots,n-3. \tag{2.11}$$

This condition is satisfied when $q=1$, because it has been noted already that at least one of the "?" elements in the second column of expression (2.8) is nonzero. We are going to corroborate the claim by induction, assuming that it holds for any integer q in the interval $[1,n-4]$.

The matrix $A_{i(q+1)}$ is related to $A_{i(q)}$ by the formula

$$A_{i(q+1)} = G_{i(q+1)} G_{i(q+1)-1} \cdots G_{i(q)+1} A_{i(q)} G_{i(q)+1}^T G_{i(q)+2}^T \cdots G_{i(q+1)}^T, \tag{2.12}$$

where the Givens rotation $G_{i(q)+j}$ is of type $G^{(\ell(q)-j)}$ for $j=1,2,\ldots,i(q+1)-i(q)$. The identity (2.10) shows that, as expected, the last of these rotations is of type $G^{(q+2)}$. Thus the $(q+2)$-th column of the product $A_{i(q)} G_{i(q)+1}^T G_{i(q)+2}^T \cdots G_{i(q+1)}^T$, say $\underline{v}_q$, is a linear combination of the columns of $A_{i(q)}$ whose indices are in the interval $[q+2, \ell(q)]$, and it includes a nonzero multiple of the $\ell(q)$-th column of $A_{i(q)}$. Now, when $A=A_{i(q)}$ and $\ell(q) \leq n-2$, at least one of the elements A_{ij}, $\ell(q)+2 \leq i \leq n$, $q+2 \leq j \leq \ell(q)$, is nonzero, because the $\ell(q)$-th stage has not been reached. Hence we expect $i(\underline{v}_q) \geq \ell(q)+2$ to occur, where $i(\underline{v}_q)$ is the index of the last nonzero component of $\underline{v}_q$. Moreover, the $(q+2)$-th column of $A_{i(q+1)}$ is the vector $\widehat{\underline{v}}_q = G_{i(q+1)} G_{i(q+1)-1} \cdots G_{i(q)+1} \underline{v}_q$, and the Givens rotations are such that the i-th component of $\widehat{\underline{v}}_q$ is different from the i-th component of $\underline{v}_q$ only if i is in the interval $[q+2, \ell(q)]$. Therefore $i(\underline{v}_q)$ is also the index of the last nonzero component of $\widehat{\underline{v}}_q$ when $\ell(q) \leq n-2$, which implies $\ell(q+1) \geq \ell(q)+2$. Alternatively, if $\ell(q) \geq n-1$, then the last component of $\underline{v}_q$ receives a contribution from the tridiagonal part of $A_{i(q)}$ and part of this contribution is inherited by the last component of $\widehat{\underline{v}}_q$, so our assumptions give $\ell(q+1)=n$. Thus the inequality $\ell(q+1) \geq \min[\ell(q)+2, n]$ is satisfied for every integer q in $[1,n-4]$. The claim (2.11) now follows from the inductive argument that has been mentioned, except that some of the remarks about nonzero elements do not yet have a rigorous proof.

The procedure under consideration uses $m-\widehat{m}$ Givens rotations to form a tridiagonal matrix from expression (2.8), and this number is the sum $m-\widehat{m} = \sum_{q=1}^{n-3} \{i(q+1)-i(q)\}$. Therefore expressions (2.10) and (2.11) imply the bound

$$m-\widehat{m} = \sum_{q=1}^{n-3} \{\ell(q)-q-2\} \geq \sum_{q=1}^{n-3} \min[q, n-q-2] = \begin{cases} \frac{1}{4}n^2-n+1, & n \text{ even} \\ \frac{1}{4}n^2-n+\frac{3}{4}, & n \text{ odd}, \end{cases} \tag{2.13}$$

where the last equation is elementary. We have now reached the unwelcome conclusion that the total number of Givens rotations is of magnitude n^2. It follows that $\mathcal{O}(n^3)$ operations are usually necessary to calculate Ω_{k+1} and M_{k+1} from Ω_k, M_k and $\underline{r}_k$, as mentioned in Section 1.

In Sections 3 and 4 we will make use of the remark that the bounds (2.11) and (2.13) are satisfied as equations when the matrix (2.8) has the quindiagonal structure

$$A_{\hat{m}} = \begin{pmatrix} \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \times & \times & \ddots & \vdots \\ 0 & \times & \times & \times & \ddots & 0 \\ 0 & \times & \times & \times & \ddots & \times \\ \vdots & \ddots & \ddots & \ddots & \ddots & \times \\ 0 & \cdots & 0 & \times & \times & \times \end{pmatrix}, \tag{2.14}$$

which can be verified by considering the details of the paragraph that includes equation (2.12). The author has been informed by M.R. Osborne and B.N. Parlett, however, that a different sequence of Givens similarity transformations is usually applied to make the matrix (2.14) tridiagonal. It begins by calculating $A_{\hat{m}+i} = G_{\hat{m}+i} A_{\hat{m}+i-1} G_{\hat{m}+i}^T$, $i=1,2,\ldots,\hat{\imath}$, where $G_{\hat{m}+i}$ is of type $G^{(2i+1)}$ and where $\hat{\imath}$ is the least integer that satisfies $2\hat{\imath}+2 \leq n$. These rotations are chosen to provide the form

$$A_{\hat{m}+\hat{\imath}} = \begin{pmatrix} \times & \times & 0 & 0 & 0 & \cdots & 0 \\ \times & \times & \times & 0 & 0 & \cdots & 0 \\ 0 & \times & \times & \times & \times & \ddots & \vdots \\ 0 & 0 & \times & \times & \times & \ddots & 0 \\ 0 & 0 & \times & \times & \times & \ddots & \times \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \times \\ 0 & 0 & \cdots & 0 & \times & \times & \times \end{pmatrix}. \tag{2.15}$$

Specifically, $G_{\hat{m}+1}$ is the rotation of type $G^{(3)}$ that gives $A_{42} = A_{24} = 0$ when $A = A_{\hat{m}}$ is overwritten by $A_{\hat{m}+1} = G_{\hat{m}+1} A\, G_{\hat{m}+1}^T$, but the (6,3) and (3,6) elements of the new matrix are nonzero. Therefore $G_{\hat{m}+2}$ is the $G^{(5)}$ rotation that restores these elements to zero in $A = G_{\hat{m}+2} A_{\hat{m}+1} G_{\hat{m}+2}^T$, but now A_{85} and A_{58} become nonzero. Continuing in this way we find that, for each integer i in $[2, \hat{\imath}]$, the matrix $A = A_{\hat{m}+i-1}$ has only two nonzero elements outside its quindiagonal part, namely $A_{2i+2\,2i-1}$ and $A_{2i-1\,2i+2}$, so we let $G_{\hat{m}+i}$ be of type $G^{(2i+1)}$. Thus the rotations shift the position of the nonzero element that is outside the quindiagonal part of A, except that $G_{\hat{m}+\hat{\imath}}$ pushes the offending element off the end of the matrix, leaving the structure (2.15).

We can also regard expression (2.14) as the bottom right $(n-1)\times(n-1)$

submatrix of the matrix (2.15). Then, as in the previous paragraph, a sequence of Givens similarity transformations gives the submatrix the form (2.15). This technique is applied recursively until A becomes tridiagonal. The main difference from the procedure that was described earlier is as follows. Letting $j(q)$ be the least integer such that the q-th stage is reached when $A_{j(q)}$ is formed, there are no nonzero elements outside the quindiagonal part of $A_{j(q)-1}$ for every integer q in $[2, n-2]$. Moreover, we deduce from the identities $j(1)=\widehat{m}$, $j(2)=\widehat{m}+1$ and $j(3)=\widehat{m}+\widehat{i}+1$ that the total number of rotations has the value

$$j(n-2) - \widehat{m} = \sum_{q=1}^{n-3} \{j(q+1) - j(q)\} = 1 + \sum_{q=2}^{n-3} \left\lfloor \tfrac{1}{2}(n-q) \right\rfloor, \tag{2.16}$$

where $\lfloor x \rfloor$ denotes the integer part of $x \in \mathcal{R}$.

It can be shown that the right hand sides of expressions (2.13) and (2.16) are equal. Furthermore, identical Givens rotations occur in the two procedures that have been described for transforming expression (2.14) into a tridiagonal matrix. Indeed, one can construct the second procedure from the first one by using only the fact that Givens rotations of types $G^{(\ell)}$ and $G^{(p)}$ commute if $|p-\ell| \geq 2$. On the other hand, each A in the second procedure has only $\mathcal{O}(n)$ nonzero elements, but the number of nonzero elements becomes of magnitude n^2 during the first procedure.

Because the first row and column of the matrix (2.14) are irrelevant to the methods that have been described, we take the view that they do not exist, and we regard the remainder of expression (2.14) as an $n \times n$ quindiagonal matrix, so n is now one less than before. It follows that we have been considering procedures for transforming general quindiagonal symmetric matrices into tridiagonal forms. By increasing n by 1 in formula (2.13), we deduce that $\left\lfloor \frac{1}{4}n^2 - \frac{1}{2}n + \frac{1}{4} \right\rfloor$ Givens rotations occur, which may be the least number of rotations in the general case. It is proved by Stewart (1997) that the magnitude of this number is no less than $\mathcal{O}(n^2)$.

3. Updating methods for band matrices

Let M_k be a symmetric $n \times n$ matrix of band-width $2s+1$ in expression (2.5), and let the Givens rotations $G_1, G_2, \ldots, G_m$ have the property that the band-width of M_{k+1} is also $2s+1$. Then the main result of this section is that the number of rotations can be of magnitude $m = \mathcal{O}(n^2/s)$. Therefore, by choosing s to be greater than one when n is large, we can improve greatly on the amount of work that is indicated in Section 2 for the tridiagonal case. Specifically, a suitable sequence of rotations will be derived from the following calculation and from some of the conclusions of Section 2.

We address the case when $n = 10$ and $s = 2$. Therefore the matrix $A_0 =$

$M_k+\phi_k\underline{\widehat{r}}_k\underline{\widehat{r}}_k^T$ of equation (2.4) has the form

$$\begin{pmatrix} \times & \times & \times & \cdot & \odot & \odot & \odot & \odot & \odot & \odot \\ \times & \times & \times & \times & \cdot & \odot & \odot & \odot & \odot & \odot \\ \times & \times & \times & \times & \times & \cdot & \odot & \odot & \odot & \odot \\ \cdot & \times & \times & \times & \times & \times & \cdot & \boxdot & \boxdot & \boxdot \\ \odot & \cdot & \times & \times & \times & \times & \times & \cdot & \boxdot & \boxdot \\ \odot & \odot & \cdot & \times & \times & \times & \times & \times & \cdot & \boxdot \\ \odot & \odot & \odot & \cdot & \times & \times & \times & \times & \times & \cdot \\ \odot & \odot & \odot & \boxdot & \cdot & \times & \times & \times & \times & \times \\ \odot & \odot & \odot & \boxdot & \boxdot & \cdot & \times & \times & \times & \times \\ \odot & \odot & \odot & \boxdot & \boxdot & \boxdot & \cdot & \times & \times & \times \end{pmatrix}, \tag{3.1}$$

where the nonzero part of M_k is confined to the elements that are denoted by "$\times$", and where the other elements may be nonzero because they belong to $\phi_k\underline{\widehat{r}}_k\underline{\widehat{r}}_k^T$. The differences between the notations "$\cdot$", "$\boxdot$" and "$\odot$" will be explained later. For $i=1,2,\ldots,n-4$, we overwrite the vector $\underline{\widehat{r}}_k$ by $G_i\underline{\widehat{r}}_k$, where G_i is the Givens rotation of type $G^{(n-i)}$ that causes the $(n-i+1)$-th component of the new $\underline{\widehat{r}}_k$ to be zero. Further, for each i, we also overwrite A by the matrix $A_i=G_iA_{i-1}G_i^T$. It follows that the final matrix A has zero elements in the positions that are indicated by "$\odot$" in expression (3.1). Indeed, the Givens rotations have been chosen so that the last $n-4$ elements of the final $\underline{\widehat{r}}_k$ are all zero, so $\phi_k\underline{\widehat{r}}_k\underline{\widehat{r}}_k^T$ makes nonzero contributions only to the leading 4×4 submatrix of the final A. Therefore the "$\odot$" elements under consideration are the corresponding elements of the product $G_{n-4}G_{n-3}\cdots G_1M_kG_1^TG_2^T\cdots G_{n-4}^T$. Symmetry allows us to restrict attention to the "$\odot$" element in position (i,j), where j and i are integers from the intervals $[1,3]$ and $[j{+}4,n]$, respectively. Now the i-th row of $G_{n-4}G_{n-3}\cdots G_1M_k$ is a linear combination of the $(i{-}1)$-th and all later rows of M_k, so $i\geq j{+}4$ implies that the (i,j) element of $G_{n-4}G_{n-3}\cdots G_1M_k$ is zero. Moreover, because each G_i is of type $G^{(n-i)}$, postmultiplication of a matrix by $G_1^TG_2^T\cdots G_{n-4}^T$ does not alter the first three columns of the matrix. Therefore the final A has zeros in the "$\odot$" positions as claimed.

We let i be any integer from the interval $[5,n]$, in order to explain the notation "$\boxdot$" of expression (3.1). As in the previous paragraph, we regard the i-th row of $G_{n-4}G_{n-3}\cdots G_1M_k$ as a linear combination of the last $n-i+2$ rows of M_k. It follows that the positions of the zero elements below the diagonal of the product $G_{n-4}G_{n-3}\cdots G_1M_k$ are denoted both by "$\odot$" and by "$\boxdot$" in the form (3.1). With this structure in mind, we form the matrix $G_{n-4}G_{n-3}\cdots G_1M_kG_1^TG_2^T\cdots G_{n-4}^T$ by post-multiplying $G_{n-4}G_{n-3}\cdots G_1M_k$ by the matrices G_i^T, $i=1,2,\ldots,n-4$, in sequence. Now, because G_i is a Givens rotation of type $G^{(n-i)}$, the post-multiplication by G_i^T mixes the $(n{-}i)$-th and $(n{-}i{+}1)$-th columns of the current matrix. Thus the "$\boxdot$" elements below the diagonal of expression (3.1) are made nonzero by these post-multiplications. A highly useful result is given by this construction. Specifically, because the last three elements of the fourth column of the matrix (3.1) are zero before the final G_{n-4}^T multiplication, these last three elements become proportional to the last three elements of the fifth column of the final A. Similarly, the last two elements of the fifth column of the final A are proportional to the last two elements of the final sixth column. Thus the construction of the previous paragraph provides a matrix A that has the following property. We consider the elements below the diagonal in the last $n-3$ columns of the final A that are in the "$\cdot$" and "$\boxdot$" positions of expression (3.1). Then these elements form part of a matrix of rank one.

The construction and remarks of the last two paragraphs can be generalized in the following way to any positive integers s and n that satisfy $1 \leq s \leq n-3$, where M_k is a symmetric $n \times n$ matrix of band-width $2s+1$. We set $A = M_k + \phi_k \underline{\widehat{r}}_k \underline{\widehat{r}}_k^T$ initially. Then, for $i = 1, 2, \ldots, n-s-2$, we let G_i be the Givens rotation of type $G^{(n-i)}$ that maximizes the number of zero elements at the beginning of the $(n-i+1)$-th row of $G_i A$ and we overwrite A by $G_i A\, G_i^T$. This specification is not only equivalent to the choice of G_i that has been described already, but also will take advantage of the rank one structure that has been mentioned. Thus the final A is symmetric, and its first $s+1$ rows and columns have the sparsity structure of a matrix of band-width $2s+3$. Therefore, because the remainder of A is an $(n-s-1) \times (n-s-1)$ submatrix, the complete matrix A has band-width $2s+3$ if s and n satisfy $n-s-1 \leq s+2$. Otherwise, more rotations are applied to A in order to achieve a band-width of $2s+3$. We will take advantage of the fact that $\{A_{ij} : s+2 \leq j \leq n-s-1,\ j+s+1 \leq i \leq n\}$ are elements of a matrix of rank one.

The further rotations when $s+2 < n-s-1$ should not disturb the first $s+1$ rows and columns of A, because they have the required sparsity structure already. For example, zero elements have been obtained in the "$\odot$" positions of expression (3.1), and we have to introduce zero elements into the "$\boxdot$" positions of that expression. Therefore, if we delete the first three rows and columns of the current matrix (3.1), and if we apply the procedure of the previous paragraph to the resultant 7×7 matrix, retaining $s = 2$, then the procedure will provide all the new zeros that are required. Three Givens rotations occur, and they are extended from 7×7 to 10×10 matrices by adding three rows and columns of the unit matrix in leading positions. These larger rotation matrices are the ones that are actually used. They do not disturb the band structure that has been achieved in the first three rows and columns of A, because they are of types $G^{(9)}$, $G^{(8)}$ and $G^{(7)}$. Similarly, for general s and n with $s+2 < n-s-1$, we replace A by its bottom right $(n-s-1) \times (n-s-1)$ submatrix for the moment and we decrease n to $n-s-1$, in order to apply the procedure of the previous paragraph to the reduced matrix. Thus we identify further Givens rotations, that are enlarged to the original dimensions by adding $s+1$ rows and columns of the unit matrix, and that provide a matrix A that has the required zeros in its first $2s+2$ rows and columns. This procedure can be applied recursively until A has band-width $2s+3$. The number of cycles through the recursive procedure is less than $n/(s+1)$, and each cycle requires fewer than n Givens rotations. Therefore the total number of rotations is $\mathcal{O}(n^2/s)$.

There is an unusual feature in the rank one property that is provided by the recursive procedure. Indeed, we have noted that certain elements below the diagonal of the final A are elements of a rank one matrix, and, by symmetry, the corresponding elements of the final A to the right of the diagonal are elements of the transpose of the rank one matrix. The unusual feature is that in general it is not possible for either rank one matrix to be symmetric, although we started with $\phi_k \underline{\hat{r}}_k \underline{\hat{r}}_k^T$. Fortunately, this feature does not invalidate the procedure that has been described.

A variation of the procedure can reduce the number of Givens rotations by about a factor of two when n is large. It is as follows when $n=10$ and $s=2$, for example. As before, we overwrite $A_0 = M_k + \phi_k \underline{\hat{r}}_k \underline{\hat{r}}_k^T$ by $G_3 G_2 G_1 A_0 G_1^T G_2^T G_3^T$, where G_1, G_2 and G_3 are the Givens rotations of types $G^{(9)}$, $G^{(8)}$ and $G^{(7)}$, respectively, such that the last three components of $G_3 G_2 G_1 \underline{\hat{r}}_k$ are zero. Thus, after these rotations, A has the structure

$$\begin{pmatrix}
\times & \times & \times & \Diamond & \Diamond & \Diamond & \Diamond & 0 & 0 & 0 \\
\times & \times & \times & \times & \Diamond & \Diamond & \Diamond & 0 & 0 & 0 \\
\times & \times & \times & \times & \times & \Diamond & \Diamond & 0 & 0 & 0 \\
\Diamond & \times & \times & \times & \times & \times & \Diamond & 0 & 0 & 0 \\
\Diamond & \Diamond & \times & \times & \times & \times & \times & \cdot & 0 & 0 \\
\Diamond & \Diamond & \Diamond & \times & \times & \times & \times & \times & \cdot & 0 \\
\Diamond & \Diamond & \Diamond & \Diamond & \times & \times & \times & \times & \times & \cdot \\
0 & 0 & 0 & 0 & \cdot & \times & \times & \times & \times & \times \\
0 & 0 & 0 & 0 & 0 & \cdot & \times & \times & \times & \times \\
0 & 0 & 0 & 0 & 0 & 0 & \cdot & \times & \times & \times
\end{pmatrix}, \tag{3.2}$$

where the elements in the positions "$\Diamond$" belong to the rank one matrix $(G_3 G_2 G_1 \underline{\hat{r}}_k)$ $(G_3 G_2 G_1 \underline{\hat{r}}_k)^T$. Previously we let G_4 in the $G_4 A G_4^T$ transformation be a Givens rotation of type $G^{(6)}$, which has the disadvantage of destroying the required zeros in the $(10,6)$ and $(6,10)$ positions. Instead, however, if G_4 is of type $G^{(1)}$, then the new A matrix can satisfy $A_{51} = A_{61} = A_{71} = A_{15} = A_{16} = A_{17} = 0$, and all the zeros of expression (3.2) are preserved. Further, by using a rotation of type $G^{(2)}$ next, followed by a rotation of type $G^{(3)}$, we can generate a matrix A of band-width seven as required.

Let G_i, $i = 1, 2, \ldots, 6$, be the sequence of six Givens rotations that occurs in the previous paragraph. They are chosen so that the last three elements of

$G_3G_2G_1\underline{\widehat{r}}_k$ are zero and so that both the first three and the last three elements of $G_6G_5G_4G_3G_2G_1\underline{\widehat{r}}_k$ are zero. We extend this construction to general values of s and n in the following way. We assume $s+2<n-s-1$, because otherwise A can be reduced to a matrix of band-width $2s{+}3$ by a single sweep of the recursive procedure that has been described already. We pick positive integers ℓ and p that satisfy $\ell+p=n-s-2$. Then we let G_i, $i=1,2,\ldots,\ell+p$, be Givens rotations that give the following properties when $\underline{\widehat{r}}_k$ is overwritten by $G_i\underline{\widehat{r}}_k$ for each i. If i is any integer in $[1,\ell]$, then G_i is of type $G^{(n-i)}$, and the last i components of $\underline{\widehat{r}}_k$ are zero after i rotations. Alternatively, if i is any integer in $[\ell{+}1,\ell{+}p]$, then G_i is of type $G^{(i-\ell)}$, and the first $i-\ell$ components of $\underline{\widehat{r}}_k$ are zero after i rotations. It follows that only the $(p{+}1)$-th to $(n{-}\ell)$-th components of the final $\underline{\widehat{r}}_k$ can be nonzero, so $\phi_k\underline{\widehat{r}}_k\underline{\widehat{r}}_k^T$ becomes a matrix of band-width $2s{+}3$. Further, if we set $A=M_k+\phi_k\underline{\widehat{r}}_k\underline{\widehat{r}}_k^T$ initially, and if as usual we overwrite A by $G_iA\,G_i^T$ for $i=1,2,\ldots,\ell{+}p$, then it is straightforward to verify that the final A also has band-width $2s+3$, except that there may be no zero elements in its top left $(p+1)\times(p+1)$ submatrix and in its bottom right $(\ell{+}1)\times(\ell{+}1)$ submatrix. Therefore more rotations are usually required if $\max[\ell,p]$ is greater than $s+1$. The current calculation is complete, however, in the example of the previous paragraph, because there we find the values $n=10$, $s=2$, $\ell=3$ and $p=3$.

On the other hand, if $s=2$ and $\ell=6$, for instance, then expression (3.1) shows the structure of the bottom right 10×10 submatrix of the final A of the previous paragraph, the zero elements being denoted by "$\odot$". Moreover, the elements in the "$\boxdot$" positions became zero during the calculation and were then overwritten in a way that provides a rank one structure below the diagonal as before. Therefore the bottom right $(\ell+1)\times(\ell+1)$ submatrix that requires further attention can be treated by the recursive procedure that has been described already. Indeed, one sweep of the recursive procedure is sufficient to correct this submatrix in the case $\ell\leq 2s+2$. Otherwise, the sweep provides the required band structure in the leading $s{+}1$ rows and columns of this submatrix, so the dimensions of the bottom right submatrix that requires further attention are reduced to $(\ell-s)\times(\ell-s)$. Using recursion as before, we find that this submatrix is transformed into the required band form by t applications of the procedure, where t is the least integer that satisfies $\ell\leq(t+1)(s{+}1)$. Moreover, the top left submatrix of A requires $\widehat{t}$ applications of an analogous procedure, where $\widehat{t}$ is the least nonnegative integer that has the property $p\leq(\widehat{t}{+}1)(s{+}1)$.

Next we bound the total number of Givens rotations that are required to reduce $M_k+\phi_k\underline{\widehat{r}}_k\underline{\widehat{r}}_k^T$ to a matrix of band-width $2s+3$ in the case $s+2<n-s-1$. We recall that this number is $\ell+p=n-s-2$ plus the work of treating a top left and bottom right submatrix of dimensions $(p+1)\times(p+1)$ and $(\ell+1)\times(\ell+1)$, respectively. We express ℓ as $\ell=t\,(s{+}1)+\tau$, where the value of the nonnegative integer t in the previous paragraph implies $1\leq\tau\leq s{+}1$. We also recall that the recursive procedure makes $m{-}s{-}2$ rotations when it is applied once to an $m{\times}m$

submatrix. Therefore the total number of rotations for treating the $(\ell{+}1)\times(\ell{+}1)$ submatrix has the value

$$\sum_{j=1}^{t}\{\ell - j\,(s{+}1)\} = \tfrac{1}{2}t\,\{2\ell - (t{+}1)\,(s{+}1)\} = \frac{\ell^2 - \ell\,(s{+}1) + \tau(s{+}1{-}\tau)}{2\,(s{+}1)}. \quad (3.3)$$

It follows from $1\le\tau\le s{+}1$ that this expression is at most $\frac{1}{8}(2\ell{-}s{-}1)^2/(s{+}1)$. Thus the number of Givens rotations of the complete calculation is bounded above by the sum

$$n-s-2+\frac{(2\ell-s-1)^2+(2p-s-1)^2}{8\,(s+1)}. \quad (3.4)$$

We pick values of ℓ and p that minimize this bound, namely $\ell=p=\frac{1}{2}(n-s-2)$ when $n-s$ is even and $\ell=p{+}1=\frac{1}{2}(n-s-1)$ when $n-s$ is odd. Hence the value of expression (3.4) is no greater than the number

$$n-s-2+\frac{(n-2s-2)^2+(n-2s-4)^2}{8\,(s+1)} = \frac{n^2-2n+2}{4\,(s+1)}. \quad (3.5)$$

Now the number of rotations is an integer, and the value (3.5) cannot be achieved. Thus we deduce the upper bound $\frac{1}{4}(n-1)^2/(s+1)$ on the total number of Givens rotations that occur. Moreover, by suppressing the $\tau(s{+}1{-}\tau)$ term of expression (3.3) and by setting $\ell=p=\frac{1}{2}(n-s-2)$, we find that the number of rotations is bounded below by $\frac{1}{4}(n-1)^2/(s+1)-\frac{1}{4}(s+1)$.

Having calculated a matrix A of band-width $2s{+}3$ that is similar to the matrix (2.2), we may wish to apply further Givens rotations that reduce the band-width to $2s{+}1$. We investigate the number of rotations that are made by a procedure that is analogous to the one that is considered at the end of Section 2. Specifically, letting A_0 be a general symmetric matrix of band-width $2s+3$, we overwrite $A=A_{i-1}$ by $A=A_i=G_iA_{i-1}G_i^T$ for $i=1,2,\ldots,\ell(n-s-1)$, where each G_i is a Givens rotation as usual and where the final A has the required band-width. The notation $\ell(n-s-1)$ arises from a key feature of the procedure. Indeed, there is a monotonically increasing sequence of nonnegative integers $\ell(j)$, $j=1,2,\ldots,n{-}s{-}1$, such that the first j rows and columns of $A_{\ell(j)}$ have the sparsity structure of a matrix of band-width $2s{+}1$, but, for $j\le n-s-2$, the bottom right $(n-j)\times(n-j)$ submatrix of $A_{\ell(j)}$ has band-width $2s+3$. We also define $\ell(0)=0$. In general, a nonzero element occurs in the $(j+s+2,j+1)$ position of $A_{\ell(j)}$ for each j in $[0,n-s-2]$. Therefore we assume that the sequence $\ell(j)$, $j=0,1,\ldots,n-s-1$, is strictly monotonic.

For each integer j in $[0,n-s-2]$, the procedure lets $G_{\ell(j)+1}$ be the Givens rotation of type $G^{(j+s+1)}$ that gives the matrix $A=A_{\ell(j)+1}=G_{\ell(j)+1}A_{\ell(j)}G_{\ell(j)+1}^T$ the property $A_{j+s+2\,j+1}=A_{j+1\,j+s+2}=0$. If $j\le n-2s-3$ and if $A_{j+s+2\,j+2s+3}\ne 0$, then the new A has two nonzero elements that violate the band-width of $2s+3$, namely $A_{j+s+1\,j+2s+3}$ and $A_{j+2s+3\,j+s+1}$. In this case we pick $G_{\ell(j)+2}$ to be the

Givens rotation of type $G^{(j+2s+2)}$ that restores these elements to zero when A is overwritten by $G_{\ell(j)+2} A\, G^T_{\ell(j)+2}$, but now, if $j \le n$–$3s$–4, then usually $A_{j+2s+2\, j+3s+4}$ and $A_{j+3s+4\, j+2s+2}$ become nonzero. We continue in this way as before, until the elements that violate the band-width of $2s$+3 are driven off the end of the matrix. Thus the sparsity structure that implies $A = A_{\ell(j+1)}$ is achieved. Because we use rotations of type $G^{(j+ps+p)}$, where p runs through the positive integers that satisfy $j+ps+p \le n-1$, it follows that the difference $\ell(j+1)-\ell(j)$ has the value $\lfloor (n-j-1)/(s+1) \rfloor$, provided that there are no unexpected zero matrix elements. Therefore the total number of rotations, namely $\ell(n-s-1)$, has the value

$$\sum_{j=0}^{n-s-2} \{\ell(j+1) - \ell(j)\} = \sum_{j=0}^{n-s-2} \lfloor (n-j-1)/(s+1) \rfloor = \sum_{j=1}^{n-s-1} \lfloor (j+s)/(s+1) \rfloor. \quad (3.6)$$

We seek upper and lower bounds on this sum that are free from the integer part operator. Therefore we write n in the form $n=(s+1)t+\tau$, where t and τ are integers such that τ satisfies $1 \le \tau \le s+1$. Thus expression (3.6) is the sum

$$\sum_{j=1}^{(s+1)(t-1)} \lfloor (j+s)/(s+1) \rfloor + \sum_{j=(s+1)(t-1)+1}^{(s+1)(t-1)+\tau} \lfloor (j+s)/(s+1) \rfloor = \sum_{i=1}^{t-1} i\,(s+1) + \tau t$$

$$= \tfrac{1}{2}(n-\tau)\left(\frac{n-\tau}{s+1} - 1\right) + \tau\left(\frac{n-\tau}{s+1}\right) = \frac{n^2 - n\,(s+1) + \tau(s+1-\tau)}{2\,s+2}. \quad (3.7)$$

It follows that the number of rotations is at most $(n-\frac{1}{2}s-\frac{1}{2})^2/(2s+2)$, and that usually this number is at least $(n^2-ns-n)/(2s+2)$. The difference between these bounds is only $\frac{1}{8}(s+1)$. We note also that expression (3.7) when $s=1$ has the value $\left\lfloor \frac{1}{4}n^2 - \frac{1}{2}n + \frac{1}{4} \right\rfloor$, as stated in the last paragraph of Section 2.

The results of this section provide a useful answer to the following question. If M_k is a symmetric $n \times n$ matrix of bandwidth $2s$+1, if $\underline{\widehat{r}}_k$ is a general vector in $\mathcal{R}^n$, and if Ω is a product of Givens rotations such that $M_{k+1} = \Omega^T (M_k + \phi_k \underline{\widehat{r}}_k \underline{\widehat{r}}_k^T)\, \Omega$ also has band-width $2s$+1, then how many Givens rotations are required? Specifically, a procedure for constructing M_{k+1} has been presented that is in two parts. The first part finds a matrix of bandwidth $2s$+3 using at most $\frac{1}{4}(n$–$1)^2/(s$+$1)$ rotations, and then the calculation is completed by a technique that requires not more than $\frac{1}{2}(n - \frac{1}{2}s - \frac{1}{2})^2/(s+1)$ rotations. It follows from $s \ge 1$ that the total number of rotations is bounded above by $\frac{3}{4}(n-1)^2/(s+1)$. Therefore M_{k+1} and $\Omega_{k+1} = \Omega_k \Omega$ can be calculated from M_k, Ω_k and $\underline{r}_k$ in $\mathcal{O}(n^3/s)$ computer operations.

The other main task of an iteration of a trust region method, that updates second derivative approximations by a quasi-Newton formula, is to solve the system of equations (1.8) for a few different values of the parameter λ. Since M_k has band-width $2s$+1, the work for each λ is of magnitude $s^2 n$, and we recall the assumption that the number of values of λ on each iteration is bounded above by a constant. Thus the total work per iteration is $\mathcal{O}(s^2 n + n^3/s)$. It follows that it

s	Update	Reduce	Band-solve	Total
10	8.20	16.58	0.06	24.90
20	4.41	8.02	0.20	12.83
40	2.54	4.01	0.74	8.03
60	1.86	2.66	1.54	7.60
80	1.51	1.99	2.63	8.76
100	1.39	1.57	3.92	10.80
120	1.20	1.31	5.44	13.39
140	1.08	1.07	6.99	16.13
160	1.05	0.94	8.70	19.39
200	1.02	0.64	12.31	26.28

Table 2: Some timings in seconds when $n = 400$

is optimal to let s be of magnitude $n^{2/3}$ in the procedure that has been described, so the complexity of each iteration is $\mathcal{O}(n^{7/3})$, as claimed in Section 1. Therefore, for sufficiently large n, the use of band matrices can provide a very substantial improvement over the work of solving full $n \times n$ systems of linear equations on each iteration.

4. Conclusions

Two Fortran subroutines have been written that apply the procedures of Section 3. Specifically, the subroutine "Update" is given an $n \times n$ symmetric matrix M_k of band-width $2s+1$, an $n \times n$ orthogonal matrix Ω_k, a vector $\underline{r}_k \in \mathcal{R}^n$ and a scalar ϕ_k. Then it calculates an $n \times n$ symmetric matrix M_{k+1} of band-width $2s+3$ and an $n \times n$ orthogonal matrix Ω_{k+1} that satisfy the equation

$$\Omega_{k+1} M_{k+1} \Omega_{k+1}^T = \Omega_k M_k \Omega_k^T + \phi_k \underline{r}_k \underline{r}_k^T, \tag{4.1}$$

in order that $\Omega_k M_k \Omega_k^T$ and $\Omega_{k+1} M_{k+1} \Omega_{k+1}^T$ can be factorizations of the matrices B_k and B_{k+1} in formula (2.2). The other subroutine, which is called "Reduce", is given the matrices Ω_{k+1} and M_{k+1} that have just been mentioned, and it overwrites them by $\Omega^T M_{k+1} \Omega$ and $\Omega_{k+1} \Omega$, respectively, where Ω is a product of Givens rotations such that the new M_{k+1} has band-width $2s+1$. Thus equation (4.1) remains valid. The method of "Update" is the one that uses at most $\frac{1}{4}(n-1)^2/(s+1)$ Givens rotations, and the method of "Reduce" is described in the two paragraphs that begin soon after equation (3.5).

The running times of these subroutines on a Sun 4 workstation were measured for a range of values of n and s. The dependence on s for fixed n deserves attention,

n	s	Total	Full	Ratio
50	20	0.07	0.10	1.43
100	30	0.35	0.74	2.11
200	40	1.51	5.90	3.91
400	60	7.60	48.42	6.37
800	80	36.62	397.48	10.85

Table 3: The dependence of the timings on n

because of its importance to the optimal choice of s. This choice depends also on the running time of "Band-solve", which is the Fortran subroutine that calculates the vector $\Omega_k^T \underline{d}$ from the system of equations (1.8) when M_k is an $n \times n$ symmetric matrix of band-width $2s+1$. Actual timings in seconds for all three subroutines in the case $n = 400$ are shown in Table 2. Further, each entry in the "Total" column is the "Update" time plus the "Reduce" time plus twice the "Band-solve" time, because we take the view that it is usual to solve the system (1.8) for two different values of λ on each iteration. We see that the choice $s = 60$ seems to be the best of the values of s that have been tried. It agrees with the suggestion that s should be of magnitude $n^{2/3}$, but of course the main purpose of the experiments is to investigate the influence of the unknown constant factors that scale the asymptotic estimates of the amounts of work.

Numerical results that are similar to Table 2 were found for several choices of n. Then the value of s for each n was set to the one that gave the least value in the "Total" column. The dependence on n of these values and of the corresponding "Total" times is displayed in the first three columns of Table 3. We see that there is good agreement with the $\mathcal{O}(n^{7/3})$ estimate of the complexity, because $2^{7/3}$ is approximately 5.04. The "Full" column of Table 3 presents the times to solve the system (1.6) twice when B_k is a full $n \times n$ symmetric matrix, and the last column gives the ratios of the "Full" to "Total" timings. Therefore the entries in the last column indicate the reduction that can be achieved in the amount of routine work per iteration by using the $\Omega\, M\, \Omega^T$ factorization technique that has been described. These entries confirm that the reduction factor is of magnitude $n^{2/3}$. Further, it seems to be advantageous to employ the procedures of Section 3 in optimization calculations that have 50 or more variables.

The $\mathcal{O}(n^{7/3})$ complexity of each iteration that is found in Section 3 can be reduced, because that complexity depends strongly on the fact that the procedures so far update the elements of the orthogonal matrix Ω explicitly. Indeed, when A is overwritten by $G\, A\, G^T$, where G is a Givens rotation, then Ω is overwritten by $\Omega\, G^T$, in order that the product $\Omega\, A\, \Omega^T$ is preserved. Now, if Ω is full and if A has band-width $2s+1$, then the calculations of $\Omega\, G^T$ and $G\,A\,G^T$ require $\mathcal{O}(n)$ and $\mathcal{O}(s)$ operations, respectively. Furthermore, if A is augmented by a matrix of rank one, then the work of finding the corresponding rank one matrix for the new A is independent of n. It follows that, if we exclude the updating of Ω from the procedures of Section 3, then, on each iteration, the total work of these procedures is only $\mathcal{O}(n^2)$ operations, which is the product of s and the number of Givens rotations.

Instead of updating Ω explicitly, one can retain all the Givens rotations that occur. Then, rather than overwriting Ω by $\Omega\, G$, it is sufficient to add G to a list of Givens rotations. Let Ω_0 be the orthogonal matrix when the list was started, and let G_i, $i=1,2,\ldots,\ell$, be the entries in the list, where the integer ℓ may become larger than n^2. Thus the current Ω is the product

$$\Omega = \Omega_0 G_1^T G_2^T \cdots G_\ell^T. \tag{4.2}$$

Therefore the vector $\underline{\widehat{g}}_k = \Omega_k^T \underline{g}_k$, which is required for the system of equations (1.8), is computed by forming $\underline{v} = \Omega_0^T \underline{g}_k$, and by overwriting $\underline{v}$ by $G_i \underline{v}$ for $i=1,2,\ldots,\ell$. Furthermore, after deriving a suitable value of λ and the corresponding vector $\Omega_k^T \underline{d}$ from these equations, we obtain the trial step $\underline{d}$ in the following way. We overwrite $\Omega_k^T \underline{d}$ by $G_i^T(\Omega_k^T \underline{d})$ for $i = \ell, \ell-1, \ldots, 1$, which yields the vector $\Omega_0^T \underline{d}$, and it is pre-multiplied by Ω_0. Moreover, the calculation of $\underline{\widehat{r}}_k = \Omega_k^T \underline{r}_k$ from $\underline{r}_k$ is similar to the calculation of $\underline{\widehat{g}}_k$ from $\underline{g}_k$. The complexity of these operations of an iteration is $\mathcal{O}(n^2+\ell)$, the n^2 term being present because there need not be any sparsity in Ω_0.

It follows that the product form (4.2) gives a gain in efficiency only if ℓ is sufficiently small. We have found, however, that about $\frac{3}{4}(n-1)^2/(s+1)$ Givens rotations are introduced by the procedures of Section 3 that make M_{k+1} inherit from M_k the band-width of $(2s+1)$. Therefore we consider the possibility of applying these procedures for only $(t-1)$ consecutive iterations, where t is a prescribed integer, and then we begin a new list of Givens rotations by changing Ω_0. If Ω_0 is replaced at the end of the k-th iteration, we require the matrix $B_{k+1} = B_k + \phi_k \underline{r}_k \underline{r}_k^T$ to be available, because we set Ω_0 to a product of Givens rotations that causes $\Omega_0^T B_{k+1} \Omega_0$ to have band-width $2s+1$. This calculation is done straightforwardly, using one Givens rotation for each zero element that has to be introduced below the diagonal of $\Omega_0^T B_{k+1} \Omega_0$. Therefore the number of rotations is at most $\frac{1}{2}(n-s-1)(n-s)$. Because Ω_0 is formed explicitly, the total work is bounded above by $\mathcal{O}(n^3)$, which is tolerable for sufficiently large t. We make B_{k+1} available by applying formula (2.2) for every k, which adds another $\mathcal{O}(n^2)$ operations to each iteration.

By replacing Ω_0 every t iterations, we restrict the value of ℓ in the product (4.2) to $\mathcal{O}(n^2 t/s)$. Therefore, by sharing the $\mathcal{O}(n^3)$ work of the replacement over t iterations, we find that on average an iteration has the complexity

$$\mathcal{O}(n^2 + n^2 t/s + n^3/t + n\, s^2), \tag{4.3}$$

where the last term comes from the solution of the banded system of equations (1.8). The magnitudes of s and t that minimize this expression are $n^{3/5}$ and $n^{4/5}$, respectively. Thus the average work per iteration is only $\mathcal{O}(n^{11/5})$, and the amount of storage for the list of Givens rotations is also of this magnitude. Therefore we have improved on the $\mathcal{O}(n^{7/3})$ complexity that is noted at the end of Section 3.

Finally, we address the importance of our conclusions to practical optimization calculations. If there is no useful structure in the second derivatives of the objective function, then, by allowing more variables, the given procedure may increase substantially the range of unconstrained optimization problems that can be solved by trust region algorithms. On the other hand, it is usual for there to be some structure when n is large. One can be optimistic about the practical value of our results, however, by taking the view that the identification and exploitation of structure may demand much effort from the computer user. Therefore our procedures may be particularly valuable in cases when the user would prefer not to consider the structure of his or her problem.

Acknowledgement
I am very grateful to M. Andramonov for the attention that he gave to this research when he visited Cambridge in 1994. I also thank M.R. Osborne, B.N. Parlett and J.D. Pryce for helpful discussions on the number of Givens similarity transformations that are necessary to decrease the band-width of a symmetric matrix.

References

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

J.J. Moré (1983), "Recent developments in algorithms and software for trust region methods", in *Mathematical Programming The State of the Art*, eds. A. Bachem, M. Grötschel and B. Korte, Springer–Verlag (Berlin), pp. 258–287.

J.J. Moré and D.C. Sorensen (1983), "Computing a trust region step", *SIAM J. Sci. Stat. Comput.*, Vol. 4, pp. 553–572.

T. Steihaug (1983), "The conjugate gradient method and trust regions in large scale optimization", *SIAM J. Numer. Anal.*, Vol. 20, pp. 626–637.

D.E. Stewart (1997), "A graph-theoretic model of symmetric Givens operations and its implications", *Linear Algebra Appl.*, Vol. 257, pp. 311-320.

J.H. Wilkinson (1965), *The Algebraic Eigenvalue Problem*, Oxford University Press (Oxford).