æ

The Solution of the Metric STRESS and SSTRESS Problems in Multidimensional Scaling Using Newton's Method

Anthony J. Kearsley* Richard A. Tapia† Michael W. Trosset‡

October 7, 1995

Abstract

This paper considers numerical algorithms for finding local minimizers of metric multidimensional scaling problems. Both the STRESS and SSTRESS criteria are considered, and the leading algorithms for each are carefully explicated. A new algorithm, based on Newton's method, is proposed. Translational and rotational indeterminancy is removed by a parametrization that has not previously been used in multidimensional scaling algorithms. In contrast to previous algorithms, a very pleasant feature of the new algorithm is that it can be used with either the STRESS or the SSTRESS criterion. Numerical results are presented.

Key words: Metric multidimensional scaling, STRESS criterion, SSTRESS criterion, unconstrained optimization, Newton's method.

*Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892. This author was generously supported by a Patricia R. Harris Fellowship.

†Department of Computational and Applied Mathematics and Center for Research in Parallel Computation, Rice University. This author was supported in part by NSF Coop. Agr. No. CCR-8809615, AFOSR 89-0363, DOE DEFG05-86ER25017, and ARO 9DAAL03-90-G-0093.

‡Department of Computational and Applied Mathematics (adjunct), Rice University; Department of Psychology (adjunct), University of Arizona; and Consultant, P. O. Box 40993, Tucson, AZ 85717-0993. This author was supported in part by NSF Coop. Agr. No. CCR-8809615, as a visiting member of the Center for Research in Parallel Computation, Rice University, August 1993 and 1994.

1 Introduction

Multidimensional scaling (MDS) is a general term for a vast collection of data analytic techniques. As defined by de Leeuw and Heiser [10], *scaling* refers to techniques that construct a configuration of points in a target metric space from information about interpoint distances, and MDS is scaling in the case that the target space is Euclidean. Kruskal and Wish [27] provided an elementary introduction to basic MDS methodology, as well as many enlightening examples.

The present paper addresses two very specific, but very important problems in MDS. As in classical MDS [39, 40, 17], two assumptions are made about the nature of the information provided about the interpoint distances. Formally, a symmetric $n \times n$ matrix $\Delta = (\delta_{ij})$ is called a dissimilarity matrix if $\delta_{ij} \geq 0$ (nonnegative elements) and $\delta_{ii} = 0$ (zero diagonal elements). From a given dissimilarity matrix Δ, a *two-way* MDS algorithm constructs a configuration of points in a Euclidean space of specified dimension p. For a configuration $x_1, \ldots, x_n \in R^p$, the $n \times p$ configuration matrix X is the matrix whose rows are the $x_i^t, i = 1, \ldots, n$. From X it is easy to compute the Euclidean interpoint distance matrix $D(X) = (d_{ij})$. The objective of two-way MDS is to construct a configuration for which the interpoint distances d_{ij} somehow approximate the given dissimilarities δ_{ij}.

Historically, MDS techniques that minimize some measure of discrepancy between the configuration's interpoint distances and the given dissimilarities δ_{ij} have been termed *metric*. In contrast, *nonmetric* techniques minimize some measure of discrepancy between the interpoint distances and a set of dissimilarity matrices whose elements have the same rank ordering as the given δ_{ij}. Early MDS techniques, e.g. the methods of Torgerson [39], were exclusively metric. However, since the pioneering work of Shepard [34, 35] and Kruskal [25, 26], the psychometric and statistical communities have tended to emphasize nonmetric MDS. Nevertheless, metric MDS has remained critically important, because solving nonmetric MDS problems typically involves repeatedly solving metric MDS subproblems. In recent years, there has been renewed interest, e.g. by de Leeuw [7], in developing efficient methods for solving these subproblems. Furthermore, in the last decade, techniques related to MDS have been studied by computational chemists, e.g. Crippen and Havel [5], interested in deducing molecular structure from information about interatomic distances. Because procedures such as NMR spectroscopy do not distort distances in the nonlinear ways that human perceptions of psychophysical phenomena typically do, it is metric MDS that is of interest in this context. Thus, the study of metric MDS remains of fundamental importance.

The classical metric approach of Torgerson [39] having fallen from favor, most modern formulations of metric MDS entail the minimization of one of two measures of the discrepancy between distances and dissimilarities. The STRESS criterion, proposed by Kruskal [25] for nonmetric MDS, is based on the squared errors between the distances and the dissimilarities. The SSTRESS criterion, popularized by Takane, Young, and de Leeuw [37] for nonmetric MDS, is based on the squared errors between the squared distances and the squared dissimilarities. Thus, both the metric STRESS ($r = 1/2$) and SSTRESS ($r = 1$) problems are special cases of the following constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i<j} w_{ij} \left[\left(d_{ij}^2\right)^r - \left(\delta_{ij}^2\right)^r \right]^2 \\ \text{subject to} & D \in \mathcal{D}_n(p), \end{array} \tag{1}$$

where the w_{ij} are nonnegative weights and $\mathcal{D}_n(p)$ is the set of all $n \times n$ matrices whose elements can be realized as the interpoint distances of n points in R^p. In practice, one often sets each $w_{ij} = 1$; however, one can use the weights either to accomodate missing data (by setting the appropriate $w_{ij} = 0$) or to weight more reliably measured dissimilarities more heavily. In most applications, the dimension of the configuration space is small; in the case of molecular conformation, of course, one always sets $p = 3$.

The purpose of the present paper is to describe an implementation of Newton's method for the efficient solution of Problem (1) in the special cases $r = 1$ and $r = 1/2$. In Section 2 we discuss several fundamental concepts in metric MDS and numerical analysis. This section provides the necessary background for our review of the leading metric MDS algorithms (Section 3) and for our description of a more efficient algorithm (Section 4). We present numerical results in Section 5 and assess what we have accomplished in Section 6.

2 General Considerations

Because it is not obvious how to write the constraint $D \in \mathcal{D}_n(p)$ in Problem (1) as a standard equality or inequality constraint, it cannot be managed by the standard techniques of mathematical programming. Therefore, virtually all treatments of MDS problems employing either STRESS or SSTRESS parametrize the distances by expressing them in terms of the configuration coordinates, i.e. by writing

$$d_{ij}^2 = d_{ij}^2(X) = \sum_{k=1}^{p}(x_{ik} - x_{jk})^2. \tag{2}$$

Substituting (2) into (1) eliminates the constraint but complicates the objective function, resulting in the unconstrained optimization problem

$$\text{minimize} \quad \sum_{i<j} w_{ij} \left[\left(\sum_{k=1}^{p}(x_{ik} - x_{jk})^2 \right)^r - \left(\delta_{ij}^2 \right)^r \right]^2. \tag{3}$$

Henceforth, we restrict attention to this parametrization of the metric STRESS and SSTRESS problems.

The remainder of this section discusses several fundamental issues from the theory and practice of unconstrained optimization that are germane to the efficient solution of Problem (3). We believe that an appreciation of these issues is essential to our critique of previous algorithms (Section 3) and our presentation of a new algorithm (Section 4). More detailed discussions of these issues can be found in the well-known book by Dennis and Schnabel [12].

Let us begin by noting the distinction between local and global minimizers. Ideally, we would like to find a global minimizer of Problem (3). Unfortunately, whereas the science of local optimization is highly advanced, the science of global optimization is still in its infancy. To date, all of the important methods proposed for metric MDS have been iterative algorithms for finding local minimizers. In this paper, we are content to improve on these algorithms. Typically, one attempts to find a "good" local minimizer by finding a good initial configuration from which to start iterating. It has also been suggested that global minimizers might be more readily obtained by exploiting the geometry of the cone of distance matrices (Glunt, Hayden, and Liu [14] demonstrated that all of the local minimizers of the metric SSTRESS problem lie on the surface of a sphere, the global minimizer being the one of greatest norm) or by parametrizing the configuration in a larger (than p) number of dimensions. Thus far, effective algorithms based on these ideas have not been forthcoming. Finally, several researchers have applied global optimization methods to the metric STRESS problem. The modularized DG-II package of Havel [20] attempts to improve on a local minimizer by the use of simulated annealing. A detailed study by Groenen [18] suggests that tunneling methods are preferable to simulated annealing.

In the modern theory of computational optimization, Newton's method continues to be the method of choice for finding local solutions of unconstrained optimization problems. Under well-known standard assumptions about smoothness and nonsingularity, it is not difficult to establish local and fast (quadratic) convergence of the method. Critics of Newton's method direct their comments to the need for calculating second derivatives, the need for solving a linear system of equations at each iteration, the implied need for smoothness and nonsingularity, and the implied nonglobal convergence. While the general theory is sharp, and all of the above may be valid criticisms across the full spectrum of applications, experience has shown that, in a particular application, not all of these criticisms need apply or be restrictive. What is at issue in the present paper is the applicability of these criticisms to the metric STRESS and SSTRESS problems of MDS.

Users of optimization algorithms are often (and understandably) confused by the distinction between local and global convergence. What is at issue here is *not* the type of minimizer, but where the algorithm must start in order that convergence to a local minimizer be guaranteed. Theory tells us that, if Newton's method starts sufficiently near a local solution, then the sequence of iterates will rapidly converge to that solution. However, this property does not preclude the possibility of choosing a starting point from which the sequence of iterates may fail to converge to any local minimizer. Modifications of locally convergent methods that eliminate this undesirable possibility are called *globalization strategies*, and algorithms that are guaranteed to converge to a local minimizer from (essentially) every starting point are called *globally convergent.*

Actually, decades of experience have demonstrated that the semi-local properties of Newton's method are usually quite good — much better, in fact, than the theory predicts. Convergence and fast convergence are usually not restricted to a very small neighborhood of the solution, as many vendors of awkward hybrid methods would have us believe. Nevertheless, Newton's method *per se* is not globally convergent. Two fundamental globalization strategies for Newton's method, line searches and trust regions, are discussed in Chapter 6 of Dennis and Schnabel [12]. Both are based on the idea that each step taken by the algorithm should decrease the value of the objective function.

Extensive experience with Newton's method has also demonstrated that damping the Newton step, i.e. choosing step length less than one, often improves the global behavior of Newton's method. However, not choosing step length one locally may preclude the fast convergence. Concern for these two aspects of Newton's method has resulted in the so-called back-tracking line search strategies, in which one always considers the full Newton step before damping, and one implements damping in a manner that ensures the full Newton step near the solution.

Line search strategies retain the Newton step direction but shorten its length. In contrast, trust region strategies constrain the step length but allow other choices of the step direction. This is accomplished by searching for steps that minimize a quadratic approximation to the objective function at the current iterate, subject to an upper bound on the step length. Because there is no finite way of exactly solving this quadratic subproblem, various approximate solutions have been suggested. The most commonly used are the "hook" step of Hebden [21] and Moré [29], and the "double dogleg" step due to Powell [32].

In general, the presence of a singular Hessian matrix at a solution dramatically slows — and may even preclude — the local convergence of Newton's method. Moreover, if the solutions are not isolated, then the Hessian matrix is necessarily singular at a solution. For this reason, we prefer problems with isolated solutions. Notice, however, that the objective function in Problem 3 is invariant under isometric transformations of the configuration, so that *every* minimizer belongs to a connected set of minimizers. What has happened is that the reparametrization from (1) to (3) introduced a considerable amount of redundancy. To develop an efficient algorithm for solving Problem (3), it is desirable to remove this redundancy. As we shall see in Section 3, different researchers have addressed this need in different ways.

Finally, we consider the computational issues of calculating second derivatives and solving systems of linear equations at each iteration of Newton's method. A critical issue in deciding to use the method is the viability of performing these computations. It is certainly naive to believe that this viability can be determined by looking at only one iteration. Rather, the complete picture must be considered. Discarding Newton's method in favor of an algorithm that produces cheap iterates is of no value if the number of iterations needed to solve the problem is prohibitively large. This is often the case for gradient methods, unless only a very limited amount of accuracy is needed. It follows that whether or not Newton's method will be successsful in a particular application can only be decided by careful study of the application, in conjunction with careful numerical experimentation.

3 Previous Algorithms

We now review the most important of the algorithms that have been proposed for metric MDS. Historically, researchers have developed completely different algorithms for the metric STRESS and SSTRESS problems. In contrast, we believe that one of the attractive features of our approach is that we have developed a single algorithm that works well on both problems.

Our primary emphasis in this section is on the numerical algorithms that have been used to generate sequences of iterates. However, we are also concerned with the approaches that different researchers have taken to isolate local minimizers, and with the devices that they have used to obtain starting points for their algorithms.

3.1 STRESS

The STRESS criterion is the objective function in Problem (3) obtained by setting $r = 1/2$. Let us write the STRESS criterion as

$$\sigma(X) = \sum_{i<j} w_{ij} \left[\left(\sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^{1/2} - \delta_{ij} \right]^2 .$$

Kruskal [26, 24] proposed minimizing $\sigma(X)$ by an *ad hoc* gradient method in which the step length is determined by the angle between the present and preceding gradients. Guttman [19] observed that the stationary equation $\nabla\sigma(X) = 0$ can be written as $X = C(X)X$, where the matrix-valued function C depends on the dissimilarity matrix Δ, and suggested that the sequence defined by the *Guttman transform* $X^{k+1} = C(X^k)X^k$ should converge to a stationary point of σ. This turned out to be essentially correct, and all of the algorithms for which convergence has been demonstrated are based on this idea.

The first rigorous analysis of convergence was supplied by de Leeuw [6] and elaborated upon by de Leeuw and Heiser [9] and de Leeuw [7]. Despite the fact that σ is not everywhere differentiable (because the square root function is not differentiable at 0, σ is not differentiable at X if some $d_{ij}(X) = 0$, i.e. if some points in the configuration coalesce), the Guttman sequence is globally convergent to a connected set of local minimizers. In fact, de Leeuw (1984) [8] proved that $\sigma(X)$ *is* differentiable at all local minimizers, so that differentiability can be assumed for local convergence analysis. (Note that a consequence of this fact is that points cannot coalesce in optimal STRESS configurations.) The first such analysis was undertaken by de Leeuw [7], who concluded that "in almost all cases convergence is linear, with a convergence [constant] close to unity." (p. 163). This analysis explains the empirically observed fact that convergence of MDS algorithms is usually very slow.

To establish convergence, de Leeuw [7] assumed that the configurations were centered at the origin. (This can be ensured by starting with a centered configuration, since this property is preserved by the Guttman transform.) This assumption removes some, but not all, of the isometric indeterminacy that characterizes Problem (3). De Leeuw subsequently observed that

> "The discussion in the previous sections shows that at least part of the difficulty with proving actual convergence of our iterations comes from the rotational indeterminacy of multidimensional scaling. ...If we eliminate rotational indeterminacy, then we eliminate these difficulties." (p. 175).

If rotational indeterminacy is eliminated (de Leeuw suggested rotating to principal components), then local minimizers may be isolated and de Leeuw obtained the rate at which the Guttman sequence converges to an isolated local minimizer.

Actually, there were other reasons to anticipate the linear convergence of the Guttman sequence. If differentiability of $\sigma(X)$ is assumed, then it is well-known that the sequence can be written as the iterates of a weighted gradient algorithm, $X^{k+1} = X^k - (1/2)V^+\nabla\sigma(X^k)$, for a certain fixed matrix V^+. In contrast, the method of steepest descent is a gradient algorithm that produces iterates of the form $X^{k+1} = X^k - \alpha_k\nabla\sigma(X^k)$, where α_k is a (positive) real number. It is generally understood that gradient algorithms, which do not exploit information about second derivative behavior, typically exhibit linear rates of convergence.

Despite the slow convergence of gradient methods, their use has been strongly emphasized in the MDS literature. For example, Kruskal [24] distinguished between unconstrained optimization methods whose memory requirements are linear (Class 1, e.g. gradient methods) and more than linear (Class 2, e.g. Newton's method) in the number of variables, and commented:

> "Although Classes 1 and 2 have both been used in this field, Class 1 has been used much more often. In addition to the high cost of memory during computing, this may be due to the fact that high accuracy solutions are almost never needed in this field due to the substantial random error which we typically find in the data. Since the solution is meaningful only up to a certain level due to random error in the input, there is no need to obtain a solution which is accurate to a much higher level. Hence the higher speed of convergence for Class 2 methods does not have so great an attraction." (p. 315).

A more modern assessment of these issues is long overdue. First, both computers and computational mathematics have advanced enormously in the last seventeen years. Second, the problems of molecular conformation are very different from the problems of psychology. The number of objects is typically much greater (a protein molecule may contain thousands of atoms), and the dissimilarities are typically much more accurate.

One fairly conservative possibility for accelerating the convergence of the Guttman sequence is to modify the step choice in the corresponding gradient algorithm. This is precisely the motivation for Kruskal's [26, 24] angle-dependent gradient method. De Leeuw and Heiser [9] presented a simple device that "approximately halves the number of iterations required to obtain a given precision, at no extra cost." (p. 513). De Leeuw (1988) [7] suggested that this improvement was what could generally be expected from such modifications, and concluded that "one should always study the second derivatives of the loss function at the stopping point of the algorithm." (p. 179).

In Sections 4 and 5, we will argue that it is desirable to study the second derivative at each iteration of the algorithm. The present aversion to so doing appears to be largely due to the perception that it is a prohibitively expensive course of action. Thus, recent advances have attempted compromises. One such compromise is the gradient method of Barzilai and Borwein [1], which determines step length using a Rayleigh quotient that approximates an eigenvalue of the Hessian matrix. Local convergence properties of this method were established by Raydan [33].

Glunt, Hayden, and Raydan [15] applied the Barzilai and Borwein method to the problem of minimizing $\sigma(X)$. The authors assumed that the starting configuration is centered, in which case all subsequent configurations are necessarily centered; however, they did not attempt to remove rotational indeterminacy. They refer to this new algorithm as the spectral gradient method, and they found that its use decreased the cpu time required by de Leeuw's [7] implementation of the Guttman sequence (which they called the *majorization algorithm*) by a factor of 10–20. Even greater acceleration is obtained with use of a preconditioner (Glunt, Hayden, and Raydan [16]).

In its present form, the spectral gradient algorithm has faster local convergence than the majorization algorithm, but it sacrifices global convergence. (Raydan is presently attempting to develop an effective globalization strategy for the Barzilai and Borwein method.) The lack of global convergence requires construction of a good starting configuration; however, as Glunt, Hayden, and Raydan [15] point out, even globally convergent methods require good starting configurations to obtain "good" local minimizers. Because their technique for constructing a starting configuration for the spectral gradient method involves solving a metric SSTRESS problem, and is actually the same technique employed by Glunt, Hayden, and Liu [14], we defer discussing it until Section 3.2. Virtually all of the methods that have been proposed in the MDS literature for constructing starting configurations involve solving an MDS problem that is presumed to be easier than whichever one is actually under investigation.

3.2 SSTRESS

The SSTRESS criterion is the objective function in Problem (3) obtained by setting $r = 1$. Computationally, it is considerably more pleasant than the STRESS criterion (in particular, it is everywhere smooth), and was in fact proposed for this reason. In consequence, researchers have been somewhat less reluctant to use second order methods on the metric SSTRESS problem than on the metric STRESS problem.

For the special case of dimension $p = n$, an extensive analysis of the metric SSTRESS problem was provided by Glunt, Hayden, Hong, and Wells [13]. In this case, the dimension of the solution is not constrained, the constraint set $\mathcal{D}_n(p)$ is convex, and a global minimizer of SSTRESS in the unweighted case can be obtained using the authors' Modified Alternating Projection (MAP) algorithm. Of course, the solution is typically of very high dimension. Because most applications require a low-dimensional configuration, e.g. $p = 2, 3$, the configuration constructed by the MAP algorithm is not of much interest *per se*.

Until recently, the best algorithm for the case of dimension $p < n - 1$ was the one proposed by Browne [4]. To understand the basis for this algorithm, it is necessary to briefly digress and consider the classical metric scaling technique of Torgerson [39].

Given a dissimilarity matrix Δ, let $\Delta * \Delta$ denote the Hadamard product of Δ with itself, so that the elements of $\Delta * \Delta$ are the squared dissimilarities. Let $\tau : A \to B$ denote the linear operator defined by

$$b_{ij} = -\frac{1}{2}(a_{ij} - \bar{a}_{i\cdot} - \bar{a}_{\cdot j} + \bar{a}_{\cdot\cdot}),$$

often called "double centering" by psychometricians. Then a well known embedding theorem from classical distance geometry states that an $n \times p$ configuration matrix X satisfies $D(X)$ if and only if $\tau(\Delta * \Delta) = XX^t$. Thus, the cone of distance matrices is parametrized by the cone of positive semidefinite matrices of rank $\leq p$, and a configuration matrix can be obtained from a positive semidefinite matrix by factorization.

In case Δ is not a distance matrix, Torgerson [39] proposed solving the optimization problem

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}\left[(B - \tau(\Delta * \Delta))^2\right] \\ \text{subject to} & B \in \Omega_n(p). \end{array} \tag{4}$$

The objective function in this problem was subsequently dubbed STRAIN, making Problem (4) the metric STRAIN problem. The metric STRAIN problem has the very attractive feature that one can compute an explicit global solution B^*. Let

$$\lambda_1 \geq \cdots \geq \lambda_n$$

denote the eigenvalues of $\tau(\Delta * \Delta)$, let $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, and let $U \Lambda U^t$ be the spectral decomposition of $\tau(\Delta * \Delta)$. Define $\bar{\Lambda}$ by $\bar{\lambda}_i = \max(\lambda_i, 0)$ for $i = 1, \ldots, p$ and $\bar{\lambda}_i = 0$ for $i = p+1, \ldots, n$. Then $B^* = U \bar{\Lambda} U^t$. For further details about the metric STRAIN problem (and the embedding result on which it is based), see the review article by Trosset [41].

We now return to Browne's [4] algorithm for the metric SSTRESS problem. Given a dissimilarity matrix Δ, the SSTRESS criterion in the unweighted case is

$$f(X) = \operatorname{tr}\left[(\Delta * \Delta - D(X) * D(X))^2\right].$$

To eliminate isometric indeterminacy, Browne introduced a "duplicate" configuration matrix Y and penalized X for departures from Y. The discrepancy between X and Y was measured using the STRAIN criterion. Thus, for a constant "deceleration" scalar $\alpha \in (0, 1]$, Browne proposed minimizing the penalized objective function

$$f^*(X, Y) = f(X) + 4\frac{1-\alpha}{\alpha} \operatorname{tr}\left[\left(YY^t - XX^t\right)^2\right]. \tag{5}$$

Browne suggested choosing the initial X configuration to be the solution to the metric STRAIN problem. His algorithm for minimizing (5) involves alternately minimizing $f^*(X_n, Y)$ for X_n fixed to obtain $Y_n = X_n$, then minimizing $f^*(X, Y_n)$ for Y_n fixed to obtain X_{n+1}. This is the method of variable alternation for reducible nonlinear programming; in the context of MDS, it is usually called the method of alternating least squares (ALS).

To accomplish the nontrivial minimization subproblem in the ALS formulation, Browne's algorithm uses Newton's method. Because this fact has been emphasized in the MDS literature, we stress that Browne's algorithm is *not* equivalent to applying Newton's method to the (unweighted) metric SSTRESS problem. In fact, when ALS converges, it usually does so at only a linear rate.

The superiority of Browne's algorithm over its predecessors was described by Glunt, Hayden, and Liu [14], who stated:

> "A number of algorithms have been proposed for the solution of the [metric SSTRESS] problem by researchers in multidimensional scaling. An algorithm due to de Leeuw and Takane was modified by Browne (1987) by adding a Newton Raphson step (henceforth called the NR method) and resulted in the best algorithm known to us for finding a local minimum solution of the [metric SSTRESS] problem. [NR] either finds the global minimum (about 90% of the time in our examples) or a local minimum with objective function near the global minimum. Furthermore, NR is orders of magnitude faster than competing algorithms and hence NR is our current yardstick for measuring success." (p. 770).

Measured by the Browne yardstick, the algorithm subsequently proposed by Glunt, Hayden, and Liu [14] is extremely impressive. In its first phase, the MAP algorithm is used to (approximately) solve the metric SSTRESS problem with $p = n$. (Glunt, Hayden, Hong, and Wells [13] had found that MAP was approximately four times faster than Browne's algorithm for solving this problem.) The MAP solution is then used to produce a starting point for the second phase, in which a penalty term is added to the objective function (to remove translation invariance) and a local minimizer is obtained by use of "a standard (say quasi-Newton) unconstrained nonlinear optimization routine, with analytic gradient computed by ...." (p. 788). Glunt, Hayden, and Liu found that this "two-phase" algorithm was approximately ten times faster than Browne's algorithm.

Let us make several observations about the algorithm of Glunt, Hayden, and Liu [14]. First, it is not at all clear that the expense of using the MAP algorithm in the first phase is justified. Let Δ^0 denote the given dissimilarity matrix and let $D^0 \in \mathcal{D}_n(p)$ denote the distance matrix obtained from the MAP algorithm. Then Glunt, Hayden, and Liu (and also Glunt, Hayden, and Raydan [15] for the metric STRESS problem; see Section 3.1) chose, as a starting configuration matrix X^0, a configuration that solves the metric STRAIN problem with $\Delta = D^0$. We see no obvious reason why this should be superior to Browne's [4] considerably less expensive choice of a configuration that solves the metric STRAIN problem with $\Delta = \Delta^0$, the original dissimilarity matrix.

Second, although Glunt, Hayden, and Liu [14] were quite careful to remove translation invariance, their penalty function does not remove rotational invariance. This was overlooked by the authors, who mistakenly believed that the Hessian matrix of their objective function is necessarily positive definite at local minimizers. Of course, it is not difficult to modify the penalty function so that rotational invariance is also removed. This was done, for example, by Tarazaga and Trosset [38], who used the parametrization $B = XX^t$ to study optimization problems defined on the set of symmetric positive semidefinite matrices of rank $\leq p$. The difficulty with this entire approach, however, is that it demands a great deal of the penalty function. In practice, devices of this sort tend to lead to a deterioration of the local behavior of the algorithm.

Finally, although Glunt, Hayden, and Liu [14] emphasized their use of the analytic gradient vector, they declined to use the analytic Hessian matrix. In fact, they opined that "The formula for the Hessian appears too complicated to be computationally helpful in the general case." (p. 779). This was also the opinion of Glunt, Hayden, and Raydan [15] with regard to the metric STRESS problem. As we have already remarked, however, extensive numerical experimentation is usually required to determine whether or not it pays to compute second derivatives. We now consider an algorithm that, it turns out, makes extremely efficient use of such information.

4 A New Algorithm

We begin by writing Problem (3) as a standard nonlinear least squares problem. To do this, let $m = n(n-1)/2$ denote the number of interpoint dissimilarities (or distances), define $G_r : R^{n\times p} \to R^m$ to have component functions of the form

$$\left(\sum_{k=1}^{p}(x_{ik} - x_{jk})^2\right)^r - \left(\delta_{ij}^2\right)^r$$

for $i < j$, and let $W \in R^{m\times m}$ be the diagonal matrix whose diagonal elements are the weights w_{ij} associated with the corresponding interpoint dissimilarities δ_{ij}. Then Problem (3) can be written as the nonlinear least squares problem

$$\text{minimize} \quad f_r(X) = \tfrac{1}{2}G_r(X)^t W G_r(X). \tag{6}$$

In Section 2, we remarked that the objective function $f_r(X)$ is invariant under isometric transformations of the configuration matrix X. For reasons discussed there, we want to remove the unnecessary degrees of freedom that result from translational and rotational invariance. As illustrated by each of the methods reviewed in Section 3, there is a long tradition in MDS of removing translational invariance by centering the configuration at the origin. Rotational invariance has variously been removed by expensive procedures such as principal component analysis or simply ignored. We propose to abandon these traditions and make use of an elementary device that has so far been neglected by MDS researchers.

In an early article concerned with finding molecular configurations that minimize the Lennard-Jones potential energy function, Hoare and Pal [23] described an elementary way of removing translational and rotational invariance in R^3. One simply constrains one point in the configuration to lie at the origin, specifies two coordinate axes, constrains a second point to lie along the first specified axis, and constrains a third point to lie in the plane determined by the two axes. For example, for $X \in R^{n\times 3}$, we parametrized Problem

(6) by setting

$$X = \begin{bmatrix} 0 & 0 & 0 \\ x_1 & 0 & 0 \\ x_2 & x_3 & 0 \\ x_4 & x_5 & x_6 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}. \tag{7}$$

Unless the selected points are collinear, this parametrization removes the isometric invariance of $f_r(X)$. In general, one selects p points, fixing $p-k+1$ coordinates of point k, for $k = 1, \ldots, p$. So long as the selected points are not contained in a subspace of dimension $p-2$, this parametrization removes the isometric invariance of $f_r(X)$.

Most of our numerical experiments have been performed for $X \in R^{n\times 3}$. In these experiments, we have found that collinearity of the selected points is very rarely encountered. In theory, of course, it may be that the three selected points are collinear in the minimizing configuration. One way of dealing with this possibility is to reparametrize, using a different triple of points, whenever the original triple is nearly collinear. If one can afford the expense of computing the metric STRAIN solution, then one can examine it to discover a triple that is almost certainly not collinear. (This also allows one to determine whether or not the dissimilarity matrix actually is a distance matrix of dimension less than p, e.g. *all* points collinear, is available.) For the remainder of this paper, we assume that the points have been labelled in such a way that we are fixing the indicated coordinates of the first p points.

The parametrization defined by (7) appears to be quite standard in the literature on minimizing the energy of molecular configurations. For example, Northby [31] used it quite casually to preclude translation or rotation of the configuration. We are somewhat at a loss to explain why this parametrization is so rarely used in MDS. (The only example of which we are aware is by Groenen [18], who exploited it for *global* optimization of the metric STRESS problem by a multi-level single-linkage clustering algorithm.) One possibility is that MDS researchers have been unduly influenced by Torgerson's [39] formulation of the metric STRAIN problem. Young's and Householder's [43] original solution of the embedding problem (the case in which the dissimilarity matrix actually is a distance matrix) placed the nth point of the configuration at the origin. Torgerson observed that, with "fallible data," different solutions are obtained according to which point is labelled the nth. As an antidote, he introduced the double centering operator that we have denoted by τ, which leads to configurations that are centered at the origin. Thus, for the metric STRAIN problem, configurations are centered in order to specify a solution that does not depend on the indexing of the points. However, it is quite clear that solutions to Problems (1), (3), and (6) do not depend on the indexing of the points. Hence, there is no substantive reason to require that solutions to the metric STRESS or SSTRESS problems be centered at the origin. Of course, if a centered solution is desired for aesthetic reasons, then one can always translate a noncentered solution after it has been obtained.

The parametrization defined by (7) reduces the number of free variables in Problem (6), from np to $N = np-p(p+1)/2$. Thus, the simplified problem has objective function $f_r : R^N \to R$, with $G_r : R^N \to R^m$. Without this reduction, f_r is invariant under isometric transformations, a consequence of which is that the Hessian matrix is singular at every point in the domain. (It should be noted, of course, that this reduction does not guarantee that the Hessian matrix will not be singular.) In particular, without this reduction it is impossible for the Hessian matrix to be positive definite at solutions of Problem (6).

The simple structure of f_r makes it a straightforward matter to derive analytic expressions for ∇f_r and $\nabla^2 f_r$. These expressions are not expensive to evaluate; in fact, f_r, ∇f_r, and $\nabla^2 f_r$ can be computed in $O(N)$ floating point operations. Using this analytic information results in algorithms with smaller truncation errors and better stability properties — see Boggs and Dennis [3] for an error analysis.

Because Problem (6) is a nonlinear least squares problem, it can be solved reasonably efficiently by applying any good general nonlinear least squares algorithm. Such algorithms have been developed by Moré, Garbow, and Hillstrom [30], by Dennis, Gay, and Welsch [11], and by Boggs, Byrd, Donaldson, and Schnabel [2]. These algorithms, however, incorporate conservative precautions and very delicate globalization strategies designed for highly nonlinear problems. In contrast, the nonlinearity of the residual functions in Problem (6) is very mild. When the above algorithms are applied to mildly nonlinear problems, they tend to require more computations and result in longer run times than are really necessary. Therefore, it makes sense to develop a more specialized algorithm for Problem (6).

It is important to appreciate that the Hessian matrix $\nabla^2 f_r(X)$ is very dense. In many applications, it is considerably less expensive to compute an alternative matrix, $\nabla G_r(X) \cdot \nabla G_r(X)^t$. This substitution leads to the *Gauss-Newton* algorithms, which are particularly effective on so-called "small residual" problems. See Moré [29] for a complete analysis of the well-known Levenberg-Marquardt version of this algorithm.

In numerical analysis, a small residual problem is one for which the value of the objective function at the solution is small compared to a typical value of the objective function. For example, $f_r(X^*) << f_r(X^0)$ would be suggestive of a small residual problem. In this relative sense, Problem (6) is typically a small residual problem. However, the availability and density of the Hessian matrix render negligible the relative savings from using $\nabla G_r(X) \cdot \nabla G_r(X)^t$ instead of $\nabla^2 f_r(X)$. In fact, for very large problems, the zero-nonzero structure of $\nabla G_r(X) \in R^{N \times m}$ is quite difficult to store and multiply, whereas the dense but much smaller matrix $\nabla^2 f_r(X) \in R^{N \times N}$ is easy to store and manipulate. In this way, Problem (6) differs from most small residual nonlinear least square problems.

All of the above considerations combine to suggest the viability of simply applying Newton's method to Problem (6). The specific algorithm that we implemented can be summarized as follows, where $G_r^i : R^N \to R$ denotes the ith component function of f_r and I denotes the $N \times N$ identity matrix. To facilitate implementation, we indicate relevant sections of Dennis and Schnabel [12]. Their Appendix A provides pseudocode for these modules.

Globalized Newton's Method Algorithm

1. Construct an initial configuration X^0.

2. Specify the convergence tolerances and a radius $\mu^0 > 0$ for the initial model trust region.

3. Compute the gradient $\nabla f_r(X^k)$ and the Hessian
$$H^k = \sum_{i=1}^{m} \left[\nabla G_r^i(X^k) \cdot \nabla G_r^i(X^k)^t + G_r^i(X^k) \cdot \nabla^2 G_r^i(X^k) \right].$$

4. Determine the step direction s^k by minimizing the local quadratic model for f_r in the model trust region. This is equivalent to solving the system of equations $(H^k + \mu^k I)s^k = -\nabla f_r(X^k)$. See Dennis and Schnabel [12], Section 6.4. Because there is no finite method for exactly solving this subproblem, we settle for an approximate solution. Our preference is for the hook step solution of Hebden [21] and Moré [29]. See Dennis and Schnabel [12], Section 6.4.1 and Algorithms A6.4.1 and A6.4.2.

5. Determine the step length α^k by a backtracking line search. See Dennis and Schnabel [12], Section 6.3.2 and Algorithm A6.3.1.

6. Set $X^{k+1} \leftarrow X^k + \alpha^k s^k$.

7. Check convergence. See Dennis and Schnabel [12], Section 7.2 and Algorithm A7.2.1.

8. Update the radius of the model trust region, i.e. compute μ^{k+1}. See Dennis and Schnabel [12], Section 6.4.3 and Algorithm A6.4.5.

9. Set $k \leftarrow k + 1$ and go to Step 3.

We investigated two strategies for constructing the initial configuration X^0. These strategies represent two possible compromises in the unavoidable tradeoff between the quality of $f_r(X^0)$ and the expense of computing X^0. Historically, MDS researchers have eschewed inexpensive initial configurations and have attempted to obtain initial configurations of high quality. The primary reason for this is concern about local minimizers — it is certainly plausible that better initial configurations will allow the algorithm to find better local minimizers.

In this spirit, our first strategy for constructing an initial configuration was to compute (a translation/rotation of) the metric STRAIN solution. As described in Section 3.2, this is precisely how Browne [4] constructed initial configurations and is very similar to the strategy employed by Glunt, Hayden, and Liu [14] and Glunt, Hayden, and Raydan [15]. This construction requires computing the spectral decomposition of the symmetric $n \times n$ matrix $\tau(\Delta * \Delta)$. Although modern methods for computing the spectral decomposition are very efficient (see Sorensen [36] for an explanation of the method that we employed), the fact that the matrix $\tau(\Delta * \Delta)$ is extremely dense means that implementing this strategy may still be fairly expensive when n is large, as is often the case for molecular conformation problems.

In case computing resources are limited, we also investigated a second, less expensive strategy for constructing an initial configuration. This construction proceeds sequentially, beginning with the placement of the first point at the origin. In general, the jth point is placed at a location determined by the preceding p points. Specifically, given $x_{j-p}, \ldots, x_{j-1}$, the coordinates of x_j are determined by solving the elementary least squares problem

$$\text{minimize} \quad \sum_{i=j-p}^{j-1} w_{ij} \left[\left(\sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^r - \left(\delta_{ij}^2 \right)^r \right]^2 . \tag{8}$$

Notice the similarity of Problems (8) and (3). Solving Problem (8) optimizes the location of x_j with respect to x_i $(i = j-p, \ldots, j-1)$, either approximating the p dissimilarites δ_{ij} with the p distances d_{ij} (the STRESS criterion, for $r = 1/2$) or approximating the p squared dissimilarites δ_{ij}^2 with the p squared distances d_{ij}^2 (the SSTRESS criterion, for $r = 1$). This construction turns out to be considerably less expensive than computing the metric STRAIN solution, and our experience suggests that the algorithm typically converges to the same local minimizer from either initial configuration.

When points in a configuration coalesce, i.e. when the algorithm steps to a configuration matrix X^k with two or more identical rows, the performance of the algorithm deteriorates and numerical difficulties are sometimes encountered. Naturally, we would prefer to avoid such configurations. For the STRESS problem, de Leeuw [8] has shown that points cannot coalesce at local minimizers, so there is no reason to ever consider such configurations. (In contrast, points can coalesce at local minimizers of the SSTRESS problem, and also at solutions of the STRAIN problem. This is one argument that can be advanced for preferring the STRESS criterion.) We found that, in practice, configurations with coalescing points are rarely encountered if all points in the initial configuration are distinct. For this reason, we actually implemented a slightly modified version of our second strategy for choosing an initial configuration. If solving Problem (8) placed x_j too close to any previous points, then the location of x_j was perturbed. We required that no interpoint distance in the initial configuration be smaller than the smallest (strictly positive) dissimilarity, i.e.

$$\min_{ij} d_{ij}(X^0) \geq \min_{ij} \delta_{ij}.$$

This condition was easy to enforce, and in practice it effectively inhibited the coalescence of points in subsequent configurations X^k.

We experimented with several globalization strategies. Both of the standard trust region methods performed extremely well. Extensive numerical experimentation suggested that the hook step method of Hebden [21] and Moré [29] slightly outperformed the double dogleg step method of Powell [32]. When a line search method was used *instead* of a trust region method, the overall performance of the algorithm deteriorated. Notice, however, that we do backtrack from the trust region step. This is somewhat nontraditional, but extensive numerical experimentation suggested that it improved the overall performance of the algorithm.

Following Section 7.2 of Dennis and Schnabel [12], our convergence criterion combined three different conditions, developed to answer the following heuristic questions:

1. "Have we solved the problem?"
2. "Have we ground to a halt, either because the algorithm has converged or simply because it has stalled?"
3. "Have we exhausted our resources?"

To ascertain if the problem has been solved, we check to see if we have found a stationary configuration, i.e. a configuration at which the gradient of the objective function is sufficiently close to zero. In case the problem is badly scaled, we employ a relative measure of magnitude. Let typx_i denote the user's estimates of typical magnitudes of the coordinates of X, and let $\mathrm{typ}f$ denote the user's estimate of a typical magnitude of $f_r(X)$. Then the condition suggested by Dennis and Schnabel [12] is

$$\max_{1\leq i\leq N} \frac{|[\nabla f_r(X^k)]_i| \max(|[X^k]_i|, \mathrm{typ}x_i)}{\max(|f_r(X^k)|, \mathrm{typ}f)} \leq \epsilon_1. \tag{9}$$

To ascertain if the algorithm has ground to a halt, we check to see if the size of the step is sufficiently close to zero. Again, we employ a relative measure of magnitude, viz.

$$\max_{1\leq i\leq N} \frac{|[s^k]_i|}{\max(|[X^k]_i|, \mathrm{typ}x_i)} \leq \epsilon_2. \tag{10}$$

Finally, we limit resources by limiting the number of iterations. The algorithm continues until either condition (9) or condition (10) obtains, or until the maximum number of iterations is reached. æ

5 Numerical Experiments

The algorithm presented in Section 4 was tested on several large, unweighted metric STRESS problems. For small problems, algorithmic efficiency is obviously less important. Currently, the most important source of large metric MDS problems is computational chemistry. For this reason, our test problems were designed to approximate molecular structure and configurations were constructed in $p = 3$ dimensions.

There were several reasons for our focus on the STRESS criterion. First, STRESS problems appear to be more difficult than SSTRESS problems. Second, it is our impression that SSTRESS has declined in popularity in recent years. Third, STRESS seems particularly suitable for molecular conformation because, unlike SSTRESS, points cannot coalesce in optimal STRESS configurations. Fourth, unlike SSTRESS, there is an established computer program, SMACOF-I [22], for minimizing the STRESS criterion. The SMACOF-I program, which is an implementation of the majorization method discussed in Section 3.1, provides a "gold standard" to which our algorithm can be compared. It should be noted that SMACOF-I is highly regarded in the literature. For example, McFarlane and Young [28] recently stated that "In the context of an interactive and dynamic graphics system, the method of choice is SMACOF-I ...; it is the fastest algorithm that optimizes the [STRESS criterion]." (p. 26).

We began by selecting three molecules: Cranbin ($n = 394$ atoms), Deoxyribonucleic acid ($n = 566$ atoms), and Glycopeptide antibiotic ($n = 122$ atoms). Configuration coordinates in R^3 for these molecules are available from the Brookhaven Protein Bank. From these coordinates, we computed the Euclidean interatomic distances, $\bar{d}_{ij}$. These distances were then perturbed to obtain dissimilarities.

For each molecule, we constructed three dissimilarity matrices, each having a different magnitude of error. The true interatomic distances $\bar{d}_{ij}$ were multiplied by errors drawn from a lognormal distribution, an error model proposed by Wagenaar and Padmos [42] that was recently employed by Groenen [18]. Specifically, for each $\bar{d}_{ij}$, we generated a pseudorandom number z_{ij} from a standard normal distribution. We then set $\sigma = g(\log 10)/1.95996$, for $g = 1, 2, 3$, and $\delta_{ij} = \bar{d}_{ij}\exp(\sigma z_{ij})$.

For each of the nine dissimilarity matrices that we obtained, we attempted to solve the metric STRESS problem in $p = 3$ dimensions using three different algorithms: our implementation of Newton's method, described in Section 4; and each of the two updating schemes available in SMACOF-I, "Guttman transforms" and "relaxed updates." Heiser and de Leeuw [22] argued that the latter scheme squares the convergence constant, thereby halving the number of iterations required for convergence by the Guttman sequence.

For each of the nine dissimilarity matrices, we also considered three strategies for choosing an initial configuration: the metric STRAIN solution, which happens to be the default initial configuration computed by SMACOF-I; the less expensive configuration described in Section 4; and five randomly generated configurations. In order for the randomly generated configurations to be meaningful, it is important to ensure that they are reasonably scaled.

Given a dissimilarity matrix Δ, a fairly natural way to generate a random initial $n \times p$ configuration matrix Y is to do the following. Let $m = n(n-1)/2$, let $S = \sum_{i<j}\delta_{ij}$, and let $\sigma^2 = S/(2pm)$. For each configuration coordinate, generate a pseudorandom number z_{ij} from a standard normal distribution and take $y_{ij} = \sigma z_{ij}$. It is easily verified that this procedure produces configurations for which the expected squared interpoint distance equals the average squared dissimilarity in Δ.

To perform the indicated numerical experiments, we had to overcome several technical obstacles. First, it was necessary to slightly modify SMACOF-I so that sufficient memory was allocated to solve the very large problems that we considered. Furthermore, despite the fact that it does not require second derivatives, SMACOF-I's memory demands actually increase with n *faster* than do the memory demands of Newton's method. For this reason, the size of the problems that we were able to consider on our platform (Sun SparcStation 10) was smaller for SMACOF-I than for our algorithm. This reverses Kruskal's [24] argument that second order methods are disadvantaged because they require more memory.

A more serious difficulty is the convergence criterion used by SMACOF-I, which stops when an iteration fails to decrease the value of the STRESS function by at least ϵ_3. (The value of ϵ_3 can be specified by the user; the SMACOF-I default is $\epsilon_3 = 10^{-5}$.) This criterion may stop the algorithm prematurely, as failure to take a step that sufficiently decreases the value of the objective function does not necessarily mean that one is near a local minimizer. We addressed this difficulty by using each solution found by SMACOF-I as an initial configuration for our algorithm, thereby establishing if further decrease was possible.

Finally, it should be noted that these experiments required a certain amount of recoordinatization. Configurations must be centered if they are to be used as initial configurations for SMACOF-I, whereas our algorithm uses a different parametrization.

Because SMACOF-I was written in single precision, all of our experiments were performed using 32-bit IEEE floating point arithmetic. Naturally, our algorithm produces more accurate solutions when it is run in double precision. Our algorithm stopped if either condition (9) or condition (10) was satisfied. We used tolerances of $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-8}$. Typically, the algorithm stopped because (9) was satisfied. The stopping criterion tolerance for SMACOF-I was $\epsilon_3 = 2 \times 10^{-6}$. Each run of each algorithm was permitted a maximum of 250 iterations.

The results of our experiments are tabled in the Appendix. From these tables, several patterns emerge. First, the hope that SMACOF-I's relaxed updating scheme requires half as many iterations as Guttman transforms appears to be overly optimistic. Using the STRAIN solution as the initial configuration (the default for SMACOF-I), the ratio of the number of iterations for relaxed updates to the number of iterations by Guttman transforms ranged from .5926 to 2.0541, with a median of .7353.

Next, it is quite interesting to note that the number of SMACOF-I iterations does not increase greatly with decreasing quality of the initial configuration. The median ratio of the number of iterations from the inexpensive initial configuration to the number of iterations from the STRAIN solution was actually *less* than unity (.5833 for Guttman transforms, .6800 for relaxed updates). For random initial configurations, the corresponding median ratios were only 1.0556 for Guttman transforms and 1.1071 for relaxed updates. There are several possible explanations for this phenomenon. First, it may be a consequence of excellent global, but slow local, convergence properties of SMACOF-I. Second, it may be that SMACOF-I is finding different local minimizers from different initial configurations. Third, it may be that SMACOF-I is stopping prematurely. Below, we will argue that SMACOF-I *does* have a tendency to stop prematurely.

The performance of our algorithm was very different from the behavior of SMACOF-I. From the STRAIN solution, the number of iterations until convergence ranged from 12 to 29. The median ratio of the number of iterations from the inexpensive initial configuration to the number of iterations from the STRAIN solution was 2.5625. From random initial configurations, the number of iterations until convergence was never less than 158 and the algorithm failed to converge in 250 iterations on several occasions. The median ratio of the *smallest* number of iterations from one of five random initial configuration to the number of iterations from the STRAIN solution was 7.6957. These results clearly illustrate the fast local convergence of Newton's method and the virtue of starting from a good initial configuration.

The most intriguing results have to do with the quality of the solutions obtained by the different algorithms from the different initial configurations. When solutions obtained by SMACOF-I were used as initial configurations for our algorithm, Newton's method always took additional steps and usually decreased the value of the objective function. From the SMACOF-I solutions obtained using Guttman transforms (relaxed updates) from the STRAIN solution, our algorithm took a median of 18 (18) additional steps and further decreased the STRESS value by a median of 0.77 (0.45) percent. Considering the stringency of the tolerances in the convergence criteria, this an appreciable amount. When other SMACOF-I solutions were used, the numbers of additional steps and the relative decreases in STRESS were typically even greater.

For each dissimilarity matrix, starting SMACOF-I from different initial configurations typically produced different final STRESS values. In comparison, for each dissimilarity matrix our algorithm typically produced fairly homogenous final STRESS values regardless of the initial configuration. In particular, our algorithm typically recovered roughly the same final STRESS value when started from SMACOF-I solutions with rather different STRESS values. These results strongly suggest that SMACOF-I has a tendency to stop prematurely.

To some extent, SMACOF-I's tendency to stop prematurely can be counteracted by specifying an extremely stringent tolerance for its convergence criterion. We suspect, however, that the actual difficulty is more fundamental, as it is well known that the stopping criterion of sufficient decrease in the objective function has a *general* tendency to induce this behavior. Even more interesting is the question of whether the convergence criterion is entirely to blame or whether the algorithm itself is partially responsible. We believe that, because the STRESS function is extremely shallow (i.e. fairly large regions of configurations have very similar STRESS values), any algorithm that fails to exploit second order information will experience difficulty actually finding a minimizer.

æ

6 Discussion

Both the metric STRESS and SSTRESS problems are mildly nonlinear least squares problems for which local minimizers can be efficiently obtained by the methods of modern numerical optimization. These problems are well-suited to the straightforward application of Newton's method. Accordingly, we believe that the second order algorithm that we have presented represents a substantial improvement on the first order methods most commonly used in current practice. An additional, very appealing feature of this approach is that the same algorithm can be used with either the STRESS or the SSTRESS criterion.

This paper has focussed exclusively on algorithms for finding solutions that correspond to fixed dissimilarity matrices. It is the fact that the dissimilarities are fixed that is the defining characteristic of metric MDS. Although metric MDS is a central mathematical problem of MDS, in most applications it does not represent the entire problem. For this reason, we conclude by briefly considering the importance of metric MDS in MDS, and the potential role of the algorithm that we have presented.

In computational chemistry, MDS is sometimes used to construct molecular configurations from measurements of interatomic distances. Because a physical molecule actually does exist in R^3, dissimilarity matrices that lead to configurations with small objective function values are to be expected and preferred. Therefore, instead of simply solving the metric MDS problem defined by setting the dissimilarites equal to the measured interatomic distances, it is standard practice to begin by smoothing the data, obtaining a dissimilarity matrix that is more nearly a distance matrix. Nevertheless, and regardless of the exact procedure by which the dissimilarites are obtained, the solution of a metric MDS problem is an important component of the complete analysis.

Algorithms for solving metric MDS problems play a clearly defined role in Havel's [20] modularized DG-II package for the determination of protein structure from distance constraints obtained from nuclear magnetic resonance (NMR) spectroscopy. The majorization module of DG-II finds local minimizers of metric STRESS problems using de Leeuw's [7] majorization algorithm, described in Section 3.1. It would be a simple matter to replace this module with an implementation of the Newton's method algorithm that we have proposed.

A slightly different approach to determining molecular structure is the data box algorithm of Glunt, Hayden, and Raydan [15], in which the dissimilarities are allowed to vary subject to bound constraints determined by the error structure of the measurement process. This algorithm employs the method of alternating least squares (ALS), whereby one alternately optimizes the dissimilarity variables for a fixed configuration and the configuration coordinates for a fixed dissimilarity matrix. The latter subproblems are metric STRESS problems, which the authors solve using their spectral gradient algorithm. It would be a simple matter to substitute the Newton's method algorithm for the spectral gradient algorithm.

Finally, as defined by Kruskal [25], nonmetric MDS allows the dissimilarities to vary subject to order constraints. As exemplified by the popular ALSCAL algorithm of Takane, Young, and de Leeuw [37], ALS is often used to solve nonmetric MDS problems. This means that, like the data box algorithm, many nonmetric MDS algorithms entail solving metric MDS subproblems. Again, it would be a simple matter to use the Newton's method algorithm to solve these subproblems.

Thus, metric MDS problems appear in a variety of contexts. Many of these contexts involve large configurations and/or the successive solution of repeated metric MDS subproblems. Accordingly, one can reasonably anticipate that the very efficient Newton's method algorithm for solving these problems will find a variety of applications.

æ

Acknowledgements

We would like to thank the individuals who so generously contributed to our research. Mark Gockenbach took an interest in our work and made many helpful observations. Tom Hayden and Marcos Raydan shared the results of their own studies of the metric STRESS and SSTRESS problems. David Xi and especially Garry King kindly explained some of the relevant concerns of computational biochemistry and NMR spectroscopy, and also provided the molecular data sets that we used in our numerical experiments. Finally, we are especially grateful to Andrea Reiff, whose assistance in the development of our algorithm was invaluable. Andrea patiently considered and discussed the details of myriad possible approaches, discovering the flaws in many of them. æ

Appendix: Results of Numerical Experiments

The following tables report numbers of iterations and final STRESS values for the numerical experiments described in Section 5. Each table corresponds to one of nine dissimilarity matrices. For each dissimilarity matrix, seven initial configurations were generated: the metric STRAIN solution, the inexpensive (Cheap) initial configuration described in Section 4, and five random initial configurations. Each row in each table corresponds to one initial configuration.

From each initial configuration, three algorithms were used to solve the metric STRESS problem: the algorithm proposed in Section 4 (Newton), SMACOF-I with Guttman transforms, and SMACOF-I with relaxed updates. For our algorithm, we report the number of iterations (Iter) and the final STRESS value. For the SMACOF-I algorithms, we report the number of iterations (It-1) and the final STRESS value (Stress-1). We also report the number of iterations (It-2) and the final STRESS value (Stress-2) obtained by starting our algorithm from the solution obtained by SMACOF-I.

If an algorithm stopped because the maximum number of iterations (250) was reached, then the number of iterations is reported as 250 and no STRESS value is reported. To facilitate interpretation, all STRESS values were multiplied by 10^4.

Cranbin, g=1

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	13	3356.61	27	3463.58	15	3356.53	25	3466.35	18	3356.45
Cheap	121	3356.59	41	3712.13	19	3356.44	40	3715.03	40	3356.62
Random	219	3356.67	89	3915.29	159	3356.24	51	3723.99	226	3356.82
Random	194	3356.62	56	3821.28	147	3356.80	49	3665.21	213	3357.36
Random	250	—	45	3741.43	191	3356.03	36	3688.70	107	3356.87
Random	183	3356.61	71	3970.98	212	3356.31	60	3731.75	186	3356.44
Random	197	3356.62	31	3600.30	214	3356.49	29	3818.09	145	3356.37

Cranbin, g=2

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	12	3492.08	36	3545.62	18	3489.77	24	3546.67	16	3490.30
Cheap	77	3492.34	21	3625.96	96	3492.52	21	3526.41	93	3490.76
Random	175	3492.02	27	3670.84	126	3491.26	26	3506.19	145	3491.40
Random	220	3492.21	67	3940.87	243	3492.79	41	3719.92	140	3491.09
Random	235	3491.98	72	3593.05	118	3492.16	41	3618.18	109	3492.86
Random	244	3492.06	38	3766.69	199	3490.69	37	3603.03	231	3489.61
Random	219	3492.19	44	3697.08	145	3491.44	33	3564.30	218	3491.56

Cranbin, g=3

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	16	4191.81	81	4211.96	35	4191.53	48	4210.67	26	4191.58
Cheap	41	4192.36	17	4303.55	195	4192.37	17	4333.25	232	4191.75
Random	163	4191.99	49	4410.31	137	4191.84	41	4221.38	242	4191.94
Random	223	4191.87	52	4349.89	216	4191.56	29	4221.12	99	4191.63
Random	228	4191.69	54	4264.91	248	4192.78	31	4313.75	213	4192.09
Random	250	—	24	4741.30	219	4192.16	22	4338.27	227	4191.73
Random	250	—	43	4639.14	239	4191.27	41	4226.51	248	4191.99

Deoxyribonucleic acid, g=1

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	29	3751.12	43	3811.26	18	3750.99	28	3809.96	39	3751.62
Cheap	117	3751.42	39	3908.92	59	3751.26	32	3901.68	48	3751.51
Random	209	3751.12	61	3841.47	249	3756.15	38	3841.14	250	—
Random	250	—	96	3892.18	191	3755.50	72	3891.78	239	3751.52
Random	250	—	29	3852.61	218	3754.71	25	3861.31	225	3751.85
Random	250	—	54	3959.23	250	—	49	3911.53	241	3751.58
Random	214	3752.07	39	3898.98	192	3758.60	31	3881.71	212	3751.95

Deoxyribonucleic acid, g=2

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	26	3799.74	27	3800.10	18	3793.96	26	3804.24	16	3799.77
Cheap	63	3799.21	13	3853.66	59	3798.38	13	3810.19	63	3798.28
Random	250	—	31	3916.85	250	—	27	4011.68	227	3795.86
Random	213	3798.99	28	3859.18	234	3793.63	27	3833.83	223	3795.03
Random	250	—	36	3912.82	195	3797.02	30	3811.45	250	—
Random	163	3799.14	39	3812.82	178	3792.19	36	3903.63	246	3794.56
Random	250	—	39	3880.97	242	3796.10	33	3880.97	241	3795.47

Deoxyribonucleic acid, g=3

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	20	5096.57	34	5100.42	19	5096.36	25	5174.04	18	5096.21
Cheap	51	5098.66	17	5118.60	50	5098.55	17	5132.66	51	5095.01
Random	233	5098.38	25	5138.20	201	5096.41	24	5138.20	215	5095.77
Random	221	5097.23	36	5104.62	219	5097.86	33	5104.83	189	5097.23
Random	236	5096.20	38	5182.18	192	5099.16	36	5182.42	221	5095.36
Random	250	—	89	5144.55	185	5098.05	70	5144.55	250	—
Random	229	5095.45	42	5317.46	250	—	40	5133.07	239	5095.37

Glycopeptide antibiotic, g=1

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	23	3002.20	31	3171.19	41	3002.21	20	3009.68	6	3002.84
Cheap	51	3000.59	54	3117.53	112	3005.87	23	3136.31	208	3005.78
Random	240	3004.27	27	3194.01	241	3005.68	24	3124.39	234	3005.11
Random	216	3001.03	51	3183.93	140	3008.95	22	3125.97	144	3005.52
Random	173	2999.92	50	3187.22	116	3005.34	22	3167.22	121	3008.75
Random	181	3005.81	22	3170.75	157	3005.12	21	3113.93	135	3007.75
Random	158	3005.60	29	3154.47	169	3006.08	23	3114.01	79	3007.74

Glycopeptide antibiotic, g=2

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	24	3126.98	42	3128.90	8	3126.08	35	3129.38	12	3126.71
Cheap	48	3125.93	22	3436.67	11	3128.29	21	3436.98	22	3127.02
Random	217	3126.75	23	3424.70	156	3127.94	21	3328.31	17	3126.40
Random	167	3126.99	89	3329.54	129	3128.37	56	3229.72	18	3126.57
Random	212	3126.08	27	3427.31	134	3126.94	22	3332.69	38	3126.72
Random	183	3127.00	29	3229.37	181	3126.03	23	3231.74	42	3126.61
Random	188	3126.18	21	3439.29	189	3126.19	21	3416.71	112	3127.10

Glycopeptide antibiotic, g=3

	Newton		SMACOF-I, Guttman transforms				SMACOF-I, Relaxed updates			
	Iter	Stress	It-1	Stress-1	It-2	Stress-2	It-1	Stress-1	It-2	Stress-2
Strain	23	3816.61	37	3845.37	34	3815.69	76	3824.59	23	3816.15
Cheap	63	3815.45	89	3921.57	89	3815.58	31	3921.75	61	3815.65
Random	187	3816.91	29	3921.99	111	3815.86	28	3917.11	115	3815.57
Random	191	3816.12	27	3923.58	250	—	18	3919.83	132	3815.64
Random	177	3815.96	38	3987.99	153	3815.25	21	3896.54	121	3815.42
Random	198	3815.99	39	3916.14	162	3816.32	31	3917.57	148	3815.59
Random	199	3815.63	37	3918.25	241	3815.26	32	3900.48	250	—

æ

References

[1] J. Barzilai and J. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[2] P. T. Boggs, R. H. Byrd, J. R. Donaldson, and R. B. Schnabel. ODRPACK: software for weighted orthogonal distance regression. *ACM Transactions on Mathematical Software*, 15:100–200, 1991.

[3] P. T. Boggs and J. E. Dennis. A stability analysis for perturbed nonlinear iterative methods. *Mathematics of Computation*, 30:1–17, 1976.

[4] M. Browne. The Young-Householder algorithm and the least square multidimensional scaling of squared distance. *Journal of Classification*, 4:175–190, 1987.

[5] G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation.* John Wiley & Sons, New York, 1988.

[6] J. de Leeuw. Applications of convex analysis to multidimensional scaling. In J. R. Barra, F. Brodeau, G. Romier, and B. van Cutsem, editors, *Recent Developments in Statistics*, pages 133–145, North-Holland Publishing Company, Amsterdam, 1977.

[7] J. de Leeuw. Convergence of the majorization method for multidimensional scaling. *Journal of Classification*, 5:163–180, 1988.

[8] J. de Leeuw. Differentiability of Kruskal's Stress at a local minimum. *Psychometrika*, 49:111–113, 1984.

[9] J. de Leeuw and W. Heiser. Multidimensional scaling with restrictions on the configuration. In P. R. Krishnaiah, editor, *Multivariate Analysis*, North-Holland Publishing Company, Amsterdam, 1980.

[10] J. de Leeuw and W. Heiser. Theory of multidimensional scaling. In P. R. Krishnaiah and I. N. Kanal, editors, *Handbook of Statistics*, chapter 13, pages 285–316, North-Holland Publishing Company, Amsterdam, 1982.

[11] J. E. Dennis, D. M. Gay, and R. E. Welsch. An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software*, 7:348–368, 1981.

[12] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[13] W. Glunt, T. L. Hayden, S. Hong, and J. Wells. An alternating projection algorithm for computing the nearest Euclidean distance matrix. *SIAM Journal on Matrix Analysis and Applications*, 11:589–600, 1990.

[14] W. Glunt, T. L. Hayden, and W.-M. Liu. The embedding problem for predistance matrices. *Bulletin of Mathematical Biology*, 53:769–796, 1991.

[15] W. Glunt, T. L. Hayden, and M. Raydan. Molecular conformations from distance matrices. *Journal of Computational Chemistry*, 14:114–120, 1993.

[16] W. Glunt, T. L. Hayden, and M. Raydan. Preconditioners for distance matrix algorithms. *Journal of Computational Chemistry*, 15:227–232, 1994.

[17] J. C. Gower. Some distance properties of latent root and vector methods in multivariate analysis. *Biometrika*, 53:315–328, 1966.

[18] P. J. F. Groenen. *The Majorization Approach to Multidimensional Scaling.* DSWO Press, Leiden, The Netherlands, 1993.

[19] L. Guttman. A general nonmetric technique for finding the smallest coordinate space for a configuration of points. *Psychometrika*, 33:469–506, 1968.

[20] T. F. Havel. An evaluation of computational strategies for use in the determination of protein structure from distance constraints obtained by nuclear magnetic resonance. *Progress in Biophysics and Molecular Biology*, 56:43–78, 1991.

[21] M. D. Hebden. *An Algorithm for Minimization Using Exact Second Derivatives.* Report TP515, Atomic Energy Research Establishment, Harwell, England, 1973.

[22] W. J. Heiser and J. de Leeuw. *SMACOF-I.* Technical Report UG-86-02, Department of Data Theory, University of Leiden, Leiden, The Netherlands, 1986.

[23] M. R. Hoare and P. Pal. Physical cluster mechanics: statics and energy surfaces for monatomic systems. *Advances in Physics*, 20:161–196, 1971.

[24] J. B. Kruskal. Multidimensional scaling and other methods for discovering structure. In K. Enslein, A. Ralston, and H. S. Wilf, editors, *Statistical Methods for Digital Computers*, chapter 12, pages 296–339, John Wiley & Sons, New York, 1977. Volume III of *Mathematical Methods for Digital Computers.*

[25] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[26] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29:28–42, 1964.

[27] J. B. Kruskal and M. Wish. *Multidimensional Scaling.* Sage Publications, Beverly Hills, 1978. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-001.

[28] M. McFarlane and F. W. Young. Graphical sensitivity analysis for multidimensional scaling. *Journal of Computational and Graphical Statistics*, 3:23–33, 1994.

[29] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116, Springer Verlag, Berlin, 1977. *Lecture Notes in Mathematics*, 630.

[30] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. *User's Guide for MINPACK-1.* Technical Report ANL 80-74, Argonne National Laboratories, 1980.

[31] J. A. Northby. Structure and binding of Lennard-Jones clusters: $13 \leq n \leq 147$. *Journal of Chemical Physics*, 87:6166–6177, 1987.

[32] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114, Gordon and Breach, London, 1970.

[33] M. Raydan. On the Barzilai and Borwein choice of a steplength for the gradient method. *IMA Journal of Numerical Analysis*, 13:321–326, 1993.

[34] R. N. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27:125–140, 1962.

[35] R. N. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. II. *Psychometrika*, 27:219–246, 1962.

[36] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM Journal on Matrix Analysis*, 13:357–385, 1991.

[37] Y. Takane, F. W. Young, and J. de Leeuw. Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. *Psychometrika*, 42:7–67, 1977.

[38] P. Tarazaga and M. W. Trosset. An optimization problem on subsets of the symmetric postive semidefinite matrices. *Journal of Optimization Theory and Applications*, 79:513–524, 1993.

[39] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17:401–419, 1952.

[40] W. S. Torgerson. *Theory and Methods of Scaling.* John Wiley & Sons, New York, 1958.

[41] M. W. Trosset. *The Formulation and Solution of Multidimensional Scaling Problems.* Technical Report TR93-55, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892, 1993. Submitted for publication.

[42] W. A. Wagenaar and P. Padmos. Quantitative interpretation of Stress in Kruskal's method multidimensional scaling technique. *British Journal of Mathematical and Statistical Psychology*, 24:101–110, 1971.

[43] G. Young and A. S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938.