An Interval Newton Method for Nonlinear Least Squares

by
R. Baker Kearfott
Department of Mathematics
U.S.L. Box 4-1010
Lafayette, LA 70504

Abstract

Various researchers have studied interval Newton methods in conjunction with generalized bisection for reliable computation of all roots and for global optimization within a box in n-space. Such methods are especially advantageous when rigorous bounds on the optimal parameters are required or when it is unknown whether local optima are unique.

These interval Newton methods could be applied directly to nonlinear least squares problems. However, that would involve computing interval values for the second derivatives of the original residual functions. Besides contributing to the total computational cost, such lengthy calculations could possibly return unacceptably wide result intervals. To avoid this, we propose interval extensions of the Gauss-Newton method.

The Gauss-Newton method is not guaranteed to be locally convergent for large residual problems. However, it is always stationary at roots of the normal equations, and appropriate interval extensions will preserve an important property of interval Newton methods.

We compare computationally our interval Gauss-Newton methods to more usual interval Newton methods. Our test set includes problems from the literature, simple large-residual problems, and nonlinear exponential fitting problems.

Keywords: nonlinear least squares, interval mathematics, Gauss-Newton method

AMS Subject classifications: 65K10, 65H10.

1. INTRODUCTION AND PURPOSE

Interval Newton methods have been studied for some time as techniques for *reliably* finding *all* solutions to systems of nonlinear equations. That is, they address the following problem.

> Find, *with certainty*, approximations to *all* solutions of the nonlinear system:

(1.1) $$f_i(x_1,x_2,...,x_n) = 0, \; 1 \le i \le n,$$

> where bounds a_i and b_i are known such that:

$$a_j \le x_j \le b_j \text{ for } 1 \le j \le n.$$

We write $F(X) = 0$, where $F = (f_1,f_2,...,f_n)$ and $X = (x_1,x_2,...,x_n)$. We denote the box given by the inequalities on the variables x_j by $\mathbf{X}$.

These methods are based on computing an image box $\mathbf{K}$ whose i-th coordinate extent is $[c_i,d_i]$, where

(1.2) $$c_i \le N_i(x_1,x_2,...,x_n) \le d_i \text{ for } \le i \le n$$

for every $X = (x_1,x_2,...,x_n) \in \mathbf{X}$, where $N_i(X)$ is the i-th component of the image of X after one iteration of Newton's method. If $\mathbf{K} \subset \mathbf{X}$, then, under certain conditions, we may conclude that there is a solution of F within $\mathbf{X}$, and that any solutions of F within $\mathbf{X}$ must be within $\mathbf{K}$.

To obtain better bounds on these solutions, we replace X by K and repeat the process. Even if $K \not\subseteq X$, any solutions of F in X must be in $K \cap X$ (since Newton's method is stationary at solutions), so we may replace X by $K \cap X$ and iterate. If this does not reduce all of the coordinate intervals $[a_i,b_i]$ sufficiently, we may form two boxes by bisecting one of the coordinate intervals; we then consider each of these two boxes separately. If, on the other hand, $K \cap X$ is empty, then there are no roots of F in X, so we may discard X.

If we apply this entire process (iteration and possible bisection) to each of these boxes separately, we will, with certainty, find all roots of F within X. (See [5] for a theoretical discussion of this algorithmic structure). We review the technique we employ here for computing K in Section 2. (Also see [3, sect. 4].)

For specific convergence results for such interval Newton methods, see [8], [10], [13], [14], and numerous other works. For an introduction to the interval mathematics generally used to compute K, see [9] or [1]. For a general framework and survey, see [4]. For a practical variant, see [3]. For systematic computational test on a battery of standard problems and for comparisons with alternate techniques, see [6]. For an abstract analysis of the bisection process, see [5].

Interval Newton methods as described above require evaluation of the Jacobian matrix of *F* over the entire box **X** using interval arithmetic. This usually need not affect the practicality of an interval Newton method; in fact the overall cost could still be less than alternatives, and we gain total reliability (cf. [6]). The Jacobian evaluation can be combined with in-line symbolic manipulators (such as in [16]), automatic differentiation techniques (as in [15]), or with general routines for evaluating functions with specific forms (such as in [7]), so that programming the Jacobian matrix evaluation and explicit compiler (or precompiler) support of interval arithmetic are not issues.

As with classical non-interval methods, the situation is slightly different for global nonlinear optimization and nonlinear least squares computations. We may transform the general nonlinear optimization case to (1.1) by identifying the gradient of the objective function with *F*. Except for the fact that the Jacobian matrix involves second-order partial derivatives, the cost of the resulting interval Newton method is then similar to the cost of the general problem (1.1) (except that additional information can be employed in the optimization problem; cf. [4]).

For the nonlinear least squares problem with m functions and n variables, the objective function is of the form

$$\Phi(X) = \frac{1}{2} \sum_{j=1}^{m} f_j^2(X), \tag{1.3}$$

so that its gradient is $G(X) = J(X)^T F(X)$, where $F(X)$: $\mathrm{R}^n \to \mathrm{R}^m$ is the vector function whose j-th component is f_j and where $J(X)$ is the m by n Jacobian matrix of G. Thus the Jacobian matrix of G is of the form

$$H(X) = J^T(X)J(X) + S(X)$$

where

$$S(X) = \sum_{j=1}^{m} f_j(X)\ H_j(X) \tag{1.4}$$

where H_j is the Hessian matrix of f_j.

Unless symbolic preprocessing is performed to first obtain a compact representation for G, evaluation of H for a Newton method thus would require evaluation of $mn + mn^2$ matrix components, where m may be large. Furthermore, the large number of algebraic operations required to evaluate S could cause the computed intervals for H to be too wide.

In the classical Gauss-Newton method to find roots of G, we replace the true derivative matrix H by J^TJ in the Newton iteration equation. Thus, we need not compute S, but we obtain convergence under more restricted conditions than Newton's method. In particular, suppose that X^* satisfies $G(X^*) = 0$, denote the smallest eigenvalue of $J^T(X^*)J(X^*)$ by λ, and denote the largest eigenvalue of $S(X^*)$ (i.e. $||S(X^*)||_2$) by σ. Then we are assured that the Gauss-Newton method converges to X^* only when $\lambda < \sigma$. (See eg. Theorem 10.2.1, p. 222 ff. of [2].) The case $\lambda \geq \sigma$ can occur when the residuals f_j are relatively large at X^* and when the f_j are also sufficiently nonlinear.

In non-interval software, the Gauss-Newton method is combined with the method of steepest descent to assure convergence (cf. eg. [2, ch. 6]).

An interval Gauss-Newton method can be practical because of the following two observations. First, despite possible divergence, the Gauss-Newton method will always be stationary at solutions to $G(X) = 0$. Second, even though $\sigma \geq \lambda$, the Gauss-Newton method often will converge in *some* of the coordinates of X. (This depends on the relationship of the eigenvalues and eigenvectors of J^TJ and S and the direction of these eigenvectors in relation to the coordinate axes; examples in Section 4 provide some insight into this.) Thus, we may combine the interval Gauss-Newton method with generalized bisection to obtain an adequately rapid algorithm which is guaranteed to find the global optimum of Φ subject to $X \in \mathbf{X}$.

In Section 2, we describe interval extensions of the Gauss-Newton method. In Section 3, we briefly describe the environment in which we tested these methods. In Section 4, we describe the test problems, and in Section 5, we present numerical results and conclusions. We summarize in Section 6.

2. NOTATION AND INTERVAL EXTENSIONS OF THE GAUSS-NEWTON METHOD

In this section, we assume familiarity with interval arithmetic; see [1] or [9], etc. for an introduction. Here, lowercase letters (e.g. x_i) will denote real numbers, uppercase letters (e.g. X) will denote real vectors, upper case boldface letters (e.g. $\mathbf{X}$) will denote interval vectors (i.e. boxes), and lowercase boldface letters (e.g. $\mathbf{x}_i$) will denote intervals.

We model the method on the structure presented in [3] for (1.1). There, if $\mathbf{X}$ is a box, $X \in \mathbf{X}$, and $M \in \mathbf{X}$ is fixed, the mean value theorem gives

$$F(X) \subseteq F(M) + J(\mathbf{X})(X - M), \qquad (2.1)$$

where $J(\mathbf{X})$ is an interval extension of the Jacobian matrix by components. If we replace $F(X)$ by 0 and multiply both members of (2.1) by a nonsingular matrix Y, where Y approximates $[J(M)]^{-1}$, we obtain

$$(2.2) \qquad \mathbf{0} \in B + \mathbf{P}\ (\mathbf{X} - M),$$

where $B = Y\ F(M)$ and $\mathbf{P} = Y\ J(\mathbf{X})$. (Note that $\mathbf{P}$ is approximately a diagonal matrix, when the widths of the box $\mathbf{X}$ are small.) Interval Newton methods consist of using interval arithmetic to solve (2.2) for X. This will give a set $K(\mathbf{X})$ which contains the set $\{X \in \mathbf{X} \mid F(X) = \mathbf{0}\}$. A particularly successful such solution procedure appears in [3]. The resulting $K(\mathbf{X})$ can serve in a generalized bisection method as outlined in the introduction.

In [3], F: $\mathbb{R}^n \to \mathbb{R}^n$. Even so, (2.1) still holds if F: $\mathbb{R}^n \to \mathbb{R}^m$ with $m > n$. We are thus tempted to multiply (1.1) by an approximate generalized inverse Y to obtain an analogue of (2.2). However, if $Y = [J(M)]^+$ (where "$+$" denotes generalized inverse), then

$$Y = [J^T(M)J(M)]^{-1}J^T(M).$$

If, furthermore, $\mathbf{X}$ contains the single point X, then

$$\mathbf{P}^{-1}\ B = [J^T(M)J(X)]^{-1}J^T(M)F(M),$$

which is a single iteration of the Gauss-Newton method if $X = M$. However, if $X^* \in \mathbf{X}$ is a stationary point and $M \neq X^*$, then, unless $m = n$ (or unless the Gauss-Newton method is convergent), we cannot be sure that X^* is contained in the corresponding solution to (2.2). Thus, if this scheme is used to compute K, then we may have $K(X) \cap \mathbf{X}$ empty even though X contains a stationary point (as our computational experiments verify). This scheme, which we term the *heuristic scheme*, is thus unreliable.

An alternate procedure involves first premultiplying both sides of (2.1) by the interval matrix $J^T(X)$. We then take

(2.3)' $$Y \simeq [J^T(M)J(M)]^{-1}$$

so that we obtain (analogously to the derivation of (2.2))

(2.3) $$\mathbf{0} \in \mathbf{B} + \mathbf{P}\ (X - M),$$

where $\mathbf{B}$ is now an *interval* n-vector. In this case, $K(\mathbf{X})$ obtained by solving (2.3) must always contain stationary points of Φ within $\mathbf{X}$ (since the Gauss-Newton method is stationary at such points); thus, the associated method of bisection must always find the global optimum of Φ. We term this scheme the *deterministic method*.

For an interesting observation, suppose in (2.3) that Y = $[J^T(M)J(M)]^{-1}$. Denote $J(M)$ by J and denote $J(\mathbf{X})$ by $\mathbf{J}$ and assume $[\mathbf{J}^T\mathbf{J}]^{-1}$ can be computed. Then

$$\mathbf{P}^{-1}\ \mathbf{B} = \left\{[(\mathbf{J}^T\mathbf{J})^{-1}(J^TJ)][(J^TJ)^{-1}\ \mathbf{J}^T]\right\} F.$$

The matrix in curved parentheses can be interpreted as an *interval* generalized inverse.

Note that $J^T(M)J(M)$ can be ill-conditioned. This could cause problems in the algorithm, and, in particular, could result in wider coordinate widths in $K(X)$ (as computed in (2.3)) than necessary. This suggests use of an alternate preconditioner based on orthogonal transformations of the Jacobian matrix. In particular, we may write $J(M)$ in terms of its singular value decomposition

(2.4) $\quad J(M) = U \Sigma V'$,

where U and V are orthogonal matrices, where V' denotes the transpose of V, and where $\Sigma = (\sigma_{i,j})$ with $\sigma_{i,j} = 0$ if $i \neq j$ and where $\sigma_{i,i}$ is the i-th singular value of $J(M)$, for $1 \leq i \leq n$. We may then change the coordinate system representing F to obtain

(2.5)
$$\bar{F}(\mathbf{W}) = U'F\ (V\ \mathbf{W}), \text{ where}$$
$$\mathbf{W} = V'\ X.$$

We then form $\mathbf{W} = V'\ X$, and we identify W with the vector of midpoints of $\mathbf{W}$. An interval Jacobian of $\bar{F}$ is then given by

(2.6) $\quad \bar{J}(\mathbf{W}) = U'J(X)\ V \simeq \Sigma$.

We may now proceed as in (2.3), except that we replace X by $\mathbf{W}$, F by $\bar{F}$, and by $\bar{J}$. Since $\bar{J}^T(W)\bar{J}(W)$ is approximately diagonal (with the smallest diagonal element occuring last), additional preconditioning should not be necessary. Thus, solution of

$$(2.7) \qquad \mathbf{0} \in \overline{\boldsymbol{B}} + \overline{\boldsymbol{P}}\,(\boldsymbol{W} - \boldsymbol{W})$$

where $\overline{B} = \overline{J}^{T}(W)\overline{F}(W)$ and where $\overline{P} = \overline{J}^{T}(W)\overline{J}(W)$ and $\overline{\boldsymbol{P}}$ are analogous to $\boldsymbol{B}$ and $\boldsymbol{P}$ in (2.3)) should give smaller intervals than the solution of (2.3). However, since solution of (2.7) is embedded in a generalized bisection method based on $\boldsymbol{X}$, $\boldsymbol{W}$ must be computed before solving (2.7), and $V\ \overline{\boldsymbol{K}}$ must be computed before continuing the bisection algorithm (where $\overline{\boldsymbol{K}}$ is the solution of (2.7)). These computations may counterbalance the advantages of the better coordinate system, and we thus need to test the method. We term the scheme based on (2.7) the *singular value decomposition method.*

The final method we consider is the usual interval Newton method applied to the gradient function. That is, we begin with

$$(2.4) \qquad G(X) \in G(M) + H(X)(X - M),$$

as in (2.1), where G is the gradient of Φ and H is the Hessian matrix of Φ. We term this method the *exact Hessian method.*

Despite the fact that interval iterations of the exact Hessian method are theoretically quadratically convergent to stationary points of Φ (whereas the interval Gauss-Newton method is at most linearly convergent), the generalized bisection method associated with it is not necessarily more efficient than the deterministic method or the singular value decomposition method. This is both because much more computation may be involved in evaluating $H(X)$ and because these internal computations may magnify the widths of the resulting intervals more than the computations on J. These possibilities are examined in Section 4.

3. GENERAL ASPECTS OF THE TEST CODE

The generalized method of bisection in which we tested the three interval nonlinear least squares methods described in Section 2 is an improved version of that in [6]; it is similar to the code in [7]. Aside from changes in the data structures which do not affect the reported test quantities, the major differences from the code used for [6] are:

(i) utilization of the objective function Φ to reject some of the boxes produced during bisection (cf. [4, sect. 6.2.3]);

(ii) an improved method for determining which coordinate interval of X to bisect, when bisection is required;

(iii) modifications in the test to determine when to do another iteration of the interval (Gauss-)Newton method and when to bisect a coordinate interval instead;

(iv) use of a generic routine to evaluate F and J when the components of F are polynomials;

(v) use of a portable form of rigorous directed rounding in the interval arithmetic;

(vi) use of the rigorous interval arithmetic to evaluate the point function value $F(M)$; and

~~(vi)~~ (vii) a modification of the scheme which determines when to accept a box produced by the algorithm as a final answer.

In [6], the code chose the k-th coordinate interval $[a_k, b_k]$ of a box X to bisect, where

$$(3.1) \qquad (b_k - a_k) = \max_{1 \le j \le n} (b_j - a_j).$$

Here and in [7], we base the coordinate selection on the interval extension of F defined by

$$(3.2) \qquad \bar{F}(X) = F(M) + J(X)\ (X - M),$$

where $J(X)$ is a componentwise interval extension of the Jacobian matrix. Let the i-th component of $\bar{F}(X)$ be $[c_i, d_i]$, and denote the (i,j)-th entry of $J(X)$ by $J_{i,j} = [J_{i,j,1}, J_{i,j,2}]$. Then

$$(3.3) \qquad s_j = \left\{ \max_{1 \le i \le m} |J_{i,j,1}|, |J_{i,j,2}| \right\} (b_j - a_j)$$

is a measure of the maximum effect the size $(b_j - a_j)$ has on the uncertainty in the values of any component of F. We choose the coordinate index k in which to bisect such that

$$(3.4) \qquad s_k = \max_{1 \le j \le n} s_j .$$

Computational experiments with $m = n$ (solving (1.1)) indicate that (3.4) is generally a more effective criterion than (3.1), and is seldom, if ever, worse. Also, (3.4) seems to be more robust with respect to scaling than (3.1).

Here, to determine when to bisect and when to iterate the interval Gauss-Newton method, we examine the ratio of volumes of K and $K \cap X$. In particular, we introduce a tuning parameter τ such that

$$(3.5) \qquad \text{If} \quad \frac{\text{vol}\,(K \cap X)}{\text{vol}\,(X)} \begin{Bmatrix} \le \tau & \text{then iterate the method.} \\ \ge \tau & \text{then bisect a coordinate interval.} \end{Bmatrix}$$

For solving (1.1), $\tau = .6$ seemed reasonable; we use that value here.

In [6], we coded the interval computations for the f_i and $J_{i,j}$ separately for each problem. Here and in [7], we use generic routines similar to those in [12] for F, J, (and S, where applicable); we thus need only input the coefficients and degrees (with respect to each variable). However, since interval arithmetic is only subdistributive (cf. eg. [9, p. 13]) and since the generic routines do not make any attempt to factor expressions, the interval function computations here usually give somewhat wider intervals than those in [6]. This could cause the computational cost to be somewhat more. (For the nonpolynomial systems here, we coded the computations explicitly.)

Directed rounding is required for total reliability. In particular, extensive investigation revealed it (and an associated interval evaluation of $F(M)$) necessary to reliably solve problem 11 from the test set in [6], on some of the computers at our disposal.

Here, we assume that, if $K(\mathbf{X}) \subset \mathbf{X}$ (where the containment is strict), then there is a unique stationary point of Φ in $\mathbf{X}$. Such a result has been proven for various interval Newton methods which solve (1.1) (cf. eg. [10], [13, p. 263], [14]); it is conjectured to be true more generally in [4]. We have discovered no instances where this procedure leads to erroneous results, except for the heuristic method.

4. THE TEST SET

The set of test problems has the following three components:

(i) selected problems from the test set from [6] for general nonlinear systems codes;

(ii) a set of simple problems with $m > n$ and large residuals; and

(iii) A nonpolynomial problem with $m > n$ from the test set in [11].

4.1 The Test Set from [6]

These problems serve as a benchmark for the overall procedure. The heuristic method reduces to the usual interval Newton method, since $m = n$ and hence $J^{+} = J^{-1}$. However, $J^{T}F = 0$ is a more ill-conditioned system than $F = 0$, and these problems would reveal the effect of this on the deterministic method and on the exact Hessian method.

We selected problems 1, 2, 9, 12, 13, 14, 15, 16, and 17 from the original test set. We eliminated problems 3, 4, 10, and 11 since, as nonlinear least squares problems, they were too ill-conditioned to be solved by considering 10000 or less sub-boxes. We eliminated problems 5, 6, 7, and 8 from our tables since these linear problems are solved trivially in one step by all of the methods.

4.2 Some Model Problems with Large Residuals

We designed these problems to examine in a simple setting how non-convergence of the Gauss-Newton method affects the overall generalized bisection algorithm in conjunction with the heuristic, deterministic, and exact Hessian methods. The problems consist of the following.

R1. *An example where the Gauss-Newton method converges.*

$$f_1 = x_1 + x_2 + 1$$
$$f_2 = x_1 - x_2 - 1$$
$$f_3 = x_2^2 - x_2 + 1$$
$$f_4 = x_2^2 + x_2 - 1$$

FR1

Global minima in $[-4\times10^2, 4\times10^2]$: at $X^* = (0,0)$.

Remarks: $J^T(X^*)J(X^*)$ = diag(2,4), while $S(0,0)$ is the zero matrix. Thus, the Gauss-Newton method is locally convergent.

R2. *A simple example where the* ~~*Gauss-Newton method diverges.*~~ very convergently

$$f_1 = x_1 + x_2 + 1$$
$$f_2 = x_1 - x_2 - 1$$
$$f_3 = x_2^2 - x_2 + 1$$
$$f_4 = -x_2^2 + x_2 - 1$$

FR2

not determined at 0

[illegible] of x

PHI (F, G, H) [illegible]

This one may converge.

Write a routine to test it.

Global minima in $[-4\times10^2, 4\times10^2]$: at $X^* = (0,0)$.

Remarks: This problem differs from R1 only in the second-order term in f_4. Still, $J^T(X^*)J(X^*) =$ diag(2,4), but $S(0,0) =$ diag(0,4), so $\lambda = \tau = 4$. Numerical experimentation reveals that the Gauss-Newton method ~~is locally divergent~~ converges slowly in x_2, but that x_1 converges to 0 in one iteration.

R3. *A diagonal system for which the Gauss-Newton method diverges.*

$$f_1 = 10\,x_1^2 + x_1 + 1$$
$$f_2 = 10\,x_1^2 - x_1 + 1$$
$$f_3 = .25x_2^2 + x_2 + 1$$
$$f_4 = .25x_2^2 - x_2 + 1$$

Global minima in $[-4\times10^2, 4\times10^2]$: at $X^* = (0,0)$.

Remarks: $J^T(X^*)J(X^*) =$ diag(2,2), but $S(0,0) =$ diag$(\overset{40,\,1000}{20,.5})$, so $\lambda = 2$ and $\tau = \underset{40}{20}$. The Gauss-Newton method is clearly divergent in x_2 and convergent in x_1. This problem thus poses a test of the effectiveness of generalized bisection in complementing the interval Gauss-Newton method.

4.3 Some Additional and non-polynomial problems

Exponential nonlinear least squares problems sometimes occur in practice. With this in mind and for ease of coding, we have selected the following from the test set in [11].

E1. *Jennrich and Sampson Function (problem 6 in [11]):*

$$f_i = 2 + 2i - [\exp(ix_1) + \exp(ix_2)],$$
$$1 \le i \le 10.$$

Global minima in [0,2] × [0,2]: at $x_1 \simeq .25782$, $x_2 \simeq .25782$; Φ at minimum is less than 111.518.

E2. *Gaussian function (problem 9 in [11]):*

$$f_i = x_1 \exp\left[\frac{-x_2 \ (t_i - x_3)^2}{2}\right] - y_i$$

Global minima in [0,1] × [0,2] × [-.5,.5]: at $x_1 \simeq .3989561$, $x_2 \simeq .1000019$, $x_3 \simeq 0$; Φ at minimum is less than 1.0621×10^{-4}.

✓ put in table of x_i and t_i

5. NUMERICAL RESULTS AND CONCLUSIONS

We present algorithmic performance for the set of problems from [6] posed as nonlinear least squares problems in Table 1. Performance data for the simple problems with large residuals appears in Table 2. Performance data for the Jennrich/Sampson function and for the Gaussian fitting function appear in tables 3 and 4, respectively.

4.1 Explanation of the tables

In all of the tables, we have the following.

NFTEST is equal to the number of sub-boxes of the original box which were tested for stationary points during the generalized bisection process;

NFUN represents the total number of interval function evaluations;

NJAC represents the total number of interval Jacobian matrix evaluations (and Hessian matrix evaluations, in the case of the exact Hessian method).

The estimated total work is defined to be n NFUN + m n NJAC for the heursitic method (HNLS), deterministic method (DNLS), and singular value decomposition method (SVD), where n is the number of variables and m is the number of functions. For the exact Hessian method (XHES), the estimated total work is defined to be n NFUN + m n $(1 + n)$ NJAC. These measures are based on the assumption that each partial derivative is of equal difficulty to evaluate as a single component of the function.

The test effectiveness E is a measure of how much the diameter of the box must be reduced by splitting along a coordinate direction before either $K(X) \cap X$ = is empty or else $K(X) \subset X$. It is defined by

$$(5.1) \qquad E = 2^{-[L/n]}$$

where L is the maximum level in the binary tree defined by the coordinate splitting process. In instances where each coordinate direction is bisected an equal number of times and iteration of the interval Newton method gives no reduction in the widths before $K(X) \subset X$, E represents the factor of relative diameter reduction necessary in the bisection process. Thus, larger values indicate the method is more effective. For example, since the initial coordinate widths for the problems in tables 1 and 2 are 800 and since the minimum box width is $10^{-5}/16 = 6.25 \times 10^{-7}$, the minimum possible E in these tables is 7.8×10^{-10}.

The number of variables n and the number of functions m are equal in all of the problems in Table 1; there, n is given in the second column. Also in Table 1, the first column identifies the problems by the numbers they are given in [6].

As is indicated in Section 4 above, $n = 2$ and $m = 4$ for all of the problems in Table 2. Likewise, $n = 2$ and $m = 10$ for Table 3, and $n = 3$ and $m = 15$ for Table 4.

5.2 Analysis of the results

5.2.1 FOR THE SQUARE NONLINEAR SYSTEMS TEST SET

In Table 1, the heuristic method is reliable since it reduces to the usual interval Newton method when $m = n$. We note that, in all five measures of efficiency and for all problems, it is better than the other methods. (For instance, the deterministic method needed to consider 11.4 times as many boxes.) This is forced partly by the fact that the condition number of J^TJ is the square of that of J. In particular, if

$$Y = \{m[J^T(X)J(X)]\}^{-1} = A^{-1}$$

where m(*) denotes the midpoint matrix, then

$$Y\ J^T(X)J(X) = Y\ (A + S) = I + H\ S,$$

where S is the matrix of widths of the elements of $J^T(X)J(X)$. If the elements of S are all of the form $[-a,a]$, then there are elements of $H\ S$ of the form $[-b,b]$, where $b = \|H\|_\infty$.

Another interesting observation from Table 1 is that, overall, the deterministic nonlinear least squares method (i.e. the interval Gauss Newton method) took only about 70% as much work as the exact Hessian method, when the additional derivative evaluations are not taken into account, and only 56% as much work when they are. The exact Hessian method needed to consider slightly fewer boxes in problems 9, 12, and 15, but was worse in other measures.

Finally, overall in Table 1, the singular value decomposition method seems to be at least 17% more efficient than the deterministic method, when the costs of the extra linear algebra computations are not included. Exceptions are problems 2 and 16.

5.2.2 FOR THE MODEL PROBLEMS WITH LARGE RESIDUALS

The solution to each of these problems lies at $x_1 = x_2 = 0$. However, the heuristic method signalled a unique root at $x_1 \simeq 0$, $x_2 \simeq -1$ in problems R1 and R2, although it did give the correct result for problem R3. Therefore, the heuristic method should not be seriously considered in this context.

Overall in these problems, the exact Hessian method did not need to consider as many boxes as either the deterministic method or the singular value decomposition method. This would be expected since these are large residual problems. The exception to this was problem R1, for which the Gauss-Newton method converges. However, the estimated total work, which takes account of the extra cost of evaluating second derivatives, was greater in all cases. Also, the number of function and Jacobian evaluations was greater except for R2; this indicates that, though the interval Newton method with the exact Hessian matrix converged, its effectiveness at reducing the box size was less than the Gauss-Newton based methods combined with generalized bisection.

We note also that, for the deterministic method, the test effectiveness differed from that for the exact Hessian method by less than one order of magnitude; in any case it was always at least 10^6 greater than the value it would attain if we applied pure generalized bisection, without reducing the box widths by iteration of $X \leftarrow K(X) \cap X$. We thus conclude that our interval extension of the Gauss-Newton method is effective even in cases where the Gauss-Newton method is non-convergent.

By all measures, the deterministic method was significantly cheaper than the singular value decomposition method for these three problems. In contrast to some of the problems from [6], the quantity $J^T J$ is well-conditioned in problems R1, R2, and R3. Thus, the primary consideration in the performance difference is the widening of intervals introduced by transforming between coordinate systems in the singular value decomposition method.

5.2.3 FOR THE EXPONENTIAL FITTING PROBLEMS

Note that problem E1 is a large-residual problem, whereas problem E2 is a small residual problem.

For both problems, the heuristic method incorrectly concluded that there were no stationary points of Φ within the initial box. We thus should ignore the performance results for the heuristic method.

For the large-residual problem, the deterministic method. exact Hessian method, and singular value decomposition method were comparable in efficiency when second-derivative evaluations were not counted. The deterministic method was significantly better by all measures for the small-residual problem.

5.3 Conclusions

The heuristic method is highly superior for square systems, where it gives rigorous results. (General nonlinear systems should not be recast in terms of the normal equations.)

However, it does not give meaningful results in other instances, and should not be considered for them.

In general, the exact Hessian method is not as efficient as the interval extensions of the Gauss-Newton method when they are both combined with generalized bisection. This is even true for some systems for which the point version of the Gauss-Newton method is divergent. It is especially so when we take account of the extra work needed to evaluate second derivatives for the exact Hessian method.

The deterministic method is superior to the singular value decomposition method for problems for which the matrix J^TJ is ill-conditioned. However, the singular value decomposition method can be better in some cases. We conjecture that, within a given problem, preconditioning with $(J^TJ)^{-1}$ may work better for some subboxes, while preconditioning with the singular value decomposition may be more appropriate for others. A hybrid algorithm based on K_h, where

$$K_h(X) = K(X) \cap \overline{K}(X),$$

where K is the interval solution of (2.3), and where $\overline{K}$ is the interval solution of (2.7), may be more efficient than either the deterministic method or the singular value decomposition method alone.

6. SUMMARY AND FURTHER WORK

We have compared two interval extensions of the Gauss-Newton method to the interval Newton method obtained by using an interval evaluation of the exact Hessian matrix. For problems with n functions and n variables, we have also compared these interval extensions of the Gauss-Newton method to "heuristic method", which reduces to a usual interval Newton method.

The problem set included a battery of problems from the literature with n functions and n variables, a set of three problems to test behavior in cases for which the usual Gauss-Newton method diverges, and a set of two nonlinear exponential fitting problems.

The interval Gauss-Newton methods compare favorably with the method based on the exact Hessian matrix. They also seem practical for exponential fitting applications. They would be especially desirable in this context when rigorous bounds on the optimal parameters are required or when there isn't a unique local optimum.

Portable interval sofware for nonlinear systems is about to be made available. (See [7].) Addition of several modules, including generic exponential fitting function and Jacobian routines, would make use of the techniques described here practical.

Additional research needs to be done on the behavior of the algorithms on more general problems.

Acknowledgement: I wish to thank David Gay for the conversation which lead me to consider coordinate transformations in the context of interval Gauss-Newton methods.

Bibliography

1. Alefeld, G., and Herzberger, J. *Introduction to Interval Computations*, Academic Press, New York, etc., 1983.

[illegible]

2. Dennis, J. E., and Schnabel, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Least Squares*, Prentice-Hall, Englewood Cliffs, NJ, , 1983.

3. Hansen, E. R., and Sengupta, S. *Bounding solutions of systems of equations using interval arithmetic, BIT 21* (1981), 203-211.

4. Hansen, E. R., and Walster, G. W. "Nonlinear equations and optimization", manuscript, Lockheed MSC, Dept. 62-81, Bldg. 563, P. O. Box 504, Sunnyvale, CA 94086

5. Kearfott, R. B. "Abstract generalized bisection and a cost bound", *Math. Comput. 49*, 179 (July, 1987), 187-202.

6. Kearfott R. B. "Some tests of generalized bisection", *ACM Trans. Math. Software 13*, 3 (Sept., 1987).

IMACS...

7. Kearfott, R. B.~~, and Novoa, M.~~ "A program for generalized bisection", in manuscript.

8. Moore, R. E. A test for existence of solutions to nonlinear systems, *SIAM J. Numer. Anal. 14*, 4 (Sept. 1977), 611-615.

9. Moore, R. E. *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

10. Moore, R. E., and Qi, L. A successive interval test for nonlinear systems, *SIAM J. Numer. Anal. 19*, 4 (Aug. 1982), 845-850.

11. More', J. J., Garbow, B. S., and Hillstrom, K. E. Testing unconstrained optimization software, *ACM Trans. Math. Software* 7, 1 (Mar. 1981), 17-41.

12. Morgan, A. P. *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems*, Prentice-Hall, Englewood Cliffs, 1987.

13. Neumaier, A. Interval iteration for zeros of systems of equations, *BIT 25*, 1 (1985), 256-273.

14. Qi, L. "A note on the Moore test for nonlinear systems", *SIAM J. Numer Anal. 19*, 4 (Aug., 1982), 851-857.

15. Rall, L. B. *Automatic Differentiation: Techniques and Applications*, Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg, New York, 1981.

16. Shearer, J. M., and Wolfe, M. A. "ALGLIB, a simple symbol manipulation package", *Comm. ACM 28*, 8 (Aug., 1985), 820-825.

						est. tot.	
no.	n	method	NFTEST	NFUN	NJAC	work	eff.
1	2	HNLS	8	22	21	128	5.0E-01
		DNLS	77	124	98	640	1.3E-01
		XHES	93	144	111	1574	1.3E-01
		SVD	77	121	94	618	1.3E-01
2	2	HNLS	42	69	52	346	1.8E-01
		DNLS	191	213	150	1026	4.4E-02
		XHES	206	230	158	2344	4.4E-02
		SVD	207	224	157	1076	4.4E-02
9	2	HNLS	3	12	12	72	7.1E-01
		DNLS	143	187	175	1074	7.8E-03
		XHES	128	211	207	1250	1.1E-02
		SVD	81	124	124	744	3.9E-03
12	3	HNLS	438	773	617	7872	1.3E-01
		DNLS*	4068	4560	3161	42129	3.1E-04
		XHES*	3707	4419	2992	40185	7.8E-03
		SVD	2684	3547	2594	33987	9.8E-03
13	3	HNLS	1	1	1	12	7.9E-01
		DNLS	1	1	1	12	7.9E-01
		XHES	1	1	1	12	7.9E-01
		SVD	1	1	1	12	7.9E-01
14	2	HNLS	14	34	32	196	3.5E-01
		DNLS	99	132	93	636	4.4E-02
		XHES	110	144	101	692	4.4E-02
		SVD	73	112	89	580	6.3E-02
15	2	HNLS	1	2	2	12	7.1E-01
		DNLS	623	712	554	3640	2.0E-03
		XHES	597	729	602	3866	9.8E-04
		SVD	228	294	263	1640	4.4E-02
16	4	HNLS	1	2	2	40	8.4E-01
		DNLS	9	15	15	300	5.0E-01
		XHES	25	36	34	688	3.5E-01
		SVD	83	84	84	1680	2.1E-01
17	5	HNLS	149	225	212	6425	4.4E-01
		DNLS	2281	2391	1512	49755	1.9E-01
		XHES	5813	5977	3826	125535	1.3E-01
		SVD	1787	1817	1422	44635	1.1E-01
OTALS:		HNLS	657	1140	951	15103	4.4E-01
		DNLS	7492	8335	5759	99212	3.5E-02
		XHES	10680	11891	8032	176146	4.5E-02
		SVD	5221	6324	4828	84972	6.0E-02
atios:							
DNLS/HNLS:			11.40	7.31	6.06	6.57	0.08
DNLS/XHES:			0.70	0.70	0.72	0.56	0.79
DNLS/SVD:			1.43	1.32	1.19	1.17	0.59

able 1. Comparison of the four nonlinear least squares methods when applied to some problems with square Jacobians.

					est. tot.	test
no.	method	NFTEST	NFUN	NJAC	work	effectiveness
R1	HNLS	1	1	1	12	7E-01
	DNLS	21	76	62	800	3E-03
	XHES	25	79	63	1828	1E-03
	SVD	94	99	90	1116	2E-04
R2	HNLS*	1	1	1	12	7E-01
	DNLS	19	51	42	540	6E-02
	XHES	6	29	26	740	5E-01
	SVD	28	60	43	584	4E-02
R3	HNLS	457	457	403	5052	1E-09
	DNLS	53	133	87	1228	2E-04
	XHES	53	199	153	4468	7E-04
	SVD	187	263	209	2724	2E-04
OTALS:	HNLS	459	459	405	5076	9E-04
	DNLS	93	260	191	2568	3E-03
	XHES	84	307	242	7036	8E-03
	SVD	309	422	342	4424	1E-03
atios:						
NLS/HNLS:		0.2026	0.5664	0.4716	0.5059	3.56359
NLS/XHES:		1.1071	0.8469	0.7892	0.3649	0.39685
NLS/SVD:		0.3009	0.6161	0.5584	0.5804	2.51984

Table 2. Comparison of the four methods on simple functions for which the Gauss-Newton method diverges. See the text for additional explanation.

				est. tot.	test
method	NFTEST	NFUN	NJAC	work	effectiveness
HNLS*	8	9	7	230	3E-01
DNLS	813	829	791	24110	4E-06
XHES	813	829	791	55750	4E-06
SVD	817	918	892	27020	9E-05
Ratios:					
DNLS/HNLS:	101.63	92.11	113.00	104.83	0.00001
DNLS/XHES:	1.00	1.00	1.00	0.43	1
DNLS/SVD:	1.00	0.90	0.89	0.89	0.04419

Table 3. Results for the Jennrich/Sampson nonlinear exponential fitting problem.

				est. tot.	test
method	NFTEST	NFUN	NJAC	work	effectiveness
HNLS*	561	596	515	32115	4E-02
DNLS	261	317	278	17265	1E-01
XHES	314	358	314	61890	8E-02
SVD	355	386	335	20865	6E-02
atios:					
NLS/HNLS:	0.47	0.53	0.54	0.54	2.51984
NLS/XHES:	0.83	0.89	0.89	0.28	1.25992
NLS/SVD:	0.74	0.82	0.83	0.83	1.58740

Table 4. Results for a nonlinear Gaussian fitting problem.