HOW UP-TO-DATE ARE LOW-RANK UPDATES?

Andreas Griewank† and Andrea Walther
Institute for Scientific Computing, Technical University Dresden, Dresden, Germany

ABSTRACT
For several decades quasi-Newton methods based on low-rank secant updates have been widely applied to many small to medium sized nonlinear equations and optimization problems. Their adaptation to large and structured problems has not always been successful. We review some convergence results for secant methods and some examples regarding the cost of derivative matrices, report some recent results from a parallel implementation of Broyden's method, and propose an unsymmetric rank-one Jacobian update based on direct and adjoint derivative information. It may be applied in particular to Jacobians in constrained optimization, either with full storage or in a limited memory version. We report some numerical results on a discretized second order ODE and conclude with an outlook on future developments.

Key words: secant updates, automatic differentiation, constrained optimization, compact perturbation, Broyden update.

MSC: 65D25, 49M37, 65Y99.

RESUMEN
Los métodos quasi-Newton basados en actualizaciones de la secante de bajo rango han sido ampliamente usados por varias décadas para resolver ecuaciones no lineales y problemas de optimización de pequeña o mediana dimensión. La adaptación de estos métodos a la solución de problemas extensos y estructurados no ha sido siempre exitosa.

1. INTRODUCTION AND MOTIVATION

The Davidon-Fletcher-Powell (DFP) formula was proposed nearly fifty years ago [Davidson, (1959)] and stimulated a lot of research into related updating formulas during the following two decades. In practical implementations it was superseded by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula, which is still extensively used in computer codes for unconstrained and constrained optimization. DFP or BFGS and convex combinations of these rank-two formulas are useful for approximating symmetric positive definite Hessian matrices of moderate dimensions, say in the hundreds. For larger problems so-called limited memory variants haven been developed [Nocedal (1980)] which are currently widely considered the most effective general purpose technique for large-scale unconstrained minimization. For the solution of systems of nonlinear equations the unsymmetric rank-one update formula due to [Broyden (1965)] yields similar advantages, though its has not been quite as successful as BFGS in the symmetric positive definite case. One possible explanation is that the Broyden formula is strongly dependent problem scaling whereas the BFGS and DFP are in some sense invariant with respect to linear transformations on the variables, is also an advantage of Newton's method in the unsymmetric case. The same desirable property is achieved by the two-sided-rank-one formula (TR1) described in Section 4 of this paper. All these schemes are referred to as secant methods because they obtain information about.

In the past two decades the development of equation solvers and optimization algorithms has been primarily aimed at the treatment of large and very large problems. In either case, we face locally the problem of finding the root of a vector function F: $\mathbb{R}^n \to \mathbb{R}^n$. Its Jacobian is square and in the optimization case also symmetric, i.e. a Hessian. At first thought the formulas mentioned above might seem unsuitable for large-scale purposes because they immediately destroy sparsity, which is frequently considered a typical property of problems in very many variables. Also, to approximate a square Jacobian or Hessian of order n from a poor initial guess by a sequence of rank-one or rank-two corrections one must take in general n or n/2 update steps, respectively. However, especially on discretizations of operator equations one rarely wants to take more than a hundred steps even if n is in the thousands. For Newton's method one can establish under fairly general assumptions on the operator equation and its discretization [Allower **et al**. (1986)] that the number of steps needed to obtain a certain solution accuracy is completely independent of the mesh-size and thus the dimension n. In other words, the key properties of Newton's method and especially its Q-quadratic rate of convergence are mesh-invariant. This means that on all sufficiently fine grids the sequence of Newton iterates closely corresponds to the one theoretically generated by Newton's method on the underlying problem in function space.

Like linear invariance this highly desirable property is not necessarily shared by the secant methods mentioned above. More specifically, the Q-superlinear rate of convergence established for a wide variety of low-rank updating methods by Broyden, Dennis and Morè (1973) in a finite dimensional setting does not automatically generalize even to separable Hilbert spaces. Counter examples in the space l_2 of square summable real sequences were given in Stoer (1984) and Griewank (1987). The key difficulty is that the unit sphere of directions is no longer compact in l_2 and all other infinite dimensional linear spaces. As a consequence the operator discrepancy between the approximating Jacobians or Hessians and their exact values need not become small, no matter how many updates haven been performed and even if they are applied in orthogonal, or more generally conjugate, directions.

To recover guaranteed Q-superlinear convergence one has to assume that already the initial operator discrepancy is relatively compact. More specifically, the singular values of the difference between the identity operator and the Jacobian or Hessian at the solution, premultiplied by the inverse of their initial approximations, must converge to zero. In other words, we must be able to precondition the given gradient or other vector function such that it is a compact perturbation of the identity. This situations arises for example in the Chandrasekhar equation

$$F(x)(\mu) = x(\mu) - \left(1 - \frac{c}{2}\int_0^1 \frac{\mu x(v)dv}{\mu + v}\right)^{-1} = 0.$$

which we will use as a test problem. The same is true for much more interesting applications, for example in optimal control of PDEs [Hinze-Kunish (1999)]. The intimate connection between preconditioning and Jacobian or Hessian initialization is often not fully appreciated.

Let us suppose the compactness assumption is satisfied - what then are the advantages of secant updating methods? Probably the foremost motivation for the design of secant updating methods was the avoidance of derivative evaluations in the nonlinear case. However, besides this considerable increase in user-friendliness compared to Newton's method, there is another very significant gain, namely the reduction of the linear algebra effort per iteration from order n^3 to order n^2 in the dense case. For limited memory implementations the storage requirement and per step operations count drop down further to O(kn), where k is the number of iterations since the last reset. These very significant savings may be important even on linear problems, where the evaluation of the Hessian of Jacobian is usually no issue at all. To bolster our conclusion that low-rank updates may be useful even on large-scale problems we discuss the costs of Newton's method in Section 2, review some classical results on secant methods in Section 3, introduce the new two-sided rank-one update in Section 4 and consider possible further developments in the final Section 5.

2. ON THE COST OF NEWTON'S METHOD

To decide whether and when updating methods are competitive, one has to consider the alternatives, namely Newton's method and its inexact variants. Exact Newton steps for a square system $F(x) = 0 \in \mathbb{R}^n$ are usually obtained by explicitly forming the Jacobian $F'(x) \in \mathbb{R}^{n\times n}$ and then factoring it into the product of two triangular matrices. Inexact methods typically compute the Newton-step approximately by an inner iteration based on Jacobian-vector products and possibly vector-Jacobian products. The convergence properties of the popular Krylov subspace methods, especially CG for the symmetric positive definite and GMRES for the general case, are dependent on the eigenvalues or singular values of the discrepancy between the possibly preconditioned Jacobians and the identity. Convergence of the inner iteration in much fewer than n steps can again only be expected if the eigen -or singular- values are heavily clustered, usually at zero. This is exactly the situation where secant updating methods perform also well in terms of the number of (outer) iterations. If on the other hand, the number of inner iterations is a significant fraction of n, then one might as well use the exact variant of Newton's method to which we will restrict our attention from now on.

For general nonlinear problems the cost of forming and factoring a Jacobian or Hessian is still almost impossible to estimate a priori. There are obvious upper bounds, namely the derivative matrix F'(x) can be estimated at roughly n times the cost of evaluating the residual F(x) itself and a LU factorization involves $n^3/3$ fused multiply-adds. For large n these costs are likely to be prohibitive, so that one will wish to detect and exploit sparsity and other problem structure. Using well-known matrix-compression techniques [Griewank (2000)] one can reduce the cost factor for evaluating a sparse Jacobian by differencing or using the forward mode of automatic differentiation (AD) from n to $\hat{n}$, the maximal number of nonzero elements per row of this matrix. In the so-called reverse, or adjoint, mode of AD the same compression technique can be

applied column-wise, in which case the cost factor is roughly equal to m, the maximal number of nonzero entries per column. Hence we may summarize

$$\frac{OPS(F'(x))}{OPS(F(x))} \le 5 \min(\hat{n},\hat{m}) \le 5 \min(n,m) \qquad (1)$$

where we have also allowed for the rectangular case $F' \in \mathbb{R}^{m \times n}$ with $m \le n$. Of particular interest is the case $m = 1$, where F' is a gradient. It is then obtained in the reverse mode of AD at essentially the same cost as the underlying scalar function F(x).

In general, the upper bound (1) is by no means a good estimate of the minimal cost for computing the Jacobian. For instance on square problems with some (nearly) dense rows and columns, as for example an arrowhead matrix, the right hand side is 5n but combinations of row and column compression yield the Jacobian at a much lower cost [Griewank (2000)]. As first demonstrated in Griewank (1993)] the relative cost of evaluating a Jacobian depends not only on its sparsity pattern, but on the internal structure of the given procedure for evaluating the underlying vector-function. The attempt to minimize the operations count for computing a Jacobian or Hessian leads to a hard combinatorial optimization problem [Griewank-Naumann (2002). It seems doubtful whether too much effort should be invested in solving this meta-problem, especially since exact Newton-steps can sometimes be computed more directly by factoring an extended system without forming the Jacobian at all [Griewank (1990)]. Rather than trying to analyze the cost of Jacobians, Hessians and Newton-steps in general, we will just highlight the issues on an algebraically simple example.

Consider the following generalization of an example used by Speelpenning, which occurs similarly as Cobb-Douglas function [Douglas (1934)] in the economics literature

$$f(x) = \prod_{j=1}^{n} x_j^{\gamma_j}.$$

Assuming that all variables x_j are nonzero one can express the gradient and Hessian of f as

$$g(x) = \nabla f(x) = f(x)\,(\gamma_j / x_j)_{j=1}^{n}$$

and

$$H(x) = \nabla^2 f(x) = g\, g^T / f \; - D$$

where $D = f\,\mathrm{diag}\,(\gamma_j / x_j^2)_{j=1}^{n}$ is diagonal. By inspection one deduces the operations counts

$$OPS(f) \sim 2n,\ OPS(g) \sim 4n,\ OPS(H) \sim \tfrac{1}{2} n^2$$

As the Hessian has in general $n^2/2$ distinct entries, which all vary as a function of x, the last estimate must apply to any method for evaluating the Hessian, whatsoever. Hence we see that in agreement with (1) the gradient g has essentially the same complexity as the underlying function f but the Hessian is roughly n times as expensive. However, this is really only true if we insist on computing the Hessian as a symmetric $n \times n$ array of real numbers, which is in some sense a redundant and thus inappropriate representation. H instead we compute and store simply D in addition to f and g, then not only the application of H to any vector r can be computed with effort of order n, but the same is true for its inverse according to the Sherman-Morrison-Formula

$$-H^{-1}r = \left(I - \frac{D^{-1} g\, g^T}{f + g^T D^{-1} g} \right) D^{-1} r.$$

The situation is slightly different if we consider a vector function $F(x) = (Fi(X))_{i=1,\ldots,m}$ with the component functions

$$F_i \equiv \prod_{j=1}^{n} x_j^{\gamma_{ij}} \quad \text{for } i = 1,\ldots,m.$$

Then the Jacobian F'(x) is formed by the gradients of the F_i and we obtain the cost estimate

$$OPS(F'(x)) = \sum_{i=1}^{m} OPS(\nabla F_i) \sim m\, n \sim OPS(F).$$

In other words the whole Jacobian F´(x) has about the same complexity as the underlying vector function F(x). In general this equivalence is by no means true, and even here it applies only under the tacit assumption that the exponents γ_{ij} are largely distinct. If they are not certain common subexpressions can be evaluated only once for several components $F_i(X)$, while the cost for evaluating the gradients ∇F_i need not go down accordingly.

To see this consider for example the square case with $\gamma_{ij} = |i - j|$. Than we have the identity $F_i = F_{i-1}\ (x_1 x_2 \ldots X_{i-1})/(x_i x_{i+2} \ldots x_n)$ which obviously allows the evaluation of F in O(n) arithmetic operations. With a little care all divisions can be avoided so that no branching is A needed if one of the x_j is zero or very small. The Jacobian on the other hand has still n^2 distinct entries given by $(\partial F_i/\partial x_j = F_i\, |i - j| / x_j$ for $1 \le i, i \le n)$. Hence computing the Jacobian explicitly is one order of magnitude more expensive than evaluating the underlying vector function F(x). However, as in the symmetric case above we see that an explicit computation of F´(x) is inappropriate for the following reason. F´(x) equals the Toeplitz matrix $(|i - j|)_{1\le j\le n}$ pre- and post-multiplied by the matrices $diag(F_i)$ and $diag(1/x_j)$, respectively. Using superfast solvers for Toeplitz systems [Bojanczyk, K. **et al**. (1995)] Newton steps can thus be computed at a cost of $O(n \log^2 n)$ operations. Obviously it would be a rather formidable challenge to develop a general methodology for detecting and exploiting this kind of structure in a vector-function given by an evaluation procedure.

For general exponents γ_{ij} and without the use of fast solvers of the Strassen type, the cost of factorizing the dense Jacobian F'(x) and thus the cost of computing an exact Newton step will be cubic in n and hence still an order of magnitude more expensive than evaluating that square matrix. Thus we conclude that it still makes plenty of sense to persue approaches that avoid forming and factoring Jacobians.

3. SOME RESULTS ON BROYDEN'S METHOD

Throughout this section subscripts represent iteration counters rather than vector components. Provided full steps are taken, all Newton-like methods take the form

$$x_{k+1} = x_k + s_k \qquad \text{with} \qquad -A_k s_k = F_k \equiv F(x_k)$$

where

$$A_k \approx F_k' \approx \frac{\partial F(x_k)}{\partial x} \in \mathbb{R}^{n\times n}$$

After each step the approximation A_k is updated to a new version satisfying the so-called secant condition

$$A_{k+1} s_k = y_k \equiv F_{k+1} - F_k \in \mathbb{R}^n$$

This linear equation is satisfied by an affine subspace of dimension $n^2 - n$, from which Broyden (1965) suggested to select the matrix closest to the current A_k with respect to the Frobenius norm. As a result one obtains

$$A_{k+1} = A_k + r_k s_k s_k^T \in \mathbb{R}^{n\times n} \quad \text{with } r_k \equiv y_k - A_k s_k = F_{k+1} \tag{2}$$

which is clearly a rank-one correction, or update, of A_k. Applying the same arguments to the inverse updating problem $A_{k+1}^{-1} y_k = s_k$ one arrives after some manipulation at the so called "bad" Broyden formula

$$A_{k+1} = A_k + r_k y_k^T A_k / y_k^T A_k s_k \tag{3}$$

Since for both formulas $(A_{k+1} - A_k)s_k = r_k$, one sees immediately that either the new residual $r_k = F_{k+1}$ is small -indicating progress towards the solution- or there is a sizable correction from A_k to A_{k+1} along s_k. In finite

dimensional Euclidean space it then follows under mild I regularity assumptions [Broyden **et al**. (1973] that we have local Q-superlinear convergence in that

$$\frac{\left\| r_{k+1} \right\|}{\left\| s_k \right\|} \underset{k}{\to} 0 \qquad \text{and} \qquad \frac{\left\| x_{k+1} - x_* \right\|}{\left\| x_k - x_* \right\|} \underset{k}{\to} 0$$

provided x_0 and A_0 are sufficiently close to a root $x_* \in F^{-1}(0)$ and the Jacobian $F'(x_*)$, respectively. Using a suitable line-search or trust-region stabilization these last two locality conditions can be significantly relaxed.

To understand the function space situation let us consider the discrepancies

$$D_k \equiv A_k - F'(x_*) = D_0 + \sum_{j=0}^{k-1} r_j s_j^T$$

which can be interpreted in the separable Hilbert space l^2 in the obvious manner. Since the accumulated corrections $D_k - D_0$ have rank k it is clear that the essential norm

$$\sigma_{ess}(D_k) \equiv \inf\{ \left\| D_k - C \right\| \mid \text{linear } C\text{: } l^2 \mapsto l^2, \text{ rank}(C) < \infty \}$$

remains constantly equal to $\sigma_{ess}(D_0)$. The essential norm is important because it has been shown [Griewank, (1987)] that

$$\limsup \left\| F_k \right\|^{1/k} = \limsup \left\| x_k - x_* \right\|^{1/k} \leq \sigma_{ess}(D_0)$$

with equality being obtained for almost all initial conditions. Hence it is usually necessary even for R-superlinear convergence -which is always implied by Q-superlinear convergence– that $\sigma_{ess}(D_k) = 0$. Provided A_0 has a bounded inverse this condition is equivalent to $\sigma_{ess}(A_0^{-1}D_0) = 0$ and equivalently $I - A_0^{-1}F'(x_*)$ being a compact operator, which is true exactly when its singular values converge to zero.

The theoretical results are confirmed by the convergence histories plotted in Figure 1 and Figure 2. The Chandrasekhar integral equations was discretized by central differences yielding a nonlinear system of equations with a dense Jacobian, which is a compact perturbation of the identity I to which A_0 was initialized. The slope of the logarithmized of the residuals norm plotted against the iterations counter becomes quite steep indicating superlinear convergence, before round-off invalidates the asymptotic analysis. The plot was obtained for a grid size of 0.01 but was essentially the same for all sizes smaller than 0.1.

The residual convergence history in Figure 2 was obtained on a central difference discretization of the linear convection diffusion equation

$$0 = -\Delta u + (v,w)\nabla u = \nabla u = f \text{ on } \Omega$$

This time the initialization $A_0 = I$ is very poor since the discrepancy between the identity and the Laplace operator is not even bounded, let alone compact.

As a consequence the convergence is very slow and strongly dependent on the grid-size. The run-time became also very long because towards the end our limited memory implementation along the lines of [Deuflhard **et al**. (1990)] had to store and manipulate more than 300 vectors in each iterations. In contrast this was not a problem on the Chandrasekhar equation, where only 16 vectors needed to be stored until full working accuracy in the solution had been achieved.

Hence we may conclude that secant updating still makes sense on large problems, provided they are relatively compact perturbations of an invertible linear operator. Considerable efforts were undertaken to incorporate sparsity into secant updating. While thus is quite easy theoretically, the practical results were mostly unconvincing. Since the resulting updates are usually of full rank all savings with regards to the Newton step calculation is lost, so that one arrives in some sense at the worst of both worlds, the step calculation is expensive and the Jacobian approximation is still not very good.

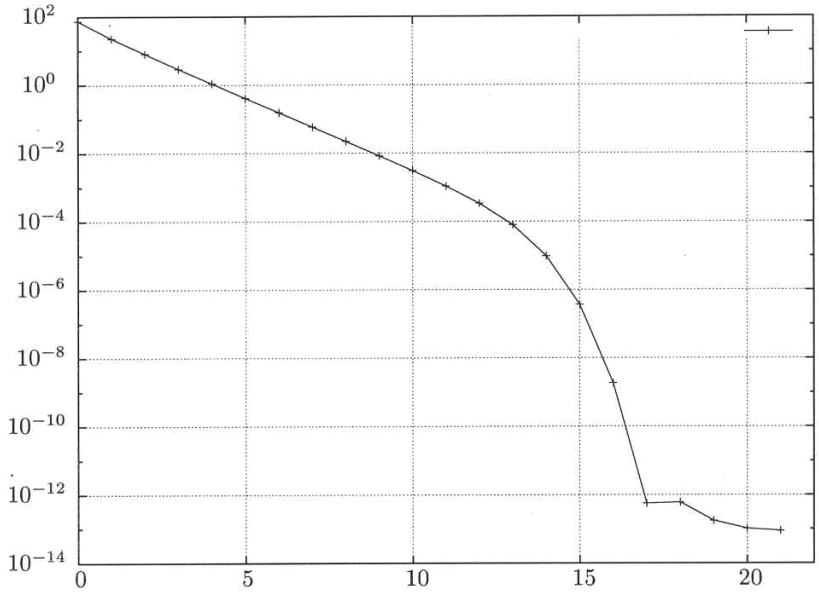

Figure 1. Residual Convergence of Aroyden on Chandrasekhar Equation where Compactness Condition is Satisfies.

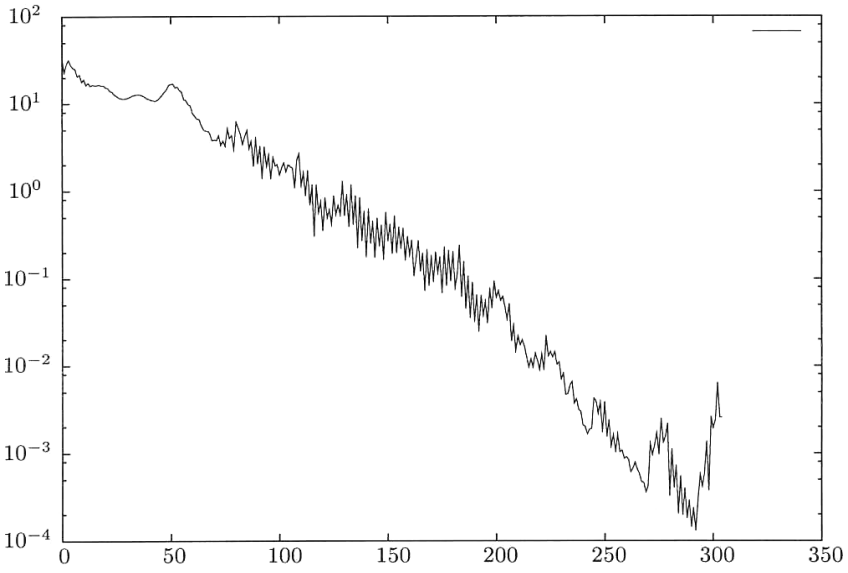

Figure 2. Residual Convergence of Broyden on Convenction Diffusion Equation where Compactness Condition is Violated.

4. THE TWO-SIDED-RANK-ONE UPDATE (TRL)

In view of the very intensive research activity on secant updates in the sixties, seventies, and eighties, it may seem rather belated and even preposterous to propose yet another formula. However, there is one crucial aspect that has changed through the advent of reverse mode automatic differentiation, namely, the availability of adjoint secant information as defined below. To derive and motivate the new update we consider the classical problem of equality constrained optimization.

Minimizing a scalar objective function f(x) subject to a vector constraint $c(x) = 0 \in \mathbb{R}^m$ is locally equivalent to computing saddle points of the Lagrange function

$$L(x, \lambda) \equiv f(x) + \lambda^T c(x).$$

In other words one has to solve the nonlinear KKT-system

$$0 = [g(x, \lambda), c(x)] \equiv [\nabla f(x) + \lambda^T c'(x), c(x)] \in \mathbb{R}^{n+m} \tag{4}$$

According to (1) the gradient $g(x, \lambda) = L_z(x, \lambda)$ can be evaluated with an effort

$$OPS\{g(x, \lambda), c(x)\} \sim OPS\{f(x), c(x)\},$$

whereas the evaluation of the complete constraint Jacobian c'(x) may well be m times as expensive. In other words, even though the matrix c'(x) occurs explicitly in the algebraic definition of the vector $g(x, \lambda)$, the latter may be an order of magnitude cheaper to obtain than the former.

To illustrate this crucial effect we consider again the example

$$c_i(x) \equiv \prod_{j=1}^{n} x_j^{|i-j|} = c_{i-1}(x) \prod_{j=1}^{i-1} x_j \Big/ \prod_{j=i}^{n} x_j$$

Hence $c \in \mathbb{R}^m$ for some $m \le n$ can clearly be evaluated with an effort of O(n). However, c' is dense and therefore costs at least mn arithmetic operations. In contrast the vector-Jacobian product $\lambda^T c'(x) = (h_j/x_j)_{j=1...n}$ can also be calculated at a cost of O(n) using the recurrence

$$h_j \equiv \sum_{i=1}^{n} \lambda_i |i - j| c_i = h_{j-1} + \sum_{i=1}^{j-1} \lambda_i c_i - \sum_{i=j}^{n} \lambda_i c_i .$$

Subsequently the h_j can be divided by x_j to yield the jth component of $\lambda^T c'(x)$. It must be stressed once more that any division can be avoided, both in the evaluation of the c_i and in the resulting reverse mode procedure for $\lambda^T c'(x)$ generated automatically.

Whether or not the Jacobian $c'(x)$ is significantly more expensive than $g(x, \lambda)$, we also have to consider the cost of factorizing the constraint Jacobian $c'(x)$, which is needed to obtain a suitable representation of its nullspace. In the last example with m a certain fraction of n, an LU or QR factorization of the Jacobian will be one order of magnitude more expensive than its evaluation. Hence it makes sense to consider secant methods simply for the purpose of reducing the linear algebra effort. Suppose we have approximations

$$\left[B_k, A_k^T\right] \approx g'(x,\lambda) = \left[\nabla^2 f + \sum \lambda_i \nabla^2 c_i(x), c'(x)^T\right] \in \mathbb{R}^{n\times(n+m)}$$

where B_k and A_k will be referred to as approximate Hessian and Jacobian, respectively. Here and throughout the remainder of the paper subscripts represent again itaration counters. Then the quasi-Newton step (s_k, σ_k) on the KKT system (4) is defined as solution of the linear system

$$\begin{bmatrix} B_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} s_k \\ \sigma_k \end{bmatrix} = - \begin{bmatrix} g_k \\ c_k \end{bmatrix} \tag{5}$$

where $g_k = g(x_k, \lambda_k)$ and $c_k \equiv c(x_k)$ at the current point $(x_k, \lambda_k) \in \mathbb{R}^{n+m}$. After the step to $(x_{k+1}, \lambda_{k+1}) = (x_k, \lambda_k) + (s_k,\sigma_k)$ we obtain new values $g_{k+1} \in \mathbb{R}^n$ and $c_{k+1} \in \mathbb{R}^m$, which immediately yields the direct secant condition

$$A_{k+1}s_k = y_k \equiv c_{k+1} - c_k \approx c'(x_k)s_k \tag{6}$$

For updating the Hessian we'll also assume that the intermediate gradient $g_{k+1/2} = g(x_{k+1}, \lambda_k)$ has been evaluated, possibly as part of a line-search procedure. Then we may impose the secant condition

$$B_{k+1}s_k = w_k \equiv g_{k+1/2} - g_k \in \mathbb{R}^n \tag{7}$$

which is fairly standard in constraint optimization codes. A natural way to satisfy this condition is to apply the symmetric-rank-one (SRI) update

$$B_{k+1} = B_k + \varepsilon_k \frac{(w_k - B_k s_k)(w_k - B_k s_k)^T}{(w_k - B_k s_k)^T s_k} \tag{8}$$

where ε_k is a damping factor that can be selected to avoid singularity or blow-up of B_{k+1}. While the vector w_k represents the change in $g(x, \lambda)$ due to the shift in the primary variables x there is also a shift $\sigma_k = \lambda_{k+1} - \lambda_k$ in the dual variables, which leads to the adjoint secant condition

$$\sigma_k^T A_{k+1} = \mu_k^T \equiv \sigma_k^T c'(x_{k+1}) = g_{k+1} - g_{k+1/2} \tag{9}$$

In the linear case the two conditions (6) and ((9) are exactly consistent in that for $A = c'$

$$\sigma_k^T y_k = \sigma_k^T A s_k = \mu_k^T s_k .$$

Provided this scalar value is nonzero, the formula

$$A_{k+1} \equiv A_k + \delta_k \frac{(y_k - A_k s_k)(\mu_k^T - \sigma_k^T A_k)}{\mu_k^T s_k - \sigma_k^T A_k s_k} \tag{10}$$

is the unique rank-one-update satisfying both secant conditions when the damping factor δ_k equals

1. Because both direct and adjoint secant information enters into this formula we call it the two-sided-rank-one update (TR1).

Still assuming linearity of c we find that the discrepancy $D_k = A_k - A$ satisfies the recurrence

$$D_{k+1} = D_k - D_k s_k \sigma_k^T D_k \big/ \sigma_k^T D_k s_k \Rightarrow \text{rank}(D_{k+1}) = \text{rank}(D_k) - 1 \tag{11}$$

so that we must have $D_k = 0$ after $k \leq m$ undamped updates. In other words, TR1 being a generalization of SR1 shares the a priori desirable property of heredity, but it also suffers from its main drawback, namely blow up when the denominator but not the numerator vanish.

In Griewank-Walther (2002) we have developed constructive conditions on the damping parameters δ_k and ε_k that limit the change in the determinant of the full KKT matrix (5) to a specified interval about 1. These conditions and in fact the whole updating procedure for A_k and B_k are shown to be invariant with respect to linear transformations on the variable vector x and the constraint vector c. Like for Newton's method and the classical symmetric secant formulas BFGS ans DFP this key property holds even if f and c are nonlinear. Its achievement is crucially dependent on the scaling information inherent in the adjoint secant condition (9), which is based on reverse mode differentiation.

In order to examine the properties of TR1 in isolation from other algorithmic aspects we consider the case $m = n$, where $c(x) = 0$ defines a locally unique feasible point x_* assuming $\det(c'(x_*) \neq 0$. Then the KKT system becomes block triangular and effectively decomposes into the equation pair

$$c(x) = 0 \text{ and } \nabla f(x) + \lambda^T c'(x) = 0$$

The second equation depends on the first and is of course linear in the Lagrange multiplier, which measure sensitivity of the objective function value $f(x_*)$ with respect to perturbations in the constraint $c(x_*) = 0$. The Jacobian of the second equation is the transposed of the Jacobian of the first. Naturally, we keep only one copy and update it according to the TR1 formula (10). The test equation considered is the central difference discretization of the second order ODE

$$\frac{\partial^2}{\partial t^2} x(t) + \delta \frac{\partial}{\partial t} x(t) + \gamma \exp(x(t)) = 0 \text{ for } 0 < t < 1$$

As objective function we used the corresponding discretization of a linear functional

$$f_i(X) \equiv \int_0^1 \omega_i(t) x(t) dt$$

with the weighting defined by $\omega_0 = 1$, $\omega_1 = 8t(1 - t)^2$, or $\omega_2 = 2 - 7.6t(1 - t)$. Figure 3 displays the dependence of the number of iterations on the number of grid points, where the initial approximation A_0 for all secant updating methods was chosen as the negative identity divided by the mesh-size. Except in the case of Newton's method, which performs mesh-independent, there is a very strong growth of the iterations count with respect to the number of grid points n. It appears to be at least quadratic in case of the standard "good" Broyden formula (2) but only linear in case of the supposedly "bad" Broyden formula (3) and the newly proposed TR1 update (10). Even though the problem is mildly nonlinear, TR1 converges in just a little more than n steps as might be expected on the basis of its heredity property (11).

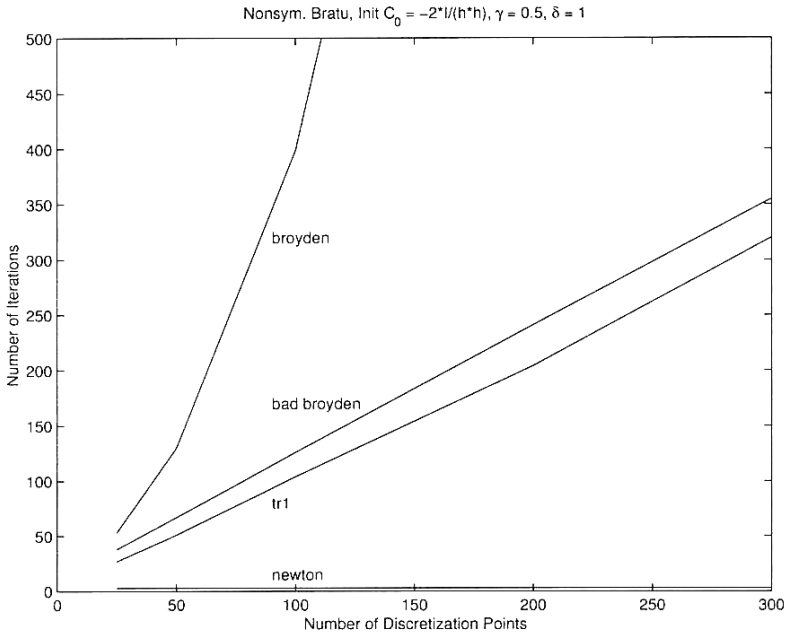

Figure 3.
Iterations for Newton and quasi-Newton from “simple” Initialization.

In contrast when A_0 is initialized as the discretization of the leading second order derivative term, all secant methods perform also mesh-independently as shown in Figure 4. This observation agrees with the theory discussed in Section 2 as the compactness condition is now satisfied. TR1 takes about half as many iterations as "bad" Broyden. That seems reasonable since, loosely speaking, TR1 takes in twice as much information as SRI per step. Also two linear systems have to be solved per step, so that the computational effort should be almost exactly the same, even in a limited memory implementation. However, in the latter case there would be a halving of the memory requirement for TRI compared to the Broyden formulas. Of course not too much should be made of a single example, and in any case, we are primarily interested in situation where the simultaneous solution of an adjoint system is required or desirable for some intrinsic reason.

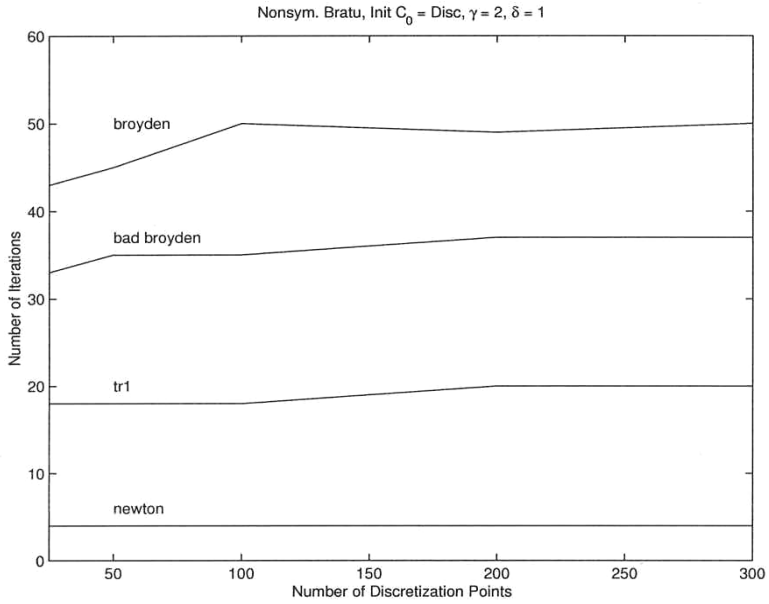

Figure 4.
Iterations for Newton and quasi-Newton from “smart” Initialization.

To check the influence of the adjoint system on the iterative solution of the direct one we have conducted the numerical experiment displayed in Figure 4. Instead of considering some norm of the residual at the sequence of iterates we have plotted the values of the second objective function defined above. For the sequence labeled with 0 we used the gradient of the same right objective in defining the adjoints sytem and thus the TRI update. For the sequence labeled with + we used *wrong* second objective for that purpose. As we had hoped the right, consistent, choice leads to considerably faster convergence of the function values to their limit. Neither the iterates generated by Newton's method nor the ones generated by "bad" Broyden depend themselves on the adjoint system so that there is only one sequence of function values for each of them.

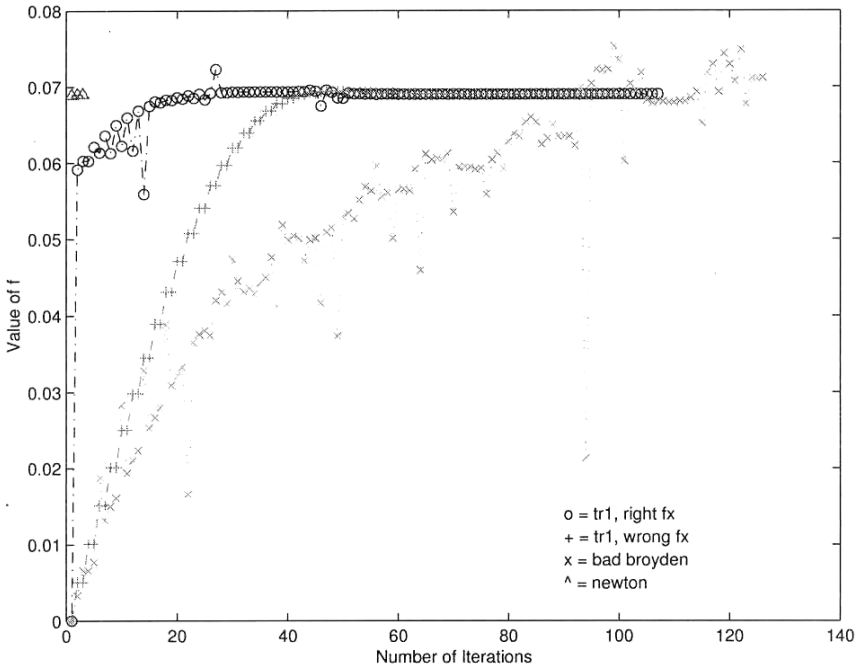

Figure 5.
Init $A_0 = 2 * l/(h * h)$, $\gamma = 0.5$, $\delta = 1$. Function Value for Newton, Bad Broyden, and TR1 with [In-] Consistent Objective (n = 100).

While Newton converges very rapidly, "bad" Broyden takes about a hundred and fifty steps to come up with a reasonable approximation to the function value. This was to be expected since we used the poor initialization for both secant methods, which did however not inhibit TR1 nearly as much. In view of the special nature of the test problem and the fact that no meaningful runtime comparisons were possible on our preliminary Matlab implementation the relative performance of TR1 and Newton cannot yet be assessed.

5. SUMMARY AND OUTLOOK

For several decades quasi-Newton methods based on low-rank secant updates have been widely applied to many small to medium sized nonlinear equations and optimization problems. Their adaption to large and structured problems has not always been successful. Also, the original motivation of avoiding any analytical derivative evaluations, because they were deemed impossible in practice, has been partly invalidated by algorithmic differentiation methods. However, exact values of complete Jacobians and Hessians do come at a varying price, and even when cheap to evaluate, they usually do not allow the kind of savings in linear algebra calculations that low rank updating achieves. This makes the latter still attractive even on linear problems.

In general we expect that secant updating methods will be competitive on large-scale problems, provided the following two conditions are met. Firstly, they must be made invariant with respect to linear transformations on the variables. Secondly, they may not be called upon to correct discrepancies in the Jacobian or Hessian that correspond to a noncompact operator for an underlying problem in function space. Encouraged by preliminary results on small equality constrained optimization problems [Griewank-Walther (2001)] we are currently in the process of developing an efficient constrained optimization code based on updating the Jacobian and Hessian with the symmetric- and two-sided rank-one formula, respectively.

Another promising application of the TR1 formula is the numerical solution of stiff ODEs, possibly in the context of optimal control. There the Jacobian, or a suitable approximation of the right-hand-side, is needed for implicit numerical integrators or the solution of the co-state equation in optimal control. By the very nature of ODE problems the numerical method should achieve invariance with respect to linear transformations of the variables and thus similarity transformations of the Jacobian. The fact that neither variant of the Broyden update can achieve this may explain why their usage in the ODE context has not been very successful. Simultaneous solution of the state and costate equation in optimal control provides naturally a pair of direct and adjoint secant conditions and thus appropriate information for the TR1 update.

6. ACKNOWLEDGMENTS

The authors are grateful to Jorge S. González for conducting the numerical experiments on Broyden's method reported in Section 2.

REFERENCES

ALLGOWER, E.L. **et al**. (1986): “A mesh independence principle for operator equations and their discretizations”, **SIAM J. Numer. Anal**, 23:160-169.

BOJANCZYK, A.W.; R.P. BRENT and F.R. de HOOG (1995): “Stability analysis of a general Toeplitz-system solver”, **Numer. Algorithms** 10, 225-244.

BROYDEN, C.G. (1965): “A class methods for solving nonlinear simultaneous equations”, **Mathematics of Computation** 19, 577-593.

BROYDEN, C.G.; J.E. DENNIS and J.J. MORE (1973): “On the local and superlinear convergence of quasi-Newton methods”, **J. Inst. Math. Appl**., 12, 223-245.

CUTHBERT, T.R., Jr. (1987): **Optimization Using Personal Computers**, John Wilay & Sons, New York.

DAVIDON, W.C. (1959): “Variable Metric Methods for Minimization”, **Tech**. **Rep. ANL-5990**, Argonne National Laboratory, Argonne, IL.

DEUFLHARD, P.; R. FREUD and A. WALTER (1990): “Fast secant methods for the iterative solution of large nonsymmetric linear systems”, **IMPACT of Computing in Science and Engineering**, 2, 244-276.

DENNIS, J.E., Jr. and R.B. SCHNABEL (1979): “Least change secant updates for quasi-Newton methods”, **SIAM Rev**., 21, 443-459.

DOUGLAS, P.H. (1934): **The Theory of Wages**, Macmillan Co., New York.

GRIEWANK, A. and A. WALTHER (2002): “Maintaining factorized KKT Systems subject to Rank-one Updates of Hessians and Jacobians”, **Technical Report Preprint IOKOMO-03-2002**, TV Dresden, Submitted to ZAMM.

_____ (2001): “On Constrained Optimization by Adjoint based quasi-Newton Methods”, To appear in **Optimization Methods and Software**. Technical Report Preprint IOKOMO-08-2001, TU Dresden.

GRIEWANK, A. (2002): “Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation”, Number 19 in **Frontiers in Appl. Math.** SIAM, Philadelphia.

_____ (1993): “Some bounds on the complexity of gradients, Jacobians, and Hessians”, in **Complexity in Nonlinear Optimization**, P.M. Pardalos, ed., World Scientific, River Edge, New York, 128-161.

_____ (1990): ”Direct calculation of Newton steps without accumulating Jacobians”, in **Large-Scale Numerical Optimization**, T.F. Coleman and Y. Li, eds., SIAM, Philadelphia, 115-137.

_____ (1987): “The Local Convergence of Broyden's Method on Lipschitzian Problems in Hilbert Spaces”, **SIAM J. Numer. Anal**., 24:684-705.

HINZE, M. and K. KUNISCH (1999): “Second order methods for optimal control of time-dependent fluid flow”, Bericht Nr. 165, Spezialforschungsbereich Optimierung und Kontrolle, Institut für Mathematik, Karl-Franzens-Vniversitat Graz, to appear **in SIAM J. Control and Optimization.**

GRIEWANK, A. and V. NAUMANN (2002): “Accumulating Jacobians by Vertex, Edge or Face Elimination”, To appear in **CARI'02**. Technical Report Preprint IOKOMO-01-2002, TV Dresden.

NOCEDAL, J. (1980): “Updating quasi-Newton matrices with limited storage”, **Mathematics of Computation**, 35, 773-782.

STOER, J. (1984): “The convergence of matrices generated by Rank-2 methods from the restricted β-class of Broyden”, **Numer. Math**., 44, 37-52.