Adjoint Broyden a la GMRES*

Andreas Griewank[†1], Sebastian Schlenkrich[2], and Andrea Walther[2]

[1]Institut für Mathematik, HU Berlin
[2]Institut für Wissenschaftliches Rechnen, TU Dresden

October 21, 2007

Abstract

It is shown here that a compact storage implementation of a quasi-Newton method based on the adjoint Broyden update reduces in the affine case exactly to the well established GMRES procedure. Generally, storage and linear algebra effort per step are small multiples of $n \cdot k$, where n is the number of variables and k the number of steps taken in the current cycle. In the affine case the storage is exactly $(n + k) \cdot k$ and in the nonlinear case the same bound can be achieved if adjoints, i.e. transposed Jacobian-vector products are available. A transposed-free variant that relies exclusively on Jacobian-vector products (or possibly their approximation by divided differences) requires roughly twice the storage and turns out to be somewhat slower in our numerical experiments reported at the end.

Keywords: nonlinear equations, quasi-Newton methods, adjoint based update, compact storage, generalized minimal residual, Arnoldi process, automatic differentiation

1 Introduction and Motivation

As shown in [SGW06, GSW06, SW06] the adjoint Broyden method described below has some very nice properties, which lead to strong theoretical convergence properties and good experimental results. A standard objection to low rank updating of approximate Jacobians is that their storage and manipulation involves per step $\mathcal{O}(n^2)$ locations and operations, respectively, since sparsity and other structure seems immediately lost. In the case of unconstrained optimization this drawback has been overcome by very successful limited memory variants [NW99] of the quasi-Newton method BFGS, which in the case of quadratic objectives and thus affine gradients reduce to conjugate gradients, the method of choice for positive definite linear systems. Since GMRES has a similar status with respect to the iterative solution of nonsymmetric systems it is a natural idea to implement a nonlinear solver that reduces automatically to GMRES on affine systems. As it turns out this is the case for a suitable implementation of the adjoint Broyden method. The insight gained from the affine scenario also helps us in dealing with singularities and other contingencies in the general case.

The paper is organized as follows. In Section 2 we describe the adjoint Broyden scheme and its main properties. In Section 3 we develop a compact storage implementation with several variants depending on the derivative vectors that are available. These are all equivalent in the affine case for which we show in Section 4 that the iterates are identical to the ones produced by GMRES, provided a linearly exact line-search is employed. Nevertheless,

*Partially supported by the DFG Research Center MATHEON "Mathematics for Key Technologies", Berlin
†Corresp. author: e-mail: griewank@mathematik.hu-berlin.de, Fax: +49-30-2093-5859

our methods are geared towards the general, nonlinear scenario, where the basic schem is guaranteed to converge [Sch07, Sec. 4.3.2], provided singularity of the actual Jacobian is exluded. Finally, in Section 6 we report comparative numerical results, mostly on nonlinear problems.

2 Description of the quasi-Newton method

We consider the iterative solution of a system of nonlinear equations

$$F(x) \;=\; 0,$$

assuming that $F : \mathbb{R}^n \to \mathbb{R}^n$ has a Lipschitz continuously differentiable Jacobian $F'(x) \in \mathbb{R}^{n\times n}$ in some neighborhood $\mathcal{N} \subset \mathbb{R}^n$ of interest. Given an initial estimate x_0 reasonably close to some root $x_* \in F^{-1}(0) \cap \mathcal{N}$ and an easily invertible approximation A_{-1} to $F'(x_*)$ we may apply our algorithms to the transformed problem

$$0 \;=\; \tilde{F}(\tilde{x}) \equiv F(x_0 + A_{-1}^{-1}\,\tilde{x})$$

Therefore we will assume without loss of generality that the original problem has been rewritten such that for some scaling factor $0 \neq \iota \in \mathbb{R}$

$$A_{-1} = I\iota \quad \text{and} \quad x_0 = 0$$

This assumption on A_{-1} greatly simplifies the notation without effecting the mathematical relations for any sensible algorithm.

Throughout the paper we use the convention that the subscript k labels all quantities related to the iterate x_k as well as all quantities concerning the step form x_{k-1} to x_k. Hence the basic iteration is

$$x_k = x_{k-1} + \alpha_k s_k \quad \text{with} \quad A_{s-1}s_k = -\delta_k F_{k-1} \quad \text{and} \quad \alpha_k \in \mathbb{R} \ni \delta_k$$

where $F_{k-1} \equiv F(x_{k-1})$. After each iteration the Jacobian approximation A_{k-1} is updated to a new version A_k in a way that distinguishes various quasi-Newton methods and is our principal concern in this section. The scalar δ_k allows for (near) singularity of the approximate Jacobian A_{k-1} and α_k represents the line-search multiplier, both of whom will be discussd below. Whenever discrepancies are computed or symbolically represented, we subtract the (more) exact quantity from the (more) approximate quantity. This is just a notational convention concerning the selection of signs in forming differences.

The Rank-one Update

Our methods are based on the following update formula.

Definition 1 (Adjoint Broyden update)
For a given matrix $A_{k-1} \in \mathbb{R}^{n\times n}$ and a current *point $x_k \in \mathbb{R}^n$ set*

$$A_k = A_{k-1} - v_k v_k^\top \left(A_{k-1} - F_k'\right) \quad \textit{with} \quad F_k' \equiv F'(x_k) \tag{1}$$

where $v_k = \sigma_k/\|\sigma_k\|$ with $\sigma_k \in \mathbb{R}^n$ chosen according to one of the three options:

'Residual': $\sigma_k = -F_k$

'Tangent': $\sigma_k = \left(A_{k-1} - F_k'\right)s_k \quad$ *for some* $s_k \in \mathbb{R}^n \setminus \{0\}$

'Secant': $\sigma_k = A_{k-1}s_k - (F_k - F_{k-1})/\alpha_k \quad$ *for some* $\alpha_k \in \mathbb{R} \setminus \{0\}$

It can be easily seen that the formula represents the least change update with respect to the Frobenius matrix norm in order to satisfy the *adjoint tangent condition*

$$A_k^\top \sigma_k \;=\; F_k'^\top \sigma_k$$

The residual choice has the nice property that after the update

$$A_k^\top F_k \;=\; F_k'^\top F_k \;\equiv\; \nabla f(x_k) \quad \text{for} \quad f(x) \equiv \|F(x)\|^2/2$$

so that the gradient of the squared residual norm is reproduced exactly. Throughout the paper $\|\cdot\|$ denotes the Euclidean norm of vectors and the corresponding induced 2-norm of matrices.

When σ_k is selected according to the tangent or secant option, the *primal tangent condition* $A_k s_k = F_k' s_k$ is satisfied approximately in that

$$\|(A_k - F_k')s_k/\|s_k\|\| \;=\; \mathcal{O}(\|x_k - x_{k-1}\|)$$

When a full quasi-Newton step $s_k = -A_{k-1}^{-1} F_{k-1}$ with $\alpha_k = 1 = \delta_k$ has been taken then the residual and the secant options are identical. The secant option reduces to the tangent option as $\alpha_k \to 0$ or when F is affine in the first place.

Throughout the paper we will allow the choice $\alpha_k = 0$, which amounts to a pure tangent update step on the Jacobian without any change in the iterate x_k itself. Several such primally stationary iterations may be interpreted as part of an inexact Newton method, which approximately solves the linearization of the given vector function at the current primal point x_k.

Heredity Properties

In the case of an affine function $F(x) \equiv Ax - b$ the tangent and secant options yield identically

$$\sigma_k \;=\; (A_{k-1} - A)s_k \;=\; D_{k-1}s_k \quad \text{with} \quad D_{k-1} \equiv A_{k-1} - A \in R^{n\times n}$$

Then it follows from (1) that the discrepancy matrices D_k satisfy the recurrence

$$D_k \;=\; D_{k-1} - \frac{D_{k-1}s_k s_k^\top D_{k-1}^\top D_{k-1}}{\|D_{k-1}s_k\|^2} \;=\; (I - v_k v_k^\top)D_{k-1}$$

Form this projective identity one sees immediately that the nullspaces of D_k and its transposed $D_k^\top$ grow monotonically with each update and must encompass the whole space $\mathbb{R}^n$ after at most n updates that are well defined in that their denominator does not vanish. In other words in the affine case the tangent and secant updates exhibit direct and adjoint hereditary in that

$$A_k s_j \;=\; A s_j \quad \text{and} \quad A_k^\top \sigma_j \;=\; A^\top \sigma_j \quad \text{for} \quad 0 \le j \le k$$

When the residual update is applied intermittently without $\sigma_k \in \text{range}(D_{k-1})$ and thus $v_k \notin \text{range}(D_{k-1})$ the direct heredity is maintained but adjoint heredity may be lost. Such updates can be viewed as a reset and are expectedt to be particularly useful in the nonlinear case.

Jacobian Initialization

It is well known for unconstrained optimization by some variant of BFGS that, starting from an initial Hessian approximation of the form $I\iota$ the performance may be strongly dependent on the choice of the scalar $\iota \neq 0$. This is so in general, even though on quadratic problems with exact line-searches the iterates are mathematically invariant with respect to $\iota \neq 0$. Hence we will also look here for a suitable initial scaling.

Another aspect of the initializationn is that in order to agree with GMRES on affine problems, we have to begin with a residual update using $\sigma_0 = -F_0$ before the very first iteration. This implies in the affine case that for all subsequent residual gradients $\nabla f(x_k) = F_k^\top F_k' = F_k^\top A_k$, which ensure for the quasi Newton-steps

$$s_{k+1} = -\delta_k A_k^{-1} F_k \quad \text{that} \quad \nabla f(x_k)^\top s_{k+1} = F_k^\top F_k' s_{k+1} = -\delta_k \|F_k\|^2$$

For $\delta_k > 0$ we have therefore descent, a property that need not hold in the nonlinear situation as we well discuss below. Starting form $A_{-1} = I\iota$ with any ι we obtain by the initial residual update

$$A_0 = I\iota - v_0 v_0^\top (I\iota - F_0') \quad \text{with} \quad \det(A_0) = \iota^{n-1} v_0^\top F_0' v_0$$

A reasonable idea for choosing ι seems to minimize the Frobenius norm of the resulting update from A_{-1} to A_0. This criterion leads to $\iota = v_0^\top F_0' v_0$, a number that may be positive or negative but unfortunately also zero. That exceptional situation arises exactly if $\det(A_0) = 0$ with the nullvector being v_0 irrespective of the choice of ι. In any case we have by Cauchy-Schwartz inequality

$$\left| v_0^\top F_0' v_0 \right| \ \leq \ \|F_0' v_0\|$$

where the right hand side does not vanish provided F_0' is nonsingular as we will assume throughout. Hence we conclude that

$$\iota \ \equiv \ \text{sign}(v_0^\top F_0' v_0) \|F_0' v_0\|$$

can be used as initial scaling. Should the first component be zero the sign can be selected arbitrarily from $\{+1, -1\}$. We could be a little bit more sophisticated here and choose the size $|\iota|$ as the Frobenius norm of the first extended Hessenberg matrix $H_0 \in \mathbb{R}^{2\times 1}$ generated by GMRES, but that complicates matters somewhat in requiring some look-ahead, especially in the nonlinear situation.

Ocurrence and Handling of Singularity

As we have seen above the contingency $\det(A_k) = 0$ may arise theoretically already when $k = 0$. In practice we are much more likely to encounter nearly singular A_k for which the full quasi-Newton directions $s_{k+1} = -A_k^{-1} F_k$ become excessively large and strongly effected by round-off. Provided we update along a null-vector whenever F_{k-1} is not in the range of $F'(x)$ we have even theoretically at most one null direction according to the following lemma.

Lemma 2 (Rank Drop at most One)
If $A_{k-1} s_k = -\delta_k F_{k-1}$ for $\delta_k \in \mathbb{R}$ with $s_k \neq 0$ and $F_k' s_k \neq 0$, then the tangent option $\sigma_k = (A_{k-1} - A) s_k$ ensures for the update (1) that

$$rank(A_k) - rank(A_{k-1}) \begin{cases} = 1 & \text{if } \delta_k = 0 \text{ and } F_k' s_k \notin range(A_{k-1}) \\ = 0 & \text{if } \delta_k = 0 \text{ and } F_k' s_k \in range(A_{k-1}) \\ \in \{0, 1\} & \text{if } \delta_k \neq 0 \text{ and } F_k' s_k \notin range(A_{k-1}) \\ \in \{-1, 0\} & \text{if } \delta_k \neq 0 \text{ and } F_k' s_k \in range(A_{k-1}) \end{cases}$$

Proof:
The tangent update always takes the explicit form

$$A_k = A_{k-1} - \big(A_{k-1} - F_k'\big) s_k s_k^\top \big(A_{k-1} - F_k'\big)^\top \big(A_{k-1} - F_k'\big) / \|(A_{k-1} - F_k') s_k\|^2$$

If $F_k' s_k \in \text{range}(A_{k-1})$ the range of A_k is contained in that of A_{k-1} so that the rank cannot go up, which implies immediately the forth case as a rank-one update can only change the rank by one up or down. If $F_k' s_k \notin \text{range}(A_{k-1})$ then multiplication of the above equation

from the right by a prospective nullvector v shows that the coefficient of $F'_k s_k$ and thus the whole rank one term must vanish. Hence v must already be a nullvector of A_{k-1} and thus the rank cannot go down, which implies in particular the third case. When $\delta_k = 0$ and thus $A_{k-1}s_k = 0$ the update simplifies to $A_k = A_{k-1} + F'_k s_k s_k^\top F_k'^\top F'_k / \|F'_k s_k\|^2$ so that s_k is a nullvector of A_{k-1} but not a null-vector of A_k. Hence we have also proven the assertion for the first case, as there can be no new null-vector as observed above. In the remaining second case all nullvectors of A_{k-1} that are orthogonal to $(F'_k s_k)^\top F'_k$ are also nullvectors of A_k and there is exactly one additional nullvector, which we may construct as follows. Let $F'_k s_k = A_{k-1} v_k$. Then there is one value $\gamma \in \mathbb{R}$ such that

$$A_k(v_k + \gamma s_k) = F'_k s_k \big(1 - \gamma - s_k^\top F_k'^\top F'_k v_k / \|F'_k s_k\|^2\big) \;=\; 0$$

□

The lemma has the following algorithmic consequences. If A_0 has at least rank $n-1$ and we select s_k as a nullvector, i.e. set $\delta_k = 0$, whenever A_{k-1} is singular, then the rank of the approximations A_k cannever drop below $n-1$. We will call this approach of setting $\delta_k = 0$ as soon as A_{k-1} is singular, the *full rank strategy*. Exactly which value $\delta_k \neq 0$ we choose when A_{k-1} is nonsingular does not make much difference in the affine case, but is of course quite important in the nonlinear case, unless we perform an exact line-search such that the scaling of s_k becomes irrelevant. We can only deviate from the full rank strategy when the approximate Jacobian A_{k-1} is singular but F_{k-1} still happens to be in its range. Then we might still choose $\delta_k \neq 0$ and determine s_k as some solution to the consistent linear system $A_{k-1}s_k = -F_{k-1}\delta_k$. This choice of s_k is even theoretically nonunique and practically subject to severe numerical instability, especially in the nonlinear scenario.

3 Smooth formulation via Adjugate

In the affine situation we will see that the singularly consistent linear systems cannever occur and that the resulting property $\mathrm{rank}(A_k) \geq n-1$ is related to the well-known fact that the Hessenberg matrix H_k in the Arnoldi process never suffers a rank drop of more than one, provided the system matrix itself is nonsingular. To define s_{k+1} uniquely as a smooth function of A_k and F_k we may set $\delta_{k+1} = \det(A_k)$ and use the adjugate $\mathrm{adj}\,(\mathrm{A_k})$ defined as the continuous solution to the identity

$$A_k\,\mathrm{adj}\,(\mathrm{A_k}) = \det(\mathrm{A_k})\mathrm{I} = \mathrm{adj}\,(\mathrm{A_k})\mathrm{A_k}$$

The entries of $\mathrm{adj}\,(\mathrm{A_k})$ may be defined as the co-factors of A_k Then we may define the steps consistently and nicely bounded via

$$s_{k+1} \equiv -\,\mathrm{adj}\,(\mathrm{A_k})\mathrm{F_k} \quad \Rightarrow \quad \mathrm{A_k s_{k+1}} = -\det(\mathrm{A_k})\mathrm{F_k}$$

If $\mathrm{rank}(A_k) = n-1$ there exist nonzero nullvectors u_k and $w_k \in \mathbb{R}^n$ such that

$$\mathrm{adj}\,(\mathrm{A_k}) \;=\; \mathrm{w_k u_k^\top} \neq 0 \quad \text{with} \quad \mathrm{A_k w_k} = 0 \quad \text{and} \quad \mathrm{u_k^\top A_k} = 0$$

Then the above formula yields the step

$$s_{k+1} = -w_k u_k^\top F_k \;\in\; \mathrm{kern}(A_k)$$

so that we have

$$s_{k+1} = 0 \quad \Leftrightarrow \quad F_k = 0 \quad \text{or} \quad 0 \neq u_k \perp F_k \neq 0$$

where the second possibility can only occur when A_k is singular. The first represents regular termination because the system is solved, whereas the second possibility indicates premature break down of the method if it is indeed defined in terms of the adjugate. It means that the linear system $A_k s_{k+1} = -F_k$ is singular but still consistent as F_k happens to lie in the

range $\{u_k\}^\perp$ of A_k. Hence nonzero solutions s_{k+1} would exist but not be unique and in the presence of round-off possibly very large. Fortunately this contingency can not occur in the affine scenario as we will see in Section 5. If it does in the nonlinear case we may define s_{k+1} as some nonzero null-vector of A_k, which is essentially unique as long as $\mathrm{rank}(A_k) = n-1$ irrespective of whether F_k is in its range or not. Alternatively we may reset A_k to A_0 as discussed above with $F_0 = F_k$, which certainly ensures that the subsequent step is well-defined.

The use of the adjugate is more of an aesthetic device in view of the affine scenario that is of particular interest in this paper. It does however alleviate the need to distinguish the cases $\mathrm{rank}(A_k) = n$ and $\mathrm{rank}(A_k) = n-1$ in proofs and other developments. The numerical computation of $s_{k+1} = -\,\mathrm{adj}\,(\mathrm{A_k})\mathrm{F_k}$ can be performed simply and stably on the basis of an LU- or QR factorization of A_k. To have a better chance of obtaining a descent direction one may multiply the step by $\mathrm{sign}(\det(A_k))$, which guarantees descent according to (2) in the affine case. More realiable for the nonlinear case would be to evaluate always the directional derivative $\nabla f(x_k)^\top s_{k+1}$ and if necessary switch the sign of s_{k+1} before entering the line-search.

Line-Search Requirements

The line-search from [Gri86] sketched below makes no assumption regarding the directional derivative and thus may produce negative step-multipliers. Moreover, if $s_k \neq 0$ is selected as arbitrary null-vector of A_k whenever $\det(A_k) \neq 0$, then that line-search ensures convergence from within level sets of f in which the actual Jacobian $F'(x)$ has no singularities. That is true even if A_0 is initialized to the null matrix, which would leave a lot of indeterminacy for the first n step selections.

The least-squares calculation at the heart of the GMRES procedure may be effected in our quasi-Newton method through an appropriate line-search. Since for affine $F(x) = Ax - b$ the function

$$\tilde{f}_k(\alpha) \;\equiv\; f(x_{k-1} + \alpha s_k) \;=\; \|F_{k-1} + \alpha A s_k\|^2/2$$

is quadratic, just three values of $\tilde{f}_k$ or two values and one directional derivative will be enough to compute the exact minimizer $\alpha_k \in \mathbb{R}$. Alternatively, we may interpolate the vector function itself by

$$\tilde{F}_k(\alpha) \;\equiv\; (1-\alpha)F_{k-1} + \alpha F(x_{k-1} + s_k)$$

on the basis of F_{k-1} and $F(x_{k-1}+s_k)$ alone. In the affine situation we have exactly $\tilde{f}_k(\alpha) = \|\tilde{F}_k(\alpha)\|^2/2$ so that the two approaches are equivalent and yield the optimal multiplier

$$\alpha_k^* \;=\; -\frac{F_{k-1}^\top A s_k}{\|A s_k\|^2} \;\equiv\; \frac{s_k^\top A_{k-1}^\top A s_k}{\|A s_k\|^2}$$

The multiplier α_k^* may be negative or even zero but it always renders the new residual $F_k = F_{k-1} + \alpha_k A s_k$ exactly orthogonal to $A s_k$. This orthogonality is crucial to proving the equivalence with GMRES and we will call any line-search yielding such an α_k^* in the affine case as **linearly exact**. Throughout we will refer to the step $x_k - x_{k-1} = \alpha_k s_k$ as

$$\textbf{trivial}:\ \alpha_k s_k = 0\,, \quad \textbf{full}:\ \alpha_k = 1/\delta_k\,, \quad \textbf{singular}:\ \det(A_{k-1}) = 0\,, \quad \textbf{exact}:\ \alpha_k = \alpha_k^*\,.$$

In the nonlinear situation we may have to perform several interpolations as described in [Gri86] before an acceptable α_k is reached. As we will see in the final section our line-search based on vector interpolation rarely requires more than one readjustment of α_k from the initial estimate $\alpha_k = 1/\delta_k$. Of course in the affine case the initial guess does not matter at all if at least one interpolation is performed so that α_k^* is reached.

Algorithmic Specification

Putting the pieces together we get the following algorithm

Algorithm 3 (Adjoint Broyden)

Initialize:	*Set* $x_0 = 0$ *and* $A_0 = I\iota - v_0\, v_0^\top (I\iota - F_0')$ *with*
	$v_0 = F_0/\|F_0\|$ *and* $\iota = sign(v_0^\top F_0' v_0)\|F_0' v_0\|$, *set* $k = 1$
Iterate:	*Compute* $s_k = \mp \mathrm{adj}\,(\mathrm{A_{k-1}})\mathrm{F_{k-1}}$ *and*
	define σ_k *by the tangent or secant option.*
Terminate:	*If* $\|\sigma_k\| \le \varepsilon$ *return* $x_k = x_{k-1} + s_k/\delta_k$ *and stop*
Update:	*Increment* $x_k = x_{k-1} + \alpha_k s_k$ *for some* $\alpha_k \in \mathbb{R}$
	set $v_k = \sigma_k/\|\sigma_k\|$, *update* $A_k = A_{k-1} - v_k\, v_k^\top (A_{k-1} - F_k')$
	and continue with **Iterate** *for* $k = k+1$

The algorithm involves at each iteration one evaluation of F_{k-1}, one of $v_k^\top F_k'$ a few trial values for F_k during the line-search. In terms of linear algebra we have to compute the step s_k by solving a system in the approximated Jacobian $A_{k-1}$and then update an appropriate representation of it to that of A_k. This means that both linear algebra subtasks require $O(n^2)$ operations and the storage requirement is n^2 or $1.5 * n^2$ floating point numbers for a QR and LU version, respectively.

4 Compact Storage Implemention

In order to reduce storage and linear algebra at least for early iterations we consider the additive expansion

$$A_k = I\iota - \sum_{j=0}^{k} v_j\, v_j^\top \left(A_{j-1} - F_j'\right) .$$

Abbreviating

$$V_k \equiv \left[v_0,\, v_1, \ldots,\, v_k\right] \in \mathbb{R}^{n\times(k+1)} \quad \text{and} \quad W_k \equiv \left[F_0'^\top v_0, \ldots, F_k'^\top v_k\right] \in \mathbb{R}^{n\times(k+1)}$$

we obtain the following representation of A_k and its inverse.

Lemma 4 (Factorized Representation)
With $L_k^{-1} \in \mathbb{R}^{(k+1)\times(k+1)}$ *the lower triangular part of* $V_k^\top V_k$ *including its diagonal we have*

$$A_k = I\iota - V_k\, L_k \left(\iota V_k - W_k\right)^\top \quad \textit{and} \quad \det(A_k) = \det(H_k)\, \iota^{n-k-1}$$

where

$$H_k \equiv W_k^\top\, V_k - \iota R_k \quad \textit{and} \quad R_k \equiv V_k^\top\, V_k - L_k^{-1} \in \mathbb{R}^{(k+1)\times(k+1)}$$

with R_k *being strictly upper triangular. Sherman-Morrison-Woodbury yields the inverse*

$$A_k^{-1} = I/\iota + V_k\, H_k^{-1} \left(V_k - W_k/\iota\right)^\top$$

if $\det(A_k) \neq 0$ *and in any case the adjugate*

$$\mathrm{adj}\,(\mathrm{A_k}) = \det(\mathrm{A_k})\mathrm{I}/\iota + \iota^{\mathrm{n-k-1}}\mathrm{V_k}\,\mathrm{adj}\,(\mathrm{H_k})(\mathrm{V_k} - \mathrm{W_k}/\iota)^\top$$

Proof: For $k = -1$ the first assertion holds trivially with all matrices other than $A_{-1} = I\iota$ vanishing completely. The induction from $k-1$ to k works as follows

$$\begin{aligned}
A_k &= A_{k-1} - v_k\, v_k^\top \big(A_{k-1} - F_k'\big) \;=\; \big(I - v_k\, v_k^\top\big) A_{k-1} + v_k\, v_k^\top F_k' \\
&= I\iota + \big(I - v_k\, v_k^\top\big)\big(A_{k-1} - I\iota\big) - v_k\, v_k^\top \big(I\iota - F_k'\big) \\
&= I\iota + \big(I - v_k\, v_k^\top\big) V_{k-1}\, L_{k-1} \big(W_{k-1} - \iota V_{k-1}\big)^\top - v_k\, v_k^\top \big(I\iota - F_k' -\big) \\
&= I\iota - \big[V_{k-1},\, v_k\big] \begin{bmatrix} I_k \\ -v_k^\top\, V_{k-1} \end{bmatrix} L_{k-1} \big(\iota V_{k-1} - W_{k-1} -\big)^\top - v_k\, v_k^\top \big(\iota I - F_k'\big) \\
&= I\iota - \big[V_{k-1},\, v_k\big] \begin{bmatrix} L_{k-1} & 0 \\ -v_k^\top\, V_{k-1}\, L_{k-1} & 1 \end{bmatrix} \begin{bmatrix} \big(\iota V_{k-1} - W_{k-1}\big)^\top \\ v_k^\top \big(I\iota - F_k'\big) \end{bmatrix} \\
&= I\iota - V_k\, L_k \big(\iota V_k - W_k\big)^\top .
\end{aligned}$$

Hence we have proven the representation of A_k provided L_k is shown to be the inverse of the upper triangular part of $V_k^\top V_k$ assuming this relation holds for L_{k-1}. That last part of the induction holds since

$$\begin{bmatrix} L_{k-1}^{-1} & 0 \\ v_k^\top\, V_{k-1} & 1 \end{bmatrix} \begin{bmatrix} L_{k-1} & 0 \\ -v_k^\top\, V_{k-1}\, L_{k-1} & 1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}$$

so that the matrix in the middle represents indeed L_k.

Assuming first $\det(H_k) \neq 0$ we obtain according to the Shermann-Morrison-Woodbury formula the inverse

$$\begin{aligned}
A_k^{-1} =& \frac{1}{\iota} \Big[I + V_k \big[I - L_k \big(V_k - W_k/\iota\big)^\top V_k\big]^{-1} L_k \big(V_k - W_k/\iota\big)^\top \Big] \\
=& \;\frac{1}{\iota} \Big[I + V_k \big(L_k^{-1}\iota - \iota V_k^\top\, V_k + W_k^\top V_k\big)^{-1} \big(\iota V_k - W_k\big)^\top \Big]
\end{aligned}$$

which can obviously be rewritten in the asserted form using the matrices R_k and H_k. The adjugate is obtained by multiplying both sides with $\det(A_k) = \det(H_k)\iota^{n-k-1}$. □

Since L_k is not needed explicitly we can implement the adjoint Broyden method storing the two $n \times (k+1)$ matrices V_k, W_k and the matrix $H_k \in \mathbb{R}^{(k+1)\times(k+1)}$ in factorized or inverted form. For small k this is certainly much less than the usual dense LU or QR implementation of A_k. However, as k approaches n it is significantly more even if we do not store $V_k^\top V_k$ which is only needed for the application of A_k itself.

Limited Memory Strategy

Since we have managed to eliminate the intermediate approximations A_j from the representation of A_k and its inverse or adjugate, it is in fact quite easy to throw out or amalgamate older directions v_j and the corresponding adjoints $v_j^\top F_j'$ from V_k and W_k, respectively. Then the corresponding rows and columns of $V_k^\top V_k$ and most importantly H_k disappear or are merged as well, which amounts to rank-two correction that is easily incorporated into the inverse or a factorization. Hence we have the capacity to always only use a window of m comparatively recent pieces of direct and adjoint secant information, a strategy that is used very successfully in limited memory BFGS. In a first test implementation we simply choose a fixed maximum m and (over)write v_k for $k > m$ into the $[(k-1) \mod m] + 1$-th columns of V_m. Obviously W_m and H_m are treated accordingly.

As we will show below, we find in the affine case that V_k is orthogonal so that $L_k = I$, $R_k = 0$ and H_k is actually upper Heisenberg, i.e., has only one nonvanishing subdiagonal. In the limited memory variant the orthogonality of V_k is maintained but the Hessenberg property of H_k is lost.

Step calculation variants

Using some temporary $(k+1)$ vector t the actual computation of the next quasi-Newton step $s_{k+1} = -\delta_k A_k^{-1} F_k$ can be broken down into the subtasks

(i) Multiply $t \equiv (\delta_k/\iota) W_k^\top F_k$

(ii) Multiply and decrement $t \mathrel{-}= V_k^\top F_k \delta_k$

(iii) Solve $H_k\, t = t$

(iv) Multiply and decrement $s_{k+1} = (\delta_k/\iota) F_k - V_k\, t$

The most promising savings are possible in the first step since we have

$$W_k^\top F_k \equiv \left[v_j^\top F_j' F_k\right]_{j=0\ldots k} \approx \left[v_j^\top F_k' F_k\right]_{j=0\ldots k} \equiv V_k^\top (F_k' F_k)$$

The approximation holds as equality exactly in the linear case, where F' is constant and thus very nearly in the smooth case. The vector on the right hand side represents in fact newer derivative information than then original one on the left. So we can get by without storing W_k at all, which pretty much halves the total storage requirement as long as $k \ll n$.

However, there is another critical issue namely how we build up the matrix H_k. Its compared to H_{k-1} new k-th column and row are given by

$$\left[v_j^\top F_j'\, v_k\right]_{j=0\ldots k} \approx V_k^\top F_k'\, v_k \in \mathbb{R}^{k+1} \quad \text{and} \quad v_k^\top F_k'\, V_k \approx \left[v_k^\top\, F_j'\, v_j\right]_{j=0\ldots k} \in \mathbb{R}^{k+1}\ .$$

For the column we may simply use the approximation based on the single, current directional derivative $F_k' v_k$. For the row we have at least three different choices. Firstly, we can compute the adjoint $v_k^\top F_k'$ but do not store it for any longer. Secondly, we store all the directional derivative $F_j' v_j$ for $j = 0 \ldots k$. Finally, we can relay on the near upper Hessenberg property of H_k and only compute the last two entries $v_k^\top\, F_{k-1}'\, v_{k-1}$ and $v_k^\top\, F_k'\, v_k$. The third option requires virtually no extra storage other than that of V_k and H_k in Hessenberg form. In that way the whole calculation would reduce almost exactly to the GMRES procedure except for the strictly upper triangular correction R_k, which is theoretically zero in the linear case.

For the solution of the nonlinear test problems in Section 6 we used the following three variants of the adjoint Broyden method:

(0) original adjoint Broyden update method storing V_k, W_k, and the QR factorization of H_k. This requires evaluation of $F(x_k)$ and $v_k^\top F'(x_k)$ at each iterate.

(1) minimal storage implementation using only V_k, the QR factorization of H_k, and approximating $W_k^\top v \approx V_k^\top (F_k' v)$. Requires evaluation of $F(x_k)$, $F'(x_k)s_k$, and $v_k^\top F'(x_k)$.

(2) forward mode based implementation using V_k, $Z_k = \left[F_j' v_j\right]_{j=0\ldots k}$, the QR factorization of H_k, and approximating $W_k^\top v \approx V_k^\top (F_k' v)$, $v_k^\top F_k' V_k \approx v_k^\top Z_k$. Requires $F(x_k)$ and $F'(x_k)v_k$

Of course, it is also possible to implement method (2) based on finite differences approximations to the directional derivatives $F'(x)v$. However, preliminary numerical tests showed that convergence of this variant is rather unreliable. For affine problems the Jacobian of F is constant and hence the variants (0) to (2) yield up to round-off identical numerical results.

5 Reduction to GMRES/FOM in affine Case

For the following result we assume that the adjoint Broyden method is applied with virtually arbitrary step-multipliers α_k. Naturally whenever $\alpha_k = 0$ we have to apply the tangent update, which could however also be approximated by a divided difference. Now we obtain the main theorem of this paper.

Theorem 5 *Suppose the algorithm 3 is applied in exact arithmetic with stopping tolerance $\varepsilon = 0$ to an affine system $F(x) = Ax - b$ with $det(A) \neq 0$. Then:*

(i) If $\iota > 0$ the iteration performs exactly the Arnoldi process irrespective of the choice of $\alpha_k \in \mathbb{R}$. If $\iota < 0$ the v_k and the corresponding entries in H_k may differ in sign. arrives at a first for which .

(ii) With $\hat{k} \leq n$ the first index such that $\sigma_{\hat{k}} = 0$ we have $x_* = x_{\hat{k}-1} + s_{\hat{k}}/\delta_{\hat{k}} = A^{-1}b$. This final step is well defined and must be taken as $F_{\hat{k}-1} \neq 0 \neq s_{\hat{k}}$ and $\delta_{\hat{k}} = \det(A_{\hat{k}-1}) \neq 0$.

(iii) For $k < \hat{k}$ all full steps $x_k = x_{k-1} + s_k/\delta_k$ with $\delta_k = \det(A_{k-1})$ (would) lead to points that coincide with the k-th iterate of the full orthogonalization method (FOM).

(iv) If (linearly) exact α_k^* are used throughout the resulting iterates x_k coincides with those generated by GMRES.

Proof: In the affine case we may always use the tangent option for σ_k so that the only impact of the step size choices α_k on the principal quantities A_k and v_k appears to be via the the residuals $F_k = Ax_k - b$. As we will see, there is in fact no such dependence, but we can certainly state already now that for any particular sequence of values α_k there must be a certain first $\hat{k}$ for which $\sigma_{\hat{k}} = 0$. The adjoint heredity property discussed in Section 2 implies that for $\hat{k} > k > j \geq 0$

$$\sigma_k^\top \sigma_j = s_k^\top D_{k-1}^\top s_j = s_k^\top 0 = 0$$

so that $V_k^\top V_k = I_k$ and consequently $L_k = I_k, R_k = 0$ in the representations of A_k and $\mathrm{adj}\,(\mathrm{A_k})$. Assuming that $F(0) = b \neq 0$ and $\det(A) \neq 0$ we find that $1 \leq \hat{k} \leq n$ since no more than n othogonal directions v_k can exist $\mathbb{R}^n$.

Now we establish the following relations by induction on $k = 1, 2, \ldots, \hat{k}$

(v) $v_{k-1} \in K_k \equiv \mathrm{span}\{b, Ab, \ldots, A^{k-1}b\} = \mathrm{span}\{v_0, v_1, \ldots, v_{k-1}\} \subset \mathbb{R}^n$

(vi) $F_{k-1} \in K_1 + AK_{k-1} = K_k$

(vii) $s_k \in K_k \ni x_k$

All three assertions hold clearly for $k = 1$ where the Krylov subspace K_1 is just the span of $F(0) = -b$ and $v_0 = \sigma_0/\|\sigma_0\|$ is selected by the residual option $\sigma = -F(0) = b$. To progress from k to $k+1$ we note that

$$\sigma_k = D_{k-1}s_k = A_{k-1}s_k - As_k = -F_{k-1}\delta_k - As_k \in K_k + AK_k = K_{k+1}$$

which proves *(v)* since v_k is colinear to σ_k and orthogonality proves that their span is the whole of K_k. Similarly we have

$$F_k = F_{k-1} + \alpha_k As_k \ \in \ K_k + AK_k = K_{k+1}$$

which proves *(vi)*. From the representation of $\mathrm{adj}\,(\mathrm{A_k})$ in Lemma 3 we see that by *(v)* up to and including v_k

$$s_{k+1} = -\,\mathrm{adj}\,(\mathrm{A_k})\mathrm{F_k} \in \mathrm{K_{k+1}} + \mathrm{range}(\mathrm{V_k}) \subset \mathrm{K_{k+1}}$$

which proves *(vii)* as the assertion for $x_{k+1} = x_k + \alpha_k s_k$ is obvious. Since the v_j are orthogonal and span succesively the Krylov subspaces K_k they must be identical (up to sign changes) to the bases generated by the Arnoldi process. As a consequence it is well know that each $Av_{i-1} \in K_{i+1}$ is a linear combination of the v_j with $j = 0 \ldots i$ so that there is an upper Hessenberg $(k+2) \times (k+1)$ matrix $\bar{H}_k$ such that

$$AV_k = V_k \bar{H}_k \quad \text{and} \quad V_k^\top A V_k = H_k$$

Here H_k is for any $k < \hat{k}$ exactly the $(k+1) \times (k+1)$ matrix occurring in Lemma 2 and can be obtained from $\bar{H}_k$ by simply leaving of the last row. It would be nice to be show that the subdiagonal elements of H_k are positive to have complete coincidence with Arnoldi but that is not an essential property. In any case it follows form our Lemma X in agreement with [Saa03] that $\det(A) \neq 0$ implies that the rectangular matrix $\bar{H}_k$ has always full column rank $(k+1)$ and H_k has therefore at least the rank k. Hence the adjugates $\mathrm{adj}\,(\mathrm{A_k})$ and $\mathrm{adj}\,(\mathrm{H_k})$ are always nontrivial. Moreover, since the elements in the subdiagonal of H_k are all nonzero we know that the a left nullvector $t_k^\top$ of H_k must have a nontrivial first component is it exists at all.

Now we can proof the remaining assertions in an explicit fashion. Firstly we obtain for the step s_{k+1} using the factorized representation of the adjugate from Lemma 4 and the identity $W_k = AV_k$ with $\delta_{k+1} = \det(A_k)$

$$\begin{aligned} s_{k+1} &= -\left[\delta_{k+1} I/\iota + \iota^{n-k-1} V_k \,\mathrm{adj}\,(\mathrm{H_k}) \mathrm{V_k^\top} (\mathrm{I} - \mathrm{A}/\iota)\right] F_k \\ &= -\iota^{n-k-1} V_k \,\mathrm{adj}\,(\mathrm{H_k}) \mathrm{V_k^\top}\, \mathrm{F_k} \end{aligned}$$

where we have used that $F_k = V_k V_k^\top F_k$ so that the AF_k term cancels out.

$$\begin{aligned} \sigma_{k+1} &= (A_k - A) s_{k+1} \\ &= -F_k \delta_{k+1} + \iota^{n-k-1} A V_k \,\mathrm{adj}\,(\mathrm{H_k}) \mathrm{V_k^\top}\, \mathrm{F_k} \\ &= -\left[\delta_{k+1} I - \iota^{n-k-1} A V_k \,\mathrm{adj}\,(\mathrm{H_k}) \mathrm{V_k^\top}\right] F_k \\ &= -\left[\delta_{k+1} I - \iota^{n-k-1} A V_k \,\mathrm{adj}\,(\mathrm{H_k}) \mathrm{V_k^\top}\right] F_0 \\ &= -\delta_{k+1} F_0 + \iota^{n-k-1} A V_k \,\mathrm{adj}\,(\mathrm{H_k}) \mathrm{e_0} \|\mathrm{F_0}\| \end{aligned}$$

where e_0 is the first Cartesian basis vector. The last simplifications come about because $F_k - F_0 \in AK_k$ belong to the null space of the matrix in square brackets and $v_0 = F_0/\|F_0\|$. Hence we see that indeed the σ_k and thus the v_k and A_k for $k < \hat{k}$ are completely indepedendent of the choice of α_k which may produce an arbitrary residual in $F_0 + AV_k$. Moreover it follows from Cramers rule that the last component in the vector $\mathrm{adj}\,(\mathrm{H_k})\mathrm{e_k}$ is exactly the product of the k subdiagonal elements of the Hessenberg matrix H_k, which are well known to be positive in the Arnoldi process. Hence this property is maintained by induction if $\iota > 0$.

Now let us consider the final situation $\sigma_{\hat{k}} = 0$. By definition the previous σ_k for $k < \hat{k}$ and thus the F_k for $k < \hat{k} - 1$ and the corresponding subdiagonals of H_k cannot vanish. Thus we must have

$$0 = \delta_{\hat{k}} F_{\hat{k}-1} - A V_{\hat{k}-1} \,\mathrm{adj}\,(\mathrm{H_{\hat{k}-1}}) \mathrm{e_0} \|\mathrm{F_0}\|$$

Since the colums of AV_k are linearly independent and $\mathrm{adj}\,(\mathrm{H_{\hat{k}-1}})\mathrm{e_0}\|\mathrm{F_0}\|$ cannot be zero neither $F_{\hat{k}-1}$ nor $\delta_{\hat{k}}$ can vanish so that the last step $s_{\hat{k}}$ is neither zero nor singular. That implies that $F_{\hat{k}} = F_{\hat{k}-1} + A s_{\hat{k}} \delta_{\hat{k}} = -\sigma_{\hat{k}} \delta_{\hat{k}} = 0$. Generally we have after each full step that F_k is a multiple of σ_k, which belongs to the orthogonal complement of K_k. That is exactly the defining property of an FOM iterate so that we have now proven (ii) and (iii).

Since α_k is obtained by a line-search minimizing $\|F_{k-1} + \alpha A s_k\|_2^2$ we must have exactly $F_k^\top A s_k = 0$. We now proof by induction on $k < \hat{k}$ the defining property of GMRES namely that $F_k^\top A s_j$ for all $0 < j \leq k$. It does hold for $k = 1 = j$ as we have just shown. Since for $k > 1$

$$F_k = F_{k-1} + \alpha_k A s_k = -\alpha_k D_{k-1} s_k + (1 - \alpha_k) F_{k-1} = \sigma_k + (1 - \alpha_k) F_{k-1}$$

the orthogonality of F_k to all As_j for $j < k$ follows from the induction hypothesis $F_{k-1}^\top A s_j$ and the fact that $\sigma_k \perp K_k \ni A s_j$.

□

To illustrate the above result in an extreme situation let us consider the case where A or more generally AA_0^{-1} is equal to the right shift matrix so that for any vector $u = (\mu_1, \mu_2, \ldots \mu_{n-1}, \mu_n)^\top \in \mathbb{R}_n$

$$A(\mu_1, \mu_2, \ldots \mu_{n-1}, \mu_n)^\top = (\mu_2, \mu_3, \ldots \mu_{n-1}, \mu_1)^\top$$

In other words A is zero except for 1's in the subdiagonal and the $(1, n)$ element. Since $AA^\top = I$ this cyclic permutation matrix is orthogonal and thus certainly normal, which according to the usual linear algebra folklore suggests that GMRES should not do too badly. In fact we find for the right hand side $b = (1, 0 \ldots 0, 0)^\top$ and $x_0 = 0$ that by GMRES also $x_k = 0$ for $k = 1 \ldots n-1$ and only the very last, namely n-th step leads to the solution $x_n = x_* = (0, 0 \ldots 0, 1)^\top$. Moreover the v_k are the Cartesian basis vectors e_k and all matrices $H_k = V_k^\top A V_k$ have the null-vectors $s_{k+1} = e_{k+1}$, which means in particular that FOM is never defined.

6 Numerical results

The adjoint Broyden methods are applied to several nonlinear equation problems. The subset of nonlinear equation problems with variable dimension of the Moré test set [MGH81] is selected. The results for these test problems should give an overview of the performance of the variants of the adjoint Broyden method. Additionally, three specific test problems are selected to investigate the convergence properties of the adjoint Broyden methods in more detail. For that purpose the problem dimensions and initial states are varied. The iteration is globalized by a derivative-free line search in the range of F. This line search was proposed in [Gri86] to prove global convergence of Broyden's method and it is adapted to the adjoint Broyden's method in [Sch07, Sec. 4.3.2].

The compact storage variants of the adjoint Broyden method are implemented in the code `abrnlq2` given as Matlab and C routine. For the considered test problems and the Matlab code derivatives are evaluated by applying AD by hand. The application of the C code uses the AD tool ADOL-C. As proposed in Section 4, we consider three variants of the algorithm. These variants are either applied to the original function or to the preconditioned function choosing $A_{-1} = F'(x_0)$ or $A_{-1} = \left[F(x_0)^\top F'(x_0) F(x_0) / F(x_0)^\top F(x_0)\right] I$.

The nonlinear equation problems with scalable dimension of the Moré test set are given in Table 1.

Table 1: Nonlinear equation problems of Moré test set

Number	Name	Reference
(21)	Extended Rosenbrock function	[Spe75]
(22)	Extended Powell singular function	[Spe75]
(26)	Trigonometric function	[Spe75]
(27)	Brown almost-linear function	[Bro69]
(28)	Discrete boundary value function	[MC79]
(29)	Discrete integral equation function	[MC79]
(30)	Broyden tridiagonal function	[Bro65]
(31)	Broyden banded function	[Bro71]

The column *Number* represents the number of the problem in [MGH81]. Additionally the performance of the adjoint Broyden updates is examined in more detail for three specific test problems:

Test function 1: The discrete integral equation function (29) in the Moré test set given by $x = (x_{(i)})_{i=1\ldots n}$, $F(x) = (f_i(x))_{i=1\ldots n}$, and

$$f_i(x) = x_{(i)} + \frac{h}{2}(1-t_i)\sum_{j=1}^{i} t_j(x_{(j)} + t_j + 1)^3 + \frac{h}{2}t_i \sum_{j=i+1}^{n} (1-t_j)(x_{(j)} + t_j + 1)^3.$$

Here $h = 1/(n+1)$ and $t_i = ih$. The function F is differentiable and its Jacobian is dense. The default initial iterate is chosen by $x_0 = (t_i(t_i - 1))_{i=1\ldots n}$.

Test function 2: The extended Rosenbrock function (21) in the Moré test set given by $x = (x_{(i)})_{i=1\ldots n}$, $F(x) = (f_i(x))_{i=1\ldots n}$, and

$$f_i(x) = \begin{cases} 10\left(x_{(i+1)} - x_{(i)}^2\right) & \text{if } i \text{ odd} \\ 1 - x_{(i-1)} & \text{if } i \text{ even} \end{cases}.$$

The function is differentiable and its Jacobian is tridiagonal. The default initial iterate is chosen by $x_0 = (-1.2, 1, -1.2, 1, \ldots)$.

Test function 3: A matrix $X \in \mathbb{R}^{d\times d}$ is sought as the matrix cube root for a given real diagonalizable matrix $Z \in \mathbb{R}^{d\times d}$, i.e.,

$$X^3 = X \cdot X \cdot X = Z. \tag{2}$$

The eigenvalue decomposition of $Z = TDT^{-1}$ yields the diagonal matrix $D = diag\{\lambda_1, \ldots, \lambda_d\}$. Denoting

$$D^{1/3} = diag\{\lambda_1^{1/3}, \ldots, \lambda_d^{1/3}\},$$

one obtains for $X = TD^{1/3}T^{-1}$ the identity

$$X^3 = TDT^{-1} = Z.$$

Thus problem (2) has a solution and can be formulated as nonlinear equation problem by

$$F(X) = X^3 - Z = 0 \in \mathbb{R}^{d\times d}$$

with dimension $n = d^2$. In the implementation the matrix X is associated row-wise with the state vector $x = (x_{(i)})$, where

$$x_{([k-1]d+l)} = X_{k,l} \quad \text{for } k, l = 1, \ldots, d.$$

Here we choose $Z = tridiag(-1, 2, -1)$. As default initial iterate the identity matrix $X_0 = I \in \mathbb{R}^{d\times d}$ is used. Note that the (i,j)-th entry of X impacts all elements in the i-th row as well as the j-th column of X^2. Consequently the same entry impacts all elements of X^3, which means that the Jacobian of this test function $F(X)$ is dense and has thus d^4 nonzero entries.

Convergence results for Moré test set functions To illustrate the performance of the adjoint Broyden update methods, the number of iterations needed to reach convergence with a reasonable tolerance are compared. Additionally, the run times required for the whole iteration process are stated. For that purpose, the C version of the program is compiled using *gcc 4.1* and executed on a PC with AMD Athlon(tm) 64 X2 Dual Core Processor 3800+ with 2 GHz and 512 KB cache size.

The results for the higher dimensional nonlinear equation problems of the Moré test set with default initial iterates are displayed in Table 2. The compact storage representation of the adjoint Broyden method is compared to the full storage representation based on updating an LU factorization of A_k. The update is evaluated by an algorithm of Bennett [Ben65]. The numbers in the first column refer to the number of the test problem in [MGH81]. If not otherwise stated, these tests are performed for the dimension $n = 1000$ using the initial iterates as proposed in the test set. The iteration is performed up to a tolerance of $tol_F = 10^{-14}$ in the residual $\|F(x_i)\|_2$ and at most 500 iterations.

Table 2: Results of Moré test set for default initial iterates

Test problem		full adj. Broy.	adjoint Broyden variant (0)	(1)	(2)
(21)	(a)	–	183	190	–
	(b)		0.64	0.59	
	(c)		177; 0	184; 0	
(21)	(a)	15	14	20	–
(P1)	(b)	0.36	0.40	0.78	
	(c)	24; 24	24; 24	26; 26	
(22)	(a)	–	44	44	–
	(b)		0.05	0.05	
	(c)		9; 4	14; 7	
(22)	(a)	28	28	28	–
(P1)	(b)	0.68	0.79	1.10	
	(c)	0; 0	0; 0	0; 0	
(26)[1]	(a)	14	13	14	116
	(b)	0.36	0.03	0.04	0.41
	(c)	3; 1	3; 0	2; 0	117; 7
(26)[1]	(a)	17	17	21	–
(P1)	(b)	0.43	0.49	0.85	
	(c)	1; 0	1; 0	4; 1	
(27)[2]	(a)	9	9	9	226
	(b)	3.1e-4	3.9e-4	4.1e-4	0.13
	(c)	1; 0	0; 0	0; 0	0; 0
(27)[2]	(a)	237	237	276	–
(P1)	(b)	9.4e-3	0.17	0.27	
	(c)	464; 464	464; 464	547; 545	

Test problem		full adj. Broy.	adjoint Broyden variant (0)	(1)	(2)
(28)	(a)	4	4	4	4
(P1)	(b)	0.10	0.13	0.16	0.08
	(c)	0; 0	0; 0	0; 0	0; 0
(29)	(a)	8	7	8	8
	(b)	5.39	5.13	8.75	5.27
	(c)	0; 0	0; 0	0; 0	0; 0
(29)	(a)	5	5	6	6
(P1)	(b)	3.25	3.98	6.86	4.18
	(c)	0; 0	0; 0	0; 0	0; 0
(30)	(a)	51	51	53	89
	(b)	1.26	0.09	0.09	0.14
	(c)	2; 0	2; 0	1; 0	1; 0
(30)	(a)	15	15	15	18
(P1)	(b)	0.37	0.44	0.61	0.35
	(c)	0; 0	0; 0	0; 0	0; 0
(31)[3]	(a)	55	42	30	70
	(b)	1.42	0.10	0.09	0.17
	(c)	18; 0	10; 0	3; 0	58; 0
(31)[3]	(a)	19	19	18	36
(P1)	(b)	0.49	0.54	0.72	0.68
	(c)	0; 0	0; 0	0; 0	1; 0

(P1) preconditioned problem with $A_{-1} = F'(x_0)$, (a) iteration counts, (b) run times in seconds, (c) additional linesearch trials; sign change in step multiplier, default problem dimension $n = 1000$

[1] Initial iterate is chosen with $x_0 = \frac{1}{2}\hat{x}_0$ with $\hat{x}_0$ as proposed in the test set. Otherwise no convergence is achieved for dimension $n = 1000$.

[2] dimension is $n = 10$, tolerance is $tol_F = 10^{-12}$

[3] tolerance is $tol_F = 10^{-12}$

As one can see nothing is gained by the compact storage implementations when the initial Jacobian $F'(x_0)$ is evaluated, factorized and then used as a preconditioner, which is mathematically equivalent to starting the adjoint Broyden method with $A_0 = F'(x_0)$. Then there is essentially no saving with regards to the linear algebra effort. However on the test problems 21 and 22 our dense implementation of full adjoint Broyden does not work at all, where as the first two compact storage versions work quite nicely. Judging by our experience so far the trouble of evaluating adjoint vectors, i.e. row-vector Jacobian products seems to pay off since the third version based exclusively on Jacobian-vector products performes significantly worse on these smooth but nonlinear problems. All three versions generate identical iterates on affine problems, of course. On test problem 3 with diagonal preconditioning the first two compact storage versions generate virtually the same iterates as the full storage vesion but the run-time is reduced by a factor of ten, which is not surprising since $n = 1000$. Similar benefits are obtained for problems 30 and 31 when the preconditioner is a multiple of the identity. Here preconditioning by the initial Jacobian reduces the number of iterations but does prolong the runtime significantly. Hence we may conclude that the compact storage implementation is indeed quite efficient, especially when the overall number of steps is only a fraction of the problem dimension.

In addition the problems of the Moré test set are solved for initial iterates further away from the solution. The approach of multiplying the initial iterate by a scalar factor to test the performance of a method is suggested in [MGH81]. Table 3 displays the required iterations and run times. The choice of the dimension n and the tolerance tol_F for these test problems is the same as before.

Table 3: Results of Moré test set for distant initial iterates

Test problem		full adj. Broy.	adjoint Broyden variant (0)	(1)	(2)
(21), (P1) $100x_0$	(a)	6	4	12	–
	(b)	0.15	0.13	0.46	
	(c)	2; 2	2; 2	4; 4	
(22) $100x_0$	(a)	–	45	65	–
	(b)		0.05	0.09	
	(c)		9; 5	27; 12	
(22), (P1) $100x_0$	(a)	34	34	34	–
	(b)	0.81	0.92	1.30	
	(c)	0; 0	0; 0	0; 0	
(26) $-10x_0$	(a)	–	34	24	130
	(b)		0.08	0.08	0.49
	(c)		18; 10	5; 2	180; 89
(26), (P1) $-10x_0$	(a)	47	47	48	185
	(b)	1.21	1.34	1.94	3.88
	(c)	1; 0	1; 0	1; 0	11; 7
(27)[4] $20x_0$	(a)	18	18	148	–
	(b)	5.9e-4	8.4e-4	0.04	
	(c)	7; 3	3; 1	164; 54	
(27)[4], (P1) $20x_0$	(a)	58	58	93	–
	(b)	2.1e-3	4.7e-3	0.01	
	(c)	63; 44	40; 18	110; 92	

Test problem		full adj. Broy.	adjoint Broyden variant (0)	(1)	(2)
(28), (P1) $100x_0$	(a)	24	24	24	–
	(b)	0.59	0.69	0.96	
	(c)	1; 0	1; 0	2; 0	
(29) $100x_0$	(a)	20	16	14	38
	(b)	18.08	10.57	14.87	25.17
	(c)	1; 0	0; 0	0; 0	9; 0
(29), (P1) $100x_0$	(a)	26	26	26	–
	(b)	16.71	17.51	28.22	
	(c)	0; 0	0; 0	0; 0	
(30)[5] $100x_0$	(a)	–	–	67	–
	(b)			0.12	
	(c)			8; 0	
(30)[5], (P1) $100x_0$	(a)	56	57	57	–
	(b)	1.37	1.62	2.28	
	(c)	20; 0	20; 0	17; 0	
(31)[5] $10x_0$	(a)	–	–	48	–
	(b)			0.16	
	(c)			7; 0	
(31)[5], (P1) $10x_0$	(a)	35	35	35	–
	(b)	0.90	1.03	1.44	
	(c)	8; 0	8; 0	10; 0	

(P1) preconditioned problem with $A_{-1} = F'(x_0)$, (a) iteration counts, (b) run times in seconds, (c) additional linesearch trials; sign change in step multiplier, default problem dimension $n = 1000$

[4] dimension is $n = 10$, tolerance is $tol_F = 10^{-12}$

[5] tolerance is $tol_F = 10^{-12}$

From remoter initial points the difference between Version 1 and 2 of the compact storage implementation becomes more marked. The latter requires only about half the storage but seems to do a better job at discarding older information as described at the end of Section 4. Hence it succeeds on problems 30 and 31 with diagonal preconditioning where the original method fails. Obviously, some kind of restart must be developed, especially in view of problem 27 where the iteration counts exceeds the dimension. In such cases one also needs a transition from the compact to the full storage scheme, which is yet to be developed.

Performance on special test problems 1-3 For the specific test problem functions, varying problem dimensions and initial iterates the Matlab version of the adjoint Broyden variants are compared to the build-in Matlab function `fsolve` for the solution of nonlinear equations. For the test functions 1 and 3 $tol_F = 10^{-14}$ and for the test function 2 $tol_F = 10^{-12}$. The maximal number of iterations allowed is again $i_{max} = 500$. Here the run times for the preconditioned problems include the run time to evaluate and factorize the initial Jacobian. Apparently Matlab uses some version of the Levenberg Marquardt method, which leads to significantly smaller iterations counts compared to the diagonally preconditioned adjoint Broyden method. However, the total runtimes are always significantly larger. Presumably, because a lot of effort goes into the differencing for Jacobian approximations.

For remote initial points the preconditioning may not pay even in terms of the iteration number and certainly with respect to the run-time. Obviously the very cheap diagonal preconditiong approach is a good idea and sometimes makes the difference between success and failure. So we have also fixed the the diagonal scaler ι at the initial point, whereas we the compact storage representation allows easily to readjust it repeatedly at virtually no extra cost.

Table 4: Results of test function 1

Test problem			adjoint Broyden variant (0)	(1)	(2)	fsolve
default initial iterate $100x_0$, varying problem dimension n						
$n = 10$		(a)	16	14	37	
		(b)	1.5e-2	1.2e-2	2.5e-2	
$n = 10$	(P1)	(a)	21	21	–	12
		(b)	2.3e-2	2.2e-2		0.21
$n = 10$	(P2)	(a)	22	21	–	
		(b)	2.2e-2	2.0e-2		
$n = 100$		(a)	16	14	37	
		(b)	5.5e-2	5.4e-2	0.14	
$n = 100$	(P1)	(a)	26	26	–	13
		(b)	0.14	0.15		3.38
$n = 100$	(P2)	(a)	22	22		
		(b)	9.1e-2	0.10	–	
$n = 1000$		(a)	16	14	38	
		(b)	1.48	1.72	3.41	
$n = 1000$	(P1)	(a)	26	26	–	15
		(b)	34.89	36.53		515
$n = 1000$	(P2)	(a)	22	22	–	
		(b)	1.83	2.78		
default problem dimension $n = 100$, varying scaling of initial iterate						
$10x_0$		(a)	8	8	12	
		(b)	3.3e-2	3.5e-2	4.3e-2	
$10x_0$	(P1)	(a)	8	8	10	8
		(b)	8.2e-2	8.6e-2	8.4e-2	1.9
$10x_0$	(P2)	(a)	9	9	10	
		(b)	4.2e-2	4.3e-2	4.2e-2	
$500x_0$		(a)	49	18	98	
		(b)	0.21	6.8e-2	0.51	
$500x_0$	(P1)	(a)	75	75	–	19
		(b)	0.40	0.46		4.6
$500x_0$	(P2)	(a)	32	32	–	
		(b)	0.12	0.14		
$1000x_0$		(a)	81	20	101	
		(b)	0.42	7.4e-2	0.54	
$1000x_0$	(P1)	(a)	121	121	–	23
		(b)	0.74	0.82		5.6
$1000x_0$	(P2)	(a)	36	36	–	
		(b)	0.15	0.17		

(P1) preconditioned problem with $A_{-1} = F'(x_0)$, (P2) preconditioned problem with $A_{-1} = \left[F(x_0)^\top F'(x_0)F(x_0)/F(x_0)^\top F(x_0)\right] I$, (a) iteration counts, (b) run times

Table 5: Results of test function 2

Test problem			adjoint Broyden variant (0)	(1)	fsolve
default initial iterate x_0, varying problem dimension n, $tol_F = 1e-12$					
$n = 10$		(a)	188	206	
		(b)	0.38	0.50	
$n = 10$	(P1)	(a)	14	19	17
		(b)	1.4e-2	1.4e-2	0.19
$n = 10$	(P2)	(a)	146	148	
		(b)	0.19	0.19	
$n = 100$		(a)	182	189	
		(b)	0.34	0.36	
$n = 100$	(P1)	(a)	14	19	23
		(b)	2.6e-2	3.0e-2	0.40
$n = 100$	(P2)	(a)	144	145	
		(b)	0.22	0.21	
$n = 1000$		(a)	183	189	
		(b)	2.0	1.7	
$n = 1000$	(P1)	(a)	14	19	18
		(b)	1.3	1.9	12.0
$n = 1000$	(P2)	(a)	144	145	
		(b)	1.5	1.1	
default problem dimension $n = 100$, varying scaling of initial iterate					
$2x_0$		(a)	28	371	
		(b)	1.8e-2	3.3	
$2x_0$	(P1)	(a)	5	11	25
		(b)	1.8e-2	2.2e-2	0.28
$2x_0$	(P2)	(a)	181	174	
		(b)	0.41	0.34	
$10x_0$		(a)	–	497	
		(b)		9.0	
$10x_0$	(P1)	(a)	4	10	10
		(b)	1.7e-2	2.1e-2	0.16
$10x_0$	(P2)	(a)	437	387	
		(b)	6.9	4.3	
$100x_0$		(a)	–	–	
		(b)			
$100x_0$	(P1)	(a)	4	11	13
		(b)	1.7e-2	2.1e-2	0.14
$100x_0$	(P2)	(a)	–	–	
		(b)			

(P1) preconditioned problem with $A_{-1} = F'(x_0)$, (P2) preconditioned problem with $A_{-1} = \left[F(x_0)^\top F'(x_0)F(x_0)/F(x_0)^\top F(x_0)\right] I$, (a) iteration counts, (b) run times in seconds, default problem dimension $n = 100$, default initial iterate $100x_0$

Table 6: Results of test function 3

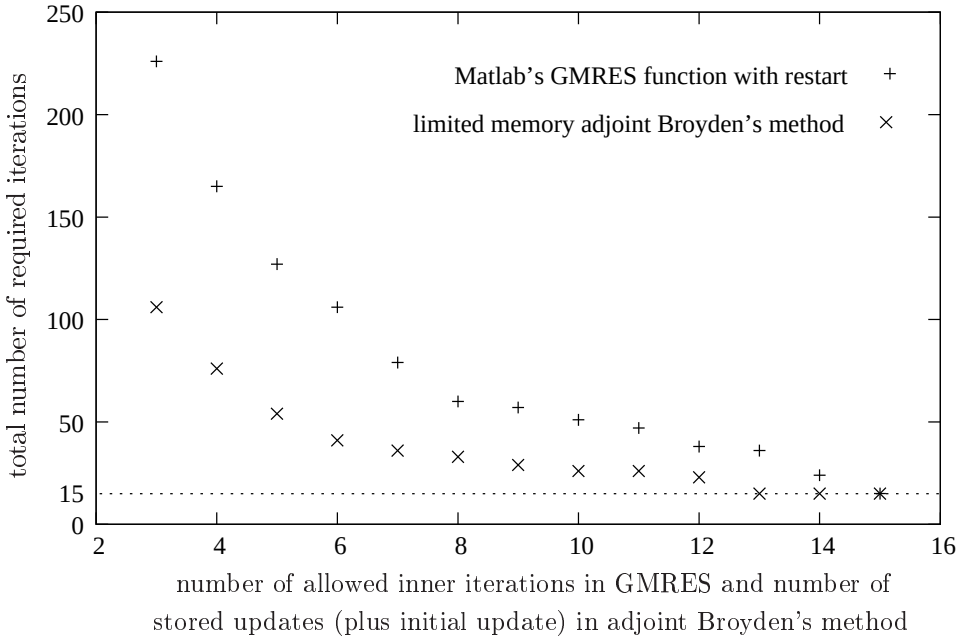
Test problem			adjoint Broyden variant (0)	(1)	(2)	fsolve
default initial iterate x_0, varying problem dimension n						
$n = 100$		(a)	23	32	–	
		(b)	3.1e-2	5.6e-2		
$n = 100$	(P1)	(a)	18	19	–	7
		(b)	8.2e-2	9.3e-2		0.24
$n = 100$	(P2)	(a)	18	19	–	
		(b)	3.1e-2	3.8e-2		
$n = 1024$		(a)	42	48	–	
		(b)	0.94	1.7		
$n = 1024$	(P1)	(a)	39	40	–	9
		(b)	15.0	15.9		52.4
$n = 1024$	(P2)	(a)	39	40	–	
		(b)	0.91	1.36		
$n = 4900$		(a)	74	79	–	
		(b)	16.2	27.1		
$n = 4900$	(P1)	(a)	65	67	–	10
		(b)	704	724		2772
$n = 4900$	(P2)	(a)	65	67	–	
		(b)	14.8	21.9		
default problem dimension $n = 100$, varying scaling of initial iterate						
$10x_0$		(a)	65	27	145	
		(b)	9.0e-2	4.4e-2	0.33	
$10x_0$	(P1)	(a)	35	34	–	15
		(b)	0.11	0.13		0.52
$10x_0$	(P2)	(a)	35	34	–	
		(b)	5.5e-2	6.7e-2		
$100x_0$		(a)	210	36	212	
		(b)	0.84	5.7e-2	0.80	
$100x_0$	(P1)	(a)	49	49	–	24
		(b)	0.14	0.17		0.82
$100x_0$	(P2)	(a)	49	50	–	
		(b)	7.3e-2	0.10		
$1000x_0$		(a)	–	46	175	
		(b)		7.2e-2	0.57	
$1000x_0$	(P1)	(a)	60	60		31
		(b)	0.16	0.20	–	1.08
$1000x_0$	(P2)	(a)	60	60	–	
		(b)	9.1e-2	0.12		

Comparison of limited memory appraches Finally, we report some preliminary results on the limited memory implementation sketched in Section 4, where the columns of V_k, W_k and H_k are periodically overwritten once k exceeds a certain limit m. We use a linear test problem so that there is no mathematical difference between our various compact storage versions and we may use the one that has almost exactly the same memory requirement as GMRES, namely it stores the $m \times n$ matrix V and the $m \times m$ matrix H.

Figure 1 compares Matlab's GMRES solver with restart to the limited memory implementation of the adjoint Broyden's method. For this test the 2D Poisson equation with Dirichlet boudary conditons on a square domain and five-point discretization is solved. Although this yields a symmetric linear problem which could be tackled by a GC method we use it here to compare the general nonsymmetric solvers. The dimension of the test problem is $n = 100$ and it is solved upto a tolerance of $tol_F = 10^{-12}$.

Without limiting the memory GMRES and adjoint Broyden are mathematically equivalent Krylov space methods and reach the required tolerance at the 15th iteration. By restricting the number m we destroy the Krylov subspace property and the convergence becomes significantly slower. As one can see from the plot in 1 GMRES(m) takes about twice as many steps as our 'periodic' adjoint Broyden version. That may be explainable by the fact that on GMRES with restart every m steps utilizes on average information in only $m/2$ directions about the problem function, whereas adjoint Broyden uses m of them throughout. Strictly speaking this means also that as far as the linear algebra is concerned the GMRES(m) iterations cost only about half as much, though there is probably a lot of common overhead, especially if m is small. Our main concern is of course the number of iterations since we assume that each function evaluation is quite expensive.

Figure 1: Comparison of limited memory adjoint Broyden and GMRES



7 Conclusion and Outlook

In this paper we have developed several compact storage implementations of the Adjoint Broyden method and shown that on affine problems they all yield identical iterates to GMRES. For that result we assumed exact line-searches, which is quite natural and realistic in the affine case. From a numerical linear algebra point of view our treatment is somewhat unsatisfactory in that we have barely given any consideration to issues of round-off propagation. In particular we have not worried about the fact that applying the compact representation of the inverse of A_k or its adjugate to the current residual amounts to orthogonalisation by unmodified Gram-Schmidt. From a more nonlinear point of view getting approximating Jacobians right with a couple of digits is already a quite satisfactory achievement so that numerical effects at level of the machine precision or even its root are of little concern. Nevertheless it should be investigated whether one my design at an implementation for general nonlinear problems that automatically reduces to the standard GMRES procedure on affine problems.

For the nonlinear scenario of greater importance are issue related to the diagonal (re)scaling of the initial Jacobian and the thorny issue if and how to reset the procedure when the storage limits are reached or older information appears to become obsolete. For that purpose one might monitor the subdiagonal entries in the projected Jacobian H_k or the entries in R. They must all vanish exactly in the affine case and should therefore be rather small near the roots of smooth functions. Their relative size might also allow a smarter selection of the directions to be discarded.

8 Aknowledgements

The first author performed his research for his paper at the IRISA Rennes, where he greatly benefited from the hospitality and the GMRES expertise of Bernard Phlippe and his colleagues.

References

[Ben65] J.M. Bennett. Triangular Factors of Modified Matrices. *Numerische Mathematik*, 7:217–221, 1965.

[Bro65] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19:577–593, 1965.

[Bro69] K. M. Brown. A quadratic convergent Newton-like method based upon Gaussian elimination. *J. Numer. Anal.*, 6:560–569, 1969.

[Bro71] C. G. Broyden. The convergence of an algorithm for solving sparse nonlinear systems. *Math. Comp.*, 25:285–294, 1971.

[Gri86] A. Griewank. The "global" convergence of Broyden-like methods with a suitable line search. *J. Austral. Math. Soc. Ser. B*, 28:75–92, 1986.

[GSW06] A. Griewank, S. Schlenkrich, and A. Walther. A quasi-Newton method with optimal R-order without independence assumption. *MATHEON Preprint 340*, 2006. Submitted to Opt. Meth. and Soft.

[MC79] J. J. Moré and M. Y. Cosnard. Numerical solution of nonlinear equations. *TOMS*, 5:64–85, 1979.

[MGH81] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *TOMS*, 7:17–41, 1981.

[NW99] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[Saa03] Y Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[Sch07] S. Schlenkrich. *Adjoint-based Quasi-Newton Methods for Nonlinear Equations*. Sierke Verlag, in press, 2007.

[SGW06] S. Schlenkrich, A. Griewank, and A. Walther. Local convergence analysis of TR1 updates for solving nonlinear euations. MATHEON Preprint 337, 2006.

[Spe75] E. Spedicato. Computational experience with quasi-Newton algorithms for minimization problems of moderately large size. *Rep. CISE-N-175, Segrate (Milano)*, 1975.

[SW06] S. Schlenkrich and A. Walther. Global convergence of quasi-Newton methods based on Adjoint Tangent Rank-1 updates. *TU Dresden Preprint MATH-WR-02-2006*, 2006. Submitted to Applied Numerical Mathematics.