An SQP Augmented Lagrangian BFGS Algorithm for Constrained Optimization

R. H. Byrd
R. A. Tapia
Yin Zhang

CRPC-TR89007
May, 1989

Center for Research on Parallel Computati
Rice University
P.O. Box 1892
Houston, TX 77251-1892

Revised November, 1990.

An SQP Augmented Lagrangian BFGS Algorithm for Constrained Optimization [*]

R. H. Byrd [†], R. A. Tapia [‡] and Yin Zhang[§]

Revised November, 1990

Abstract

In this research we present an effective algorithm for nonlinearly constrained optimization using the structured augmented Lagrangian secant update recently proposed by Tapia. The algorithm is globally defined, and uses a new and reliable method for choosing the Lagrangian augmentation parameter that does not require prior knowledge of the true Hessian. We present considerable numerical experimentation with this algorithm, both embedded in a merit-function line search SQP framework, and without line search. We compare the algorithm to the widely-used damped BFGS secant update of Powell, which, like ours, was designed to circumvent the lack of positive definiteness in the Hessian of the Lagrangian. We also establish that when our algorithm converges it converges R-superlinearly, which is a strong result in that it makes no assumptions on the approximate Hessian or the augmentation parameter. An immediate corollary is a new result in unconstrained optimization: whenever the unconstrained BFGS secant method converges, it does so Q-superlinearly. Our study has led us to the conclusion that when properly implemented Tapia's structured augmented Lagrangian BFGS secant update has strong theoretical properties, and in experiments is very competitive with Powell's damped BFGS update.

Keywords: BFGS secant method, augmented Lagrangian, SQP methods, superlinear convergence, constrained optimization.

Abbreviated Title: An SQP BFGS Algorithm.

[*] Research sponsored by SDIO/IST/ARO,AFOSR 85-0243, and DOE DEFG05-86ER 25017, and NSF grant CCR8702403, AFOSR grant 85-0251, and ARO contract DAAL03-88-K-0086

[†] Department of Computer Science, University of Colorado, Boulder, Colorado 80309

[‡] Department of Mathematical Sciences, Rice University, Houston, Texas 77251-1892

[§] Department of Mathematics and Statistics, University of Maryland Baltimore County, Catonsville, Maryland 21228.

1 Introduction

In this work, we will be concerned with the equality constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x), \\ \text{subject to} & h(x) = 0, \end{array} \tag{1.1}$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R}$, $h : \mathbf{R}^n \rightarrow \mathbf{R}^m$ $(m < n)$ and f and h are generally nonlinear. The Lagrangian function associated with problem (1.1) is the function

$$\ell(x, \lambda) = f(x) + \lambda^T h(x) \tag{1.2}$$

where $\lambda \in \mathbf{R}^m$ is called the vector of Lagrange multipliers or simply the Lagrange multiplier. We will be examining algorithms for solving this problem based on successive quadratic programming that make use of a modification of the Lagrangian in (1.2), the *augmented Lagrangian.*

As usual ∇ will denote the gradient operator, ∇^2 the Hessian operator and subscripts on these quantities signify partial differentiation. We will denote $\nabla f(x)$ by $g(x)$ and the matrix whose columns are $\nabla h_1(x), \nabla h_2(x), ..., \nabla h_m(x)$ by $A(x)$. On occasion, we employ the convention of writing g_k for $g(x_k)$ and g_* for $g(x_*)$, and similarly for other functions and other arguments. This usage should be clear from the context. We will use x_* to denote a local solution of problem (1.1) and λ_* to denote a Lagrange multiplier vector, satisfying $\nabla_x \ell(x_*, \lambda_*) = 0$.

In unconstrained optimization the BFGS secant update has emerged as the secant update of choice. The convergence analysis of BFGS secant methods requires that the Hessian matrix that is being approximated be positive definite at the solution. Furthermore, this requirement is satisfied at any nonsingular local minimizer.

It is well-known that a formal extension of the BFGS secant method can be made from unconstrained optimization to constrained optimization (problem (1.1)) by employing the so-called successive quadratic programming (SQP) framework. In anticipation of our later needs, we now state this formal extension in a line search globalization environment.

Algorithm 1.1 (Line Search SQP Lagrangian BFGS Method) *Given $x_0 \in \mathbf{R}^n$ and a symmetric $B_0 \in \mathbf{R}^{n \times n}$, for $k = 1, 2, ...,$ until convergence do*

$$\begin{aligned} x_{k+1} &= x_k + \tau_k d_k, \\ \lambda_{k+1} &= \Lambda(x_k, x_{k+1}, B_k), && (1.3) \\ s_k &= x_{k+1} - x_k, \\ y_k^\ell &= \nabla_x \ell(x_{k+1}, \lambda_{k+1}) - \nabla_x \ell(x_k, \lambda_{k+1}), && (1.4) \\ B_{k+1} &= B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k^\ell y_k^{\ell\,T}}{y_k^{\ell\,T} s_k}. && (1.5) \end{aligned}$$

where the line search direction d_k is the solution of the quadratic programming subproblem

$$\begin{array}{ll} \text{minimize} & g_k^T d + \frac{1}{2} d^T B_k d, \\ \text{subject to} & h_k + A_k^T d = 0, \end{array} \tag{1.6}$$

and the step-length τ_k is chosen to decrease a given merit (line search) function. The matrix B_k is interpreted as an approximation to $\nabla_x^2 \ell(x_k, \lambda_k)$. The function Λ in (1.3) is an updating formula for λ. A common choice for λ_{k+1} in (1.3) is the multiplier associated with the solution d_k of the subproblem (1.6). Observe that B_{k+1} satisfies the secant equation $B_{k+1}s_k = y_k^\ell$.

There is a major flaw in Algorithm 1.1. This flaw will be obvious once we invoke the following assumptions, which are standard in the theory of quasi-Newton methods for problem (1.1). They will be assumed throughout this paper.

Assumptions

A1 f and h_i have second derivatives which are Lipschitz continuous in an open, convex neighborhood $D \subset \mathbf{R}^n$ of the local solution x_*.

A2 $A(x_*)$ has full rank.

A3 $p^T \nabla_x^2 \ell(x_*, \lambda_*)p > 0$ for all $p \neq 0$ satisfying $A(x_*)^T p = 0$.

Note that A2 implies that λ_* is unique.

The deficiency of Algorithm 1.1 is that the local convergence theory for BFGS secant methods requires $\nabla_x^2 \ell(x_*, \lambda_*)$ to be positive definite and yet satisfaction of this condition is not guaranteed by the standard assumptions A1-A3.

When a line search globalization strategy is added to a BFGS secant method, it is essential that the approximate Hessian matrices B_k be positive definite. The well-known hereditary positive definiteness property of the BFGS secant update is that positive definite B_k leads to positive definite B_{k+1} if and only if $y_k^{\ell T} s_k > 0$. If $\nabla_x^2 \ell(x_*, \lambda_*)$ is not positive definite, we cannot guarantee the condition $y_k^{\ell T} s_k > 0$ even locally, i.e., for x_k and x_{k+1} near x_*, let alone globally. The desire to enforce this condition globally will play a major role in the present research.

Alternative formulations of the SQP Lagrangian BFGS secant method which circumvent the lack of positive definiteness of $\nabla_x^2 \ell(x_*, \lambda_*)$ have been challenging researchers now for many years. Perhaps the first alternative considered was replacing the Lagrangian with the augmented Lagrangian associated with problem (1.1), (see Han [12], and Tapia [23]). This latter function is

$$L(x, \lambda, \rho) = \ell(x) + \frac{\rho}{2} h(x)^T h(x), \quad (\rho \geq 0). \tag{1.7}$$

Observe that the Hessian of the augmented Lagrangian at a local solution of problem (1.1) has the form

$$H_*(\rho) \equiv \nabla_x^2 L(x_*, \lambda_*, \rho) = \nabla_x^2 \ell(x_*, \lambda_*, \rho) + \rho A(x_*) A(x_*)^T. \tag{1.8}$$

It is well-known that for any augmentation parameter ρ greater than a threshold value $\bar{\rho}$, $H_*(\rho)$ is positive definite; therefore, if y_k is defined as (we will use y_k as a generic term and different choices of y_k will be denoted by different superscripts)

$$y_k^L = \nabla_x L(x_{k+1}, \lambda_{k+1}, \rho) - \nabla_x L(x_k, \lambda_{k+1}, \rho), \tag{1.9}$$

we can guarantee that near the solution ${y_k^L}^T s_k > 0$ for ρ sufficiently large.

We arrive at the (line search) SQP augmented Lagrangian BFGS secant method for problem (1.1) by replacing y_k^ℓ in (1.4) with y_k^L from (1.9). The Broyden-Dennis-Moré theory was used by Han [12], Tapia [24] and Glad [9] to establish local and Q-superlinear convergence in the pair (x, λ) for a version of this algorithm under the standard assumptions A1–A3. Fontecilla, Steihaug and Tapia [8] showed that the convergence in x is actually Q-superlinear.

Though theoretically attractive, this alternative has serious practical problems. First, *a priori* knowledge of the threshold value $\bar{\rho}$ for a given problem is generally unavailable. Second, the attempt to use large ρ seems to present severe numerical problems; see the examples given by Tapia [24] and Nocedal and Overton [16]. See Appendix B of Tapia [25] for some interesting comments on this issue. We emphasize that y_k^L given by (1.9) has the serious disadvantage that at some iterations it may not be possible to choose ρ sufficiently large so that ${y_k^L}^T s_k$ is positive (even though it must be possible near the solution).

Another direction taken to circumvent the lack of positive definiteness of $\nabla_x^2 \ell(x_*, \lambda_*)$ is to use the BFGS secant update in the context of reduced Hessian (or projected Hessian) methods. In contrast to full Hessian methods, reduced Hessian methods approximate the Hessian restricted to the null space of the Jacobian of the constraints, where it is expected to be positive definite. Since the concern of the present work is full Hessian methods, we refer interested readers to Coleman and Conn [4], Nocedal and Overton [16], and Byrd and Nocedal [2] for further references on reduced Hessian methods. Fenyes [6] and Fontecilla [7] proposed full Hessian methods which have some of the flavor of the reduced Hessian methods.

Powell [19] proposed another modification to the (line search) SQP Lagrangian BFGS secant method which compensates for the lack of positive definiteness in the Hessian at the solution. Despite the fact that the true Hessian of the Lagrangian may not be positive definite at a solution, Powell chose to maintain a positive definite matrix by modifying y_k^ℓ whenever necessary. The modified y_k^P (say) has the form

$$y_k^P = \theta_k y_k^\ell + (1 - \theta_k) B_k s_k, \tag{1.10}$$

where the parameter θ_k is contained in $(0, 1]$. Notice from (1.5), if $\theta_k = 0$, then $B_{k+1} = B_k$; while if $\theta_k = 1$, we obtain B_{k+1} as the full BFGS update of B_k. For this reason, with Griewank [10], we refer to the use of (1.10) in (1.5) as the damped update. Powell chose θ_k so that

$${y_k^P}^T s_k \geq \eta s_k^T B_k s_k$$

is always satisfied for some $\eta \in (0, 1)$. More specifically, the number $\theta_k \in (0, 1]$ is given the value

$$\theta_k = \begin{cases} 1, & {y_k^\ell}^T s_k \geq \eta s_k^T B_k s_k \\ (1 - \eta) s_k^T B_k s_k / (s_k^T B_k s_k - {y_k^\ell}^T s_k), & \text{otherwise.} \end{cases}$$

A value for η of 0.2 was proposed in [19] and 0.1 in [21]. This technique preserves positive definiteness of B_k even far from the solution, and therefore the subproblems (1.6) are always well-posed.

Powell's damped BFGS secant method has proved to be computationally very successful (see Hock and Schittkowski [14], for example). However, there a proof of local convergence is not known for this algorithm. Given convergence, Powell [18] proves an R-superlinear rate, but only under the assumption of uniform bounds involving the approximate Hessians. Practically, although Powell's damped BFGS update works very well in general, it does sometimes encounter difficulties (see Powell [21]).

Recently, Tapia [25] suggested two new BFGS secant updates based on the structure of the augmented Lagrangian. He was able to prove that the corresponding SQP methods gave local and Q-superlinear convergence in the variable x under the standard assumptions and the assumption that the augmentation parameter ρ was greater than a threshold value $\bar{\rho}$. No guidelines or heuristics were given for choosing the augmentation parameter ρ.

It is worth mentioning that all the above techniques except for Powell's damped update have been restricted primarily to a local framework.

The objective of the current research is to first develop effective guidelines for choosing the augmentation parameter in Tapia's BFGS structured augmented Lagrangian secant algorithm (SALSA). This choice must produce globally a y_k such that $y_k^T s_k > 0$ so that the positive definiteness of approximate Hessians will be maintained. We then describe a practical implementation of SALSA, and make a theoretical and experimental investigation of its behavior.

The bulk of our numerical study of SALSA will be accomplished by using it in an SQP framework in conjunction with a line search on an ℓ_1 merit-function. Because of the demonstrated effectiveness of Powell's damped BFGS algorithm (which we will refer to as PDA) on many problems, we compare SALSA and PDA in this context. However, in order to demonstrate that differences observed are not purely consequences of the line search strategy employed we also include comparisons of the local versions of both algorithms (i.e. without line search).

Our theoretical results are an advance over what has been shown about SALSA and other augmented Lagrangian based SQP methods. We analyze the algorithm and its adaptive procedure for choosing the augmentation parameter ρ_k, without assuming that this parameter is chosen greater than some threshold value. We show, under only Assumptions A1-3 and no assumptions whatsoever on the approximate Hessians and the choice of the augmentation parameter, that if SALSA converges, then the convergence in x is R-superlinear. This is similar to, although somewhat stronger than, the result of Powell for PDA which is mentioned above. Additionally, our theorem implies, as an immediate corollary, the new result that under the standard assumptions only, i.e, no assumptions on the approximate Hessians, whenever the BFGS secant method for unconstrained optimization converges it converges Q-superlinearly.

This paper is organized as follows. In Section 2, we briefly present SALSA as Tapia proposed it. In Section 3, we discuss some critical issues concerning the globalization and implementation of SALSA and describe a complete algorithm. In particular, we develop a choice for the augmentation parameter ρ, propose a merit function and present the complete line search algorithm. Section 4 is devoted entirely to demonstrating the convergence rate result discussed above. Our numerical results comparing SALSA and Powell's damped BFGS algorithm are given in Section 5. Section 6 contains concluding remarks.

2 The Use of Structure in the Augmented Lagrangian

SALSA was designed to take advantage of structure present in the Hessian of the augmented Lagrangian function for problem (1.1). By way of motivation, observe that the Hessian of the augmented Lagrangian (1.8) displays significant structure in that there is a clear separation between the first and second order information.

Recall that the Lagrangian $\ell(x, \lambda)$ is given by (1.2) and that the augmented Lagrangian $L(x, \lambda, \rho)$ is given by (1.7). We use the superscript ℓ and L to denote quantities associated with the Lagrangian and the augmented Lagrangian, respectively. The superscript S is used in place of the superscript L when the quantity in question has been derived using the structure of the Hessian of the augmented Lagrangian.

¿From the definitions of y_k^L and y_k^ℓ (see (1.9) and (1.4)),

$$\begin{aligned} y_k^L &= y_k^\ell + \rho(A_{k+1} h_{k+1} - A_k h_k) \\ &= y_k^\ell + \rho(\textstyle\sum_{i=1}^m h_{k+1}^{(i)} \nabla_x^2 h_{k+1}^{(i)} + A_{k+1} A_{k+1}^T) s_k + O(\|s_k\|^2) \\ &= y_k^\ell + \rho A_{k+1} A_{k+1}^T s_k + O(\sigma_k^2), \end{aligned}$$

where in this case we use the superscript (i) to denote the i-th component of the vector h_k and

$$\sigma_k = \max\{(\|x_{k+1} - x_*\|, \|x_k - x_*\|\}. \tag{2.1}$$

Eliminating the second-order term of σ_k from y_k^L, we have

$$y_k^S = y_k^\ell + \rho A_{k+1} A_{k+1}^T s_k. \tag{2.2}$$

It should be noted that the use of y_k^S in place of y_k^L does not prevent the local analysis for secant methods from being carried out since the difference between y_k^L and y_k^S is $O(\sigma_k^2)$.

For the sake of completeness we present the line search SQP structured augmented Lagrangian BFGS secant algorithm – SALSA in its entirety instead of merely making appropriate changes in Algorithm 1.1.

Algorithm 2.1 (SALSA) *Given $x_0 \in \mathbf{R}^n$ and a symmetric positive definite matrix $B_0 \in \mathbf{R}^{n \times n}$, for $k = 1, 2, \ldots$, until convergence do*

$$\begin{aligned} x_{k+1} &= x_k + \tau_k d_k, \\ \lambda_{k+1} &= \Lambda(x_k, x_{k+1}, , B_k), \qquad (2.3) \\ s_k &= x_{k+1} - x_k, \\ y_k^S &= \nabla_x \ell(x_{k+1}, \lambda_{k+1}) - \nabla_x \ell(x_k, \lambda_{k+1}) + \rho A_{k+1} A_{k+1}^T s_k, \qquad (2.4) \\ B_{k+1}^L &= B_k^L - \frac{B_k^L s_k s_k^T B_k^L}{s_k^T B_k^L s_k} + \frac{y_k^S y_k^{S\,T}}{y_k^{S\,T} s_k}. \qquad (2.5) \end{aligned}$$

where the line search direction d_k is the solution of the quadratic programming subproblem (1.6) with B_k^L in place of B_k, and the step-length τ_k is chosen to decrease a given merit function. The matrix B_{k+1}^L is interpreted as an approximation to $\nabla_x^2 L(x_{k+1}, \lambda_{k+1}, \rho)$.

In SALSA, the approximate Hessian of the augmented Lagrangian B_{k+1}^L satisfies the following structured form of the augmented Lagrangian secant equation

$$B_{k+1}^L s_k = y_k^S = y_k^\ell + \rho A_{k+1} A_{k+1}^T s_k. \tag{2.6}$$

For ρ large enough, the local positivity of $y_k^{S\,T} s_k$ is guaranteed and consequently the hereditary positive definiteness of B_k^L is achieved. Even globally, $y_k^{S\,T} s_k$ can be made positive by increasing ρ, as long as $A_{k+1}^T s_k \neq 0$. To see this notice that

$$y_k^{S\,T} s_k = y_k^{\ell\,T} s_k + \rho \|A_{k+1}^T s_k\|_2^2. \tag{2.7}$$

However, as discussed in Section 3, some back-up strategy is needed to make $y_k^{S\,T} s_k > 0$ when $A_{k+1}^T s_k$ is numerically zero and $y_k^{\ell\,T} s_k \leq 0$.

It is interesting to note that while we have been viewing SALSA as an SQP augmented Lagrangian secant method, it can be equivalently viewed as an SQP structured Lagrangian secant method. To see this recall that $\nabla_x^2 L_* = \nabla_x^2 \ell_* + \rho A_* A_*^T$; thus it is quite natural to consider B_{k+1}^ℓ defined by

$$B_{k+1}^\ell = B_{k+1}^L - \rho A_{k+1} A_{k+1}^T.$$

Now from (2.6), we see that B_{k+1}^ℓ satisfies the Lagrangian secant equation

$$B_{k+1}^\ell s_k = y_k^\ell. \tag{2.8}$$

Moreover, B_{k+1}^ℓ is positive definite on the null space of A_{k+1}^T, since on this space it coincides with B_{k+1}^L. It also follows that the corresponding quadratic programming subproblem (1.6) using B_{k+1}^ℓ will have the same solution. Hence SALSA can be viewed as an SQP Lagrangian secant method with the highly desirable property that B_{k+1}^ℓ is positive definite on the null space of A_{k+1}^T.

In SALSA the structure in the Hessian of the augmented Lagrangian was utilized only in the definition of y_k^S, but not in the definition of B_k^L. Tapia [25] considered utilizing the structure in both definitions and derived what he called the augmented scale BFGS secant update. Essentially, he was able to show that this complete use of structure led to cancellations throughout the SQP method and the resulting algorithm could be viewed as an SQP Lagrangian secant method where only the part of the BFGS secant update corresponding to the scale was changed.

Initially, we experimented with the SQP augmented scale BFGS secant method and found that it does not lend itself to a line search globalization. This is due to the fact that the Hessian approximations are not necessarily positive definite. For this reason, we decided to restrict our attention to SALSA. However, the augmented scale BFGS secant update may find use in a trust-region globalization.

3 Development of SALSA

In the previous section we have discussed why we believe that the SALSA updating procedure, that is using (2.5) with (2.4), should be a good one. However, several important issues associated in the development of the algorithm SALSA remain to be addressed. In this section we first discuss a weighted form of the augmentation, and then we take up the essential issue of choosing the augmentation parameter ρ. We discuss the issues of subproblem solution, multiplier estimates, and line search, which must be addressed for any SQP algorithm, and finally we give a precise statement of the algorithm. We mention that the current version of the code is given primarily for the purpose of testing the viability of SALSA and performing numerical comparative studies. Further effort is needed to optimize each component of this algorithm.

3.1 Weighted Augmentation

The Hessian of the (unweighted) augmentation term $\rho h(x)^T h(x)$ at any feasible point, in particular at a solution x_*, is of the form $\rho A(x)A(x)^T$. If the constraints are badly scaled, then the matrix $A(x)A(x)^T$ may be ill-conditioned (here the condition number of a singular matrix is defined to be the ratio of its largest and smallest non-zero singular values) and can have negative effects on the updating process through the use of $A_k A_k^T s_k$ in y_k^S. It is natural to scale the constraints by using a weighted augmentation term $\rho h(x)^T W(x) h(x)$ which produces at x_* a well conditioned Hessian matrix $\rho A(x_*)W(x_*)A(x_*)^T$. The matrix $W(x) \in \mathbf{R}^{m \times m}$ is called a weighting matrix and should be positive definite in the area of interest. Under the assumption that $A(x)$ has full rank for all x_k, a good choice for the weighting matrix seems to be

$$W(x) = [A(x)^T A(x)]^{-1}$$

because we can write

$$A_k W_k A_k^T = A_k(A_k^T A_k)^{-1} A_k^T = Y_k Y_k^T,$$

where Y_k is any orthonormal basis for the range space of A_k. Clearly, the matrix $Y_k Y_k^T$ always has unity condition number. Moreover, as long as a weighting matrix $W(x)$ and its inverse are uniformly bounded in norm, all our theoretical results remain valid. Based on the above consideration, we therefore use the matrices $Y_k Y_k^T$ instead of $A_k A_k^T$ in our algorithm. Specifically, we define

$$y_k^S = y_k^\ell + \rho_k Y_{k+1} Y_{k+1}^T s_k. \tag{3.1}$$

In our computational experiments, this weighting technique worked somewhat better on the whole, and we did find examples for which it significantly improved the robustness of the algorithm when compared to the unweighted version.

3.2 Choice of Parameter ρ

A fundamental issue in using the augmented Lagrangian in a secant algorithm is the choice of the augmentation parameter ρ, and this is thus an issue for SALSA also. Although, as mentioned in the introduction, any value of ρ greater than the threshold value $\bar{\rho}$ will make the Hessian of the augmented Lagrangian $H_*(\rho) \equiv \nabla_x^2 L_* = \nabla_x \ell_*^2 + \rho A_* A_*^T$ positive definite, $\bar{\rho}$ depends on the unknown matrix $\nabla_x \ell_*^2$.

The practice of choosing a large ρ from the very beginning has proved to be computationally ineffective for the SQP augmented Lagrangian secant method. Not surprisingly, as was also observed by Martinez [15], we found that the same ineffectiveness also exists for the structured version, SALSA.

An alternative approach that we consider here is to choose ρ_k just large enough so that $y_k^{S\,T} s_k$ is positive. The formulation of SALSA provides a natural framework for doing this. As we can see from the definition of y_k^S in (2.4), ρ can be increased whenever needed to make $y_k^{S\,T} s_k$ *sufficiently* positive, as long as $A_{k+1}^T s_k \neq 0$. The difficult question here is what is meant by sufficiently positive. Suppose we choose ρ_k just large enough so that $y_k^{S\,T} s_k = \beta s_k^T s_k$. If β is a very small positive constant then B_{k+1} will be nearly singular (having an eigenvalue less than or equal to β) whenever $y_k^{\ell\,T} s_k \leq 0$. If β is reasonably large, then we get a poor approximation to H_* whenever the smallest eigenvalue of the reduced Hessian $Z_*^T H_* Z_*$, where Z_* is an orthonormal basis for the null space of A_*^T, is much less than β (provided that s_k has a significant component in the null space). However, if we impose the condition

$$\frac{y_k^{S\,T} s_k}{\|Y_{k+1}^T s_k\|^2} \geq \nu \tag{3.2}$$

we have a condition on $y_k^{S\,T} s_k$ that is inactive when s_k is in the null space of A_{k+1}^T and the Hessian of the Lagrangian is positive definite on that null space, and avoids near singularity of B_{k+1}^L when s_k has a significant range space component $Y_{k+1}^T s_k$. Consequently, as will be shown in Theorem 3.1, imposing the condition (3.2) solves the problem near the solution.

However, when this positive definiteness fails, as it may far from the solution, we argue that this bound should be larger. This is because the term $y_k^S y_k^{S\,T} / y_k^{S\,T} s_k$ in the BFGS updating formula (2.5) can get excessively large when $y_k^{S\,T} s_k$ is small relative to $\|y_k^S\|^2$. To demonstrate this phenomenon, let us suppose that we are in a situation where $y_k^{\ell\,T} s_k < 0$ and $\|Y_{k+1}^T s_k\| \ll \|y_k^\ell\|$. If ρ_k is chosen such that $y_k^{S\,T} s_k$ is comparable to $\|Y_{k+1}^T s_k\|^2$, then $y_k^{S\,T} s_k \ll \|y_k^\ell\|$. On the other hand, the magnitude of $\|y_k^S\|$ (i.e., $\|y_k^\ell + \rho_k Y_{k+1} Y_{k+1}^T s_k\|$) can be as large as as the dominant term $\|y_k^\ell\|$. Consequently, the rank-one matrix $y_k^S y_k^{S\,T} / y_k^{S\,T} s_k$ can be excessively large, since its unique nonzero eigenvalue is $\|y_k^S\|^2 / y_k^{S\,T} s_k$. As a result, the newly updated matrix B_{k+1}^L could be badly ill-conditioned. To see this, observe that a lower bound for the spectrum condition number of B_{k+1}^L is

$$\frac{y_k^{S\,T} B_{k+1}^L y_k^S / y_k^{S\,T} y_k^S}{s_k^T B_{k+1}^L s_k / s_k^T s_k} \geq \left(\frac{\|y_k^S\| \, \|s_k\|}{y_k^{S\,T} s_k} \right)^2 .$$

In deriving the above estimate we used the facts $B_{k+1}^L s_k = y_k^S$ and

$$y_k^{S\,T} B_{k+1}^L y_k^S \geq (y_k^{S\,T} y_k^S)^2 / y_k^{S\,T} s_k.$$

Now it should be clear that the condition number of B_{k+1}^L will be large when $y_k^{S\,T}s_k$ is small relative to $\|y_k^S\|\,\|s_k\|$. In experiments we have observed algorithm failures due to this behavior. However these failures were avoided by requiring in addition that $y_k^{S\,T}s_k \geq |y_k^\ell s_k|$.

Therefore, combining this condition with (3.2) yields the following strategy for choosing ρ_k at each iteration. Whenever $\|Y_{k+1}^T s_k\|$ is *sufficiently positive*, we choose $\rho_k > 0$ such that

$$y_k^{S\,T}s_k = y_k^{\ell\,T}s_k + \rho_k s_k^T Y_{k+1}Y_{k+1}^T s_k \geq \max\{|y_k^{\ell\,T}s_k|, \nu\|Y_{k+1}^T s_k\|^2\} \tag{3.3}$$

where ν is a positive constant. The condition of $\|Y_{k+1}^T s_k\|$ being sufficiently positive will be discussed in the next subsection. In that case we need a back-up strategy, which will also be discussed in the next subsection.

It is straightforward to show that (3.3) is equivalent to requiring

$$y_k^{S\,T}s_k \geq \max\{\frac{\rho_k}{2}, \nu\}\|Y_{k+1}^T s_k\|^2. \tag{3.4}$$

In our implementation, we set $\nu = 0.01$. We choose ρ_k to be the smallest nonnegative value satisfying (3.3), which implies that we choose $\rho_k = 0$ if

$$y_k^{\ell\,T}s_k \geq \nu\|Y_{k+1}^T s_k\|^2. \tag{3.5}$$

It can be seen easily that when s_k is in the null space of A_{k+1}^T and x_k is near x_*, which implies $y_k^{\ell\,T}s_k > 0$ under Assumption A3, condition (3.5) will hold. On the other hand, when the step has a significant range space component $Y_{k+1}Y_{k+1}^T s_k$, near-singularity of B_{k+1}^L is avoided because $s_k^T B_{k+1}^L s_k \geq \nu\|Y_{k+1}^T s_k\|^2$. Moreover, the condition $y_k^{S\,T}s_k \geq |y_k^{\ell\,T}s_k|$ is designed to prevent the deterioration of B_{k+1}^L due to relatively small $y_k^{S\,T}s_k$. Our computational experiments have shown that this heuristic condition works quite well. In addition, as a result of enforcing (3.3) y_k^S has the following nice property.

Theorem 3.1 *Under Assumptions A1-A3, if ρ_k is chosen to satisfy (3.3), then there is a constant M_1 such that*

$$y_k^{S\,T}s_k \geq M_1\|s_k\|^2 \tag{3.6}$$

for all x_k and x_{k+1} sufficiently close to x_* and λ_{k+1} sufficiently close to λ_*.

Proof: Let $\hat\rho$ be some value such that $H_*(\hat\rho) = \nabla_x^2\ell(x_*,\lambda_*) + \hat\rho A(x_*)W(x_*)A(x_*)^T$ is positive definite, and let μ_1 be the smallest eigenvalue of $H_*(\hat\rho)$.

Case 1: $\|Y_{k+1}^T s_k\|^2 \leq \mu_1\|s_k\|^2/(3\hat\rho)$. Then for some constant $C > 0$,

$$\begin{aligned}
y_k^{S\,T}s_k &\geq y_k^{\ell\,T}s_k \\
&\geq s_k^T H_*(\hat\rho)s_k - C\sigma_k\|s_k\|^2 - \hat\rho s_k^T Y_{k+1}Y_{k+1}^T s_k \\
&\geq (\mu_1 - C\sigma_k)\|s_k\|^2 - \hat\rho\|Y_{k+1}^T s_k\|^2 \\
&\geq \frac{\mu_1}{3}\|s_k\|^2
\end{aligned}$$

when $\sigma_k = \max(\|x_k - x_*\|, \|x_{k+1} - x_*\|, \|\lambda_{k+1} - \lambda_*\|) \le \mu_1/(3C)$.

Case 2: $\|Y_{k+1}^T s_k\|^2 > \mu_1 \|s_k\|^2/(3\hat{\rho})$. Then by (3.3)

$$y_k^{S\,T} s_k \ge \nu \|Y_{k+1}^T s_k\|^2 \ge \frac{\nu\mu_1}{3\hat{\rho}} \|s_k\|^2. \tag{3.7}$$

In either case our result holds with $M_1 = \min[\frac{\mu_1}{3}, \frac{\nu\mu_1}{3\hat{\rho}}]$. □

Note that the only property of the matrix Y_{k+1} used in the proof was the existence of $\hat{\rho}$ such that $H_*(\hat{\rho})$ is positive definite. This means that Theorem 3.1 also holds for any choice of y_k such that $y_k - y_k^S = O(\sigma_k \|s_k\|)$ or one using A_{k+1} in place of Y_{k+1}.

A nice feature of this result is that it shows that we can pick ρ_k so that y_k^S acts as though $H_*(\rho_k)$ were positive definite (it satisfies (3.6)) even though we don't know whether we have chosen ρ_k large enough to make $H_*(\rho_k)$ positive definite.

3.3 A Back-up Strategy

Theorem 3.1 seems to indicate that we have a good strategy for choosing the augmentation parameter and maintaining positive definiteness of B_k in a neighborhood of the solution. In fact it actually allows us to maintain positive definiteness whenever $Y_{k+1}^T s_k$ is nonzero.

However, in the above strategy for making $y_k^{S\,T} s_k$ positive, there is one case that the structured augmented Lagrangian approach is incapable of handling; that is when

$$Y_{k+1}^T s_k = 0 \text{ and } y_k^{\ell\,T} s_k \le 0. \tag{3.8}$$

This is analogous to the case $y_k^T s_k \le 0$ in unconstrained optimization. We have shown that this will not happen when the current iterate is already close to a solution, but globally this may happen. In addition, if $y_k^{\ell\,T} s_k \le 0$, and $\|Y_{k+1}^T s_k\|$ is not zero but very small, the choice of ρ given by (3.3) would be excessively large. Therefore in these cases we need a back-up strategy for preserving positive definiteness, and we need a rule for deciding between the back-up strategy and the SALSA update.

A possible option for such a back-up strategy is to just not update, i.e., set $B_{k+1}^L = B_k^L$ whenever the case (3.8) occurs. However, in experiments with this strategy we have observed that once an update is skipped, the algorithm often continues not to update for a number of iterations without much progress, requiring a large number of iterations to solve the problem. The problem with the not-to-update strategy seems to be its sacrifice of a self-correcting mechanism. This sacrifice may cause problems in the following way. Suppose the not-to-update strategy is invoked when the step s_k is very small due to very large elements in the matrix B_k^L as well as small $\|h(x_k)\|$. Because s_k is small, $x_{k+1}(= x_k + s_k)$ will be close to x_k. Since $y_k^{\ell\,T} s_k < 0$, we would like $s_k^T B_{k+1}^L s_k$ to be small. Instead, the update is skipped and B_{k+1}^L continues to be large. As a result, s_{k+1} is again very small and has a direction close to that of s_k (because $B_{k+1}^L = B_k^L$ and $x_{k+1} \approx x_k$). At the $(k+1)$-st iteration the update will be skipped again and this process can be repeated for many steps.

Having been convinced that skipping updates is not a good strategy, we adopt the following back-up strategy. Noting from (3.1) that y_k^ℓ is augmented by a constant times the projection of s_k

on the range space of A_{k+1}, it seems natural to use s_k itself whenever its projection on the range space of A_{k+1} is too small. Therefore, whenever (3.5) is violated and

$$\|Y_{k+1}^T s_k\| < \min\{\beta_1, \|s_k\|\}\|s_k\|, \tag{3.9}$$

we replace $Y_{k+1}Y_{k+1}^T s_k$ in (3.1) by s_k. Here $\beta_1 < 1$ is a small positive number. We choose the value $\beta_1 = 0.01$, which seems to work well experimentally. When using this back-up strategy we choose ρ_k such that

$$y_k^{S\,T} s_k = y_k^{\ell\,T} s_k + \rho_k s_k^T s_k \geq \max(|y_k^{\ell\,T} s_k|, \nu\|Y_{k+1}^T s_k\|^2) \tag{3.10}$$

is satisfied, which is analogous to condition (3.3).

Condition (3.9) is designed to ensure that the back-up strategy is eventually turned off as x_k approaches x_*. This is due to the fact that $\|Y_{k+1}^T s_k\|$ is of order $O(\|s_k\|^2)$. This is the subject of the following result.

Theorem 3.2 *Assume A1-A3. If condition (3.9) holds, x_k and x_{k+1} are sufficiently close to x_* and λ_{k+1} is sufficiently close to λ_*, then condition (3.5) is satisfied and therefore the back-up strategy is not selected.*

Proof: Suppose condition (3.9) holds and let $Z_{k+1} \in \mathbf{R}^{(n-m)\times n}$ be such that its columns form an orthonormal basis for the null space of A_{k+1}^T. It follows from $\|Y_{k+1}^T s_k\| < \beta_1\|s_k\|$ and $s_k = Z_{k+1}Z_{k+1}^T s_k + Y_{k+1}Y_{k+1}^T s_k$ that

$$\|Y_{k+1}^T s_k\|^2 < \frac{\beta_1^2}{1-\beta_1^2}\|Z_{k+1}^T s_k\|^2.$$

Substituting the above into $\|Y_{k+1}^T s_k\| < \|s_k\|^2$, we obtain

$$\|Y_{k+1}^T s_k\| < \|Z_{k+1}^T s_k\|^2 + \|Y_{k+1}^T s_k\|^2 < \frac{\|Z_{k+1}^T s_k\|^2}{1-\beta_1^2}. \tag{3.11}$$

Let

$$\bar{G}_k = \int_0^1 \nabla_x^2 \ell(x_k + \tau s_k, \lambda_{k+1}) d\tau.$$

Then, we have

$$\begin{aligned} y_k^{\ell\,T} s_k &= s_k^T \bar{G}_k s_k = s_k^T Z_{k+1}(Z_{k+1}^T \bar{G}_k Z_{k+1}) Z_{k+1}^T s_k \\ &+ s_k^T Y_{k+1}(Y_{k+1}^T \bar{G}_k Y_{k+1}) Y_{k+1}^T s_k + 2 s_k^T Y_{k+1}(Y_{k+1}^T \bar{G}_k Z_{k+1}) Z_{k+1}^T s_k \\ &= s_k^T Z_{k+1}(Z_{k+1}^T \bar{G}_k Z_{k+1}) Z_{k+1}^T s_k + O(\|Z_{k+1}^T s_k\|^3) \end{aligned}$$

by (3.11). By Assumption A3, for x_k and x_{k+1} sufficiently close to x_* and λ_{k+1} to λ_*,

$$s_k^T Z_{k+1}(Z_{k+1}^T \bar{G}_k Z_{k+1}) Z_{k+1}^T s_k \geq \mu\|Z_{k+1}^T s_k\|^2$$

for some constant $\mu > 0$. Therefore, by (3.11)

$$y_k^{\ell\,T} s_k \geq \frac{\mu}{2}\|Z_{k+1}^T s_k\|^2 \geq \frac{\mu}{2}(1-\beta_1^2)\|Y_{k+1}^T s_k\| \geq \nu\|Y_{k+1}^T s_k\|^2,$$

for $\|Y_{k+1}^T s_k\|$ sufficiently small, and condition (3.5) is satisfied. □

3.4 Subproblem Sdution and Multiplier Estimates

Our procedure for computing the solution of (1.6) is as follows. A QR decomposition of A_k is first performed, namely

$$A_k = \begin{pmatrix} Y_k & Z_k \end{pmatrix} \begin{pmatrix} R_k \\ 0 \end{pmatrix} = Y_k R_k, \qquad (3.12)$$

where $Y_k \in \mathbf{R}^{n\times m}$ is an orthonormal basis for the range space of A_k, $Z_k \in \mathbf{R}^{n\times(n-m)}$ is an orthonormal basis for the null space of A_k^T, and R_k is an m by m upper triangular matrix. The solution d_k of the subproblem (1.6) is given by

$$d_k = Y_k Y_k^T d_k + Z_k Z_k^T d_k, \qquad (3.13)$$

where

$$Y_k^T d_k = -R_k^{-T} h_k, \text{ and } Z_k^T d_k = -(Z_k^T B_k^L Z_k)^{-1} Z_k^T (g_k + B_k^L Y_k Y_k^T d_k). \qquad (3.14)$$

The multiplier associated with the QP subproblem (1.6) is

$$\lambda_{k+1}^{QP} = -(A_k^T A_k)^{-1} A_k^T (g_k + B_k^L d_k), \qquad (3.15)$$

We use this multipler estimate in defining y_k^ℓ. Another possible choice for the multiplier estimate is the least-squares estimate

$$\lambda_{k+1}^{LS} = -(A_{k+1}^T A_{k+1})^{-1} A_{k+1}^T g_{k+1}. \qquad (3.16)$$

However, in our experiments we found that use of this value resulted in significantly more failures than the use of (3.15). Therefore, we will use the QP multiplier estimate to form y_k^ℓ in our numerical tests, that is

$$y_k^\ell = \nabla_x \ell(x_{k+1}, \lambda_{k+1}^{QP}) - \nabla_x \ell(x_k, \lambda_{k+1}^{QP}). \qquad (3.17)$$

3.5 Line Search

In order to test the viability of SALSA in a line search globalization framework, we need to specify a merit function for the algorithm. Our purpose here is not to determine the best merit function, but to use a simple robust function to provide some context for testing our updating strategy. We choose a merit function of the form

$$\phi(x, w) = f(x) + \sum_{i=1}^{m} w^{(i)} |h^{(i)}(x)|. \qquad (3.18)$$

This type of merit function was first used in an SQP algorithm by Han [13] and was later also used by Powell [19].

Let $\phi_k(\tau) = \phi(x_k + \tau d_k, w_k)$, $\tau \geq 0$ and let $\phi_k'(0)$ be the directional derivative of $\phi(x, w_k)$ with respect to x in the direction d_k — the solution of subproblem (1.6). Then

$$\phi_k'(0) = g_k^T d_k - \sum_{i=1}^{m} w^{(i)} |h_k^{(i)}|,$$

which follows from the fact that d_k satisfies the constraints of subproblem (1.6)

$$\nabla h_k^{(i)\,T} d_k = -h_k^{(i)}, \ \ i = 1, 2, \ldots, m.$$

It has been shown by Han [13] that a sufficient condition for $\phi_k'(0) < 0$ is

$$w_k^{(i)} > |(\lambda_{k+1}^{QP})^{(i)}| \tag{3.19}$$

for all i, where λ_{k+1}^{QP} is the Lagrange multiplier associated with the k-th QP subproblem. Han [13] proves a global convergence result assuming (3.19) is eventually satisfied for a constant w. This holds under his conditions if the weights are chosen to be monotone increasing. However, it has been observed that the performance of this merit function is rather sensitive to the choice of the weights w. Too large a w can also slow down convergence. Powell [19] first used in his code VFO2AD a strategy that allowed w to fluctuate, more specifically,

$$w_k^{(i)} = \max\{|(\lambda_{k+1}^{QP})^{(i)}|, 0.5(|(\lambda_{k+1}^{QP})^{(i)}| + w_{k-1}^{(i)})\}.$$

Though this strategy has been shown [14] to be computationally successful, it does not meet Han's condition for global convergence. Moreover, Chamberlain [3] constructed an example that shows that Powell's strategy of choosing w can lead to cycling instead of convergence.

We performed numerical tests using monotonically increasing weights and found that this strategy resulted in a large number of failures with both Powell's method and SALSA. This was particularly true when we used non-standard starting points which were far from the solutions or ill-conditioned initial Hessian approximations. It seemed to occur fairly often that an early estimate of the Lagrange multiplier would be much larger than the true multiplier. Then the corresponding large weight, kept large by the monotonicity requirement, would cause the line search to take very short steps, sometimes leading to failure.

In order to have a more meaningful comparison in a realistic environment, we used the following simple nonmonotone strategy. We define at the k-th iteration

$$w_k^{(i)} = \mu_k(|(\lambda_{k+1}^{QP})^{(i)}| + \delta), \ \ i = 1, 2, ..., m,$$

where $\mu_k \geq 1$ and $\delta > 0$ (here we choose $\delta = 0.0001$) and λ_{k+1}^{QP} is the Lagrange multiplier estimate obtained by solving the quadratic programming subproblem. Although a value of $\mu_k = 1$ does give a descent direction (see (3.19)), we found that we were able to take full steps more often if μ_k was chosen large enough so that

$$\phi_k'(0) \leq -|g_k^T d_k|.$$

Consequently, the formula we used for choosing μ_k was

$$\mu_k = \max\{1, 2g_k^T d_k / (\sum_{i=1}^{m} |(|(\lambda_{k+1}^{QP})^{(i)}| + \delta) h_k^{(i)}|\}.$$

This is somewhat similar to a condition proposed by Powell [20] in the context of a monotone strategy. Of course, we can make no global convergence guarantees for this nonmonotone strategy, and it is certainly possible that instances of cycling like those discussed by Chamberlain [3] could occur. However, based on our experiments, the likelihood of cycling seems to be extremely low (it was never observed) for equality constrained problems. In addition, it should be noted that most proofs of convergence involving quasi-Newton methods and merit functions (except that given in [2] for the reduced Hessian case) assume the boundedness of $\|B_k\|$ or $\|B_k^{-1}\|$, a property whuch, even locally, does not follow from our analysis of this method. Thus even if we used monotone increasing weights, we would have only a very weak guarantee of global convergence.

A back-tracking strategy is used in our line search to determine a step-length τ_k satisfying the sufficient decrease condition

$$\phi_k(\tau_k) \leq \phi_k(0) + \alpha\tau_k\phi_k'(0) \tag{3.20}$$

where $0 < \alpha < \frac{1}{2}$. Here we choose $\alpha = 0.1$. We always start from $\tau_k^{(0)} = 1$. If $\tau_k^{(j)}$ satisfies (3.20), we let $\tau_k = \tau_k^{(j)}$; otherwise,

$$\tau_k^{(j+1)} = \max\left\{0.1, \min\left\{0.9, \frac{0.5\,\phi_k'(0)\tau_k^{(j)}}{\phi_k(0) + \phi_k'(0)\tau_k^{(j)} - \phi_k(\tau_k^{(j)})}\right\}\right\}\tau_k^{(j)}.$$

The formula on the right-hand side comes from a restricted quadratic interpolation. We limit the number of back-trackings to 10; if $j > 10$, we abort the line search and terminate the algorithm. The above back-tracking procedure is basically the one used by Powell [19].

It is well-known that the non-smoothness of the merit function $\phi(x, w)$ may prevent a step-length of one from being taken near the solution even though it is a good choice. This phenomenon is commonly called the Maratos effect. It is certainly an issue that should be adequately addressed in a production code, but it does not happen very often and we therefore took no specific measures to combat it. The Maratos effect does not appear to have been a major factor in our numerical experiments; in only a very small number of cases was a step-length of less than one taken within the last three iterations of a run and it never happened within the last two iterations.

3.6 Algorithm Description

Now we are ready to describe the complete form of SALSA. We suppose that all the quantities involved in the algorithms have already been evaluated before they are used.

Algorithm 3.1 (SALSA)

Step 0 *Choose positive constants tol* > 0, ν, β_1, *a positive integer mxiter,* $x_0 \in \mathbf{R}^n$ *and a symmetric positive definite matrix* $B_0^L \in \mathbf{R}^{n\times n}$. *Set* $k = 0$.

Step 1 *If the stopping criterion* $\|(Z_k^T g_k, h_k)\|_2 \leq tol$ *is satisfied, exit.*

Step 2 *If B_k^L is numerically indefinite, stop; otherwise solve the subproblem (1.6) for the search direction d_k and the QP Lagrange multiplier estimate λ_{k+1}^{QP}, using (3.13), (3.14) and (3.15).*

Step 3 *Perform the line search to determine the step-length τ_k. If the number of back-tracking iterations exceeds 10, stop; otherwise, set $x_{k+1} = x_k + \tau_k d_k$ and $s_k = x_{k+1} - x_k$.*

Step 4 *Calculate y_k^ℓ given by (3.17). If $y_k^{\ell T} s_k \geq \nu \|Y_{k+1}^T s_k\|^2$, set $y_k^S = y_k^\ell$. Otherwise, set $y_k^S = y_k^\ell + \rho_k v_k$, where*

$$\rho_k = (\max\{|y_k^{\ell T} s_k|, \nu \|Y_{k+1}^T s_k\|^2\} - y_k^{\ell T} s_k)/\|v_k\|^2,$$

$$v_k = \begin{cases} Y_{k+1} Y_{k+1}^T s_k, & \textit{if } \|Y_{k+1}^T s_k\| \geq \min\{\beta_1, \|s_k\|\}\|s_k\|, \\ s_k, & \textit{otherwise.} \end{cases}$$

Use the updating formula (2.5) to obtain B_{k+1}^L.

Step 5 *If $k > mxiter$, then stop (too many iterations); otherwise, increment k by one and go to* **Step 1.**

Since Powell's damped BFGS method is one of the most efficient methods currently available, for the purpose of comparison we also implemented Powell's damped BFGS method and ran it side by side along with SALSA. Our implementation of Powell's damped BFGS algorithm is the following and for simplicity we will refer it as the PD algorithm or simply PDA.

Algorithm 3.2 (PDA) *All steps are identical to Algorithm 3.1 (SALSA) except for* **Step 4** *where Powell's damped BFGS update is used.*

Evidently, discrepancies in the numerical performance of Algorithms SALSA and PDA should be largely due to the use of the two different updating schemes: the structured augmented Lagrangian BFGS update or Powell's damped BFGS update.

4 Convergence Rate of SALSA

Now we consider the convergence rate of the algorithm developed in the previous section. In this paper we will analyze only the local behavior of SALSA. Therefore we will assume that the sequence generated by SALSA converges to a local minimizer satisfying Assumptions A1-3, and that a step-length of one is eventually taken at each iteration. A proof of convergence based on a line search on the merit function as in [1] would require more knowledge of merit functions than currently exists. As already mentioned, augmented Lagrangian quasi-Newton algorithms have been analyzed by Han [12] and by Tapia [24], [25] under the assumption that ρ_k is chosen larger than the threshold value and is eventually constant. Their analysis is similar to the Broyden, Dennis and Moré theory for unconstrained optimization and establishes that, if x_0 and B_0 are sufficiently good initial approximations, then the sequence $\{(x_k, \lambda_k)\}$ converges to (x_*, λ_*) Q-superlinearly. Actually, Tapia [25] established that $x_k \rightarrow x_*$ Q-superlinearly. Because of our weaker and more implementable assumptions on the choice of ρ, we cannot prove local convergence when B_0 is a good enough approximation, but we can prove that if the iterates converge to the solution they converge R-superlinearly.

We would like our analysis to apply to a wider class of implementations of SALSA than the detailed Algorithm 3.1. To achieve this we will base our analysis on the following generalized version of SALSA, which differs from Algorithm 3.1 in that step-lengths of one are always taken, stopping conditions are removed, and a wider class of augmentation terms and multiplier estimates is allowed.

Algorithm 4.1 (Generalized Local Version of SALSA)

Step 0 *Initialize $x_0 \in \mathbf{R}^n$ and a symmetric positive definite matrix $B_0^L \in \mathbf{R}^{n \times n}$. Set $k = 0$.*

Step 1 *Solve the subproblem (1.6) for the search direction d_k using (3.13) and (3.14).*

Step 2 *Set $x_{k+1} = x_k + d_k$.*

Step 3 *Choose the matrix $\hat{A}_{k+1} = A_{k+1} W_{k+1}^{\frac{1}{2}} + O(\|s_k\|)$, where W_{k+1} is taken from a bounded set of positive definite matrices whose inverses are also bounded.*

Step 4 *Calculate $y_k^\ell = \nabla_x \ell(x_{k+1}, \lambda_{k+1}) - \nabla_x \ell(x_k, \lambda_{k+1})$. If $y_k^{\ell\,T} s_k \geq \nu \|\hat{A}_{k+1}^T s_k\|^2$, set $y_k^S = y_k^\ell$. Otherwise, set $y_k^S = y_k^\ell + \rho_k v_k$, where*

$$\rho_k = (\max\{|y_k^{\ell\,T} s_k|, \nu \|\hat{A}_{k+1}^T s_k\|^2\} - y_k^{\ell\,T} s_k)/\|v_k\|^2,$$

$$v_k = \begin{cases} \hat{A}_{k+1}\hat{A}_{k+1}^T s_k, & \text{if } \|\hat{A}_{k+1}^T s_k\| \geq \min\{\beta_1, \|s_k\|\}\|s_k\|, \\ s_k, & \text{otherwise} \end{cases}$$

Use the updating formula (2.5) to obtain B_{k+1}^L.

Step 5 *Increment k by one and go to* **Step 1.**

Note that we do not specify the Lagrange multiplier estimate λ_k in y_{k+1}^ℓ in Step 4 of the algorithm; however in the theory we will require that $\lambda_k \rightarrow \lambda_*$. (For some choices of multiplier estimate such as λ_k^{LS} convergence of the multipliers is a consequence of convergence of $\{x_k\}$, but this is not immediate for λ_k^{QP}.) The form of $\hat{A}_k$ allows many possible choices for y_k^S depending on the choice of W_{k+1} (see Section 3.1). It is easy to see that the following choices of y_k^S are of the specified form, $y_k^\ell + \rho_k A_{k+1} W_{k+1} A_{k+1}^T s_k + O(\|s_k\|^2)$, and thus covered by our analysis.

$$\begin{aligned} y_k^S &= y_k^\ell + \rho_k Y_{k+1} Y_{k+1}^T s_k, \\ y_k^S &= y_k^\ell + \rho_k A_{k+1} A_{k+1}^T s_k, \\ y_k^S &= y_k^\ell + \rho_k Y_k Y_k^T s_k, \\ y_k^S &= y_k^\ell + \rho_k A_k (h_{k+1} - h_k), \end{aligned}$$

In the analysis to follow,-we will use y_k in place of y_k^S and B_k in place of B_k^L for simplicity.

The main purpose of this section will be to prove the following result.

Theorem 4.1 *Assume that the sequence $\{x_k\}$ is generated by Algorithm 4.1, and Assumptions A1-3 hold. If $x_k \rightarrow x_*$ and $\lambda_k \rightarrow \lambda_*$, then $x_k \rightarrow x_*$ R-superlinearly.*

In order to prove this convergence theorem, however, we first define some useful quantities and prove the intermediate results, Lemmas 4.1-4.4. After proving the theorem we will then point out an interesting application to unconstrained optimization. Note that Theorem 4.1 is similar to the rate of convergence result proved by Powell for his damped algorithm, except that this result makes no boundedness assumptions on the approximate Hessians. Our analysis uses some of the techniques developed by Powell in his proof.

By Assumptions A1-3 we know that there is a value $\hat{\rho} \geq 0$ such that the matrix $\nabla_x^2 L(x_*, \lambda_*, \hat{\rho})$ is positive definite. Given the uniform boundedness of $\{W_{k+1}^{-1}\}$, this value may also be chosen so that

$$s_k^T(y_k^\ell + \hat{\rho}\hat{A}_{k+1}\hat{A}_{k+1}^T s_k) > 0 \tag{4.1}$$

for x_k, x_{k+1}, and λ_k sufficiently close to their solution values. For purposes of analysis we select one such $\hat{\rho}$ and we define the matrix $H_* = \nabla_x^2 L(x_*, \lambda_*, \hat{\rho})$, which will be used as a weighting matrix. We define two quantities which measure the accuracy of B_k along the step direction s_k: the ratio of quadratic forms,

$$q_k = \frac{s_k^T B_k s_k}{s_k^T H_* s_k}, \tag{4.2}$$

and

$$\cos\theta_k = \frac{s_k^T B_k s_k}{\|H_*^{\frac{1}{2}} s_k\| \|H_*^{-\frac{1}{2}} B_k s_k\|}, \tag{4.3}$$

the cosine of the angle between $B_k s_k$ and $H_* s_k$, measured in the $H_*^{-\frac{1}{2}}$ weighted norm. These two quantities, which ideally have value one, thus measure how close the magnitude and direction of $B_k s_k$ correspond to the magnitude and direction of $H_* s_k$. We now show that these two quantities provide rough bounds on the ratio of $\|s_k\|$ to the error, and on the ratio of successive errors. We will also use the notation $e_k = x_k - x_*$ in what follows.

Lemma 4.1 *Given Assumptions A1-3, there exist constants γ_1 and γ_2 such that if x_k is sufficiently close to x_*, and s_k solves (1.6) then*

$$\gamma_1(1 + \frac{q_k}{\cos\theta_k})^{-1} \leq \frac{\|s_k\|}{\|e_k\|} \leq \gamma_2(\frac{1}{\cos\theta_k} + \frac{1}{q_k}) \tag{4.4}$$

and

$$\frac{\|e_{k+1}\|}{\|e_k\|} \leq 1 + \gamma_2\left(\frac{1}{\cos\theta_k} + \frac{1}{q_k}\right). \tag{4.5}$$

Proof: By the way the step is computed, $\|B_k s_k\| \geq \|Z_k^T B_k s_k\| = \|Z_k^T g_k\|$. Therefore,

$$\begin{aligned} \|s_k\| &\geq \frac{\|s_k\|}{\|B_k s_k\|}\|Z_k^T g_k\| \\ &= \frac{s_k^T B_k s_k}{\|B_k s_k\|\|s_k\|}\frac{s_k^T s_k}{s_k^T B_k s_k}\|Z_k^T g_k\| \\ &\geq \gamma_1' \frac{s_k^T B_k s_k}{\|H_*^{-\frac{1}{2}} B_k s_k\|\|H_*^{\frac{1}{2}} s_k\|}\frac{s_k^T H_* s_k}{s_k^T B_k s_k}\|Z_k^T g_k\| \end{aligned}$$

for some constant γ_1', since H_* is positive definite. Thus

$$\|s_k\| \geq \gamma_1' \frac{\cos\theta_k}{q_k}\|Z_k^T g_k\|.$$

Looking at the normal component of the step we see that

$$\|s_k\| \geq \|A_k(A_k^T A_k)^{-1} A_k^T s_k\| = \| - A_k(A_k^T A_k)^{-1} h_k\| \geq \hat{\gamma}_1 \|h_k\|,$$

for some constant $\hat{\gamma}_1$. Then, in the neighborhood of a minimizer satisfying Assumptions A1-3 we have

$$\begin{aligned} \|x_k - x_*\| &\leq \gamma(\|Z_k^T g_k\| + \|h_k\|), \\ &\leq \frac{\gamma q_k}{\gamma_1' \cos\theta_k}\|s_k\| + \frac{\gamma}{\hat{\gamma}_1}\|s_k\|, \end{aligned}$$

and the left inequality of (4.4) follows immediately.

To establish the other side note that

$$\begin{aligned} s_k^T s_k &= \frac{s_k^T s_k}{s_k^T B_k s_k} s_k^T (Z_k Z_k^T B_k s_k + Y_k Y_k^T B_k s_k) \\ &\leq \frac{s_k^T s_k}{s_k^T B_k s_k}\left(\|s_k\|\|Z_k^T g_k\| + \|Y_k^T s_k\|\|B_k s_k\|\right). \end{aligned}$$

Therefore

$$\|s_k\| \leq \frac{s_k^T s_k}{s_k^T B_k s_k}\|Z_k^T g_k\| + \hat{\gamma}_2 \frac{\|B_k s_k\|\|s_k\|}{s_k^T B_k s_k}\|h_k\|,$$

and since H_* is positive definite,

$$\|s_k\| \leq \gamma_2' \left(\frac{s_k^T H_* s_k}{s_k^T B_k s_k} + \frac{\|H_*^{-\frac{1}{2}} B_k s_k\|\|H_*^{\frac{1}{2}} s_k\|}{s_k^T B_k s_k}\right)\left(\|Z_k^T g_k\| + \|h_k\|\right),$$

from which the right inequality follows immediately. Inequality (4.5) follows from (4.4) upon noting that, by the triangle inequality,

$$\frac{\|e_{k+1}\|}{\|e_k\|} \leq 1 + \frac{\|s_k\|}{\|e_k\|}.$$

□

Actually the previous lemma could have been proved with any positive definite matrix replacing H_* in the definition of q_k and $\cos\theta_k$. However, in the next lemma the use of H_* is essential to establishing the more precise result that if q_k and $\cos\theta_k$ are sufficiently close to 1 then the ratio of successive errors can be made arbitrarily small.

Lemma 4.2 *Under the conditions of Lemma 4.1,*

$$\begin{aligned} \|e_{k+1}\| &= O(\|e_k\|^2 + \|Z_k^T(B_k - H_*)s_k\|) && (4.6) \\ &= O\left(\|e_k\|^2 + \left(\frac{q_k^2}{\cos^2\theta_k} - 2q_k + 1\right)^{\frac{1}{2}} \|s_k\|\right). && (4.7) \end{aligned}$$

Proof: First we decompose the error into two parts and consider each separately. Observe that

$$\begin{aligned} \|Z_k^T H_* e_{k+1}\| &= \|Z_k^T H_*(e_k + s_k)\| \\ &= \|Z_k^T [\nabla L(x_k, \lambda_*, \hat{\rho}) - (\nabla L(x_*, \lambda_*, \hat{\rho}) + B_k s_k] \\ &\quad + Z_k^T (H_* - B_k)s_k)\| + O(\|e_k\|^2) \\ &= \|Z_k^T [\nabla L(x_k, \lambda_*, \hat{\rho}) - g_k + (H_* - B_k)s_k)]\,\| \\ &\quad + O(\|e_k\|^2) \\ &\leq \|Z_k^T (H_* - B_k)s_k)\| + O(\|e_k\|^2). \end{aligned}$$

The range space component of the error is given by

$$\begin{aligned} A_k^T e_{k+1} &= A_k^T e_k + A_k^T s_k && (4.8) \\ &= h_k + O(\|e_k\|^2) - h_k = O(\|e_k\|^2). && (4.9) \end{aligned}$$

The total error is related to these two parts by

$$\|e_{k+1}\| = \left\| \begin{bmatrix} Z_k^T H_* \\ A_k^T \end{bmatrix}^{-1} \begin{bmatrix} Z_k^T H_* e_{k+1} \\ A_k^T e_{k+1} \end{bmatrix} \right\|,$$

and by Assumptions A1-3 the matrix

$$\begin{bmatrix} Z_k^T H_* \\ A_k^T \end{bmatrix}^{-1} = \left[Z_k (Z_k^T H_* Z_k)^{-1} \;\; H_*^{-1} A_k (A_k^T H_*^{-1} A_k)^{-1} \right]$$

is bounded for all x_k in some neighborhood of x_*. Therefore

$$\|e_{k+1}\| = O\left(\left\| \begin{matrix} Z_k^T H_* e_{k+1} \\ A_k^T e_{k+1} \end{matrix} \right\| \right) = O(\|e_k\|^2 + \|Z_k^T (B_k - H_*)s_k\|)$$

which is just (4.6).

To establish (4.7), note that

$$\frac{\|H_*^{-\frac{1}{2}}(B_k - H_*)s_k\|^2}{\|H_*^{\frac{1}{2}} s_k\|^2} = \frac{q_k^2}{\cos^2\theta_k} - 2q_k + 1,$$

which, since H_* is nonsingular, implies that the right hand side of (4.6) is of the same order as the right hand side of (4.7).

□

Having established the effect of the quantities q_k and $\cos\theta_k$ on the length of the computed step and the error at the next point, we now consider the issue of how these two key quantities are related to the BFGS update. To that end we define, for any positive definite matrix B, the quantity

$$\psi(B) = \mathrm{trace}(H_*^{-\frac{1}{2}} B H_*^{-\frac{1}{2}}) - \log\det(H_*^{-\frac{1}{2}} B H_*^{-\frac{1}{2}}),$$

which may be considered as a measure of the deviation of B from H_*. Note that ψ is a strictly convex function over the set of positive definite matrices, and it has a unique minimizer at $B = H_*$ as is discussed by Byrd and Nocedal [1].

We now show that $\{\rho_k\}$ is bounded and that the update has an important self-correcting property with respect to ψ. Close to the solution if q_k or θ_k deviates significantly from 1, and if s_k is close to the null space, then $\psi(B)$ is decreased (i.e. B_{k+1} is closer to H_*). The self-correction relation (4.10) established below is analogous to the one of Lemma 7 in [18] except that it uses the Ψ function in a manner similar to equation (2.9) of [1] instead of a weighted Frobenius norm.

Lemma 4.3 *If A1-3 are satisfied, then for all x_k and x_{k+1} sufficiently close to x_* and all λ_{k+1} sufficiently close to λ_*, there exists an upper bound for the augmentation parameter ρ_k chosen by Algorithm 4.1. In addition, for any bounded choice of ρ_k, if B_k is positive definite the updated matrix produced by the algorithm satisfies*

$$\psi(B_{k+1}) \le \psi(B_k) - \frac{q_k}{\cos^2\theta_k} + \log q_k + 1 + \gamma_4\sigma_k + \gamma_3\left(\frac{\|h_k\|}{\|s_k\|}\right)^2, \tag{4.10}$$

where $\sigma_k = \max\{\|e_k\|, \|e_{k+1}\|, \|\lambda_{k+1} - \lambda_*\|\}$, and γ_3 and γ_4 are constants.

Proof: By Theorem 3.2, sufficiently close to the solution, the backup strategy is not used, and the value of ρ_k chosen in **Step** 4 of Algorithm 4.1 is the smallest value satisfying (3.3), or equivalently (3.4). Since the value $\hat{\rho}$ is such that sufficiently close to x_* (4.1) holds, then it follows that a value of ρ_k as large as $2\hat{\rho} + \nu$ will satisfy (3.4).

By the definition of ψ, and since $\det(B_{k+1}) = (y_k^T s_k / s_k^T B_k s_k)\det(B_k)$

$$\begin{aligned}\psi(B_{k+1}) &= \psi(B_k) + \mathrm{trace}(H_*^{-\frac{1}{2}}(-\frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k})H_*^{-\frac{1}{2}}) - \log(\frac{y_k^T s_k}{s_k^T B_k s_k}) && (4.11)\\ &= \psi(B_k) - \frac{\|H_*^{-\frac{1}{2}} B_k s_k\|^2}{s_k^T B_k s_k} + \frac{y_k^T H_*^{-1} y_k}{y_k^T s_k} - \log\frac{y_k^T s_k}{s_k^T H_* s_k} + \log\frac{s_k^T B_k s_k}{s_k^T H_* s_k}. && (4.12)\end{aligned}$$

By Steps 2 and 3 of the algorithm,

$$\begin{aligned} y_k &= H_* s_k - \hat{\rho} A_{k+1} A_{k+1}^T s_k + \rho_k \hat{A}_{k+1} \hat{A}_{k+1}^T s_k + O(\sigma_k\|s_k\|)\\ &= H_* s_k + A_{k+1}(\rho_k W_{k+1} - \hat{\rho} I) A_{k+1}^T s_k + O(\sigma_k\|s_k\|).\end{aligned}$$

This means that

$$\begin{aligned} y_k^T H_*^{-1} y_k &= y_k^T s_k y_k^T H_*^{-1} A_{k+1}(\rho_k W_{k+1} - \hat{\rho} I) A_{k+1}^T s_k + +O(\sigma_k \|s_k\|^2) \\ &= y_k^T s_k + s_k^T A_{k+1}(\rho_k W_{k+1} - \hat{\rho} I) A_{k+1}^T s_k \\ &\quad + s_k^T A_{k+1}(\rho_k W_{k+1} - \hat{\rho} I) A_{k+1}^T H_*^{-1} A_{k+1}(\rho_k W_{k+1} - \hat{\rho} I) A_{k+1}^T s_k + O(\sigma_k \|s_k\|^2), \end{aligned}$$

so that using Theorem 3.1, the fact that $A_{k+1}^T s_k = h_k + O(\sigma_k \|s_k\|)$, and the uniform bound on W_{k+1},

$$\frac{y_k^T H_*^{-1} y_k}{y_k^T s_k} = 1 + O\left(\frac{\|h_k\|}{\|s_k\|}\right)^2 + O(\sigma_k). \tag{4.13}$$

In addition,

$$y_k^T s_k = s_k^T H_* s_k + s_k^T A_{k+1}(\rho_k W_{k+1} - \hat{\rho} I) A_{k+1}^T s_k + O(\sigma_k \|s_k\|^2)$$

so that

$$-\log \frac{y_k^T s_k}{s_k^T H_* s_k} = -\log[1 - O\left(\frac{\|h_k\|}{\|s_k\|}\right)^2 + O(\sigma_k)] \tag{4.14}$$

$$= O\left(\frac{\|h_k\|}{\|s_k\|}\right)^2 + O(\sigma_k) \tag{4.15}$$

for σ_k and $\|h_k\|/\|s_k\|$ sufficiently small. Since, by Theorem 3.1, $y_k^T s_k / s_k^T H_* s_k$ is bounded away from zero (4.15) also holds if either σ_k or $\|h_k\|/\|s_k\|$ are not small. Substituting (4.13) and (4.15) into (4.12) and using (4.2) and (4.3) we get

$$\psi_{k+1} \leq \psi_k - \frac{q_k}{\cos^2 \theta_k} + \log q_k + 1 + O(\sigma_k) + O\left(\frac{\|h_k\|}{\|s_k\|}\right)^2$$

□

To analyze the iterates produced by the algorithm we would like bounds on the ratios $\frac{\|s_j\|}{\|e_j\|}$ and $\frac{\|e_{j+1}\|}{\|e_j\|}$. Such bounds would hold at each iterate if we had bounds on the quantities $\|B_k\|$ and $\|Z_k^T B_k^{-1} Z_k\|$ as is shown in [18], but we have not assumed and cannot establish such bounds on B_k. However, the self-correcting property of Lemma 4.3 based on the departure of q_k and $\cos\theta_k$ from 1, can be used together with the bounds in Lemma 4.1 to bound the average behavior of any large subset of the iterates.

Lemma 4.4 *Assume that the sequence $\{x_k\}$ is generated by Algorithm 4.1 and that Assumptions A1-3 hold. If $x_k \to x_*$, and $\lambda_k \to \lambda_*$, then there is a constant β such that for any $k > 0$ and any subset S of $\{1, ..., k\}$,*

$$[\prod_{j \in S} \frac{\|e_{j+1}\|}{\|e_j\|}] < \beta^k. \tag{4.16}$$

In addition, for any $p \in (0,1)$ there are constants β_1 and β_2 such that for any $k > 0$ the set

$$\mathcal{J}_k = \{j \in [1,k] : \beta_1 \leq \frac{\|s_j\|}{\|e_j\|} \leq \beta_2\} \tag{4.17}$$

contains at least pk elements.

Proof: Summing up the recursion (4.10) established in the previous lemma we have that

$$0 \le \psi_{k+1} \le \psi_0 + \sum_{j=0}^{k}[-\frac{q_j}{\cos^2\theta_j} + \log q_j + 1 + \gamma_4\sigma_j + \gamma_3\left(\frac{\|h_j\|}{\|s_j\|}\right)^2]$$

Since $\|A(x)\|$ is uniformly bounded near x_*, the quantity $\|h_k\|/\|s_k\|$ is bounded above for all k so that

$$0 \le \sum_{j=0}^{k}[-\frac{q_j}{\cos^2\theta_j} + \log q_j + 1] + k\gamma',$$

for some constant γ'. Alternatively,

$$\sum_{j=0}^{k}[\frac{q_j}{\cos^2\theta_j} - \log q_j - 1] \le k\gamma'. \tag{4.18}$$

Now note that by Lemma 4.1 for any j (since we may assume without loss of generality that $\gamma_2 \ge 1$),

$$\begin{aligned}
\log\frac{\|e_{j+1}\|}{\|e_j\|} &\le \log\gamma_2(1 + \frac{1}{\cos\theta_j} + \frac{1}{q_j}) \\
&= \log\gamma_2 + \log(q_j + \frac{q_j}{\cos\theta_j} + 1) - \log q_j \\
&\le \log\gamma_2 + \frac{q_j}{\cos\theta_j} + q_j - \log q_j \\
&\le \log\gamma_2 + \frac{q_j}{\cos\theta_j} + 2q_j - 3\log q_j \\
&\le \log\gamma_2 + 3(\frac{q_j}{\cos\theta_j} - \log q_j).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\sum_{j\in S}\log\frac{\|e_{j+1}\|}{\|e_j\|} &\le 3\sum_{j\in S}[\frac{q_j}{\cos^2\theta_j} - \log q_j - 1] + k(3 + \log\gamma_2) \\
&\le 3\sum_{j=0}^{k}[\frac{q_j}{\cos^2\theta_j} - \log q_j - 1] + k(3 + \log\gamma_2) \le (3\gamma' + 3 + \log\gamma_2)k.
\end{aligned}$$

by the fact that all terms in the sum are nonnegative and by (4.18). The nonnegativity of the terms in the sum follows from the fact that

$$\frac{q_j}{\cos^2\theta_j} - \log q_j - 1 = (-\log\cos^2\theta_j) + (\frac{q_j}{\cos^2\theta_j} - 1 - \log\frac{q_j}{\cos^2\theta_j}), \tag{4.19}$$

and, by the properties of the logarithm, both expressions in parentheses are nonnegative.

The first result follows by taking the exponential of both sides of (4.19), and letting $\beta = \gamma_2 e^{3\gamma'+3}$. To establish the second result, we apply to (4.18) the same argument as in the proof of Theorem 2.1 of Byrd and Nocedal [1]. The relation (4.18) implies that for any k at least pk of the (nonnegative) terms in the sum are less than or equal to $\gamma'/(1-p)$. For these terms (4.19) implies a positive lower bound on $\cos\theta_j$ and upper and lower bounds on q_j. Then the existence of the constants β_1 and β_2 in (4.17) follows from Lemma 4.1. □

Now we are ready to prove our main result, which we restate here.

Theorem 4.1 *Assume that the sequence $\{x_k\}$ is generated by Algorithm 4.1 and Assumptions A1-3 hold. If $x_k \to x_*$, and $\lambda_k \to \lambda_*$, then $x_k \to x_*$ R-superlinearly.*

Proof: Suppose that the convergence is not R-superlinear. Then there exists a positive constant r and a subsequence $\mathcal{K}$ such that

$$\|e_k\| > r^k \text{ for all } k \in \mathcal{K}. \tag{4.20}$$

We will derive a contradiction from this assumption. Consider the recursion established in Lemma 4.3,

$$\psi_{k+1} \leq \psi_k - \frac{q_k}{\cos^2 \theta_k} + \log q_k + 1 + \gamma_4 \sigma_k + \gamma_3 \left(\frac{\|h_k\|}{\|s_k\|} \right)^2 .$$

Let

$$\mathcal{P}_k = \left\{ j \in [1,k] : \frac{\|h_j\|}{\|s_j\|} \geq \sqrt{\frac{\sigma_j}{\gamma_3}} \right\} . \tag{4.21}$$

and let $\pi_k = \frac{1}{k}|\mathcal{P}_k|$ where $|.|$ denotes cardinality.

Case 1: $\{\pi_k\}_{k \in \mathcal{K}}$ converges to 0.

Note that for $k \in \mathcal{K}$ Lemma 3.3 implies that

$$0 \leq \psi_{k+1} \leq \psi_0 + \sum_{j=0}^{k} [-\frac{q_j}{\cos^2 \theta_j} + \log q_j + 1 + \gamma_4 \sigma_j + \sigma_j] + k \pi_k \gamma_5$$

for a constant γ_5 since $\|h_j\|/\|s_j\|$ is uniformly bounded above. Therefore

$$\frac{1}{k} \sum_{j=0}^{k} [\frac{q_j}{\cos^2 \theta_j} - \log q_j - 1] \leq \frac{1}{k} \psi_0 + \frac{1}{k} \sum_{j=0}^{k} (\gamma_4 + 1) \sigma_j + \pi_k \gamma_5 . \tag{4.22}$$

Since we are assuming that $\{e_k\}$ and a subsequence of $\{\pi_k\}$ converge to 0, the right hand side and, thus, the left hand side of (4.22) converge to 0 for this subsequence. Therefore for any $\delta > 0$ there exists k_0 such that if $k > k_0$ and $k \in \mathcal{K}$ then

$$\frac{1}{k} \sum_{j=0}^{k} [\frac{q_j}{\cos^2 \theta_j} - \log q_j - 1] \leq \frac{\delta}{2} .$$

Since each summand is nonnegative, this implies that $q_j / \cos^2 \theta_j - \log q_j - 1 \leq \delta$ for at least $k/2$ values of $j \leq k$.

Now note that

$$\frac{q_j}{\cos^2 \theta_j} - \log q_j - 1 = [\frac{q_j}{\cos^2 \theta_j} - 1 - \log \frac{q_j}{\cos^2 \theta_j}] + [-\log \cos^2 \theta_j]$$

and both quantities in square brackets are nonnegative so that by choosing δ sufficiently small we can make $|q_j - 1|$ and $1 - \cos \theta_j$ arbitrarily small for half the iterates. By Lemma 4.1 the quantity $\|s_j\|/\|e_j\|$ is bounded above for those iterates. Now consider (4.7), and note that the quantity

$(q_k^2/\cos^2\theta_k - 2q_k + 1)^{1/2}$ is zero when $q_k = \cos\theta_k = 1$ and is continuous at that point, so by (4.7) $\|e_{j+1}\|/\|e_j\|$ can be made arbitrarily small for those iterates.

Therefore we have that for any $\epsilon > 0$ there exists k_0 such that if $k > k_0, k \in \mathcal{K}$ then $\|e_{j+1}\|/\|e_j\| < \epsilon$ for $k/2$ values of $j \le k$. Let $S = \{j \le k : \|e_{j+1}\|/\|e_j\| < \epsilon\}$. This implies that

$$\prod_{j=1}^{k} \frac{\|e_{j+1}\|}{\|e_j\|} = \prod_{j\in S} \frac{\|e_{j+1}\|}{\|e_j\|} \prod_{j\notin S} \frac{\|e_{j+1}\|}{\|e_j\|} \le \epsilon^{\frac{k}{2}}\beta^k,$$

using the bound (4.16).

By choosing ϵ small enough we see that $\left(\prod_{j=1}^{k} \frac{\|e_{j+1}\|}{\|e_j\|}\right)^{\frac{1}{k}}$ is arbitrarily small for all sufficiently large $k \in \mathcal{K}$ thus contradicting (4.20) in Case 1.

Case 2: There is an infinite subset $\mathcal{K}' \subset \mathcal{K}$ and a constant $\hat{\pi} > 0$ such that $\pi_k \ge \hat{\pi}$ for all $k \in \mathcal{K}'$.

Apply Lemma 4.4 with $p > 1 - \frac{\hat{\pi}}{2}$ (note that $\hat{\pi} \le 1$). Consider $k \in \mathcal{K}'$ and $\mathcal{J}_k$, the set of iterates defined by (4.17). Now define the set

$$\mathcal{T}_k = \{j \in \mathcal{P}_k \cap \mathcal{J}_k : j - 1 \in \mathcal{J}_k\}.$$

The number of elements in $\mathcal{P}_k$ that are not in $\mathcal{T}_k$ is no more than the number of indices $j \le k$ such that j or $j-1$ is not in $\mathcal{J}_k$, which is at most twice the cardinality of the set $[1,k] - \mathcal{J}_k$. Therefore,

$$\begin{aligned} |\mathcal{T}_k| &\ge |\mathcal{P}_k| - 2(k - |\mathcal{J}_k|) \\ &\ge \hat{\pi}k - 2(k - pk) \\ &= (\hat{\pi} + 2p - 2)k = \tau k. \end{aligned} \tag{4.23}$$

where $\tau = \hat{\pi} + 2p - 2$ is positive by our choice of p.

For any $j \in \mathcal{T}_k$, by (4.17) and (4.21).

$$\|e_j\|^{\frac{3}{2}} \le \frac{1}{\beta_1}\|s_j\|\|e_j\|^{\frac{1}{2}} \le \frac{\sqrt{\gamma_3}}{\beta_1}\|h_j\|$$

Expanding this h_j we get

$$\|h(x_j)\| \le \|h(x_{j-1}) + A_{j-1}^T s_{j-1}\| + \gamma_6\|s_{j-1}\|^2 = \gamma_6\|s_{j-1}\|^2$$

for some constant γ_6. Applying (4.17) at $j-1$ gives,

$$\|e_j\|^{\frac{3}{2}} \le \frac{\sqrt{\gamma_3}\gamma_6}{\beta_1}\|s_{j-1}\|^2 \le \frac{\sqrt{\gamma_3}\gamma_6\beta_2^2}{\beta_1}\|e_{j-1}\|^2.$$

Thus for the at least τk indices in $\mathcal{T}_k$ we have

$$\|e_j\| \le \gamma_7\|e_{j-1}\|^{\frac{4}{3}} \tag{4.24}$$

where $\gamma_7 = (\sqrt{\gamma_3}\gamma_6\beta_2^2/\beta_1)^{2/3}$.

Now since the sequenee converges we can choose k_0 so that $\gamma_7 \|e_j\|^{\frac{1}{3}} \leq \left(\frac{r}{2\beta}\right)^{\frac{1}{r}}$ for all $j \geq k_0 - 1$, where r is as in (4.20) and, without loss of generality, may be assumed to satisfy $\frac{r}{2\beta} < \frac{r}{2} <\leq 1$. Therefore, for any $k > k_0$ such that $k \in \mathcal{K}'$ we have, using Lemma 4.4, (4.24) and (4.23),

$$\begin{aligned}
\|e_k\| &= \|e_{k_0-1}\| \prod_{j=k_0}^{k} \frac{\|e_j\|}{\|e_{j-1}\|} \\
&= \|e_{k_0-1}\| \prod_{j\in[k_0,k]-\mathcal{T}_k} \frac{\|e_j\|}{\|e_{j-1}\|} \prod_{j\in\mathcal{T}_k\cap[k_0,k]} \frac{\|e_j\|}{\|e_{j-1}\|} \\
&\leq \|e_{k_0-1}\|\beta^k \prod_{j\in\mathcal{T}_k\cap[k_0,k]} (\gamma_7 \|e_{j-1}\|^{\frac{1}{3}}) \\
&\leq \|e_{k_0-1}\|\beta^k \left(\frac{r}{2\beta}\right)^{\frac{1}{r}|\mathcal{T}_k\cap[k_0,k]|} \\
&\leq \|e_{k_0-1}\|\beta^k \left(\frac{r}{2\beta}\right)^{k-\frac{k_0}{r}} \\
&\leq \|e_{k_0-1}\| \left(\frac{r}{2\beta}\right)^{-\frac{k_0}{r}} \left(\frac{r}{2}\right)^{k}
\end{aligned}$$

For k sufficiently large this violates the assumption (4.20) for Case 2. Thus the convergence must be R-superlinear. □

Although we have assumed in Theorem 4.1 that both sequences $\{x_k\}$ and $\{\lambda_k\}$ are convergent, it is interesting to note that if λ_k is given by the least-squares multiplier estimate (3.16), then convergence of $\{\lambda_k\}$ follows from convergence of $\{x_k\}$, so the assumption of multipler convergence is not needed in Theorem 5.1.

Theorem 5.1 establishes an R-superlinear rate of convergence, and we do not now see any way to strengthen the result to show Q-superlinear convergence for our algorithm. However, if we instead choose ρ_k to be fixed and sufficiently large, we can prove the following.

Corollary 4.1 *Consider a modification of Algorithm 4.1 where in Step 4, for sufficiently large k, ρ_k is chosen to be equal to a constant greater than $\hat{\rho}$ satisfying (3.3) for all large k. Then if $x_k \rightarrow x_*$ and $\lambda_k \rightarrow \lambda_*$, it follows that $x_k \rightarrow x_*$ Q-superlinearly. That is*

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \rightarrow 0. \tag{4.25}$$

Proof: Note that as long as ρ_k satisfies (3.3) and is bounded then Lemma 4.3 holds, and thus Theorem 4.1 holds also. Then by Theorem 4.1 we still have R-superlinear convergence which implies

$$\sum_{k=0}^{\infty} \|x_k - x_*\| < \infty. \tag{4.26}$$

However, the modified algorithm is equivalent, for large k, to the fixed ρ version of SALSA analyzed by Tapia in [25]. It then follows from (4.26) and Theorem 7.2 of [25] that convergence of x_k is Q-superlinear. □

A Corollary in the Unconstrained Case. It is interesting to note that if we apply Theorem 4.1 in the case of unconstrained optimization it implies a new result about the convergence of the unconstrained BFGS method.

Corollary 4.2 *If x_* is a local minimizer of the function $f(x)$ such that $\nabla^2 f(x_*)$ is nonsingular, and the sequence $\{x_k\}$ generated by the BFGS method with step-length one converges to x_*, then the convergence is Q-superlinear.*

Proof: Note that Algorithm 4.1 applied to a problem with no constraints is simply the BFGS method. By Theorem 4.1, if the iterates converge to the solution they do so R-superlinearly. This of course implies that

$$\sum_{k=0}^{\infty} \|x_k - x_*\| < \infty. \tag{4.27}$$

By Theorems 3.4 and 3.5 of Dennis and Moré [5] this implies the sequence converges Q-superlinearly. □

Recently, A. Griewank [11] has shown us an alternative proof of Corollary 4.2 using techniques developed in [10].

5 Numerical Experiments

The algorithms described in Section 3 have been programmed and tested on a SUN 3/50 Workstation in double precision FORTRAN with a machine epsilon of about 2×10^{-16}. The tolerance for the stopping criterion was chosen as $tol = 10^{-6}$ and the allowed maximum iteration number was $mxiter = 100$. All the other parameters used in the algorithm are as specified in Section 3. In particular, $\nu = \beta_1 = 0.01$. We now give a couple of details about our numerical experiments that are not stated in the description of the algorithms.

5.1 Experiment Description

In our implementation we always set the initial Hessian approximation to a scalar multiple of the identity matrix. A pre-update scaling proposed by Oren and Spedicato [17] for use with the BFGS secant method for unconstrained optimization has been adapted to both algorithms SALSA and PDA to give this scalar. Following Shanno and Phua [23] we perform the pre-update scaling only at the first iteration.

In the unconstrained case the scaling factor is chosen so that the spectrum of the initial approximation B_0 overlaps the spectrum of the true Hessian of the objective near x_0. Now in SALSA we are approximating the Hessian of the augmented Lagrangian and in PDA we are approximating the Hessian of the standard Lagrangian. These facts indicate that we should set $B_0^L = \eta I$ where $\eta > 0$ is a pre-update scaling factor that for SALSA would naturally be given by

$$\eta = y_0^{S\,T} s_0 / s_0^T s_0. \tag{5.1}$$

For PDA it is appropriate to use

$$\eta = \begin{cases} y_0^{\ell\,T} s_0 / s_0^T s_0, & y_0^{\ell\,T} s_0 > 0 \\ 1, & \text{otherwise.} \end{cases} \tag{5.2}$$

Observe that according to our construction of SALSA, the factor η given by (5.1) will always be positive and therefore SALSA will always take advantage of the pre-update scaling. However, from (5.2) we see that this is not the case for the factor based on PDA. In order to ensure that any differences between the numerical performance of SALSA and PDA were not due to this difference in pre-update scaling, we used the choice (5.2) in for both SALSA and PDA in our experiments. This decision put SALSA at a slight disadvantage, as numerical experimentation showed that the choice (5.1) led to slightly better performance for SALSA than did the choice (5.2).

As can be seen from the algorithm description, the algorithms, SALSA and PDA, are forced to terminate in the following three situations:

1. B_k^L is numerically indefinite. This is the situation when the Cholesky factorization of $Z_k^T B_k Z_k$ can not be carried out or $s_k^T B_k s_k \leq 0$.

2. the number of back-trackings in the line search exceeds 10;

3. the number of iterations exceeds *mxiter*.

All three of these cases will be called *irregular terminations* in contrast to the regular termination which occurs when the stopping criterion is satisfied. In addition, the algorithms are stopped if a matrix A_k is found to be numerically rank deficient. However, this situation only occurred once in the entire sequence of experiments.

A set of 44 test problems has been chosen from Hock and Schittkowski [14] and Schittkowski [22]. A precise description of these problems can be found in the above two references. All the problems are numbered as in these references. Problems with numbers less than 200 (29 problems) are from Hock and Schittkowski [14] and the rest (15 problems) are from Schittkowski [22]. For those of the problems having inequality constraints, only the constraints active at the solution are included. Linearly constrained problems have been excluded from our test set.

Most of the test problems are so well-conditioned that the identity matrix is often too good an approximation matrix to really test the robustness of an algorithm. In order to test the robustness of algorithms SALSA and PDA, from each given standard test problem we construct four scaled variants. We first define a diagonal matrix D_q by

$$D_{ii} = 1 + \left(1 - \frac{i-1}{n-1}\right)(10^{-q} - 1), \; i = 1, 2, \ldots, n, \tag{5.3}$$

where $q \in \mathbf{R}^n$ is a control parameter. In our tests, for each given objective function f and constraint function h, we solve the following five problems

$$\begin{array}{ll} \text{minimize} & f(D_q x), \\ \text{subject to} & h(D_q x) = 0, \end{array} \tag{5.4}$$

for $q = 0, 1, 2, 3, 4$. Obviously, $q = 0$ corresponds to the original problem and $q > 0$ to the scaled variants. If the Hessian matrix of a function $f(x)$ is $H(x)$, then after the diagonal scaling, the Hessian of $f(D_q x)$ is $D_q H(x) D_q$. Since the condition number of D_q^2 is 10^{2q}, if $H(x_*)$ is well conditioned, then for q large, in general $D_q H(x_*) D_q$ will be relatively ill-conditioned compared with $H(x_*)$.

The starting points x_0 are chosen as

$$x_0 = x_s + (\gamma - 1)(x_s - x_*), \tag{5.5}$$

where x_s are the standard starting points given in [14] and [22]. However, for Problems 12, 316–322, 336 and 338, we use $x_0 = (10^{-4}, \cdots, 10^{-4})$ instead of the given $x_0 = 0$ because $A(0)$ has zero columns and therefore is not of full rank. It is easy to see that

$$\|x_0 - x_*\| = |\gamma| \, \|x_s - x_*\|.$$

The number γ is thus used to control the distance $\|x_0 - x_*\|$ and was given different values as described in Section 5.2. For each problem, we let the integer q vary from 0 to 4. The total number of test cases is 220.

In the sequel, by one function evaluation we mean an evaluation of the $(m+1)$-vector $[f(x), h(x)]$. Similarly, one gradient evaluation represents an evaluation of the $n \times (m+1)$ matrix $[g(x)\ A(x)]$. Since the algorithms require only one gradient evaluation per iteration, the number of iterations needed for a run is always one less than the number of gradient evaluations because iterations are counted from 0.

5.2 Numerical Results

It is interesting to see how the two updating methods, SALSA and PDA, behave locally without a line search. After deactivating the line search subroutine as well as the pre-update scaling (because without a line search the information obtained from the first iteration is usually unreliable), we ran both SALSA and PDA on the 220 test cases always using step-length one and starting from the standard starting points x_s given in [14] and [22] (i.e., we set $\gamma = 1$ in (5.5)). It turns out that the standard starting points are fairly close to the solutions because for all the problems at least one of the two algorithms converged for at least one value of q. We will call this test (220 test cases) the local test.

We also tested SALSA and PDA with the line search procedure described in Section 3.5 and with the pre-update scaling (5.2) on the same set of test problems. As already mentioned, the standard starting points as given in [14] and [22] are generally fairly close to the solutions. In order to test the algorithms in a realistic global environment, we set $\gamma = 10$ for the starting points defined in (5.5) but with a few exceptions. Because for all the q-values both algorithms failed to converge for Problem 72, we still set $\gamma = 1$ for this problem. We ran the two algorithms with the above prescribed starting points and with the line search subroutine on the 220 test cases for the pre-update scaling (5.2). We will call this test (220 test cases) the global test.

Detailed information on both the local and the global tests that used the pre-update scaling (5.2) can be found in Tables 3–6 in the Appendix. In Table 1, we list the average numbers of function and gradient evaluations required by SALSA and PDA. To distinguish the standard test problems with its scaled variants, we present the results for $q = 0$ (standard) and for $q > 0$ (scaled) separately. The average number for each category is taken over all test cases in that category for which both SALSA and PDA converged. For the local test, since the number of function evaluations is always equal to the number of gradient evaluations, only one number is given for each category. For the global test, in each category the average number of function evaluations is given, followed by the average number of gradient evaluations separated with a colon. The rest of the table should be self-explanatory.

Table 1: Average Numbers of Function and Gradient Evaluations

Local Test				Global Test			
$q = 0$		$q = 1, 2, 3, 4$		$q = 0$		$q = 1, 2, 3, 4$	
SALSA	PDA	SALSA	PDA	SALSA	PDA	SALSA	PDA
23	21	32	31	24:29	27:29	28:33	30:37

As one can see from Table 1, the numbers of function and gradient evaluations required by SALSA and PDA are comparable for test cases where both algorithms converged. Therefore, we infer based on our numerical experiments that as far as efficiency is concerned, SALSA and PDA appear comparable.

However, we observe that SALSA has displayed a somewhat higher degree of robustness. This can be seen from Table 2 where the irregular termination behavior of SALSA and PDA is summarized. As can be seen from the table, for the total number of 450 test cases, PDA had more irregular terminations than SALSA did (59 vs. 42). However, since most of PDA's irregular terminations occurred in the local test, it does seem that the line search and the scalings helped to narrow the gap in robustness between SALSA and PDA.

Table 2: Number of Irregular Terminations

Local Test				Global Test			
$q = 0$		$q = 1, 2, 3, 4$		$q = 0$		$q = 1, 2, 3, 4$	
SALSA	PDA	SALSA	PDA	SALSA	PDA	SALSA	PDA
4	9	20	26	2	6	16	18

We close this section_by providing some additional observations obtained from our numerical tests. Among all the updates made by SALSA in our tests, the back-up strategy was used about 24 percent of the time. Of course, the choice of β_1 in (3.9) affects how often the back-up strategy is used and a decrease in the value of β_1 will result in less usage of the back-up strategy. As a comparison, we also ran SALSA using Powell's damped BFGS update as a back-up strategy instead of the one described in Section 3.3. Very similar results were obtained, though Powell's damped BFGS update, as the back-up strategy, was used slightly more often and the number of function evaluations was slightly increased.

6 Concluding Remarks

SALSA appears to have certain theoretical advantages over PDA. On the one hand, if a value for the augmentation parameter happens to be picked up that is greater than the threshold value, under standard assumptions, it will have local and Q-superlinear convergence. Local convergence has not yet been established for Powell's damped BFGS method. On the other hand, if the augmentation parameter happens to be smaller than the threshold value, we have established, under much weaker and more realistic assumptions than those that were assumed by Powell, that SALSA will, if it converges, converge at an R-superlinear rate as has been proved for Powell's damped BFGS method. As an immediate corollary, we have that if the BFGS secant method in unconstrained optimization converges it gives Q-superlinear convergence.

Our numerical experiments have shown that for a fairly large set of test problems the overall numerical performance of SALSA was moderately better than that of PDA in terms of robustness as measured by the number of irregular terminations. The higher degree of robustness of SALSA is likely due to the fact that B_k is not involved in y_k^S but is involved in y_k^P (see (3.1) and (1.10)).

Based on the established convergence results and our computational experiments, we have been led to the conclusion that in addition to its strong theoretical properties, the structured augmented Lagrangian BFGS secant method if properly implemented also performs experimentally at least as well as Powell's damped BFGS secant method.

References

[1] R. H. Byrd and J. Nocedal. An analysis of reduced Hessian methods for constrained optimization. Technical report CU-CS-398-88, Dept. of Computer Science, University of Colorado, 1988, (to appear in *Math. Prog.*).

[2] R. H. Byrd and J. Nocedal. A tool for the analysis of quasi-newton methods with application to unconstrained optimization. *SIAM J. Num. Analy.*, 26:727–739, 1989.

[3] R. M. Chamberlain. Some examples of cycling in variable metric method for constrained optimization. *Math. Prog.*, 16:378–383, 1979.

[4] T. F. Coleman and A. R. Conn. On the local convergence of a quasi-Newton method for the nonlinear programming problem. *SIAM J. Num. Analy.*, 21:755-769, 1984.

[5] J. E. Dennis Jr. and J. J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. *Math. Comp*, 28:549–560, 1974.

[6] P. Fenyes. *Partitioned quasi-Newton methods for nonlinear equality constrained optimization.* PhD thesis, Dept. of Computer Science, Cornell University, 1987.

[7] R. Fontecilla. Local convergence of secant methods for nonlinear constrained optimization. *SIAM J. Num. Anal.*, 25:692–712, 1988.

[8] R. Fontecilla, T. Steihaug, and R. A. Tapia. A convergence theory for a class of quasi-Newton methods for constrained optimization. *SIAM J. Num. Anal.*, 24:1133–1151, 1987.

[9] S.T. Glad. Properties of updating methods for the multipliers in augmented Lagrangians. *J.O.T.A.*, 28:135–156, 1979.

[10] A. Griewank. The global convergence of partitioned BFGS on semi-smooth problems with convex decompositions. ANL/MCS-TM-105, Mathematics and Computer Science Division, Argonne National Laboratory, 1987.

[11] A. Griewank. Private communication, 1989.

[12] S-P. Han. Superlinear convergent variable metric algorithms for general nonlinear programming. *Math. Prog.*, 11:263–282, 1976.

[13] S-P. Han. A globally convergent method for nonlinear programming. *J.O.T.A.*, 22:297–309, 1977.

[14] W. Hock and K. Schittkowski. *Test examples for nonlinear programming codes (Lecture notes in eco. and math. systems 187).* Springer-Verlag, Berlin, 1981.

[15] H. J. Martinez. *A numerical investigation on the BFGS update.* Master's thesis, Dept. Math. Sciences, Rice University, 1986.

[16] J. Nocedal and M. Overton. Projected Hessian updating algorithms for nonlinear constrained optimization. *SIAM J. Num. Anal.*, 22:821–850, 1985.

[17] S.S. Oren and E. Spedicato. Optimal conditioning of self-scaling variable metric algorithms. *Math. Prog.* 10, pp. 70-90, 1976.

[18] M. J. D. Powell. The convergence of variable metric method for nonlinearly constrained optimization calaulation. in *Nonlinear Programming 3*, O. Mangasarian, R. Meyer, and S. Robinson, editors, Academic Press, New York, 1978.

[19] M. J. D. Powell. A_fast algorithm for nonlinearly constrained optimization calaulation. in *Numerical analysis Proceedings Dundee 1977*, G. A. Watson, editor, Springer-Verlag, 1978.

[20] M. J. D. Powell. Extensions to subroutine VFO2AD. in *System modeling and optimization, Lecture notes in control and information sciences 38*, R. F. Drenick and F. Kozin, editors, Springer-Verlag, New York, 1982.

[21] M. J. D. Powell. The performance of two subroutines for constrained optimization on some difficult test problems. in *Numerical Optimization 1984*, P. T. Boggs, R. Byrd, and R. Schnabel, editors. SIAM, 1985.

[22] K. Schittkowski. *More test examples for nonlinear programming codes (Lecture notes in eco. and math. systems 282)*. Springer-Verlag, Berlin, 1987.

[23] D. F. Shanno and K. H. Phua. Matrix conditioning and nonlinear optimization. *Math. Prog.*, 14:145–160, 1978.

[24] R. Tapia. Diagonalized multiplier methods and quasi-Newton methods for constrained optimization. *J.O.T.A.*, 22:135–194, 1977.

[25] R. Tapia. On secant updates for use in general constrained optimization. *Math. Comp.*, 181-202, 1988.

Appendix: Tables

Detailed information on our numerical experiments is given here.

In all the tables, the problem are numbered after Hock and Schittkowski [14] and Schittkowski [22] and are specified in the first column along with the corresponding numbers of variables and constraints $(n : m)$.

For each value of q, listed in the tables are $ng : nf$ — the numbers of gradient and function evaluations respectively, as well as the final values of $\|\nabla \ell(x_k, \lambda_k)\|_2$ when the algorithms terminate.

The irregular terminations are indicated by boxes around the values of $\|\nabla \ell(x_k, \lambda_k)\|_2$ which are greater than $tol = 10^{-6}$. The symbols "Inf" and "NaN" in the tables stand for "Infinity" for "Not a Number" under the IEEE floating point standard as implemented in the operating system SunOS 4.0.3. Basically, both indicate that a floating point overflow has occurred.

The three types of irregular terminations as listed in Section 5 can be distinguished as follows. If the number of gradient evaluations is 101, then the algorithm was stopped because the maximum number of iterations was exceeded. If a pair $ng : nf$ is followed by an asterisk "*", then the algorithm was stopped because the maximum number of back-tracking steps in the line search was exceeded. Otherwise, the irregular terminations was due to the numerical indefiniteness of the Hessian approximation matrix. For Problem 72 in the local test of Powell's Damped BFGS

Algorithm (PDA) the blank entry indicates the algorithm was terminated because the matrix A_k was found to be rank deficient.

Table 3: SALSA without line search

Prob.#	$q = 0$		$q = 1$		$q = 2$		$q = 3$		$q = 4$	
$(n : m)$	$ng : nf$	$\|\nabla\ell\|$	$ng : nf$	$\|\nabla\ell\|$	$ng : nf$	$\|\nabla\ell\|$	$ng : nf$	$\|\nabla\ell\|$	$ng : nf$	$\|\nabla\ell\|$
6 (2:1)	9:9	.2D-09	17:17	.4D-08	21:21	.7D-10	27:27	.5D-06	31:31	.2D-07
7 (2:1)	15:15	.2D-07	14:14	.6D-09	17:17	.3D-06	18:18	.2D-06	21:21	.1D-07
10 (2:1)	12:12	.1D-06	12:12	.1D-06	15:15	.3D-07	15:15	.3D-07	16:16	.8D-06
11 (2:1)	8:8	.2D-07	12:12	.1D-07	17:17	.4D-06	23:23	.3D-10	25:25	.8D-09
12 (2:1)	51:51	.6D-09	46:46	.5D-06	35:35	.1D-06	33:33	.2D-06	25:25	.8D-06
26 (3:1)	34:34	.5D-06	31:31	.5D-06	36:36	.1D-05	43:43	.8D-06	47:47	.6D-06
27 (3:1)	34:34	.6D-06	22:22	.5D-07	8:8	.9D-06	4:4	.2D-06	4:4	.2D-08
29 (3:1)	13:13	.3D-08	14:14	.4D-07	31:31	.7D-06	34:34	.2D-07	41:41	.4D-07
39 (4:2)	13:13	.8D-06	14:14	.2D-06	19:19	.4D-06	24:24	.9D-06	10:10	.5D-07
40 (4:3)	8:8	.5D-07	33:33	.9D-06	6:6	.2D-07	6:6	.4D-11	6:6	.3D-11
43 (4:2)	12:12	.4D-07	17:17	.3D-07	39:39	.5D-07	57:57	.2D-06	88:88	.2D-06
46 (5:2)	101:101	$\boxed{\text{.2D+00}}$	16:16	$\boxed{\text{.4D+60}}$	29:29	.1D-06	38:38	.2D-06	38:38	.3D-06
47 (5:3)	68:68	.4D-07	23:23	.3D-07	36:36	.8D-06	52:52	.1D-06	79:79	.7D-06
56 (7:4)	14:14	.5D-06	101:101	$\boxed{\text{.3D+02}}$	10:10	.5D-07	11:11	.9D-06	12:12	.5D-06
60 (3:1)	11:11	.8D-06	19:19	.3D-06	24:24	.7D-07	39:39	.2D-08	37:37	.6D-06
61 (3:2)	11:11	.3D-11	11:11	.5D-09	14:14	.6D-06	15:15	.2D-07	16:16	.2D-06
63 (3:2)	8:8	.6D-06	13:13	.4D-08	14:14	.2D-06	15:15	.9D-07	15:15	.8D-06
65 (3:1)	10:10	.7D-07	27:27	.2D-07	28:28	.3D-07	37:37	.8D-08	40:40	.4D-07
66 (3:2)	7:7	.3D-06	27:27	.9D-07	101:101	$\boxed{\text{NaN}}$	101:101	$\boxed{\text{NaN}}$	101:101	$\boxed{\text{NaN}}$
71 (4:3)	6:6	.4D-07	14:14	.5D-06	19:19	.3D-07	40:40	.2D-07	63:63	.2D-06
72 (4:2)	21:21	.2D-06	31:31	.3D-06	101:101	$\boxed{\text{NaN}}$	50:50	$\boxed{\text{.7D+00}}$	43:43	$\boxed{\text{.7D+00}}$
77 (5:2)	43:43	.4D-06	101:101	$\boxed{\text{.1D-01}}$	31:31	.1D-06	37:37	.4D-06	43:43	.2D-06
78 (5:3)	8:8	.4D-07	51:51	.2D-06	32:32	.3D-06	29:29	.1D-06	8:8	.6D-07
79 (5:3)	11:11	.1D-06	15:15	.8D-08	49:49	.3D-07	83:83	.2D-07	75:75	.5D-08
80 (5:3)	7:7	.8D-08	33:33	.4D-07	32:32	.2D-06	39:39	.1D-08	6:6	.8D-07
81 (5:3)	10:10	.6D-08	22:22	.2D-06	101:101	$\boxed{\text{NaN}}$	101:101	$\boxed{\text{.1D+00}}$	6:6	.8D-07
93 (6:2)	33:33	.3D-06	6:6	$\boxed{\text{.1D+28}}$	4:4	$\boxed{\text{.2D+44}}$	24:24	$\boxed{\text{.2D+01}}$	29:29	$\boxed{\text{.4D+07}}$
100 (7:2)	51:51	.3D-06	28:28	.5D-06	37:37	.2D-06	47:47	.1D-06	52:52	.6D-06
104 (8:4)	27:27	.4D-06	101:101	$\boxed{\text{.5D+09}}$	101:101	$\boxed{\text{.5D+02}}$	101:101	$\boxed{\text{.5D+02}}$	101:101	$\boxed{\text{.5D+05}}$
106 (8:6)	43:43	$\boxed{\text{.1D+15}}$	29:29	$\boxed{\text{.2D+09}}$	89:89	$\boxed{\text{.8D+08}}$	61:61	.3D-06	61:61	.6D-06
216 (2:1)	25:25	.1D-06	9:9	.2D-07	22:22	.1D-09	29:29	.5D-07	38:38	.7D-07
219 (4:2)	18:18	.2D-07	22:22	.2D-07	25:25	.2D-06	30:30	.6D-07	17:17	.2D-06
316 (2:1)	60:60	.2D-08	60:60	.4D-07	41:41	.4D-08	37:37	.4D-08	27:27	.3D-06
317 (2:1)	59:59	.2D-09	54:54	.3D-07	48:48	.4D-07	36:36	.1D-07	28:28	.5D-11
318 (2:1)	62:62	.4D-06	60:60	.3D-08	43:43	.8D-07	59:59	.6D-09	28:28	.3D-11
319 (2:1)	59:59	.9D-11	59:59	.6D-08	53:53	.8D-09	44:44	.2D-07	33:33	.1D-06
320 (2:1)	49:49	.2D-06	70:70	.3D-07	53:53	.3D-10	43:43	.2D-06	37:37	.1D-07
321 (2:1)	44:44	.8D-06	71:71	.4D-08	53:53	.3D-10	46:46	.7D-06	44:44	.2D-08
322 (2:1)	25:25	.3D-11	39:39	.4D-06	64:64	.4D-08	50:50	.3D-06	51:51	.6D-08
335 (3:2)	25:25	.7D-08	28:28	.3D-07	28:28	.3D-06	37:37	.4D-07	46:46	.2D-06
336 (3:2)	48:48	.2D-08	75:75	.7D-07	76:76	.2D-06	72:72	.1D-05	72:72	.8D-09
338 (3:2)	56:56	.4D-06	43:43	.2D-09	36:36	.2D-07	36:36	.5D-08	30:30	.1D-07
355 (4:1)	7:7	$\boxed{\text{.3D+47}}$	41:41	.2D-07	101:101	$\boxed{\text{.3D+00}}$	24:24	$\boxed{\text{.3D+38}}$	59:59	$\boxed{\text{.3D+68}}$
373 (9:6)	101:101	$\boxed{\text{.2D+04}}$	14:14	.5D-06	21:21	.1D-07	26:26	.8D-07	32:32	.2D-06
375 (10:9)	26:26	.4D-10	16:16	.2D-07	20:20	.4D-06	15:15	.1D-07	16:16	.1D-09

Table 4: PDA without line search

Prob.#	$q = 0$		$q = 1$		$q = 2$		$q = 3$		$q = 4$	
$(n : m)$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$
6 (2:1)	13:13	.5D-09	14:14	.3D-08	18:18	.1D-11	21:21	.8D-08	26:26	.2D-13
7 (2:1)	15:15	.4D-08	21:21	.2D-10	47:47	.1D-06	52:52	.2D-07	46:46	.2D-08
10 (2:1)	12:12	.1D-06	13:13	.2D-06	17:17	.8D-10	20:20	.1D-08	22:22	.3D-08
11 (2:1)	8:8	.2D-07	13:13	.5D-07	18:18	.3D-06	24:24	.8D-08	32:32	.1D-06
12 (2:1)	21:21	.2D-11	24:24	.2D-10	39:39	.2D-09	28:28	.9D-06	30:30	.2D-06
26 (3:1)	34:34	.5D-06	31:31	.5D-06	35:35	.5D-06	43:43	.8D-06	46:46	.9D-06
27 (3:1)	31:31	.3D-07	22:22	.5D-07	12:12	.5D-06	4:4	.2D-06	4:4	.2D-08
29 (3:1)	49:49	$\boxed{\text{.2D+109}}$	13:13	.9D-07	25:25	.2D-07	29:29	.4D-07	35:35	.2D-07
39 (4:2)	13:13	.8D-06	14:14	.2D-06	16:16	.8D-06	18:18	.2D-08	10:10	.4D-07
40 (4:3)	8:8	.2D-07	39:39	.4D-08	6:6	.2D-07	6:6	.4D-11	6:6	.3D-11
43 (4:2)	12:12	.4D-07	16:16	.5D-08	44:44	.4D-08	63:63	.1D-06	87:87	.6D-06
46 (5:2)	62:62	.6D-06	89:89	.5D-06	23:23	$\boxed{\text{Inf}}$	39:39	.9D-06	43:43	.8D-07
47 (5:3)	54:54	.1D-06	29:29	.9D-07	65:65	.4D-07	64:64	.2D-06	87:87	.3D-06
56 (7:4)	101:101	$\boxed{\text{NaN}}$	7:7	.7D-06	10:10	.5D-07	12:12	.8D-06	13:13	.8D-06
60 (3:1)	11:11	.8D-06	20:20	.7D-07	28:28	.2D-08	27:27	.1D-06	37:37	.3D-06
61 (3:2)	11:11	.3D-11	15:15	.8D-06	19:19	.4D-08	22:22	.3D-07	24:24	.6D-07
63 (3:2)	8:8	.6D-06	15:15	.5D-09	21:21	.2D-09	19:19	.4D-07	22:22	.5D-08
65 (3:1)	14:14	.1D-06	22:22	.8D-06	32:32	.4D-06	98:98	.3D-07	95:95	.3D-06
66 (3:2)	7:7	.3D-06	28:28	.5D-08	101:101	$\boxed{\text{.2D+07}}$	75:75	.7D-08	101:101	$\boxed{\text{NaN}}$
71 (4:3)	6:6	.4D-07	15:15	.6D-09	23:23	.2D-06	51:51	.6D-06	73:73	.5D-09
72 (4:2)	21:21	.4D-06	37:37	.1D-06	101:101	$\boxed{\text{NaN}}$	——	$\boxed{}$	43:43	$\boxed{\text{.7D+00}}$
77 (5:2)	44:44	.3D-06	17:17	.8D-06	27:27	.3D-06	31:31	.2D-06	39:39	.7D-07
78 (5:3)	8:8	.4D-07	22:22	.8D-06	14:14	$\boxed{\text{.1D+49}}$	54:54	.5D-06	8:8	.1D-06
79 (5:3)	11:11	.1D-06	20:20	.2D-06	42:42	.9D-07	84:84	.1D-06	101:101	$\boxed{\text{.3D+02}}$
80 (5:3)	7:7	.8D-08	20:20	.1D-06	35:35	.5D-07	101:101	$\boxed{\text{NaN}}$	6:6	.8D-07
81 (5:3)	9:9	.8D-08	27:27	.3D-07	54:54	.7D-07	92:92	.4D-06	6:6	.8D-07
93 (6:2)	35:35	.3D-06	101:101	$\boxed{\text{NaN}}$	5:5	$\boxed{\text{.1D+47}}$	34:34	$\boxed{\text{.4D-01}}$	44:44	$\boxed{\text{.1D+08}}$
100 (7:2)	50:50	.5D-06	28:28	.5D-06	37:37	.2D-06	47:47	.1D-06	60:60	.3D-06
104 (8:4)	27:27	.4D-06	92:92	$\boxed{\text{.6D+14}}$	84:84	$\boxed{\text{.2D-01}}$	101:101	$\boxed{\text{.3D+00}}$	101:101	$\boxed{\text{.9D+01}}$
106 (8:6)	29:29	.5D-08	32:32	.1D-08	57:57	$\boxed{\text{.6D+12}}$	51:51	$\boxed{\text{.2D+14}}$	101:101	$\boxed{\text{.2D+21}}$
216 (2:1)	26:26	.1D-09	9:9	.2D-07	26:26	.4D-11	37:37	.3D-09	50:50	.5D-09
219 (4:2)	18:18	.2D-07	20:20	.9D-08	23:23	.3D-06	35:35	.1D-06	18:18	.2D-06
316 (2:1)	37:37	.3D-06	40:40	.2D-07	29:29	.3D-10	31:31	.1D-07	32:32	.2D-06
317 (2:1)	17:17	$\boxed{\text{.6D+02}}$	37:37	.5D-06	28:28	.5D-06	31:31	.2D-09	32:32	.2D-07
318 (2:1)	17:17	$\boxed{\text{.1D+03}}$	33:33	.9D-08	28:28	.6D-07	30:30	.2D-07	32:32	.2D-07
319 (2:1)	18:18	$\boxed{\text{.8D+02}}$	22:22	$\boxed{\text{.2D+03}}$	29:29	.7D-07	30:30	.1D-07	31:31	.3D-07
320 (2:1)	15:15	$\boxed{\text{.1D+03}}$	22:22	$\boxed{\text{.4D+02}}$	34:34	.1D-05	29:29	.1D-08	30:30	.2D-07
321 (2:1)	15:15	$\boxed{\text{.2D+02}}$	37:37	.7D-07	41:41	.2D-07	28:28	.1D-07	29:29	.4D-06
322 (2:1)	55:55	.1D-05	45:45	.8D-12	39:39	.1D-07	35:35	.4D-08	27:27	.5D-08
335 (3:2)	25:25	.1D-07	32:32	.1D-07	28:28	.3D-07	39:39	.3D-07	51:51	.3D-06
336 (3:2)	32:32	.8D-10	54:54	.2D-06	50:50	.2D-07	65:65	.1D-07	82:82	.5D-08
338 (3:2)	19:19	$\boxed{\text{.1D+03}}$	13:13	$\boxed{\text{.2D+06}}$	15:15	$\boxed{\text{.1D+02}}$	16:16	$\boxed{\text{.2D+01}}$	18:18	$\boxed{\text{.1D-02}}$
355 (4:1)	6:6	$\boxed{\text{.7D+106}}$	6:6	$\boxed{\text{.2D+84}}$	23:23	$\boxed{\text{.1D+71}}$	35:35	$\boxed{\text{.6D+61}}$	22:22	$\boxed{\text{.1D+60}}$
373 (9:6)	25:25	.3D-06	14:14	.5D-06	21:21	.1D-07	26:26	.2D-06	30:30	.2D-06
375 (10:9)	14:14	.2D-07	15:15	.1D-07	16:16	.4D-06	22:22	.3D-07	20:20	.5D-06

Table 5: SALSA with line search

Prob.#	$q = 0$		$q = 1$		$q = 2$		$q = 3$		$q = 4$	
$(n : m)$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$	$ng : nf$	$\|\nabla \ell\|$
6 (2:1)	13:19	.2D-07	15:16	.3D-07	26:30	.6D-09	29:31	.8D-08	38:43	.6D-06
7 (2:1)	18:20	.8D-08	20:21	.5D-07	26:28	.2D-09	24:24	.8D-06	28:32	.4D-07
10 (2:1)	17:19	.2D-10	18:20	.1D-05	17:17	.1D-06	19:19	.3D-07	19:19	.3D-06
11 (2:1)	11:12	.3D-07	15:18	.2D-06	21:26	.3D-09	21:26	.1D-07	30:34	.1D-08
12 (2:1)	23:31	.9D-08	23:28	.2D-09	20:27	.8D-11	38:63	.2D-08	30:43	.3D-07
26 (3:1)	32:34	.9D-06	31:31	.8D-06	34:35	.7D-06	47:48	.8D-06	53:54	.6D-06
27 (3:1)	51:63	.2D-06	31:39	.6D-07	61:72	.1D-07	17:17	.8D-06	4:4	.2D-07
29 (3:1)	27:32	.3D-07	15:15	.3D-07	19:22	.5D-09	23:25	.7D-06	29:32	.3D-06
39 (4:2)	18:20	.3D-06	25:27	.5D-07	22:22	.4D-06	28:28	.5D-07	14:14	.6D-06
40 (4:3)	10:11	.1D-06	22:23	.9D-06	7:7	.8D-08	7:7	.7D-12	7:7	.5D-12
43 (4:2)	26:37	.2D-06	29:42	.8D-06	37:52	.8D-06	47:69	.4D-08	55:81	.3D-06
46 (5:2)	101:116	$\boxed{.7D-02}$	25:29	.7D-06	41:47	.3D-06	60:70	.2D-06	89:99	.9D-06
47 (5:3)	57:61	.1D-06	22:23	.1D-07	40:44	.4D-06	36:38	.2D-06	69:76	.8D-06
56 (7:4)	20:27	.9D-07	40:45	.3D-07	41:80	$\boxed{.7D-05}$	12:12	.2D-07	15:15	.5D-06
60 (3:1)	23:25	.4D-06	25:26	.8D-07	31:33	.1D-06	35:36	.1D-05	41:42	.4D-06
61 (3:2)	25:36	.2D-10	17:20	.8D-07	14:16	.1D-06	16:17	.1D-06	17:18	.2D-09
63 (3:2)	9:9	.6D-09	10:10	.2D-08	11:12	.4D-07	12:13	.2D-07	13:14	.1D-07
65 (3:1)	24:34	.3D-06	44:65	.7D-06	29:35	.7D-06	31:36	.7D-08	34:39	.2D-07
66 (3:2)	8:8	.2D-09	18:27	.3D-07	6:7	$\boxed{.8D+00}$	6:21*	$\boxed{.8D+00}$	8:37*	$\boxed{.8D+00}$
71 (4:3)	9:10	.6D-07	10:10	.4D-08	11:12	.3D-06	12:15	.7D-09	16:23	.3D-06
72 (4:2)	26:27	.3D-06	31:31	.3D-06	32:35	$\boxed{.8D+00}$	89:114	$\boxed{.7D+00}$	39:42	$\boxed{.7D+00}$
77 (5:2)	78:90	.8D-06	24:26	.2D-06	29:30	.4D-06	34:35	.9D-06	39:40	.3D-08
78 (5:3)	28:39	.1D-06	11:12	.6D-06	41:43	.6D-06	54:77	.6D-06	8:8	.3D-06
79 (5:3)	17:18	.9D-07	20:21	.1D-07	22:23	.6D-07	40:49	.5D-07	59:72	.8D-07
80 (5:3)	9:9	.7D-06	18:18	.4D-06	32:36	.6D-07	65:100	.6D-06	6:6	.6D-06
81 (5:3)	19:22	.4D-06	24:24	.9D-07	42:48	.3D-06	93:103	.4D-06	6:6	.6D-06
93 (6:2)	30:35	.2D-06	84:108	.5D-07	72:108	.4D-07	101:205	$\boxed{.3D+01}$	37:172	$\boxed{.2D+04}$
100 (7:2)	82:93	.1D-06	53:60	.2D-06	55:61	.1D-06	68:76	.4D-06	75:86	.5D-07
104 (8:4)	25:26	.8D-06	28:31	.1D-06	33:38	.8D-06	14:30*	$\boxed{.2D-01}$	26:48*	$\boxed{.1D+01}$
106 (8:6)	40:42	.1D-08	40:63*	$\boxed{.2D+14}$	37:52	$\boxed{.4D+16}$	60:65	.9D-06	80:98	.3D-08
216 (2:1)	17:20	.4D-10	16:18	.1D-06	14:14	.4D-07	13:16	.2D-07	13:21	.1D-05
219 (4:2)	39:49	.9D-08	47:59	.4D-07	36:37	.3D-06	43:45	.5D-06	50:52	.5D-07
316 (2:1)	25:28	.9D-07	23:29	.2D-06	16:18	.3D-08	17:19	.2D-08	17:19	.7D-09
317 (2:1)	14:21	.2D-09	42:59	.3D-06	16:17	.2D-06	16:17	.5D-07	16:17	.2D-07
318 (2:1)	19:22	.4D-07	21:25	.2D-07	17:18	.1D-07	16:17	.1D-06	16:17	.8D-08
319 (2:1)	22:33	.2D-07	31:45	.1D-05	19:20	.3D-08	18:18	.2D-07	17:17	.4D-06
320 (2:1)	13:15	.1D-05	22:33	.1D-06	25:29	.2D-07	19:19	.9D-06	20:21	.3D-07
321 (2:1)	15:16	.5D-06	16:19	.1D-09	26:35	.5D-06	25:26	.3D-06	22:24	.6D-06
322 (2:1)	19:22	.2D-06	19:23	.1D-12	14:16	.3D-06	17:18	.6D-07	21:21	.3D-08
335 (3:2)	23:30	.2D-07	79:176	$\boxed{.3D-01}$	42:56	.3D-07	28:36	.6D-09	22:27	.6D-06
336 (3:2)	20:27	.8D-07	21:25	.2D-07	29:45	.1D-05	31:52	.1D-07	19:57*	$\boxed{.3D-02}$
338 (3:2)	9:9	.1D-07	15:17	.2D-06	20:28	.2D-07	22:33	.4D-06	30:46	.4D-06
355 (4:1)	84:112	.9D-06	32:39	.8D-07	93:166	.1D-06	101:150	$\boxed{.6D+00}$	101:188	$\boxed{.1D-01}$
373 (9:6)	101:206	$\boxed{.3D+01}$	101:203	$\boxed{.4D+01}$	34:41	.6D-06	42:51	.2D-07	48:58	.4D-06
375 (10:9)	11:11	.5D-06	15:16	.5D-07	16:17	.3D-06	16:16	.6D-06	18:18	.8D-10

Table 6: PDA with line search

Prob.#	$q=0$		$q=1$		$q=2$		$q=3$		$q=4$	
$(n:m)$	$ng:nf$	$\|\nabla\ell\|$	$ng:nf$	$\|\nabla\ell\|$	$ng:nf$	$\|\nabla\ell\|$	$ng:nf$	$\|\nabla\ell\|$	$ng:nf$	$\|\nabla\ell\|$
6 (2:1)	13:16	.3D-10	14:17	$\boxed{\text{.5D-05}}$	25:25	.1D-13	29:34	.4D-12	32:33	.3D-12
7 (2:1)	17:21	.5D-06	35:42	.3D-09	34:38	.2D-06	36:41	.3D-07	35:39	.6D-06
10 (2:1)	17:19	.2D-10	18:20	.1D-06	18:18	.6D-06	23:23	.2D-09	25:25	.9D-09
11 (2:1)	11:12	.3D-07	16:18	.7D-08	23:28	.5D-07	28:32	.3D-10	34:39	.1D-07
12 (2:1)	23:28	.2D-10	23:29	.5D-07	23:29	.5D-07	27:35	.9D-09	32:41	.3D-10
26 (3:1)	32:34	.9D-06	31:31	.8D-06	36:37	.6D-06	47:48	.8D-06	53:54	.6D-06
27 (3:1)	46:55	.2D-07	50:69	.2D-06	101:132	$\boxed{\text{.2D-01}}$	18:18	.3D-06	4:4	.2D-07
29 (3:1)	31:51	$\boxed{\text{.8D+08}}$	15:15	.3D-07	21:23	.9D-06	22:25	.2D-06	28:31	.4D-06
39 (4:2)	18:20	.3D-06	20:20	.4D-06	31:32	.2D-07	28:29	.4D-07	19:20	.4D-06
40 (4:3)	10:11	.1D-06	22:22	.9D-06	7:7	.8D-08	7:7	.7D-12	7:7	.5D-12
43 (4:2)	26:39	.3D-07	29:42	.5D-07	41:59	.1D-06	56:83	.3D-06	62:92	.3D-06
46 (5:2)	101:126	$\boxed{\text{.1D-03}}$	26:30	.4D-06	101:134	$\boxed{\text{.1D-05}}$	99:133	.8D-06	101:135	$\boxed{\text{.2D-05}}$
47 (5:3)	58:62	.6D-06	22:23	.1D-07	41:47	.3D-06	36:38	.2D-06	72:94	.9D-06
56 (7:4)	18:25	.9D-06	33:36	.5D-07	48:58*	$\boxed{\text{.3D+23}}$	12:12	.2D-07	15:15	.5D-06
60 (3:1)	24:27	.4D-07	23:24	.1D-07	32:33	.6D-07	36:38	.2D-06	41:42	.4D-06
61 (3:2)	18:21	.2D-06	15:16	.5D-06	17:18	.3D-10	20:20	.9D-08	21:22	.8D-08
63 (3:2)	9:9	.6D-09	10:10	.2D-08	12:12	.6D-06	15:15	.3D-08	18:18	.1D-07
65 (3:1)	25:37	.8D-06	27:42	.8D-09	29:35	.2D-07	36:44	.7D-06	35:49	.3D-06
66 (3:2)	8:8	.2D-09	18:24	.8D-06	26:35	.3D-06	36:43	.6D-06	36:50	.1D-07
71 (4:3)	9:10	.6D-07	10:10	.4D-08	12:13	.2D-08	16:19	.3D-09	29:38	.2D-06
72 (4:2)	26:27	.8D-06	32:33	.2D-06	35:62*	$\boxed{\text{.7D+00}}$	48:72*	$\boxed{\text{.3D+00}}$	43:43	$\boxed{\text{.7D+00}}$
77 (5:2)	80:89	.2D-06	23:24	.1D-06	29:30	.4D-06	34:35	.9D-06	38:39	.5D-07
78 (5:3)	24:32	.2D-08	11:12	.6D-06	29:34	.9D-07	67:102	.1D-07	8:8	.3D-06
79 (5:3)	17:18	.9D-07	20:21	.2D-07	21:22	.5D-07	48:54	.2D-07	48:57	.8D-07
80 (5:3)	9:9	.7D-06	18:18	.4D-06	27:28	.2D-06	62:95	.3D-07	6:6	.6D-06
81 (5:3)	18:20	.7D-08	23:25	.6D-07	33:34	.3D-06	90:94	.7D-06	6:6	.6D-06
93 (6:2)	29:34	.8D-07	87:111	.3D-07	33:41	.6D-06	59:74	.1D-06	62:138	$\boxed{\text{.2D+06}}$
100 (7:2)	83:96	.2D-06	53:60	.8D-06	54:59	.2D-06	61:66	.9D-06	66:70	.2D-06
104 (8:4)	25:26	.8D-06	28:31	.1D-06	36:40	.5D-07	14:30*	$\boxed{\text{.2D-01}}$	24:40*	$\boxed{\text{.8D+00}}$
106 (8:6)	42:45	$\boxed{\text{.2D+19}}$	32:42	$\boxed{\text{.2D+16}}$	45:55	$\boxed{\text{.4D+09}}$	63:68	.1D-05	64:92	$\boxed{\text{.6D+21}}$
216 (2:1)	17:20	.4D-10	16:18	.1D-06	14:14	.4D-07	17:19	.2D-09	27:35	.5D-06
219 (4:2)	49:57	.2D-07	80:115	.6D-08	52:60	.6D-06	46:85*	$\boxed{\text{.4D+00}}$	47:62	.2D-07
316 (2:1)	33:63	.3D-06	23:29	.2D-06	23:30	.8D-07	26:31	.2D-07	27:34	.2D-07
317 (2:1)	16:23	.2D-06	30:48	.3D-06	24:38	.2D-07	24:28	.4D-06	33:48	.3D-07
318 (2:1)	14:18	.5D-06	19:24	.4D-07	22:28	.9D-06	25:31	.8D-07	29:46	.2D-09
319 (2:1)	17:23	.7D-07	18:22	.3D-06	26:40	.2D-06	25:36	.1D-07	27:36	.2D-07
320 (2:1)	16:20	.6D-06	15:17	.6D-07	29:43	.1D-07	24:33	.2D-07	29:40	.9D-11
321 (2:1)	17:20	.2D-06	16:19	.1D-09	26:31	.8D-09	32:52	.1D-06	29:43	.2D-06
322 (2:1)	12:14	$\boxed{\text{.4D+01}}$	20:26	.4D-07	14:16	.3D-06	26:37	.5D-06	29:43	.2D-08
335 (3:2)	25:31	.3D-06	42:64	.8D-09	35:45	.5D-09	36:41	.1D-06	26:31	.4D-07
336 (3:2)	21:25	.4D-08	21:28	.3D-07	36:66	$\boxed{\text{.5D-02}}$	24:34	.9D-07	26:44	.1D-09
338 (3:2)	9:9	.1D-07	15:16	.1D-09	23:29	.1D-08	28:41	.6D-07	12:22	.3D-06
355 (4:1)	101:136	$\boxed{\text{.2D-01}}$	35:42	.4D-07	93:166	.1D-06	101:230	$\boxed{\text{.7D-01}}$	101:155	$\boxed{\text{.1D-01}}$
373 (9:6)	101:223	$\boxed{\text{.3D+01}}$	101:199	$\boxed{\text{.4D+01}}$	34:41	.6D-06	42:52	.4D-06	48:58	.4D-06
375 (10:9)	11:11	.5D-06	17:17	.7D-06	17:17	.5D-06	20:20	.3D-07	22:22	.4D-06