

AN IMPLICIT FILTERING ALGORITHM FOR OPTIMIZATION OF FUNCTIONS WITH MANY LOCAL MINIMA *

P. GILMORE[†] AND C. T. KELLEY[†]

Abstract. In this paper we describe and analyze an algorithm for certain box constrained optimization problems that may have several local minima. A paradigm for these problems is one in which the function to be minimized is the sum of a simple function, such as a convex quadratic, and high frequency, low amplitude terms which cause local minima away from the global minimum of the simple function. Our method is gradient based and therefore the performance can be improved by use of quasi-Newton methods.

Key words. filtering, projected gradient algorithm, quasi-Newton method

AMS(MOS) subject classifications. 65H10, 65K05, 65K10

1. Introduction. In this paper we describe and analyze an algorithm for bound constrained optimization problems that may have several local minima. The type of problem we have in mind is one in which the function to be minimized is the sum of a simple function, such as a convex quadratic, and high frequency, low amplitude terms which cause the local minima. Of particular interest is the case in which the amplitude of the high frequency components decays to zero near the local minima of the simple function. This algorithm, at various stages of its development, has been applied to such problems by a group in the departments of Mathematics and Electrical and Computer Engineering at North Carolina State University to a variety of optimization problems that arise in computer aided design of microwave devices [17], [16], [19], [20]. The algorithm is an extension of the projected gradient method [1] and as such is simple to implement and its performance can be improved by application of quasi-Newton methods. The purpose of this theoretical paper is to analyze the convergence properties of the method. The algorithm discussed in this paper was designed for the specific applications described fully in [17], [16], [19], and [20]. These papers put the numerical properties of the algorithm in context.

An example of the type of problem we have in mind is plotted in Figure 1.1, taken from [20], which is a graph of negative of the power added efficiency of a simulated semiconductor device against the real and imaginary parts of the second harmonic of load impedance, which are constrained to lie in the interval $[0, 80]$. The small amplitude, high frequency perturbation which dies off near the optimal point, $(0, 0)$, is clearly visible.

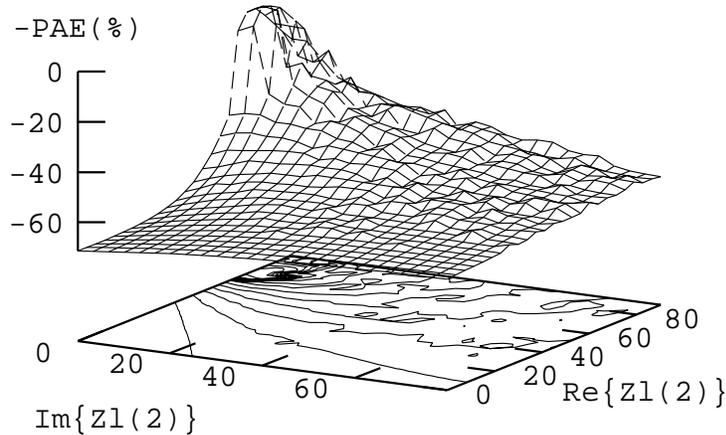
In this section we begin by discussing in general terms the class of problems we seek to solve, then mention some other possibilities, and give a brief description of our approach. We formally describe the basic form of our algorithm in § 2. In § 3 we relate the output of the algorithm to a class of problems like that represented in Figure 1.1. In § 4 and § 5 we state and prove some convergence results.

We seek to minimize a function f subject to simple bound constraints:

$$x \in \Omega = \{x \mid l^i \leq x^i \leq u^i\},$$

* This document was printed on January 12, 1994.

[†] North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, N. C. 27695-8205, USA. This research was supported by National Science Foundation grant #DMS-9024622, U. S. Army grant #DAAL0189K0906, and Air Force Office of Scientific Research grant #AFOSR-FQ8671-9101094. Computing activity was partially supported by an allocation of time from the North Carolina Supercomputing Center.

FIG. 1.1. *Power Added Efficiency*

where x^i denotes the i th component of the vector $x \in R^N$, but that we can only observe $\hat{f} = f + \phi$, where ϕ is small in magnitude relative to f but has high frequency oscillations that cause local minima. We do not require that ϕ be smooth or even continuous.

One way to avoid such local minima is to filter the high frequency components from some expansion of \hat{f} , by means of a discrete Fourier transform, for example. In this way one might expect that the filtered form of \hat{f} is a good approximation to f and does not have as many local minima as \hat{f} does. By applying a conventional minimization algorithm to the filtered form of \hat{f} , one might find the minimizer of f up to the accuracy allowed by the noise in the observation. By changing the filter as an iteration progresses, to admit higher frequencies near the minimizer, or by restarting the iteration with a filter that admits higher frequencies after convergence, one might hope to even avoid local minima in f itself. Another advantage of refining the filter as the iteration progresses is to deal with problems, such as the one represented in Figure 1.1, for which the perturbation ϕ is much smaller near the minimizer of f than elsewhere. Such problems were encountered in [17], [16], [19], and [20] in which noise from model errors was reduced near the solution of the optimization problem.

The disadvantage of applying a filter to \hat{f} is that one must sample the entire variable space in order to use the filter. An example of such an algorithm has been reported recently in [12]. An advantage from the point of view of this paper, but not from that of [12], of such an algorithm is that large amplitude high frequency terms may be eliminated and therefore the iterates may avoid steep valleys. In the work reported in [17], [16], [19], and [20], steep valleys in the objective could be attributed to errors. Filtering algorithms might not be appropriate in situations where steep valleys are significant.

Another approach is stochastic smoothing [13] where f is replaced by an average and the averages are changed as the iteration progresses. Almost sure convergence to the global minimum is proved in [13]. We do not prove such a strong result

for the algorithm we propose. The advantages our algorithm offers are simplicity and efficiency of implementation in that there is no preprocessing of the objective function and the analysis is completely deterministic. The price paid for this efficiency and simplicity is that some features of the objective, such as steep valleys, may be missed. This is reflected in the convergence results, which, in broad terms, assert that a local minimum of the unperturbed function will be identified up to the accuracy permitted by the perturbation ϕ . However, for the types of problems considered in [17], [16], [19], and [20], the decay of ϕ near the minimum allowed for high accuracy. We quantify this in Theorem 3.1.

We should also mention the multidirectional search algorithm for unconstrained problems proposed in [18]. This algorithm is based on the Nelder-Mead simplex algorithm. The multidirectional search algorithm uses a simplex that is rotated expanded and contracted as the algorithm proceeds. The size of this simplex corresponds to the length of the scales used in the implicit filtering algorithm. At the beginning of the optimization process the initial simplex is taken to be relatively large. This could, in principal, allow the multidirectional search algorithm to avoid local minima caused by the low amplitude high frequency term, although this was not the purpose of its design and is not covered by its analysis. As the algorithm proceeds the size of the simplex will decrease allowing the structure of the merit function to be resolved to a finer level of resolution. Torczon proved convergence of this algorithm to local minima for continuous functions in [18]. The cost of application of the algorithm in [18] is roughly the same as ours if centered differences are used to approximate gradients. In [9] Dennis and Torczon showed how the multidirectional search algorithm could be efficiently implemented on parallel processor computers.

Rather than sample the variable space as a true filtering algorithm would, we propose use of a finite-difference gradient based method, for example the projected gradient method [1], with the step size in the difference chosen as it would be if ϕ were floating point roundoff. For example, in the case of forward differences $h \approx \sqrt{\|\phi\|_\infty}$. Since $\|\phi\|_\infty$ is not known, we apply the finite-difference gradient based method, decrease h after convergence, and apply the finite-difference gradient based method again. We could terminate this outer iteration after a predetermined smallest value of h is reached or if we determine that no further progress is being made. We refer to the algorithm as *implicit filtering* because we use the differencing to “step over” the noise ϕ at varying levels of resolution, hence implicitly filtering the objective. This algorithm is, therefore, deterministic in both its implementation and its analysis.

A significant difference from the alternative approaches listed above is that the performance in the terminal phase of the iteration can be accelerated by a quasi-Newton method. This was very important from the point of view of the applications discussed in [17], [16], [19], and [20]. In those papers the SR1 update was found to be very useful and roughly 10% more efficient than the BFGS update. In § 4 we present some simple examples using the secant update for one dimensional problems that illustrate this point.

2. Specification of the Algorithm. As a basic algorithm we use the projected-gradient-Armijo algorithm from [1]. The form we use employs finite difference gradients. We let ∇_h be some finite difference gradient, using, for example, forward or centered differences, computed with step h . The algorithm takes as its input the function to be minimized, an initial iterate x , which is overwritten with an approximation to the minimizer, a stepsize h for the finite difference, a function $\tau(h)$ used for termination, a minimum step size $\bar{\alpha}$ for the line search, and a (small) parameter

σ used to measure sufficient decrease. In the specification of Algorithm `projgrad` we will use the notation

$$(2.1) \quad x(\alpha, h, f) = \mathcal{P}(x - \alpha \nabla_h f),$$

where \mathcal{P} denotes the projection onto the feasible set

$$\mathcal{P}(x)^i = \begin{cases} u^i & x^i > u^i \\ x^i & l^i \leq x^i \leq u^i \\ l^i & x^i < l^i. \end{cases}$$

In the description of the algorithms that follow and in the discussion in § 5 the Euclidean norm will be denoted by $\|\cdot\|$ and the ℓ^∞ norm $\|\cdot\|_\infty$ with a subscript.

ALGORITHM 2.1. *Algorithm* `projgrad`($f, x, h, \tau, \bar{\alpha}, \sigma$)

1. $k = 0$
2. Compute $\nabla_h f$.
 - (a) If $\|x - x(1, h, f)\| \leq \tau(h)$ terminate successfully.
3. Set $\alpha = 1$.
4. (a) If $\alpha < \bar{\alpha}$ terminate unsuccessfully.
 - (b) Compute $f(x(\alpha, h, f))$.
 - (c) If

$$f(x) - f(x(\alpha, h, f)) \geq \frac{\sigma \|x - x(\alpha, h, f)\|^2}{\alpha}$$

set $x = x(\alpha, h, f)$, $k = k + 1$ and go to step 2.

(d) $\alpha = \beta\alpha$

The roles of the parameters σ and β are the same as in standard discussions of the Armijo rule [1], [8], [14]. The role of $\bar{\alpha}$ is that of a safeguard to keep from reducing the step size too often when $-\nabla_h f$ is not a descent direction and to determine when no further reduction in h should be done. We have used $\bar{\alpha} = \beta^{10}$ in the application work reported in [17], [16], [19], and [20]. The analysis in § 5 can be directly extended to more general line search rules, such as the polynomial models in [8]. The cubic model was used in [17], [16], [19], and [20]. Extension to a trust region approach such as that in [5] or [15] should also be possible.

The basic form of the algorithm we propose in this paper requires a decreasing, finite, sequence of difference steps $\{h_i\}_{i=1}^m$, called *scales* and consists of the repeated application of Algorithm `projgrad` to \hat{f} .

ALGORITHM 2.2. *Algorithm* `imfilter`($\hat{f}, x, \{h_i\}, \tau, \bar{\alpha}, \sigma$)

- for $i = 1, \dots, m$
 - call `projgrad`($\hat{f}, x, h_i, \tau, \bar{\alpha}, \sigma$)

Passing through the decreasing sequence of scales is intended to have the effect of changing the filter as the iteration progresses to admit higher frequencies in \hat{f} . However, this is only a heuristic, as we do no actual filtering. Our strategy for selection of the sequence $\{h_i\}$ is also a heuristic. We make some more remarks on that later in this section. Note that h_i , like the temperature in an annealing algorithm, the simplex size in multidirectional search, or the time step in the algorithm in [12] in a sense measures the resolution of the optimizer. The important difference in Algorithm `imfilter` from these algorithms is that it is gradient based. Therefore it has the simultaneous advantages that it does not require sampling of the variable space, can be analyzed with Taylor series methods, is simple to implement, and can be accelerated by quasi-Newton methods.

At this point we make a few remarks on the goals and properties of Algorithm `imfilter`. As is the case with other filtering algorithms, such as that given in [12], our algorithm may miss global minima caused by large amplitude “spikes” in ϕ . We view this as desirable and do not think of our algorithm as a global optimization method, but rather as a method for dealing with a particular class of noisy functions. In the work reported in [17], [16], [19], and [20], spikes represented error and were best avoided.

A proper criterion for determining if Algorithm `imfilter` has succeeded is also a question. Even if each call to `projgrad` terminates successfully with a solution x , there is no guarantee that a second call to `imfilter` would leave x invariant since ϕ could change the output from the early calls to `projgrad`. Therefore it might be necessary to restart `imfilter`. We show in § 3 that such restarts are not necessary if ϕ decays sufficiently rapidly near a global minimum of f . We set as our goal the computation of x such that x is a minimum at every scale.

DEFINITION 2.1. *x is a minimum at all scales if `projgrad` leaves x invariant for all $h = h_1, \dots, h_m$.*

We compute a minimum at all scales by Algorithm `imfilter` and restarting, if necessary, until each call to `projgrad` leaves x invariant.

ALGORITHM 2.3. *Algorithm `allscale`($f, x, h, \tau, \bar{\alpha}, \sigma$)*

- *Until each call to `projgrad` leaves x invariant:
call `imfilter`($f, x, \{h_i\}, \tau, \bar{\alpha}, \sigma$)*

The central theoretical contributions of this paper are to show how a minimum at all scales is related to a global minimum for functions of the type plotted in Figure 1.1, where the perturbation decays near a minimum of f , and to give conditions on f, ϕ, τ , and $\bar{\alpha}$ under which Algorithm `allscale` terminates in finitely many steps and returns a minimum at all scales. We do these things in § 3, § 4, and § 5.

3. Minima at All Scales: Examples and Characterization. The idea, first advocated in [17], [16], [19], and [20], that a minimum at all scales, and not a global minimum or a minimum of an explicitly filtered function, should be the goal of the iteration is central to the algorithm and reflects the motivating problems where the amplitude of ϕ decays near the minimum. We begin with a theorem that illustrates the relationship between a minimum at all scales and a global minimum. We follow that with two examples to illustrate the ideas. A feature of these theorems and their proofs is the variety of subtle relations between the size of the perturbation and the curvature of f and the parameters $\{h_i\}, \sigma, \bar{\alpha}$ in the specification of the algorithm. These relationships are further explored in § 4 and § 5.

In the following theorem, as in [17], [16], [19], and [20], we use $\tau(h) = \bar{\tau}h$, for some $\bar{\tau} > 0$ and $h_k = \mu^k h_0$ for some $\mu \in (0, 1)$. We will assume that ∇_h is computed in Ω in a way that is at least first order accurate. We must make precise our assumptions on f and the way that ϕ decays near the global minimum of f .

ASSUMPTION 3.1. *Assume that $\hat{f} = f + \phi$, that f has a global minimum in Ω at x^* , ∇f is Lipschitz continuous in Ω with Lipschitz constant L . Assume that*

$$\Omega = \{x \mid l^i \leq x^i \leq u^i\} \subset R^N,$$

diameter $D_\Omega = \max(u^i - l^i) < \infty$. Assume that ∇_h is computed with forward, backward, or centered differences. Finally assume that there are $c_0, c_1, e_-,$ and $M_\phi >$

0 such that for all $x \in \Omega$

$$(3.1) \quad \begin{aligned} \|x - \mathcal{P}(x - \nabla f(x))\| &\geq c_0 \|x - x^*\|, \\ \|\nabla_h f - \nabla f(x)\| &\leq c_1 h, \\ |\phi(x)| &\leq M_\phi \max\{\|x - x^*\|^2, e_-^2\}. \end{aligned}$$

THEOREM 3.1. *Assume that Assumption 3.1 holds. Then if M_ϕ is sufficiently small there are C_1, C_2 and K , such that if*

$$(3.2) \quad h_0 \geq \frac{2C_1 M_\phi D_\Omega}{c_0 - C_2 M_\phi},$$

$h_k = \mu^k h_0$, for some $\mu \in (0, 1)$ and

$$\|x - x(1, h_k, \hat{f})\| = \|x - \mathcal{P}(x - \nabla_{h_k} \hat{f}(x))\| \leq \bar{\tau} h_k$$

for $k = 1, \dots, m$, then

$$\|x - x^*\| \leq K h_-,$$

where $h_- = \max(e_-, h_m)$.

Conversely, there is $\bar{\tau}_0$ such that if $\bar{\tau} \geq \bar{\tau}_0$ and $h_m \geq e_- \geq \|x - x^*\|$ then x is a minimum at all scales.

Proof. We note that Assumption 3.1 implies that there are C_1 and C_2 such that if $\bar{\phi} = C_1 M_\phi$ and $\nu = C_2 M_\phi$ then either $\|x - x^*\| < e_-$ or

$$(3.3) \quad \|\nabla_h \phi(x)\| \leq \bar{\phi} \|x - x^*\|^2 / h + \nu \max(e, h).$$

In fact, for forward or backward differences we have

$$\|\nabla_h \phi(x)\| \leq 2M_\phi \sqrt{N} (e + h)^2 / h,$$

leading to $C_1 = 2\sqrt{N}$ and $C_2 \leq 6\sqrt{N}$. Using second order centered differences would reduce C_1 and C_2 by factors of two but not eliminate ν entirely. Perhaps because of this reduction of $\bar{\phi}$ and ν , centered differences were found to reduce the sensitivity of the algorithm on the size of ϕ in [17]. We have, for M_ϕ sufficiently small, that

$$(3.4) \quad \bar{\phi} \leq \frac{c_0 - \nu}{4} \text{ and } 0 < \nu < c_0.$$

We let $e = \|x - x^*\|$. Our assumptions imply that either $e \leq e_-$ or for each $k = 0, \dots, m$

$$\begin{aligned} \bar{\tau} h_k &\geq \|x - \mathcal{P}(x - \nabla_h \hat{f}(x))\| \\ &\geq \|x - \mathcal{P}(x - \nabla_h f(x))\| - \|\nabla_h \phi(x)\| \\ &\geq \|x - \mathcal{P}(x - \nabla f(x))\| - \|\nabla f(x) - \nabla_h f(x)\| - \|\nabla_h \phi(x)\| \\ &\geq \|x - \mathcal{P}(x - \nabla f(x))\| - c_1 h_k - \left(\frac{\bar{\phi} e^2}{h_k} + \nu \max(e, h_k)\right) \\ &\geq c_0 e - c_1 h_k - \left(\frac{\bar{\phi} e^2}{h_k} + \nu \max(e, h_k)\right). \end{aligned}$$

Therefore, either $e \leq e_-$ or for each $k = 0, \dots, m$

$$(3.5) \quad c_0 e \leq \|x - \mathcal{P}(x - \nabla_{h_k} f(x))\| \leq (\bar{\tau} + c_1)h_k + \frac{\bar{\phi}e^2}{h_k} + \nu \max(e, h_k).$$

Setting $k = 0$ and using (3.2) implies that either $e \leq h_0$ or

$$(c_0 - \nu)e \leq (\bar{\tau} + c_1)h_k + \frac{(c_0 - \nu)e}{2}$$

and hence

$$(3.6) \quad e \leq \kappa h_0 = \mu^{-1} \kappa h_1,$$

where

$$\kappa = 2 \frac{\bar{\tau} + c_1}{c_0 - \nu}.$$

We now assume that M_ϕ is small enough so that

$$(3.7) \quad \frac{\bar{\phi} \max(\mu^{-1} \kappa, 2)}{c_0 - \nu} \leq 1/2$$

which is usually stronger (3.4).

We use (3.6) to induct on k and show that either $e \leq e_-$, or $e \leq \max(\kappa, 1)h_k$. Assume that $e \leq \kappa h_{k-1} \leq \mu^{-1} \kappa h_k$, which we have verified above for $k = 1$. The goal of the induction step is to show that either $e \leq e_-$, $e \leq h_k$, or $e \leq \kappa h_k$. Assume that $e > e_-$ and $e > h_k$, then we may apply (3.5) and (3.7) to conclude

$$e \leq \kappa h_k / 2 + \frac{\bar{\phi}e^2}{(c_0 - \nu)h_k} \leq \kappa h_k / 2 + \frac{\bar{\phi}\mu^{-1}\kappa}{c_0 - \nu} e \leq \kappa h_k / 2 + e/2.$$

Hence, either $e \leq e_-$, $e \leq h_k$, or $e \leq \kappa h_k$ and the induction is complete. This completes the proof of the forward part of the result with $K = \max(\kappa, 1)$.

To prove the converse, we note that if $h \geq e_- \geq e$ then

$$\|x - x(\alpha, h, \hat{f})\| \leq L e_- + c_1 h + \bar{\phi} e_- + \nu h \leq \bar{\tau} h$$

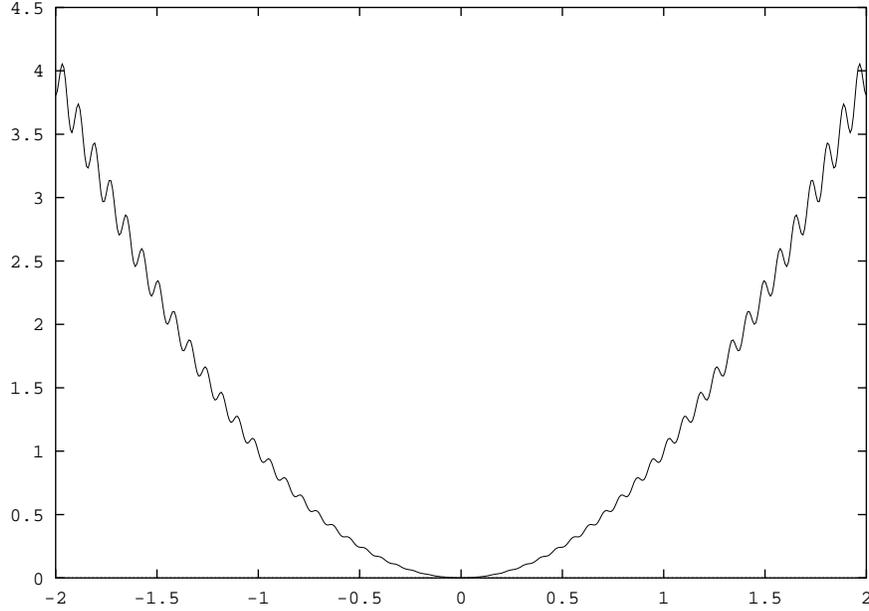
if $\bar{\tau} \geq \bar{\tau}_0 = L + c_1 + \bar{\phi} + \nu$. \square

It is important in the proof of Theorem 3.1 that h_0 be large enough to begin the induction. Heuristically, an initial scale that is too small could lead to entrapment in a local minimum and that possibility is eliminated by (3.2). The same failure could be caused by selection of a value of $\bar{\tau}$ that is too small. The perturbation itself must be small, which is the role of assumption (3.7). Left unresolved is the issue of whether the outer iteration or the line search in **projgrad** will terminate. A deeper examination of the relation of the termination criteria, *i. e.* $\bar{\tau}$, and the line search, *i. e.* the parameter σ , to the size of ϕ will be done in § 4 and § 5.

The parameter $\bar{\tau}_0$ is so large that many scales will be rejected, that is Algorithm **projgrad** will terminate on entry, before the iteration begins to make progress. The search for a good heuristic for the choice of $\bar{\tau}$ is an open problem.

Note that the accuracy of the finite difference plays no direct role in the ultimate accuracy of the iteration because of the presence of the $\nu \max(e, h)$ term in (3.3). To see how this term arises, assume that there is C such that

$$\phi(x) \leq C \|\phi\|_\infty \|x - x^*\|^2.$$

FIG. 3.1. $\hat{f} = f + \phi, e_- = 0$ 

For forward differences we have

$$\|\nabla_h \phi(x)\| \leq 2C\sqrt{N}\|\phi\|_\infty(e+h)^2/h$$

leading to the estimates

$$\bar{\phi} \leq 2C\sqrt{N}\|\phi\|_\infty \text{ and } \nu \leq 6C\sqrt{N}\|\phi\|_\infty.$$

Using second order centered differences would reduce the estimates of $\bar{\phi}$ and ν by factors of two but not eliminate ν entirely. Perhaps because of this reduction of $\bar{\phi}$ and ν , centered differences were found to reduce the sensitivity of the algorithm on the size of ϕ in [17], [16], [19], and [20].

As an example of the type of function we consider we take $f = x^2$ and $\phi = \epsilon x^2 \cos(80x)$ on the interval $[-2, 2]$. Here $e_- = 0$. We take $\bar{\tau} = 2$, $\mu = 1/2$, and apply forward differencing. Using the notation in the proof we see that

$$c_0 = 2, c_1 = 1, \bar{\phi} = 2\epsilon, \text{ and } \nu = 6\epsilon.$$

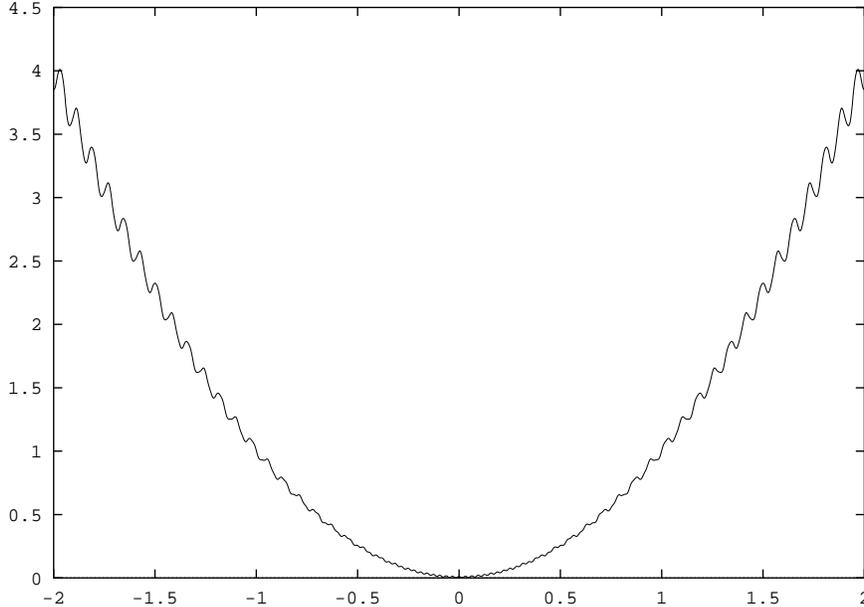
Hence $\kappa = 6/(2 - 6\epsilon)$ Hence (3.7) holds if

$$24\epsilon/(2 - 6\epsilon)^2 \leq 1/2$$

which holds for any $\epsilon \leq 1/20$. As $D_\Omega = 4$, setting $\epsilon = 1/20$ and $h_0 = 2$ will satisfy the assumptions of Theorem 3.1. A plot of \hat{f} with $\epsilon = 1/20$ is in figure 3.1.

In the applications it was rare that $e_- = 0$. Typically the noise in \hat{f} decayed near the minimum, but not to zero. In this case the minimum could only be resolved to a

FIG. 3.2. $\hat{f}_2 = f + \phi_2, e_- \neq 0$



level of size proportional to the square root of the minimum noise. As an example of such a function consider $\hat{f}_2 = x^2(x) + \phi_2(x)$ where $\Omega = [-2, 2]$ and

$$\phi_2(x) = .75x^2 \cos(80x)/20 + .25 * \cos(100x)^2/20.$$

\hat{f}_2 satisfies the assumptions of Theorem 3.1 with $e_- = .25/20$. We plot e_- in figure 3.2. The perturbations in these examples appear small, but clearly result in substantial local minima.

The proof of Theorem 3.1 also provides insight into how Algorithm 2.2 is intended to function. For functions satisfying the hypothesis of Theorem 3.1, application of `projgrad`($\hat{f}, x, h_i, \tau, \bar{\alpha}$) return x_i with $\|x_i - x^*\| \leq K \max(e_-, h_i)$. This should reduce the size of ϕ for the next iterate and allows for the reduction of h .

4. Convergence Results. In this section we apply a technical result proved in § 5 to give conditions under which Algorithm `imfilter` will converge to a minimum at all scales for objectives that satisfy Assumption 3.1. We then give some simple examples to illustrate the behavior of the algorithm and show how quasi-Newton methods can improve the performance of the algorithm.

In § 5 we prove the following result.

THEOREM 4.1. *Let Assumption 3.1 hold and assume that $M_\phi \leq 1$. Let $\beta \leq 1/2$ and $\sigma \in (0, 1/4)$ be given. Let $\tau(h) = \bar{\tau}h$. Then there are $\bar{\tau}_1$ and $\eta > 0$ so that if $\bar{\tau} \geq \bar{\tau}_1$, $M_\phi^{1/3}/\bar{\tau}^2 < \eta$, $\|x - x^*\| \geq e_-$, and $h \geq M_\phi^{1/3}\|x - x^*\|$, then Algorithm `projgrad`($\hat{f}, x, h, \tau, \bar{\alpha}, \sigma$) will terminate successfully and return x such that $\|x - x(1, h, \hat{f})\| \leq \bar{\tau}h$. Moreover, either x is unchanged or \hat{f} is reduced by at least $\sigma \bar{\alpha} \bar{\tau}^2 h^2$.*

From the technical result Theorem 4.1 we can directly obtain conditions that insure that Algorithm **imfilter** will terminate successfully with a minimum at all scales. We consider sequences of scales of the form $h_k = \mu^k h_0$ with $\mu \in (0, 1)$.

THEOREM 4.2. *Assume that Assumption 3.1 holds, that $M_\phi \leq 1$, that $M_\phi^{1/3}/\bar{\tau}^2 < \eta$, and that $\bar{\tau} > \bar{\tau}_0$. Let $\mu \in (0, 1)$ be such that $M_\phi^{1/3}K \leq \mu$. Then if $h_0 \geq D_\Omega/2$, $h_k = \mu^k h_0$, and $h_m \geq e_-$, then Algorithm **imfilter**, will terminate and return a minimum at all scales.*

Proof. Either $\|x - x^*\| \leq e_-$ and we terminate with a minimum at all scales by Theorem 3.1 or by Theorem 4.1, Algorithm **projgrad**($\hat{f}, x, h_0, \tau, \bar{\alpha}, \sigma$) will terminate with $\|x - x(1, h_0, \hat{f})\| \leq \bar{\tau}h_0$. We may then conclude from Theorem 3.1 that

$$e = \|x - x^*\| \leq Kh_0.$$

If we require that $M_\phi^{1/3}K \leq \mu$ then either $e < e_-$ or

$$\mu h_0 \geq M_\phi^{1/3}Kh_0 \geq M_\phi^{1/3}e,$$

and we may apply Algorithm **projgrad**($\hat{f}, x, h, \tau, \bar{\alpha}, \sigma$) with $h = \mu h_0$.

We may continue this, replacing h by μh as the iteration progresses. Eventually the iteration will terminate at a minimum at all scales since the reduction in h implies a reduction in the bound for e , and any x such that $e \leq e_-$ is a minimum at all scales.

□

Note that the restart feature of Algorithm **allscale** was not used in Theorem 4.2. It was not needed in the work reported in [17], [16], [19], and [20] either. A corollary of Theorem 4.2 is that if **projgrad** fails, as it can if there are too many step size reductions, then $e \leq e_-$. Hence monitoring the Armijo rule is a way to detect the limit for the scales.

COROLLARY 4.3. *Let the assumption of Theorem 4.2 hold. If Algorithm **imfilter** is initiated as in the statement of that result and μh_m is the first scale for which Algorithm **projgrad** fails, then $K\mu h_m \geq e_-$.*

While Corollary 4.3 does not guarantee that $Kh_m \geq e_- \geq K\mu h_m$, we have used it with success in the applications to determine when to terminate the iteration when no *a priori* knowledge of the size of ϕ was available.

We applied algorithm **imfilter** to the function

$$\hat{f}(x) = 2x^2 + x^2 \cos(80x)/6$$

on the interval $[-2, 2]$. We used $x_0 = -1.75$, $h_0 = 2$, and $h_i = 2^{-i}h_0$ for $i = 1, \dots, 12$. We used $\bar{\tau} = 2$. In Table 4.1 we tabulate h , the iteration counter i_p for Algorithm **projgrad** for that h , the number of stepsize reductions i_A at that iteration, x , $\nu(x) = |x - \mathcal{P}(x - \nabla_h \hat{f}(x))|$, and $\hat{f}(x)$. The last entry for each value of h corresponds to the terminal iterate for Algorithm **projgrad**. After evaluation of \hat{f} at the initial iterate, the cost in function evaluations is like that for the projected gradient method. For each value of h and i_p one function evaluation is performed to compute $\nabla_h \hat{f}$, $\hat{f}(x)$ being provided by the termination at the previous value of h or i_p . i_A counts the number of additional function evaluations for each value of i_p required to obtain sufficient decrease in the Armijo rule. For Table 4.1 a total of 39 function evaluations are performed. An interesting feature of the table is how the iterates can remain unchanged as the scale is reduced. It is normal behavior of algorithm **imfilter** to

pass through more than one scale with the termination criterion satisfied on entry before finally taking a step and changing the iterate. The iteration terminated at a minimum at all scales and hence a full application of Algorithm `allscale` would terminate as well.

We also applied Algorithm `imfilter` to

$$\hat{f}(x) = 2x^2 + .75x^2 \cos(80x)/6 + .25 \cos(100x)/6.$$

On the interval $[-2, 2]$, $x_0 = -1.75$, $h_0 = 2$, and $h_i = 2^{-i}h_0$ for $i = 1, \dots, 12$, and $\bar{\tau} = 2$. Here $e_- \approx .25/6$. Since $e_- \neq 0$ we might expect Algorithm `projgrad`, and hence Algorithm `imfilter`, to fail when $e < e_-$. This happens as we can see from the large number of stepsize reductions in the latter phases of the iteration reported in Table 4.2. The version of Algorithm `imfilter` used in the applications would have terminated the iteration at when $h = .5$ and $i_p = 2$ (since $\bar{\alpha} = 2^{-10}$), finding a minimum at all scales for the scales $\{h_i\}_{i=0}^6$.

The example in Table 4.2 illustrates the heuristic we use to estimate e_- , *i. e.* the point where further reductions in h give no advantage. When the Armijo rule in Algorithm `projgrad` fails, we conclude that the nature of the problem has changed and that a minimum scale has been found. While this heuristic is certainly far from a theorem, and examples can easily be constructed for which it fails, we found it to be very useful in the applications reported in [17], [16], [19], and [20].

We close this section with remarks on enhancements that improve the performance of Algorithm `allscale` in practice. First of all, for the work reported in [17], [16], [19], and [20] experience showed that it was not necessary to call `imfilter` more than once, hence the loop in Algorithm `allscale` that tests the invariance of `imfilter`, which is necessary for the theoretical results, was not used in the practical results reported in [17], [16], [19], and [20]. Convergence in the final phases of the iteration was improved by using a projected SR1 [3] iteration such as that proposed in [6]. The SR1 update performed somewhat better than the BFGS update in the preliminary experiments for our work in [17], [16], [19], and [20]. This is consistent with other reports [7], [11], [10], and we used it in the computations reported in [17], [16], [19], and [20]. We point out that for problems of moderate size, a projected Newton formulation [2] would be equally desirable if the Hessian could be computed accurately and cheaply, but that was certainly not the case for the problems considered in [17], [16], [19], and [20] where function evaluations were quite expensive making Hessian evaluation too expensive.

To illustrate the benefits of incorporation of a quasi-Newton update into Algorithm `projgrad` apply a secant update to the examples tabulated in Table 4.1 and Table 4.2. 28 function evaluations were required for this computation, an improvement over the 29 for the results in Table 4.1.

One could also use a different forward difference step for each component of $\nabla_h f$. Our implementation in [17], [16], [19], and [20] did this by scaling the feasible set to the unit cube in R^N . We let the largest scale $h_0 = .5$ be half the diameter of the cube and let $h_i = .5h_{i-1}$. Selection of the smallest scale is problem dependent and was determined in [17], [16], [19], and [20] by physical estimates of the error in the objective function.

TABLE 4.1
One sweep of implicit filtering. $\epsilon_- = 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	-1.750e+0	2.947e+0	6.024e+0
1	0	1	-1.750e+0	3.750e+0	6.024e+0
1	1	2	7.442e-1	2.744e+0	1.016e+0
1	2	0	-5.564e-1	2.895e-1	6.637e-1
5.000e-1	0	1	-5.564e-1	1.315e+0	6.637e-1
5.000e-1	1	3	1.011e-1	1.336e+0	2.004e-2
5.000e-1	2	0	-6.595e-2	6.735e-1	9.087e-3
2.500e-1	0	0	-6.595e-2	2.221e-1	9.087e-3
1.250e-1	0	0	-6.595e-2	1.685e-2	9.087e-3
6.250e-2	0	1	-6.595e-2	1.450e-1	9.087e-3
6.250e-2	1	4	6.536e-3	1.603e-1	9.161e-5
6.250e-2	2	0	-3.480e-3	1.111e-1	2.616e-5
3.125e-2	0	0	-3.480e-3	4.603e-2	2.616e-5
1.562e-2	0	0	-3.480e-3	1.809e-2	2.616e-5
7.812e-3	0	0	-3.480e-3	1.834e-3	2.616e-5
3.906e-3	0	0	-3.480e-3	6.596e-3	2.616e-5
1.953e-3	0	1	-3.480e-3	1.081e-2	2.616e-5
1.953e-3	1	2	1.925e-3	1.252e-2	8.020e-6
1.953e-3	2	0	-1.204e-3	9.849e-4	3.140e-6
9.766e-4	0	1	-1.204e-3	3.100e-3	3.140e-6
9.766e-4	1	3	3.461e-4	3.614e-3	2.595e-7
9.766e-4	2	0	-1.056e-4	1.658e-3	2.418e-8

TABLE 4.2
One sweep of implicit filtering. $\epsilon_- \neq 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	-1.750e+0	2.948e+0	6.064e+0
1	0	1	-1.750e+0	3.750e+0	6.064e+0
1	1	2	7.354e-1	2.735e+0	1.041e+0
1	2	0	-5.934e-1	3.195e-1	6.996e-1
5.000e-1	0	1	-5.934e-1	1.281e+0	6.996e-1
5.000e-1	1	13	4.701e-2	1.268e+0	4.200e-3
5.000e-1	2	17	4.686e-2	1.267e+0	4.195e-3
5.000e-1	3	0	4.686e-2	1.267e+0	4.195e-3

TABLE 4.3
One sweep of secant method implicit filtering. $\epsilon_- = 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	-1.750e+0	2.947e+0	6.024e+0
1	0	1	-1.750e+0	3.750e+0	6.024e+0
1	1	0	7.442e-1	2.744e+0	1.016e+0
1	2	0	-5.291e-1	7.538e-2	5.559e-1
5.000e-1	0	1	-5.291e-1	1.109e+0	5.559e-1
5.000e-1	1	3	2.520e-2	1.065e+0	1.224e-3
5.000e-1	2	0	-8.757e-3	9.626e-1	1.631e-4
2.500e-1	0	0	-8.757e-3	4.999e-1	1.631e-4
1.250e-1	0	0	-8.757e-3	1.970e-1	1.631e-4
6.250e-2	0	0	-8.757e-3	8.672e-2	1.631e-4
3.125e-2	0	0	-8.757e-3	2.655e-2	1.631e-4
1.562e-2	0	0	-8.757e-3	3.974e-3	1.631e-4
7.812e-3	0	1	-8.757e-3	2.063e-2	1.631e-4
7.812e-3	1	1	1.560e-3	2.319e-2	5.272e-6
7.812e-3	2	0	-1.169e-3	1.173e-2	2.961e-6
3.906e-3	0	0	-1.169e-3	3.390e-3	2.961e-6
1.953e-3	0	0	-1.169e-3	8.344e-4	2.961e-6
9.766e-4	0	1	-1.169e-3	2.950e-3	2.961e-6
9.766e-4	1	1	3.056e-4	3.439e-3	2.024e-7
9.766e-4	2	0	-9.131e-5	1.720e-3	1.806e-8

TABLE 4.4
One sweep of secant method implicit filtering. $\epsilon_- \neq 0$.

h	i_p	i_A	x	$\nu(x)$	$f(x)$
2	0	0	-1.750e+0	2.948e+0	6.064e+0
1	0	1	-1.750e+0	3.750e+0	6.064e+0
1	1	0	7.354e-1	2.735e+0	1.041e+0
1	2	0	-5.489e-1	2.269e-1	6.405e-1
5.000e-1	0	1	-5.489e-1	1.269e+0	6.405e-1
5.000e-1	1	1	8.569e-2	1.238e+0	3.338e-2
5.000e-1	2	0	-7.101e-2	6.495e-1	3.008e-2
2.500e-1	0	0	-7.101e-2	1.893e-1	3.008e-2

5. Proof of Theorem 4.1. Throughout this section we will assume that the assumptions of Theorem 4.1 hold. As in section 1 $\nabla_h \hat{f}(x)$ represent a forward, backward, or centered difference approximation to $\nabla f(x)$. In this section, since the context will be clear, we abbreviate $x(\alpha, h, \hat{f})$ by $x(\alpha)$.

We will require several lemmas.

LEMMA 5.1. *Let Assumption 3.1 hold. Let $h \geq M_\phi^{1/3} \|x - x^*\|$. Then there is M_1 such that*

$$(5.1) \quad \|\nabla_h \phi\| \leq M_1 M_\phi^{1/3} h.$$

Proof. As in the proof of Theorem 3.1,

$$\|\nabla_h \phi(x)\| \leq 2M_\phi \sqrt{N} (c + h)^2 / h$$

and hence, for forward differences,

$$\|\nabla_h \phi(x)\| \leq 2M_\phi^{1/3} \sqrt{N} (M_\phi^{1/3} + 1)^2 h.$$

This completes the proof with $M_1 = 8\sqrt{N}$. \square

LEMMA 5.2. *Let Assumption 3.1 hold. Let $h \geq M_\phi^{1/3} \|x - x^*\|$. Then there is M_2 such that*

$$(5.2) \quad \|\nabla_h \hat{f}(x) - \nabla f(x)\| \leq M_2 h.$$

Proof. Equation (5.2) is a direct consequence of Lemma 5.1 and Assumption 3.1 with $M_2 = c_1 + (c_0 + M_1)M_\phi^{1/3}$. \square

The next lemma, which we give without proof, is a direct consequence of the projection theorem, [4].

LEMMA 5.3. *Let $x \in \Omega$, $0 < \alpha \leq 1$, and $x(\alpha) = x(\alpha, h, \hat{f})$. Then*

1. $\|x - x(\alpha)\| \leq \alpha \|\nabla_h \hat{f}(x)\|$.
2. for any $i \in \{1, \dots, n\}$, if $x^i(1) = x^i - \nabla_h \hat{f}(x)$ then $x^i(\alpha) = x^i - \alpha \nabla_h \hat{f}(x)$.
3. If $\|x - x(1)\| \geq \bar{\tau}h$ then $\|x - x(\alpha)\| \geq \alpha \bar{\tau}h$.
4. $\alpha \nabla_h \hat{f}(x)^T (x - x(\alpha)) \geq \|x - x(\alpha)\|^2$.

Next we will prove a lemma that specifies an interval for the step size for which the criteria for sufficient decrease given in the description of Algorithm **projgrad** is always satisfied.

LEMMA 5.4. *Let Assumption 3.1 hold and assume that $M_\phi \leq 1$. Let $\beta \leq 1/2$, $\bar{\alpha} \leq 1/(2L + 8)$, and $\sigma \in (0, 1/4)$ be given. Let $\tau(h) = \bar{\tau}h$. Then there are $\eta > 0$ so that if $M_\phi^{1/3}/\bar{\tau}^2 < \eta$, $\|x - x^*\| \geq e_-$, and $h \geq M_\phi^{1/3} \|x - x^*\|$, then if $\|x - x(1)\| \geq \bar{\tau}h$ then the generalized Armijo step size rule*

$$(5.3) \quad \hat{f}(x) - \hat{f}(x(\alpha)) \geq \sigma \frac{\|x_k - x_k(\alpha)\|^2}{\alpha}$$

is satisfied for all α with

$$(5.4) \quad \frac{1}{2L + 8M_\phi} \leq \alpha \leq \frac{3}{2L + 8M_\phi}.$$

Hence there is m such that (5.3) is satisfied with $\alpha = \beta_m$.

Proof. Let $\delta(\alpha) = x - x(\alpha)$ and assume that $\bar{\tau} \geq \bar{\tau}_0$, so that the converse part of Theorem 3.1 holds. Therefore $\epsilon > \epsilon_-$.

Using the definition of $\hat{f}(x)$, and the fundamental theorem of calculus we obtain,

$$\begin{aligned}
 \hat{f}(x) - \hat{f}(x(\alpha)) &= f(x) - f(x(\alpha)) + \phi(x) - \phi(x(\alpha)) \\
 &= \delta(\alpha)^T \int_0^1 \nabla f(x - t\delta(\alpha)) dt + \phi(x) - \phi(x(\alpha)) \\
 (5.5) \quad &= \delta(\alpha)^T \left(\nabla f(x) + \int_0^1 (\nabla f(x - t\delta(\alpha)) - \nabla f(x)) dt \right) \\
 &\quad + \phi(x) - \phi(x(\alpha)) \\
 &\geq \delta(\alpha)^T \nabla f(x) - \frac{L\|x-x(\alpha)\|^2}{2} + \phi(x) - \phi(x(\alpha))
 \end{aligned}$$

Now assume that $\|x - x(1)\| \geq \bar{\tau}h$, and hence $\epsilon \geq \epsilon_-$. We estimate the parts of the right hand side of (5.5) in turn. First, by Lemma 5.2 and Lemma 5.3,

$$\delta(\alpha)^T \nabla f(x) \geq \frac{\|\delta(\alpha)\|^2}{\alpha} - \frac{M_2\|\delta(\alpha)\|^2}{\alpha\bar{\tau}} \geq \left(1 - \frac{M_2}{\bar{\tau}}\right) \frac{\|\delta(\alpha)\|^2}{\alpha}.$$

We set $\bar{\tau}_1 = \max(\bar{\tau}_0, 4M_2)$ and require $\bar{\tau} \geq \bar{\tau}_1$ to obtain

$$(5.6) \quad \delta(\alpha)^T \nabla f(x) \geq \frac{3\|\delta(\alpha)\|^2}{4\alpha}$$

Using Assumption 3.1 we have, as we assume $\epsilon \geq \epsilon_-$,

$$|\phi(x(\alpha))| \leq M_\phi \|x(\alpha) - x^*\|^2 \leq 2M_\phi (\|\delta(\alpha)\|^2 + \epsilon^2).$$

By the estimates

$$M_\phi^{1/3} \epsilon \leq h \leq \|\delta(\alpha)\| / (\alpha\bar{\tau})$$

and Lemma 5.3 we have

$$\begin{aligned}
 |\phi(x(\alpha))| &\leq 2M_\phi (\|\delta(\alpha)\|^2 + M_\phi^{-2/3} h^2) \\
 &\leq \frac{\|\delta(\alpha)\|^2}{\alpha} \left(2\alpha M_\phi + \frac{2M_\phi^{1/3}}{\alpha\bar{\tau}^2} \right).
 \end{aligned}$$

Since

$$|\phi(x)| \leq M_\phi \epsilon^2 \leq M_\phi^{1/3} h^2 \leq \frac{\|\delta(\alpha)\|^2}{\alpha} \frac{M_\phi^{1/3}}{\alpha\bar{\tau}^2}$$

We have, since $\alpha \leq 1$,

$$(5.7) \quad |\phi(x) - \phi(x(\alpha))| \leq \frac{\|\delta(\alpha)\|^2}{\alpha} \left(2M_\phi + \frac{3M_\phi^{1/3}}{\alpha\bar{\tau}^2} \right).$$

Using (5.6) and (5.7) in (5.5) yields

$$(5.8) \quad \hat{f}(x) - \hat{f}(x(\alpha)) \geq \frac{\|\delta(\alpha)\|^2}{\alpha} (3/4 - D_0/\alpha - D_2\alpha)$$

where

$$D_0 = \frac{3M_\phi^{1/3}}{\bar{\tau}^2} \text{ and } D_2 = 2M_\phi + L/2$$

If $M_\phi^{1/3}/\bar{\tau}^2$ is small enough so that

$$(5.9) \quad D_0 D_2 \leq 1/64$$

then

$$-D_2 \alpha^2 + \alpha/2 - D_0 \geq 0,$$

for all $\alpha \in [1/(4D_0), 3/(4D_0)]$. Hence $3/4 - D_0/\alpha - D_2\alpha \geq \sigma$ holds for all $\sigma \in (0, 1/4)$ and all $\alpha \in [1/(4D_0), 3/(4D_0)]$. This completes the proof. \square

To complete the proof of Theorem 4.1 we note that Lemma 5.4 implies that if $\|x - x(1)\| > \bar{\tau}h$, then the line search in Algorithm `projgrad`($f, x, h, \tau, \bar{\alpha}, \sigma$) will return $x(\alpha)$ and, using Lemma 5.3,

$$\hat{f}(x) - \hat{f}(x(\alpha)) \geq \sigma \frac{\|x - x(\alpha)\|^2}{\alpha} \geq \sigma \frac{\alpha^2 \bar{\tau}^2 h^2}{\alpha} \geq \sigma \bar{\alpha} \bar{\tau}^2 h^2.$$

Boundedness of Ω and continuity of f then imply that Algorithm `projgrad` must terminate successfully.

Acknowledgments. The authors would like to thank Griff Bilbro, Bob Trew, and Tom Winslow of the Department of Electrical and Computer Engineering at North Carolina State University and Dan Stoneking of M/A-COM Corporation for their input to the development of the algorithm. We also appreciate Winslow's creation of Figure 1.1.

REFERENCES

- [1] D. B. BERTSEKAS, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Autom. Control, (1976), pp. 174–184.
- [2] ———, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.
- [3] C. G. BROYDEN, *Quasi-Newton methods and their application to function minimization*, Math. Comp., 21 (1967), pp. 368–381.
- [4] P. CIARLET, *Introduction to numerical linear algebra and optimization*, Cambridge University Press, New York, NY, 1988.
- [5] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Global convergence of a class of trust region algorithms for optimization problems with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.
- [6] ———, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.
- [7] ———, *Convergence of quasi-Newton matrices generated by the symmetric rank one update*, Math. Programming A, 50 (1991), pp. 177–195.
- [8] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, Prentice-Hall, Englewood Cliffs, N.J., 1983.
- [9] J. E. DENNIS AND V. TORCZON, *Direct search methods on parallel machines*, SIAM J. Optimization, 1 (1991), pp. 448 – 474.
- [10] C. T. KELLEY, E. W. SACHS, AND B. WATSON, *A pointwise quasi-Newton method for unconstrained optimal control problems, II*, J. Optim. Theory Appl., 71 (1991), pp. 535–547.
- [11] H. KHALFAN, R. H. BYRD, AND R. B. SCHNABEL, *A theoretical and experimental study of the symmetric rank one update*, Tech. Report CU-CS-489-90, University of Colorado at Boulder, December 1990.

- [12] J. KOSTROWICKI AND L. PIELA, *Diffusion equation method of global minimization: Performance for standard test functions*, *J. Optim. Theory Appl.*, (1991), pp. 269–284.
- [13] J. KREIMER AND R. Y. RUBINSTEIN, *Smoothed functionals for optimization problems*, *SIAM J. Numer. Anal.*, (1988), pp. 470–487.
- [14] J. ORTEGA AND W. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
- [15] T. STEihaug, *The conjugate gradient method and trust regions in large scale optimization*, *SIAM J. Numer. Anal.*, 20 (1983), pp. 626–637.
- [16] D. STONEKING, G. BILBRO, R. TREW, P. GILMORE, AND C. T. KELLEY, *Yield optimization using a GaAs process simulator coupled to a physical device model*, in *Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits*, IEEE, 1991, pp. 374–383.
- [17] ———, *Yield optimization using a GaAs process simulator coupled to a physical device model*, *IEEE Transactions on Microwave Theory and Techniques*, 40 (1992), pp. 1353–1363.
- [18] V. TORCZSON, *On the convergence of the multidimensional direct search*, *SIAM J. Optimization*, 1 (1991), pp. 123–145.
- [19] T. A. WINSLOW, R. J. TREW, P. GILMORE, AND C. T. KELLEY, *Doping profiles for optimum class B performance of GaAs mesfet amplifiers*, in *Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits*, IEEE, 1991, pp. 188–197.
- [20] ———, *Simulated performance optimization of GaAs MESFET amplifiers*, in *Proceedings IEEE/Cornell Conference on Advanced Concepts in High Speed Devices and Circuits*, IEEE, 1991, pp. 393–402.