SIAM J. OPTIM.
Vol. 16, No. 1, pp. 170–192

A NEW CONJUGATE GRADIENT METHOD WITH GUARANTEED DESCENT AND AN EFFICIENT LINE SEARCH*

WILLIAM W. HAGER† AND HONGCHAO ZHANG†

Abstract. A new nonlinear conjugate gradient method and an associated implementation, based on an inexact line search, are proposed and analyzed. With exact line search, our method reduces to a nonlinear version of the Hestenes–Stiefel conjugate gradient scheme. For any (inexact) line search, our scheme satisfies the descent condition $\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le -\frac{7}{8}\|\mathbf{g}_k\|^2$. Moreover, a global convergence result is established when the line search fulfills the Wolfe conditions. A new line search scheme is developed that is efficient and highly accurate. Efficiency is achieved by exploiting properties of linear interpolants in a neighborhood of a local minimizer. High accuracy is achieved by using a convergence criterion, which we call the "approximate Wolfe" conditions, obtained by replacing the sufficient decrease criterion in the Wolfe conditions with an approximation that can be evaluated with greater precision in a neighborhood of a local minimum than the usual sufficient decrease criterion. Numerical comparisons are given with both L-BFGS and conjugate gradient methods using the unconstrained optimization problems in the CUTE library.

Key words. conjugate gradient method, unconstrained optimization, convergence, line search, Wolfe conditions

AMS subject classifications. 90C06, 90C26, 65Y20

DOI. 10.1137/030601880

1. Introduction.

We develop a new nonlinear conjugate gradient algorithm for the unconstrained optimization problem

$$\min\{f(\mathbf{x}) : \mathbf{x} \in \Re^n\}, \tag{1.1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable. The iterates $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$, ... satisfy the recurrence

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k,$$

where the stepsize α_k is positive and the directions $\mathbf{d}_k$ are generated by the rule

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k^N \mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, \tag{1.2}$$

$$\beta_k^N = \frac{1}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k}\left(\mathbf{y}_k - 2\mathbf{d}_k\frac{\|\mathbf{y}_k\|^2}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k}\right)^{\mathsf{T}}\mathbf{g}_{k+1}. \tag{1.3}$$

Here $\|\cdot\|$ is the Euclidean norm, $\mathbf{g}_k = \nabla f(\mathbf{x}_k)^{\mathsf{T}}$, and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$; the gradient $\nabla f(\mathbf{x}_k)$ of f at $\mathbf{x}_k$ is a row vector and $\mathbf{g}_k$ is a column vector. If f is a quadratic and α_k is chosen to achieve the exact minimum of f in the direction $\mathbf{d}_k$, then $\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_{k+1} = 0$, and the formula (1.3) for β_k^N reduces to the Hestenes–Stiefel scheme [22]. In this paper, however, we consider general nonlinear functions and an inexact line search.

As explained in our survey paper [19], the nonlinear conjugate gradient scheme developed and analyzed in this paper is one member of a one-parameter family of conjugate gradient methods with guaranteed descent. Different choices for the parameter

*Received by the editors November 18, 2003; accepted for publication December 10, 2004; published electronically September 8, 2005. This research was supported by National Science Foundation grant 0203270.

http://www.siam.org/journals/siopt/16-1/60188.html

†Department of Mathematics, University of Florida, Gainesville, FL 32611-8105 (hager@math.ufl.edu, http://www.math.ufl.edu/∼hager; hzhang@math.ufl.edu, http://www.math.ufl.edu/∼hzhang).

correspond to differences in the relative importance of conjugacy versus descent. The specific scheme analyzed in this paper is closely connected with the memoryless quasi-Newton scheme of Perry [30] and Shanno [36]. In particular, the scheme (1.2)–(1.3) can be obtained by deleting a term in the Perry–Shanno scheme. If $\mathbf{d}_{k+1}$ is the direction generated by the new scheme (1.2)–(1.3), then the direction $\mathbf{d}_{k+1}^{PS}$ of the Perry–Shanno scheme can be expressed as

$$\mathbf{d}_{k+1}^{PS} = \frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{s}_k}{\|\mathbf{y}_k\|^2}\left(\mathbf{d}_{k+1} + \frac{\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_{k+1}}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k}\mathbf{y}_k\right), \tag{1.4}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. We observe in section 2 that the $\mathbf{d}_{k+1}$ term in (1.4) dominates the $\mathbf{y}_k$ term to the right when the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$ is sufficiently small and f is strongly convex. In this case, the directions generated by the new scheme are approximate multiples of $\mathbf{d}_{k+1}^{PS}$. The Perry–Shanno scheme, analyzed further in [34, 37, 39], has global convergence for convex functions and for an inexact line search [36], but in general, it does not necessarily converge, even when the line search is exact [33]. Of course, the Perry–Shanno scheme is convergent if restarts are employed; however, the speed of convergence can decrease. Han, Liu, and Yin [21] proved that if a standard Wolfe line search is employed, then convergence to a stationary point is achieved when $\lim_{k\to\infty}\|\mathbf{y}_k\|_2 = 0$ and the gradient of f is Lipschitz continuous.

Although we are able to prove a global convergence result for (1.2)–(1.3) when f is strongly convex, our analysis breaks down for a general nonlinear function since β_k^N can be negative. Similar to the approach [13, 20, 38] taken for the Polak–Ribière–Polyak [31, 32] version of the conjugate gradient method, we establish convergence for general nonlinear functions by restricting the lower value of β_k^N. Although restricting β_k^N to be nonnegative ensures convergence, the resulting iterates may differ significantly from those of (1.2)–(1.3), and convergence speed may be reduced, especially when f is quadratic. In our restricted scheme, we dynamically adjust the lower bound on β_k^N in order to make the lower bound smaller as the iterates converge:

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \bar{\beta}_k^N\mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, \tag{1.5}$$

$$\bar{\beta}_k^N = \max\left\{\beta_k^N, \eta_k\right\}, \quad \eta_k = \frac{-1}{\|\mathbf{d}_k\|\min\{\eta, \|\mathbf{g}_k\|\}}, \tag{1.6}$$

where $\eta > 0$ is a constant; we took $\eta = .01$ in the experiments of section 5.

For this modified scheme, we prove a global convergence result with inexact line search. When $\|\mathbf{g}_k\|$ tends to zero as k grows, it follows that η_k in (1.6) tends to $-\infty$ as k grows when $\mathbf{d}_k$ is bounded. Moreover, for strongly convex functions, we show that $\mathbf{d}_k$ is bounded. In this case, where $\mathbf{d}_k$ is bounded, the scheme (1.5)–(1.6) is essentially the scheme (1.2)–(1.3) when k is large since η_k tends to $-\infty$.

Another method related to (1.2)–(1.3) is the Dai–Liao version [7] of the conjugate gradient method, in which β_k^N in (1.2) is replaced with

$$\beta_k^{DL} = \frac{1}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k}(\mathbf{y}_k - t\mathbf{s}_k)^{\mathsf{T}}\mathbf{g}_{k+1}, \tag{1.7}$$

where $t > 0$ is a constant parameter. Numerical results are reported in [7] for $t = 0.1$ and $t = 1$; for different choices of t, the numerical results are quite different. The method (1.2)–(1.3) can be viewed as an adaptive version of (1.7) corresponding to $t = 2\|\mathbf{y}_k\|^2/\mathbf{s}_k^{\mathsf{T}}\mathbf{y}_k$.

With conjugate gradient methods, the line search typically requires sufficient accuracy to ensure that the search directions yield descent [6, 16]. Moreover, it has been shown [9] that for the Fletcher–Reeves [12] and Polak–Ribière–Polyak [31, 32] conjugate gradient methods, a line search that satisfies the strong Wolfe conditions may not yield a direction of descent for a suitable choice of the Wolfe line search parameters, even for the function $f(\mathbf{x}) = \lambda\|\mathbf{x}\|^2$, where $\lambda > 0$ is a constant. An attractive feature of the new conjugate gradient scheme, which we now establish, is that the search directions always yield descent when $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$, a condition which is satisfied when f is strongly convex, or the line search satisfies the Wolfe conditions.

THEOREM 1.1. *If* $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$ *and*

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \tau\mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, \tag{1.8}$$

for any $\tau \in [\beta_k^N, \max\{\beta_k^N, 0\}]$, *then*

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} \leq -\frac{7}{8}\|\mathbf{g}_{k+1}\|^2. \tag{1.9}$$

Proof. Since $\mathbf{d}_0 = -\mathbf{g}_0$, we have $\mathbf{g}_0^{\mathsf{T}}\mathbf{d}_0 = -\|\mathbf{g}_0\|^2$, which satisfies (1.9). Suppose $\tau = \beta_k^N$. Multiplying (1.8) by $\mathbf{g}_{k+1}^{\mathsf{T}}$, we have

$$\begin{aligned}
\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} &= -\|\mathbf{g}_{k+1}\|^2 + \beta_k^N \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \\
&= -\|\mathbf{g}_{k+1}\|^2 + \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \left(\frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k} - 2\frac{\|\mathbf{y}_k\|^2 \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k}{(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2} \right) \\
&= \frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k) - \|\mathbf{g}_{k+1}\|^2(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2 - 2\|\mathbf{y}_k\|^2(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k)^2}{(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2}.
\end{aligned} \tag{1.10}$$

We apply the inequality

$$\mathbf{u}^{\mathsf{T}}\mathbf{v} \leq \frac{1}{2}(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2)$$

to the first term in (1.10) with

$$\mathbf{u} = \frac{1}{2}(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)\mathbf{g}_{k+1} \quad \text{and} \quad \mathbf{v} = 2(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k)\mathbf{y}_k$$

to obtain (1.9). On the other hand, if $\tau \neq \beta_k^N$, then $\beta_k^N \leq \tau \leq 0$. After multiplying (1.8) by $\mathbf{g}_{k+1}^{\mathsf{T}}$, we have

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + \tau\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k.$$

If $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \geq 0$, then (1.9) follows immediately since $\tau \leq 0$. If $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k < 0$, then

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + \tau\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \leq -\|\mathbf{g}_{k+1}\|^2 + \beta_k^N \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k$$

since $\beta_k^N \leq \tau \leq 0$. Hence, (1.9) follows by our previous analysis. ∎

By taking $\tau = \beta_k^N$, we see that the directions generated by (1.2)–(1.3) are descent directions when $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$. Since η_k in (1.6) is negative, it follows that

$$\bar{\beta}_k^N = \max\left\{\beta_k^N, \eta_k\right\} \in [\beta_k^N, \max\{\beta_k^N, 0\}].$$

Hence, the direction given by (1.5) and (1.6) is a descent direction. Dai and Yuan [8, 10] present conjugate gradient schemes with the property that $\mathbf{d}_{k+1}^{\mathsf{T}}\mathbf{g}_{k+1} < 0$ when $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k > 0$. If f is strongly convex or the line search satisfies the Wolfe conditions, then $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k > 0$ and the Dai–Yuan schemes yield descent. Note that in (1.9) we bound $\mathbf{d}_{k+1}^{\mathsf{T}}\mathbf{g}_{k+1}$ by $-(7/8)||\mathbf{g}_{k+1}||^2$, while for the schemes [8, 10], the negativity of $\mathbf{d}_{k+1}^{\mathsf{T}}\mathbf{g}_{k+1}$ is established.

Our paper is organized as follows: In section 2 we prove convergence of (1.2)–(1.3) for strongly convex functions, while in section 3 we prove convergence of (1.5)–(1.6) for more general nonlinear functions. In section 4 we develop a new line search that is both efficient and highly accurate. This line search exploits properties of linear interpolants to achieve rapid convergence of the line search. High accuracy is achieved by replacing the sufficient decrease criterion in the Wolfe conditions with an approximation that can be evaluated with greater precision in a neighborhood of a local minimum. In section 5 we compare the Dolan–Moré [11] performance profile of the new conjugate gradient scheme to the profiles for the L-BFGS (limited memory Broyden–Fletcher–Goldfarb–Shanno) quasi-Newton method [25, 28], the Polak–Ribière–Polyak PRP+ method [13], and the Dai–Yuan schemes [8, 10] using the unconstrained problems in the test problem library CUTE (constrained and unconstrained testing environment) [4].

2. Convergence analysis for strongly convex functions. Although the search directions generated by either (1.2)–(1.3) or (1.5)–(1.6) are always descent directions, we need to constrain the choice of α_k to ensure convergence. We consider line searches that satisfy either the Goldstein conditions [14],

$$\delta_1\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le f(\mathbf{x}_k+\alpha_k\mathbf{d}_k)-f(\mathbf{x}_k) \le \delta_2\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k, \tag{2.1}$$

where $0 < \delta_2 < \frac{1}{2} < \delta_1 < 1$ and $\alpha_k > 0$, or the Wolfe conditions [40, 41],

$$f(\mathbf{x}_k+\alpha_k\mathbf{d}_k)-f(\mathbf{x}_k) \le \delta\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k, \tag{2.2}$$

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \ge \sigma\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k, \tag{2.3}$$

where $0 < \delta \le \sigma < 1$. As in [8], we do not require the "strong Wolfe" condition $|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k| \le -\sigma\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k$, which is often used to prove convergence of nonlinear conjugate gradient methods.

LEMMA 2.1. *Suppose that* $\mathbf{d}_k$ *is a descent direction and* ∇f *satisfies the Lipschitz condition*

$$\|\nabla f(\mathbf{x})-\nabla f(\mathbf{x}_k)\| \le L\|\mathbf{x}-\mathbf{x}_k\|$$

for all $\mathbf{x}$ *on the line segment connecting* $\mathbf{x}_k$ *and* $\mathbf{x}_{k+1}$*, where* L *is a constant. If the line search satisfies the Goldstein conditions, then*

$$\alpha_k \ge \frac{(1-\delta_1)}{L}\frac{|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}. \tag{2.4}$$

If the line search satisfies the Wolfe conditions, then

$$\alpha_k \ge \frac{1-\sigma}{L}\frac{|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}. \tag{2.5}$$

Proof. For the convenience of the reader, we include a proof of these well-known results. If the Goldstein conditions hold, then by (2.1) and the mean value theorem,

we have

$$\begin{aligned}\delta_1\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k &\le f(\mathbf{x}_k+\alpha_k\mathbf{d}_k)-f(\mathbf{x}_k)\\ &= \alpha_k\nabla f(\mathbf{x}_k+\xi\mathbf{d}_k)\mathbf{d}_k\\ &\le \alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k+L\alpha_k^2\|\mathbf{d}_k\|^2,\end{aligned}$$

where $\xi\in[0,\alpha_k]$. Rearranging this inequality gives (2.4).

Subtracting $\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k$ from both sides of (2.3) using the Lipschitz condition gives

$$(\sigma-1)\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le (\mathbf{g}_{k+1}-\mathbf{g}_k)^{\mathsf{T}}\mathbf{d}_k \le \alpha_k L\|\mathbf{d}_k\|^2.$$

Since $\mathbf{d}_k$ is a descent direction and $\sigma<1$, (2.5) follows immediately. ∎

We now prove convergence of the unrestricted scheme (1.2)–(1.3) for the case when f is strongly convex.

THEOREM 2.2. *Suppose that f is strongly convex and Lipschitz continuous on the level set*

$$\mathcal{L}=\{\mathbf{x}\in\mathbb{R}^n : f(\mathbf{x})\le f(\mathbf{x}_0)\}. \tag{2.6}$$

That is, there exist constants L and $\mu>0$ such that

$$\begin{aligned}&\|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})\|\le L\|\mathbf{x}-\mathbf{y}\| \text{ and}\\ &\mu\|\mathbf{x}-\mathbf{y}\|^2\le(\nabla f(\mathbf{x})-\nabla f(\mathbf{y}))(\mathbf{x}-\mathbf{y})\end{aligned} \tag{2.7}$$

for all $\mathbf{x}$ and $\mathbf{y}\in\mathcal{L}$. If the conjugate gradient method (1.2)–(1.3) is implemented using a line search that satisfies either the Wolfe or the Goldstein conditions in each step, then either $\mathbf{g}_k=\mathbf{0}$ for some k, or

$$\lim_{k\to\infty}\ \mathbf{g}_k=\mathbf{0}. \tag{2.8}$$

Proof. Suppose that $\mathbf{g}_k\ne\mathbf{0}$ for all k. By the strong convexity assumption,

$$\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k=(\mathbf{g}_{k+1}-\mathbf{g}_k)^{\mathsf{T}}\mathbf{d}_k\ge\mu\alpha_k\|\mathbf{d}_k\|^2. \tag{2.9}$$

Theorem 1.1 and the assumption $\mathbf{g}_k\ne\mathbf{0}$ imply that $\mathbf{d}_k\ne\mathbf{0}$. Since $\alpha_k>0$, it follows from (2.9) that $\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k>0$. Since f is strongly convex over $\mathcal{L}$, f is bounded from below. After summing over k the upper bound in either (2.1) or (2.2), we conclude that

$$\sum_{k=0}^{\infty}\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k>-\infty.$$

Combining this with the lower bound for α_k given in Lemma 2.1 and the descent property (1.9) gives

$$\sum_{k=0}^{\infty}\frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2}<\infty. \tag{2.10}$$

By Lipschitz continuity (2.7),

$$\|\mathbf{y}_k\|=\|\mathbf{g}_{k+1}-\mathbf{g}_k\|=\|\nabla f(\mathbf{x}_k+\alpha_k\mathbf{d}_k)-\nabla f(\mathbf{x}_k)\|\le L\alpha_k\|\mathbf{d}_k\|. \tag{2.11}$$

Utilizing (2.9) and (1.3), we have

$$\begin{aligned}|\beta_k^N| &= \left|\frac{\mathbf{y}_k^\mathsf{T}\mathbf{g}_{k+1}}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k} - 2\frac{\|\mathbf{y}_k\|^2\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}}{(\mathbf{d}_k^\mathsf{T}\mathbf{y}_k)^2}\right| \\ &\le \frac{\|\mathbf{y}_k\|\|\mathbf{g}_{k+1}\|}{\mu\alpha_k\|\mathbf{d}_k\|^2} + 2\frac{\|\mathbf{y}_k\|^2\|\mathbf{d}_k\|\|\mathbf{g}_{k+1}\|}{\mu^2\alpha_k^2\|\mathbf{d}_k\|^4} \\ &\le \frac{L\alpha_k\|\mathbf{d}_k\|\|\mathbf{g}_{k+1}\|}{\mu\alpha_k\|\mathbf{d}_k\|^2} + 2\frac{L^2\alpha_k^2\|\mathbf{d}_k\|^3\|\mathbf{g}_{k+1}\|}{\mu^2\alpha_k^2\|\mathbf{d}_k\|^4} \\ &\le \left(\frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)\frac{\|\mathbf{g}_{k+1}\|}{\|\mathbf{d}_k\|}. \qquad (2.12)\end{aligned}$$

Hence, we have

$$\|\mathbf{d}_{k+1}\| \le \|\mathbf{g}_{k+1}\| + |\beta_k^N|\|\mathbf{d}_k\| \le \left(1 + \frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)\|\mathbf{g}_{k+1}\|.$$

Inserting this upper bound for $\mathbf{d}_k$ in (2.10) yields

$$\sum_{k=1}^{\infty}\|\mathbf{g}_k\|^2 < \infty,$$

which completes the proof. □

We now observe that the directions generated by the new conjugate gradient update (1.2) point approximately in the Perry–Shanno direction (1.4) when f is strongly convex and the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$ is sufficiently small. By (2.9) and (2.11), we have

$$\frac{|\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}|}{|\mathbf{d}_k^\mathsf{T}\mathbf{y}_k|}\|\mathbf{y}_k\| \le \frac{L}{\mu}|\mathbf{u}_k^\mathsf{T}\mathbf{g}_{k+1}| = c_1\epsilon\|\mathbf{g}_{k+1}\|, \qquad (2.13)$$

where $\mathbf{u}_k = \mathbf{d}_k/\|\mathbf{d}_k\|$ is the unit vector in the direction $\mathbf{d}_k$, ϵ is the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$, and $c_1 = L/\mu$. By the definition of $\mathbf{d}_{k+1}$ in (1.2), we have

$$\|\mathbf{d}_{k+1}\|^2 \ge \|\mathbf{g}_{k+1}\|^2 - 2\beta_k^N\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}. \qquad (2.14)$$

By the bound for β_k^N in (2.12),

$$|\beta_k^N\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}| \le c_2|\mathbf{u}_k^\mathsf{T}\mathbf{g}_{k+1}|\|\mathbf{g}_{k+1}\| = c_2\epsilon\|\mathbf{g}_{k+1}\|^2, \qquad (2.15)$$

where c_2 is the constant appearing in (2.12). Combining (2.14) and (2.15), we have

$$\|\mathbf{d}_{k+1}\| \ge \sqrt{1 - 2c_2\epsilon}\|\mathbf{g}_{k+1}\|.$$

This lower bound for $\|\mathbf{d}_{k+1}\|$ and the upper bound (2.13) for the $\mathbf{y}_k$ term in (1.4) imply that the ratio between them is bounded by $c_1\epsilon/\sqrt{1 - 2c_2\epsilon}$. As a result, when ϵ is small, the direction generated by (1.2) is approximately a multiple of the Perry–Shanno direction (1.4).

3. Convergence analysis for general nonlinear functions. Our analysis of (1.5)–(1.6) for general nonlinear functions exploits insights developed by Gilbert and Nocedal in their analysis [13] of the PRP+ scheme. Similar to the approach taken in [13], we establish a bound for the change $\mathbf{u}_{k+1} - \mathbf{u}_k$ in the normalized direction $\mathbf{u}_k = \mathbf{d}_k/\|\mathbf{d}_k\|$, which we use to conclude, by contradiction, that the gradients cannot be bounded away from zero. The following theorem is the analogue of [13, Lem. 4.1]; it differs in the treatment of the direction update formula (1.5).

LEMMA 3.1. *If the level set* (2.6) *is bounded and the Lipschitz condition* (2.7) *holds, then for the scheme* (1.5)–(1.6) *and a line search that satisfies the Wolfe conditions* (2.2)–(2.3)*, we have*

$$\mathbf{d}_k \neq \mathbf{0} \quad \textit{for each } k \textit{ and} \quad \sum_{k=0}^{\infty} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2 < \infty$$

whenever $\inf\{\|\mathbf{g}_k\| : k \geq 0\} > 0$.

Proof. Define $\gamma = \inf\{\|\mathbf{g}_k\| : k \geq 0\}$. Since $\gamma > 0$ by assumption, it follows from the descent property, Theorem 1.1, that $\mathbf{d}_k \neq \mathbf{0}$ for each k. Since $\mathcal{L}$ is bounded, f is bounded from below, and by (2.2) and (2.5), the following Zoutendijk condition [42] holds:

$$\sum_{k=0}^{\infty} \frac{(\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < \infty.$$

Again, the descent property yields

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|\mathbf{d}_k\|^2} \leq \sum_{k=0}^{\infty} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} \leq \frac{64}{49} \sum_{k=0}^{\infty} \frac{(\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < \infty. \tag{3.1}$$

Define the quantities

$$\beta_k^+ = \max\{\bar{\beta}_k^N, 0\}, \quad \beta_k^- = \min\{\bar{\beta}_k^N, 0\}, \quad \mathbf{r}_k = \frac{-\mathbf{g}_k + \beta_{k-1}^- \mathbf{d}_{k-1}}{\|\mathbf{d}_k\|}, \quad \delta_k = \beta_{k-1}^+ \frac{\|\mathbf{d}_{k-1}\|}{\|\mathbf{d}_k\|}.$$

By (1.5)–(1.6), we have

$$\mathbf{u}_k = \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|} = \frac{-\mathbf{g}_k + (\beta_{k-1}^+ + \beta_{k-1}^-)\mathbf{d}_{k-1}}{\|\mathbf{d}_k\|} = \mathbf{r}_k + \delta_k \mathbf{u}_{k-1}.$$

Since the $\mathbf{u}_k$ are unit vectors,

$$\|\mathbf{r}_k\| = \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| = \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\|.$$

Since $\delta_k > 0$, it follows that

$$\begin{aligned} \|\mathbf{u}_k - \mathbf{u}_{k-1}\| &\leq \|(1+\delta_k)(\mathbf{u}_k - \mathbf{u}_{k-1})\| \\ &\leq \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| + \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\| \\ &= 2\|\mathbf{r}_k\|. \end{aligned} \tag{3.2}$$

By the definition of β_k^- and the fact that $\eta_k < 0$ and $\bar{\beta}_k^N \geq \eta_k$ in (1.6), we have the following bound for the numerator of $\mathbf{r}_k$:

$$\begin{aligned} \| - \mathbf{g}_k + \beta_{k-1}^- \mathbf{d}_{k-1}\| &\le \|\mathbf{g}_k\| - \min\{\bar\beta_{k-1}^N, 0\}\|\mathbf{d}_{k-1}\| \\ &\le \|\mathbf{g}_k\| - \eta_{k-1}\|\mathbf{d}_{k-1}\| \\ &\le \|\mathbf{g}_k\| + \frac{1}{\|\mathbf{d}_{k-1}\|\min\{\eta,\gamma\}}\|\mathbf{d}_{k-1}\| \\ &\le \Gamma + \frac{1}{\min\{\eta,\gamma\}}, \end{aligned} \tag{3.3}$$

where

$$\Gamma = \max_{\mathbf{x}\in\mathcal{L}} \|\nabla f(\mathbf{x})\|. \tag{3.4}$$

Let c denote the expression $\Gamma + 1/\min\{\eta,\gamma\}$ in (3.3). This bound for the numerator of $\mathbf{r}_k$ coupled with (3.2) gives

$$\|\mathbf{u}_k - \mathbf{u}_{k-1}\| \le 2\|\mathbf{r}_k\| \le \frac{2c}{\|\mathbf{d}_k\|}. \tag{3.5}$$

Finally, by squaring (3.5), summing over k, and utilizing (3.1), we complete the proof. ∎

THEOREM 3.2. *If the level set* (2.6) *is bounded and the Lipschitz condition* (2.7) *holds, then for the scheme* (1.5)–(1.6) *and a line search that satisfies the Wolfe conditions* (2.2)–(2.3)*, either* $\mathbf{g}_k = \mathbf{0}$ *for some* k*, or*

$$\liminf_{k\to\infty} \|\mathbf{g}_k\| = 0. \tag{3.6}$$

Proof. We suppose that both $\mathbf{g}_k \neq \mathbf{0}$ for all k and $\liminf_{k\to\infty}\|\mathbf{g}_k\| > 0$. In the following, we obtain a contradiction. Defining $\gamma = \inf\{\|\mathbf{g}_k\| : k \ge 0\}$, we have $\gamma > 0$ due to (3.6) and the fact that $\mathbf{g}_k \neq \mathbf{0}$ for all k. The proof is divided into the following three steps:

I. *A bound for* $\bar\beta_k^N$. By the Wolfe condition $\mathbf{g}_{k+1}^\mathsf{T}\mathbf{d}_k \ge \sigma\mathbf{g}_k^\mathsf{T}\mathbf{d}_k$, we have

$$\mathbf{y}_k^\mathsf{T}\mathbf{d}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)^\mathsf{T}\mathbf{d}_k \ge (\sigma - 1)\mathbf{g}_k^\mathsf{T}\mathbf{d}_k = -(1-\sigma)\mathbf{g}_k^\mathsf{T}\mathbf{d}_k. \tag{3.7}$$

By Theorem 1.1,

$$-\mathbf{g}_k^\mathsf{T}\mathbf{d}_k \ge \frac{7}{8}\|\mathbf{g}_k\|^2 \ge \frac{7}{8}\gamma^2.$$

Combining this with (3.7) gives

$$\mathbf{y}_k^\mathsf{T}\mathbf{d}_k \ge (1-\sigma)\frac{7}{8}\gamma^2. \tag{3.8}$$

Also, observe that

$$\mathbf{g}_{k+1}^\mathsf{T}\mathbf{d}_k = \mathbf{y}_k^\mathsf{T}\mathbf{d}_k + \mathbf{g}_k^\mathsf{T}\mathbf{d}_k < \mathbf{y}_k^\mathsf{T}\mathbf{d}_k. \tag{3.9}$$

Again, the Wolfe condition gives

$$\mathbf{g}_{k+1}^\mathsf{T}\mathbf{d}_k \ge \sigma\mathbf{g}_k^\mathsf{T}\mathbf{d}_k = -\sigma\mathbf{y}_k^\mathsf{T}\mathbf{d}_k + \sigma\mathbf{g}_{k+1}^\mathsf{T}\mathbf{d}_k. \tag{3.10}$$

Since $\sigma < 1$, we can rearrange (3.10) to obtain

$$\mathbf{g}_{k+1}^\mathsf{T}\mathbf{d}_k \ge \frac{-\sigma}{1-\sigma}\mathbf{y}_k^\mathsf{T}\mathbf{d}_k.$$

Combining this lower bound for $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k$ with the upper bound (3.9) yields

$$\left|\frac{\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k}{\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k}\right| \le \max\left\{\frac{\sigma}{1-\sigma}, 1\right\}. \tag{3.11}$$

By the definition of $\bar{\beta}_k^N$ in (1.6), we have

$$\bar{\beta}_k^N = \beta_k^N \text{ if } \beta_k^N \ge 0 \quad \text{and} \quad 0 \ge \bar{\beta}_k^N \ge \beta_k^N \text{ if } \beta_k^N < 0.$$

Hence, $|\bar{\beta}_k^N| \le |\beta_k^N|$ for each k. We now insert the upper bound (3.11) for $|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k|/|\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k|$, the lower bound (3.8) for $\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k$, and the Lipschitz estimate (2.11) for $\mathbf{y}_k$ into the expression (1.3) to obtain

$$\begin{aligned}
|\bar{\beta}_k^N| &\le |\beta_k^N| \\
&\le \frac{1}{|\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k|}\left(|\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}| + 2\|\mathbf{y}_k\|^2\frac{|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k|}{|\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k|}\right) \\
&\le \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma\|\mathbf{s}_k\| + 2L^2\|\mathbf{s}_k\|^2\max\left\{\frac{\sigma}{1-\sigma}, 1\right\}\right) \\
&\le C\|\mathbf{s}_k\|, \qquad (3.12)
\end{aligned}$$

where Γ is defined in (3.4), and where C is defined as follows:

$$C = \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma + 2L^2 D\max\left\{\frac{\sigma}{1-\sigma}, 1\right\}\right), \tag{3.13}$$

$$D = \max\{\|\mathbf{y}-\mathbf{z}\| : \mathbf{y}, \mathbf{z} \in \mathcal{L}\}. \tag{3.14}$$

Here D is the diameter of $\mathcal{L}$.

II. *A bound on the steps* $\mathbf{s}_k$. This is a modified version of [13, Thm. 4.3]. Observe that for any $l \ge k$,

$$\mathbf{x}_l - \mathbf{x}_k = \sum_{j=k}^{l-1}\mathbf{x}_{j+1} - \mathbf{x}_j = \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\mathbf{u}_j = \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\mathbf{u}_k + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|(\mathbf{u}_j - \mathbf{u}_k).$$

By the triangle inequality,

$$\sum_{j=k}^{l-1}\|\mathbf{s}_j\| \le \|\mathbf{x}_l - \mathbf{x}_k\| + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\|\mathbf{u}_j - \mathbf{u}_k\| \le D + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\|\mathbf{u}_j - \mathbf{u}_k\|. \tag{3.15}$$

Let Δ be a positive integer, chosen large enough that

$$\Delta \ge 4CD, \tag{3.16}$$

where C and D appear in (3.13) and (3.14). Choose k_0 large enough that

$$\sum_{i\ge k_0}\|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2 \le \frac{1}{4\Delta}. \tag{3.17}$$

By Lemma 3.1, k_0 can be chosen in this way. If $j > k \geq k_0$ and $j - k \leq \Delta$, then by (3.17) and the Cauchy–Schwarz inequality, we have

$$\begin{aligned}\|\mathbf{u}_j - \mathbf{u}_k\| &\leq \sum_{i=k}^{j-1} \|\mathbf{u}_{i+1} - \mathbf{u}_i\| \\ &\leq \sqrt{j-k}\left(\sum_{i=k}^{j-1} \|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2\right)^{1/2} \\ &\leq \sqrt{\Delta}\left(\frac{1}{4\Delta}\right)^{1/2} = \frac{1}{2}.\end{aligned}$$

Combining this with (3.15) yields

$$\sum_{j=k}^{l-1} \|\mathbf{s}_j\| \leq 2D, \tag{3.18}$$

when $l > k \geq k_0$ and $l - k \leq \Delta$.

III. *A bound on the directions* $\mathbf{d}_l$. By (1.5) and the bound on $\bar{\beta}_k^N$ given in step I, we have

$$\|\mathbf{d}_l\|^2 \leq (\|\mathbf{g}_l\| + |\bar{\beta}_{l-1}^N|\|\mathbf{d}_{l-1}\|)^2 \leq 2\Gamma^2 + 2C^2\|\mathbf{s}_{l-1}\|^2\|\mathbf{d}_{l-1}\|^2,$$

where Γ is the bound on the gradient given in (3.4). Defining $S_i = 2C^2\|\mathbf{s}_i\|^2$, we conclude that for $l > k_0$,

$$\|\mathbf{d}_l\|^2 \leq 2\Gamma^2\left(\sum_{i=k_0+1}^{l}\prod_{j=i}^{l-1} S_j\right) + \|\mathbf{d}_{k_0}\|^2\prod_{j=k_0}^{l-1} S_j. \tag{3.19}$$

Above, the product is defined to be 1 whenever the index range is vacuous. Let us consider as follows a product of Δ consecutive S_j, where $k \geq k_0$:

$$\begin{aligned}\prod_{j=k}^{k+\Delta-1} S_j = \prod_{j=k}^{k+\Delta-1} 2C^2\|\mathbf{s}_j\|^2 &= \left(\prod_{j=k}^{k+\Delta-1} \sqrt{2}C\|\mathbf{s}_j\|\right)^2 \\ &\leq \left(\frac{\sum_{j=k}^{k+\Delta-1} \sqrt{2}C\|\mathbf{s}_j\|}{\Delta}\right)^{2\Delta} \leq \left(\frac{2\sqrt{2}CD}{\Delta}\right)^{2\Delta} \leq \frac{1}{2^\Delta}.\end{aligned}$$

The first inequality above is the arithmetic-geometric mean inequality, the second is due to (3.18), and the third comes from (3.16). Since the product of Δ consecutive S_j is bounded by $1/2^\Delta$, it follows that the sum in (3.19) is bounded, and the bound is independent of l. This bound for $\|\mathbf{d}_l\|$, independent of $l > k_0$, contradicts (3.1). Hence, $\gamma = \liminf_{k\to\infty} \|\mathbf{g}_k\| = 0$. □

4. Line search. The line search is an important factor in the overall efficiency of any optimization algorithm. Papers focusing on the development of efficient line search algorithms include [1, 2, 16, 24, 26, 27]. The algorithm [27] of Moré and Thuente is used widely; it is incorporated in the L-BFGS limited memory quasi-Newton code of Nocedal and in the PRP+ conjugate gradient code of Liu, Nocedal, and Waltz.

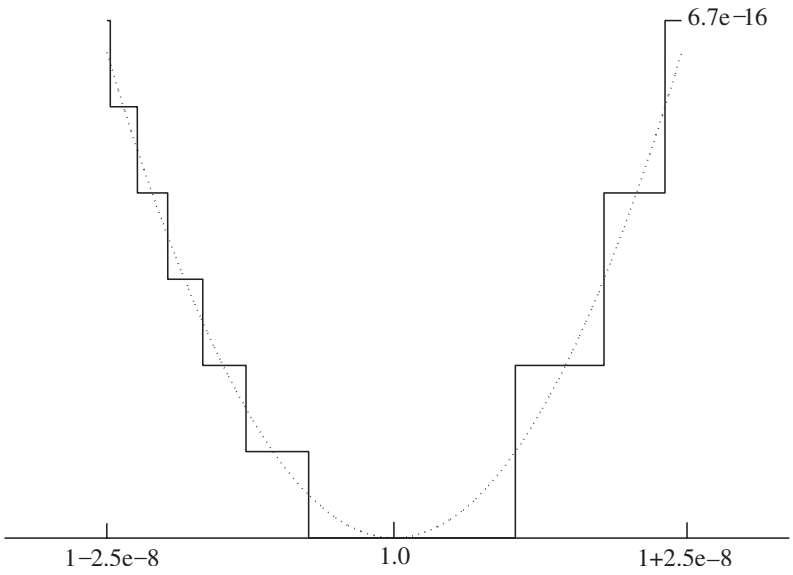

FIG. 4.1. *Numerical and exact graphs of* $F(x) = 1 - 2x + x^2$ *near* $x = 1$.

The approach we use to find a point satisfying the Wolfe conditions (2.2)–(2.3) is somewhat different from the earlier work cited. To begin, we note that there is a fundamental numerical issue connected with the first Wolfe condition, (2.2). In Figure 4.1 we plot $F(x) = 1 - 2x + x^2$ in a neighborhood of $x = 1$.

The graph, generated by a MATLAB program using a Sun workstation, is obtained by evaluating F at 10,000 values of x between $1 - 2.5 \times 10^{-8}$ and $1 + 2.5 \times 10^{-8}$ and by connecting the computed points on the graph by straight line segments. The true graph is the parabola in Figure 4.1, while the computed graph is piecewise constant.

When devising an algorithm to minimize a smooth function, we often visualize the graph as smooth. But, in actuality, the computer's representation of the function is piecewise constant. Observe that there is an interval of width 1.8×10^{-8} surrounding $x = 1$, where F vanishes. Each point in this interval is a minimizer of the computer's F. In contrast, the true F has a unique minimum at $x = 1$. The interval around $x = 1$, where F is flat, is much wider than the machine epsilon 2.2×10^{-16}. This relatively large flat region is a result of subtracting nearly equal numbers when F is evaluated. In particular, near $x = 1$, $1 - 2x$ is near -1, while x^2 is near $+1$. Hence, when the computer adds $1 - 2x$ to x^2, it is, in essence, subtracting nearly equal numbers. It is well known that there is a large relative error when nearly equal numbers are subtracted; the width of the flat interval near $x = 1$ is on the order of the square root of the machine epsilon (see [15]).

Now consider the function $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$. If $\phi(0)$ corresponds to a point in the flat part of Figure 4.1 near $x = 1$, then the first Wolfe condition, (2.2), is never satisfied, assuming $\mathbf{d}_k$ is a descent direction, since the right side of (2.2) is always negative and the left side can be only nonnegative. On the other hand, when we compute with 16 significant digits, we would like to be able to compute a solution to the optimization problem with 16-digit accuracy. We can achieve this accuracy by looking for a zero of the derivative. In Figure 4.2 we plot the derivative $F'(x) =$

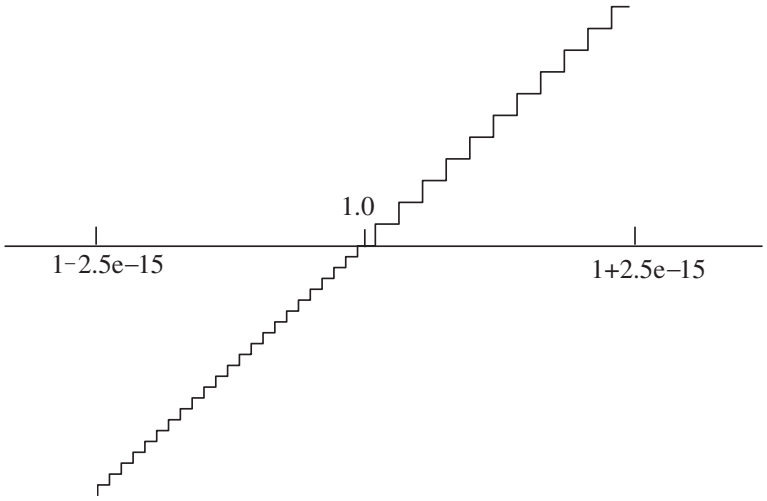

FIG. 4.2. *The numerical graph of the derivative* $F'(x) = 2(x-1)$ *near* $x = 1$.

$2(x-1)$ of the function in Figure 4.1 near $x = 1$. Since the interval where F' vanishes has width 1.6×10^{-16}, we can locate the zero of F' (Figure 4.2) with accuracy on the order of the machine epsilon 2.2×10^{-16}, while the minimum of F in Figure 4.1 is determined with accuracy on the order of the square root of the machine epsilon. Figures 4.1 and 4.2 are extracted from [17].

This leads us to introduce the *approximate Wolfe conditions*,

$$(2\delta - 1)\phi'(0) \geq \phi'(\alpha_k) \geq \sigma\phi'(0), \tag{4.1}$$

where $\delta < \min\{.5, \sigma\}$. The second inequality in (4.1) is identical to the second Wolfe condition, (2.3). The first inequality in (4.1) is identical to the first Wolfe condition, (2.2), when f is quadratic. For general f, we now show that the first inequality in (4.1) and the first Wolfe condition agree to the order of α_k^2. The interpolating (quadratic) polynomial q that matches $\phi(\alpha)$ at $\alpha = 0$, and $\phi'(\alpha)$ at $\alpha = 0$ and $\alpha = \alpha_k$, is

$$q(\alpha) = \frac{\phi'(\alpha_k) - \phi'(0)}{2\alpha_k}\alpha^2 + \phi'(0)\alpha + \phi(0).$$

For such an interpolating polynomial, $|q(\alpha) - \phi(\alpha)| = O(\alpha^3)$. After replacing ϕ with q in the first Wolfe condition, we obtain the first inequality in (4.1) (with an error term of order α_k^2). We emphasize that this first inequality is an approximation to the first Wolfe condition. On the other hand, this approximation can be evaluated with greater precision than the original condition when the iterates are near a local minimizer, since the approximate Wolfe conditions are expressed in terms of a derivative, not the difference of function values.

With these insights, we terminate the line search when either of the following conditions holds:

T1. The original Wolfe conditions (2.2)–(2.3) are satisfied.

T2. The approximate Wolfe conditions (4.1) are satisfied and

$$\phi(\alpha_k) \le \phi(0) + \epsilon_k, \tag{4.2}$$

where $\epsilon_k \ge 0$ is an estimate for the error in the value of f at iteration k. For the experiments in section 5, we took

$$\epsilon_k = \epsilon |f(\mathbf{x}_k)|, \tag{4.3}$$

where ϵ is a (small) fixed parameter. We would like to satisfy the original Wolfe conditions, so we terminate the line search whenever they are satisfied. On the other hand, when $\mathbf{x}_{k+1}$ and $\mathbf{x}_k$ are close together, numerical errors may make it impossible to satisfy (2.2). If the function value at $\alpha = \alpha_k$ is not much larger than the function value at $\alpha = 0$, then we view the iterates as close together, and we terminate when the approximate Wolfe conditions are satisfied.

We satisfy the termination criterion by constructing a nested sequence of (bracketing) intervals, which converge to a point satisfying either T1 or T2. A typical interval $[a, b]$ in the nested sequence satisfies the following *opposite slope condition*:

$$\phi(a) \le \phi(0) + \epsilon_k, \quad \phi'(a) < 0, \quad \phi'(b) \ge 0. \tag{4.4}$$

Given a parameter $\theta \in (0, 1)$, the *interval update rules* are specified in the following procedure "interval update." The input of this procedure is the current bracketing interval $[a, b]$ and a point c generated by either a secant step or a bisection step, as will be explained shortly. The output of the procedure is the updated bracketing interval $[\bar{a}, \bar{b}]$.

INTERVAL UPDATE. $[\bar{a}, \bar{b}] = update\ (a, b, c)$.

U0. If $c \notin (a, b)$, then $\bar{a} = a$, $\bar{b} = b$, and return.

U1. If $\phi'(c) \ge 0$, then $\bar{a} = a$, $\bar{b} = c$, and return.

U2. If $\phi'(c) < 0$ and $\phi(c) \le \phi(0) + \epsilon_k$, then $\bar{a} = c$, $\bar{b} = b$, and return.

U3. If $\phi'(c) < 0$ and $\phi(c) > \phi(0) + \epsilon_k$, then set $\hat{a} = a$, $\hat{b} = c$, and do the following:

- a. Set $d = (1 - \theta)\hat{a} + \theta\hat{b}$; if $\phi'(d) \ge 0$, then set $\bar{b} = d$, $\bar{a} = \hat{a}$, and return.
- b. If $\phi'(d) < 0$ and $\phi(d) \le \phi(0) + \epsilon_k$, then set $\hat{a} = d$ and go to step a.
- c. If $\phi'(d) < 0$ and $\phi(d) > \phi(0) + \epsilon_k$, then set $\hat{b} = d$ and go to step a.

After completing U1–U3, we obtain a new interval $[\bar{a}, \bar{b}] \subset [a, b]$ whose endpoints satisfy (4.4). The loop embedded in U3a–c should terminate since the interval width $\hat{b} - \hat{a}$ tends to zero and, at $\hat{a}$ and $\hat{b}$, the following conditions hold:

$$\phi'(\hat{a}) < 0, \quad \phi(\hat{a}) \le \phi(0) + \epsilon_k,$$
$$\phi'(\hat{b}) < 0, \quad \phi(\hat{b}) > \phi(0) + \epsilon_k.$$

The input c for the update routine is generated by polynomial interpolation. The interpolation is done in a special way to ensure that the line search interval shrinks quickly. In Figure 4.3, where ϕ' is concave, an initial secant step using function values at a and b yields a point $\bar{b}$ to the right of the zero. A second secant step using function values at $\bar{b}$ and b yields a point $\bar{a}$ to the left of the zero. On the other hand, if ϕ' is convex as shown in Figure 4.4, then an initial secant step using function values at a and b yields a point $\bar{a}$ to the left of the zero. A second secant step using function values at a and $\bar{a}$ yields a point $\bar{b}$ to the right of the zero. Hence, whether ϕ' is convex or concave, a pair of secant steps, implemented in this way, will update one side of the interval, bracketing the zero, and then the other side.

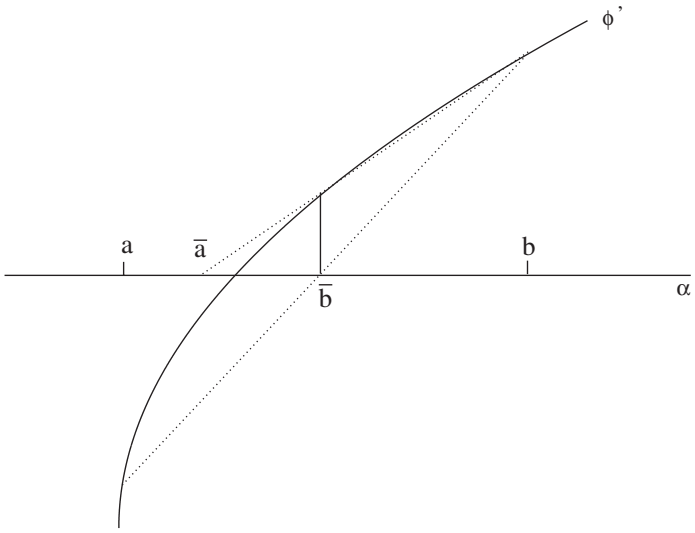

FIG. 4.3. *A pair* of secant *steps applied to a concave* ϕ'.

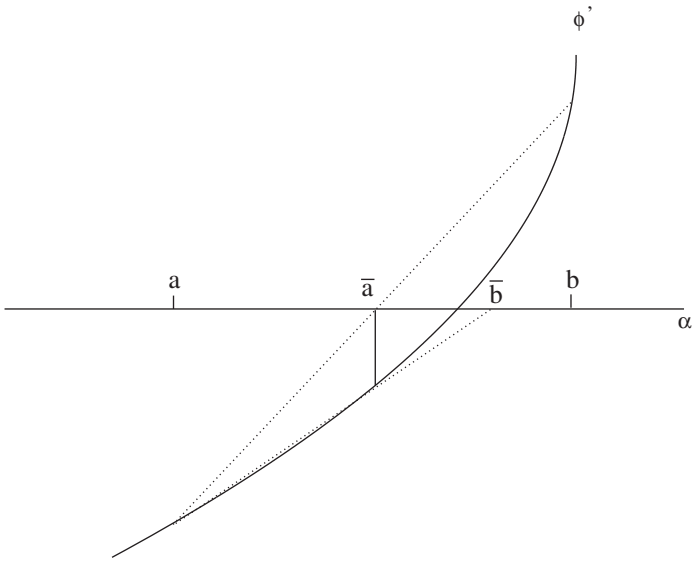

FIG. 4.4. *A pair* of secant *steps applied to a convex* ϕ'.

If c is obtained from a secant step based on function values at a and b, then we write

$$c = \text{ secant } (a,b) = \frac{a\phi'(b) - b\phi'(a)}{\phi'(b) - \phi'(a)}.$$

In general, we do not know whether ϕ' is convex or concave. Consequently, the pair of secant steps is generated by a routine denoted secant^2 defined in the following way.

DOUBLE SECANT STEP. $[\bar{a}, \bar{b}] = \text{secant}^2\ (a, b)$.
S1. $c =$ secant (a, b) and $[A, B] =$ update (a, b, c).
S2. If $c = B$, then $\bar{c} =$ secant (b, B).
S3. If $c = A$, then $\bar{c} =$ secant (a, A).
S4. If $c = A$ or $c = B$, then $[\bar{a}, \bar{b}] =$ update $(A, B, \bar{c})$. Otherwise, $[\bar{a}, \bar{b}] = [A, B]$.

If we assume the initial interval $[a, b]$ in the secant step satisfies (4.4), then c lies between a and b. If $c = B$, then U1 is satisfied and ϕ' is nonnegative at both b and B. In this case, corresponding to Figure 4.3, we attempt a secant step based on the values of ϕ' at b and B. The attempted secant step fails if $\bar{c}$ lies out the interval $[a, b]$, in which case the update simply returns the initial interval $[A, B]$. If $c = A$ in S3, then U2 is satisfied and ϕ' is negative at both a and A. In this case, corresponding to Figure 4.4, we attempt a secant step based on the values of ϕ' at a and A.

Assuming ϕ is not monotone, an initial interval $[a, b] = [a_0, b_0]$ satisfying (4.4) can be generated by sampling $\phi(\alpha)$ for various choices of α. Starting from this interval, and initializing $k = 0$, we now give a complete statement of the line search used for the numerical experiments in section 5, beginning with a list of the parameters.

Line search/CG_DESCENT *parameters.*
δ - range $(0, .5)$, used in the Wolfe conditions (2.2) and (4.1)
σ - range $[\delta, 1)$, used in the Wolfe conditions (2.3) and (4.1)
ϵ - range $[0, \infty)$, used in the approximate Wolfe termination (T2)
θ - range $(0, 1)$, used in the update rules when the potential intervals $[a, c]$ or $[c, b]$ violate the opposite slope condition contained in (4.4)
γ - range $(0, 1)$, determines when a bisection step is performed (L2 below)
η - range $(0, \infty)$, used in the lower bound for β_k^N in (1.6).

ALGORITHM. *Line search.*
L0. Terminate the line search if either (T1) or (T2) is satisfied.
L1. $[a, b] =$ secant$^2(a_k, b_k)$.
L2. If $b - a > \gamma(b_k - a_k)$, then $c = (a + b)/2$ and $[a, b] =$ update (a, b, c).
L3. Increment k, set $[a_k, b_k] = [a, b]$, and go to L0.

The line search is terminated whenever a point is generated for which either T1 or T2 holds.

THEOREM 4.1. *Suppose that ϕ is continuously differentiable on an interval $[a_0, b_0]$, where* (4.4) *holds. If $\delta < 1/2$, then the line search algorithm terminates at a point satisfying either* T1 *or* T2.

Proof. Due to the bisection step L2, the interval width $b_k - a_k$ tends to zero. Since each interval $[a_k, b_k]$ satisfies the opposite slope condition (4.4), we conclude that $\phi'(a_k)$ approaches 0. Hence, T2 holds for k sufficiently large. ∎

We now analyze the convergence rate of the secant^2 iteration. Since the root convergence order [29] of the secant method is $(1 + \sqrt{5})/2$, the order of convergence for a double secant step is $(1 + \sqrt{5})^2/4$. However, the iteration secant^2 is not a conventional double secant step since the most recent iterates are not always used to compute the next iterate; our special secant iteration was devised to first update one side of the bracketing interval and then the other side. This behavior is more attractive than a high convergence order. We now show that the convergence order of secant^2 is $1 + \sqrt{2} \approx 2.4$, slightly less than $(1 + \sqrt{5})^2/4 \approx 2.6$.

THEOREM 4.2. *Suppose that ϕ is three times continuously differentiable near a local minimizer α^*, with $\phi''(\alpha^*) > 0$ and $\phi'''(\alpha^*) \neq 0$. Then for a_0 and b_0 sufficiently close to α^* with $a_0 \leq \alpha^* \leq b_0$, the iteration*

$$[a_{k+1}, b_{k+1}] = \text{secant}^2(a_k, b_k)$$

converges to α^*. Moreover, the interval width $|b_k - a_k|$ tends to zero with root convergence order $1 + \sqrt{2}$.

Proof. Suppose that $\phi'''(\alpha^*) > 0$. The case $\phi'''(\alpha^*) < 0$ is treated in a similar way. Our double secant step, as seen in Figure 4.4, is

$$a_{k+1} = \text{ secant } (a_k, b_k) \quad \text{and} \quad b_{k+1} = \text{ secant } (a_k, a_{k+1}). \tag{4.5}$$

It is well known (e.g., see [3, p. 49]) that the error in the secant step $c = \text{ secant } (a, b)$ can be expressed as

$$c - \alpha^* = (a - \alpha^*)(b - \alpha^*)\frac{\phi'''(\xi)}{2\phi''(\bar{\xi})},$$

where $\xi, \bar{\xi} \in [a, b]$. Hence, for our double secant step, we have

$$\begin{bmatrix} \alpha^* - a_{k+1} \\ b_{k+1} - \alpha^* \end{bmatrix} = \begin{bmatrix} C_k(\alpha^* - a_k)(b_k - \alpha^*) \\ D_k(\alpha^* - a_k)^2(b_k - \alpha^*) \end{bmatrix}, \tag{4.6}$$

where C_k and D_k are constants depending on the second and third derivatives of ϕ near α^*; C_k approaches $\phi'''(\alpha^*)/2\phi''(\alpha^*)$ as a_k and b_k approach α^*, while D_k approaches C_k^2.

Let $\mathbf{E}_k$ denote the error vector

$$\mathbf{E}_k = \begin{bmatrix} a_k - \alpha^* \\ b_k - \alpha^* \end{bmatrix}.$$

Given any $\lambda \in (0, 1)$, it follows from (4.6) that there exists a neighborhood $\mathcal{N}$ of α^* with the property that whenever $a_k < \alpha^* < b_k$ with a_k and $b_k \in \mathcal{N}$, C_k and D_k are bounded and $\|\mathbf{E}_{k+1}\| \leq \lambda \|\mathbf{E}_k\|$. Consequently, the iteration (4.5) is convergent whenever $a_0 < \alpha^* < b_0$ with a_0 and $b_0 \in \mathcal{N}$.

Let $\bar{C}$ and $\bar{D}$ denote the maximum values for C_k and D_k, respectively, when a_k and $b_k \in \mathcal{N}$, and consider the following recurrence:

$$\begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} = \begin{bmatrix} \bar{C}A_kB_k \\ \bar{D}A_k^2B_k \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} A_0 \\ B_0 \end{bmatrix} = \begin{bmatrix} \alpha^* - a_0 \\ b_0 - \alpha^* \end{bmatrix}. \tag{4.7}$$

Since $C_k \leq \bar{C}$ and $D_k \leq \bar{D}$, it follows that $\alpha^* - a_k \leq A_k$ and $b_k - \alpha^* \leq B_k$ for each k. In other words, A_k and B_k generated by (4.7) bound the error in a_k and b_k, respectively.

Defining the variables

$$v_k = \log(A_k\sqrt{\bar{D}}) \quad \text{and} \quad w_k = \log(\bar{C}B_k),$$

we have

$$\begin{bmatrix} v_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ w_k \end{bmatrix}. \tag{4.8}$$

The solution is

$$\begin{bmatrix} v_k \\ w_k \end{bmatrix} = \frac{2v_0 + \sqrt{2}w_0}{4}(1 + \sqrt{2})^k \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} + \frac{-2v_0 + \sqrt{2}w_0}{4}(1 - \sqrt{2})^k \begin{bmatrix} -1 \\ \sqrt{2} \end{bmatrix}.$$

Observe that both v_0 and w_0 are negative when a_0 and b_0 are near α. Since $1+\sqrt{2} > |1-\sqrt{2}|$, we conclude that for k large enough,

$$\begin{bmatrix} v_k \\ w_k \end{bmatrix} \le \frac{2v_0 + \sqrt{2}w_0}{8}(1+\sqrt{2})^k \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}.$$

Hence, the root convergence order is $1+\sqrt{2}$. Since

$$b_k - a_k \le |b_k - \alpha^*| + |a_k - \alpha^*|,$$

$b_k - a_k$ converges to zero with root convergence order $1+\sqrt{2}$. □

5. Numerical comparisons. In this section we compare the CPU time performance of the new conjugate gradient method, denoted CG_DESCENT, to the L-BFGS limited memory quasi-Newton method of Nocedal [28] and Liu and Nocedal [25], and to other conjugate gradient methods as well. Comparisons based on other metrics, such as number of iterations or number of function/gradient evaluations, are given in [18], where extensive numerical testing of the methods is done. We considered both the PRP+ version of the conjugate gradient method, developed by Gilbert and Nocedal [13], where the β_k associated with the Polak–Ribière–Polyak conjugate gradient method [31, 32] is kept nonnegative, and versions of the conjugate gradient method developed by Dai and Yuan in [8, 10], denoted CGDY and DYHS, which achieve descent for any line search that satisfies the Wolfe conditions (2.2)–(2.3). The hybrid conjugate gradient method DYHS uses

$$\beta_k = \max\{0, \min\{\beta_k^{HS}, \beta_k^{DY}\}\},$$

where β_k^{HS} is the choice of Hestenes and Stiefel [22] and β_k^{DY} appears in [8]. The test problems are the unconstrained problems in the CUTE [4] test problem library.

The L-BFGS and PRP+ codes were obtained from Jorge Nocedal's Web page at http://www.ece.northwestern.edu/∼nocedal/software.html. The L-BFGS code is authored by Jorge Nocedal, while the PRP+ code is coauthored by Guanghui Liu, Jorge Nocedal, and Richard Waltz. In the documentation for the L-BFGS code, it is recommended that between 3 and 7 pairs of vectors be used for the memory. Hence, we chose 5 pairs of vectors for the memory. The line search in both codes is a modification of subroutine CSRCH of Moré and Thuente [27], which employs various polynomial interpolation schemes and safeguards in satisfying the strong Wolfe line search conditions.

We also manufactured a new L-BFGS code by replacing the Moré–Thuente line search with the new line search presented in our paper. We call this new code L-BFGS*. The new line search would need to be modified for use in the PRP+ code to ensure descent. Hence, we retained the Moré–Thuente line search in the PRP+ code. Since the conjugate gradient algorithms of Dai and Yuan achieve descent for any line search that satisfies the Wolfe conditions, we are able to use the new line search in our experiments with CGDY and with DYHS. All codes were written in Fortran and compiled with f77 (default compiler settings) on a Sun workstation.

For our line search algorithm, we used the following values for the parameters:

$$\delta = .1, \quad \sigma = .9, \quad \epsilon = 10^{-6}, \quad \theta = .5, \quad \gamma = .66, \quad \eta = .01.$$

Our rationale for these choices was the following: The constraints on δ and σ are $0 < \delta \le \sigma < 1$ and $\delta < .5$. As δ approaches 0 and σ approaches 1, the line search

terminates more quickly. The chosen values $\delta = .1$ and $\sigma = .9$ represent a compromise between our desire for rapid termination and our desire to improve the function value. When using the approximate Wolfe conditions, we would like to achieve decay in the function value, if numerically possible. Hence, we made the small choice $\epsilon = 10^{-6}$ for the error tolerance in (4.3). When restricting β_k in (1.6), we would like to avoid truncation if possible, since the fastest convergence for a quadratic function is obtained when there is no truncation at all. The choice $\eta = .01$ leads to infrequent truncation of β_k. The choice $\gamma = .66$ ensures that the length of the interval $[a, b]$ decreases by a factor of $2/3$ in each iteration of the line search algorithm. The choice $\theta = .5$ in the update procedure corresponds to the use of bisection. Our starting guess for step α_k in the line search was obtained by minimizing a quadratic interpolant.

In the first set of experiments, we stopped whenever

$$\text{(5.1)} \qquad \text{(a) } \|\nabla f(\mathbf{x}_k)\|_\infty \le 10^{-6} \quad \text{or} \quad \text{(b) } \alpha_k \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \le 10^{-20} |f(\mathbf{x}_{k+1})|,$$

where $\|\cdot\|_\infty$ denotes the maximum absolute component of a vector. In all but three cases, the iterations stopped when (a) was satisfied—the second criterion essentially says that the estimated change in the function value is insignificant compared to the function value itself.

The CPU time in seconds and the number of iterations, function evaluations, and gradient evaluations for each of the methods are posted on the following Web site: http://www.math.ufl.edu/∼hager/papers/CG. In running the numerical experiments, we checked whether different codes converged to different local minimizers; we only provide data for problems in which all six codes converged to the same local minimizer. The numerical results are now analyzed.

The performance of the six algorithms, relative to CPU time, was evaluated using the profiles of Dolan and Moré [11]. That is, for each method, we plot the fraction P of problems for which the method is within a factor τ of the best time. In Figure 5.1, we compare the performance of the four codes CG_DESCENT, L-BFGS*, L-BFGS, and PRP+. The left side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that were successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time. Since the top curve in Figure 5.1 corresponds to CG_DESCENT, this algorithm is clearly the fastest for this set of 113 test problems with dimensions ranging from 50 to 10,000. In particular, CG_DESCENT is fastest for about 60% (68 out of 113) of the test problems, and it ultimately solves 100% of the test problems. Since L-BFGS* (fastest for 29 problems) performed better than L-BFGS (fastest for 17 problems), the new line search led to improved performance. Nonetheless, L-BFGS* was still dominated by CG_DESCENT.

In Figure 5.2 we compare the performance of the four conjugate gradient algorithms. Observe that CG_DESCENT is the fastest of the four algorithms. Since CGDY, DYHS, and CG_DESCENT use the same line search, Figure 5.2 indicates that the search direction of CG_DESCENT yields quicker descent than the search directions of CGDY and DYHS. Also, DYHS is more efficient than CGDY. Since each of these six codes differs in the amount of linear algebra required in each iteration and in the relative number of function and gradient evaluations, different codes will be superior in different problem sets. In particular, the fourth ranked PRP+ code in Figure 5.1 still achieved the fastest time in 6 of the 113 test problems.

In our next series of experiments, shown in Table 5.1, we explore the ability of the algorithms and line search to accurately solve the test problems.

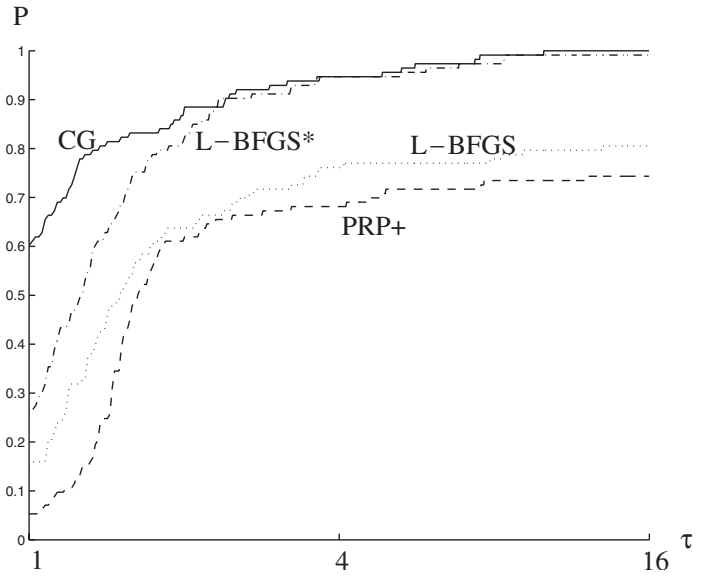

FIG. 5.1. *Performance profiles.*

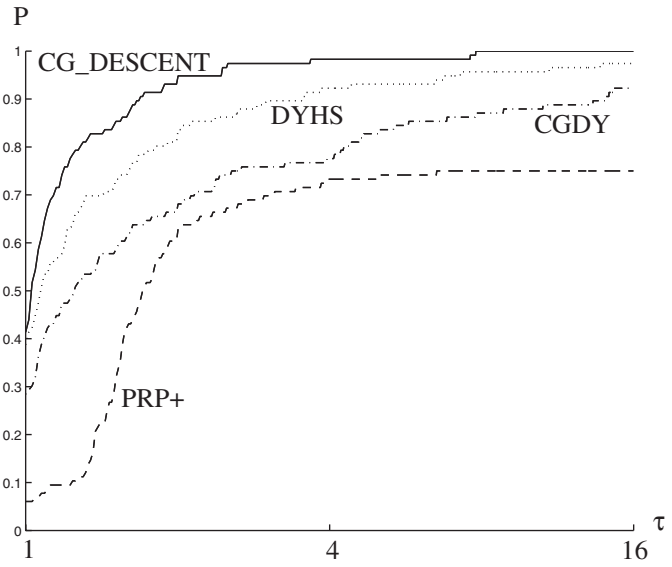

FIG. 5.2. *Performance profiles of conjugate gradient methods.*

In this series of experiments, we repeatedly solve six test problems, increasing the specified accuracy in each run. For the initial run, the stopping condition was $\|\mathbf{g}_k\|_\infty \le 10^{-2}$, and in the last run, the stopping condition was $\|\mathbf{g}_k\|_\infty \le 10^{-12}$. The test problems used in these experiments, and their dimensions, were the following:

TABLE 5.1
Solution time versus tolerance.

Tolerance $\|\mathbf{g}_k\|_\infty$	Algorithm	Problem #1	#2	#3	#4	#5	#6
10^{-2}	CG_DESCENT	5.22	2.32	0.86	0.00	1.57	10.04
	L-BFGS*	4.19	1.57	0.75	0.01	1.81	14.80
	L-BFGS	4.24	2.01	0.99	0.00	2.46	16.48
	PRP+	6.77	3.55	1.43	0.00	3.04	17.80
10^{-3}	CG_DESCENT	9.20	5.27	2.09	0.00	2.26	17.13
	L-BFGS*	6.72	6.18	2.42	0.01	2.65	19.46
	L-BFGS	6.88	7.46	2.65	0.00	3.30	22.63
	PRP+	12.79	7.16	3.61	0.00	4.26	24.13
10^{-4}	CG_DESCENT	10.79	5.76	5.04	0.00	3.23	25.26
	L-BFGS*	11.56	10.87	6.33	0.01	3.49	31.12
	L-BFGS	12.24	10.92	6.77	0.00	4.11	33.36
	PRP+	15.97	11.40	8.13	0.00	5.01	F
10^{-5}	CG_DESCENT	14.26	7.94	7.97	0.00	4.27	27.49
	L-BFGS*	17.14	16.05	10.21	0.01	4.33	36.30
	L-BFGS	16.60	16.99	10.97	0.00	4.90	F
	PRP+	21.54	12.09	12.31	0.00	6.22	F
10^{-6}	CG_DESCENT	16.68	8.49	9.80	5.71	5.42	32.03
	L-BFGS*	21.43	19.07	14.58	9.01	5.08	46.86
	L-BFGS	21.81	21.08	13.97	7.78	5.83	F
	PRP+	24.58	12.81	15.33	8.07	7.95	F
10^{-7}	CG_DESCENT	20.31	11.47	11.93	5.81	5.93	39.79
	L-BFGS*	26.69	25.74	17.30	12.00	6.10	54.43
	L-BFGS	26.47	F	17.37	9.98	6.39	F
	PRP+	31.17	F	17.34	8.50	9.50	F
10^{-8}	CG_DESCENT	23.22	12.88	14.09	9.68	6.49	47.50
	L-BFGS*	28.18	33.19	20.16	16.58	6.73	63.42
	L-BFGS	32.23	F	20.48	14.85	7.67	F
	PRP+	33.75	F	19.83	F	10.86	F
10^{-9}	CG_DESCENT	27.92	13.32	16.80	12.34	7.46	56.68
	L-BFGS*	32.19	38.51	26.50	26.08	7.67	72.39
	L-BFGS	33.64	F	F	F	8.50	F
	PRP+	F	F	F	F	11.74	F
10^{-10}	CG_DESCENT	33.25	13.89	21.18	13.21	8.11	65.47
	L-BFGS*	34.16	50.60	29.79	33.60	8.22	79.08
	L-BFGS	39.12	F	F	F	9.53	F
	PRP+	F	F	F	F	13.56	F
10^{-11}	CG_DESCENT	38.80	14.38	25.58	13.39	9.12	77.03
	L-BFGS*	36.78	55.70	34.81	39.02	9.14	88.86
	L-BFGS	F	F	F	F	9.99	F
	PRP+	F	F	F	F	14.44	F
10^{-12}	CG_DESCENT	42.51	15.62	27.54	13.38	9.77	78.31
	L-BFGS*	41.73	60.89	39.29	43.95	9.97	101.36
	L-BFGS	F	F	F	F	10.54	F
	PRP+	F	F	F	F	15.96	F

1. FMINSURF (5625)
2. NONCVXU2 (1000)
3. DIXMAANE (6000)
4. FLETCBV2 (1000)
5. SCHMVETT (10000)
6. CURLY10 (1000)

These problems were chosen somewhat randomly; however, we did not include any problem for which the optimal cost was zero. When the optimal cost is zero

while the minimizer $\mathbf{x}$ is not zero, the estimate $\epsilon|f(\mathbf{x}_k)|$ for the error in function value (which we used in the previous experiments) can be very poor as the iterates approach the minimizer (where f vanishes). These six problems all have nonzero optimal cost. The times reported in Table 5.1 differ slightly from the times reported at the Web site http://www.math.ufl.edu/∼hager/papers/CG due to timer errors and the fact that the computer runs were done at different times. In Table 5.1, F means that the line search terminated before the convergence tolerance for $\|\mathbf{g}_k\|$ was satisfied. According to the documentation for the line search in the L-BFGS and PRP+ codes, "Rounding errors prevent further progress. There may not be a step which satisfies the sufficient decrease and curvature conditions. Tolerances may be too small."

As can be seen in Table 5.1, the line search based on the Wolfe conditions (used in the L-BFGS and PRP+ codes) fails much sooner than the line search based on both the Wolfe and the approximate Wolfe conditions (used in CG_DESCENT and L-BFGS*). Roughly speaking, a line search based on the Wolfe conditions can compute a solution with accuracy on the order of the square root of the machine epsilon, while a line search that also includes the approximate Wolfe conditions can compute a solution with accuracy on the order of the machine epsilon.

6. Conclusions. We have presented a new conjugate gradient algorithm for solving unconstrained optimization problems. Although the update formulas (1.2)–(1.3) and (1.5)–(1.6) are more complicated than previous formulas, the scheme is relatively robust in numerical experiments. We prove that it satisfies the descent condition $\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le -\frac{7}{8}\|\mathbf{g}_k\|^2$, independent of the line search procedure, as long as $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$. For (1.5)–(1.6), we prove global convergence under the standard (not strong) Wolfe conditions. A new line search was introduced that utilizes the "approximate Wolfe" conditions; this approximation provides a more accurate way to check the usual Wolfe conditions when the iterates are near a local minimizer. Our line search algorithm exploits a double secant step, denoted secant^2, shown in Figures 4.3 and 4.4, that is designed to achieve rapid decay in the width of the interval which brackets an acceptable step. The convergence order of secant^2, given in Theorem 4.2, is $1 + \sqrt{2}$. The performance profile for our conjugate gradient algorithm (1.5)–(1.6), implemented with our new line search algorithm, was higher than those of the well-established L-BFGS and PRP+ methods for a test set consisting of 113 problems from the CUTE library.

Acknowledgments. Initial experimentation with a version of the line search algorithm was done by Anand Ramasubramaniam in his Master's thesis [35]. The line search used for the numerical experiments of section 5 also exploited a result of Holly Hirst [23], which is roughly the following: For a line search done with "quadratic accuracy," the conjugate gradient method retains its n-step local quadratic convergence property [5]. By [16, Lem. 5], an iteration based on a quadratic interpolant achieves suitable accuracy. Hence, when possible, we always started our line search with a step based on quadratic interpolation. Constructive comments by the referees and by associate editor Jorge Nocedal are gratefully acknowledged.

REFERENCES

[1] M. Al-Baali, *Decent property and global convergence of the Fletcher–Reeves method with in exact line search*, IMA J. Numer. Anal., 5 (1985), pp. 121–124.

[2] M. Al-Baali and R. Fletcher, *An efficient line search for nonlinear least squares*, J. Optim. Theory Appl., 48 (1984), pp. 359–377.

[3] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley, New York, 1978.

[4] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint, *CUTE: Constrained and unconstrained testing environments*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.
[5] A. I. Cohen, *Rate of convergence of several conjugate gradient algorithms*, SIAM J. Numer. Anal., 9 (1972), pp. 248–259.
[6] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y.-X. Yuan, *Convergence properties of nonlinear conjugate gradient methods*, SIAM J. Optim., 10 (1999), pp. 345–358.
[7] Y. H. Dai and L. Z. Liao, *New conjugate conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim., 43 (2001), pp. 87–101.
[8] Y. H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., 10 (1999), pp. 177–182.
[9] Y. H. Dai and Y. Yuan, *Nonlinear Conjugate Gradient Methods*, Shang Hai Science and Technology Publisher, Beijing, 2000.
[10] Y. H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient method for unconstrained optimization*, Ann. Oper. Res., 103 (2001), pp. 33–47.
[11] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Programming, 91 (2002), pp. 201–213.
[12] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149–154.
[13] J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21–42.
[14] A. A. Goldstein, *On steepest descent*, SIAM J. Control, 3 (1965), pp. 147–151.
[15] W. W. Hager, *Applied Numerical Linear Algebra*, Prentice–Hall, Englewood Cliffs, NJ, 1988.
[16] W. W. Hager, *A derivative-based bracketing scheme for univariate minimization and the conjugate gradient method*, Comput. Math. Appl., 18 (1989), pp. 779–795.
[17] W. W. Hager, *Numerical Linear Algebra: Practical Insights and Applications*, in preparation.
[18] W. W. Hager and H. Zhang, *CG_DESCENT, A conjugate gradient method with guaranteed descent*, ACM Trans. Math. Software, to appear.
[19] W. W. Hager and H. Zhang *A survey of nonlinear conjugate gradient methods*, Pacific J. Optim., submitted.
[20] J. Han, G. Liu, D. Sun, and H. Yin, *Two fundamental convergence theorems for nonlinear conjugate gradient methods and their applications*, Acta Math. Appl. Sinica, 17 (2001), pp. 38–46.
[21] J. Y. Han, G. H. Liu, and H. X. Yin, *Convergence of Perry and Shanno's memoryless quasi-Newton method for nonconvex optimization problems*, OR Trans., 1 (1997), pp. 22–28.
[22] M. R. Hestenes and E. L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.
[23] H. Hirst, *N-step Quadratic Convergence in the Conjugate Gradient Method*, Ph.D. dissertation, Department of Mathematics, Pennsylvania State University, State College, PA, 1989.
[24] C. Lemarechal, *A view of line-searches*, in Optimization and Optimal Control, Lecture Notes in Control and Inform. Sci. 30, A. Auslender, W. Oettli, and J. Stoer, eds., Springer-Verlag, Heidelberg, 1981, pp. 59–79.
[25] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Programming Ser. B, 45 (1989), pp. 503–528.
[26] J. J. Moré and D. C. Sorensen, *Newton's method*, in Studies in Numerical Analysis, G. H. Golub, ed., Mathematical Association of America, Washington, DC, 1984, pp. 29–82.
[27] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.
[28] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comp., 35 (1980), pp. 773–782.
[29] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics in Appl. Math. 30, SIAM, Philadelphia, 2000.
[30] J. M. Perry, *A Class of Conjugate Gradient Algorithms with a Two Step Variable Metric Memory*, Discussion paper 269, Center for Mathematical Studies in Economics and Management Science, Northwestern University, Chicago, 1977.
[31] E. Polak and G. Ribière, *Note sur la convergence de méthodes de directions conjuguées*, Rev. Française Informat. Recherche Opérationnelle, 3 (1969), pp. 35–43.
[32] B. T. Polyak, *The conjugate gradient method in extreme problems*, USSR Comp. Math. Math. Phys., 9 (1969), pp. 94–112.
[33] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method*, in Numerical Analysis, Lecture Notes in Math. 1066, D. F. Griffiths, ed., Springer, Berlin, 1984, pp. 122–141.

[34] M. J. D. Powell, *Restart procedures for the conjugate gradient method*, Math. Programming, 12 (1977), pp. 241–254.
[35] A. Ramasubramaniam, *Unconstrained Optimization by a Globally Convergent High Precision Conjugate Gradient Method*, Master's thesis, Department of Mathematics, University of Florida, Gainesville, FL, 2000.
[36] D. F. Shanno, *On the convergence of a new conjugate gradient algorithm*, SIAM J. Numer. Anal., 15 (1978), pp. 1247–1257.
[37] D. F. Shanno, *Globally convergent conjugate gradient algorithms*, Math. Programming, 33 (1985), pp. 61–67.
[38] C. Wang, J. Han, and L. Wang, *Global convergence of the Polak–Ribière and Hestenes–Stiefel conjugate gradient methods for the unconstrained nonlinear optimization*, OR Trans., 4 (2000), pp. 1–7.
[39] D. F. Shanno and K. H. Phua, *Remark on algorithm* 500, ACM Trans. Math. Software, 6 (1980), pp. 618–622.
[40] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235.
[41] P. Wolfe, *Convergence conditions for ascent methods.* II: *Some corrections*, SIAM Rev., 13 (1971), pp. 185–188.
[42] G. Zoutendijk, *Nonlinear programming, computational methods*, in Integer and Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, 1970, pp. 37–86.