Pertanika J. Sci. & Technol. 22 (1): 213 - 224 (2014)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Diagonal Preconditioned Conjugate Gradient Algorithm for Unconstrained Optimization

Choong Boon Ng*, Wah June Leong and Mansor Monsi

Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ABSTRACT

The nonlinear conjugate gradient (CG) methods have widely been used in solving unconstrained optimization problems. They are well-suited for large-scale optimization problems due to their low memory requirements and least computational costs. In this paper, a new diagonal preconditioned conjugate gradient (PRECG) algorithm is designed, and this is motivated by the fact that a pre-conditioner can greatly enhance the performance of the CG method. Under mild conditions, it is shown that the algorithm is globally convergent for strongly convex functions. Numerical results are presented to show that the new diagonal PRECG method works better than the standard CG method.

Keywords: Unconstrained optimization, conjugate gradient method, preconditioning, diagonal approximation for Hessian

INTRODUCTION

The nonlinear conjugate gradient (CG) method was designed to solve the following unconstrained optimization problem:

$$\min_{x \in R^n} f(x), \tag{1}$$

where $f: R^n \to R$ is a continuously differentiable function (Ismail Mohd *et al.*, 2007). The iterative formula of the CG method is given by

$$x_{k+1} = x_k + \lambda_k d_k \tag{2}$$

and

$$d_k = -g_k + \gamma_k d_{k-1}, \tag{3}$$

where, $d_0 = -g_0$ for $k = 0,1,\ldots n$ and $g_k = \nabla f(x_k)$, λ_k were chosen to satisfy some line search conditions along the search direction, d_k, and γ_k is a scalar parameter.

The idea of incorporating a pre-conditioner to the CG method is initiated by Raydan (1997), where the spectral gradient is combined with the conjugate gradient

Article history:
Received: 2 February 2012
Accepted: 30 October 2012

E-mail addresses:
boon85@live.com.my (Choong Boon Ng),
leongwj@upm.edu.my (Wah June Leong),
mmonsi@upm.edu.my (Mansor Monsi)
*Corresponding Author

ISSN: 0128-7680

directions. This is an iterative algorithm to generate a sequence $x_k, k = 0,1,2,\ldots,n$ as presented in (1) and (2), where d_k is a spectral gradient search direction in the successive iterations generated by:

$$d_{k+1} = -\beta_k g_{k+1} + \gamma_k s_k, \tag{4}$$

where β_k is a parameter that defines different CG methods, and $s_k = x_{k+1} - x_k$ (Andrei, 2010).

Unexpectedly, the spectral gradient choice associated in this algorithm shows more efficient results than sophisticated CG methods in many cases. It showed that spectral gradient and conjugate gradient combination produced more efficient algorithms (Raydan, 1997).

Andrei (2007) presented a new preconditioned conjugate gradient (PRECG) method, where the scaled memory-less BFGS update was used as the pre-conditioner. The scaling factor in the pre-conditioner was selected as a matrix, which was reset when the Powell restarted criterion (Powell, 1977) holds to ensure that the search directions would be descent directions. Consider (1), where the function *f* has continuous partial derivatives, and d_k is a search direction generated by:

$$d_{k+1} = -\theta_k g_{k+1} + \gamma_k s_k, \tag{5}$$

For $k = 0,1,2,\ldots,n,$ where θ_k is a parameter to be computed, and g_k denoting $\nabla f(x_k)$ is selected to minimize *f* along the search direction, d_k, with $s_k = x_{k+1} - x_k = \lambda_k d_k$, and γ_k is a scalar parameter to be determined. $x_0 \in R^n$ is an arbitrary initial value and the iterative process is initialized with an initial point x_0 and $d_0 = -g_0$.

From the success of the spectral gradient method used by Raydan (1997) and Andrei's scaled memory-less BFGS method (Andrei, 2007) in the preconditioning technique, we developed a new pre-conditioner, D_k, which is a diagonal matrix based on both the spectral gradient and matrix preconditioning ideas.

DIAGONAL PRECONDITIONED CONJUGATE GRADIENT ALGORITHM

To incorporate the correct inverse Hessian information into the preconditioner, D_k, we let the diagonal pre-conditioner D_k to satisfy the weak-secant equation of Dennis and Wolkowicz (1993), as follows:

$$y_k^T s_k = y_k^T D_k y_k \tag{6}$$

With this aim, we let $\mathrm{s_k} = x_k - x_{k-1}$ and $\mathrm{y_k} = g_k - g_{k-1}$, and consider the minimization problem:

$$\min \frac{1}{2} \| D_k - I \|_F^2 \tag{7a}$$

$$\text{s.t.} y_k^T \left(D_k - I \right) y_k = y_k^T s_k - y_k^T y_k \tag{7b}$$

where $\|\cdot\|_F$ denotes the standard Frobenius norm.

Since the objective function in (7a) and the feasible set is convex, it gives a unique solution for problems (7a) and (7b). Using the method of Lagrangian function, we can obtain:

$$D_k - I = \frac{(y_k^T y_k - y_k^T s_k)}{y_k^T G_k y_k} G_k \tag{8}$$

where $G_k = diag\left(\left((s)_k^{(1)}\right)^2, \ldots, \left((s)_k^{(n)}\right)^2\right)$.

Finally, by substituting $y_k^T G_k y_k = tr\left(G_k^2\right)$, where $tr(.)$ denotes the trace operator, it gives a diagonal pre-conditioner, which satisfies (6), as follows:

$$D_k = I + \frac{(y_k^T s_k - y_k^T y_k)}{tr(G_k^2)} G_k \tag{9}$$

It is shown that the diagonal preconditioner D_k in (9) is a special class of diagonal Hessian approximation derived by Leong *et al*. (2010), Farid and Leong (2011), Leong *et al*. (2011), Farid *et al*. (2010), Leong and Hassan (2009), and Hassan *et al.* (2009).

In this case, the new PRECG method search direction d_{k+1} is given by:

$$d_{k+1} = -D_{k+1} g_{k+1} + \theta_{k+1}\left(\frac{g_{k+1}^T s_k}{y_k^T s_k}\right) y_k - \left[\left(1 + \theta_{k+1}\frac{y_k^T y_k}{y_k^T s_k}\right)\frac{g_{k+1}^T s_k}{y_k^T s_k} - \theta_{k+1}\frac{g_{k+1}^T y_k}{y_k^T s_k}\right] s_k \tag{10}$$

The proposed pre-conditioner D_k is in diagonal matrix form, where the storage requirements is of $O(n)$. Moreover, this pre-conditioner D_k satisfies the weak-secant equation of Dennis and Wolkowicz (1993), which is a valid approximation of inverse Hessian.

The PRECG algorithm has the following steps:

Step 1. Given $x_0 \in R^n$, set $d_0 = -g_0$, $\lambda_0 = 1/\|g_0\|$ and $k = 0$. Update $x_1 = x_0 + \lambda_0 d_0$

Step 2. For $k \geq 1$, calculate λ_k which satisfying Wolfe conditions. Compute the direction d_k as in (10). Update the variables $x_{k+1} = x_k + \lambda_k d_k$. Then, compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 3. Test for the stopping condition. The iterations are stopped if stopping condition is satisfied. Else, set $k = k + 1$ and go to Step 2.

The Assumption 1 below is to guarantee the existing G is bounded.

Assumption 1

1. $f(x)$ is twice continuously differentiable and G denotes the matrix of second derivatives of $f(x)$.
2. G is bounded, that is $m_1 \| x \|^2 \leq x^T \nabla^2 f(x) x \leq m_2 \| x \|^2$, where $0 < m_1 \leq m_2$.
3. Function $f(x)$ is strongly convex and has Lipschitz continuous on gradient in the level set $L_0 = \{ x \in R^n : f(x) \leq f(x_0) \}$, where there exists constants $\mu > 0$ and L such that $(\nabla f(x) - \nabla f(y))^T (x - y) \geq \mu \| x - y \|^2$ and $\| \nabla f(x) - \nabla f(y) \| \leq L \| x - y \|$, for all x and y from L_0.

This proposed pre-conditioner D_k is proved to be bounded in such a way that we can expect the corresponded PRECG method to converge globally. With the aim to show that the proposed preconditioner D_k is bounded, so we have the following lemma:

Lemma 1: Assume that $\| D_0 \|_F \leq \sigma_0$, where σ_0 is a constant. Then, for all $k \geq 0$, $\| D_k \|_F \leq \sigma_k$, where σ_k is some constant and $\sigma_k \geq 0$. If we can show that the diagonal pre-conditioner D_k satisfies $\| D_k \|_F \leq \sigma_k$, and then the diagonal pre-conditioner D_k is bounded above.

Proof: Let $D_k = diag\left(d_k^{(i)}\right)$, $G_k = diag(y_k^{(1)^2}, \ldots, y_k^{(n)^2})$ and $y_k^{(M)}$ be the largest component of y_k.

Then from (9), we have

$$d_k^{(i)} = 1 + \frac{(y_k^T s_k - y_k^T y_k)}{tr(G_k^2)} ((y)_k^{(i)})^2.$$

It follows from Assumption 1 that we have,

$$d_k^{(i)} \leq 1 + \frac{\frac{1}{m_2} \| y_k \|^2 - \| y_k \|^2}{\sum (y_k^{(i)})^4} (y_k^{(M)})^2.$$

and the fact that $\| y_k \|^2 = \sum (y_k^{(i)})^2 \leq n (y_k^{(M)})^2$ gives

$$d_k^{(i)} \leq 1 + \frac{n \left| \frac{1}{m_2} - 1 \right|}{\sum (y_k^{(i)})^4} (y_k^{(M)})^4.$$

Finally, it leads to $d_k^{(i)} \leq 1 + n \left| \frac{1}{m_2} - 1 \right|$.

Hence, we have $\| D_k \| \leq \sigma_k$ where $\sigma_k = \sqrt{n + n^2 \left| \frac{1}{m_2} - 1 \right|}$. □

Below is the convergence result of our new algorithm when the objective function $f(x)$ satisfies Assumption 1 (iii).

Theorem 1.1 If at every step of the conjugate gradient given in (2) with the step length λ_k selected to satisfy the Wolfe conditions (Wolfe, 1969) and d_{k+1} is given by (10), then either $g_k = 0$ for some *k* or $\lim_{k \to \infty} g_k = 0$.

Proof: From $d_0 = -g_0$, we have $g_0^T d_0 = -\| g_0 \|^2 \leq 0$. When (10) is multiplied by g_{k+1}^T:

$$g_{k+1}^T d_{k+1} \leq \frac{1}{(y_k^T s_k)^2} \Big[-\sigma_{k+1} \| g_{k+1} \|^2 + 2\theta_{k+1} \left(g_{k+1}^T y_k \right) \left(g_{k+1}^T s_k \right) \left(y_k^T s_k \right) - \; g^T{}_{+1} s \quad y^T s \quad - \quad {}_{+1}(y^T y\ )(g^T$$

$$- \left(g_{k+1}^T s_k \right)^2 \left(y_k^T s_k \right) - \theta_{k+1} (y_k^T y_k)(g_{k+1}^T s_k)^2 \Big]$$

and with $u = (s_k^T y_k) g_{k+1}$ and $v = (g_{k+1}^T s_k) y_k$, we can then get the following by applying the inequality $u^T v \leq \frac{1}{2} (\| u \|^2 + \| v \|^2)$ to the second term of the right hand side of the above inequality:

$$g_{k+1}^T d_{k+1} \leq -\frac{\left(g_{k+1}^T s_k \right)^2}{y_k^T s_k}. \tag{11}$$

Therefore, by Wolfe's condition, $g_{k+1}{}^T s_k \geq \beta_2 g_k{}^T s_k$, $g_{k+1}^T d_{k+1} < 0$ for every $k = 0, 1, 2, \ldots, n.$

By strong convexity, we have $y_k^T d_k = (g_{k+1} - g_k)^T d_k \geq \mu \ \lambda_k \| d_k \|^2$.

Here $g_k \neq 0$ implies for all *k* by Theorem 1.1, where $g_k^T d_k < 0$. *f* is bounded from below due to its strongly convex over L_0. By summing over *k*, the Wolfe conditions $f(x_{k+1}) < f(x_k) + \beta_1 g_k{}^T s_k$, we have:

$$\sum_{k=0}^{\infty} \lambda_k g_k^T d_k \to -\infty.$$

Consider that d_k is a descent direction and the lower bound for λ_k which satisfies the Wolfe condition, $g_{k+1}{}^T s_k \geq \beta_2 g_k{}^T s_k$, then

$$\lambda_k \geq \frac{1 - \sigma_2 \left| g_k^T d_k \right|}{L \| d_k \|^2}.$$

and follows with

$$\sum_{k=1}^{\infty} \frac{\left|g_k^T d_k\right|^2}{\| d_k \|^2} < \infty. \tag{12}$$

Using the inequality of Cauchy and by strong convexity, we have $y_k^T s_k \geq \mu \| s_k \|^2$ and get

$$g_{k+1}^T d_{k+1} \leq -\frac{(g_{k+1}^T s_k)^2}{y_k^T s_k} \leq -\frac{\| g_{k+1} \|^2 \| s_k \|^2}{\mu \| s_k \|^2} = -\frac{\| g_{k+1} \|^2}{\mu}.$$

Hence, from (12) it follows that,

$$\sum_{k=0}^{\infty} \frac{\| g_k \|^4}{\| d_k \|^2} < \infty. \tag{13}$$

From (10), when θ_{k+1} is selected by spectral gradient, the direction d_{k+1} will then satisfy:

$$\| d_{k+1} \| \leq \left(\frac{2}{\mu} + \frac{2L}{\mu^2} + \frac{L^2}{\mu^3} \right) \| g_{k+1} \|. \tag{14}$$

By inserting the upper bound (14) for d_k in (13) will yield the following:

$$\sum_{k=0}^{\infty} \| g_k \|^2 < \infty,$$

which completes the proof. □

NUMERICAL RESULTS

In this section, we discuss some numerical experiments that are conducted in order to test the performance of our new gradient method for unconstrained optimization against the standard CG method.

We compare the performance of a Fortran implementation of our new algorithm with the standard CG algorithm on a set of 50 large-scale unconstrained optimization test problems in extended or generalized form(Andrei, 2008). All tests are run on a 2.6 GHz Pentium IV with 512MB of RAM and all algorithms are coded in Fortran commands. We have considered a number of variables $n = 1000, 2000, \ldots, 10000$ for each problem. For all the test runs, the termination condition is $\| g_k \| \leq 10^{-6}$. The accumulated number of iterations and the average of the norm of gradient are used to compare the effectiveness of the results.

For convenience, the following abbreviations are used to identify a particular conjugate gradient method.

1. SCG: The standard conjugate gradient method (without pre-conditioner).
2. PRECG: The preconditioned conjugate gradient method with

$$D_k = I + \frac{(y_k^T s_k - y_k^T y_k)}{Tr(G_k^2)} G_k .$$

Table 2 gives a summary of comparison results between PRECG algorithm and the standard CG method. The symbol *prob* and $\| g_k \|$ mean the number of the test problems and the norm of the gradient of the function, respectively. The $Iter$ means total iteration calls. Table 3 gives the comparison results of the number of function evaluation for all the methods. Meanwhile, Table 4 summarizes the performance of the PRECG algorithm versus the SCG algorithm on 50 problems, which achieved the least number of iteration and a lower value of gradient norms. The test problems are listed in Table 1.

From Table 4, the PRECG algorithm performs better than the SCG algorithm to achieve a minimum norm of gradients, with 31 problems out of 50 problems, as compared to the SCG algorithm, which is only achieved for 16 problems. In more specific, the percentage of efficiency for the PRECG algorithm is 30% compared to the SCG algorithm.

TABLE 1
Test Problems and their corresponding problem number (prob) (refer to Andrei, 2008)

Problem	Test problems
1	Extended Freudenstein & Roth Function
2	Extended Trigonometric Function
3	Extended Rosenbrock Function
4	Extended White & Holst Function
5	Extended Beale Function
6	Extended Penalty Function
7	Perturbed Quadratic Function
8	Raydan 1 Function
9	Raydan 2 Function
10	Diagonal 1 Function
11	Diagonal 2 Function
12	Diagonal 3 Function
13	Hager Function
14	Generalized Tridiagonal 1 Function
15	Extended Tridiagonal 1 Function
16	Extended Three Expo Terms Function
17	Generalized Tridiagonal 2 Function
18	Diagonal 4 Function
19	Diagonal 5 Function
20	Extended Himmelblau Function
21	Generalized PSC1 Function

TABLE 1 (continue)

Problem	Test problems
22	Extended PSC1 Function
23	Extended Powell Function
24	Extended Block-Diagonal BD1 Function
25	Extended Maratos Function
26	Extended Cliff Function
27	Quadratic Diagonal Perturbed Function
28	Extended Wood Function
29	Extended Hiebert Function
30	Quadratic QF1 Function
31	Extended Quadratic Penalty QP1 Function
32	Extended Quadratic Penalty QP2 Function
33	Quadratic QF2 Function
34	Extended EP1 Function
35	Extended Tridiagonal 2 Function
36	BDQRTIC (CUTE) Function
37	TRIDIA (CUTE) Function
38	ARWHEAD (CUTE) Function
39	NONDIA (CUTE) Function
40	NONDQUAR (CUTE) Function
41	DQDRTIC (CUTE) Function
42	EG2 (CUTE) Function
43	DIXMAANA (CUTE) Function
44	DIXMAANB (CUTE) Function
45	DIXMAANC (CUTE) Function
46	DIXMAANE (CUTE) Function
47	Partial Perturbed Quadratic PPQ1 Function
48	BroydenTridiagonal Function
49	Almost Perturbed Quadratic Function
50	Tridiagonal Perturbed Quadratic Function

TABLE 2
A comparison of the CG and PRECG methods in terms of total iteration calls and gradient norm

prob	CG algorithm		PRECG algorithm	
	Iter	$\|\| g_k \|\|$	*Iter*	$\|\| g_k \|\|$
1	91	1.47e-05	71	3.85e-06
2	701	3.97e-06	696	2.83e-06
3	248	9.27e-06	240	7.08e-06
4	305	9.36e-06	301	5.79e-06

TABLE 2 (continue)

prob	CG algorithm		PRECG algorithm	
	Iter	$\lVert g_k \rVert$	*Iter*	$\lVert g_k \rVert$
6	1067	1.12e-06	1061	1.93e-06
7	7080	5.72e-06	7076	5.31e-06
8	6057	4.45e-06	5991	4.32e-06
9	30	3.57e-07	30	3.57e-07
10	7680	3.84e-06	7256	4.31e-06
11	4773	1.90e-06	3954	1.77e-06
12	16969	6.97e-05	17245	6.71e-05
13	7896	1.84e-06	9732	1.50e-06
14	370	2.55e-06	380	2.34e-06
15	102	2.64e-05	68	1.78e-05
16	67	3.33e-06	60	6.12e-06
17	546	2.44e-06	532	2.71e-06
18	20	1.76e-06	31	4.32e-11
19	30	2.81e-10	30	2.81e-10
20	70	5.66e-06	63	6.29e-07
21	6075	2.17e-06	7134	1.79e-06
22	80	4.50e-09	89	3.38e-06
23	545	1.95e-05	573	2.88e-05
24	219	2.74e-05	2286	-
25	448	9.91e-05	447	1.75e-05
26	207	1.26e-05	16059	-
27	3026	9.15e-06	3131	8.83e-06
28	989	1.98e-05	1079	1.94e-05
29	512	4.12e-06	520	1.95e-06
30	6923	5.82e-06	7195	5.42e-06
31	528	1.17e-06	685	5.51e-06
32	128	1.98e-07	250	1.74e-07
33	7836	4.49e-06	8283	4.50e-06
34	18	6.56e-05	18	6.56e-05
35	319	3.44e-06	331	2.93e-06
36	18300	5.19e-05	18572	1.08e-04
37	19080	7.18e-03	18649	4.60e-02
38	31	1.29e-07	30	1.39e-07
39	298	2.51e-07	109	2.30e-08
40	19465	6.46e-05	18261	9.28e-06
41	51	2.53e-07	50	2.50e-07
42	16779	6.03e-04	11096	2.41e-04

TABLE 2 (continue)

prob	CG algorithm		PRECG algorithm	
	Iter	$\|\| g_k \|\|$	*Iter*	$\|\| g_k \|\|$
44	149	2.26e-06	193	5.17e-06
45	174	4.89e-06	245	1.36e-06
46	4015	6.14e-06	3967	6.41e-06
47	1107	1.72e-05	1053	1.60e-05
48	634	2.22e-06	645	2.83e-06
49	7360	5.17e-06	7002	4.98e-06
50	6826	6.57e-06	7149	6.08e-06

TABLE 3
A comparison of the methods in terms of the Total Number of Function Evaluation for all n

Pro	Methods (Number of functions calls)		*Pro*	Methods (Number of functions calls)	
	CG	PRECG		CG	PRECG
1	265	178	26	300	16147
2	1076	1068	27	4912	5090
3	489	482	28	1875	2030
4	613	546	29	1097	1085
5	233	188	30	9084	9456
6	25945	26366	31	12937	15435
7	9281	9228	32	308	590
8	8468	8376	33	10314	10902
9	90	90	34	59	59
10	21881	17385	35	582	553
11	6999	5781	36	478435	468394
12	354441	355758	37	25118	24330
13	238400	298785	38	101	80
14	5306	5210	39	1379	224
15	209	156	40	33513	32610
16	138	140	41	132	130
17	884	860	42	468361	294247
18	60	81	43	136	179
19	90	90	44	248	2808
20	160	153	45	298	3677
21	67721	67912	46	5521	5159
22	180	189	47	1773	1713
23	1015	1050	48	1010	1052
24	474	3294	49	9649	9253
25	970	932	50	8986	9386

TABLE 4
A comparison of CG and PRECG (All problems)

	CG performs better	PRECG performs better	Equal performance
function calls	21	26	3
norm of gradient	16	31	3

CONCLUSION

In this paper, we presented the performance of a new diagonal preconditioned conjugate gradient (PRECG) method. Accordingly, the PRECG method was also compared against the SCG method based on 50 benchmark problems. Based on our numerical experiments, the PRECG method has been shown to outperform the SCG method. Thus, it is concluded that the introduction of PRECG method is worthwhile.

REFERENCES

Andrei, N. (2008). An unconstrained optimization test functions collection. *Advanced Modeling & Optimization, 10*(1), 147-161.

Andrei, N. (2007). Scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. *Optimization Methods & Software, 22*(4), 561-571.

Andrei, N. (2010). Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. *European Journal of Operational Research*, *204*(3), 410-420.

Dennis, J. E., & Wolkowicz, H. (1993). Sizing and least-change secant methods. *SIAM Journal on Numerical Analysis*, *30*(5), 1291-1314.

Farid *et al*. (2010). A new two-step gradient-type method for large-scale unconstrained optimization. *Computers and Mathematics with Applications*, *59*(10), 3301-3307.

Farid, & Leong (2011). An improved multi-step gradient-type method for large scale optimization. *Computers and Mathematics with Applications*, *61*(11), 3312-3318.

Fletcher, R., & Reeves, C.M. (1964). Function minimization by conjugate gradients. *The Computer Journal*, *7*(2), 149-154.

Hassan *et al*. (2009). A new gradient method via quasi-Cauchy relation which guarantees descent. *Journal of Computational and Applied Mathematics*, *230*(1), 300-305.

Hestenes, M. R., & Stiefel, E.L. (1952). Methods of conjugate gradients for solving linear systems, J. *Research Nat. Bur. Standards*, *49*, 409-436.

Ismail bin Mohd *et al*. (2007). Unimodality Tests for Global Optimization of Single Variable Functions Using Statistical Methods. *Malaysian Journal of Mathematical Sciences, 1*(2), 205-215.

Leong, & Hassan (2009). A restarting approach on symmetric rank one update for unconstrained optimization. *Computational Optimization and Applications*, *42*(3), 327-334.

Leong *et al*. (2010). A monotone gradient method via weak secant equation for unconstrained optimization. *Taiwanese Journal of Mathematics*, *14*(2), 412-423.

Polak, E., & Ribiere, G. (1969). Note sur la convergence de directions conjugees, Rev. *Francaise Informat Recherche Opertionelle, 16*, 35-43.

Raydan, M. (1997). The Barzilain and Borwein gradient method for the large unconstrained minimization problem. *SIAM Journal on Optimization*, *7*, 26–33.

Wolfe, P. (1969). Convergence conditions for ascent methods. *SIAM Review, 11*, 226-235.