A Descent Conjugate Gradient Method for Optimization Problems

Ayinde Semiu*, *Member, IAENG,* Osinuga Idowu, Adio Adesina, Agboola Sunday, Adelodun Joseph, Uka Uchenna and Awe Olufisayo

***Abstract*—Over the years, a considerable number of conjugate gradient methods have been proposed based on modifications on the well-known classical conjugate gradient methods. These methods were shown to have satisfied descent condition taking into consideration the strong Wolfe line search and other line search schemes. Convergence of objective functions were also guarantied. In this study, a decent conjugate gradient method for solving unconstrained non-linear optimization problems is developed. Algorithm of the proposed method was well developed by constructing its update parameter. Descent properties of the method based on some assumptions on the objective function were established. The convergence analysis of the method showed that it converges globally taking into consideration the strong Wolfe conditions . Dolan and More performance profile was used to compare the numerical strength of this method with other methods, showing clear evidence of better performance of the new method in the profiles tested.**

***Index Terms*—global convergence; unconstrained optimization; strong Wolfe conditions; descent direction; step length.**

I. INTRODUCTION

An unconstrained optimization problem of the form

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

with $f : R^n \longrightarrow R$, where f is continuously differentiable is central to problems involving decision making arising from engineering, social sciences, sciences etc. In fact most of the optimization problems found in theoretical fields are reducible to problems of the form (1). In this paper we aim to construct an efficient conjugate gradient method (CG method) to solve large-scale problem (1).

Manuscript received March 11, 2024; revised July 31, 2024.

Ayinde, Semiu is a lecturer in the Department of Basic Sciences (Mathematics Unit), Babcock University, Ilishan-Remo, Ogun State, Nigeria. (Corresponding author, phone: +2348056131809, e-mail: ayindes@babcock.edu.ng).

Osinuga, Idowu is a Professor in the Department of Mathematics, College of Physical Sciences, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria. (e-mail: osinuga08@gmail.com).

Adio, Adesina is an Associate Professor in the Department of Basic Sciences (Mathematics Unit), Babcock University, Ilishan-Remo, Ogun State, Nigeria. (e-mail: adioa@babcock.edu.ng).

Agboola, Sunday is an Associate Professor in the Department of Mathematics, Nigerian Army University Biu, Nigeria. (e-mail: agboolasunday70@gmail.com).

Adelodun, Joseph is a Professor in the Department of Basic Sciences (Mathematics Unit), Babcock University, Ilishan-Remo, Ogun State, Nigeria. (e-mail: adelodunj@babcock.edu.ng).

Uka, Uchenna is a lecturer in the Department of Basic Sciences (Mathematics Unit), Babcock University, Ilishan-Remo, Ogun State, Nigeria. (e-mail: ukau@babcock.edu.ng).

Awe, Olufisayo is a postgraduate student in the Department of Computer Science, Red River College, Canada. (e-mail: aweolufisayoademi@gmail.com).

CG-methods have been shown to be effective in solving (1) using the following iterative formula:

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

The step length $\alpha_k > 0$ is determined from line search schemes known as strong Wolfe conditions:

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\delta\alpha_k g_k^T d_k \tag{3}$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k \tag{4}$$

for $0 < \delta \leq \sigma < 1$
with the direction d_k given by

$$d_k = \begin{cases} -g_k & k = 0, \\ -g_k + \beta_k d_{k-1} & k \geq 1. \end{cases} \tag{5}$$

$\beta_k \in R$ is a scalar referred to as the CG coefficient or update parameter where g_k represents the gradient of $f(x_k)$.
Hestenes and Stiefel [12] in 1952 presented an algorithm to solve algebraic equations. This algorithm later became the first CG method that solved the unconstrained optimization problem (1). CG methods are normally formed from different constructions of β_k. Other CG-methods developed in this line which formed the first generation of CG-methods referred to as classical CG-methods include:

$$\begin{cases} \beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \\ \beta_k^{FR} = \frac{||g_k||^2}{||g_{k-1}||^2}, \\ \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{||g_{k-1}||^2}, \\ \beta_k^{LS} = \frac{-g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}, \\ \beta_k^{DY} = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}}, \\ \beta_k^{CD} = \frac{-||g_k||^2}{d_{k-1}^T g_{k-1}}. \end{cases} \tag{6}$$

Where $||.||$ represents the euclidean norm and

$$y_{k-1} = g_k - g_{k-1}. \tag{7}$$

See [8], [11], [16], [20], [21] or [23] for more details and related work on the classical CG-methods.

A. Related Works

Among the classical methods, of interest to our work are LS and DY methods.
In 1991, Liu and Storey [16] determined β_k by using the conjugacy condition

$$d_{k-1}^T S_k d_k = 0, \quad (8)$$

where S_k is the Hessian matrix of f. In order to circumvent the computation of second derivative and matrices storage, (8) was replaced by

$$d_k^T(g_k - g_{k-1}) = 0. \quad (9)$$

Meanwhile, with β_k being suggested as

$$\beta_k = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})} \quad (10)$$

and using the fact that $d_{k-1}g_k = 0$, they proposed the following update parameter

$$\beta_k = \frac{-g_k^T(g_k - g_{k-1})}{d_{k-1}^T g_{k-1}}. \quad (11)$$

The descent and convergence properties of the method were presented.
Moreover, Dai and Yuan [7] in 1999 took into consideration deficiency found in the global convergence properties of well known CG methods to propose a method with strong global convergence properties. Their motivation propelled them to investigate CG method that generates descent direction when the standard Wolfe conditions (3) and (4) are satisfied. Let d_k be a descent direction with $d_{k-1}^T g_{k-1} < 0$. An update parameter β_k which defines a descent direction d_k was determined with the requirement that

$$-||g_k||^2 + \beta_k g_k^T d_{k-1} < 0. \quad (12)$$

$\beta_k > 0$ was assumed. They defined a parameter

$$\tau = \frac{||g_k||^2}{\beta_k}. \quad (13)$$

Hence, (12) was said to be equivalent to

$$\tau_k > g_k^T d_{k-1}. \quad (14)$$

By letting $\tau_k = d_{k-1}^T y_{k-1}$, they proposed the following formula.

$$\beta_k = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}}. \quad (15)$$

Their method is globally convergent when conditions (3) and (4) are satisfied.

As a result of the efficiency, ease of implementation and low memory status of CG-methods, a remarkable number of CG methods and hybrid CG methods were introduced and are still being introduced. A greater number of the proposed methods came as modifications of the classical CG-methods. Some of these methods include: Abubakar *et al* [2] proposed LS-type method under standard Wolfe and Armijo-like schemes; Dai and Liao [6] considered an inexact line search conjugacy scheme that reduces to a classical exact line search conjugacy scheme to solve (1). Dai and Yuan [7] worked on a CG-method based on Wolfe scheme that converges globally; Dai and Yuan [8] studied an hybrid version of CG-method for (1); Hu and Storey [13] worked on an algorithm that uses conjugate gradient coefficient on which certain conditions were placed; Jiang *et al* [14] presented a three-term conjugate algorithm that is based on LS CG method; Jie and Zhong [15] worked on three-term CG method for problem (1); Liu and Storey [16] studied inexact line search on conjugacy to solve (1); Lu *et al* [17] presented a modified Dai-Liao conjugacy condition based on a new quasi-Newton equation to solve (1); Malik *et al* [19] proposed a three-term CG method with a unique search direction that satisfied descent condition and Zheng [24] introduced a secant equation into Dai-Liao conjugacy condition.

B. Motivation

A considerable number of the proposed methods that solve (1) efficiently come as linear combinations or hybrids of the classical methods. For instance, [1] combined PRP and LS methods to solve (1). [5] developed a new Dai-Liao-type method that incorporates DY method for (1). [9] proposed an hybrid CG method that combined CD and DY methods. [18] applied the second inequality condition of the strong Wolfe conditions to modify the conjugate parameters of PRP and HS methods for their two methods to solve (1). In [23], hybrid method which is a linear combination of DY and HS methods was proposed. Ayinde, *et al* [4] combined AYO with DHS and DLS (see [25]) methods to give two hybrid methods. Moreover, the convergence property and practical performance of DY [7] and LS [16] respectively are of interest. In practice, LS performs better than DY, on the other hand, convergence property of DY surpasses that of LS and other well known CG methods. Our inspiration comes from [1] and [23] and also from the properties of LS and DY methods which when combine can produce an efficient CG method.
The construction of the proposed method is initiated from (5) alongside other equations to arrive at a formula which yields different well known classical methods when the constant terms are altered. Discussion on the construction of the proposed CG method is given under problem formulation in section 2. Section 3 discusses the proposed algorithm, the descent and convergence properties of the method. Section 4 covers the implementation of the proposed algorithm for nonlinear problems, parameter specification, numerical results and their discussion. Section 5 gives the conclusion.

II. Problem Formulation

This section discusses the construction of the conjugate coefficient for the proposed CG method. The starting point is the direction d_k given in (5).
From (5), we have

$$d_k = -g_k + \beta_k d_{k-1}. \quad (16)$$

Multiply (16) by g_k to give

$$g_k^T d_k = -||g_k||^2 + \beta_k g_k^T d_{k-1}. \quad (17)$$

From (17),

$$\beta_k = \frac{g_k^T d_k + ||g_k||^2}{g_k^T d_{k-1}}. \quad (18)$$

Multiply (7) by g_k to give

$$g_k^T y_{k-1} = ||g_k||^2 - g_k g_{k-1}. \quad (19)$$

Thus,

$$||g_k||^2 = g_k^T y_{k-1} + g_k g_{k-1}. \quad (20)$$

By substituting (20) in (18),

$$\beta_k = \frac{g_k^T d_k + g_k^T y_{k-1} + g_k g_{k-1}}{g_k^T d_{k-1}}. \quad (21)$$

Similarly, multiply (7) by d_{k-1} to give

$$d_{k-1}^T y_{k-1} = g_k^T d_{k-1} - g_{k-1}^T d_{k-1}. \quad (22)$$

From (22), we have

$$g_k^T d_{k-1} = d_{k-1}^T y_{k-1} + g_{k-1}^T d_{k-1}. \quad (23)$$

Substitute (23) in (21) to give

$$\beta_k = \frac{g_k^T d_k + g_k^T y_{k-1} + g_k g_{k-1}}{d_{k-1}^T y_{k-1} + g_{k-1}^T d_{k-1}}. \quad (24)$$

At this point, we shall introduce constants $a,\ b,\ c,\ e,\ f \in R$ such that

$$\beta_k^A = \frac{a g_k^T d_k + b g_k^T y_{k-1} + f g_k g_{k-1}}{c d_{k-1}^T y_{k-1} + e g_{k-1}^T d_{k-1}}. \quad (25)$$

From (6),

$$-\beta_k^{LS} = \frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} \quad (26)$$

and from ([7], p.180),

$$\frac{d_k^T g_k}{d_{k-1}^T g_{k-1}} = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}}. \quad (27)$$

Interestingly, we note that if $a = 0$, $e = 0$, $b = 1$, $c = 1$ and $f = 0$ in (25), $\beta_k^A = \beta_k^{HS}$,
if $a = 1$, $e = 1$, $b = 0$, $c = 0$ and $f = 0$ in (25), $\beta_k^A = \beta_k^{DY}$ and
if $a = 0$, $e = 1$, $b = -1$, $c = 0$ and $f = 0$ in (25), $\beta_k^A = \beta_k^{LS}$.
Therefore, this paper proposes a descent CG-method as follows: Let $a = 1$, $b = 1$, $c = 0$, $e = 1$ and $f = 0$ in (25) to give

$$\beta_k^{AOAAH} = \frac{g_k^T d_k}{g_{k-1}^T d_{k-1}} + \frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}}. \quad (28)$$

By using (27), the proposed CG coefficient becomes

$$\beta_k^{AOAAH} = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}} + \frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} \quad (29)$$

The choice of values of the constants a, b, c, e and f is to achieve a linear combination of DY and LS CG methods. Equations (2) and (5) with β_k^{AOAAH} will be called AOAAH-CG method.

III. Proposed Algorithm and Properties

In this section, the algorithm, descent properties and convergence results for the proposed method are presented.

3.1 Algorithm (new β_k)

Step 1:Let $x_0 \in R^n$ be the initial point with $\epsilon > 0$, set $k = 0$ and $d_0 = -g_0$. If $||g_k|| \le \epsilon$, then stop;
Step 2: Determine $\alpha_k > 0$ using conditions (3)-(4);
Step 3: Compute β_k^{AOAAH} and find $\{x_k\}$, $\{g_k\}$ and $\{d_k\}$;
Step 4: Set $k = k + 1$ and then proceed to step 2.

3.2 Descent Properties

The descent properties of the new method are addressed in the following theorem.

Theorem 3.2.1. *Let d_k and g_k be generated by the CG-method (29) algorithm as presented in subsection 3.1 . Then, d_k satisfies the following condition.*

$$g_k^T d_k \le -c||g_k||^2 \quad (30)$$

or

$$d_k^T g_k < 0, \quad (31)$$

for each $k \ge 0$.

Proof: It is obvious by induction that

$$d_0^T g_0 = -||g_0||^2 \quad (32)$$

is true for the case when $k = 0$.
Let (30) be true when $k \ge 1$. Multiplying (16) by g_k to give

$$d_k^T g_k = -||g_k||^2 + \beta_k^{AOAAH} d_{k-1}^T g_k. \quad (33)$$

Hence,

$$d_k^T g_k = -||g_k||^2 + (\frac{||g_k||^2}{d_{k-1}^T y_{k-1}} + \frac{g_k^T y_k}{d_{k-1}^T g_{k-1}}) d_{k-1}^T g_k. \quad (34)$$

From (26) and (27), we have

$$d_k^T g_k = -||g_k||^2 + (\beta_k^{DY} - \beta_k^{LS}) d_{k-1}^T g_k. \quad (35)$$

Case 1. If $\beta_k^{LS} > 0$. Since $\beta_k^{DY} > 0$, (See [7]) therefore,

$$d_k^T g_k \le -||g_k||^2 + \beta_k^{DY} d_{k-1}^T g_k. \quad (36)$$

By [7], $c = \frac{1}{1+\sigma}$. Hence,

$$d_k^T g_k \le -(\frac{1}{1+\sigma})||g_k||^2. \quad (37)$$

Therefore, sufficient descent property is satisfied.

Case 2. If $\beta_k^{LS} < 0$. Then,

$$d_k^T g_k = -||g_k||^2 + (\beta_k^{DY} - \beta_k^{LS}) d_{k-1}^T g_k \quad (38)$$

$$= -||g_k||^2 + (\frac{||g_k||^2}{d_{k-1}^T y_{k-1}} + \frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}) d_{k-1}^T g_k. \quad (39)$$

$\beta_k^{LS} < 0$ implies $g_k^T y_{k-1} < 0$, $g_k^T y_{k-1} = ||g_k||^2 - g_k g_{k-1} < 0$ and $g_k g_{k-1} > 0$. Since $g_{k-1}^T d_{k-1} < 0$ (see [5]) then,

$$d_k^T g_k \le -||g_k||^2 + (\frac{||g_k||^2}{d_{k-1}^T y_{k-1}} - \frac{g_k^T g_{k-1}}{d_{k-1}^T g_{k-1}}) d_{k-1}^T g_k. \quad (40)$$

$$< (\frac{||g_k||^2}{d_{k-1}^T y_{k-1}} - \frac{g_k^T g_{k-1}}{d_{k-1}^T g_{k-1}}) g_{k-1}^T d_{k-1} < 0. \quad (41)$$

Therefore, sufficient descent property is satisfied.

3.3 Convergence Results
The following assumptions on the objective function is required for the convergence results.
Assumptions 3.3.1
(1) Set
$$U = \{x|f(x) \leq f(y)\} \tag{42}$$
bounded with $y \in R^n$.
(2) Given a neighborhood V of U, f is continuous and differentiable in V with its gradient $g(x)$ and Lipschitz constant L satisfying
$$||g(x) - g(y)|| \leq L||x - y|| \tag{43}$$
for any $x, y \in V$.

In addition, by Assumption 3.3.1 there are two constants D and ϕ that satisfy
$$||x - y|| \leq D \tag{44}$$
for any $x, y \in V$
and
$$||g(x)|| \leq \phi \tag{45}$$
for any $x \in V$.
The following results are on convergence for the new method.
Lemma 3.3.2. *Let the conditions in the assumption 3.3.1 be satisfied and for any equation of the form (5) with descent direction d_k where α_k satisfies the Wolfe conditions in (3) and (4). Then,*
$$\sum_{i=1}^{\infty} \frac{(g_k^T d_k)^2}{||g_k||^2} < \infty. \tag{46}$$
Proof: The comprehensive proof of Lemma 3.3.2 can be found in [7].
Theorem 3.3.3. *Suppose the conditions in the assumption 3.3.1 are satisfied and that x_k is determined by the CG-method (29) algorithm in section 2. Then,*
$$\lim_{k\to\infty} \inf ||g_k|| = 0. \tag{47}$$
Proof: **Case 1.** When $\beta_k^{LS} > 0$. We prove the result by contradiction.
Let
$$\lim_{k\to\infty} \inf ||g_k|| \neq 0. \tag{48}$$
Given that
$$||g_k|| > 0, \tag{49}$$
there exits a constant $n > 0$, where
$$||g_k|| > n \quad \forall k. \tag{50}$$
From (16),
$$d_k + g_k = \beta_k d_{k-1}. \tag{51}$$
Square of both sides of (51) is taken with $\beta_k = \beta_k^{AOAAH}$ to have
$$||d_k||^2 = -||g_k||^2 - 2d_k^T g_k + (\beta_k^{AOAAH})^2 \ ||d_{k-1}||^2. \tag{52}$$
(52) is divided by $(d_k^T g_k)^2$ to give
$$\frac{||d_k||^2}{(d_k^T g_k)^2} = -\frac{||g_k||^2}{(d_k^T g_k)^2} - \frac{2d_k^T g_k}{(d_k^T g_k)^2} + \frac{(\beta_k^{AOAAH})^2 \ ||d_{k-1}||^2}{(d_k^T g_k)^2}. \tag{53}$$
Since $\beta_k^{AOAAH} \leq \beta_k^{DY}$ when $\beta_k^{LS} > 0$ for $k \geq 1$.
Therefore, result follows from [7].
Case 2. When $\beta_k^{LS} < 0$.
Since $d_{k-1}^T g_{k-1} < 0$ (see [5]).
$$\beta_k^{AOAAH} = \frac{||g_k||^2}{d_{k-1}^T(g_k - g_{k-1})} + \frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}. \tag{54}$$
From (27)
$$\beta_k^{AOAAH} = \frac{d_k^T g_k}{d_{k-1}^T g_{k-1}} + \frac{g_k y_{k-1}}{d_{k-1}^T g_{k-1}}. \tag{55}$$
Since $g_k^T y_{k-1} < 0$ it implies that$-g_k^T y_{k-1} > 0$ and by (30), we have
$$\beta_k^{AOAAH} \leq \frac{c||g_k||^2}{|-d_{k-1}^T g_{k-1}|} + \frac{|-g_k^T y_{k-1}|}{|-d_{k-1}^T g_{k-1}|} \tag{56}$$
$$\leq \frac{c||g_k||^2 + ||-g_k|||g_k - g_{k-1}|}{|-d_{k-1}^T g_{k-1}|}. \tag{57}$$
By (43) and (44),
$$\beta_k^{AOAAH} \leq \frac{c||g_k||^2 + ||g_k||LD}{|-d_{k-1}^T g_{k-1}|} \tag{58}$$
$$= \frac{c||g_k||^2 + ||g_k||LD}{||d_{k-1}||||g_{k-1}||}. \tag{59}$$
$$\beta_k^{AOAAH} \leq \frac{||g_k||(c||g_k|| + LD)}{||d_{k-1}||||g_{k-1}||}. \tag{60}$$
By (45)and [22, (Theorem 5.8)] where $||g_{k-1}|| \geq c_1$, we have
$$\beta_k^{AOAAH} \leq \frac{||g_k||(c\phi + LD)}{c_1||d_{k-1}||}. \tag{61}$$
From (16),
$$||d_k|| = ||g_k|| + |\beta_k^{AOAAH}|||d_{k-1}||. \tag{62}$$
$$||d_k|| \leq ||g_k|| + \frac{||g_k||(c\phi + LD)}{c_1||d_{k-1}||}||d_{k-1}||. \tag{63}$$
$$||d_k|| \leq (1 + \frac{(c\phi + LD)}{c_1})||g_k||. \tag{64}$$
The result of using (30), (64) together with Zoutendijk's condition in [26] is
$$\sum_{k\geq 1} \frac{(g_k^T d_k)^2}{||d_k||^2} < +\infty. \tag{65}$$
Inequality (65) obviously implies (47).

IV. Proposed Algorithm Implementation for Nonlinear Problems

This section presents the numerical experiment and results done to evaluate the performance of our proposed algorithm with $\beta_k^{AOAAH} = \beta_k$ in comparison with other notable algorithms available in the literature.

A. Parameter Specification

We set $\delta = 0.0001$, $\sigma = 0.9$ for the algorithm code for all the formulas. $t = 1$ was selected for NEW+ CG method [24]. All the algorithms were implemented on MATLAB $R2015a$, HP 650 windows 10 OS and RAM 3GB. The algorithm will terminate if $||g_k|| \leq 10^{-6}$.

B. Numerical Results and Discussion

In this subsection, tables for the numerical experiment and graphs showing graphical representations of the performance indices of the proposed method against the existing methods and analysis of the results of the experiment are given. On the tables, $Abrev.$ represents abbreviations of the test problems while $Dim.$ represents dimensions of the problems. Table I shows the set of test problems drawn from [3] with their initial points considered for the experiment. Computation was done by using conditions (3)-(4) for all the formulas for comparison. We adopt the following abbreviations in the tables: Extended Penalty-EP, Quadratic-QU, Extended MCCORMCK-EK, Diagonal 4-DI, Extended Himmelblau-HE, Extended Baele-EB, Raydan 1-RY1, Raydan 2-RY2, Extended Three Exponential Terms-E3, Extended Tridiagonal2-E2, ARWHEAD-AR, DQDRTIC-DQ, MDF EXPLIN 1-ME, RMODF COSINE-CO, RMODF SINE-SI, Extended Rosenbrock ER, Extended Booth- EX, Chebyquad-CB, Quadratic Diagonal Perturbed-QP and Generalized PSCI-PS.

Table I. List of test problems and initial points.

s/n	$Abrev.$	$Dim.$	$initial\ points$
1	EP	2	$(1, 2, ..., n)$
2	QU	2	$(1 - 1/1, ..., 1 - 1/n$
3	ER	2	$(1, 1, ..., 1)$
4	ER	100	$(1, 1, ..., 1)$
5	ER	500	$(1, 1, ..., 1)$
6	ER	1000	$(1, 1, ..., 1)$
7	$D4$	2	$(1, 1, ..., 1)$
8	$D4$	100	$(1, 1, ..., 1)$
9	$D4$	500	$(1, 1, ..., 1)$
10	DI	1000	$(1, 1, ..., 1)$
11	DI	10000	$(1, 1, ..., 1)$
12	DI	50000	$(1, 1, ..., 1)$
13	DI	100000	$(1, 1, ..., 1)$
14	HE	2	$(1, 1, ..., 1)$
15	HE	100	$(1, 1, ..., 1)$
16	HE	500	$(1, 1, ..., 1)$
17	HE	1000	$(1, 1, ..., 1)$
18	HE	10000	$(1, 1, ..., 1)$
19	HE	50000	$(1, 1, ..., 1)$
20	HE	100000	$(1, 1, ..., 1)$
21	EB	2	$(1, 0.8, ..., 1, 0.8)$
22	EB	500	$(1, 0.8, ..., 1, 0.8)$
23	EB	1000	$(1, 0.8, ..., 1, 0.8)$
24	EB	10000	$(1, 0.8, ..., 1, 0.8)$
25	$RY1$	2	$(1, 1, ..., 1)$
26	$RY1$	100	$(1, 1, ..., 1)$
27	$RY1$	500	$(1, 1, ..., 1)$
28	$RY1$	1000	$(1, 1, ..., 1)$
29	$RY1$	10000	$(1, 1, ..., 1)$
30	$RY1$	50000	$(1, 1, ..., 1)$
31	$RY1$	100000	$(1, 1, ..., 1)$
32	$RY2$	2	$(1, 1, ..., 1)$
33	$RY2$	100	$(1, 1, ..., 1)$
34	$RY2$	500	$(1, 1, ..., 1)$
35	$RY2$	1000	$(1, 1, ..., 1)$

Table II. List of test problems and initial points contd.

s/n	$Abrev.$	$Dim.$	$initial\ points$
36	$RY2$	10000	$(1, 1, ..., 1)$
37	$RY2$	50000	$(1, 1, ..., 1)$
38	$RY2$	100000	$(1, 1, ..., 1)$
39	$E3$	2	$(0.1, 0.1, ..., 0.1)$
40	$E3$	100	$(0.1, 0.1, ..., 0.1)$
41	$E3$	500	$(0.1, 0.1, ..., 0.1)$
42	$E3$	1000	$(0.1, 0.1, ..., 0.1)$
43	$E2$	2	$(1, 1, ..., 1)$
44	$E2$	500	$(1, 1, ..., 1)$
45	AR	2	$(1, 1, ..., 1)$
46	AR	100	$(1, 1, ..., 1)$
47	DQ	2	$(3, 3, ..., 3)$
48	DQ	100	$(3, 3, ..., 3)$
49	DQ	500	$(3, 3, ..., 3)$
50	DQ	1000	$(3, 3, ..., 3)$
51	DQ	10000	$(3, 3, ..., 3)$
52	ME	2	$(1, 1, ..., 1)$
53	ME	100	$(1, 1, ..., 1)$
54	ME	500	$(1, 1, ..., 1)$
55	ME	1000	$(1, 1, ..., 1)$
56	ME	10000	$(1, 1, ..., 1)$
57	ME	50000	$(1, 1, ..., 1)$
58	ME	100000	$(1, 1, ..., 1)$
59	CO	2	$(1, 1, ..., 1)$
60	CO	100	$(1, 1, ..., 1)$
61	CO	500	$(1, 1, ..., 1)$
62	CO	1000	$(1, 1, ..., 1)$
63	CO	10000	$(1, 1, ..., 1)$
64	SI	2	$(1, 1, ..., 1)$
65	SI	100	$(1, 1, ..., 1)$
66	SI	500	$(1, 1, ..., 1)$
67	SI	1000	$(1, 1, ..., 1)$
68	SI	10000	$(1, 1, ..., 1)$
69	SI	50000	$(1, 1, ..., 1)$
70	SI	100000	$(1, 1, ..., 1)$
71	EK	2	$(1, 1, ..., 1)$
72	EK	100	$(1, 1, ..., 1)$
73	EK	500	$(1, 1, ..., 1)$
74	EK	1000	$(1, 1, ..., 1)$
75	EK	10000	$(1, 1, ..., 1)$
76	EX	2	$(1, 3, ..., 1, 3)$
77	EX	100	$(1, 3, ..., 1, 3)$
78	EX	500	$(1, 3, ..., 1, 3)$
79	EX	1000	$(1, 3, ..., 1, 3)$
80	EX	10000	$(1, 3, ..., 1, 3)$
81	EX	1000	$(1, 3, ..., 1, 3)$
82	EX	10000	$(1, 3, ..., 1, 3)$
83	CB	2	$(1, 1, ..., 1)$
84	CB	100	$(1, 1, ..., 1)$
85	CB	500	$(1, 1, ..., 1)$
86	CB	1000	$(1, 1, ..., 1)$
87	CB	10000	$(1, 1, ..., 1)$
88	QP	2	$(0.5, 0.5, ..., 0.5)$
89	PS	2	$(3, 0.1, ..., 3, 0.1)$

Table III. Numerical result CPU and values of function f.

s/n	$Prob.$	$Dim.$	$AOAAH$	DY	LS	$NEW+$
			CPU/FN	CPU/FN	CPU/FN	CPU/FN
1	EP	2	$0.245/2.17$	F/F	F/F	F/F
2	QU	2	$0.16/-1.06$	$0.2/-1.06$	$0.17/9.27E-01$	$0.144/-1.06$
3	ER	2	$4.792/1.49E-15$	$0.539/2.82E-18$	$1.717/4.37E-14$	F/F
4	ER	100	$3.516/1.98E-13$	$0.565/1.41E-16$	$1.997/1.24E-1$	F/F
5	ER	500	$3.66/9.23E-17$	F/F	$29.079/3.39E-16$	F/F
6	ER	1000	$3.825/1.14E-16$	F/F	$19.615/3.95E-16$	F/F
7	DI	2	$0.163/5.75E-16$	$0.171/2.28E-14$	$0.27/6.53E-17$	$0.169/7.25E-18$
8	DI	100	$0.203/2.85E-16$	$0.209/1.45E-14$	$0.348/7.18E-18$	$0.709/7.86E-18$
9	DI	500	$0.194/1.42E-16$	$0.221/8.24E-15$	$0.371/3.59E-17$	F/F
10	DI	1000	$0.18/2.84E-16$	$0.223/1.65E-14$	$0.377/7.18E-17$	$0.247/1.77E-16$
11	DI	10000	$0.47/2.83E-16$	$0.413/6.25E-15$	$1.22/3.36E-17$	F/F
12	DI	50000	$1.432/1.41E-16$	$1.157/1.05E-14$	$4.736/7.89E-18$	$19.286/2.00E-09$
13	DI	100000	$3.132/2.81E-16$	$2.134/2.10E-14$	$10.18/1.58E-17$	$16.938/F$
14	HE	2	$0.302/1.43E-15$	$0.942/1.24E-14$	$0.59/2.94E-15$	$0.34/1.23E-15$
15	HE	100	$0.412/3.93E-15$	$1.21/1.64E-14$	$0.685/8.61E-16$	$0.296/3.92E-15$
16	HE	500	$0.417/4.39E-16$	$1.137/8.08E-15$	$0.768/1.15E-15$	$0.367/2.04E-15$
17	HE	1000	$0.446/9.62E-16$	$1.228/2.94E-15$	$0.992/7.71E-15$	$0.395/4.08E-15$
18	HE	10000	$1.038/5.04E-15$	$2.965/7.50E-15$	$3.299/4.47E-15$	$1.197/3.33E-15$
19	HE	50000	$65.4/1.88E-16$	$219.6/1.14E-14$	$218.3/9.00E-16$	$526.9/1.03E-14$
20	HE	100000	$7.456/3.45E-15$	$22.24/1.26E-14$	$14.31/4.47E-15$	$15.64/1.09E-15$
21	EB	2	$4.114/9.31E-13$	$5F/F$	$2.393/4.52E-01$	F/F
22	EB	500	$2.756/F$	F/F	F/F	$6.146/F$
23	EB	1000	$7.874/8.65E-13$	F/F	F/F	F/F
24	EB	10000	$38.827/1.19E-12$	F/F	F/F	F/F
25	$RY1$	2	$0.073/F$	$0.053/2.71E+13$	F/F	$0.078/F$
26	$RY1$	100	$0.017/143000$	$0.015/143000$	$0.015/143000$	$0.013/143000$
27	$RY1$	500	$0.008/8.73E+07$	$0.008/8.73E+07$	$0.013/8.73E+07$	$0.012/8.73E+07$
28	$RY1$	1000	$0.008/1.39E+09$	$0.009/1.39E+09$	$0.011/1.39E+09$	$0.008/1.39E+09$
29	$RY1$	10000	$0.014/1.39E+13$	$0.012/1.39E+13$	$0.015/1.39E+13$	$0.012/1.39E+13$
30	$RY1$	50000	$0.031/8.68E+15$	$0.057/8.68E+15$	$0.071/8.68E+15$	$0.064/8.68E+15$
31	$RY1$	100000	$0.039/1.39E+17$	$0.035/1.39E+17$	$0.053/1.39E+17$	$0.047/1.39E+17$
32	$RY2$	2	$0.102/2.00$	$0.177/2.00$	$0.087/2.00$	$0.015/2.00$
33	$RY2$	100	$0.121/1.00E+02$	$0.218/1.00E+02$	$0.108/1.00E+02$	$0.086/1.00E+02$
34	$RY2$	500	$0.107/5.00E+02$	$0.215/5.00E+02$	$0.079/5.00E+02$	$0.052/5.00E+02$
35	$RY2$	1000	$0.11/1.00E+03$	$0.222/1.00E+03$	$0.099/1.00E+03$	$0.048/1.00E+03$
36	$RY2$	10000	$0.193/1.00E+04$	$0.37/1.00E+04$	$0.151/1.00E+04$	$0.078/1.00E+04$
37	$RY2$	50000	$0.342/5.00E+04$	$0.631/5.00E+04$	$0.238/5.00E+04$	$0.508/5.00E+04$
38	$RY2$	100000	$0.572/1.00E+05$	$1.01/1.00E+05$	$0.377/1.00E+05$	$2.611/1.00E+05$
39	$E3$	2	$0.284/2.56E+00$	$0.467/2.56E+00$	$0.32/2.56E+00$	$0.296/2.56E+00$
40	$E3$	100	$0.367/128$	$0.541/1.28E+02$	$0.454/128$	$0.641/128$
41	$E3$	500	$0.394/6.40E+02$	$0.547/6.40E+02$	$0.48/6.40E+02$	$0.701/6.40E+02$
42	$E3$	1000	$0.909/1.28E+03$	$0.884/1.28E+03$	$0.486/1.28E+03$	$6.4/1.28E+03$
43	$E2$	2	$1.016/3.90E-01$	$0.28/3.90E-01$	F/F	F/F
44	$E2$	500	$4.231/1.94E+02$	F/F	F/F	F/F
45	AR	2	$0.201/2.28E-14$	$0.251/2.12E-14$	$0.397/4.75E-14$	$0.213/1.11E-14$

Table IV. Numerical result of CPU and values of function f contd.

s/n	$Prob.$	$Dim.$	$AOAAH$	DY	LS	$NEW+$
			CPU/FN	CPU/FN	CPU/FN	CPU/FN
46	AR	100	$14.454/9.70E+03$	F/F	F/F	F/F
47	DQ	2	$0.211/0.00E+00$	$0.366/0.00E+00$	$2.224/F$	$0.504/0.00E+00$
48	DQ	100	$4.14/4.12E-14$	$1.25/2.74E-14$	$1.326/4.69E-15$	$6.088/2.21E-15$
49	DQ	500	$2.769/1.51E-13$	$0.546/1.12E-15$	$1.206/2.13E-14$	$0.753/2.37E-14$
50	DQ	1000	$3.37/1.40E-14$	$1.403/7.39E-14$	$0.915/2.18E-15$	$1.001/3.01E-14$
51	DQ	10000	$7.169/3.03E-14$	$3.623/2.24E-14$	$4.034/1.82E-14$	$1.76/5.00E-14$
52	ME	2	$0.179/2.00E+00$	$0.103/2.00E+00$	$0.581/2.00E+00$	$0.412/2.00E+00$
53	ME	100	$0.327/100$	$0.125/100$	$0.824/100$	$0.514/100E$
54	ME	500	$0.322/5.00E+02$	$0.116/5.00E+02$	$0.708/5.00E+02$	$0.488/5.00E+02$
55	ME	1000	$0.344/1.00E+03$	$0.116/1.00E+03$	$0.721/1.00E+03$	$0.534/1.00E+03$
56	ME	10000	$0.52/1.00E+04$	$0.166/1.00E+04$	$1.168/1.00E+04$	$0.703/1.00E+04$
57	ME	50000	$1.17/5.00E+04$	$0.366/5.00E+04$	$2.085/5.00E+04$	$1.343/5.00E+04$
58	ME	100000	$1.875/1.00E+05$	$0.602/1.00E+05$	$3.233/1.00E+05$	$2.34/1.00E+05$
59	CO	2	$0.091/-1.00$	$0.089/-1.00$	$0.19/-1.0$	$0.132/-1.00$
60	CO	100	$0.116/-50.0$	$0.288/-50.0$	$0.32/-50.0$	$0.207/F$
61	CO	500	$3.52/-2.50E+2$	$0.528/-2.50E+2$	$0.29/-2.50E+02$	F/F
62	CO	1000	$3.75/-5.00E+2$	$0.546/-5.00E+2$	$0.34/-5.00E+02$	F/F
63	CO	10000	$10.55/-5.00E+3$	$0.446/-5.00E+3$	$2.65/-5.00E+03$	F/F
64	SI	2	$0.13/-1.00$	$0.145/-1.00$	$0.5/-1.00$	$0.33/-1.00$
65	SI	100	$0.185/-50.0$	$0.201/-50.0$	$0.659/-50.0$	$0.348/F$
66	SI	500	$0.15/-2.50E+2$	$0.173/-2.50E+2$	$0.60/-2.50E+02$	$0.4/-2.50E+2$
67	SI	1000	$0.15/-5.00E+2$	$0.175/-5.00E+2$	$0.63/-5.00E+02$	$0.42/-5.00E+2$
68	SI	10000	$0.23/-5.00E+3$	$0.258/-5.00E+3$	$1.02/-5.00E+03$	$0.62/-5.00E+3$
69	SI	50000	$0.6/-2.50E+04$	$0.579/-2.50E+04$	$1.958/-2.50E+04$	$13.47/F$
70	SI	100000	$0.874/-5.00E+4$	$1.002/-5.00E+04$	$3.182/-5.00E+04$	$15.502/F$
71	EK	2	$0.38/-4.96E-3$	$0.327/-4.96E-3$	F/F	F/F
72	EK	100	$0.279/-95.7$	$0.253/-95.7$	$0.446/-95.7$	$0.835/-97$
73	EK	500	$0.44/-1.24$	$0.385/-1.24$	F/F	F/F
74	EK	1000	$0.46/-2.48$	$0.4/-2.48$	F/F	F/F
75	EK	10000	$0.89/-2.48E+1$	$0.689/-2.48E+1$	F/F	F/F
76	EX	2	$0.158/1.51E-13$	$0.199/2.06E-14$	$0.278/6.90E-15$	$0.128/4.44E-15$
77	EX	100	$0.311/2.06E-15$	$0.243/5.52E-14$	$0.335/1.84E-13$	$0.216/9.22E-15$
78	EX	500	$0.244/8.58E-15$	$0.247/2.13E-14$	$0.342/1.10E-13$	$0.143/1.74E-14$
79	EX	1000	$0.255/1.57E-14$	$0.256/4.26E-14$	$0.384/8.06E-15$	$0.164/5.42E-16$
80	EX	10000	$0.471/2.58E-14$	$0.485/6.55E-07$	$1.003/4.08E-14$	$0.284/5.42E-15$
81	EX	50000	$1.909/5.16E-14$	$1.498/1.67E-14$	$3.64/2.04E-13$	$1.529/5.63E-14$
82	EX	100000	$5.228/1.27E-13$	$2.868/3.35E-14$	$7.654/2.09E-14$	$3.386/6.28E-14$
83	CB	2	$0.011/1.33$	$0.015/1.33$	$0.016/1.33$	$0.034/1.33$
84	CB	100	$0.227/1.33$	$0.0210/1.33$	F/F	$0.214/F$
85	CB	500	$0.209/1.33$	$0.11/1.33$	F/F	$1.007/F$
86	CB	1000	$0.121/1.33$	$0.17/1.33$	F/F	$1.645/F$
87	CB	10000	$0.1/1.33$	$0.2/1.33$	F/F	$20.576/F$
88	QP	2	$0.206/3.63E-02$	$0.036/3.63E-02$	F/F	$0.326/F$
89	PS	2	$0.18/7.73E-01$	$0.211/7.73E-01$	$0.365/7.73E-01$	$0.22/7.73E-01$

Table V. Numerical result of number of iterations and gradient norm.

s/n	$Prob.$	$Dim.$	$AOAAH$	DY	LS	$NEW+$
			ITR/GN	ITR/GN	ITR/GN	ITR/GN
1	EP	2	$23/8.92E-07$	F/F	F/F	F/F
2	QU	2	$19/9.71E-07$	$27/4.80E-07$	$17/3.21E-07$	$18/8.32E-07$
3	ER	100	$335/9.06E-07$	$66/2.75E-07$	$155/7.71E-07$	F/F
4	ER	2	$462/9.96E-07$	$66/3.89E-08$	$145/3.06E-07$	F/F
5	ER	500	$397/5.76E-07$	F/F	$2540/8.65E-07$	F/F
6	ER	1000	$368/8.42E-07$	F/F	$1679/8.87E-07$	F/F
7	DI	2	$18/8.64E-07$	$21/8.78E-07$	$25/8.07E-07$	$19/2.53E-07$
8	DI	100	$20/6.08E-07$	$24/7.01E-07$	$29/2.67E-07$	$59/1.78E-07$
9	DI	500	$21/4.29E-07$	$25/5.27E-07$	$29/5.98E-07$	F/F
10	DI	1000	$21/6.07E-07$	$25/7.46E-07$	$29/8.46E-07$	$25/2.17E-07$
11	DI	10000	$22/6.06E-07$	$27/4.60E-07$	$31/5.79E-07$	F/F
12	DI	50000	$23/4.27E-07$	$28/5.95E-07$	$33/2.80E-07$	$14/8.85E-07$
13	DI	100000	$23/6.04E-07$	$28/8.41E-07$	$33/3.96E-07$	$182/F$
14	HE	2	$40/5.08E-07$	$123/8.44E-07$	$53/6.48E-07$	$38/4.49E-07$
15	HE	100	$43/8.54E-07$	$133/9.83E-07$	$58/3.72E-07$	$30/7.95E-07$
16	HE	500	$46/2.56E-07$	$139/7.63E-07$	$70/3.75E-07$	$42/5.80E-07$
17	HE	1000	$46/3.81E-07$	$142/4.44E-07$	$82/9.63E-07$	$42/8.20E-07$
18	HE	10000	$46/8.45E-07$	$145/6.67E-07$	$100/7.91E-07$	$44/7.40E-07$
19	HE	50000	$47/1.76E-07$	$161/9.18E-07$	$64/3.81E-07$	$210/7.28E-07$
20	HE	100000	$49/8.35E-07$	$155/8.51E-07$	$64/8.03E-07$	$69/4.63E-07$
21	EB	2	$509/9.22E-07$	F/F	$4301/9.76E-07$	F/F
22	EB	500	$183/F$	F/F	F/F	$317/F$
23	EB	1000	$576/8.67E-07$	F/F	F/F	F/F
24	EB	10000	$657/8.52E-07$	F/F	F/F	F/F
25	$RY1$	2	$10/0.00E+00$	$6/0.00E+00$	F/F	$9/F$
26	$RY1$	100	$1/4.74E-120$	$1/4.74E-120$	$1/4.74E-120$	$1/4.74E-120$
27	$RY1$	500	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$
28	$RY1$	1000	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$
29	$RY1$	10000	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$
30	$RY1$	50000	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$
31	$RY1$	100000	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$
32	$RY2$	2	$13/4.24E-07$	$24/6.52E-07$	$9/4.65E-08$	$6/3.13E-09$
33	$RY2$	100	$14/1.87E-07$	$27/8.80E-07$	$9/3.29E-07$	$9/2.69E-09$
34	$RY2$	500	$14/4.18E-07$	$28/9.30E-07$	$9/7.35E-07$	$6/4.95E-08$
35	$RY2$	1000	$14/5.91E-07$	$29/8.35E-07$	$10/1.92E-08$	$6/7.00E-08$
36	$RY2$	10000	$15/3.77E-08$	$31/7.93E-07$	$10/6.06E-08$	$6/2.22E-07$
37	$RY2$	50000	$15/8.42E-08$	$32/8.38E-07$	$10/1.36E-07$	$14/5.72E-12$
38	$RY2$	100000	$15/1.19E-07$	$33/7.53E-07$	$10/1.92E-07$	$31/1.13E-07$
39	$E3$	2	$38/7.95E-07$	$51/8.04E-07$	$35/7.47E-07$	$35/9.77E-07$
40	$E3$	100	$39/6.63E-07$	$60/9.41E-07$	$41/9.52E-07$	$71/6.12E-07$
41	$E3$	500	$46/8.64E-07$	$68/9.77E-07$	$43/5.91E-07$	$81/9.90E-07$
42	$E3$	1000	$105/6.09E-07$	$107/9.86E-07$	$43/4.91E-07$	$778/9.52E-07$
43	$E2$	2	$119/9.85E-07$	$38/8.97E-07$	F/F	F/F
44	$E2$	500	$499/5.95E-07$	F/F	F/F	F/F
45	AR	2	$28/6.51E-07$	$35/5.32E-07$	$44/9.90E-07$	$29/2.99E-07$

Table VI. Numerical result of number of iterations and gradient norm contd.

s/n	$Prob.$	$Dim.$	$AOAAH$	DY	LS	$NEW+$
			ITR/GN	ITR/GN	ITR/GN	ITR/GN
46	AR	100	$1928/9.16E-07$	F/F	F/F	F/F
47	DQ	2	$29/3.64E-07$	$50/6.65E-07$	$263/F$	$69/5.79E-07$
48	DQ	100	$455/8.61E-07$	$143/8.15E-07$	$128/7.17E-07$	$721/9.92E-07$
49	DQ	500	$357/9.86E-07$	$72/2.58E-07$	$118/9.96E-07$	$98/9.10E-07$
50	DQ	1000	$413/9.07E-07$	$175/8.02E-07$	$85/8.39E-07$	$125/4.12E-07$
51	DQ	10000	$426/9.80E-07$	$227/6.72E-07$	$133/6.17E-07$	$110/7.31E-07$
52	ME	2	$25/8.34E-07$	$14/2.90E-07$	$69/9.89E-07$	$57/8.58E-07$
53	ME	100	$38/9.64E-07$	$16/2.79E-07$	$80/9.41E-07$	$63/8.66E-07$
54	ME	500	$45/9.61E-07$	$16/6.25E-07$	$85/8.46E-07$	$69/9.33E-07$
55	ME	1000	$48/9.68E-07$	$16/8.84E-07$	$86/9.97E-07$	$71/8.44E-07$
56	ME	10000	$57/9.43E-07$	$18/3.80E-07$	$93/8.80E-07$	$76/8.75E-07$
57	ME	50000	$60/8.22E-07$	$18/8.50E-07$	$97/9.49E-07$	$68/8.12E-07$
58	ME	100000	$61/9.47E-07$	$19/9.09E-07$	$99/9.32E-07$	$69/8.52E-07$
59	CO	2	$12/2.70E-07$	$12/1.07E-07$	$21/2.47E-07$	$17/7.23E-07$
60	CO	100	$13/4.88E-07$	$12/7.55E-07$	$23/5.10E-07$	$17/F$
61	CO	500	$448/1.04E-08$	$72/6.62E-08$	$29/7.34E-07$	F/F
62	CO	1000	$448/1.47E-08$	$72/9.36E-08$	$31/8.11E-07$	F/F
63	CO	10000	$561/9.19E-08$	$36/5.81E-07$	$113/6.60E-07$	F/F
64	SI	2	$18/6.13E-07$	$20/5.28E-07$	$59/8.09E-07$	$47/8.03E-07$
65	SI	100	$20/4.98E-07$	$22/8.70E-07$	$67/9.60E-07$	$30/F$
66	SI	500	$21/3.77E-07$	$24/4.54E-07$	$71/8.79E-07$	$56/9.53E-07$
67	SI	1000	$21/5.33E-07$	$24/6.41E-07$	$72/9.95E-07$	$58/7.58E-07$
68	SI	10000	$22/5.71E-07$	$26/4.73E-07$	$78/8.24E-07$	$62/7.58E-07$
69	SI	50000	$37/5.33E-07$	$36/6.43E-07$	$36/9.58E-07$	$29/7.58E-07$
70	SI	100000	$52/8.30E-07$	$36/9.09E-07$	$37/8.63E-07$	$30/5.60E-07$
71	EK	2	$52/6.11E-07$	$46/7.05E-07$	F/F	F/F
72	EK	100	$32/1.31E-07$	$26/5.41E-07$	$41/3.18E-07$	$96/8.86E-07$
73	EK	500	$59/2.87E-07$	$53/5.85E-07$	F/F	F/F
74	EK	1000	$59/4.03E-07$	$53/8.27E-07$	F/F	F/F
75	EK	10000	$61/9.32E-08$	$58/4.40E-07$	F/F	F/F
76	EX	2	$21/9.15E-07$	$27/8.62E-07$	$28/4.85E-07$	$17/3.98E-07$
77	EX	100	$31/2.50E-07$	$30/9.74E-07$	$29/9.45E-07$	$20/4.54E-07$
78	EX	500	$32/2.34E-07$	$32/4.63E-07$	$34?7.34E-07$	$19/7.86E-07$
79	EX	1000	$32/3.17E-07$	$32/6.55E-07$	$35/3.95E-07$	$21/1.39E-07$
80	EX	10000	$32/4.63E-07$	$34/9.83E-07$	$36/4.29E-07$	$21/4.39E-07$
81	EX	50000	$23/4.33E-07$	$27/5.64E-07$	$81/9.44E-07$	$37/F$
82	EX	100000	$23/6.12E-07$	$27/7.97E-07$	$83/8.54E-07$	$29/F$
83	CB	2	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$	$1/0.00E+00$
84	CB	100	$27/7.14E-07$	$2/2.68E-16$	F/F	$24/F$
85	CB	500	$26/8.49E-07$	$2/3.16E-15$	F/F	$112/F$
86	CB	1000	$16/9.14E-07$	$2/2.36E-15$	F/F	$212/F$
87	CB	10000	$2/6.41E-08$	$2/1.98E-15$	F/F	$2075/F$
88	QP	2	$24/9.44E-07$	$2/3.43E-16$	F/F	$15/F$
89	PS	2	$22/5.61E-07$	$26/6.70E-07$	$34/6.74E-07$	$25/8.89E-07$

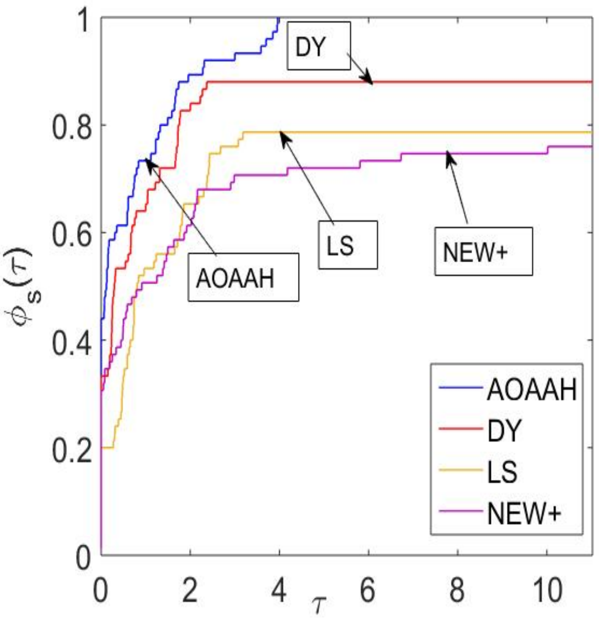

Fig. 1: Number of iterations (ITR)

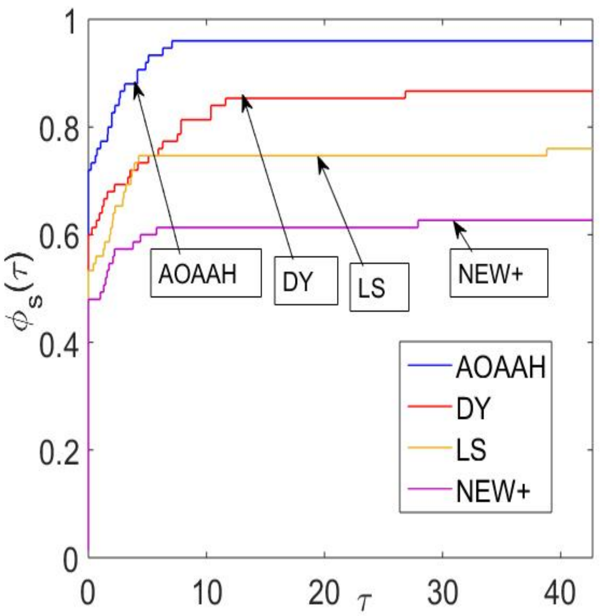

Fig. 2: Value of function f (FV)

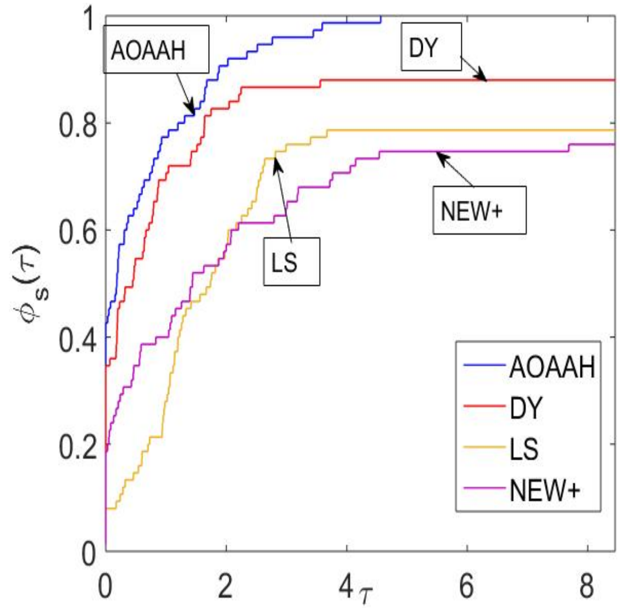

Fig. 3: CPU time

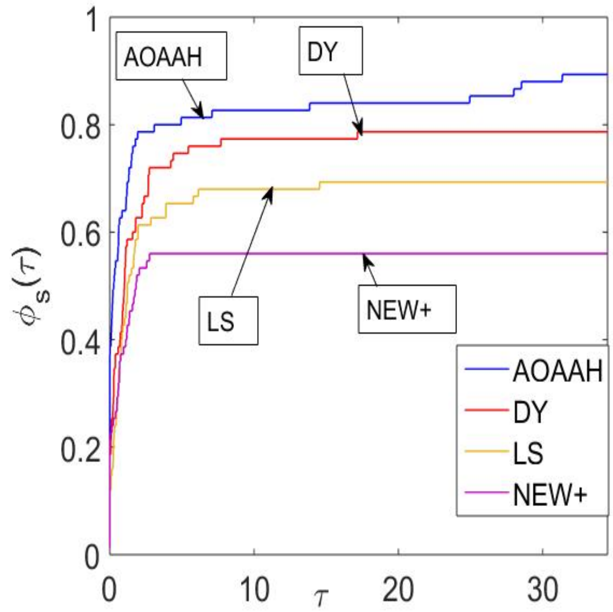

Fig. 4: Gradient norm (GN)

Test results in Tables III to VI generated from test problems in Tables I and II report the performance profiles of AOAAH-CG method against DY-CG [7], LS-CG [16] and NEW+-CG [24] methods respectively. Parameters used to determine the numerical strength of AOAAH-CG method against these CG methods were the number of iterations (ITR), the value of function (FN), the CPU time (CPU) and gradient norm (GN) respectively. Dimensions: 2, 100, 500, 1000 and 10000, 50000, 100000 were considered to report results in Tables III to VI. F in the tables stands for failure of an algorithm to solve a problem. Graphical representations showing the numerical strength of all the methods were generated from the drawing tools of Dolan and More [10]: Given set S of t_s methods to be compared, let R be the set of n_r test functions. The tools are implemented with the fact that $U_{r,s}$ can either be ITR, F, GN or CPU time for each method S and problem R with different methods being compared by using the ratio

$$u_{r,s} = \frac{U_{r,s}}{min\{U_{r,s} : s \in S \ and \ r \in F\}}. \tag{66}$$

However, the overall distribution function for $u_{r,s}$ is defined by

$$\phi_s(\tau) = \frac{1}{n_r}|R \in R : logu_{r,s}| \leq \tau \tag{67}$$

where $\tau \geq 0$. The probability that $u_{r,s}$ is within a factor $\tau \geq 1$ in relation to the method s is $\phi_s(\tau)$. So also, the probability that one method will outperform other methods

is $\phi_s(\tau)$ for the value $\tau = 1$. If $u_i = u_{r,s}$ for some parameter u_i, the chosen method $s \in S$ will fail to solve a problem.
Figures (1-4) highlight the profile strengths of the four methods tested based on number of iterations (ITR), values of function (FN), CPU time (CPU) and gradient norm (GN) respectively. Figure 1 shows that the proposed algorithm AOAAH is more robust than the other three algorithms in the areas of number of iterations. Figure 2 indicates that AOAAH algorithm is the top performer in term of value of functions followed by DY method while the worst performer is the NEW+ algorithm. In figure 3, AOAAH method solves more problems than other algorithms in term of CPU time. Figure 4 shows that AOAAH method outperformed the other three methods in the area of gradient norm. The overall performance of the four algorithms tested shows that AOAAH method is more robust and effective than the other three methods.

V. Conclusions

A new descent CG method was proposed in this paper as a result of wide acceptability of CG methods in solving any problem of the form (1). The new method was constructed with the inclusion of constants to give a hybrid CG method which is a linear combination of known classical methods. This method was tested based on the Wolfe line search scheme. Descent and convergence of the new method were established. Numerical strength of the method showed that it is promising and solved problems better than the chosen existing methods. For further studies, constants a, b, c, e, f can be varied to achieve different descent conjugate gradient methods.

References

[1] Abubakar, A. B., Kumam, P., Malik, M., and Ibrahim, A. H. A hybrid conjugate gradient based approach for solving unconstrained optimization and motion control problems. *Mathematics and Computers in Simulation,* 201, 640-657, 2022.

[2] Abubakar, A. B., Malik, M., Kumam, P., Mohammad, H., Sun, M., Ibrahim, A. H., and Kiri, A. I. . A Liu-Storey-type conjugate gradient method for unconstrained minimization problem with application in motion control. *Journal of King Saud University-Science*, 34(4), 101923, 2022.

[3] Andrei N., 2011. Open Problems in Nonlinear Conjugate Gradient Algorithms for Unconstrained Optimization,. *Bulletin of the Malaysian Mathematical Sciences Society* vol. 34, no. 2, pp. 319-330, 2011.

[4] Ayinde, S. A., Agboola, S. O., Adelodun, J. F. and Hassan, S. O. Two new conjugate gradient methods in unconstrained optimization problems. *Annals of Mathematics and Computer Science,* vol. 18, pp. 26-39, 2023.

[5] Ayinde, S. A., Osinuga, I. A., Adio, A. K. and Adelodun, J. F. A new Dai-Liao-type conjugate gradient method for unconstrained optimization problems. *Engineering Letters,* vol. 31(3), 1281-1290, 2023.

[6] Dai, Y. and Liao, L. New conjugacy conditions and related nonlinear conjugate gradient methods, *Applied Mathematics and Optimization*. vol. 43, pp. 87-101, 2001.

[7] Dai, Y. and Yuan, Y. A nonlinear conjugate gradient method with a strong global convergence property, *SIAM Journal of Optimization,* vol. 10, pp. 177-182, 1999.

[8] Dai, Y. and Yuan, Y. An efficient hybrid conjugate gradient method for unconstrained optimization, *Annals of Operations Research,* vol.103, pp.33-47, 2001.

[9] Deepho, J., Abubakar, A. B., Malik, M., and Argyros, I. K. Solving unconstrained optimization problems via hybrid CD-DY conjugate gradient methods with applications. *Journal of Computational and Applied Mathematics,* 405,113823, 2022.

[10] Dolan, E. D. and More, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming,* vol. 91, pp. 201-213, 2002.

[11] Fletcher, R. and Reeves, C. M. Function minimization by conjugate gradients. *Computer Journal*, vol. 7, pp. 149-154, 1964.

[12] Hestenes, M. R. and Stiefel, E. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards,* vol. 49, pp. 409-435, 1952.

[13] Hu Y. and Storey C., Global convergence result for conjugate gradient methods. *Journal of Optimization Theory and Applications,* vol. 71, pp. 399-405, 1991.

[14] Jiang, X. Z., Yang, H. H., Yin, J. H., Liao, W. A three-term conjugate gradient algorithm with restart procedure to solve image restoration problems, *J. Computational. Applied Mathematics,* 424, 115020, 2023.

[15] Jie, G. and Zhong W., A new three-term conjugate gradient algorithm with modified gradient-differences for solving unconstrained optimization problems, *AIMS Mathematics,* 8(2), 2473-2488, 2023.

[16] Liu, Y. and Storey, C., Efficient generalized conjugate gradient algorithms, part 1: Theory, *Journal of Optimization Theory and Applications,* 69, 129-137, 1991.

[17] Lu, J., Yuan, G. and Wang, Z., "A modified Dai-Liao conjugate gradient method for solving unconstrained optimization and image restoration problems", *Journal of Applied Mathematics and Computing,* vol. 68, pp. 681-703, 2022.

[18] Ma, G., Lin, H., Jin, W., Han, D. Two modified conjugate gradient methods for unconstrained optimization with applications in image restoration problems. *Journal of Applied Mathematics and Computing,* 68, 4733-4758, 2022.

[19] Malik, M., Sulaiman, I. M., Abubakar, A. B., and Ardaneswari, G. A new family of hybrid three-term conjugate gradient method for unconstrained optimization with application to image restoration and portfolio selection, *AIMS Mathematics,* 8(1), 1-28, 2023.

[20] Polak, E. and Ribiere, G. Note sur la convergence de directions conjugate, *Analysis- Mode 'elisation Math 'ematique et Analyse Num 'erique,* vol. 3, no. R1, 35-43, 1969.

[21] Polyak, B., The conjugate gradient method in extremal problems, *USSR Computational Mathematics and Mathematical Physics*, vol. 9, pp. 94-112, 1969.

[22] Stanimirovic, P. S., Ivanov, B. and Mosic, D. A survey of gradient methods for solving nonlinear optimization. *Electronic Research Archive,* vol. 28, no. 4, pp. 1573-1624, 2020.

[23] Xu, X. and Kong, F. New hybrid conjugate gradient methods with the generalized Wolfe line search. *Springer Plus,* vol.5, 10.1186/s40064-016-2522-9, 2016.

[24] Zheng, Y. A family of Dai-Liao conjugate gradient methods with modified secant equation for unconstrained optimization, *RAIRO Operations Research,* vol. 55, pp. 3281-3291, 2021.

[25] Zheng, Y. and Zheng, B. Two new Dai-Liao-type conjugate gradient methods for unconstrained optimization problems, *Journal of Optimization Theory and Applications,* 175, pp 502-509, 2017.

[26] Zoutendijk, G., Nonlinear programming computational methods, in Integer and Nonlinear Programming. *J. Abadie (Ed.), Linear and Nonlinear Programming North-Holland, Amsterdam*, pp. 37-86, 1970.