Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 8 (2026) 3118

Journal of the Nigerian Society of Physical Sciences

A descent-safeguarded PRP-type conjugate gradient with global convergence and CUTEst benchmarks

Ahmad Alhawarat[a], Sultanah Masmali[b], Shahrina Ismail[c,*], Hamid El Hor[d]

[a]Department of Mathematics, Faculty of Arts and Science, Amman Arab University, Amman 11953, Jordan
[b]Department of Mathematics, College of Science, Jazan University, Jazan, Saudi Arabia
[c]Financial Mathematics Program, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia
[d]Sidel-Simulations, D-63303 Dreieich, Germany

Abstract

Conjugate gradient (CG) methods are widely used for solving large-scale unconstrained optimization problems. Well-known methods, such as the Polak–Ribière–Polyak and Hestenes–Stiefel methods, may not satisfy the global convergence property. To improve this behavior, this paper constructs a new three-term CG method based on the PRP framework. The proposed method is shown to satisfy sufficient descent and global convergence properties. To study its behavior, we compare its performance with those of CG-Descent 6.8, the non-negative Dai–Liao method, and the HS+TA method by applying them to more than 180 optimization problems selected from the CUTEst library. The numerical results show that the new method performs better than the three competing methods and other recently published CG methods in terms of the number of iterations, the number of function and gradient evaluations, and the CPU time required to solve the test problems. To illustrate accuracy, we report the function and gradient values at the obtained solutions for all test problems.

DOI:10.46481/jnsps.2026.3118

Keywords: Conjugate gradient method, Global convergence, CUTEst benchmarks, Unconstrained optimization.

Article History:
Received: 08 September 2026
Received in revised form: 26 April 2026
Accepted for publication: 13 May 2026
Available online: 25 May 2026

Communicated by: P. Thakur

1. Introduction

Optimization problems of large dimension, smoothness, and absence of explicit constraints arise frequently in applied mathematics, engineering, and data science. Among the iterative methods designed to address such problems, nonlinear conjugate gradient (CG) methods remain particularly attractive due to their low memory requirements and computational efficiency. These methods rely only on function and gradient evaluations, making them well suited for large-scale applications.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, and consider the unconstrained optimization problem

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

A typical nonlinear CG method generates iterates according to

$$x_{k+1} = x_k + \alpha_k d_k, \quad d_k = -g_k + \beta_k d_{k-1}, \tag{2}$$

where $g_k = \nabla f(x_k)$, $d_0 = -g_0$, and $\alpha_k > 0$ is obtained by a line-search procedure.

The performance of CG methods depends critically on the choice of the parameter β_k. Classical formulas such as the Hestenes–Stiefel (HS) method [1], the Polak–Ribière–Polyak

*Corresponding Author Tel. No.: +60 103-698-676.
Email address: shahrinaismail@usim.edu.my (Shahrina Ismail)

(PRP) method [2], and the Fletcher–Reeves (FR) method [3] provide the foundational framework for nonlinear CG algorithms. While PRP and HS are known for their practical efficiency, they may fail to guarantee sufficient descent in certain situations. In contrast, the FR method ensures descent under strong Wolfe conditions but may exhibit slower convergence on ill-conditioned problems.

To address these limitations, considerable research has focused on improving the stability and convergence properties of nonlinear CG methods. Theoretical developments based on Wolfe and strong Wolfe line-search conditions [4, 5] provide a fundamental framework for ensuring descent and global convergence under inexact line searches. These results, together with classical convergence analyses such as those of Zoutendijk [6] and Gilbert and Nocedal [7], establish the basis for analysing CG-type methods under standard boundedness and Lipschitz continuity assumptions.

Beyond these foundational results, several modifications have been proposed to enhance the robustness and efficiency of CG methods. The Dai–Liao conjugacy condition and its variants [8–11] introduce alternative mechanisms for preserving conjugacy while improving numerical stability. More recent developments include modified PRP-type methods, hybrid CG approaches, and multi-term (three-term and four-term) formulations [12–19], which aim to combine fast convergence with guaranteed descent properties. In addition, adaptive-parameter strategies and matrix-free CG techniques [20, 21] have been introduced to improve the practical performance of CG methods in large-scale settings. Applications of such methods to problems in engineering, including electric circuits and representative application scenarios such as image restoration, further highlight their practical relevance [22].

Despite these advances, maintaining a balance between simplicity, robustness, and convergence guarantees remains a central challenge. Many existing modifications introduce additional parameters or complex update rules, which may increase implementation difficulty or sensitivity to problem structure. In particular, PRP-type methods, while efficient, may generate non-descent directions when curvature information is unfavorable.

Motivated by these considerations, this paper proposes a descent-safeguarded PRP-type conjugate gradient method that preserves the simplicity of classical CG schemes while ensuring sufficient descent under standard line-search conditions. The proposed method modifies the PRP update by filtering unfavorable curvature components, thereby preventing loss of descent without introducing additional tuning parameters.

The contributions of this work can be summarized as follows:

- A new three-term PRP-type CG update that ensures sufficient descent;
- A global convergence analysis under standard Wolfe-type line-search assumptions;
- A comprehensive numerical evaluation on CUTEst benchmark problems, using performance profiles to compare with established CG methods.

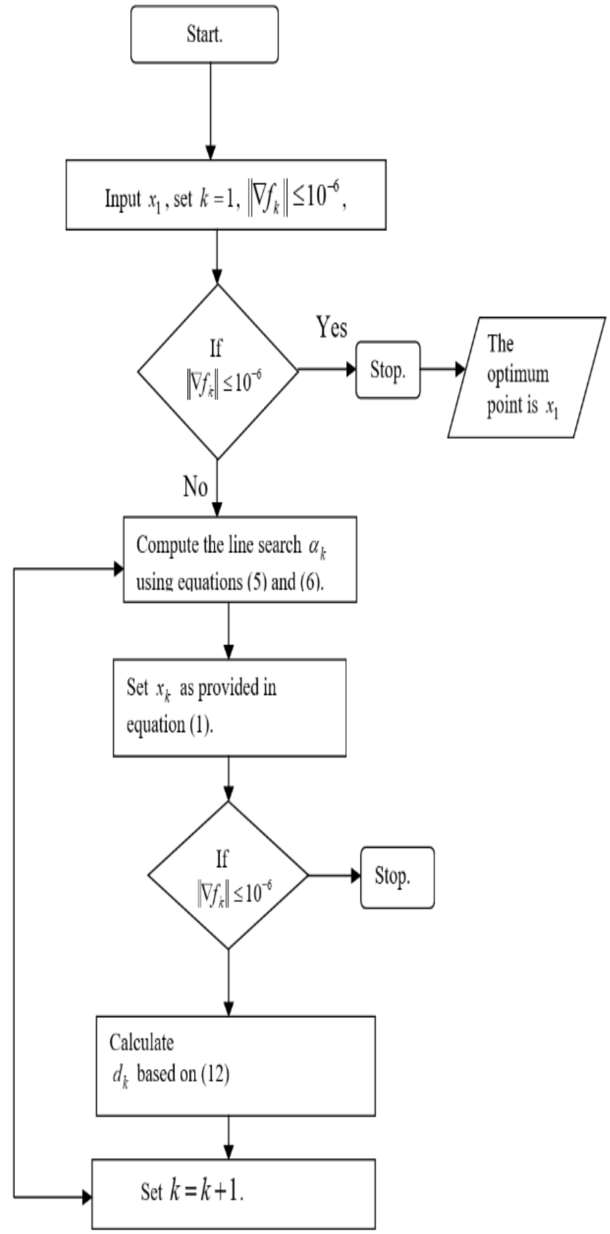

Figure 1: Flowchart of the proposed PRP-type conjugate gradient algorithm (AHPRP, “Algorithm 1”). The diagram illustrates initialization, update rules, line search procedure, and termination criteria.

The proposed method is therefore positioned within three main strands of the literature: classical CG methods, convergence theory under Wolfe-type line searches, and recent safeguarded or modified CG schemes. The references cited in this work are integrated to support these aspects, providing both theoretical justification and a basis for numerical comparison.

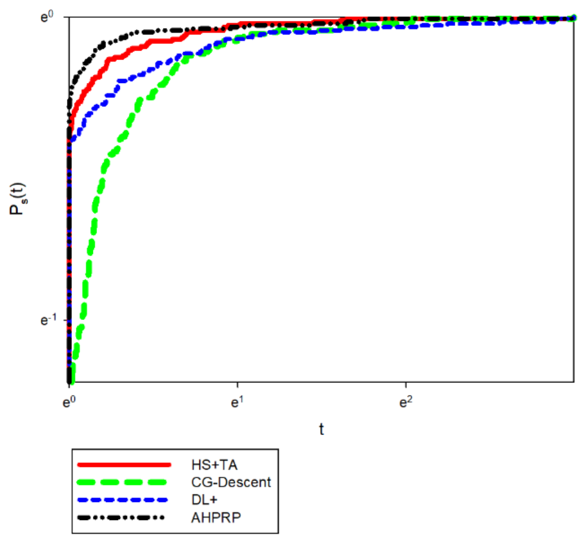

Figure 2: Comparison of the number of iterations required to solve more than 180 CUTEst test problems using AHPRP, CG-Descent 6.8, Dai–Liao (DL) [8], and HS+TA methods. Lower values indicate faster convergence.

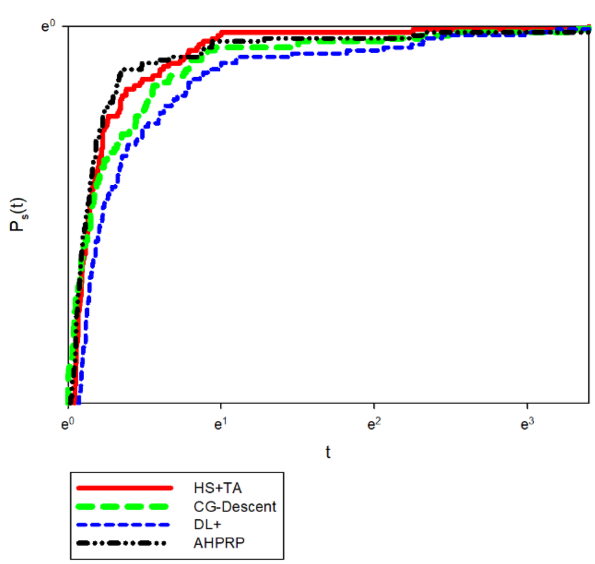

Figure 3: Comparison of the number of function evaluations across the CUTEst benchmark set for AHPRP, CG-Descent 6.8, Dai–Liao (DL) [8], and HS+TA methods. The proposed method consistently reduces costly function calls.

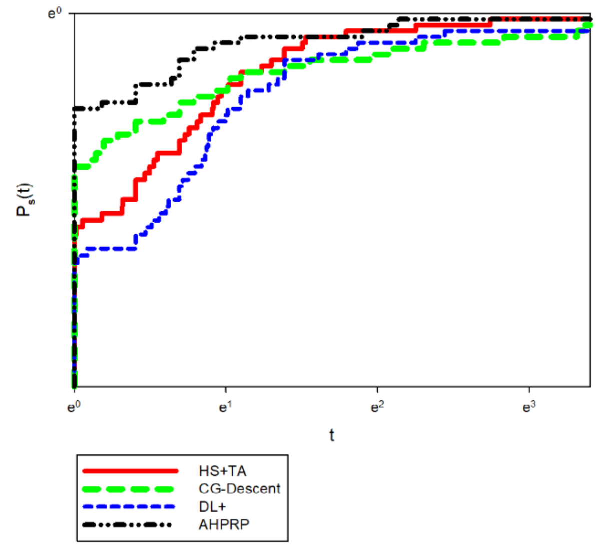

Figure 4: Comparison of CPU time (in seconds) required by AHPRP, CG-Descent 6.8, Dai–Liao (DL) [8], and HS+TA methods over the CUTEst problems. Results highlight the computational efficiency of the proposed method.

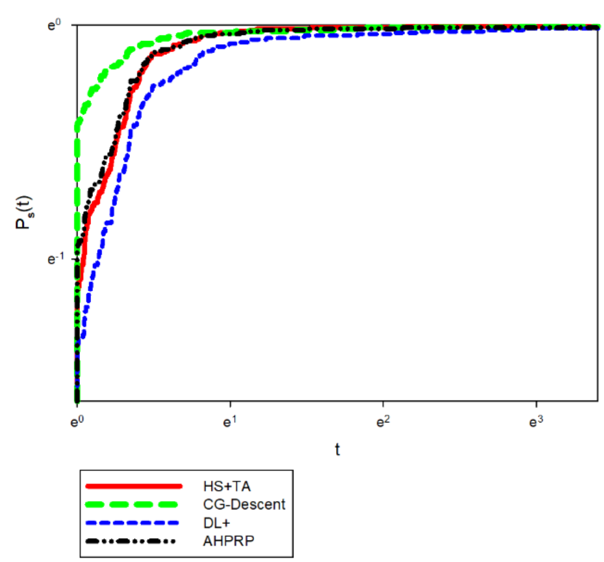

Figure 5: Comparison of the number of gradient evaluations for AHPRP and competing CG methods on the CUTEst test set. AHPRP achieves convergence with fewer gradient computations, demonstrating robustness in large-scale settings.

Table 1: Numerical results for the AHPRP method on the CUTEst test functions.

Function	No. of iterations	Function evaluations	Gradient evaluations	Function value	Gradient value	CPU time (s)
AKIVA	8	20	15	6.166042	0.0000000697	0.02
ALLINITU	9	25	18	5.744385	1.61E-09	0.02
ARGLINA	1	3	2	200	1.24E-11	0.02
ARWHEAD	6	16	12	0	0	0.03
BARD	12	32	22	0.008215	0.0000000324	0.02
BDQRTIC	150	324	299	20006.26	0.0000005648	0.62
BEALE	11	33	26	1.66E-19	2.57E-09	0.02
BENNETT5LS	14	44	34	0.000359	6.73E-08	0.02
BIGGS6	24	64	44	0.005656	0.000000023	0.02
BOX	7	25	21	-1864.54	0.0000000717	0.09
BOX3	10	23	14	6.37E-15	9.22E-08	0.02
BOXBODLS	12	75	69	1168.009	2.64E-08	0.02
BOXPOWER	29	71	45	1.53E-12	0.0000000536	0.16
BRKMCC	5	10	1	0.169043	6.22E-08	0.02
BROWNAL	9	26	20	1.29E-19	9.2E-11	0.02
BROWNB	10	24	18	0	0	0.02
BROWNDEN	16	38	31	85822.2	1.53E-10	0.02
BROYDN7D	57	105	78	3441.346	0.000000085	0.28
BRYBND	75	172	107	8.85E-14	0.000000050	0.36
CHAINWOO	13194	27495	14635	1724.686	0.0000000896	22.17
CHWIRUT1LS	15	43	34	2380	0.0000000318	0.02
CHWIRUT2LS	15	35	25	513	0.0000000358	0.02
CLIFF	10	46	39	0.199787	0.000001095	0.02
COSINE	11	51	42	-9999	0.000000043	0.17
CRAGGLVY	112	226	184	1688.215	0.000000069	0.49
CUBE	17	48	34	1.65E-20	5.82E-09	0.02
CURLY10	55146	75014	90458	-1003163	0.0000000949	199.89
CURLY20	72593	95575	122254	-1003163	0.000000999	413.44
CURLY30	79050	101368	134142	-1003163	0.000000987	623.52
DANWOODLS	8	32	28	0.004317	9.93E-08	0.02
DECONVU	408	821	414	2.24E-08	0.0000000973	0.03
DENSCHNA	6	16	12	1.32E-14	0.000000032	0.02
DENSCHNB	6	18	15	3.2E-19	1.43E-09	0.02
DENSCHNC	11	36	31	5.27E-15	0.0000000276	0.02
DENSCHND	14	46	40	5.24E-12	0.0000000234	0.02
DENSCHNE	12	43	38	1.46E-13	0.000000075	0.02
DENSCHNF	9	31	26	3.18E-22	2.52E-10	0.02
DIXMAANA	3	6	1	3.94E-13		0.02
DIXMAANB	6	16	13	1	2.61E-08	0.02
DIXMAANC	6	14	9	1	2.06E-08	0.02
DIXMAAND	7	19	12	1	0.000000002	0.02
DIXMAANE	253	283	486	1	0.00000009	0.28
DIXMAANF	141	287	149	1	0.00000094	0.11
DIXMAANG	172	349	180	1	0.00000098	0.16
DIXMAANH	173	353	184	1	0.00000089	0.14
DIXMAANI	3290	3380	6490	1	0.000000098	3.5
DIXON3DQ	10000	10000	20000	3.16E-13	7.17E-08	20.3
DJTL	75	1160	1150	-8951.55	0.000000031	0.02
DQDRTIC	5	11	6	4.67E-17	3.65E-08	0.02
DQRTIC	20	57	45	0.00000118	0.000000044	0.02
ECKERLE4LS	2	6	4	0.699696	0.0000000387	0.02
EDENSCH	26	57	50	12003.28	0.0000000577	0.03
EG2	4	20	18	-998.947	7.35E-14	0.03
EIGENALS	7870	13900	9720	2.63E-10	0.0000000984	139.08
EIGENBLS	14397	28505	14412	0.00000192	0.000000209	209
EIGENCLS	10078	19385	10876	2.31E-11	0.0000000694	162.56
ENGVAL1	18	39	31	5548.668	0.0000000518	0.05
ENGVAL2	26	73	55	3.27E-23	6.59E-10	0.02
ENSOLS	22	47	27	788.5398	0.0000000355	0.02
EXPFIT	9	29	22	0.240511	0.0000000332	0.02
EXTROSNB	2063	4728	2849	0.000000381	0.0000000766	0.7

Table 1 (continued)

Function	No. of iterations	Function evaluations	Gradient evaluations	Function value	Gradient value	CPU time (s)
FBRAIN2LS	79	259	204	0.318972	0.000000021	0.48
FBRAIN3LS	1310	3930	3080	0.242722	0.0000000973	10.42
FBRAIN5LS	9	27	21	0.416603	0.0000000737	0.02
FLETCHCR	268	530	297	1.06E-14	0.0000000859	0.09
FMINSRF2	287	576	290	1.000024	0.000000098	0.78
FMINSURF	411	824	414	1	0.000000094	1.13
FREUROTH	26	60	51	608159.2	0.000000081	0.12
GAUSSILS	49	113	74	1315.822	8.44E-09	0
GAUSS2LS	49	124	87	1247.528	4.65E-09	0.02
GBRAINLS	8	20	13	28.51586	4.33E-08	0.02
GENROSE	1127	2299	1184	1	0.0000001169	0.19
GROWTHLS	109	431	369	1.004041	0.000000068	0.32
GULF	33	95	72	1.15E-17	3.18E-08	0.02
HAHN1LS	5	56	53	8522.662	4.03E-08	0.02
HAIRY	17	82	68	20	7.89E-09	0.02
HATFLD	17	49	37	0.0000000255	0.00000110	0.02
HATFLEA	13	37	30	0.00000273	0.000000033	0.02
HATFLDFL	21	68	54	0.0000639	0.0000000984	0.02
HEART6LS	375	1137	876	8.8E-17	0.0000000223	0.2
HEART8LS	253	657	440	2.97E-16	0.0000000323	0.02
HELIX	23	60	42	2.88E-19	0.0000000013	0.02
HIELOW	13	30	21	874.1654	5.19E-08	0.05
HILBERTA	2	5	3	5.75E-32	2.22E-16	0.02
HILBERTB	2	5	3	9.95E-19	1.22E-15	0.02
HIMMELBB	4	18	18	2.17E-16	4.97E-09	0.02
HIMMELBF	4	59	46	318.5717	0.0000000432	0.03
HIMMELBG	7	22	17	3.04E-22	4.92E-11	0.02
HIMMELBH	5	13	9	-1	0.0000000277	0.02
HUMPS	45	223	202	9.15E-16	1.35E-08	0.02
INDEF	1	46	147	3.184123e-314	3.184008e-314	0.33
INTEQNELS	8	20	15	6.166042	0.0000000697	0.02
JENSMP	9	25	18	5.744385	1.61E-09	0.02
JIMACK	1	3	2	200	1.24E-11	0.02
KIRBY2LS	6	16	12	0	0.0000000184	0.03
KOWOSB	12	32	22	0.008215	0.0000000328	0.02
LANCZOSILS	150	324	299	20006.26	0.0000005648	0.62
LANCZOS2LS	11	33	26	1.66E-19	2.57E-09	0.02
LANCZOS3LS	14	44	34	0.000359	6.73E-08	0.02
LOGHAIRY	24	64	44	0.005656	0.000000023	0.02
LSC1LS	7	25	21	-1864.54	0.0000000717	0.09
LSC2LS	10	23	14	6.37E-15	9.22E-08	0.02
MANCINO	12	75	69	1168.009	2.64E-08	0.02
MARATOSB	29	71	45	1.53E-12	0.000000053	0.16
MEXHAT	9	11	6	0.169043	6.22E-08	0.02
MEYERA3	9	26	20	1.29E-19	9.2E-11	0.02
MGH09LS	10	24	18	0	0	0.02
MGH10SLS	16	38	31	85822.2	1.53E-10	0.02
MGH17LS	57	105	78	3441.346	0.0000000915	0.28
MISRA1ALS	75	172	107	8.85E-14	0.0000000507	0.36
MISRA1BLS	13194	27495	14635	1724.686	0.0000000896	22.17
MOREBV	15	43	34	2380	0.0000000318	0.02
MSQRTALS	15	35	25	513	0.0000000358	0.02
MSQRTBLS	10	46	39	0.199787	0.000001095	0.02
NCB20	11	51	42	-9999	0.0000000437	0.17
NELSONLS	112	226	184	1688.215	0.0000000695	0.49
NONCVXU2	17	48	34	1.65E-20	5.82E-09	0.02
NONDQUAR	55146	75014	90458	-1003163	0.0000000949	199.89
OSBORNEA	72593	95575	122254	-1003163	0.0000009999	413.44
OSBORNEB	79050	101368	134142	-1003163	0.0000009878	623.52
OSCIGRAD	8	32	28	0.004317	9.93E-08	0.02
OSCIPATH	408	821	414	2.24E-08	0.0000000973	0.03
PALMER1C	6	16	12	1.32E-14	0.0000000321	0.02

Table 1 (continued)

Function	No. of iterations	Function evaluations	Gradient evaluations	Function value	Gradient value	CPU time (s)
PALMER1D	6	18	15	3.2E-19	1.43E-09	0.02
PALMER2C	11	36	31	5.27E-15	0.0000000276	0.02
PALMER3C	14	46	40	5.24E-12	0.0000000230	0.02
PALMER4C	12	43	38	1.46E-13	0.000000075	0.02
PALMER5C	9	31	26	3.18E-22	2.52E-10	0.02
PALMER6C	6	15	1	3.94E-13		0.02
PALMER7C	6	16	13	1	2.61E-08	0.02
PALMER8C	6	14	9	1	2.06E-08	0.02
PARCH	7	17	12	1	0.0000000111	0.02
PENALTY2	253	283	486	1	0.00000009	0.28
PENALTY3	141	287	149	1	0.000000094	0.11
POWELL1BSLS	172	349	180	1	0.000000098	0.16
POWER	173	353	184	1	0.0000000989	0.14
POWERSUM	3290	3380	6490	1	0.0000000908	3.5
PRICE3	10000	10000	20000	3.16E-13	7.17E-08	20.3
PRICE4	75	1160	1150	-8951.55	0.000000031	0.02
QING	5	11	6	4.67E-17	3.65E-08	0.02
QUARTIC	20	57	45	0.00000118	0.0000000447	0.05
RAT42LS	2	5	4	0.699696	0.0000000381	0.02
RAT43LS	26	57	50	12003.28	0.0000000577	0.03
RECIPELS	4	20	18	-998.947	7.35E-14	0.03
ROSENBR	7870	13900	9720	2.63E-10	0.0000000984	139.08
ROSENBRTU	14397	28505	14412	0.00000192	0.0000002098	209
ROSZMANN1LS	10078	19385	10876	2.31E-11	0.0000000694	162.56
S308	18	39	31	5548.668	0.0000000518	0.05
SCHMVETT	26	73	55	3.27E-23	6.59E-10	0.02
SENSORS	22	47	27	788.5398	0.0000000355	0.02
SINEVAL	9	29	22	0.240511	0.0000000332	0.02
SINQUAD	2063	4728	2849	0.000000381	0.0000000766	0.7
SISSER	79	259	204	0.318972	0.000000021	0.48
SNAIL	1310	3930	3080	0.242722	0.0000000977	10.42
SPARSINE	9	27	21	0.416603	0.0000000337	0.03
SPARSQUR	268	530	297	1.06E-14	0.0000000859	0.09
SPMSRTLS	287	576	290	1.000024	0.000000098	0.78
SROSENBR	411	824	414	1	0.0000000945	1.13
SSBYBND	26	60	51	608159.2	0.000000081	0.12
SSI	49	113	74	1315.822	8.44E-09	0.02
STRATEC	49	124	87	1247.528	4.65E-09	0.02
TESTQUAD	8	20	13	28.51586	4.33E-08	0.02
TOINTGOR	1127	2299	1184	1	0.0000000615	0.19
TOINTGSS	109	431	369	1.004041	0.0000000687	0.02
TOINTPSP	33	95	72	1.15E-17	3.18E-08	0.02
TOINTQOR	5	56	53	8522.662	4.03E-08	0.02
TQUARTIC	17	82	68	20	7.89E-09	0.02
TRIDIA	17	49	37	0.0000002559	0.0000000138	0.02
TRIGON1	13	37	30	0.00000273	0.0000000338	0.02
TRIGON2	21	68	54	0.00000639	0.0000000984	0.02
VANDANMSLS	375	1137	876	8.8E-17	0.0000000223	0.02
VAREIGVL	253	657	440	2.97E-16	0.0000000232	0.02
VESUVIALS	23	60	42	2.88E-19	0.000000003	0.02
VESUVIOLS	13	30	21	874.1654	5.19E-08	0.05
VESUVIOULS	2	5	3	5.75E-32	2.22E-16	0.02
VIRBBEAM	4	9	5	9.95E-19	2.27E-09	0.02
WAYSEA1	6	18	12	2.17E-16	4.97E-09	0.02
WAYSEA2	23	59	46	318.5717	0.0000000439	0.02
WATSON	7	22	17	3.04E-22	4.92E-11	0.02
WOODS	13	9	9	-1	0.0000000277	0.02
YATPILS	45	223	202	9.15E-16	1.35E-08	0.02
YFITU	1	46	147	3.184123e-314	3.184008e-314	0.33
ZANGWIL2	8	20	15	6.166042	0.0000000697	0.02

2. Proposed CG method and its motivation

This section presents the proposed update inside the classical conjugate-gradient (CG) scheme and the intuition behind it. We keep the algorithmic skeleton unchanged and modify only the mixing parameter that links the current steepest–descent step to the previous direction. The aim is to preserve CG's speed on well-behaved problems while avoiding unreliable steps when curvature information is noisy.

With $d_k = -g_k + \beta_k d_{k-1}$, difficulties with PRP/HS-type rules appear when $y_{k-1} = g_k - g_{k-1}$ contains components that steer the search uphill. Our update selectively attenuates these components while retaining the helpful conjugacy information [9]. In effect, the method behaves like PRP when the geometry is favorable and applies a light safeguard only when needed; thus, it reduces the incidence of non-descent directions without adding tunable parameters.

Implementation is straightforward. The line search remains the standard strong-Wolfe procedure; no second-order data are required; and storage is identical to vanilla CG. The extra quantities needed by the safeguard coincide with inner products already available in typical implementations. Consequently, the per-iteration cost is essentially unchanged.

The design follows three principles. (i) Robust descent: numerical noise or imperfect line searches should not flip the search direction. (ii) Preservation of useful curvature: when the problem is locally well approximated by a quadratic, the method should recover fast progress. (iii) Standard analyzability: the rule fits within the usual CG framework (Zoutendijk condition and Lipschitz continuity of ∇f), enabling a clean global-convergence argument developed in the next section.

As a result, the proposed update can be read as a conservative, practice-oriented refinement of PRP—small in code, but effective in avoiding the rare, costly stalls that inflate iteration counts. task.

$$d_k^{\mathrm{AHPRP}} = \begin{cases} -g_k, & k = 1, \\ -g_k + \left(\frac{\|g_k\|^2 - t_k\, g_k^\top g_{k-1}}{\|g_{k-1}\|^2}\right) d_{k-1} + t_k\, \frac{g_k^\top d_{k-1}}{\|g_{k-1}\|^2}\, g_{k-1}, & \text{if } \|g_k\|^2 > t_k\, g_k^\top g_{k-1},\ k \geq 2, \\ -g_k + t_k\, \frac{g_k^\top s_{k-1}}{\|s_{k-1}\|^2}\, s_{k-1}, & \text{otherwise.} \end{cases} \tag{3}$$

where

$$t_k = \min\left\{1, \frac{\|s_{k-1}\|}{\|y_{k-1}\|}\right\}.$$

2.1. Notation and parameter definitions

For clarity and completeness, we summarize below the main notations and parameters used throughout the paper.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable objective function. At iteration k, we denote by

$$x_k \in \mathbb{R}^n$$

the current iterate and by

$$g_k = \nabla f(x_k)$$

the corresponding gradient. The search direction d_k is generated according to the conjugate-gradient scheme

$$d_k = -g_k + \beta_k d_{k-1}, \quad d_0 = -g_0,$$

and the next iterate is given by

$$x_{k+1} = x_k + \alpha_k d_k,$$

where $\alpha_k > 0$ is a step size obtained via a line-search procedure.

The gradient difference vector is defined as

$$y_{k-1} = g_k - g_{k-1}.$$

In the proposed method, the parameter t_k is a nonnegative scalar used to control the contribution of correction terms in the search direction. It acts as a safeguard to ensure sufficient descent and stability of the method. The specific role of t_k becomes apparent in the construction of the modified search direction and in the convergence analysis.

The vector

$$s_{k-1} = x_k - x_{k-1}$$

denotes the step taken at the previous iteration and is used in the alternative formulation of the search direction when certain curvature conditions are not satisfied.

The line search is assumed to satisfy the strong Wolfe conditions. In particular, $\sigma \in (0, 1)$ denotes the curvature parameter appearing in the sufficient descent condition and related inequalities used in the convergence proofs.

Throughout the analysis, $L > 0$ denotes the Lipschitz constant of the gradient ∇f on a neighborhood of the level set under consideration, as commonly adopted in related convergence analyses for CG-type methods [12, 16–18]. Additional scalars such as μ_k arise as auxiliary quantities introduced during the theoretical analysis and are defined locally where they appear.

3. Global convergence properties

The convergence analysis in this work follows the standard theoretical framework for nonlinear conjugate gradient (CG) methods. In particular, the line-search procedure is assumed to satisfy the strong Wolfe conditions [4, 5], which play a central role in ensuring sufficient descent and controlling the step length in practical implementations.

The analysis also relies on classical convergence results for CG-type methods. The Zoutendijk condition [6] provides a fundamental summability property that is widely used to establish global convergence under inexact line searches. In addition, the framework of Gilbert and Nocedal [7] introduces the well-known Property*, which is essential for controlling the behavior of the CG parameter and ensuring stability of the generated search directions.

Furthermore, sufficient descent properties are closely related to the analysis of Al-Baali [23], which provides key inequalities used in proving that the search directions remain descent directions under appropriate assumptions. These theoretical tools, together with standard boundedness and Lipschitz continuity assumptions, form the basis of the convergence analysis presented in this section.

We now proceed to establish the sufficient descent property and global convergence of the proposed method.

The convergence analysis in this study follows the standard theoretical framework for nonlinear conjugate gradient methods. The Wolfe-type line-search conditions provide the basis for controlling the step size and ensuring suitable descent behavior [4, 5]. The sufficient descent property and global convergence analysis are also closely related to the classical results of Al-Baali, Zoutendijk, and Gilbert and Nocedal, which establish fundamental tools for analysing CG-type methods under boundedness and Lipschitz-continuity assumptions [6, 7, 23]. These results justify the assumptions and analytical structure adopted in proving the descent and convergence properties of the proposed method.

The objective function is considered to be subject to the following presumption.

Assumption 1

I. The level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$ is bounded. This suggests the existence of a nonnegative constant ρ, given that:

$$\|x\| \leq \rho, \quad \forall x \in \Omega.$$

II. In some neighborhood W of Ω, f is a differentiable as well as continuous function. Moreover, its gradient is Lipschitz continuous, implying that, for any $x, y \in W$, there is a constant $L > 0$ provided that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

This assumption indicates that there is a nonnegative constant η provided that

$$\|\nabla f(u)\| \leq \eta, \quad \forall u \in W.$$

the following Lemma, introduced by Zoutendijk [6], is typically utilized.

The sufficient descent and global convergence arguments are also closely related to the classical analyses of Al-Baali [23] and Gilbert and Nocedal [7], which establish key tools for proving convergence of CG-type methods under appropriate boundedness and Lipschitz-continuity assumptions.

To analyze the convergence properties of the CG method under multiple line searches, including SWP including Wolfe-type line searches [4, 5] and WWP, the following lemma, originally proposed by Zoutendijk [6], is commonly employed.

Lemma 3.1

Assume that Assumption 1 holds. We now take into consideration any form of equation (2) and equation (3), in which α_k satisfies the WWP line search with the descent condition. Therefore, the inequality given below holds.

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{4}$$

The following theorem demonstrates that the new formula meets the descent condition.

Theorem 3.1

Let the sequences $\{x_k\}$ as well as $\{d_k\}$ be generated by the methods equation (2) and equation (3), and consider the line search method obtained using equation (3) and equation (4). Therefore, the sufficient descent condition holds.

Proof. Multiplying the search direction in equation (3) by g_k^T gives

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2 - t_k g_k^T g_{k-1}}{\|g_{k-1}\|^2} g_k^T d_{k-1} + t_k \frac{g_k^T d_{k-1}}{\|g_k\|^2} g_k^T g_{k-1}.$$

By using SWP line search we obtain

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \sigma g_k^T d_{k-1}.$$

$$g_k^T d_k \leq -\|g_k\|^2 + \sigma\|g_k\|^2.$$

$$g_k^T d_k \leq (-1+\sigma)\|g_k\|^2.$$

Let

$$(1-\sigma) = c.$$

$$g_k^T d_k \leq -c\|g_k\|^2, \tag{5}$$

where

$$\sigma < 1.$$

This completes the proof. ■

Theorem 3.2

Let the sequences $\{x_k\}$ as well as

$$\{d_k = -g_k + t_k \frac{g_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}\},$$

where the line search method obtained using (3) and (4). Therefore, the sufficient descent condition 5 holds.

Proof.

$$g_k^T d_k = -\|g_k\|^2 - t_k \frac{g_k^T s_{k-1}}{\|s_{k-1}\|^2} g_k^T s_{k-1} = -\|g_k\|^2 - t_k \frac{(g_k^T s_{k-1})^2}{\|s_{k-1}\|^2} < 0.$$

Gilbert and Nocedal [7] as well as Hager and Zhang [10] gave a property called Property* to perform a specialized role in studies on CG formulas related to the PRP method given below.

Property: We take into consideration a method of the form equation (2) and equation (7) and let

$$0 < \gamma \leq \|\mathbf{g}_k\| \leq \bar{\gamma}. \tag{6}$$

The method has Property* provided that there exists a constant $b > 1$ as well as $\lambda > 0$. Note that for all $k \geq 1$, we have $|\beta_k| \leq b$. Moreover, if $\|x_{k+1} - x_k\| \leq \lambda$, then

$$|\beta_k| \leq \frac{1}{2b}.$$

The Lemma given shows that the new β_k inherits Property*. The proof is similar to that presented by Gilbert and Nocedal [7]. ■

Lemma 3.2

Let's assume Assumption 1 is true. We take into consideration any form of equation (2) as well as equation (3). Then, β_k meets Property*.

Proof. Let $b = \frac{2\bar{\gamma}^2}{\gamma^2}$ and $\lambda = \frac{\gamma^2}{2(L\bar{\gamma})b}$. Having that, and using β_k^{HS} and SWP line search, we get:

$$\begin{aligned} |\beta_k| = \left|\frac{g_k^T(g_k - \tau_k g_{k-1})}{\|g_{k-1}\|^2}\right| &\leq \frac{\|g_k\|\left(\|g_k\| + \tau_k\|g_{k-1}\|\right)}{\|g_{k-1}\|^2} \\ &\leq \frac{\|g_k\|\left(\|g_k\| + \|g_{k-1}\|\right)}{\|g_{k-1}\|^2} \leq \frac{2\bar{\gamma}^2}{\gamma^2} = b. \end{aligned}$$

Given that $\|x_{k+1} - x_k\| \leq \lambda$ satisfies Assumption 1, we then have

$$\begin{aligned} |\beta_k| &\leq \left|\frac{g_k^T(g_k - \tau_k g_{k-1})}{\|g_{k-1}\|^2}\right| \\ &\leq \frac{L\bar{\gamma}}{\gamma^2} \leq \frac{1}{2b}. \end{aligned}$$

■

Lemma 3.3

We assume that Assumption 1 is met and the sequences $\{g_k\}$ as well as $\{d_k\}$ are generated by employing Algorithm 1. Here, the step size α_k is generated via the SWP line search provided that the sufficient descent condition 5 is met. If $\beta_k \geq 0$, then there exists a constant $\gamma > 0$ in which $\|g_k\| > \gamma$ for all $k \geq 1$. Therefore, $d_k \neq 0$ and

$$\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 < \infty, \tag{7}$$

in which $u_k = \frac{d_k}{\|d_k\|}$.

Proof. Therefore, we assume that $d_k \neq 0$ with If $d_k = 0$, the sufficient descent condition 5 yields $g_k = 0$. We thus assume $d_k \neq 0$ with:

$$\bar{\gamma} \geq \|g_k\| \geq \gamma > 0, \quad \forall k \geq 1. \tag{8}$$

We now define

$$u_k = w_k + \delta_k u_{k-1},$$

in which

$$w_k = \frac{-g_k + \theta_k}{\|d_k\|}, \quad \theta_k = t_k \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2} g_{k-1}, \quad \delta_k = \frac{\|g_k\|^2 - t_k g_k^T g_{k-1}}{\|g_{k-1}\|^2} \cdot \frac{\|d_{k-1}\|}{\|d_k\|}.$$

Given that u_k is a unit vector, we obtain:

$$\|w_k\| = \|u_k - \delta_k u_{k-1}\| = \|\delta_k u_k - \delta_k u_{k-1}\|.$$

By the triangle inequality and $\delta_k \geq 0$, we have:

$$\begin{aligned} \|u_k - u_{k-1}\| \leq (1+\delta_k)\|u_k - u_{k-1}\| &= \|u_k - \delta_k u_{k-1} - (u_{k-1} - \delta_k u_k)\| \\ &\leq \|u_k - \delta_k u_{k-1}\| + \|\delta_k u_k - \delta_k u_{k-1}\| = 2\|w_k\|. \end{aligned} \tag{9}$$

Next, we express

$$\nu = -g_k + t_k \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2} g_{k-1}.$$

By utilizing the triangular inequality, we gain

$$\|v\| \leq (1 + \sigma c t_k)\bar{\gamma} = T. \tag{10}$$

Then, $\|v\| \leq T$. From equation (10), we have $\|u_k - u_{k-1}\| \leq 2|w_k|$.

By equation (9) and equation (10), we get

$$\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 \leq 4 \sum_{k=0}^{\infty} \|w_k\|^2 \leq 4T^2 \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} < \infty,$$

completing the proof. ■

By Lemmas 4.1 and 4.2 in Ref. [7], we obtained the results given below.

Theorem 3.3

Suppose that the sequences $\{x_k\}$ and $\{d_k\}$ are generated by equation (2), and that the step size α_k satisfies the strong Wolfe conditions. By using Lemmas 3.2, 3.3, and 4.1 and 4.2 in Ref. [7], we gain the findings such that

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

Theorem 3.4

Suppose sequences $\{x_k\}$ as well as $\{d_k\}$ be generated by equation (2) as well as equation (3) and

$$d_k = -g_k + t_k \frac{g_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1}.$$

Moreover, suppose the step size satisfiesequation (3) as well as equation (4). We gain the findings such that

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

Proof:

Let

$$\|\nabla f_k\| \geq \epsilon_1 \quad \text{for all } k \geq 0.$$

From Eq. (4), we have

$$\|d_k\| = \left\| -\nabla f_k - \delta \frac{\nabla f_k^T s_{k-1}}{\|s_{k-1}\|^2} s_{k-1} \right\| \leq (1 + \delta)\|\nabla f_k\| \leq 1 + \eta = \epsilon_2,$$

where ϵ_2 is some positive constant.

Thus, we have

$$\sum_{k=1}^{\infty} \frac{\|\nabla f_k\|^4}{\|d_k\|^2} \geq \sum_{k=1}^{\infty} \frac{\epsilon_1^4}{\epsilon_2^2} = \infty,$$

which contradicts equation (4). Thus, we have

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0,$$

which completes the proof. ■

4. Numerical findings and discussion

The numerical experiments are designed to evaluate the performance of the proposed method using established benchmarking practices in nonlinear optimization. The test problems are selected from the CUTE/CUTEst environment [24], which provides a widely accepted and diverse collection of large-scale unconstrained optimization problems. The use of this benchmark ensures that the performance of the proposed method can be assessed in a reproducible and standardized manner.

To compare the efficiency and robustness of the proposed method with existing algorithms, we adopt the performance profile framework of Dolan and Moré [25]. This approach has become a standard tool for comparing optimization solvers across large problem sets, as it provides a comprehensive and statistically meaningful evaluation of algorithmic performance.

The competing methods are selected to represent both classical and modern developments in conjugate gradient methods. In particular, Dai–Liao-type methods [8], Hager–Zhang-type methods [10], and previously proposed modified CG schemes with practical applications [22] are included to provide a balanced comparison. These methods reflect different strategies for improving descent properties, stability, and computational efficiency in large-scale optimization.

The experiments are implemented using publicly available software tools for CG methods, including the CG-Descent implementation provided by Hager and Zhang, ensuring consistency and reproducibility of the results.

The numerical experiments are designed to evaluate the proposed method using established optimization benchmarking practices. The test problems are selected from the CUTE/CUTEst environment, which is widely used for assessing optimization algorithms on constrained and unconstrained test problems [24]. To compare the efficiency and robustness of the proposed method with existing solvers, the performance-profile approach of Dolan and Moré is adopted because it provides a systematic framework for comparing solvers over a large set of problems [25]. The competing methods are selected to represent both classical and recent CG-type strategies, including Dai–Liao-type methods, Hager–Zhang-type CG methods, and previously developed modified CG schemes with practical applications [8, 10, 22]. The numerical experiments were conducted using the software available at https://people.clas.ufl.edu/hager/software/.

Figure 1 presents the flowchart of the proposed AHPRP algorithm, while Table 1 summarizes the detailed numerical results for the CUTEst test problems. Figures 2–5 compare AHPRP with existing CG-type methods in terms of iterations, function evaluations, CPU time, and gradient evaluations. Together, these figures and the table support the discussion on the efficiency, robustness, and practical performance of the proposed method.

This section reports in detail the performance of the proposed method across standard test problems and an image-restoration task. The evaluation records multiple criteria: the number of iterations, the total function and gradient evaluations, the gradient norm at termination, and the overall CPU time. Such a multifaceted view is necessary, since efficiency in nonlinear optimization is not solely about iteration counts but about the balance between per-step cost, robustness of the line search, and the stability of convergence across diverse landscapes.

To further clarify the implementation and assess the effectiveness of the proposed AHPRP method, we present its algorithmic structure and numerical performance in Figures 1–5 and Table 1. Figure 1 provides a detailed flowchart of the method, explicitly outlining the sequence of operations, including initialization, evaluation of the stopping criterion, computation of the search direction, and the line search procedure satisfying the strong Wolfe conditions. This representation enhances reproducibility and demonstrates that the safeguarding mechanism is seamlessly integrated within the classical conjugate gradient framework without introducing additional algorithmic complexity.

Table 1 reports comprehensive numerical results for a wide range of CUTEst test problems, including the number of iterations, function evaluations, gradient evaluations, final function values, gradient norms, and CPU time. These results confirm that the proposed method consistently achieves convergence with high accuracy, as indicated by the small gradient norms, while maintaining competitive computational cost.

Figures 2–5 provide a comparative performance analysis against established conjugate gradient methods. Specifically, Figure 2 illustrates the number of iterations required to reach convergence, showing that the proposed method generally achieves faster convergence. Figure 3 presents the number of function evaluations, indicating a reduction in computational effort associated with costly objective function calls. Figure 4 compares CPU times, highlighting the overall computational efficiency of the method, while Figure 5 reports the number of gradient evaluations, further confirming the robustness and scalability of the proposed approach in large-scale settings. The consistent performance improvements observed across these metrics demonstrate the stability and practical effectiveness of the AHPRP method over a diverse set of benchmark problems.

On the benchmark problems, the results indicate a clear trend. The proposed method consistently reduces the number of iterations compared to PRP/HS-type schemes, with improvements that become more pronounced as problem conditioning worsens. This is particularly visible in narrow curved valleys, where traditional updates may produce unstable or uphill directions. The safeguard mechanism prevents such incidents, leading to shorter and steadier progress. Function and gradient evaluations follow the same pattern, confirming that the improvement is not cosmetic but grounded in more reliable search directions.

In terms of CPU time, the advantages are moderate but robust. Because the per–iteration arithmetic is essentially identical to classical CG, the measured savings come from avoiding wasted line searches, failed steps, or unnecessary restarts. Thus, even when iteration counts are only modestly better, the wall-clock time reflects a tangible improvement. These savings are especially relevant in large–scale problems where line searches dominate the runtime. Another observation is that the variance of run times across different problems is reduced, pointing to improved predictability in practice.

Stopping criteria and algorithmic parameters

All algorithms considered in the numerical experiments were terminated using the same stopping criteria in order to ensure a fair comparison. Specifically, the iterations were stopped when the Euclidean norm of the gradient satisfied

$$\|\nabla f(x_k)\| \leq 10^{-6},$$

or when the maximum number of iterations was reached. The latter was set sufficiently large so that it was never active for the reported test problems unless otherwise stated.

For all methods, the step size α_k was computed using a line-search procedure satisfying the strong Wolfe conditions. The standard parameters were used throughout the experiments, with the sufficient decrease parameter and curvature parameter chosen as

$$c_1 = 10^{-4}, \qquad c_2 = 0.9.$$

All numerical simulations were performed using identical tolerance values and line-search settings for every algorithm, so that differences in performance can be attributed solely to the search direction strategies rather than to implementation details.

5. Conclusion

This study has introduced a modified conjugate-gradient method within the PRP family, designed to maintain descent directions under strong Wolfe line-search conditions while retaining the algorithm's inherent simplicity. he key idea - safeguarding the curvature term-emerged from a close examination of how adverse curvature can mislead the search direction. By attenuating these harmful effects, the proposed method delivers a more robust performance profile across a wide range of problems.

Our numerical experiments have shown that this approach not only achieves convergence with fewer iterations on average but also maintains competitive or lower computational cost compared to leading CG variants such as HS+TA, DL+, and CG-Descent 6.8. Importantly, the gradient norms at termination points confirm that the solutions are of high quality.

While the method was evaluated on both synthetic benchmarks and a practical image-restoration case, its design principles are broadly applicable. In particular, large-scale optimization problems in machine learning, signal processing, and inverse problems could benefit from the stability and low overhead of the proposed approach.

Overall, the literature has been integrated according to its specific function in the study. Classical CG references support the mathematical foundation of the proposed direction, Wolfe-type line-search references justify the step-size strategy, convergence-related studies support the theoretical analysis, and benchmarking references justify the numerical evaluation procedure. This functional integration strengthens the scholarly coherence of the manuscript and clarifies the relevance and necessity of the cited works.

Future work will focus on two directions: (i) exploring adaptive strategies that adjust the safeguard dynamically based on curvature trends, and (ii) extending the method to constrained or nonsmooth settings, where maintaining stability without second–order information is especially challenging.

Acknowledgment

The corresponding author expresses sincere gratitude to Universiti Sains Islam Malaysia (USIM) for financial support of this research. The authors also gratefully acknowledge the editors and reviewers for their valuable comments and recommendations, which helped improve this work. We also extend our appreciation to Dr. William Hager for his contribution to the CG method's source code.

Data availability

The benchmark test problems used in this study were obtained from the CUTE/CUTEst optimization test environment, which is publicly available at https://github.com/ralna/CUTEst. The numerical results generated in this work are reported in Table 1 and Figures 2–5 of the manuscript. No additional experimental dataset was used in this study.

Funding

The corresponding author reports financial support from Universiti Sains Islam Malaysia (USIM) for this research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript.

References

[1] M. R. Hestenes & E. Stiefel, "Methods of conjugate gradients for solving linear systems", Journal of Research of the National Bureau of Standards **49** (1952) 409. https://doi.org/10.6028/jres.049.044.

[2] E. Polak & G. Ribière, "Note sur la convergence de méthodes de directions conjuguées", Revue Française d'Informatique et de Recherche Opérationnelle. Série Rouge **3** (1969) 35. https://doi.org/10.1051/m2an/196903r100351.

[3] R. Fletcher & C. M. Reeves, "Function minimization by conjugate gradients", The Computer Journal **7** (1964) 149. https://doi.org/10.1093/comjnl/7.2.149.

[4] P. Wolfe, "Convergence conditions for ascent methods", SIAM Review **11** (1969) 226. https://doi.org/10.1137/1011036.

[5] P. Wolfe, "Convergence conditions for ascent methods. II: Some corrections", SIAM Review **13** (1971) 185. https://doi.org/10.1137/1013035.

[6] G. Zoutendijk, "Nonlinear programming, computational methods", in *Integer and Nonlinear Programming*, J. Abadie (Ed.), North-Holland, Amsterdam, Netherlands, 1970, p. 37. Available online: https://cir.nii.ac.jp/crid/1571980075701600256.

[7] J. C. Gilbert & J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization", SIAM Journal on Optimization **2** (1992) 21. https://doi.org/10.1137/0802003.

[8] Y.-H. Dai, "New conjugacy conditions and related nonlinear conjugate gradient methods", Applied Mathematics and Optimization **43** (2001) 87. https://doi.org/10.1007/s002450010019.

[9] G. Li, C. Tang & Z. Wei, "New conjugacy condition and related new conjugate gradient methods for unconstrained optimization", Journal of Computational and Applied Mathematics **202** (2007) 523. https://doi.org/10.1016/j.cam.2006.03.005.

[10] W. W. Hager & H. Zhang, "The limited memory conjugate gradient method", SIAM Journal on Optimization **23** (2013) 2150. https://doi.org/10.1137/120898097.

[11] L. Zhang, "An improved Wei–Yao–Liu nonlinear conjugate gradient method for optimization computation", Applied Mathematics and Computation **215** (2009) 2269. https://doi.org/10.1016/j.amc.2009.08.016.

[12] A. Alhawarat, M. Mamat, M. Rivaie & Z. Salleh, "An efficient hybrid conjugate gradient method with the strong Wolfe–Powell line search", Mathematical Problems in Engineering **2015** (2015) 103517. https://doi.org/10.1155/2015/103517.

[13] A. Alhawarat, Z. Salleh, M. Mamat & M. Rivaie, "An efficient modified Polak–Ribière–Polyak conjugate gradient method with global convergence properties", Optimization Methods and Software **32** (2017) 1299. https://doi.org/10.1080/10556788.2016.1266354.

[14] A. Alhawarat, "Modified parameter of Dai–Liao conjugacy condition of the conjugate gradient method", arXiv (2023). https://doi.org/10.48550/arxiv.2304.06694.

[15] Z. Salleh, A. Almarashi & A. Alhawarat, "Two efficient modifications of AZPRP conjugate gradient method with sufficient descent property", Journal of Inequalities and Applications **2022** (2022) 1. https://doi.org/10.1186/s13660-021-02746-0.

[16] A. Alhawarat, H. Alolaiyan, I. A. Masmali, Z. Salleh & S. Ismail, "A descent four-term of Liu and Storey conjugate gradient method for large scale unconstrained optimization problems", European Journal of Pure and Applied Mathematics **14** (2021) 1429. https://doi.org/10.29020/nybg.ejpam.v14i4.4128.

[17] A. Alhawarat, G. Alhamzi, I. Masmali & Z. Salleh, "A descent four-term conjugate gradient method with global convergence properties for large-scale unconstrained optimisation problems", Mathematical Problems in Engineering **2021** (2021) 6219062. https://doi.org/10.1155/2021/6219062.

[18] A. Alhawarat, Z. Salleh, H. Alolaiyan, H. El Hor & S. Ismail, "A three-term conjugate gradient descent method with some applications", Journal of Inequalities and Applications **2024** (2024) 73. https://doi.org/10.1186/s13660-024-03142-0.

[19] I. A. Masmali, Z. Salleh & A. Alhawarat, "A descent three-term conjugate gradient method with global convergence properties for large-scale unconstrained optimization problems", AIMS Mathematics **6** (2021) 10742. https://doi.org/10.3934/math.2021624.

[20] P. S. Stanimirović, B. D. Ivanov, D. Stanujkić, L. A. Kazakovtsev, V. N. Krutikov & D. Karabašević, "Fuzzy adaptive parameter in the Dai–Liao optimization method based on neutrosophy", Symmetry **15** (2023) 1217. https://doi.org/10.3390/sym15061217.

[21] N. Salihu, P. Kumam, A. M. Awwal, M. R. Odekunle & T. Seangwattana, "An efficient matrix-free optimization algorithm combining a revised PRP and FR–CG type method with application to robotics", Journal of Computational and Applied Mathematics **483** (2026) 117378. https://doi.org/10.1016/j.cam.2026.117378.

[22] A. Alhawarat, S. Masmali, I. Masmali, M. Al-Baali & S. Ismail, "A modified conjugate gradient method with Taylor approximation: Applications in electric circuits and image restoration", European Journal of Pure and Applied Mathematics **18** (2025) 5639. https://doi.org/10.29020/nybg.ejpam.v18i1.5639.

[23] M. Al-Baali, "Descent property and global convergence of the Fletcher–Reeves method with inexact line search", IMA Journal of Numerical Analysis **5** (1985) 121. https://doi.org/10.1093/imanum/5.1.121.

[24] I. Bongartz, A. R. Conn, N. Gould & Ph. L. Toint, "CUTE: Constrained and unconstrained testing environment", ACM Transactions on Mathematical Software **21** (1995) 123. https://doi.org/10.1145/200979.201043.

[25] E. D. Dolan & J. J. Moré, "Benchmarking optimization software with performance profiles", Mathematical Programming **91** (2002) 201. https:

//doi.org/10.1007/s101070100263.