Menemui Matematik
(Discovering Mathematics)

journal homepage: https://persama.org.my/dismath/home

An Enhanced Spectral Gradient Method for Large-Scale Unconstrained Optimisation with Efficient Inverse Hessian Approximation

Dick Mun Chang[1], Hong Seng Sim[1*], Yong Kheng Goh[1], Sing Yee Chua[2] and Wah June Leong[3]

[1]*Centre for Mathematical Sciences, Universiti Tunku Abdul Rahman, 43000 Kajang, Selangor*
[2]*Centre for Advanced Sustainable Materials Research, Universiti Tunku Abdul Rahman, 43000 Kajang, Selangor*
[3]*Institute of Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor*

msimhs@utar.edu.my
*Corresponding author

Received:2 Jun 2025
Accepted: 12 August 2025

ABSTRACT

This study introduces the Enhanced Spectral Gradient (ESG) method, a modified optimisation technique designed for large-scale unconstrained problems. The ESG method employs a direct diagonal approximation of the inverse Hessian matrix, derived by minimising the log-determinant norm of the Hessian, thereby avoiding costly matrix inversions while maintaining numerical stability. By incorporating a weak secant condition and a computationally efficient update rule, ESG achieves superior convergence properties compared to existing methods. To validate its performance, we conducted extensive numerical experiments on 96 test problems from the CUTE and Andrei collections, spanning dimensions from 10 to 10,000. Benchmarking against seven state-of-the-art gradient-based methods (including MDG, CG variants, and Steepest Descent) revealed that ESG consistently outperforms competitors for large performance ratios ($\tau > 5$), as measured by iterations, function evaluations, and CPU time. Profiling graphs demonstrated ESG's robustness, particularly in high-dimensional settings, where its low-cost backtracking line search and minimal implementation overhead offer practical advantages. The ESG method's drop-in compatibility with existing optimisation frameworks, makes it an attractive candidate for integration into scientific computing libraries. This work bridges theoretical innovation with practical utility, providing a scalable and efficient solution for large-scale optimisation challenges.

Keywords: Spectral Gradient, unconstrained optimisation, inverse Hessian approximation, large-scale problems, performance profiling.

INTRODUCTION

Optimisation is a fundamental research problem across various disciplines, including mathematics, engineering, computer science, and economics. At its core, optimisation involves identifying the best solution from a set of feasible solutions to the optimal value of a function while adhering to predefined constraints. A critical step in this process is formulating an objective function (or cost function), which quantifies the quantity to be minimised (e.g., error, cost) or maximised. The goal is to determine the input parameters that yield the optimal value of this function, often under constraints that can be expressed as either equality or inequality conditions.

Optimisation techniques have been widely applied in numerous fields, such as neural networks, computer vision, and image processing. Various optimisation algorithms have been developed to tackle such problems, including gradient descent, Newton's method, and genetic algorithms. These methods are essentially iterative in nature. The methods refine the solution iteratively until convergence to an optimal (or near-optimal) point is achieved. The choice of algorithm depends on factors such as the problem's convexity, the differentiability of the objective function, and whether the goal is to find global or local optima (Nocedal and Wright, 2006).

Among these methods, the Steepest Descent (SD) method is one of the oldest and simplest iterative optimisation techniques; it updates solutions in the direction of the negative gradient. While computationally efficient, SD suffers from slow convergence near minima due to its zig-zagging behaviour (Cauchy, 1847). Subsequently, more advanced methods, such as the Conjugate Gradient (CG) and Quasi-Newton approaches, address these limitations but introduce new challenges. For instance, the quasi-Newton method approximates the Hessian matrix to accelerate convergence but requires costly matrix inversions, making it computationally expensive for high-dimensional problems (Nocedal and Wright, 2006).

To mitigate these computational burdens, Barzilai and Borwein (1988) proposed a family of spectral gradient methods. Unlike traditional quasi-Newton methods that compute full-rank Hessian matrices, Barzilai and Borwein approximate the Hessian matrix with a carefully picked scalar. Effectively, the Hessian matrix is replaced with a constant diagonal matrix. Sim et al. (2019) proposed variable diagonal damping factors in the approximated matrix. The variable diagonal elements allow decreasing the value of the objective function by all components simultaneously.

This paper presents an improved spectral gradient method that employs a spectral parameter to approximate the inverse Hessian matrix. Building upon the variable diagonal spectral method proposed by Sim et al., we develop an enhanced formulation featuring a variable spectral approximation of the inverse Hessian that can be directly incorporated into unconstrained optimisation update rules. The proposed method's efficacy is demonstrated through comprehensive numerical experiments, which compare its performance against established gradient-based optimisation techniques. These comparisons are conducted using standardized test problems from the CUTE collection (Bongartz et al., 1995) and additional benchmark problems (Andrei, 2008)

APPROXIMATION OF THE INVERSE HESSIAN MATRIX USING A DIAGONAL MATRIX

Consider a typical unconstrained optimisation problem:

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

In this context, $f(x)$ represents the objective function, while x denotes an $n \times 1$ decision vector. A typical iterative formula for updating optimisations with the spectral gradient method is:

$$x_{k+1} = x_k - {B_k}^{-1} g_k,$$

where x_k is the decision vector for the optimisation problem at step k, while g_k denotes the gradient vector. The matrix B_k in the standard spectral gradient methods is a constant diagonal matrix that aims to mimic the effects of the Hessian matrix.

Sim et al. (Sim et al., 2019) introduced a diagonal matrix approximation for the Hessian matrix, with the diagonal components that are following an updating formula given by:

$$B_{k+1}{}^{(i)} = \frac{1}{1+\omega\left(s_k{}^{(i)}\right)^2},$$

where *i* denotes the diagonal component of matrix *B*, and *ω* is approximated as:

$$\omega \approx \frac{s_k{}^T s_k - s_k{}^T y_k}{\sum_{i=0}^{n}\left(s_k{}^{(i)}\right)^4}.$$

Here, $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

However, this formulation could be further simplified by explicitly approximating the inverse Hessian, *H_k*, to fully eliminate the need for matrix inversion. We want H_k to be a diagonal and positive-definite matrix that serves as an approximation of $B_k{}^{-1}$. We start with the derivation of the updating formula for H_k by considering the log-determinant norm as the objective function:

$$\Psi(H_k) = \operatorname{tr}(H_k) - \ln\left(\det\left(H_k\right)\right),$$

where tr(H_k) and det(H_k) are the trace and determinant of H_k.

We formulate an optimisation problem using the log-determinant norm, constrained by a *weak secant condition*.

$$\min \Psi(H_{k+1}),$$

$$\text{s.t. } y_k{}^T H_{k+1} y_k = y_k{}^T s_k.$$

This choice of norm promotes an ideal inverse Hessian with *balanced eigenvalues*. The *trace term* penalises excessively large eigenvalues, preventing aggressive step sizes in optimisation. Meanwhile, the *log-determinant term* discourages excessively small eigenvalues, thereby avoiding ill-conditioning and ensuring numerical stability.

Let $H_{k+1} = diag(H_{k+1}{}^{(1)}, \ldots, H_{k+1}{}^{(n)})$ and $y_k = (y_k{}^{(1)}, \ldots, y_k{}^{(n)})$, the objective function and the constraint can be written as:

$$\min \left(\sum_{i=1}^{n} H_{k+1}{}^{(i)}\right) - \ln\left(\prod_{i=1}^{n} H_{k+1}{}^{(i)}\right),$$

$$\text{s.t. } \left(\sum_{i=1}^{n} (y_k{}^{(i)})^2 H_{k+1}{}^{(i)}\right) - y_k{}^T s_k = 0.$$

The Lagrangian is defined as:

$$L(\lambda, \omega) = \omega\left[\left(\sum_{i=1}^{n} (y_k{}^{(i)})^2 H_{k+1}{}^{(i)}\right) - y_k{}^T s_k\right] + \left(\sum_{i=1}^{n} H_{k+1}{}^{(i)}\right) - \ln\left(\prod_{i=1}^{n} H_{k+1}{}^{(i)}\right),$$

where ω is the Lagrange multiplier. Taking the derivative and setting it to zero yields:

$$\frac{\partial L}{\partial H_{k+1}{}^{(i)}} = 1 - \frac{1}{H_{k+1}{}^{(i)}} + \omega\left(y_k{}^{(i)}\right)^2 = 0, i = 0,1,2,\ldots,n,$$

that gives:

$$H_{k+1}{}^{(i)} = \frac{1}{1+\omega\left(y_k{}^{(i)}\right)^2}, i = 1,2,\ldots,n.$$

Substituting this into the weak secant condition, we get an expression:

$$A(\omega) = \sum_{i=1}^{n}\left(\frac{1}{1+\omega\left(y_k{}^{(i)}\right)^2}\right) - y_k{}^T s_k.$$

Note that $A'(\omega) < 0$, $A(\omega)$ is monotonically decreasing for $\omega \in [0,\infty)$. Thus, when $y_k{}^T y_k > y_k{}^T s_k$, $A(\omega) = 0$ has a unique positive solution. In order to speed up the computation of Hk, we avoid solving $A(\omega) = 0$ directly, but approximate the value of ω by applying one step of the Newton-Raphson method with the initial value $\omega_0 = 0$. Consequently, the resulting Lagrange multiplier for step k is approximately

$$\omega_k \approx \omega_0 - \frac{A(\omega)}{A'(\omega)} = \frac{y_k{}^T y_k - y_k{}^T s_k}{\sum_{i=1}^{n}\left(y_k{}^{(i)}\right)^4}.$$

For cases where $y_k{}^T y_k < y_k{}^T s_k$, we will adopt the standard practice for most quasi-Newton methods of applying the Oren-Luenberger scaling (Luenberger, D.G., 1984). Combining both scenarios, we have the new updating formula for H_{k+1}:

$$H_{k+1} = \begin{cases} \operatorname{diag}\left(H_{k+1}^{(1)},\cdots,H_{k+1}^{(n)}\right), & \text{if } y_k^T y_k > y_k^T s_k, \\ \dfrac{y_k^T s_k}{y_k^T y_k} I, & \text{otherwise.} \end{cases}$$

ALGORITHM IMPLEMENTATION

To ensure an optimal step size and stable iterations, we employ a standard iterative method with backtracking line search in our general unconstrained optimisation $\min_{x \in R^n} f(x)$ problem. Let the step length be α_k and α_k satisfies the Armijo condition (Armijo, 1966):

$$f(x_k + \alpha d_k) \le f(x_k) + c\alpha g_k^T d_k,$$

where $0 < c < 1$.

Algorithm 1: Backtracking Armijo Line Search

```
1. Initialize $c \in (0,1), \tau \in (0,1)$ and initial step length, $\alpha$.
2. Check if $f(x_k + \alpha d_k) \le f(x_k) + c\alpha g_k^T d_k$,
3. If satisfied, set $\alpha_k = \alpha$ and update $x_{k+1} = x_k + \alpha d_k$ .
4. Otherwise, reduce $\alpha$ via $\alpha_{k+1} = \tau\alpha_k$ and go to Step 2.
```

Combining the proposed updating rule of H_{k+1} and the Armijo line search, we have the novel enhanced spectral gradient method:

Algorithm 2: Enhanced Spectral Gradient Method

```
1. Initialize $k = 0, x_0 \in \Re, H_0 = I,$ and tolerance $\epsilon$.
2. Compute $g_0 = \nabla f(x_0)$ and $d_0 = -g_0$.
```

```
3. If $||g_k|| \leq \epsilon$, terminate; else compute $\alpha_k$ via backtracking.
4. Update $x_{k+1} = x_k + \alpha_k d_k$.
5. Compute $g_{k+1} = \nabla f(x_{k+1})$.
6. Update $H_{k+1}$ using Eq. (18).
7. Compute $d_{k+1} = -H_{k+1} g_{k+1}$.
8. Increment $k$ and repeat from Step 3.
```

NUMERICAL EXPERIMENTS AND COMPARISONS

Here we report numerical experiments on an extensive 96 tester problems from the Constrained and Unconstrained Testing Environment (CUTE) dataset (Bongartz et al., 1995), as well as the Unconstrained Optimization Test Functions Collection by Andrei (Andrei, 2008). Both collections of test functions are non-linear, user customisable variable sizes, from both synthetic and real-world problems.

The benchmarking with these test functions is carried out via the evaluation of the profiling graphs in terms of: (1) computational time, (2) number of iterations, and (3) number of function evaluations. All experiments were conducted on a 10th Gen Intel Core i5 processor.

Our benchmarking process is to compare our proposed Enhanced Spectral Gradient (ESG) method against seven state-of-the-art gradient-based optimisation technique that commonly implemented in scientific packages. The acronyms of the algorithms used are listed below:

1.	**ESG**	– Enhanced Spectral Gradient (proposed method)
2.	**MDG**	– Multiple Damping Gradient (Sim et al., 2019)
3.	**CG-FR**	– Conjugate Gradient (Fletcher-Reeves, 1960)
4.	**CG-PRP**	– Conjugate Gradient (Polak-Ribière-Polyak, 1969)
5.	**CG-LS**	– Conjugate Gradient (Liu-Storey, 1991)
6.	**CG-DY**	– Conjugate Gradient (Dai-Yuan, 1999)
7.	**CG-HZ**	– Conjugate Gradient (Hager-Zhang, 2006)
8.	**SD**	– Steepest Descent (Cauchy, 1847)

The above algorithms are implemented in Python programming language and tested on a total of 7,680 test problems of various sizes. The specifications of the implementation details are outlined as follows:

- Total test functions: 7,680.
- Dimension of x: Evaluated across a range from 10 to 10,000 dimensions.
- Stopping criteria:
    - Solution is considered as converged when $||g_k|| < 10^{-4}$.
    - Algorithm would stop when reaching a maximum iteration of 10,000.
- Parameters for backtracking Armijo (BTA) condition: $c = 0.1$, τ=0.5, initial step length α=1, and a maximum of 15 backtracking iterations per line search.

PERFORMANCE EVALUATION RESULTS

The benchmark of ESG with the other optimisation algorithms are assessed using the standard performance profiling graphs popularised by Dolan and Moré (2002). The following show the profiling graphs on different performance metrics (iterations, function calls, and CPU time) across all methods.

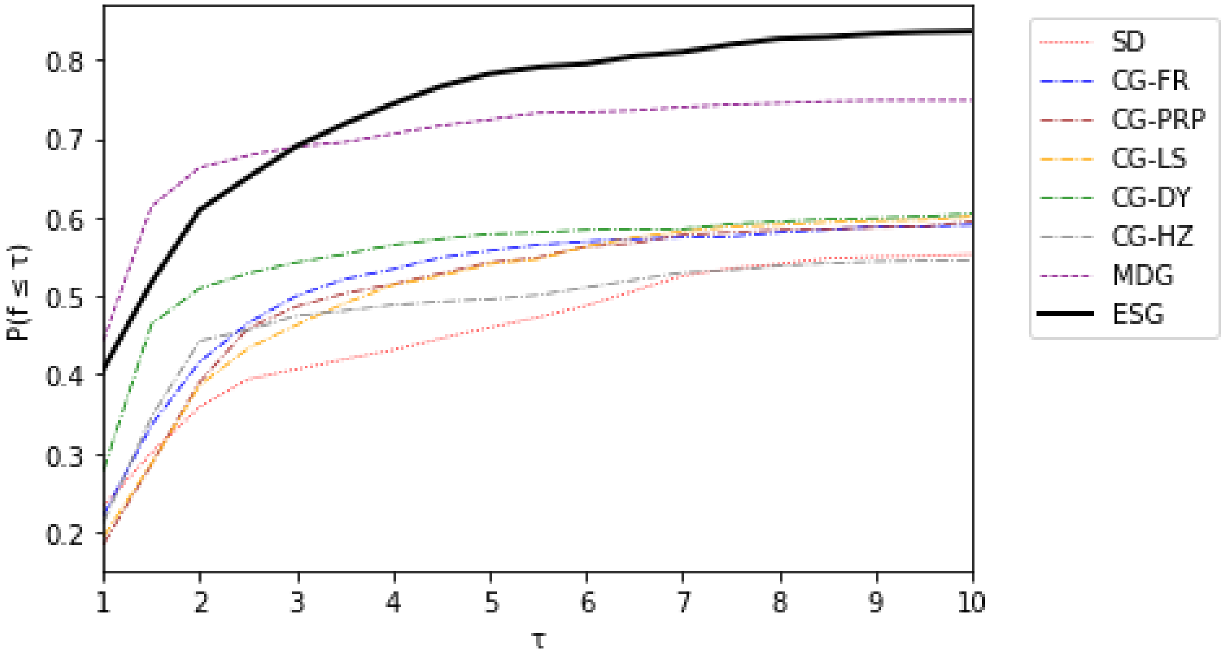

Figure 1: Profiling graph by number of iterations

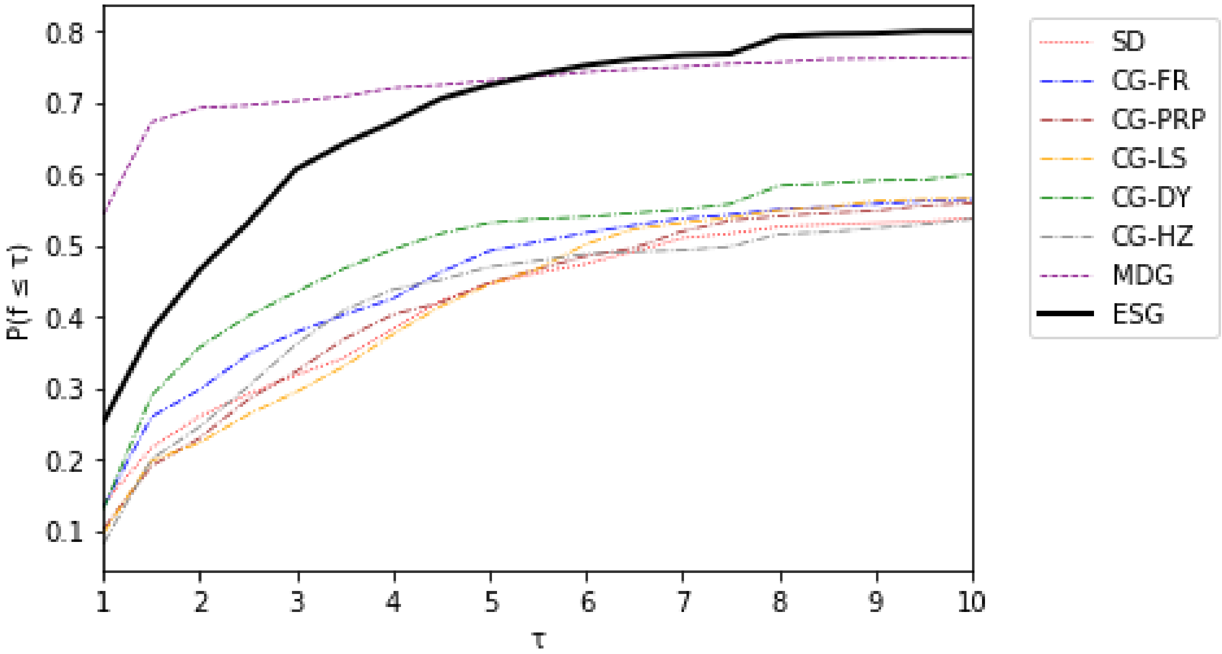

Figure 2: Profiling graph by number of function calls

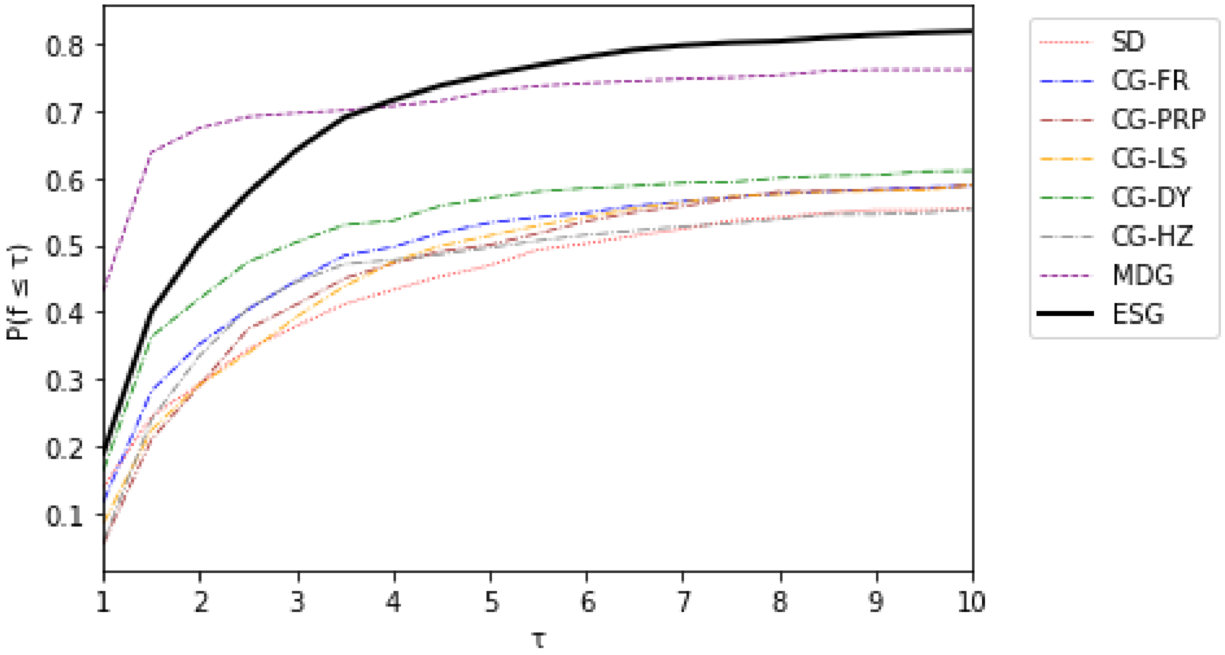

Figure 3: Profiling graph by computation time.

In Figure 1 to Figure 3, the horizontal axis is the performance ratio, τ, in log-scale. The performance ratio compares an algorithm's performance on a specific problem to that of the best-performing algorithm. The vertical axis shows the cumulative probability of the performance ratio, indicating the overall performance ranking of each individual algorithm. A line with higher cumulative probability for large value of performance ratio indicates superior performance. The dark solid line in the figures is the Enhanced Spectral Gradient method, which clearly out-performs other algorithms for large performance ratio.

It is worth noting, for small values of τ, that CG methods often exhibit improved convergence in specific test problems due to incorporation of the stringent line search strategy in these CG methods to achieve optimal convergence. This leads to the comparable performance of ESG with these CG methods for τ < 2.

In comparison, the MDG method outperforms all CG methods in efficiency, as it integrates damping with the line search strategy (Sim et al., 2019). However, due to its use of an approximate Hessian inverse, the MDG method may demand more computational time than the ESG method when solving high-dimensional optimization problems.

Unsurprisingly, for larger values of τ, τ > 5, the ESG method outperforms both the MDG method and other CG methods. This advantage stems from ESG's more direct inverse Hessian approximation, whereas MDG might incurs higher computational costs due to matrix inversion in its update formula. Consequently, ESG could be a replacement candidate algorithm for an efficient and practical alternative for large-scale optimization, especially when a low-cost backtracking line search is the preferred approach.

CONCLUSION

This study presents the Enhanced Spectral Gradient (ESG) method, an improved diagonal spectral gradient approach designed for large-scale unconstrained optimization. The ESG method employs a direct diagonal approximation of the inverse Hessian matrix, derived by minimising the log-determinant norm of the Hessian matrix. To assess its effectiveness, we conducted numerical experiments comparing ESG with the MDG method and various conjugate gradient (CG) methods. The profiling graph results demonstrate that ESG consistently outperforms other gradient-based methods in overall efficiency.

The ESG method offers particular practical advantages as it maintains compatibility with standard backtracking line search techniques. Its straightforward implementation makes it an attractive option for integration into existing unconstrained optimisation libraries. The transition to ESG requires minimal modifications, simply replacing the gradient update formula while preserving current stopping criteria and line search procedures. This minor adjustment can potentially yield significant performance improvements, making ESG an efficient upgrade for current optimisation frameworks.

ACKNOWLEDGEMENT

This research was supported by the Universiti Tunku Abdul Rahman (UTAR) through UTAR Research Fund (IPSR/RMC/UTARRF/2025-C1/S02).

REFERENCES

Andrei, N. (2008), An unconstrained optimisation test functions collection. *Advanced Modelling and Optimisation,* **10(1):** 147-161.

Armijo, L. (1966), Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific J. Math*. **16(1)**: 1-3.

Barzilai, J. and Borwein, J. (1988), Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, **8:** 141-148.

Bongartz, I., Conn., A. R., Gould., N. I. M., Toint., Ph. L. (1995), CUTE: constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, **21:** 123-160.

Cauchy, A. (1847), Méthode générale pour la résolution des systemes d'équations simultanées [General method for solving systems of simultaneous equations]. *Comp. Rend. Sci. Paris*, **25:** 536-538.

Dai, Y. H., Yuan, Y. (1999), A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on Optimisation,* **10(1):** 177-182.

Dolan, E., Moré, J. (2002), Benchmarking optimisation software with performance profiles. *Math Program,* **91**: 201-213.

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

Hager, W. W., Zhang, H. (2005), A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimisation,* **2(1):** 35-58.

Liu, Y., and Storey, C. (1991), Efficient Generalized Conjugate Gradient Algorithms, Part 1. *Theory. J. Optim. Theory Appl.,* **69:** 129-137.

Luenberger, D.G. (1984), *Linear and Nonlinear Programming, 2nd ed.* Reading: Addison-Wesley.

Nocedal, J. and Wright, S. J. (2006), *Numerical Optimisation, 2nd ed*. New York: Springer.

Polak, E. and Ribiere, G. (1969), Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique,* **3:** 35-43.

Polyak, B. T. (1969), The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics* **9(4):** 94–112.

Sim, H.S., Leong, W.J., and Chen, C.Y. (2019), Gradient Method with Multiple Damping for Large Scale Unconstrained Optimization. *Optimization Letters,* **13**: 617-632.