Preconditioning Nonlinear Conjugate Gradient with Diagonalized Quasi-Newton

Alp Dener
adener@anl.gov
Argonne National Laboratory
Lemont, Illinois

Adam Denchfield
adench2@uic.edu
University of Illinois at Chicago
Chicago, Illinois

Todd Munson
tmunson@mcs.anl.gov
Argonne National Laboratory
Lemont, Illinois

ABSTRACT

Nonlinear conjugate gradient (NCG) methods can generate search directions using only first-order information and a few dot products, making them attractive algorithms for solving large-scale optimization problems. However, even the most modern NCG methods can require large numbers of iterations and, therefore, many function evaluations to converge to a solution. This poses a challenge for simulation-constrained problems where the function evaluation entails expensive partial or ordinary differential equation solutions. Preconditioning can accelerate convergence and help compute a solution in fewer function evaluations. However, general-purpose preconditioners for nonlinear problems are challenging to construct. In this paper, we review a selection of classical and modern NCG methods, introduce their preconditioned variants, and propose a preconditioner based on the diagonalization of the BFGS formula. As with the NCG methods, this preconditioner utilizes only first-order information and requires only a small number of dot products. Our numerical experiments using CUTEst problems indicate that the proposed preconditioner successfully reduces the number of function evaluations at negligible additional cost for its update and application.

CCS CONCEPTS

• **Theory of computation → Continuous optimization**; **Quadratic programming**; **Nonconvex optimization**.

KEYWORDS

conjugate gradient, preconditioning, quasi-Newton, optimization

ACM Reference Format:
Alp Dener, Adam Denchfield, and Todd Munson. 2019. Preconditioning Nonlinear Conjugate Gradient with Diagonalized Quasi-Newton. In *Proceedings of the Platform for Advanced Scientific ComputingConference (PASC '19), June 12–14, 2019, Zurich, Switzerland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3324989.3325712

1 INTRODUCTION

Linear conjugate gradient (CG) methods have been widely used to iteratively solve linear systems of the form $Ax = b$ where A is a symmetric, positive definite matrix since their debut in 1952 when Hestenes and Stiefel [25] introduced the first of its kind. The core idea is to minimize the strongly convex quadratic function,

$$\min_x \phi(x) := \frac{1}{2}x^T Ax - b^T x, \tag{1}$$

by taking steps along directions that are conjugate with respect to the A matrix. At each iteration, this approach yields a search direction, d_k, that is a linear combination of the residual, $r_k = Ax_k - b$, and the previous direction, d_{k-1}, such that

$$d_k = -r_k + \beta_k d_{k-1}, \tag{2}$$

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3}$$

where

$$\alpha_k = \frac{r_k^T d_k}{d_k A d_k} \tag{4}$$

is the one-dimensional minimizer of $\phi(\cdot)$ along $x_k + \alpha_k d_k$, and

$$\beta_k = \frac{\|r_k\|^2}{\|r_{k-1}\|^2} \tag{5}$$

where $\|\cdot\|$ is the Euclidean norm throughout this paper.

Fletcher and Reeves extended this method in 1964 to minimize nonlinear functions $f(x) : \mathbb{R}^n \to \mathbb{R}$ by making two changes [19]. First, the residual is replaced with the gradient of the objective function, $g_k = \nabla f(x_k)$, and, second, the step length α_k is computed via a line search that minimizes the nonlinear function along the search direction. An overview of the method is available in Alg. 1. This approach has proven effective, particularly for large-scale optimization because it can generate search directions economically using only first-order derivative information and few dot products.

For strongly convex quadratic functions, and using exact line searches, the Fletcher-Reeves nonlinear conjugate gradient (NCG) method reduces to the Hestenes-Stiefel method and guarantees convergence exactly within n iterations [19]. Converseley, its numerical performance on general nonlinear functions with inexact line searches leaves much to be desired, taking many small steps and exhibiting slow convergence. Considerable work has been done by many researchers since the 1950s to develop NCG variants that offer improved convergence, robustness, and efficiency over the Fletcher-Reeves method on such problems. We review a selection of these methods in Section 2.

In the present work, however, we are interested primarily in preconditioning. NCG methods can generate search directions economically, with a small memory footprint, but can take a large number of nonlinear iterations to solve large-scale problems. This is a well-understood trade-off between convergence rates and storage requirements for first-order gradient-based methods [37]. We

PASC '19, June 12–14, 2019, Zurich, Switzerland

ACM ISBN 978-1-4503-6770-7/19/06…$15.00
https://doi.org/10.1145/3324989.3325712

posit that effective preconditioning of NCG can improve this behavior by accelerating convergence at a negligible storage cost. Our work is motivated by an emerging class of large-scale problems particularly in data assimilation, where the function and gradient evaluations depend on expensive simulations (e.g. solution of partial differential equations). Example applications include optical tomography [1], seismic inversion [17], and weather forecasting [18]. Preconditioned nonlinear CG methods have the potential of making a significant impact in these fields.

For linear problems, preconditioning modifies the system of equations in order to improve the eigenvalue distribution of A. Instead of $Ax = b$, we solve the system $(C^{-T}AC^{-1})\hat{x} = C^{-T}b$, where C is a nonsingular matrix and $\hat{x} = Cx$. This process and choices of preconditioners are well understood for linear problems; however, extending preconditioning to NCG methods remains an open question with little consensus.

We address this need for nonlinear preconditioning by establishing a mathematical connection between NCG and limited-memory quasi-Newton methods and drawing on quasi-Newton Hessian initializations to develop effective preconditioners for NCG methods. We begin by highlighting the selection of NCG methods we use in our numerical experiments, and we then introduce a nonlinear quasi-Newton preconditioner framework applicable to any NCG method. We implement this approach as a bound-constrained NCG method in PETSc/TAO Version 3.10 [5, 14], and evaluate its performance on 119 bound-constrained CUTEst test problems [21].

2 REVIEW OF METHODS

The classical NCG methods developed from the 1950 to the 1990s focused primarily on different ways to compute the β_k parameter. Hager and Zhang [23] present a comprehensive review of these methods and discuss their convergence properties. In the present work, we briefly review the original Fletcher-Reeves [19] NCG method and its most popular improvement, the Polak-Ribière-Polyak method [33, 34]. However, our primary interest lies in the self-scaling memoryless BFGS (SSML-BFGS) method by Perry [32] and Shanno [36], as well as the modern Hager-Zhang [22] method and the Dai-Kou [13] family that relate to it.

Fletcher-Reeves

Fletcher and Reeves [19] extended the Hestenes-Stiefel linear CG method to nonlinear problems with a straightforward replacement of the linear residual with the gradient of the nonlinear function, such that

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}. \tag{6}$$

Although globally convergent, it has quickly been superseded by faster and more robust NCG methods. We have chosen to include Fletcher-Reeves in our numerical studies because of its historical significance. An overview of this algorithm is provided in Alg. 1.

Polak-Ribière-Polyak

The propensity of the Fletcher-Reeves method to get stuck in place while taking many small steps motivated Polak and Ribière [33] and Polyak [34] to modify the numerator such that

Algorithm 1: Fletcher-Reeves nonlinear conjugate gradient method.

```
Data: $x_0$, $\epsilon$, $K$
Result: estimate for the optimal solution $x^*$
Evaluate $g_0 = \nabla f(x_0)$
Initialize $d_0 = -g_0$ and $k = 0$
while $\|g_k\| > \epsilon$ and $k < K$ do
    Find $\alpha_k$ to minimize $f(x_k + \alpha_k d_k)$
    Update $x_{k+1} = x_k + \alpha_k d_k$
    Evaluate $g_{k+1} = \nabla f(x_{k+1})$
    Compute $\beta_{k+1}^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}$
    Set $d_{k+1} = -g_{k+1} + \beta_{k+1}^{FR} d_k$ and $k = k + 1$
end
```

$$\beta_k^{PRP} = \frac{g_k^\top y_k}{\|g_{k-1}\|^2}, \tag{7}$$

where $y_k = g_k - g_{k-1}$. When the subsequent gradients are orthogonal, $g_k^T g_{k-1} = 0$, the Polak-Ribière-Polyak (PRP) method reduces to the Fletcher-Reeves method. However, the inclusion of the previous gradient in this term introduces a built-in “reset” mechanism to the algorithm, where $\beta_k^{PRP} \to 0$ as $\|g_k - g_{k-1}\| \to 0$. In other words, when the steps make little or no improvement in the gradient of the objective function, the step direction automatically reduces to the steepest descent direction.

Researchers have observed, however, that the PRP method may not generate valid descent directions with inexact line searches. To address this issue, Powell suggested a modification to the PRP method in 1984 [35] such that $\beta_k^{PRP+} = \max\{\beta_k^{PRP}, 0\}$. We include both the original and the modified PRP methods in our numerical studies.

Self-Scaling Memoryless BFGS (Perry-Shanno)

The foundation of the SSML-BFGS algorithm was first discovered by Perry *et al.* in 1977 [32] as a means of developing a nonlinear conjugate gradient algorithm with memory (i.e., stored information from past iterates) that could address the challenges in applying full-memory quasi-Newton methods to large-scale problems where it is impractical to store the Hessian matrix. This method was the first effort of its kind, preceding Nocedal’s introduction of the limited-memory two-loop BFGS formula in 1980 [28].

Perry’s algorithm was later reinterpreted as memoryless BFGS by Shanno in 1978 [36] and improved into the final SSML-BFGS method with the introduction of a scaling term, τ_k, such that

$$d_k = -\hat{H}_k g_k, \tag{8}$$

where

$$\hat{H}_k = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) \tau_k I \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k} \tag{9}$$

with $s_k = x_k - x_{k-1}$. The expression in (9) is easily recognizable as a limited-memory BFGS approximation to the inverse Hessian with only one update and with the initial Hessian defined as $\hat{H}_0 = \tau_k I$. The resulting algorithm is also often referred to as the Perry-Shanno scheme.

Leveraging the substitution $s_k = \alpha_{k-1} d_{k-1}$ yields a step direction notation that takes on the modified NCG structure

$$d_k = \tau_k \left(-g_k + \gamma_k^{PS} y_k + \beta_k^{PS} d_{k-1} \right), \tag{10}$$

where

$$\gamma_k^{PS} = \frac{g_k^T d_{k-1}}{y_k^T d_{k-1}}, \tag{11}$$

$$\beta_k^{PS} = \frac{1}{y_k^T d_{k-1}} \left[g_k^T y_k - \left(\frac{y_k^T y_k}{y_k^T d_{k-1}} + \frac{\alpha_{k-1}}{\tau_k} \right) g_k^T d_{k-1} \right]. \tag{12}$$

In our numerical experiments with SSML-BFGS, we adopt the Oren and Spedicato scaling factor [31]

$$\tau_k = \frac{y_k^T s_k}{y_k^T y_k} = \frac{\alpha_{k-1} y_k^T d_{k-1}}{y_k^T y_k}, \tag{13}$$

which reduces (12) to

$$\beta_k^{PS} = \frac{1}{y_k^T d_{k-1}} \left(g_k^T y_k - 2 \frac{y_k^T y_k}{y_k^T d_{k-1}} g_k^T d_{k-1} \right). \tag{14}$$

Hager-Zhang

The SSML-BFGS scheme requires that the Hessian approximation $\hat{H}_k$ be periodically reset to the identity matrix to guarantee global convergence; this process can, however, decrease the rate of convergence. Hager and Zhang [22] sought to improve this behavior with a restriction that deletes the $\gamma_k^{PS} y_k$ term in (10) and adopts the Oren and Spedicato scaling in (13) such that

$$d_k = -g_k + \beta_k^{HZ+} d_{k-1}, \tag{15}$$

where

$$\beta_k^{HZ+} = \max\{\beta_k^{HZ}, \eta_k^{HZ}\}, \tag{16}$$

$$\beta_k^{HZ} = \frac{1}{y_k^T d_{k-1}} \left(g_k^T y_k - 2 \frac{y_k^T y_k}{y_k^T d_{k-1}} g_k^T d_{k-1} \right), \tag{17}$$

and

$$\eta_k^{HZ} = \frac{-1}{\|d_{k-1}\| \min\{0.01, \|g_{k-1}\|\}}. \tag{18}$$

Similar to Powell's modification to the PRP method, the β_k^{HZ+} is restricted with a lower bound to guarantee global convergence. However, Hager and Zhang's truncation dynamically adjusts this lower bound such that $\eta_k^{HZ} \to -\infty$ as $\|g_k\| \to 0$ in order to accelerate convergence.

Hager and Zhang also developed an approximate Wolfe line search tailored to their NCG method. Our implementation does not include this line search and instead relies on a general purpose Morè-Thuente line search [26] that can take step lengths greater than 1 in order to account for potentially poor scaling in the NCG direction.

Dai-Kou

Building on Hager and Zhang's efforts, Dai and Kou [13] explored alternatives to deleting the $\gamma_k^{PS} y_k$ term in SSML-BFGS and developed a family of methods such that

$$d_k = -g_k + \beta_k^{DK+} d_{k-1}, \tag{19}$$

where

$$\beta_k^{DK+} = \max \left\{ \beta_k^{DK}(\tau_k), \eta^{DK} \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} \right\}, \tag{20}$$

$$\beta_k^{DK}(\tau_k) = \frac{g_k^T y_k}{y_k^T d_{k-1}} - \left(\tau_k + \frac{y_k^T y_k}{s_k^T y_k} - \frac{s_k^T y_k}{s_k^T s_k} \right) \frac{g_k^T s_k}{y_k^T d_{k-1}}, \tag{21}$$

and $\eta^{DK} \in [0, 1)$ is a scalar parameter, chosen to be 0.5 by Dai and Kou in their experiments. As before, β_k^{DK+} is truncated to guarantee global convergence, this time with a positive term that produces a downhill direction that Dai and Kou have observed to be a better restart direction than steepest descent in their numerical experiments. The most effective member of this family of methods, and the one we include in our numerical experiments, is defined by the Oren and Luenberger scaling factor [30]

$$\tau_k = \frac{s_k^T y_k}{s_k^T s_k}, \tag{22}$$

which reduces (21) to

$$\beta_k^{DK} = \frac{g_k^T y_k}{y_k^T d_{k-1}} - \frac{y_k^T y_k}{s_k^T y_k} \frac{g_k^T s_k}{y_k^T d_{k-1}}. \tag{23}$$

Similar to Hager and Zhang, Dai and Kou also developed a matching improved Wolfe line search algorithm tailored to their method. As before, we do not implement this line search and instead evaluate the effect of our proposed preconditioning approach using the Morè-Thuente line search for all NCG methods.

3 PRECONDITIONING

The task of preconditioning linear CG methods is a well-understood subject viewed through the lens of reducing the condition number of the constant coefficient matrix. The linear system is altered such that we solve $C^{-T} A C^{-1} \hat{x} = C^{-T} b$ where $\hat{x} = Cx$.

In practice, however, the C matrix is never directly used. Instead, a stationary symmetric positive-definite preconditioner matrix $M = CC^T$ is constructed such that $M^{-1} \approx A^{-1}$ and $M^{-1} A \approx I$. The resulting preconditioner is applied to the residual in the β_k definition, replacing it with $z_k = M^{-1} r_k$ such that the Hestenes-Stiefel method becomes

$$\beta_k = \frac{z_k^T r_k}{z_{k-1}^T r_{k-1}}. \tag{24}$$

It is possible, in theory, to perform the same modification in the nonlinear case. Instead of the original function, $f(x)$, we minimize the "preconditioned" function $\hat{f}(x) = f(\hat{x})$ with respect to the modified variables $\hat{x} = Cx$ where C represents a non-singular linear transformation. Propagating the substitution, we obtain the step direction for the modified function,

$$\hat{d}_k = -\hat{g}_k + \hat{\beta}_k \hat{d}_{k-1}, \tag{25}$$

where $\hat{g}_k = C^T g_k$ and $\hat{d}_k = C^{-1} d_k$. The Fletcher-Reeves method in the original variables then becomes

$$d_k = -M g_k + \hat{\beta}_k d_{k-1}, \tag{26}$$

where

$$\hat{\beta}_k = \frac{g_k^T M g_k}{g_{k-1}^T M g_{k-1}} \tag{27}$$

and $M = C_k C_k^T$. As before, this factorization of M is never used in practice and C is never formed. However, the form of the factorization guarantees that M is always symmetric positive-definite under the assumption that the variable transformation is non-singular. Consequently, in practical applications, we seek symetric positive-definite preconditioners that approximate the inverse of the Hessian at each iteration, such that $M\nabla_x^2 f(x_k) \approx I$. For a closer look at the derivation and effect of this preconditioning, we refer the reader to Hager and Zhang [23].

The above approach can also be considered analogous to nonlinear right-preconditioning for Newton's method, where the preconditioner enters the system on the "right" through a variable transformation. This terminology mimics the linear case, where the preconditioner right-multiplies the coefficient matrix. In contrast, left-preconditioning aims to construct a new function that has the same solution as the original but is easier to solve. Brune *et al.* provide a comprehensive review of these approaches in the scope of solving nonlinear partial differential equations, and re-interpret them as composable nonlinear solvers [7]. They demonstrate that the nonlinear right-preconditioning of Newton-Krylov methods reduces to a multiplicative composition of Newton-Krylov with second solver. Unfortunately, this reduction is not possible for NCG, and the variable transformation must be propagated through the algorithm in order to derive different update formulas for each NCG method. This limitation has led Brune *et al.* to restrict their NCG results to left-preconditioners only. In the present work, we derive the right-preconditioned NCG formulas and investigate their performance on bound-constrained optimization problems.

For an efficient and effective choice of M, we draw inspiration from limited-memory quasi-Newton methods. The Perry-Shanno scheme reviewed in Section 2 utilizes a single-update BFGS approximation to compute the step direction. In this formulation, the scale factor τ_k appears where the initial Hessian would be in a traditional limited-memory BFGS method. In other words, Perry-Shanno utilizes a scalar approximation of $\hat{H}_0 = \tau_k I$ as the initial Hessian on top of which a single BFGS update is applied.

Such scalar Hessian initializations are commonly used in limited-memory quasi-Newton methods. For instance, $\hat{H}_0 = (y_k^T y_k / s_k^T y_k) I$ is a popular choice for the BFGS method, as recommended by Nocedal and Wright [29]. However, more sophisticated Hessian initializations also exist. Gilbert and Lemaréchal [20] proposed a sparse Hessian initialization, $\hat{H}_0 = \rho_k \mathrm{diag}(h_k)$, where

$$h_k = h_{k-1} + \frac{y_k \circ y_k}{y_k^T s_k} - \frac{(h_{k-1} \circ s_k)^2}{s_k^T (h_{k-1} \circ s_k)} \tag{28}$$

and

$$\rho_k = \frac{y_k^T (h_k \circ y_k)}{y_k^T s_k}. \tag{29}$$

This initialization is a diagonalization of the full-memory BFGS update where the matrix-vector products have been replaced by Hadamard products (denoted by ∘) with the h_k vector representing the diagonal entries of the initial Hessian. The rescaling parameter ρ_k was added based on numerical experiments that revealed the BFGS formula's inability to rapidly modify a diagonal matrix.

In the present work, we adapt Gilbert and Lemaréchal's Hessian initialization as a preconditioner in NCG methods. We replace the previously static preconditioner M with the dynamically updated preconditioner $M_k = \rho_k^{-1} \mathrm{diag}(h_k^{-1})$ and modify the NCG formulas as follows:

- **Fletcher-Reeves**
$$\hat{\beta}_k^{FR} = \frac{g_k^T M_k g_k}{g_{k-1}^T M_k g_{k-1}} \tag{30}$$
- **Polak-Ribière-Polyak**
$$\hat{\beta}_k^{PRP} = \frac{g_k^T M_k y_k}{g_{k-1}^T M_k g_{k-1}} \tag{31}$$
- **Hager-Zhang**
$$\hat{\beta}_k^{HZ} = \frac{1}{y_k^T d_{k-1}} \left(g_k^T M_k y_k - 2 \frac{y_k^T M_k y_k}{y_k^T d_{k-1}} g_k^T d_{k-1} \right) \tag{32}$$
- **Dai-Kou**
$$\hat{\beta}_k^{DK} = \frac{g_k^T M_k y_k}{y_k^T d_{k-1}} - \frac{y_k^T M_k y_k}{s_k^T y_k} \frac{g_k^T s_k}{y_k^T d_{k-1}} \tag{33}$$

We conclude the description of the preconditioner with two remarks.

Remark. *Preconditioned formulas for modern CG methods based on the Perry-Shanno scheme (e.g., Hager-Zhang and Dai-Kou) are derived by replacing the initial Hessian in the BFGS formula with the preconditioner matrix and retracing the original derivation steps for the methods that involve modifying and/or dropping the $\gamma_k y_k$ term. This approach is equivalent to performing the $\hat{g}_k = C_k^{-T} g_k$ and $\hat{d}_k = C_k d_k$ substitutions as in the case for classic NCG methods (e.g., Fletcher-Reeves and Polak-Ribière).*

Remark. *The idea of accelerating NCG with quasi-Newton information is not new; the potential was first identified by Buckley [8] and Nazareth [27] in the context of exploring mathematical connections between CG and quasi-Newton methods. Andrei utilized this connection in accelerating NCG methods with a scalar-scaling based on quasi-Newton updates [3, 4]. There has also been significant interest in developing limited-memory quasi-Newton-like conjugate gradient methods that utilize iteration history [9, 24]. Most recently, Caliciotti* et al. *have investigated quasi-Newton updates derived from modified scant equations as preconditioners for NCG methods [2, 12]. While effective, these approaches nonetheless increase the memory footprint of NCG methods, which detracts from NCG's key advantage over quasi-Newton methods for large-scale applications. In contrast, the proposed diagonalized BFGS preconditioner accumulates information from an unlimited number of past iterates without storing any history, and econdes more information than a scalar scaling without increasing storage requirements.*

4 NUMERICAL STUDIES

We now investigate the numerical performance of the selected NCG methods with and without the proposed preconditioning. The methods are implemented in PETSc/TAO as a bound-constrained algorithm utilizing a Moré-Thuente line search [26] and an active-set estimation based on Bertsekas' work [6].

We note that our NCG implementations are not exact replicas of the original authors' works. We implement only the β_k scalar definitions for each NCG method and compare them independently of any accompanying specialized line search. Consequently, we acknowledge that the results presented in this section may differ from the authors' own observations. Our goal is not to exhaustively compare the methods to each other but instead demonstrate a versatile preconditioning approach that can improve the performance of NCG methods.

Our numerical experiments cover a diverse set of 119 problems from the bound-constrained CUTEst [21] test set, with the number of variables ranging from 2 to 10^5. In all experiments, the NCG iteration limit is set to $1,000$ and the absolute convergence tolerance to $\|g_k\| \leq 10^{-5}$. All tests are performed on a 2018 MacBook Pro with a 3.1 GHz quad-core Intel Core i7 CPU.

We construct performance profiles using the Dolan and Morè methodology [16]. For a given NCG method $c \in C$ solving a CUTEst problem $p \in \mathcal{P}$, the cost measure is defined as

$$t_{p,c} = \text{\# of function evals. to solve problem } p \text{ with method } c.$$

This cost is normalized by the best NCG method for each problem such that

$$r_{p,c} = \frac{t_{p,c}}{\min\{t_{p,\hat{c}} : \hat{c} \in C\}}.$$

Performance of each method is then given by

$$P_c(r_{p,c} \leq \pi : c \in C) = \frac{1}{n_p}\text{size}\{p \in \mathcal{P} : r_{p,c} \leq \pi\},$$

which describes the probability for NCG method $c \in C$ to have a cost ratio $r_{p,c}$ that is within a factor of $\pi \in \mathbb{R}_+$ of the best NCG method. We display this π factor in logarithmic scale to improve visibility of small differences between methods.

We begin by comparing the baseline NCG methods without any preconditioning in Fig. 1. As expected, we observe that Fletcher-Reeves underperforms in the presence of nonconvex problems and an inexact line search. However, we refrain from drawing any other conclusions from the relative performance of more modern NCG methods, since many of them rely on specialized line search approaches we have not implemented.

Fig. 2 shows the performance of the NCG methods with our preconditioner. Unexpectedly, preconditioning degrades the convergence of the Fletcher-Reeves method. However, all remaining methods are improved significantly by preconditioning, with the newest methods deriving the greatest benefit.

Fig. 3 offers a direct base versus preconditioned method comparison of the three NCG methods that have benefited the most from preconditioning. The ++ symbol denotes the variants that include the proposed preconditioner. We observe that the preconditioner significantly accelerates convergence and offers greater robustness on this problem set.

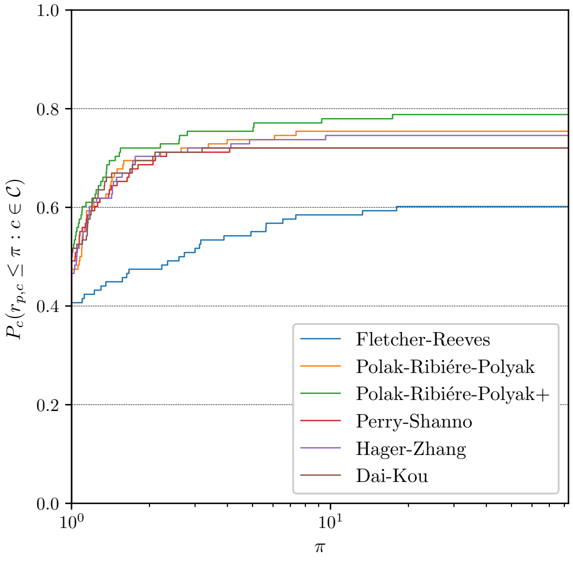

Figure 1. Comparison of selected NCG methods without preconditioning.

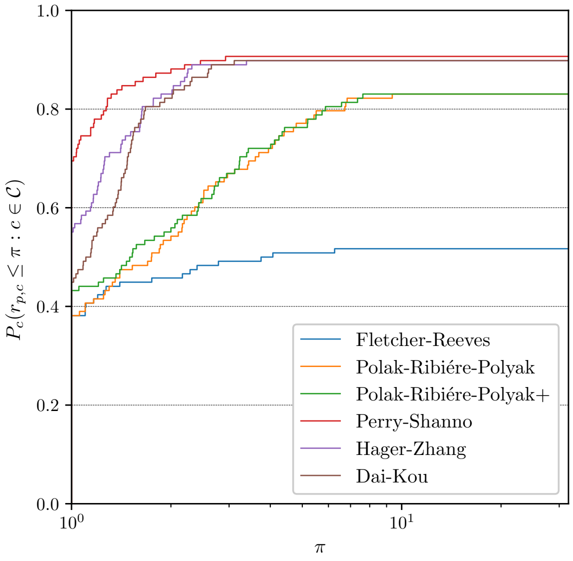

Figure 2. Comparison of selected NCG methods with our preconditioner.

As a common benchmark case, Fig. 4 provides a comparison of the preconditioned methods against the L-BFGS-B method with the history size set to 5 iterations. Our L-BFGS-B implementation

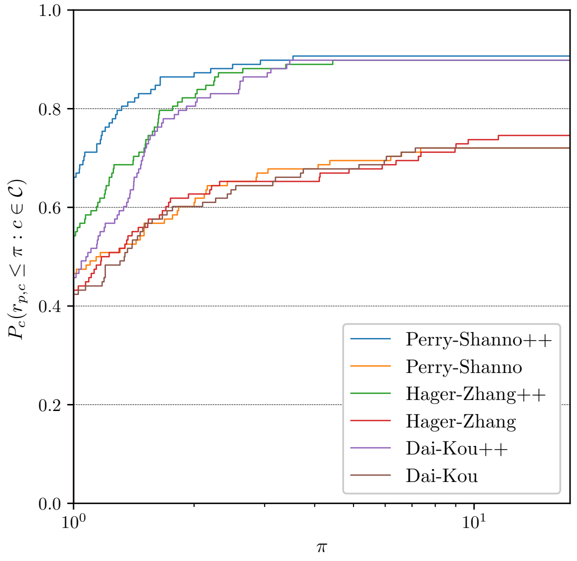

Figure 3. Effect on preconditioning on three best NCG methods.

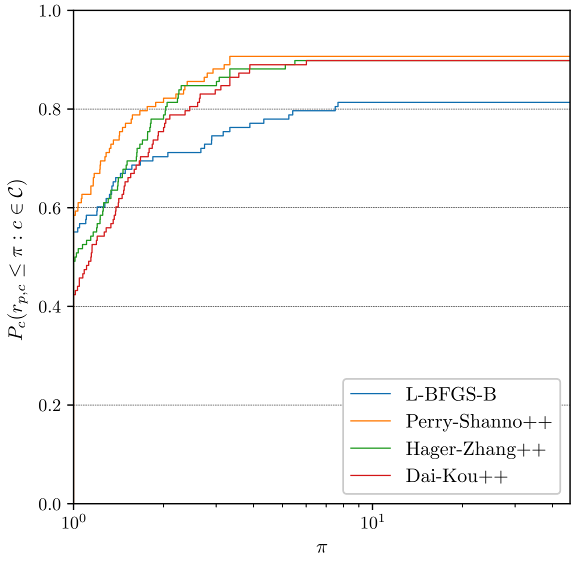

Figure 4. Benchmarking preconditioned NCG methods against L-BFGS-B.

in PETSc/TAO utilizes the same active-set estimation and Moré-Thuente line search as the NCG methods. The preconditioned NCG methods significantly outperform L-BFGS-B while maintaining a smaller memory footprint.

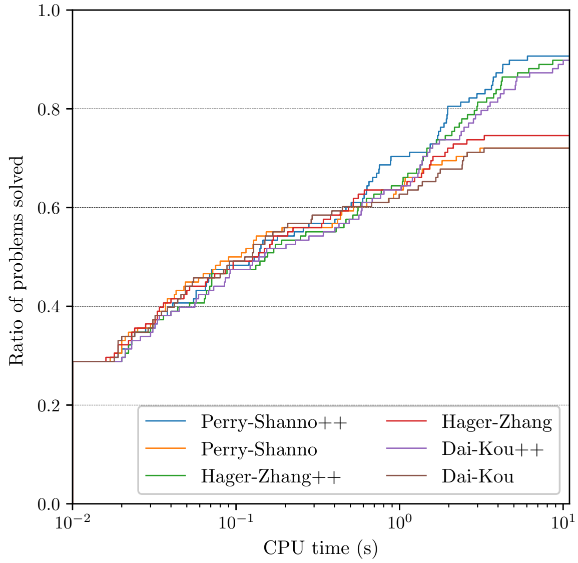

Figure 5. Compute time impact of the preconditioner.

Finally, in Fig. 5, we evaluate the computational cost of the additional algebra operations required by the preconditioner updates and applications. The results show that the cost of updating and applying the preconditioner is offset by the lower number of function evaluations, resulting in virtually no compute time increases compared with the base methods. Since function evaluations for CUTEst problems are relatively cheap and fast, the computational cost of the preconditioner update and application are nearly negligible, and the potential for speedup is significant on problems with expensive function or gradient evaluations.

5 CONCLUSIONS

We have reviewed a selection of classical and modern nonlinear conjugate gradient (NCG) methods, introduced their preconditioned versions, and developed a flexible preconditioner based on the diagonalization of the BFGS formula originally proposed by Gilbert and Lemaréchal for limited-memory quasi-Newton Hessian initializations. Our preconditioned NCG methods are implemented in PETSc/TAO, which allows us to switch between the different NCG formulas andturn preconditioning on and off at runtime.

Our numerical experiments indicate that the proposed preconditioner significantly accelerates convergence and improves robustness at negligible additional algebra cost on all tested NCG methods except for Fletcher-Reeves. Although we have not yet closely analyzed the cause of this behavior for Fletcher-Reeves, we suspect that it may be because of the method's reliance on exact line searches and the error introduced by the small change assumption for the preconditioner (i.e., $C_{k-1} \approx C_k$).

We have also observed that modern NCG methods that have evolved from the Perry-Shanno scheme appear to receive the greatest benefit from preconditioning. It may be worthwhile to study and analyze whether this is due to the shared limited-memory quasi-Newton origin between the preconditioner and the NCG methods themselves or whether modern NCG methods would generally benefit more from other preconditioners as well. In future efforts, we aim to further explore unifying connections between NCG and limited-memory quasi-Newton methods and study a broader preconditioning framework based on the diagonalization of the restricted Broyden class [10, 11, 15].

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program through the FASTMath Institute under Contract No. DE-AC02-06CH11357 at Argonne National Laboratory.

REFERENCES

[1] Gassan S Abdoulaev, Kui Ren, and Andreas H Hielscher. 2005. Optical tomography as a PDE-constrained optimization problem. *Inverse Problems* 21, 5 (2005), 1507.

[2] Caliciotti Andrea, Fasano Giovanni, and Roma Massimo. 2017. Novel preconditioners based on quasi–Newton updates for nonlinear conjugate gradient methods. *Optimization Letters* 11, 4 (2017), 835–853.

[3] Neculai Andrei. 2007. A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. *Applied Mathematics Letters* 20, 6 (2007), 645–650.

[4] Neculai Andrei. 2010. Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. *European Journal of Operational Research* 204, 3 (2010), 410–420.

[5] Satish Balay, Shrirang Abhyankar, Mark F Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D Gropp, Dinesh Kaushik, Matthew G Knepley, Dave A May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. 2018. *PETSc Users Manual*. Technical Report ANL-95/11 - Revision 3.10. Argonne National Laboratory. http://www.mcs.anl.gov/petsc

[6] Dimitri P Bertsekas. 1982. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization* 20, 2 (1982), 221–246.

[7] Peter R Brune, Matthew G Knepley, Barry F Smith, and Xuemin Tu. 2015. Composing scalable nonlinear algebraic solvers. *SIAM Rev.* 57, 4 (2015), 535–565.

[8] A Buckley. 1978. Extending the relationship between the conjugate gradient and BFGS algorithms. *Mathematical Programming* 15, 1 (1978), 343–348.

[9] A Buckley and A LeNir. 1983. QN-like variable storage conjugate gradients. *Mathematical programming* 27, 2 (1983), 155–175.

[10] Richard H Byrd, Dong C Liu, and Jorge Nocedal. 1992. On the behavior of Broyden's class of quasi-Newton methods. *SIAM Journal on Optimization* 2, 4 (1992), 533–557.

[11] Richard H Byrd, Jorge Nocedal, and Ya-Xiang Yuan. 1987. Global convergence of a cass of quasi-Newton methods on convex problems. *SIAM J. Numer. Anal.* 24, 5 (1987), 1171–1190.

[12] Andrea Caliciotti, Giovanni Fasano, and Massimo Roma. 2018. Preconditioned nonlinear conjugate gradient methods based on a modified secant equation. *Appl. Math. Comput.* 318 (2018), 196–214.

[13] Yu-Hong Dai and Cai-Xia Kou. 2013. A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM Journal on Optimization* 23, 1 (2013), 296–320.

[14] Alp Dener, Adam Denchfield, Todd Munson, Jason Sarich, Stefan Wild, Steven Benson, and Lois Curfman McInnes. 2018. *TAO Users Manual*. Technical Report ANL/MCS-TM-322 - Revision 3.10. Argonne National Laboratory. https://www.mcs.anl.gov/petsc

[15] Alp Dener and Todd Munson. 2019. Accelerating limited-memory quasi-Newton convergence for large-scale optimization. In *International Conference on Computational Science (submitted)*.

[16] Elizabeth D Dolan and Jorge J Moré. 2002. Benchmarking optimization software with performance profiles. *Mathematical Programming* 91, 2 (2002), 201–213.

[17] Ioannis Epanomeritakis, Volkan Akçelik, Omar Ghattas, and Jacobo Bielak. 2008. A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion. *Inverse Problems* 24, 3 (2008), 034015.

[18] Mike Fisher, Jorge Nocedal, Yannick Trémolet, and Stephen J Wright. 2009. Data assimilation in weather forecasting: a case study in PDE-constrained optimization. *Optimization and Engineering* 10, 3 (2009), 409–426.

[19] Roger Fletcher and Colin M Reeves. 1964. Function minimization by conjugate gradients. *Comput. J.* 7, 2 (1964), 149–154.

[20] Jean Charles Gilbert and Claude Lemaréchal. 1989. Some numerical experiments with variable-storage quasi-Newton algorithms. *Mathematical Programming* 45, 1-3 (1989), 407–435.

[21] Nicholas IM Gould, Dominique Orban, and Philippe L Toint. 2015. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications* 60, 3 (2015), 545–557.

[22] William W Hager and Hongchao Zhang. 2005. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization* 16, 1 (2005), 170–192.

[23] William W Hager and Hongchao Zhang. 2006. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization* 2, 1 (2006), 35–58.

[24] William W Hager and Hongchao Zhang. 2013. The limited memory conjugate gradient method. *SIAM Journal on Optimization* 23, 4 (2013), 2150–2168.

[25] Magnus Rudolph Hestenes and Eduard Stiefel. 1952. *Methods of conjugate gradients for solving linear systems*. Vol. 49. NBS Washington, DC.

[26] Jorge J Moré and David J Thuente. 1994. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software (TOMS)* 20, 3 (1994), 286–307.

[27] Larry Nazareth. 1979. A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms. *SIAM J. Numer. Anal.* 16, 5 (1979), 794–800.

[28] Jorge Nocedal. 1980. Updating quasi-Newton matrices with limited storage. *Mathematics of computation* 35, 151 (1980), 773–782.

[29] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization. 2nd ed.* (2nd ed. ed.). New York, NY: Springer.

[30] Shmuel S Oren and David G Luenberger. 1974. Self-scaling variable metric (ssvm) algorithms: Part i: Criteria and sufficient conditions for scaling a class of algorithms. *Management Science* 20, 5 (1974), 845–862.

[31] Shmuel S Oren and Emilio Spedicato. 1976. Optimal conditioning of self-scaling variable metric algorithms. *Mathematical Programming* 10, 1 (1976), 70–90.

[32] JM Perry et al. 1977. A class of conjugate gradient algorithms with a two-step variable-metric memory. *Discussion Papers* 269 (1977).

[33] Elijah Polak and Gerard Ribière. 1969. Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle. Série rouge* 3, 16 (1969), 35–43.

[34] Boris T Polyak. 1969. The conjugate gradient method in extremal problems. *U. S. S. R. Comput. Math. and Math. Phys.* 9, 4 (1969), 94–112.

[35] Michael JD Powell. 1984. Nonconvex minimization calculations and the conjugate gradient method. In *Numerical analysis*. Springer, 122–141.

[36] David F Shanno. 1978. On the convergence of a new conjugate gradient algorithm. *SIAM J. Numer. Anal.* 15, 6 (1978), 1247–1257. https://doi.org/10.1137/0715085

[37] Garrett Vanderplaats. 2002. Very large scale optimization. In *8th Symposium on Multidisciplinary Analysis and Optimization*. 4809.