Accelerating Limited-Memory Quasi-Newton Convergence for Large-Scale Optimization

Alp Dener and Todd Munson

Argonne National Laboratory, Argonne, IL 60439, USA

Abstract. Quasi-Newton methods are popular gradient-based optimization methods that can achieve rapid convergence using only first-order derivatives. However, the choice of the initial Hessian matrix upon which quasi-Newton updates are applied is an important factor that can significantly affect the performance of the method. This fact is especially true for limited-memory variants, which are widely used for large-scale problems where only a small number of updates are applied in order to minimize the memory footprint. In this paper, we introduce both a scalar and a sparse diagonal Hessian initialization framework, and we investigate its effect on the restricted Broyden-class of quasi-Newton methods. Our implementation in PETSc/TAO allows us to switch between different Broyden class methods and Hessian initializations at runtime, enabling us to quickly perform parameter studies and identify the best choices. The results indicate that a sparse Hessian initialization based on the diagonalization of the BFGS formula significantly improves the base BFGS methods and that other parameter combinations in the Broyden class may offer competitive performance.

1 Introduction

Quasi-Newton methods are a variation of Newton's method where the Jacobian or the Hessian is approximated using the secant condition. Since their inception in the late-1950s by Davidon [10,11] and Fletcher and Powell [19], quasi-Newton methods have been widely used in solving nonlinear systems of equations, especially in optimization applications. For a comprehensive review of these methods, see Dennis and Moré [15] and Nocedal and Wright [32].

Our interest in quasi-Newton methods is motivated by the computational cost and difficulty in calculating exact Hessians for large-scale or partial differential equation (PDE)-constrained optimization problems. In particular, for reduced-space methods for PDE-constrained problems where the PDE constraint is eliminated via the implicit function theorem, constructing exact second-order information at each iteration requires as many adjoint solutions as the number of optimization variables [33]. Computing Hessian-vector products without computing the Hessian itself can be computationally cheaper [3,24,28], but the matrix-free nature of this approach poses additional difficulties in preconditioning the systems [13,14].

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

Limited-memory quasi-Newton methods circumvent these issues by directly constructing approximations to the inverse Hessian using only first-order information; however, they also typically exhibit slower convergence than truncated-Newton methods [27]. Our goal is to investigate the so-called restricted Broyden class of quasi-Newton methods and develop new strategies to accelerate their convergence in order to minimize the number of function and gradient evaluations.

For a given bound-constrained optimization problem,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x), \\ \text{s.t.} \quad & x_l \leq x \leq x_u, \end{aligned} \tag{1}$$

with $f : \mathbb{R}^n \to \mathbb{R}$ as the objective function and $g_k = \nabla f(x_k)$ as its gradient at the k^{th} iteration, Broyden's method [4] constructs the approximate Hessian with the update formula,

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi(s_k^T B_k s_k) v_k v_k^T, \tag{2}$$

where $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$,

$$v_k = \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k},$$

and ϕ is a scalar parameter. In the active-set approach, the inverse of this approximate Hessian is applied to the negative gradient and a projected line search is performed along the resulting step direction. The process repeats until the projected gradient norm is reduced below a prescribed tolerance or an iteration limit is reached. Many methods are available for estimating the index set of active variables [2,7,26,31]; however, our focus in the present work is the quasi-Newton approximation.

Methods in the Broyden class are defined by the different scalar values of the parameter ϕ. The restricted Broyden class, in particular, limits the choice of ϕ to the range $[0, 1]$, which guarantees that the updates are symmetric positive-definite provided that $s_k^T y_k > 0$. The most well-known methods of this type are the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [5,18,22,34] and Davidon-Fletcher-Powell (DFP) [11] methods, which are recovered with $\phi = 0$ and $\phi = 1$, respectively. The restricted Broyden-class formulation in (2) can also be rewritten as a convex combination of the BFGS and DFP methods, such that

$$B_{k+1} = (1 - \phi) B_{k+1}^{BFGS} + \phi B_{k+1}^{DFP}. \tag{3}$$

The limited-memory variant of the restricted Broyden update takes the form

$$B_{k+1} = B_0 + \sum_m \left[\frac{y_m y_m^T}{y_m^T s_m} - \frac{B_m s_m s_m^T B_m}{s_m^T B_m s_m} + \phi(s_m^T B_m s_m) v_m v_m^T \right], \tag{4}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

where B_0 is an initial Hessian, $m = \{\max(0, k - M + 1) \dots k\}$ is the index set for the sequence of quasi-Newton updates, and M is the maximum number of updates to be applied to the initial Hessian (i.e.: the "memory" size). Since Newton-type optimization algorithms seek to apply the inverse of the Hessian matrix to the gradient, the Sherman-Morrison-Woodbury formula [25] is utilized to update an approximation to the inverse of the Hessian matrix directly, such that

$$H_{k+1} = H_0 + \sum_m \left[\frac{s_m s_m^T}{s_m^T y_m} - \frac{H_m y_m y_m^T H_m}{y_m^T H_m y_m} + \psi_m (y_m^T H_m y_m) w_m w_m^T \right], \quad (5)$$

where

$$w_m = \frac{s_m}{y_m^T s_m} - \frac{H_m y_m}{y_m^T H_m y_m}$$

and

$$\psi_m = \frac{(1-\phi)(y_m^T s_m)^2}{(1-\phi)(y_m^T s_m)^2 + \phi (y_m^T H_m y_m)(s_m^T B_m s_m)}.$$

In practice, the limited-memory formula is often implemented in a matrix-free fashion where only M of the (s_k, y_k) vector pairs are stored and the action of the approximate Hessian – the product between the approximate inverse of the Hessian and a given vector – is defined by multiplying (5) with a vector. With this approach, the H_m terms inside the summation recurse into their own quasi-Newton formulas and are implemented with nested loops. Some special cases, such as the BFGS formula, can be unrolled into two independent loops that minimize the number of operations [8]. For a more comprehensive look at this approach, we refer the reader to [17].

In this paper, we investigate a framework for constructing scalar or sparse diagonal choices for the initial Hessian B_0 in (4). Our software implementation of (5) including the sparse Hessian initialization is available as part of PETSc/TAO Version 3.10 [1,12]. We leverage PETSc extensibility to explore values of ϕ and other parameters associated with the initial Hessian at runtime to study the convergence and performance of our approach on the complete set of 119 bound-constrained CUTEst test problems [23].

2 Hessian Initialization

The choice of a good initial Hessian H_0 in limited-memory quasi-Newton methods is critically important to the quality of the Hessian approximation. It has been well documented that the scaling of the approximate Hessian depends on this choice and dramatically affects convergence [21,29]. Our goal is to develop a modular framework that can generate effective initializations that preserve the symmetric positive-definite property of the restricted Broyden class methods and are easily invertible as part of efficient matrix-free applications of limited-memory quasi-Newton formulas (e.g., two-loop L-BFGS inversion [8]). To that

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

end, we construct our initial Hessians in scalar and sparse diagonal forms, the latter of which is based on the restricted Broyden-class formula.

The first step is to address the starting point x_0 when there is no accumulated information with which to construct either scalar or diagonal initializations. It is common for the matrix at iteration 0 to be set to a multiple $B_0 = \rho_0 I$ of the identity that promotes acceptance of the unit-step length by the line search; however, no good general strategy exists for choosing a suitable value for ρ_0. Gilbert and Lemaréchal [20] proposed $\rho_0 = 2\Delta/||g_0||_2^2$, where Δ is a user-supplied parameter that represents the expected decrease in $f(x)$ at the first iteration. We use

$$\rho_0 = \begin{cases} 2/||g_0||_2^2 & \text{for } f(x_0) = 0 \\ 2|f(x_0)|/||g_0||_2^2 & \text{otherwise,} \end{cases} \tag{6}$$

which has proven to be an effective choice across our numerical experiments and eliminates a user-defined parameter from the algorithm. This choice also appears to be related to more recent investigations into scaled gradient descent steps with an a priori estimation of the local minimum [9], with $f(x^*) = 0$ where x^* is the minimizer. Both the scalar and sparse diagonal B_0 constructions we introduce below leverage this initial scalar choice.

2.1 Scalar Formulation

Scalar Hessian initializations restrict the estimate to a positive scalar multiple of the identity matrix, such that $B_0 = \rho_k I$ during iteration k. For BFGS matrices, a common and well-understood choice has been

$$\rho_k = \frac{y_k^T y_k}{y_k^T s_k}, \tag{7}$$

which is an approximation to an eigenvalue of $\nabla^2 f(x_k)$ [32].

Our scalar construction begins with the recognition that (7) is the positive solution to the scalar minimization problem

$$\rho_k = \underset{\rho>0}{\operatorname{argmin}} ||\frac{1}{\rho} y_k - s_k||_2^2, \tag{8}$$

which is also a least-squares solution to the secant equation $B_0^{-1} y_k = s_k$ with $B_0 = \rho I$ [6]. We then introduce a new parameter $\alpha \in [0, 1]$ such that $B_0 = \rho^{2\alpha-1} I$ and we solve the modified least-squares problem,

$$\rho_k = \underset{\rho>0}{\operatorname{argmin}} ||\rho^{-\alpha} y_k - \rho^{-(1-\alpha)} s_k||_2^2. \tag{9}$$

After constructing the optimality conditions and solving for ρ, we arrive at the following values:

1. If $\alpha = 0$, then

$$\rho_k = \frac{y_k^T s_k}{s_k^T s_k}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

2. If $\alpha = 1/2$, then

$$\rho_k = \sqrt{\frac{y_k^T y_k}{s_k^T s_k}}$$

3. If $\alpha = 1$, then

$$\rho_k = \frac{y_k^T y_k}{y_k^T s_k}$$

   Note: this value corresponds to the commonly used eigenvalue estimate in (7).
4. Otherwise, ρ_k is the positive root of the quadratic equation,

$$\alpha(y_k^T y_k)\rho^2 - (2\alpha - 1)(y_k^T s_k)\rho + (\alpha - 1)(s_k^T s_k) = 0.$$

Since $s_k^T s_k$ and $y_k^T y_k$ cannot be negative and are zero only for a zero step length, the scalar Hessian approximation preserves symmetric positive-definiteness for any (s_k, y_k) update that satisfies the Wolfe conditions.

2.2 Sparse Diagonal Formulation

The sparse diagonal formulation constructs an initial Hessian as a diagonal matrix, $B_0 = \text{diag}(b_k)$, at iteration k defined by and stored as the vector of diagonal entries b_k. Specifically, we construct this diagonal vector using the full-memory restricted Broyden formula in (2), such that

$$\begin{aligned} b_{k+1} = b_k &+ (1-\theta)\left[\frac{y_k \circ y_k}{y_k^T s_k} - \frac{(b_k \circ s_k)^2}{s_k^T (b_k \circ s_k)}\right] \\ &+ \theta\left[\left(\frac{1}{y_k^T s_k} + \frac{s_k^T (b_k \circ s_k)}{(y_k^T s_k)^2}\right)(y_k \circ y_k) - \frac{2(s_k \circ b_k \circ y_k)}{y_k^T s_k}\right]. \end{aligned} \tag{10}$$

This expression is the expanded version of the convex combination notation in (3), where $(1-\theta)$ and θ correspond to the BFGS and DFP components, respectively. Since we compute only diagonal entries, all matrix-vector products have been replaced by Hadamard products with the previous diagonal. As in Broyden's method, $\theta = 0$ corresponds to a pure BFGS formulation, while $\theta = 1$ recovers DFP.

This initialization is a full-memory approach; the diagonal entries of B_0 are explicitly stored in b_k and updated with every accepted new iterate. Consequently, B_0 contains information from all iterates traversed in the optimization instead of only the last M iterates stored for the limited-memory formula, but without the large memory cost of storing dense matrices.

Gilbert and Lemaréchal have explored a similar initial Hessian using diagonalizations of the BFGS formula only [20] and reported the need to rescale the diagonal to account for the inability to rapidly modify it in large steps. To that

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

end, we redefine the initial Hessian as $B_0 = \sigma_k^{2\alpha-1}\text{diag}(b_k)$ and compute the rescaling factor σ_k by seeking the least-squares solution to the secant equation, $B_0^{-1}y_k = s_k$, such that

$$\sigma_k = \underset{\sigma}{\text{argmin}}||\sigma^{-\alpha}(b_k^{-0.5} \circ y_k) - \sigma^{-(1-\alpha)}(b_k^{0.5} \circ s_k)||_2^2. \tag{11}$$

Note that the expression inside the l_2-norm is equivalent to the secant equation in residual form, restructured so that the solution can be more easily expressed in the form of quadratic roots. The solution yields the following values:

1. If $\alpha = 0$, then
$$\sigma_k = \frac{y_k^T s_k}{s_k^T(b_k \circ s_k)}.$$
2. If $\alpha = 1/2$, then
$$\sigma_k = \sqrt{\frac{y_k^T(b_k^{-1} \circ y_k)}{s_k^T(b_k \circ s_k)}}.$$
3. If $\alpha = 1$, then
$$\sigma_k = \frac{y_k^T(b_k^{-1} \circ y_k)}{y_k^T s_k}.$$
4. Otherwise, ρ_k is the positive root of the quadratic equation,
$$\alpha\left[y_k^T(b_k^{-1} \circ y_k)\right]\sigma^2 - (2\alpha-1)(y_k^T s_k)\sigma + (\alpha-1)\left[s_k^T(b_k \circ s_k)\right] = 0.$$

As with the scalar initialization, the sparse diagonal B_0 remains positive definite for any (s_k, y_k) pair that satisfies the Wolfe conditions. Nonetheless, we have encountered cases where numerical problems surface in finite precision arithmetic. Therefore, we safeguard all the methods by checking whether the denominators are equal to zero and setting their value to a small constant, 10^{-8}, if so.

3 Numerical Studies

We now investigate the numerical performance of our proposed scalar and sparse Hessian initializations and study the parameter space of the user-controlled scalar factors to determine useful recommendations. Our quasi-Newton implementation in PETSc/TAO utilizes an active-set estimation based on the work of Bertsekas [2] and is discussed in further detail in the TAO manual [12]. The step direction is globalized via a projected Moré-Thuente line search [30], which is capable of taking step lengths greater than 1.

Our numerical experiments are based on 119 bound-constrained problems from the CUTEst test set [23], covering a diverse range of problems from 2 to

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

10^5 variables. In all cases presented in this section, we set the quasi-Newton memory size to $M = 5$ updates, limit the maximum number of iterations to $1,000$, and require convergence to an absolute tolerance of $||g^*||_2 \leq 10^{-6}$.

Performance profiles are constructed by using the methodology proposed by Dolan and Moré [16]. For a given CUTEst problem $p \in \mathcal{P}$ and solver configuration $c \in \mathcal{C}$, we define a cost measure

$$t_{p,c} = \text{function evaluations required to solve problem } p \text{ with configuration } c$$

and normalize it by the best configuration for each problem, such that

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

Performance of each configuration is then given by

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

which describes the probability for configuration $c \in \mathcal{C}$ to have a cost ratio $r_{p,c}$ that is within a factor of $\pi \in \mathbb{R}$ of the best configuration.

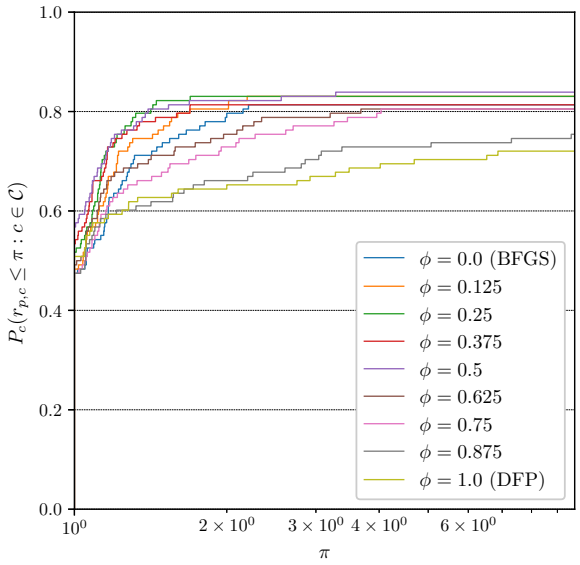
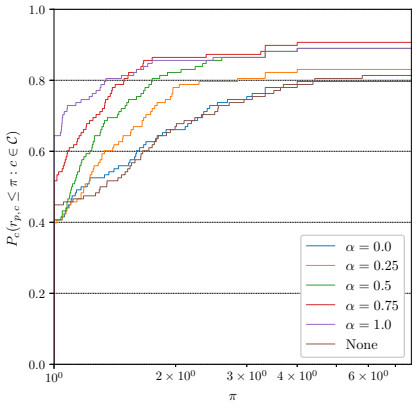


Fig. 1. Parameter study for restricted Broyden convex combination factor.

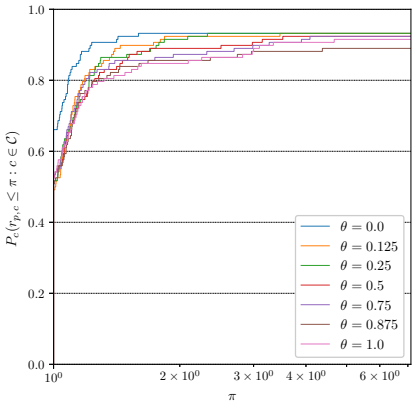
We begin our analysis with a sweep through the ϕ parameter space in the restricted Broyden updates in Fig. 1. For this study we turn off all Hessian

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

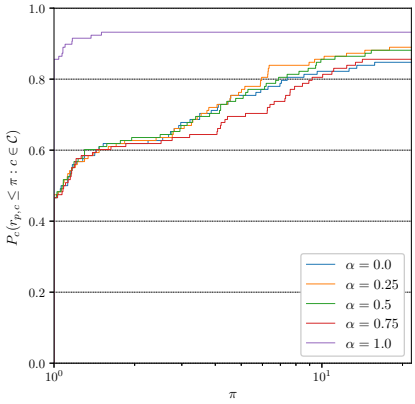
initialization (i.e., $H_0 = I$) and investigate only the relative performances of the raw Broyden-class methods. Note that $\phi = 0$ and $\phi = 1$ are included, which correspond to the BFGS and DFP methods, respectively. The results indicate that ϕ values in range $(0, 0.5]$ produce Hessian approximations that outperform BFGS, with $\phi = 0.5$ yielding the best performance.



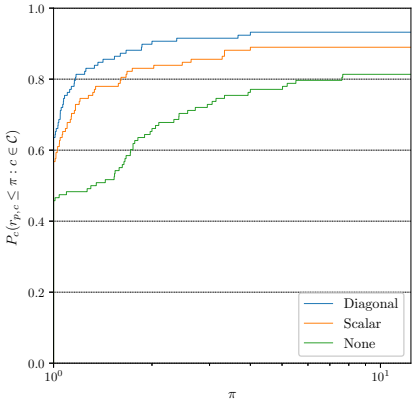
(a) Scalar H_0 initialization.



(b) Diagonal H_0 initialization.



(c) Rescaling of diagonal H_0.



(d) Comparison of best H_0 configurations.

Fig. 2. Parameter study for Hessian initialization with the L-BFGS update.

For the next study, we used the L-BFGS method as the basis for analyzing the effects of different H_0 initialization methods. We start with the scalar H_0 in Fig. 2a and investigate the effect of the α parameter. Here, setting $\alpha = 1.0$ recovers

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

the widely used $y_k^T s_k / y_k^T y_k$ factor that approximates an eigenvalue of $\nabla^2 f(x_k)^{-1}$. As expected, this choice yields the best scaled Hessian approximation and the lowest number of function evaluations on most problems. Surprisingly, however, the results indicate that $\alpha = 0.75$ remains competitive in cost, while converging on two additional test problems.

Fig. 2b shows the relative performances of the sparse diagonal H_0 initialization. For this study, we fix the rescaling parameter to $\alpha = 1.0$. We investigate the effect of the rescaling parameter below. Not surprisingly, the results indicate that $\theta = 0.0$, which corresponds to a "full-memory" BFGS diagonal, produce the best convergence improvement for L-BFGS updates.

Fixing $\theta = 0.0$ as the best case for the sparse diagonal H_0, we now perform a parameter sweep through the rescaling term in Fig. 2c. The results indicate that $\alpha = 1.0$ produces the most well-scaled initialization for the sparse diagonal case. Additional experiments not shown have failed to recover a better diagonal initialization at different α and θ combinations.

Table 1. Select problems for comparison of H_0 methods in L-BFGS.

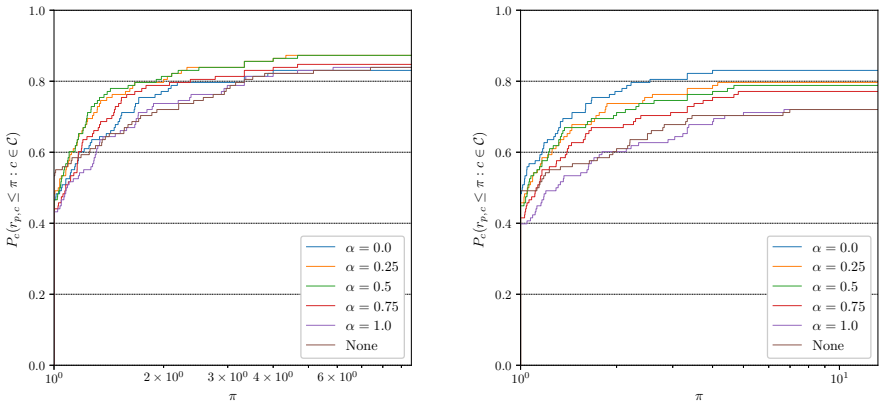
	# of vars.			**# of iterations**		
	Total	**Free**	**Active**	$H_0 = I$	$H_0 = \rho_k I$	$H_0 = \sigma_k \text{diag}(h_k)$
EXPLIN	1200	52	1148	213	170	109
EXPQUAD	1200	1119	81	N/A	314	113
BDEXP	5000	5000	0	36	18	16
TORSIONB	5625	3624	1852	165	151	132
JNLBRNGA	10000	6359	3641	N/A	327	299
OBSTCLBL	10000	7057	2943	131	130	113

In Fig. 2d, we compare the best-case configurations for both H_0 initialization types to the raw L-BFGS results. The sparse diagonal initialization enables L-BFGS to solve over 90% of the bound-constrained CUTEst test set in under $1,000$ iterations and accelerates convergence on all problems, offering a significant improvement over both the scalar and identity initialization methods. Statistics for a subset of the problems from this plot are available in Table 1.

We also explore additional H_0 parameters to accelerate convergence of DFP and the best Broyden method at $\phi = 0.5$. Fig. 3a and Fig. 3b show the parameter study for the scalar initialization in the Broyden and DFP methods, respectively. The best case for DFP at $\alpha = 0$ yields a scalar H_0 that is the dual of the best scalar term for BFGS (i.e.: interchanging roles for s_k and y_k). This mimics the duality between the DFP and BFGS formulas themselves. Additionally, the best case for Broyden's method matches the α parameter to the convex combination term of $\phi = 0.5$. This observation suggests that the best scalar initialization parameter, α, for any member of the Broyden-class method may be the same as the convex combination term ϕ that defines the method.

In Fig. 4, we compare these scalar H_0 terms with one another and against the best BFGS initializations above. Results show that selecting the correct scalar

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39



(a) Broyden method with $\phi = 0.5$.

(b) DFP method.

Fig. 3. Scalar Hessian initialization for Broyden ($\phi = 0.5$) and DFP methods.

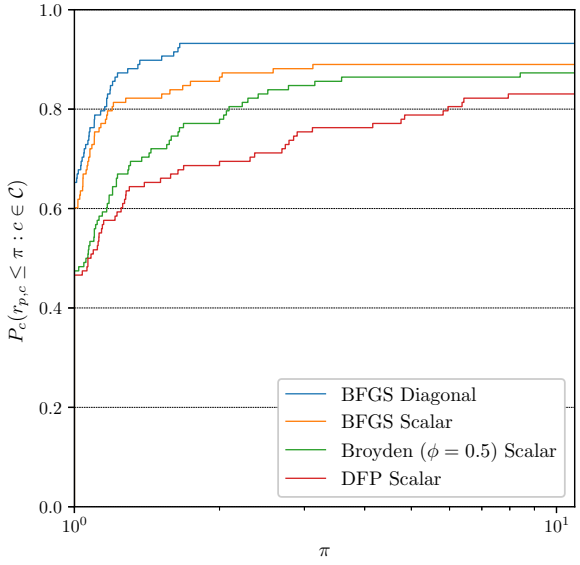


Fig. 4. Comparison of best H_0 definitions for selected quasi-Newton methods.

H_0 for each method significantly improves all quasi-Newton methods tested and makes the $\phi = 0.5$ Broyden method competitive with BFGS in the number of problems solved. Our observations indicate that a more comprehensive pa-

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

rameter study may reveal other Broyden-class methods with different convex combination and Hessian initialization terms that offer competitive or better performance than BFGS on some problems. Preliminary numerical experiments we have conducted suggest that the DFP method does not benefit from a diagonal Hessian initialization; however, we aim to utilize our flexible quasi-Newton framework to explore the diagonal H_0 with other Broyden-class methods in the near future.

4 Conclusions

We have introduced a flexible framework for constructing both scalar and sparse diagonal, positive-definite Hessian initializations for limited-memory quasi-Newton methods based on the restricted Broyden-class updates. Our implementation in PETSc/TAO allows us to rapidly change parameters and shift between different members of the Broyden-class methods and select the form for the Hessian initializations at runtime.

Our numerical experiments indicate that intermediate values of ϕ in the Broyden-class outperform the base BFGS formula for a significant subset of the bound-constrained CUTEst problems. We also compare different scalar initializations for different quasi-Newton methods; the results suggest that the best possible α parameter in our H_0 formulation tracks with the convex combination parameter ϕ that defines members of the Broyden-class.

We demonstrate that the diagonal Hessian initialization successfully accelerates BFGS convergence at minimal additional memory and algebra cost compared with scalar initializations. Our preliminary experience testing similar initializations with DFP and other Broyden-class methods suggests that other parameter values may reveal alternative quasi-Newton methods that are competitive with BFGS on large-scale optimization problems. We hope to leverage our flexible Broyden-class quasi-Newton algorithm to further investigate these possibilities in the future.

Acknowledgments

This work was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

References

1. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.10, Argonne National Laboratory (2018), http://www.mcs.anl.gov/petsc
2. Bertsekas, D.P.: Projected newton methods for optimization problems with simple constraints. SIAM Journal on control and Optimization **20**(2), 221–246 (1982)
3. Biros, G., Ghattas, O.: Parallel lagrange–newton–krylov–schur methods for pde-constrained optimization. part i: The krylov–schur solver. SIAM Journal on Scientific Computing **27**(2), 687–713 (2005)
4. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. Mathematics of computation **19**(92), 577–593 (1965)
5. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms 1. general considerations. IMA Journal of Applied Mathematics **6**(1), 76–90 (1970)
6. Burke, J.V., Wiegmann, A., Xu, L.: Limited memory bfgs updating in a trust-region framework. SIAM Journal on Optimization (2008)
7. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing **16**(5), 1190–1208 (1995)
8. Byrd, R.H., Nocedal, J., Schnabel, R.B.: Representations of quasi-newton matrices and their use in limited memory methods. Mathematical Programming **63**(1-3), 129–156 (1994)
9. D'Alves, C.: A Scaled Gradient Descent Method for Unconstrained Optimization Problems With A Priori Estimation of the Minimum Value. Ph.D. thesis (2017)
10. Davidon, W.C.: Variable metric method for minimization, argonne natl. Tech. rep., Argonne National Laboratory (1959)
11. Davidon, W.C.: Variable metric method for minimization. SIAM Journal on Optimization **1**(1), 1–17 (1991)
12. Dener, A., Denchfield, A., Munson, T., Sarich, J., Wild, S., Benson, S., McInnes, L.C.: Tao users manual. Tech. Rep. ANL/MCS-TM-322 - Revision 3.10, Argonne National Laboratory (2018), https://www.mcs.anl.gov/petsc
13. Dener, A., Hicken, J.E.: Matrix-free algorithm for the optimization of multidisciplinary systems. Structural and Multidisciplinary Optimization **56**(6), 1429–1446 (2017)
14. Dener, A., Hicken, J.E., Kenway, G.K., Martins, J.: Enabling modular aerostructural optimization: Individual discipline feasible without the jacobians. In: 2018 Multidisciplinary Analysis and Optimization Conference. p. 3570 (2018)
15. Dennis, Jr, J.E., Moré, J.J.: Quasi-newton methods, motivation and theory. SIAM review **19**(1), 46–89 (1977)
16. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical programming **91**(2), 201–213 (2002)
17. Erway, J.B., Marcia, R.F.: On solving large-scale limited-memory quasi-newton equations. Linear Algebra and its Applications **515**, 196–225 (2017)
18. Fletcher, R.: A new approach to variable metric algorithms. The computer journal **13**(3), 317–322 (1970)
19. Fletcher, R., Powell, M.J.: A rapidly convergent descent method for minimization. The computer journal **6**(2), 163–168 (1963)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39

20. Gilbert, J.C., Lemaréchal, C.: Some numerical experiments with variable-storage quasi-newton algorithms. Mathematical programming **45**(1-3), 407–435 (1989)
21. Gill, P.E., Leonard, M.W.: Reduced-hessian quasi-newton methods for unconstrained optimization. SIAM Journal on Optimization **12**(1), 209–237 (2001)
22. Goldfarb, D.: A family of variable-metric methods derived by variational means. Mathematics of computation **24**(109), 23–26 (1970)
23. Gould, N.I., Orban, D., Toint, P.L.: Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Computational Optimization and Applications **60**(3), 545–557 (2015)
24. Haber, E., Ascher, U.M.: Preconditioned all-at-once methods for large, sparse parameter estimation problems. Inverse Problems **17**(6), 1847 (2001)
25. Hager, W.W.: Updating the inverse of a matrix. SIAM review **31**(2), 221–239 (1989)
26. Hager, W.W., Zhang, H.: A new active set algorithm for box constrained optimization. SIAM Journal on Optimization **17**(2), 526–557 (2006)
27. Hicken, J., Alonso, J.: Comparison of reduced-and full-space algorithms for pde-constrained optimization. In: 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. p. 1043 (2013)
28. Hinze, M., Pinnau, R.: Second-order approach to optimal semiconductor design. Journal of optimization theory and applications **133**(2), 179–199 (2007)
29. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Mathematical programming **45**(1-3), 503–528 (1989)
30. Moré, J.J., Thuente, D.J.: Line search algorithms with guaranteed sufficient decrease. ACM Transactions on Mathematical Software (TOMS) **20**(3), 286–307 (1994)
31. Moré, J.J., Toraldo, G.: On the solution of large quadratic programming problems with bound constraints. SIAM Journal on Optimization **1**(1), 93–113 (1991)
32. Nocedal, J., Wright, S.J.: Numerical optimization 2nd (2006)
33. Papadimitriou, D., Giannakoglou, K.: Direct, adjoint and mixed approaches for the computation of hessian in airfoil design problems. International journal for numerical methods in fluids **56**(10), 1929–1943 (2008)
34. Shanno, D.F.: Conditioning of quasi-newton methods for function minimization. Mathematics of computation **24**(111), 647–656 (1970)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_39