



A faster Broyden method

Eric Kvaalen

► To cite this version:

Eric Kvaalen. A faster Broyden method. BIT Numerical Mathematics, 1991, 31 (2), pp.369-372. 10.1007/BF01931297 . hal-04066969

HAL Id: hal-04066969

<https://hal.science/hal-04066969v1>

Submitted on 13 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

A FASTER BROYDEN METHOD

ERIC KVAALLEN

IMI Institute for Research and Development, Box 313, Haifa 31002, Israel.

Abstract.

A variation of the Broyden update is found to require less operations and to work as well as the usual Broyden update.

Keywords: Broyden method, systems of equations, roots, quasi-Newton methods.

1. Introduction.

In C. G. Broyden's 1965 paper "A Class of Methods for Solving Nonlinear Simultaneous Equations," [1] two such methods were given. Method One has become the accepted "Broyden method," whereas Method Two was not discussed further because Broyden found that "successive step vectors . . . were identical and further progress became impossible." The aim of the present note is to indicate that this conclusion was apparently invalid. Broyden's second method has been tried on several types of problems (including those given by Broyden) and has encountered no such difficulty. In fact, the second method has at least one advantage over the usual Broyden method in that it requires less operations to do the update. A shortcut is also suggested which can reduce the computation to about half of the usual method. The number of iterations needed to converge may be different from Method One, but in comparisons there has usually been no difference. Broyden's second method has been used successfully in the past for solving systems of ordinary differential equations by Brown et al. [2].

2. Broyden's two methods.

Our problem is to find a vector x which satisfies

$$(2.1) \quad f(x) = 0$$

where f is a vector-valued function. In both Broyden methods, the i th approximation of the solution, x_i , is replaced by a better one, namely

$$(2.2) \quad x_{i+1} = x_i + t_i H_i f_i$$

where $f_i = f(x_i)$, H_i is the negative inverse of an approximation B_i of the Jacobian matrix of the function evaluated near the solution or near x_i . Then if t_i is equal to one, and the approximations are good, the procedure will converge rapidly to the solution. After each iteration, the matrix H_i is updated to H_{i+1} to make it a better approximation of the negative inverse of the Jacobian. Method One and Method Two differ in how this update is done.

In both methods, H_{i+1} satisfies two conditions. First, it is given by the difference of H_i and a rank-one matrix, that is, one that can be written as the outer product of two vectors:

$$(2.3) \quad H_{i+1} = H_i - uv^T$$

(in which u and v actually depend on i). Second, H_{i+1} should express the relationship between the change in x and the change in f . That is,

$$(2.4a) \quad H_{i+1}(f_{i+1} - f_i) = x_i - x_{i+1}, \quad \text{or}$$

$$(2.4b) \quad H_{i+1} \Delta f = -\Delta x.$$

This gives, with (2.3), $u(v^T \Delta f) = \Delta x + H_i \Delta f$ or

$$(2.5) \quad u = (\Delta x + H_i \Delta f) / (v^T \Delta f).$$

From this it is seen that v can be almost arbitrary (it cannot be orthogonal to Δf) and u will then depend on v . In Method One, v is chosen to be $H_i^T \Delta x$ while for Method Two it is chosen to be Δf .

From these formulae one can see that the number of multiplications and additions involved in the calculation of the update for Method One with n variables is about $3n^2$, whereas in Method Two it is only about $2n^2$.

At this point, a comment should be made regarding the implementation of either of these methods. The approximately n^2 multiplications and additions normally needed for finding the next value of x by Equation 2.2 can actually be avoided. First note that the term $H_i \Delta f$ needed in the calculation of u can be calculated as $H_i f_{i+1} - H_i f_i$, the latter part of which has already been found. If we save $H_i f_{i+1}$, we can find $H_{i+1} f_{i+1}$

$$(2.6) \quad H_{i+1} f_{i+1} = H_i f_{i+1} - u(v^T f_{i+1}).$$

Equation 2.6 involves only about $2n$ multiplications and additions.

In Method One without Equation 2.6, the number of multiplications and additions is of order $4n^2$, whereas in Method Two with Equation 2.6, it is of order $2n^2$ per iteration. The use of these two improvements thus halves this basic amount of

computation. However, if n is small or if much time is needed in the evaluation of f , then this improvement is of minor importance.

3. Test cases.

Broyden gives the results of ten sample problems, but explicitly states only six of them. Each of these six has been retried with Method Two and found to converge in the same number of iterations as with Method One, except Case 10 which does not converge with either method due to a poor initial guess. The method proposed here was faster than the usual method in each case, ranging from 6% faster for a problem of two variables, to 70% faster for a problem of twenty variables. In both methods the initial Jacobian was estimated by finite differences.

As a further test, the problems given by Moré and Cosnard [3] were tried. In every case in which Method One was able to solve the problem without adjusting the step size or restarting, Method Two was also able to do so in the same number of iterations. For those problems that could not be solved so easily, a programme was used which would cut the step size in half whenever the original step caused the norm of the function to increase, and which also recalculated the Jacobian when “in trouble.” This algorithm succeeded in solving more of the problems, using both methods, without much difference in number of iterations. Only in one problem (“Brown’s almost linear problem”), with starting point very far from the solution, did Method Two fail (due to floating point overflow or singular Jacobian), while Method One succeeded after many iterations. But with the original starting point, Method Two succeeded, and with less iterations.

4. The invariance of Broyden’s second method with linear transformation.

There is a further property of Broyden’s second method, having to do with the scaling of the variables, which may or may not be considered an advantage over the first method. In Method One, if one transforms the problem by performing a linear transformation on the independent variables, then the iterations will not follow the same path as before. However, with Method Two, the path will be the same (but transformed). It should be pointed out that Method One has the same property with respect to linear transformation of the function.

5. Conclusions.

Broyden’s second method, coupled with an improved algorithm, seems to have advantages over his first method, with no apparent disadvantages. In short, Broyden’s second method should be given a second chance.

Acknowledgement.

I would like to thank C. G. Broyden for his interest and for conversations and correspondence regarding this note. Thanks also to Mordochai Shacham for his suggestions.

REFERENCES

1. Broyden, C. G., *A class of methods for solving nonlinear simultaneous equations*, Math. of Computations, Vol. 19, pages 577–593 (1965).
2. P. N. Brown, A. C. Hindmarsh and H. F. Walker, *Experiments with quasi-Newton methods in solving stiff ODE systems*, SIAM J. Sci. Stat. Comp., Vol. 6, No. 2, pages 297–313 (April 1985).
3. J. J. Moré and M. Y. Cosnard, *Numerical solution of nonlinear equations*, ACM Trans. on Mathematical Software, Vol. 5, No. 1, pages 64–85 (March 1979).