Available online at www.sciencedirect.com

SCIENCE DIRECT

ELSEVIER

Journal of Computational and Applied Mathematics 169 (2004) 161–169

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

A modified Newton method with cubic convergence: the multivariate case

H.H.H. Homeier[a,b,*]

[a]*Institut für Physikalische und Theoretische Chemie, Universität Regensburg, D93040 Regensburg, Germany*
[b]*science+computing ag, Ingolstädter Str. 22, D80807 München, Germany*

Received 17 August 2003; received in revised form 9 November 2003

Abstract

Recently, a modification of the Newton method for finding a zero of a univariate function with local cubic convergence has been introduced. Here, we extend this modification to the multi-dimensional case, i.e., we introduce a modified Newton method for vector functions that converges locally cubically, without the need to compute higher derivatives. The case of multiple roots is not treated. Per iteration the method requires one evaluation of the function vector and solving two linear systems with the Jacobian as coefficient matrix, where the Jacobian has to be evaluated twice. Since the additional computational effort is nearly that of an additional Newton step, the proposed method is useful especially in difficult cases where the number of iterations can be reduced by a factor of two in comparison to the Newton method. This much better convergence is indeed possible as shown by a numerical example. Also, the modified Newton method can be advantageous in cases where the evaluation of the function is more expensive than solving a linear system with the Jacobian as coefficient matrix. An example for this is given where numerical quadrature is involved. Finally, we discuss shortly possible extensions of the method to make it globally convergent.

MSC: 41A25; 65D99

Keywords: Rootfinding; Newton method; Generalized Bloch equation; Coupled algebraic equations; Affine-invariant Newton technique; Hybrid method

Nonlinear equation systems are of immense importance for applications in many sciences and engineering. For instance, large examples of such systems arise from coupled-cluster equations and also from the generalized Bloch equation in quantum chemical ab initio calculations where also

[*] Corresponding author. science+computing ag, Ingolstädter Str. 22, D80807 München, Germany. Tel.: +49-171-629-0224; fax: +49-89-356-386737.

E-mail address: herbert.homeier@na-net.ornl.gov (H.H.H. Homeier).

0377-0427/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2003.12.041

special solution methods have been developed (see, e.g., [12–14,17] and references therein). Despite the fact that this field is researched since decades (see [19] and references therein), there is still room for new methods. There are problems that are difficult to tackle using standard solvers [3–8,12–14].

As a generalization of the scalar case [11], we introduce a modified Newton method. Under suitable conditions it is proved to converge locally cubically. We also present some numerical examples. Classes of problems are sketched where the modified method might be advantageous. Finally, we discuss shortly how global convergence could be achieved in analogy to the affin-invariant Newton techniques (ANT) of Deuflhard and coworkers [9,10,15,16].

We treat methods for solving the equation $F(x^{(*)})=0$ where $F:\mathbb{R}^d\to\mathbb{R}^d$ is a vector function.

The Newton method uses the iteration function

$$\Phi_F(x)=x-A_F(x)F(x), \tag{1}$$

where $A_F(x)=[J_F(x)]^{-1}$ is the inverse of the Jacobian of F,

$$J_F(x)=DF(x)=\left(\left.\frac{\partial F_\alpha}{\partial x_\beta}\right|_x\right),\quad \alpha,\beta=1..d \tag{2}$$

and thus, an iteration sequence $x^{(n+1)}=\Phi_F(x^{(n)})$ is obtained for some starting value $x^{(0)}$. Since $F(x^{(*)})=0$ entails $D\Phi_F(x^{(*)})=0$, the convergence to the fixed point $x^{(*)}$ of Φ_F, i.e., to the root $x^{(*)}$ of F is locally quadratic if the Jacobian matrix $J_F(x^{(*)})$ is invertible. Thus, for $e^{(n)}=x^{(n)}-x^{(*)}$, we have $\|e^{(n+1)}\|=\mathrm{O}(\|e^{(n)}\|^2)$. It is well-known, that one can avoid the computation of the matrix inverse $A_F(x)$ by solving a linear system with the Jacobian as coefficient matrix, and thus, Eq. (1) is to be regarded as a more formal definition.

In the sequel, we consider only the case that $J_F(x^{(*)})$ is invertible. Thus, modifications of the Newton method for multiple roots will not be treated in the present contribution.

In the univariate case, the following theorem was proven recently [11]:

Theorem 1. *Assume that the function $f:\mathbb{R}\to\mathbb{R}$ is sufficiently smooth in a neighborhood of its root x_*, where $f'(x_*)\neq 0$, that the iteration function $\Psi_f:x\mapsto x-f(x)/f'(x-f(x)/(2f'(x)))$ satisfies $|\Psi_f'''(x)|\leqslant M$ for some constant M in that neighborhood. Then the iterative scheme $x_{n+1}=\Psi_f(x_n)$ converges cubically to x_* in a neighborhood of x_*.*

The multivariate analogon to the iterative scheme given in the above theorem is given by the scheme $x^{(n+1)}=\Psi_F(x^{(n)})$ starting from some initial vector $x^{(0)}$. Here, we define $\Psi_F:x\mapsto\Psi_F(x)$ by

$$\begin{aligned} x \quad &\mapsto z \qquad\quad : \quad J_F(x)(x-z)=F(x)/2,\\ (x,z) &\mapsto \Psi_F(x) : J_F(z)(x-\Psi_F(x))=F(x). \end{aligned} \tag{3}$$

Thus, the first of these equations is used to calculate an intermediate vector z by solving a linear system with a coefficient matrix as given by the Jacobian at x, and then, the new vector $\Psi_F(x)$ is calculated by solving a further linear system with a coefficient matrix as given by the Jacobian at the vector z.

Formally, we can write

$$\Psi_F(x)=x-A_F\left(x-\tfrac{1}{2}A_F(x)F(x)\right)F(x). \tag{4}$$

This may be interpreted geometrically in the following way: The tangent hyperplane going through the point $P=(x,F(x))$ that is used in the Newton method, is replaced by a hyperplane through P that is parallel to the tangent hyperplane at $Q=(z,F(z))$, where z corresponds to a "half Newton step". In this way, the curvature of F in the direction of the Newton step is taken into account. Note that $F(z)$ is not computed.

Now we prove the main result of this contribution:

Theorem 2. *Assume that the function $F:\mathbb{R}^d \to \mathbb{R}^d$ has a root $x^{(*)}$ where the Jacobian $J_F(x^{(*)})$ is invertible. Further, assume that F is sufficiently smooth in a neighborhood of this root such that, for all x in that neighborhood, the Jacobian $J_F(x)$ is invertible with inverse $A_F(x)$, the Newton method converges quadratically and, further*

$$\max_{1\leqslant\alpha,\beta,\gamma\leqslant d}\left|\frac{\partial^3 F}{\partial x_\alpha\partial x_\beta\partial x_\gamma}(x)\right|\leqslant M$$

for some constant M holds in that neighborhood. Then the iterative scheme

$$x^{(n+1)}=\Psi_F(x^{(n)})=x^{(n)}-A_F\left(x^{(n)}-\tfrac{1}{2}A_F(x^{(n)})F(x^{(n)})\right)F(x^{(n)}) \tag{5}$$

converges cubically to $x^{(*)}$ in a neighborhood of $x^{(*)}$.

Proof. Define $z^{(n)}=x^{(n)}-A_F(x^{(n)})F(x^{(n)})/2$, $e^{(n)}=x^{(n)}-x^{(*)}$, and $d^{(n)}=z^{(n)}-x^{(*)}$. Then, since the Newton method converges quadratically, we have $x^{(n)}-A_F(x^{(n)})F(x^{(n)})=x^{(*)}+\mathrm{O}(\|e^{(n)}\|^2)$, whence

$$z^{(n)}=x^{(n)}/2+x^{(*)}/2+\mathrm{O}(\|e^{(n)}\|^2)\,.$$

Thus,

$$d^{(n)}=e^{(n)}/2+\mathrm{O}(\|e^{(n)}\|^2).$$

Using Taylor expansion, we have (we always sum over all double greek indices from 1 to d and put $\partial_\alpha=\partial/\partial x_\alpha$)

$$F_\alpha(x^{(n)})=e^{(n)}_\beta\partial_\beta F_\alpha(x^{(*)})+\tfrac{1}{2}\,e^{(n)}_\beta\partial_\beta e^{(n)}_\gamma\partial_\gamma F_\alpha(x^{(*)})+\mathrm{O}(\|e^{(n)}\|^3)$$

and

$$\partial_\beta F_\alpha(x^{(*)})=\partial_\beta F_\alpha(z^{(n)})-d^{(n)}_\gamma\partial_\gamma\partial_\beta F_\alpha(z^{(n)})+\mathrm{O}(\|d^{(n)}\|^2).$$

Substituting the latter equation into the first term of the rhs of the former equation, and using

$$\tfrac{1}{2}\,e^{(n)}_\beta e^{(n)}_\gamma(\partial_\beta\partial_\gamma F_\alpha(x^{(*)})-\partial_\gamma\partial_\beta F_\alpha(z^{(n)}))=\mathrm{O}(\|e^{(n)}\|^3),$$

we obtain

$$F_\alpha(x^{(n)})=e^{(n)}_\beta\partial_\beta F_\alpha(z^{(n)})+\mathrm{O}(\|e^{(n)}\|^3).$$

But since we have

$$F_\alpha(x^{(n)})=(x^{(n)}-x^{(n+1)})_\beta\partial_\beta F_\alpha(z^{(n)})$$

we obtain by subtracting both relations

$$0 = e_\beta^{(n+1)} \partial_\beta F_\alpha(z^{(n)}) + \mathrm{O}(\|e^{(n)}\|^3).$$

Since $J(z^{(n)})$ is invertible for $z^{(n)}$ sufficiently close to $x^{(*)}$, we obtain the desired result

$$e^{(n+1)} = \mathrm{O}(\|e^{(n)}\|^3).$$

This ends the proof. □

Now, numerical examples are given for which both the Newton and the modified Newton method will be compared. The calculations were done using `MAPLE V`™ 5 [1,2] on a personal computer requiring 16 decimal digits for all numerical evaluations.

In the first example, F corresponds to a coupled system of quadratic equations. It concerns the so-called generalized Bloch equation (see, e.g., [17]). This equation is used in quantum chemical ab initio calculations and can also be solved by direct iteration [12–14]. The generalized Bloch equation can be expressed as

$$UHU = HU \tag{6}$$

in terms of $m \times m$ matrices U and H, where H is Hermitean and U is idempotent $U^2 = U$.

We set (block matrix notation)

$$U = \begin{pmatrix} 1_k & 0_{k\times(m-k)} \\ X & 0_{(m-k)\times(m-k)} \end{pmatrix}, \tag{7}$$

where 1_k denotes the $k \times k$ unit matrix, $0_{u\times v}$ denotes the $u \times v$ matrix of zeroes, and X is the $(m-k) \times k$ matrix

$$X = \begin{pmatrix} x_1 & x_{m-k+1} & \cdots & x_{d-m+k+1} \\ x_2 & x_{m-k+2} & \cdots & x_{d-m+k+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m-k} & x_{2(m-k)} & \cdots & x_d \end{pmatrix} \tag{8}$$

with $d = (m-k)k$. Then, $U^2 = U$ is satisfied automatically. Also note the special notation for the entries of the matrix X that shows that X may be regarded as a d dimensional vector x. Inserting this into Eq. (6), one easily obtains a set of d coupled quadratic equations in the unknowns $x_1, \ldots, x_d$.

In our example, we put

$$H_{i,j} = \frac{1}{1 + |i-j|/40}. \tag{9}$$

We consider two cases:

Case 1: $m = 6$, $k = 2$ which results in $d = 8$, iterations are stopped for $\|UHU - HU\| \leqslant 10^{-14}$.

Case 2: $m = 8$, $k = 3$ which results in $d = 15$, iterations are stopped for $\|UHU - HU\| \leqslant 10^{-13}$.

In all cases and for both the Newton and the modified Newton method, we start from $x^{(0)} = 0$.

The results for both cases are displayed in Table 1.

First, we discuss Case 1: Here, the estimates for the matrix X at the end agree to approximately 13 decimal digits.

Table 1
Displayed are the error norms $\|UHU - HU\|$ for the Newton and the modified Newton method after the iteration step for both cases

	Case 1		Case 2	
Step	Newton	Mod. Newton	Newton	Mod. Newton
1	3.65	1.42	9.86	2.34
2	0.64	0.033	1.93	0.24
3	0.090	0.011	0.27	0.0047
4	1.42	0.00019	5630.99	0.00032
5	0.33	$0.12{\cdot}10^{-8}$	1407.75	$0.12{\cdot}10^{-6}$
6	0.066	$0.25{\cdot}10^{-14}$	351.93	$0.29{\cdot}10^{-14}$
7	0.031		87.95	
8	0.0046		21.95	
9	0.00031		5.45	
10	0.000011		1.33	
11	$0.13{\cdot}10^{-7}$		0.30	
12	$0.19{\cdot}10^{-13}$		59.37	
13	$0.20{\cdot}10^{-14}$		14.87	
14			3.74	
15			0.95	
16			0.24	
17			0.091	
18			0.020	
19			0.00072	
20			$0.92{\cdot}10^{-6}$	
21			$0.46{\cdot}10^{-13}$	

The modified Newton method is slightly superior to the Newton method in case 1 in two respects: To achieve convergence with 6 steps of the modified Newton method requires similar but somewhat lower numerical efforts than 12 steps of the Newton methods since 6 evaluations of F are saved. Further, the reduction of the error norm is smoother for the modified Newton method.

In Case 2, both methods yield a solution, but these are different. This is not surprising since the solution to a set of coupled algebraic equations does not need to be unique (nor real). The resulting approximation for $x^{(*)}$ is much closer to the starting vector $x^{(0)} = 0$ in the modified Newton method. This corresponds to the fact, that again, the decrease of the error norm is much smoother for the modified Newton method. The Newton method seems to be more likely to run into problems that result in a ragged performance profile in terms of the error norm. As a result of this behavior, for case 2, the modified Newton method is much superior to the Newton method since the numerical effort for the 6 iteration steps is somewhat less than that of 12 iteration steps of the Newton method.

As a second example, we treat the two-dimensional vector function F with components

$$F_1 : (x, y) \mapsto \int_0^{x+3y} \left(1 + \exp\left(2\sqrt[8]{|t|}\right)\right)^{-1} \mathrm{d}t + xy,$$

$$F_2 : (x, y) \mapsto \exp(-x) - \exp(-4y). \tag{10}$$

Table 2
Displayed are the error norms $\|x^{(n)}\|$ for the Newton and the modified Newton method for the vector function (10), and the total execution times up to iteration n (for a representative run using numerical quadrature, in seconds). The starting vector was $(2, 0.6)$

	Newton		Mod. Newton	
n	Error	Time	Error	Time
1	1.12	8	$3.02{\cdot}10^{-1}$	9
2	$8.58{\cdot}10^{-2}$	16	$1.08{\cdot}10^{-2}$	17
3	$1.58{\cdot}10^{-2}$	24	$7.30{\cdot}10^{-5}$	25
4	$4.37{\cdot}10^{-3}$	33	$7.05{\cdot}10^{-7}$	34
5	$3.09{\cdot}10^{-4}$	41	$3.84{\cdot}10^{-9}$	42
6	$1.87{\cdot}10^{-5}$	50	$1.07{\cdot}10^{-11}$	51
7	$7.35{\cdot}10^{-7}$	59	$1.39{\cdot}10^{-14}$	59
8	$1.78{\cdot}10^{-8}$	67	$4.72{\cdot}10^{-18}$	69
9	$2.55{\cdot}10^{-10}$	76		
10	$2.05{\cdot}10^{-12}$	84		
11	$8.75{\cdot}10^{-15}$	93		
12	$3.45{\cdot}10^{-17}$	101		
13	$1.59{\cdot}10^{-17}$	110		

Table 3
Displayed are the error norms $\|x^{(n)}\|$ for the Newton and the modified Newton method for the vector function (10), and the total execution times up to iteration n (for a representative run using the analytical expression, in seconds). The starting vector was $(2, 0.6)$

	Newton		Mod. Newton	
n	Error	Time	Error	Time
1	1.12	0	$3.02{\cdot}10^{-1}$	0
2	$8.58{\cdot}10^{-2}$	1	$1.08{\cdot}10^{-2}$	1
3	$1.58{\cdot}10^{-2}$	1	$7.30{\cdot}10^{-5}$	1
4	$4.37{\cdot}10^{-3}$	1	$7.05{\cdot}10^{-7}$	2
5	$3.09{\cdot}10^{-4}$	2	$3.84{\cdot}10^{-9}$	2
6	$1.87{\cdot}10^{-5}$	2	$1.11{\cdot}10^{-11}$	3
7	$7.35{\cdot}10^{-7}$	3	$1.20{\cdot}10^{-13}$	3
8	$1.78{\cdot}10^{-8}$	3		
9	$2.55{\cdot}10^{-10}$	4		
10	$2.23{\cdot}10^{-12}$	6		
11	$2.73{\cdot}10^{-14}$	6		

This function has a zero at (0,0). In Tables 2 and 3, we compare the Newton method and the present modification. Although an analytical expression for the integral in F_1 is available,

$$\int_0^z \left(1+\exp\left(2\sqrt[8]{|t|}\right)\right)^{-1} \mathrm{d}t = z + 315\sqrt[8]{z}\,\phi(7,-X) - 4z^{7/8}\ln(1+X)$$

$$
\begin{aligned}
&-\frac{315}{2}\,\phi(8,-X) - 14z^{3/4}\phi(2,-X)\\
&+42z^{5/8}\phi(3,-X) - 105\,\sqrt{z}\phi(4,-X)\\
&+210z^{3/8}\phi(5,-X) - 315\,\sqrt[4]{z}\,\phi(6,-X)\\
&-\frac{127}{7680}\,\pi^{8} \qquad (11)
\end{aligned}
$$

for $X = \exp\left(2\,\sqrt[8]{z}\right)$ and $z > 0$, in terms of the polylog function denoted by

$$\phi(a,r) = \sum_{n=1}^{\infty} \frac{r^n}{n^a}, \qquad (12)$$

we first evaluate the integral by numerical quadrature for illustrative purposes (Table 2).

Then, the time per iteration is strongly dominated by the numerical quadrature of the integral in F_1. Hence, the time per iteration is nearly equal for the Newton and the modified Newton method. The modified method is seen to be superior in this example. To reach comparable accuracies requires about 50–70% more computational time for the Newton method.

If the analytical expression is used instead, we obtain the results in Table 3. Apart from much lower overall execution times, more or less the same picture as in Table 2 is observed. This means that the evaluation of the vector function is still rather costly in comparison to the other operations required for a Newton step.

Of course, the second example is especially chosen to illustrate the potential of the modified Newton method.

However, the modified method is expected to be at least competitive to the Newton method (as in the one-dimensional case) if the costs for the evaluation of the vector function are comparable to the costs for the evaluation of the Jacobian and the solution of the corresponding linear systems. Under such conditions, two steps of the modified method and three steps of the Newton method have approximately the same costs and the same order of convergence: Starting at a distance ε from $x^{(*)}$ the error is $O(\varepsilon^6)$ both after two steps of the modified method or after three steps of the Newton method.

Such cost relations can occur for instance (a) in low-dimensional problems, (b) in cases where the Jacobian is sparse, (c) for Jacobians of a special structure (banded, triangular etc.) making inversion or solution of the corresponding linear systems cheap and (d) in cases involving numerical quadrature similar to the second example.

It is not claimed that the present modification of the Newton method is always superior to the Newton method.

In many cases, the evaluation of the vector function F costs much less than the evaluation of the Jacobian and/or the solution of the corresponding linear systems. This means that the costs for one step of the modified Newton method are approximately equal to that of two steps of the Newton method. Starting at a distance ε to $x^{(*)}$ the error after one step of the modified method is $O(\varepsilon^3)$ and after two steps of the Newton method is $O(\varepsilon^4)$ for small ε. Thus, close to $x^{(*)}$, the Newton method should be superior.

But as noted for the first example, the modified Newton method may provide a smoother decrease of the error norm than the Newton method. A better understanding of such a behavior at larger distances from $x^{(*)}$ would be very interesting. Such an investigation should provide conditions under which a single step of the modified method can be superior to two steps of the Newton method. The search for such conditions is a promising line for future work.

It is to be expected that there are cases where the modified method has problems similar to the Newton method, e.g., due to the fact that the Jacobian is numerically singular. Thus, in order to use the present method in a robust solver, one has to combine it with other concepts such as trust regions or steepest descent methods, similar to hybrid methods involving the Newton method like Powell's method [18] or like HYBRD1 from MINPACK, available from NETLIB. Even more promising is the use of ANT of Deuflhard and coworkers [9,10,15,16] that can be used to make the Newton method globally convergent. There, one uses a carefully chosen damping factor λ with $0 < \lambda \leqslant 1$ to reduce the Newton step in such a way that too large steps are avoided. This damping strategy is chosen to be affin-invariant. This means that for any invertible matrix $M \in GL(d)$ and any starting vector $x^{(0)}$ the application of the method to the vector function $G(x) = MF(x)$ will yield the same sequence of iterates $x^{(n)}$ as application to $F(x)$ itself.

We note that, similarly to the Newton method, the modified Newton method (without damping) is affine invariant. The proof is straightforward and left to the reader. Thus, one may use all the concepts of Deuflhard's ANT also for the modified method with probably small modifications. This, however, is beyond the scope of the present contribution.

Acknowledgements

The author thanks the *Deutsche Forschungsgemeinschaft* and the *Fonds der Chemischen Industrie* for financial support.

References

[1] B.W. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, S.M. Watt, Maple V Language Reference Manual, Springer, Berlin, 1991.

[2] B.W. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, S.M. Watt, Maple V Library Reference Manual, Springer, Berlin, 1991.

[3] D. Dent, M. Paprzycki, A. Kucaba-Pietal, Performance of solvers for systems of nonlinear equations, in: Proceedings of the 15th Annual Conference on Applied Mathematics, University of Central Oklahoma, Edmond, OK, 1999, pp. 67–77.

[4] D. Dent, M. Paprzycki, A. Kucaba-Pietal, Studying the numerical properties of solvers for systems of nonlinear equations, in: D. Bainov (Ed.), Proceedings of the Ninth Colloquium on Difference Equations, Utrecht, VSP, 1999, pp. 113–118.

[5] D. Dent, M. Paprzycki, A. Kucaba-Pietal, Testing convergence of nonlinear system solvers, in: Proceedings of the First Southern Symposium on Computing, The University of Southern Mississippi, December 4–5, 1998, 1999. http://pax.st.usm.edu/cmi/fscc98_html/processed/.

[6] D. Dent, M. Paprzycki, A. Kucaba-Pietal, Recent advances in solvers for nonlinear algebraic equations, Comput. Assist. Mech. Eng. Sci. (CAMES) 7 (2000) 493–505.

[7] D. Dent, M. Paprzycki, A. Kucaba-Pietal, Studying the performance nonlinear systems solvers applied to the random vibration test, in: Svetozar Margenov, Jerzy Wasniewski, Plamen Y. Yalamov (Eds.), Large-Scale Scientific Computing, Third International Conference, LSSC 2001, Sozopol, Bulgaria, June 6–10, 2001, Revised Papers, Lecture Notes in Computer Science, Vol. 2179, Springer, Berlin, 2001, pp. 471–478.

[8] D. Dent, M. Paprzycki, A. Kucaba-Pietal, Comparing solvers for large systems of nonlinear algebraic equations, in: Proceedings of the Southern Conference on Computing, The University of Southern Mississippi, October 26–28, 2000, 2002. http://www.sc.usm.edu/conferences/scc2/papers/Dent-etal.ps.

[9] P. Deuflhard, A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting, Numer. Math. 22 (1974) 289–315.

[10] P. Deuflhard, A relaxation strategy for the modified Newton method, in: R. Bulirsch, W. Oettli, J. Stoer (Eds.), Optimization and Optimal Control, Lecture Notes in Mathematics, Vol. 477, Springer, Berlin, 1975, pp. 59–73.

[11] H.H.H. Homeier, A modified Newton method for root finding with cubic convergence, J. Comput. Appl. Math. 157 (2003) 227–230.

[12] H. Meißner, J. Paldus, Direct iterative solution of the generalized Bloch equation. II. A general formalism for many-electron systems, J. Chem. Phys. 113 (7) (2000) 2594–2611.

[13] H. Meißner, J. Paldus, Direct iterative solution of the generalized Bloch equation. III. Application to H_2-cluster models, J. Chem. Phys. 113 (7) (2000) 2612–2621.

[14] H. Meißner, J. Paldus, Direct iterative solution of the generalized Bloch equation. IV. Application to H_2 LiH, BeH, and CH_2, J. Chem. Phys. 113 (7) (2000) 2622–2637.

[15] U. Nowak, L. Weimann, GIANT—a software package for the numerical solution of very large systems of highly nonlinear equations, Technical Report TR 90-11, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1990. available from the ZIB ftp-server.

[16] U. Nowak, L. Weimann, A family of Newton codes for systems of highly nonlinear equations, Technical Report TR 91-10, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1991. available from the ZIB ftp-server.

[17] J. Paldus, Coupled cluster theory, in: S. Wilson, G.H.F. Diercksen (Eds.), Methods in Computational Molecular Physics, Plenum Press, New York, 1992, pp. 99–194.

[18] M.J.D. Powell, A hybrid method for nonlinear equations, in: P. Rabinowitz (Ed.), Numerical Methods for Nonlinear Algebraic Equations, Gordon and Breach, London, 1970, pp. 87–114.

[19] W.C. Rheinboldt, Methods for Solving System of Nonlinear Equations, SIAM, Philadelphia, 1998.