

An incomplete Hessian Newton minimization method and its application in a chemical database problem

Dexuan Xie · Qin Ni

Received: 8 November 2007 / Revised: 20 December 2007
© Springer Science+Business Media, LLC 2008

Abstract To efficiently solve a large scale unconstrained minimization problem with a dense Hessian matrix, this paper proposes to use an incomplete Hessian matrix to define a new modified Newton method, called the incomplete Hessian Newton method (IHN). A theoretical analysis shows that IHN is convergent globally, and has a linear rate of convergence with a properly selected symmetric, positive definite incomplete Hessian matrix. It also shows that the Wolfe conditions hold in IHN with a line search step length of one. As an important application, an effective IHN and a modified IHN, called the truncated-IHN method (T-IHN), are constructed for solving a large scale chemical database optimal projection mapping problem. T-IHN is shown to work well even with indefinite incomplete Hessian matrices. Numerical results confirm the theoretical results of IHN, and demonstrate the promising potential of T-IHN as an efficient minimization algorithm.

Keywords Unconstraint minimization · Incomplete Hessian matrix · Convergence analysis · Truncated Newton · Chemical database analysis

This project was supported by the National Science Foundation (DMS-0241236, USA), the National Institutes of Health (PHS R01 EB005825-01, USA), the Natural Science Foundation of China (10471062), and Jiangsu Province (BK2006184, China).

D. Xie (✉)
Department of Mathematical Sciences, University of Wisconsin, 3200 N Cramer Street,
EMS Building, Room E403, Milwaukee, WI 53211, USA
e-mail: dxie@uwm.edu

Q. Ni
Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016,
People's Republic of China

1 Introduction

Let f be a twice continuously differentiable multivariable function defined on a bounded neighboring domain, \mathcal{D} , of the n -dimensional Euclidean real vector space R^n . We consider a large scale unconstrained optimization problem:

$$\text{Find } x^* \in \mathcal{D} \subset R^n \quad \text{such that} \quad f(x^*) = \min\{f(x) \mid x \in \mathcal{D}\}, \quad (1)$$

where x^* lies in the interior of \mathcal{D} , $f(x)$ and the gradient vector $g(x)$ (i.e., the first derivative of f at x) are expensive to be evaluated, and the Hessian matrix $H(x)$ (i.e., the second derivative of f at x) is dense, and available to be evaluated analytically. In practice, however, when n is large enough, it is infeasible to evaluate the whole Hessian $H(x)$ due to the computer cost of computing and storage. Typical examples of (1) often arise from biomolecular energy function minimization problems and chemical database optimal projection problems as well as many other scientific and engineering applications. The minimization problem of biomolecular potential energy function is one of the fundamental tasks in biomolecular simulations [10] while solving the optimal projection mapping problem is a key step in a large scale chemical database analysis [10, 13, 14]. Developing efficient numerical minimization algorithms is essential in these application fields.

Currently, typical algorithms for solving (1) include the steepest descent method (SD), the nonlinear conjugate gradient method (CG), the limited-memory BFGS method (L-BFGS) [4–6], and the discrete truncated Newton method (D-TN) [6, 7]. They all do not require any evaluations of the Hessian matrices but gradient vectors. In SD and CG, the gradient vectors are employed to construct descent search directions. To gain faster minimization algorithms, L-BFGS and D-TN use the gradient vectors to construct approximate Hessian matrices to define them as modified Newton type algorithms [6]. For example, in D-TN, an approximate Hessian matrix is generated implicitly by using gradient vectors to construct a finite difference approximation to the Hessian-vector product, which is the only place where $H(x)$ occurs in the truncated Newton method (TN) [1]. However, such a finite difference formula is inherently numerically unstable, which may disturb the numerical behaviors of D-TN significantly.

In this paper, we intend to study the idea of constructing an approximate Hessian, $M(x)$, directly from the Hessian matrix $H(x)$, which is available in Problem (1) and should be used, even partially, in developing fast minimization algorithms. With a properly selected sparse pattern and sparse matrix techniques, we can simply construct $M(x)$ as an incomplete Hessian matrix, and evaluate it in a fast way based on the available computer computing and storage capability. We then substitute it to the Hessian matrix $H(x)$ of a classic modified Newton method to yield an incomplete Hessian Newton method (IHN). With a properly selected $M(x)$, IHN is expected to be numerically stable, easy to be implemented, and have high computer performance and fast convergence rate.

The focus of this paper is to study some basic convergence properties of IHN. For this purpose, we simply assume that $M(x)$ is symmetric, positive definite in domain \mathcal{D} . Following the classic quasi-Newton theory, we prove that IHN converges

globally, and has both an R-linear rate of convergence and a Q-linear rate of convergence when $M(x)$ is properly selected. We also prove that the Wolfe conditions hold for IHN with a line search step length of one when the number of IHN iterations is large enough.

As an application, we construct a particular IHN for solving the chemical database optimal projection mapping problem as described in [10, 13, 14]. In this application, the entries of $H(x)$ that correspond to the pairwise distances within a certain short range become a dominating part of $H(x)$. Thus, they can be selected as the nonzero entries of $M(x)$, yielding a good approximation of $H(x)$. In this paper, we construct $M(x)$ by a distance cut-off strategy, and express both $H(x)$ and $M(x)$ in terms of Kronecker products to display the dense and sparse matrix structures of $H(x)$ and $M(x)$. Such a sparse expression of $M(x)$ is valuable in programming $M(x)$ by sparse matrix techniques. To confirm our theoretical results, we carry out numerical experiments on IHN with a real chemical dataset. Numerical results show that the rate of convergence of IHN can be close to that of the classic modified Newton method when the incomplete Hessian is properly selected. Even with a very sparse incomplete Hessian (a block diagonal matrix with each block being a 2 by 2 matrix), IHN was still found to have a much faster rate of convergence than SD.

However, assuming all the incomplete Hessian matrices to be positive definite is often too strong to be satisfied in practice. To release this assumption and further reduce the computing cost, we use the truncated Newton strategy given in [12] to modify IHN as a descent search direction method, and call it the truncated IHN method (T-IHN) for clarity. T-IHN is shown to converge globally even with indefinite incomplete Hessian matrices. To numerically study the convergence rate and computer performance, we develop a MATLAB program package for T-IHN for solving the chemical database problem using sparse matrix techniques. We then compare the convergence and performance of T-IHN with that of SD, BFGS, and D-TN. Here SD and BFGS were implemented by calling the minimization solver routine *fminunc* from the MATLAB library, and the MATLAB program of D-TN is the same as that of T-IHN except that it uses the Euler forward finite difference formula to approximate the Hessian-vector product.

Numerical results show that the T-IHN using an incomplete Hessian with about 60 percent of zero entries has a faster rate of convergence and a better performance than BFGS. T-IHN took less CPU time by a factor of about 2.09 than BFGS for a dataset of 300 members. T-IHN was also found to have a close rate of convergence as D-TN and a better performance than D-TN. In this test, T-IHN took less CPU time by a factor of about 2.6 than D-TN. Here we did not compare T-IHN with L-BFGS since the MATLAB library does not contain any L-BFGS program routine. Note that L-BFGS usually has a slower rate of convergence than BFGS. Hence, it can be expected that T-IHN has better performances in both convergence and CPU time than L-BFGS.

We also made tests on T-IHN with two very sparse incomplete Hessian matrices: one has about 97.43% of zero entries, and the other has about 99.67% of zero entries. Even so, T-IHN was still found to have much better performances in both convergence rate and CPU time than SD. It took less CPU time by a factor of up to 7.68 than SD. These numerical results demonstrate the promising potential of T-IHN as an efficient solver of minimization problem (1).

The remainder of the paper is organized as follows. We define IHN in Sect. 2, and present its basic convergent properties in Sect. 3. We then describe the IHN and T-IHN methods for solving the database problem in Sect. 4. Finally, the numerical results on IHN and T-IHN are presented in Sect. 5.

2 The IHN method

Let $g(x)$, $H(x)$ and $M(x)$ denote the gradient vector, Hessian matrix, and incomplete Hessian matrix of f at $x \in \mathcal{D}$, respectively. We assume that both $H(x)$ and $M(x)$ are symmetric, positive definite in \mathcal{D} . The IHN iterative sequence $\{x_k\}$ for solving (1) is defined in the form

$$x_{k+1} = x_k + \alpha_k p_k, \quad k = 0, 1, 2, \dots, \quad (2)$$

where x_0 is a given initial iterate in \mathcal{D} , p_k is a search direction satisfying

$$M(x_k)p_k = -g(x_k), \quad (3)$$

and α_k is a step length satisfying the Wolfe conditions

$$f(x_{k+1}) \leq f(x_k) + c_1 \alpha_k g(x_k)^T p_k \quad \text{and} \quad g(x_{k+1})^T p_k \geq c_2 g(x_k)^T p_k \quad (4)$$

for $0 < c_1 < \frac{1}{2} < c_2 < 1$. Clearly, $p_k = -M(x_k)^{-1}g_k$, which is a descent search direction in the sense that

$$g(x_k)^T p_k < 0 \quad \text{for } k = 0, 1, 2, \dots \quad (5)$$

Denote m_{ij} and h_{ij} as the (i, j) th entry of $M(x)$ and $H(x)$, respectively. The sparse pattern P of $M(x)$ is a set of index pairs (i, j) at which $m_{ij} \neq 0$. With a given P , the incomplete Hessian matrix $M(x)$ is defined by

$$m_{ij}(x) = \begin{cases} h_{ij}(x) & \text{for } (i, j) \in P, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, a selection of P depends on the problem to be solved and the capacity of a computer to be used for implementation. In the extreme cases, we can set $P = P_f$ and P_d , where

$$P_d = \{(i, i) \mid i = 1, 2, \dots, n\} \quad \text{and} \quad P_f = \{(i, j) \mid i, j = 1, 2, \dots, n\}. \quad (6)$$

Obviously, the matrices $M(x)$ with P_f and P_d are respectively the original Hessian matrix $H(x)$ and the diagonal matrix with $h_{11}(x), h_{22}(x), \dots$, and $h_{nn}(x)$ as the diagonal entries. Hence, the IHN with P_f returns to the classic Newton method.

3 The convergence analysis of IHN

Let $\|\cdot\|$ be the 2-norm of a vector/matrix. For clarity, we sometimes write $f(x_k)$, $g(x_k)$, $H(x_k)$, $M(x_k)$, $g(x^*)$, $H(x^*)$, and $M(x^*)$ as f_k , g_k , H_k , M_k , g^* , H_* , and M_* , respectively. Following the general quasi-Newton theory (e.g., see pp. 43–45 in [6]), we have the global convergence theorem for IHN as below.

Theorem 1 *Let $\{x_k\}$ be a sequence of IHN iterates defined in (2). If there exists a positive constant η such that*

$$\|M_k\| \|M_k^{-1}\| \leq \eta \quad \text{for all } k, \tag{7}$$

then $\lim_{k \rightarrow \infty} \|g(x_k)\| = 0$ for any $x_0 \in \mathcal{D}$.

To discuss the convergence rates of IHN, we make Assumptions 1 and 2.

Assumption 1 The IHN iterative sequence $\{x_k\}$ converges to $x^* \in \mathcal{D}$ for any $x_0 \in \mathcal{D}$. Here $g(x^*) = 0$ and both $M(x^*)$ and $H(x^*)$ are positive definite.

Assumption 2 There exists a positive integer, k_0 , such that Wolfe condition (4) is satisfied with $\alpha_k = 1$ for all $k \geq k_0$.

The following two definitions will be used in our IHN analysis.

Definition 1 The convergence rate of $\{x_k\}$ is R-linear if there exists a number r between 0 and 1 such that $\limsup_{k \rightarrow \infty} \|x_k - x^*\|^{1/k} = r$.

Definition 2 The convergence rate of $\{x_k\}$ is Q-linear if there exists a constant $0 < c < 1$ and a positive integer k_0 such that $\|x_{k+1} - x^*\| \leq c \|x_k - x^*\|$ whenever $k \geq k_0$.

Using the similar arguments in both [9] and the proof of Theorem 6.1 in [4], we can prove the R-linear convergence for IHN.

Theorem 2 *If Assumption 1 and condition (7) hold, then the IHN iterative sequence $\{x_k\}$ is R-linearly convergent.*

The following corollary gives an easy-to-check sufficient condition for the IHN using the sparse pattern P_d defined in (6).

Corollary 1 *Let Assumption 1 hold and the sparse pattern of incomplete Hessian matrix M_k be given with P_d in (6). If the diagonal elements of H_k are positive and bounded, then IHN is convergent both globally and R-linearly.*

Proof We only need to prove that (7) is satisfied. Let $h_{ii}^{(k)}$ for $i = 1, \dots, n$ be the diagonal elements of H_k . By the assumption, there exist two positive constants b_1

and b_2 such that $b_1 \leq h_{ii}^{(k)} \leq b_2$ for all i and k . Thus,

$$\|M_k\| \|M_k^{-1}\| = \frac{\max_{1 \leq i \leq n} |h_{ii}^{(k)}|}{\min_{1 \leq i \leq n} |h_{ii}^{(k)}|} \leq \frac{b_2}{b_1}.$$

Hence, the proof follows from Theorems 1 and 2. \square

To discuss the Q-linear rate of convergence, we need Lemmas 1 and 2.

Lemma 1 *If Assumption 1 holds, then for any $\eta > 0$, there exists $\epsilon > 0$ such that $H(x)$ and $M(x)$ are positive definite, $\|M(x)^{-1} - M(x^*)^{-1}\| < \eta$, and $\|g(x) - g(x^*) - H(x^*)(x - x^*)\| < \eta \|x - x^*\|$ whenever $\|x - x^*\| \leq \epsilon$.*

The above lemma can be proved by similar arguments to the ones from Sects. 2.3.3 and 3.1.6 in [8].

Lemma 2 *If H and M are symmetric, positive definite, then*

$$\lambda_{\min}(M^{-1}H) - 1 \leq \frac{x^T(H - M)x}{x^T Mx} \leq \lambda_{\max}(M^{-1}H) - 1, \quad \forall x \neq 0, \quad (8)$$

where $\lambda_{\min}(M^{-1}H)$ and $\lambda_{\max}(M^{-1}H)$ denote the smallest and largest eigenvalues of $M^{-1}H$, respectively.

Proof Let $M^{1/2}$ be the square root matrix of M . Set $\tilde{H} = M^{-1/2}HM^{-1/2}$ and $y = M^{1/2}x$ for any nonzero $x \in \mathcal{D}$. It is easy to show that $x^T Hx = y^T \tilde{H}y$, $y^T y = x^T Mx$, and \tilde{H} is similar to $M^{-1}H$. Hence,

$$\lambda_{\min}(M^{-1}H)y^T y \leq y^T \tilde{H}y \leq \lambda_{\max}(M^{-1}H)y^T y,$$

from which it is then easy to obtain (8). \square

Since M_* is assumed to be symmetric, positive definite, its square root matrix $M_*^{1/2}$ exists and satisfies that $M_* = M_*^{1/2}M_*^{1/2}$. Thus, a new norm, $\|\cdot\|_*$, can be well defined by

$$\|x\|_* = \|M_*^{1/2}x\|, \quad \forall x \neq 0. \quad (9)$$

Under this new norm, we obtain the Q-linear convergence rate for IHN in Theorem 3. Here the spectral radius of a matrix, say A , is denoted by $\rho(A)$, which is defined as the largest of the modules of the eigenvalues of A .

Theorem 3 *Let Assumptions 1 and 2 hold. If*

$$\rho(M_*^{-1}H_*) < 2, \quad (10)$$

then the IHN iterative sequence $\{x_k\}$ has a Q-linear rate of convergence under the norm $\|\cdot\|_$ defined in (9).*

Proof Set $\bar{H}_* = M_*^{-1/2} H_* M_*^{-1/2}$, and denote by λ_j as the j th eigenvalue of \bar{H}_* for $j = 1, 2, \dots, n$. Clearly, \bar{H}_* is symmetric, positive definite because of Assumption 1 so that all its eigenvalues are positive. Further, \bar{H}_* is similar to $M_*^{-1} H_*$; thus, both $M_*^{-1} H_*$ and \bar{H}_* have the same eigenvalues. Hence, $I - M_*^{-1} H_*$ has eigenvalues $1 - \lambda_j$ for $j = 1, 2, \dots, n$, and from (10) it follows that $0 < \lambda_j < 2$ or $|1 - \lambda_j| < 1$ for $j = 1, \dots, n$. Therefore, $\rho(I - M_*^{-1} H_*) = \max_{1 \leq i \leq n} |1 - \lambda_i| < 1$, and we can select a positive real number, t , such that $\rho(I - M_*^{-1} H_*) \leq t$.

We next set $E^* = M_* - H_*$ and $\bar{E}^* = M_*^{-1/2} E^* M_*^{-1/2}$. It is easy to see that \bar{E}^* is symmetric, and similar to $I - M_*^{-1} H_*$. Hence, $\|\bar{E}^*\| = \rho(\bar{E}^*) = \rho(I - M_*^{-1} H_*)$.

With (2) and Assumption 1, we can get that

$$\begin{aligned} & M_*^{1/2}(x_{k+1} - x^*) \\ &= M_*^{1/2}(x_k - M_k^{-1}g_k - x^*) = M_*^{1/2}(-M_k^{-1}(g_k - g^*) + (x_k - x^*)) \\ &= -M_*^{1/2}M_k^{-1}(g_k - g^* - H_*(x_k - x^*)) \\ &\quad - M_*^{1/2}(M_k^{-1} - M_*^{-1})H_*(x_k - x^*) + M_*^{-1/2}E^*(x_k - x^*). \end{aligned} \tag{11}$$

The last term of the above expression can then be estimated as below:

$$\begin{aligned} \|\bar{E}^* M_*^{1/2}(x^* - x_k)\| &= \|\bar{E}^* M_*^{1/2}(x^* - x_k)\| \\ &\leq \|\bar{E}^*\| \|x^* - x_k\|_* = \rho(\bar{E}^*) \|x^* - x_k\|_* \\ &= \rho(I - M_*^{-1} H_*) \|x^* - x_k\|_* \leq t \|x^* - x_k\|_*. \end{aligned} \tag{12}$$

To estimate the other two terms of (11), we set $\eta = \frac{1}{2}(1 + t)$, and

$$\mu = \max\{\|H_*\|, \|M_*\|, \|M_*^{-1}\|, \|M_*^{1/2}\|, \|M_*^{-1/2}\|\}. \tag{13}$$

Obviously, $t < \eta < 1$, and there exists a sufficiently small positive number, γ , satisfying

$$t + \mu^2(2\mu + \gamma)\gamma \leq \eta. \tag{14}$$

For the above $\gamma > 0$, using the continuity of $H(x)$ and $M(x)$ at x^* and similar arguments in the proof of Lemma 1, we can show that there exists positive number ϵ such that

$$\|M(y)^{-1} - M(x^*)^{-1}\| \leq \gamma, \tag{15}$$

and

$$\|M_*^{1/2}(g(y) - g(x^*) - H(x^*)(y - x^*))\| \leq \gamma \|M_*^{1/2}(y - x^*)\| \tag{16}$$

whenever $\|y - x^*\| \leq \epsilon$.

Furthermore, from the definition of the convergence of sequence $\{x_k\}$ it can follow that there exists integer $k_1 > 0$ such that $\|x_k - x^*\| \leq \epsilon$ for $k \geq k_1$. Hence, by (13) and (15), we can obtain that

$$\|M_*^{1/2}(M_k^{-1} - M_*^{-1})H_*M_*^{-1/2}\| \leq \gamma\mu^3,$$

and

$$\|M_k^{-1}\| \leq \|M_*^{-1}\| + \|M_k^{-1} - M_*^{-1}\| \leq \mu + \gamma \quad \text{for } k \geq k_1.$$

Finally, applying the above two expressions, (12), (14) and (16) to (11) gives

$$\begin{aligned} \|x_{k+1} - x^*\|_* &= \|M_*^{1/2}(x_{k+1} - x^*)\| \\ &\leq \|M_*^{1/2}M_k^{-1}(g_k - g^* - H_*(x_k - x^*))\| \\ &\quad + \|M_*^{1/2}(M_k^{-1} - M_*^{-1})H_*(x_k - x^*)\| + \|M_*^{-1/2}E^*(x_k - x^*)\| \\ &\leq (\gamma\mu^2(2\mu + \gamma) + t)\|M_*^{1/2}(x_k - x^*)\| \\ &\leq \eta\|M_*^{1/2}(x_k - x^*)\| = \eta\|x_k - x^*\|_* \end{aligned} \quad (17)$$

whenever $k \geq k_1$. This completes the proof of Theorem 3. \square

From the above proof we see that the condition $\rho(I - M_*^{-1}H_*) < 1$ follows from (10), and is then used in the remaining part of the proof. Thus, we have

Corollary 2 *Let Assumptions 1 and 2 hold. If*

$$\rho(I - M_*^{-1}H_*) < 1, \quad (18)$$

then IHN has a Q -linear rate of convergence under the norm $\|\cdot\|_$.*

According to the theory of linear stationary iterative methods [15], (18) gives a sufficient and necessary condition to guarantee the convergence of an iterative method constructed by the matrix splitting $H_* = M_* - N_*$ with $N_* = M_* - H_*$ for solving the linear system with coefficient matrix H_* . Thus, (18) is a natural condition to construct an efficient IHN.

The condition (10) in Theorem 3 depends on the solution x^* , which is difficult to be verified. To improve it, we propose a sufficient condition that only depends on the current iterate in the following corollary.

Corollary 3 *Let Assumptions 1 and 2 hold. If H_k is positive definite and there exist positive constant $\eta < 2$ and positive integer k_1 such that*

$$\|M_k^{-1}H_k\| \leq \eta, \quad \forall k > k_1, \quad (19)$$

then IHN has a Q -linear rate of convergence under the norm $\|\cdot\|_$.*

Proof From the property of matrix norm and the assumptions it follows that

$$\rho(M_k^{-1}H_k) \leq \|M_k^{-1}H_k\| \leq \eta \quad \text{for all } k > k_1.$$

Letting $k \rightarrow \infty$ in the above expression immediately gives condition (10). Thus, the proof is followed from Theorem 3. \square

A sufficient condition is given in the following theorem to guarantee that IHN satisfies Assumption 2.

Theorem 4 *Let Assumption 1 hold. If the eigenvalues $\lambda_i^{(k)}$ for $i = 1, \dots, n$ of $M_k^{-1}H_k$ have lower and upper bounds λ_l and λ_u and there exists positive integer k_1 such that*

$$1 - c_2 < \lambda_l \leq \lambda_i^{(k)} \leq \lambda_u < 2 - 2c_1 \quad \text{for } i = 1, \dots, n \text{ and } k \geq k_1, \quad (20)$$

where c_1 and c_2 are the two constants in the Wolfe conditions of (4), then Assumption 2 holds for IHN.

Proof By Taylor expansion, we can get that

$$\begin{aligned} f(x_k + p_k) - f(x_k) - c_1 g_k^T p_k \\ = \left(\frac{1}{2} - c_1\right) g_k^T p_k + \frac{1}{2} p_k^T (g_k + H_k p_k) + \frac{1}{2} p_k^T (H(u_k) - H_k) p_k, \end{aligned} \quad (21)$$

where $u_k = x_k + t_k p_k$ with $0 \leq t_k \leq 1$. Set

$$r = \frac{1}{2} \lambda_{\min}(M(x^*)), \quad \text{and} \quad \omega = \frac{r}{2} \min\{2 - 2c_1 - \lambda_u, \lambda_l + c_2 - 1\}. \quad (22)$$

From Assumption 1 and (20) it can be shown that both r and ω are positive. Thus, an upper bound for $\lambda_u - 1$ and a lower bound for $\lambda_l - 1$ can be estimated in terms of r and ω as below:

$$\begin{aligned} \lambda_u - 1 &\leq \frac{\lambda_u + 2 - 2c_1}{2} - 1 = 1 - 2c_1 - \frac{1}{2}(2 - 2c_1 - \lambda_u) \\ &\leq 1 - 2c_1 - \frac{\omega}{r}, \end{aligned} \quad (23)$$

and

$$\begin{aligned} \lambda_l - 1 &\geq \frac{\lambda_l + 1 - c_2}{2} - 1 = -c_2 + \frac{1}{2}(\lambda_l + c_2 - 1) \\ &\geq -c_2 + \frac{\omega}{r}. \end{aligned} \quad (24)$$

Further, by the continuity of H , we can select $\epsilon > 0$ such that

$$\|H(z) - H(x)\| \leq \omega \quad (25)$$

whenever $\|x - x^*\| \leq \epsilon$ and $\|z - x^*\| \leq \epsilon$. Based on Assumption 1, we can also find positive integer $k_2 \geq k_1$ such that

$$\lambda_{\min}(M_k) \geq r, \quad \|x_k - x^*\| \leq \epsilon, \quad \|x_k + p_k - x^*\| \leq \epsilon, \quad (26)$$

and $H(x_k)$ is positive definite for all $k \geq k_2$. As the result of the last two inequality of (26),

$$\|u_k - x^*\| = \|t_k(x_k + p_k - x^*) + (1 - t_k)(x_k - x^*)\| \leq \epsilon.$$

Hence, with (25) and the second inequality of (26), we can get that

$$|p_k^T (H(u_k) - H_k) p_k| \leq \omega \|p_k\|^2, \quad \forall k \geq k_2. \quad (27)$$

Also, by (3), (8) and (23), the term $p_k^T (g_k + H_k p_k)$ of (21) is estimated as

$$\begin{aligned} p_k^T (g_k + H_k p_k) &= p_k^T (H_k - M_k) p_k \leq (\lambda_u - 1) p_k^T M_k p_k \\ &\leq \left(1 - 2c_1 - \frac{\omega}{r}\right) p_k^T M_k p_k. \end{aligned} \quad (28)$$

Therefore, applying (3), the first one of (26), (27), and (28) into (21) gives

$$\begin{aligned} &f(x_k + p_k) - f(x_k) - c_1 g_k^T p_k \\ &\leq -\left(\frac{1}{2} - c_1\right) p_k^T M_k p_k + \left(\frac{1}{2} - c_1 - \frac{\omega}{2r}\right) p_k^T M_k p_k + \frac{1}{2} \omega \|p_k\|^2 \\ &= -\frac{\omega}{2r} p_k^T M_k p_k + \frac{1}{2} \omega \|p_k\|^2 \\ &\leq -\frac{1}{2} \omega \|p_k\|^2 + \frac{1}{2} \omega \|p_k\|^2 = 0, \quad \forall k \geq k_2. \end{aligned} \quad (29)$$

This completes the proof of the first inequality of (4) for Assumption 2.

We next prove the second inequality of (4) for Assumption 2.

By the mean value theorem and (3),

$$\begin{aligned} g(x_k + p_k)^T p_k &= g_k^T p_k + p_k^T H_k p_k + p_k^T (H(v_k) - H_k) p_k \\ &= p_k^T (H_k - M_k) p_k + p_k^T (H(v_k) - H_k) p_k, \end{aligned} \quad (30)$$

where $v_k = x_k + t'_k p_k$ with $0 \leq t'_k \leq 1$. Similar to (27), we obtain

$$|p_k^T (H(v_k) - H_k) p_k| \leq \omega \|p_k\|^2, \quad \forall k \geq k_2. \quad (31)$$

A combination of (8) with (20) and (24) gives

$$p_k^T (H_k - M_k) p_k \geq (\lambda_l - 1) p_k^T M_k p_k \geq \left(-c_2 + \frac{\omega}{r}\right) p_k^T M_k p_k. \quad (32)$$

Thus, by (3), (26), (30), (31), and (32),

$$\begin{aligned} &g(x_k + p_k)^T p_k - c_2 g_k^T p_k \\ &= c_2 p_k^T M_k p_k + p_k^T (H_k - M_k) p_k + p_k^T (H(v_k) - H_k) p_k \\ &\geq c_2 p_k^T M_k p_k + \left(-c_2 + \frac{\omega}{r}\right) p_k^T M_k p_k - \omega \|p_k\|^2 \\ &= \frac{\omega}{r} p_k^T M_k p_k - \omega \|p_k\|^2 \geq 0 \quad \text{for } k \geq k_2. \end{aligned}$$

This completes the proof of Theorem 4. \square

4 The IHN and T-IHN methods for chemical database problem

As an application, we construct IHN and T-IHN for solving the chemical database optimal projection mapping problem described in [10, 13, 14]. The key step is the construction of incomplete Hessian matrix M_k , which is presented in details in this section.

With the notation of [13], we recall the database problem as below.

Let a chemical database of n members have been characterized as an $n \times m$ matrix X :

$$X = (X_1, X_2, \dots, X_n)^T = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}, \tag{33}$$

where $X_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$ stands for the i th member of the database, x_{ij} denotes the value of the j th chemical descriptor for i th member, and the distance $\delta_{ij} = \|X_i - X_j\|$ measures the similarity of the i th member with j th member. A key step in the efficient visualization protocol proposed in [13, 14] is to solve the following unconstrained minimization problem:

$$E(Y_1^*, Y_2^*, \dots, Y_n^*) = \min_{\forall Y_i \in R^l} E(Y_1, Y_2, \dots, Y_n), \tag{34}$$

where $Y_i = (y_{i1}, y_{i2}, \dots, y_{il})^T$, l is a positive integer less than m , and the objective function E is defined by

$$E(Y_1, Y_2, \dots, Y_n) = \frac{1}{4} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \omega_{ij} |d(Y_i, Y_j)^2 - \delta_{ij}^2|^2. \tag{35}$$

Here $d(Y_i, Y_j)^2 = \sum_{k=1}^l (y_{ik} - y_{jk})^2$, and ω_{ij} is the weight constant, which is set as $\omega_{ij} = 1/\delta_{ij}^4$ if $\delta_{ij} \geq \eta$ and $\omega_{ij} = 1$ if $\delta_{ij} < \eta$. The parameter η is a small positive number such as 10^{-12} . In [13, 14], the above minimization problem was solved by D-TN with an initial guess generated from SVD/PCA (singular value decomposition or principal component analysis) [2, 3].

Set $Y = (Y_1, Y_2, \dots, Y_n)^T$. To construct the incomplete Hessian matrix $M(Y)$, we first find the Hessian $H(Y)$ of $E(Y)$ as below:

$$H(Y) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n H_{ij}(Y), \tag{36}$$

where $H_{ij}(Y)$ denotes the second derivative of the term $\frac{\omega_{ij}}{4} [d(Y_i, Y_j)^2 - \delta_{ij}^2]^2$ with respect to Y . We express $H_{ij}(Y)$ as an $n \times n$ block matrix with each block entry, $h_{\mu\nu}$, being an $l \times l$ matrix. It is easy to find that

$$h_{\mu\nu} = \begin{cases} \Pi_{ij} & \text{if } \mu = \nu = i \text{ or } \mu = \nu = j, \\ -\Pi_{ij} & \text{if } \mu = i, \nu = j \text{ or } \mu = j, \nu = i, \\ 0 & \text{otherwise,} \end{cases}$$

where Π_{ij} denotes an $l \times l$ matrix defined by

$$\Pi_{ij} = \omega_{ij}(r_{ij}I_l + 2R_{ij}R_{ij}^T).$$

Here $r_{ij} = d(Y_i, Y_j)^2 - \delta_{ij}^2$, $R_{ij} = Y_i - Y_j$, and I_l is the $l \times l$ identity matrix. Clearly, H_{ij} is sparse with only four nonzero block entries, h_{ii} , h_{jj} , h_{ij} , and h_{ji} , where $h_{ii} = h_{jj}$ and $h_{ij} = h_{ji}$. In terms of Kronecker product \otimes ,¹ $H_{ij}(Y)$ can be expressed as

$$H_{ij}(Y) = (e_i e_i^T - e_i e_j^T - e_j e_i^T + e_j e_j^T) \otimes \Pi_{ij},$$

where e_i denotes the i th standard unit vector of R^n . Thus, if we define $\Pi_{ii} = 0$, then the Hessian $H(Y)$ can be written as

$$\begin{aligned} H(Y) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n (e_i e_i^T - e_i e_j^T - e_j e_i^T + e_j e_j^T) \otimes \Pi_{ij} \\ &= \sum_{i=1}^n \left(e_i e_i^T \otimes \sum_{j=1}^n \Pi_{ij} \right) - \sum_{i < j} (e_i e_j^T + e_j e_i^T) \otimes \Pi_{ij}. \end{aligned} \tag{37}$$

Clearly, the first and second terms of (37) give the main block diagonal part and the off-diagonal part of $H(Y)$, respectively. From the above expression it is easy to see that $H(Y)$ is a full dense matrix of $N \times N$ with $N = nl$.

Using the distance cutoff strategy, we define the incomplete Hessian matrix $M(Y)$ as follows: With a given cutoff radius, $\tau > 0$, we construct the sparse pattern P by

$$P = \{(i, j) \mid \|X_i - X_j\| \leq \tau \text{ for } i, j = 1, 2, \dots, n\}.$$

We then define $M(Y)$ by

$$M(Y) = \sum_{i=1}^n \left(e_i e_i^T \otimes \sum_{j=1}^n \Pi_{ij} \right) - \sum_{(i,j) \in P} (e_i e_j^T) \otimes \Pi_{ij}. \tag{38}$$

Clearly, the above matrix of $M(Y)$ is symmetric due to the symmetry of the submatrix Π_{ij} and the definition of P . Hence, in computer implementation, we can only evaluate and store the upper triangular part of $M(Y)$ to reduce the costs of computing and storage.

To properly control the sparsity of $M(Y)$, we propose to select a value of the cutoff radius τ by the formula

$$\tau = \xi \left[\frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n \delta_{ij}^2 \right]^{1/2}, \tag{39}$$

¹The Kronecker product \otimes of an $m \times n$ matrix, say $A = (a_{ij})_{m \times n}$, with a $\mu \times \nu$ matrix B is defined as a $\mu m \times \nu n$ matrix in the form $A \otimes B = (a_{ij} B)_{\mu m \times \nu n}$.

where ξ is an adjusting factor for the sparsity of $M(Y)$. If $\xi = 1$, τ gives a mean value of the distances between each pair of the database members. In other words, about a half of the entries of $M(Y)$ may be zero. As the value of ξ is reduced to zero, $M(Y)$ becomes the block diagonal matrix $M_d(Y)$:

$$M_d(Y) = \sum_{i=1}^n \left(e_i e_i^T \otimes \sum_{j=1}^n \Pi_{ij} \right). \tag{40}$$

From the block Jacobi convergence theory (see p. 111 in [15], for example) it follows that $\rho(I - M_d(Y^*)^{-1} H(Y^*)) < 1$ if $H(Y^*)$ and $2M_d(Y^*) - H(Y^*)$ are positive definite and all the main diagonal block entries of $H(Y^*)$ are symmetric, positive definite. Hence, by Corollary 2, it is claimed that the IHN method with the incomplete Hessian of (40) is Q-linearly convergent.

With the incomplete Hessian $M(Y)$ of (38), we obtain IHN for solving the database problem (34).

We next describe T-IHN for solving the database problem (34).

The T-IHN iterative sequence $\{x_k\}$ is defined in the same form as the one in (2) except that the descent search direction p_k is selected as either an iterate of the preconditioned conjugate gradient method (PCG) [2] for solving (3) or $-g_k$ (in the worst case) according to the truncated Newton strategy given in [12]. For clarity, the scheme for generating p_k is presented in Algorithm 1. The initial iterate x_0 of T-IHN is generated by using the SVD/PCA scheme given in [13].

Algorithm 1 (Defining the descent search direction p_k for T-IHN)

Let B_k be a preconditioner for M_k , and w_j represent the j th PCG iterate for solving (3). Set $\eta_k = \min\{c_k/k, \|g_k\|\}$, and give $\epsilon > 0$ and $\text{IT}_{\text{PCG}} > 0$ (e.g., $c_k = 0.5$, $\epsilon = 10^{-6}$ and $\text{IT}_{\text{PCG}} = 80$). The k th descent search direction p_k of the T-IHN method is selected by the following steps:

1. [INITIALIZATION]
 - Set $j = 1$, $w_1 = 0$, $r_1 = -g_k$, and $d_1 = z_1$,
 where z_1 solves the linear system $B_k z_1 = -r_1$.
2. [SINGULARITY TEST]
 - If either $|r_j^T z_j| \leq \delta$ or $|d_j^T M_k d_j| \leq \delta$ (e.g., $\delta = 10^{-10}$),
 exit the algorithm with $p_k = w_j$ (for $j = 1$, set $p_k = -g_k$).
3. Compute $\alpha_j = r_j^T z_j / d_j^T M_k d_j$ and $w_{j+1} = w_j + \alpha_j d_j$.
4. [DESCENT DIRECTION TEST]
 - If $g_k^T w_{j+1} \geq g_k^T w_j + \delta$,
 exit the algorithm with $p_k = w_j$ (for $j = 1$, set $p_k = -g_k$).
5. Compute $r_{j+1} = r_j - \alpha_j M_k d_j$.
6. [TRUNCATION TEST]
 - If $\|r_{j+1}\| \leq \eta_k \|g_k\|$ or $j + 1 > \text{IT}_{\text{PCG}}$,
 exit the algorithm with $p_k = w_{j+1}$.
7. Compute $\beta_j = r_{j+1}^T z_{j+1} / r_j^T z_j$, and $d_{j+1} = z_{j+1} + \beta_j d_j$,
 where z_{j+1} solves the linear system $B_k z_{j+1} = r_{j+1}$.
8. Increase j to $j + 1$ and go to Step 2.

As shown in [12], a descent search direction is produced from Algorithm 1 even with indefinite M_k or B_k . Thus, T-IHN is a descent search direction method, whose global convergence follows immediately from the general theory of descent methods [6].

The performance of T-IHN can be improved significantly if a good preconditioner B_k can be selected. However, since M_k is often indefinite for this database problem, its preconditioning causes difficulties in both theory and practice. We avoid such difficulties in this paper by simply setting B_k to be an identity matrix.

5 Numerical results

We developed two MATLAB program packages for IHN and T-IHN for solving the database problem (34), respectively. The IHN program package is only used for numerically studying the convergence behaviors of IHN. Thus, we simply store the incomplete Hessian M_k into a full matrix array and solve the related linear systems by a direct solver. In the T-IHN package, we evaluated and stored only the nonzero block entries of the upper triangular part of M_k , and evaluated the product of M_k with a nonzero vector using only the nonzero block entries of M_k . In both IHN and T-IHN packages, the step length α_k was calculated by calling the line search program from the MATLAB library, where we used $c_1 = 0.01$, $c_2 = 0.9$, and an initial guess of one (based on the result of Theorem 4). The test data sets were selected from a large chemical database provided by the Medical College of Wisconsin, in which each member consists of one rat's renal reactions to different physiological and medical experiments.

As comparisons, we also solved the database problem (34) by the classic Newton method, SD, BFGS, and D-TN. Here BFGS and SD were implemented by calling the minimization program routine, *fminunc*, from the MATLAB library with the option of *HessUpdate* as 'bfgs' and 'steepdesc', respectively. The other options for BFGS and SD included the scaled-identity matrix as the initial Hessian approximation, and the default mixed cubic and quadratic polynomial line search method, which is the same as the one used in the IHN and T-IHN packages. The D-TN program package was the same as the T-IHN package except that the Hessian-vector product $H_k d$ was approximated by the Euler forward finite difference approximation:

$$H_k d \approx \frac{g(x_k + hd) - g(x_k)}{h}, \quad (41)$$

where x_k is the k th D-TN iterate, d is a vector, and h is set as

$$h = \max \left\{ \frac{\varsigma}{\max\{10\varsigma, \|d\|\}}, 0.1\varsigma \right\}$$

with $\varsigma = 2\sqrt{\epsilon_m}(1 + \|x_k\|\sqrt{N})$, $\epsilon = 10^{-10}$, and $N = ln$. The above h is the same as the default setting for D-TN within TNPACK [11]. With (41), each evaluation of $H_k d$ requires one new gradient evaluation. All the tests on SD, BFGS, D-TN, IHN and T-IHN used the same MATLAB program routine we wrote for computing function

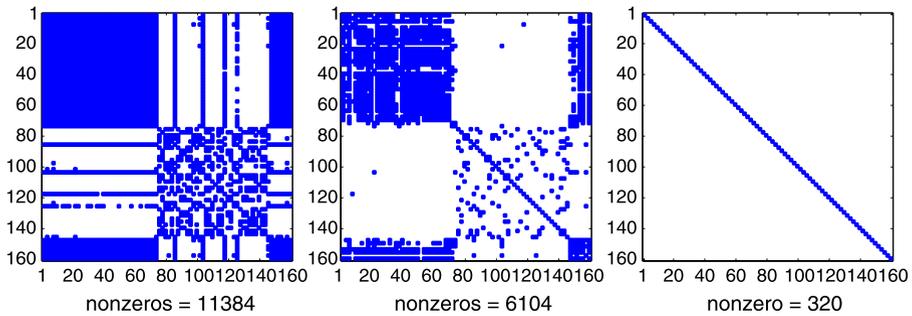


Fig. 1 The sparse patterns of the three incomplete Hessian matrices of 160×160 with the sparsity ratio $\rho = 44.46\%$, 23.84% and 1.25% (from left to right)

Table 1 Comparisons of the convergence of IHN with that of Newton and SD for solving the database problem (34) with $n = 80$ and $l = 2$

Minimizer	Iterations	Final E	Final $\ g\ $
Classic Newton	53	20.663	6.74×10^{-7}
IHN $\rho = 44.46\%$	68	21.334	9.43×10^{-7}
IHN $\rho = 23.84\%$	105	20.776	9.62×10^{-7}
IHN $\rho = 1.25\%$	279	20.538	9.76×10^{-7}
SD	7184	20.633	9.75×10^{-7}

values E and gradient vectors g_k . They also used the same iteration stopping rule (i.e., $\|g(x_k)\| < 10^{-6}$), the same initial iterate generated from SVD (except IHN), and the same datasets with $m = 9$ and $l = 2$. The numerical experiments were made via MATLAB version R2006a on a laptop computer (Latitude D610 with Intel Pentium(R) M 1.86 GHz processor, and 1 GB RAM) at the University of Wisconsin-Milwaukee.

In the numerical tests on IHN, we used a dataset with $n = 80$. As required by the IHN analysis, an initial guess x_0 was selected such that all the Hessian and incomplete Hessian matrices H_k and M_k were positive definite (we checked them by evaluating their eigenvalues). Three incomplete Hessian matrices were constructed by using $\xi = 0.5, 0.2235$ and 0 , which gave the cutoff radius $\tau = 185.26, 82.81$ and 0 , respectively, according to (39). The resulted three incomplete Hessian matrices were found to have that $\rho = 44.46\%$, 23.84% and 1.25% , respectively, where ρ is the percentage of nonzero entries of an $N \times N$ sparse matrix defined by

$$\rho = \frac{100(\text{Total Number of Nonzero Entries})}{N^2}\%$$

Their sparse patterns were plotted in Fig. 1. From this figure we see that the incomplete Hessian with $\rho = 1.25\%$ is a block diagonal matrix with each block being a 2 by 2 matrix.

Figures 2 and 3 and Table 1 compare the convergence behaviors of the IHN using the above three sparse incomplete Hessian matrices with that of the Newton and SD methods. Here the minimum point x^* used in computing errors $\|x_k - x^*\|$ for each method was found previously by this method. From the figures we see that the convergence speed of IHN becomes decreasing as the sparsity percentage ρ is reduced.

Fig. 2 Comparisons of the absolute errors of IHN with that of the classic Newton method for solving the database problem (34) with $n = 80$ and $l = 2$

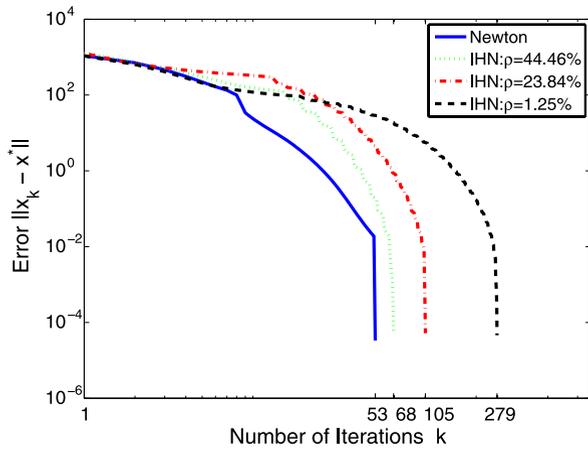
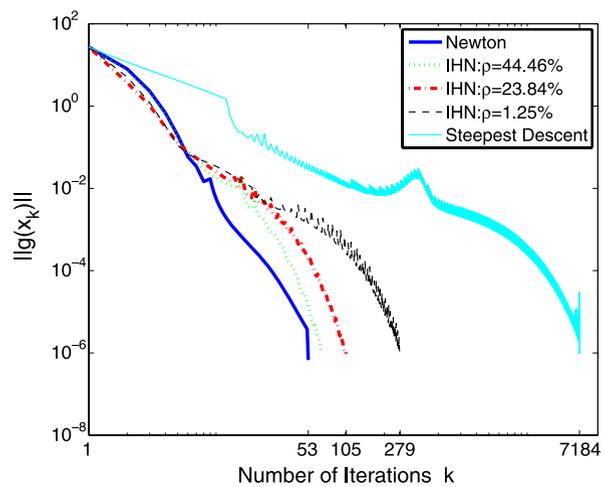


Fig. 3 Comparisons of the gradient norms of IHN with that of the Newton and SD methods for solving the database problem (34) with $n = 80$ and $l = 2$



With $\rho = 44.46\%$, the convergence rate of IHN was close to that of the classic Newton method. In the case of $\rho = 1.25\%$, where incomplete Hessian $M(x_k)$ is a block diagonal matrix with each block being a 2 by 2 matrix, IHN was still found to have a much faster convergence speed than SD.

In the numerical experiments on T-IHN, we selected a dataset of 300 members ($n = 300$) from the large database. We also constructed three incomplete Hessian matrices using $\xi = 0.5, 0.1$ and 0, which resulted in the cutoff radius $\tau = 179.89, 35.978$ and 0, and the sparsity percentage $\rho = 40.22\%, 2.58\%$ and 0.33% , respectively. We plotted the sparse patterns of the incomplete matrices with $\rho = 40.22\%$ and 2.58% in Figs. 4 and 5. It is interesting to note that the nonzero entries of M_k are distributed across the whole matrix. The one with $\rho = 0.33\%$ is a block diagonal matrix with each block being a 2 by 2 matrix.

Table 2 gives the performance data on the T-IHN, D-TN, BFGS, and SD methods for solving the database problem (34) with $n = 300$ and $l = 2$. Here, *funcCount* de-

Fig. 4 The sparse pattern of the 600×600 incomplete Hessian matrix with the sparsity ratio $\rho = 40.2\%$

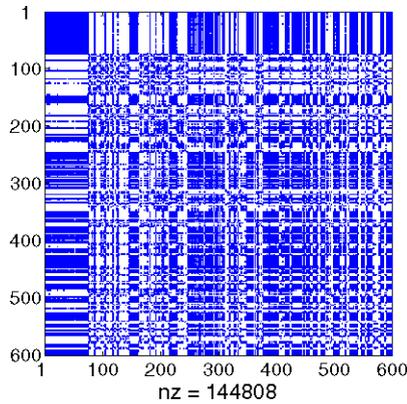


Fig. 5 The sparse pattern of the 600×600 incomplete Hessian matrix with $\rho = 2.58\%$

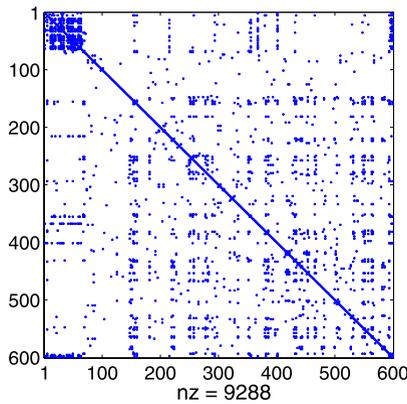


Table 2 Comparisons of the convergence and performance of T-IHN with that of D-TN, BFGS, and SD for solving the database problem (34) with $n = 300$ and $l = 2$. Here the CPU time is measured in seconds

Minimizer	Iterations	Final E	Final $\ g\ $	funcCount	CPU
BFGS	196	330.67	1.19×10^{-6}	199	25.04
D-TN	19 (846)	330.67	4.45×10^{-6}	909	27.26
SD	3531	330.67	2.76×10^{-5}	6683	373.15
T-IHN $\rho = 40.22\%$	21 (714)	330.67	9.14×10^{-6}	83	11.28
T-IHN $\rho = 2.58\%$	186 (1964)	330.67	9.46×10^{-6}	440	48.56
T-IHN $\rho = 0.33\%$	246 (9403)	330.67	8.85×10^{-6}	553	61.25

notes the number of calling the program routine for evaluating E and g as well as M (T-IHN only), the number in parentheses is the total number of CG iterations within the D-TN and T-IHN methods, and the computer CPU time is measured by the MATLAB time functions *tic* and *toc*, where *tic* saves the current time that *toc* uses later to measure the elapsed time in seconds. Comparisons of the convergence processes

Fig. 6 Comparisons of the gradient norms of T-IHN with that of D-TN and BFGS for solving the database problem (34) with $n = 300$ and $l = 2$. The convergence rate of T-IHN is shown to be able to be close to that of D-TN and faster than that of BFGS

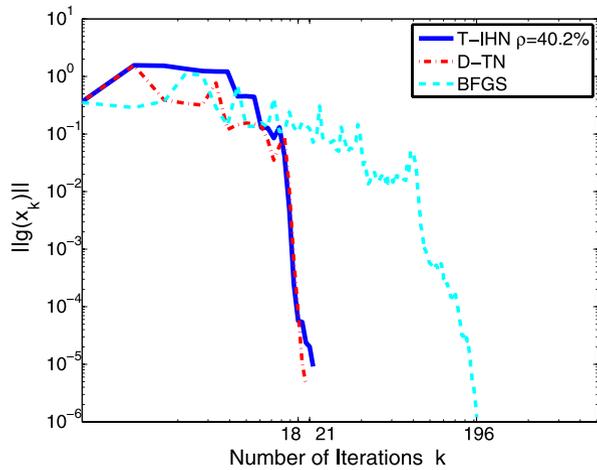
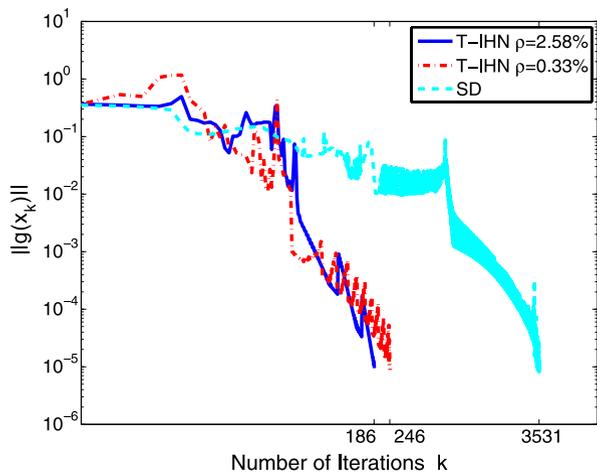


Fig. 7 Comparisons of the gradient norms of T-IHN with that of SD for solving the database problem (34) with $n = 300$ and $l = 2$. Even with two very sparse incomplete Hessian matrices, T-IHN is shown to have a much faster convergence speed than SD



of T-IHN, D-TN, BFGS, and SD in terms of gradient norms are displayed in Figs. 6 and 7.

From Table 2 and Fig. 6 we see that the T-IHN with $\rho = 40.22\%$ (i.e., about 60 percentage of the entries are zero) not only had a rate of convergence that is close to D-TN and faster than BFGS, but also had better performances than both D-TN and BFGS. In these tests, T-IHN took less CPU time by a factor of 2.4 than D-TN and by a factor of 2.2 than BFGS.

As shown in Table 2, even with a very sparse incomplete Hessian matrix, T-IHN still had a much faster convergence speed and better performance than SD. The T-IHN using the incomplete Hessian with $\rho = 2.58\%$ and 0.33% reduced the total CPU time of SD by the factors of about 7.68 and 6.09, respectively.

These numerical results demonstrate the promising potential of T-IHN as an efficient algorithm for solving the minimization problem (1). In our sequent work, we intend to further improve the convergence rate of T-IHN and study the precondition-

ing issue for T-IHN to further improve the performance of T-IHN for solving a very large database problem.

Acknowledgements The authors are grateful to Professor Dan Beard from the Medical College of Wisconsin for providing the chemical database. They are also indebted to the two anonymous referees for valuable comments and suggestions.

References

1. Dembo, R.S., Steihaug, T.: Truncated-Newton algorithms for large-scale unconstrained optimization. *Math. Program.* **26**, 190–212 (1983)
2. Golub, G.H., van Loan, C.F.: *Matrix Computations*, 2nd edn. Johns Hopkins University Press, Baltimore (1986)
3. Jolliffe, I.T.: *Principal Component Analysis*. Springer, New York (1986)
4. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. *Math. Program.* **45**, 503–528 (1989)
5. Nash, S.G., Nocedal, J.: A numerical study of the limited memory BFGS method and the truncated Newton method for large scale optimization. *SIAM J. Optim.* **1**, 358–371 (1991)
6. Nocedal, J., Wright, S.J.: *Numerical Optimization*. 2nd edn. Springer, New York (2006)
7. O’Leary, D.P.: A discrete Newton algorithm for minimizing a function of many variables. *Math. Program.* **23**, 20–33 (1982)
8. Ortega, J.M., Rheinboldt, W.C.: *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York (1970)
9. Powell, M.J.D.: Some global convergence of a variable metric algorithm for minimization without exact linear search. In: *Nonlinear Programming. SIAM-AMS Proceedings*, vol. IX. SIAM, Philadelphia (1976)
10. Schlick, T.: *Molecular Modeling and Simulation, an Interdisciplinary Guide*. Springer, New York (2002)
11. Schlick, T., Fogelson, A.: TNPACK—A truncated Newton minimization package for large-scale problems: I. Algorithm and usage. *ACM Trans. Math. Softw.* **18**, 46–70 (1992)
12. Xie, D., Schlick, T.: Efficient implementation of the truncated-Newton algorithm for large-scale chemistry applications. *SIAM J. Optim.* **9**, 132–154 (1999)
13. Xie, D., Singh, S.B., Fluder, E.M., Schlick, T.: Principal component analysis combined with truncated-Newton minimization for dimensionality reduction of chemical databases. *Math. Program.* **95**, 161–185 (2003)
14. Xie, D., Tropsha, A., Schlick, T.: An efficient projection protocol for chemical databases: the singular value decomposition combined with truncated-Newton minimization. *J. Chem. Inf. Comput. Sci.* **40**, 167–177 (2000)
15. Young, D.M.: *Iterative Solution of Large Linear System*. Academic Press, New York (1971)