# An interior-point gradient method for large-scale totally nonnegative least squares problems*

Michael Merritt and Yin Zhang

Technical Report TR04-08
Department of Computational and Applied Mathematics
Rice University, Houston, Texas 77005, U.S.A.

May, 2004

**Abstract**

We study an interior-point gradient method for solving a class of so-called totally nonnegative least squares problems. At each iteration, the method decreases the residual norm along a diagonally scaled negative gradient direction with a special scaling. We establish the global convergence of the method, and present some numerical examples to compare the proposed method with a few similar methods including the affine scaling method.

## 1 Introduction

The non-negative least squares (NNLS) problem,

$$\min_{x \geq 0} q(x) \equiv \frac{1}{2} \|Ax - b\|^2 \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $b \in \mathbb{R}^m$, is a common optimization problem arising from many applications. Without loss of generality, we will always assume that $A$ has full rank. In this paper we consider a special class of NNLS problem as defined below.

**Definition 1.** *If $b > 0$, and $A \geq 0$ with no zero rows or columns, then we call problem (1) a totally nonnegative least squares (TNNLS) problem.*

*This work was supported in part by DOE/LANL Contract 03891-99-23.

The NNLS problem is a (strictly) convex quadratic program with necessary and sufficient optimality (KKT) conditions

$$
\begin{aligned}
x \circ (A^T A x - A^T b) &= 0, &\text{(2a)}\\
A^T A x - A^T b &\geq 0, &\text{(2b)}\\
x &\geq 0, &\text{(2c)}
\end{aligned}
$$

where $\circ$ is component-wise multiplication; or simply put, $\min(x, A^T Ax - A^T b) = 0$ where the minimum is taken component-wise. For any TNNLS problem, the following simple fact holds.

**Proposition 1.** *In any TNNLS problem, $A^T b > 0$ and $A^T Ax > 0$ for any $x > 0$.*

A number of algorithms can be used to solve the NNLS problem, such as active-set methods, most notably the NNLS algorithm (Ref. 1), and the more recent primal-dual interior-point methods, see (Ref. 2), for example. Typically, these methods require matrix factorizations or updates, and can become overly expensive for very large-scale problems. Gradient-type methods, such as gradient projection methods (Refs. 3-4), only require matrix-vector multiplications, but typically have very slow convergence. Nevertheless, when high-accuracy solutions are not necessary, gradient-type methods can still be methods of choice for very large-scale applications. In this paper, we introduce a new gradient-type method for the TNNLS problem and establish its convergence. The new method can be considered an extension to the classic affine-scaling method, but since it utilizes the special structure of the TNNLS problem, it can offer a much faster convergence than the affine-scaling method at least for some problem instances, as will be illustrated later in this paper.

In section 2, we will present our new algorithm for the TNNLS problem and also state the convergence result for it. The proof of the convergence result will be developed in Section 3. Section 4 includes a couple of numerical examples for illustration purposes.

In this paper, most superscripts will refer to iteration counts and subscripts to vector components. We will use $\mathbb{N}$ to denote the set of natural numbers. The symbol $\|\cdot\|$ will denote the Euclidean norm.

## 2 An Interior-Point Gradient Algorithm

The algorithm we introduce below is basically a scaled gradient descent method where at each iteration the descent direction $p$ is the negative gradient direction $-\nabla q$ scaled by a positive vector $d > 0$, i.e.,

$$p := -d \circ \nabla q = -d \circ (A^T Ax - A^T b).$$

Like any interior-point method, the iterates are kept positive all the time by appropriate choices of step length. We will call algorithms of this type *interior-point gradient* (IPG) algorithms. The particular method we propose for solving the TNNLS problem is given below.

**IPG Algorithm:**

Initialize $x^0 > 0$ and set $k = 0$. Let $\tau \in (0, 1)$.

**Do**

1. Compute $\nabla q^k = A^T A x^k - A^T b$, and set $p^k = -d^k \circ \nabla q^k$, where $d_i^k = x_i^k / (A^T A x^k)_i$ for $i = 1, \cdots, n$.
2. Choose $\tau_k \in [\tau, 1)$ and set $\alpha_k = \min(\tau_k \hat{\alpha}_k, \alpha_k^*)$, where $\hat{\alpha}_k = \max\{\alpha : x^k + \alpha p^k \geq 0\}$, and $\alpha_k^* = -(p^k)^T \nabla q^k / (p^k)^T A^T A p^k$.
3. Set $x^{k+1} = x^k + \alpha_k p^k$, increment $k$ and go to step 1.

**End**

Note that $\alpha_k^*$ is the exact minimizer of $q(x^k + \alpha p^k)$ and $\hat{\alpha}_k$ is the step to reach the boundary of the nonnegative orthant. Thus, the step length $\alpha_k$ is chosen to ensure the next iterate is as close as possible to the exact minimizer in the $p^k$ direction without getting too close to the boundary (because $\tau_k < 1$).

The scaling vector $d$ with $d_i^k = x_i^k / (A^T A x^k)_i$ was motivated by the work of Lee and Seung (Ref. 5), even though they studied a very different algorithm for a very different problem. For the TNNLS problem, the positivity of the scaling vector $d$ is guaranteed by Proposition 1 and the positivity of the iterates. Note that if we instead use the scaling vector $d = x^2$, where the square is taken component-wise, this method becomes the classic affine-scaling algorithm (Ref. 6).

Since the sequence $\{q(x^k)\}$ is clearly monotonically decreasing and bounded below by zero, it must converge to some value $q^* \geq 0$, namely

$$\lim_{k \to \infty} q(x^k) = q^* \geq 0. \tag{3}$$

The main result of this paper is the following convergence result for the IPG algorithm, whose proof will be developed in the next section.

**Theorem 1.** *Let the NNLS problem (1) be totally nonnegative. Then the sequence of iterates generated by the IPG Algorithm converges to the unique solution of (1).*

## 3 Proof of Convergence

It is easy to see that (i) $p^k$ is a descent direction for $q(x)$ at $x^k$ whenever $\nabla q(x^k) \neq 0$, and (ii) $q(x)$ decreases monotonically along the iterates. We also recall that $A$ is assumed to be full rank and $m \geq n$, so $A^T A$ is positive definite. We need to develop several technical lemmas.

**Lemma 1.** *Let $\hat{\alpha}_k$ and $\alpha_k^*$ be defined as in the Step (2) of the IPG Algorithm. Then $\hat{\alpha}_k > 1$ and $\alpha_k^* \geq 1$. Consequently, $\alpha_k \geq \tau_k > \tau > 0$.*

*Proof.* Let us omit the superscripts $k$. Note the subscript $i$ refers to the $i$-th component of that vector. Consider for $x > 0$,

$$(x+p)_i = x_i - d_i\,(A^T A x - A^T b)_i = x_i\,\frac{(A^T b)_i}{(A^T A x)_i}.$$

Since all the quantities involved in the right-hand side are positive, clearly $x + p > 0$. This proves $\hat{\alpha}_k > 1$. Now we prove that $\alpha^* \geq 1$. Again, by substitution we calculate

$$\alpha^* = \frac{(e-u)^T \mathrm{Diag}(x \circ A^T A x)(e-u)}{(e-u)^T \mathrm{Diag}(x)(A^T A)\mathrm{Diag}(x)(e-u)}, \tag{4}$$

where $\mathrm{Diag}(x)$ is the diagonal matrix with the vector $x$ on its diagonal (similarly for the other diagonal matrix), $e$ is the vector of all ones, and $u_i = (A^T b)_i/(A^T A x)_i$. For any vector $v \in \mathbb{R}^n$, we calculate

$$\begin{aligned}
& v^T \mathrm{Diag}(x \circ A^T A x) v - v^T \mathrm{Diag}(x)(A^T A)\mathrm{Diag}(x) v \\
= \; & \sum_{i=1}^{n} v_i^2 x_i (A^T A x)_i - \sum_{i,j=1}^{n} (A^T A)_{ij} x_i x_j v_i v_j \\
= \; & \sum_{i,j=1}^{n} (A^T A)_{ij} x_i x_j v_i^2 - \sum_{i,j=1}^{n} (A^T A)_{ij} x_i x_j v_i v_j \\
= \; & \frac{1}{2}\left( \sum_{i,j=1}^{n} (A^T A)_{ij} x_i x_j (v_i^2 + v_j^2) - 2 \sum_{i,j=1}^{n} (A^T A)_{ij} x_i x_j v_i v_j \right) \\
= \; & \frac{1}{2} \sum_{i,j=1}^{n} (A^T A)_{ij} x_i x_j (v_i - v_j)^2 \geq 0,
\end{aligned}$$

since $A^T A \geq 0$ and $x > 0$. Thus, for $v = e - u$ and together with (4), we have $\alpha^* \geq 1$. Obviously, since $\alpha_k = \min(\tau_k \hat{\alpha}_k, \alpha^*)$ and $\tau_k \in [\tau, 1)$ for $\tau > 0$, we have $\alpha_k \geq \tau_k \geq \tau > 0$. $\square$

**Lemma 2.** *For the IPG Algorithm,*

$$\lim_{k\to\infty} \nabla q(x^k)^T (x^{k+1} - x^k) = \lim_{k\to\infty} \nabla q(x^k)^T p^k = 0. \tag{5}$$

*Proof.* Noting that $\alpha_k \leq \alpha_k^*$, we calculate

$$\begin{aligned}
q\left(x^{k+1}\right) &= q(x^k + \alpha_k p^k) \\
&= q(x^k) + \alpha_k (p^k)^T \nabla q^k + \frac{\alpha_k^2}{2}(p^k)^T A^T A p^k \\
&= q(x^k) + \frac{\alpha_k}{2}(p^k)^T \nabla q^k + \frac{\alpha_k}{2}(p^k)^T A^T A p^k \left(\alpha_k + \frac{(p^k)^T \nabla q^k}{(p^k)^T A^T A p^k}\right) \\
&= q(x^k) + \frac{\alpha_k}{2}(p^k)^T \nabla q^k + \frac{\alpha_k}{2}(p^k)^T A^T A p^k (\alpha_k - \alpha_k^*) \\
&\leq q(x^k) + \frac{\alpha_k}{2}(p^k)^T \nabla q^k
\end{aligned}$$

where $\nabla q^k \equiv \nabla q(x^k)$. Therefore,

$$q(x^0) \geq q(x^0) - q\left(x^{k+1}\right) = \sum_{j=0}^{k} \left(q(x^j) - q\left(x^{j+1}\right)\right) \geq -\frac{1}{2}\sum_{j=0}^{k} \alpha_j (p^j)^T \nabla q^j \geq 0,$$

from which we conclude that $\alpha_k (p^k)^T \nabla q^k = {\nabla q^k}^T (x^{k+1} - x^k) \to 0$ as $k \to \infty$. Since $\alpha^k \geq \tau > 0$ by Lemma 1, we also have $(p^k)^T \nabla q^k \to 0$ as $k \to \infty$. $\square$

**Lemma 3.** *The iterate sequence generated by the IPG Algorithm is bounded.*

*Proof.* Since $\{q(x^k)\}$ decreases monotonically, $\|Ax^k - b\| \leq \|Ax^0 - b\|$, implying

$$\|Ax^k\| \leq \|Ax^0 - b\| + \|b\| := M_1.$$

Hence, the sequence $\{Ax^k\}$ is bounded. It follows from the identity $x^k = (A^T A)^{-1} A^T (Ax^k)$

$$\|x^k\| \leq \|(A^T A)^{-1} A^T\| \|Ax^k\| \leq \|(A^T A)^{-1} A^T\| M_1 := M_2,$$

which proves the lemma. $\square$

**Lemma 4.** *If $\{x^k : k \in \mathcal{M} \subseteq \mathbb{N}\}$ is a convergent subsequence of the IPG iterates with the limit $\hat{x} \in \mathbb{R}^n$, then*

$$\hat{x} \circ \nabla q(\hat{x}) = 0.$$

*Proof.* By Lemma 2, $(p^k)^T \nabla q^k \to 0$ as $k \to \infty$. Since $(p^k)^T \nabla q^k$ is a sum of non-positive terms, each term must go to zero as well. That is, for $i = 1, \ldots, n$,

$$\begin{aligned}
& \lim_{k \to \infty} \frac{x_i^k}{(A^T A x^k)_i} (A^T A x^k - A^T b)_i^2 = 0 \\
\Rightarrow \quad & \lim_{k \in \mathcal{M}} \frac{x_i^k}{(A^T A x^k)_i} (A^T A x^k - A^T b)_i^2 = 0 \\
\Rightarrow \quad & \frac{\hat{x}_i}{(A^T A \hat{x})_i} (A^T A \hat{x} - A^T b)_i^2 = 0,
\end{aligned}$$

proving the result. $\square$

**Lemma 5.** *Let $\{x^k\}$ be the sequence of iterates generated by the IPG Algorithm 2. If $\{x^k\}$ does not converge, then there exist two distinct limit points $x^\alpha$ and $x^\beta$ and subsequences $\{x^k : k \in \mathcal{M} \subset \mathbb{N}\}$ and $\{x^{k+1} : k \in \mathcal{M} \subset \mathbb{N}\}$ such that*

$$\lim_{k\in\mathcal{M}} x^k = x^\alpha \quad \textit{and} \quad \lim_{k\in\mathcal{M}} x^{k+1} = x^\beta.$$

*Proof.* By Lemma 3, the iterate sequence is bounded. Therefore, there exists a convergent subsequence with a limit that we label $x^\alpha$. Lemma 4 implies $x^\alpha \circ \nabla q(x^\alpha) = 0$. Moreover, by (3) $q(x^\alpha) = q^*$. Let us define the index set $N = \{i : (\nabla q(x^\alpha))_i \neq 0\}$. Then clearly $x^\alpha_N = 0$ where $x^\alpha_N$ is the sub-vector of $x^\alpha$ consisting of those components corresponding to the index set $N$. By continuity, there exists $\epsilon > 0$ such that $[\nabla q(x)]_N \neq 0$ at any $x \in \mathcal{B}(x^\alpha, \epsilon)$, which is the open ball centered at $x^\alpha$ with radius $\epsilon$.

We now show that $x^\alpha$ is the only limit point of $\{x^k\}$ in $\mathcal{B}(x^\alpha, \epsilon)$. Suppose for contradiction that there exists another limit point $\hat{x} \in \mathcal{B}(x^\alpha, \epsilon)$. Then $q(\hat{x}) = q^*$, $\hat{x} \circ \nabla q(\hat{x}) = 0$, and $\hat{x}_N = 0$. The fact that $x^\alpha_N = \hat{x}_N = 0$ implies $\nabla q(x^\alpha)^T(\hat{x} - x^\alpha) = 0$. Since $q(x)$ is strictly convex and $x^\alpha \neq \hat{x}$, we have

$$q^* = q(\hat{x}) > q(x^\alpha) + \nabla q(x^\alpha)^T(\hat{x} - x^\alpha) = q(x^\alpha) = q^*,$$

a contradiction. Therefore, $x^\alpha$ is indeed the only limit point in $\mathcal{B}(x^\alpha, \epsilon)$. In addition, we will conveniently assume that there are in fact no other limit points on the boundary of $\mathcal{B}(x^\alpha, \epsilon)$ either; otherwise we only need to replace $\epsilon$ by $\epsilon/2$.

If $\{x^k\}$ does not converge, there must exist infinitely many points of $\{x^k\}$ inside $\mathcal{B}(x^\alpha, \epsilon)$ and infinitely many outside. We can thus select an infinite subset $\mathcal{M}_0 \subset \mathbb{N}$ of indices such that

$$\{x^k : k \in \mathcal{M}_0\} \subset \mathcal{B}(x^\alpha, \epsilon) \;\text{ and }\; \{x^{k+1} : k \in \mathcal{M}_0\} \subset \mathbb{R}^n \backslash \mathcal{B}(x^\alpha, \epsilon).$$

Since the iterates are bounded, both of the above subsequences themselves have convergent subsequences so that we can further select $\mathcal{M} \subseteq \mathcal{M}_0$ such that

$$\lim_{k\in\mathcal{M}} x^k = x^\alpha \in \mathcal{B}(x^\alpha, \epsilon) \;\text{ and }\; \lim_{k\in\mathcal{M}} x^{k+1} = x^\beta \in \mathbb{R}^n \backslash \overline{\mathcal{B}(x^\alpha, \epsilon)},$$

for some $x^\beta$ where $\overline{\mathcal{B}}$ denotes the closure of $\mathcal{B}$. Clearly, $x^\alpha \neq x^\beta$. $\square$

Now we are ready to prove Theorem 1. We will prove the result in two stages, first showing that the iterates converge, and then showing that the limit satisfies the optimality conditions.

**Proof of Theorem 1:**

*Proof.* First, we will prove by contradiction that the iterates converge. Suppose they do not converge. Then, by Lemma 5, there exist two distinct limit points $x_\alpha \neq x_\beta$ of $\{x^k\}$ such that for some $\mathcal{M} \subset \mathbb{N}$,

$$\lim_{k\in\mathcal{M}} x^k = x^\alpha \quad \text{and} \quad \lim_{k\in\mathcal{M}} x^{k+1} = x^\beta.$$

It follows from Lemma 2 that $\nabla q(x_\alpha)^T(x_\beta - x_\alpha) = 0$, and from (3) $q(x_\alpha) = q(x_\beta) = q^*$. However, since $q(x)$ is a strictly convex function, $q(x_\beta) > q(x_\alpha) + \nabla q(x_\alpha)^T(x_\beta - x_\alpha) = q(x_\alpha)$, a contradiction. Thus, the iterates must converge.

Next, we prove the iterates converge to the optimal solution of the TNNLS problem by verifying the KKT conditions (2).

Let $x^*$ be the limit of $\{x^k\}$. Since $x^* \geq 0$ and $x^* \circ \nabla q(x^*) = 0$ (by Lemma 4), we only need to show that $\nabla q(x^*) \geq 0$.

Suppose some component $(\nabla q(x^*))_i < 0$. Then by complementarity $x_i^* = 0$. Continuity of $\nabla q(x)$ implies $(\nabla q(x^k))_i < 0$ for all sufficiently large $k$. Note that for all components $i$,

$$(\nabla q(x^k))_i\,(x^{k+1} - x^k)_i = \alpha^k (\nabla q(x^k))_i\, p_i^k = -\alpha^k d_i^k (\nabla q(x^k))_i^2 \leq 0$$

since $\alpha^k > 0$ and $d^k > 0$. So, $x_i^{k+1} \geq x_i^k$ for all $k$ sufficiently large, contradicting the fact that $x_i^k \to x_i^* = 0$. This completes the proof. □

# 4 Numerical Examples

The convergence behavior of the IPG algorithm depends on the scaling vector $d$. The proposed choice in this paper is $d_i = x_i/(A^TAx)_i$. In this section we present a couple of numerical examples to illustrate that the proposed scaling seems to compare favorably with the well-know affine scaling.

We consider an instance of the NNLS problem (1) where the data $(A, b)$ are generated by the Matlab scripts:

```
rand('state',0);
m = 800; n = 400; density = 0.05; condA = 1.e+2;
A = sprand(m,n,density,1/condA); b = rand(m,1);
```

and we set the random-number generator to the state of 0 so that the same data can be re-generated. The elements of the matrix $A$ and the right-hand side vector $b$ are uniformly distributed "pseudo-random" numbers between zero and one.

For the IPG algorithm, we use four different scalings: $d = x^p$ for exponent $p = 1, 2, 3$, and the proposed choice $d_i = x_i/(A^TAx)_i$. We again note that $d = x^2$ is the affine scaling.

We ran the IPG algorithm with the above four scalings and a maximum number of 1000 iterations. In Figure 1, we plot the relative errors, $[q(x^k) - q(x^*)]/q(x^*)$, against the iteration number, where $x^*$ is the optimal solution computed by an active-set method.

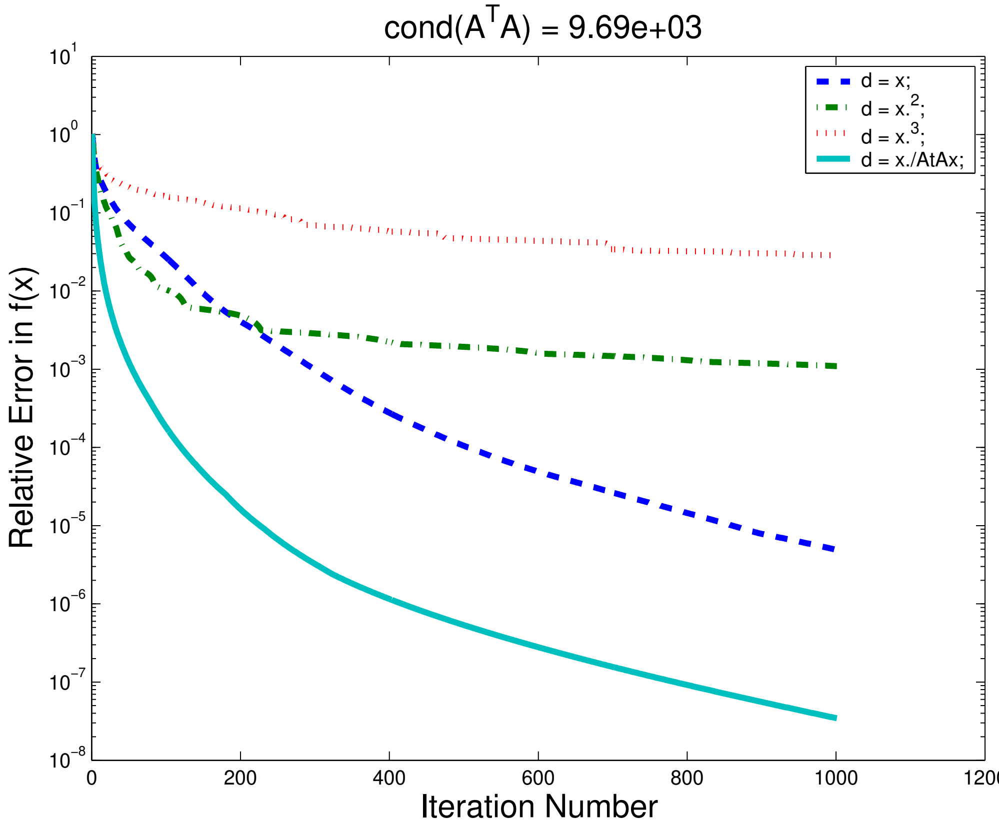


Figure 1: Relative errors $[q(x^k) - q(x^*)]/q(x^*)$ for 4 scaling sequences (I)

As can be seen from Figure 1, the four scalings have led to very different convergence behaviors on this particular test problem. The scaling $d = x^3$ has the slowest convergence, followed by $d = x^2$ and then $d = x$, while the scaling $d_i = x_i/(A^T Ax)_i$ has the fastest convergence. The differences in the behavior of convergence for these four scalings are quite significant. In Figure 2, we repeat the experiment on data where the problem is much more ill-conditioned ($\text{cond}(A^T A) \approx 10^{15}$). For this ill-conditioned problem, the advantage of the proposed scaling becomes more pronounced. It seems relatively unaffected by the ill-conditioning in comparison to other scalings.

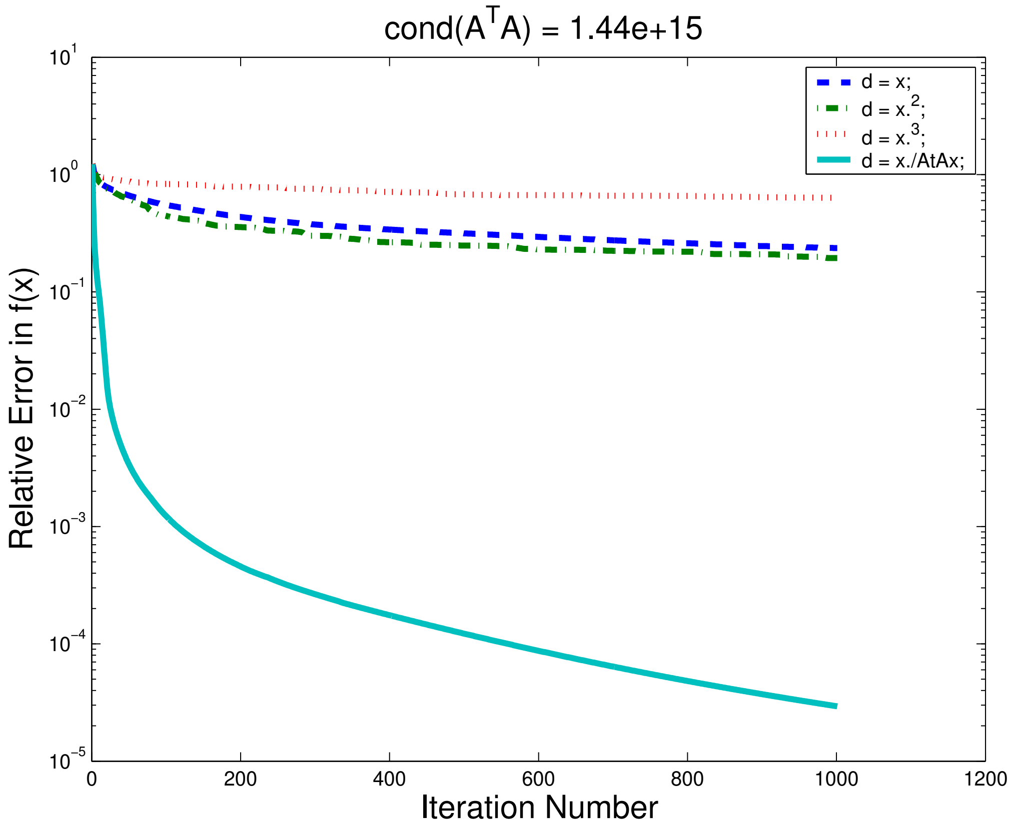


Figure 2: Relative errors $[q(x^k) - q(x^*)]/q(x^*)$ for 4 scaling sequences (II)

# 5 Concluding Remarks

In this paper we have presented an interior-point gradient algorithm with a new scaling suitable for solving the totally nonnegative least squares (TNNLS) problems. We have established global convergence for this method. Preliminary numerical examples indicate that the proposed algorithm seems promising for solving large-scale TNNLS problems when highly accurate solutions are not necessary, though more extensive numerical testing is still needed. Convergence results for a wider class of interior-point gradient methods applicable to more general problems have recently been obtained by the second author of this paper (Ref. 7).

## References


[1] LAWSON, C. and HANSON, R., *Solving Least Squares Problems*, Chapter 23. Prentice Hall, 1974.

[2] WRIGHT, S., *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, 1997.

[3] GOLDSTEIN, A., *Convex programming in Hilbert space.* Bull. Amer. Math. Soc., 70: pp. 709–710, 1964.

[4] LEVITIN, E. S. and POLYAK, B. T., *Constrained minimization methods.* USSR Comput. Math. Math. Phys., Vol. 6: pp. 1–50, 1966.

[5] LEE, D. D. and SEUNG, H. S., *Algorithms for Non-negative matrix factorization.* Advances in Neural Information Processing Systems, Vol. 13: pp. 556–562, 2001.

[6] DIKIN, I. I., *Iterative solution of problems of linear and quadratic programming.* Sov. Math. Doklady, 8: pp. 674–675, 1967.

[7] ZHANG, Y., *Interior-Point Gradient Methods with Diagonal-Scalings for Simple-Bound Constrained Optimization.* CAAM Dept. Technical Report TR04-06, Rice University, Houston, Texas, 2004.