COMPUTING MINIMUM NORM SOLUTION OF LINEAR SYSTEMS OF EQUATIONS BY THE GENERALIZED NEWTON METHOD

SAEED KETABCHI*
Department of Applied Mathematics
Faculty of Mathematical Sciences
University of Guilan, Rasht, Iran

HOSSEIN MOOSAEI
Department of Mathematics, Faculty of Science
University of Bojnord, Bojnord, Iran

M. PARANDEGAN
Department of Applied Mathematics
Faculty of Mathematical Sciences
University of Guilan, Rasht, Iran

HAMIDREZA NAVIDI
Shahed University, Tehran, Iran

(Communicated by Wenyu Sun)

Abstract. The aim of this paper is to find the minimum norm solution of a linear system of equations. The proposed method is based on presenting a view of solution on the dual exterior penalty problem of primal quadratic programming. To solve the unconstrained minimization problem, the generalized Newton method was employed and to guarantee its finite global convergence, the Armijo step size regulation was adopted. This method was tested on all systems selected in NETLIB 1. Numerical results were compared with the MOSEK Optimization Software 2 on linear systems in NETLIB (Table 1) and on linear systems generated by the Linear systems generator (Table 2).

1. Introduction. When a linear system has multiple solutions, it may be important to select a particular solution, and in these cases, a natural choice would be the solution with the minimum norm [2][10].
Consider the following problem:

\[
\min \frac{1}{2} \|x\|^2 \quad \text{s.t.} \quad Ax = b, \ x \geq 0, \quad (1)
\]

where \( A \in R^{m \times n} \) and \( b \in R^m \).

2010 Mathematics Subject Classification. Primary: 90C06; Secondary: 90C20.
Key words and phrases. piecewise quadratic programs, minimum norm solution, generalized Newton method, Armijo rule, penalty parameter.
* Corresponding author: Saeed Ketabchi.
1 www.netlib.org
2 www.mosek.com.
The solution can be obtained by minimizing dual exterior penalty problem of primal quadratic programming \[10, 11\]. The alternative method offers another way to solve this problem \[7, 9\]. The use of dual exterior penalty problem of primal quadratic programming \[1\] leads to minimization function, which is piecewise quadratic, convex, and only once differentiable. On the other hand, for this function, the ordinary Hessian is not applicable. Here, for the minimization of this function, the generalized Newton method \[10, 11, 12\], along with the Armijo step size regulation \[1, 3, 4, 13, 14\] were used. Both methods and generalized Newton method were discussed in details in Sections 2 and 3. In the last section, a comparison of the test results with \textit{MOSEK} was presented.

As for our notations, a few points need to be made. By denoting \(n\)-dimensional real space by \(\mathbb{R}^n\), we mean the transpose of matrix \(A\) and Euclidean norm and \(\infty\)-norm, respectively. \(a_+\) replaces negative components of the vector \(a\) by zeros.

2. Main Results. In this section, we explain two approaches discussed in the introduction.

Consider the following function

\[ h(u) = f(A^T u) - g(b^T u), \]

where \(f\) is an arbitrary differentiable function on \(\mathbb{R}^n\) to \(\mathbb{R}\), and \(g\) is an arbitrary real and differentiable function. The next result provides sufficient conditions to determine minimum norm solution for the problem \[1\].

**Theorem 2.1.** Let \(u^*\) be a stationery point for function \(h\) and assume that there exist a vector \(v^* \in \mathbb{R}^m\) such that

\[ (A^T v^*)_+ = \nabla f(A^T u^*) / g'(b^T u^*), \]

then \(x^* = (A^T v^*)_+ / g'(b^T u^*)\) is the solution to problem \[1\].

**Proof.** Let \(u^*\) be a stationery point for \(h\) and \(v^* \in \mathbb{R}^m\) satisfies \[2\], then we will have \(\nabla h(u^*) = A \nabla f(A^T u^*) - bg'(b^T u^*) = 0\) and from here \(A(A^T v^*)_+ - b = 0\). This means \((A^T v^*)_+\) is the feasible solution to \[1\].

Since \(x^* = (A^T v^*)_+\), it follows

\[ x^* - A^T v^* \geq 0, \]

\[ x^* (x^* - A^T v^*) = 0, \]

\[ Ax^* - b = 0. \]

Thus from \[4\], \[5\], and \[6\], it can be concluded that \(x^*\) satisfies Karush-Kuhn-Tucker optimality conditions for problem \[1\].

In the following section, two approaches based on the Theorem 2.1 for solving problem \[1\] are described.

2.1. **First approach.** For the first approach, we consider

\[ h(u) = \frac{1}{2} \| (A^T u)_+ \|^2 - \varepsilon b^T u. \]

As shown in \[10, 11\] there exists some positive \(\bar{\varepsilon}\) so that if we choose \(\varepsilon \in (0, \bar{\varepsilon})\), then \(x^* = \frac{1}{\varepsilon}(A^T u^*)_+\) will be the minimum norm solution of \[1\]. Therefore, from
and from Theorem 2.1 we have the following:
\[
\nabla f(A^T u^*) = (A^T u^*)_+, \quad g'(b^T u^*) = \varepsilon, \quad v^* = \frac{u^*}{\varepsilon}.
\]
In fact this approach is based on the dual exterior penalty problem.

2.2. **Second approach.** In the second approach we consider \( h \) as follows,
\[
 h(u) = \| (A^T u)_+ \|^2 + \| b^T u - \rho \|^2, \quad \rho > 0.
\]
(8)

If the system \( Ax = b, \ x \geq 0 \) is feasible, then \((\rho - b^T u^*) > 0 \) and \( x^* = \frac{1}{\rho - b^T u^*} (A^T u^*)_+ \) will be the minimum norm solution of (1) (see [2]). As such, from [2] and from Theorem 2.1 we have the following:
\[
\nabla f(A^T u^*) = 2(A^T u^*)_+, \quad 2g'(b^T u^*) = b^T u^* - \rho, \quad v^* = \frac{u^*}{\rho - b^T u^*}.
\]
This approach is based on the alternative method.

3. **Generalized Newton Method.** In this section, the generalized Newton method for solving the unconstrained optimization problem for the first approach is explained. Function \( h(u) = \frac{1}{2}\| (A^T u)_+ \|^2 - \varepsilon b^T u \) is piecewise quadratic, convex, and differentiable, but it is not twice differentiable.

By assuming \( u \) and \( s \in \mathbb{R}^m \), for gradient of \( h(u) \), we have
\[
\| \nabla h(u) - \nabla h(s) \| \leq \| A \| \| A^T \| \| u - s \|,
\]
this means \( \nabla h \) is globally Lipschitz continuous with constant \( K = \| A \| \| A^T \| \). Thus, for this function, the existence of a generalized Hessian is approved, which is defined as \( m \times m \) symmetric positive semidefinite matrix [8]
\[
\nabla^2 h(u) = AD(z)A^T,
\]
where \( D(z) \) denotes an \( n \times n \) diagonal matrix with \( i \)-diagonal element and \( z_i \) is equal to 1 if \( A^T u(\varepsilon)_i > 0 \); otherwise, it is equal to 0. Therefore, the generalized Newton method can be used to solve this problem, and it requires a line-search algorithm to obtain global termination (see [13]). In the following algorithm, the generalized Newton method is applied with a line-search based on the Armijo rule.

---

**Generalized Newton Method with the Armijo Rule**

Choose any \( u_0 \in \mathbb{R}^m \) and \( \varepsilon > 0 \)

\( i=0; \)

while \( \| \nabla h(u_i) \| \geq \varepsilon \)

Choose \( \alpha_i = \max\{s, s\delta, s\delta^2, \ldots\} \) such that
\[
 h(u_i) - h(u_i + \alpha_i d_i) \geq -\alpha_i \mu \nabla h(u_i) d_i,
\]
where \( d_i = -\nabla^2 h(u_i)^{-1} \nabla h(u_i) \), \( s > 0 \) be a constant, \( \delta \in (0, 1) \) and \( \mu \in (0, 1) \).

\( u_{i+1} = u_i + \alpha_i d_i \)

\( i = i + 1; \)

end

---

In this algorithm, the generalized Hessian may be singular, so we used a modified Newton direction Cholesky factorizations as follows:
\[
M^T M = (\nabla^2 h(u_k) + \gamma I_m), \quad d_k = -(M^T M)^{-1} \nabla h(u_k),
\]
where $M$ is an upper triangular matrix, $\gamma$ is a small positive number and $I_m$ is the identity matrix of $m$ order.

Now, the following iterative process can be introduced,

$$u_{k+1} = u_k + \alpha_k d_k,$$

and if

$$u^* = \arg \min_{u \in R^m} \left\{ -\varepsilon b^T u + \frac{1}{2} \| (A^T u)_+ \|^2 \right\},$$

then $x^* = \frac{1}{\varepsilon} (A^T u^*)_+$ is the minimum norm solution to (1).

The proof for the finite global convergence of this algorithm is given in [1, 11].

4. Numerical Results. This section presents some numerical results on linear systems in NETLIB (See Table 1) along with various randomly generated systems (See Table 2) to test the iterative process (9). The test system generator creates a random matrix $A$ for a given $m, n$ and density $d$. The elements of $A$ are uniformly distributed between $-50$ and $+50$. These linear systems are generated using the following MATLAB code:

```
Linear systems generator
%Sgen: Generate random solvable System: Ax = b ;x>=0,
%Input: m,n,d(ensity); Output: A,b,x;
%m=input('Enter m:')
n=input('Enter n:')
d=input('Enter d:')(abs(x)+x)/2
pl=inline('(abs(x)+x)/2')
A=sprand(m,n,d);A=100*(A-0.5*spones(A)); x=sparse(10*pl(rand(n,1)));
b=A*x;
format short e;
[norm(A*x-b)]
```

In Table(1), a comparison has been drawn between the MOSEK Optimization Software for convex quadratic problems (cqpMosek) and our method, which were implemented in MATLAB 7.6 code (ssGNewton) on systems in NETLIB. The results of the comparison on various randomly generated systems, are shown in Table(2).

In the following solved examples, the starting vector is $u^0 = 0_m$, and $\varepsilon = (n/m)10^{-5}$, $tol = 10^{-10}$. For this purpose, we used 5000 GHz, AMD 64 Athlon X2 Dual Core with 2 Gigabyte of memory. The total time of computations in each example is provided in the second column of the table.

5. Conclusion. In this paper, we proposed the use of fast generalized Newton method and Armijo rule to obtain the minimum norm solution for a linear system. We probed its finite global convergence with the Armijo step size regulation. The solution and convergence of this method were analyzed. In support of the predicted theory, several test examples were solved using this method. A comparison of the numerical results was made by tabulating. The computational results on test large-scale linear systems demonstrated the effectiveness of the ssGNewton compared to the MOSEK Optimization Software for convex quadratic problems (cqpMosek).

---

3 The Matlab code of this paper is available from the authors upon request.
Table 1. Comparison of \textit{ssGNewton} and \textit{cqpMosek} on systems in NETLIB. In all solved examples $\|(-x^*)_+\| = 0$.

| Systems | Method   | \textit{time (sec)} | $\|x^*\|$ | $\|Ax^* - b\|_\infty$ |
|---------|----------|----------------------|------------|--------------------------|
| 25fe47  | ssGNewton | 1.51                 | $3.31 \times 10^3$ | $3.43 \times 10^{-9}$ |
|         | cqpMosek | 1.36                 | $3.31 \times 10^3$ | $1.91 \times 10^{-14}$ |
| 80bau3b | ssGNewton | 0.95                 | $4.13 \times 10^3$ | $1.18 \times 10^{-12}$ |
|         | cqpMosek | 0.80                 | $4.13 \times 10^3$ | $2.90 \times 10^{-14}$ |
| addittle| ssGNewton | 0.05                 | $4.3 \times 10^2$ | $2.27 \times 10^{-13}$ |
|         | cqpMosek | 0.35                 | $4.3 \times 10^2$ | $7.18 \times 10^{-14}$ |
| astro   | ssGNewton | 0.06                 | $6.34 \times 10^2$ | $6.39 \times 10^{-14}$ |
|         | cqpMosek | 0.31                 | $6.34 \times 10^2$ | $1.13 \times 10^{-14}$ |
| creb    | ssGNewton | 13.20                | $6.24 \times 10^2$ | $1.62 \times 10^{-9}$  |
|         | cqpMosek | 2.31                 | $6.24 \times 10^2$ | $1.61 \times 10^{-9}$  |
| maroser7| ssGNewton | 2.86                 | $1.41 \times 10^5$ | $2.54 \times 10^{-11}$ |
|         | cqpMosek | 55.20                | $1.41 \times 10^5$ | $3.27 \times 10^{-14}$ |
| pds02   | ssGNewton | 2.30                 | $1.61 \times 10^5$ | $1.40 \times 10^{-7}$  |
|         | cqpMosek | 0.51                 | $1.61 \times 10^5$ | $8.2 \times 10^{-9}$   |
| ken13   | ssGNewton | 9.09                 | $2.53 \times 10^4$ | $4.39 \times 10^{-9}$  |
|         | cqpMosek | 2.09                 | $2.53 \times 10^4$ | $1.71 \times 10^{-9}$  |
| osu14   | ssGNewton | 60.10                | $1.19 \times 10^5$ | $4.10 \times 10^{-8}$  |
|         | cqpMosek | 4.40                 | $1.19 \times 10^5$ | $7.82 \times 10^{-9}$  |
| agg3    | ssGNewton | 0.27                 | $7.65 \times 10^5$ | $3.59 \times 10^{-8}$  |
|         | cqpMosek | 0.40                 | $7.65 \times 10^5$ | $2.32 \times 10^{-10}$ |
| crea    | ssGNewton | 1.25                 | $1.62 \times 10^3$ | $4.13 \times 10^{-6}$  |
|         | cqpMosek | 0.61                 | $1.62 \times 10^3$ | $2.15 \times 10^{-6}$  |

\textbf{Remark:} By using the "format long e" command in MATLAB, if $x^*_{\text{ssGNewton}}$ is the optimal solution for \textit{ssGNewton} and $x^*_{\text{cqpMosek}}$ is the optimal solution for
Table 2. Comparison of \textit{ssGNewton} and \textit{cqpMosek} on randomly generated systems. 'OOM' denotes out of memory. In all solved examples $\|(-x^*)_+\| = 0$.

| $m$, $n$, $d$ | Method       | time (sec) | $\|x^*\|$ | $\|Ax^* - b\|_\infty$ |
|--------------|--------------|------------|------------|-------------------|
| $1000 \times 1200 \times 0.1$ | \textit{ssGNewton} | 6.92 | $1.87 \times 10^6$ | $7.73 \times 10^{-12}$ |
|               | \textit{cqpMosek} | 14.70 | $1.87 \times 10^9$ | $5.45 \times 10^{-12}$ |
| $1000 \times 1500 \times 0.01$ | \textit{ssGNewton} | 3.94 | $1.90 \times 10^2$ | $1.13 \times 10^{-12}$ |
|               | \textit{cqpMosek} | 3.10 | $1.90 \times 10^2$ | $6.80 \times 10^{-12}$ |
| $2000 \times 10000 \times 0.1$ | \textit{ssGNewton} | 29.10 | $2.94 \times 10^2$ | $4.54 \times 10^{-11}$ |
|               | \textit{cqpMosek} | 122.00 | $2.94 \times 10^2$ | $6.18 \times 10^{-11}$ |
| $4500 \times 5000 \times 0.1$ | \textit{ssGNewton} | OOM | – | – |
|               | \textit{cqpMosek} | 339.00 | $3.95 \times 10^2$ | $3.89 \times 10^{-8}$ |
| $1000 \times (5 \times 10^6) \times 10^{-6}$ | \textit{ssGNewton} | 19.20 | $2.02 \times 10^2$ | $2.27 \times 10^{-13}$ |
|               | \textit{cqpMosek} | OOM | – | – |
| $1000 \times 100000 \times 0.01$ | \textit{ssGNewton} | 4.54 | $2.45 \times 10^2$ | $3.27 \times 10^{-11}$ |
|               | \textit{cqpMosek} | 9.10 | $2.45 \times 10^2$ | $3.63 \times 10^{-11}$ |
| $100 \times (2 \times 10^6) \times 0.001$ | \textit{ssGNewton} | 4.35 | $8.13 \times 10^1$ | $5.09 \times 10^{-11}$ |
|               | \textit{cqpMosek} | 30.40 | $8.13 \times 10^1$ | $4.69 \times 10^{-11}$ |
| $(2.5 \times 10^5) \times (3 \times 10^5) \times 10^{-6}$ | \textit{ssGNewton} | 70.12 | $1.41 \times 10^3$ | $4.10 \times 10^{-3}$ |
|               | \textit{cqpMosek} | 5.32 | $1.41 \times 10^3$ | $4.90 \times 10^{-3}$ |
| $10^5 \times (2 \times 10^5) \times 10^{-6}$ | \textit{ssGNewton} | 9.85 | $7.65 \times 10^2$ | $2.87 \times 10^{-8}$ |
|               | \textit{cqpMosek} | 3.26 | $7.65 \times 10^2$ | $2.27 \times 10^{-13}$ |

\textit{cqpMosek}, then for all solved systems we have

$$\|x^*_\text{ssGNewton}\| < \|x^*_\text{cqpMosek}\|.$$  

For example in the problem "80bau3b"

$$\|x^*_\text{ssGNewton}\| = 4.129965300963039e + 003,$$

$$\|x^*_\text{cqpMosek}\| = 4.129965305352692e + 003,$$

or in the problem "osa14"

$$\|x^*_\text{ssGNewton}\| = 1.195823216292518e + 005,$$
\[ \| x^*_{\text{capMosek}} \| = 1.195823217807026e + 005. \]

Acknowledgments. The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the manuscript.

REFERENCES

[1] L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, *Pacific Journal of Mathematics*, 16 (1966), 1–3.
[2] T. V. Anh, An extragradient method for finding minimum-norm solution of the split equilibrium problem, *Acta Mathematica Vietnamica*, 44 (2016), 1–18.
[3] M. S. Bazaraa, H. D. Sherali and C. M. Shetty, *Nonlinear Programming Theory and Algorithms*, John Wiely and Sons, 1999.
[4] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 2014.
[5] Yu. G. Evtushenko, A. I. Golikov and N. Mollaverdi, Augmented Lagrangian method for large-scale linear programming problems, *Optimization Methods and Software*, 20 (2005), 515–524.
[6] Yu. G. Evtushenko, A. I. Golikov, S. Ketabchi and N. Mollaverdi, The augmented Lagrangian function for the linear programming problems, *Dynamics of non- homogeneous systems (Ed. YuSPopkov)*, *Russian Academy of Sciences Institute for System Analysis*, 2 (2004), 101–106.
[7] A. I. Golikov and Yu. G. Evtushenko, Theorems of the alternative and their applications in numerical methods, *Comput. Maths. and Math. Phys*, 43 (2003), 338–358.
[8] J. B. Hiriart-Urruty, J. J. Strodiot and V. H. Nguyen, Generalized Hessian matrices and second-order optimality conditions for problems $C^{1,1}$ data, *Applied Mathematics and Optimization*, 11 (1984), 43–56.
[9] S. Ketabchi and E. Ansari-Piri, On the solution set of convex problems and its numerical application, *Journal of Computational and Applied Mathematics*, 206 (2007), 288–292.
[10] C. Kanzow, H. Qi and L. Qi, On the minimum norm solution of linear programs, *Journal of Optimization Theory and Applications*, 116 (2003), 333-345.
[11] O. L. Mangasarian, A Newton method for linear programming, *Journal of Optimization Theory and Applications*, 121 (2004), 1–18.
[12] O. L. Mangasarian, A finite Newton method for classification, *Optimization Methods and Software*, 17 (2002), 913–930.
[13] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Science, 1999.
[14] W. Sun and Y. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, New York, 2006.

Received July 2015; 1st revision January 2017; final revision May 2017.

E-mail address: sketabchi@guilan.ac.ir
E-mail address: hmoosaei@gmail.com, moosaei@ub.ac.ir
E-mail address: m.parandegan@gmail.com
E-mail address: navidi@shahed.ac.ir