Computer Science Journal of Moldova, vol.**17**, no.1(49), 2009

Computational Experiments on the Tikhonov Regularization of the Total Least Squares Problem

Maziar Salahi* Hossein Zareamoghaddam

Abstract

In this paper we consider finding meaningful solutions of ill-conditioned overdetermined linear systems $Ax \approx b$, where A and b are both contaminated by noise. This kind of problems frequently arise in discretization of certain integral equations. One of the most popular approaches to find meaningful solutions of such systems is the so called total least squares problem. First we introduce this approach and then present three numerical algorithms to solve the resulting fractional minimization problem. In spite of the fact that the fractional minimization problem is not necessarily a convex problem, on all test problems we can get the global optimal solution. Extensive numerical experiments are reported to demonstrate the practical performance of the presented algorithms.

Keywords: Linear systems, Total least squares, Tikhonov regularization, Newton method, Bisection method.

1 Introduction

In this paper we aim to find meaningful solutions for the linear systems of the form

$$Ax \approx b, \tag{1}$$

where $A \in R^{m\times n}$, $b \in R^m$, $m \geq n$ are both contaminated by noise. This kind of systems frequently arise in discretization of certain integral equations [3].

* This author is a corresponding author

If A is ill-conditioned, a quite effective procedure to find a reasonably good solution for (1) is to use the regularized least squares approach. Perhaps the best known regularization technique is due to Tikhonov [6], which solves the following minimization problem rather than classical least squares one:

$$\min_{x \in R^n} \|Ax - b\|^2 + \rho \|x\|^2 , \tag{2}$$

where ρ is a positive constant.

It is worth mentioning that when $\rho = 0$ (the classical least squares approach), and A is ill-conditioned, then the solution of (2) might have large norm, while for positive ρ it is not the case. It is not easy to find the exact value of ρ, however there have been some studies on this subject [5]. It is obvious that (2) is a convex minimization problem and its optimal solutions should satisfy

$$(A^T A + \rho I)x = A^T b. \tag{3}$$

We may use existing efficient iterative algorithms like conjugate gradient methods to solve (3).

Another most popular approach to deal with such systems is the so called total least squares problem [1, 4]. This approach leads to a fractional nonconvex minimization problem. In this paper we present three efficient algorithms to solve it. Extensive numerical results are reported to show the efficiency of the discussed algorithms.

2 Total least squares problem

In this approach, one aims to find a feasible system by minimal changes in problem data i.e.,

$$\begin{aligned} \min_{x,E,r} \quad & \|E\|^2 + \|r\|^2 \\ & (A + E)x = b + r. \end{aligned} \tag{4}$$

The optimal E and r values are given in the following theorem.

Theorem 1. *The optimal E and r values of problem (4) are given by*

$$r^* = \frac{Ax^* - b}{1 + \|x^*\|^2}, \ E^* = -\frac{Ax^* - b}{1 + \|x^*\|^2} x^{*T}$$

where x^* is the optimal solution of

$$\min_{x \in R^n} \frac{\|Ax - b\|^2}{1 + \|x\|^2}. \tag{5}$$

Proof. The minimization problem in (4) can be written as two minimization problems as follows:

$$\begin{aligned} \min_{x \in R^n} \min_{E,r} \quad & \|E\|^2 + \|r\|^2 \\ & (A + E)x = b + r. \end{aligned}$$

Let us first consider the inner minimization problem. Obviously it is a convex optimization problem, therefore the KKT conditions are necessary and sufficient for optimality that follows:

$$\begin{aligned} & 2E^* + \lambda^* x^T = 0 \\ & 2r^* - \lambda^* = 0 \\ & (A + E^*)x - b - r^* = 0, \end{aligned} \tag{6}$$

where the vector λ^* denotes the lagrange multipliers. From the second equation of (6) we have $\lambda^* = 2r^*$ and subsequently from the first equation we have $E^* = -r^* x^T$. Finally, the last equation implies that

$$r^* = \frac{Ax - b}{1 + \|x\|^2},$$

and subsequently

$$E^* = -\frac{Ax - b}{1 + \|x\|^2} x^T.$$

Now the objective function of inner minimization problem becomes

$$\frac{\|Ax - b\|^2}{1 + \|x\|^2}.$$

Thus if x^* be an optimal solution of this problem, then the proof is completed. $\square$

Therefore, by solving this minimization problem we have a modified linear system which is feasible. Since the original system is ill-conditioned, then the solution of (5) might be meaningless from practical point of view due to the large norm. Thus we can stabilize the solution by utilizing the Tikhonov regularization technique. The regularized problem becomes:

$$\min_{x \in R^n} f(x) := \frac{\|Ax - b\|^2}{1 + \|x\|^2} + \rho \|x\|^2 , \tag{7}$$

where ρ is a nonnegative parameter. As it is obvious, problem (7) is not known to be convex or concave in general. In the sequel we present several numerical algorithms, which can help us to solve (7) up to global optimality. First let us derive the gradient and hessian of the objective function of (7) as follows:

$$\nabla f(x) = \frac{2A^T(Ax - b)}{1 + \|x\|^2} - \frac{2\|Ax - b\|^2 x}{(1 + \|x\|^2)^2} + 2\rho x$$

$$\begin{aligned} \nabla^2 f(x) = \quad & \frac{2A^T A}{1 + \|x\|^2} - \frac{4x(A^T(Ax - b))^T}{(1 + \|x\|^2)^2} + 2\rho I - \frac{4A^T(Ax - b)x^T}{(1 + \|x\|^2)^2} + \\ & \left(\frac{8xx^T}{(1 + \|x\|^2)^3} - \frac{2}{(1 + \|x\|^2)^2} I \right) \|Ax - b\|^2 . \end{aligned}$$

The first approach which we utilize to tackle (7) numerically is the classical Newton method. During this process one might end up with an iterate when the hessian is singular or very close to singularity, but a slight perturbation of it usually resolves this bad behavior. Although the objective function of (7) is not known to be convex, but for most of the test problems we have considered, it yields a global solution as it will be shown in the next section. The structure of the algorithm is as follows:

Newton Based Algorithm

Inputs: An accuracy parameter $\epsilon > 0$;
A regularization parameter ρ;
A parameter δ usually 10^{-4};
A starting point $x_0 \in R^n$.
begin
i=0;
while $\|\nabla f(x_i)\| \geq \epsilon$
Find an appropriate α by Armijo line search and let
$x_{i+1} = x_i - \alpha(\nabla^2 f(x_i) + \delta I)^{-1}\nabla f(x_i)$.
i=i+1;
end
end

In the sequel we present another algorithm by an old idea due to Dinkelbach [2] which uses an equivalent formulation of the problem (7) to solve it. It is obvious that

$$\min_{x\in R^n}\{\frac{\|Ax-b\|^2}{1+\|x\|^2} + \rho\,\|x\|^2\} \leq t$$

is equivalent to

$$\min_{x\in R^n}\{\|Ax-b\|^2 - t(1+\|x\|^2) + \rho(\|x\|^2 + \|x\|^4)\} \leq 0. \tag{8}$$

Now let us define

$$\Phi(t) = \min_{x\in R^n}\{\|Ax-b\|^2 - t(1+\|x\|^2) + \rho(\|x\|^2 + \|x\|^4)\}.$$

Lemma 1. *The function $\Phi(t)$ is a strictly decreasing function.*

Proof. Let $t_1 < t_2$ and x_{t_1} be the point for which

$$\Phi(t_1) = \|Ax_{t_1} - b\|^2 - t_1(1+\|x_{t_1}\|^2) + \rho(\|x_{t_1}\|^2 + \|x_{t_1}\|^4).$$

Then we have

$$\Phi(t_1) > \|Ax_{t_1} - b\|^2 - t_2(1+\|x_{t_1}\|^2) + \rho(\|x_{t_1}\|^2 + \|x_{t_1}\|^4) \geq \Phi(t_2).$$

Therefore, $\Phi(t_1) > \Phi(t_2)$. □

We further have that $\Phi(0) > 0$ and

$$\Phi(\|b\|^2) \leq \|A0 - b\|^2 - \|b\|^2 (1 + \|0\|^2) + \rho(\|0\|^2 + \|0\|^4) = 0$$

Therefore function $\Phi(t)$ has a unique root in the interval $[0, \|b\|^2]$. Now our goal is to find this root. First in the Next lemma we prove that this gives us the global minimum of (7). Then explain how to find the root numerically.

Lemma 2. *The root of the function $\Phi(t)$ gives the global minimum of problem (7).*

Proof. Let t^* be the root of $\Phi(t)$. Then

$$\min_{x \in R^n} \{\|Ax - b\|^2 - t^*(1 + \|x\|^2) + \rho(\|x\|^2 + \|x\|^4)\} = 0.$$

Let x^* be the point on which this minimum happens. Then for any $x \in R^n$ one has

$$\|Ax - b\|^2 - t^*(1 + \|x\|^2) + \rho(\|x\|^2 + \|x\|^4) \geq 0,$$

or

$$\frac{\|Ax - b\|^2}{1 + \|x\|^2} + \rho \|x\|^2 \geq t^*.$$

This further implies that

$$\min_{x \in R^n} \{\frac{\|Ax - b\|^2}{1 + \|x\|^2} + \rho \|x\|^2\} \geq t^*,$$

but at least we know that equality holds when $x = x^*$. Thus x^* is the global minimum of (7). □

Now, to find the root of function $\Phi(t)$ we utilize the bisection algorithm to reduce the initial interval $[0, \|b\|^2]$ and also the classical Newton method to solve the corresponding minimization problem. Similar to the fractional case, here also we do not know whether the objective function is convex or not. But this simple procedure leads us to the global minimum for all test problems.

However, as we are aware, the bisection method is usually too slow, so the third approach which we consider is as follows. First we perform a few iterations of bisection algorithm, then crossover to formulation (7) rather than finding the root of function Φ. This combined algorithm finds the global solution much faster compared to both previous algorithms. In the next section we report extensive numerical testing which demonstrates the practical performance of the three presented algorithms.

3 Computational experiments

Test problems in Tables 1 and 2 are taken from [5] which contains ill-posed linear systems arising from certain integral equations, and problems on Table 3 are taken from University of Florida sparse matrix collection. The implementation of the algorithms are done in MATLAB 7.4 on a pentium M 1.7GHz laptop with 1 GB of memory. All test problems are square, however they can easily be made overdetermined by repeating some constraints with slightly different right hand side and still the same observations, which will be given in the sequel, hold.

For all test problems the coefficient matrices are either singular or very close to singularity. Moreover, for problems in Tables 1 and 2 we have the exact solution and for problems in Table 3 we consider all one vector as the exact solution. Furthermore the noisy system is generated by perturbing A and b by adding '$1e-3*randn(size(A))$' and '$1e-3*randn(size(b))$' respectively. Since the coefficient matrix is singular or very close to singularity, then either system $Ax=b$ is infeasible or its solution might have very large norm. Therefore the total least squares approach is utilized to find an appropriate feasible system with a meaningful solution.

In all tables x_s and x^* denote the exact and computed solution of problems, respectively and $\|Ax^*-b\|$ denotes the violation of the computed solution from the original system. The numbers in all parenthesis are for the classical Newton method, bisection method, and bisection-Newton method (crossover) respectively. For all test problems we have

Table 1. **Comparison of Newton, bisection-Newton and crossover algorithms**

problem	ρ	$\|Ax^* - b\|$	$\|x^*\|$	$\|x_s\|$	time
baart-100	0.001	(70.155, 0.0093, 0.0093)	(20.844, 1.2016, 1.2016)	1.2533	(0.65,0.8,0.45)
	0.1	(0.1690, 0.1690, 0.1690)	(1.0143, 1.0143, 1.0143)	1.2533	(0.5,0.9,0.5)
	1	(0.5223, 0.5223, 0.5223)	(0.7929, 0.7929, 0.7929)	1.2533	(0.18,0.95,0.55)
	10	(1.4873, 1.4873, 1.4873)	(0.4485, 0.4485, 0.4485)	1.2533	(0.16,1.2,0.65)
baart-500	0.001	(70.162, 0.0084, 0.0084)	(20.846, 1.2036, 1.2036)	1.2533	(13,18.5,9.6)
	0.1	(0.1678, 0.1678, 0.1678)	(1.0154, 1.0154, 1.0154)	1.2533	(8.9,21.2,13.4)
	1	(0.5216, 0.5216, 0.5216)	(0.7932, 0.7932, 0.7932)	1.2533	(3.2,21.5,14)
	10	(1.4869, 1.4869, 1.4869)	(0.4485, 0.4485, 0.4485)	1.2533	(3.2,23.6,9.6)
baart-1000	0.001	(70.1654, 0.0075, 0.0075)	(20.8443, 1.2068, 1.2068)	1.2533	(229,131,62)
	0.1	(0.167, 0.167, 0.167)	(1.0161, 1.0161, 1.0161)	1.2533	(25.7,153.2,69.8)
	1	(0.5212, 0.5212, 0.5212)	(0.7937, 0.7933, 0.7933)	1.2533	(22.4,155.4,72.5)
	10	(1.4867, 1.4867, 1.4867)	(0.4486, 0.4486, 0.4486)	1.2533	(19,163,77)
heat-100	0.001	(0.7048, 0.047, 0.047)	(0.7279, 1.739, 1.739)	2.4623	(0.13,0.9,0.3)
	0.1	(0.2636, 0.2636, 0.2636)	(0.6915, 0.6915, 0.6915)	2.4623	(0.1,0.27,0.12)
	1	(0.42081, 0.42081, 0.42081)	(0.1519, 0.1519, 0.1519)	2.4623	(0.1,0.26,0.1)
	10	(0.46329, 0.46329, 0.46329)	(0.0148, 0.0148, 0.0148)	2.4623	(0.1,0.25,0.12)
heat-500	0.001	(3.6139, 0.1831, 0.1831)	(7.5382, 3.3458, 3.3458)	5.5034	(5.1,20.4,10)
	0.1	(0.6448, 0.6448, 0.6448)	(1.3441, 1.3441, 1.3441)	5.5034	(3.9,23.3,9)
	1	0.8974, 0.8974, 0.8974)	(0.4768, 0.4768, 0.4768)	5.5034	(3.5,23,11)
	10	(1.0335, 1.0335, 1.0335)	(0.03618, 0.03618, 0.03618)	5.5034	(3.2,21,10)
heat-1000	0.001	(1.5452, 0.3286, 0.3286)	(2.1473, 4.3463, 4.3463)	7.7829	(30,151.5,63)
	0.1	(0.9633, 0.9633, 0.9633)	(1.7141, 1.7141, 1.7141)	7.7829	(34,181.5,63.5)
	1	(1.2513, 1.2513, 1.2513)	(0.7321, 0.7321, 0.7321)	7.7829	(28.2,165.8,65.2)
	10	(1.459, 1.459, 1.459)	(0.0578, 0.0578, 0.0578)	7.7829	(22.7,157,77)

Table 2. **Comparison of Newton, bisection-Newton and crossover algorithms**

problem	ρ	$\|Ax^* - b\|$	$\|x^*\|$	$\|x_s\|$	time
shaw-100	0.001	(0.5862, 0.5862, 0.5862)	(9.3802, 9.3802, 9.3802)	9.982	(0.18,0.28,0.17)
	0.1	(6.7763, 6.7763, 6.7763)	(6.093, 6.093, 6.093)	9.982	(0.1,0.3,0.15)
	1	(12.128, 12.128, 12.128)	(3.985, 3.985, 3.985)	9.982	(0.1,0.33,0.18)
	10	(16.704, 16.704, 16.704)	(2.3207, 2.3207, 2.3207)	9.982	(0.1,0.35,0.15)
shaw-500	0.001	(224.4, 3.9035, 3.9035)	(58.085, 19.253, 19.253)	22.32	(6,25.9,14.2)
	0.1	(23.265, 23.265, 23.265)	(10.37, 10.37, 10.37)	22.32	(2.7,27.6,17.3)
	1	(33.95, 33.95, 33.95)	(6.409, 6.409, 6.409)	22.32	(3.2,27.8,18.7)
	10	(41.504, 41.504, 41.504)	(3.715, 3.715, 3.715)	22.32	(3.6,31,0.14.2)
shaw-1000	0.001	(270.21, 8.1311, 8.1311)	(66.423, 25.695, 25.695)	31.566	(58.3,172.1,95.5)
	0.1	(37.78, 37.78, 37.78)	(12.82, 12.82, 12.82)	31.566	(22.5,193,118.6)
	1	(51.57, 51.57, 51.57)	(7.7873, 7.7873, 7.7873)	31.566	(25.3,199.5,125.5)
	10	(60.833, 60.833, 60833)	(4.5, 4.5, 4.5)	31.566	(28.60,207.6,101.7)
spike-100	0.001	(1.2306, 1.2306, 1.2306)	(14.121, 14.121, 14.121)	29.017	(0.2,0.3,0.15)
	0.1	(17.211, 17.211, 17.211)	(11.664, 11.664, 11.664)	29.017	(0.1,0.33,0.2)
	1	(43.986, 43.986, 43.986)	(8.4666, 8.4666, 8.4666)	29.017	(0.1,0.34,0.23)
	10	(72.392, 72.392, 72.392)	(5.4189, 5.4189, 5.4189)	29.017	(0.1,0.39,0.23)
spike-500	0.001	(1.3441, 1.3441, 1.3441)	(22.706, 22.706, 22.706)	34.67	(17,25.9,14.8)
	0.1	(33.24, 33.24, 33.24)	(21.207, 21.207, 21.207)	34.670	(6.3,32,19)
	1	(138.54, 138.54, 138.54)	(17.723, 17.723, 17.723)	34.670	(9.1,29.5,20.5)
	10	(319.12, 319.12, 319.12)	(12.607, 12.607, 12.607)	34.670	(4.4,29.3,18.4)
spike-1000	0.001	(1.9066, 1.9066)	(30.877, 30.877, 30.877)	40.645	(580,210,124)
	0.1	(51.536, 51.632, 51.536)	(29.536, 29.536, 29.536)	40.645	(26,205.9,123.7)
	1	(252.78, 252.78, 252.78)	(25.833, 25.833, 25.833)	40.645	(48.3,215,156.2)
	10	(678.16, 678.16, 678.16)	(19.205, 19.205, 19.205)	40.645	(20,220.2,152)

Table 3. **Comparison of Newton, bisection-Newton and crossover algorithms**

problem	ρ	$\|Ax^* - b\|$	$\|x^*\|$	$\|x_s\|$	time
can187	0.001	(0.4187, 0.4187, 0.4187)	(13.6, 13.6, 13.6)	13.675	(0.31,1.4,0.85)
	0.1	(16.399, 16.399, 16.399)	(11.495, 11.495, 11.495)	13.675	(0.14,0.9,0.5)
	1	(42.695, 42.695, 42.695)	(8.236, 8.236, 8.236)	13.675	(0.15,1,0.7)
	10	(67.983, 67.983, 67.983)	(5.159, 5.159, 5.159)	13.675	(0.12,0.95,0.6)
can268	0.001	(1.1281, 1.1281, 1.1281)	(16.014, 16.014, 16.014)	16.371	(8,2.4,1.4)
	0.1	(19.266, 19.266, 19.266)	(13.777, 13.777, 13.777)	16.371.	(0.35,2.7,1.8)
	1	(61.559, 61.559, 61.559)	(10.6, 10.6, 10.6)	16.371	(0.67,2.2,1.4)
	10	(112.79, 112.79, 112.79)	(6.92, 6.92, 6.92)	16.371	(0.4,2.4,1.5)
can445	0.001	(1.2475, 1.2475, 1.2475)	(20.892, 20.892, 20.892)	21.095	(6.3,7,4.3)
	0.1	(39.62, 39.62, 39.62)	(16.336, 16.336, 16.336)	21.095	(1,6.3,3.7)
	1	(84.947, 84.947, 84.947)	(11.118, 11.118, 11.118)	21.095	(1.2,5.8,3.8)
	10	(122.60, 122.60, 122.60)	(6.8154, 6.8154, 6.8154)	21.095	(1.5,7.5,4.7)
cavity07	0.001	(7.0198, 7.0198, 7.0198)	(22.403, 22.043, 22.403)	34.38	(16.7,103,65)
	0.1	(25.558, 25.558, 22.558)	(10.563, 10.563, 10.563)	34.38	(16.4,105.6,64.9)
	1	(36.13, 36.13, 36.13)	(6.7589, 6.7589, 6.7589)	34.38	(20.7,74.6,51.3)
	10	(47.42, 47.42, 47.42)	(4.1722, 4.1722, 4.1722)	34.38	(20.5,123.8,87.5)
ex21	0.001	(8.3144, 0.09386, 0.09386)	(12.006, 6.7722, 6.7722)	25.612	(7.6,21,11)
	0.1	(2.0441, 2.0441, 2.0441)	(2.619, 2.619, 2.619)	25.612	(5,22.4,11)
	1	(2.4770, 2.4770, 2.4770)	(1.3508, 1.3508, 1.3508)	25.612	(6,18.9,7.5)
	10	(2.8593, 2.8593, 2.8593)	(0.3089, 0.3089, 0.3089)	25.612	(5,24.3,11.9)
dwt869	0.001	(3.3016, 3.3016, 3.3016)	(28.916, 28.916, 28.916)	29.479	(7.6,43.7,27)
	0.1	(72.326, 72.326, 72.326)	(20.557, 20.557, 20.577)	29.479	(6.8,47.3,30.5)
	1	(133.26, 133.26, 133.26)	(13.49, 13.49, 13.49)	29.479	(6.6,28.5,18)
	10	(179.94, 179.94, 179.94)	(8.1496, 8.1496, 8.1496)	29.479	(7.5,51,34.3)
gd01a	0.001	(1.0654, 1.0654, 1.0654)	(12.541, 12.541, 12.541)	30.871	(11,278.4,27.7)
	0.1	(12.323, 12.323, 12.323)	(8.6939, 8.6939, 8.6939)	30.871	(9.4,370,99)
	1	(24.636, 24.636, 24.636)	(6.0131, 6.0131, 6.0131)	30.871	(15.2,265.9,210)
	10	(36.958, 36.958, 36.958)	(3.7135, 3.7135, 3.7135)	30.871	(11.7,420,30.9)
gd00a	0.001	(1.4398, 1.4398, 1.4398)	(12.69, 12.69, 12.69)	18.762	(1.1,10,1.2)
	0.1	(10.985, 10.985, 10.958)	(7.2358, 7.2358, 7.2358)	18.762	(0.8,11.6,2.2)
	1	(16.768, 16.768, 16.768)	(4.5229, 4.5229, 4.5229)	18.762	(1.4,13,7)
	10	(21.304, 21.304, 21.304)	(2.6177, 2.6177, 2.6177)	18.762	(0.9,14,2)

used $10 * ones(n, 1)$ as the starting point[1] with four different values of the ρ parameter. Having prior information of the solution also indeed is suggested to be incorporated as the starting point selection procedure.

As our computational results show, the classical Newton method solves all problems for $\rho = 0.1, 1, 10$ faster than the other two approaches, however it fails for many problems when $\rho = 0.001$. It is worth to note that by changing the starting point to for example $100 * ones(n, 1)$ Newton method solves some of the failed problems. However, the other two approaches successfully solve all problems for all ρ values up to global optimality. Therefore, based on these computational results we may conclude that for smaller ρ values the later two approaches are preferred to Newton algorithm, specially the crossover approach, otherwise the Newton algorithm seems to find the global solution much faster.

4 Conclusions

In this paper, first we have introduced the total least squares problem to deal with approximate feasible linear systems. Then three numerical algorithms are presented to solve the resulting fractional minimization problem. Finally, several numerical examples are presented to demonstrate the practical efficiency of the presented algorithms.

References

[1] A. Beck, A. Ben-Tal. *On the solution of the Tikhonov regularization of the total least squares problem.* SIAM J. On Opt., vol. 17 (2006), pp. 98–118.

[2] W. Dinkelbach. *On nonlinear fractional programming.* Management Sci., vol. 131 (1967), pp. 492–498.

[1]For problem 'spike1000' with $\rho = 0.001$ we could not solve it even by bisection method, however by $100 * ones(n, 1)$ as the starting point the solution is given in Table 2.

[3] H.W. Engl. *Regularization methods for the stable solutions of inverse problems.* Surveys Math. Indust., vol. 3 (1993), pp. 71–143.

[4] G.H. Golub, P.C. Hansen, D.P. O'Leary. *Tikhonov regularization and total least squares.* SIAM J. Matrix Anal. Appli., vol. 21 (1999), pp. 185–194.

[5] P.C. Hansen. *Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems,* Numerical Algorithms, vol. 6 (1994), pp. 1–35.

[6] A.N. Tikhonov. *Solution of incorrectly formulated problems and regularization method.* Soveit Math. Dokl., vol. 4 (1963), pp. 1036–1038.

Maziar Salahi, Hossein Zaremoghaddam,
Received December 22, 2008

Maziar Salahi,
Department of Mathematics,
Faculty of Sciences,
University of Guilan,
Rasht, Namjoo Street, Guilan, P.O.Box 1914
E–mail: *salahim@guilan.ac.ir*

Hossein Zareamoghaddam,
Department of Mathematics,
Faculty of Sciences,
University of Guilan,
Rasht, Namjoo Street, Guilan, P.O.Box 1914
E–mail: *zareamoghaddam@yahoo.com*