Linear Algebra and its Applications 433 (2010) 1101–1109

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Minimal condition number for positive definite Hankel matrices using semidefinite programming

Suliman Al-Homidan [a,1], Mohammad M. Alshahrani [b,1], Cosmin G. Petra [c,2], Florian A. Potra [c,2,*]

[a] *Department of Mathematics & Statistics, King Fahd University of Petroleum and Minerals (KFUPM), P.O. Box 119, Dhahran 31261, Saudi Arabia*
[b] *Department of Mathematics & Statistics, King Fahd University of Petroleum and Minerals (KFUPM), P.O. Box 1258, Dhahran 31261, Saudi Arabia*
[c] *Department of Mathematics & Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA*

ARTICLE INFO

Article history:
Received 27 January 2009
Accepted 15 April 2010
Available online 26 June 2010

Submitted by V. Mehrmann

Keywords:
Hankel matrix
Minimum condition number
Semidefinite programming

ABSTRACT

We present a semidefinite programming approach for computing optimally conditioned positive definite Hankel matrices of order n. Unlike previous approaches, our method is guaranteed to find an optimally conditioned positive definite Hankel matrix within any desired tolerance. Since the condition number of such matrices grows exponentially with n, this is a very good test problem for checking the numerical accuracy of semidefinite programming solvers. Our tests show that semidefinite programming solvers using fixed double precision arithmetic are not able to solve problems with $n > 30$. Moreover, the accuracy of the results for $24 \leqslant n \leqslant 30$ is questionable. In order to accurately compute minimal condition number positive definite Hankel matrices of higher order, we use a Mathematica 6.0 implementation of the SDPHA solver that performs the numerical calculations in arbitrary precision arithmetic. By using this code, we have validated the results obtained by standard codes for $n \leqslant 24$, and we have found optimally conditioned positive definite Hankel matrices up to $n = 100$.

* Corresponding author. Tel.: +1 410 455 2429; fax: +1 410 455 1066.
E-mail addresses: homidan@kfupm.edu.sa (S. Al-Homidan), mmogib@awalnet.net.sa (M.M. Alshahrani), cpetra1@math.umbc.edu (C.G. Petra), potra@math.umbc.edu (F.A. Potra).

[1] Research supported by King Fahd University of Petroleum and Minerals (KFUPM).
[2] Research supported in part by the National Science Foundation under Grant 0728878.

0024-3795/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2010.04.052

1. Introduction

A Hankel matrix is a square matrix having constant anti-diagonal elements. A general Hankel matrix H can therefore be written as

$$H = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots & \cdots & h_n \\ h_2 & h_3 & h_4 & \cdots & h_n & h_{n+1} \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ h_n & h_{n+1} & \cdots & \cdots & \cdots & h_{2n-1} \end{bmatrix}. \tag{1}$$

Complex Hankel matrices can be well-conditioned, whereas real positive definite Hankel matrices are known to be very ill-conditioned. Indeed, the condition number of a positive definite Hankel matrix is bounded below by $3 \cdot 2^{(n-6)}$ which is very large even for relatively small values of n [15]. Beckermann [7] gave a better lower bound of the form $\gamma^{n-1}/(16n)$, with $\gamma \approx 3.210$. In this paper we obtain, for any given value of $n \leq 100$, a positive definite Hankel matrix with minimum condition number.

Several authors have developed algorithms for the factorization or inversion of Hankel matrices [8,11,16]. In [5], a positive definite Hankel matrix is constructed using the Cholesky factorization. In some application areas, such as digital signal processing and control theory, one has to compute a positive semidefinite Hankel matrix that is closest, in some sense, to a given covariance matrix computed from a data sequence. The problem of preserving the structure while approximating a low rank Hankel matrix, also arises in many applications (see [1–3]). When using interior point methods for solving such problems, it is important for some algorithms to start from within the cone of positive semidefinite Hankel matrices, i.e., the initial point must be a positive definite Hankel matrix [4,6]. Having an optimally conditioned positive definite Hankel matrix as a starting point, would be highly desirable in such applications.

The problem of finding an optimally conditioned Hankel matrix of order n, $\widehat{H}_n$, was analyzed by Varah [16], who obtained some far reaching theoretical results, and presented an iterative numerical algorithm for computing such matrices. The algorithm uses Vandermonde factorizations and bootstrapping techniques. As observed by the author "the algorithm produces a sequence of positive definite Hankel matrices with decreasing condition numbers, which appears to converge ..., although we have no convergence proof" [16, pp. 311]. Numerical results are given for $n \leqslant 16$. From the high precision results obtained by the method proposed in this paper, it follows that Varah's results are correct for $n \leqslant 14$, while the condition numbers of the matrices found by Varah for $n = 15$ and $n = 16$ are slightly higher than the optimal condition numbers ($985{,}970 > \kappa(\widehat{H}_{15}) \approx 985{,}955.14$ and $2{,}890{,}409 > \kappa(\widehat{H}_{16}) \approx 2{,}890{,}366.56$). By using a heuristic technique based on the Cholesky factorization, Alshahrani and Al-Homidan [5] have managed to compute positive definite Hankel matrices with decent, but not optimal, condition number for $n < 30$. They analyze in detail their technique for $n = 20$, where they obtain a positive definite Hankel matrix whose condition number is 7.2776×10^8. By comparison, our high precision method gives $\kappa(\widehat{H}_{16}) \approx 2.71 \times 10^8$.

The high precision method we alluded to in the above paragraph, is based on formulating the problem of finding an optimally conditioned Hankel matrix as a semidefinite programming problem (SDP), and by applying an interior point method implemented in (variable) high precision arithmetic for its numerical solution. Unlike previous approaches, our approach is guaranteed to find $\widehat{H}_n$ within any desired tolerance.

As mentioned above, the condition number of a positive definite Hankel matrix grows exponentially with n. Hence, the problem of computing such a matrix can be used to test the accuracy of SDP codes. In the following sections we state the problem and give detailed results by applying the SDP solvers SDPT3 3.0 [14], SDPA-M 6.2 [10], SeDuMi 1.2 [13] and SDPHA [9] to this problem. The first four solvers are implemented in Matlab (with some modules written in C). Since the condition number grows exponentially with n, and these codes use fixed double precision arithmetic, they are not able to solve problems with $n > 30$. Moreover, the accuracy of the results for $26 \leqslant n \leqslant 30$ is questionable. In order to elucidate these issues, in this paper we present some results obtained by a Mathematica 6.0 implementation of the SDPHA solver that performs the numerical calculations in arbitrary precision

arithmetic. To distinguish this implementation from the original SDPHA, we will refer to it as SDPHAM6, M6 standing for Mathematica 6. In the last section of the paper, we validate the numerical results obtained with the above mentioned Matlab-based solvers, by running the Mathematica version of SDPHA in high precision arithmetic. The code will be also used to find larger positive definite Hankel matrices (up to $n = 100$).

2. The problem

The set of all $n \times n$ real Hankel matrices will be denoted by $\mathcal{H}_n$ and the set of all $n \times n$ symmetric matrices by $\mathcal{S}_n$. Any $n \times n$ real Hankel matrix is uniquely determined by a vector $h \in \mathbb{R}^{2n-1}$, so that it is convenient to denote the matrix H from (1) by $H(h)$. Clearly any real Hankel matrix is symmetric, i.e., $\mathcal{H}_n \subset \mathcal{S}_n$.

The problem of finding a positive definite Hankel matrix with the smallest condition number possible can be written in the following form

$$\begin{aligned} &\min_{h\in\mathbb{R}^{2n-1}} \quad k(H(h)),\\ &\text{subject to } H(h) \succ 0, \end{aligned} \tag{2}$$

where $k(A) = \lambda_{max}(A)/\lambda_{min}(A)$ is the condition number of the matrix A and $H(h) \succ 0$ means that $H(h)$ is positive definite.

Varah showed in [16] that positive definite Hankel matrices of minimal condition can be found among the matrices of the form

$$\overline{H}(h) = \begin{bmatrix} h_1 & 0 & h_2 & 0 & \cdots & & & \\ 0 & h_2 & 0 & \cdots & & & & \\ h_2 & 0 & \cdots & & & & & \\ \vdots & \vdots & & & \ddots & & \vdots & \vdots \\ & & & & & \cdots & h_2 & 0 \\ & & & & \cdots & h_2 & 0 & h_1 \end{bmatrix}, \tag{3}$$

where $h = [h_1 h_2 \ldots h_p]^T \in \mathbb{R}^p$, with

$$p = \begin{cases} n/2, & \text{if } n \text{ is even} \\ (n+1)/2, & \text{if } n \text{ is odd.} \end{cases} \tag{4}$$

Varah's result allows us to reduce the size of problem (2) by a factor of about four. Accordingly, in this paper we will compute $\widehat{H}_n$ by solving the problem

$$\begin{aligned} &\min_{h\in\mathbb{R}^{p}} \quad k(\overline{H}_n(h)),\\ &\text{subject to } \overline{H}_n(h) \succ 0. \end{aligned} \tag{5}$$

3. Semidefinite programming

The semidefinite programming (SDP) problem in primal standard form is:

$$\begin{aligned} &(P)\ \text{minimize}_X\ C \bullet X,\\ &\quad \text{subject to } A_i \bullet X = b_i, i = 1, 2, \ldots, m, X \succeq 0, \end{aligned} \tag{6}$$

where all $A_i, C \in \mathcal{S}_n$, $b \in \mathbb{R}^m$ are given, $X \in \mathcal{S}_n$ is the primal variable, and $C \bullet X = \text{trace}(CX)$. This optimization problem is a convex optimization problem, since its objective function and constraints are convex. The dual problem of (6) is

$$(D)\ \text{maximize}_y\ b^T y,$$

$$\text{subject to } \sum_{i=1}^{m} y_i A_i \preceq C, \tag{7}$$

where $y \in \mathbb{R}^m$ is the vector of dual variables. Problems (6) and (7) include many problems as special cases and have many applications, in particular, (2).

Our problem can be expressed in the form (7) as follows:

$$\begin{aligned} &\text{maximize } \mu, \\ &\text{such that } I \succeq \overline{H}(h) \succeq \mu I, \end{aligned} \tag{8}$$

or, equivalently,

$$\begin{aligned} &\text{maximize } \mu, \\ &\text{such that } \begin{bmatrix} \overline{H}(h) & 0 \\ 0 & \mu I - \overline{H}(h) \end{bmatrix} \preceq \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned} \tag{9}$$

Observe that if the vector $h^* \in \mathbb{R}^p$, and the number $\mu^* \in \mathbb{R}$, are a solution of problem (9), then we must have $\widehat{H}_n = \overline{H}(h^*)$ and $\mu^* = 1/\kappa(\widehat{H}_n)$.

The above problem is an SDP problem in dual form (7) of dimension $p+1$. To see this, we identify

$$\begin{aligned} &b = [1\,0\ldots 0]^T \in \mathbb{R}^{p+1}, \quad y = [\mu\, h_1 \ldots h_p]^T \in \mathbb{R}^{p+1}, \\ &C = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, \\ &A_k = \begin{bmatrix} E_{2k-3} + E_{2n-2k+3} & 0 \\ 0 & -E_{2k-3} - E_{2n-2k+3} \end{bmatrix}, \ k = 2, \ldots, p\,, \\ &A_{p+1} = \begin{cases} \begin{bmatrix} E_n & 0 \\ 0 & -E_n \end{bmatrix}, & \text{if } n \text{ is odd} \\ \begin{bmatrix} E_{n-1} + E_{n+1} & 0 \\ 0 & -E_{n-1} - E_{n+1} \end{bmatrix}, & \text{if } n \text{ is even,} \end{cases} \end{aligned}$$

where the $n \times n$ matrix E_k is defined to be

$$E_k(i,j) = \begin{cases} 1, & \text{if } i+j = k+1, \\ 0, & \text{otherwise,} \end{cases}$$

for any $k = 1, 2, \ldots, 2n-1$.

4. Double precision computations of $\widehat{H}_n$

We have first tried to compute minimal condition number positive definite Hankel matrices of size n by feeding problem (9) to double precision SDP solvers. We have used the SDP solvers SDPT3-4.0, SDPA-M 6.2, SeDuMi 1.05, and the Matlab implementation of SDPHA.

In our formulation, the maximum eigenvalue is forced to be equal to 1, so that the minimum eigenvalue is $\lambda_{min} = \mu^* = 1/\kappa(\widehat{H}_n)$. Since the condition number increases exponentially with n, this poses severe problems for all SDP solvers using double precision arithmetic, even for relatively small values of n. We measure the accuracy of the double precision SDP solvers by the maximum relative error in the computed entries of the optimal Hankel matrices

$$e_n^h = \max\{|h_i^{dbl} - h_i^*|/h_i^* \ : \ i = 1, 2, \ldots, p\},$$

where h^{dbl} is obtained in double precision arithmetic by one of the four SDP solvers mentioned above, and h^* is computed in high precision arithmetic by the method described in Section 5, which is shown to be very accurate. We have also used the relative error

Table 1
Relative errors in double precision computations.

n	e_n^h				e_n^κ			
	SDPA-M	SDPT3	SeDuMi	SDPHA	SDPA-M	SDPT3	SeDuMi	SDPHA
3	8.9e-13	8.7e-10	1.8e-14	1.6e-10	5.9e-13	5.8e-10	1.2e-14	1.0e-10
4	6.4e-7	2.0e-10	6.2e-15	7.2e-13	3.2e-13	5.1e-17	2.5e-16	1.0e-16
5	1.6e-6	3.1e-5	6.1e-9	3.3e-9	2.4e-12	9.7e-10	1.3e-14	1.3e-10
6	7.4e-7	3.7e-5	4.2e-8	4.0e-11	2.2e-13	6.0e-10	2.2e-16	8.7e-16
7	6.5e-6	4.1e-5	2.6e-8	1.5e-8	2.3e-11	3.0e-9	7.3e-14	7.5e-10
8	3.9e-6	1.6e-4	2.4e-8	5.3e-9	1.2e-11	6.8e-9	6.3e-15	3.1e-15
9	5.2e-6	1.1e-4	1.2e-6	3.8e-8	2.6e-11	1.3e-8	7.2e-13	9.0e-10
10	6.5e-6	2.4e-4	4.9e-7	6.5e-9	3.3e-11	1.3e-8	1.3e-13	2.0e-14
11	3.9e-6	2.6e-4	5.5e-6	1.7e-6	1.9e-11	1.6e-7	1.0e-11	5.6e-9
12	3.0e-5	6.8e-4	7.8e-6	2.6e-6	6.5e-10	8.0e-8	1.6e-11	8.2e-13
13	2.1e-5	2.4e-4	9.0e-6	2.8e-6	3.5e-10	4.8e-7	1.2e-11	1.8e-9
14	1.1e-4	1.5e-3	6.6e-5	2.0e-6	3.3e-9	4.5e-7	5.6e-10	2.3e-12
15	3.1e-5	2.1e-1	5.8e-5	1.7e-4	2.5e-9	5.1e-3	1.4e-9	7.9e-8
16	2.9e-4		4.8e-4	4.4e-5	2.2e-8		3.1e-8	2.6e-10
17	1.1e-3		2.6e-4	8.3e-6	1.2e-7		2.6e-8	1.8e-7
18	2.4e-4		2.8e-5	5.1e-3	1.1e-8		3.4e-9	3.1e-6
19	1.6e-1		1.5e-3	4.1e-2	2.2e-3		7.3e-7	1.7e-4
20			2.4e-3	1.2e-1			6.5e-7	1.7e-3
21			8.2e-3				1.1e-5	
22			5.1e-3				2.0e-5	
23			7.9e-2				6.6e-4	
24			1.2e-2				1.8e-4	
25			7.7e-1				4.1e-2	

$$e_n^\kappa = |\kappa(\widehat{H}_n(h^{dbl})) - \kappa(\widehat{H}_n(h^*))|/\kappa(\widehat{H}_n(h^*))$$

to measure the accuracy of the computed condition number. Table 1 lists e_n^h and e_n^κ, as obtained with each SDP solver, starting at $n = 3$ and stopping when $e_n^h \geqslant 0.1$.

As evidenced by Table 1, the computation of optimally conditioned positive definite Hankel matrices in double precision arithmetic is accurate only for a limited range of n. SeDuMi was able to find the largest optimal Hankel matrices within a relative error of 0.1. However this accuracy is far from being satisfactory, and high precision arithmetic is needed to obtain accurate Hankel matrices of larger size.

The loss of accuracy of double precision computations was to be expected since, as we have pointed out, the condition number of the optimal Hankel matrices grows exponentially with n. We mention that all four SDP solvers detected this problem, and gave warnings of numerical difficulties and possible inexact results, SDPT3 starting with $n = 9$, the others starting around $n = 12$. Hence, the problem of finding optimally conditioned positive definite Hankel matrices, is indeed a good test for the accuracy of positive semidefinite solvers.

5. High precision computations with the Mathematica 6 version of SDPHA

Matlab uses double precision arithmetic, which means that the numerical values have up to about 15 digits of accuracy. This narrows the applicability of double precision codes to our model for larger values of n. On the other hand, Mathematica offers the possibility to conduct the numerical calculations in arbitrary precision arithmetic, the only limitations being the available memory of the host computer, and the execution time. Arbitrary precision arithmetic, being implemented in software, is usually much slower than double precision arithmetic, which is implemented in hardware. Within these limitations, we have managed to implement SDPHA in Mathematica 6.0, and to take advantage of the Mathematica's built-in arbitrary precision arithmetic in order to accurately compute optimally conditioned positive

Table 2
Relative distance d_{rel} between objective function μ and $\kappa(\widehat{H}_n)$.

n	d_{rel}	n	d_{rel}	n	d_{rel}	n	d_{rel}	n	d_{rel}
1	7.1e-22	21	8.4e-14	41	8.7e-25	61	1.0e-34	81	5.2e-44
2	5.4e-21	22	1.6e-14	42	4.4e-24	62	4.3e-35	82	8.2e-45
3	1.4e-20	23	7.8e-16	43	2.8e-25	63	6.0e-37	83	2.3e-46
4	1.6e-18	24	2.3e-14	44	9.6e-26	64	4.2e-36	84	1.2e-45
5	2.5e-19	25	4.6e-15	45	3.6e-27	65	7.6e-37	85	5.2e-47
6	5.1e-18	26	6.6e-16	46	7.7e-26	66	2.1e-37	86	5.8e-47
7	2.2e-19	27	1.6e-19	47	4.5e-28	67	7.6e-37	87	1.4e-46
8	1.6e-18	28	3.3e-18	48	1.0e-27	68	4.4e-38	88	3.4e-48
9	1.2e-17	29	7.6e-19	49	1.0e-28	69	9.2e-39	89	1.6e-48
10	9.0e-17	30	3.9e-18	50	3.5e-27	70	2.2e-38	90	1.7e-48
11	4.8e-19	31	6.2e-19	51	4.9e-30	71	5.5e-40	91	2.1e-50
12	4.4e-17	32	3.5e-19	52	4.9e-29	72	7.4e-39	92	2.8e-49
13	5.8e-17	33	5.5e-21	53	1.6e-29	73	9.4e-41	93	1.1e-50
14	1.0e-14	34	1.5e-19	54	8.0e-31	74	3.2e-39	94	3.5e-51
15	2.1e-16	35	6.7e-23	55	7.0e-31	75	1.1e-42	95	2.2e-51
16	1.7e-13	36	1.0e-21	56	1.0e-31	76	3.6e-41	96	5.9e-51
17	4.9e-15	37	6.1e-23	57	1.6e-31	77	4.4e-43	97	1.2e-52
18	1.8e-13	38	9.6e-22	58	6.3e-32	78	7.9e-42	98	9.2e-52
19	9.8e-12	39	6.5e-24	59	1.6e-33	79	2.4e-44	99	1.6e-52
20	1.1e-11	40	1.7e-23	60	2.5e-33	80	2.9e-43	100	6.6e-54

definite Hankel matrices for $n \leqslant 100$. We mention that SDPHA uses the homogeneous formulation from [12].

Since larger values of n require more accurate computer arithmetic, SDPHAM6 gradually increases the number of precision digits, and refines the stopping criteria with n. We observed, from the computations for $n = 2, 3, \ldots, 20$, that the objective function of the dual problem is of order $10^{-(n-2)/2}$. Our algorithm stops when the duality gap and the infeasibility measure are smaller than 10^{-n}. If the duality gap falls below this threshold value, then the objective functions of the primal and dual problems have at least n common digits, or, by taking in account their size, $n - (n-2)/2 = (n+2)/2$ common significant digits. Since small numbers, such as 10^{-n}, come into the picture, we have decided to use $2n$ digit precision arithmetic in Mathematica. For example, for $n = 50$, the objective function is expected to be order of 10^{-24} (in fact is 3.81183×10^{-24}). In this case, our algorithm uses 100 precision digits and stops when both the duality gap and the infeasibility measure are less than 10^{-50}. We applied the above rule of thumb only for $n \geqslant 20$. For $n < 20$ we have used 40 exact digits, and the algorithm stopped when both the duality gap and the infeasibility measure were less than 10^{-20}.

As shown in Section 3, the computed value of μ must theoretically be equal to the minimum eigenvalue λ_{min} of the computed Hankel matrix. Any discrepancy between μ and λ_{min}, shows loss of accuracy in numerical computations. Therefore, we have used, as a measure of accuracy, the relative distance between μ and λ_{min},

$$d_{rel} = \frac{|\mu - \lambda_{min}|}{\mu}.$$

The values of d_{rel}, for n as large as 100, are presented in Table 2. The small values of d_{rel}, show that our computations are accurate. It can also be seen that d_{rel} is close to $10^{-(n+2)/2}$, for $n = 20, 21, \ldots, 100$, showing that the expected number of significant digits is actually obtained. The decrease of d_{rel} with n, from the first column, is normal, since SDPHAM6 uses the same arithmetic for all $n < 20$.

The optimal condition numbers $\kappa(\widehat{H}_2), \ldots, \kappa(\widehat{H}_{100})$ are presented in Table 3. Because of space limitation we have listed only 3 significant digits. The full precision values, together with the software used to compute them, can be found at http://www.math.umbc.edu/~cpetra1/hankel.html.

Fig. 1 plots the logarithm of the condition number $\kappa(n)$, and the logarithm of Beckermann's lower and upper bounds,

Table 3
Condition number of optimal Hankel matrices for $n = 2, 3, \ldots, 100$.

n	$\kappa(\widehat{H}_n)$	n	$\kappa(\widehat{H}_n)$	n	$\kappa(\widehat{H}_n)$	n	$\kappa(\widehat{H}_n)$
		2	1.00e+0	3	3.00e+0	4	5.82e+0
5	1.87e+1	6	4.49e+1	7	1.47e+2	8	3.84e+2
9	1.26e+3	10	3.45e+3	11	1.13e+4	12	3.20e+4
13	1.04e+5	14	3.02e+5	15	9.85e+5	16	2.89e+6
17	9.41e+6	18	2.79e+7	19	9.07e+7	20	2.71e+8
21	8.82e+8	22	2.65e+9	23	8.62e+9	24	2.61e+10
25	8.47e+10	26	2.57e+11	27	8.36e+11	28	2.55e+12
29	8.27e+12	30	2.53e+13	31	8.21e+13	32	2.52e+14
33	8.18e+14	34	2.52e+15	35	8.16e+15	36	2.52e+16
37	8.15e+16	38	2.52e+17	39	8.17e+17	40	2.53e+18
41	8.19e+18	42	2.54e+19	43	8.22e+19	44	2.56e+20
45	8.27e+20	46	2.57e+21	47	8.32e+21	48	2.60e+22
49	8.39e+22	50	2.62e+23	51	8.46e+23	52	2.64e+24
53	8.54e+24	54	2.67e+25	55	8.63e+25	56	2.70e+26
57	8.73e+26	58	2.73e+27	59	8.83e+27	60	2.77e+28
61	8.94e+28	62	2.81e+29	63	9.06e+29	64	2.84e+30
65	9.18e+30	66	2.88e+31	67	9.31e+31	68	2.93e+32
69	9.45e+32	70	2.97e+33	71	9.59e+33	72	3.02e+34
73	9.74e+34	74	3.07e+35	75	9.89e+35	76	3.12e+36
77	1.00e+37	78	3.17e+37	79	1.02e+38	80	3.22e+38
81	1.04e+39	82	3.28e+39	83	1.05e+40	84	3.34e+40
85	1.07e+41	86	3.40e+41	87	1.09e+42	88	3.46e+42
89	1.11e+43	90	3.53e+43	91	1.13e+44	92	3.59e+44
93	1.15e+45	94	3.66e+45	95	1.18e+46	96	3.73e+46
97	1.20e+47	98	3.81e+47	99	1.22e+48	100	3.88e+48

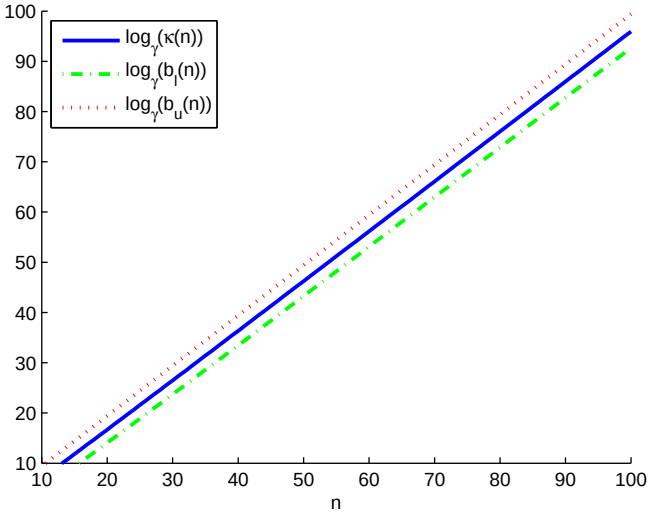

Fig. 1. Logarithm of the optimal condition number and Beckermann's bounds.

$$b_l(n) = \gamma^{n-1}/(16n), \quad b_u(n) = \gamma^n/2,$$

for $n = 3, 4, \ldots, 100$. It is clear, that the numerical values of the condition number increases exponentially with n and stays between the theoretical bounds found by Beckermann.

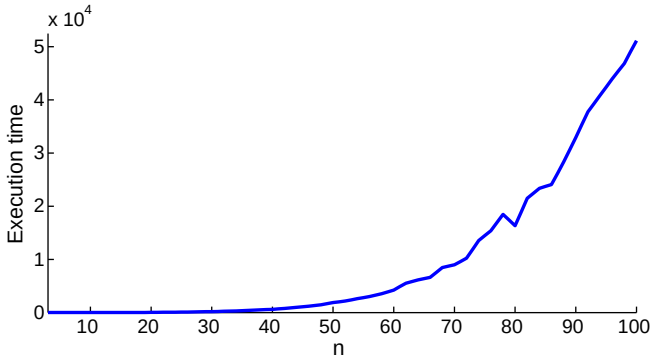

Fig. 2. Execution time in seconds for finding the optimal Hankel matrix of size n.

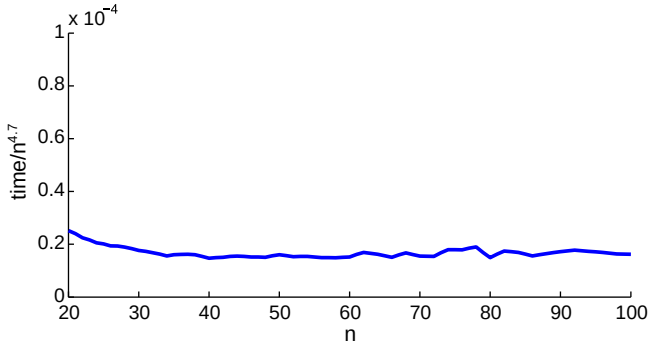

Fig. 3. The ratio between execution time and $n^{4.7}$ remains constant.

Table 4
Entries of $\widehat{H}_{100}$.

$h_1 = 1.000e0$	$h_2 = 2.179e\text{-}1$	$h_3 = 1.042e\text{-}1$	$h_4 = 6.441e\text{-}2$
$h_5 = 4.520e\text{-}2$	$h_6 = 3.422e\text{-}2$	$h_7 = 2.723e\text{-}2$	$h_8 = 2.245e\text{-}2$
$h_9 = 1.900e\text{-}2$	$h_{10} = 1.641e\text{-}2$	$h_{11} = 1.440e\text{-}2$	$h_{12} = 1.281e\text{-}2$
$h_{13} = 1.152e\text{-}2$	$h_{14} = 1.046e\text{-}2$	$h_{15} = 9.576e\text{-}3$	$h_{16} = 8.825e\text{-}3$
$h_{17} = 8.184e\text{-}3$	$h_{18} = 7.630e\text{-}3$	$h_{19} = 7.149e\text{-}3$	$h_{20} = 6.728e\text{-}3$
$h_{21} = 6.356e\text{-}3$	$h_{22} = 6.028e\text{-}3$	$h_{23} = 5.736e\text{-}3$	$h_{24} = 5.475e\text{-}3$
$h_{25} = 5.240e\text{-}3$	$h_{26} = 5.030e\text{-}3$	$h_{27} = 4.840e\text{-}3$	$h_{28} = 4.669e\text{-}3$
$h_{29} = 4.513e\text{-}3$	$h_{30} = 4.372e\text{-}3$	$h_{31} = 4.244e\text{-}3$	$h_{32} = 4.128e\text{-}3$
$h_{33} = 4.023e\text{-}3$	$h_{34} = 3.927e\text{-}3$	$h_{35} = 3.840e\text{-}3$	$h_{36} = 3.761e\text{-}3$
$h_{37} = 3.689e\text{-}3$	$h_{38} = 3.625e\text{-}3$	$h_{39} = 3.567e\text{-}3$	$h_{40} = 3.515e\text{-}3$
$h_{41} = 3.468e\text{-}3$	$h_{42} = 3.428e\text{-}3$	$h_{43} = 3.392e\text{-}3$	$h_{44} = 3.361e\text{-}3$
$h_{45} = 3.335e\text{-}3$	$h_{46} = 3.314e\text{-}3$	$h_{47} = 3.297e\text{-}3$	$h_{48} = 3.284e\text{-}3$
$h_{49} = 3.276e\text{-}3$	$h_{50} = 3.272e\text{-}3$		

The execution time needed to compute an optimal positive definite Hankel matrix, is shown in Fig. 2, for $n = 3, 4, \ldots, 100$. If the computation of $\widehat{H}_{20}$ requires less than 1 min, the computation of $\widehat{H}_{100}$ takes as much as 14.209 h. We are able to provide an estimate for the computational complexity of the method presented in this paper. Since the stopping criteria are more stringent for larger n, the number of interior-point iterations increases with n. Also, each interior-point iteration is more expensive for larger n, not only because the semidefinite programs become larger, but also because we increase the accuracy of the floating point arithmetic, so that a flop is more expensive. Based on the numerical evidence, we found that the overall complexity of calculating $\widehat{H}_n$ is $O(n^{4.7})$, as evidenced by Fig. 3.

We end this paper by listing the entries of $\widehat{H}_{100}$ with 4 significant digits in Table 4. The full precision values, together with the eigenvalues of this matrix, can be found at http://www.math.umbc.edu/~cpetra1/hankel.html.

References

[1] S. Al-Homidan, Combined methods for approximating Hankel matrix, WSEAS Trans. Syst. 1 (1) (2002) 35–41 (Selected papers from the 2nd WSEAS Multiconference and the 3rd WSEAS Symposium (Cairns/Vouliagmeni, 2001)).
[2] S. Al-Homidan, Hybrid methods for approximating Hankel matrix, Numer. Algorithms 32 (1) (2003) 57–66.
[3] S. Al-Homidan, Structured methods for solving Hankel matrix approximation problems, Pacific J. Optim. 1 (3) (2005) 599–609.
[4] S. Al-Homidan, Solving Hankel matrix approximation problem using semidefinite programming, J. Comput. Appl. Math. 202 (2) (2007) 304–314.
[5] S. Al-Homidan, M. Alshahrani, Positive definite Hankel matrices using Cholesky factorization, Comput. Methods Appl. Math. 9 (3) (2009) 221–225.
[6] M. Alshahrani, S. Al-Homidan, Mixed semidefinite and second-order cone optimization approach for the Hankel matrix approximation problem, Nonlinear Dynam. Syst. Theory 6 (3) (2006) 211–224.
[7] B. Beckermann, The condition number of real Vandermonde, Krylov and positive definite Hankel matrices, Numer. Math. 85 (4) (2000) 553–577.
[8] D.L. Boley, F.T. Luk, D. Vandevoorde, Vandermonde factorization of a Hankel matrix, Scientific Computing (Hong Kong, 1997), Springer, Singapore, 1997, pp. 27–39.
[9] N. Brixius, F.A. Potra, R. Sheng, SDPHA: a MATLAB implementation of homogeneous interior-point algorithms for semidefinite programming, Optim. Methods Softw. 11–12 (1999) 583–596.
[10] K. Fujisawa, Y. Futakata, M. Kojima, S. Matsuyama, S. Nakamura, K. Nakata, M. Yamashita, SDPA-M (Semi Definite Programming Algorithm in MATLAB) User's manual — version 6.2.0. Research report B-359, Tokyo Institute of Technology, January 2000 (Revised 2007).
[11] V. Olshevsky, M. Stewart, Stable factorization for Hankel and Hankel-like matrices, Numer. Linear Algebra Appl. 8 (6–7) (2001) 401–434. (Numerical linear algebra techniques for control and signal processing).
[12] F.A. Potra, R. Sheng, On homogeneous interior-point algorithms for semidefinite programming, Optim. Methods Softw. 9 (1998) 161–184.
[13] J.F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optim. Methods Softw. 11–12 (1999) 625–653. (Special issue on Interior Point Methods (CD supplement with software)).
[14] K.C. Toh, M.J. Todd, R. Tutuncu, SDPT3 — a Matlab software package for semidefinite programming, Optim. Methods Softw. 11 (1999) 545–581.
[15] E.E. Tyrtyshnikov, How bad are Hankel matrices? Numer. Math. 67 (2) (1994) 261–269.
[16] J.M. Varah, Positive definite Hankel matrices of minimal condition, Linear Algebra Appl. 368 (2003) 303–314.