AN INFINITELY SUMMABLE SERIES IMPLEMENTATION OF INTERIOR POINT METHODS

Romesh Saigal
Department of Industrial & Operations Engineering
University of Michigan

Technical Report 92-37

May 1992
Revised October 1993

An Infinitely Summable Series implementation of interior point methods

Romesh SAIGAL

Department of Industrial and Operations Engineering,

The University of Michigan,

Ann Arbor, Michigan 48109-2117, USA

May 1992 (Revised October 1993)

Abstract

In this paper we consider an alternative implementation of the interior point methods. In the popular implementations, a variant of sparse Cholesky factorization is intregrated with a congugate gradient type iterative method. In contrast, we set up the problem as an infinitely summable series of vectors. This series is then, iteratively, summed. At each iteration, a procedure based on "least squares" is used to obtain an estimate of the "tail" of the series. In addition, using Chebychev approximations, we show how to accelerate the convergence of this procedure. In this alternative approach, besides finding cholesky factors of some simpler matrices, only multiplications by some matrix are involved. This may be an advantage since in the later part of these methods, the linear system to be solved has a tendency to become ill-conditioned. We present some evidence (for the case of the dual affine scaling method applied to the assignment problem) which supports this conclusion.

Key words: Linear Programming, interior point methods, iterative methods, inifinitely summable series, Chebychev approximations.

Abbreviated title: Implementation of interior point methods.

1 Introduction

In this paper we consider the following linear programming problem:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

where $\boldsymbol{A}$ is a $m \times n$ matrix of full rank m; $\boldsymbol{b}$ is a m vector and $\boldsymbol{c}$ is a n vector, and consider the recent interior point methods, a study of which was started by the seminal work of Karmarkar [4]. Most of the material on these methods can be found in the following references Barnes [1], Kojima, Mizuno and Yoshise [5], Vanderbei, Meketon and Freedman [10] and the recent book Fang and Puthenpura [2]. We leave the reader to browse through these and many other works for an introduction. Our starting point is the implementation of these methods, which requires, at each iteration, the solution of a $m \times m$ linear system of equations

$$\boldsymbol{C}\boldsymbol{z} = \boldsymbol{d} \tag{1}$$

where $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T$ for some $n \times n$ diagonal matrix $\boldsymbol{D}$ which has positive diagonal elements, and $\boldsymbol{d}$ is some vector generated by the matrices $\boldsymbol{A}$, $\boldsymbol{D}$ and vectors $\boldsymbol{b}$, $\boldsymbol{c}$, the exact form depending on the particular interior point method being implemented. The popular implementations to date all use some version of the sparse Cholesky techniques, George and Liu [3], integrated with the congugate gradient method. The goal of this paper is to present another implementation methodology for solving (1).

The first approach for solving (1) as an infinite series was presented by Puthenpura, Saigal and Sinha [7] (this can also be found in Fang and Puthenpura [2]), where the solution to (1) is generated as a sum of an infinite series. In section 2 we give some details of this proposal. Another recent approach for implementation of large sparse or structured problems is presented by Saigal [9]. Here the rows of $\boldsymbol{A}$ are partitioned into two parts, with each part treated separately. In section 3 we give some details of this proposal. Our approach here builds on both these approaches, and the problem is reduced to solving the system

$$(\boldsymbol{I} - \boldsymbol{N})\boldsymbol{y} = \boldsymbol{q} \tag{2}$$

where $\boldsymbol{N}$ is an $r \times r$ symmetric positive semi-definite matrix, with spectral radius less than 1. The variation in this implementation is to solve (2) by the "power expansion"

$$(\boldsymbol{I} - \boldsymbol{N})^{-1} = \boldsymbol{I} + \boldsymbol{N} + \boldsymbol{N}^2 + \cdots \tag{3}$$

and then write the solution to (2) as the infinite sum:

$$\boldsymbol{y} = \boldsymbol{q} + \boldsymbol{N}\boldsymbol{q} + \boldsymbol{N}^2\boldsymbol{q} + \cdots \tag{4}$$

An apparant advantage of (4) is that its implementation involves only multiplications by $\boldsymbol{N}$. But, as we shall see later, in applications to interior point methods, the spectral radius of $\boldsymbol{N}$ is close to 1 (and gets closer to 1 as the algorithm proceeds to the solution). Thus the convergence rate of the sequence $\{\boldsymbol{s}^k\}$ could be very slow, where

$$\begin{aligned} \boldsymbol{s}^k &= \boldsymbol{q} + \boldsymbol{N}\boldsymbol{q} + \cdots + \boldsymbol{N}^k\boldsymbol{q} \\ &= \boldsymbol{q} + \boldsymbol{N}\boldsymbol{s}^{k-1}. \end{aligned}$$

As is easily seen the convergence rate of $\boldsymbol{s}^k$ is bounded by the spectral radius of $\boldsymbol{N}$; i.e., $\| \boldsymbol{s}^k - \boldsymbol{s}^\infty \| \leq \rho(\boldsymbol{N}) \| \boldsymbol{s}^{k-1} - \boldsymbol{s}^\infty \|$ where $\boldsymbol{s}^\infty$ is the limit of the sequence $\{\boldsymbol{s}^k\}$ and $\rho(\boldsymbol{N})$ is the spectral radius of $\boldsymbol{N}$, in our case, the magnitude of the largest eigenvalue (see, for example, [7]).

In this paper we present two methodologies for summing (4). The first approach is to generate $\boldsymbol{s}^k$, and an estimate $\hat{\boldsymbol{t}}^k$ of the tail

$$\boldsymbol{t}^k = \boldsymbol{N}^{k+1}\boldsymbol{q} + \boldsymbol{N}^{k+2}\boldsymbol{q} + \cdots$$

and to declare $\hat{\boldsymbol{y}}^k = \boldsymbol{s}^k + \hat{\boldsymbol{t}}^k$ as an estimate of the sum (4). This estimate is generated by a linear model, which is solved using "least squares". This model establishes a linear relationship between the coefficients of a polynomial whose roots contain certain eigenvalues of $\boldsymbol{N}$. It is based on the following: given the number of "similar classes" of eigenvalues of $\boldsymbol{N}$ (i.e, number of classes of eigenvalues behaving similarily when raised to increasing powers) the model estimates the coefficients of a polynomial which includes each class within its set of roots. One class easy to identify is of small eigenvalues. These all go to zero rapidly as they are raised to higher powers, and also contribute little to the tail. Our model distinguishes

these classes of eigenvalues by functions $(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^k$, of $\boldsymbol{N}$, for some integers l and k. The series

$$(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^k\boldsymbol{q},(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^{k+1}\boldsymbol{q},\cdots$$

is then used in the estimation and is generated by finite-differencing of the original series

$$\boldsymbol{q},\boldsymbol{N}\boldsymbol{q},\cdots,\boldsymbol{N}^{l+k}\boldsymbol{q},\boldsymbol{N}^{l+k+1}\boldsymbol{q},\cdots.$$

This model has the interesting property that the residual vector of the least squares problem is the same as the error vector when the estimate $\hat{\boldsymbol{y}}^k$ is used in place of the "true" solution $\boldsymbol{y}$ in (2). Using the formulation (2) for the assignment problem (presented by Saigal [9]), we show here some limited computational experience which compares a model with two classes to the one with at most 14 classes. This experience suggests that the model, for this problem, will perform well.

The second approach, which accelerates the convergence of the one presented above, uses $\boldsymbol{N}^r\boldsymbol{q}$, $r=0,1,\cdots,k'$, to generate, $\hat{\boldsymbol{s}}^{k'}$, a lower order Chebychev polynomial approximation to $\boldsymbol{s}^k$, $k>k'$; and then uses $\hat{\boldsymbol{y}}^k=\hat{\boldsymbol{s}}^{k'}+\hat{\boldsymbol{t}}^k$ as an estimate of the sum (4), where $\hat{\boldsymbol{t}}^k$ is the estimate of the tail defined earlier. Here, the series,

$$p(\boldsymbol{N})=\boldsymbol{I}+\boldsymbol{N}+\boldsymbol{N}^2+\cdots+\boldsymbol{N}^k$$

is approximated by

$$q(\boldsymbol{N})=A'_1+A'_2\boldsymbol{N}+\cdots+A'_{k'}\boldsymbol{N}^{k'}$$

and the approximation $\hat{\boldsymbol{s}}^{k'}=q(\boldsymbol{N})\boldsymbol{q}$ of $\boldsymbol{s}^k$ is added to the estimate $\hat{\boldsymbol{t}}^k$ to obtain a better estimate of the solution. A'_j are constants determined so that q is a Chebychev polynomial approximation to p.

After presenting, in sections 2 and 3, two formulations that convert equation (1) into the form (2), we develop the basic model in section 4. In section 5, we show how the estimated parameters are used to generate an estimate of the tail, and an estimate of the solution. In section 6 we present a method to modify the eigenvalue structure, while in section 7 we present an acceleration scheme based on Chebychev polynomials. Finally, in section 8, we present some computational justification for this model when used in the context of the assignment problem.

2 LQ Factorization

In this section we show how the system (1) can be transformed into (2) with an appropriate matrix $\boldsymbol{N}$. We use the methodology of Puthenpura, Saigal and Sinha [7] to show this transformation.

For this purpose we will assume that $\boldsymbol{A}$ is an orthonormal matrix, i.e., $\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{I}$. We can assume this because of the following standard theorem, and the implied transformation of the linear program that follows the statement of the theorem.

Theorem 1 *Let $\boldsymbol{A}$ be a $m \times n$ matrix of rank m. There exists a unique lower triangular matrix $\boldsymbol{L}$ and a unique orthonormal matrix $\boldsymbol{Q}$ such that $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{Q}$. The uniqueness is upto a choice of the signs of columns of $\boldsymbol{L}$ and the corresponding rows of $\boldsymbol{Q}$.*

Proof: See, for example, Wilkinson [11]. ■

A consequence of this theorem is that, without loss of generality, we may assume the matrix $\boldsymbol{A}$ is orthonormal. This can be guaranteed by the linear transformation

$$\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{L}^{-1}\boldsymbol{b}$$

of the system in the linear program. In this case then, $\boldsymbol{Q} = \boldsymbol{L}^{-1}\boldsymbol{A}$ is the required matrix. We note here that the matrix $\boldsymbol{Q}$ is dense, even when $\boldsymbol{A}$ and $\boldsymbol{L}$ may be sparse. In [7], it is shown how all the iterations can be carried out without explicitly requiring $\boldsymbol{Q}$.

Now consider $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T$, where $\boldsymbol{A}$ is orthonormal, and let $\alpha > 0$ be a large positive scalar. Then

$$\boldsymbol{C} = \alpha(\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{I} - \frac{1}{\alpha}\boldsymbol{D})\boldsymbol{A}^T)$$

has the required property provided $\alpha > \max_i D_{ii}$. In this case then

$$\begin{aligned} \boldsymbol{N} &= \boldsymbol{A}(\boldsymbol{I} - \frac{1}{\alpha}\boldsymbol{D})\boldsymbol{A}^T \\ \boldsymbol{q} &= \frac{1}{\alpha}\boldsymbol{d}. \end{aligned}$$

The following theorem appears in Puthenpura, Saigal and Sinha [7].

Theorem 2 *For an orthonormal matrix $\boldsymbol{A}$, $\|\boldsymbol{A}\|_2 \leq 1$, $\|\boldsymbol{A}^T\|_2 = 1$ and thus, for $\alpha > max_i D_{ii}$, $\boldsymbol{N}$ defined above has spectral radius less than 1. Here $\|\boldsymbol{A}\|_2 = max_{\|\boldsymbol{x}\|=1}\|\boldsymbol{A}\boldsymbol{x}\|_2$.*

3 Matrix Partitioning and Special Structures

In this section we present another example in which system (1) can be transformed into system (2). This derivation is due to Saigal [9].

Assume that the rows of $\boldsymbol{A}$ have the partition:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{F} \end{bmatrix} \quad , \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{f} \end{bmatrix} \tag{5}$$

where $\boldsymbol{E}$ and $\boldsymbol{F}$ can be readily identified. Examples where this happens include the transportation problem, the generalized upper bounded problem, and the block-diagonal linear programming problem. See, for example, Lasdon [6] for more on these special linear programs. This partition could also be generated to preserve sparsity of the Cholesky factors. The following theorem is from Saigal [9]:

Theorem 3 *There exist a lower triangular matrix $\boldsymbol{L}_1$, a matrix $\boldsymbol{L}_2$ and $\bar{\boldsymbol{D}} = \boldsymbol{F}\boldsymbol{D}\boldsymbol{F}^T - \boldsymbol{L}_2\boldsymbol{L}_2^T$, such that*

$$\boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T = \boldsymbol{L}\hat{\boldsymbol{D}}\boldsymbol{L}^T$$

$$\boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^T = \boldsymbol{L}_1\boldsymbol{L}_1^T$$

$$\boldsymbol{L}_1\boldsymbol{L}_2^T = \boldsymbol{E}\boldsymbol{D}\boldsymbol{F}^T$$

with

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_1 & \boldsymbol{0} \\ \boldsymbol{L}_2 & \boldsymbol{I} \end{bmatrix} \quad , \quad \hat{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{D}} \end{bmatrix}.$$

We now show how Theorem 3 can be used to reduce (1) into the appropriate form. Partitioning

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}^1 \\ \boldsymbol{z}^2 \end{bmatrix} \quad , \quad \boldsymbol{d} = \begin{bmatrix} \boldsymbol{d}^1 \\ \boldsymbol{d}^2 \end{bmatrix}$$

to correspond to the partition of $\boldsymbol{A}$, the following can be easily proved:

Theorem 4 *Equation(2) can be readily solved by the following steps:*

$$\boldsymbol{L}_1\bar{\boldsymbol{z}}^1 \;=\; \boldsymbol{d}^1 \tag{6}$$

$$\bar{D}z^2 = d^2 - L_2\bar{z}^1 \tag{7}$$

$$L_1^T z^1 = \bar{z}^1 - L_2^T z^2 \tag{8}$$

Proof: See Saigal [9]. ■

Note that (6) and (8) are triangular systems, while (7) has the form

$$(FDF^T - L_2L_2^T)z^2 = d^2 - L_2\bar{z}^1 \tag{9}$$

To reduce equation (7) into the form (2), define a lower triangular matrix L_3 such that

$$FDF^T = L_3L_3^T$$

and the system (9) becomes

$$(I - L_3^{-1}L_2L_2^TL_3^{-T})L_3^Tz^2 = L_3^{-1}(d^2 - L_2\bar{z}^1). \tag{10}$$

Letting $N = L_3^{-1}L_2L_2^TL_3^{-T}$, the following is proved in Saigal [9].

Theorem 5 *N is a symmetric positive semi-definite matrix, with spectral radius less than 1.*

An advantage in partitioning A into two parts is that the Cholesky factorization $EDE^T = L_1L_1^T$ and $FDF^T = L_3L_3^T$ can be found separately, and can thus be very sparse and structured. The price paid is in solving (10). For the assignment problem, both EDE^T and FDF^T are diagonal matrices, and the complexity of solving this problem is reduced to that of solving (10).

4 The Model

In this section we develop a model that will generate information useful for estimating the tail, t^k. For example, if the eigenvalues and eigenvectors of N are known, the estimate of the tail can be readily obtained by using the results of the corollary to the standard theorem that appears below:

Theorem 6 *For every $r \times r$ symmetric and positive semi-definite matrix $\boldsymbol{N}$, there exist an orthonormal matrix $\boldsymbol{P}$, and a diagonal matrix $\boldsymbol{\Lambda}$, with non-negative diagonal elements, such that $\boldsymbol{N} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T$. The diagonal entries of $\boldsymbol{\Lambda}$ are the eigenvalues of $\boldsymbol{N}$.*

Proof: See Wilkinson [11]. ■

Since $\boldsymbol{P}$ in Theorem 6 is orthonormal, the following corollary readily follows:

Corollary 7 *For some matrix $\bar{\boldsymbol{P}}$, and all $k = 1, 2, \cdots$, $\boldsymbol{N}^k\boldsymbol{q} = \bar{\boldsymbol{P}}\boldsymbol{\Lambda}^k\boldsymbol{e}$, where $\boldsymbol{e} = (1, 1, \cdots, 1)^T$, is the r-vector of all ones.*

Proof: Using Theorem 6, $\boldsymbol{N}\boldsymbol{q} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T\boldsymbol{q}$. Let $\bar{\boldsymbol{q}} = \boldsymbol{P}^T\boldsymbol{q}$ and $\bar{\boldsymbol{P}}_j = \bar{q}_j\boldsymbol{P}_j$, where $\bar{\boldsymbol{P}}_j$ is the j^{th} column of $\bar{\boldsymbol{P}}$. Since $\boldsymbol{P}$ is orthonormal, it is readily confirmed that $\boldsymbol{N}^k\boldsymbol{q} = \boldsymbol{P}\boldsymbol{\Lambda}^k\boldsymbol{P}^T\boldsymbol{q}$, and the result follows. ■

Using the result of the Corollary 7, we can rewrite the tail $\boldsymbol{t}^k = \bar{\boldsymbol{P}}(\boldsymbol{\Lambda}^{k+1} + \boldsymbol{\Lambda}^{k+2} + \cdots)\boldsymbol{e}$, and as the diagonal entries of $\boldsymbol{\Lambda}$ are non-negative and less than 1, the infinite series in the above expression sums to $\boldsymbol{\Lambda}^{k+1}(\boldsymbol{I} - \boldsymbol{\Lambda})^{-1}$, and we have a formula for the tail.

Since calculating the eigenvalues and eigenvectors of $\boldsymbol{N}$ is as difficult as solving the original system, this result is of little value. Our starting point towards thinking about the model is that a good estimate of the tail can still be obtained if only the larger eigenvalues are known. This is so because the smaller eigenvalues approach zero rapidly, as k becomes large. This is the basis of the model developed in this section. The model developed here generates estimates of coefficients of a polynomial whose roots are certain eigenvalues of $\boldsymbol{N}$, generally the larger ones. These, instead of the eigenvalues themselves, are used to calculate the estimate, $\hat{\boldsymbol{t}}^k$, of $\boldsymbol{t}^k$ (as is done in the next section).

We now develop the fundamental ideas of the model. Using the result of Corollary 7, we note that

$$\boldsymbol{N}\boldsymbol{q} = \sum_{j=1}^{r} \lambda_j \bar{\boldsymbol{P}}_j$$

and, allowing for multiplicities in the eigenvalues,

$$\boldsymbol{N}\boldsymbol{q} = \sum_{i=1}^{p} \sum_{j \in S_i} \bar{\lambda}_i \bar{\boldsymbol{P}}_j = \sum_{i=1}^{p} \bar{\lambda}_i \sum_{j \in S_i} \bar{\boldsymbol{P}}_j = \sum_{i=1}^{p} \bar{\lambda}_i \bar{\boldsymbol{Q}}_i$$

where S_i, $i = 1, \cdots, p$ is a partition of $\{1, \cdots, r\}$ with $\lambda_j = \bar{\lambda}_i$ for $j \in S_i$, i.e., S_i represents the indices of eigenvalues that have the same value $\bar{\lambda}_i$.

In addition, since

$$\boldsymbol{N}^k\boldsymbol{q} = \sum_{i=1}^{p} \bar{\lambda}_i^k \bar{\boldsymbol{Q}}_i$$

for sufficiently large k, the contribution, to $\boldsymbol{N}^k\boldsymbol{q}$, of "small" $\bar{\lambda}_i$ will be minimal. The partition of the eigenvalues into "similar" groups, and the previous statement about "small" eigenvalues, forms the basis of our model. Since the terms in the tail are associated with large values of k, we note that the contribution to it of "small" eigenvalues will be minimal. We now develop a model which attempts to estimate the contribution of other classes of eigenvalues.

For this purpose, consider the function $(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^k$ of $\boldsymbol{N}$ for some $l \geq 1$ and $k \geq 1$. We then have:

Theorem 8 *The eigenvalues of* $(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^k$ *are* $\{(1-\lambda_i)^l\lambda_i^k\}$ *where* $\{\lambda_i\}$ *are the eigenvalues of* $\boldsymbol{N}$.

Proof: Since $\boldsymbol{N} = \boldsymbol{P}\Lambda\boldsymbol{P}^T$, using the orthonormality of $\boldsymbol{P}$ we can write $(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^k = \boldsymbol{P}(\boldsymbol{I}-\Lambda)^l\Lambda^k\boldsymbol{P}^T$, and we have the result. ■

We note that for $l = 1$ and large k, the eigenvalues of $(\boldsymbol{I}-\boldsymbol{N})\boldsymbol{N}^k$ corresponding to eigenvalues of $\boldsymbol{N}$ close to 1, and small, are small. Thus the function concentrates on relatively large eigenvalues of $\boldsymbol{N}$. It is this observation that will be used to identify the sets S_i. We now investigate this function in the next theorem.

Theorem 9 *The function* $(1-x)^lx^k$ *is maximized at* $x^* = k/(k+l)$.

Proof: We have the result as this function is concave, and its derivative vanishes at x^*. ■

To understand further the relation of the function $(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^k$ to the original sequence $\boldsymbol{q}, \boldsymbol{N}\boldsymbol{q}, \boldsymbol{N}^2\boldsymbol{q}, \cdots$, and to see how to translate the sequence into the above functional form, we introduce the following *finite difference* operator on an ordered sequence of vectors $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_k$:

$$\partial\boldsymbol{x}_j = \boldsymbol{x}_j - \boldsymbol{x}_{j+1}.$$

Then,

$$\partial^{k+1}\boldsymbol{x}_j = \partial(\partial^k\boldsymbol{x}_j) = \partial^k\boldsymbol{x}_j - \partial^k\boldsymbol{x}_{j+1},$$

is the same as the operator ∂ applied $k+1$ times. The following result relates this operator to the required form.

Theorem 10 *Given* $\boldsymbol{q}, \boldsymbol{N}\boldsymbol{q}, \cdots, \boldsymbol{N}^t\boldsymbol{q}$, $\partial^s(\boldsymbol{N}^j\boldsymbol{q})$ *is well defined for all* $s+j \leq t$, *and* $\partial^s(\boldsymbol{N}^j\boldsymbol{q}) = (\boldsymbol{I}-\boldsymbol{N})^s\boldsymbol{N}^j\boldsymbol{q}$.

Proof: This is clearly true for $s=1$. Assume it true for some s and consider $\partial^{s+1}(\boldsymbol{N}^j\boldsymbol{q})$, $s+j+1 \leq t$. But this is, by definition, $\partial^s(\boldsymbol{N}^j\boldsymbol{q}) - \partial^s(\boldsymbol{N}^{j+1}\boldsymbol{q})$ with both terms well defined. Using the induction hypothesis, we see that $\partial^{s+1}(\boldsymbol{N}^j\boldsymbol{q}) = (\boldsymbol{I}-\boldsymbol{N})^s\boldsymbol{N}^j\boldsymbol{q} - (\boldsymbol{I}-\boldsymbol{N})^s\boldsymbol{N}^{j+1}\boldsymbol{q} = (\boldsymbol{I}-\boldsymbol{N})^{s+1}\boldsymbol{N}^j\boldsymbol{q}$ and we are done. ■

Using Theorem 10 one can readily develop an algorithm for generating the required functional form from the original sequence. Thus, generating $(\boldsymbol{I}-\boldsymbol{N})^l\boldsymbol{N}^k\boldsymbol{q}$ requires an l - fold finite differencing of the original sequence, which only involves vector subtractions. Since this is self evident, we omit the details.

We are now ready to develop the linear model that will enable, at iteration k, the estimation of the tail $\hat{\boldsymbol{t}}^k$ from the given sequence $\boldsymbol{q}, \boldsymbol{N}\boldsymbol{q}, \cdots, \boldsymbol{N}^k\boldsymbol{q}$. To do this, assume the eigenvalues fall into the classes $S_1, S_2, \cdots, S_p$ (which form a disjoint partition of $\{1, \cdots, r\}$); and for each class, assume that the functional from $(\boldsymbol{I}-\boldsymbol{N})^{s_i}\boldsymbol{N}^j\boldsymbol{q}$, $s_i \geq 1$, distinguishes the class S_i from the others. Based on this information, we now develop the linear model.

Let $s_i + t \leq k$, and define

$$\begin{aligned}\boldsymbol{\Delta}_t^{(i)} &= \partial^{s_i}(\boldsymbol{N}^t\boldsymbol{q}) \\ &= (\boldsymbol{I}-\boldsymbol{N})^{s_i}\boldsymbol{N}^t\boldsymbol{q} \\ &= \sum_{j=1}^{p} \bar{\boldsymbol{P}}_j(\boldsymbol{I}-\boldsymbol{\Lambda}_j)^{s_i}\boldsymbol{\Lambda}_j^t\boldsymbol{e}\end{aligned}$$

Here $\bar{\boldsymbol{P}}_j$, $\boldsymbol{\Lambda}_j$ are the corresponding matrices associated with S_j, and, as mentioned earlier, $\boldsymbol{\Delta}_t^{(i)}$ is known, and can be generated from the original sequence by finite differencing.

Under the assumption that s_i distinguishes the eigenvalues of S_i from the others, we can write:

$$\boldsymbol{\Delta}_t^{(i)} = \bar{\boldsymbol{P}}_i(\boldsymbol{I}-\boldsymbol{\Lambda}_i)^{s_i}\boldsymbol{\Lambda}_i^t\boldsymbol{e} + \varepsilon_t^{(1)} \tag{11}$$

where we treat $\varepsilon_t^{(1)}$ as the "error" of ignoring the terms pertaining to the eigenvalues not in S_i.

For each i, let $s_i + t_i \leq k$. t_i is the number of eigenvalues in S_i. Define the $r \times t_i$ matrix:

$$\boldsymbol{\Delta}^{(i)} = (\boldsymbol{\Delta}^{(i)}_{k-s_i-t_i+1}, \cdots, \boldsymbol{\Delta}^{(i)}_{k-s_i}). \tag{12}$$

Using (11), it can be readily shown that

$$\boldsymbol{\Delta}^{(i)} = \bar{\boldsymbol{P}}_i(\boldsymbol{I} - \boldsymbol{\Lambda}_i)^{s_i}\boldsymbol{\Lambda}_i^{k-s_i-t_i+1}\boldsymbol{V}_i + \boldsymbol{\Upsilon}^{(1)}$$

where

$$\boldsymbol{V}_i = \begin{bmatrix} 1 & \lambda_1^{(i)} & (\lambda_1^{(i)})^2 & \cdots & (\lambda_1^{(i)})^{t_i-1} \\ 1 & \lambda_2^{(i)} & (\lambda_2^{(i)})^2 & \cdots & (\lambda_2^{(i)})^{t_i-1} \\ . & . & . & \cdots & . \\ 1 & \lambda_{t_i}^{(i)} & (\lambda_{t_i}^{(i)})^2 & \cdots & (\lambda_{t_i}^{(i)})^{t_i-1} \end{bmatrix}$$

is a $t_i \times t_i$ Vandermonde's matrix. Here, $\lambda_j^{(i)}$ are the eigenvalues of $\boldsymbol{N}$ indexed in S_i. One would expect $\boldsymbol{\Upsilon}^{(1)}$ to be small only when such distinguishing s_i exist. We do not expect them to exist in all such systems, but believe that they will exist when these systems are generated by an interior point method. We provide some preliminary evidence of this later in this work.

Thus, for each i, we can write

$$\bar{\boldsymbol{P}}_i = \boldsymbol{\Delta}^{(i)}\boldsymbol{V}_i^{-1}\boldsymbol{\Lambda}_i^{-(k-s_i-t_i+1)}(\boldsymbol{I} - \boldsymbol{\Lambda}_i)^{-s_i} + \boldsymbol{\Upsilon}^{(2)}. \tag{13}$$

Now, $\boldsymbol{N}^k\boldsymbol{q} = \bar{\boldsymbol{P}}\boldsymbol{\Lambda}^k\boldsymbol{e} = \sum_{i=1}^{p-1}\bar{\boldsymbol{P}}_i\boldsymbol{\Lambda}_i^k\boldsymbol{e} + \bar{\boldsymbol{P}}_p\boldsymbol{\Lambda}_p^k\boldsymbol{e}$ where we will assume that S_p contains the indices of all the 'small' eigenvalues. Substituting (13) into the above, we get

$$\boldsymbol{N}^k\boldsymbol{q} = \sum_{i=1}^{p-1}\boldsymbol{\Delta}^{(i)}\boldsymbol{V}_i^{-1}\boldsymbol{\Lambda}_i^{s_i+t_i-1}(\boldsymbol{I} - \boldsymbol{\Lambda}_i)^{-s_i}\boldsymbol{e} + \varepsilon^{(2)}. \tag{14}$$

Defining, for each $i = 1, \cdots, p-1$, $\alpha^{(i)} = \boldsymbol{V}_i^{-1}\boldsymbol{\Lambda}_i^{s_i+t_i-1}(\boldsymbol{I} - \boldsymbol{\Lambda}_i)^{-s_i}\boldsymbol{e}$, and $\boldsymbol{\Delta}^{(0)} = \boldsymbol{N}^k\boldsymbol{q}$, we obtain the linear model

$$\boldsymbol{\Delta}^{(0)} = \sum_{i=1}^{p-1}\boldsymbol{\Delta}^{(i)}\alpha^{(i)} + \varepsilon^{(2)} \tag{15}$$

where $\varepsilon^{(2)}$ is the 'deterministic error', and $\alpha^{(i)}$ are the parameters to be estimated.

A comment on this model is in order here. The parameters $\alpha^{(i)}$ are complicated functions of the eigenvalues of $\boldsymbol{N}$, but by a simple analysis, can be shown to generate, through a

linear transformation, coefficients of a polynomial which has $\lambda_j^{(i)}$ as roots (besides others). Fortunately, we do not need to compute $\lambda_j^{(i)}$, and need only estimates of $\alpha^{(i)}$ (see the next section). Also, the data, $\boldsymbol{\Delta}^{(i)}$ for $i = 0, \cdots, p-1$ in the model is known, and is generated by finite differencing of the original data.

In pedagogical terms, this model sets up a linear relationship between the most recently generated data $\boldsymbol{N}^k\boldsymbol{q}$ and the past data $\boldsymbol{q}, \boldsymbol{N}\boldsymbol{q}, \cdots, \boldsymbol{N}^{k-1}\boldsymbol{q}$. In estimation of the tail, this linear relationship is assumed to hold for all future data (to be estimated) and the past data (assumed known).

This model now allows us to estimate the parameters by the "least squares" procedure, requiring the solution of the normal equation:

$$\boldsymbol{\Delta}^T\boldsymbol{\Delta}\alpha \;=\; \boldsymbol{\Delta}^T\boldsymbol{\Delta}^{(0)}. \tag{16}$$

where $\boldsymbol{\Delta} = (\boldsymbol{\Delta}^{(1)}, \cdots, \boldsymbol{\Delta}^{(p-1)})$, $\alpha = (\alpha^{(1)}, \cdots, \alpha^{(p-1)})^T$, and is a $u \times u$ system of linear equations, with $u = \sum_{i=1}^{p-1} t_i$.

5 Estimating the Tail and the Solution

Let $\hat{\alpha}$ be the estimate of the coefficient α, and be obtained by solving the model (15) by the method of least squares, i.e., by solving (16). We now show how this estimate can be used to compute, $\hat{\boldsymbol{t}}^{k-1}$, an estimate of the tail

$$\boldsymbol{t}^{k-1} = \sum_{j=0}^{\infty} \boldsymbol{N}^{k+j}\boldsymbol{q}.$$

Please note that the "known" term $\boldsymbol{N}^k\boldsymbol{q}$ has been included in the tail and the reason for this inclusion will become clear later.

Using Corollary 7, we can write

$$\begin{aligned} \boldsymbol{t}^{k-1} &= \sum_{j=0}^{\infty}\sum_{i=1}^{p-1} \bar{\boldsymbol{P}}_i\boldsymbol{\Lambda}_i^{k+j}\boldsymbol{e} + \varepsilon^{(3)} \\ &= \sum_{i=1}^{p-1} \bar{\boldsymbol{P}}_i\boldsymbol{\Lambda}_i^{k}(\boldsymbol{I} - \boldsymbol{\Lambda}_i)^{-1}\boldsymbol{e} + \varepsilon^{(3)} \end{aligned}$$

where $\varepsilon^{(3)}$ is the error vector.

Our approach now is to obtain an expression for $\bar{\boldsymbol{P}}_i$ in terms of $\boldsymbol{\Lambda}_i$, such that its substitution into the above expression can be readily identified as some function of $\alpha^{(i)}$. We could use formula (13), but it is unsatisfactory (the final terms involve both $\alpha^{(i)}$ and $\boldsymbol{\Lambda}_i$ in the simplification). We now desire an expression for $\bar{\boldsymbol{P}}_i$ that does the trick.

Define

$$\bar{\boldsymbol{\Delta}}_i^{(i)} = \partial^{s_i-1}(\boldsymbol{N}^t\boldsymbol{q}) \tag{17}$$

Using (17), for $t = k - s_i - t_i + 1, \cdots, k - s_i$, we can define

$$\bar{\boldsymbol{\Delta}}^{(i)} = \bar{\boldsymbol{P}}_i\boldsymbol{\Lambda}_i^{k-s_i-t_i+1}(\boldsymbol{I} - \boldsymbol{\Lambda}_i)^{s_i-1}\boldsymbol{V}_i + \boldsymbol{\Upsilon}^{(3)}.$$

As before, $\bar{\boldsymbol{\Delta}}^{(i)}$ is obtained from the original data by finite differencing, and is known.

Defining $\bar{\boldsymbol{P}}_i$, as in (13), and substituting in above, we get

$$\begin{aligned} \boldsymbol{t}^{k-1} &= \sum_{i=1}^{p-1} \bar{\boldsymbol{\Delta}}^{(i)}\boldsymbol{V}_i^{-1}\boldsymbol{\Lambda}_i^{s_i+t_i-1}(\boldsymbol{I} - \boldsymbol{\Lambda}_i)^{-s_i}\boldsymbol{e} + \varepsilon^{(4)} \\ &= \sum_{i=1}^{p-1} \bar{\boldsymbol{\Delta}}^{(i)}\alpha^{(i)} + \varepsilon^{(4)} \end{aligned}$$

and we can use

$$\hat{\boldsymbol{t}}^{k-1} = \bar{\boldsymbol{\Delta}}\hat{\alpha} \tag{18}$$

(where $\bar{\boldsymbol{\Delta}} = (\bar{\boldsymbol{\Delta}}^{(1)}, \cdots, \bar{\boldsymbol{\Delta}}^{(p-1)})$, and $\hat{\alpha}^T = (\hat{\alpha}^{(1)}, \cdots, \hat{\alpha}^{(p-1)})$ is the "least squares" estimate of α in (15)) as an estimate of $\boldsymbol{t}^{k-1}$. There is clearly a relationship between $\bar{\boldsymbol{\Delta}}$ and $\boldsymbol{\Delta}$.

Using the estimate $\hat{\boldsymbol{t}}^{k-1}$, defined in (18), we set

$$\begin{aligned} \hat{\boldsymbol{y}}^k &= \boldsymbol{s}^{k-1} + \hat{\boldsymbol{t}}^{k-1} \\ &= \boldsymbol{s}^{k-1} + \bar{\boldsymbol{\Delta}}\hat{\alpha} \end{aligned}$$

and use this as an estimate of the solution to (2). Of interest is the error in the equation (2) when this estimate is used. Define the vector

$$\epsilon^k = \boldsymbol{q} - (\boldsymbol{I} - \boldsymbol{N})\hat{\boldsymbol{y}}^k.$$

Indeed, a good estimate of the solution to (2), results in a smaller error ϵ^k, and a good model (15) would generate a lower residual error in the least squares setting. The next theorem asserts that these two errors, are the same, and also justifies the inclusion of $\boldsymbol{N}^k\boldsymbol{q}$ as a part of the tail.

Theorem 11 *Let $\hat{\boldsymbol{y}}^k$ be the estimate of the solution to (2) as computed above using $\hat{\alpha}$, the least squares solution to (15). Then*

$$\epsilon^k = \boldsymbol{q} - (\boldsymbol{I} - \boldsymbol{N})\hat{\boldsymbol{y}}^k = \boldsymbol{N}^k\boldsymbol{q} - \sum_{i=1}^{p-1} \boldsymbol{\Delta}^{(i)}\hat{\alpha}^{(i)}.$$

Proof:

$$\begin{aligned} \boldsymbol{q} - (\boldsymbol{I} - \boldsymbol{N})\hat{\boldsymbol{y}}^k &= \boldsymbol{q} - (\boldsymbol{I} - \boldsymbol{N})(\boldsymbol{s}^{k-1} + \bar{\boldsymbol{\Delta}}\hat{\alpha}) \\ &= \boldsymbol{q} - (\boldsymbol{I} - \boldsymbol{N})(\boldsymbol{q} + \boldsymbol{N}\boldsymbol{q} + \cdots + \boldsymbol{N}^{k-1}\boldsymbol{q} + \bar{\boldsymbol{\Delta}}\hat{\alpha}) \\ &= \boldsymbol{N}^k\boldsymbol{q} - (\boldsymbol{I} - \boldsymbol{N})\bar{\boldsymbol{\Delta}}\hat{\alpha}. \end{aligned}$$

But $(\boldsymbol{I} - \boldsymbol{N})\bar{\boldsymbol{\Delta}}_t^{(i)} = \partial(\partial^{s_i-1}(\boldsymbol{N}^t\boldsymbol{q})) = \boldsymbol{\Delta}_t^{(i)}$ and we have our result. ■

This is an important theorem, since it connects the error in the solution of (2) with the residual vector of the least-squares solution. Thus, if the residual is large, the resulting error in the equation will also be large. Hence a large residual would indicate a model modification which is defined by the number of sets S_i eigenvalues are partitioned into, the number of eigenvalues, t_i, in each set, and the "distinguishing" power s_i.

6 Separating Eigenvalues

The "goodness of fit" of the model developed in section 4 is also effected by the "separation" of the eigenvalues of $\boldsymbol{N}$ into classes that behave similarily when raised to powers. Thus for systems with well separated eigenvalues, our models will exhibit better perform. We present here a technique for enhancing the separation of eigenvalues of $\boldsymbol{N}$. This separation is acheived by rewriting the series involving higher powers of $\boldsymbol{N}$, and the fact that the eigenvalues of $\boldsymbol{N}^k$, for $k \geq 2$, are "further apart" than those of $\boldsymbol{N}$.

Consider the sum of (4).

Theorem 12 *For each $l \geq 1$, there exist a vector $\bar{\boldsymbol{q}}$ and a symmetric positive semi-definite matrix $\bar{\boldsymbol{N}}$ such that*

$$\boldsymbol{y} = \bar{\boldsymbol{q}} + \bar{\boldsymbol{N}}\bar{\boldsymbol{q}} + \bar{\boldsymbol{N}}^2\bar{\boldsymbol{q}} + \cdots$$

where the eigenvalues of $\bar{\boldsymbol{N}}$ are $(l+1)$ powers of the eigenvalues of $\boldsymbol{N}$.

Proof: For each given $l \geq 1$, the sum of (4) can be rearranged as

$$\boldsymbol{y} = (\boldsymbol{q} + \boldsymbol{N}\boldsymbol{q} + \cdots + \boldsymbol{N}^l\boldsymbol{q}) + \boldsymbol{N}^{l+1}(\boldsymbol{q} + \boldsymbol{N}\boldsymbol{q} + \cdots + \boldsymbol{N}^l\boldsymbol{q}) + \boldsymbol{N}^{2(l+1)}(\boldsymbol{q} + \boldsymbol{N}\boldsymbol{q} + \cdots + \boldsymbol{N}^l\boldsymbol{q}) + \cdots$$

and we obtain our theorem by letting $\bar{\boldsymbol{q}} = \boldsymbol{q} + \boldsymbol{N}\boldsymbol{q} + \cdots + \boldsymbol{N}^l\boldsymbol{q}$ and $\bar{\boldsymbol{N}} = \boldsymbol{N}^{l+1}$. ■

Implementation of the result of this theorem requires generating the partial sums:

$$\bar{\boldsymbol{N}}^j\bar{\boldsymbol{q}} = \boldsymbol{N}^{j(l+1)}\boldsymbol{q} + \boldsymbol{N}^{j(l+1)+1}\boldsymbol{q} + \cdots + \boldsymbol{N}^{j(l+1)+l}\boldsymbol{q}$$

from the original data, $\boldsymbol{q}, \boldsymbol{N}\boldsymbol{q}, \boldsymbol{N}^2\boldsymbol{q}, \cdots$. In any computational analysis, this "overhead" must be compared to the resulting savings.

7 Acceleration by Chebychev Polynomials

Chebychev Polynomials, (see for example, Rivlin [8]), provide a convenient and efficient mechanism for making a best lower order polynomial approximation, over a range of values of the variables, to a given polynomial. In this section we show how these polynomials can be used to accelerate the convergence of the techniques developed in this paper. Before we do this, we present a brief introduction.

The Chebychev polynomial of degree n is defined as

$$T_n(x) = a_0^{(n)} + a_1^{(n)}x + \cdots + a_n^{(n)}x^n \tag{19}$$

for all $x \in [-1, 1]$; and, the relationship between these polynomials is defined by the difference equation

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x). \end{aligned}$$

This difference equation can be used to generate coefficients of T_n, and the formulii for these coefficients can be found in [8], and other references.

Given a polynomial $p(x)$ of degree n, a lower order approximation $q(x)$ of degree m, $m < n$, is generated by first expressing $p(x)$ in terms of $T_j(x)$; and then dropping the terms

involving the higher order Chebychev polynomials. Thus, to get a m degree approximation $q(x)$ to $p(x)$, we first define $A_0, \cdots, A_n$ such that

$$p(x) = A_0T_0(x) + A_1T_1(x) + \cdots + A_nT_n(x) \tag{20}$$

and then define

$$q(x) = A_0T_0(x) + A_1T_1(x) + \cdots + A_mT_m(x). \tag{21}$$

The following is well known:

Theorem 13 $|p(x) - q(x)| \leq \sum_{j=m+1}^{n} |A_j|$ *for all* $x \in [-1, 1]$.

In our application, we will apply Chebchev approximations to the series

$$p(\boldsymbol{N}) = \boldsymbol{I} + \boldsymbol{N} + \boldsymbol{N}^2 + \cdots + \boldsymbol{N}^{k-1} \tag{22}$$

of matrices $\boldsymbol{N}$ by the lower order series

$$q(\boldsymbol{N}) = A_0 + A_1T_1(\boldsymbol{N}) + \cdots + A_{k'}T_{k'}(\boldsymbol{N}) \tag{23}$$

where $k' < k - 1$. A caution is in order here. For the ease of exposition, we have abused notation, and used the same symbols, p, q and T, having the set of reals (as in (19) - (21)) or the set of matrices (as in (22) and (23)) , as their domain and range. We can show that

Theorem 14 *For every* k

$$\|T_k(\boldsymbol{N})\|_2 \leq max_i|T_k(\lambda_i)| \leq 1.$$

Proof: Using the fact that $\boldsymbol{N} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T$ for some orthonormal matrix $\boldsymbol{P}$, we see that

$$T_k(\boldsymbol{N}) = \boldsymbol{P}(a_0^{(k)}\boldsymbol{I} + a_1^{(k)}\boldsymbol{\Lambda} + \cdots + a_k^{(k)}\boldsymbol{\Lambda}^k)\boldsymbol{P}^T$$

and thus

$$\begin{aligned} \|T_k(\boldsymbol{N})\|_2 &\leq \|a_0^{(k)}\boldsymbol{I} + a_1^{(k)}\boldsymbol{\Lambda} + \cdots + a_k^{(k)}\boldsymbol{\Lambda}^k\| \\ &= \max_i|(a_0^{(k)} + a_1^{(k)}\lambda_i + \cdots + a_k^{(k)}\lambda_i^k| \\ &= \max_i|T_k(\lambda_i)| \\ &\leq 1 \end{aligned}$$

and we are done. ■

In the estimation of the solution to (2) we will use

$$\hat{\boldsymbol{s}}^{k-1} = q(\boldsymbol{N})\boldsymbol{q}$$

instead of $\boldsymbol{s}^{k-1}$.

Since p and q are polynomials in $\boldsymbol{N}$, the error calculation of Theorem 13 will involve the eigenvalues λ_i of $\boldsymbol{N}$. This follows from the fact that $||p(\boldsymbol{N}) - q(\boldsymbol{N})|| \leq \sum_{k'+1}^{k-1} |A_j| ||T_j(\boldsymbol{N})||_2$ and, from Theorem 14, $||T_j(\boldsymbol{N})||_2 \leq \max_i |T_j(\lambda_i)|$. Since $\lambda_i \in [0,1]$, we need to modify the definition of the polynomials so that the corresponding domain of interest is $[0,1]$. This is readily done by a change of variables to obtain

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= 2x - 1 \\ T_{n+1}(x) &= 2(2x-1)T_n(x) - T_{n-1}(x). \end{aligned} \tag{24}$$

We now compute the coefficients of $T_n(x)$.

Theorem 15 *Let T_j be defined by (24). Then for each $n \geq 1$ and $j \leq n$*

$$a_j^{(n)} = \frac{(-1)^{n+j} n}{n+j} \begin{pmatrix} n+j \\ 2j \end{pmatrix} 4^j.$$

Proof: Using (24), we can readily establish that for each $n \geq 1$:

$$\begin{aligned} a_{n+1}^{(n+1)} &= 4a_n^{(n)} \\ a_n^{(n+1)} &= 4a_{n-1}^{(n)} - 2a_n^{(n)} \\ a_j^{(n+1)} &= -2a_j^{(n)} + 4a_{j-1}^{(n)} - a_j^{(n-1)}, 0 < j < n \\ a_0^{(n+1)} &= -2a_0^{(n)} - a_0^{(n-1)}. \end{aligned}$$

The result now follows by a simple induction hypothesis using $a_0^{(0)} = 1$, and $a_1^{(1)} = 2$, $a_0^{(1)} = -1$. ■

To generate the approximation (23) we need to determine the coefficients, $A_j^{(k-1)}$, $j = 0, \cdots, k-1$, such that

$$p(\boldsymbol{N}) = \sum_{j=0}^{k-1} A_j^{(k-1)} T_j(\boldsymbol{N}). \tag{25}$$

It can be readily established, using the form (19) of T_j, that $A_j^{(k-1)}$ solve the following triangular system of equations:

$$\begin{bmatrix} a_0^{(0)} & a_0^{(1)} & \cdots & a_0^{(k-1)} \\ & a_1^{(1)} & \cdots & a_1^{(k-1)} \\ & & & \cdot \\ & & & a_{k-1}^{(k-1)} \end{bmatrix} \begin{bmatrix} A_0^{(k-1)} \\ A_1^{(k-1)} \\ \cdot \\ A_{k-1}^{(k-1)} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \cdot \\ 1 \end{bmatrix}. \tag{26}$$

Theorem 16 *The solution to the system (26) is: For $j \geq 1$,*

$$A_j^{(k-1)} = \sum_{i=0}^{k-j-1} \binom{2(j+i)}{i} 4^{-j-i+1}, \tag{27}$$

with

$$A_0^{(k-1)} = 1 + \sum_{j=1}^{k-1} (-1)^{j+1} A_j^{(k-1)}. \tag{28}$$

Proof: The (26) can be rewritten as

$$\begin{bmatrix} a_0^{(0)} & \boldsymbol{a}_0 \\ & \boldsymbol{L} \end{bmatrix} \begin{bmatrix} A_0 \\ \boldsymbol{A}_0 \end{bmatrix} = \begin{bmatrix} 1 \\ \boldsymbol{e} \end{bmatrix}$$

for the appropriate upper triangular matrix $\boldsymbol{L}$, vectors $\boldsymbol{A}_0$, $\boldsymbol{a}_0$ and $\boldsymbol{e} = (1, 1, \cdots, 1)^T$. After substituting the result of Theorem 15, we can obtain the solution of (27) by solving

$$8\boldsymbol{L}\boldsymbol{A}_0 = \boldsymbol{r} \tag{29}$$

where

$$\boldsymbol{L} = \begin{bmatrix} 1 & -\binom{2}{1} & \binom{3}{1} & \cdots & (-1)^{k-1}\binom{k-1}{1} \\ & 1 & -\binom{4}{3} & \cdots & (-1)^{k}\binom{k}{3} \\ & & 1 & \cdots & (-1)^{k+1}\binom{k+1}{5} \\ & & & & \cdot \\ & & & & 1 \end{bmatrix},$$

$$\boldsymbol{A}_0 = \begin{bmatrix} A_1^{(k-1)} \\ 2A_2^{(k-1)} \\ \cdot \\ (k-1)A_{k-1}^{(k-1)} \end{bmatrix}, \boldsymbol{r} = \begin{bmatrix} 1 \\ 24^{-1} \\ \cdot \\ (k-1)4^{-k+2} \end{bmatrix}.$$

Using the standard identity

$$\binom{r}{j} - \binom{r-1}{j} = \binom{r-1}{j-1},$$

it can be readily shown that

$$\boldsymbol{L}^{-1} = \boldsymbol{M}_1^2 \boldsymbol{M}_2^2 \cdots \boldsymbol{M}_{k-2}^2$$

where

$$\boldsymbol{M}_j = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & 1 & & & \\ & & & 1 & 1 & & \\ & & & & \cdot & \cdot & \\ & & & & & 1 & 1 \end{bmatrix} \leftarrow \text{row} j$$

which specifies that columns, starting with the j^{th}, be successively added. Since $\boldsymbol{L}^{-1}$ is triangular, we can define

$$\boldsymbol{L}^{-1} = \begin{bmatrix} \hat{\boldsymbol{L}} & \boldsymbol{u} \\ & v \end{bmatrix}$$

and; if we define

$$\boldsymbol{r} = \begin{bmatrix} \hat{\boldsymbol{r}} \\ 0 \end{bmatrix} + r_{k-1}\boldsymbol{u}_{k-1}$$

where $\boldsymbol{u}_{k-1} = (0, 0, \cdots, 1)^T$ (the $(k-1)st$ unit vector),

$$\begin{aligned} \boldsymbol{L}^{-1}\boldsymbol{r} &= \begin{bmatrix} \hat{\boldsymbol{L}}^{-1}\hat{\boldsymbol{r}} \\ 0 \end{bmatrix} + r_{k-1}\boldsymbol{L}^{-1}\boldsymbol{u}_{k-1} \\ &= \begin{bmatrix} \hat{\boldsymbol{L}}^{-1}\hat{\boldsymbol{r}} + r_{k-1}\boldsymbol{u} \\ r_{k-1}v \end{bmatrix}. \end{aligned}$$

Using the last identity, the result of the theorem can be established by a straightforward induction on k. ■

Theorem 16 can now be used to derive the coefficients of the approximating polynomial (23). If

$$q(\boldsymbol{N}) = A'_0 + A'_1\boldsymbol{N} + \cdots + A'_{k'}\boldsymbol{N}^{k'}$$

then

$$\boldsymbol{A'} = \boldsymbol{L'A}$$

where $\boldsymbol{A'} = (A'_0, \cdots, A'_{k'})^T$, $A = (A_0, \cdots, A_{k'})^T$ and

$$\boldsymbol{L'} = \begin{bmatrix} a_0^{(0)} & a_0^{(1)} & \cdots & a_0^{(k')} \\ & a_1^{(1)} & \cdots & a_1^{(k')} \\ & & & \cdot \\ & & & a_{k'}^{(k')} \end{bmatrix}.$$

8 Validation of the Model

In this section, we present some computational results validating the model of Section 4. This is done by its application to the system (4) generated by an assignment problem. This problem has an optimum solution which is highy degenerate, and can thus present special problems for interior point methods. In several examples we tested this model on, the spectral radius of $\boldsymbol{N}$ approaches 1, as the algorithm proceeds to the solution (see Table 3), thus, making the summation of the series (4) very hard. This happens when the optimum solution is unique, and thus a highly degenerate vertex of the assignment polytope. In the case when the algorithm converges to an interior solution, our model performs exceptionally well (see Tables 1 and 2). These computational results are with the dual affine scaling method, see, for example, Vanderbei, Meketon and Freedman [10] for the specifics.

We now derive the matrix $\boldsymbol{N}$ used for the validation. The assignment problem is the following linear program: given n^2 positive real numbers $c_{i,j}$, $i = 1, \cdots, n$, $j = 1, \cdots, n$; find

$x_{i,j}$ that solve the following linear program:

$$\begin{array}{ll} \min & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j}x_{i,j} \\ & \sum_{j=1}^{n} x_{i,j} = 1 \qquad i = 1,\cdots,n \\ & \sum_{i=1}^{n} x_{i,j} = 1 \qquad j = 1,\cdots,n \\ & x_{i,j} \geq 0 \qquad i = 1,\cdots n, j = 1,\cdots,n. \end{array}$$

For this problem, the vertex optimum solutions are known to be highly degenerate. The dual to this problem is the linear program:

$$\begin{array}{ll} \max & \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j \\ & u_i + v_j + s_{i,j} = c_{i,j} \quad i = 1,\cdots n, j = 1,\cdots,n. \\ & s_{i,j} \geq 0 \qquad\qquad i = 1,\cdots n, j = 1,\cdots,n. \end{array}$$

where u_i, v_j are the dual variables and $s_{i,j}$ are the dual slacks.

We apply the dual affine scaling method, starting with the interior dual solution $u_i = 0$, $v_j = 0$ and $s_{i,j} = c_{i,j}(> 0)$ for each i and j. In this case the diagonal matrix, $\boldsymbol{D}$, is a $n^2 \times n^2$ matrix with the diagonal entries, $D_{i,j} = 1/s_{i,j}^2$ for $i = 1,\cdots,n$, $j = 1,\cdots,n$; and the matrix $\boldsymbol{A}$ has the partition (5) of section 3 with

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{e}^T & \cdots & \mathbf{0} \\ \cdot & \cdot & & \cdot \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{e}^T \end{bmatrix}; \boldsymbol{F} = (\boldsymbol{I}, \boldsymbol{I}, \cdots, \boldsymbol{I}),$$

where $\boldsymbol{e}$ is a n-vector of all $1's$, and $\boldsymbol{I}$ is a $n \times n$ identity matrix. Thus, $\boldsymbol{EDE}^T$ and $\boldsymbol{FDF}^T$ of section 3 are diagonal matrices with positive diagonal entries. $\boldsymbol{L}_1$ and $\boldsymbol{L}_3$ are also diagonal matrices, with $\boldsymbol{L}_2 = \boldsymbol{L}_1^{-1}\boldsymbol{FDE}^T$. The matrix $\boldsymbol{N} = \boldsymbol{L}_3^{-1}\boldsymbol{L}_2\boldsymbol{L}_2^T\boldsymbol{L}_3^{-1}$, is a $n \times n$ positive and positive semi-definite matrix with spectral radius less than 1, see, for example Saigal [9].

Computational testing of the model is performed on three sets of randomly generated assignment problems, i.e, the positive constants $c_{i,j}$ are determined randomly. Each set consistes of two similarily behaving problems, one of size 200×200 and the other 300×300. These problems are solved by the dual affine scaling method adopting the methodology of Saigal [9] presented above, and present increasing degree of difficulty to the affine scaling method and the estimation model presented in this work. On the first set of problems, the

method converges to a solution on the boundary of the primal polyhedron with a very few positive variables at value 1; on the second set the method converges to almost a vertex, with about 75% of positive variables at value 1; and, on the third set the method converges to a vertex, with all positive variables at value 1. The "step-size" in the affine scaling method is given the following values: during the first 20 iterations the values, respectivelly, are $0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.9, 0, 90, 0, 93, 0.93, 0.94, 0.95, 0.95, 0.97$ and remains at 0.97 for the $21st$ and higher iterations. Thus the method approaches the boundary gradually.

The matrix $\boldsymbol{N}$ for testing the model is generated by the affine scaling algorithm. The estimated solution $\hat{\boldsymbol{y}}$ generated by the model is not used in the method, but (2) is solved by the technique proposed in Saigal [9]. This is done to guarantee that the same matrix $\boldsymbol{N}$ is used in testing of different parameters. During each iteration of the affine scaling method, the model is tested on the resulting $\boldsymbol{N}$ with $p = 2$ and $p \leq 14$. In the later case, the iterative method automatically increases p as more entries $\boldsymbol{N}^j\boldsymbol{q}$ of the series are generated. The values $s_1 = 1$ and $t_1 = 5$ are used with $p = 2$, and for $p \leq 14$, s_i and t_i are given in the table below:

i	1	2	3	4	5	6	7	8	9	10	11	12	13
t_i	3	2	2	2	2	2	2	2	2	2	2	2	2
s_i	1	4	6	8	10	12	14	16	18	20	22	24	26

The results of this test are presented in Tables 1 and 2. At each iteration of the dual affine scaling method the matrix $\boldsymbol{N}$ is used to recursively generate $\boldsymbol{Nq}, \cdots, \boldsymbol{N}^K\boldsymbol{q}$. This sequence is then used in the model to estimate the solution to the system (2). This system is considered adequately solved when either the error "ERROR" (reported in Tables 1 and 2) of this estimate satisfies

$$\text{ERROR} \leq \min\{10^{-5}, ||\boldsymbol{q}||_2 \times 10^{-4}\}.$$

or $\boldsymbol{N}^{500}\boldsymbol{q}$ has been generated.

As can be verified from the Tables 1 and 2, the problem complexity greatly effects the model performance. The first set of problems (converging to the interior of a face) is solved quickly; i.e., a good estimate of the solution is obtained for small K. Solving (2), for

example requires, (at iteration 14) 44 or 23 matrix multiplications for the 200×200 and 44 or 24 for 300×300 case, instead of the inversion of $\boldsymbol{I} - \boldsymbol{N}$. From the tables, it appears that the second class of problems is easier to solve than the third; and for these problems, the model performs well at the earlier and the later iterations.

This limited experience suggests that the performance is enhanced when p is increased. It can be verified from Table 3, that the spectral radius of the matrix $\boldsymbol{N}$ approaches 1, specially for the harder problems, and that the model performance improves in the later iterations. This may be because the model performance is effected by the number of sets of "similar" eigenvalues, and not necessairly by the magnitude of the largest one. This phenomenon is under investigation.

We expect this experience to carry over to the general problem as well. To date no computations that confirm this have been performed.

9 Acknowledgment

The author gratefully acknowledges many helpful discussions with Teresa Lam.

References

[1] E. R. Barnes, " A variation of Karmarkar's Algorithm for solving Linear Programming Problems, " *Mathematical Programming* 36 (1986) 174 - 182.

[2] S. C. Fang and S. Puthenpura, *Linear Optimization and Extensions* , Prentice Hall, in print.

[3] A. George and J. W. Liu, *Computer solution of large definite systems* , Prentice Hall, Englewood Cliffs, New Jersey (1981).

[4] N. Karmarkar, " A new polynomial time algorithm for linear programming", *Combinatorica* 4(1984) 373 - 395.

[5] M. Kojima, S. Mizuno and A. Yoshise, " A primal-dual interior point method for linear programming ", in *Progress in Mathematical Programming: Interior Point Methods* ,

edited by N. Megiddo, Springer Verlag, New York (1989) 29-48.

[6] L. S. Lasdon, *Optimization Theory for Large Systems* The Macmillian Company, London (1970).

[7] S. Puthenpura, R. Saigal and L. P. Sinha, " Application of LQ Factorization in implementing the Karmarkar algorithm and its variants ", Technical Memorandum, No 51173 - 900205 -01TM, AT&T Bell Laboratories (1990).

[8] T. J. Rivlin, *The Chebychev Polynomials* , John Wiley and Sons, New York (1974).

[9] R. Saigal, "Matrix partitioning methods for interior point algorithms", Technical Report 92 - 39, Department of Industrial and Operations Enginering, University of Michigan, June 1992.

[10] R. Vanderbei, M. S. Meketon and B. A. Freedman, " A modification of Karmarkar linear programming algorithm ", *Algorithmatica* 1986 395 - 407.

[11] J. H. Wilkinson, *The Algebraic Eigenvalue Problem* , Oxford University Press, 1965.

200 × 200 ASSIGNMENT PROBLEM												
Iter no	Interior Facet				Almost Vertex				Vertex			
	p = 2		p ≤ 14		p = 2		p ≤ 14		p = 2		p ≤ 14	
	Error	K	K	Error	Error	K	K	Error	Error	K	K	Error
1	.23E-06	4	4	.23E-06	.21E-07	5	5	.21E-07	.38E-05	7	7	.38E-05
2	.66E-06	4	4	.66E-06	.87E-07	5	5	.87E-07	.15E-07	8	8	.15E-07
3	.22E-05	4	4	.22E-05	.66E-06	5	5	.66E-06	.84E-06	8	8	.84E-06
4	.17E-06	5	5	.17E-06	.85E-05	5	5	.85E-05	.10E-05	10	10	.10E-05
5	.30E-05	5	5	.30E-05	.25E-05	6	6	.28E-05	.10E-05	11	11	.10E-05
6	.32E-05	6	6	.34E-05	.62E-05	7	8	.58E-05	.44E-06	16	16	.44E-06
7	.99E-06	8	9	.21E-06	.57E-05	9	10	.14E-05	.41E-05	19	19	.41E-05
8	.26E-05	9	9	.48E-05	.53E-05	12	12	.20E-05	.85E-05	26	21	.35E-05
9	.36E-05	11	11	.22E-05	.38E-05	17	14	.87E-05	.71E-05	41	33	.50E-05
10	.31E-05	14	13	.11E-05	.92E-05	25	18	.71E-05	.95E-05	65	47	.12E-05
11	.42E-05	16	14	.32E-05	.76E-05	30	20	.73E-05	.92E-05	97	70	.72E-05
12	.16E-05	20	16	.10E-05	.87E-05	43	23	.91E-05	.59E-05	117	79	.50E-05
13	.11E-05	25	18	.64E-06	.91E-05	58	29	.98E-05	.82E-05	183	110	.75E-05
14	.78E-07	44	23	.12E-06	.66E-05	127	61	.89E-05	.91E-05	475	367	.86E-05
15	.72E-08	19	15	.70E-08	.99E-05	268	136	.10E-04	.20E-02	500	500	.32E-03
16	.95E-09	60	29	.10E-08	.19E-05	282	147	.18E-05	.66E+00	500	500	.52E-02
17	.33E-10	18	15	.18E-10	.17E-05	500	500	.67E-06	.13E+00	500	500	.25E-01
18	.57E-11	81	39	.57E-11	.30E-05	500	500	.12E-05	.37E-01	500	500	.15E-01
19	.19E-12	18	15	.11E-12	.85E-06	500	500	.28E-05	.40E-02	500	500	.29E-02
20	.22E-13	115	52	.36E-13	.62E-07	500	500	.92E-06	.90E-04	500	500	.72E-04
21	.13E-14	22	17	.81E-15	.20E-08	500	500	.20E-08	.31E-05	500	500	.20E-05
22	.23E-15	215	95	.23E-15	.81E-11	500	500	.81E-11	.29E-07	3	3	.29E-07
23	.60E-17	18	15	.43E-17	.12E-11	177	79	.11E-11	.42E-09	2	2	.42E-09
24	.85E-18	77	36	.11E-17	.23E-12	500	319	.87E-13	.16E-11	2	2	.16E-11
25					.60E-14	187	86	.60E-14	.68E-14	2	2	.68E-14
26					.59E-15	500	276	.41E-15	.31E-16	2	2	.31E-16
27					.30E-16	239	110	.29E-16	.20E-18	2	2	.20E-18
28					.24E-17	410	210	.25E-17	.62E-21	2	2	.62E-21
29									.27E-23	2	2	.27E-23
10									.20E-25	2	2	.20E-25
31									.11E-27	2	2	.11E-27

Table 1

300 × 300 ASSIGNMENT PROBLEM												
Iter no	Interior Facet				Almost Vertex				Vertex			
	p = 2		p ≤ 14		p = 2		p ≤ 14		p = 2		p ≤ 14	
	Error	K	K	Error	Error	K	K	Error	Error	K	K	Error
1	.16E-06	4	4	.16E-06	.44E-07	5	5	.44E-07	.43E-05	6	6	.43E-05
2	.44E-06	4	4	.44E-06	.49E-07	5	5	.49E-07	.26E-06	7	7	.26E-06
3	.20E-05	4	4	.20E-05	.43E-06	5	5	.43E-06	.89E-07	9	9	.89E-07
4	.11E-06	5	5	.11E-06	.61E-05	5	5	.61E-05	.66E-08	11	10	.44E-06
5	.24E-05	5	5	.24E-05	.10E-05	6	6	.11E-05	.23E-05	10	10	.23E-05
6	.11E-05	6	6	.12E-05	.67E-06	7	7	.60E-05	.39E-05	12	12	.39E-05
7	.86E-06	7	7	.69E-05	.67E-05	8	9	.97E-06	.47E-05	16	16	.47E-05
8	.42E-05	8	9	.98E-06	.51E-05	11	11	.21E-05	.38E-05	24	24	.38E-05
9	.49E-05	10	11	.58E-06	.91E-05	15	14	.30E-05	.35E-05	37	37	.35E-05
10	.52E-05	12	11	.93E-05	.58E-05	21	17	.20E-05	.81E-05	58	42	.20E-05
11	.44E-05	14	13	.24E-05	.69E-05	26	19	.43E-05	.98E-05	84	84	.98E-05
12	.35E-05	16	14	.33E-05	.99E-05	33	22	.54E-05	.69E-05	139	67	.38E-05
13	.20E-05	23	17	.16E-05	.81E-05	51	27	.67E-05	.83E-05	185	107	.87E-05
14	.42E-06	44	24	.31E-06	.82E-05	125	63	.90E-05	.99E-05	354	226	.91E-05
15	.81E-08	24	17	.90E-08	.77E-05	213	119	.96E-05	.51E-03	500	500	.38E-04
16	.29E-08	100	36	.26E-08	.39E-05	488	288	.38E-05	.12E+00	500	500	.46E-02
17	.47E-10	16	14	.33E-10	.37E-05	500	500	.25E-05	.14E+00	500	500	.73E-01
18	.14E-10	150	50	.16E-10	.52E-05	500	500	.43E-05	.89E-01	500	500	.56E-01
19	.39E-12	21	17	.12E-12	.18E-05	500	500	.15E-05	.32E-01	500	500	.22E-01
20	.97E-13	138	57	.91E-13	.21E-06	500	500	.57E-05	.79E-02	500	500	.52E-02
21	.20E-14	20	16	.15E-14	.16E-07	500	500	.27E-06	.27E-02	500	500	.15E-02
22	.40E-15	116	57	.35E-15	.39E-10	124	31	.38E-10	.20E-03	500	500	.25E-03
23	.24E-16	41	23	.23E-16	.30E-11	77	28	.38E-11	.11E-04	500	500	.67E-05
24	.21E-17	125	53	.22E-17	.20E-12	41	21	.13E-12	.88E-07	500	500	.34E-07
25	.25E-18	67	28	.24E-18	.22E-13	69	27	.21E-13	.27E-08	32	17	.24E-08
26					.11E-14	77	29	.71E-15	.19E-10	53	17	.14E-10
27					.93E-16	49	23	.27E-16	.55E-11	39	17	.23E-12
28					.51E-17	250	40	.11E-17	.83E-12	61	19	.99E-13
29					.73E-18	96	33	.41E-18	.45E-13	57	17	.57E-15
30									.44E-14	77	19	.37E-14
31									.17E-15	43	18	.67E-18
32									.21E-16	37	17	.15E-18
33									.12E-17	71	17	.57E-19

Table 2

THE LARGEST EIGENVALUE						
	Interior Facet		Almost Vertex		Vertex	
iter	200	300	200	300	200	300
1	.9901447238672399	.9934978175014585	.9896131723699559	.9932347666942302	.9902704011386622	.9934762989524124
2	.9900961443084537	.9934186912268195	.9899056464756422	.9933083337386703	.9901099097716491	.9933895689884312
3	.9900607937971756	.9933559054478737	.9901720102742565	.9933864437617831	.9899756389655071	.9933091899889467
4	.9899965061222104	.9932791318740413	.9905098393465249	.9935104212676482	.9898290242692684	.9932136733431952
5	.9899036721613828	.9931941062181071	.9909246054695797	.9936881369782937	.9896816509502613	.9931259144676824
6	.9898362574474263	.9931267666279024	.9914143785352492	.9939091182799246	.9895654883729561	.9930849464235598
7	.9899194893548973	.9930952597987705	.9920528292478605	.9941600282091113	.9895965563950285	.9931239002123157
8	.9902630967647999	.9931056428988306	.9927454269019744	.9944891617919949	.9900471983168925	.9932731057716612
9	.9909347495951451	.9938442214108793	.9934552173991646	.9949007307038975	.9910334510805776	.9935938447387953
10	.991698350999346	.9933668666248594	.9942917996965845	.9953666836700457	.9924207814025372	.9941748293727437
11	.9923520122869073	.9936250698922473	.9950296766226088	.9957957479586064	.9937359111635224	.9949332859291869
12	.9930303653401724	.9939487465725621	.9957786450015449	.9961745224941985	.9949046663890485	.9957201695856718
13	.9935758356117154	.9943471698211603	.9963645109429915	.9966200644015995	.9960325015350879	.9964116190494283
14	.9952617651221247	.9951934721853056	.99770385857701	.9978513542039164	.9986825570584397	.9982736953381484
15	.9932782887906138	.9945002638745208	.9982106973541132	.9986067117779875	.9994308944679383	.9987384902708429
16	.9965906303594346	.9958084656045198	.9986350889093472	.9990509284216897	.9998158719261547	.9993068865577555
17	.9931044028591927	.9942191630970714	.9993053726008107	.9993768099938902	.9999507392517608	.9996908839712081
18	.9971093266767113	.9966531473552512	.9998011882172851	.999656820722268	.9999914233005139	.9998352646075588
19	.9922950323648496	.9948769102561883	.9999726712124946	.99980850749491	.9999975689126962	.9998816294842967
20	.9973853729691724	.9967373895081607	.9999978542185801	.999939258173573	.999999824948421	.9999699825813522
21	.9936907023585453	.9935361068659556	.9999998914042579	.9999894909179856	.9999999816879248	.9999984634373612
22	.9980711062108224	.9969089383222424	.9999999841440135	.9999985974184675	.9999999995858695	.999999833594602
23	.9911968175530437	.9957296613353846	.9999999997658635	.9999999503334166	.9999999999067528	.9999999907302287
24	.9969672707273868	.9966921636270423	.9999999999165852	.9999999881180519	.9999999999997641	.9999999991419738
25		.9968005968116425	.9999999999985193	.9999999999413154	.9999999999998697	.9999999999731211
26			.999999999999502	.9999999999677954	.9999999999999836	.9999999999951627
27			.9999999999999911	.9999999999964501	.9999999999999946	.9999999999999893
28			.9999999999999983	.9999999999998771	.9999999999999996	.9999999999999925
29				.999999999999977	1.000000000000002*	.9999999999999992
30					1.000000000000002*	1.000000000000003*
31					1.000000000000003*	1.000000000000004*
32						1.000000000000004*
33						1.000000000000004*

Table 1: $*$ - No eigenvalue of N is larger than 1. This may be due to numerical problems encountered by EISPACK.