NUMERICAL ALGEBRA,
CONTROL AND OPTIMIZATION
Volume **6**, Number **4**, December **2016**

doi:10.3934/naco.2016019
pp. **437–445**

CLOSED-FORM EXPRESSION FOR THE INVERSE OF A CLASS OF TRIDIAGONAL MATRICES

Sigve Hovda

Department of Petroleum and Applied Geophysics
Norwegian University of Science and Technology
Trondheim, Norway

(Communicated by Hua Dai)

Abstract. Despite the simplicity of tridiagonal matrices, they have shown to be very resilient to closed-form solutions. We consider a class of tridiagonal stiffness matrices that stems from a variety of lumped element models in mechanical, acoustical and electrical systems. The computational efforts in such models are related to solving the generalized eigenvalue problem and finding the inverse of the stiffness matrix. To improve accuracy, it is desired to discretisize the problem as much as possible at the expense of growing matrices. This paper improves the efficiency of finding the inverse by a factor of at least three and the computational memory involved is at least halved. Moreover, the result provides an analytical expression for where the stable position is, which might be used in control systems. Surprisingly, it is the practical application itself that guides the proof.

1. **Motivation.** In this paper we develop an expression for the inverse of the n-dimensional square matrix $\mathbf{K}$, which is given by

$$\begin{bmatrix} k_1+k_2 & -k_2 & 0\dots & & 0 & 0 & \\ -k_2 & k_2+k_3 & -k_3 & \dots & 0 & 0 & 0 \\ 0 & -k_3 & k_3+k_4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & k_{n-2}+k_{n-1} & -k_{n-1} & 0 \\ 0 & 0 & 0 & \dots & -k_{n-1} & k_{n-1}+k_n & -k_n \\ 0 & 0 & 0 & \dots & 0 & -k_n & k_n+k_{n+1} \end{bmatrix},$$

where all the k_is can be any complex numbers satisfying the restriction that the determinant is not equal to zero. The matrix $\mathbf{K}$, appears in the canonical example when springs are connected to point masses in a sequence. Describing this problem leads to a system of second order ordinary differential equations, which is described in many mathematical textbooks.

A description of the mechanical aspect of the mass-spring problem is described in [13], but in this book n is limited to two. The extension to n dimensions is given in for instance the book of of Dermel [3]. The sequential mass-spring example is used several times to describe numerical methods in linear algebra such as finding

2010 *Mathematics Subject Classification.* Primary: 15A09.

Key words and phrases. Tridiagonal matrices, matrix inversion, stiffness matrix, mass-spring model, lumped element model.

eigenvalues. Chapter 12 in [12] contains a discussion of other physical systems that have the same set of differential equations. That is, these systems are analogous systems to the sequential mass-spring example. From [12], it is clear that $\mathbf{K}$ appears in mechanical systems involving rotation and electrical systems as well.

In [3], the mass spring problem is horizontal, but as soon as the gravity forces are not perpendicular to the directions of the springs, the stable position of the masses are shifted by a term that is expressed by $\mathbf{K}^{-1}$. This motivates finding an analytical expression for $\mathbf{K}^{-1}$. We will return to this discussion in section 2.

The literature on the inverse of tridiagonal matrices, as given by

$$\mathbf{T} = \begin{bmatrix} \alpha_1 & \beta_1 & 0\ldots & & 0 & 0 & \\ \gamma_1 & \alpha_2 & \beta_2 & \ldots & 0 & 0 & 0 \\ 0 & \gamma_2 & \alpha_3 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \alpha_{n-2} & \beta_{n-2} & 0 \\ 0 & 0 & 0 & \ldots & \gamma_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ 0 & 0 & 0 & \ldots & 0 & \gamma_{n-1} & \alpha_n . \end{bmatrix},$$

consists of papers involving *general approaches* and papers involving *special cases*. Meurant's paper [10] includes a well written review of the literature on the inverse of symmetric tridiagonal matrices. We will not repeat this discussion here, but focus on the main results that are relevant for this study.

The special cases involve closed-form expressions that are related to Toeplitz matrices. A Toeplitz matrix is a matrix where all $\alpha_i = \alpha$, all $\beta_i = \beta$ and all $\gamma_i = \gamma$. Paper [7] contains a closed-form expression for symmetric Toeplitz matrices, while properties of the general Toeplitz matrices are given in [9]. In [6], an analytical solution is given for a pseudo-Toeplitz matrix. This matrix is a symmetric Toeplitz, except for the fact that the first and the last element on the diagonal are different. It is clear that $\mathbf{K}$ is not a special case of any of these papers.

Typically, the goal of the general approaches is to describe general properties of the matrices, where the ultimate goal is to find numerical algorithms for efficient and stable inversion of large matrices, described in for instance [8] and [4]. Of particular interest are some papers, where the entries of the inverse are given by recursive recurrence relations.

The most general result in this direction is the result from Usmani [15], which states that

$$\mathbf{T}^{-1}_{ij} = \begin{cases} (-1)^{-i+j}\beta_i \ldots \beta_{j-1}\theta_{i-1}\phi_{j+1}/\theta_n & \text{if} \quad i \leq j \\ (-1)^{-i+j}\gamma_j \ldots \gamma_{i-1}\theta_{j-1}\phi_{i+1}/\theta_n & \text{if} \quad i > j. \end{cases} \tag{1}$$

Here, the θ_is and the ϕ_is are the recurrence relations given by

$$\begin{aligned} \theta_i &= \alpha_i\theta_{i-1} - \beta_{i-1}\gamma_{i-1}\theta_{i-2} \quad \text{for} \quad 2 \leq i \leq n \\ \phi_i &= \alpha_i\phi_{i+1} - \beta_i\gamma_i\phi_{i+2} \qquad \text{for} \quad 1 \leq i \leq n-1, \end{aligned}$$

with the initial conditions are $\theta_0 = 1$,$\theta_1 = \alpha_1$, $\phi_{n+1} = 1$ and $\phi_n = \alpha_n$. In this expression, θ_n is the determinant of the matrix and is required be nonzero for this inverse to exist. Simply inserting our problem into this framework yields

$$\begin{aligned} \mathbf{K}^{-1}_{ij} &= \begin{cases} (-1)^{-i+j} k_{i+1} \dots k_j \theta_{i-1} \phi_{j+1}/\theta_n & \text{if} \quad i \leq j \\ (-1)^{-i+j} k_{j+1} \dots k_i \theta_{j-1} \phi_{i+1}/\theta_n & \text{if} \quad i > j, \quad \text{where} \end{cases} \\ \theta_i &= (k_i + k_{i+1})\theta_{i-1} - k_i^2 \theta_{i-2} \quad \text{for} \quad 2 \leq i \leq n \\ \phi_i &= (k_i + k_{i+1})\phi_{i+1} - k_i^2 \phi_{i+2} \qquad \text{for} \quad 1 \leq i \leq n-1. \end{aligned} \tag{2}$$

where the initial conditions are $\theta_0 = 1$,$\theta_1 = k_1 + k_2$, $\phi_{n+1} = 1$ and $\phi_n = k_n + k_{n+1}$. It seems difficult to find a closed-form expression for $\mathbf{K}^{-1}$ directly from these expressions.

Another route to finding $\mathbf{K}^{-1}$ is using the fact that the inverse of an invertible irreducible tridiagonal matrix is a generator representable semiseparable matrix and vice versa. This is proven in on page 42 in corollary 1.39 in [16]. This means that the lower triangular $tril(\mathbf{K}^{-1})$ is equal to $tril(\boldsymbol{u}\boldsymbol{v}^T)$ and the upper triangular $trip(\mathbf{K}^{-1})$ is equal to $trip(\boldsymbol{p}\boldsymbol{q}^T)$, where $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{p}$ and $\boldsymbol{q}$ are column vectors of length n.

We can also use the fact that when $\mathbf{K}$ is symmetric, then $trip(\mathbf{K}^{-1}) = trip(\boldsymbol{v}\boldsymbol{u}^T)$, as reported in [1] and [5]. In [10], recursive recurrence relations for these entries are given. Translating to our problem gives

$$\begin{aligned} &v_1 = \frac{1}{d_1}, \quad v_i = \frac{k_2 \cdots k_i}{d_1 \cdots d_i}, \quad \text{for} \quad i = 2, \cdots n \quad \text{where} \\ &d_n = k_n + k_{n+1}, \quad d_i = k_i + k_{i+1} - \frac{k_{i+1}^2}{d_{i+1}}, \quad \text{for} \quad i = n-1, \cdots 1 \end{aligned} \tag{3}$$

and

$$\begin{aligned} &u_n = \frac{1}{e_n v_n}, \quad u_{n-i} = \frac{k_{n-i+1} \cdots k_n}{e_{n-i} \cdots e_n v_n}, \quad \text{for} \quad i = 1, \cdots n-1 \quad \text{where} \\ &e_1 = k_1 + k_2, \quad e_i = k_i + k_{i+1} - \frac{k_i^2}{e_{i-1}}, \quad \text{for} \quad i = 2, \cdots n. \end{aligned} \tag{4}$$

As long as the matrix is weakly diagonally dominant, which $\mathbf{K}$ is, then this algorithm is computationally stable as reported in [10]. Finding the v_is requires $6n - 3$ operations, while finding the u_is requires $6n - 2$ operations. This totals to $12n - 5$ operations and a storage space of $4n$.

To compute the full matrix, it is required to calculate $n(n+1)/2$ more operations, which dominates the computation. However, in many applications, the inverse of the matrix is multiplied by a vector or used in a way where it is not necessary to compute the full matrix. In those cases, we only need to compute the u_is and the v_is. Since $\mathbf{K}$ is weakly diagonally dominant, this is the gold standard that we compare with our new expression for $\mathbf{K}^{-1}$.

In this paper, we seek an analytical solution and we show that it computes the inverse faster and requires less storage space. Moreover, $\boldsymbol{u}$ and $\boldsymbol{v}$ can be calculated from this solution and computation times can be compared directly.

In order to find the inverse, we are inspired by Newton and the laws of physics. In section 2, the mass-spring element model is outlined and an expression for $\mathbf{K}^{-1}$ is proposed. This proposition is proven, while the computational efficiency is discussed in section 3. Some final remarks are given in the sections 4.

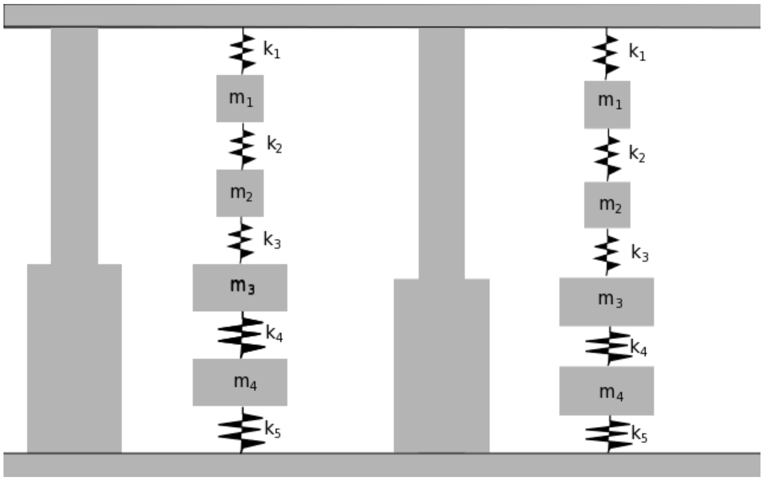

FIGURE 1. The figure shows a rod that is modeled as five springs and four blocks. On the left side, we see the rod and its model when it is nowhere in tension nor in compression. On the right side, we see the effect of gravity on the rod and its model. Since the radius of the rod is non-uniform, the m_is and all the k_is are not equal.

2. **Mass-spring element model of a rod.**

We consider a vertical rod with non-uniform radius, that is mounted at the bottom and at the top. When the rod was mounted, it was nowhere in tension nor in compression. However, because of gravity, parts of the rod become in tension and parts of the rod become in compression. We model this rod as a set of n blocks that are connected by $n+1$ spring elements. This is illustrated in figure 1.

A one dimensional coordinate system is introduced, where the positive direction is pointing downwards. The center point on each block is denoted $X_i(t)$, where t is time. In the unrealistic case where all springs are not in compression and not in tension, these coordinates are denoted $X_{s,i}$. The distance between $X_{s,i}$ and $X_i(t)$ is defined to be $q_i(t)$, that is $X_i(t) = X_{s,i} + q_i(t)$. The physical state of the rod at any time is therefore uniquely defined by the generalized coordinates $q_i(t)$.

Newton's second law on each element gives

$$\begin{aligned}
0 &= -\, m_i\ddot{q}_i + m_1 g - k_1 q_1 + k_2(q_2 - q_1) \\
0 &= -\, m_i\ddot{q}_i + m_i g - k_i(q_i - q_{i-1}) + k_{i+1}(q_{i+1} - q_i) \quad \text{for} \quad 2 \le i \le n-1 \\
0 &= -\, m_n\ddot{q}_n + m_n g - k_n(q_n - q_{n-1}) - k_{n+1} q_n,
\end{aligned} \tag{5}$$

where g is the gravity constant, m_i is the mass of block i and k_i is the spring constant of the spring above block i. Equation (5) is a system of n coupled second order ordinary differential equations. This can be written on a matrix form by

$$\mathbf{M}\ddot{\boldsymbol{q}} + \mathbf{K}\boldsymbol{q} - \boldsymbol{g} = 0, \tag{6}$$

where $\mathbf{M}$ is a diagonal matrix with the m_is on the diagonal and the elements of $\boldsymbol{g}$ are constants of the form $m_i g$. Here, the coefficients of the matrix $\mathbf{K}$ are restricted to be positive.

If we set the $\boldsymbol{g} = \mathbf{0}$ and $k_{n+1} = 0$, the mathematical description reduces to the equations in Demmel's book [3]. In Dermel's book the springs and the masses are laid out horizontally and the last spring does not exist.

If we make the coordinate transformation $\boldsymbol{q} = \boldsymbol{y} + \mathbf{K}^{-1}\boldsymbol{g}$, then equation (6) becomes

$$\mathbf{M}\ddot{\boldsymbol{y}} + \mathbf{K}\boldsymbol{y} = 0, \tag{7}$$

which is homogeneous. Notice that when the initial conditions are $\boldsymbol{y}(0) = 0$ and $\dot{\boldsymbol{y}}(0) = 0$, then all $\boldsymbol{y}$ stays at zero. The solution of this system will fluctuate around its stable position, which is $\boldsymbol{y} = 0$ [14]. The stable position in terms of $\boldsymbol{q}$ is $\mathbf{K}^{-1}\boldsymbol{g}$. In fact the elements of the vector $\mathbf{K}^{-1}\boldsymbol{g}$ are the vertical displacements of all the blocks in figure 1.

In the stable position when nothing is moving we see from Newton's first law that

$$\begin{aligned} k_1 q_1 &= \sum_{j=1}^{n} m_j g - k_{n+1} q_n \quad \text{and} \\ k_i(q_i - q_{i-1}) &= \sum_{j=i}^{n} m_j g - k_{n+1} q_n \quad \text{for} \quad 2 \le i \le n. \end{aligned}$$

Clearly, we can reason that

$$\begin{aligned} q_i &= \sum_{j=1}^{i} \left(k_j^{-1} \left(\sum_{u=j}^{n} m_u g - k_{n+1} q_n \right) \right) \\ &= \sum_{j=1}^{i} \left(k_j^{-1} \sum_{u=j}^{n} m_u g \right) - c_i k_{n+1} q_n, \end{aligned}$$

where $c_i = \sum_{j=1}^{i} k_j^{-1}$. If we insert for $i = n$, then we get an expression for q_n

$$\begin{aligned} q_n &= \sum_{j=1}^{n} \left(k_j^{-1} \sum_{u=j}^{n} m_u g \right) - c_n k_{n+1} q_n \\ &= \frac{\sum_{j=1}^{n} \left(k_j^{-1} \sum_{u=j}^{n} m_u g \right)}{1 + c_n k_{n+1}}. \end{aligned}$$

This means that

$$q_i = \sum_{j=1}^{i} \left(k_j^{-1} \sum_{u=j}^{n} m_u g \right) - \frac{c_i}{c_{n+1}} \sum_{j=1}^{n} \left(k_j^{-1} \sum_{u=j}^{n} m_u g \right),$$

which in matrix notation can be written as $\boldsymbol{q} = (\mathbf{A} - \mathbf{B})\boldsymbol{g}$, where

$$\mathbf{A} = \begin{bmatrix} c_1 & c_1 & c_1 \dots & c_1 & c_1 & c_1 & \\ c_1 & c_2 & c_2 & \dots & c_2 c_2 & c_2 & \\ c_1 & c_2 & c_3 & \dots & c_3 c_3 & c_3 & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ c_1 & c_2 & c_3 & \dots & c_{n-2} & c_{n-2} & c_{n-2} \\ c_1 & c_2 & c_3 & \dots & c_{n-2} & c_{n-1} & c_{n-1} \\ c_1 & c_2 & c_3 & \dots & c_{n-2} & c_{n-1} & c_n \end{bmatrix},$$

$$\mathbf{B} = c_{n+1}^{-1} \begin{bmatrix} c_1 c_1 & c_1 c_2 & c_1 c_3 \dots & c_1 c_{n-2} & c_1 c_{n-1} & c_1 c_n & \\ c_2 c_1 & c_2 c_2 & c_2 c_3 \dots & c_2 c_{n-2} & c_2 c_{n-1} & c_2 c_n & \\ c_3 c_1 & c_3 c_2 & c_3 c_3 \dots & c_3 c_{n-2} & c_3 c_{n-1} & c_3 c_n & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ c_{n-2} c_1 & c_{n-2} c_2 & c_{n-2} c_3 \dots & c_{n-2} c_{n-2} & c_{n-2} c_{n-1} & c_{n-2} c_n & \\ c_{n-1} c_1 & c_{n-1} c_2 & c_{n-1} c_3 \dots & c_{n-1} c_{n-2} & c_{n-1} c_{n-1} & c_{n-1} c_n & \\ c_n c_1 & c_n c_2 & c_n c_3 \dots & c_n c_{n-2} & c_n c_{n-1} & c_n c_n & \end{bmatrix}.$$

This inspires the following theorem:

Theorem 2.1. *If θ_n, as given in equation (2), is not equal to zero, then $\mathbf{K}^{-1} = \mathbf{A} - \mathbf{B}$.*

Notice that the k_is are now allowed to be complex, which is more general than in the example with the rod. Moreover, this result is in compliance with Asplund [1] and Fadeev [5]. In particular, we can use theorem 2.1 to simplify equation (3) and (4) to $u_i = c_i$ and $v_i = 1 - c_i/c_{n+1}$.

It is known that if one of the k_i's is zero, the symmetric tridiagonal problem reduces to two smaller sub-problems as discussed in [2]. In theorem 2.1, some of the k_is are allowed to be zero as long as the determinant is not zero. Division into sub-problems is therefore not necessary as long as θ_n is not equal to zero. It is interesting that this is the case, despite the fact that some of the c_is become infinite.

Contrary, when all $k_i \neq 0$ then all c_i's are finite and hence the inverse exist. To be formal:

Lemma 2.2. *If $k_i \neq 0$ for all $1 \leq i \leq n$, then $\mathbf{K}^{-1} = \mathbf{A} - \mathbf{B}$.*

Notice that in the expressions of the closed form solution, it is clear that unless any of the k_is are very close to zero, then the solution will be robust. It is basically summing reciprocal k_is. In a practical problem, we typically discretize a physical object into a set of discrete entities, which means that $\mathbf{K}$ increases in size. Moreover, the k_is increase in magnitude. If we for instance, model a uniform rod with stiffness k, then the partial stiffnesses will be k times the number of elements. This means that the more we discretisize, the more stable the inversion becomes.

Proof. In order to rigorously prove theorem 2.1, we need to prove that $\mathbf{K}(\mathbf{A}-\mathbf{B}) = \mathbf{I}$ and also that $(\mathbf{A} - \mathbf{B})\mathbf{K} = \mathbf{I}$. Both of these proofs are very similar and we will only show that $\mathbf{K}(\mathbf{A} - \mathbf{B}) = \mathbf{I}$.

It makes sense to let $\mathbf{K} = \mathbf{K_1} + \mathbf{K_2}$, where $\mathbf{K_2}$ is zero everywhere except on $\mathbf{K_2}_{nn}$ where it is equal to k_{n+1}. The proof is concluded when we have shown that $\mathbf{K_1 A} - \mathbf{K_1 B} + \mathbf{K_2}(\mathbf{A} - \mathbf{B}) = \mathbf{I}$.

First, we see that $\mathbf{K_1} = \mathbf{E}\hat{\mathbf{K}}\mathbf{E}^T$ and that $\mathbf{K_1}^{-1} = (\mathbf{E}^T)^{-1}\hat{\mathbf{K}}^{-1}\mathbf{E}^{-1} = \mathbf{A}$, where $\hat{\mathbf{K}}$ is a diagonal matrix with the n-first k_is on the diagonal and

$$\mathbf{E} = \begin{bmatrix} 1 & -1 & 0\dots & & \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & -1 & 0 \\ 0 & \dots & 0 & 1 & -1 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{E}^{-1} = \begin{bmatrix} 1 & 1 & 1\dots & & \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 1 & 1 \\ 0 & \dots & 0 & 1 & 1 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix}.$$

Since $\mathbf{K_1 A} = \mathbf{I}$, it remains to prove that $\mathbf{K_2(A - B)} = \mathbf{K_1 B}$.

As $\mathbf{K}_2$ only contains one element, it is clear that $\mathbf{K_2(A - B)}$ is a matrix with zeros everywhere, except on the last row. In fact the last row is $k_{n+1}(1 - c_n c_{n+1}^{-1})\boldsymbol{c}$, where $\boldsymbol{c} = \{c_1, c_2, c_3, ..., c_{n-2}, c_{n-1}, c_n\}$. Noting that $c_{n+1} = c_n + k_{n+1}^{-1}$, we see that the last row is $c_{n+1}^{-1}\boldsymbol{c}$.

The first row of $\mathbf{K_1 B}$ is $c_{n+1}^{-1}(k_1 c_1 + k_2 c_1 - k_2 c_2)\boldsymbol{c} = \mathbf{0}$ and the ith row is $c_{n+1}^{-1}(-k_i c_{i-1} + (k_i + k_{i+1})c_i - k_{i+1}c_{i+1})\boldsymbol{c} = \mathbf{0}$, when $2 \le i \le n-1$. Finally the nth row is $c_{n+1}^{-1}(-k_n c_{n-1} + k_n c_n)\boldsymbol{c} = c_{n+1}^{-1}\boldsymbol{c}$, which proves that $\mathbf{K_2(A - B)} = \mathbf{K_1 B}$. □

3. Computational efficiency of computing large matrices.

Computing the closed-form expression is very fast. Computing the c_i's requires $2n-1$ computations and the storage is approximately n. In many applications, the actual inverse is not needed and further calculations can be derived from the c_is. For comparison, computing the u_is and the v_is involves $4n - 1$ computations, which is much less than the $12n - 5$ computations that are needed when using equation (3) and (4). Moreover, in the special case when $k_{n+1} = 0$, the full matrix is computed in just $2n - 1$ computations. This is a drastic enhancement compared to the method in [10], where there is a $\mathcal{O}(n^2)$ term to compute the entries of the matrix.

In order to discuss the efficiency of computing the inverse of large matrices, we decided to compare computation of the u_is and the v_is when using the new method and the standard method, which is here defined as using the recursive relations in equation (3) and (4). We generated n k_is using a random generator and chose them to be between 1 and 2 to avoid that any of them could be too close to zero. The code was implemented in Matlab (Mathworks Inc) on an iMac with processor 3,5 GHz Intel Core i7 and 32 GB 1600 MHz DDR3 memory card.

Table 1 shows computational speed for various n. For large n, we see the expected linear relationship between cpu times and n. Moreover, the new method seems to be in the range of three times faster then the standard method. This is expected as the new method contains three times less computations. When $n = 10^9$, the standard method is more than four times slower. The reason for this may be that the standard method requires more storage space than the new method.

It is also worth mentioning that it is often not necessary to compute the v_i in many applications, since they are directly derived from the u_is. In this case, the computation time will be even less than what is reported here.

4. Final remarks and conclusion.

A compact formula for the inverse of a type of matrices that is frequently used in modeling mechanical and electrical systems is found. Even though, there exist stable and efficient numerical algorithms, it is clear that this method is faster than existing methods. Other practical consequences of theorem 2.1 is given below.

Matrix size n	Cpu time in seconds	
	Standard method	New method
10^5	0.01	0.00
10^6	0.07	0.03
10^7	0.73	0.40
10^8	7.62	2.22
10^9	98.94	22.19

TABLE 1. The figure shows computational speed of computing the u_is and the v_is using the standard method, ie solving equation (3) and (4) and the new method which involves computing the c_is.

The theorem provides an analytical and comprehendible insight about the stable position of the problem that is described by equation (6). The solution of the differential equations oscillates around $\mathbf{K}^{-1}\boldsymbol{g}$. If we add a damping term to the problem, the system will either be under-, over-, or critically damped. In all cases, the behavior moves toward the stable position.

In many real systems, the coefficients in the matrices may actually have a time dependence. This complicated the full solution of the differential equations, but the "stable position" can still be calculated by the inverse at each time step. Solving this system can be done by for instance adaptive observers [11]. It is possible that the closed-form expression for the inverse can help regulating the adapters.

A last point regarding the consequences of theorem 2.1, is that it actually allows a discussion of when n goes to infinity. The lumped element model simplifies the description of a spatially distributed system into a topology of n identities. This simplification can be studied by letting n go to infinity. This type of discussions are difficult without analytical expressions for the inverse.

Acknowledgments. We would like to thank the editorial staff, including the reviewers for valuable contributions on this article.

REFERENCES

[1] E. Asplund, Inverse of matrices $\{a_{ij}\}$ which satisfy $a_j = 0$ for $j > i + p$, *Mathematica Scandinavia*, **7** (1959), 57–60.

[2] W. W. Barrett, A theorem on inverse of tridiagonal matrices, *Linear Algebra and its Applications*, **27** (1979), 211–217.

[3] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.

[4] M. E. A. El-Mikkawy, On the inverse of a general tridiagonal matrix, *Applied Mathematics and Computation*, **150** (2004), 669–679.

[5] D. K. Fadeev, Properties of a matrix, inverse to a hessenberg matrix, *Journal of Sovjet Mathematics*, **24** (1984), 118–120.

[6] C. D. Fonseca, On the eigenvalues of some tridiagonal matrices, *Journal of Computational and Applied Mathematics*, **200** (2007), 283–286.

[7] G. Hu and R. F. O'Connell, Analytical inversion of symmetric tridiagonal matrices, *Journal of Physics A: Mathematical and General*, **29** (1996), 1511–1513.

[8] E. Kilic, Explicit formula for the inverse of a tridiagonal matrix by backward continued fractions, *Applied Mathematics and Computation*, **197** (2008), 345–357.

[9] R. K. Mallik, The inverse of a tridiagonal matrix, *Linear Algebra and its Applications*, **325** (2001), 109–139.

[10] G. Meurant, A review on the inverse of symmetric tridiagonal and block tridiagonal matrices, *SIAM Journal on Matrix Analysis and Applications*, **13** (1992), 707–728.

[11] K. S. Narendra and A. M. Annaswarny, *Stable Adaptive Systems*, Prentice Hall, 1989.

[12] K. U. Siddiqui and M. K. Singh, *Mechanical System Design*, New Age International, 2007.

[13] T. L. Smith and K. S. Smith, *Mechanical Vibrations : Modeling and Measurement*, Springer, 2011.

[14] F. Tisseur and K. Meerbergen, The quadratic eigenvalue problem, *Society of Industrial and Applied Mathematics, Review*, **43** (2001), 235–286.

[15] R. Usmani, Inversion of a tridiagonal jacobi matrix, *Computers & Mathematics with Applications*, **27** (1994), 59–66.

[16] R. Vandebril, M. V. Barel and N. Mastronardi, *Matrix Computations and Semiseparable Matrices: Linear Systems*, Johns Hopkins University Press, 2007.

Received June 2016; 1^{st} revision October 2016; final revision November 2016.

E-mail address: sigve.hovda@ntnu.no