TRIDIAGONAL MATRICES: CHARACTERIZATIONS, APPLICATIONS, SOLUTIONS

MORSHED, EMIROV, BROWNING, AND WANG

Abstract. The goal of this work is present both an analytic and computational review of Tridiagonal matrices. Of the many analytic benefits, we find particular interest in the closed-form solutions for many indispensable matrix properties. Also, through their inherent sparsity, Tridiagonal systems provide a computational quintessence which directly lends itself to tailor-made algorithms for solving.

1. Introduction

In mathematics, a tridiagonal matrix is a matrix that has nonzero elements only on the subdiagonal, superdiagonal and the diagonal itself. Although conventionally referred to as tridiagonal, the names band matrix with bandwidth 3 and upper and lower Hessenberg matrix may also be encountered in literature. Tridiagonal systems are of the form

$$T = [t_{ij}] = \begin{pmatrix} a_1 & b_1 & & & 0 \\ c_1 & a_2 & b_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ 0 & & & c_{n-1} & a_n \end{pmatrix} \tag{1}$$

Defining the t_{ij} element using piecewise notation we write

$$t_{ij} = \begin{cases} a_i, & \text{if } i = j; \\ b_i, & \text{if } j - i = 1; \\ c_{i-1}, & \text{if } i - j = 1; \\ 0, & \text{otherwise.} \end{cases}$$

Date: April 25, 2016.

Often, as a form of shorthand notation, the tridiagonal system may be notated as

$$T = \mathrm{Trid}[c_{n-1}, a_n, b_{n-1}],$$

where

$$c_{n-1} := (c_1, c_2, ..., c_{n-1}) \text{ is the vector along the subdiagonal,}$$

$$a_n := (a_1, a_2, ..., a_n) \text{ is the vector along the diagonal,}$$

$$b_{n-1} := (b_1, b_2, ..., b_{n-1}) \text{ is the vector along the subdiagonal.}$$

In this **general** case, we assume that we do not have equality of all elements in each diagonal vector. $c_i \neq c_j$, $a_i \neq a_j$, $b_i \neq b_j$, for some $1 \leq i, j \leq n$.

In the event that $c_i = c_j$, $a_i = a_j$, $b_i = b_j$, for all $1 \leq i, j \leq n$, we write

$$T = \mathrm{Trid}[c, a, b].$$

This special subclass of tridiagonal systems is often referred to as **Toeplitz tridiagonal** matrices and provide the benefit of allowing for closed-form solutions of some its characterizing properties.

As a subclass of the aforementioned, we have **symmetric Toeplitz tridiagonal** matrices, which are characterized by the equivalence of all elements in both the super- and subdiagonals. That is to say that $c_i = c_j$, $a_i = a_j$, $b_i = b_j$, $c_i = b_j$, for all $1 \leq i, j \leq n$. We notate them as follows:

$$T = \mathrm{Trid}[b, a, b].$$

Lastly, before proceeding to the more technical details of tridiagonal matrices, we mention the subclass of tridiagonal systems (non-Toeplitz, symmetric) characterized by $a_i = 0$ for $1 \leq i \leq n$ and $b_i = c_i$ for all $1 \leq i \leq n-1$. For the purposes of this paper we will call these systems **symmetric zero-tridiagonal** matrices and they shall be notated as

$$T = \mathrm{Trid}[b_{n-1}, a, b_{n-1}].$$

Now, let us turn our attention to some of the analytic and computational properties of tridiagonal systems.

2. Characterizations & Properties

As previously discussed, there are multiple benefits to working with tridiagonal systems over general $n \times n$ matrices. Analytically these benefits often come in the form of closed-form solutions for many of the ubiquitous matrix properties. Not only do these benefit our analysis of the subjects of matrix theory and linear algebra, they extend to the numerical realm as well. An attractively low operational cost is enjoyed when using general tridiagonal systems for iterative methods and even more so when the systems are symmetric [1]. Of course, the cost-benefit analysis must be performed to determine if these reductions will be computationally beneficial. Being that the scope of this work is limited, we will ultimately focus our attention in this section to those properties pertaining to the determinant, the eigenvalues and the inverse of each of the classes of matrices we introduced in the introduction.

Determinant

For a general, non-tridiagonal matrix, the computational cost to compute the determinant is $\mathcal{O}(n^3)$. However, when working with a general tridiagonal matrix, our cost is reduced to $\mathcal{O}(n)$ [2]. In the **general** case, the determinant can be calculated using the recursive open-form equation defined as a function of the preceding principal minors. The equation is given as

$$\delta_n = a_n\delta_{n-1} - c_{n-1}b_{n-1}\delta_{n-2}, \tag{2}$$

where $\delta_{-1} = 0$ and $\delta_0 = 1$.

However, this open-form solution is not idea. Ideally, we would like to have a closed-form solution. Thus, now that this recursive relationship has been observed, let us also observe that (2) takes the form of a homogeneous second order difference equation. That is to say that (2) is of the form

$$y_{n+2} = -A_n y_{n+1} - B_n y_n, \quad n \geq 1 \tag{3}$$

where the initial conditions $y_{-1} = 0$ and $y_0 = 1$ are given and A_n and B_n are complex variable coefficients with $B_n \neq 0$ [4]. Associating this definition with (2), we can see that for our case, each A_n and B_n will be defined in terms of the original entries of our matrix $T = \text{Trid}[c_{n-1}, a_n, b_{n-1}]$. More specifically, $A_i = -a_i$ for all $i = 1, ..., n$ and $B_i = c_{i-1}b_{i-1}$ for all $i = 1, ..., n-1$. Then as author Ranjan Mallik states on page 37 of his work, we define

$$\sigma_k := \frac{B_k}{A_{k-1}A_k}, \quad k \geq 2 \qquad \text{and} \qquad \sigma_1(n) := \frac{B_1}{A_n A_1}. \tag{4}$$

Mallik then introduces a set $S_q(L,U)$ q-tuples of natural numbers, where $q, L, U \in \mathbb{N}$, which can be examined in more detail in [4]. Thus, we write

$$S_q(L,U) = \begin{cases} \{L, L+1, ..., U\}, \text{ if } U \geq L \text{ and } q = 1 \\ \{(k_1, ..., k_q) : k_1, ..., k_q \in \{L, L+1, ..., U\}; \\ \qquad\qquad k_l - k_{l-1} \geq 2 \text{ for } l = 2, ..., q\} \\ \qquad\qquad \text{if } \; U \geq L+2 \text{ and } 2 \leq q \leq \lfloor \frac{U-L+2}{2} \rfloor \\ 0, \qquad\qquad \text{otherwise.} \end{cases} \tag{5}$$

Then, by Proposition 5 (p. 37, [4]), we write the closed-form solution of the determinant of T with initial conditions $\delta_1 = a_1$ and $\delta_2 = a_1a_2 - c_1b_1$ as

$$\begin{aligned} \delta_n &= C_{n-2}\delta_2 + D_{n-2}\delta_1 \\ &= (a_1a_2 - c_1b_1)C_{n-2} + a_1 D_{n-2}, \quad n \geq 2 \end{aligned}$$

where $C_0 = 1$, $C_1 = -A_1$, $D_0 = 0$, $D_1 = -B_1$, $D_2 = B_1A_2$ and

(6)
$$C_n = (-1)^n (A_1 \cdots A_n) \times \left(1 + \sum_{q=1}^{\lfloor n/2 \rfloor} (-1)^q \sum_{(k_0, ..., k_{q-1}) \in S_q(2,n)} (\sigma_{k_0} \cdots \sigma_{k_{q-1}}) \right)$$

(7)
$$D_n = (-1)^{n+1} B_1 (A_2 \cdots A_n) \times \left(1 + \sum_{q=1}^{\lfloor (n+1)/2 \rfloor} (-1)^q \sum_{(k_1, ..., k_{q-1}) \in S_{q-1}(3,n)} (\sigma_{k_1} \cdots \sigma_{k_{q-1}}) \right)$$

for $n \geq 2$ and $n \geq 3$ respectively and where σ_k is defined by (4) $S_q(L,U)$ is defined by (5) .

As may be discerned from (7) and (8), the calculation of the determinant of a general tridiagonal matrix can become very notationally convoluted, amongst other challenges. Therefore, it is often much more convenient, when presented with one of the subclasses mentioned in the introduction, to exploit the specifications of the system to develop simpler methods of calculation.

When presented with a **Toeplitz tridiagonal** matrix $T = \text{Trid}[c, a, b]$, the determinant can be computed by

$$\delta_n = a\delta_{n-1} - cb\delta_{n-2}, \tag{8}$$

where, as before, $\delta_{-1} = 0$ and $\delta_0 = 1$. However, we also have a closed-form solution that is considerably more straightforward and is given by

$$\delta_n = \begin{cases} \frac{1}{t_1 - t_2}[t_1^{n+1} - t_2^{n+1}], & \text{if } a^2 \neq 4cb \\ (n+1)\left(\frac{a}{2}\right)^2, & \text{if } a^2 = 4cb \end{cases},$$

where $t_1 = \dfrac{a + \sqrt{a^2 - 4cb}}{2}$ and $t_2 = \dfrac{a - \sqrt{a^2 - 4cb}}{2}$.

Similarly, we may define closed-form solutions for the other subclasses of matrices previously mentioned. However, we will suffice it to have shown the general case and subclass example.

Eigenvalues

Let the eigenvalues of a general tridiagonal matrix T be represented by λ. Then, the characteristic polynomial of T may be represented by

$$\overline{\delta_n} = \det\left\{\text{Trid}[c_{n-1}, a_n - \lambda, b_{n-1}]\right\} = 0.$$

As with the determinant, we may relate the characteristic polynomial to the recursive formula given in (2), namely

$$\overline{\delta_n} = (a_n - \lambda)\overline{\delta}_{n-1} - b_{n-1}c_{n-1}\overline{\delta}_{n-2} = 0. \tag{9}$$

In order to determine the value of λ, we must solve (4) because there is no closed-form solution for the eigenvalues of a **general** tridiagonal matrix.

When $T = \text{Trid}[c, a, b]$ is a **Toeplitz tridiagonal** matrix, the k^{th} eigenvalue of T may computed by ([3])

$$\lambda_k = a + 2\sqrt{bc}\cos\left[\frac{k\pi}{n+1}\right], \text{ for } k = 1, ..., n. \tag{10}$$

Moreover, as is made apparent in (5), for all $k = 1, ..., n$, the eigenvalues of T will be bound by

$$a - 2\sqrt{bc} \leq \lambda_k \leq a + 2\sqrt{bc}.$$

As an immediate consequence of (5), we obtain the closed form for the eigenvalues of a **symmetric Toeplitz** tridiagonal matrix. Namely,

$$\lambda_k = a + 2b\cos\left[\frac{k\pi}{n+1}\right], \text{ for } k = 1, ..., n. \tag{11}$$

as well as

$$a - 2b \leq \lambda_k \leq a + 2b. \tag{12}$$

Finally, in the case where $T = \text{Trid}[b_{n-1}, a, b_{n-1}]$ is a **symmetric zero-tridiagonal** matrix, the characteristic polynomials are of the form of Hermite polynomials. Schur showed that Hermite polynomials of even degree are irreducible and that their Galois groups are not solvable. In other words, in this circumstance, not only is the possibility of a closed-form solution unsolved but has been proven to be impossible!

3. Applications

Outside of the fields in which they first found life, namely matrix theory and linear algebra, tridiagonal matrices have been found to naturally manifest in a multitude of real-world scenarios. These systems can be found in applications such as the fast diffusion question in the field of digital imaging [6], the comparison of a wavelet-Galerkin procedure with a Crank-Nicolson-Galerkin procedure for the diffusion equation [7], and quantitative remote sensing inversion in earth science [8].

Another place where tridiagonal systems may arise is in the solution of the vertical force produced by vibrating strings with coupled oscillators with fixed end-points. Specifically, consider the given system shown in **Figure 1.**:

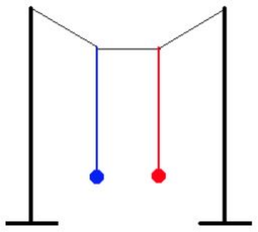

Figure 1. Coupled Oscillators with Fixed End-Points

In the given system, we have $n = 2$ equally-spaced balls, each with mass m. Then, we let y_k represent the vertical displacement of the k^{th} mass and τ be the tension in the string. Also, let ϕ_k be the angle of elevation from the k^{th} mass to the $k+1^{\text{st}}$ mass, with respect to the horizontal plane. Then, the force produced on the vertical plane may be measured by

$$my_k'' = \tau \sin \phi_k - \tau \sin \phi_{k-1}, \quad k = 1, ..., n \tag{13}$$

This is a system of second order linear constant coefficient differential equations with the boundary conditions $y_0(t) = 0$ and $y_{n+1}(t) = 0$. Thus, the solution to this problem shares many similarities with those found in our quest to compute the determinant of a general $n \times n$ tridiagonal matrix. However, (13) is nonlinear but we may make some numerical assumptions, namely that $\phi_k \approx 0$, in order to reduce our system that would represent a tridiagonal system. The reduction would be

$$\alpha^2 v_k = p^2(v_{k+1} - 2v_k + v_{k-1}), \quad k = 1, ..., n.$$

Thus, we have

(14)
$$\begin{bmatrix} -(2p^2+\alpha^2) & p^2 & & & 0 \\ p^2 & -(2p^2+\alpha^2) & p^2 & & \\ & p^2 & \ddots & \ddots & \\ & & \ddots & \ddots & p^2 \\ 0 & & & p^2 & -(2p^2+\alpha^2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

4. Solutions: Gauss Elimination vs. Gauss-Seidel

In the following section, we will compare and contrast the computational properties of the Gauss Elimination and Gauss-Seidel methods for solving systems of equations. Our focus will be restricted to symmetric, tridiagonal matrices of the form:

$$Ax = \begin{bmatrix} d_1 & a_1 & 0 & 0 & 0 & . & . & . & 0 \\ a_1 & d_2 & a_2 & 0 & 0 & . & . & . & 0 \\ 0 & a_2 & d_3 & a_3 & 0 & . & . & . & 0 \\ 0 & 0 & a_3 & d_4 & a_4 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & a_{n-3} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & 0 & . & . & 0 & a_{n-2} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & . & . & 0 & 0 & a_{n-1} & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ . \\ . \\ . \\ b_{n-1} \\ b_n \end{bmatrix} = b$$

Gauss Elimination

Let us denote the i^{th} row of A by R_i and the new i^{th} row, after performing the Gauss Elimination method, by R_i'. Also, we will represent the new i^{th} diagonal entry of A by d_i' and new i^{th} entry of b by b_i'. Then, let us perform the following operations in accordance with Gauss elimination:

$$R_2 \leftarrow R_1\big(-\frac{a_1}{d_1}\big) + R_2$$
$$R_3 \leftarrow R_2'\big(-\frac{a_2}{d_2'}\big) + R_3$$
$$\vdots$$
$$R_n \leftarrow R_{n-1}'\big(-\frac{a_{n-1}}{d_{n-1}'}\big) + R_n$$

After performing these successive operations, we obtain an updated system which, as is often the goal of Gauss elimination, has zeroes below the leading terms in each row. That is to say that

$$(15)\quad \begin{bmatrix} d_1' & a_1 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & d_2' & a_2 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & d_3' & a_3 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & d_4' & a_4 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & 0 & d_{n-2}' & a_{n-2} & 0 \\ 0 & 0 & 0 & . & . & 0 & 0 & d_{n-1}' & a_{n-1} \\ 0 & 0 & 0 & . & . & 0 & 0 & 0 & d_n' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1' \\ b_2' \\ b_3' \\ . \\ . \\ . \\ b_{n-1}' \\ b_n' \end{bmatrix}$$

where,

$$d_1' = d_1, \qquad b_1' = b_1$$
$$d_2' = d_2 - \frac{a_1^2}{d_1}, \quad b_2' = b_2 - b_1\big(\frac{a_1}{d_1}\big)$$
$$d_3' = d_3 - \frac{a_2^2}{d_2'}, \quad b_3' = b_3 - b_2'\big(\frac{a_2}{d_2'}\big)$$
$$\vdots$$
$$d_n' = d_n - \frac{a_{n-1}^2}{d_{n-1}'}, b_n' = b_n - b_{n-1}'\big(\frac{a_{n-1}}{d_{n-1}'}\big).$$

Now, let us turn our attention to the operational cost of performing the Gauss elimination method. First, since the lower diagonal is zero and the upper diagonal stays same, we do not need to compute them. Hence, we just need to calculate d_i which, for each i will cost

$$1(\text{multiplication})+1(\text{division})+1(\text{subtraction})=3.$$

In total, we will need to perform $3(n-1)$ operations on the given matrix. Similaryl, we will operate on the solution vector b, which will come at a cost of $3(n-1)$ operations. All together, our computational cost will be $6(n-1)$ for each b. Moreover, in the event that we are presented with M different solution vectors b (as could be expected in industry), the total number of operations will be $3(M+1)(n-1)$.

Then, employing the method of backward sustitution, our system can be solved by the following computational algorithms:

$$\begin{aligned} x_n &= b'_n/d'_n && \text{(1 operation)} \\ x_{n-1} &= \frac{b'_{n-1} - a_{n-1}x_n}{d'_{n-1}} && \text{(3 operations)} \\ &\vdots \\ x_i &= \frac{b'_i - a_i x_{i+1}}{d'_i} && \text{(3 operations)} \\ &\vdots \\ x_1 &= \frac{b'_1 - a_1 x_2}{d'_1} && \text{(3 operations)} \end{aligned}$$

Thus, the number of operations required for backward substitution is $3(n-1)+1$. Finally, we conclude that in order to solve the tridiagonal system of the equations using the Gauss elimination method, a total of $(3M+6)(n-1)+1$ operations must be performed.

Gauss-Seidel Method

The iterative formula given by $Qx^{(k)} = (Q-A)x^{(k-1)} + b$, will produce a sequence converging to the solution of $Ax = b$, for any $x^{(0)}$, if and only if $\rho(I - Q^{-1}A) < 1$.

In order to perform the Gauss-Seidel Method, let us define Q to be the lower triangular part of A, including the diagonal.

Then we have the following algorithms for computing the solution:

$x_1^{(k)} = \frac{1}{d_1}\left[b_1 - a_1 x_2^{(k-1)}\right]$(3 operations)

$x_i^{(k)} = \frac{1}{d_i}\left[b_i - a_{i-1}x_{i-1}^{(k)} - a_i x_{i+1}^{(k-1)}\right]$ for $i = 2, 3, ..., n-1$, (5 operations)

$x_n^{(k)} = \frac{1}{d_n}\left[b_n - a_{n-1}x_{n-1}^{(k)}\right]$ (3 operations)

Then, the number of operations for one iteration of the Gauss-Seidel method is $3 + 5(n-2) + 3 = 5n - 4$. In order to compute k iterations, we will need to perform a total of $k(5n-4)$ operations.

Comparison

Let us begin by assuming that the given system is consistent.

1. If the matrix A is not a diagonally dominant matrix then we can not apply Gauss Seidel method due to its lack of convergence. However, we may employ the Gauss elimination method with pivoting to achieve solutions to the system.

2. Now, from this point forward, let assume that the matrix A is a diagonally dominant matrix. Then, we can apply Gauss Seidel method to solve the system, where the solution will converge to the exact solution for any given error bound.
Since, Gauss elimination preserves the diagonal dominance of A, then each $|d_i'| \geq |a_i| > 0$, for all $a_i \neq 0$. If $a_i = d_i' = 0$ then $b_i' = 0$ since the system is consistent and we can ignore that row to minimize the number of operations. Therefore, with all of these possible scenarios, we do not apply the row interchanging process in the Gauss elimination method.

3. Because the number of operations in Gauss elimination $(3M+6)(n-1)+1$ depends on the amount, M, of solution vectors, if we have a high amount of different b vectors then the number of operations will also be high. However, for the Gauss Seidel method, we will just multiply it by M to achieve $Mk(5n-4)$.

4. In the Gauss Seidel method, we can control the error by either increasing or decreasing the number of iterations. Thus, we will not have an issue with round-off error. But in the Gauss elimination method, if our entries are very small or relatively close to each other, then it is very possible that we will incur a large round-off error. TMoreover, the larger our given system, the more this error will be magnified. This comes as a result of having to compute more operations each time with the succeeding round-off error, which in turn will produce an ever larger error.

Example 1:

$$A = \begin{bmatrix} 6 & 6-\epsilon & 0 \\ 6-\epsilon & 6 & \epsilon-1 \\ 0 & \epsilon-1 & 6 \end{bmatrix} \tag{16}$$

If ϵ is small enough, then $d_2' = 6 - \frac{(6-\epsilon)^2}{6} \approx 0$ in Matlab, which will not give the correct answer if we use Gauss elimination.

Example 2: Our goal here was to try and produce a large round-off error that would in turn produce largely varying results. However, as can be seen in the following, that is not the case for smaller systems such at this 3×3 matrix.

$$Ax = \begin{bmatrix} 6 & 5.9 & 0 \\ 5.9 & 6 & 1 \\ 0 & 1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \\ 7 \end{bmatrix} = b \tag{17}$$

By using Gauss-Elimination method we get

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.6667 \\ 0.0000 \\ 1.1667 \end{bmatrix}. \tag{18}$$

Solving the system with the Gauss Seidel method we will obtain an equivalent result. Namely,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1.6667 \\ 0.0000 \\ 1.1667 \end{bmatrix}. \tag{19}$$

5. Conclusion

In our quest to review the many benefits and properties of tridiagonal matrices, we have concluded that not only are these systems ideal in a numerical methods setting, but are also revered in an analytic setting due to their closed-form properties. We have seen that while these systems may arise organically in various applications, it may also be beneficial to transform general matrices into a tridiagonal form. Moreover, we have seen that the computations required for certain properties may be $\mathcal{O}(n)$ while we could expect operation counts on the order of $\mathcal{O}(n^3)$ for general systems.

Of significant importance was our development of the relationship between the recursive formula for the determinant and difference equations, which ultimately led to a closed form solution.

Of particular interest to the members of our group were the tridiagonal applications related to differential equations and have provided an vessel through which further research may be conducted.

References

[1] Geist, George A., Reduction of a General Matrix To Tridiagonal Form, http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=9AACFF293375BE056D58E22866B9BA24?doi=10.1.1.49.5689&rep=rep1&type=pdf

[2] Siegel, David and Dubbs, Clyde, The College Mathematics Journal, Computing Determinants, Vol. 18, No. 1 (Jan., 1987), pp. 48-50 http://www.jstor.org.ezproxy.saintleo.edu/stable/2686317?seq=1#page_scan_tab_contents

[3] King, Frank, University of Cambridge, https://www.cl.cam.ac.uk/teaching/2003/Probability/prob07.pdf

[4] Mallik, Ranjan K., On the Solution of a Second Order Linear Homogeneous Difference Equation with Variable Coefficients, 1996, http://ac.els-cdn.com/S0022247X97956018/1-s2.0-S0022247X97956018-main.pdf?_tid=dabd0d36-07e4-11e6-bbb9-00000aab0f27&acdnat=1461259198_416ddc3545aff22a579306b992bf37c5

[5] Kazdan, Jerry L., University of Pennsylvania, Algebraic Techniques, https://www.math.upenn.edu/~kazdan/AMCS602/tridiag-short.pdf

[6] Nashed, Zuhair M., Scherzer, Otmar, Editors, Contemporary Mathematics, Inverse Problems, Image Analysis, and Medical Imaging, p.128, Jan. 10-13, 2001,

[7] Furati, K.M., Nashed, Zuhair M., Siddiqi, Abul Hasan, Mathematical Models and Methods for Real World Systems (Lecture Notes in Pure and Applied Mathematics), p.115, 2005

[8] Freeden, Willi, Nashed, Zuhair M., Sonar, Thomas, Handbook of Geomathematics, p.785, 2005

[9] Noschese, Silvia and Pasquini, Lionello and Reichel, Lothar, Tridiagonal Toeplitz matrices: Properties and Novel Applications, 2013, http://dx.doi.org/10.1002/nla.1811

[10] Usmani, Riaz A., Inversion of a Tridiagonal Jacobi Matrix, Linear Algebra and its Applications, 1994, http://ac.els-cdn.com/0024379594904146/1-s2.0-0024379594904146-main.pdf?_tid=b0f9ad92-07e4-11e6-a84d-00000aacb35e&acdnat=1461259128_b7bad14589b53bc2d1b9cfeef0f4890f

[11] Kincaid, K., Cheney, W., Numerical Analysis: Mathematics of Scientific Computing, Third Edition

Appendix

Gauss Elimination Matlab Code

```
clear;clc;format short;
a=[5.9999999999 0.9999999999];
d=[6 6 6];
B=[11.9999999999 13.9999999998 6.9999999999];
[M,N]=size(a);
A=zeros(N+1);
for i=1:N
A(i,i+1)=a(i);
A(i+1,i)=a(i);
A(i,i)=d(i);
end
A(N+1,N+1)=d(N+1);
A
B
for i=2:N+1
    A(i,i-1)=0;
    A(i,i)=A(i,i)-(a(i-1))^2/A(i-1,i-1);
    B(i)=B(i)-(a(i-1)*B(i-1))/A(i-1,i-1);
end
A
B
x(N+1)=B(N+1)/A(N+1,N+1);
for i=N:-1:1
    x(i)=(B(i)-a(i)*x(i+1))/A(i,i);
end
x'
A*x'
a=[5.9999999999 0.9999999999];
d=[6 6 6];
B=[11.9999999999 13.9999999998 6.9999999999];
[M,N]=size(a);
A=zeros(N+1);
for i=1:N
A(i,i+1)=a(i);
A(i+1,i)=a(i);
```

```
A(i,i)=d(i);
end
A(N+1,N+1)=d(N+1);
A
B
A*x'
```

Gauss-Seidel Matlab Code

```
clear;clc;format short;
a=[5.99999 0.99999];
d=[6 7 6];
B=[11.99999 13.99998 6.99999];
[M,N]=size(a);
A=zeros(N+1);
for i=1:N
A(i,i+1)=a(i);
A(i+1,i)=a(i);
A(i,i)=d(i);
end
A(N+1,N+1)=d(N+1);
A
B
x(:,1)=[-1 4 10]';
for k=2:500
     x(1,k)=(B(1)-a(1)*x(2,k-1))/d(1);
  for i=2:N
     x(i,k)=(B(i)-a(i-1)*x(i-1,k)-a(i)*x(i+1,k-1))/d(i);
  end
  x(N+1,k)=(B(N+1)-a(N)*x(N,k))/d(N+1);
end
x(:,500)
A*x(:,500)
```