WARSAW UNIVERSITY OF TECHNOLOGY

Numerical Methods 1

Faculty of Mathematics and Information Science

Project 1

- Final Report -

By **Elie SAAD**
May 6, 2020

Contents

1 Problem Statement

Find all principal minors of A using the GE method, where $A \in \mathbb{R}^n \times n$ is a symmetric positive definite tridiagonal matrix. Compare with other methods.

2 Definitions

Definition 2.1 (Principal Minor)**.** Let A be an $n \times n$ matrix. A $k \times k$ submatrix of A formed by deleting $n - k$ rows of A, and the same $n - k$ columns of A, is called **principal submatrix** of A. The determinant of a principal submatrix of A is called a **principal minor** of A [1].

Definition 2.2 (Symmetric Matrix)**.** Let A be an $n \times n$ matrix. A is called a **symmetric matrix** if and only if $A = A^T$ [2].

Definition 2.3 (Positive Definiteness)**.** Let A be an $n \times n$ matrix. A is called a **positive definite** matrix if $x^T Ax > 0$ for every nonzero vector x [2][3].

Definition 2.4 (Tridiagonal Matrix)**.** Let A be an $n \times n$ matrix. A is called a **tridiagonal matrix** if it has nonzero entries only along the main diagonal and the immediate upper and lower diagonal. The immediate upper diagonal is called the super-diagonal and the immediate lower diagonal is called the sub-diagonal [4].

3 Proposed Solutions

3.1 Notation

The following technical notation is used:

- $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite tridiagonal matrix.
- M_k is the set of all principal matrices of A of order k $\forall k = 1, ..., n$.
- Δ_k is the set of all principal minors of M_k of order k.
- $A_i \in M_k$ is the i-th principal matrix $\forall k = 1, ..., n$ and $i = 1, ..., k$.
- $D_i \in \Delta_k$ is the principal minor of A_i $\forall k = 1, ..., n$ and $i = 1, ..., k$. It is calculated by taking the determinant of A_i, so $D_i = det|A_i|$.

- $S \in \mathbb{N}$ is the set of all elements on the diagonal of the triangular matrix $R \in \mathbb{R}^{n \times n}$, where $|S| = n$.
- Σ is the set of all the subsets $S_i \in S$ where $i = 1, ..., 2^n - 1$, and $|\Sigma| = 2^n - 1$.

3.2 Using the Improved GE Method

This is a method where we

1. apply the Improved GE (IGE) method on A,
2. find all possible principal matrices M_k,
3. calculate the principal minor D_i of every A_i.

By applying the IGE method on A, we reduce it to an upper triangular positive definite matrix. We note that applying the GE method on a tridiagonal matrix would be inefficient and thus we will provide a method to improve it for tridiagonal matrices. We know from the definition of a principal minor D_i that it is the determinant of a principal matrix A_i. We will prove that all principal minors of a matrix A are the same after applying the GE method on A. So finding these principal matrices M_k before, or after applying the GE method on A would be the same. Thus, calculating the principal minor D_i of every principal matrix A_i taken after the application of the GE method, would be calculated by taking the product of the trace of A_i. Therefore, because we have upper triangular matrices M_k, calculating the principal minors Δ_k in an algorithmic way would be as simple as iterating through all possible combinations of removing k elements from the traces of the A_i's, then taking their product.

3.2.1 Proving that the Principal Matrices are Unchanged Before and After the GE Method

Theorem 3.1. For a square matrix $A \in \mathbb{R}^{n \times n}$, applying the GE method on A then taking the principal matrices M_k, is the same as taking the principal matrices M_k then applying the GE method to them for $k = 1, ..., n$.

Proof. We will prove this theorem using proof by induction:
For the base case, we will take $n = 2$. Now, for a matrix $A \in \mathbb{R}^{2 \times 2}$ being $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. We apply the GE method to get $A' = \begin{pmatrix} a & b \\ 0 & d - \frac{c}{a}b \end{pmatrix}$. Thus, the principal matrices for A after applying the GE method are $M_1 = a$ or

$M_1 = d - \frac{c}{a}b$, and $M_0 = \begin{pmatrix} a & b \\ 0 & d - \frac{c}{a}b \end{pmatrix}$.
Now taking the principal minors of A before applying the GE method we get $M_1 = a$ or $M_1 = d$ and $M_0 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. After applying the GE method on both M_0 and M_1, we get the exact same results as above.
Now for the induction step to work, we need to assume that all cases up to the case $n = k - 1$ work. So the case for $n = k$ will work as well from the definition of the principal minor M_0 being the matrix itself.
Therefore completing the proof. □

3.2.2 Improving the GE Method for Tridiagonal Matrices

The general linear system of n equations can be written in a matrix form as follows

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{1}$$

But, in our case, the linear system that we use is a banned structure. This means that nonzero elements in the left hand side of (1) appear only around the main diagonal. Such condition may be expresses as follows

$$\text{there exist } m < n, \text{ such that } a_{i,j} = 0 \text{ for } |i - j| > m.$$

The above condition means that nonzero elements reside only on the band of the $2m + 1$ width around the main diagonal. For such systems it is not efficient to employ full version of the Gaussian eliminations as most operations would simply be carried out on the zero elements.
Now we will concentrate on the banded system which have only three diagonals which may be occupied by nonzero entries since this is our case. Such system arises in the numerical methods for solving boundary problem for one dimensional Poisson's equation

$$-y'' = f(x),$$

or more generally second order linear differential equation with variable coefficients

$$p(x)y'' + q(x)y' + r(x)y = f(x),$$

for $x \in I \subset \mathbb{R}$, where I denotes some interval. The tridiagonal system can be written in a matrix form as

$$\begin{bmatrix} d_1 & c_1 & 0 & 0 & ... & 0 \\ a_1 & d_2 & c_2 & 0 & ... & 0 \\ 0 & a_2 & d_3 & c_3 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & a_{n-2} & d_{n-1} & c_{n-1} \\ 0 & 0 & ... & 0 & a_{n-1} & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{bmatrix} \tag{2}$$

For the description of Gaussian elimination we do not need the column of unknowns, so for the sake of simplicity we use the following description of (2)

$$\left[\begin{array}{cccccc|c} d_1 & c_1 & 0 & 0 & ... & 0 & b_1 \\ a_1 & d_2 & c_2 & 0 & ... & 0 & b_2 \\ 0 & a_2 & d_3 & c_3 & ... & 0 & b_3 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & ... & a_{n-2} & d_{n-1} & c_{n-1} & b_{n-1} \\ 0 & 0 & ... & 0 & a_{n-1} & d_n & b_n \end{array}\right] \tag{3}$$

The elimination is now a process which converts this extended matrix into upper triangular one, what means that below the main diagonal are only zeroes. Of course, this process is restricted to such operations that leave the solution of the system (2) unchanged. There are two basic operations that satisfy this requirement and are enough to bring our system to the triangular form: (i) adding any row multiplied by a number to another row; (ii) swapping two rows.

Gaussian elimination method for the tridiagonal system (3) has the following basic structure

$$\begin{cases} R_1 - \text{unchanged}, \\ R_2 := R_2 - \frac{a_1}{d_1} R_1, \\ R_3 := R_3 - \frac{a_2}{d_2} R_2, \\ \vdots \\ R_n := R_n - \frac{a_{n-1}}{d_{n-1}} R_{n-1} \end{cases} \tag{4}$$

where $R_1, ..., R_n$ are the rows of the extended matrix (3). But we should also take into account the fact that rows $R_1, ..., R_n$ contain only two or three nonzero entries so the subtraction of rows in 4) basically is reduced to two values:

$$\begin{cases} d_1, c_1 - \text{unchanged}, \\ d_2 = d_2 - \frac{a_1}{d_1} c_1, b_2 = b_2 - \frac{a_1}{d_1} b_1, \\ d_3 = d_3 - \frac{a_2}{d_2} c_2, b_3 = b_3 - \frac{a_2}{d_2} b_2, \\ \vdots \\ d_n = d_n - \frac{a_{n-1}}{d_{n-1}} c_{n-1}, b_n = b_n - \frac{a_{n-1}}{d_{n-1}} b_{n-1} \end{cases} \tag{5}$$

Using the pseudo code this fragment reads as

```
for $i \leftarrow 1$ to $n$ do
    $d_i \leftarrow d_i - (a_{i-1}/d_{i-1}) * c_{i-1}$
    $b_i \leftarrow b_i - (a_{i-1}/d_{i-1}) * b_{i-1}$
end for
```

Once we have arrived at the upper triangle form we can now easily compute the solutions by the simple backwards substitution - which we will not implement as that is not our goal. But for the case at hand our upper triangle system is not a full one because in each row there are only two nonzero elements d_1, c_1 (except for the last row where there is only d_n) so we can use the restricted form of the general procedure which now will require only one operation per one unknown variable. Specifically, the system looks like below

$$\begin{cases} d_1 x_1 + c_1 x_2 = b_1, \\ d_2 x_2 + c_2 x_3 = b_2, \\ d_3 x_3 + c_3 x_4 = b_3, \\ \vdots \\ d_{n-1} x_{n-1} + c_{n-1} x_n = b_{n-1}, \\ d_n x_n = b_n, \end{cases}$$

and solutions are obtained by the following relations

$$x_n = \frac{b_n}{d_n},$$
$$x_i = \frac{b_i - c_i x_{i+1}}{d_i}, i = n-1, n-2...., 1.$$

In pseudo code we have

```
$x_{n-1} \leftarrow b_{n-1}/d_{n-1}$
for $i \leftarrow n-2$ to 1 do
    $x_i \leftarrow (b_i - c_i * x_{i+1})/d_i$
end for
```

3.2.3 Using the Binary Representation Method to Compute All the Principal Minors

After applying the IGE method discussed in section 3.2.2, we will be left with an upper triangular matrix. And as we have learned before from Linear Algebra, that computing the determinant of a triangular or a diagonal matrix is as simple as calculating its trace - meaning multiplying the elements on the diagonal. And thus - from the definition of the principal minor - we would have computed the principal minor of order n of the upper triangular factor of A

$$\begin{bmatrix} d_1 & a_1 & 0 & 0 & ... & 0 \\ 0 & d_2 & a_2 & 0 & ... & 0 \\ 0 & 0 & d_3 & a_3 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 0 & d_{n-1} & a_{n-1} \\ 0 & 0 & ... & 0 & 0 & d_n \end{bmatrix}. \tag{6}$$

By doing that we would have calculated all the principal minors of order n (since we can have no other combination of crossing out n rows and columns in an $n \times n$ matrix A). Whereas as we reduce the number of eliminated rows and columns of the matrix to compute all the principal minors of the principal matrices of order less than n, things get tricky.

Assume that we are to compute all the principal minors of principal matrices of all orders. We have $2^n - 1$ principal matrices to compute the principal minors for (this fact will be proven in section 3.2.4). Computing the principal minors - as discussed before - constitutes the multiplication of the diagonals of these principal matrices, which in turn constitutes the multiplication of the diagonal of the matrix A with different elements either added or reduced. Thus from now on in this algorithm, we will only consider the diagonals which are elements in the set S, and their different combinations for computing the principal minors. With that information and knowing that we cannot include the same element more than once, we have devised an algorithm where we would represent all the possible combinations of the elements from the set S in binary representation.

Considering the fact that we have n elements in S, the elements may or may not be unique. Then our task is reduced to calculating the product of the elements from all possible subsets S_i's of S.

Let T be the set of of size 2^n-1, of binary numbers, starting from the number

1_2 all the way up to the number $(2^n - 1)_2$. Each number in T is represented by n digits. And each element in S is represented by one digit in every element in T, either 1 or 0. The digit 1 being the representation of this element's inclusion, and 0 being the representation of this element's exclusion. Looping through T and forming the subsets S_i's where $i = 1, ..., 2^n - 1$ from the elements in S with the same indexes as the digit 1's in every element in T, would result in us forming the set Σ of all possible subsets of S. After filling up Σ, calculating all possible principal minors would be as simple as looping through Σ and taking the product of all elements of every S_i.

In pseudo code we have

```
for $i \leftarrow 1$ to $|T|$ do
    $\sigma \leftarrow \emptyset$
    for $j \leftarrow 1$ to $|T_i|$ do
        if $T_{ij} = 1$ then
            $\sigma \leftarrow \sigma \cup S_j$
        end if
    end for
    $\Sigma \leftarrow \Sigma \cup \sigma$
end for
$\Delta \leftarrow \emptyset$
for $i \leftarrow 1$ to $|\Sigma|$ do
    $\Delta \leftarrow \Delta \cup \prod_{j=1}^{|S_i|} \Sigma_{ij}$
end for
```

3.2.4 Proving the Correctness of the Binary Representation Method for Computing All Principal Minors

We will prove the correctness of the algorithm by induction over n.

Theorem 3.2. Let R be an $n \times n$ triangular matrix. Then $|\Delta| = 2^n - 1$ $\forall n$, where Δ denotes the set of all principal minors of R.

Proof. Let $P(n)$ be the predicate "An triangular matrix with dimension n has $2^n - 1$ principal minors".

We will begin with the initial step of $n = 1$, $P(1)$ is true, because the triangular matrix with one element has $2^1 - 1 = 1$ principal minor.

For the inductive step we shall prove $P(k) \implies P(k+1)$. That is, we need to prove that if a triangular matrix of dimension k has $2^k - 1$ principal minors, then a triangular matrix with dimension $k+1$ has $2^{k+1}-1$ principal minors.

Let us assume that for an arbitrary k, any triangular matrix with dimension k has $2^k - 1$ principal minors.

Let T be the matrix such that it has dimension $k+1$, and E_T be the set of all the diagonal elements of T.

Let us enumerate the elements of T and E_T:

$$T = \begin{bmatrix} d_1 & u_{1,2} & u_{1,3} & u_{1,4} & ... & u_{1,k+1} \\ 0 & d_2 & u_{2,3} & u_{2,4} & ... & u_{2,k+1} \\ 0 & 0 & d_3 & u_{3,4} & ... & u_{3,k+1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 0 & d_k & u_{k,k+1} \\ 0 & 0 & ... & 0 & 0 & d_{k+1} \end{bmatrix}, E_T = \{d_1, d_2, ..., d_{k+1}\}.$$

Let

$$S = \begin{bmatrix} d_1 & u_{1,2} & u_{1,3} & u_{1,4} & ... & u_{1,k} \\ 0 & d_2 & u_{2,3} & u_{2,4} & ... & u_{2,k} \\ 0 & 0 & d_3 & u_{3,4} & ... & u_{3,k} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 0 & d_{k-1} & u_{k-1,k} \\ 0 & 0 & ... & 0 & 0 & d_k \end{bmatrix}, E_S = \{d_1, d_2, ..., d_k\}.$$

Then S has dimension k, so S has $2^k - 1$ principal minors according to the inductive hypothesis.

Note that $E_T = E_S \cup \{d_{k+1}\}$, so every subset of E_S is also a subset of E_T. Meaning that any subset of E_T either contains d_{k+1}, or it does not contain d_{k+1}. So, if a subset of E_T does not contain d_{k+1}, then it is also a subset of E_S, and there are 2^k of those subsets. On the other hand, if a subset of E_T does contain the element d_{k+1}, then the subset is formed by including d_{k+1} in one of the 2^k subsets of E_S, so E_T has 2^k subsets containing d_{k+1}.

We have shown that E_T has 2^k subsets containing d_{k+1}, and another 2^k subsets not containing d_{k+1}, so the total number of subsets of E_T is

$$2^k + 2^k = (2)2^k = 2^{k+1}.$$

Therefore T has $2^{k+1}-1$ (excluding the empty set) principal minors. □

3.2.5 Proof of Time Complexity

Let T be the function that is equal to the running time.
First let us consider the running time of binary representation method of the algorithm. Let c_i be the cost of the command execution, where $i = 1, ..., 9$ (since we have 9 command lines in this algorithm).
We denote by t_i the time for each command to execute, where $i = 1, ..., 8$.
Going through the algorithm by command lines calculating the time of each command in relations with its cost we get

$$\begin{aligned} T(n) &= \sum_{i=1}^{n} c_i t_i \\ &= c_1(2^n) + c_2(2^n-1) + c_3(c+1)(2^n-1) + c_4 n(2^n-1) + c_5 n(2^n-1) \\ &\quad + c_6(2^n-1) + c_7 + c_8(n+1) + c_9 n \\ &= c_1 2^n + c_2 2^n - c_2 + c_3 n 2^n - c_3 n + c_3 2^n - c_3 + c_4 n 2^n - c_4 n + c_4 2^n \\ &\quad - c_4 + c_5 n 2^n - c_5 n + c_6 2^n - c_6 + c_7 + c_8 n + c_8 + c_9 n \\ &= n2^n(c_3 + c_4 + c_5) + 2^n(c_1 + c_2 + c_3 + c_4 + c_6) \\ &\quad + n(c_3 + c_4 + c_5 + c_8 + c_9) - (c_2 + c_3 + c_4 + c_6 - c_7) \\ &= n2^n(c^{(1)}) + 2^n(c^{(2)}) + n(c^{(3)}) + c^{(4)}, \end{aligned}$$

where $c^{(i)}$ is a constant with $i = 1, 2, 3, 4$. Thus the time is exponential for this algorithm with $O(n2^n)$.

Now let us consider the running time of the IGE method of the algorithm. Let c_i be the cost of the command execution, where $i = 1, 2, 3$ (since we have 3 command lines in this algorithm).
We denote by t_i the time for each command to execute, where $i = 1, 2, 3$.
Going through the algorithm by command lines calculating the time of each command in relations with its cost we get

$$\begin{aligned} T(n) &= \sum_{i=1}^{n} c_i t_i \\ &= c_1 + c_2(n+1) + c_3 n \\ &= c_1 + c_2 n + c_2 + c_3 n \\ &= n(c_2 + c_3) + c_1 + c_2 \\ &= n(c^{(1)}) + c^{(2)}, \end{aligned}$$

where $c^{(1)}$ and $c^{(2)}$ are constants. Thus the time is linear for this part of the algorithm with $O(n)$.

Therefore we have an exponential time overall with a $O(n2^n)$.

3.3 Using the Modified GE Method

This method is a modification to the previously discussed GE method, where we

1. apply the Modified GE (MGE) method on A,
2. find all possible principal matrices M_k,
3. calculate the principal minor D_i of every A_i.

The only modification done to the previous GE method is in the first step where we apply a MGE method instead of the IGE method. The MGE method takes into consideration the matrix being tridiagonal and symmetric, and just updates the diagonal elements instead of updating the diagonal elements and the elements on the lower diagonal as well. Since the elements on the lower diagonal are not important to the calculation of the determinant, they are kept intact. We will prove that applying the determinant of the MGE is the same as the determinant of the original GE method in section 3.3.2.

After applying the MGE method, we then take A with a modified diagonal and use the Binary Representation Method to compute all principal minors of A, just like we did in the IGE method before.

3.3.1 Modifying the GE Method

The tridiagonal system discussed in section 3.3.1, is written in a matrix form as equation (2). But in our case, the tridiagonal system is also symmetric. This means that the diagonal elements below the main diagonal are equal to the diagonal elements above the main diagonal. This can be written as

$$\begin{bmatrix} d_1 & a_1 & 0 & 0 & ... & 0 \\ a_1 & d_2 & a_2 & 0 & ... & 0 \\ 0 & a_2 & d_3 & a_3 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & a_{n-2} & d_{n-1} & a_{n-1} \\ 0 & 0 & ... & 0 & a_{n-1} & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{bmatrix} \tag{7}$$

For the determinant we do not need the column of unknowns and b, so for the sake of simplicity we use the following description of (7)

$$\begin{bmatrix} d_1 & a_1 & 0 & 0 & ... & 0 \\ a_1 & d_2 & a_2 & 0 & ... & 0 \\ 0 & a_2 & d_3 & a_3 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & a_{n-2} & d_{n-1} & a_{n-1} \\ 0 & 0 & ... & 0 & a_{n-1} & d_n \end{bmatrix}.$$

Thus, calculating the determinant is as simple as applying the IGE method to just the diagonal elements similar to (5) as follows

$$\begin{cases} d_1, a_1 - \text{unchanged}, \\ d_2 = d_2 - \frac{a_1^2}{d_1}, \\ d_3 = d_3 - \frac{a_2^2}{d_2}, \\ \vdots \\ d_n = d_n - \frac{a_{n-1}^2}{d_{n-1}}, \end{cases}$$

where all the a_i's are kept the same and unchanged. Using the pseudo code this fragment reads as

```
for $i \leftarrow 1$ to $n$ do
    $d_i \leftarrow d_i - (a_{i-1}/d_{i-1}) * c_{i-1}$
end for
```

3.3.2 Proof of Correctness

To prove the correctness of the MGE methods we need to show that the determinant after applying the MGE method is equal to the product of the diagonal elements of A after applying the GE method. We will prove that by using induction over n.

Theorem 3.3. Let R be an $n \times n$ tridiagonal and symmetric matrix. Then the determinant of A after applying the GE method is equal to the product of the elements on the diagonal of A after applying the MGE method.

Proof. We will prove this theorem using proof by induction:
For the base case, we will take $n = 2$. Now, for a symmetric and tridiagonal matrix $A \in \mathbb{R}^{2\times 2}$ being $A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$. We apply the GE method to get

$A_{GE} = \begin{pmatrix} a & b \\ 0 & c - \frac{b^2}{a} \end{pmatrix}$. The determinant of A_{GE} - since it is an upper triangular matrix - is the product of the elements on the diagonal of A_{GE}, thus the determinant of A_{GE} is $det(A_{GE}) = a \times c - \frac{b^2}{a}$. We then apply the MGE method to get $A_{MGE} = \begin{pmatrix} a & b \\ b & c - \frac{b^2}{a} \end{pmatrix}$. Taking the product of the diagonal elements of A_{MGE} we get $a \times c - \frac{b^2}{a} = det(A_{GE})$.
Now for the induction step to work, we need to assume that all cases up to the case $n = k - 1$ work. So we have

$$A_{GE} = \begin{bmatrix} d_1 & a_1 & 0 & 0 & ... & 0 \\ 0 & d_2 & a_2 & 0 & ... & 0 \\ 0 & 0 & d_3 & a_3 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 0 & d_{k-2} & a_{k-2} \\ 0 & 0 & ... & 0 & 0 & d_{k-1} \end{bmatrix}, A_{MGE} = \begin{bmatrix} d_1 & a_1 & 0 & 0 & ... & 0 \\ a_1 & d_2 & a_2 & 0 & ... & 0 \\ 0 & a_2 & d_3 & a_3 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & a_{k-3} & d_{k-2} & a_{k-2} \\ 0 & 0 & ... & 0 & a_{k-2} & d_{k-1} \end{bmatrix},$$

where $det(A_{GE}) = \prod_{i=1}^{k-1} d_i$. Thus the case $n = k$ follows from the definition of the GE method. Therefore $det(A_{GE}) = \prod_{i=1}^{k} d_i$. This concludes our proof. □

3.3.3 Proof of Time Complexity

The proof of the time complexity of the MGE method follows from the proof of the time complexity of the previous method. Therefore, the time complexity of the MGE method is an exponential time overall with a $O(n2^n)$.

4 Experimentation

In what follows, we will begin by testing the correctness of both the IGE and the MGE methods discussed in this document. We will test the correctness of the algorithms in a concrete example, first solved by hand, and then applied to both of the algorithms. Then finally we will test the time of both of the algorithms discussed in this document by running them on multiple matrices with increasing dimensions then plotting the time of both algorithms on a graph. All of the testing is done using the matlab code provided with this document.

4.1 Experimentation on the Correctness

Let $A \in \mathbb{R}^{3\times 3}$ be a symmetric, tridiagonal, and positive definite matrix. Assume that

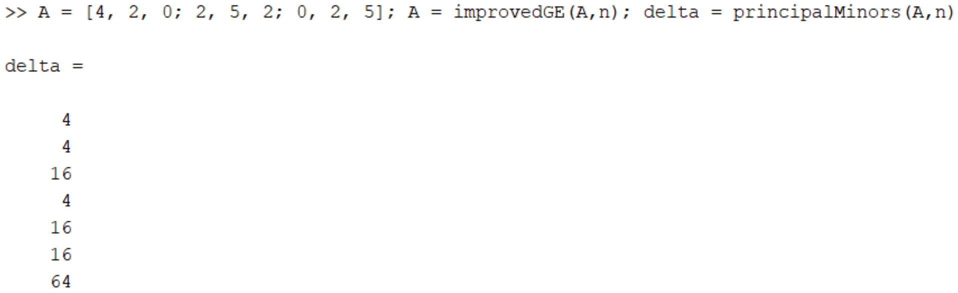
$$A = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 5 & 2 \\ 0 & 2 & 5 \end{bmatrix}.$$

Thus, applying the GE method on paper will give us

$$A_{GE} = \begin{bmatrix} 4 & 2 & 0 \\ 0 & 4 & 2 \\ 0 & 0 & 4 \end{bmatrix}.$$

So calculating the principle minors of A_{GE} becomes easy now. So the principle minors of A_{GE} are (including repetitions) $\Delta = \{4, 4, 4, 4 \times 4, 4 \times 4, 4 \times 4, 4 \times 4 \times 4\} = \{4, 4, 4, 16, 16, 16, 64\}$

Applying the IGE gives us the following

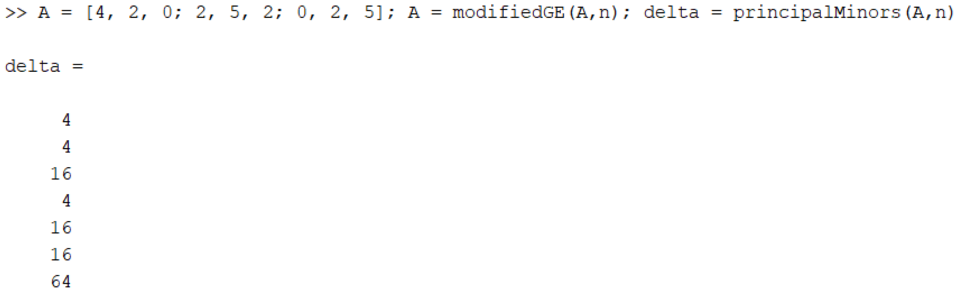


```
>> A = [4, 2, 0; 2, 5, 2; 0, 2, 5]; A = improvedGE(A,n); delta = principalMinors(A,n)

delta =

     4
     4
    16
     4
    16
    16
    64
```

Applying the MGE gives us the following



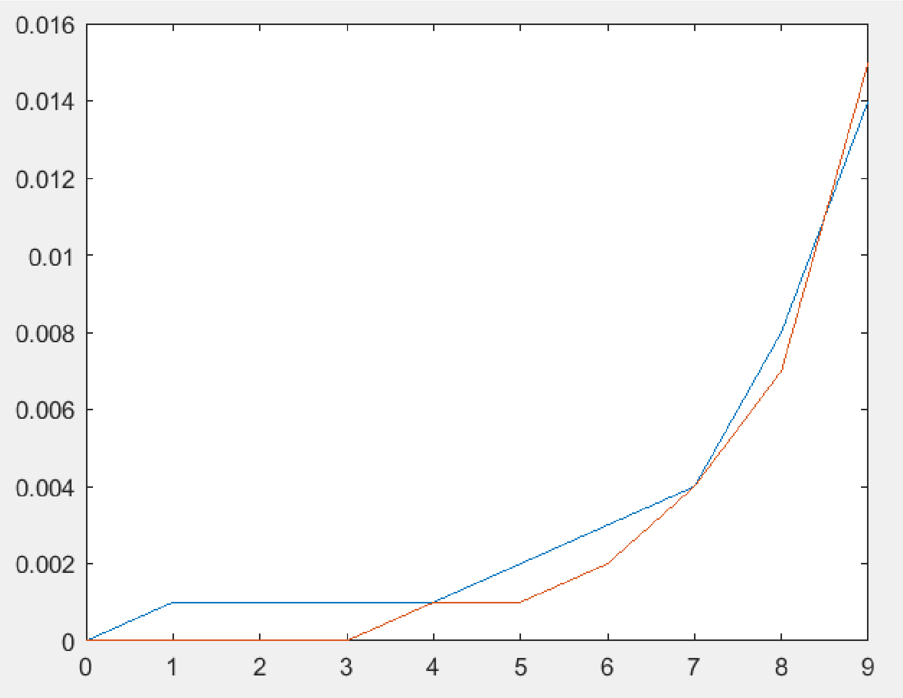
```
>> A = [4, 2, 0; 2, 5, 2; 0, 2, 5]; A = modifiedGE(A,n); delta = principalMinors(A,n)

delta =

     4
     4
    16
     4
    16
    16
    64
```

4.2 Experimentation on the Time

We have run the algorithms consecutively ten times, each time running them on a matrix that is one dimension larger than the one before, starting with the dimension two. We have obtained the following graph where the blue line is the time for the IGE and the red line is the time for the MGE



The graph shows that the time is exponential, just how we have proven it to be for both the IGE and the MGE methods.

5 Conclusion

In this document we have devised two ways of finding all of the principal minors of a tridiagonal, symmetric, and positive definite matrix A. The first being the Improved Gaussian Elimination method, and the second being the Modified Gaussian Elimination method. We have proven that both of those methods are valid methods of finding all of the principal minors of A. We have also proven that both of them have the same time complexity of time $O(n2^n)$ where n is the dimension of A. And finally we have concluded with experimentation that shows that both methods work in practice as well. This document has been accompanied with five files, being

1. ”main.m” is the file that we have used to display the time complexity,

2. ”triPosDef.m” is the file that creates a tridiagonal positive definite

symmetric matrix for our tests,

3. ”improvedGE.m” is the file that is an implementation of the IGE method described in this document,

4. ”modifiedGE.m” is the file that is an implementation of the MGE method described in this document,

5. ”principalMonis.m” is the file that is an implementation of the Binary Representation method of computing the principal minors discussed in this document.

References

[1] Massachusetts Institute of Technology, *Handout on Second Order Conditions*, 12/10/2004, ⟨http://web.mit.edu/14.102/www/notes/soc.pdf⟩, [7/4/2020], p. 1.

[2] Massachusetts Institute of Technology, *Symmetric matrices and positive definiteness*, Fall 2011, ⟨https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/positive-definite-matrices-and-applications/symmetric-matrices-and-positive-definiteness/MIT18_06SCF11_Ses3.1sum.pdf⟩, [7/4/2020], pp. 1, 2.

[3] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 3rd Edition, (American Mathematical Society, Providence, RI, 2002), p. 145.

[4] M. Yano, J. D. Penn, G. Konidaris, A. T. Patera, *DRAFT V2.1 From Math, Numerics, & Programming (for Mechanical Engineers)*, https://ocw.mit.edu/ans7870/2/2.086/S13/MIT2_086S13_Unit5_Textbook.pdf August 2003, [7/4/2020], pp. 399, 400.