

Exercise P23: A circulant matrix is one where constant diagonals 'wrap around':

$$C = \begin{bmatrix} c_1 & c_m & \dots & c_3 & c_2 \\ c_2 & c_1 & c_m & & c_3 \\ \vdots & c_2 & c_1 & \ddots & \vdots \\ c_{m-1} & & \ddots & \ddots & c_m \\ c_m & c_{m-2} & \dots & c_2 & c_1 \end{bmatrix}$$

Each entry of $C \in \mathbb{C}^{m \times m}$ is thus a function of the row/column index difference:

$$C_{jk} = \begin{cases} c_{j-k+1} & j \geq k \\ c_{m+j-k+1} & j < k \end{cases}$$

Here c_1, \dots, c_m are the entries of a column vector, namely the first column of C . Specifying the first column of a circulant matrix describes it completely.

Here is an extraordinary fact about circulant matrices: Every circulant matrix has a complete set of eigenvectors that are known in advance, without the eigenvalues. Specifically, define $f_k \in \mathbb{C}^m$ by,

$$(f_k)_j = \exp(-i(j-1)(k-1)\frac{2\pi}{m}) = e^{-i2\pi(k-1)(j-1)/m} \quad (2)$$

where, as usual $i = \sqrt{-1}$. These vectors are waves, i.e. combinations of familiar sines and cosines.

After some warm-up exercises you will show in part (e) that $Cf_k = \lambda_k f_k$.

- a. Define the periodic convolution $u * w \in \mathbb{C}^m$ of vectors $u, w \in \mathbb{C}^m$ by

$$(u * w)_j = \sum_{k=1}^m u_{\mu(j,k)} w_k$$

where,

$$\mu_{(j,k)} = \begin{cases} j - k + 1 & j \geq k \\ m + j - k + 1 & j < k \end{cases}$$

Show that $u * w = w * u$.

Solution:

Suppose the following periodic convolution for some row j , and let the $|$ denote where we 'wrap-around' indices with respect to the function μ ,

$$(u * w)_j = \sum_{k=1}^m u_{\mu(j,k)} w_k = u_j w_1 + u_{j-1} w_2 + u_{j-3} w_3 \dots u_1 w_j | + u_m w_{j+1} + u_{m-1} w_{j+2} \dots u_{j+1} w_m.$$

Now consider $(w * u)_j$,

$$(w * u)_j = \sum_{k=1}^m w_{\mu(j,k)} u_k = w_j u_1 + w_{j-1} u_2 + w_{j-3} u_3 \dots w_1 u_j + w_m u_{j+1} + w_{m-1} u_{j+2} \dots w_{j+1} u_m.$$

Note that if we reverse the indices of the partial sum before and after the $|$ we get the same sum as $(u * w)_j$. Said another way we can reverse the indices of each partial sum to get the following,

$$\begin{aligned} \sum_{k=1}^j u_{j-k+1} w_k &= \sum_{k=1}^j w_{j-k+1} u_k, \\ \sum_{k=j+1}^m u_{m+j-k+1} w_k &= \sum_{k=j+1}^m w_{m+j-k+1} u_k. \end{aligned}$$

Finally by splitting up the periodic convolution and applying substitution we get the following,

$$\begin{aligned} (u * w)_j &= \sum_{k=1}^m u_{\mu(j,k)} w_k \\ &= \sum_{k=1}^j u_{j-k+1} w_k + \sum_{k=j+1}^m u_{m+j-k+1} w_k, \\ &= \sum_{k=1}^j w_{j-k+1} u_k + \sum_{k=j+1}^m w_{m+j-k+1} u_k, \\ &= \sum_{k=1}^m w_{\mu(j,k)} u_k, \\ &= (w * u)_j. \end{aligned}$$

- b. Show that $Cu = v * u$ if C is a circulant matrix and v is the first column of C .

Solution:

Suppose that $C \in \mathbb{C}^{m \times m}$ is a circulant matrix where v is the first column and let u be some vector in \mathbb{C}^m . Consider the convolution $v * u$ and by part a, it's j th entry is given by,

$$(v * u)_j = \sum_{k=1}^m v_{\mu(j,k)} u_k = v_j u_1 + v_{j-1} u_2 + v_{j-3} u_3 \dots v_1 u_j + v_m u_{j+1} + v_{m-1} u_{j+2} \dots v_{j+1} u_m.$$

Now consider the j th entry in the Cu matrix-vector product, and note that since C is a circular matrix the j th row is defined by the row/column index difference defined in the first part of this problem, so

$$(Cu)_j = \sum_{k=1}^m C_{jk} u_k = \sum_{k=1}^m v_{\mu(j,k)} u_k = (v * u)_j.$$

- c. Show that the vectors defined in (2) are orthogonal.

Solution:

Consider two arbitrary eigenvectors f_u, f_v defined by (2). By (2) we can define the innerproduct between them to be,

$$f_u^* f_v = \sum_{j=1}^m (f_u)_j (f_v)_j.$$

Applying our definition of the eigenvector entries in (2) and simplifying we get the following,

$$\begin{aligned} f_u^* f_v &= \sum_{j=1}^m (f_u)_j (f_v)_j, \\ &= \sum_{j=1}^m e^{\frac{-i2\pi(u-1)(j-1)}{m}} e^{\frac{-i2\pi(v-1)(j-1)}{m}}, \\ &= \sum_{j=1}^m e^{\frac{(-i2\pi(u-1)(j-1)) + (-i2\pi(v-1)(j-1))}{m}}, \\ &= \sum_{j=1}^m e^{-i2\pi(j-1) \frac{(u-1)+(v-1)}{m}}, \\ &= \sum_{j=1}^m e^{-i2\pi \frac{(u+v-2)}{m} (j-1)}, \\ &= \sum_{j=0}^{m-1} e^{-i2\pi \frac{(u+v-2)}{m} j}. \end{aligned}$$

Note that we have a finite geometric sum, writing it in a closed form results in,

$$\begin{aligned} f_u^* f_v &= \frac{1 - e^{-i2\pi \frac{(u+v-2)}{m} m}}{1 - e^{-i2\pi \frac{(u+v-2)}{m}}}, \\ &= \frac{1 - e^{-i2\pi(u+v-2)}}{1 - e^{-i2\pi \frac{(u+v-2)}{m}}}. \end{aligned}$$

Writing the numerator in-terms of sin and cos we get the following,

$$f_u^* f_v = \frac{1 - \cos(2\pi(u+v-2)) + i \sin(2\pi(u+v-2))}{1 - e^{-i2\pi \frac{(u+v-2)}{m}}}.$$

Note that the sin term will always go to zero, since it will always have as an input a multiple of 2π . Similarly the cos term will return a 1. Therefore the numerator goes to 0 and we get the f_u and f_v are orthogonal.

- d. For $m = 20$, use Matlab to plot the real parts of the vectors f_1, \dots, f_5 together in a single figure.

Solution:

It seems as though I was unable to generate the right eigenvalues with the following script which implements the given formula,

Code:

```
eigenVectors = zeros(20,5);
for k = 1:5
    for j = 1:20
        eigenVectors(j,k) = exp(-1i*2*pi*(k-1)*(j-1)*(1\20));
    end
end
```

Console:

```
>> real(eigenVectors)
```

```
ans =
```

```

1      1      1      1      1
1      1      1      1      1
1      1      1      1      1
1      1      1      1      1
```

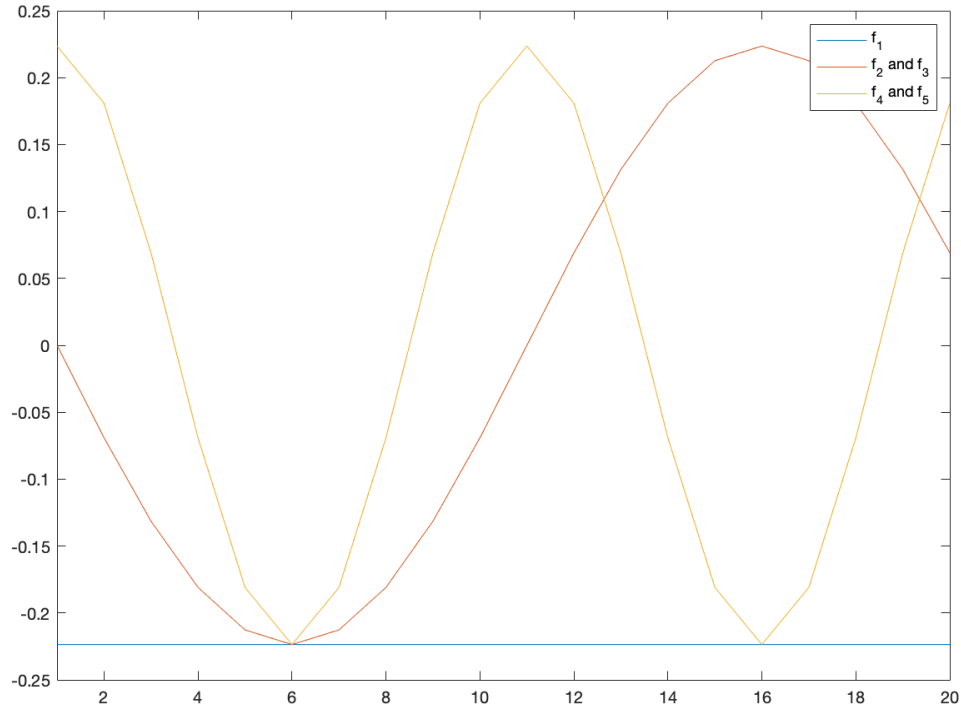
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Clearly the real parts of these eigenvectors will plot a straight line, nothing resembling a discretizing wave. I'm not really sure what is wrong with my code so I'll continue by generating a circulant matrix then calling `eig()` on it to get usable vectors,

Console:

```
A = circu([1:20]);
[vec, val] = eig(A);
hold on
plot([1:20], real(vec(:,1)))
plot([1:20], real(vec(:,2)))
plot([1:20], real(vec(:,3)))
plot([1:20], real(vec(:,4)))
plot([1:20], real(vec(:,5)))
```

Figure 1: Real Parts of Circulant Matrix Eigenvectors



- e. For the general circulant matrix C in (1) above, give a formula for the eigenvalues Λ_k , in terms of the entries c_1, \dots, c_m . that is, show via by-hand calculation that,

$$Cf_k = \lambda_k f_k.$$

Solution:

Recall from part *b* that the matrix-vector product Cfi can be rewritten as the convolution $c * fi$ where c is the first column of C . Doing so we get the following,

$$(Cfi)_j = (c * fi)_j = (fi * c)_j = \sum_{k=1}^j fi_{j-k+1}c_k + \sum_{k=j+1}^m fi_{m+j-k+1}c_k.$$

Recall the definition of the fk_j term from (2), we can factor an fi_j term from each

sum we get the following,

$$(Cfi)_j = fi_j \left(\sum_{k=1}^j \frac{fi_{j-k+1}}{fi_j} c_k + \sum_{k=j+1}^m \frac{fi_{m+j-k+1}}{fi_j} c_k \right).$$

Applying our definition of each term of the eigenvector from (2) we get,

$$\begin{aligned} (Cfi)_j &= fi_j \left(\sum_{k=1}^j \frac{fi_{j-k+1}}{fi_j} c_k + \sum_{k=j+1}^m \frac{fi_{m+j-k+1}}{fi_j} c_k \right). \\ &= fi_j \left(\sum_{k=1}^j \frac{e^{-i2\pi(i-1)(j-k+1-1)(1/m)}}{e^{-i2\pi(i-1)(j-1)(1/m)}} c_k + \sum_{k=j+1}^m \frac{e^{-i2\pi(i-1)(m+j-k+1-1)(1/m)}}{e^{-i2\pi(i-1)(j-1)(1/m)}} c_k \right). \\ &= fi_j \left(\sum_{k=1}^j e^{-i2\pi(1/m)(i-1)(j-k-j+1)} c_k + \sum_{k=j+1}^m e^{-i2\pi(1/m)(i-1)(m+j-k-j+1)} c_k \right), \\ &= fi_j \left(\sum_{k=1}^j e^{-i2\pi(1/m)(i-1)(1-k)} c_k + \sum_{k=j+1}^m e^{-i2\pi(1/m)(i-1)(m-k+1)} c_k \right), \end{aligned}$$

... I got very stuck here. This felt right but I couldn't get the sum back together.

- f. Download the `circu.m` matlab script, and generate a circulant matrix C with the first column consisting of 20 random numbers of your choice. Use the result of (e) to compute the eigenvalues λ_k , and compare these against the result of `eig()`. Also generate f_5 from (2) and verify that $cf_5 = \Lambda_5 f_5$ to high accuracy; use a vector norm.

Solution:

I'm not sure what was going on with part d and e. I was able to play around with the `circu.m` script and some formulas I found online for how the eigenvalues and eigenvectors should be calculated.

Exercise 22.1: Show that for Gaussian elimination with partial pivoting applied to any matrix $A \in \mathbb{C}^{m \times m}$, the growth factor (22.2) satisfies $\rho \leq 2^{m-1}$.

Solution:

Suppose a matrix $A \in \mathbb{C}^{m \times m}$, and recall that partial pivoting will not change $\max_{i,j} |a_{i,j}|$. Applying the first step (or L factor) in gaussian elimination to the matrix A we get a reduced matrix A_1 with zeroes below the diagonal of the first column whose entries can be described by the following,

$$a_{i,j}^1 = a_{i,j} - \frac{a_{i,1}}{a_{1,1}} a_{1,j}.$$

Recall that partial pivoting ensures us that,

$$\frac{a_{i,1}}{a_{1,1}} \leq 1.$$

Thus by substitution we obtain the following inequality,

$$|a_{i,j}^1| = |a_{i,j} - \frac{a_{i,1}}{a_{1,1}} a_{1,j}| \leq |a_{i,j} + \frac{a_{i,1}}{a_{1,1}} a_{1,j}| \leq |a_{i,j} + a_{1,j}| \leq 2 \max_{i,j} |a_{i,j}|.$$

We will continue via induction on the k th iteration of gaussian elimination with partial pivoting. Let $a_{i,j}^k$ denote the entries of the reduced matrix A_k obtained through k iterations of gaussian elimination. Following the last inequality we get,

$$a_{i,j}^{k+1} = a_{i,j}^k - \frac{a_{i,k+1}^k}{a_{k+1,k+1}^k} a_{i,k+1}^k.$$

Again since the matrix has already been pivoted we know the following,

$$\frac{a_{i,k+1}^k}{a_{k+1,k+1}^k} \leq 1.$$

Therefore, similarly to the base case, by substitution it follows that,

$$|a_{i,j}^{k+1}| \leq |a_{i,j}^k + a_{i,k+1}^k| \leq 2 \max_{i,j} |a_{i,j}^k|.$$

Since there are $m - 1$ iterations to perform gaussian elimination on an $m \times m$ matrix we get that $|a_{i,j}^{m-1}| = |u_{i,j}|$. From our induction argument we can chain the inequalities to get,

$$|u_{i,j}| \leq 2 \max_{i,j} |a_{i,j}^{m-2}| \leq 2^2 \max_{i,j} |a_{i,j}^{m-3}| \leq \dots \leq 2^{m-1} |a_{i,j}|.$$

Which gives us that,

$$\frac{|u_{i,j}|}{|a_{i,j}|} \leq 2^{m-1}.$$

Exercise 22.2: Experiment with solving 60×60 systems of equation $Ax = b$ by Gaussian elimination with partial pivoting, with A having the form in 22.4. Do you observe that the results are useless because of the growth factor of order 2^{60} ? At your first attempt you may not observe this, because the integer entries of A may prevent any rounding errors from occurring. If so, find a way to modify your problem slightly so that the growth factor is the same or nearly so and catastrophic rounding error really do take place.

Solution:

The following matlab script constructs a 60×60 matrix with the form found in 22.4. We then performed gaussian elimination on 20 random vectors, each with a maximum range double the previous one. This was done in order to observe a wider range of x solutions since the text mentions that a growth factor on the order of 2^{60} corresponds to a loss of about 60 bits of precision. We then computed the solution using the \backslash operator, although any of our previous methods for solving systems should do fine as well.

Code:

```

norms = [];
maxes = [];

%Generate Matrix
A = ( tril( zeros(60) - ones(60)*(1 - 1*10^-14), -1) + eye(60));
A(:,60) = ones(1,60);

%LU Decomposition
[L,U,P] = lu(A);

%Compute Growth factor
growthfactor = max(abs(U),[], 'all')/max(abs(A),[], 'all');

for i = 1:20
    %Generate Random Vector
    b = 2^i*rand(60,1);

    %Forcing Gaussian Elimination
    y = forwardsub(L,P*b);
    x = backsub(U,y);

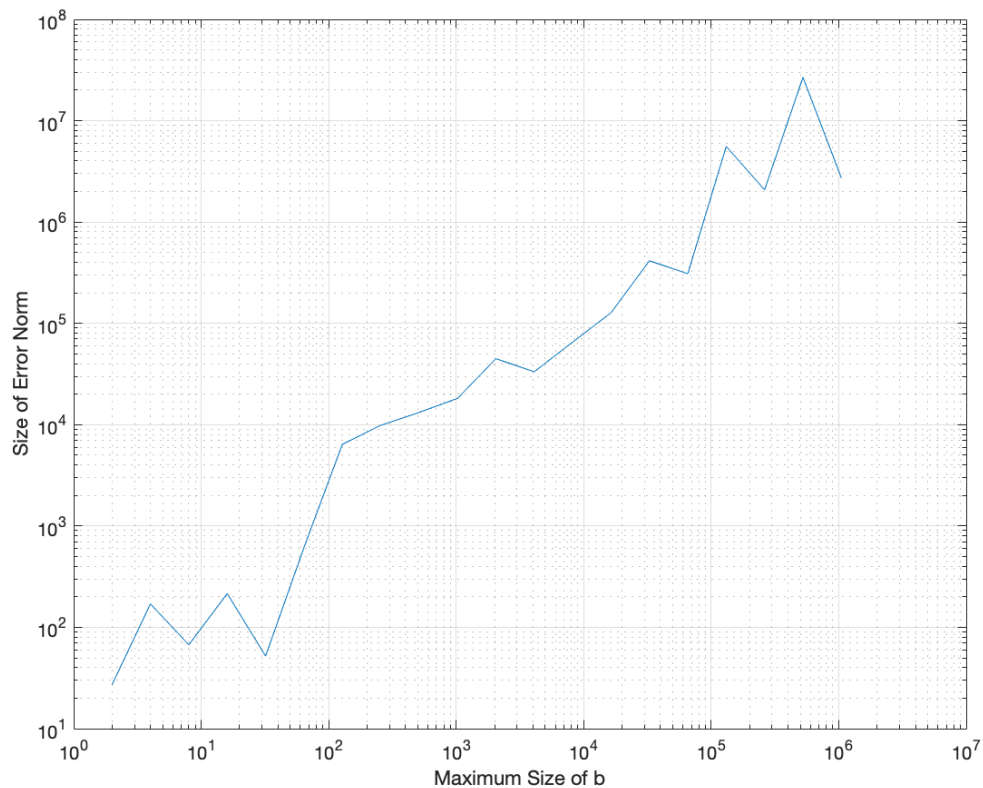
    %Solving
    xx = A\b;

    %Computing norms, and maximum difference
    norms = [norms; norm(abs(xx - x))];
    maxes = [maxes; max(abs(xx-x))];
end

```

```
%Plotting results
loglog(2.^[1:20], norms)
grid on
xlabel('Maximum Size of b')
ylabel('Size of Error Norm')
```

Figure 2: Maximum Input Size vs Error



As we can see, when given digits to loose this method of solving this specific system will loose them.

Exercise 23.2: Use the proof of Theorem 16.2 as a guide, derive Theorem 23.3 from Theorems 23.2 and 17.2.

Solution:

Let $A \in \mathbb{C}^{m \times m}$ be hermitian positive definite, and let a Cholesky factorization of A be computed by algorithm 23.1 on a machine satisfying 13.1 and 13.7. By Theorem 23.2 this Cholesky factorization exists such that the computed \tilde{R} satisfies,

$$\tilde{R}^* \tilde{R} = A + \delta A,$$

where,

$$\frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}}).$$

Recall that Theorem 17.2 states that solving systems via backwards substitution (and forwards substitution) is backwards stable such that the computed solution $\tilde{x} \in \mathbb{C}^m$ satisfies,

$$(R + \delta R)\tilde{x} = b,$$

for some upper(lower)-triangular matrix $\delta R \in \mathbb{C}^{m \times m}$ with the property that,

$$\frac{\|\delta R\|}{\|R\|} = O(\epsilon_{\text{machine}}).$$

Now consider solving the hermitian positive definite system $Ax = b$ via Cholesky factorization. Substituting our computed Cholesky factorization and applying 17.2 to both \tilde{R}^* and \tilde{R} factors we get the following,

$$b = (\tilde{R}^* + \delta \tilde{R}^*)(\tilde{R} + \delta \tilde{R})x = (\tilde{R}^* \tilde{R} + \tilde{R}^* \delta \tilde{R} + \delta \tilde{R}^* \tilde{R} + \delta \tilde{R}^* \delta \tilde{R})x$$

From Theorem 23.2 we know that $\tilde{R}^* \tilde{R} = A + \delta A$ and therefore,

$$b = (A + \delta A + \tilde{R}^* \delta \tilde{R} + \delta \tilde{R}^* \tilde{R} + \delta \tilde{R}^* \delta \tilde{R})x.$$

Which can be written in the form,

$$b = (A + \Delta A)x$$

where $\Delta A = \delta A + \tilde{R}^* \delta \tilde{R} + \delta \tilde{R}^* \tilde{R} + \delta \tilde{R}^* \delta \tilde{R}$.

Now we must show that each term in the perturbation is small relative to A . First consider $\tilde{R}^* \delta \tilde{R}$ and by theorem 17.2 we know that,

$$\frac{\|\tilde{R}^* \delta \tilde{R}\|}{\|A\|} \leq \frac{\|\tilde{R}^*\|}{\|A\|} \|\delta \tilde{R}\| = O(\epsilon_{\text{machine}}).$$

Similarly we get that,

$$\frac{\|\delta \tilde{R}^* \tilde{R}\|}{\|A\|} \leq \|\delta \tilde{R}^*\| \frac{\|\tilde{R}\|}{\|A\|} = O(\epsilon_{\text{machine}}).$$

Applying 17.2 twice we get,

$$\frac{\|\delta \tilde{R}^* \delta \tilde{R}\|}{\|A\|} \leq \frac{\|\delta \tilde{R}^*\| \|\delta \tilde{R}\|}{\|A\|} = O(\epsilon_{\text{machine}}^2).$$

Thus we know that,

$$\frac{\|\delta A\|}{\|A\|} \leq \frac{\|\delta A\|}{\|A\|} + \frac{\|\tilde{R}^* \delta \tilde{R}\|}{\|A\|} + \frac{\|\delta \tilde{R}^* \tilde{R}\|}{\|A\|} + \frac{\|\delta \tilde{R}^* \delta \tilde{R}\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

Therefore solving linear systems with Cholesky factorization is backwards stable.

Exercise 23.3: Reverse Software Engineering of ”\”. the following Matlab session records a sequence of tests of the elapsed times for various computation on a workstation manufactured in 1991. For each part, try to explain: (i) Why was this experiment carried out? (ii) Why did the result come out as it did? Your answers should refer to formulas from the text for flop counts. The Matlab queries `help chol` and `help slash` may help in your detective work.

- a. (i) The test constructs a positive definite 200x200 matrix and forms a system with a randomly generated b vector. Following the ”\” flowchart we know that this system is solved via Cholesky factorization. (ii) 23.4 states that Cholesky factorization takes on the order of $\frac{1}{3}m^3$ flops.
- b. (i) This test involves solving the same system from the previous test. I am not really sure why this experiment was carried out, maybe they wanted to run it again to get another reading. (ii) The results seem to be inline with the previous test, perhaps slightly faster from cpu instruction management.
- c. (i) In this test, we take A from the previous test and form $A2$ by dividing the last term in the first column by 2. Therefore $A2$ is no longer symmetric. Following the ”\” flowchart we can see that this system is solved via LU factorization. (ii) Given that 20.8 states that LU factorization takes on the order of $\frac{2}{3}m^3$ a benchmark of 2.0361 seconds (2 times Cholesky) is expected.
- d. (i) This test we form $A3$ by subtracting .9 times the smallest eigenvalue of A from the diagonal. The matrix is still positive definite, maybe this test was used to measure the performance of Cholesky when the matrix was only just barely positive definite. (ii) I think an edge case, where the matrix only barely fits the requirements for Cholesky would be reflected in an error/sensitivity analysis not a performance benchmark. The Algorithm will still perform on the order of $\frac{1}{3}m^3$ flops and take about the same time.

- e. (i) In this test A_4 is formed by subtracting 1.1 times the smallest eigenvalue of A from the diagonal. Here the matrix is not positive definite, so this test is for when Cholesky cannot be used. (ii) As was discussed in the class discord it is likely that this test was used to benchmark the performance of a 'fail Cholesky then perform LU' subroutine (despite the modern flowchart leading to LDL solver) which would explain where a time of 2.9624 comes from. 1 second for Cholesky and 2 for LU.
- f. (i) In this test an upper triangular matrix, A_5 is taken from A . This test is likely used to test a triangular solver like forwards and back substitution. (ii) The text states that the work required for back substitution is on the order of m^2 flops. A time like .1261 that is significantly less than the other test is expected.
- g. (i) In this test we form A_6 by letting the lower leftmost entry of A_5 equal to its upper rightmost entry. This matrix is no longer triangular or hermitian, so Matlab must perform an LU factorization. Comparing it to test 2, I think this test was used to compare the LU performance between a sparse almost symmetric matrix and a full almost symmetric matrix. (ii) Around 2 seconds for an LU factorization is roughly in line with what we saw for test c.

Exercise 24.1: For each of the following statements, prove that it is true or give an example to show it is false. Throughout, $A \in \mathbb{C}^{m \times m}$ unless otherwise indicated, and 'ew' stands for eigenvalue. The corresponding abbreviation for eigenvector is 'ev'.

- a. If λ is an ew of A and $\mu \in \mathbb{C}$, then $\lambda - \mu$ is an ew of $A - I\mu$.

Solution:

Suppose that λ is an ew of A and $\mu \in \mathbb{C}$. Now consider the following characteristic equation,

$$\det(I(\lambda - \mu) - (A - I\mu)) = \det(I\lambda - I\mu - A + I\mu) = \det(I\lambda - A).$$

By definition we know that,

$$\det(I\lambda - A) = 0.$$

.

- b. If A is real and λ is an ew of A , then so is $-\lambda$.

Solution:

This statement is false. Consider the following matrix,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

It is a real matrix whose eigenvalues are all positive and contained on the diagonal.

- c. If A is real and λ is an ew of A then so is $\bar{\lambda}$.

Solution:

Suppose A is real and λ is an ew of A . Since λ is an ew it must be a root to the characteristic polynomial. If λ is real, then $\lambda = \bar{\lambda}$ it is trivially an ew. If λ is imaginary then $\bar{\lambda}$ must also be a root to the characteristic polynomial, by the complex conjugate root theorem. Therefore $\bar{\lambda}$ is an ew.

- d. If λ is an ew of A and A is nonsingular, then λ^{-1} is an ew of A^{-1} .

Solution:

Suppose λ is an ew of A and A is nonsingular. Let v be an ev such that,

$$Av = \lambda v.$$

Since A is nonsingular, apply A^{-1} to both sides,

$$A^{-1}Av = A^{-1}\lambda v,$$

$$v = \lambda A^{-1}v,$$

$$\lambda^{-1}v = A^{-1}v.$$

Therefore λ^{-1} must be an ew.

- e. If all the ew's of A are zero, then $A = 0$.

Solution:

This statement is false. Any non-zero triangular matrix with zeros on the diagonal works as a counterexample.

- f. If A is hermitian and λ is an ew of A , then $|\lambda|$ is a singular value of A .

Solution:

This statement is proven by Theorem 5.5 in the text.

g. If A is diagonalizable and all its ew's are equal, then A is a diagonal.

Solution:

Suppose that A is diagonalizable and all its ew's are equal to λ . By definition there exists an X such that following is true,

$$A = X\Lambda X^{-1}$$

Since all the ew are the same we know that,

$$A = XI\lambda X^{-1},$$

$$A = \lambda XX^{-1},$$

$$A = \lambda I.$$

Thus A must be diagonal.