Problem 0.1. Watkins 4.1.6

Solution. The $i$-th column of $AV$ is $A$ times the $i$-th column of $V$, $v_i$. The $i$-th column of $U\Sigma$ is similarly $U$ times the $i$-th column of $\Sigma$. But the $i$-th column of $\Sigma$ is zero except for $\sigma_i$ in the $i$-th row. Thus, the $i$-th column of $U\sigma$ is $\sigma_i u_i$. Thus, as desired, $Av_i = \sigma_i u_i$. A similar argument works for the other equations.

If we have $AV = U\Sigma$, since $V$ is orthogonal, it is easy to solve this for $A$ by multiplying both sides by $V^T$ on the right. Thus, $A = U\Sigma V^T$. □

Problem 0.2. Watkins 4.2.5

Solution. We calculate

$$\frac{\|Av_n\|_2}{\|v_n\|_2} = \frac{\|Av_n\|_2}{\|v_n\|_2} = \sigma_n \frac{\|v_n\|_2}{\|v_n\|_2} = \sigma_n,$$

and so we know that

$$\minmag(A) = \min \frac{\|Ax\|_2}{\|x\|_2} \leq \sigma_n,$$

since we found a specific vector which made the fraction at least that small.

Suppose that $x = \sum c_i v_i$, which we can do since the $v_i$ are a basis for $\mathbb{R}^n$. Notice that $\|x\|_2 = \sum |c_i|$, since the $v_i$ are orthonormal. Then

$$\|Ax\|_2 = \|\sum c_i Av_i\|_2$$

$$= \|\sum c_i \sigma_i u_i\|_2$$

$$\geq \|\sum c_i \sigma_i u_i\|_2$$

$$\geq |\sigma_n| \|\sum c_i u_i\|_2$$

$$\geq \sigma_n \sqrt{\sum |c_i|^2}$$

$$= \sigma_n \|x\|_2.$$

The third line is true because $\sigma_n$ is the smallest singular value. The next to last line holds because the $u_i$ are orthonormal, and so we use the Pythagorean theorem.

This inequality shows that $\|Ax\|_2 / \|x\|_2 \geq \sigma_n$, and so $\minmag(A)$ is both bigger and smaller than $\sigma_n$. Thus $\minmag(A) = \sigma_n$. Notice that we showed that $v_n$ already achieved that minimum magnification. □
Problem 0.3. Watkins 4.2.14

Solution. Suppose $A = U\Sigma V^T$. Let $A_\epsilon = U(\Sigma + \epsilon I)V^T$, where $I$ is the identity matrix (or the equivalent when $A$ is not square.) Notice that $A_\epsilon$ was defined in its SVD form. Even the diagonal elements of $\Sigma + \epsilon I$ are in the correct order. Thus, since none of those singular values can be zero, $A_\epsilon$ has full rank.

Notice that $A - A_\epsilon$ is equal to $U(\epsilon I)V^T$. This is, again, in SVD form, and so the largest singular value of $A - A_\epsilon$ is $\epsilon$. Thus $\|A - A_\epsilon\|_2 = \epsilon$.

Problem 0.4. Watkins 5.2.2

Solution. If $A$ is nonsingular, then $Ax = 0$ has a unique solution. That solution must be 0, and so there cannot be a nonzero vector $x$ such that $Ax = 0x = 0$. Thus 0 cannot be an eigenvalue of $A$.

If 0 is not an eigenvalue of $A$, then there is no vector $x \neq 0$ such that $Ax = 0x = 0$. Thus the only solution to $Ax = 0$ is $x = 0$, and so $A$ is nonsingular.

Problem 0.5. Watkins 5.2.14a,b

Solution. (a) Since the characteristic equation is $\lambda^2 = 0$, the only eigenvalue is $\lambda = 0$. To be an eigenvector, then, we need $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$. It is easy to see that this means that $y = 0$. But then the only eigenvector is $[1,0]$ and its multiples.

(b) The matrix is $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

Problem 0.6. Watkins 5.2.17

Solution. Using the geometric SVD theorem (Theorem 4.1.3), It is easy to see that

$$A^T A v_i = A^T \sigma_i u_i = \sigma^2_i v_i,$$

and similarly, $AA^T u_i = \sigma_i^2 u_i$. Thus they are eigenvectors, associated with the right eigenvalues. They are linearly independent because the columns of an orthogonal matrix are orthonormal, and thus linearly independent.

Problem 0.7. Watkins 5.3.6

Solution. The characteristic polynomial is

$$(9 - \lambda)(2 - \lambda) - 1^2 = \lambda^2 - 11\lambda + 17 = 0.$$ 

The eigenvalues are thus 1.8599 and 9.1401. (The larger one is the dominant one.) Using that, we need to find the null space of $\begin{bmatrix} -0.1401 & 1 \\ 1 & -7.1401 \end{bmatrix}$, which we find by row reduction. The two rows are multiples of each other, so we need to find solutions of $\begin{bmatrix} -0.1401 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \tilde{0}$, and so $y = 0.1401x$, and so our null space is spanned by $[1, 0.1401]$, as found in their calculation.
The errors after each step, starting with the original \([1; 1]\), were, 0.85995, 0.15995, 0.031988, 0.0064867, etc.. The ratios of these are 0.186, 0.19999, 0.20279. The theoretical convergence rate was \(|\lambda_2/\lambda_1| = 0.2035\), so these are very close.

The agreement is so good because there are only 2 dimensions, and so \(x = c_1v_1 + c_2v_2\) is simple. In larger matrices, I wouldn’t expect the agreement to be so good, since the extra terms would create complexity. □

**Problem 0.8.** Watkins 5.3.8

*Solution.* If we run a few iterations, we get

\[
\begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow \begin{bmatrix} b \\ a \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ b \end{bmatrix} \Rightarrow \begin{bmatrix} b \\ a \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ b \end{bmatrix}.
\]

This doesn’t converge because the two eigenvalues are \(\lambda = \pm 1\). Thus, \(|\lambda_1| = |\lambda_2|\), and so our convergence argument fails. □

**Problem 0.9.** Watkins 5.3.13

*Solution.* If \((\lambda, v)\) is an eigenpair of \(A\), then \(Av = \lambda v\). This can be rearranged as \(\frac{1}{\lambda}v = A^{-1}v\), and so \((\frac{1}{\lambda}, v)\) is an eigenpair of \(A^{-1}\). □

**Problem 0.10.** Watkins 5.3.14

*Solution.* If \((\lambda, v)\) is an eigenpair of \(A\), then \(Av = \lambda v\). Then

\[(A - \rho I)v = Av - \rho v = \lambda v - \rho v = (\lambda - \rho)v.
\]

Thus \((\lambda - \rho, v)\) is an eigenpair of \(A - \rho I\). □

**Problem 0.11.** Watkins 5.3.15

*Solution.* (a) The eigenvalues of a diagonal matrix are the diagonal values, and their eigenvectors are their associated standard basis functions. So, our three eigenpairs are

\[(2.99, [1; 0; 0]), \quad (1.99, [0; 1; 0]), \quad (1, [0; 0; 1]).\]

(b) If \(\rho = 0.99\), then the eigenpairs of \((A - \rho I)\) are

\[(2, [1; 0; 0]), \quad (1, [0; 1; 0]), \quad (0.01, [0; 0; 1]).\]

Thus, the eigenpairs of \((A - \rho I)^{-1}\) are

\[(1/2, [1; 0; 0]), \quad (1, [0; 1; 0]), \quad (100, [0; 0; 1]).\]

Using direct power method, we get

\[
\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 0.5 \\ 0.005 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 2.5 \times 10^{-1} \\ 2.5 \times 10^{-5} \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 1.25 \times 10^{-1} \\ 1.25 \times 10^{-7} \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 6.25 \times 10^{-2} \\ 6.25 \times 10^{-10} \end{bmatrix}.
\]

This converges to \([1; 0; 0]\).
Using inverse iteration (i.e., the power method with the inverse matrix,) we get
\[
\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.005 \\ 0.01 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2.5 \times 10^{-5} \\ 1 \times 10^{-4} \\ 1 \times 10^{-6} \end{bmatrix} \Rightarrow \begin{bmatrix} 1.25 \times 10^{-7} \\ 1 \times 10^{-8} \end{bmatrix} \Rightarrow \begin{bmatrix} 6.25 \times 10^{-10} \\ 1 \times 10^{-10} \end{bmatrix}.
\]
This converges to \([0; 0; 1]\). This obviously converged much faster. That's because, for \((A - \rho I)^{-1}\), the ratio \(|\lambda_2/\lambda_2| = 100\), while for \((A - \rho I)\), the ratio is only 2.

(c) Using inverse iteration on \((A - \rho I)\) for \(\rho = 2\) converges to the eigenvector \([0; 1; 0]\).

\[
\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} -0.010101 \\ 1 \\ 0.01 \end{bmatrix} \Rightarrow \begin{bmatrix} 1.0203 \times 10^{-4} \\ 1 \times 10^{-4} \end{bmatrix} \Rightarrow \begin{bmatrix} -1.0306 \times 10^{-6} \\ 1 \times 10^{-6} \end{bmatrix}.
\]

Using inverse iteration on \((A - \rho I)\) for \(\rho = 3\) converges to the eigenvector \([1; 0; 0]\).

\[
\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 0.009901 \\ 0.005 \end{bmatrix} \Rightarrow \begin{bmatrix} 9.8030 \times 10^{-5} \\ 2.5 \times 10^{-5} \end{bmatrix} \Rightarrow \begin{bmatrix} 9.7059 \times 10^{-7} \\ 1.25 \times 10^{-7} \end{bmatrix}.
\]
(extra) If you solve the normal equations, you need to solve $A^T Ax = A^T b$. But, if $A$ is even moderately ill-conditioned, then $A^T A$ will be very ill-conditioned, since the condition number is squared. That means your answer is likely to be very inaccurate. Though we didn’t prove it, the QR decomposition doesn’t have this same problem.

(Also, note that the same kind of thing is true for the Cholesky decomposition and solving $Ax = b$. One reason the Cholesky decomposition is nice is that it is more numerically stable than the LU decomposition.)

\[\square\]

**Problem 0.14.** Explain why iterative methods are necessary to find eigenvalues and eigenvectors of matrices.

**Solution.** By a theorem proven by Abel, there is no general formula to find the roots of polynomials with degree greater than four. Since finding the eigenvalues of a $n \times n$ matrix is equivalent to finding the roots of a $n$-th degree polynomial, any exact algorithm for finding eigenvalues would be an exact algorithm for finding roots, an impossibility. Thus, any algorithm we use must be approximate, and so we need to use iterative methods to find eigenvalues and eigenvectors.

\[\square\]

**Problem 0.15.** Write a function in MATLAB that takes as input a square matrix $A$, a guess for an eigenvalue $\rho$ and returns as output the eigenvalue and eigenvector found using the shift-and-invert strategy. Use basic programming, except you may use MATLAB’s built in matrix computation commands such as addition, subtraction, “division,” and multiplication, if needed. To test your program, if you enter the matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix}$ and $\rho = 1$, you should find the eigenvalue 0.19825 and the eigenvector $(-0.33455, 1, -0.57726)$.

**Solution.** Here is one version of my program. For simplicity, I used a fixed 1000 iterations.

```matlab
function [vector, value]=SAI(A, r)

n=length(A);
vector=rand(n,1);
A = A - r * eye(n); % replacing A with A-r*I

for i=1:1000
    % First, we find (A-r*I)^-1 * vector
    vector=A\vector;

    % next we find the largest entry.
    value=vector(1);
    for j=2:n
        if abs(vector(j))>abs(value)
            value=vector(j);
        end
    end
end
```


end
end

%finally, we normalize.
vector=vector/value;
end

value=value^(-1+r); %we have to change the eigenvalue to find the
%eigenvalue of A, rather than of (A-r*I)^-1.

%The eigenvector, however, is the same.

Here is a slightly more sophisticated one that checks whether our algorithm is
converging. I use a “while” loop instead of a for loop. A while loop keeps running
while the condition given is true. It checks the condition at the beginning of each
run of the loop, and does not run the code if the condition given is false.

function [vector, value]=SAI(A, r)

n=length(A);
vector=rand(n,1);

A=A-r*eye(n); %replacing A with A-r*I
error = 1000; %just some large number.

while error>10^-5
  %save the old vector for later use
  oldvector=vector;

  %First, we find (A-r*I)^-1 * vector
  vector=A\vector;

  %next we find the largest entry.
  value=vector(1);
  for j=2:n
    if abs(vector(j))>abs(value)
      value=vector(j);
    end
  end

  %finally, we normalize.
  vector=vector/value;

end
%our error is how much our eigenvector changed in the last step.
%This is not fool-proof, but it works well.
error = norm(vector-oldvector);
end

value=valueˆ−1+r; %we have to change the eigenvalue to find the
eigenvalue of A, rather than of (A−r*I)ˆ−1.

%The eigenvector, however, is the same.

Problem 0.16. Let $A = \begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix}$. This matrix’s dominant eigenvalue is approximately 8. Start from the vector (0, 1) and use the power method (by hand, for 3 iterations) to estimate the dominant eigenvalue and its associated eigenvector.

Solution. The eigenvalue is approximately 8.53 with eigenvector (0.266, 1).

$\begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \end{bmatrix} \Rightarrow \begin{bmatrix} 1/4 \\ 1 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.25 \\ 8.5 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.26 \\ 1 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix} \begin{bmatrix} 0.26 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.26 \\ 8.53 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.266 \\ 1 \end{bmatrix}$

The last multiplier was 8.53, so that is our estimate for the eigenvalue.