Problem 0.1. Watkins 1.3.4

Solution.

\[
\begin{bmatrix}
2 \\
3 \\
2 \\
9
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
2/2 \\
3 \\
2 \\
9
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
1 \\
(3-1\cdot1)/2 \\
2 \\
9
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
1/2 \\
(2-3\cdot1-1\cdot2)/-1 \\
(9-4\cdot1-1\cdot2-3\cdot3)/3
\end{bmatrix}
= \begin{bmatrix}
1 \\
2 \\
3 \\
4
\end{bmatrix}
\]

Problem 0.2. Watkins 1.3.15

Solution. My strategy was to start from \( n \) then subtract off the counters \( i, j \). The hard part is making sure that my indices are referring to the right entries. The general shape of the program is the same as forward substitution. As you should always do, I picked some matrices at random to double check it works, and it does.

```matlab
function b = backsub(A,b)

n=length(b);

for i=0:n-1
    for j=0:i-1
        b(n-i)=b(n-i)-A(n-i ,n-j)*b(n-j);
    end
    if A(n-i ,n-i)==0
        error(”Your matrix is singular.”)
    end
    b(n-i) = b(n-i)/A(n-i ,n-i);
end
```

\[\square\]
Problem 0.3. Watkins 1.3.25

Solution.

\[
1 + 2 + 3 \cdots + n - 1 + n - 1 + n - 2 + n - 3 \cdots + n = n + n + n \cdots + n
\]

for a total of \(n(n - 1)\), since there are \(n - 1\) \(n's\). This was double the original sum, so \(\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}\). \(\square\)

Problem 0.4. Watkins 1.4.15

Solution. (a) Directly, \([x, y]A[x; y] = 4x^2 + 9y^2 > 0\) if either \(x\) or \(y\) is nonzero. Thus \(A\) is positive definite. You could also prove this by computing the Cholesky decomposition.

(b) \(R = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\)

(c) Since the Cholesky factor is the unique upper-triangular matrix with positive diagonal, our \(R\) need to have non-positive diagonals. That suggests these matrices: \(\begin{bmatrix} 2 & 0 \\ 0 & -3 \end{bmatrix}, \begin{bmatrix} -2 & 0 \\ 0 & 3 \end{bmatrix}, \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix}\).

(d) For any \(n \times n\) positive definite matrix, the only choice we have is the sign of the diagonal terms. Since there are \(n\) diagonal terms, that means there are \(2^n\) possible upper-triangular matrices. \(\square\)

Problem 0.5. Watkins 1.4.21

Solution. I won’t show work here, but \(R = \begin{bmatrix} 4 & 1 & 2 & 1 \\ 0 & 3 & 2 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}\). After forward substitution, you should have reduced \(b\) to \(\begin{bmatrix} 8 \\ 6 \\ 5 \\ 1 \end{bmatrix}\). After back substitution, you should have the solution \(x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\). \(\square\)

Problem 0.6. Watkins 1.4.62

Solution. Though there’s more than one way to prove this, the way I had in mind was using the Cholesky decomposition. If \(A\) is positive definite, then there
is an upper triangular $R$ with positive diagonal, and $A = R^T R$. By the hint, $|A| = |R^T R| = |R^T||R|$. Since the determinant of a triangular matrix is the product of its diagonal entries, then $|R| = |R^T| > 0$. Thus $|A| > 0$. □

**Problem 0.7.** Watkins 1.3.7

**Solution.** Here’s my code. There are other ways to do it, but this is a relatively efficient way to set it up.

```matlab
function b=forsub(A,b)

n=length(b);

%first, check how many leading zeros there area
for i=1:n
    if b(i)~=0 %finds first nonzero term in b
        z=i-1; %saves how many zeros there were
        break %"break" forces matlab out of the for loop.
    end
end

%now we do forward subs, ignoring the first rows
for i = z+1:n
    for j=z+1:i-1
        %there is a typo in book that implies this
        %should be j=z+1:n. It ’s wrong. (pg 27)
        b(i) = b(i)-A(i,j)*b(j);
    end
    if A(i,i)==0
        error("Your matrix is singular.")
    end
    b(i) = b(i)/A(i,i);
end
```

□

**Problem 0.8.** Watkins 1.4.52

**Solution.** Suppose that there was some $a_{ii} \leq 0$, instead of them all being positive. Let $x = e_i$, i.e., the vector with 1 in the $i$-th slot, and the rest 0. Then $x^T Ax = a_{ii} \leq 0$, which means that $A$ cannot be positive definite. That contradicts our assumption that $A$ is positive definite, and so we could not have that $a_{ii} \leq 0$. Thus $a_{ii} > 0$. □

**Problem 0.9.** Briefly explain why it is useful to calculate the Cholesky decomposition when solving $Ax = b$ for a number of different $b$’s, as compared to just using standard row reduction.
Solution. Standard row reduction takes about $\frac{2}{3}n^3$ flops, while the Cholesky decomposition plus forward/back substitution takes $\frac{1}{3}n^3$ flops, a saving of half. But the real reason it’s more useful is that you only have to compute it once. You can use the same $R$ for multiple $b$, meaning you only have to do the relatively cheap $2n^2$ flops for forward and backward substitution, rather than the full $O(n^3)$ for solving it directly. □