Question 1

Parts A through C

The functions are defined in the `math425hw1.m` file, under the section `%% Question 1`.

Question 2

For this question, the matrix A is defined as

$$A = \begin{pmatrix} -8 & -2 & 3 & 1 \\ 1 & -2 & 0 & 2 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix}.$$

All relevant computations are provided in the `math425hw1.m` file, under the section `%% Question 2`.

Part A

The first row operation would be to make the first non-pivot position in the first column zero. Namely, $\frac{1}{8}R_1 + R_2 \to R_2$:

$$\underset{\substack{\parallel \\ A}}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 1 & -2 & 0 & 2 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix}} \xrightarrow{\frac{1}{8}R_1+R_2\to R_2} \underset{\substack{\parallel \\ A_1}}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix}}$$

Part B

To get the corresponding elementary matrix E_1, we apply the same row operation to the identity matrix, I_4.

$$\underset{\substack{\parallel \\ I_4}}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}} \xrightarrow{\frac{1}{8}R_1+R_2\to R_2} \underset{\substack{\parallel \\ E_1}}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/8 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}$$

Then, as demonstrated in class, computing E_1A yields A_1.

$$E_1A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/8 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -8 & -2 & 3 & 1 \\ 1 & -2 & 0 & 2 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix} = A_1$$

Part C

Continuing down the first column, we move on to the third row. To make the entry zero, we would do: $-\frac{1}{2}R_1 + R_3 \to R_3$.

$$\underset{A_1}{\underset{\parallel}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix}}} \xrightarrow{-\frac{1}{2}R_1+R_3\to R_3} \underset{A_2}{\underset{\parallel}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 4 & 1 & -1 & -1 \end{pmatrix}}}$$

Part D

And again, we apply the same row operation to the identity matrix to obtain the corresponding elementary matrix E_2.

$$\underset{I_4}{\underset{\parallel}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}} \xrightarrow{-\frac{1}{2}R_1+R_3\to R_3} \underset{E_2}{\underset{\parallel}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}}$$

Again, computing E_2A_1 should yield A_2.

$$E_2A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 4 & 1 & -1 & -1 \end{pmatrix} = A_2$$

Part E

Continuing the process to obtain an upper triangular matrix, we then apply $\frac{1}{2}R_1 + R_4 \to R_4$ to A_2.

$$\underset{\displaystyle A_2}{\underset{\shortparallel}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 4 & 1 & -1 & -1 \end{pmatrix}}} \xrightarrow{\frac{1}{2}R_1+R_4\to R_4} \underset{\displaystyle A_3}{\underset{\shortparallel}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 0 & 0 & 1/2 & -1/2 \end{pmatrix}}}$$

Then, apply the same operation to I_4 to obtain E_3.

$$\underset{\displaystyle I_4}{\underset{\shortparallel}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}} \xrightarrow{\frac{1}{2}R_1+R_4\to R_4} \underset{\displaystyle E_3}{\underset{\shortparallel}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 1 \end{pmatrix}}}$$

For the purpose of maintaining our pattern, I will compute E_3A_2 to show that it yields A_3.

$$E_3A_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 4 & 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 0 & 0 & 1/2 & -1/2 \end{pmatrix} = A_3$$

"Luckily," the last operation also took care of the second column for us. So we now move to the third column by applying $-\frac{1}{3}R_3 + R_4 \to R_4$ to A_3.

$$\underset{\displaystyle A_3}{\underset{\shortparallel}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 0 & 0 & 1/2 & -1/2 \end{pmatrix}}} \xrightarrow{-\frac{1}{3}R_3+R_4\to R_4} \underset{\displaystyle A_4}{\underset{\shortparallel}{\begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 0 & 0 & 0 & -1 \end{pmatrix}}}$$

Again, apply the same operation to I_4 to obtain E_4.

$$\underset{\displaystyle I_4}{\underset{\shortparallel}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}} \xrightarrow{-\frac{1}{3}R_3+R_4\to R_4} \underset{\displaystyle E_4}{\underset{\shortparallel}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/3 & 1 \end{pmatrix}}}$$

Once more, computing E_4A_3 should yield A_4, which is the upper triangular matrix we want to achieve. Hence, we denote $A_4 = U$.

$$E_4A_3 = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -1/3 & 1\end{pmatrix}\begin{pmatrix}-8 & -2 & 3 & 1\\ 0 & -9/4 & 3/8 & 17/8\\ 0 & 0 & 3/2 & 3/2\\ 0 & 0 & 1/2 & -1/2\end{pmatrix} = \begin{pmatrix}-8 & -2 & 3 & 1\\ 0 & -9/4 & 3/8 & 17/8\\ 0 & 0 & 3/2 & 3/2\\ 0 & 0 & 0 & -1\end{pmatrix} = A_4 = U$$

From parts (c) through (e), we found that $E_iA_{i-1} = A_i$ at the end of each row operation i (and for the sake of this argument, let's define $A_0 = A$). In other words, applying the elementary matrix E_i obtained from the corresponding row operation i yields the same result as applying the row operation to the matrix A_{i-1}.

More clearly:

$$\begin{aligned} E_1A &= A_1 & & \\ E_2A_1 &= A_2 &\implies& E_2E_1A = A_2\\ E_3A_2 &= A_3 &\implies& E_3E_2E_1A = A_3\\ E_4A_3 &= A_4 &\implies& E_4E_3E_2E_1A = A_4 = U \end{aligned}$$

Part F

From parts (b) through (f), we just computed the elementary matrices $E_1, \ldots, E_4$. Using MATLAB to compute their inverse, we see that it simply flips the sign of the non-zero entry below the diagonal.

$$E_1 = \begin{pmatrix}1 & 0 & 0 & 0\\ 1/8 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix} \implies E_1^{-1} = \begin{pmatrix}1 & 0 & 0 & 0\\ -1/8 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$E_2 = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ -1/2 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix} \implies E_2^{-1} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 1/2 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$E_3 = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 1/2 & 0 & 0 & 1\end{pmatrix} \implies E_3^{-1} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ -1/2 & 0 & 0 & 1\end{pmatrix}$$

$$E_4 = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -1/3 & 1\end{pmatrix} \implies E_4^{-1} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1/3 & 1\end{pmatrix}$$

Part G

From part (g), we have that:

$$E_4E_3E_2E_1A = U$$

Applying $L = E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1}$:

$$E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1} E_4 E_3 E_2 E_1 A = E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1} U$$

On the left-hand side, $E_i^{-1} E_i = I$ for each $i = 4, 3, 2, 1$:

$$\begin{aligned} E_1^{-1} E_2^{-1} E_3^{-1} \cancel{E_4^{-1} E_4} E_3 E_2 E_1 A &= E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1} U \\ E_1^{-1} E_2^{-1} \cancel{E_3^{-1} E_3} E_2 E_1 A &= E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1} U \\ E_1^{-1} \cancel{E_2^{-1} E_2} E_1 A &= E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1} U \\ \cancel{E_1^{-1} E_1} A &= E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1} U \\ A &= LU \end{aligned}$$

Since L is composed of the inverses of the elementary matrices $E_1, \dots, E_4$ (which are lower triangular), the resulting product $L = E_1^{-1} E_2^{-1} E_3^{-1} E_4^{-1}$ is also a lower triangular matrix.

We note that the elementary matrices (and subsequently, its inverse) are in lower triangular form because we are performing Gaussian elimination in a way such that the column i is always less than row j. As a consequence, the non-zero entry (j, i) is *always* under the main diagonal.

Using MATLAB, we do indeed find that U is the upper triangular matrix arrived in part (g).

As such we conclude that, the LU factorization of the matrix is

$$A = LU = \begin{pmatrix} -8 & -2 & 3 & 1 \\ 1 & -2 & 0 & 2 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix}$$

where $L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1/8 & 1 & 0 & 0 \\ 1/2 & 0 & 1 & 0 \\ -1/2 & 0 & 1/3 & 1 \end{pmatrix}$ and $U = \begin{pmatrix} -8 & -2 & 3 & 1 \\ 0 & -9/4 & 3/8 & 17/8 \\ 0 & 0 & 3/2 & 3/2 \\ 0 & 0 & 0 & -1 \end{pmatrix}$.

Part H

With the LU factorization of A in the previous part, we can solve the system $A\vec{\mathbf{x}} = \vec{\mathbf{b}}$ in two steps using our `myLinearSolution` function.

First, solve $L\vec{\mathbf{y}} = \vec{\mathbf{b}}$:

$$L\vec{\mathbf{y}} = \vec{\mathbf{b}} \implies \vec{\mathbf{y}} = \texttt{myLinearSolution}(L,\ \vec{\mathbf{b}}) = \begin{pmatrix} -2 \\ 23/4 \\ -4 \\ 4/3 \end{pmatrix}$$

Then, solving $U\vec{\mathbf{x}} = \vec{\mathbf{y}}$ gives us the solution:

$$U\vec{\mathbf{x}} = \vec{\mathbf{y}} \implies \vec{\mathbf{x}} = \texttt{myLinearSolution(}U\texttt{, }\vec{\mathbf{y}}\texttt{)} = \begin{pmatrix} {}^{16}\!/_{27} \\ -{}^{109}\!/_{27} \\ -{}^{4}\!/_{3} \\ -{}^{4}\!/_{3} \end{pmatrix}$$

And sure enough, we can verify our solution by computing $A\vec{\mathbf{x}} = \vec{\mathbf{b}}$ with our $\vec{\mathbf{x}}$.

$$A\vec{\mathbf{x}} = \begin{pmatrix} -8 & -2 & 3 & 1 \\ 1 & -2 & 0 & 2 \\ -4 & -1 & 3 & 2 \\ 4 & 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} {}^{16}\!/_{27} \\ -{}^{109}\!/_{27} \\ -{}^{4}\!/_{3} \\ -{}^{4}\!/_{3} \end{pmatrix} = \begin{pmatrix} -2 \\ 6 \\ -5 \\ 1 \end{pmatrix} = \vec{\mathbf{b}}$$

Question 3

In this question, we are referring to the system $A\vec{\mathbf{x}} = \vec{\mathbf{b}}$, where A is a regular $n \times n$ matrix such that

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$$

and $a_{i,j}$ are the entries of the corresponding row i and column j.

Part A

Again, under the assumption that $a_{i,i} \neq 0$ for all $i = 1, \ldots, n$, the first step for us is to convert the coefficient matrix A into an upper triangular matrix U.

First, we need to compute the ratio for the needed to eliminate the entries below the diagonal. In class, we denoted this ratio $\ell_{i,j} = -\frac{a_{j,i}}{a_{i,i}}$. We need to compute this for each entries *below* the diagonal, that is, $n - i$ rows. For a system of $n \times n$ matrix, we perform this calculation at least $n - 1$ times. As such, the number of division operations we perform to calculate the ratio is:

$$\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$$

Then, we perform the row operation, by scaling the row by the ratio and updating it i.e., performing the operation $\ell_{i,j}R_i + R_j \to R_j$. By the magic of MATLAB, we can do this in one step. However, to count the number of operations, we need to break it down:

- First, the ratio is multiplied to each of the entries in the row R_i. At column i, we will multiply the ratio to entries to the right of the pivot i.e., $n-i$ times.
  - And also add it to the corresponding entry in row R_j, but we'll ignore it since we're not counting addition/subtraction.
- Then, we need to repeat the above for all of the rows below. Which, again is $n-i$ rows.

As such, the number of multiplication operations we perform updating the entries is:

$$\sum_{i=1}^{n-1}(n-i)(n-i) = \frac{n(2n-1)(n-1)}{6}$$

At this stage, our $n \times n$ matrix A should be transformed into an upper triangular matrix U in the form

$$U = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & u_{n,n} \end{pmatrix}$$

where $u_{i,j}$ are the entries of the corresponding row i and column j.

Now, we can begin to perform backward substitution to solve for $\vec{\mathbf{b}}$. Let's denote the transformed column vector as $\vec{\mathbf{b}'}$ and its entries b'_i for row i. Our augmented matrix should look like this

$$\left[U \mid \vec{\mathbf{b}'}\right] = \left(\begin{array}{ccccc|c} u_{1,1} & u_{1,2} & \cdots & \cdots & u_{1,n} & b'_1 \\ 0 & u_{2,2} & & & \vdots & \vdots \\ \vdots & & \ddots & & \vdots & \vdots \\ \vdots & & & u_{n-1,n-1} & u_{n-1,n} & b'_{n-1} \\ 0 & \cdots & \cdots & \cdots & u_{n,n} & b'_n \end{array}\right)$$

which is equivalent to the system of equation

$$\begin{cases} u_{1,1}x_1 + u_{1,2}x_2 + \cdots + u_{1,n}x_n = b'_1 \\ 0 \qquad\quad u_{2,2}x_2 + \cdots + u_{2,n}x_n = b'_2 \\ \vdots \qquad\qquad \ddots \\ \vdots \qquad\qquad u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b'_{n-1} \\ 0 \cdots\cdots\cdots u_{n,n}x_n = b'_n \end{cases}$$

where $\vec{\mathbf{x}} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^\top$.

First, we begin solving from the bottom, starting at row n:

$$
\begin{aligned}
x_n &= \frac{b'_n}{u_{n,n}} \\
x_{n-1} &= \frac{b'_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}} \\
x_{n-2} &= \frac{b'_{n-2} - u_{n-2,n}x_n - u_{n-2,n-1}x_{n-1}}{u_{n-2,n-2}} \\
&\vdots \\
x_i &= \frac{b'_i - u_{i,n}x_n - u_{i,n-1}x_{n-1} - \cdots - u_{i,i+1}x_{i+1}}{u_{i,i}} \\
&\vdots \\
x_1 &= \frac{b'_1 - u_{1,n}x_n - \cdots - u_{1,2}x_2}{u_{1,1}}
\end{aligned}
$$

As we can see, at row n, we have one division operation. Then:

- At row $n-1$, there is one multiplication and one division operation; a total of 2.
- At row $n-2$, there are two multiplication and one division operation; a total of 3.
- At row i, there will be $i-1$ multiplication and one division operation; a total of i operations.

As such, the total number of multiplication and division operations for performing back substitution is:

$$1 + 2 + \cdots + (n-2) + (n-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

At the end of this, we should have the solution to our system, $A\vec{\mathbf{x}} = \vec{\mathbf{b}}$. And so, the total number of multiplication and division operations needed to find the solution $\vec{\mathbf{x}}$ by Gaussian elimination and backward substitution is:

$$
\begin{aligned}
\sum_{i=1}^{n-1}(n-i) + \sum_{i=1}^{n-1}(n-i)(n-i) + \sum_{i=1}^{n-1} i &= \frac{n(n-1)}{2} + \frac{n(2n-1)(n-1)}{6} + \frac{n(n-1)}{2} \\
&= \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n
\end{aligned}
$$

Thus, the degree is 3 and the leading term is $\dfrac{1}{3}$.

Part B

If A is invertible, then there exists a matrix P such that $PA = AP = I_n$. P is said to be the inverse of A, denoted A^{-1}. To find A^{-1}, we can set up a system $AP = I_n$. Then, we have that

$$AP = I_n \iff A\begin{pmatrix} | & & | \\ \vec{\mathbf{p}}_1 & \cdots\cdots & \vec{\mathbf{p}}_n \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ \vec{\mathbf{e}}_1 & \cdots\cdots & \vec{\mathbf{e}}_n \\ | & & | \end{pmatrix}$$

where $\vec{\mathbf{e}}_i$ are the zero column vectors with a single 1 entry at row i. By matrix multiplication, we can them decompose $AP = I_n$ into n separate systems.

$$\begin{gathered} A\vec{\mathbf{p}}_1 = \vec{\mathbf{e}}_1 \\ A\vec{\mathbf{p}}_2 = \vec{\mathbf{e}}_2 \\ \vdots \\ A\vec{\mathbf{p}}_n = \vec{\mathbf{e}}_n \end{gathered}$$

And so, to find the inverse, we would need to perform Gaussian elimination and backward substitution on the augmented matrix $\left[A \mid \vec{\mathbf{e}}_i\right]$ for each $i = 1, \ldots, n$.

In part (a), we already found the number of multiplication and division operations needed to find the solution $\vec{\mathbf{x}}$. Now, we're performing the same operations, but for $\vec{\mathbf{p}}_1 \ldots \vec{\mathbf{p}}_n$. As such, the total number of operations needed to find A^{-1} is:

$$\begin{aligned} \sum_{i=1}^{n}\left(\frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n\right) &= n\left(\frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n\right) \\ &= \frac{1}{3}n^4 + \frac{1}{2}n^3 - \frac{5}{6}n^2 \end{aligned}$$

We still need to solve the original system by computing $\vec{\mathbf{x}} = A^{-1}\vec{\mathbf{b}}$. Since A^{-1} is an $n \times n$ matrix and $\vec{\mathbf{b}}$ is an $n \times 1$ column vector, the number of multiplication operations is n^2.

As such, the total number of multiplication and division operations needed to find the solution $\vec{\mathbf{x}}$ by computing the inverse A^{-1} is:

$$\begin{aligned} \sum_{i=1}^{n}\left(\frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n\right) + \sum_{i=1}^{n} n &= \frac{1}{3}n^4 + \frac{1}{2}n^3 - \frac{5}{6}n^2 + n^2 \\ &= \frac{1}{3}n^4 + \frac{1}{2}n^3 + \frac{1}{6}n^2 \end{aligned}$$

Thus, the degree is 4 and the leading coefficient is $\frac{1}{3}$.

Part C

In part (a), we found that solving the system through Gaussian elimination and backward substitution involves $\dfrac{n^3}{3}$ operations. Whereas in part (b), simply computing the inverse A^{-1} itself requires $\dfrac{n^4}{3}$ operations, which grows significantly faster.

Thus, for large n, solving the system using Gaussian elimination and backward substitution will more more efficient than calculating A^{-1} and multiplying by $\vec{\mathbf{b}}$.

Question 4

Part A

First, we notice that the Hilbert matrix is symmetric. The entries in the main diagonal are the unit fractions in descending order. Interestingly, the denominators of the pivot entries are ascending odd numbers, starting from one in the first column. For example, below are 3×3 and 5×5 Hilbert matrices, denoted H_3 and H_5, respectively.

$$H_3 = \begin{pmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{pmatrix}, \qquad H_5 = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{pmatrix}$$

Using MATLAB, we find their respective inverse to be the following:

$$H_3^{-1} = \begin{pmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{pmatrix}, \quad H_5^{-1} = \begin{pmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{pmatrix}$$

Looking at the inverses, we notice that:

- all of the entries are integers;
- the entries on the main diagonal are all positive;
- the first entry $(1, 1)$ for an $n \times n$ Hilbert matrix H_n is always n^2;
- the sign for each entry are alternating, with the first being positive; and
- by the property of symmetric matrices, their inverses are also symmetric.

I can't help but to also notice that there *might* be a pattern for the entries on the major diagonal, but it doesn't appear to be immediately obvious.

In `math425hw1.m`, under the `%% Question 4` section, I've generated additional Hilbert matrices up to $n = 15$, which are omitted here for the sake of brevity.

Part B

To illustrate the issue of numerical inaccuracy, we first generate a random 15×1 column vector $\vec{\mathbf{x}}$ using `rand`. This is the "true solution" that we should get.

$$\vec{\mathbf{x}} = \texttt{rand(15, 1)} = \begin{pmatrix} {}^{1079}/_{3083} \\ {}^{358}/_{1821} \\ \vdots \\ {}^{967}/_{1703} \end{pmatrix}$$

Then, calculate a $\vec{\mathbf{b}}$ by applying the 15×15 Hilbert matrix, H_{15}.

$$H_{15}\vec{\mathbf{x}} = \vec{\mathbf{b}} = \texttt{hlib(15) * x} = \begin{pmatrix} {}^{1604}/_{1149} \\ {}^{551}/_{520} \\ \vdots \\ {}^{226}/_{633} \end{pmatrix}$$

Theoretically, applying H_{15}^{-1} to both sides of $H_{15}\vec{\mathbf{x}} = \vec{\mathbf{b}}$ should give us the original $\vec{\mathbf{x}}$. However, using MATLAB we find that the results vary by a lot. The first entry appears to be correct, however, the difference between the following rows start the grow.

$$\vec{\mathbf{x}}_{\text{actual}} = H_{15}^{-1}\vec{\mathbf{b}} = \texttt{inv(hilb(15)) * b} = \begin{pmatrix} {}^{1079}/_{3083} \\ {}^{1529}/_{7778} \\ \vdots \\ -{}^{97}/_{1069} \end{pmatrix}$$

To illustrate this, the following shows the difference between the expected values of $\vec{\mathbf{x}}$ and the computed values $\vec{\mathbf{x}}_{\text{actual}}$ for this specific randomly-generated vector.

$$\vec{\mathbf{x}}_\Delta = \vec{\mathbf{x}}_{\text{actual}} - \vec{\mathbf{x}} = \begin{pmatrix} {}^{1079}/_{3083} \\ {}^{1529}/_{7778} \\ {}^{971}/_{3863} \\ {}^{1138}/_{1877} \\ {}^{419}/_{817} \\ {}^{2797}/_{10380} \\ {}^{869}/_{544} \\ {}^{2405}/_{1111} \\ {}^{-1383}/_{217} \\ {}^{1855}/_{162} \\ {}^{-8031}/_{422} \\ {}^{2147}/_{141} \\ {}^{-14464}/_{4359} \\ {}^{2397}/_{662} \\ {}^{-97}/_{1069} \end{pmatrix} - \begin{pmatrix} {}^{1079}/_{3083} \\ {}^{358}/_{1821} \\ {}^{695}/_{2768} \\ {}^{1436}/_{2331} \\ {}^{567}/_{1198} \\ {}^{339}/_{964} \\ {}^{1935}/_{2329} \\ {}^{1263}/_{2158} \\ {}^{199}/_{362} \\ {}^{1595}/_{1739} \\ {}^{327}/_{1144} \\ {}^{2261}/_{2986} \\ {}^{2476}/_{3285} \\ {}^{751}/_{1974} \\ {}^{967}/_{1703} \end{pmatrix} \approx \begin{pmatrix} 0 \\ -0.00001\ldots \\ 0.00027\ldots \\ -0.00975\ldots \\ 0.03956\ldots \\ -0.08219\ldots \\ 0.76659\ldots \\ 1.57945\ldots \\ -6.92299\ldots \\ 10.5334\ldots \\ -19.3166\ldots \\ 14.4697\ldots \\ -4.07192\ldots \\ 3.24040\ldots \\ -0.65856\ldots \end{pmatrix}$$

Interestingly, repeated runs of this experiment consistently show that the first entry is always accurate, while the largest absolute differences between the actual and expected values typically occur around rows 10 to 15. The reason for this pattern is beyond my pay grade.

Question 5

First, we can express the matrices $A = LU$ as:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{2,1} & 1 & \ddots & 0 \\ \vdots & & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & u_{n,n} \end{pmatrix}$$

I have no idea how you would achieve it without Gaussian elimination. But I believe through careful calculations you can compute the entries of L and U directly. Though, I can't imagine that it would be more efficient than the Gaussian elimination and backward substitution method since you would likely need to perform as many multiplication operations to compute the entries for L and U.

Once we have compute the LU factorization for a matrix A, the likely advantage is that we can express a system $A\vec{\mathbf{x}} = \vec{\mathbf{b}}$ as $LU\vec{\mathbf{x}} = \vec{\mathbf{b}}$.

Then, if we let $\vec{\mathbf{y}} = U\vec{\mathbf{x}}$ we can express the system as $L\vec{\mathbf{y}} = \vec{\mathbf{b}}$. Since L and U are in lower and upper triangular form, we would only need to perform forward and backward substitution to solve for $\vec{\mathbf{x}}$ for any given $\vec{\mathbf{b}}$. This would be more efficient than performing Gaussian elimination for the entire system again. As we found in 3(a), backward (and therefore forward) substitution only grows as n^2, where as performing Gaussian elimination grows as n^3.