Linear Systems
Motivation

• “Matrices are bad, vector spaces are good”
  • That is, don’t think of matrices as a collection of numbers
  • Instead, think of columns as vectors in a high dimensional space
• We don’t have great intuition going from $R^1$ to $R^2$ to $R^3$ to $R^n$ (for large $n$)
• Thinking about vectors in high dimensional spaces is a good way of gaining intuition about what’s going on
• Linear algebra, as a mathematical area, contains a lot of machinery for dealing with, discussing, and gaining intuition about vectors in high dimensional spaces
• So, while we will cover the essentials of linear algebra, we will do it from the viewpoint of understanding higher dimensional spaces
System of Linear Equations

• System of equations: $3c_1 + 2c_2 = 6$ and $-4c_1 + c_2 = 7$

• Matrix form: $\begin{pmatrix} 3 & 2 \\ -4 & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 7 \end{pmatrix}$ or $Ac = b$

• Given $A$ and $b$, determine $c$

• Theoretically, there is a unique solution, no solution, or infinite solutions

• Ideally, software would determine whether there was a unique solution, no solution, or infinite solutions. In the last case, it would list a parameterized family of solutions. Unfortunately, this is difficult to accomplish.

• Note: in this class, $x$ will be typically be used for data, and $c$ will typically be used for unknowns (such as for the unknown parameters of a neural network)
“Zero”

• One of the basic issues that has to be confronted is the concept of “zero”
• When dealing with large numbers (e.g. Avogadro’s number: $6.022e23$) zero can be quite large
  • E.g. $6.022e23 - 1e7 = 6.022e23$ in double precision, making $1e7$ behave like “zero”
• When dealing with small numbers (e.g. $1e - 23$), “zero” is much smaller
  • In this case, on the order of $1e - 39$ in double precision

• Mixing big and small numbers often wreaks havoc on algorithms
• So, we typically non-dimensionalize and normalize to make equations $O(1)$ as opposed to $O(“big”)$ or $O(“small”)$. 
Row/Column Scaling

• Consider:

\[
\begin{pmatrix}
3e6 & 2e10 \\
1e-4 & 0
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_2
\end{pmatrix} = 
\begin{pmatrix}
5e10 \\
6
\end{pmatrix}
\]

• **Row Scaling** - divide first row by 1e10 to obtain:

\[
\begin{pmatrix}
3e-4 & 2 \\
1e-4 & 0
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_2
\end{pmatrix} = 
\begin{pmatrix}
5 \\
6
\end{pmatrix}
\]

• **Column Scaling** - define a new variable \( c_3 = (1e-4)c_1 \) to obtain:

\[
\begin{pmatrix}
3 & 2 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
c_3 \\
c_2
\end{pmatrix} = 
\begin{pmatrix}
5 \\
6
\end{pmatrix}
\]

• Final result is much easier to treat with finite precision arithmetic

• Solve for \( c_3 \) and \( c_2 \), and then \( c_1 = (1e4)c_3 \)
Transpose and Symmetry

- **Elements** of a matrix are often referred to by their row and column.
- For example, $a_{ik}$ is the element of matrix $A$ in row $i$ and column $k$.

- **Transpose** swaps the row and column of every entry.
- $A^T$ moves element $a_{ik}$ to row $k$ column $i$ (and vice versa).

- The matrix size changes when it’s non-square: \( \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \)

**Symmetric Matrices** have $A^T = A$ meaning that $a_{ik} = a_{ki}$ for all $i$ and $k$. 
Square Matrix

- A size $m \times n$ matrix has $m$ rows and $n$ columns
- For now, let’s consider square $n \times n$ matrices
- We will consider (non-square) rectangular matrices with $m \neq n$ a bit later
Solvability

• **Singular** – $A$ is singular when it is not invertible (does not have an inverse)

• Various ways of showing this:
  • At least one column is linearly dependent on others (as we have seen before)
  • The **determinant** is zero: $\det A = 0$
  • $A$ has a nonempty **null space**, i.e. $\exists c \neq 0$ with $Ac = 0$

• **Rank** - maximum number of linearly independent columns

• Singular matrices have rank $< n$ (the # of columns), i.e. rank-deficient, and have either no solution or infinite solutions

• A **nonsingular** square matrix has an inverse: $AA^{-1} = A^{-1}A = I$
  • so $Ac = b$ can be solved for $c$ via $c = A^{-1}b$

• **Note:** we typically do not compute the inverse, but instead have a solution algorithm that exploits its existence
Matrices as Vectors (an example)

• Recall $Ac = \sum_k c_k a_k$ where the $a_k$ are the columns of $A$

• Consider $Ac = 0$ or $\sum_k c_k a_k = 0$

• If one column is a linear combination of others, then the linear combination weights can be used to obtain $Ac = 0$ with $c$ nonzero
  • This nonzero $c$ is in the null space of $A$, and $A$ is singular

• Conversely, if the only solution to $Ac = 0$ is $c$ identically 0, then no column is linearly dependent on the others
  • Thus $A$ is nonsingular
Diagonal Matrices

- All off-diagonal entries are 0
- Equations are decoupled, and easy to solve
- E.g. \[
\begin{pmatrix}
5 & 0 \\
0 & 2
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_2
\end{pmatrix}
= 
\begin{pmatrix}
10 \\
-1
\end{pmatrix}
\]
has \(5c_1 = 10\) and \(2c_2 = -1\) so \(c_1 = 2\) and \(c_2 = -0.5\)
- A zero on the diagonal indicates a singular system, which has no solution (e.g. \(0c_1 = 10\)) or infinite solutions (e.g. \(0c_1 = 0\))
- The determinant of a diagonal matrix is obtained by multiplying all the diagonal elements together
- Thus, a 0 on the diagonal implies a zero determinant and a singular matrix
Upper Triangular Matrices

• All entries below the diagonal are 0
• Nonsingular when the diagonal elements are all nonzero
  • Determinant is obtained by multiplying all the diagonal elements together
• Solve via back substitution

\[
\begin{pmatrix}
5 & 3 & 1 \\
0 & 1 & -1 \\
0 & 0 & 5
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_2 \\
c_3
\end{pmatrix}
=
\begin{pmatrix}
0 \\
10 \\
10
\end{pmatrix}
\]

• E.g. consider

• Start at the bottom with $5c_3 = 10$ or $c_3 = 2$, and move upwards one row at a time. Next, $c_2 - c_3 = 10$ or $c_2 - 2 = 10$ or $c_2 = 12$. Then, $5c_1 + 3c_2 + c_3 = 0$ or $5c_1 + 36 + 2 = 0$ or $c_1 = -38/5 = -7.6$
Lower Triangular Matrices

• All entries above the diagonal are 0
• Nonsingular when the diagonal elements are all nonzero
  • Determinant is obtained by multiplying all the diagonal elements together
• Solve via forward substitution

\[
\begin{pmatrix}
5 & 0 & 0 \\
-1 & 1 & 0 \\
1 & 3 & 5
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_2 \\
c_3
\end{pmatrix}
=
\begin{pmatrix}
10 \\
10 \\
0
\end{pmatrix}
\]

• E.g. consider

• Start at the top with \( 5c_1 = 10 \) or \( c_1 = 2 \), and move downwards one row at a time. Next, \( -c_1 + c_2 = 10 \) or \( -2 + c_2 = 10 \) or \( c_2 = 12 \). Then, \( c_1 + 3c_2 + 5c_3 = 0 \) or \( 2 + 36 + 5c_3 = 0 \) or \( c_3 = -38/5 = -7.6 \)
Elimination Matrix

- Standard basis vectors: \( \hat{e}_i = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \) with a 1 in the \( i \)-th row/entry

- Given \( \begin{pmatrix} a_{1k} \\ \vdots \\ a_{ik} \\ a_{i+1,k} \\ \vdots \\ a_{mk} \end{pmatrix} \), define \( m_{ik} = \frac{1}{a_{ik}} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{i+1,k} \\ \vdots \\ a_{mk} \end{pmatrix} \) and \( M_{ik} = I_{m \times m} - m_{ik} \hat{e}_i^T \)

- \( M_{ik} \) is a size \( m \times m \) elimination matrix that subtracts multiples of row \( i \) from rows \( > i \) in order to create zeroes in column \( k \)
Elimination Matrix

- Let \( a_1 = \begin{pmatrix} 2 \\ 4 \\ -2 \end{pmatrix} \)

- \( M_{11} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 0 \\ 4 \\ -2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \) and \( M_{11} a_1 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \)

- \( M_{21} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \frac{1}{4} \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{pmatrix} \) and \( M_{21} a_1 = \begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix} \)
Elimination Matrix Inverse

- Inverse of an elimination matrix is $L_{ik} = M_{ik}^{-1} = I_{mxm} + m_{ik} \hat{e}_i^T$

- $L_{ik}$ is a size $m \times m$ elimination matrix that adds multiples of row $i$ to rows $> i$ in order to reverse the effect of $M_{ik}$

- $L_{11} = M_{11}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$

- $L_{21} = M_{21}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/2 & 1 \end{pmatrix}$
Combining Elimination Matrices

- \[ M_{i_1k_1} M_{i_2k_2} = I - m_{i_1k_1} \hat{e}_{i_1}^T - m_{i_2k_2} \hat{e}_{i_2}^T \] when \( i_1 < i_2 \) but not when \( i_1 > i_2 \)

\[
M_{11}M_{21} = \begin{pmatrix}
1 & 0 & 0 \\
-2 & 1 & 0 \\
1 & 1/2 & 1
\end{pmatrix}
\]

- \[ L_{i_1k_1} L_{i_2k_2} = I + m_{i_1k_1} \hat{e}_{i_1}^T + m_{i_2k_2} \hat{e}_{i_2}^T \] when \( i_1 < i_2 \) but not when \( i_1 > i_2 \)

\[
L_{11}L_{21} = \begin{pmatrix}
1 & 0 & 0 \\
2 & 1 & 0 \\
-1 & -1/2 & 1
\end{pmatrix}
\]
Gaussian Elimination

- Consider
  \[
  \begin{pmatrix}
  2 & 4 & -2 \\
  4 & 9 & -3 \\
  -2 & -3 & 7
  \end{pmatrix}
  \begin{pmatrix}
  c_1 \\
  c_2 \\
  c_3
  \end{pmatrix} =
  \begin{pmatrix}
  2 \\
  8 \\
  10
  \end{pmatrix}
  \]

- \( M_{11}A = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{pmatrix} = \begin{pmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{pmatrix} \) and \( M_{11}b = \begin{pmatrix} 2 \\ 4 \\ 12 \end{pmatrix} \)

- \( M_{22}M_{11}A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{pmatrix} \) and \( M_{22}M_{11}b = \begin{pmatrix} 2 \\ 4 \\ 8 \end{pmatrix} \)

- Then, solve the upper triangular
  \[
  \begin{pmatrix}
  2 & 4 & -2 \\
  0 & 1 & 1 \\
  0 & 0 & 4
  \end{pmatrix}
  \begin{pmatrix}
  c_1 \\
  c_2 \\
  c_3
  \end{pmatrix} =
  \begin{pmatrix}
  2 \\
  4 \\
  8
  \end{pmatrix}
  \) via back substitution
LU Factorization

- Gaussian Elimination gives an upper triangular \( U = M_{n-1,n-1} \cdots M_{22}M_{11}A \)
- Using inverses, \( A = L_{11}L_{22} \cdots L_{n-1,n-1}M_{n-1,n-1} \cdots M_{22}M_{11}A = L_{11}L_{22} \cdots L_{n-1,n-1}U \)
- Since \( L_i i_1 L_i i_2 = I + m_{i_1 i_1} \hat{e}_{i_1}^T + m_{i_2 i_2} \hat{e}_{i_2}^T \) when \( i_1 < i_2 \), \( L = L_{11}L_{22} \cdots L_{n-1,n-1} \) is lower triangular and \( A = LU \)

Here \( L = L_{11}L_{22} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \)

\[ A = \begin{pmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{pmatrix} = LU \]
LU Factorization

• Factorizing $A = LU$ facilitates solving $Ac = b$

• In order to solve $LUc = b$, define an auxiliary variable $\hat{c} = Uc$

• First, solve $L\hat{c} = b$ for $\hat{c}$ via forward substitution

• Second, solve $Uc = \hat{c}$ for $c$ via back substitution

• Note: the LU factorization is only computed once, and then can be used afterwards on many right hand side ($b$) vectors
Pivoting

- $A = \begin{pmatrix} 0 & 4 \\ 4 & 9 \end{pmatrix}$ requires division by zero in order to create $M_{11}$

- **(Partial) Pivoting** - swap rows to use the largest (magnitude) element in the column under consideration
  - Don’t forget to swap the right hand side $b$ too

- **Full Pivoting** swap rows and columns to use the largest possible element
  - Don’t forget to change the order of the unknowns $c$

- When considering column $k$, can only swap with rows/columns $\geq k$
Permutation Matrix

• Constructed by switching the 2 rows of $I$ that one wants swapped
  \[
  P_{13} = \begin{pmatrix}
  0 & 0 & 1 \\
  0 & 1 & 0 \\
  1 & 0 & 0
  \end{pmatrix}, \text{ and } P_{13}A \text{ swaps the first and third rows of } A
  \]

• Permutation matrices are their own inverses (swapping again restores the rows)

• Switching rows $i_1$ and $i_2$ moves a 1 from $a_{i_1i_1}$ to $a_{i_2i_1}$ as well as from $a_{i_2i_2}$ to $a_{i_1i_2}$, preserving symmetry (i.e. $P_{i_1i_2}^T = P_{i_1i_2}$)

• To swap the first and third unknowns: $Ac = AP_{13}P_{13}c = (AP_{13})(P_{13}c)$ where $P_{13}c$ swaps the unknowns and $AP_{13}$ swaps the columns (to see this, consider $(AP_{13})^{TT} = (P_{13}A^T)^T$ which swaps the rows of $A^T$)
Full Pivoting

- Let $P_{ri}$ be the permutation matrix that (potentially) switches row $i$ with a row $> i$
- Let $P_{ck}$ be the permutation matrix that (potentially) switches column $k$ with a col $> k$
- Then full pivoting can be written as:

$$
(M_{n-1,n-1}P_{r_{n-1}} \ldots M_{22}P_{r_{2}}M_{11}P_{r_{1}}AP_{c_{1}}P_{c_{2}} \ldots P_{c_{n-1}})(P_{c_{n-1}} \ldots P_{c_{2}}P_{c_{1}}c)
$$

- Once known, $P_r = P_{r_{n-1}} \ldots P_{r_{2}}P_{r_{1}}$ and $P_c = P_{c_{n-1}} \ldots P_{c_{2}}P_{c_{1}}$ can be used to do all the permutations ahead of time (the resulting matrix doesn’t require pivoting)
- $Ac = b$ becomes $(P_rAP_c^T)(P_c c) = P_r b$ or $A_pc_p = b_p$; then, $A_p = L_p U_p$ can be computed without pivoting
- Subsequently, given any right hand side $b$, solve $L_p U_p c_p = P_r b$ to find $c_p$ using forward/back substitution, and then $c = P_c^T c_p$
Sparsity

• Most large matrices (of interest) operate on variables that only interact with a sparse set of other variables

• This makes the matrix sparse (as opposed to dense), i.e. most entries are identically 0

• However, the inverse of a sparse matrix can contain an unwieldy amount of non-zero entries

• E.g. the 3D Poisson equation on a relatively small $100^3$ Cartesian grid has an unknown for each of the $10^6$ grid points

• For each unknown, the discretized Poisson equation depends on the unknown itself and its 6 immediate Cartesian grid neighbors

• Thus, the size $10^6 \times 10^6$ matrix has only $7 \times 10^6$ nonzero entries

• But, the inverse potentially has $10^{12}$ nonzero entries!
Computing the Inverse

- When $A$ is relatively small (and dense), one might compute $A^{-1}$

- Since $AA^{-1} = I$, the solution $c_k$ to $Ac_k = \hat{e}_k$ is the $k$-th column of $A^{-1}$

- First, compute $A_P = L_P U_P$ as usual

- Then, solve $Ac_k = \hat{e}_k$ repeatedly ($n$ times, once for each column)