The Three-Day Plan

• Geometry of Linear Algebra
  ▸ Vectors, matrices, basic operations, lines, planes, homogeneous coordinates, transformations

• Solving Linear Systems
  ▸ Gaussian Elimination, LU and Cholesky decomposition, over-determined systems, calculus and linear algebra, non-linear least squares, regression

• The Spectral Story
  ▸ Eigensystems, singular value decomposition, principle component analysis, spectral clustering
Vectors

• A vector is a motion in space:

\[
\begin{bmatrix}
  2 \\
  1 \\
\end{bmatrix}
\]

• Vectors can be added

\[
\begin{bmatrix}
  2 \\
  1 \\
\end{bmatrix} + \begin{bmatrix}
  1 \\
  -3 \\
\end{bmatrix} = \begin{bmatrix}
  3 \\
  -2 \\
\end{bmatrix}
\]
Dot Products

- Produces a **scalar**

- Measures the similarity in direction of two vectors.

\[
\begin{bmatrix}
2 \\
1
\end{bmatrix} \cdot \begin{bmatrix}
3 \\
2
\end{bmatrix} = 2 \times 3 + 1 \times 2 = 8
\]

- Geometry: Project one vector into the direction of the other. Then rescale by length of the second vector.

- What’s the dot product of your thesis and final project?
Checkpoint

• What is the dot product of a vector with itself?

• What is the dot product of two orthogonal vectors?
Computing orthogonal vectors

- How can we compute vectors whose dot product will be zero?

- 2D?
  - How many vectors are perpendicular?

- 3D?
  - How many vectors are perpendicular?
Constructing perpendicular vectors in 3D

- It’s actually too easy... we can ask how many vectors are simultaneously perpendicular to two vectors.

- I.e., where do the two planes intersect? (along a line!)
  - (how many perpendicular vectors?)
Cross Product (3D)

- Orthogonal vector can be constructed in 3D by cross product:

\[
\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}.
\]

\[
\mathbf{a} \times \mathbf{b} = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix} \mathbf{i} - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix} \mathbf{j} + \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} \mathbf{k}
\]
Checkpoint

View from eye
Lines and Hyperplanes

- What’s the relationship between a line, a plane, and a hyperplane?

- Is there a generic way of writing them, regardless of dimension?

\[ n \cdot x = b \]

- What do \( n \) and \( b \) mean, geometrically?

\( n \) is the normal to the hyperplane
\( b \) is the distance (in the direction of the normal) from the hyperplane to the origin...

E.g. consider point \( bn \), which lies on the plane.
Planes versus Lines

• In 2D, a hyperplane and a line have the same form.

  ▶ A **line** “pins down” all *but* one degree of freedom
    - Can move in one dimension up and down the line

  ▶ A **hyperplane** “pins down” only *one* degree of freedom.
    - Can move in all *but* one degree of freedom.

• In short:
  ▶ A single linear equation yields a hyper-plane.
  ▶ N-1 simultaneous linear equations yields a line.
Checkpoint

• A point is the intersection of how many hyperplanes?

• What point on a hyperplane defined by n and b is closest to some other point p?

• Why is this formulation better than $y=mx + b$?
Matrix: The Basics

- A matrix is a rectangular array of numbers.
  
  - Has geometric interpretation too... but first, let's start with something simple.

- When can matrices be added together? (And how does addition work?)
Matrix-vector product

• Mechanics of matrix-vector multiplication:

\[
\begin{bmatrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{bmatrix}
\begin{bmatrix}
  j \\
  k \\
  l
\end{bmatrix}
= \begin{bmatrix}
  aj + bk + cl \\
  dj + ek + fl \\
  gj + hk + il
\end{bmatrix}
\]

\[
\begin{bmatrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{bmatrix}
\begin{bmatrix}
  j \\
  k \\
  l
\end{bmatrix}
= \begin{bmatrix}
  aj + bk + cl \\
  dj + ek + fl \\
  gj + hk + il
\end{bmatrix}
= \begin{bmatrix}
  a \\
  d \\
  g
\end{bmatrix} j + \begin{bmatrix}
  b \\
  e \\
  h
\end{bmatrix} k + \begin{bmatrix}
  c \\
  f \\
  i
\end{bmatrix} l
\]

• Two different interpretations:
  ▶ Dot product of rows with vector
  ▶ Linear combination of columns

\[
\begin{bmatrix}
  2 & -1 \\
  1 & 1
\end{bmatrix}
\begin{bmatrix}
  3 \\
  -4
\end{bmatrix}
\]
Linear Systems

• We can use a variable instead of a vector, which gives us a linear system.

\[
\begin{bmatrix}
2 & -1 \\
1 & 1
\end{bmatrix} x = \begin{bmatrix}
2 \\
4
\end{bmatrix}
\]

• What simultaneous equations does this represent?
Matrix Geometry: Row story

- Each row of a linear system represents a hyperplane. (In 2D, that’s also a line!)
- The solution to the system is the intersection of those hyperplanes
Matrix Geometry: Column Story

- Each column can be interpreted as a vector
  - How far do we go in each direction?

\[
\begin{bmatrix}
2 & -1 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
2 \\
4
\end{bmatrix}
\]

\[
\begin{bmatrix}
2 \\
1
\end{bmatrix}
x_1 + 
\begin{bmatrix}
-1 \\
1
\end{bmatrix}
x_2 = 
\begin{bmatrix}
2 \\
4
\end{bmatrix}
\]
Matrix-matrix product

- Think of the right hand side as a list of column vectors.
  - Each is transformed separately via a matrix-vector product.

\[
\begin{bmatrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{bmatrix}
\begin{bmatrix}
  j & k & l \\
  m & n & o \\
  p & q & r
\end{bmatrix}
= 
\begin{bmatrix}
  - & - & - \\
  - & - & gk + hn + iq \\
  - & - & -
\end{bmatrix}
\]

- Finger sweeping rule should be second nature!
  - Left finger sweeps left to right
  - Right finger sweeps top to bottom

Saturday, September 10, 11
Checkpoint

• Which of the following multiplications are sensible?
Matrices as projections

- Matrix multiplication projects from one space to another.

\[
\begin{bmatrix}
\text{Data in original coordinate system}
\end{bmatrix}
= \begin{bmatrix}
A
\end{bmatrix}
\begin{bmatrix}
\text{Data projected into new coordinate system}
\end{bmatrix}
\]

Direction of new $x$ in old coordinate system - we're taking a dot product with it!

Direction of old $x$ in new coordinate system.
Rotation Matrices

• A special kind of projection
  ‣ Preserves scale, distance, and relative orientation. (It’s *rigid.*)
  ‣ It’s orthonormal too!

• Write a rotation matrix for 45 degree rotation around the Z axis.

\[
\begin{bmatrix}
cos(\theta) & -\sin(\theta) & 0 \\
\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Direction of new x in old coordinate system—we’re taking a dot product with it!

Direction of old x in new coordinate system.
Rotation

- Consider “rotate +90 degrees around the Z axis”

- Use right-hand to represent the coordinate frame

- Which way is +90 degrees?
  - Point right thumb in the direction of the axis of rotation
  - Fingers curl in the positive direction
Translation Matrices

• How do we do a translation?

There's no way to do it without homogeneous coordinates. Ask class... show that we're stuck.
Homogenous Coordinates

- We’ll introduce a new convention, *homogenous coordinates*.
- We write points just the way we did before, but add an extra row:
  - The extra row is a *scale factor* for the whole vector.

\[
p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{becomes} \quad p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
\]

What point does this correspond to?

\[
\begin{bmatrix} 10 \\ 20 \\ 15 \\ 5 \end{bmatrix}
\]
Translation

- Suppose I want to shift all objects by $T_x, T_y, T_z$:

\[
\begin{bmatrix}
    x + T_x \\
    y + T_y \\
    z + T_z \\
    1
\end{bmatrix}
= \begin{bmatrix} ? \end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z \\
    1
\end{bmatrix}
\]
Rotation

• How about a rotation of 90 degrees around the Z axis?

› (This time in homogeneous coordinates)

\[
\begin{bmatrix}
-y \\
x \\
z \\
1
\end{bmatrix}
= 
\begin{bmatrix}
? \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
More complex rotations

• How about rotating 37.2 degrees around the vector 
  \[ \begin{bmatrix} 0.3 & 0.6 & 0.4 \end{bmatrix} \]

  ▶ Don’t worry. Nobody can do that in their head!
  ▶ But it’ll look like this:

  \[
  \begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
  \end{bmatrix}
  =
  \begin{bmatrix}
  R_{00} & R_{01} & R_{02} & 0 \\
  R_{10} & R_{11} & R_{12} & 0 \\
  R_{20} & R_{21} & R_{22} & 0 \\
  0 & 0 & 0 & 1
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y \\
  z \\
  1
  \end{bmatrix}
  \]
Rigid-Body Transformations

• The product of two rigid-body transformations is always another rigid-body transformation!

  ‣ Does order of multiplication matter?

• So no matter how the object has been translated or rotated, we can describe its position with a single 4x4 matrix, which has the structure:

\[
\begin{bmatrix}
x' \\
y' \\
z' \\
1
\end{bmatrix} =
\begin{bmatrix}
R_{00} & R_{01} & R_{02} & T_x \\
R_{10} & R_{11} & R_{12} & T_y \\
R_{20} & R_{21} & R_{22} & T_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
How to “undo” a transformation?
Matrix Inverse

• Definition: $AA^{-1} = I$

• Dot product interpretation
  ▶ Find a vector whose dot product with the first row is one, and whose dot product with every other row is zero.
  ▶ Repeat this for every row.

• Distributive property: (Prove it)
  $$(AB)^{-1} = B^{-1}A^{-1}$$
When does the inverse exist?

- Inverse exists iff the matrix has full rank.

- The “row” story: the matrix describes a set of intersecting hyperplanes.
  - Is their intersection a point? A line? Something higher dimensional?

- The “column” story: the column vectors describe “directions” that you can move in. Can you get everywhere?

- Easy cases: rigid-body transformations? rectangular matrices?

- We’ll talk more about this in the next lectures.
Transpose

• Definition:

\[ A^T_{i,j} = A_{j,i} \]

• Allows us to write dot product:

\[ a^T b \]

• Distributive property:

\[ (AB)^T = B^T A^T \]

• Sometimes a transpose is an inverse! (When?)

orthonormal matrices... draw an example with rows v1, v2, v3... noting that \( \text{dot}(v_1,v_2) = 0 \)

we won’t use \( \cdot \) anymore.

Distributive property seems familiar... it’s like the inverse rule.
Checkpoint

• Is the following matrix invertible?

\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{pmatrix}
\]

• What’s the inverse of:
  - \text{RotateZ}(3.5) \ast \text{Translate}(1,2,0) \ast \text{Translate}(-2,2,3)

• R is a rotation matrix. Simplify the expression:

\[
(R^T B)^{-1} (A^T R)^T (B^{-1} A)^{-1}
\]