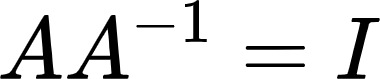
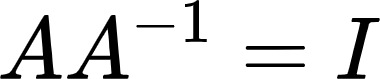
**Midterm 2**

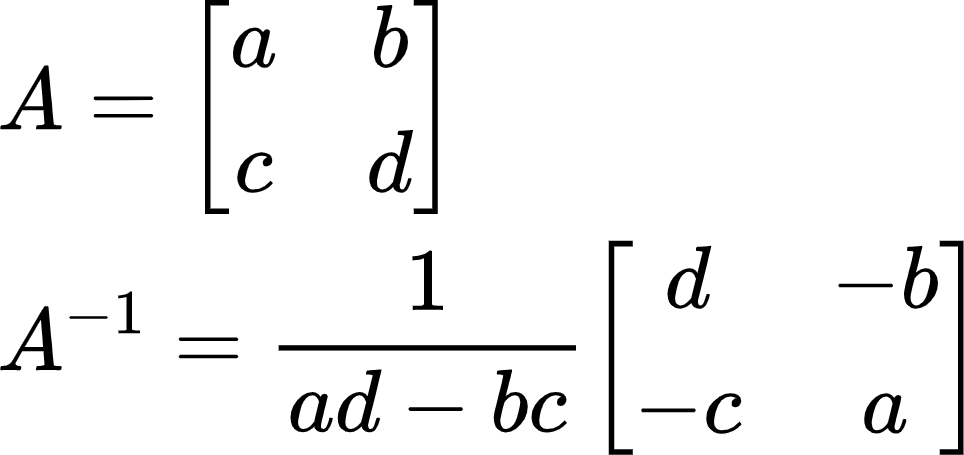
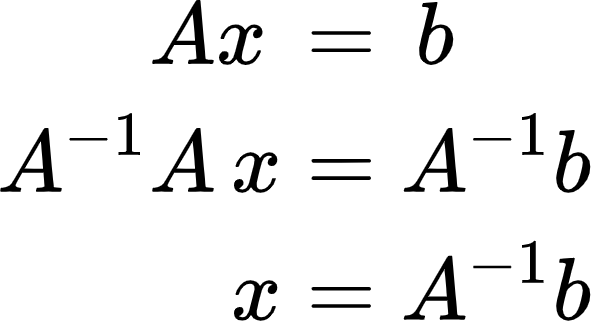
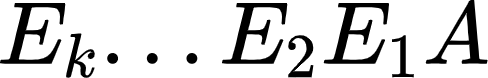
**Midterm 1 Review:** [Midterm 1 Study Guide](https://docs.google.com/document/d/1cDebjXM7Ptmn9ReloB8LKhDaLnorc3OQggCdJVojFxw/edit?usp=sharing)

**Section 2.2: The Inverse of a Matrix**

**Definitions**

* Invertible Matrix
  + An *n x n* matrix A where 
* Inverse of a Matrix
  + *A-1* where 
* Singular
  + Not invertible
* Determinant of a 2x2 matrix
  + ad - bc
* Elementary Matrix (E)
  + Matrix obtained by performing a **single** row operation on an **identity matrix**
  + Are invertible: inverse of an elementary matrix, E, is another elementary matrix of the same type that **transforms E back to I**
  + **All** elementary matrices are invertible
* Row Equivalent Matrices
  + Matrices that can **transform into each other** through a sequence of elementary row operations

**Key Notes**

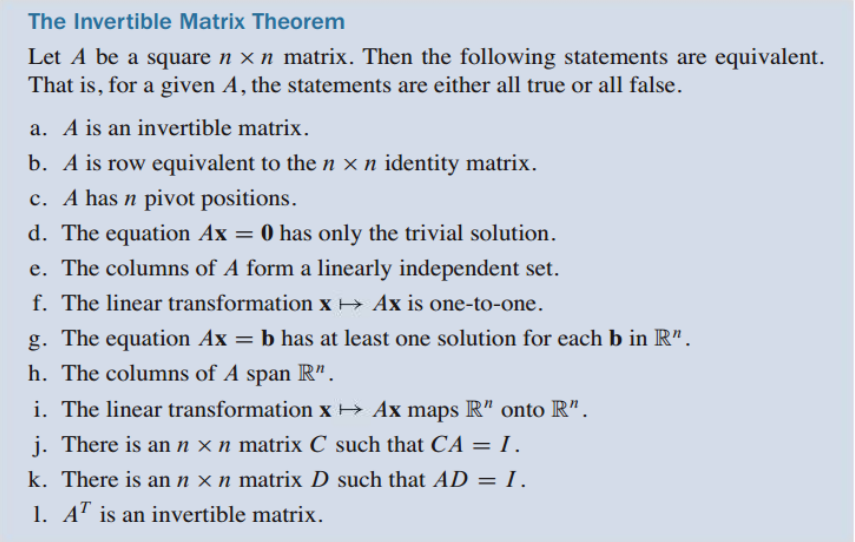
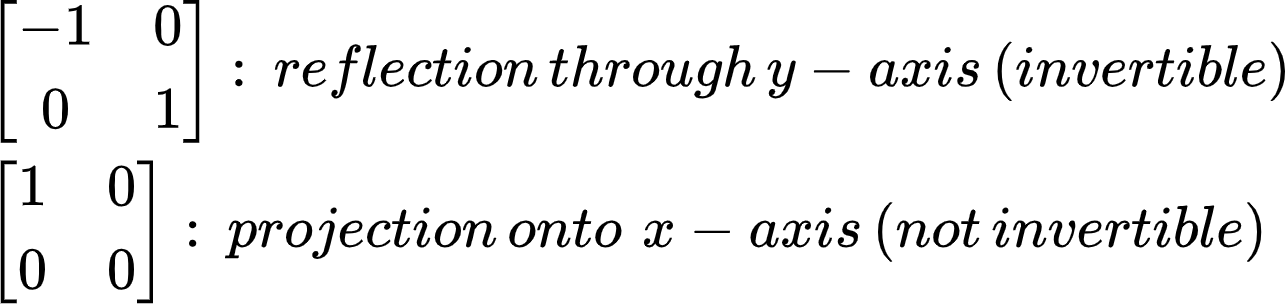
* Invertible = **non**singular
* **Not** invertible = singular
* Inverse of a 2x2 Matrix
  + 
    - If (ad - bc) = 0, then A is **not invertible**
* Ax = b can be rewritten using inverses **only if A is invertible**
  + 
  + Of course, you can still use the **row reduction** method to solve Ax = b
  + For all b in Rn, x = A-1b is a **unique solution**
    - Invertible matrices have **no free variables**
    - Unique solution
* Product of *n x n* invertible matrices is invertible
  + Inverse of product is the product of the inverses in **reverse order**
* When an elementary row operations is performed on an *m x n* matrix A, the resulting matrix can be written as EA
  + What if we had **multiple elementary row operations** on A?
    - 
* Method to find the inverse
  + Row reduce A to the identity matrix while performing the **same row operations on the identity matrix** at the same time
  + [A | I] => [I | A-1]
* A matrix is invertible if and only if it is **row equivalent** to the identity
  + Pivots in every row and column **(onto & one-to-one)**

**Section 2.3: Characterizations of Invertible Matrices**

**Definitions**

* Linear Transformation
  + Mapping between two vector spaces (Rn’s) that preserves all vector addition & scalar properties
* Invertible Linear Transformation
  + Linear transformation *T: Rn -> Rn* is invertible if there is **another linear transformation S***: Rn -> Rn* such that:
    - S(T(x)) = x for all x in Rn
    - T(S(x)) = x for all x in Rn
  + Equivalent to saying:
    - A-1Ax = (I)x

**Key Notes**

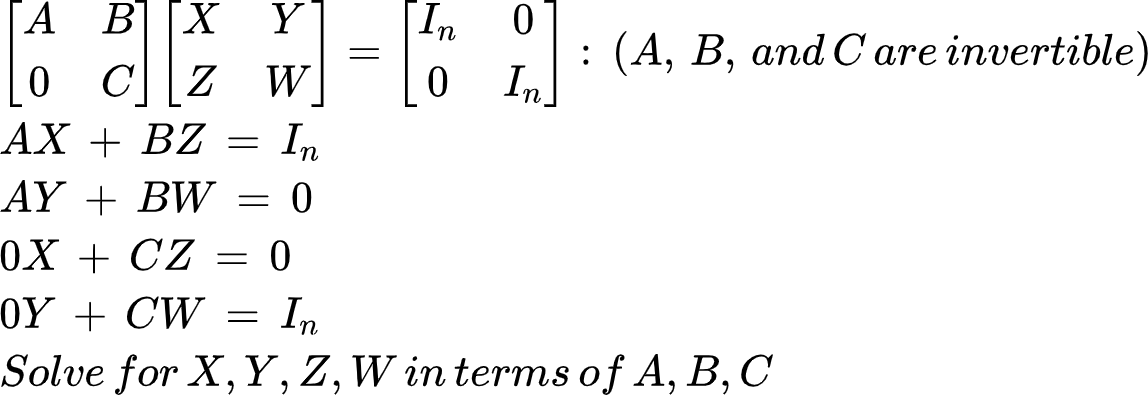
* The IMT
  + 
* Let A and B be square matrices:
  + If AB = I, then A and B are **both invertible**
  + B = A-1 & A = B-1
* How to determine if a linear transformation is **invertible**?
  + Let a matrix A represent the linear transformation
  + If A is invertible, then the linear transformation is invertible
    - 

**Section 2.4: Partitioned Matrices**

**Definitions**

* Partitioned Matrix
  + Matrix divided up into separate blocks
* Block Diagonal Matrix
  + A partitioned matrix where all blocks except the main diagonal are 0’s
  + Is invertible if the main diagonal blocks are invertible

**Key Notes**

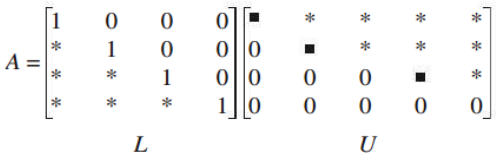
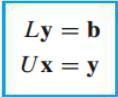
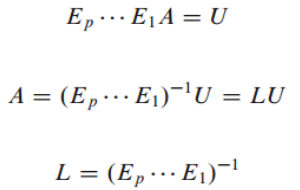
* Adding 2 partitioned matrices A and B
  + A and B must be the **same size** and partitioned in the **exact same way**
    - Add **block by block**
* Scaling partitioned matrices
  + Scale block by block
* Multiplying 2 partitioned matrices A and B (A\*B)
  + Column partition of A **must equal** row partition of B
  + Number of columns in partition A = number of rows in partition B
    - Just like multiplying regular matrices
    - (2 x 2) \* (2 x 1) => (2 x 1)
    - (3 x 4) \* (4 x 1) => (3 x 1)
* Inverses of Partitioned Matrices
  + 

**Section 2.5: Matrix Factorizations**

**Definitions**

* Factorization of a matrix
  + Expression of a matrix as the product of two or more matrices
* Row interchanges
  + Swapping rows when row reducing
* Lower triangular matrix
  + Entries above the main diagonal are all 0’s
* Upper triangular matrix
  + Entries below the main diagonal are all 0’s
* Algorithm for an LU Factorization
  + 1. Reduce *A* to an echelon form *U* by a sequence of row replacement operations, if possible.
  + 2. Place entries in *L* such that the *same sequence of row operations* reduces L to I.

**Key Notes**

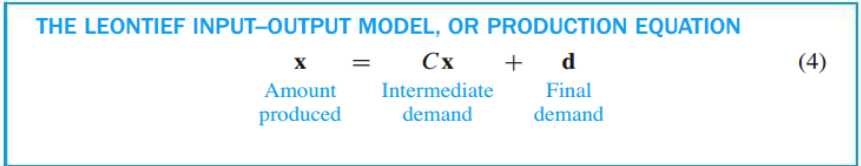
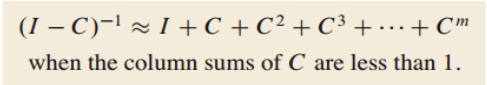
* LU Factorization
  + Why do we use it?
    - More efficient to solve a sequence of equations with the same coefficient matrix (Ax = b1, Ax = b2, ... , Ax = bn) by LU factorization than row reducing the equations every single time
* Let A be an *m x n* matrix that can be row reduced to echelon form **without row exchanges**, then:
  + 
    - **L**: *m x m* lower triangular matrix with one’s on the main diagonal
    - **U**: *m x n* echelon form of A
* Rewriting Ax = b using A = LU
  + 
  + Ax = b -> L(Ux) = b
* The LU Factorization Algorithm
  + How do we get U?
    - Row reduce A to echelon form using only row replacements that add a multiple of one row to another below it
  + How do we get L?
    - Take the row replacement operations you did on A when getting echelon form
      * Basically: find the **elementary matrices** that transform A into U
    - Then, **reverse** the signs and input them in their respective spots in the *m x m* identity matrix
      * Replace the 0’s below the main diagonal with the row replacement “coefficients”
      * Basically: after finding all the elementary matrices, take their **inverses**
  + 
* Using the LU Decomposition
  + After constructing A = LU, solve A*x* = LU*x* = b by:
    - 1. Forward solve for *y* in L*y* = *b*
      * R1(x) + R2 -> R2
      * Modify rows **below** using **above rows**
    - 2. Backwards solve for *x* in U*x* = *y*
      * R2(x) + R1 -> R1
      * Modify rows **above** using **below rows**

**Section 2.6: The Leontief Input-Output Model**

**Definitions**

* Production vector in Rn (x)
  + Lists the output of each sector for one year
* Final demand vector (d)
  + Lists the value of goods and services produced for the consumers (nonproductive part of the economy)
* Intermediate demand (Cx)
  + The demand for goods and services that the producers (sectors) need as inputs for their own production
    - Ex: electricity sector needs inputs from the water sector and vice versa
* Consumption matrix (C)
  + How much each sector consumes from other sectors in terms of percentages
* Column sum
  + The sum of the entries in a column

**Key Notes**

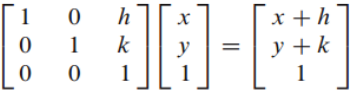
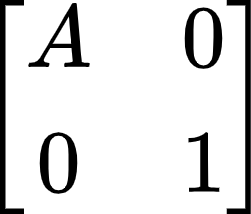
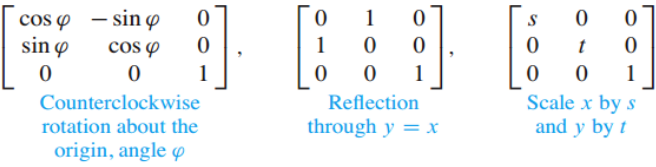
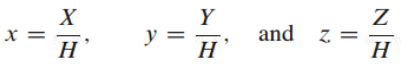
* The Leontief Input-Output Model (Production Equation)
  + 
  + Can be rewritten as:
    - (I - C)x = d
      * Solve for x (amount produced) by row reduction
    - x = (I - C)-1 \* d
      * Solve for x (amount produced) by multiplying
* For a good economy, the column sum of each sector should be **less than 1**
  + A sector should in general require less than one unit’s worth of inputs to produce one unit of output
* Output vector (x)
  + xi: entry i of vector x
    - Number of units produced by sector *i*
* Internal consumption (C)
  + 2 equivalent ways of defining entries of C where an entry is ci, j:
    - Sector *i* sends a proportion of its units to sector *j*
    - Sector *j* requires a proportion of the units created by sector *i*
* Consumption matrix (Cx)
  + Total output for each sector (per one unit) is the sum of the columns for each sector
* A Formula for (I - C)-1
  + As an economy is introduced to a demand vector, the equation starts off as:
    - x = d
  + However, production will require intermediate demand from other sectors, and then that intermediate demand will require more inputs from even more sectors
    - x = d + Cd + C2d + C3d + …
      * => (I + C + C2 + C3)d
  + 
    - We can approximate (I - C)-1 by making *m* as large as possible
      * Add as many intermediate demands as we can
* Economic Importance of Entries in (I - C)-1
  + Entries used to predict how the production *x* will have to change when the final demand *d* changes
    - Remember: x = (I - C)-1 \* d
  + The entries in each column of (I - C)-1 are the *increased* amounts each sector has to produce in order to satisfy *an increase of 1 unit* in the final demand

**Section 2.7: Applications to Computer Graphics**

**Definitions**

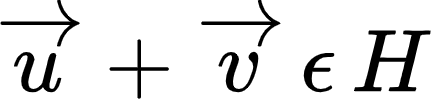
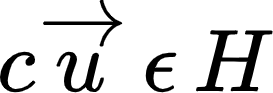
* Homogeneous coordinates
  + Each point (x, y) in R2 can be identified with the point (x, y, 1) on the plane in R3 that lies one unit above the xy - plane
* Composite transformations
  + Multiplication of 2 or more basic transformations

**Key Notes**

* Why do we use homogeneous coordinates?
  + Translations are **not** linear transformations
* Homogeneous coordinates are allowed to be **scalars**
  + (3, 5, 1) = (6, 10, 2)
* (x, y) -> (x + h, y + k)
  + Translation cannot be represented by an R2 matrix multiplication
  + (x, y, 1) -> (x + h, y + k, 1)
  + 
    - Translation not possible if we used a 2x2 identity matrix
* Linear transformations in R2 represented with homogeneous coordinates are written as partitioned matrices:
  +  where A is a 2x2 matrix
  + Examples
    - 
* Composite Transformations
  + “Add” on more transformation matrices **to the left** of the other transformations
    - First transformation is always the **rightmost** (modifies the x vector first)
* Homogeneous 3D Coordinates
  + (X, Y, Z, H) are homogeneous coordinates for (x, y, z) if H ≠ 0 and
    - 

**Section 2.8: Subspaces of Rn**

**Definitions**

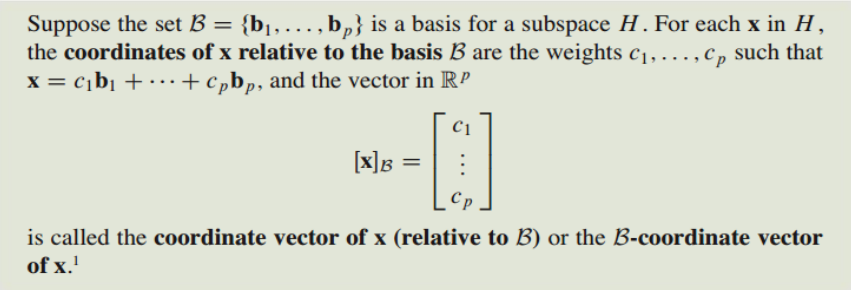
* Subset of Rn
  + Any collection of vectors that are in Rn
* Subspace of Rn
  + A subset H in Rn that has 3 properties:
    - The zero vector is in H
    - (closed under addition)
    -  (closed under scalar multiplication)
  + Subspace can be written as the **Span{}** of some amount of linearly independent vectors
* Column Space of a Matrix A **(*m x n*)**
  + Col A: the subspace of **Rm** spanned by ***{a1 , … , an}***
  + Essentially all the **pivot columns**
* Null Space of a Matrix A **(*m x n*)**
  + Null A: the subspace of **Rn** spanned by the set of all vectors ***x*** that solve ***Ax = 0***
* Basis for a Subspace ***H*** of ***Rn***
  + A linearly independent set in ***H*** that spans ***H***
    - **DOES NOT CONTAIN THE ZERO VECTOR (BECAUSE IT IS LINEARLY INDEPENDENT) UNLIKE THE SPAN**
* Standard Basis for ***Rn***
  + ***{e1 , … , en}***

**Key Notes**

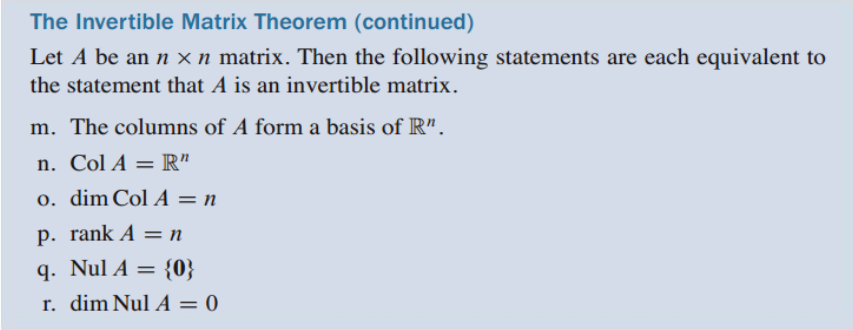
* If ***v1*** and ***v2*** are in ***Rn*** and ***H* = *Span{v1 , v2}***, then ***H*** is a subspace of***Rn***
  + ***v1*** and ***v2*** must be in **Rn** for this relation to work
* For ***v1***, … , ***vp*** in ***Rn***, the set of all linear combinations of ***v1***, … , ***vp*** is a subspace of ***Rn***
  + ***Span{v1 , … , vp}***= subspace spanned by ***v1 , … , vp***
* Is **b** in the column space of **A**?
  + Same as : Is **b** a linear combination of **A**?
  + Same as : Is **b** in the Span of **A**?
* Is ***H*** a subspace of **Rn?**
  + Basically asking if *H* has **n linearly independent** columns
  + Does *H* have **no free variables?**
* Subspaces vs. Bases
  + Subspaces => **Span**{v1 , … , vn}
    - **Includes the 0 vector**
  + Bases => {v1 , … , vn}
* Defining a basis for column space A
  + Number of entries for each vector = number of rows in matrix A
  + Number of vectors in the basis = number of pivot columns
  + What vectors can you include in the basis?
    - Scalar multiples
    - The identity matrix columns **only if** every column is pivotal in A
* Finding the Column Space
  + Row reduce the matrix
    - **Row operations do not affect linear dependence relations**
  + Determine the pivot columns
  + Create a basis/subspace using the pivot columns in the **original matrix**
    - Not the row reduced one
  + If **every** column is **linearly independent**, then the elementary vectors are included in the column space
    - Linear combinations of elementary vectors can get you **any column** of the original matrix
* Finding the Null Space
  + Determine all the free variables
  + Rewrite system in **parametric vector form**
  + Vectors created in parametric vector form generate the null space

**Section 2.9: Dimension and Rank**

**Definitions**

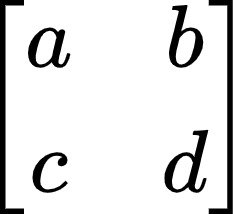
* Coordinates
  + Weights that map our vectors to get to some point in the span of the vectors
* Coordinate Vector
  + 
* Dimension of a Subspace
  + dim H: the number of vectors in a basis of H
  + dim{0} = 0
* Rank of a Matrix A
  + Dimension of the column space of A
  + Number of pivots in A

**Key Notes**

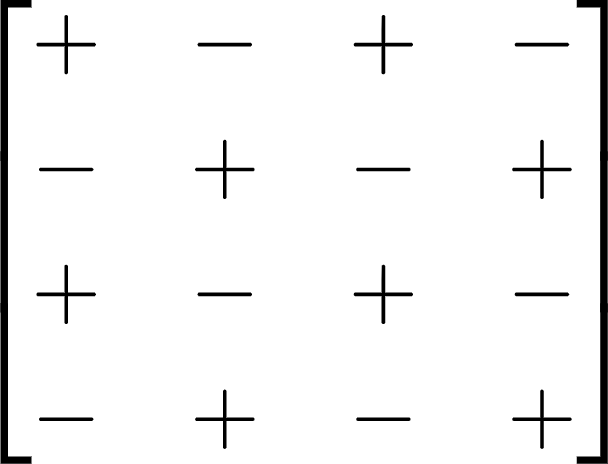
* Why we choose to write bases:
  + Each vector in ***H*** can be written in **only one way** as a linear combination of the **basis** vectors
* A plane through 0 in R3 is two-dimensional
  + 3x3 matrix A has 2 pivots
* A line through 0 in R2 is one-dimensional
  + 2x2 matrix A has one pivot
* Any two choices of bases of a non-zero subspace *H* have the **same dimension**
  + dim Rn = n
  + dim(Col A) = number of pivots
  + dim(Null A) = number of free variables
* dim(Col A) = rank A
* Rank Theorem
  + If A has n columns, then:
    - rank A + dim(Null A) = n
  + Number of pivots + number of free variables = number of columns
* Basis Theorem
  + Any two bases for a subspace have the same dimension (cardinality)
  + Many choices for the basis of a subspace
* Continuation of the Invertible Matrix Theorem with Rank
  + 

**Section 3.1: Introduction to Determinants**

**Definitions**

* *Ai j* submatrix
  + Delete the *i*th row and *j*th column of matrix A
  + Remaining elements will form the new submatrix
* Determinant for a 2x2
  + A =  -> det A = ad - bc
* Cofactor expansion
  + A way to solve determinants for square matrices that are 3x3 and greater

**Key Notes**

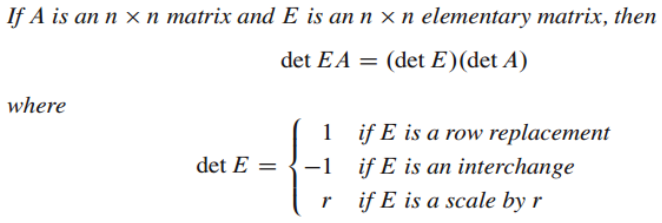
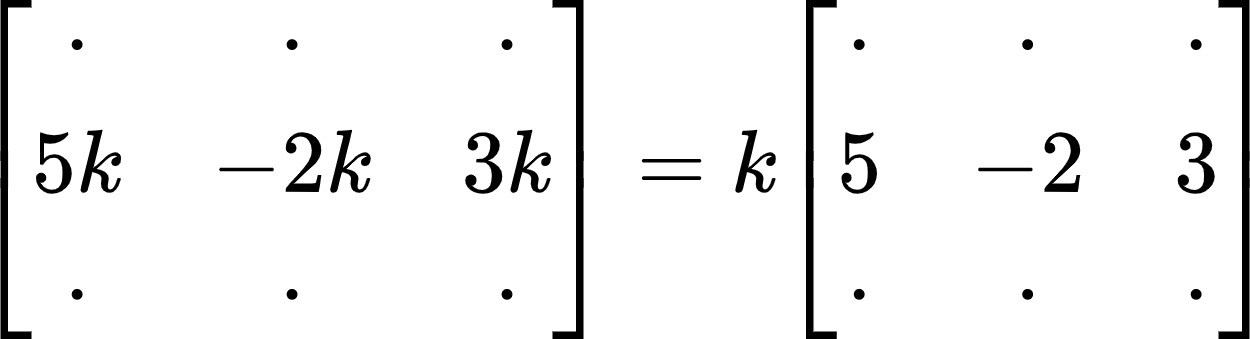
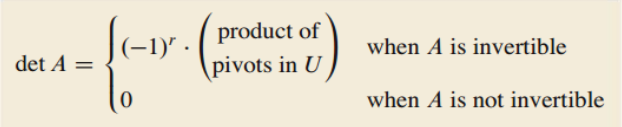
* Signs of cofactor expansions
  + Depends on position of element *ai j* in the matrix
  + 
* Shortcut for finding the determinant
  + Row reduce to REF
    - Effects of row operations on determinant covered in 3.2
  + Multiply all the numbers on the main diagonal

**Section 3.2: Properties of Determinants**

**Definitions**

* Column Operations
  + Same effect on determinants as row operations
  + This is true because the determinant of A = determinant of AT (transpose)

**Key Notes**

* Row operations on determinants
  + Row replacement: nothing
  + Row swap: multiply determinant by negative one
  + Row scale: multiply determinant by scale
* Summary of elementary matrices’ determinants
  + 
* More specific example of row scaling on determinants
  + 
  + Row divided by k
    - Determinant is multiplied by 1/k
* If A is invertible (every column is pivotal)
  + det A ≠ 0
* If A is not invertible
  + det A = 0
  + At least one entry on the main diagonal of REF is 0
* 
* When A is not invertible, the rows are linearly dependent
  + If A is square, then the columns are also linearly dependent
* det A = det AT
* det AB = (det A)(det B)
* det *A*-1 = 1 / (det *A*)

**Section 3.3: Volume and Linear Transformations**

**Definitions**

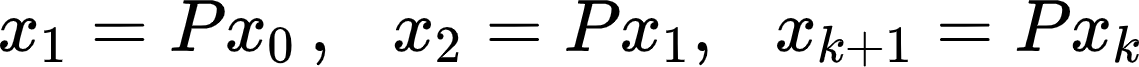
* Parallelepiped: a parallelogram in Rn where n > 2

**Key Notes**

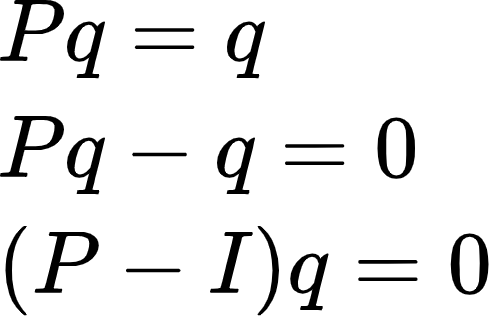
* If A is a 2x2 matrix:
  + Area of the parallelogram determined by the columns of A is **| det A |**
* If A is a 3x3 matrix:
  + Area of the **parallelepiped** determined by the columns of A is **| det A |**
* Row/column swaps and replacements do not affect the **absolute value** of the determinant
* Linear transformations on a parallelepiped
  + Area of T(S) = | det A | \* { area of S }
    - T: linear transformation determined by matrix A
    - S: parallelogram

**Section 4.9: Applications to Markov Chains**

**Definitions**

* Probability vector
  + A vector with **nonnegative** entries that **sum to 1**
* Stochastic matrix
  + A **square** matrix whose columns are **probability vectors**
* Markov Chain
  + A sequence of probability vectors {x0, x1, x2, …} together with a stochastic matrix {P} such that:
    - 
* Steady State Vector
  + A probability *q* such that *Pq = q*
  + Every stochastic matrix has a steady state vector
* Regular stochastic matrix
  + Stochastic matrix where some power of it will contain only **strictly positive entries**
    - Pk where all entries > 0

**Key Notes**

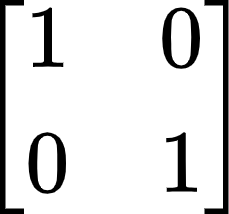
* How to find the next outcome of a Markov Chain?
  + Simply multiply P by xk to find xk+1
* How to find a steady state vector?
  + 
  + After finding a basis for the null space of (P - I) q = 0, remember to make sure that the **column sum is 1**
    - Steady state vector is a **probability vector**
* The initial state has **no effect** on the long term behavior of the Markov Chain

**Section 5.1: Eigenvectors and Eigenvalues**

**Definitions**

* Eigenvector of an *n x n* matrix A:
  + **Nonzero** vector x such that Ax = λx for some scalar λ
* Eigenvalue of A:
  + A scalar λ where there is a **nontrivial solution** x of Ax = λx
* Eigenspace of an eigenvalue
  + Contains the zero vector and **all eigenvectors corresponding to λ**

**Key Notes**

* Determine if a vector *x* is an eigenvector
  + A\*x => see if product is a scalar multiple of *x*
* Finding the eigenvector from an eigenvalue (7)
  + Solve (A - 7I)x = 0
  + Then, do the parametric vector form of what you have left
* Finding the eigenvalue λ
  + Solve (A - λI)x = 0 for a **nontrivial solution**
  + Find the set of all solutions to the **null space** of (A - λI)
* Eigenvalues of a **triangular matrix** are the entries on the **main diagonal**
* 0 is an eigenvalue of A if and only if A is **not invertible**
  + Ax = 0x
  + Ax = 0: x is a nontrivial solution if A is not invertible
* Eigenvectors that correspond to distinct eigenvalues are **linearly independent**
  + Opposite is not always true
    - : eigenvectors are linearly independent but have the **same eigenvalue**

**Section 5.2: The Characteristic Equation**

**Definitions**

* The Characteristic Polynomial:
  + det(A - λI)
* The Characteristic Equation
  + det(A - λI) = 0
* Trace
  + Sum of the diagonal entries in a matrix
* Algebraic Multiplicity of an Eigenvalue
  + The number of times the eigenvalue shows up as roots of the characteristic polynomial
* Geometric Multiplicity of an Eigenvalue
  + The dimension of Null (A - λI) for a given eigenvalue λ

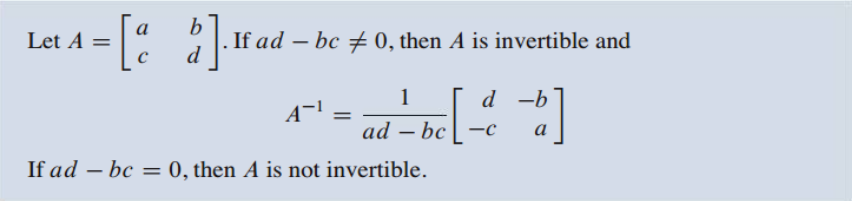
**Key Notes**

* How to find eigenvalues?
  + Solve (A - λI)x = 0 for a **nontrivial solution**
  + Find the set of all solutions to the **null space** of (A - λI)
* Continuation of IMT
  + For A: *n x n* matrix, A is invertible if and only if:
    - The number 0 **is not** an eigenvalue of A
    - The determinant of A **is not 0**
* Finding the characteristic polynomial using **trace** and **determinant** for a characteristic polynomial of **2**
  + λ2 - λ(trace) + det A
* **Warnings:**
  + Cannot determine the eigenvalues of a matrix from its reduced from
  + Row operations **change** the eigenvalues

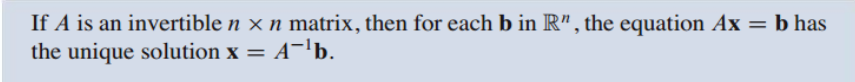
**Theorems**

**Chapter 2**

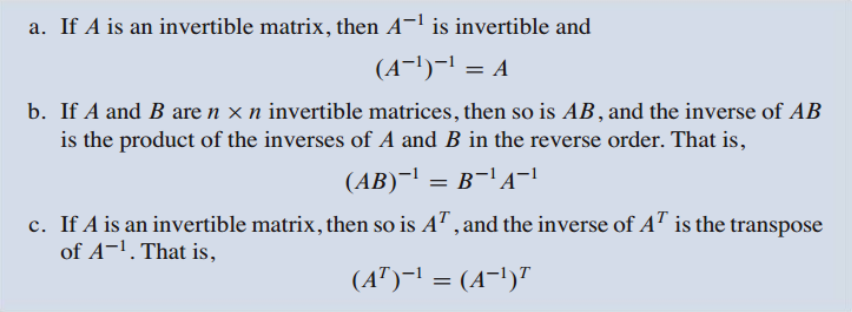
**Theorem 4: Finding the Inverse of a 2x2 Matrix**

****

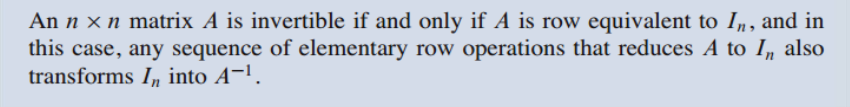
**Theorem 5: Alternate Method of Finding the Solution Set**

****

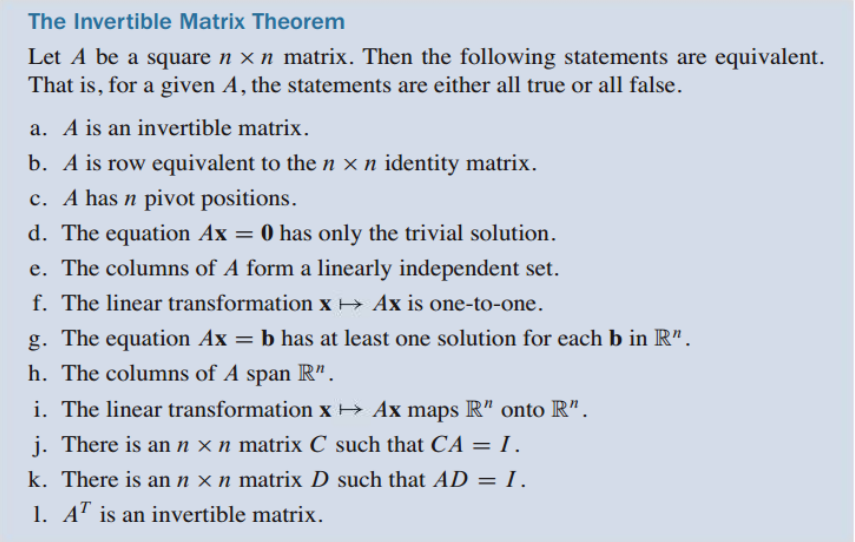
**Theorem 6: Properties of Invertible Matrices**

****

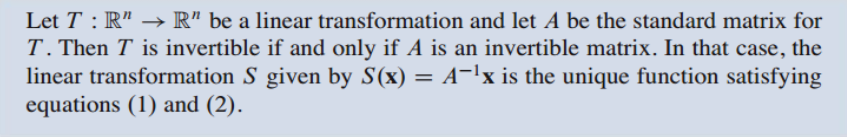
**Theorem 7: Finding the Inverse of a Matrix**

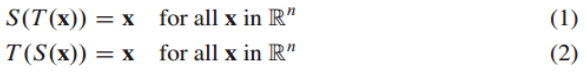
****

**Theorem 8: The Invertible Matrix Theorem**

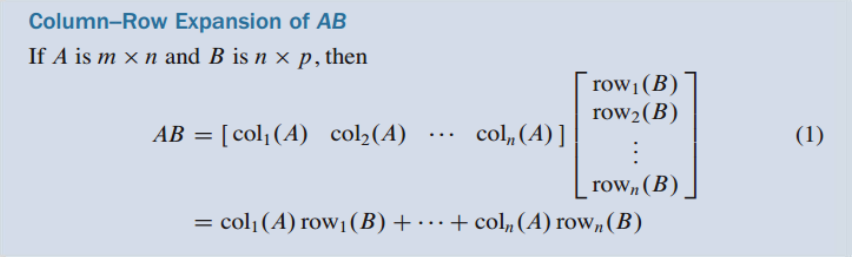
****

**Theorem 9: Invertible Linear Transformations**

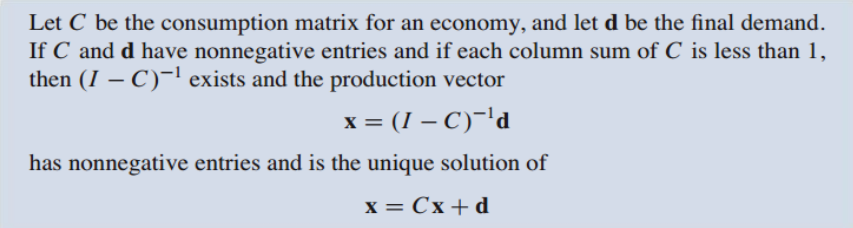
****

****

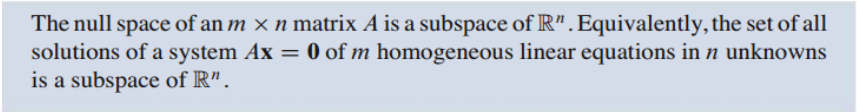
**Theorem 10: Column-Row Expansion of AB**

****

**Theorem 11: Solving the Output Vector (x)**



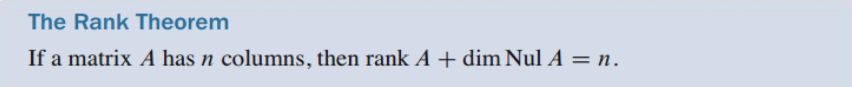
**Theorem 12: Finding the Null Space of Matrix A**

****

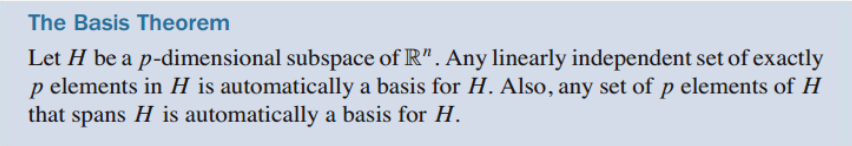
**Theorem 13: Determining the Column Space of Matrix A**

****

**Theorem 14: The Rank Theorem**

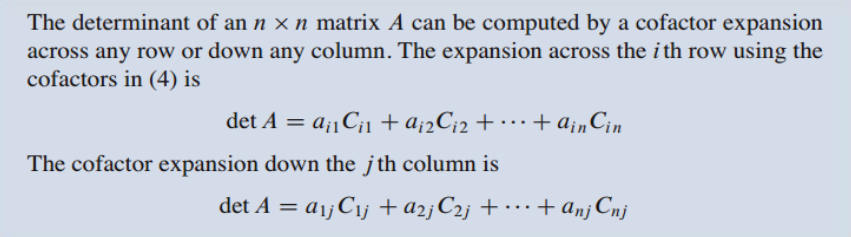
****

**Theorem 15: The Basis Theorem**

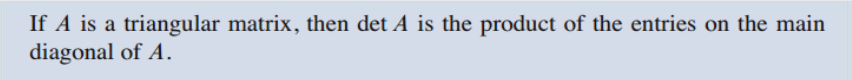
****

**Chapter 3**

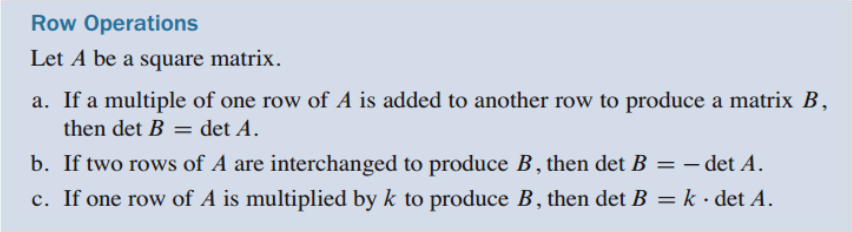
**Theorem 1: Cofactor Expansion to find Determinants**

****

**Theorem 2: Shortcut to Computing Determinant**

****

**Theorem 3: Row Operations on Determinants**

****

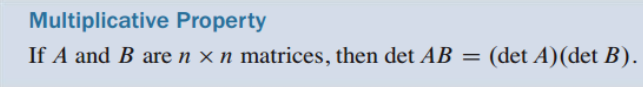
**Theorem 4: IMT DLC: Determinant**

****

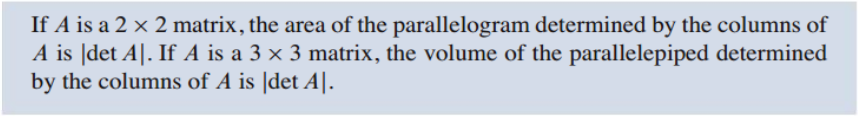
**Theorem 5: Transpose Equivalence for Determinants**

****

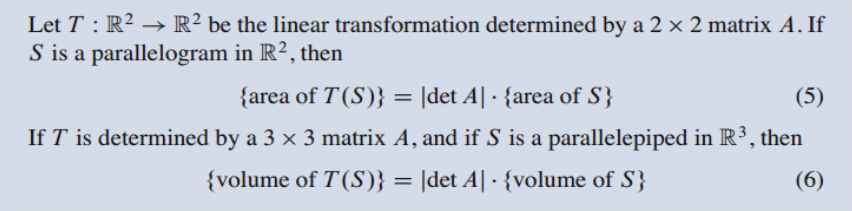
**Theorem 6: Multiplicative Property of Determinants**

****

**Theorem 9: Determinants as Area and Volume**

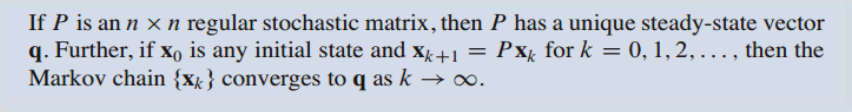
****

**Theorem 10: Linear Transformations on Area/Volume**

****

**Chapter 4**

**Theorem 18: Long-term Behavior of a Markov Chain**

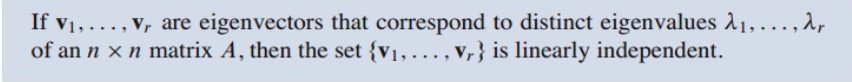
****

**Chapter 5**

**Theorem 1: Eigenvalues of a Triangular Matrix**

****

**Theorem 2: Eigenvectors for Distinct Eigenvalues**

****