FINAL EXAM LOGISTICS

The final exam will be held on **Monday July 25th from 9am-11am**. Please arrive at the computer lab (Tech E117) about 10 minutes before 9am. The exam is comprehensive (it will involve all material we covered in this class). Here is a summary of some (not necessarily all) of the concepts that we have learned this summer.

LINEAR ALGEBRA

Systems of linear equations
Further reading: Kolman ch. 1.1 (course webpage)

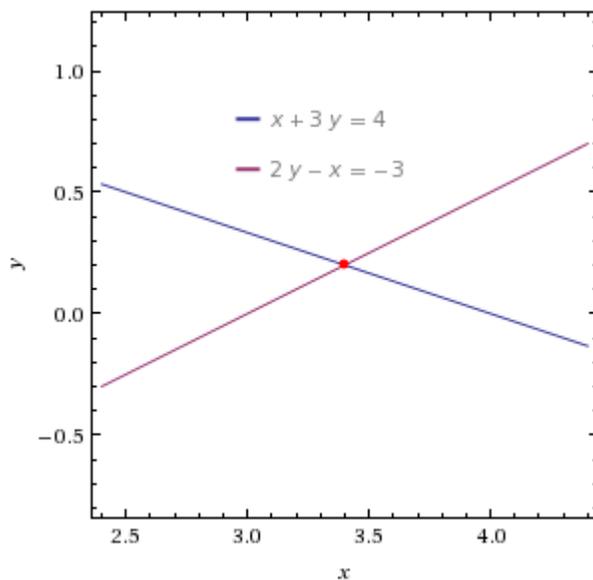A system of linear equations (or a linear system) is a set of equations of the form

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

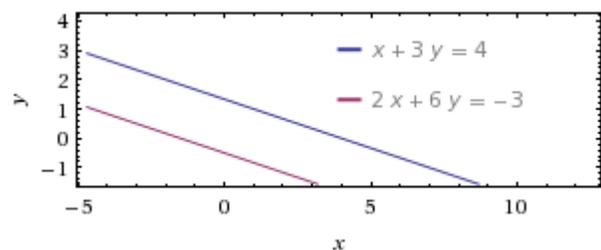$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$

$$\cdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

Where $a_{11}, a_{12}, \ldots$ are constants (real numbers in this class). The solution to a linear system is a set of numbers $x_1, x_2, \ldots$ which make all the equations simultaneously true. In 2-dimensions, each equation represents a straight line in $\mathbb{R}^2$. The solution is the point (x, y) where all the lines intersect. If two equations represent lines which are parallel and non-overlapping, there are no solutions. If two equations represent lines which are parallel and overlapping, there are infinite solutions (any point along either line satisfies both equations).

A 2x2 linear system with one unique solution   A 2x2 linear system with no solutions

This categorization of solutions holds for all linear systems, regardless of the number of variables or equations.

> **A system of linear equations has either: (1) no solution, or (2) exactly one unique solution, or (3) infinitely many solutions.**

Sample problems
- Kolman ch. 1.1 exercises 1-14 (you should be able to graph the 2D systems and determine if there is zero, one, or infinitely many solutions.
- Kolman ch. 1.1 exercises 15, 16.
- Quiz 1 Problem 1

Row reduction and reduced echelon form
Further reading: D.C. Lay ch. 1.2

We can transform a matrix into a **row-equivalent matrix** by applying any of three possible **row operations**: (1) multiply every entry in a row by a non-zero constant, or (2) swap the positions of two rows, or (3) replace a row with itself plus a non-zero multiple of another row. By applying row operations in series we can transform any matrix into an echelon matrix or reduced echelon matrix

> A rectangular matrix is in **echelon form** if it has the following three properties:

1. All nonzero rows are above any rows of all zeros.
2. Each leading entry of a row is in a column to the right of the leading entry of the row above it.
3. All entries in a column below a leading entry are zeros.

> If a matrix in echelon form satisfies the following additional conditions, then it is in **reduced echelon form**

4. The leading entry in each nonzero row is 1.
5. Each leading 1 is the only nonzero entry in its column.

A **pivot position** in a matrix A is a location in A that corresponds to a leading 1 in the reduced echelon form of A. A **pivot column** is a column of A that contains a pivot position.

Solving systems of linear equations using row reduction
A linear system

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$

$$\ldots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

can be rewritten in an augmented matrix containing the coefficients a_{ij} and solution vector b_i:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{bmatrix}$$

Each column (except the last) in the augmented matrix correspond to a variable x_1, x_2, etc. Each row in the augmented matrix corresponds to an equation. The variables corresponding to columns containing a **pivot position** in the augmented matrix are called **basic variables.** The variables corresponding to columns with no **pivot position** are called **free variables.**

You can tell by looking at the echelon form of a matrix to determine if the corresponding linear system has zero, one, or infinite solutions.

1. If there is a pivot in every column, except the last, of the augmented matrix, the system has one unique solution. If the matrix is in reduced echelon form, the entries in the last column is the solution.
2. If there are any columns (except the last column) **without a pivot position** (i.e. free variables), and there are no contradictions (i.e. a row such as $\begin{bmatrix} 0 & 0 & 2 \end{bmatrix}$) then there are infinite solutions
3. If there are any columns (except the last column) **without a pivot position** (i.e. free variables), and there is one or more contradictions, then there are no solutions.

Where to find practice problems
- D.C. Lay ch 1.2 exercises 3, 4 (row reduction); 7-14 (finding solutions to linear systems)
- Quiz 2 Problem 1
- Homework 2, first hand-written problems

Matrix multiplication and the matrix inverse
Further reading: D.C. Lay ch. 2.1

In class way learned how to perform **matrix multiplication**. See the posted reading or slides for a refresher of the procedure for multiplying matrices. Remember the following important points about matrix multiplication:

1. Matrix multiplication is **not commutative**, i.e. $AB \neq BC$.
2. Matrices can only be multiplied if their inner dimensions match, e.g. for A=[4x2 matrix] and B=[2x3] matrix, we can perform AB=[4x3 matrix] but not BA.
3. Matrix multiplication is **associative**, i.e. $A(BC) = (AB)C$.
4. Matrix multiplication is a **linear operator**, i.e. for appropriately sized matrices A, B, and C, and real numbers r and s, $A(rB + sC) = rAB + sAC$.

Matrix inverse
A square matrix A is invertible if there is a matrix A^{-1} such that $A^{-1}A = AA^{-1} = I$, where I is the identity matrix (ones along the diagonal, zeros everywhere else). The identity matrix acts like the number 1 in that $IA = AI = A$ for any matrix A. Remember:

- A matrix **must** be square to be invertible (i.e. same number of rows and columns).
- Not every square matrix is invertible.

We saw how to calculate the inverse of a matrix using row reduction. First, create a larger matrix $\begin{bmatrix} A & I \end{bmatrix}$ with the matrix to invert on the left and the identity matrix on the right. Then use row operations to find

the reduced row echelon form of the entire matrix. If the matrix is invertible, you should be left with $[I \quad A^{-1}]$.

Solving matrix equations using matrix inverses

You can algebraically solve matrix equations by using matrix inverses. For example, solve $Ax = b$ for x.

$$Ax = b$$

$$A^{-1}Ax = A^{-1}b \qquad \text{Multiply both equations \textbf{on the left} by } A^{-1}$$

$$Ix = A^{-1}b \qquad A^{-1}A \text{ becomes } I \text{ due to the definition of the inverse}$$

$$x = A^{-1}b \qquad Ix \text{ simply returns } x \text{ (identity matrix acts like a "1")}$$

Where to find practice problems
- Matrix multiplication: D.C. Lay ch. 2.1 exercises 1 and 2, HW_01
- Matrix inverses: D.C. Lay ch. 2.2 exercises 1-8, 14, 16, 20 (hard), 29-32
- Homework 3

Linear models and linear regression
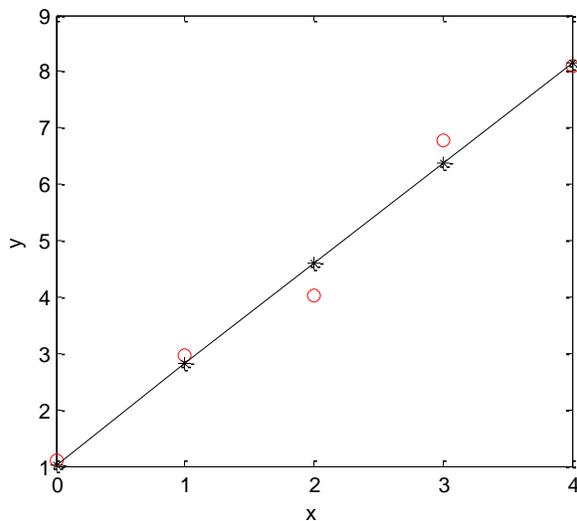
A linear model is an equation designed to explain experimental data, having the form

$$\hat{y} = a_0 f_0(x) + a_1 f_1(x) + a_2 f_a(x) + \cdots$$

For n data points (x, y), the equation above can be written as a matrix equation with n rows for the model predictions $\hat{y}_i$ as $Xa = \hat{y}$. The "best-fit" coefficients $a_i, \ldots$ are found by solving the "normal equations"

$$X^T X a = X^T y$$

where y are the actual observed values, while $\hat{y}$ are the predicted values from the model. In the example to the left, the red circles correspond to the measured values, y, while the black stars correspond to the model predicted values $\hat{y}$.

In this class, we are mostly concerned with creating the "design matrix" X. For examples, see the posted reading on linear regression, as well quiz 3.

MATLAB

Scalar arithmetic and assignment

The assignment operator "=" in MATLAB is used for storing values in variables. For example

```
a = 1;
```
Assign the value 1 to the variable a

```
b = a + 2
```
Evaluate the expression 1 + 2 and assign the result, 3 to b

You can evaluate mathematical expressions using the operators "*" (multiplication), "/" (division), "+" (addition), "-" (subtraction), and "^" (raising to a power).

Remember:

- Every time you use the assignment operator, you replace the previous value stored in that variable.
- The term to the left of the assignment operator must be a valid **target** for assignment, that is either a variable name (e.g. `a = …`), or an address in an array (e.g. `a(1, 2) = …`).

Matrix creation and indexing

Create matrices using square brackets [] and separate items on the same row with "," and rows with ";". Join arrays inside square brackets using "," to join horizontally and ";" to join vertically.

`A = [1, 2; 3, 4]` $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

`B = [A; [5, 6]]` $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

`C = [B, [7; 8; 9]]` $C = \begin{bmatrix} 1 & 2 & 7 \\ 3 & 4 & 8 \\ 5 & 6 & 9 \end{bmatrix}$

Use index notation to access values inside of an array. See the MATLAB "cheat sheet" posted on the course webpage for more details. For practice, consider the following problems. Suppose A is defined in MATLAB as

$$A = \begin{bmatrix} 3 & 6 & 9 & 3 & 1 & 8 & 6 & 7 & 8 \\ 4 & 6 & 2 & 0 & 8 & 7 & 2 & 2 & 2 \\ 10 & 8 & 8 & 8 & 7 & 9 & 8 & 5 & 6 \\ 7 & 1 & 1 & 3 & 9 & 8 & 7 & 4 & 2 \\ 8 & 6 & 10 & 7 & 9 & 8 & 6 & 8 & 6 \end{bmatrix}$$

Determine the output of the following MATLAB statements, or if the statement would result in an error, explain why.

```
a) A(:, [2, 4])
b) A(1:7, [2, 4])
c) A([3, 5], [1, 1]) + A([4, 2], [6, 7])
d) A(3:end, 5:end)
e) [A(:, 2:end), A(:, 1)]
f) [A(:, 2:end), A(1, :)]
g) [A(:, 2:end)'; A(1, 1:2:9)]
```

Conditional expressions and `if` statements

A Boolean statement is an expression which returns either **True** or **False**. Boolean statements are typically created in MATLAB using comparison operators such as "==" (equal to), ">=" (greater than or equal to), or "<" (less than). In addition, the operators "&" (and), "|" (or), and "~" (not) allow us to combine multiple Boolean expressions into compound expressions.

For practice, try filling in the rest of the following truth table:

A	B	A | B	A & B	~(A | B)	A | ~B	~(A|B) | B
T	T	T	T			
T	F	T	F			
F	T	T	F			
F	F	F	F			

In MATLAB, if statements are used to execute code only if a certain condition is met

```
if a < 4

    do something

elseif a < 10

    do something else

else

    do a different thing if neither of the above conditions were true

end
```

You can include as many elseif statements as you want and you do not need to include an else statement.

List iteration with `for` loops

For loops are used for repeating code a fixed number of times. Each time, the iterator variable takes on the next value in the list. Remember, the iterator changes each time the loop runs, **but the list cannot change**.

Examples of problems to solve with a for loop:

- adding up all the numbers in a vector
- finding all the factors of a composite number (i.e. try to divide an integer by every integer less than itself)
- performing an operation on every element in a list

Conditional iteration with `while` loops

While loops are used for repeating code until a certain condition (i.e. a Boolean expression) is no longer true. In addition the examples we've seen in class (homework, quizzes, labs), while loops could be used for:

- checking whether a number N is prime. Create a variable f which increases by one each time the loops runs, and try to divide N by f each time and check whether the remainder is zero).
- reading all the lines in a file, until you reach the end
- getting keyboard input from the user (until the user presses a certain key)