ME751
Advanced Computational Multibody Dynamics

Introduction

January 19, 2010

© Dan Negrut, 2010
ME751, UW-Madison

Dan Negrut
University of Wisconsin, Madison

Before we get started…

- Today:
  - Discuss Syllabus
  - Other schedule related issues
  - Discuss class structure
  - Review
    - Basic Matrix Algebra
    - Lagrange Multiplier Theorem

Instructor: Dan Negrut

- Polytechnic Institute of Bucharest, Romania
  - B.S. – Aerospace Engineering (1992)
- The University of Iowa
  - Ph.D. – Mechanical Engineering (1998)
- MSC.Software
  - Product Development Engineer 1998-2004
- The University of Michigan
  - Adjunct Assistant Professor, Dept. of Mathematics (2004)
- Division of Mathematics and Computer Science, Argonne National Laboratory
  - Visiting Scientist (2005, 2006)
- The University of Wisconsin-Madison, Joined in Nov. 2005
  - Research: Computer Aided Engineering (tech lead, Simulation-Based Engineering Lab)
  - Focus: Computational Dynamics (http://sbel.wisc.edu/)

Good to know…

- Time 9:30 – 10:45 AM [Tu, Th]
- Room 2106
- Office 2035ME
- Phone 608 890-0914
- E-Mail negrut@engr.wisc.edu
- Course Webpage:
  - https://learnuw.wisc.edu – solution to HW problems and grades
  - http://sbel.wisc.edu/Courses/ME751/2010/index.htm - for slides, audio files, examples covered in class, etc.

- Grader: Naresh Khude (khude@wisc.edu)

- Office Hours:
  - Monday 3 – 4 PM
  - Wednesday 3 – 4 PM
  - Friday 3 – 4 PM
  - Stop by any other time in the afternoon, I'll try to help if I'm in

Text

- **Edward J. Haug: Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods (1989)**

  - Book is out of print
  - Author provided PDF copy of the book, available free of charge at Learn@UW
  - The material in the book will be supplemented with notes
  - Available at Wendt Library (on reserve)
  - We'll cover Chapters 9 and 11

Other Sources of Information Used

- Dynamics of Multibody Systems, by Ahmed A. Shabana, 3rd ed., 2007
- Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, by U. Ascher and L. Petzold, SIAM, 1998
- Solving Ordinary Differential Equations I: Nonstiff Problems, by E. Hairer, S. Norsett, G. Wanner, 1993
- Solving Ordinary Differential Equations II: Stiff and differential-algebraic Problems (Second Revised Edition) by E. Hairer and G. Wanner, 2002
- Numerical Methods in Multibody Dynamics, E. E. Soellner and C. Fuhrer, 2002 (out of print)
- CUDA Programming Guide, Version 2.3, NVIDIA Corporation, July 2009

History of Class

- This class started back in the day by Professor John Uicker
- Class used to be called “Matrix Methods in the Design and Analysis of Mechanisms”
- Changed title to “Advanced Computational Multibody Dynamics”
  - Not official, need to follow up on some bureaucratic aspects
- Emphasis on the use of the computer to predict the dynamics (time evolution) of system of mutually interacting rigid bodies
  - Key words: **predict**, **time evolution**, **rigid bodies, mutually interacting**

Course Related Information

- Offering class for the first times, rough edges super likely to show up
- Trying to alleviate this: I plan to upload the lectures + sound
- Why bother?
  - PPT notes + lack of book means you might find yourself in a situation where things don’t add up.
  - Good if you miss class
  - Allows me to review my teaching and hopefully improve

Course Related Information [Cntd.]

- Handouts will be printed out and provided before most of the lectures
- The class has a website: http://sbel.wisc.edu/Courses/ME751/2010/index.htm
- PPT lecture slides will be made available online at lab website
- Homework solutions will be posted at Learn@UW
- Grades will be maintained online at Learn@UW
- Syllabus will be updated as we go and will contain info about
  - Topics covered
  - Homework assignments and due dates
  - Midterm exam date
- Syllabus available at the course website

Grading

Homework	40%
Midterm Exam	20%
Final Project	40%
	100%

NOTE:

- Score related questions (homeworks/exams) must be raised prior to next class after the homeworks/exam is returned.

Homework

- Shooting for weekly homeworks
  - Assigned at the end of each class
  - Typically due one week later, unless stated otherwise
  - No late homework accepted
  - I anticipate to have about 11 or 12 homeworks

Exams & Final Project

- One midterm exams, on 04/22
  - Exam will be in the evening, starting at 7:15 pm

- No final exam

- Final Project
  - You have to select a topic by 03/11
  - Final project is individual
  - I will provide an overview of topics selected by you on 03/18
  - You are encouraged to select a topic that ties into your research
  - In general, anything is ok as long as it is related to one of the following
    - Rigid body dynamics
    - Numerical integration methods for dynamic systems (not necessarily mechanic)
    - High performance computing

Final Project, Example Topics

- Investigation of animating rotation with Quaternion curves
- Practical implications of the parameterization of rotations using the exponential map
- Parallel collision detection on the GPU
- Comparison of penalty and DVI approaches for frictional contact in rigid body dynamics
- Extension of the functionality of any open source dynamics engine (ChronoEngine, IMP, etc.)
- On the use of iterative methods in multibody dynamics
- Rigid-Deformable body co-simulation for applications in biomechanics
- Methods of uncertainty quantification in multibody dynamics
- Advanced modeling in open source dynamics engine with validation in ADAMS
- Investigation of a ADAMS-MATLAB co-simulation approach for mechatronics applications
- Development of a computer game using an open source physics engine

Final Project, Evaluation

- Each student will have a one hour presentation of his/her final project
  - 40 minute PPT presentation reports on your contributions/innovations/achievements
  - 20 minutes set aside for Q&A
- Each students should schedule after consultation with the instructor the Final Project presentation during the last week of class or finals' week
- Your presentation will be videotaped and posted at Learn@UW for the rest of the students to review if interested
- Any student can participate in any final project presentation of interest
- To allow you to focus on the final project no HW will be assigned and no new material covered in the last two weeks of class

Last Two Weeks of Class

- Trying to bring in colleagues from industry
- Tentative schedule, last two weeks:
  - April 27 – Dr. Andrei Schaffer, of MSC.Software
  - April 29 – Dr. Jonathan Cohen, of NVIDIA Research
  - May 4 – Field Trip (John Deere and NADS) – not confirmed yet
  - May 6 – Richard Tonge, of NVIDIA PhysX

Scores and Grades

<u>Score</u>	<u>Grade</u>
94-100	A
87-93	AB
80-86	B
73-79	BC
66-72	C
55-65	D

- Grading will <u>not</u> be done on a curve
- Final score will be rounded to the nearest integer prior to having a letter assigned
  - 86.59 becomes AB
  - 86.47 becomes B

Class Goals

- Given a general mechanical system, understand how to generate in a **systematic** and **general** fashion the equations that govern the time evolution of the mechanical system
  - These equations are called the equations of motion (EOM)

- Have a basic understanding of the techniques (called numerical methods) used to solve the EOM
  - We'll rely on MATLAB to implement/illustrate some of the numerical methods used to solve EOM

- Focus on rigid bodies connected through joints and mutually interacting through contact and friction

Why/How do bodies move?

- Why?
  - The configuration of a mechanism changes in time based on **forces** and **motions** applied to its components
    - Forces
      - Internal (reaction forces)
      - External, or applied forces (gravity, compliant forces, etc.)
    - Motions
      - Somebody prescribes the motion of a component of the mechanical system

- How?
  - They move in a way that obeys Newton's second law
    - Caveat: there are *additional* conditions (constraints) that need to be satisfies by the time evolution of these bodies, and these constraints come from the joints that connect the bodies (to be covered in detail later…)

30,000 Feet Perspective…

MECHANICAL SYSTEM
=
BODIES + JOINTS + FORCES

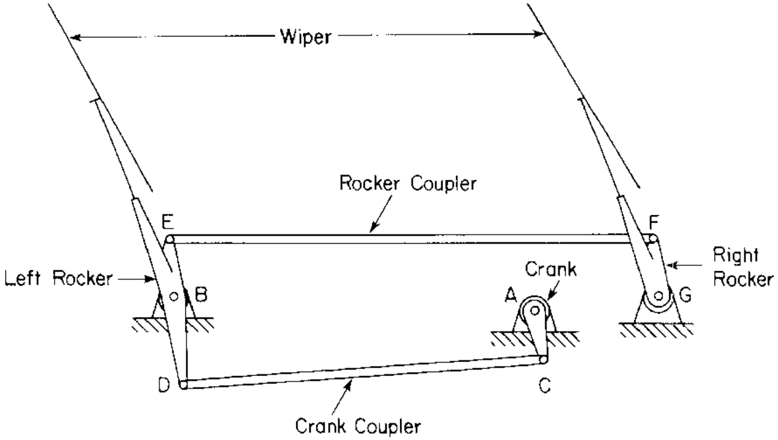
THE SYSTEM CHANGES ITS CONFIGURATION IN TIME

WE WANT TO BE ABLE TO PREDICT & CHANGE/CONTROL HOW SYSTEM EVOLVES

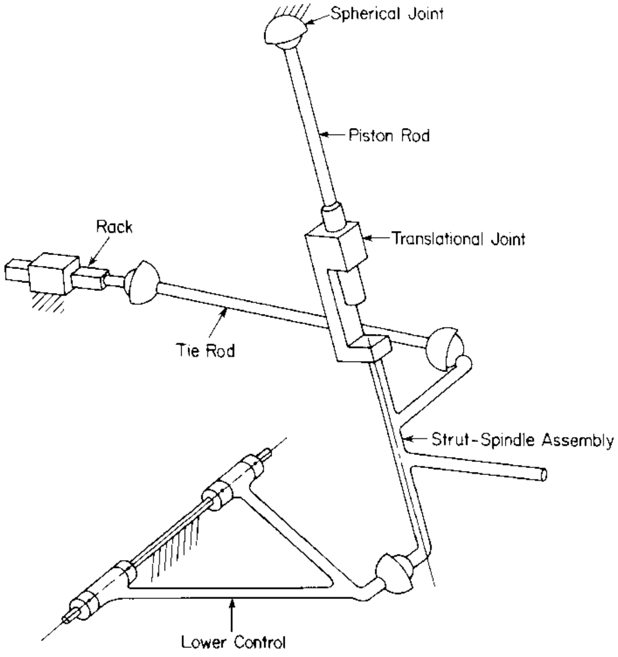
Applications of Multibody Dynamics

- Mechanical Engineering
- Biomechanics
- Rheology
- Computer Gaming
- Movie Industry

Dynamic Systems, Examples

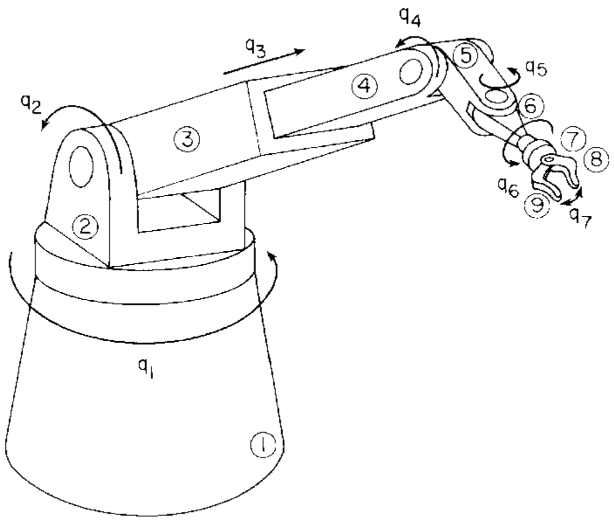


Windshield wiper mechanism

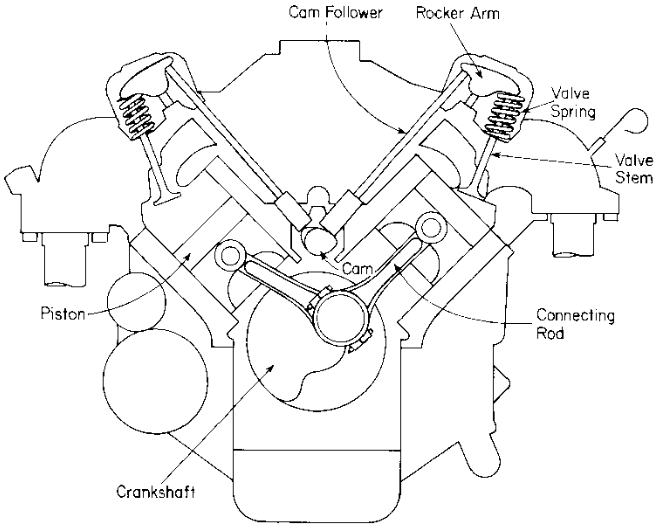


McPherson Strut Front Suspension

More examples …



Robotic Manipulator



Cross Section of Engine

Nomenclature

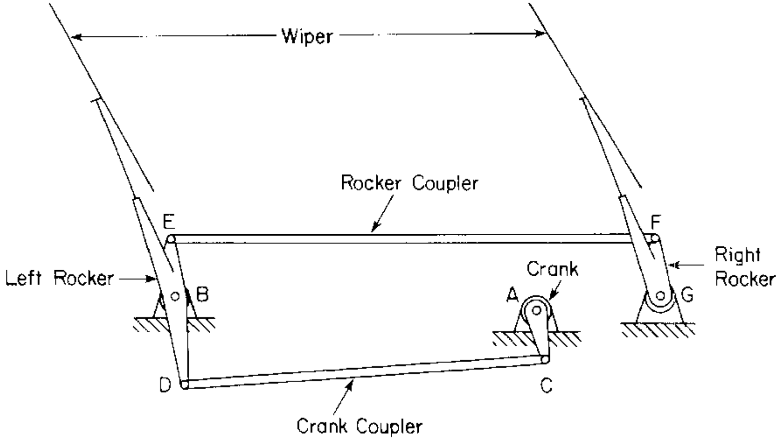
- Mechanical System, definition:
  - A collection of interconnected rigid **bodies** that can move relative to one another, consistent with **joints** that limit relative motions of pairs of bodies

- Why type of analysis might one be interested in in conjunction with a mechanical system?
  - Kinematics analysis
  - Dynamics analysis
  - Inverse Dynamics analysis
  - Equilibrium analysis

23

Kinematics Analysis

- Concerns the motion of the system **independent** of the forces that produce the motion
- Typically, the time history of one body in the system is prescribed
- We are interested in how the rest of the bodies in the system move
- Requires the solution linear and nonlinear systems of equations



Windshield wiper mechanism

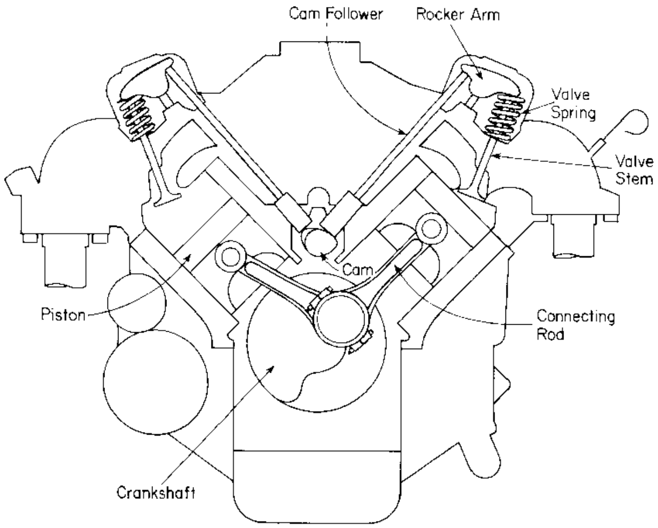
Dynamics Analysis

- Concerns the motion of the system that is due to the action of applied forces/torques

- Typically, a set of forces acting on the system is provided. Motions can also be specified on some bodies

- We are interested in how each body in the mechanism moves

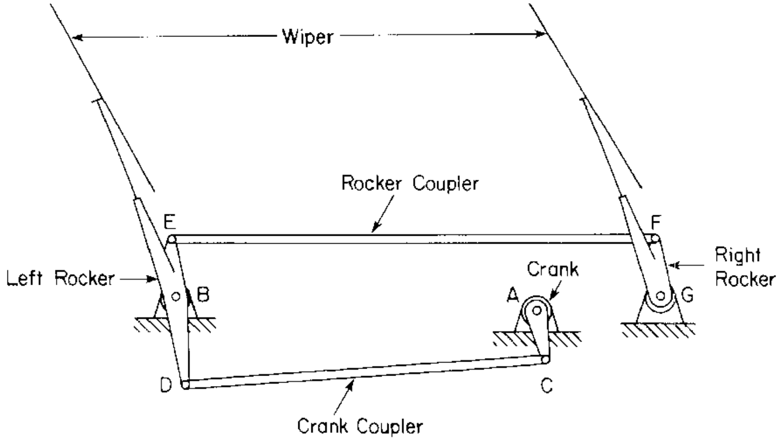
- Requires the solution of a combined system of differential and algebraic equations (DAEs)



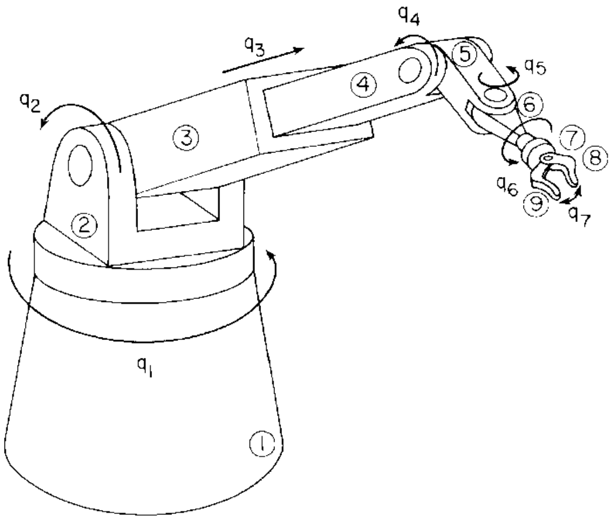
Cross Section of Engine

Inverse Dynamics Analysis

- It is a hybrid between Kinematics and Dynamics
- Basically, one wants to find the set of forces that lead to a certain desirable motion of the mechanism
- Widely used in Controls…



Windshield wiper mechanism



Robotic Manipulator

What is the slant of this course?

- When it comes to dynamics, there are several ways to approach the solution of the problem, that is, to find the time evolution of the mechanical system

  - The ME240 way
    - Very much solution is approached on a case-by-case fashion
    - Typically works for small problems, not clear how to go beyond textbook cases

  - Use a graphical approach
    - Intuitive
    - Doesn't scale particularly well
    - Somewhat cumbersome to bring computers into it

  - Use a computational approach (emphasized in this class)
    - Leverages the power of the computer
    - Relies on a unitary approach to finding the time evolution of any mechanical system
      - Sometimes the approach might seem an overkill, but it's general, and remember, it's the computer that does the work and not you

The Computational Slant…

- Recall title of the class: "Advanced Computational Multibody Dynamics"

- The topic is approached from a computational perspective, that is:
  - A) Establish a simple way of describing mechanical systems

  - B) Given a description of a system from A), produce a simple but general way to formulate the equations of motion (equations that govern the time evolution of the system)

  - C) Identify numerical solution methods capable of approximating the solution of the equations of motion (Newton Raphson, Euler, Runge-Kutta, etc.)

  - D) Post-process results to gain insights into the dynamics of the system (Usually it this requires a GUI for plotting results and generating movies with the dynamics of the system)

Overview of the Class

- Introduction
  - General considerations regarding the scope and goal of Kinematics and Dynamics
  - Review of Linear Algebra: Focus on geometric vectors and matrix-vector operations
  - Review of Calculus: partial derivatives, handling time derivatives, chain rule
- Spatial Cartesian Kinematics
  - Learn how to position and orient a rigid body in 3D space
  - Introduces the concept of kinematic constraint as the mathematical building block used to represent joints in mechanical systems
  - Poses the Kinematics Analysis problem, basically Chapter 9 of the Haug book
- Spatial Cartesian Dynamics
  - Learn how to formulate the equations that govern the time evolution of a mechanical system
  - Basically Chapter 11 of the Haug book
- Numerical Integration Methods
  - Methods for the solution of ordinary differential equations (ODEs)
  - Methods for the solution of differential algebraic equations (DAEs)
- Methods for handling frictional contact problems in rigid body dynamics
  - Penalty Methods
  - Differential Variational Inequality Methods
- High Performance Computing
  - Why/How
  - Intro to GPU Computing

MATLAB, ADAMS, and C

- MATLAB will be used on a couple of occasions for HW
  - It'll be the vehicle used to formulate and solve the equations governing the time evolution of mechanical systems

- ADAMS is a commercial software package used in industry
  - Two short ADAMS presentations will be made in class: 02/04/2010
  - A set of ADAMS tutorials are available on-line
    - http://sbel.wisc.edu/Courses/ME751/2010/tutorialsADAMS.htm
  - Currently the most widely used multibody dynamics simulation software

- C will be used when discussing opportunities for High-Performance Computing (HPC) in determining the dynamics of mechanical systems

ADAMS

- Automatic Dynamic Analysis of Mechanical Systems

- It says Dynamics in name, but goes beyond that
  - Kinematics, Statics, Quasi-Statics, etc.

- Philosophy behind software package
  - Offer a pre-processor (ADAMS/View) for people to be able to generate models
  - Offer a solution engine (ADAMS/Solver) for people to be able to find the time evolution of their models
  - Offer a post-processor (ADAMS/PPT) for people to be able to animate and plot results

- It now has a variety of so-called vertical products, which all draw on the ADAMS/Solver, but address applications from a specific field:
  - ADAMS/Car, ADAMS/Rail, ADAMS/Controls, ADAMS/Linear, ADAMS/Hydraulics, ADAMS/Flex, ADAMS/Engine, etc.

- I used to work for six years in the ADAMS/Solver group

End: General Introduction

Begin: Review of Linear Algebra

Notation Conventions

- A bold upper case letter denotes matrices
  - Example: **A**, **B**, etc.

- A bold lower case letter denotes a vector
  - Example: **v**, **s**, etc.

- A letter in italics format denotes a scalar quantity
  - Example: a, b_1

Matrix Review

- What is a matrix?

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_1^T \\ \boldsymbol{\alpha}_2^T \\ \cdots \\ \boldsymbol{\alpha}_m^T \end{bmatrix}$$

- Matrix addition:

$$\mathbf{A} = [a_{ij}], \qquad 1 \le i \le m, \qquad 1 \le j \le n$$

$$\mathbf{B} = [b_{ij}], \qquad 1 \le i \le m, \qquad 1 \le j \le n$$

$$\mathbf{C} = \mathbf{A} + \mathbf{B} = [c_{ij}], \qquad c_{ij} = a_{ij} + b_{ij}$$

  - Addition is commutative

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$$

Matrix Multiplication

- Remember the dimension constraints on matrices that can be multiplied:
  - # of columns of first matrix is equal to # of rows of second matrix

$$\mathbf{A} = [a_{ij}], \qquad \mathbf{A} \in \mathbb{R}^{m*n}$$
$$\mathbf{C} = [c_{ij}], \qquad \mathbf{C} \in \mathbb{R}^{n*p}$$
$$\mathbf{D} = \mathbf{A} \cdot \mathbf{C} = [d_{ij}], \qquad \mathbf{D} \in \mathbb{R}^{m*p}$$
$$d_{ij} = \sum_{k=1}^{n} a_{ij} c_{kj}$$

- This operation is not commutative

- Distributivity of matrix multiplication with respect to matrix addition:

$$(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$$

Matrix-Vector Multiplication

- A column-wise perspective on matrix-vector multiplication

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} = \sum_{i=1}^{n} \mathbf{a}_i v_i$$

  - Example:

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} 1 & 4 & 2 & 0 \\ 2 & 3 & 1 & 1 \\ -1 & 0 & 1 & -1 \\ 0 & 1 & -1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \\ 0 \end{bmatrix} \cdot (1) + \begin{bmatrix} 4 \\ 3 \\ 0 \\ 1 \end{bmatrix} \cdot (2) + \begin{bmatrix} 2 \\ 1 \\ 1 \\ -1 \end{bmatrix} \cdot (-1) + \begin{bmatrix} 0 \\ 1 \\ -1 \\ -2 \end{bmatrix} \cdot (1) = \begin{bmatrix} 7 \\ 8 \\ -3 \\ 1 \end{bmatrix}$$

- A row-wise perspective on matrix-vector multiplication:

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} \boldsymbol{\alpha}_1^T \\ \boldsymbol{\alpha}_2^T \\ \dots \\ \boldsymbol{\alpha}_m^T \end{bmatrix} \mathbf{v} = \begin{bmatrix} \boldsymbol{\alpha}_1^T \mathbf{v} \\ \boldsymbol{\alpha}_2^T \mathbf{v} \\ \dots \\ \boldsymbol{\alpha}_m^T \mathbf{v} \end{bmatrix}$$

Matrix Review [Cntd.]

- Scaling of a matrix by a real number: scale each entry of the matrix

$$\alpha \cdot \mathbf{A} = \alpha \cdot [a_{ij}] = [\alpha \cdot a_{ij}]$$

  - Example:

$$(1.5) \cdot \begin{bmatrix} 1 & 4 & 2 & 0 \\ 2 & 3 & 1 & 1 \\ -1 & 0 & 1 & -1 \\ 0 & 1 & -1 & -2 \end{bmatrix} = \begin{bmatrix} 1.5 & 6 & 3 & 0 \\ 3 & 4.5 & 1.5 & 1.5 \\ -1.5 & 0 & 1.5 & -1.5 \\ 0 & 1.5 & -1.5 & -3 \end{bmatrix}$$

- Transpose of a matrix **A** dimension m$\times$n: a matrix **B**=**A**$^{\mathbf{T}}$ of dimension n$\times$m whose (i,j) entry is the (j,i) entry of original matrix **A**

$$\begin{bmatrix} 1 & 4 & 2 & 0 \\ 2 & 3 & 1 & 1 \\ -1 & 0 & 1 & -1 \\ 0 & 1 & -1 & -2 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 & -1 & 0 \\ 4 & 3 & 0 & 1 \\ 2 & 1 & 1 & -1 \\ 0 & 1 & -1 & -2 \end{bmatrix}$$

Linear Independence of Vectors

- Definition: linear independence of a set of m vectors, $\mathbf{v}_1,\ldots,\mathbf{v}_m$:

$$\mathbf{v}_1,\ldots.,\mathbf{v}_m \in \mathbb{R}^n$$

- The vectors are linearly independent if the following condition holds

$$\alpha_1\mathbf{v}_1 + \ldots. + \alpha_m\mathbf{v}_m = 0 \qquad \Rightarrow \qquad \alpha_1 = \ldots = \alpha_m = 0$$

- If a set of vectors are not linearly independent, they are called dependent
  - Example:

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \qquad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 4 \end{bmatrix} \qquad \mathbf{v}_3 = \begin{bmatrix} 1 \\ -3 \\ -6 \end{bmatrix}$$

  - Note that $\mathbf{v}_1$-2$\mathbf{v}_2$-$\mathbf{v}_3$=0

Matrix Rank

- Row rank of a matrix
  - Largest number of rows of the matrix that are linearly independent
  - A matrix is said to have full row rank if the rank of the matrix is equal to the number of rows of that matrix
- Column rank of a matrix
  - Largest number of columns of the matrix that are linearly independent
- NOTE: for each matrix, the row rank and column rank are the same
  - This number is simply called the rank of the matrix
  - It follows that

$$\mathrm{rank}(C) = \mathrm{rank}(C^T)$$

Matrix Rank, Example

- What is the row rank of the matrix **J**?

$$\mathbf{J} = \begin{bmatrix} 2 & 1 & -1 & 0 \\ 4 & -2 & -2 & 1 \\ 0 & -4 & 0 & 1 \end{bmatrix}$$

- What is the rank of **J**?

Matrix & Vector Norms

- Norm of a vector
  - Definitions: norm 1, norm 2 (or Euclidian), and Infinity norm

$$||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i| \qquad ||\mathbf{x}||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2} \qquad ||\mathbf{x}||_\infty = \max |x_i|$$

- Norm of a matrix
  - Definition: norm 1, norm 2 (or Euclidian), and Infinity

$$||\mathbf{A}|| = \sup_{\mathbf{x}\neq 0} \frac{||\mathbf{Ax}||_p}{||\mathbf{x}||_p}$$

$$||\mathbf{A}||_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}| \qquad ||\mathbf{A}||_2 = \sqrt{\rho(\mathbf{A}^T\mathbf{A})} \qquad ||\mathbf{A}||_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|$$

Matrix & Vector Norms, Exercise

- Find norm 1, Euclidian, and Infinity for the following matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix}$$

- Answer:

$$\|\mathbf{A}\|_1 = 6 \qquad \|\mathbf{A}\|_2 \approx 5.46 \qquad \|\mathbf{A}\|_\infty = 7$$