Colorado State University

ENGR 513 – Modeling Simulation and Experiments

Thomas Bradley
Professor, Systems Engineering
Spring 2020

Opening

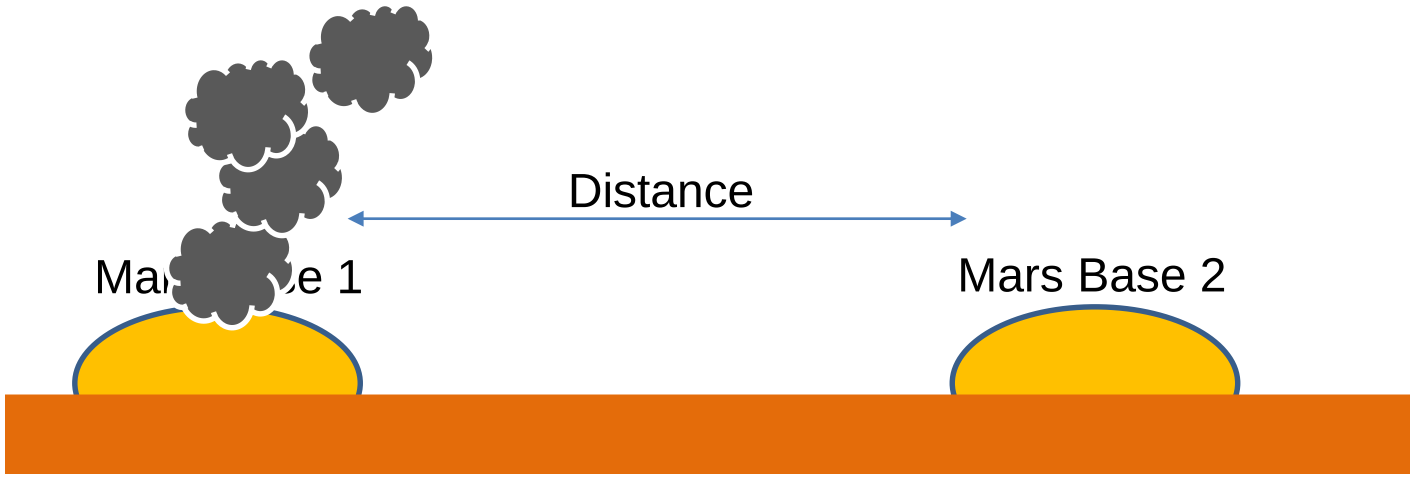
- Lectures
  - Introduction to Midterm
  - Laboratories - Midterm data gathering
  - MDA

Midterm Project

- Individual midterm project, you can work together, but turn in your own work.
- I'll provide the project scenario and guidance
- You demonstrate your capabilities in MSE.

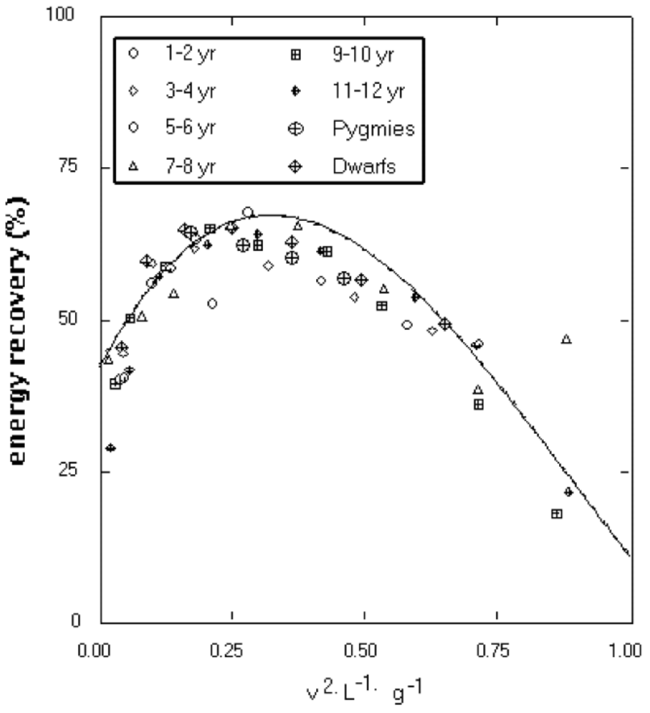
Midterm Project

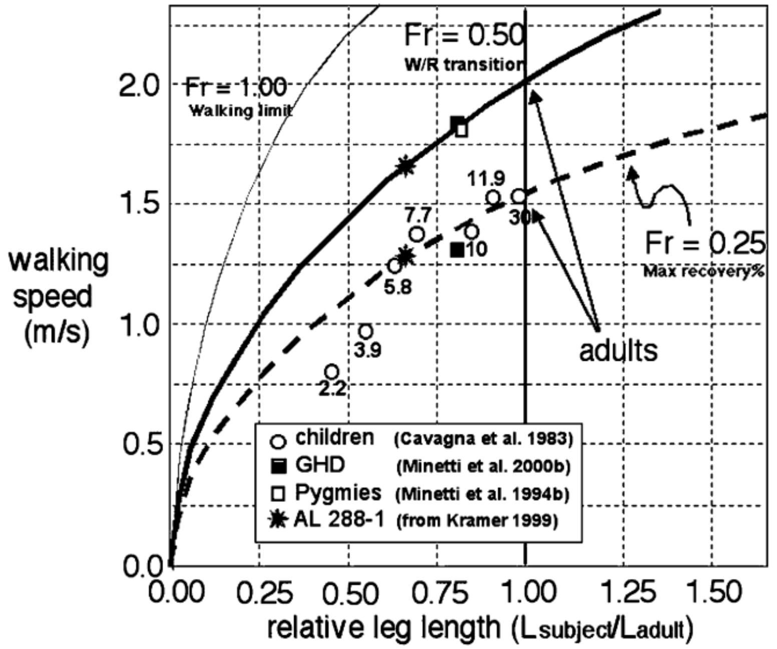
- We want to plan a colony on Mars, but we want each base to be walkable distance from the others in case of emergency
  - How fast do people walk on Mars anyhow?



Midterm Project

- Well, it turns out that exercise scientists have measured the energy consumption and non-dimensional characteristics of walking.

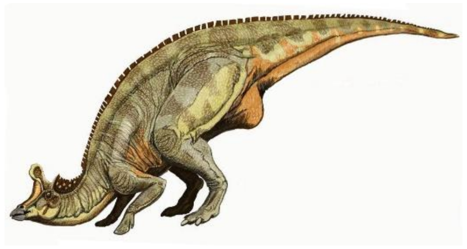




Midterm Project

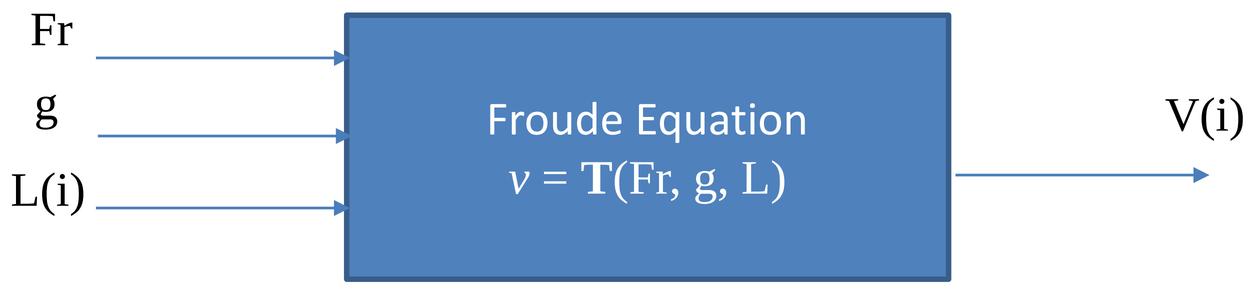
- They can characterize gait transition from walking to running using the Froude Number

$$\mathrm{Fr} = \frac{v^2}{g\,L}$$



  - v=velocity (m/s)
  - g=acceleration due to gravity (m/s/s)
  - L=leg length (ground to hip joint) (m)
- At Fr = 0.25, humans are at their most efficient
- At Fr = 0.5, humans transition from walk/run

Midterm Project



$$\sigma_x^2 \simeq \sigma_u^2 \left(\frac{\partial x}{\partial u}\right)^2 + \sigma_v^2 \left(\frac{\partial x}{\partial v}\right)^2 + \cdots$$

The normal Kline McClintock method

$$\sigma_v{}^2 = \sigma_T{}^2 \left(\frac{\partial v}{\partial T}\right)^2 + \sigma_L{}^2 \left(\frac{\partial v}{\partial L}\right)^2$$

The tool function is not perfect though

$$\sigma_v{}^2 = \sigma_T{}^2 + \sigma_L{}^2 \left(\frac{\partial T}{\partial L}\right)^2$$

The Sensitivity of the tool to its output is 1.0

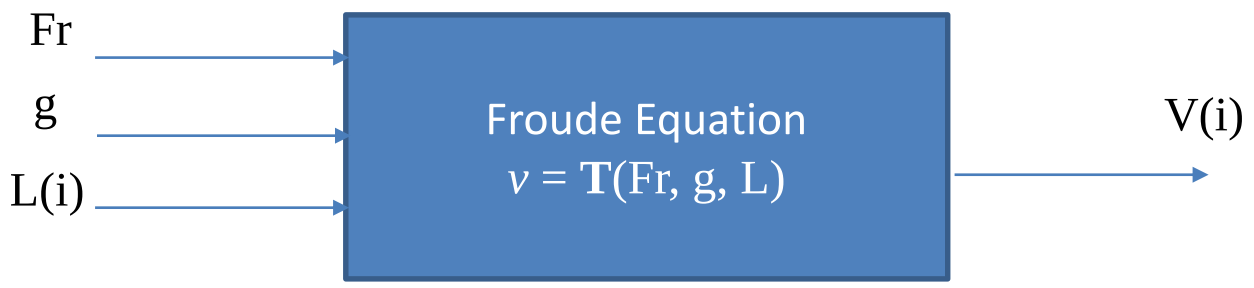
*L is an uncertain input
*All models are deterministic models

Part 1

- **Establish Purpose and Scope**
- Describe the background and scope of your project. Describe and document your simulation's purpose and scope.

Part 2

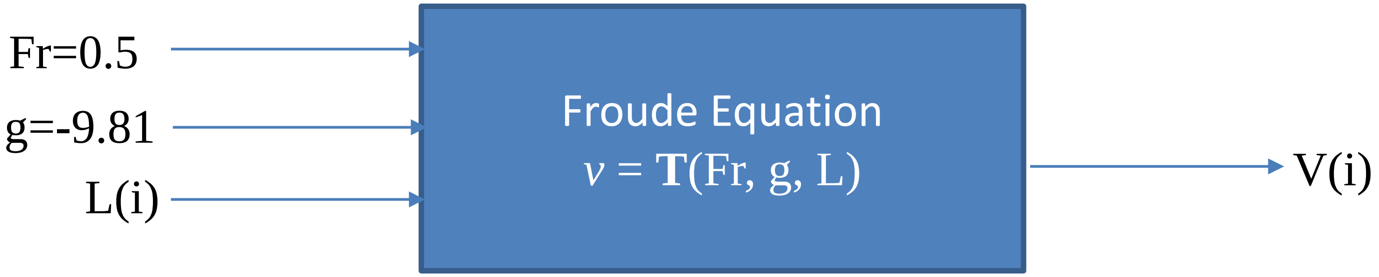
- Perform Validation Procedures Including Uncertainty Quantification

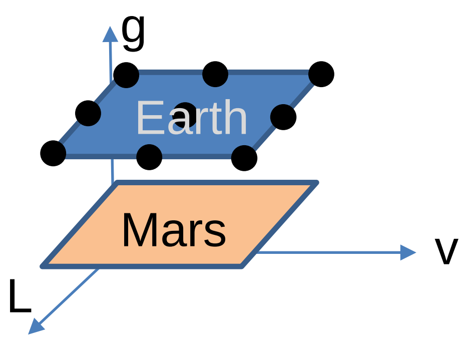


- How will we quantify the error in this model of walking velocity?

Part 2

- You will have to validate your model of walking velocity using data that you gather from subjects on earth.



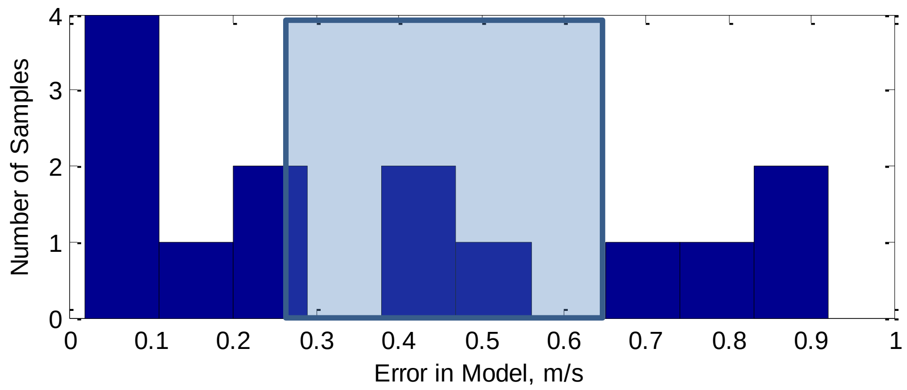


How much error is there in the walking model? Characterize it by calculating the uncertainty in *V* based on

1) validation against experimental mean
2) experimental uncertainty

Colorado State University

Midterm Project



- Collect walking speed data on your friends (not yourself)
- Calculate the average relative error associated with your tool from two sources:
  - Validation error
  - Experimental uncertainty

Midterm Review

```
clear all
close

% Leg Length (m)    Speed (m/s)
data = [0.9271  2.777777778
0.9652  3.144654088
0.9271  2.380952381
0.9398  2.288329519
1.1778  2.364066194
1.0668  2.450980392
1.0287  2.717391304
0.88138 2.610966057
0.90932 2.380952381
0.90932 2.427184466
0.9779  2.475247525
1.016   2.444987775];

leg_length = data(:,1);
speed = data(:,2);

speed_model = sqrt(0.5*9.81*leg_length);
mean_relative_speed_err = mean((speed_model - speed)./speed);
```

- Let’s Calculate the Mean Error
- Let’s calculate the CI about the estimation of the error mean

$$\left(\bar{x} - t\left(\frac{\alpha}{2}, n-1\right) * S/\sqrt{n}\right) < \mu < \left(\bar{x} + t\left(\frac{\alpha}{2}, n-1\right) * S/\sqrt{n}\right)$$

```
relative_speed_CI = -tinv((1-0.68)/2, length(data)-1) * std(data(:,2))/sqrt(length(data));
variance = relative_speed_CI.^2;
```

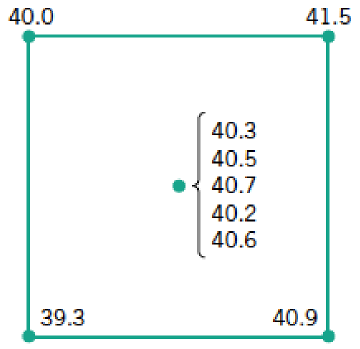
Midterm Review

- Now let’s consider that pure error estimate, pretend that I went out and gathered that data:

```
leg_length_pee = [1.000 1.000 1.000 1.000 1.000]';
speed_pee = [2.33 2.35 2.45 2.1 2.24]';
mean_relative_pee_err = mean((mean(speed_pee)- speed_pee)./speed_pee);
variance_pee = (std(speed_pee)./mean(speed_pee)).^2;
total_tool_variance = variance + variance_pee;
```

Estimate prediction uncertainty

Calculate the aleatory experimental uncertainty using a *pure error estimate*, by repeating the experiment on yourself multiple times



$$s_{exp} = \mathrm{stdev}([e_{THB1}, e_{THB2}, e_{THB3}, e_{THB4}, e_{THB5}])$$

$$90\%\ \mathrm{CI} = \pm 1.65\mathrm{sigma}$$

Sum up the variances to get the tool uncertainty:

$$s_{total}^2 = s_{exp}^2 + s_{model}^2$$

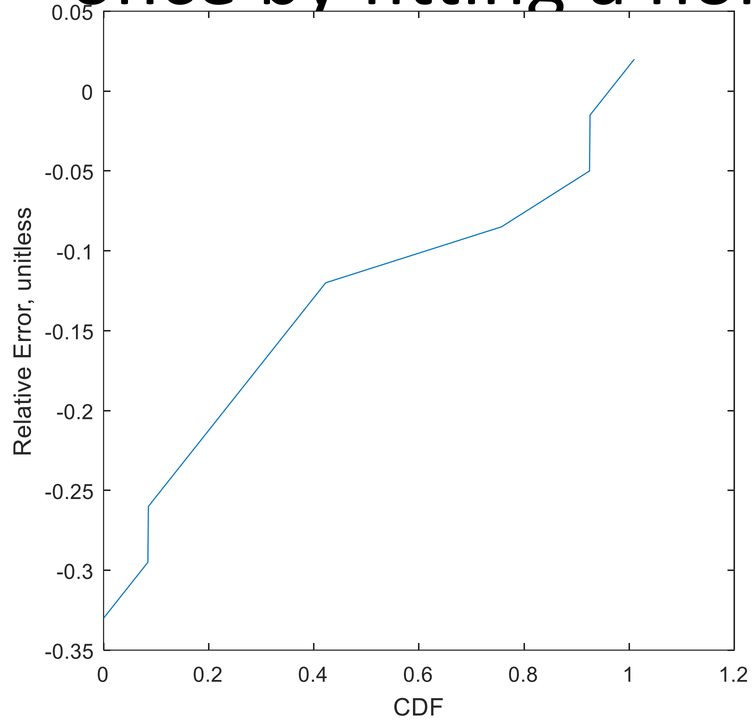
Kline-Mclintock Error Propagation

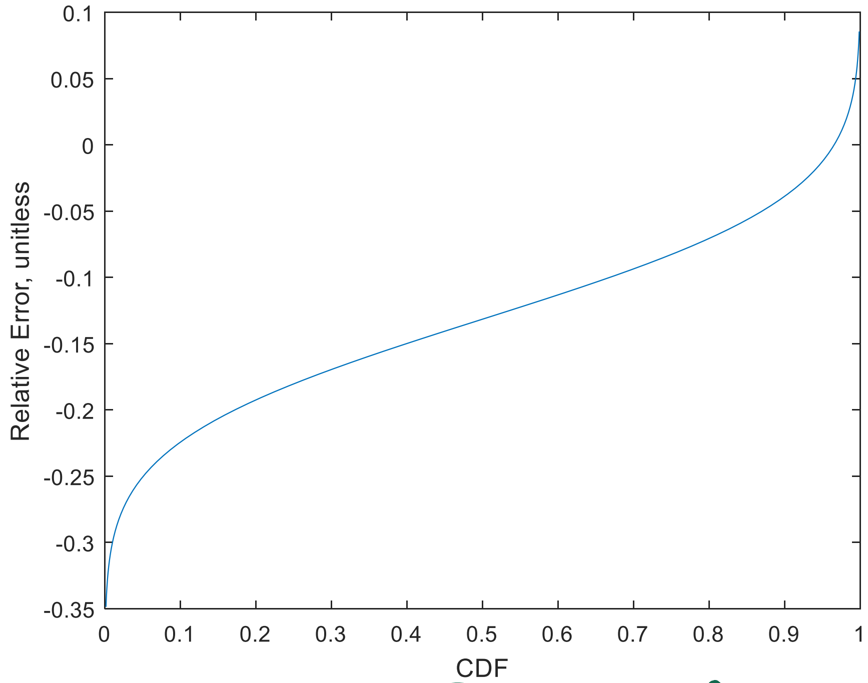
$$\sigma_v{}^2 = \sigma_T{}^2 + \sigma_L{}^2 \left(\frac{\partial T}{\partial L}\right)^2$$

- We just calculated the Tool Error Variance ($\sigma_T{}^2$)
- Its easy to calculate the leg length of your astronaut variance ($\sigma_L{}^2$= σ_L* σ_L)
- How do we get $\frac{\partial T}{\partial L}$?
- $\frac{\partial T}{\partial L} = \frac{\partial}{\partial L}\sqrt{Fr * g * L} = \frac{\partial}{\partial L} T(Fr, g, L)$

Monte Carlo Simulation

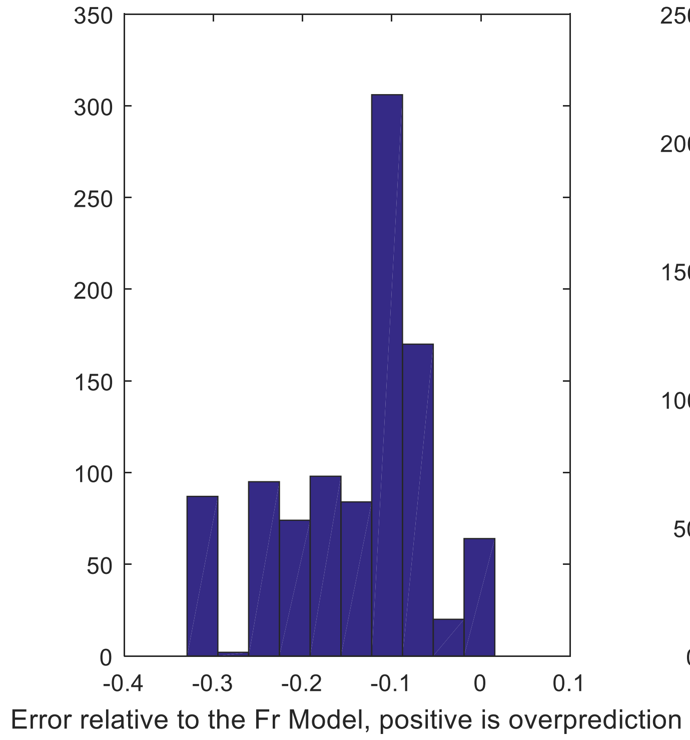
- I did this analysis in two ways, one without any fitting of the distribution of relative error, once by fitting a normal distribution

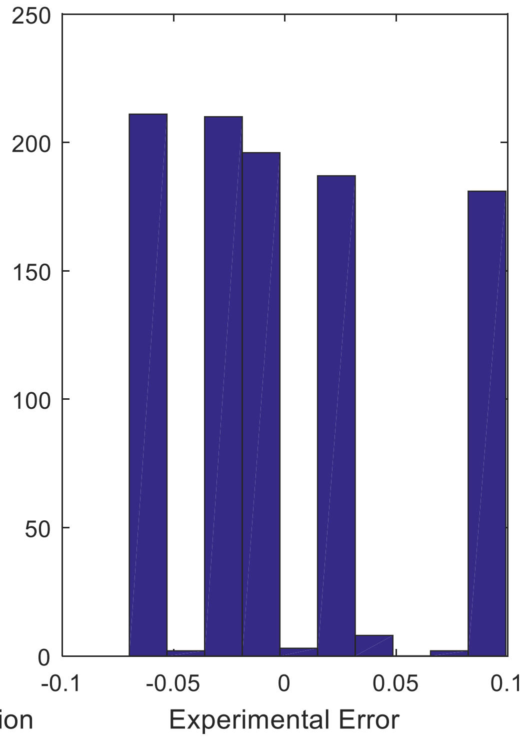


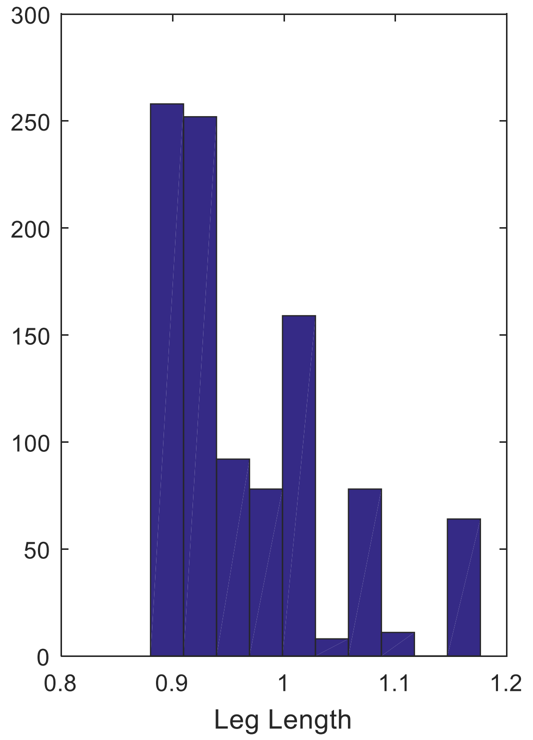


Monte Carlo Simulation

- Using the measured distributions

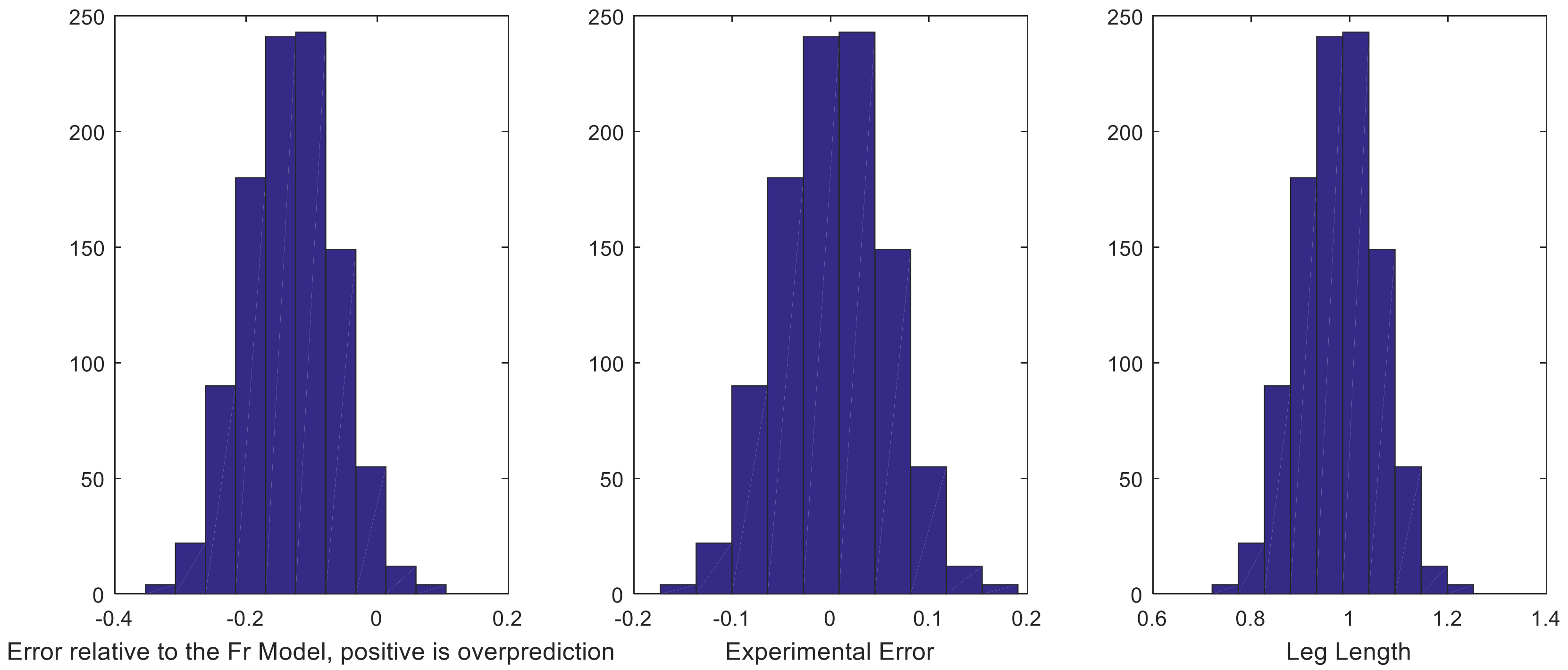






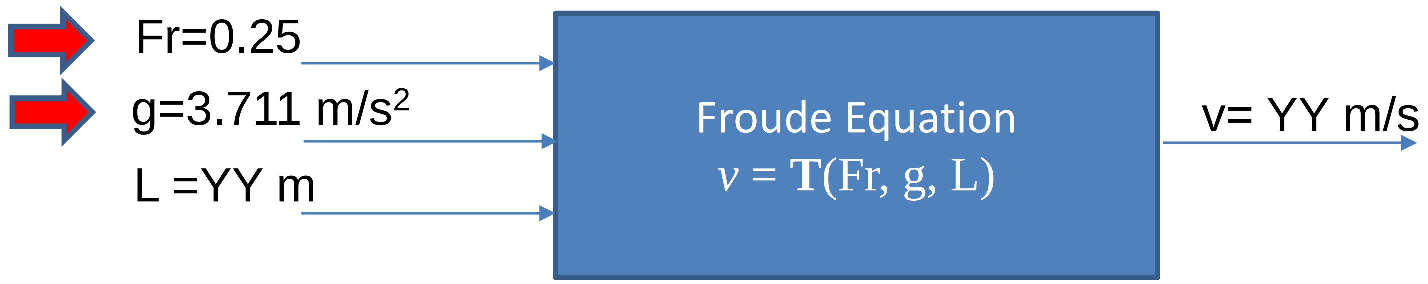
Monte Carlo Simulation

- Using fitted distributions



Part 3

- **Perform Simulation at Expected Values on Mars**



- **Propagate Uncertainty from both:**
  - **Experimentally-measured leg length (input)**
  - **Tool Uncertainty (due to prediction error)**
- **Present expected value and 90% Certainty bounds on the result ($V_{walkingonMars}$)**

Part 4

- **Conclusions**
  - On the basis of your results, provide a graphical and textual summary of the findings, results and recommendations. Draw some conclusions about the role of the tools that we have developed in providing decision support for this example problem. Provide me with some critique or suggestions for the final project.

Part 4

- **Conclusions should also include**
  - Provide me with some critique or suggestions for the project. What parts of the exercise surprised you (positive or negative). I will evaluate this section based on the type of learning that that you demonstrate on the basis of Bloom’s Taxonomy (link).

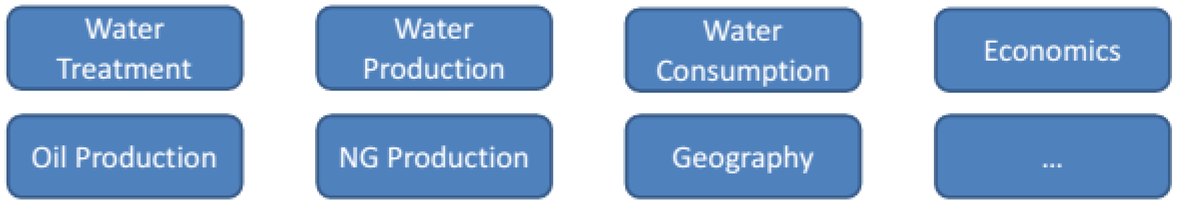
http://mars.nasa.gov/multimedia/resources/mars-posters-explorers-wanted/

https://www.sciencedirect.com/science/article/pii/S0021929012000723?via%3Dihub

Colorado State University

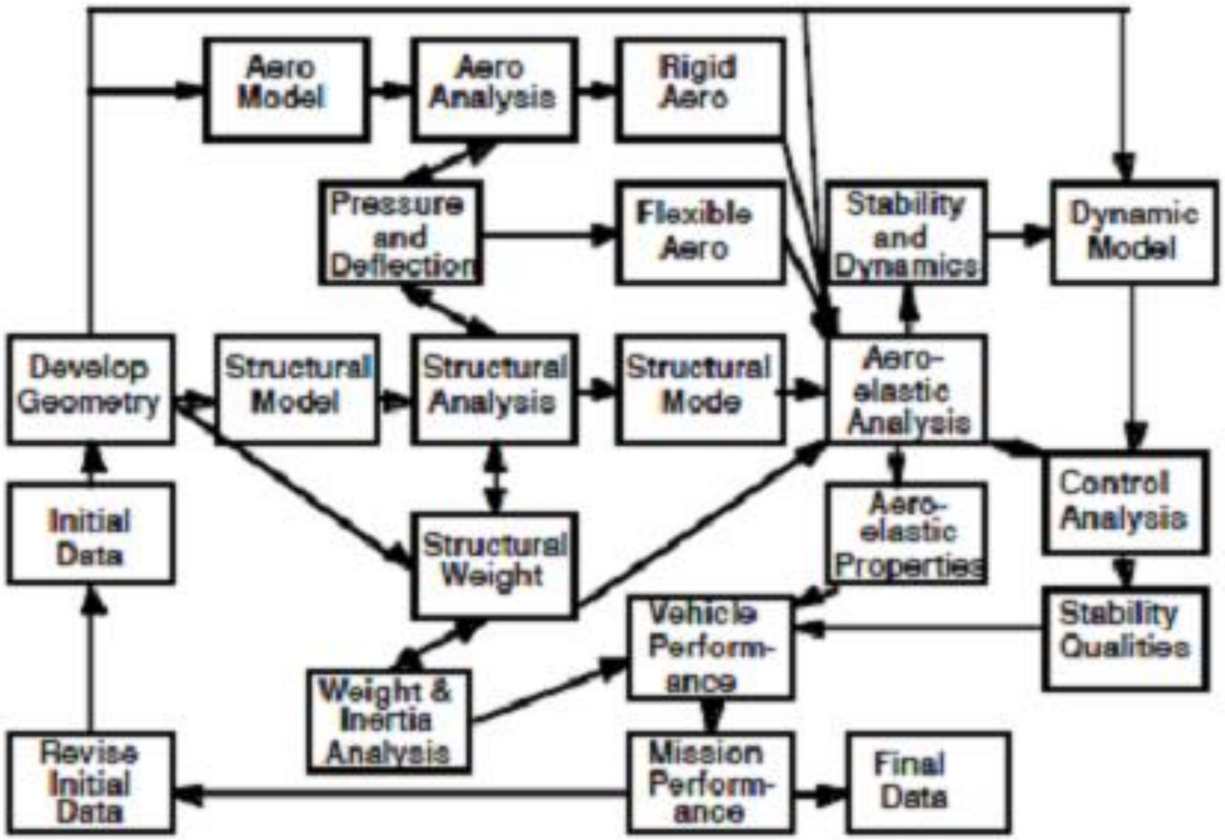
Multi-disciplinary Analysis

- In any complicated domain, modeling the tradeoffs among options requires multi-disciplinary analysis
- For instance, to calculate the GHG/emissions of Natural Gas, you need models of:



- They are all inter-related….

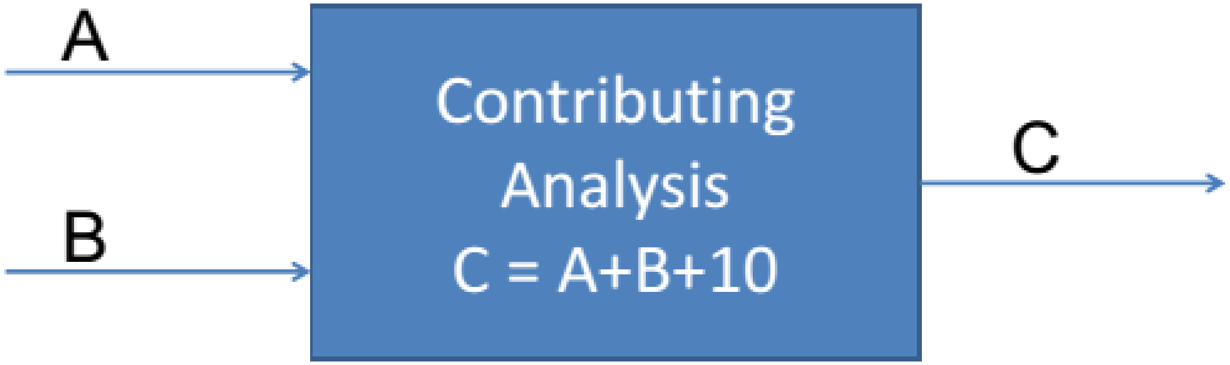
Aircraft Design Example



"DeMAID/GA user's guide-design manager's aid for intelligent decomposition with a genetic algorithm," NASA, Hampton, VA, TM-110 241, 1996.

Multi-disciplinary Analysis

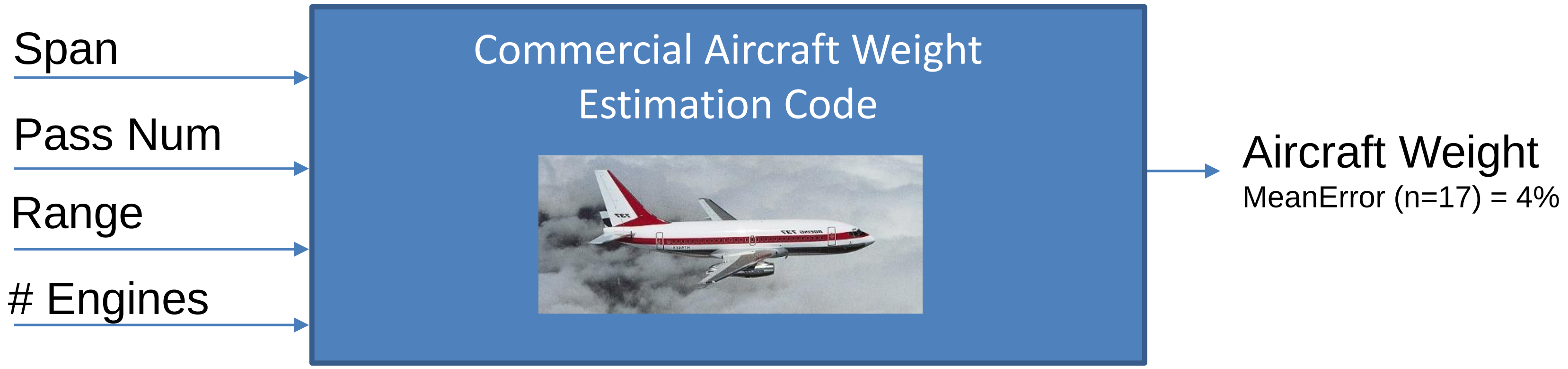
- Functions (CAs) have *causality* and are *atomic subsystems*,



In MATLAB code:

```
function [C] = ContributingAnalysis(A,B)
          C=A+B+10;
end
```

Multi-disciplinary Analysis



- This model has already been
  - constructed, verified, validated, qualified, uncertainty quantified, etc.

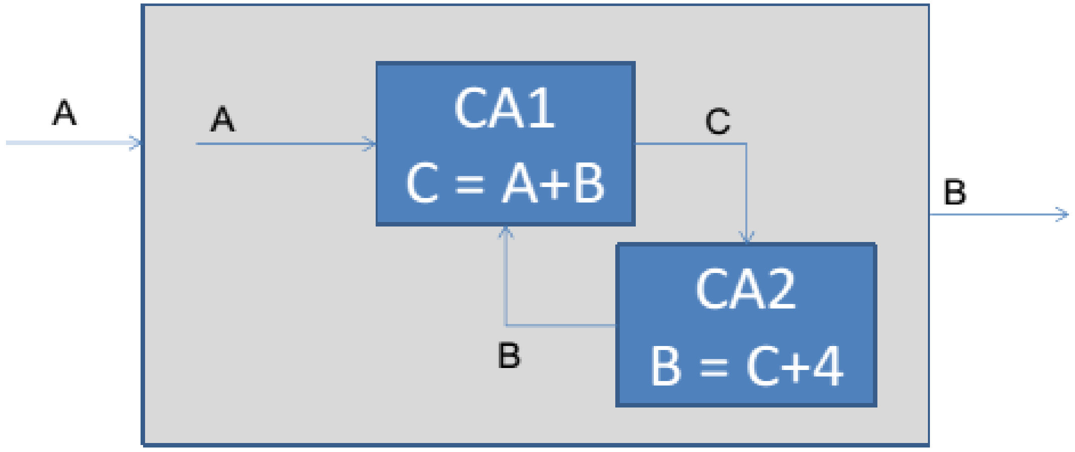
- We want to use it in our analyses

Multi-disciplinary Analysis

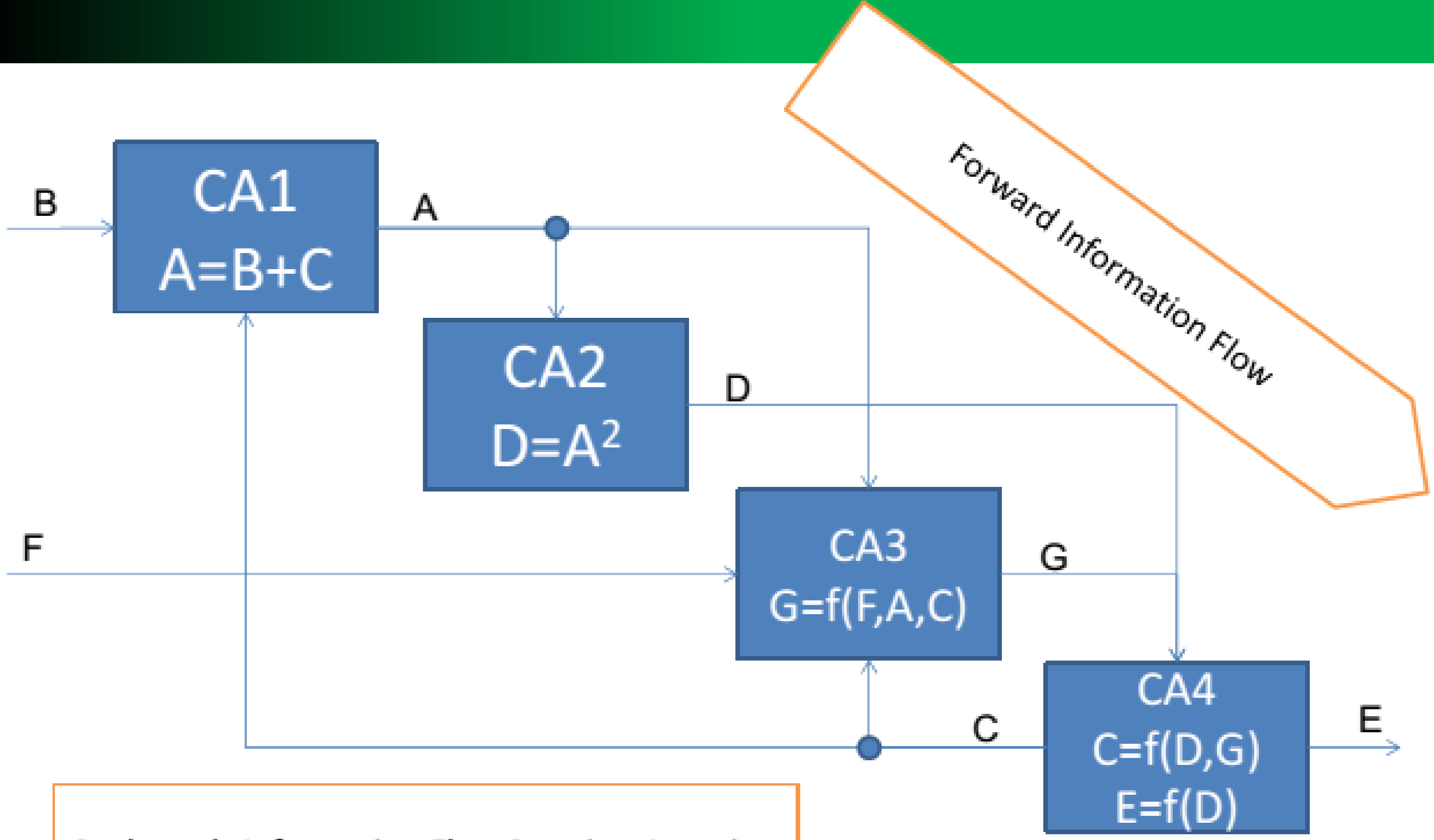
- Combining CAs into MDAs
- Example:

CA1: C=A+B

CA2: B=C+4



Solving Multi-disciplinary Analyses

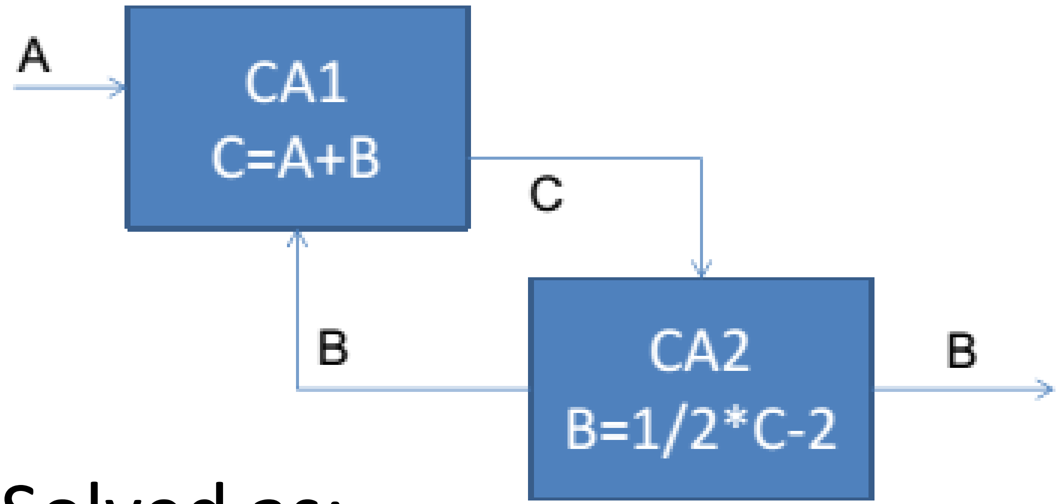


Backwards Information Flow Requires Iteration

Colorado State University

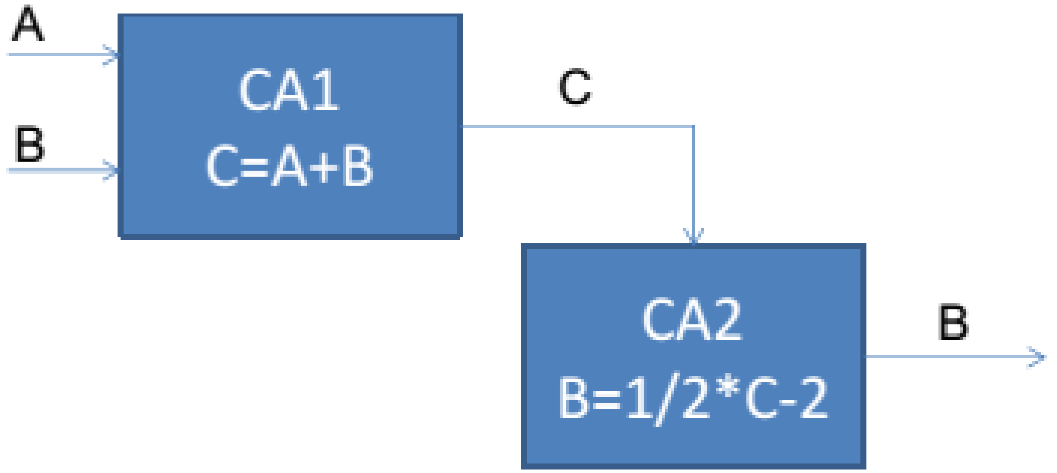
DSM + Iterative Solver = MDA

- Composed as:



CA1			
Input	Input	Output	
		CA2	
		Input	Output
A	B	C	B
2.500	1.000	3.500	-0.250
2.500	-0.250	2.250	-0.875
2.500	-0.875	1.625	-1.188
2.500	-1.188	1.313	-1.344
2.500	-1.344	1.156	-1.422
2.500	-1.422	1.078	-1.461
2.500	-1.461	1.039	-1.480
2.500	-1.480	1.020	-1.490
2.500	-1.490	1.010	-1.495
2.500	-1.495	1.005	-1.498
2.500	-1.498	1.002	-1.499
2.500	-1.499	1.001	-1.499

- Solved as:



Solution

$$\text{A = 2.5} \rightarrow \boxed{\text{MDA}} \rightarrow \text{B = -1.5}$$

Benefits of the MDA

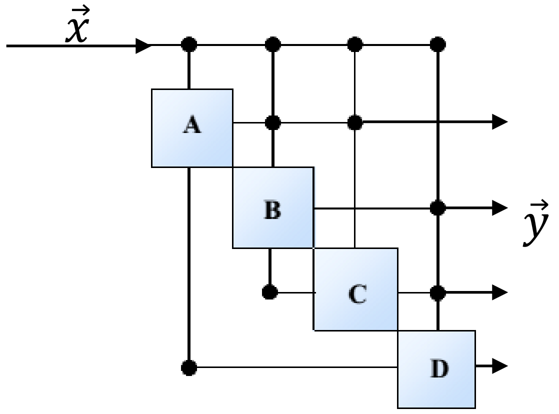
- Represents complicated, legacy, codes or processes
- Scalable, structured way to compose and analyze large simulations
  - Can be parallelized, can be “structured”
- Can represent design processes (MDO)
- Appeals to linear algebra analogies

Solving MDAs

- Let's describe a medium-sized MDA (Ref, Ref)
- $\vec{x}$ is a vector of inputs

  >> $\vec{x}$ = [x y z]'
- $\vec{y}$ is a vector of outputs

  >> $\vec{y}$ = [a b c d]'
- We have 4 Contributing Analyses (A,B,C,D)
- What is $\vec{y}$, at $\vec{x}$ = [5 5 5]' ?



$$a = \frac{1}{40}\left(\begin{array}{l} 2.48374x + \dfrac{4.746265d^2}{x+3.2874} + 3.23872dx + [y - 3.48572]x \\ + \dfrac{[d-4.34721]y}{[x+5.424]}2.3421 + y \end{array}\right)$$

$$b = \frac{1}{40}\left(xac + \frac{a}{c^2 + 0.02375}\right)$$

$$c = 2 + \frac{1}{60}\left(x^2[x - 5.67834] + ax[a - 6.3432]\right)$$

$$d = \frac{1}{1200}\left(bc[c + 0.0345] + z^2[b^2 + 0.34721]\right)$$

Solving MDAs

- We can just solve this by guessing the values of $\vec{y}$ = [a,b,c,d] that are needed to calculate each CA
  - Let $\vec{y}_0$ = [a,b,c,d] = [1,1,1,1]
- Calculate the difference between the values of $\vec{y}$ that went into the calculation and the values that come out, force that difference to zero
  - Calculate $\overrightarrow{zero_k} = \vec{y}_k - \vec{y}_{k\text{-}1}$
- When $\vec{y}_k$ is converged, all equations in the MDA are true

Solving MDAs

```
function [a,b,c,d] = tool1(y_vect,x_vect);

% Establish all variables
a = y_vect(1);
b = y_vect(2);
c = y_vect(3);
d = y_vect(4);

x = x_vect(1);
z = x_vect(3);
y = x_vect(2);

% Perform Function
a =
(1/40)*(2.48374*x+((4.746265*(d^2))/(x+3.2874))
+3.23872*d*x+x*(y-3.48572)+((y*(d-
4.3472))/(x+5.424))*2.3421+y);
```

```
function [a,b,c,d] = tool2(y_vect,x_vect);

% Establish all variables
a = y_vect(1);
b = y_vect(2);
c = y_vect(3);
d = y_vect(4);

x = x_vect(1);
z = x_vect(3);
y = x_vect(2);

% Perform Function
b = (1/40)*(x*a*c+(a/((c^2)+0.02375)));
```

Solving MDAs

```
function [a,b,c,d] = tool3(y_vect,x_vect);

% Establish all variables
a = y_vect(1);
b = y_vect(2);
c = y_vect(3);
d = y_vect(4);

x = x_vect(1);
z = x_vect(3);
y = x_vect(2);

% Perform Function
c = 2+(1/60)*((x^2)*(x-5.67834)+a*x*(a-
6.3432));
```

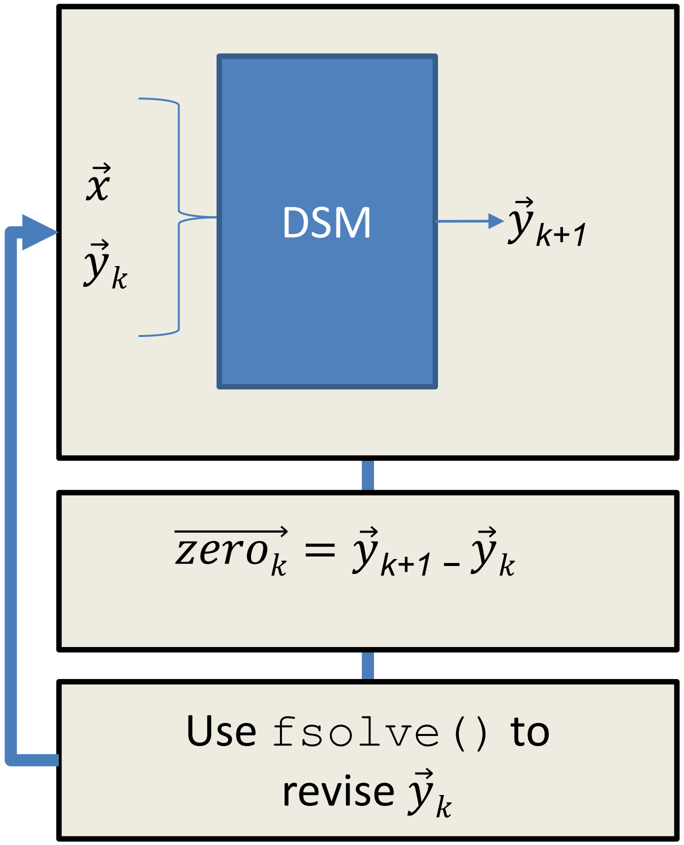
```
function [a,b,c,d] = tool4(y_vect,x_vect);

% Establish all variables
a = y_vect(1);
b = y_vect(2);
c = y_vect(3);
d = y_vect(4);

x = x_vect(1);
z = x_vect(3);
y = x_vect(2);

% Perform Function
d =
(1/1200)*(b*c*(c+0.0345)+(z^2)*((b^2)+0.34721));
```

Design Structure Matrix



$\vec{x}$
$\vec{y}_k$
DSM
$\vec{y}_{k+1}$
$\overrightarrow{zero}_k = \vec{y}_{k+1} - \vec{y}_k$
Use `fsolve()` to revise $\vec{y}_k$

```
function zero = List_of_CAs(y_vect,x_vect)

a = y_vect(1);
b = y_vect(2);
c = y_vect(3);
d = y_vect(4);

x = x_vect(1);
y = x_vect(2);
z = x_vect(3);

[a,b,c,d] = tool1(y_vect,x_vect);
zero(1)  =  a - y_vect(1);
[a,b,c,d] = tool2(y_vect,x_vect);
zero(2)  =  b - y_vect(2);
[a,b,c,d] = tool3(y_vect,x_vect);
zero(3)  =  c - y_vect(3);
[a,b,c,d] = tool4(y_vect,x_vect);
zero(4)  =  d - y_vect(4);
```

DSM + Iterative Solver = MDA

```
% Set the design variables that are the input to the MDA (equivalent to x vector)
des_vars = [5 5 5]

% Initialize the internal CA variables that are the outputs of the CAs
CA_vars_guess = [1 1 1 1];

% Call MATLAB’s function solver to set the change in CA vars to zero
CA_vars_converged = fsolve('List_of_CAs',CA_vars_guess, optimset('Display','iter'),des_vars)
```

```
des_vars = [5 5 5]
        CA_vars_converged =
         0.5060   0.0988   1.4712   0.0076
des_vars = [6 6 6]
        CA_vars_converged =
        0.7724   0.2104   1.7627   0.0123
```