Multidisciplinary Design Optimization (MDO): Complex Systems and Introduction

AE 6310: Optimization for the Design of Engineered Systems

Spring 2017

Dr. Glenn Lightsey

Lecture Notes Developed By Dr. Brian German

MDO Problems

Problems we're used to:

$$\begin{aligned} \min_{\mathbf{X}} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{X}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{X}) = \mathbf{0} \end{aligned}$$

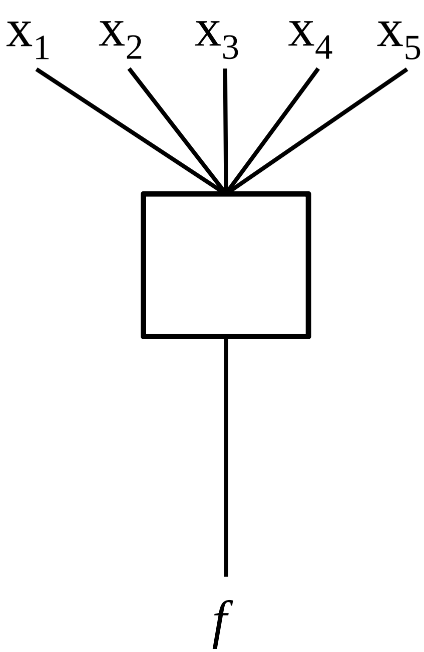
MDO problems:

$$\begin{aligned} \min_{\mathbf{X}} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{X}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{X}) = \mathbf{0} \end{aligned}$$

Seems similar right?

What's New?

Problems we're used to:

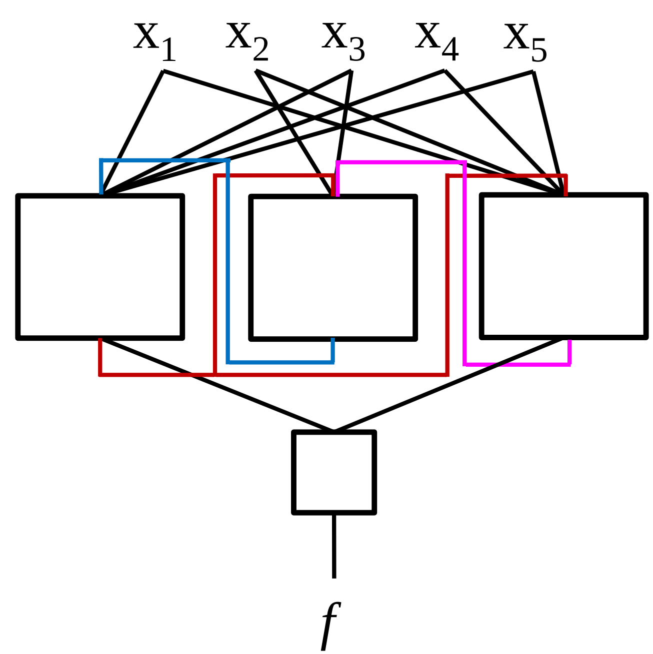


Design variables

Analyses (typically computer codes)

Objective Function (and/or constraints)

MDO problems:



What's New?

Problems we're used to:

- Design variables feed into one analysis.
- Objective function is an output of the analysis.
- Data flow and process are unidirectional.

MDO Problems:

- Design variables feed into multiple analyses.
- Objective function is a function one or several of the analysis function outputs.
- Data flow and process involves iteration due to interdependencies among analysis functions.

MDO is Used for Complex Engineered Systems

Complex system:

- “An assembly of interacting members that is difficult to understand as a whole.” (Allison 2004)
- Characterized by “emergent behavior”

There are *many* definitions and contexts for complex systems beyond the domain of MDO. Some even consider MDO problems to be just “complicated” but not “complex”…

Complex Systems - Implications

The behavior of the system in its entirety and its internal workings is difficult to comprehend.

Possible causes:

- Large scale (large number of design variables)
- Nonlinearity of individual analysis functions
- Interactions

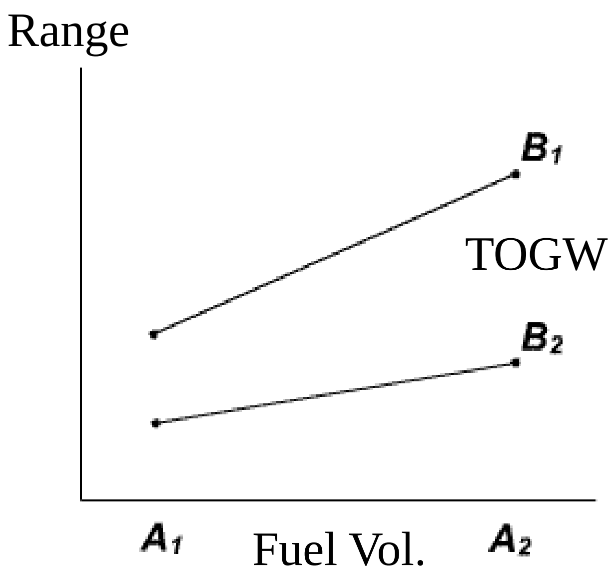
Example: it is unlikely for an engine designer to know how changes in his engine design will affect the aircraft's structural design in detail.

Complex Systems - Interactions

Interaction:

The system's response to changes in one analysis function is affected by the state of another analysis function.

Example: the same increase in fuel volume will have a different effect on max range if TOGW = 1,000 lb vs. if TOGW = 10,000 lb.



Interaction can occur between design variables in any nonlinear function. However, the "separation" of analysis functions makes interaction more difficult to detect and deal with.

Complex Systems – Decomposition

It is inefficient to analyze or optimize a complex system as a whole.

MDO therefore takes the following approach:

> **Partition (decompose)** the system into subsystems and analyze the subsystems and their interactions individually.

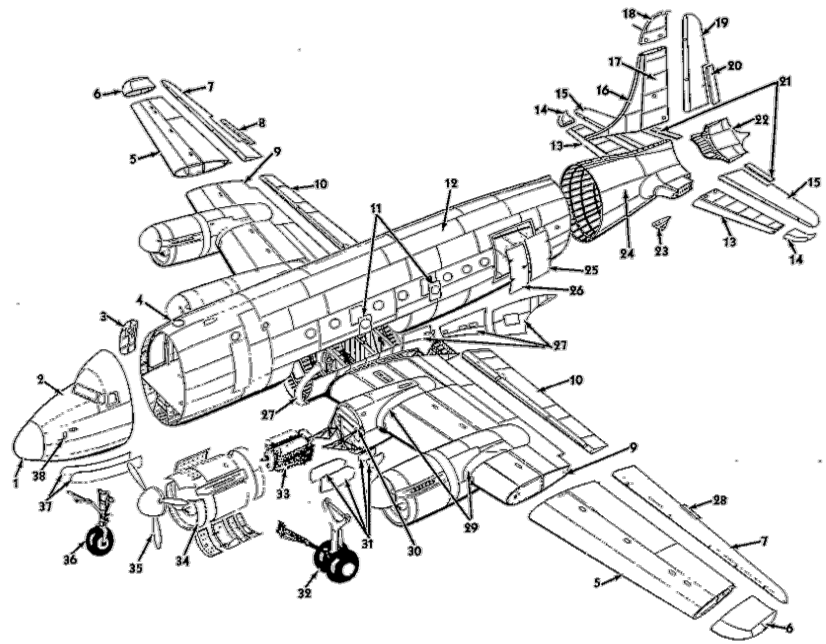
Sometimes this partitioning is “natural” based on historical design approaches or existing analysis codes.

In other cases, we design the partitioning in counter-intuitive ways to make the problem easier.

Complex Systems – Decomposition

Partitioning schemes may be based on:

- ❖ Physical/geometric boundaries
- ❖ Function of components
- ❖ *Discipline* of analysis
- ❖ Sequence of analyses
- ❖ Organizations within a company's management structure



(http://www.projectnorthstar.ca/html/northstarprimer/ch03s03.html)

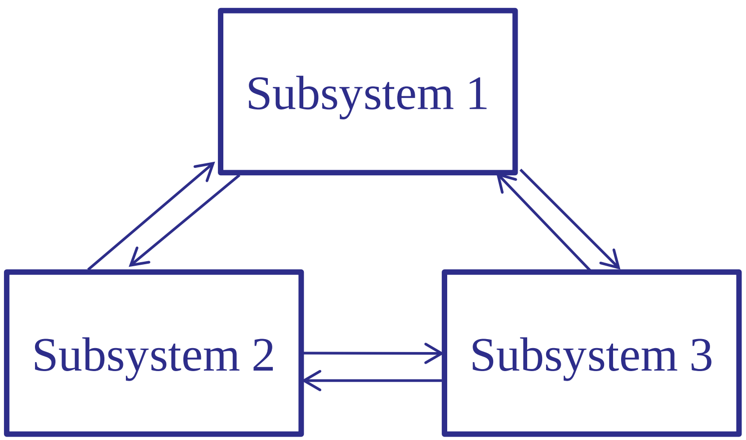
Partitioned elements may be called:

- ➢ Subsystems, disciplines, members, elements, components, subspaces, subproblems

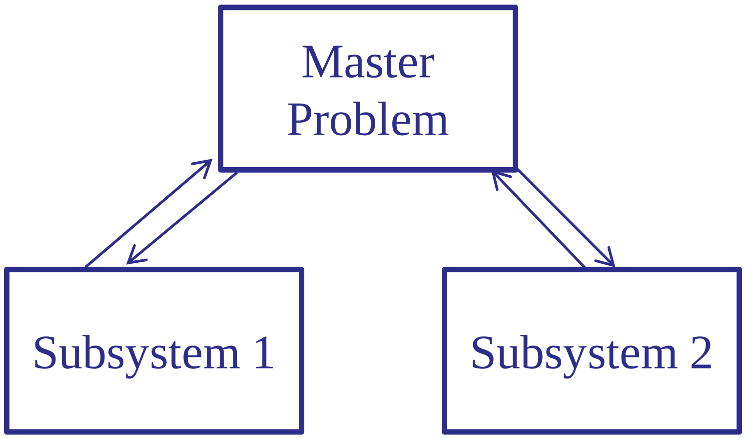
Complex Systems – Hierarchy

Partitioning/decomposition – dividing a large box into smaller boxes.

Hierarchy– how these boxes communicate.



Non-hierarchic



Hierarchical (Parent-Child)

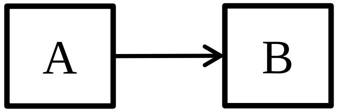
Problems not hierarchical in nature can be formulated as such by using additional system constraints… More about this later.

Complex Systems – Coupling

Interaction exists when one subsystem's analysis requires the output of another subsystem as an input.

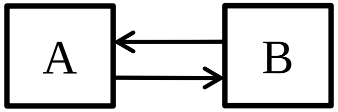
If B is dependent on A but A is not dependent on B, then the analysis is unidirectional.

Example: A is station 1 in an assembly line and B is station 2.



If A is dependent on B *and* B is dependent on A, then the two subsystems are said to be "**coupled**."

Can extend to multiple subsystems.



Complex Systems – Coupling

Example – Aeroelastic analysis of a wing involving two teams/analyses:

- Structural team/analysis:

  Wing Deflection = f (Load Distribution)

- Aerodynamics team/analysis:

  Load Distribution = f (Wing Deflection)

"Strong" coupling – changes in a subsystem have relatively large effects on the responses of coupled subsystems.

Coupling strength can be viewed as the interaction strength between subsystems.

Example: if the wing's overall rigidity is high, the coupling will be weaker than if the wing is more deformable.

Multidisciplinary Design Optimization (MDO)

As we have discussed, characteristics of complex systems make system analysis and design difficult.

Multidisciplinary design optimization (MDO) is the name of the technical field that has developed a class of formal approaches to the design of complex engineered systems.

The AIAA MDO Technical Committee defines MDO as:

> "A methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena." (MDO TC Technical Report 1991)

Multidisciplinary Design Optimization (MDO)

Unlike previous optimization algorithms we have covered, MDO methods are better described as "*architectures*."

- Optimization algorithms are tools used within the architectures.
- MDO is concerned with subsystem partitioning, communication, and coordination.

MDO architectures are often mapped to organizational structures with subsystems corresponding to individual experts or design teams.

- The study of MDO is also concerned with improving the *design process* as well as the resulting design.

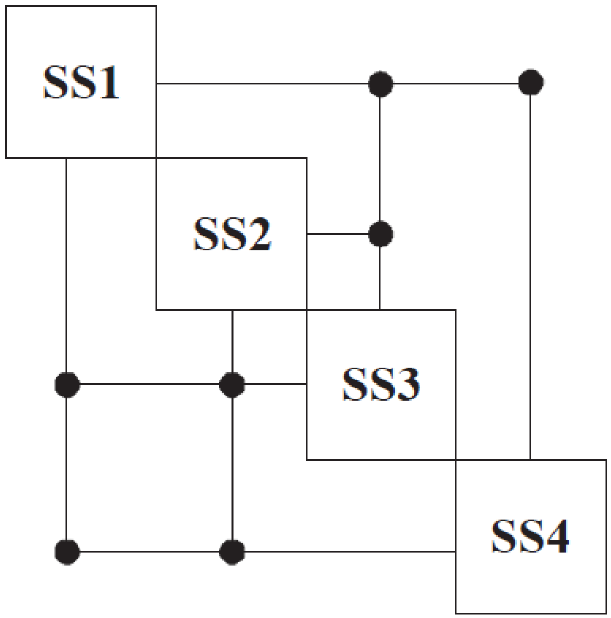
Visualizing MDO Problems – DSM

A common method of visually representing MDO problems is with a **Design Structure Matrix (DSM)**. A DSM is also known as a dependency structure, N^2 diagram, interaction matrix, etc.

Nodes in the upper triangular region denote feed-forward relationships

Nodes in the lower triangular region denote feedbacks.

The presence of feedback requires iteration or other techniques to find a consistent solution.

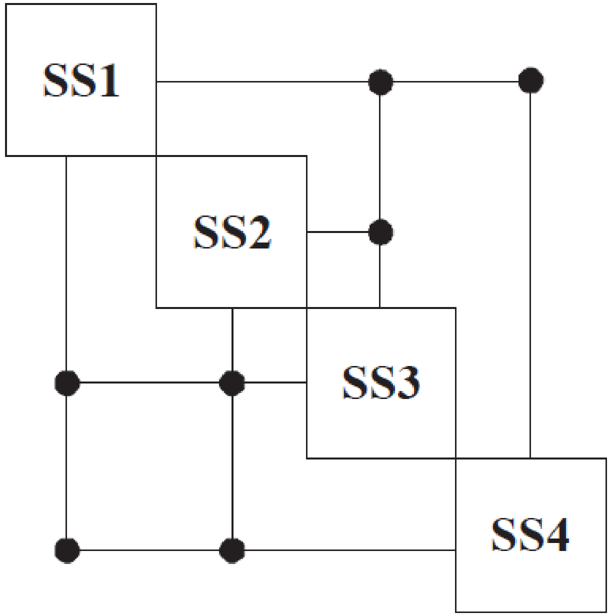


(Allison 2004)

Visualizing MDO Problems – DSM

Coupled subsystems are connected by nodes in both the upper and lower triangles (e.g. SS2 and SS3).

The DSM implies an “order” in which the subsystem analyses are executed: Start in the upper left and move to the lower right



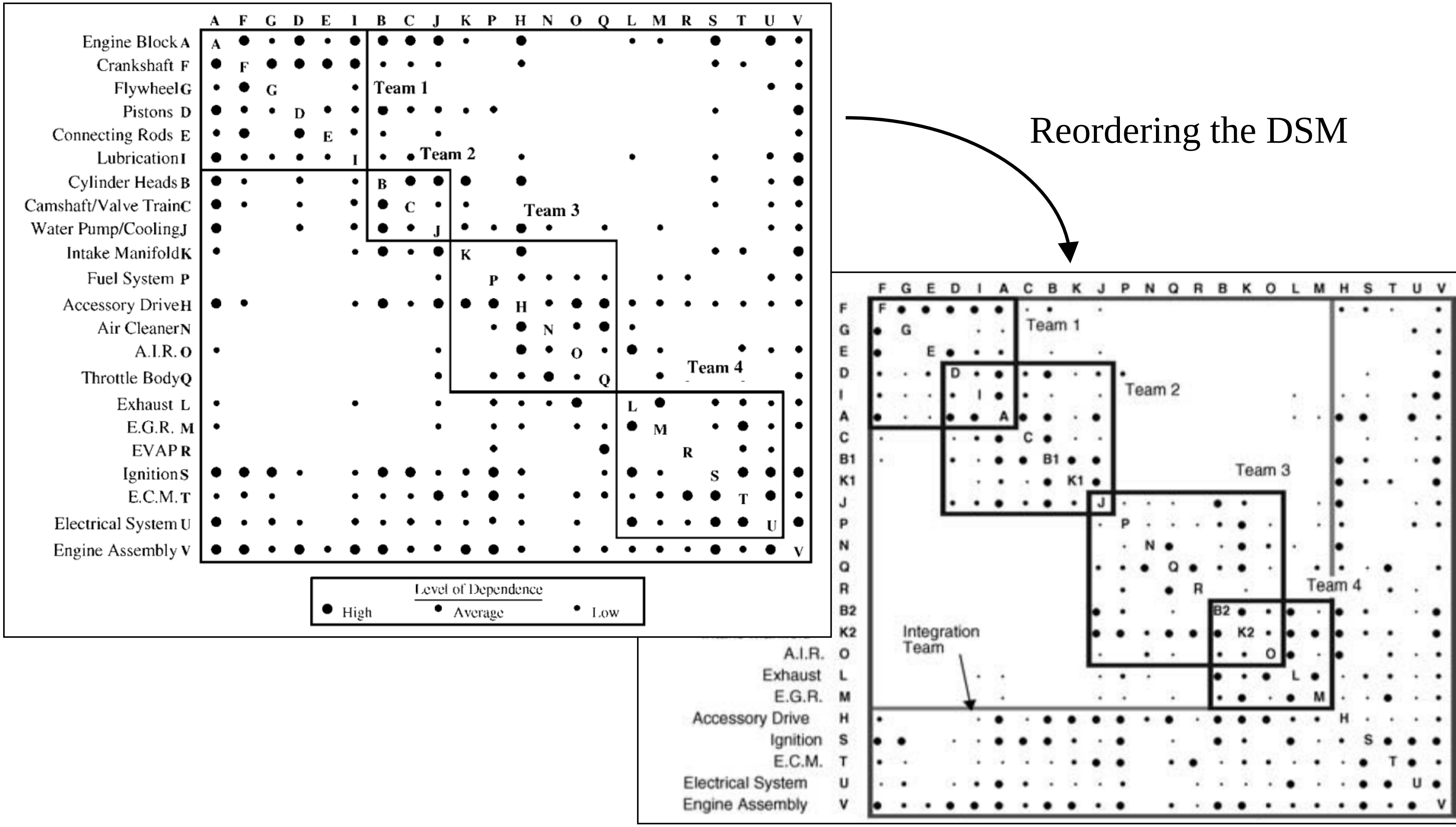
(Allison 2004)

Subsystem order can be rearranged to reduce the number and length of feedback connections.

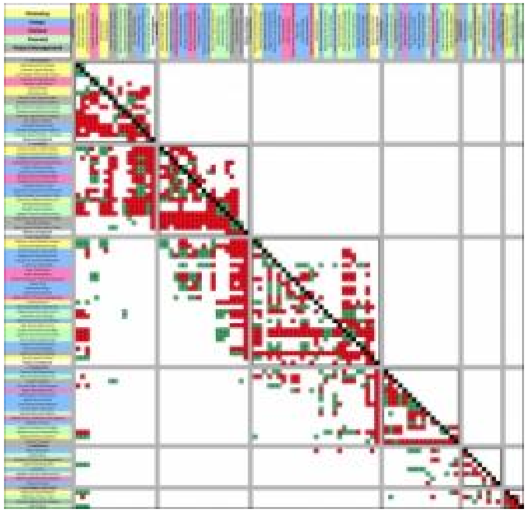
A DSM is abstract and hides complexities associated with:

- Number of variables represented by each node
- Strength of coupling relationships
- Effects on convergence behavior

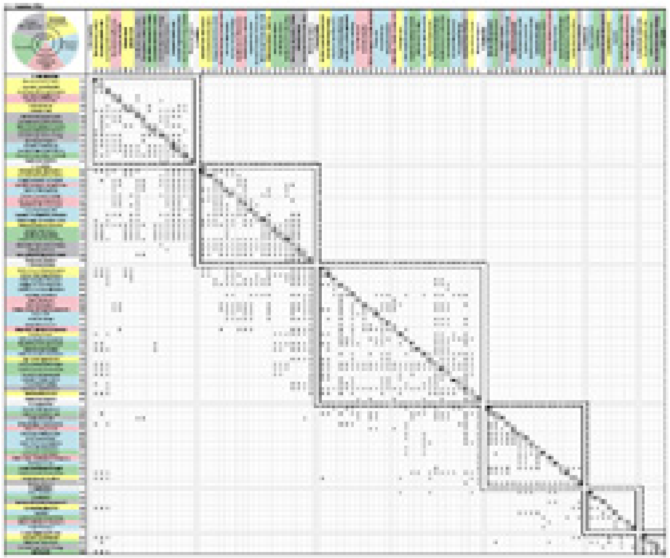
Visualizing MDO Problems – DSM Examples



Visualizing MDO Problems – DSM Examples



http://cdn.physorg.com/newman/gfx/news/2012/betterproduc.jpg



http://web.mit.edu/cre/alumni/i/bulloch-and-sullivan_2009_dsm_w250.jpg

Analysis vs. Design/Optimization

Analysis – determining the response of a system described by a set of design variables.

Design/Optimization – determining the set of design variable values that produce the “best” system response.

MDO stems from **Multidisciplinary Analysis (MDA)**.

The study of both in a multidisciplinary context is sometimes called **Multidisciplinary Design Analysis and Optimization (MDAO)**.

MDA Solvers

Solver– an algorithm used to solve a system of nonlinear equations for a ***consistent*** solution that, otherwise, cannot be solved explicitly.

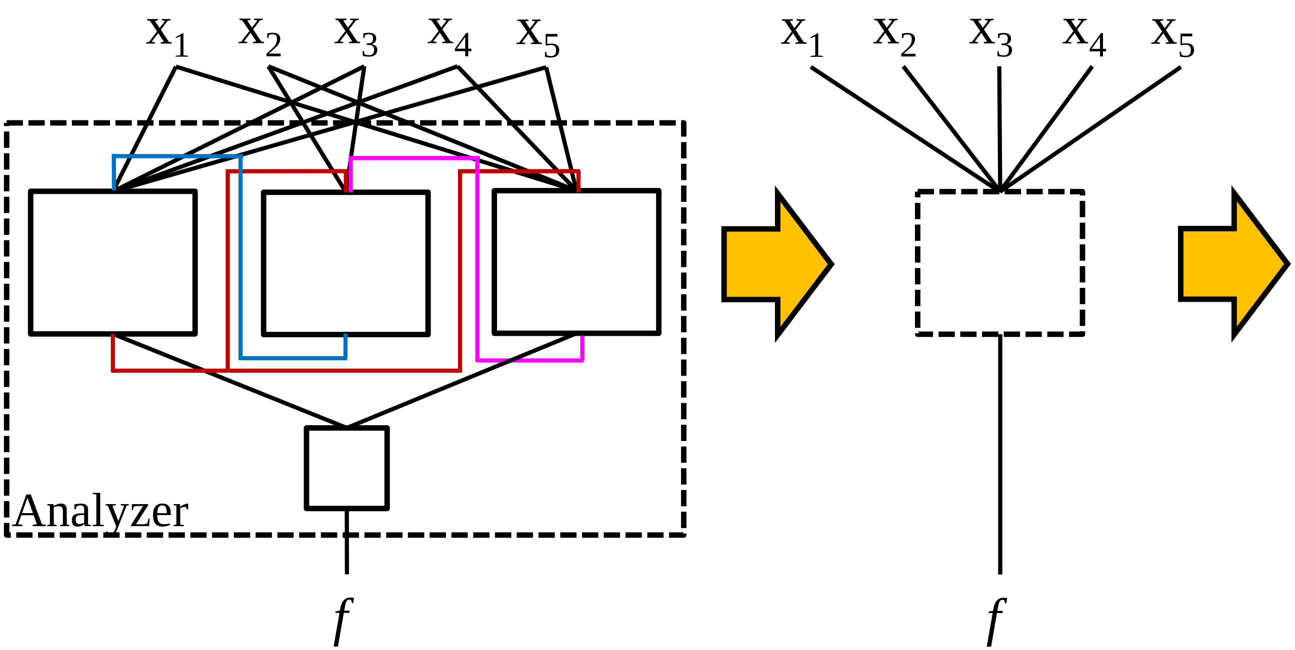
Synonyms for “solvers” include “analyzers” and “drivers.”

Fixed Point Iteration (FPI) – the most commonly used and easily understood solver.

Newton-Raphson – a more powerful solver requiring derivative information.

MDA Solvers

Solvers allow complicated functions to seem simple to the outside observer.

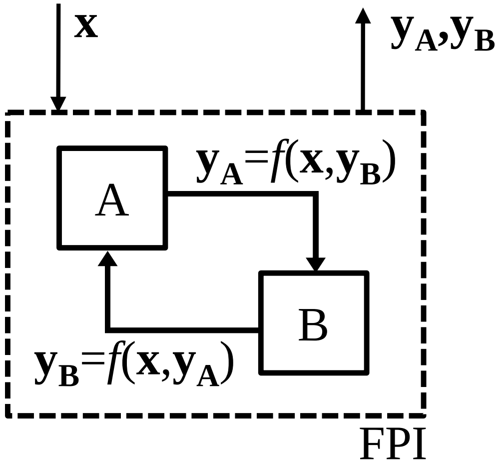


$$\begin{aligned} \min_{\mathbf{X}} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{X}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{X}) = \mathbf{0} \end{aligned}$$

Fixed Point Iteration (FPI) Algorithm

For a coupled two subsystem problem, the steps for FPI are:

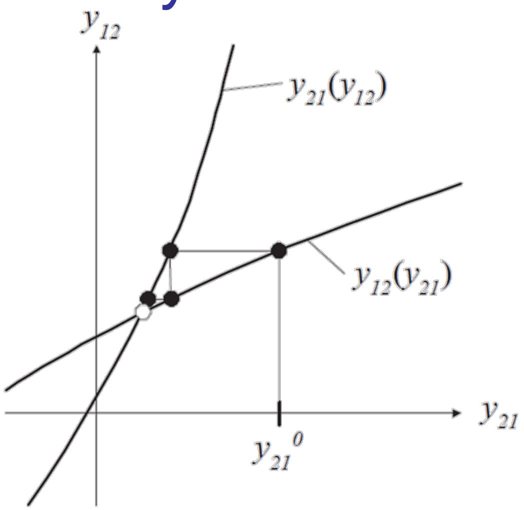
1. Calculate A's output using x and a guess for B's output, $y_{B,0}$.
2. Calculate B's output using x and A's output, y_A.
3. Calculate A's output using x and B's output, y_B.
4. Repeat steps 2 and 3 until the solution changes between iterations are smaller than some specified tolerance.



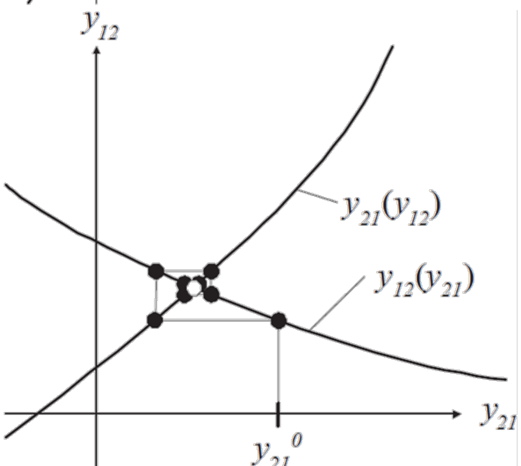
The converged solution is called a *fixed point*.

FPI – Convergence

Depending on the problem, FPI convergence may not occur or may produce a suboptimal solution:



- Monotonic convergence
  - Attractive FP results in convergence

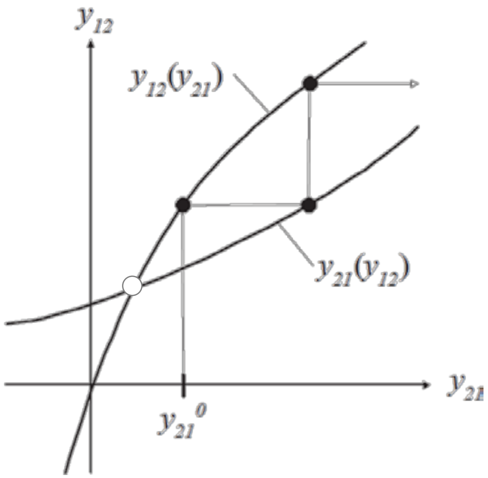


- Oscillatory convergence
  - Orthogonal solution curves results in an infinite loop, despite the existence of an attractive FP

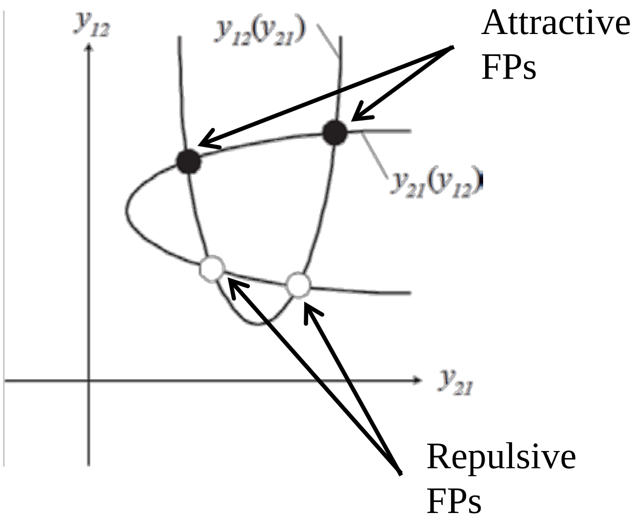
Allison, 2004

FPI – Convergence

- Divergence
  - Due to repulsive FP



- Multiple fixed points
  - Converged solution depends on initial guess
  - Converged solution may not be optimal
  - FPI cannot find repulsive FPs



FPI with Relaxation

Some of the problems with FPI can be improved by incorporating *relaxation.*

Here is a nice description for you to review:

http://sci.tech-archive.net/Archive/sci.math.num-analysis/2005-12/msg00040.html

FPI – Convergence

A system of the form,

$$\boldsymbol{y}^{k+1} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}^k)$$

will converge to a single fixed point if

$$\left\| \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}} \right\| \leq \rho < 1$$

where ρ is a real number and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}$ is the Jacobian matrix of the system.