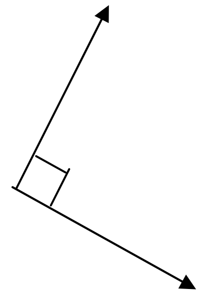
Constructing Orthogonal Arrays

Learning Objectives

- Introduce & explore orthogonality
- Study the standard OAs
- Practice computing DOF of an experiment
- Learn how to select a standard OA
- Introduce means to modify OAs
- Consider studying interactions in OAs

What is orthogonality?

- Geometry



- Vector algebra $\vec{x} \cdot \vec{y} = 0$
- Robust design
  - Form *contrasts* for the columns (i)

$$w_{i1} + w_{i2} + w_{i3} \cdots + w_{i9} = 0$$

  - Inner product of contrasts must be zero

$$w^{<i>} \cdot w^{<j>} = 0$$

Before Constructing an Array

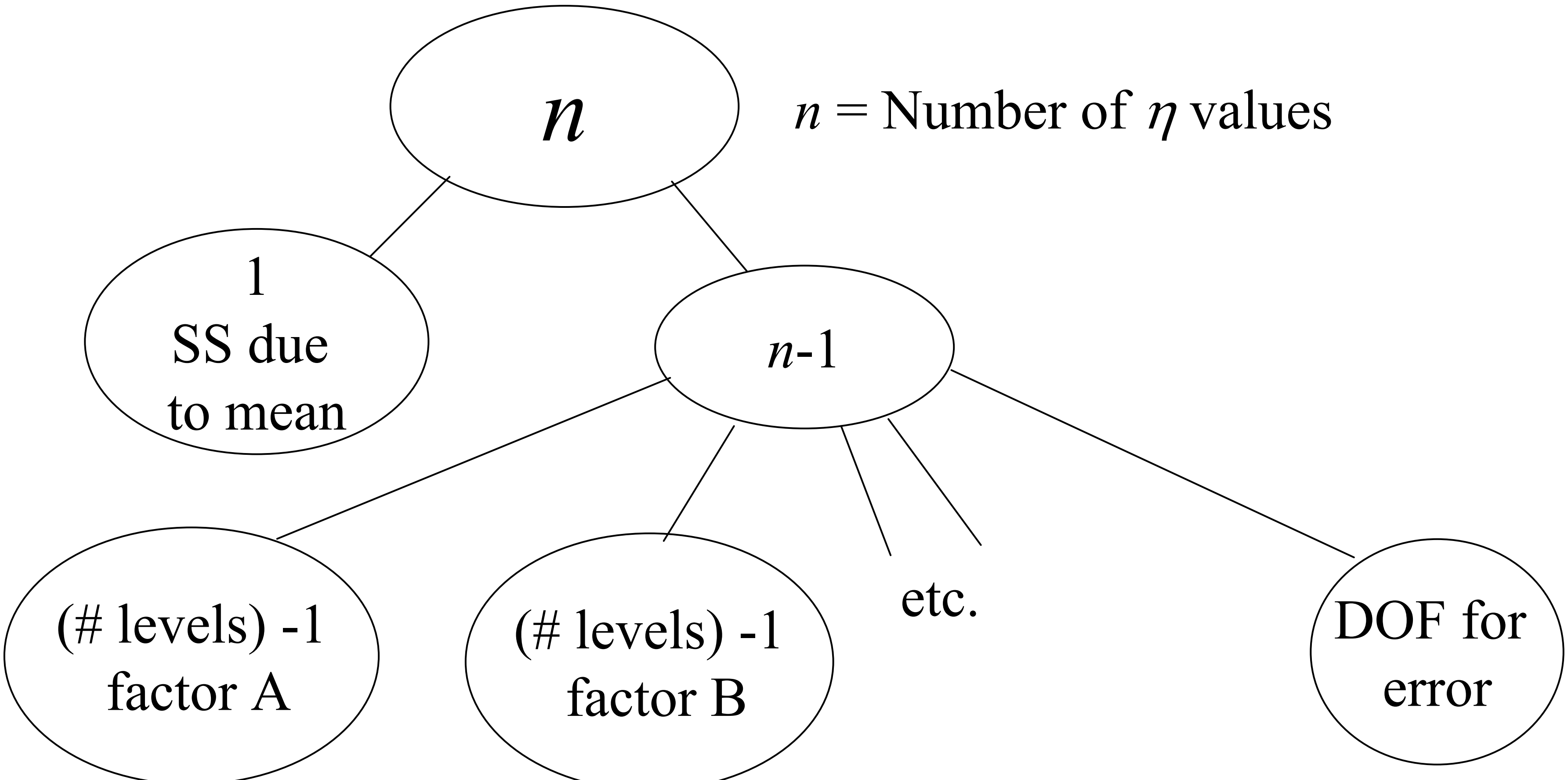
We must define:

- Number of factors to be studied
- Number of levels for each factor
- 2 factor interactions to be studied
- Special difficulties in running experiments

Counting Degrees of Freedom

- Grand mean
  - 1
- Each control factor (e.g., A)
  - (# of levels of A -1)
- Each two factor interaction (e.g., AxB)
  - (DOF for A)x(DOF for B)
- Example -- $2^1 \mathrm{x} 3^7$

Breakdown of DOF



DOF and Modeling Equations

- Additive model

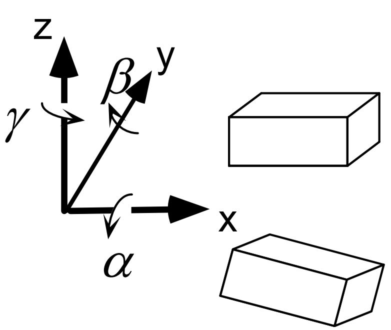
$$\eta(A_i, B_j, C_k, D_l) = \mu + a_i + b_j + c_k + d_l + \cancelto{0}{e}$$

- How many parameters are there?

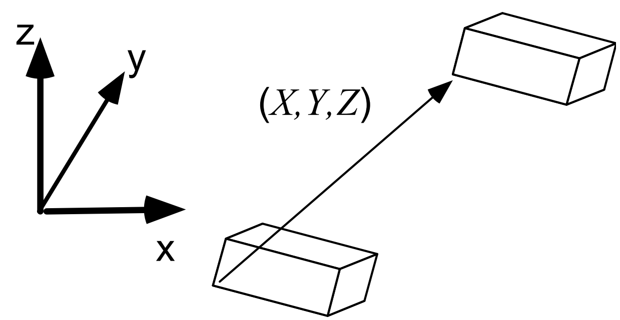
- How many additional equations constrain the parameters?

DOF -- Analogy with Rigid Body Motion

- How many parameters define the position and orientation of a rigid body?
- How do we remove these DOF?



Rotation



Translation

Notation for Matrix Experiments

$$L_9 (3^4)$$

Number of experiments

Number of levels

Number of factors

$$9=(3-1)\times 4+1$$

Standard Orthogonal Arrays

- See table 7.1 on Phadke page 152
- Note: You can never use an array that has fewer rows than DOF req'd
- Note: The number of factors of a given level is a **maximum**
- You can put a factor with fewer columns into a column that has more levels
  - But NOT fewer!

Standard Orthogonal Arrays

Orthogonal Array	Number of Rows	Maximum Number of Factors	Maximum Number of Columns at These Levels			
			2	3	4	5
L_4	4	3	3	-	-	-
L_8	8	7	7	-	-	-
L_9	9	4	-	4	-	-
L_{12}	12	11	11	-	-	-
L_{16}	16	15	15	-	-	-
L'_{16}	16	5	-	-	5	-
L_{18}	18	8	1	7	-	-
L_{25}	25	6	-	-	-	6
L_{27}	27	13	1	13	-	-
L_{32}	32	31	31	-	-	-
L'_{32}	32	10	1	-	9	-
L_{36}	36	23	11	12	-	-
L'_{36}	36	16	3	13	-	-
L_{50}	50	12	1	-	-	11
L_{54}	54	26	1	25	-	-
L_{64}	64	63	63	-	-	-
L'_{64}	64	21	-	-	21	-
L_{81}	81	40	-	40	-	-

Difficulty in Changing Levels

- Some factor levels cost money to change
  - Paper airplane
  - Other examples?
- Note: All the matrices in Appendix C are arranged in increasing order of number of level changes required (left to right)
- Therefore, put hard to change levels in the leftmost columns

Choosing an Array -- Example 1

- 1 two level factor
- 5 three level factors
- What is the number of DOF
- What is the smallest standard array that will work?

Choosing an Array -- Example 2

- 2 two level factor
- 3 three level factors
- What is the number of DOF
- What is the smallest standard array that will work?

Dummy Levels

- Turns a 2 level factor into a 3 level factor (or a 3 to a 4 etc.)
- By creating a “new” level A3 that is really just A1 (or A2)
- Let’s consider example 2
- Question -- What will the factor effect plot look like?

Dummy Levels Preserve Orthogonality

- Let's demonstrate this for Example 2
- But **only** if we assign the dummy level consistently

Considerations in Assigning Dummy Levels

- Desired accuracy of factor level effect
  - Examples?
- Cost of the level assignment
  - Examples?
- Can you assign dummy levels to **more than one factor** in a matrix experiment?
- Can you assign **more than one dummy level** to a single factor?

Compounding Factors

- Assigns two factors to a single column by *merging* two factors into one

Before		After
A_1	B_1	$C_1=A_1B_1$
A_2	B_2	$C_2=A_1B_2$
		$C_3=A_2B_2$

Compounding Factors -- Example

- 3 two level factors
- 6 three level factors
- What is the smallest array we can use?

- How can compounding reduce the experimental effort?

Considerations in Compounding

- Balancing property not preserved between compounded factors

$C_1=A_1B_1$

$C_2=A_1B_2$

$C_3=A_2B_2$

- Main effects confounded to some degree
- ANOVA becomes more difficult

Interaction Tables

- To avoid confounding A and B with AxB, leave a column unassigned
- To know which column to leave unassigned, use an **interaction table**

Interaction Table Example

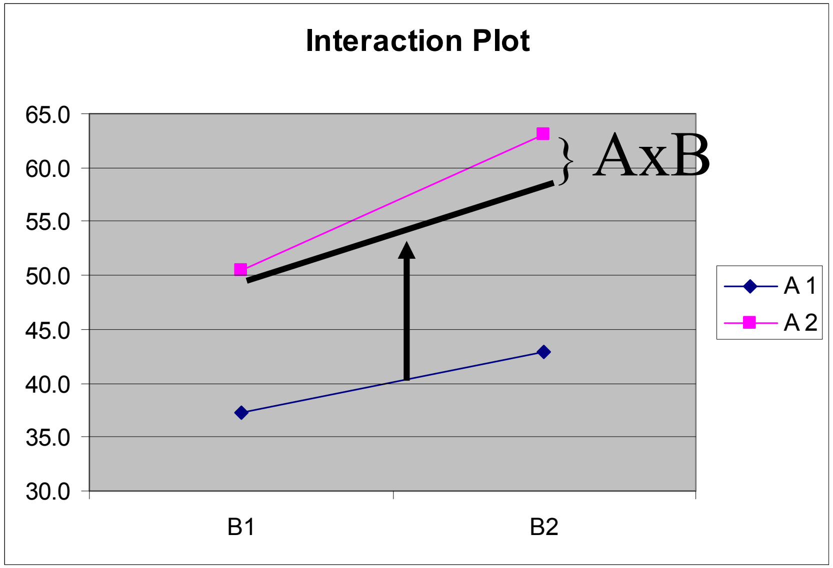
- We are running an L8
- We believe that CF4 and CF6 have a significant interaction
- Which column do we leave open?

Two Level Interactions in L_4

- AxB Interaction =

$$(y_{A_2B_2} - y_{A_1B_2}) - (y_{A_2B_1} - y_{A_1B_1})$$

- As you learned from the noise experiment

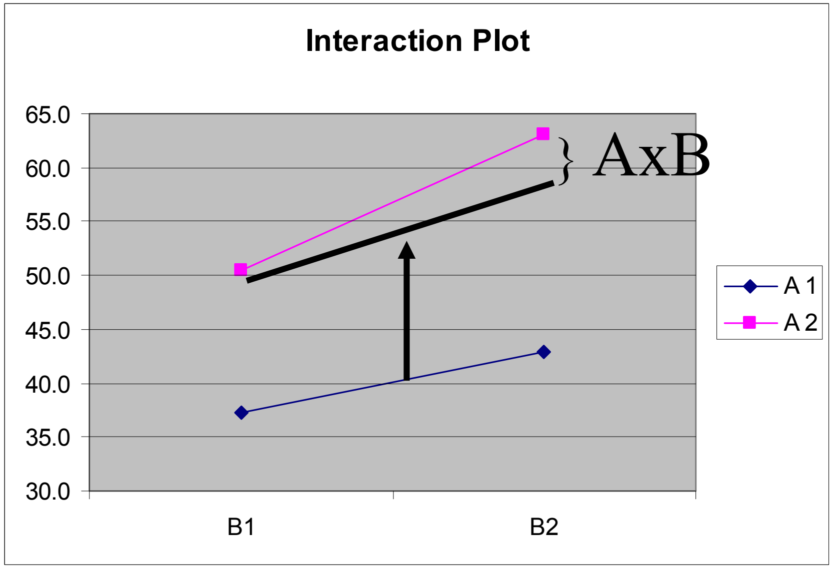


Interactions in Larger Matrices

- AxB Interaction =

$$(\bar{y}_{A_2B_2} - \bar{y}_{A_1B_2}) - (\bar{y}_{A_2B_1} - \bar{y}_{A_1B_1})$$

- Average the rows with the treatment levels listed above



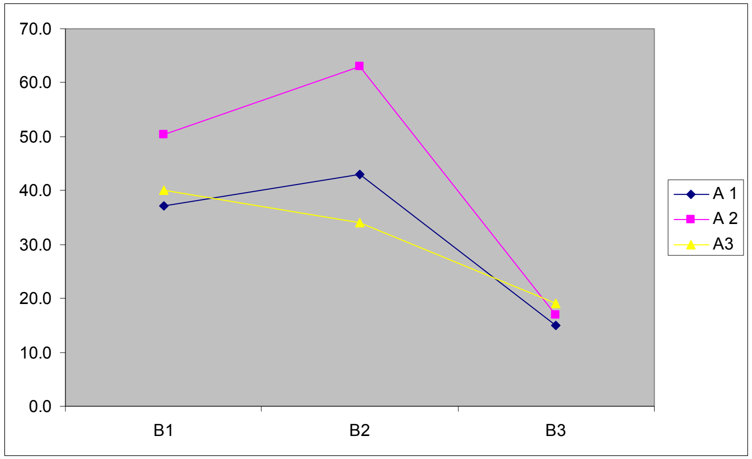
Two Factor Interaction Numerical Example

- 4x6 = $(\bar{y}_{A_2B_2} - \bar{y}_{A_1B_2}) - (\bar{y}_{A_2B_1} - \bar{y}_{A_1B_1})$

Run	c1	4x6	c3	A	c5	B	c7	
1	1	1	1	1	1	1	1	1.2
2	1	1	1	2	2	2	2	1.7
3	1	2	2	1	1	2	2	2.1
4	1	2	2	2	2	1	1	2.6
5	2	1	2	1	2	1	2	4.9
6	2	1	2	2	1	2	1	3.9
7	2	2	1	1	2	2	1	0.9
8	2	2	1	2	1	1	2	1.1

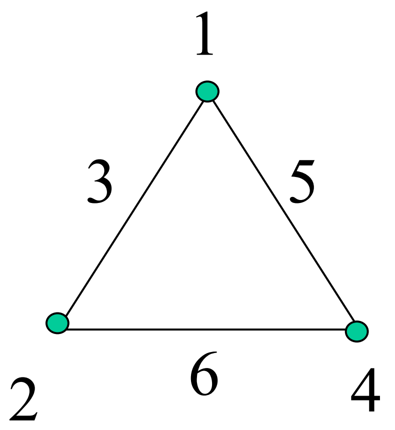
Three Level Interactions

- AxB has 4 DOF
- Each CF has 2DOF
- Requires two unassigned columns (the *right* ones)



Linear Graphs

- To study interaction between CF dot and CF dot, leave CF on connecting line unassigned



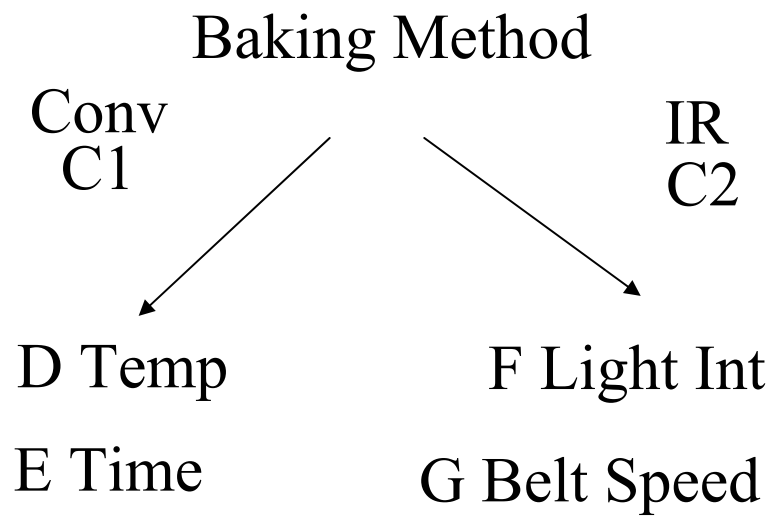
e.g., L_8

Column Merging

- Can turn 2 two level factors into a 4 level factor
- Can turn 2 three level factors into a six level factor
- Need to strike out interaction column (account for the right number of DOF!)
- Example on an L_8

Branching Design

- One control factor determines the appropriate choice of other control factors
- Strike out the parent x child column to preserve the balancing property



Branching Design

Oven Type

Temp / Light Int

T

I

Run	c1	c2	c3	c4	c5	c6	c7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

Is the balancing property preserved?

How can we recover balance?

Next Steps

- Homework #7 due on Lecture 10
- Next session tomorrow
  - Read Phadke Ch. 10
  - Read “Planning Efficient Software Tests”
  - Tought questions:
    - What does software do?
    - How is software different from hardware?
    - How does this affect the application of RD?
- Quiz on Constructing Arrays