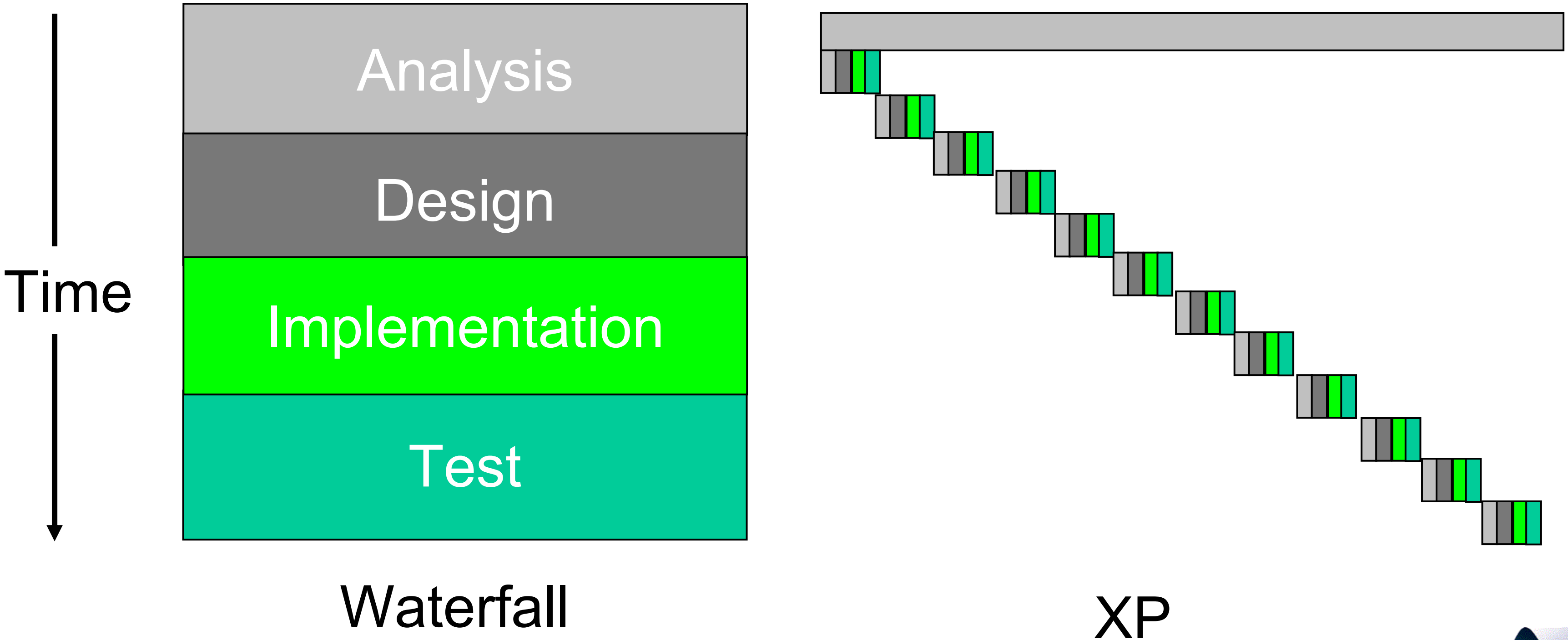
ESD.33 -- Systems Engineering

# Session #14
# **Extreme Programming**



Dan Frey

# Plan for the Session

➡ Comments on Robust Design

- Extreme Programming
  - Beck_Extreme Programming.pdf
- Pair Programming
  - Williams_Pair Programming.pdf
- Next steps

# Mal Atherton

## Rolls-Royce, Controls Lead Engineer

I think a lot of students got lost towards the end today, because some of the details of the array types were difficult to cover in the short time we had.

# Ari Dimitriou

## Chief Signal Processing Engineer, Raytheon

I was wondering if you could point me to more reading material on your DOE results. …

…a common Russian Submariner saying is "Better is the worst enemy of good enough". I feel the analysis you are making with OFAT may be supporting that argument...

# Outer Array

- Induce the same noise factor levels for each row in a balanced manner

| | | | | E1 | E1 | E2 | E2 |
|---|---|---|---|---|---|---|---|
| | | | | F1 | F2 | F1 | F2 |
| | | | | G1 | G2 | G2 | G1 |
| A1 | B1 | C1 | D1 | *1* | *2* | *3* | *4* |
| A1 | B2 | C2 | D2 | *5* | *6* | *7* | *8* |
| A1 | B3 | C3 | D3 | *9* | *10* | *11* | *12* |
| A2 | B1 | C2 | D3 | *13* | *14* | *15* | *16* |
| A2 | B2 | C3 | D1 | *17* | *18* | *19* | *20* |
| A2 | B3 | C1 | D2 | *21* | *22* | *23* | *24* |
| A3 | B1 | C3 | D2 | *25* | *26* | *27* | *28* |
| A3 | B2 | C1 | D3 | *29* | *30* | *31* | *32* |
| A3 | B3 | C2 | D1 | *33* | *34* | *35* | *36* |

inner x outer = L9xL4= 36

# Single Arrays

Example of a suggested design:

- 32 run single array $2^{10-5}$
- 7 control factors, 3 noise factors
- Design generators

A=1, B=2, C=3, D=4, E=234, F=134, G=123,
a=5, b=124, c=1245

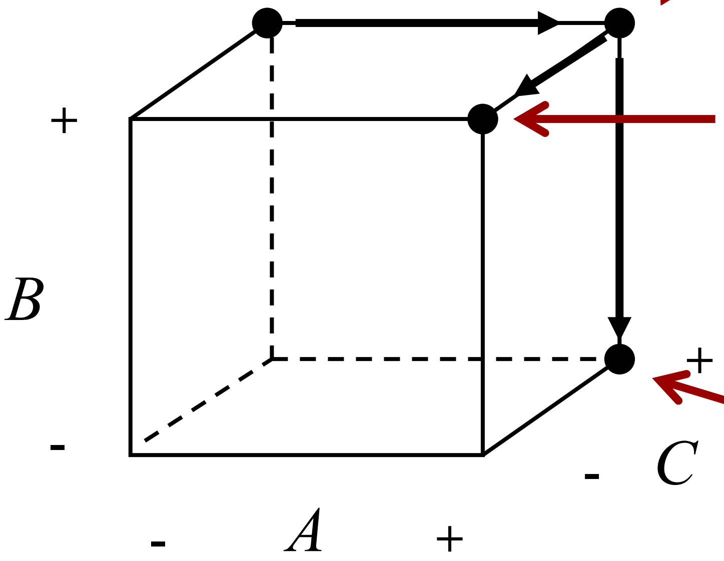
Wu, C. F. J, and H., M. Hamada, 2000, *Experiments: Planning Analysis, and Parameter Design Optimization,* John Wiley & Sons, New York.

# Adaptive One Factor at a Time Experiments

Do an experiment

Change <u>one</u> factor

If there is an improvement, retain the change



If the response gets worse, go back to the previous state

Stop after you've changed every factor

# Results

| | | Basic | | | Fitted | | |
|---|---|---|---|---|---|---|---|
| Method | Experiments | WH | low $w$ | 2nd order | WH | low $w$ | 2nd order |
| $2^7 \times 2^3$ | 1,024 | 60% | 81% | 58% | 50% | 58% | 40% |
| $2^7 \times 2^{3-1}_{III}$ | 512 | 44% | 80% | 52% | 45% | 58% | 40% |
| $2^{10-4}$ | 64 | 8% | 8% | 56% | 18% | 9% | 38% |
| $2^{10-5}$ | 32 | 9% | 3% | 33% | 16% | 9% | 17% |
| $2^{7-4}_{III} \times 2^{3-1}_{III}$ | 32 | 12% | 8% | 51% | 16% | 25% | 38% |
| $OFAT \times 2^{3-1}_{III}$ | 32 | 39% | 56% | 43% | 36% | 42% | 35% |
| $OFAT \times OFAT$ | 32 | 31% | 37% | 41% | 33% | 31% | 27% |
| $2^{10-6}$ | 16 | 4% | 4% | 8% | 4% | 2% | 0% |

# How are Fractional Factorial Designs Formed?

## Matlab Function “fracfact”

```
function [x, conf] = fracfact(gen)
%FRACFACT generates a two-level fractional factorial design.
%
%   X = FRACFACT(GEN) produces the fractional factorial design defined
%   by the generator string GEN.  GEN must be a sequence of "words" separated
%   by spaces.  If the generators string consists of P words using K letters
%   of the alphabet, then X will have N=2^K rows and P columns.
```

# Hadamard Matrix

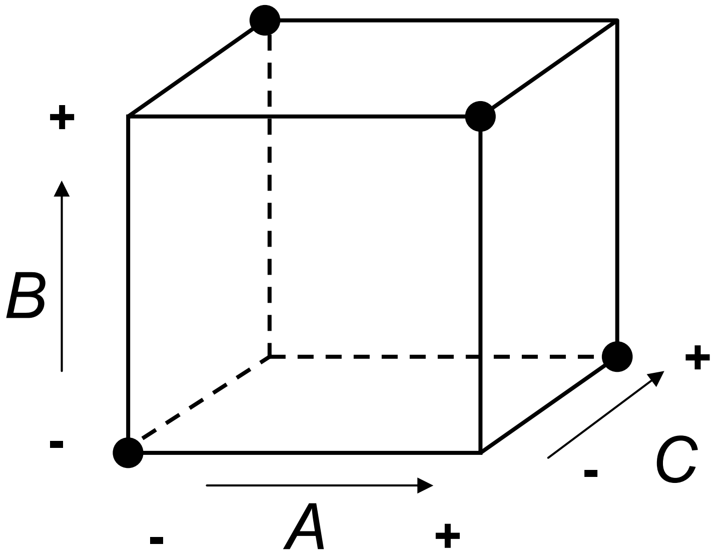
- They are orthogonal $\mathbf{H}^T\mathbf{H} = \mathbf{I}$
- They have only +1 and -1
- Three basic ones exist $\mathbf{H}_2$, $\mathbf{H}_{12}$, and $\mathbf{H}_{20}$
- Others can be constructed recursively

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \mathbf{H}_{2N} = \begin{bmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{bmatrix}$$

- They enable construction of $OA$s

# Fractional Factorial Experiments

## Cuboidal Representation



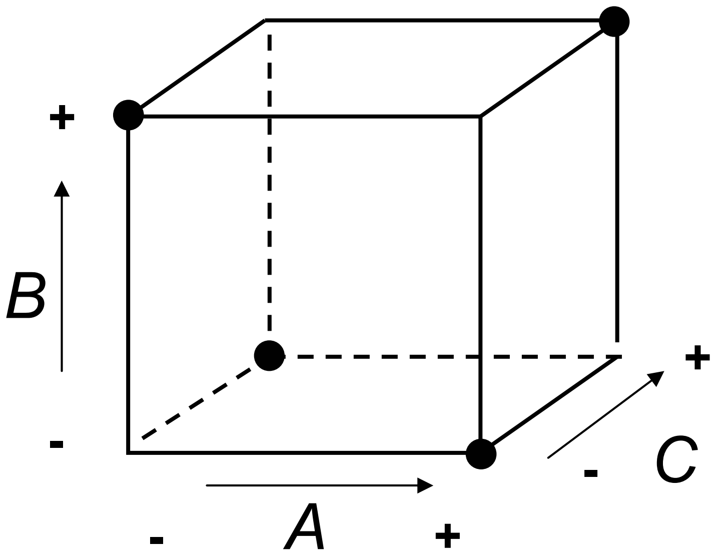
“generator”
is
$ABC$

“defining relation”
is
$I=ABC$

This is a $2^{3-1}$ fractional factorial.

# Fractional Factorial Experiments

## Cuboidal Representation



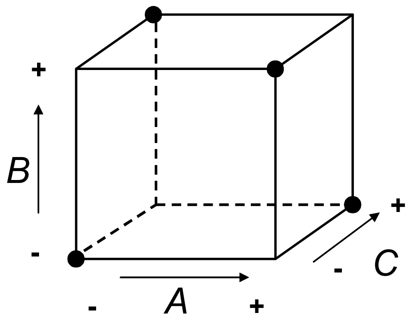
“generator”
is
*-ABC*

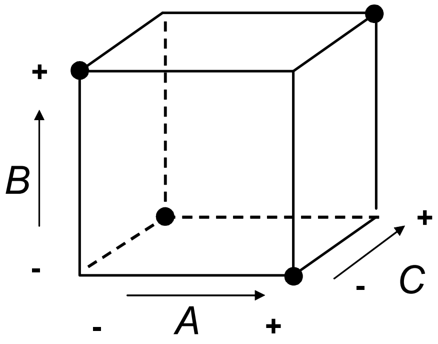
“defining relation”
is
*I=-ABC*

This is the other $2^{3-1}$ fractional factorial.

# Families of Fractional Factorials

- “In practice, it does not matter which fraction is actually used.  Both fractions belong to the same family”
  - Montgomery, D., *Design and Analysis of Experiments*





# John Arruda

Hamilton Sundstrand Engine Systems
Manager - Engine Control Systems & Flight Test Group

- You stated during the lecture that the order of the Control Factors on slide 20 made a difference and that this would result in different tests being conducted. I agree with that. ... What is the approach for deciding which permutation of the balanced orthogonal array to test or does it matter? …would the Factor Effect Plots generated as per slide 24 show the same information, i.e., identify those factors that generate the most improvement independent of which orthogonal array you ran for a given set of factors?

# Greg Andries

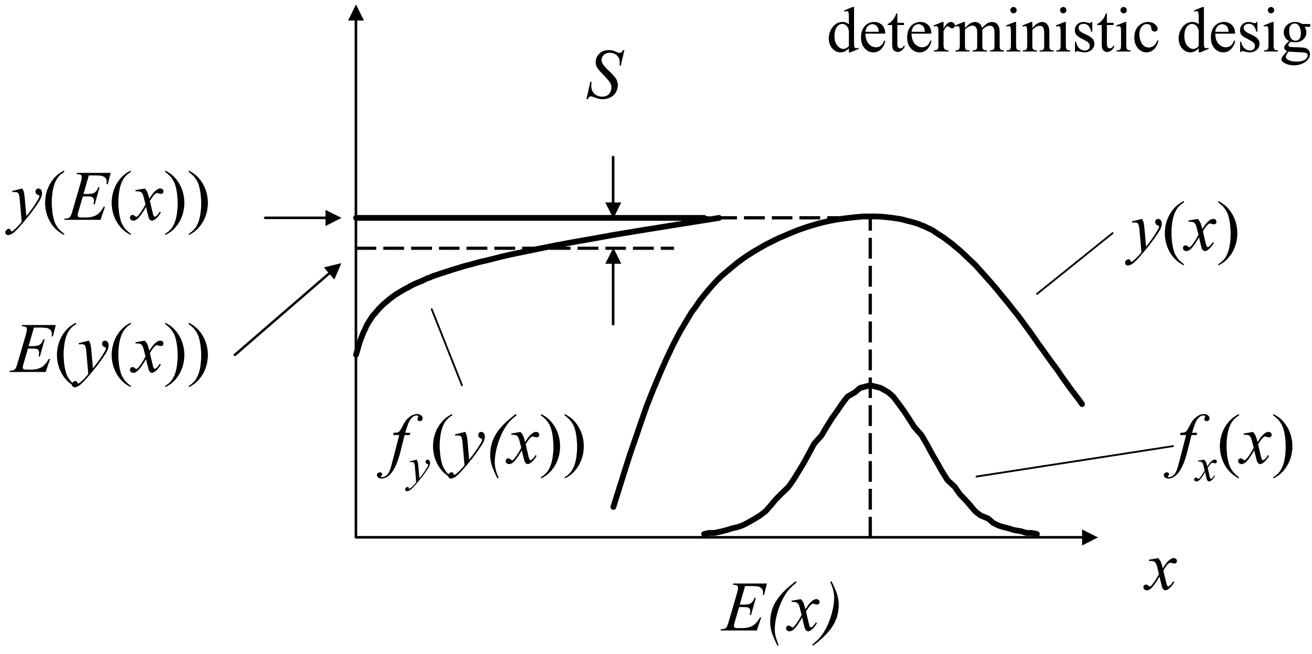
Pratt & Whitney, F135 TAD Validation Manager

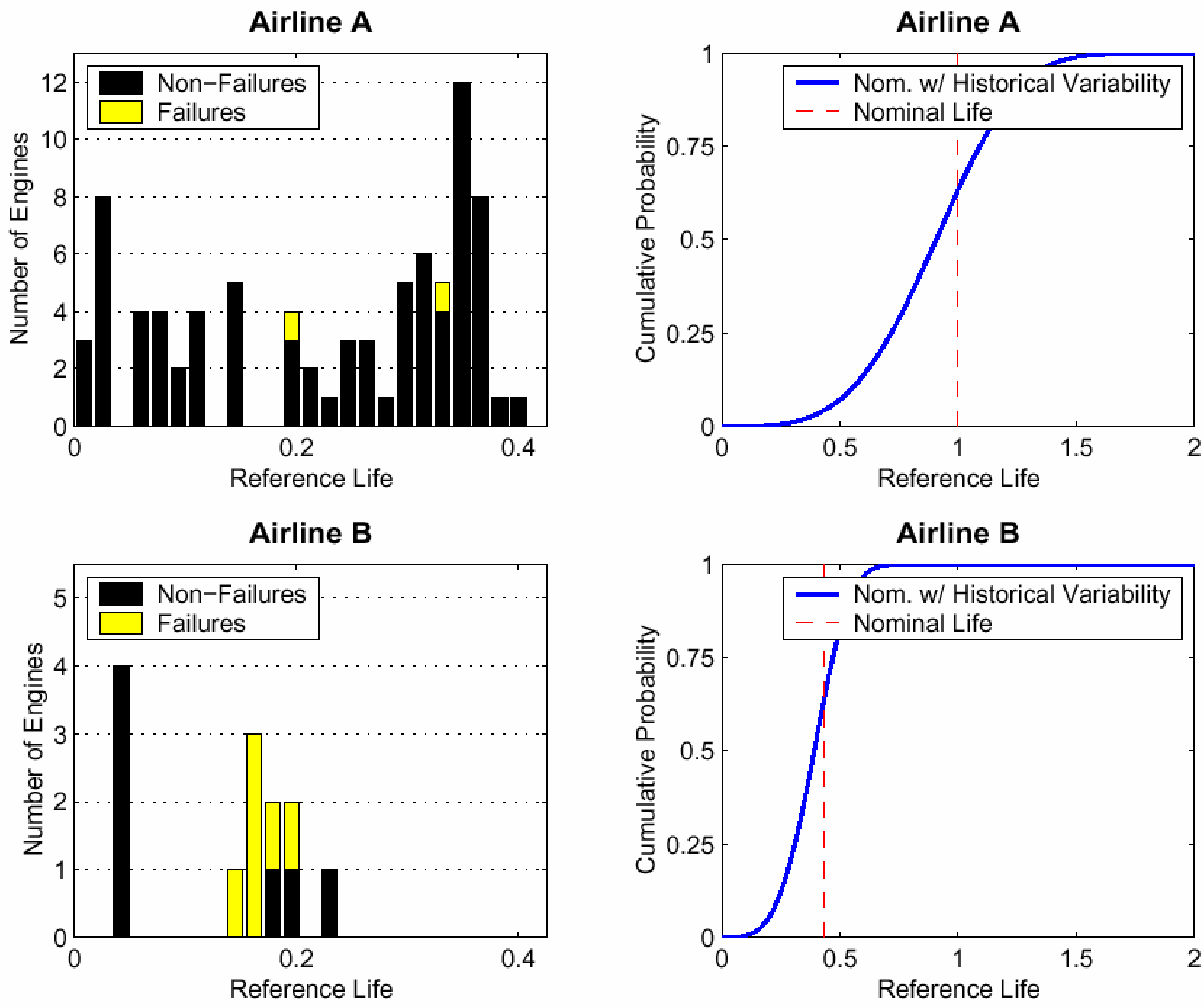
A perfect P&W example of what Dr. Frey is talking about would be cruise TSFC optimization.  There are a number of factors that could be changed to a given engine cycle that could contribute to a reduction in TSFC. … software scheduling changes of variable geometry … turbine flow area change … aero improvement to the compression system...

# Expectation Shift

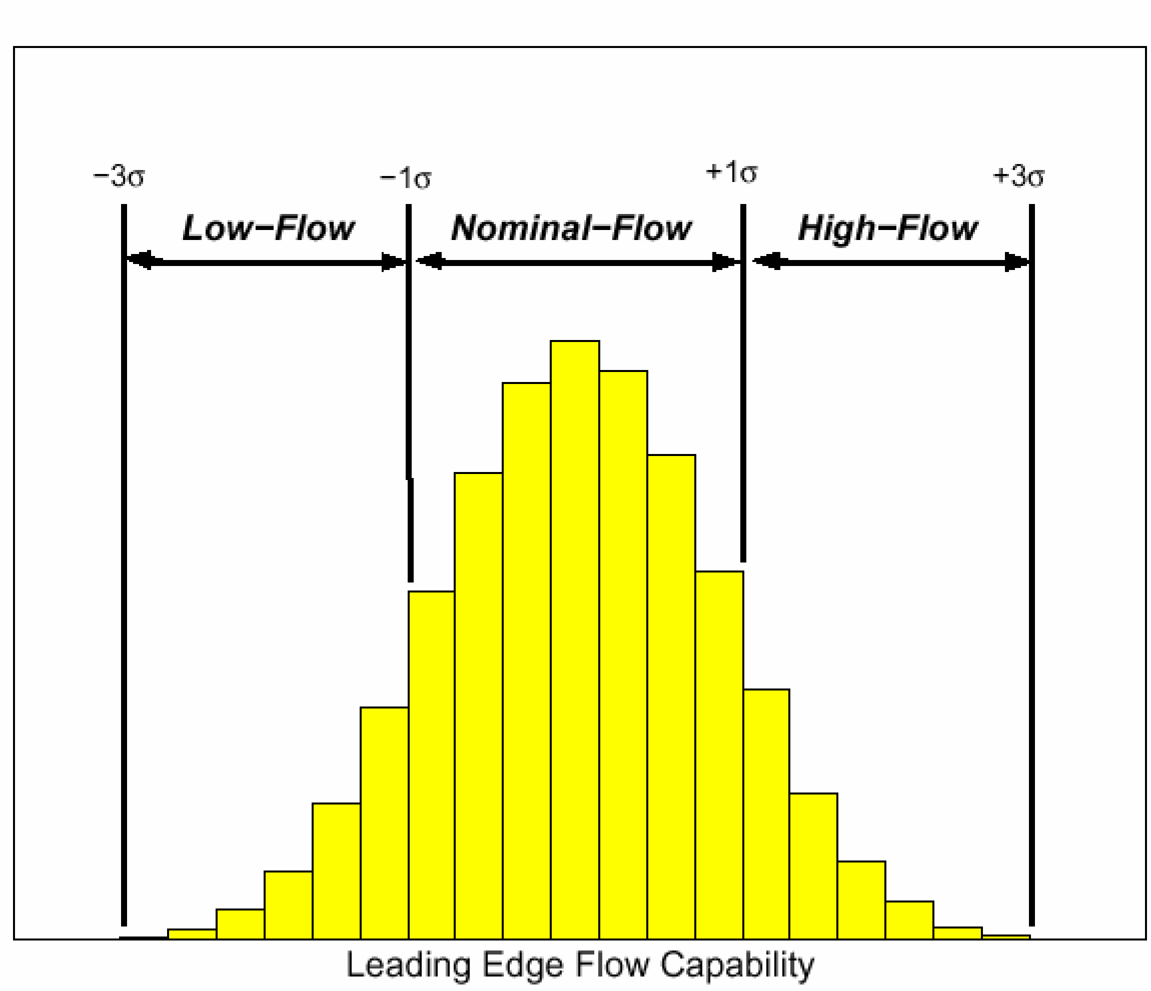
$S = E(y(\mathbf{x})) - y(E(\mathbf{x}))$

Under utility theory (DBD), *S* is a key difference between probabilistic and deterministic design

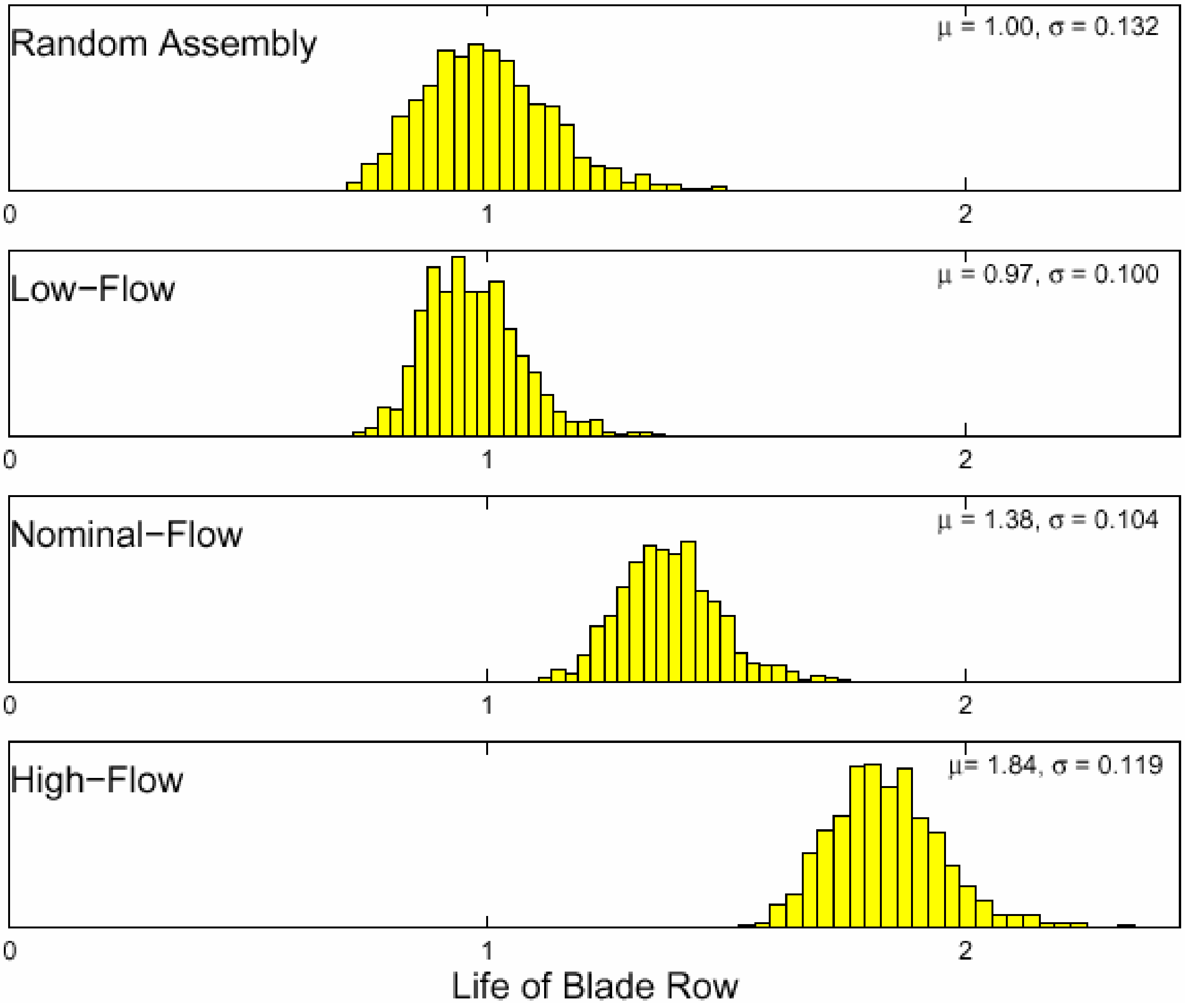




Sidwell, C. V., 2004, “On the Impact of Variability and Assembly on Turbine Blade Cooling Flow and Oxidation Life,” Ph.D. Thesis, MIT.



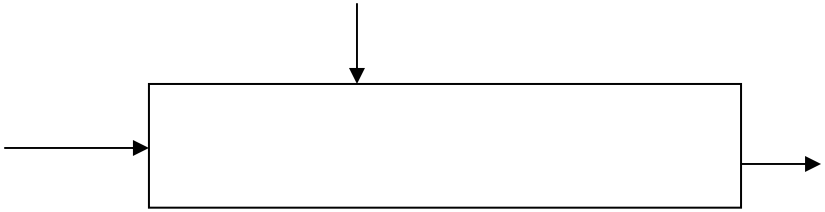
Sidwell, C. V., 2004, “On the Impact of Variability and Assembly on Turbine Blade Cooling Flow and Oxidation Life,” Ph.D. Thesis, MIT.



Sidwell, C. V., 2004, "On the Impact of Variability and Assembly on Turbine Blade Cooling Flow and Oxidation Life," Ph.D. Thesis, MIT.

# Classifying Robustness Inventions

Patent #4,487,333 –
"Fluid Dispensing System"



**Noise**

**Signal**

**Response**

Patent #5,024,105 –
Viscosity-insensitive
variable-area flowmeter

Patent #5,483,840 –
"System for
Measuring Flow"

# Mal Atherton

Rolls-Royce, Controls Lead Engineer

In my experience, the main problem is the tendency to regard the spec (tolerance based) as the benchmark for all design decisions. We even have trouble convincing the company to allow us to perform robustness tests…Go fix the spec and stop asking for expensive and time consuming robustness tests we are told. … robustness is a design property that we should care about rather than just meeting the spec. Is this an appropriate way to view the issue?

# History of Tolerances

- pre 1800 -- Craft production systems
- 1800 -- Invention of machine tools & the English System of manufacture
- 1850 -- Interchangeability of components & the American system of manufacture

Jaikumar, Ramachandran, 1988, *From Filing and Fitting to Flexible Manufacture*

# Craft Production

- Drawings communicated rough proportion and function
- Drawings carried no specifications or dimensions
- Production involved the master, the model, and calipers

# The English System

- Greater precision in machine tools
- General purpose machines
  - Maudslay invents the slide rest

- Accurate measuring instruments
  - Micrometers accurate to 0.001 inch
- Engineering drawings
  - Monge "La Geometrie Descriptive"
  - Orthographic views and dimensioning
- Parts made to fit to one another
  - Focus on perfection of fit

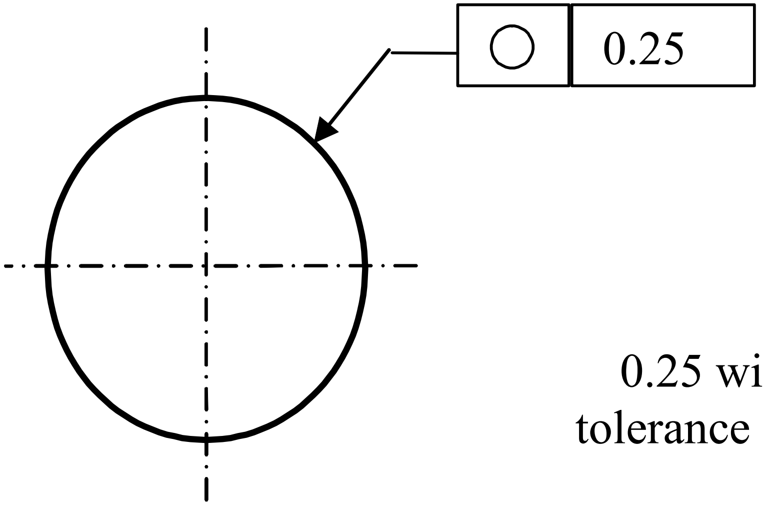
# The American System

- Interchangeability required for field service of weapons
- Focus on management of clearances
- Go-no go gauges employed to ensure fit
- Allowed parts to be

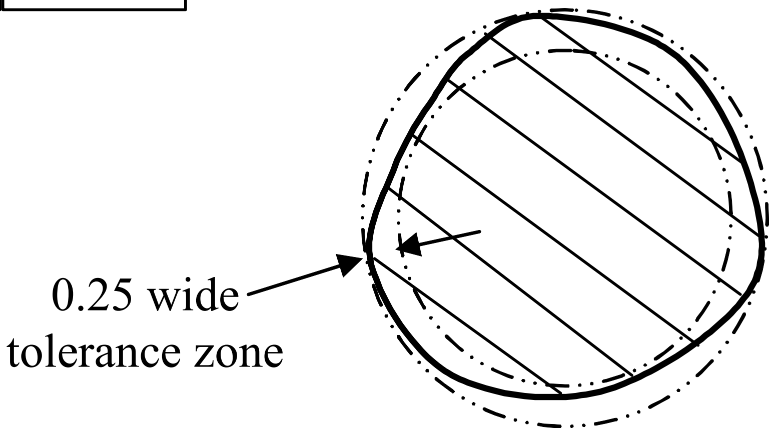
made in large lots

Go - no go gauges

# Tolerance of Form



**THIS ON A DRAWING**

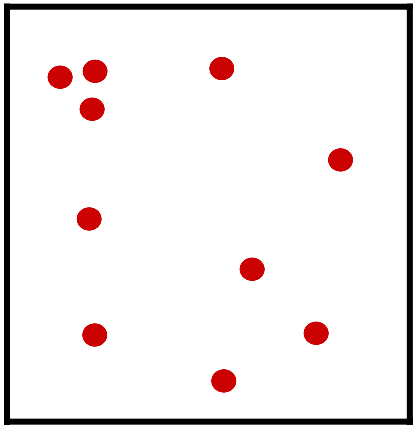


**MEANS THIS**

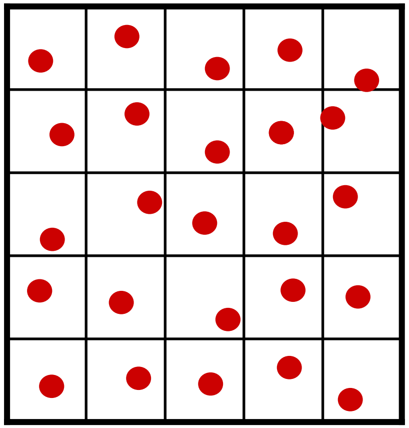
# Basic Tolerancing Principles ref. ANSI Y14.5M

- Each dimension must have a tolerance
- Dimensions of size, form, and location must be complete
- No more dimensions than necessary shall be given
- Dimensions should not be subject to more than one interpretation
- Do not specify manufacturing method

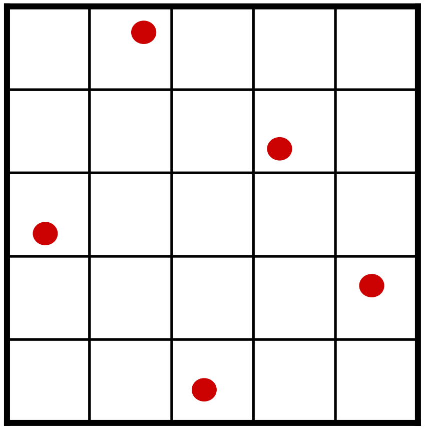
# Sampling Techniques for Computer Experiments



Random Sampling



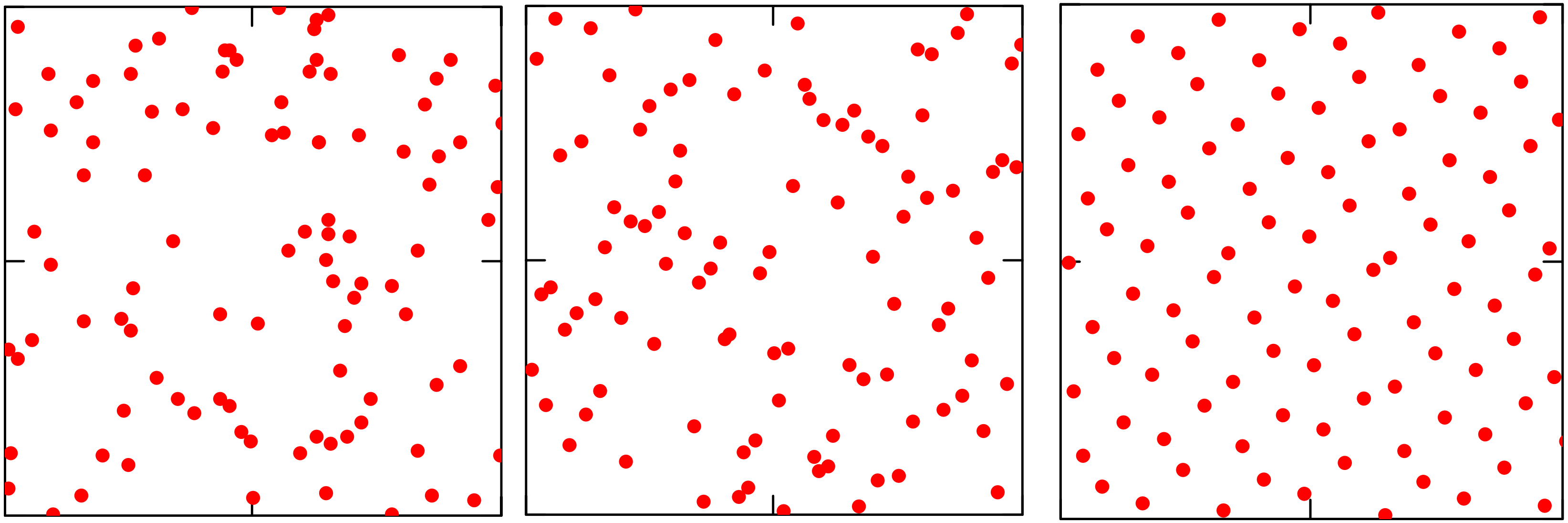
Stratified Sampling



Latin Hypercube Sampling

# Hammersley Sequence Sampling

- A sampling scheme design for low “discrepancy”
- Demonstrated to converge to 1% accuracy 3 to 40 times more quickly than LHS
- Still generally requires >100 samples
  - [Kalagnanam and Diwekar, 1997]



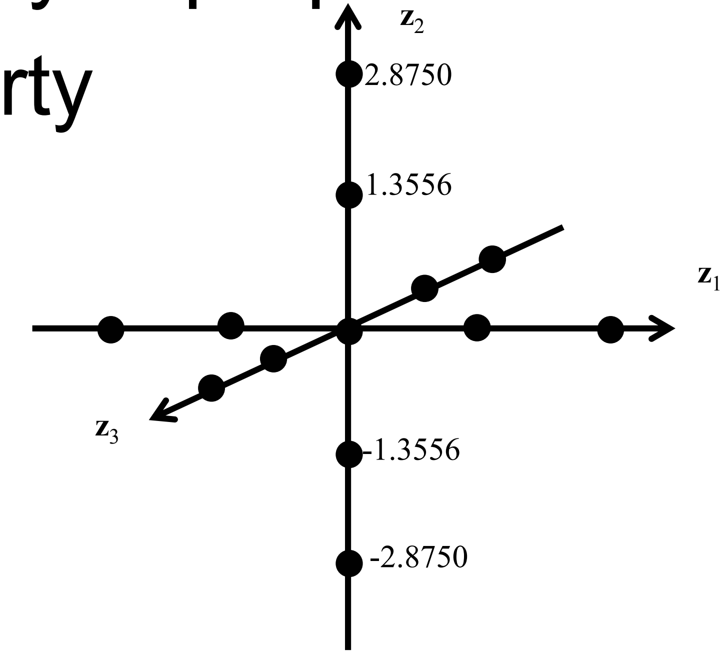
Monte Carlo

Latin

Hammersley

# Proposed Method

- Simply extend quadrature to many variables
- Will be exact to if factor effects of 4th polynomial order linearly superpose
- Lacks projective property
- Poor divergence



# Why Neglect Interactions?

$$\eta(\mathbf{z}) = \beta_0 + \sum_{i=1}^{n} \beta_i \mathbf{z}_i + \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i \le j}}^{n} \beta_{ij} \mathbf{z}_i \mathbf{z}_j + \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i \le j}}^{n} \sum_{\substack{k=1 \\ k \le k}}^{n} \beta_{ijk} \mathbf{z}_i \mathbf{z}_j \mathbf{z}_k + \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i \le j}}^{n} \sum_{\substack{k=1 \\ k \le j}}^{n} \sum_{\substack{l=1 \\ l \le k}}^{n} \beta_{ijkl} \mathbf{z}_i \mathbf{z}_j \mathbf{z}_k \mathbf{z}_l$$

If the response is polynomial

$$\sigma^2(\eta(\mathbf{z})) = \sum_{i=1}^{n} \left( \beta_i^{\ 2} + 2\beta_{ii}^{\ 2} + 6\beta_i \beta_{iii} + 15\beta_{iii}^{\ 2} + 24\beta_{ii}\beta_{iiii} + 96\beta_{iiii}^{\ 2} \right) +$$

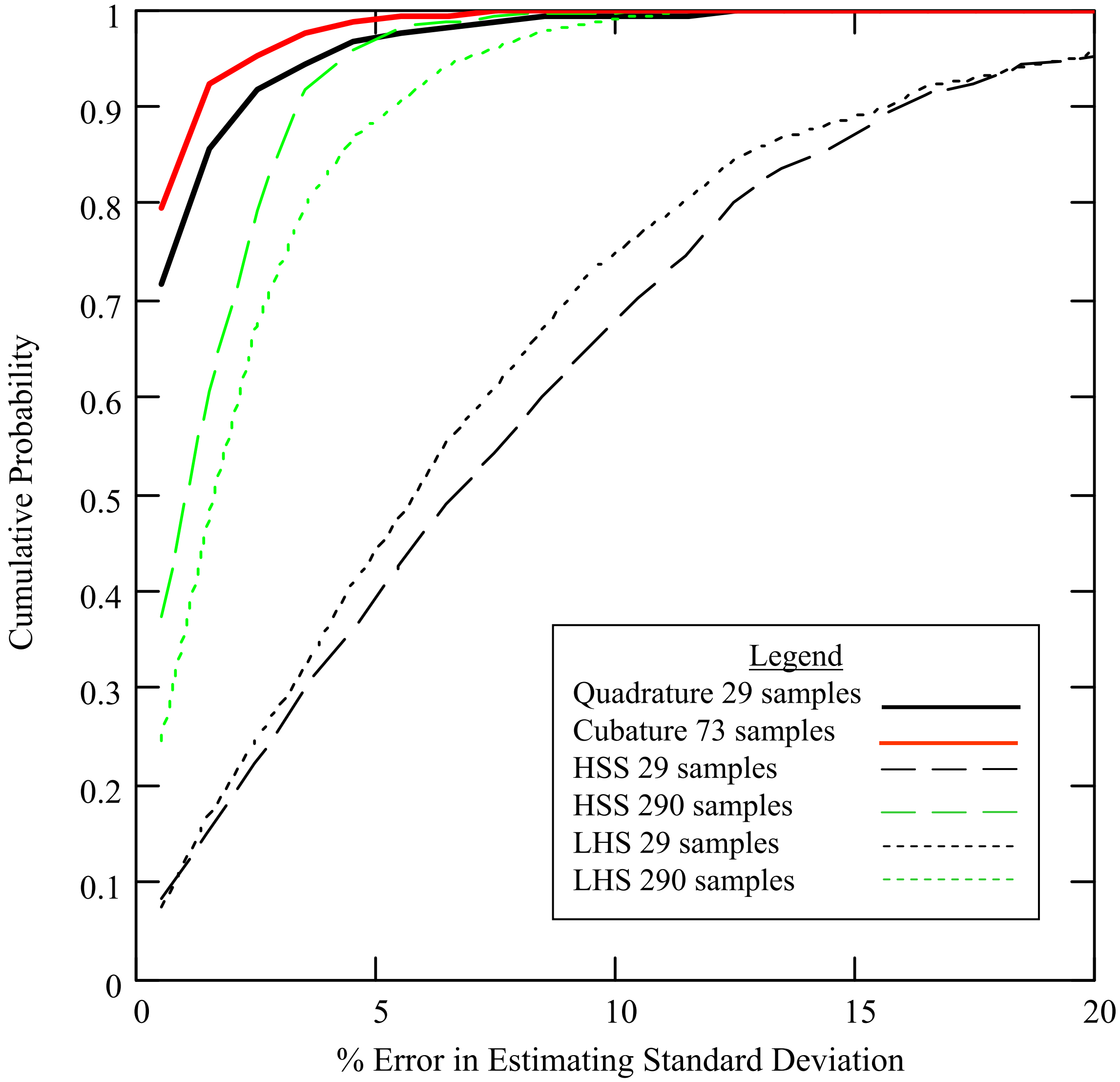
$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ i<j}}^{n} \begin{pmatrix} \beta_{ij}^{\ 2} + 3\beta_{iij}^{\ 2} + 3\beta_{ijj}^{\ 2} + 15\beta_{iiij}^{\ 2} + 15\beta_{ijjj}^{\ 2} + 8\beta_{iijj}^{\ 2} + \\ 2\beta_i \beta_{ijj} + 2\beta_j \beta_{iij} + 4\beta_{ii}\beta_{iijj} + 4\beta_{jj}\beta_{iijj} + 6\beta_{ij}\beta_{iiij} + \\ 6\beta_{ij}\beta_{ijjj} + 6\beta_{iii}\beta_{ijj} + 6\beta_{jjj}\beta_{iij} + 24\beta_{iiii}\beta_{iijj} + 24\beta_{jjjj}\beta_{iijj} \end{pmatrix} +$$

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ i<j}}^{n} \sum_{\substack{k=1 \\ j<k}}^{n} \begin{pmatrix} \beta_{ijk}^{\ 2} + 3\beta_{iijk}^{\ 2} + 3\beta_{ijjk}^{\ 2} + 3\beta_{ijkk}^{\ 2} + 2\beta_{iij}\beta_{jkk} + 2\beta_{iik}\beta_{jjk} + \\ 2\beta_{ijj}\beta_{ikk} + 4\beta_{iijj}\beta_{jjkk} + 4\beta_{iijj}\beta_{iikk} + 4\beta_{iikk}\beta_{jjkk} + 6\beta_{iiij}\beta_{ijkk} + \\ 6\beta_{iiik}\beta_{ijjk} + 6\beta_{ijjj}\beta_{ijkk} + 6\beta_{jjjk}\beta_{iijk} + 6\beta_{jkkk}\beta_{iijk} \end{pmatrix} +$$

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ i<j}}^{n} \sum_{\substack{k=1 \\ j<k}}^{n} \sum_{\substack{l=1 \\ k<l}}^{n} \begin{pmatrix} \beta_{ijkl}^{\ 2} + 2\beta_{iijj}\beta_{kkll} + 2\beta_{iikk}\beta_{jjll} + 2\beta_{iill}\beta_{jjkk} + 2\beta_{iijk}\beta_{jkll} \\ + 2\beta_{ijjk}\beta_{ikll} + 2\beta_{ijkk}\beta_{ijll} + 2\beta_{iijl}\beta_{jkkl} + 2\beta_{ijjl}\beta_{ikkl} + 2\beta_{ijll}\beta_{ijkk} \end{pmatrix}$$

Then the effects of single factors have larger contributions to $\sigma$ than the mixed terms.

# Fourth Order – RWH Model Fit to Data



$d$=7

4$d$+1=29

$d^2$+3$d$+3=73

# Continuous-Stirred Tank Reactor

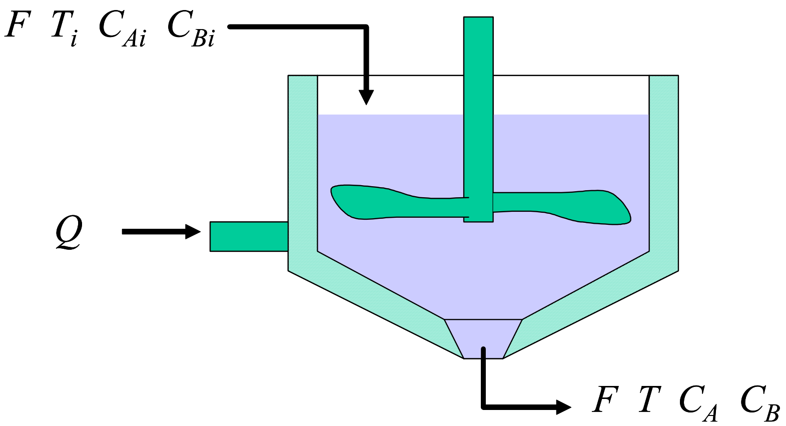
- Objective is to generate chemical species B at a rate of 60 mol/min

$$Q = F\rho C_p (T - T_i) + V(r_A H_{RA} + r_B H_{RB})$$

$$C_A = \frac{C_{Ai}}{1 + k_A^0 e^{-E_A/RT}\tau} \quad C_B = \frac{C_{Bi} + k_A^0 e^{-E_A/RT}\tau C_A}{1 + k_B^0 e^{-EB/RT}\tau}$$

$$-r_A = k_A^0 e^{-E_A/RT} C_A$$

$$-r_B = k_B^0 e^{-E_B/RT} C_B - k_A^0 e^{-E_A/RT} C_A$$

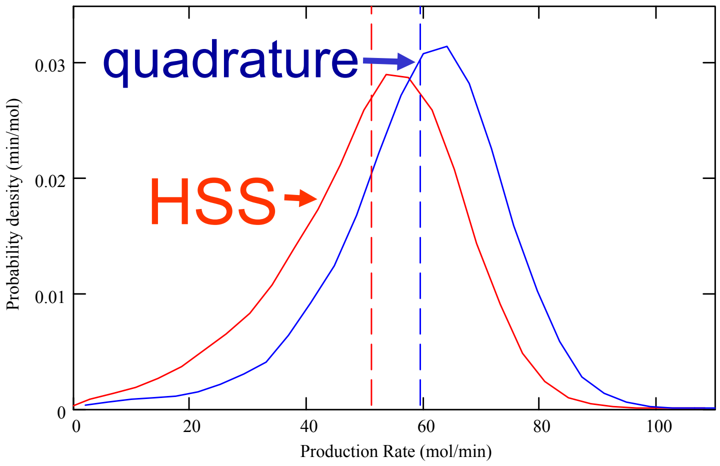


Adapted from Kalagnanam and Diwekar, 1997, “An Efficient Sampling Technique for Off-Line Quality Control”, *Technometrics* (39 (3) 308-319.

# Comparing HSS and Quadrature

## Hammersley Sequence

- Required ~150 points
- 1% accuracy $\sigma^2$
- $\downarrow\sigma^2$ from 1,638 to 232
- Nominally on target
- Mean 15% off target



## Quadrature

- Used 25 points
- 0.3% accuracy in $\mu$
- 9% accuracy in $(y\text{-}60)^2$ far from optimum
- 0.8% accuracy in $(y\text{-}60)^2$ near to optimum
- Better optimum, on target and slightly lower variance
- $E(L(y)) = 208.458$

# Plan for the Session

- Comments on Robust Design
- Extreme Programming
  - Beck_Extreme Programming.pdf
- Pair Programming
  - Williams_Pair Programming.pdf
- Next steps

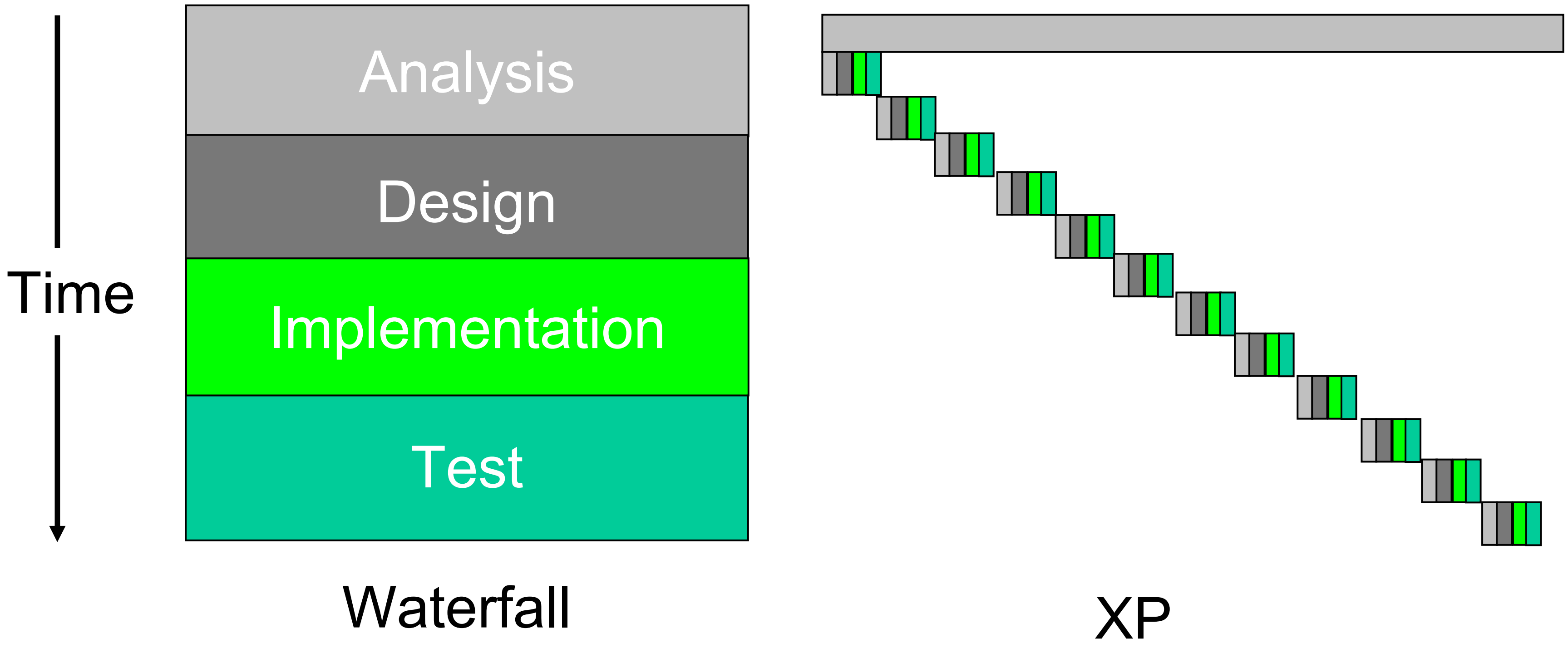
# Recap of “No Silver Bullet”

- What did Fred Brooks Say?
- What is hard about software?
- “Promising attacks on the conceptual essence”
  - Buy versus build
  - Requirements refinement and rapid prototyping
  - Incremental development (grow, don’t build)
  - Great designers

# Roots of XP

- Christopher Alexander
- Spiral development
- Scrum
- Evolutionary delivery
- Kent Beck
  - 1996 – 4 principles
  - 1999 – evangelized
  - 2004 – fairly wide use

# What is XP?



XP turns the conventional software process sideways.
What if we didn’t settle for taking a cleaver to the waterfall?
What if we could throw it in a blender?

# XP Practices

- Planning game
- Small releases
- Metaphor
- Simple design
- Tests
- Refactoring
- Pair programming
- Continuous integration
- Collective ownership
- On-site customer
- 40 hour weeks
- Open workspace
- Just rules

# Stories

- Story = A use case that can fit on an index card
- Each story defines something the software should be able to do
- Estimate the resources required to implement each story
- Collect a set of stories to form a release
- Each story must be testable

# Unit Testing

- “If there is a technique at the heart of XP it is unit testing”
- Tests are what would convince the customer that the stories are completed
- Programmers write their OWN tests
- Write the tests BEFORE the code
- Test run automatically
- Tests are permanent and accumulate

# Pair Programming

- Programmers sign up for tasks for which they take responsibility for
- Responsible programmer finds a partner
- The pair shares a single machine
  - One person codes
  - The other critiques, helps, etc
- More later

# Studies of Pair Progamming

- 15 experienced programmers, 5 individuals, 5 pairs – ~40% faster, higher quality [Nosek, 1998]
- 41 senior students, self selected to pair or individual programming – ~40-50% faster, more test cases passed [Williams, 2000]
- A good amount of anecdotal evidence from industry case studies

# Other Arguments for Pair Programming

- Mistake avoidance – “two sets of eyes”
- Programmers like it
- If there is turn-over, you retain knowledge
- Facilitates learning from peers
- Organizational unity

# What is XP good for?

- Products where the requirements are highly uncertain
- Modest sized projects / teams

# Other “Agile” Methods

- Scrum
- XP
- Crystal Orange
- DSDM

# Next Steps

- Continue preparing for exam
  - Exam posted next Tuesday AFTER session
- See you at Tuesday’s session
  - General Electric Aircraft Engine case study
  - 8:30AM Tuesday, 27 July
- Reading assignment
  - www.geae.com/education/engines101
  - Cumpsty_Jet Propulsion ch4.pdf