1. MOTIVATION AND BACKGROUND

The Design-Oriented Analysis (D-OA) Paradigm

Premise:

Not much of what you learned in school has turned out to be much use

Premise:

Not much of what you learned in school has turned out to be much use

So why are you here, listening to another professor?

Premise:

Not much of what you learned in school has turned out to be much use

So why are you here, listening to another professor?

Because you are going to see a completely different approach, from the get-go:

Premise:

Not much of what you learned in school has turned out to be much use

So why are you here, listening to another professor?

Because you are going to see a completely different approach, from the get-go:

You don't solve equations simultaneously; instead:

You solve them sequentially

Premise:

Not much of what you learned in school has turned out to be much use

So why are you here, listening to another professor?

Because you are going to see a completely different approach, from the get-go:

You don't solve equations simultaneously; instead:

You solve them sequentially

An Engineer's story

Premise:

Not much of what you learned in school has turned out to be much use

So why are you here, listening to another professor?

Because you are going to see a completely different approach, from the get-go:

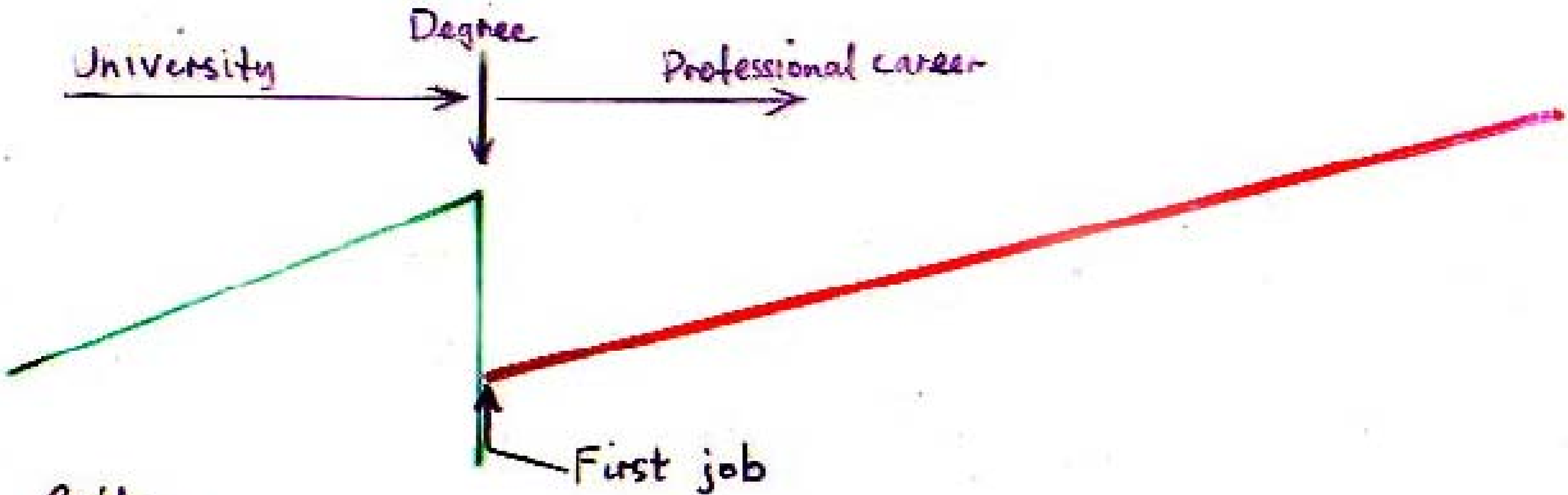
You don't solve equations simultaneously; instead:

You solve them sequentially

An Engineer's story

Falling off a cliff

Most of us "fall off a cliff" when we begin our first job



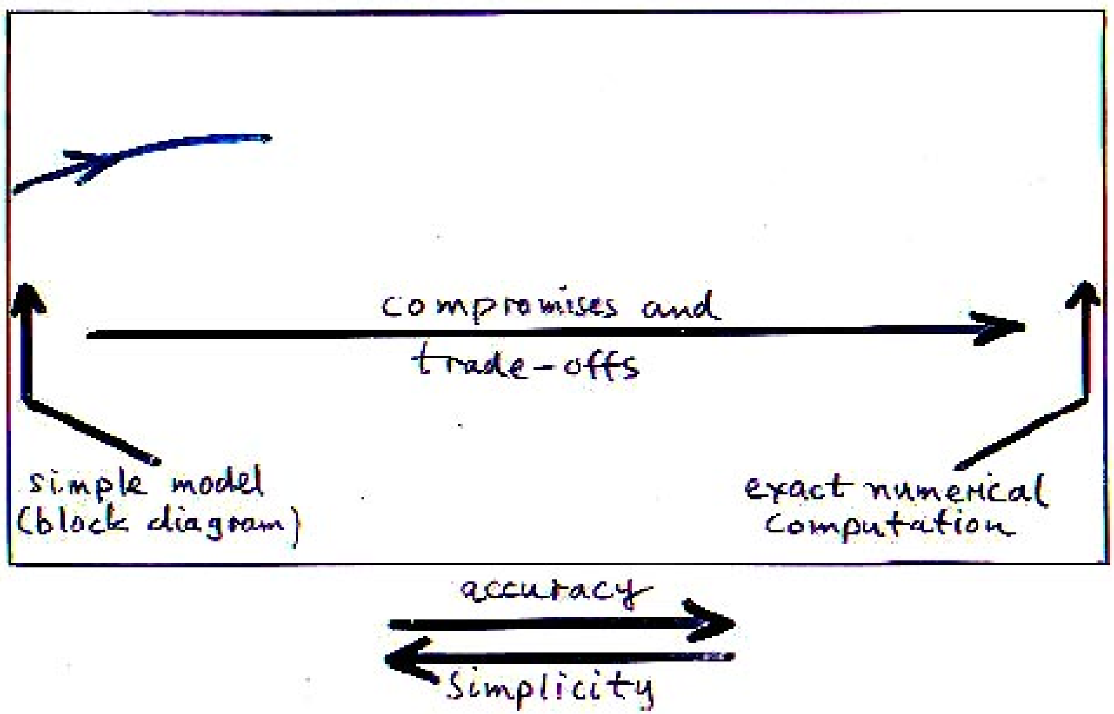
Problem:

New graduate engineers are unable to translate the principles and methods they have learned to the real world.

What can be done?

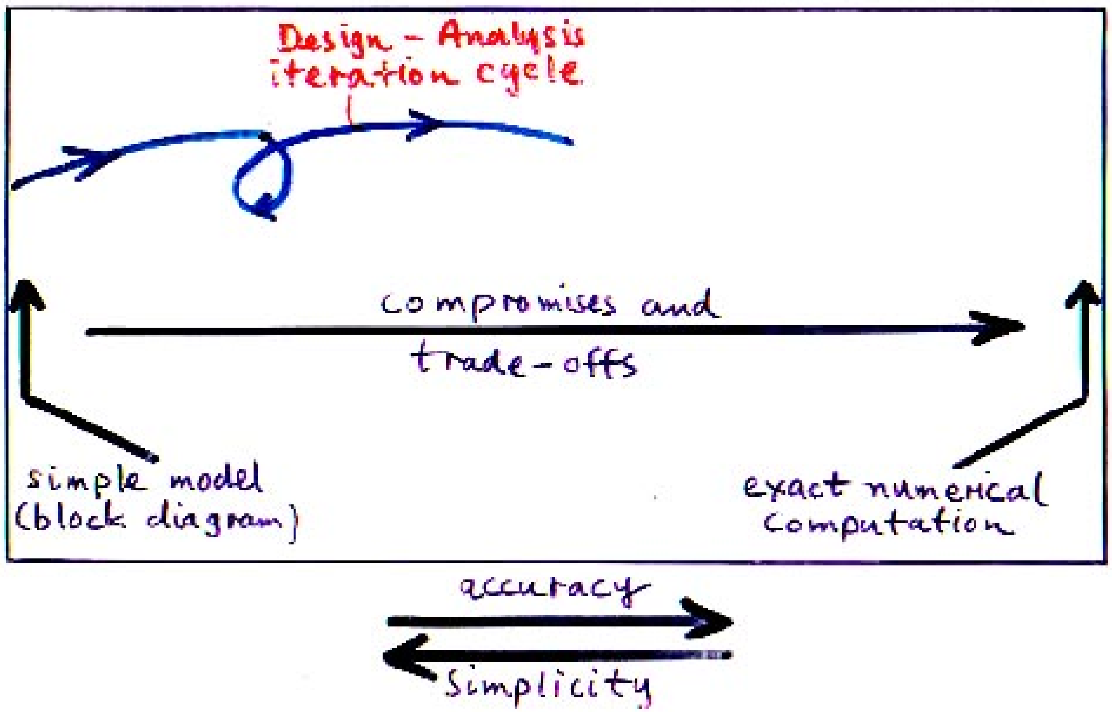
The Design Process

Design iteration loops



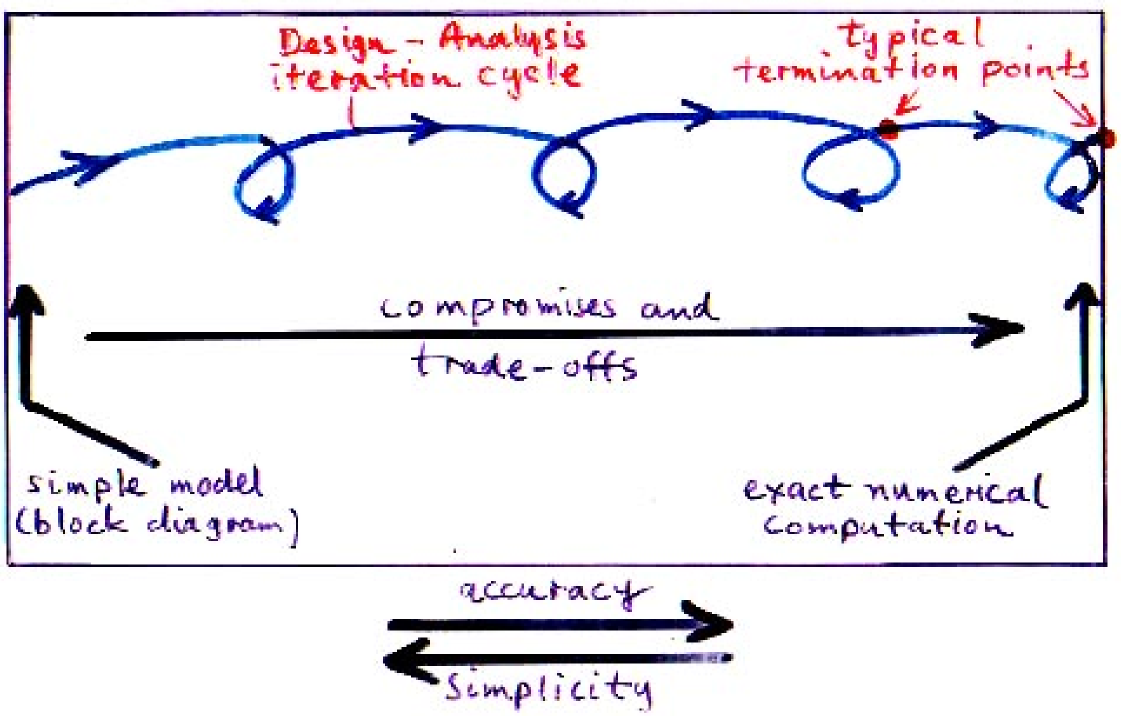
The Design Process

Design iteration loops

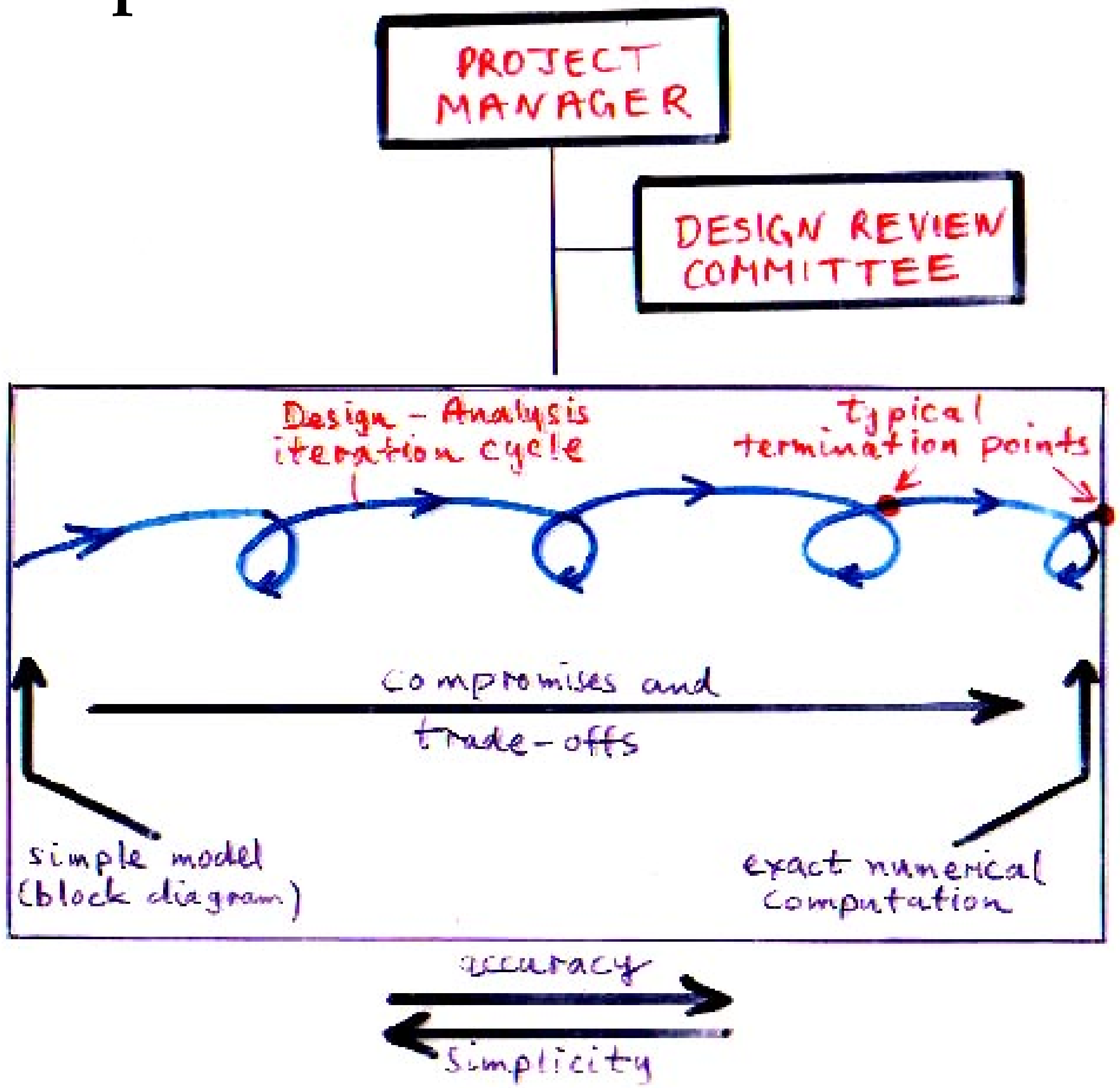


The Design Process

Design iteration loops



The design process consists of a succession of iteration loops:



"How to present the results" is important:

1. If you are a design engineer writing a report or appearing before a design review committee;

2. If you are a Test, Reliability, or System Integration Engineer dealing with someone else's design.

The D-OA approach is valuable for *all* these engineers.

Realization:

Design is the Reverse of Analysis,

Realization:

Design is the Reverse of Analysis, because:

The Starting Point of the Design Problem

(the Specification)

Realization:

Design is the Reverse of Analysis, because:

The Starting Point of the Design Problem (the Specification) is the Answer to the Analysis Problem

Conventional problem-solving approach:

1. Put everything into the model and simplify later.
2. Postpone approximation as long as possible, and don't even dare to make an approximation unless you can justify it on the spot.
3. The "answer" is acceptable in whatever form it emerges from the algebra.
4. The more work you do, the more valuable the result.
5. Every problem is a brand-new problem, and requires a brand-new strategy to solve it.

This is a recipe for failure!

Syndromes of Technical Disability:

Algebraic diarrhoea, which leads to

Algebraic paralysis

Fear of approximation

Syndromes of Technical Disability:

Algebraic diarrhoea, which leads to

Algebraic paralysis

Fear of approximation

The negative results of the conventional paradigm are often masked while the student is in school.

Why does the conventional approach fail?

Mathematicians tell us:

# of equations must = # of unknowns

Engineers face:

# of equations <<< # of unknowns

but have to solve the problem, anyway.

How can we overcome the negative results of the conventional approach?

1. Divide and Conquer:

It's easier to solve many simpler problems than one large one.

2. We must make the equations we do have *work harder* by expressing them in "Low Entropy" form.

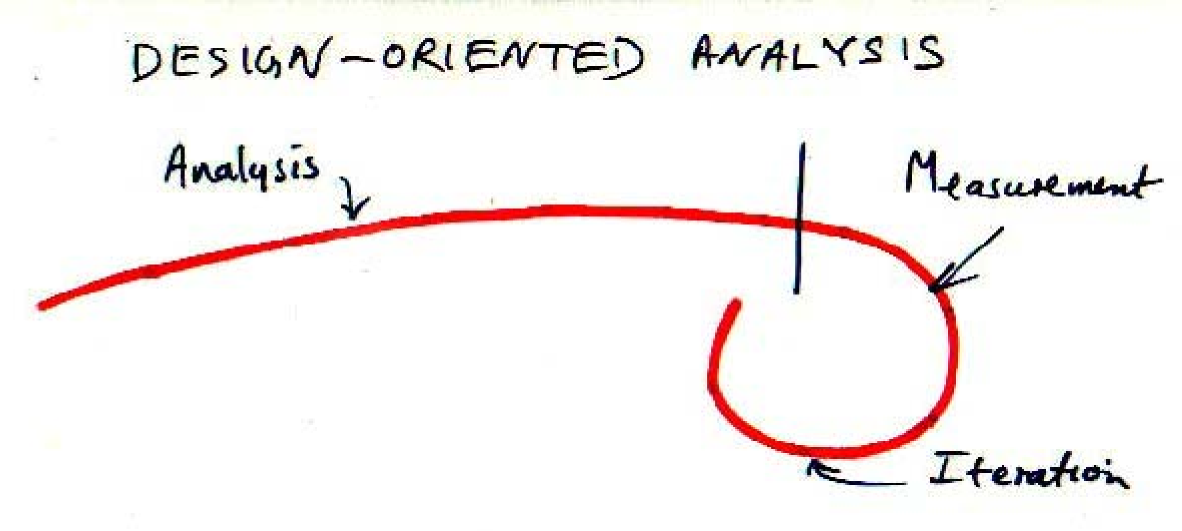
A *High Entropy Expression* is one in which the arrangement of terms and element symbols conveys no information other that obtained by substitution of numbers.

2. We must make the equations we do have *work harder* by expressing them in "Low Entropy" form.

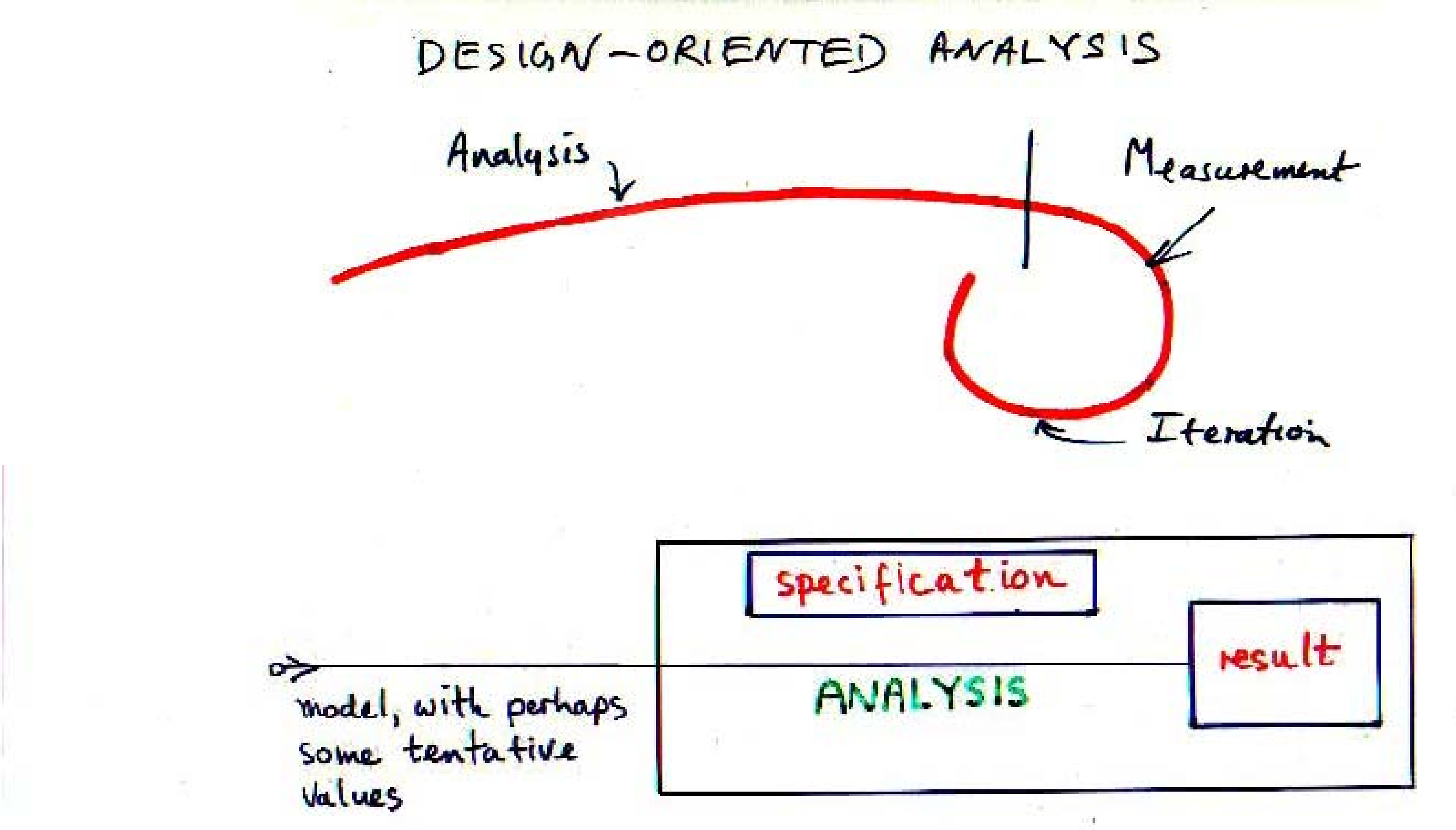
A *High Entropy Expression* is one in which the arrangement of terms and element symbols conveys no information other that obtained by substitution of numbers.

A *Low Entropy Expression* is one in which the terms and element symbols are ordered and grouped so that their physical origin and relative importance are apparent. Only in this way can one *change* the values in an informed manner in order to *change* the analysis answer (that is, to make it meet the Specification).

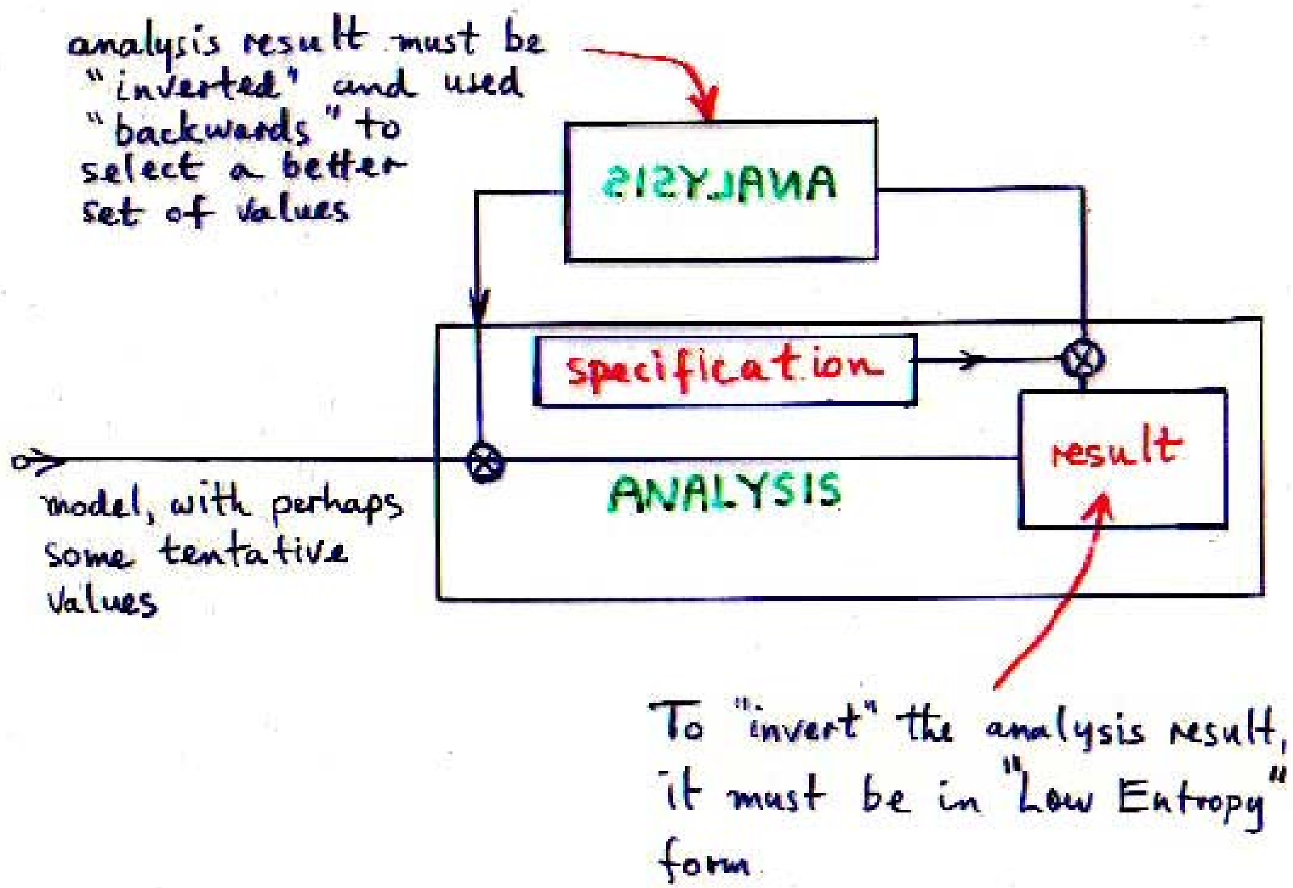
Low Entropy Expressions are essential in order to navigate the Design Iteration Loop:



Low Entropy Expressions are essential in order to navigate the Design Iteration Loop:



Low Entropy Expressions are essential in order to navigate the Design Iteration Loop:



Design-Oriented Analysis (D-OA) keeps the entropy low at every step along the way to a low entropy result.

Design-Oriented Analysis (D-OA) keeps the entropy low at every step along the way to a low entropy result.

The way to do this is:

Avoid solving simultaneous equations.

Design-Oriented Analysis (D-OA) keeps the entropy low at every step along the way to a low entropy result.

The way to do this is:
Avoid solving simultaneous equations.

Instead, follow the signal path from input to output by Thevenin/Norton reduction and/or voltage/current dividers.

Design-Oriented Analysis (D-OA) keeps the entropy low at every step along the way to a low entropy result.

The way to do this is:
Avoid solving simultaneous equations.

Instead, follow the signal path from input to output by Thevenin/Norton reduction and/or voltage/current dividers.

There may be many such paths (algorithms), each of which gives a different Low Entropy Expression.

Design-Oriented Analysis (D-OA) keeps the entropy low at every step along the way to a low entropy result.

The way to do this is:
Avoid solving simultaneous equations.

Instead, follow the signal path from input to output by Thevenin/Norton reduction and/or voltage/current dividers.

There may be many such paths (algorithms), each of which gives a different Low Entropy Expression.

Avoid multiplying out the series/parallel combinations.

How can we overcome the negative results of the conventional approach?

How can we overcome the negative results of the conventional approach?

3. Recognize that we *don't want* an exact answer: it would be too complicated to use, even if we could get it.

How can we overcome the negative results of the conventional approach?

3. Recognize that we *don't want* an exact answer: it would be too complicated to use, even if we could get it.

Therefore, subsititute for the missing equations with: *inequalities, approximations, assumptions,* and *tradeoffs*.

D-OA problem-solving approach: D-OA Rules

1. Put only enough into the model to get the answer you need.

2. Make all the approximations you can, as soon as you can, justified or not. Plow through the problem leaving behind you a wake of assumptions and approximations. ***You can't lose by trying*****.**

3. Figure out in advance as many of the quantities as you can that you want to have in the answer, and put them into the statement of the problem as soon as possible – ***even into the circuit model.***

4. The less work you do, the more valuable the result. ***You*** **control the algebra. You** ***make*** **the algebra come out in low entropy form by applying strategic mental energy before and during the math.**

5. Every problem in not unique. There are problem solving strategies that apply to almost all engineering problems.

This is a recipe for success!

The benefits of applying the D-OA Rules are:

1. You can *fend off* algebraic paralysis.

The benefits of applying the D-OA Rules are:

1. You can *fend off* algebraic paralysis.

2. Approximations are *good* things, not an admission of defeat.

The benefits of applying the D-OA Rules are:

1. You can *fend off* algebraic paralysis.

2. Approximations are *good* things, not an admission of defeat.

3. Algebra is *malleable*; you have *choices*.
You are *empowered* to exercise control: the math is your *slave*, not your master.

D-OA is the only kind of analysis worth doing!

TOPIC STRUCTURE

Getting Results:

Low Entropy Expressions
Ch 2

Presenting Results
Ch 3 Ch 4 Ch 5

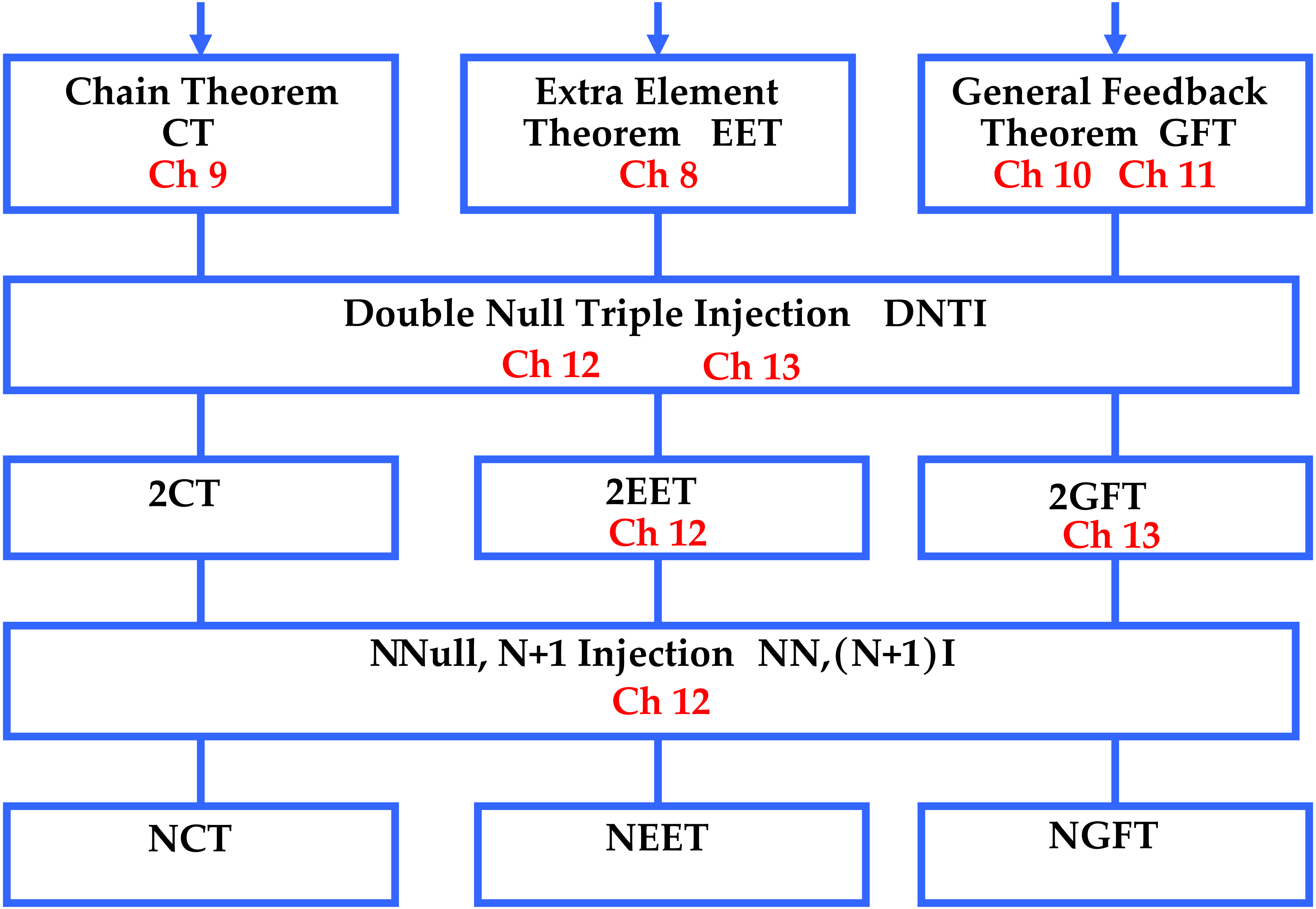
Combining Results
Ch 6

Extending Results:

Input/Output Impedance Theorem I/O IT
Ch 7

Null Double Injection NDI
Ch 8

Dissection Theorem DT
Ch 9



Examples: Chs 14 - 18