Session 3

Cushioning, Crowding and Quantization

Worksheet Functions
Circular References
Implicit Intersection

Last revised: July 6, 2011

Review of last time: Analysis and Synthesis

- Analysis and Synthesis
  - Decomposing and Recombining
  - Arrays often result — array manipulation required
- Techniques of array manipulation
  - Array arithmetic (+, -, *, /, ^, ...)
  - Matrix arithmetic (`MMULT`, `TRANSPOSE`)
  - Use Matrix arithmetic when you want to perform a synthesis
  - Use Array arithmetic between analysis and synthesis

Why conservative modeling?

- Modeling is inexact
  - The validity of a model is based on plausibility, not proof
  - Parameters of a model are either guessed or measured
    In either case, true values are rarely known exactly
  - The system you're modeling is a moving target
    What was once well-understood may evolve out from underneath your model
- Some sources of change (there are many more)
  - Competitive environment
  - Legal environment
  - Technological advances
  - Customer behavior and requirements
  - Your own organization
- For these reasons we often build safety cushions into models

Cushioning

Cushioning is the practice of including safety margins in models to mitigate the effects of approximations and change

- Even when you *can* find an optimum solution:
  - You might still want to a safety margin or cushion
  - You might not know the initial conditions exactly
- Examples
  - Cash reserves greater than the model predicts you need
  - Extra staff, stock or other resources during periods of high volume
  - Leasing more space than needed to allow for faster expansion

Crowding

Crowding is the opposite of cushioning —
you intentionally plan for resources smaller than needed

- Examples
  - Plan for doubling up in offices
  - Planned shortages in computer equipment
  - Planned shortages of staff — planned overtime
  - Planned wages or salaries below prevailing rates — planned turnover
- Usually, this is bad business
- But when you have to do it, it's good to know how

Quantization

Quantization occurs when resources are available in discrete units, or when attributes change discontinuously

- Sometimes vendors require orders in discrete units
  - No “odd lots”
  - Full trailer load, full roll of carpet, full cases
  - Also for labor: full week, full shift
  - Buildings, vehicles, people are not available in fractional units (except by time-share)
- Minimum quantity requirements
- To get what you need, you must sometimes acquire more (or less) than you need

Computational issues

- The difference between what you actually need and what you plan for is the *planned margin*
- Planned margins
  - Can be additive or multiplicative
  - Can be constant
  - Size can depend on base value, time, or some other quantity
- When the model describes temporal behavior
  - Planned margin can vary with time
  - Planned margin can depend on
    - This period
    - Previous, next, or other period
- When there is dependence on "this period," watch out for circular reference errors (more on this later)

Excel functions for cushions, crowding and quantization

- CEILING(*number*, *significance*)
  - Returns nearest integer multiple of *significance* on other side of *number* from 0
  - *number* and *significance* must have the same sign
- FLOOR(*number*, *significance*)
  - Returns nearest multiple of *significance* between 0 and *number*
  - *number* and *significance* must have the same sign
- INT(*number*)
  - Returns most positive integer closest to *number*
- ROUND(*number*, *numberOfDigits*)
  - Returns multiple of $10^{-numberOfDigits}$ that is closest to *number*
- TRUNC(*number*, *numberOfDigits*)
  - Returns multiple of $10^{-numberOfDigits}$ that is closest to *number*, between *number* and 0
- MROUND(*number*, *base*)
  - Returns *number* rounded to the nearest multiple of *base*
  - *number* and *base* must have the same sign

See On Line Help

◆ ExcelFunctions

Modeling techniques for cushions

- You're planning purchase of trucks for a utility company fleet. You want to make sure that you always have enough trucks ready to roll. If the number you need is N, you can pad N with:
  - A constant integer

    `N+Pad`
  - A computed integer

    `ROUND(N*(1+Pad),0)` gives an integer cushion such that result is a constant percentage greater than the computed value
  - A time-dependent pad

    `N+<reference>`

    Here we compute the actual value in `<reference>`

Circular reference errors

- Spreadsheets calculate the value in a cell by first computing values of cells it depends on
- If this leads back to the original cell, we say there is a "circular reference"
- Most spreadsheets cannot deal with circular references
  - Excel has an iterative recalculation mode
  - Generally, performance is so poor that you should avoid using it
- Circular reference errors are more frequent when the cushioning or crowding factor depend on the previous or next period

Unwinding circular references

- Simple case:
  - Y = 3 - X
  - X = 1 + Y
- These are simple, linear simultaneous equations
- The solution is X = 2, Y = 1
- If you implement X and Y as cells, Excel cannot resolve the solution because you get a circular reference error
- Best approach: solve the equations, then implement that solution

Unwinding circular references, continued

- In most common cases, dependencies are linear:

C_{OH}	Cash on hand
C_o	Cash before interest expense
D_N	New debt
D_{CP}	Cumulative prior debt
C_{MIN}	Cash minimum
i	Interest rate per period

If new debt is required:

$$C_{OH} = C_o - i(D_N + D_{CP}) + D_N = C_{MIN}$$

$$D_N = C_{MIN} - (C_o - i(D_N + D_{CP}))$$

If new debt is not required:

$$D_N = 0$$

$$C_{OH} = C_o - i(D_N + D_{CP})$$

To eliminate circularity, solve equations symbolically

Examples

- Ordering fleet vehicles with various planned margin strategies
  - Example1
- Simple circularity
  - Example2
- Diagram of a more complex circularity
  - Example3
- Minimum cash on hand with circularity
  - Example4

Implicit intersection

- Excel interprets non-array formulas that refer to ranges using a technique called “implicit intersection”
- If the formula is an array formula, the interpretation proceeds normally.
- Otherwise Excel retrieves the value of the leftmost (topmost) cell in the independent range that is in the same row (column) as the dependent cell
- If there is no such cell, a `#VALUE` error occurs

◆ ImplicitIntersection

Preview of next time: Temporal Response

- Excel functions for accessing array elements:
  `INDEX, OFFSET`
- References can be combined to produce new references using reference operators: Colon, Comma and Space
- Time evolution and temporal response model business processes
  - Convolution applies when system behavior is time-independent and additive
  - The Convolve macro implements convolution
  - Also implement convolution with tables

Special Note: Read the session notes for Session 4 to prepare