Axisymmetric Composite Panel with Insert: Reliability Analysis

There are two types of panel models, and four methods for calculating reliability. We're currently focusing on panels with a through the thickness insert, using models in 1D and 2D.

The COMSOL model files are located in the directories 'Insert1D' and 'Insert2D' respectively. Each input directory contains 4 files.

1. **getInputParms.m** and **runTrial.m** are generic wrapper files that find the input parameters and send them to the model.
2. **getSandwichParms.m** contains all of the stochastic parameters to be input to the model. Each parameter has 3 values: mean, standard deviation, and a distribution index. I've only implemented three distributions so far: 1. Normal, 2. Lognormal, and 3. Uniform. FORM and Line Sampling require that parameter distributions be transformed into the standard normal distribution, so adding additional distributions will require some more work.
3. **runSandwichInsert1D.m** and **runSandwichInsert2D.m** are the actual COMSOL model files that create the model, solve it, and calculate distance to the failure surface for each set of parameters. In the 'Insert1D' and 'Insert2D' directories, these files are set up as functions that take input parameters and return failure criteria. However, in the directory 'SingleRun', there are standalone versions that can be run individually to see the effect of changing different parameters. Note that the applied load is a global parameter, *g_Load*.

In the main directory, I've combined all of the different reliability methods into one file called **RunTest.m**. It takes in 4 parameters, *x*, *sCase*, *inParams*, and *bWarning*:

1. *x* is a vector of load cases, and can be of any length. The file will run an analysis to determine the probability of failure at each applied load in *x*.
2. *sCase* indicates the type of analysis and the type of model
    1. Monte Carlo – 1D Model
    2. Monte Carlo – 2D Model
    3. FORM – 1D Model
    4. FORM – 2D Model
    5. Line Sampling – 1D Model
    6. Line Sampling – 2D Model
    7. Subset Sampling – 1D Model
    8. Subset Sampling – 2D Model
3. *inParams* gives the input parameters to the method – these vary with the technique
    1. Monte Carlo
        1. *inParams.f* – Value at which failure occurs – for these simulations, *inParams.f* = 0.
        2. *inParams.n* – Number of samples – can be a vector with a different value for each load.
    2. FORM
        1. *inParams.f* – Same as Monte Carlo – critical value of performance function.
    3. Line Sampling
        1. *inParams.f* – Same
        2. *inParams.n* – Again, number of samples
    4. Subset Sampling – where the inputs are interesting:
        1. *inParams.f* – Same
        2. *inParams.nLevels* – Number of levels of conditional probability
        3. *inParams.nN* – Number of samples at each conditional level
        4. *inParams.nP* - Inverse of conditional probability
        5. *inParams.pWin* – Size, in standard deviations, of the search window for the Markov Monte Carlo chain
4. *bWarning* is a flag which determines whether the program outputs a line after each load case – default is false.

RunTest.m takes care of sorting the output files in the appropriate directory, 'Data_MC', 'Data_FORM', 'Data_LINE', and 'Data_SUB'. Each load case gets saved as a separate .mat file. There are some variations in some of the data that is saved, but 3 values will always appear in the data files:

1. *Pf* – The calculated probability of failure for the load
2. *NN* – The number of function evaluations
3. *t* – The total running time

EvalTest.m can be used to process the data. It takes two parameters, *x* and *sCase*. Again, *x* is the vector of loads, and *sCase* indicates the type of analysis and model, with the same numbering system as **RunTest.m**. It provides three outputs – *p, t,* and *n*. *p* is a vector of probabilities of failure, with one entry for each load case. *t* is a vector of simulation times, and *n* is a vector of simulation iterations.