

ABSTRACT

YUAN, XI. Sensitivity and Reliability Analysis of Pipe Components. (Under the direction of Vernon C. Matzen.)

In this dissertation, design of experiments (DOE) and finite element analysis (FEA) methods are employed in a sensitivity study on the B_2 stress index of ASME Boiler and Pressure Vessel (BPV) Code, Section III Equation 9 for elbow components. The effect on the index of loading condition, pipe size, pipe schedule, flange location, material type, temperature, pressure, bend angle and radius type is investigated using Latin Hypercube DOE method. Loading condition, pipe size, pipe schedule and material type are identified as the most influential factors from DOE study. An analytical equation is constructed from FEA results to describe the relationship between B_2 and those four factors. Also, the First Order Reliability Method (FORM), Monte Carlo sampling technique and finite element analysis are all employed to conduct an ASME Code calibration on the reliability of straight pipe components. The results are compared with those from a closed form solution.

**SENSITIVITY AND RELIABILITY ANALYSIS OF PIPE
COMPONENTS**

BY

XI YUAN

A DISSERTATION SUBMITTED TO THE GRADUATE FACULTY OF
NORTH CAROLINA STATE UNIVERSITY
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

CIVIL ENGINEERING

RALEIGH

2006

APPROVED BY:


VERNON C. MATZEN

Chair of Advisory Committee


ABHINAV GUPTA

Co-chair of Advisory Committee


KERRY S. HAVNER


JOHN W. BAUGH

To Those Who Love Me

BIOGRAPHY

Mr. Xi Yuan, graduated from Department of Mechanics, Beijing University with a Bachelor Degree in 1988 and a MS Degree in 1991. He joined Chinese State Bureau of Seismology and Chinese Academy of Sciences as a research engineer thereafter. He came to North Carolina State University to pursue a Ph.D. Degree in Civil Engineering in 1995. Mr. Yuan has been working as a research assistant at the Center for Nuclear Power Plant Structures, Equipment and Piping at NCSU and technical support and quality control engineer at Engineous Software, Inc.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to Professor Vernon C. Matzen for his inspiration and guidance throughout this research work, and to Professor Kerry S. Havner for his strong support during the most difficult time of program change. Also, I would like to extend my thanks to the other committee members, Professor Abhinav Gupta and Professor John Baugh, for their help and support.

The author wishes to express his sincere appreciation to his colleagues at Engineous Software, Inc. To Dr. Siu Tong, Mr. Mike Lemmons, Mr. Jeff Bond, Mr. JP Evans, Mr. Randy Elliott, Dr. Brett Wujek, Dr. Pat Koch, Mr. Judah Greenblatt, Dr. Charles Yuan and Dr. Mike Sheh.

This research was supported by the Center for Nuclear Power Plant Structures, Equipment and Piping and the Department of Civil Engineering. North Carolina Supercomputing Center provided high performance computational facility. The assistance of NCSC support staff, Dr. Robert H. Mallett of Mallett Technology, Inc are greatly appreciated.

Finally, I would like to thank my parents, my brother, my wife and two sons for their encouragement, financial support and help throughout my education.

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
NOTATION	x
1. INTRODUCTION	1
2.SENSITIVITY ANALYSIS OF B2 STRESS INDEX	5
2.1 Literature Review.....	5
2.2 Research Scope and Approach.....	8
2.3 Process Integration and Automation.....	10
2.4 Curvature Change of Out-of-Plane Bending or Torsion.....	15
2.5 Results and Discussion.....	23
2.6 Conclusion.....	26
3. RELIABILITY ANALYSIS OF STRAIGHT PIPE COMPONENTS ..	27
3.1 Background.....	27
3.2 Load and Resistance Factor Design Method.....	28
3.3 Saigal’s Closed Form Solution.....	30
3.4 Code Calibration Using Finite Element Method.....	34
3.5 iSIGHT Implementation.....	38
3.6 Code Calibration Results.....	38
4. SUMMARY AND FUTURE RESEARCH RECOMMENDATIONS .	43
5. REFERENCES.....	46

6. APPENDIX A.....	48
7. APPENDIX B.....	102
8. APPENDIX C.....	106

LIST OF TABLES

	Page
Table 2.4.1 Time Histories of Displacement, Curvature and Moment	21
Table 2.5.1 Preliminary DOE Study Result Summary	23
Table 2.5.2 Parameter Values for Full Factorial DOE Study	24
Table 3.4.1 Diameter and Thickness of Straight Pipe Components	37
Table 3.4.2 Types of Distributions, Mean Values and Coefficient of Variations	37

LIST OF FIGURES

	Page
Figure 2.1.1 Bend Angle for Elbow Component.....	6
Figure 2.1.2 Twice Elastic Slope Definition	7
Figure 2.3.1 DOE Sampling Techniques.....	14
Figure 2.4.1 Definition for Different Parts of Elbow Component	17
Figure 2.4.2 Global Coordinate System for Elbow Component (oblique view).....	18
Figure 2.4.3 Global Coordinate System for Elbow Component (plane view)	18
Figure 2.4.4 Rotation Calculation for Out-of-Plane Bending and Torsion	19
Figure 2.4.5 Typical Moment- $\Delta\alpha$ Curve under Out-of-Plane Bending Load	22
Figure 2.5.1 Comparison of NCSU Equation and Finite Element Result.....	25
Figure 3.2.1 Reliability Index for Standard Normal Distribution	30
Figure 3.4.1 Moment Curvature Curve for Bilinear Kinematic Hardening Material..	36
Figure 3.6.1 Reliability Index for Elastic-Perfectly Plastic Material under Service Level C.....	39
Figure 3.6.2 Reliability Index for Elastic-Perfectly Plastic Material under Service Level D.....	40
Figure 3.6.3 Reliability Index for Bilinear Kinematic Hardening Material under Service Level C.....	40

Figure 3.6.4 Reliability Index for Bilinear Kinematic Hardening Material under
Service Level D..... 41

NOTATION

- B_2 = primary stress index for bending
 c_1, c_2 = coefficients in an equation for B_2
 e_1, e_2 = exponents in an equation for B_2
 D_o = outside diameter of pipe
 h = characteristic bend parameter, tR/r_m^2
 $M_{CL_{elbow}}$ = Twice-elastic slope collapse moment for the elbow
 $M_{CL_{straight\ pipe}}$ = Twice-elastic slope collapse moment for the straight pipe
 P = internal pressure
 P_a = allowable working pressure, $P_a = \frac{2S_m t}{D_o - 2yt}$
 R = nominal bend radius of elbow
 r_m = mean pipe radius, $(D_o-t)/2$
 S_m = allowable design stress intensity value
 S_y = yield stress
 S_u = ultimate stress
 t = nominal wall thickness
 y = 0.4
 α = bend angle of elbow, in radians
 θ = rotation of a virtual line segment between two nodes
ANOVA = Analysis of Variance

Chapter 1

Introduction

ASME Boiler and Pressure Vessel Code Section III (the Code) [1] provides guidelines for the design of piping in nuclear power plants. Generally speaking, the design approaches in the current ASME Code fall into two categories: design by detailed analysis and design by rules along with simplified analysis.

Compared to the design by rules approach, the design by detailed analysis approach is more expensive since nonlinear analysis is usually required. It is not surprising, then, that the design by rules approach is widely used in nuclear power plants piping designs.

In the design by rules approach, semi-empirical design equations are developed using classic limit analysis and engineering experience. For the convenience of design equations development, the stresses that exist in piping system are divided into three categories: primary, secondary and peak. In the ASME Code, Primary stress is defined as any normal stress or shear stress developed by an imposed loading which is necessary to satisfy the laws of equilibrium of external and internal forces and moments. The basic characteristic of primary stress is that it is not self-limiting. Secondary stress is a normal stress or shear stress developed by the constraint of adjacent material or by self-constraint of the structure. The basic characteristic of a secondary stress is that it is self-limiting. Peak stress is that increment of stress that is additive to the primary plus secondary stress by reason of local discontinuities or local thermal stress, including stress concentrations. The basic characteristic of a peak stress is that it does not cause any noticeable distortion and is only a possible source of a fatigue crack or brittle fracture.

Three design equations, numbered (9), (10) and (11) in the Code, are developed for primary stress, primary plus secondary stress and peak stress, respectively. These design equations are employed in conjunction with linear elastic analysis to design Class I piping in nuclear power plants. Class 1 is the Reactor Coolant Loop and all its branches up to the second isolation valve on the branch. It has the highest safety requirements.

Of the three design equations, Equation (9) is the one that has attracted the most research interest since it is generally regarded as being overly conservative. It is given as follows:

$$B_1 \frac{PD_o}{2t} + B_2 \frac{D_o}{2I} M_i \leq 1.5S_m \quad (1.1.1)$$

for service level A loading. In this equation, B_1 and B_2 are stress indices for internal pressure and moment loads, respectively

In the ASME Code, loads are categorized into four different service levels: A, B, C and D. Service level A corresponds to operating conditions, and behavior is expected to remain elastic. Service level B corresponds to upset loading condition and the behavior may be slightly inelastic. However, the plant will keep operating following the loading event. Service level C corresponds to emergency loading. In this loading case, a plant would shut down following the event, but could be repaired. Service level D, the most severe, corresponds to faulted loading condition. This loading event would cause significant permanent damage, but no violation of the radiation barrier. A plant would not be expected to reopen following level D loading.

For B, C and D service levels, linear elastic analysis is not strictly appropriate. To accommodate the inelastic behavior in an approximate and, one hopes, a conservative manner while still allowing linear analysis, the right hand side of Eqn. (1.1.1) is allowed

to exceed the material's yield stress. Thus, the value of the right hand side of this equation becomes $1.8 S_m$, $2.25 S_m$ and $3.0 S_m$ for levels B, C and D, respectively. For carbon steel, S_m is $\min(2/3 S_y, 1/3 S_u)$. Assuming that the $2/3 S_y$ option governs, the right hand side becomes S_y , $1.2 S_y$, $1.5 S_y$ and $2 S_y$ for levels A-D, respectively.

If B_1 is equal to $1/2$, and B_2 is equal to 1, then Eqn. (1.1.1) is an approximation to the maximum Tresca stress in a thin straight pipe. To adapt this equation for piping components other than straight pipes, the Code uses modified values for B_1 and B_2 . For curved pipe or butt welding elbows, for example [Code, section NB-3683.7],

$$B_2 = \frac{1.30}{h^{2/3}} \geq 1.0 \tag{1.1.2}$$

$$B_1 = -0.1 + 0.4h \text{ but not } < 0 \text{ or greater than } 0.5$$

where h is the characteristic bend parameter defined as tR/r_m^2 , R is the nominal bend radius of elbow and r_m is the mean pipe radius. See the Notation Section for a complete list of symbols and their definitions.

This definition of the B_2 stress index came from Rodabaugh and George's early work [2] on the stress intensification factor. According to research work by Touboul et al. [3], this B_2 definition is said to be overly conservative in some cases but unsafe in others.

An alternative definition of the B_2 stress index introduced by Matzen and Tan in 2002 [6] takes the form:

$$B_2 = \frac{M_{CL_{straight\ pipe}}}{M_{CL_{elbow}}} \tag{1.1.3}$$

The background for this definition is described briefly in the next section. See [6] for a detailed history of the B_2 stress index.

In Chapter 2 of this dissertation, an investigation of the B_2 stress index for elbow components will be presented. In this research, the effects of nine parameters on the B_2 stress index were first evaluated using a sensitivity analysis method. Then a more comprehensive study was conducted on a few of the most influential parameters as identified from the sensitivity analysis. In both studies, the B_2 stress indices were calculated using finite element analysis with Eqn. (1.1.3). These studies resulted in sets of B_2 values vs. the elbow parameter “ h ” for various bend angles and internal pressures. A curve fitting technique was then used to obtain an “optimal” B_2 equation that can be evaluated without recourse to finite element analysis.

In order to achieve safe and affordable pipe designs, a thorough understanding of the safety factors of ASME Code design equation is requisite. Currently, those safety factors are based on simplified analysis, experimental data and engineering experience. A more rational way to evaluate the safety factors is reliability-based design methods. These methods formally address uncertainty and randomness using probabilistic analyses.

As a first step in applying reliability methods to piping design, finite element analysis and the Advanced First Order Reliability Analysis Method (AFORM) will be applied to a series of straight pipe components in Chapter 3 of this dissertation. The results will be compared with those from a closed-form solution. The extension to elbow behavior, while necessary for a full investigation of Equation (1.1.1), is beyond the scope of this dissertation.

Chapter 2

Sensitivity Analysis of B_2 stress index

2.1 Literature Review

In 1957, Rodabaugh and George [2] performed a theoretical study on the effect of internal pressure on the stress intensification factor i , which played a role in earlier Codes comparable to B_2 in the current Code. For smaller values of h , i.e. for thinner elbows, they found the effect of pressure to be significant; but for larger h , i.e. for thicker elbows, it was negligible. They proposed using the following approximate equation to consider the effect of internal pressure:

$$i_p = \frac{i}{1 + \frac{S}{E} X_i} \geq 1 \quad (2.1.1)$$

where $i = \frac{0.9}{h^{2/3}} \geq 1$, $S = Pr_m/t$, E is Young's modulus, and $X_i = 3.25 \left(\frac{r_m}{t} \right)^{3/2} \left(\frac{R}{r_m} \right)^{2/3}$

In 1988, Touboul et al. [3] presented an appraisal of various experimental results. Their purpose was to determine the influence of material, h , bend angle α , and pressure on the B_2 stress index. They started with an equation proposed by Dodge and Moore [4]:

$$B_2(P) = \frac{B_2(0)}{1 + f(h)g(P)} \quad (2.1.2)$$

where $g(P) = \sigma_p / S_y$, $f(h) = 0.7h^{-1}$ and $B_2(0)$ is B_2 stress index with zero internal pressure, which is $1.21h^{-2/3}$ [3]. σ_p is the Von Mises stress corresponding to maximum allowable pressure.

After making an empirical correction using their experimental results, Touboul et al. ended up with the following equation:

$$B_2 = \frac{1.6h^{-2/3}(\alpha/\pi)^{0.4}}{1+0.7\sigma_p/(hS_y)} \quad (2.1.3)$$

where the bend angle α is as defined in Figure 2.1.1.

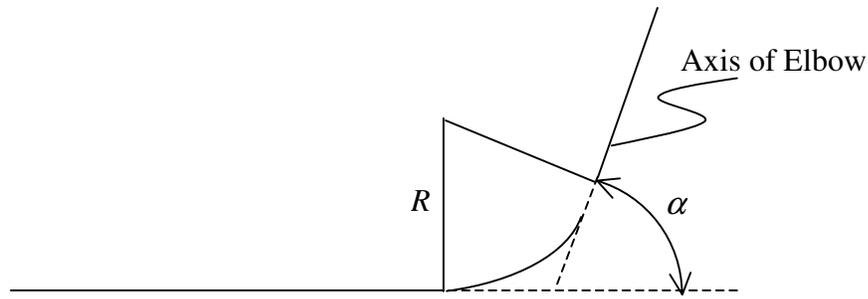


Fig 2.1.1 Bend Angle for Elbow Component

Yu [5] conducted a finite element study on the B_2 index for different values of h .

The equation used for determining B_2 was:

$$B_2 = \frac{S_y Z}{M_{CL}} \quad (2.1.4)$$

where M_{CL} is the collapse moment of the elbow and Z is the elastic section modulus.

While applying this equation to straight pipes, they found that B_2 could be less than 1.0.

As mentioned above, Matzen and Tan [6] described another procedure in 2002 for calculating the B_2 stress index, which took the form:

$$B_2 = \frac{M_{CL_{straight\ pipe}}}{M_{CL_{elbow}}} \quad (2.1.5)$$

where the collapse moments were obtained using nonlinear finite element analysis and were based on what is called the twice-elastic slope definition of collapse [Code, NB-3213.25]. The definition can be described graphically as follows: Consider the moment-curvature graph in Fig 2.1.2. Straight line 1 has the same slope as the initial slope of moment-curvature curve. Straight line 2 has a slope such that $\tan \phi = 2 \tan \theta$ relative to the ordinate. The twice-elastic slope method defines the moment value of the intersection point of line 2 and the moment-curvature curve as collapse moment M_{CL} . For many values of h , this M_{CL} is a reasonable approximation to the ultimate moment of elbows. [13]

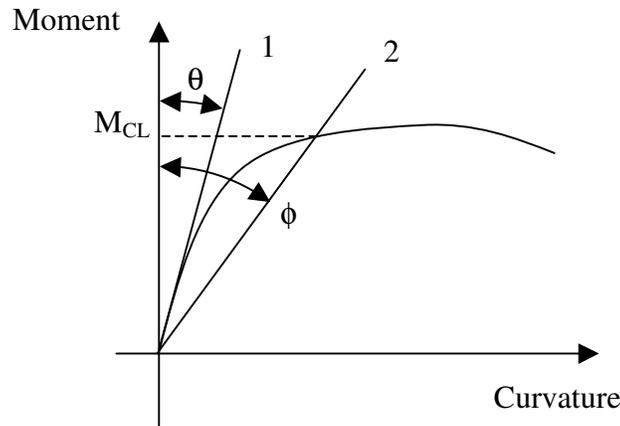


Fig 2.1.2 Twice Elastic Slope Definition

This definition of the B_2 stress index ensures that, for a straight pipe, B_2 is always 1.0, and, furthermore, it can be shown that the safety margins for an elbow component and a straight pipe segment with the same material and geometric properties are identical

[6]. In their study, Matzen and Tan considered only 90°, long radius, stainless steel 304L elbows at room temperature under in-plane bending.

In this research, the effects of various parameters on the B_2 index of elbow components are evaluated using the Matzen and Tan definition.

2.2 Research Scope and Approach

After reviewing the existing work, nine parameters were determined to affect the value of the B_2 stress index. They are:

1. Loading Type: in-plane bending (both closing and opening modes), out-of-plane bending, and torsion (twisting)
2. Size: pipe outer diameter D_o
3. Schedule: pipe wall thickness index. For the same pipe size, larger schedule represents larger wall thickness. See Appendix C for a sample pipe table.
4. Flange location: length of the straight pipe between the elbow component and the flange.
5. Material
6. Temperature: material properties are scaled up or down to reflect change in working temperature
7. Pressure: internal pressure normalized by the ASME Code allowed pressure
8. Elbow bend angle
9. Radius type: ratio of elbow radius and pipe nominal radius R/r_m . Long radius, $R/r_m=1.5$, short radius, $R/r_m=1.0$

There are, of course, many sizes, schedules, etc. We limited the selection of the parameters to the following commonly used values:

Loading \in (In-Plane Closing, In-Plane Opening, Out-of-Plane Bending, Torsion)

$D_o \in$ (2", 4", 6", 8")

Schedule \in (5, 10, 40, 80, 160)

Flange length as a multiple of $D_o \in$ (0, 1, 5) and the possible combinations for both

sides of elbow \in ((0,0), (0,1), (0,5), (1,1), (1,5), (5,5))

Material \in (Carbon Steel, Stainless Steel, Elastic-Perfectly Plastic)

Temperature \in (70°F, 400°F, 800°F)

Normalized Pressure $P/P_a \in$ (0, 0.618, 1) (where P_a is the ASME Code allowed pressure of the elbow component. See Notation for definition)

Bend Angle \in (30°, 90°, 150°)

Radius type \in (long radius, short radius)

The purpose of this study is to evaluate the effects of these nine parameters on the B_2 stress index and to develop an equation relating B_2 to the most significant of these parameters. To maximize the information from the study, ideally, all possible combinations of the nine parameters would need to be exhausted. The total number of combinations = $4 \times 4 \times 5 \times 6 \times 3 \times 3 \times 3 \times 3 \times 2 = 77,760$ and each combination requires two finite element analysis runs: one for the elbow and one for the straight pipe. This leads to more than 150,000 nonlinear finite element analysis runs and each run may take hours to execute. A rational and efficient way to reduce this number further is the formal Design of Experiments (DOE) method.

Experimental design in this context dates back to the work of Fisher in 1930s at the Rothmasted Agricultural Experimental Station in UK. A variety of DOE methods developed since that time have been applied to various disciplines, e.g. engineering,

physical and biological sciences [7]. DOE methods allow an investigator to find out what happens to the output or response (B_2 in this case) when the settings of the input variables (the nine parameters) are systematically changed.

Instead of exhausting all 77,760 possible combinations, the following two steps were taken: First, a preliminary DOE study was performed on all nine parameters. Post-processing (sensitivity analysis) the DOE results led to the elimination of the less influential parameters. Then, a more comprehensive study was performed on the remaining (most influential) parameters. An optimal equation for B_2 as a function of those most influential parameters was constructed in this second study. These two steps will be discussed in detail in next sections.

2.3 Process Integration and Automation

According to the Matzen and Tan procedure [6], the B_2 calculation requires the collapse moments of the elbow component and the straight pipe component of the same geometric and material properties. Thus each DOE calculation requires two finite element analysis runs, one for the straight pipe and one for the elbow. To avoid mistakes while creating input files, and to carry out all the finite element analysis runs more efficiently, the DOE process was automated in the generic software shell iSIGHT [13]. iSIGHT is widely used for design process automation and optimization. Most commercial finite element analysis codes can be integrated with iSIGHT and the process can be optimized by the built-in DOE, Optimization and Reliability based design algorithms.

The following are detailed steps for a typical DOE study:

1. Modeling: both elbow and straight pipe components were modeled in ANSYS [11] using element Shell 43 since this element proved suitable for quasi-static nonlinear analysis [5], [6]. The finite element model and the B_2 calculation procedure were similar to those used in [6] except for the definition of curvature change. Material model was based on a small strain assumption. Although this is not strictly appropriate, the FEA model was proved to match experiment results very well [16], [17]. When the component was under torsion or out-of-plane bending loads, the definition of curvature used in [6] was no longer valid and no existing definition was available. Thus, a new definition of curvature needed to be introduced. A detailed discussion is given in the next section.

2. Process Integration: after one elbow and one straight pipe were modeled and executed once, the input and output files were used as templates since the input and output files of all other components were similar. The differences among those files were the values of the selected parameters. Therefore, these template files were linked with the iSIGHT file parser to generate input files and read output files automatically. A set of instructions was created to guide the iSIGHT file parser to find where to write input parameters (component geometry, loading etc.) and where to read output parameters (moments and curvatures). The simulation code name and path to its executable were also specified. At run time, the whole process (write input file, run simulation code, and read output file) would be repeated without manual interruption. The detailed implementation can be found in the Appendix A.

3. Material Scaling: In the finite element analysis runs, the material properties may also vary from one run to another. A problem arises here that warrants some

discussion. Both ANSYS and ABAQUS [12] require an entire monotonic stress strain curve (true stress and true strain for large deformation purpose), but the material properties provided by the ASME code are only the 0.2% offset yield stress and the ultimate stress. The procedure to use these two parameters, along with the modulus of elasticity and the strain at the ultimate stress, to generate complete stress strain curves described in [6] is used in this work. This procedure is described as below:

Step 1: Establish a baseline engineering stress-strain curve from an ASTM test (NCSU material test data are used herein)

Step 2: At each point (i) of the baseline curve, subtract the elastic strain from the total strain to get the engineering stress-plastic strain curve.

$$e_{PL_i} = e_i - \frac{S_{i0}}{E} \quad (2.3.1)$$

where S_{i0} is the stress value of the point on the baseline curve and E is Young's modulus.

Step 3: Calculate the following two scaling factors at 0.2% yield stress and at the ultimate stress (where the corresponding plastic strain was taken to be 35%, which was the approximate value in all of the NCSU tests) using the following equations:

$$f_y = \frac{S_y}{S_{y0}} \quad \text{and} \quad f_u = \frac{S_u}{S_{u0}} \quad (2.3.2)$$

where S_y and S_u are the 0.2% yield stress and the ultimate stress given by the ASME Code for current component and S_{y0} and S_{u0} are the yield stress and the ultimate stress for baseline measured curve, respectively.

Step 4: Determine the coefficients a and b from following equations:

$$f_i = ae_{PL_i} + b \quad (2.3.3)$$

To evaluate a and b , use the two baseline points corresponding to the yield stress and the ultimate stress. Solving for a and b leads to the following:

$$a = \frac{f_u - f_y}{e_{uPL} - e_{yPL}} = \frac{f_u - f_y}{0.35 - 0.002} \text{ and } b = f_y - ae_{yPL} \quad (2.3.4)$$

Step 5: Scale the baseline stresses to obtain the scaled stresses for the current component.

$$S_i = S_{i0}f_i \quad (2.3.5)$$

Step 6: Add elastic strain to the plastic strain to get the total strain.

$$e_i = e_{PL_i} + \frac{S_i}{E} \quad (2.3.6)$$

Step 7: Convert the modified engineering stress-engineering strain curve to the true stress-true strain curve.

4. Problem Definition: Let x_1, x_2, \dots, x_9 stand for the nine parameters and assume that B_2 is a generic function of x_1, x_2, \dots, x_9 :

$$B_2 = f(x_1, x_2, \dots, x_9) \quad (2.3.7)$$

Although B_2 can be determined using Matzen and Tan's procedure [6] on any given set of the parameters, the functional form of $f()$ is not known. However, DOE

studies can provide insights into an unknown function by analyzing output changes corresponding to systematic input changes. As mentioned in the last section, the first DOE study was to identify most influential parameters. The Latin Hypercubes technique was employed to make this determination because of its efficiency and ability to take into account nonlinear effects.

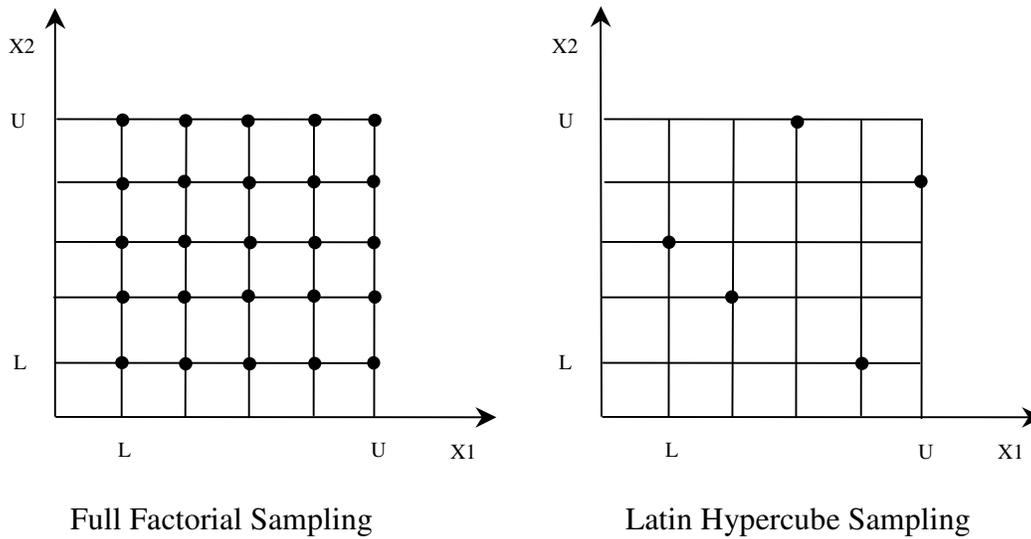


Fig 2.3.1 DOE Sampling Techniques

Figure 2.3.1 demonstrates the difference between Full Factorial sampling technique and Latin Hypercube sampling technique. Full Factorial exhausts all possible combinations of all parameter levels between lower value (L) and upper value (U). Latin Hypercube includes all possible levels of each parameter between L and U and randomly combines those levels.

To capture the trend (main effects) of all parameters while minimizing the total number of finite element analyses, only first order, second order, and interaction

effects were considered. This assumption is similar to approximating $f()$ as a full quadratic polynomial function, i.e.

$$B_2 \approx a_0 + a_1x_1 + a_2x_2 + \dots + a_9x_9 + a_{10}x_1^2 + a_{11}x_2^2 + \dots + a_{18}x_9^2 + a_{19}x_1x_2 + a_{20}x_1x_3 + \dots + a_{54}x_8x_9 \quad (2.3.8)$$

where a_0, a_1, \dots, a_{54} are coefficients of the quadratic polynomial function.

Thus the minimum number of experiments needed to determine a_0, a_1, \dots, a_{54} is 55.

In the actual study, 60 experiments were conducted to provide some redundancy.

5. Input File Generation: since iSIGHT and finite element analysis codes are not available on the same computer system, the integrated application could not be run directly. An internal application programming interface (API) command was employed to generate all 60 input files for the finite element analysis code.

6. Run FEA code in batch mode: all FEA runs were combined into one batch job and sent to a supercomputer batch queue.

7. Read Results: after all FEA simulation runs were completed, a post-processing code was run to calculate B_2 values using the ratio of the collapse moments as shown in Eqn. 2.1.5.

8. Database Scan: after all B_2 values were copied and pasted into the iSIGHT database file, a database scan could enable all iSIGHT post-process functionalities for DOE. All sixty data sets were fed into a least square fitting process to calculate the coefficients a_0, a_1, \dots, a_{54} from (2.3.8). The sensitivity coefficient of each parameter (a_1, a_2, \dots, a_9) was recorded for review.

2.4 Curvature Change of Out-of-Plane Bending or Torsion

In Matzen and Tan's B_2 study [6], only in-plane bending was considered. It is useful to review their approach to curvature calculations. The relative rotation ($\Delta\alpha$) is the difference of the rotations of the two end cross sections of the elbow component. $\Delta\alpha$ is not the curvature, but, for a straight pipe, $\Delta\alpha$ is related to the curvature through the equation:

$$\kappa = \frac{1}{\rho} = \frac{\Delta\alpha}{L} \quad (2.4.1)$$

where κ is the curvature, ρ is the radius of curvature and L is the length of the straight pipe.

For an elbow subjected to in-plane bending, L is thought of as the length of the pipe-elbow center line running from the outside end of one extension to the outside end of the other. However, since L is a constant, and only moment values are of interests, M vs. $\Delta\alpha$ plots can be used instead of M vs. κ plots to calculate the collapse moment. To calculate the relative rotation of the elbow ends, Matzen and Tan started with two vectors, each one defined by points on the extrados and intrados (points 1 and 2 shown in Figure 2.4.1 for one end of an elbow). As the elbow deformed under load, they calculated the change in rotation about the transverse axis (Z in Figure 2.4.2).

For out-of-plane bending and torsion, similar definitions of curvature are used.

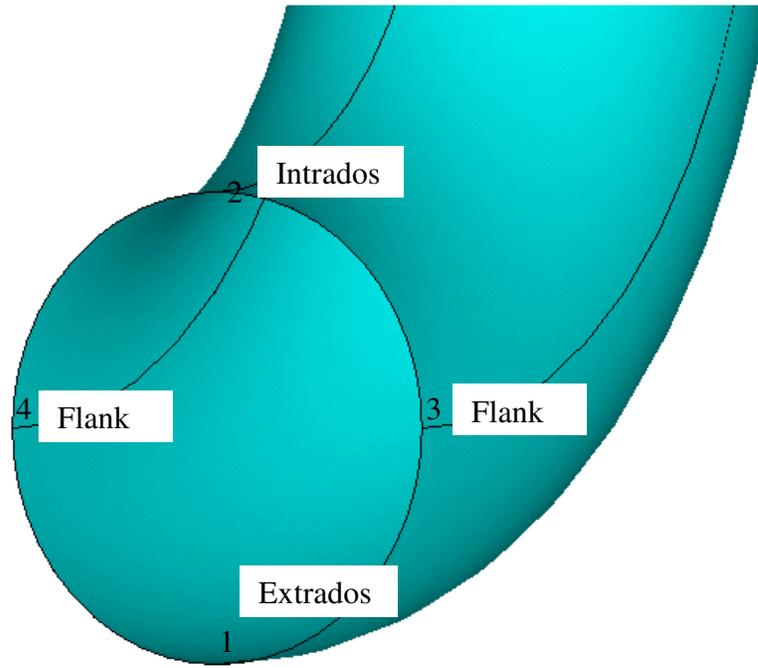


Fig. 2.4.1 Definition for Different Parts of Elbow Component

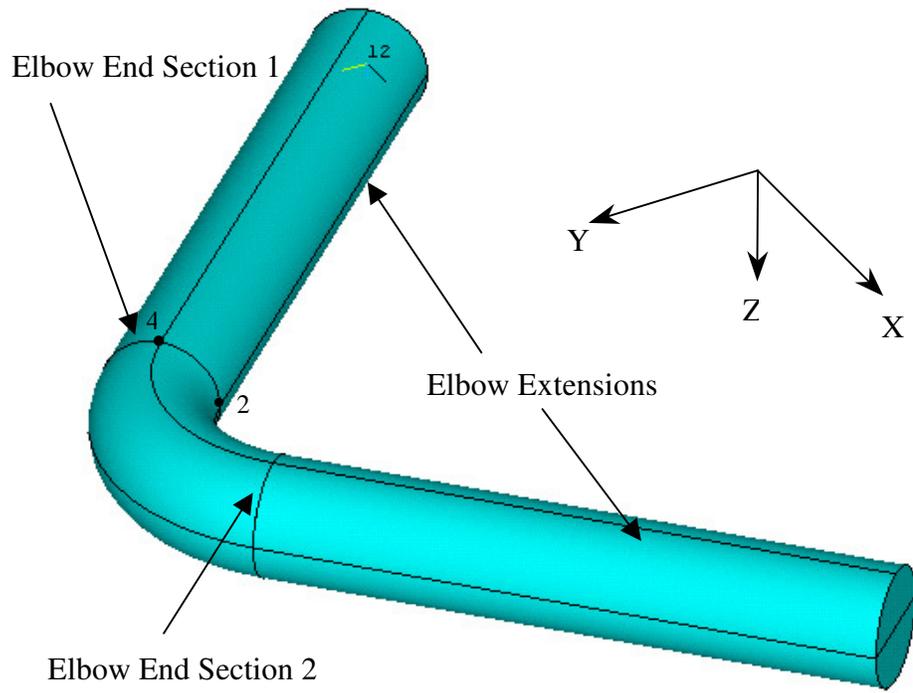


Fig. 2.4.2 Global Coordinate System for Elbow Component (oblique View)

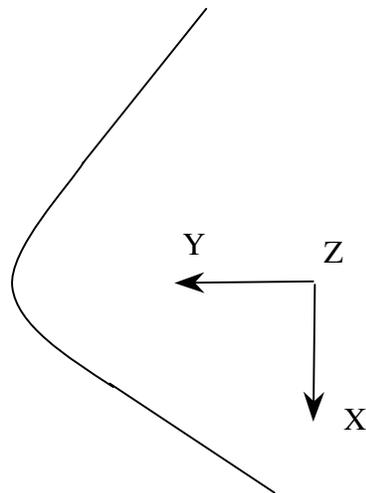


Fig. 2.4.3 Global Coordinate System for Elbow Component (plane view)

Figure 2.4.2 and Figure 2.4.3 show the coordinate system that is used in the out-of-plane bending and torsion cases. Point 1 and point 3 are on the opposite sides of point 2 and point 4, thus are not visible in this figure. In Figure 2.4.4, point 3' and 4' are positions of point 3 and 4 after deformation. To simplify the calculation, line 4'-3' is translated to line 4''-3'' so that point 4 and 4'' are coincident since translation of the line won't change its direction and thus won't affect the curvature calculation. Angle α is defined as the rotation of line 4-3 with respect to the Y-axis and β as rotation of line 4-3 with respect to the X-axis.

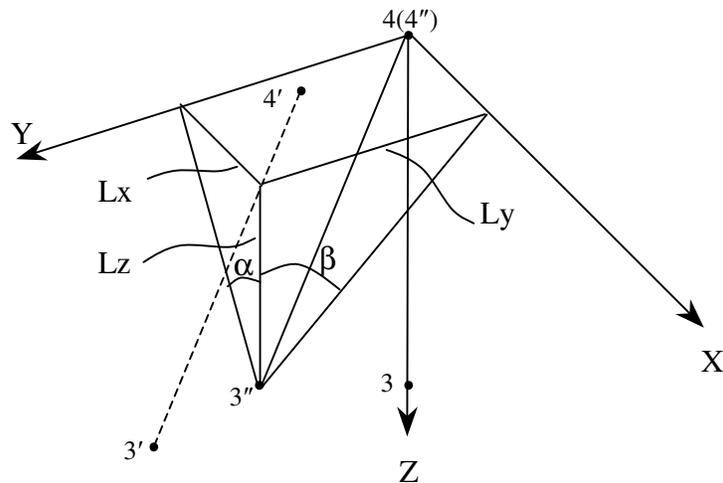


Fig. 2.4.4 Rotation Calculation for Out-of-Plane Bending and Torsion

Consider first the out-of-plane bending case, and assume that the coordinates of point 3 are (x_3, y_3, z_3) . After deformation, they become $(x_3+u_3, y_3+v_3, z_3+w_3)$. Similarly point 4,

(x_4, y_4, z_4) , becomes $(x_4+u_4, y_4+v_4, z_4+w_4)$. Herein, u_3, v_3, w_3 and u_4, v_4, w_4 are the three components of displacements of points 3, 4.

$$Lx = x_3 + u_3 - x_4 - u_4 = (x_3 - x_4) + (u_3 - u_4) \quad (2.4.2)$$

$$Lz = z_3 + w_3 - z_4 - w_4 = (z_3 - z_4) + (w_3 - w_4) \quad (2.4.3)$$

Because of the orientation of the coordinate system relative to the elbow, $x_3 = x_4$ and $z_3 - z_4 = D_{norm}$, where D_{norm} is the nominal diameter of elbow component. Displacements u_3, u_4, w_3, w_4 are obtained from outputs of finite element analysis. Thus,

$$Lx = u_3 - u_4 \quad (2.4.4)$$

$$Lz = D_{norm} + (w_3 - w_4) \quad (2.4.5)$$

and

$$\alpha = \arctan\left(\frac{Lx}{Lz}\right) \quad (2.4.6)$$

The same method can be applied to the other end plate of the elbow component, and these two angles can be used to evaluate $\Delta\alpha$.

For the torsion case, the equations are similar except the rotation calculated is about the X -axis instead of the Y -axis.

$$Ly = v_3 - v_4 \quad (2.4.7)$$

$$Lz = D_{norm} + (w_3 - w_4) \quad (2.4.8)$$

$$\beta = \arctan\left(\frac{Ly}{Lz}\right) \quad (2.4.9)$$

The best way to illustrate the procedures/calculations in curvature change is by way of an example. The example used in this illustration is described as follows:

Component: elbow

Size: 6 inch

Schedule: 40

Loading: Out-of-Plane Bending

Constraint: Fixed at one end

Material Model: Multi-linear isotropic hardening, large strain not considered

Execution of the finite element analysis for the component described above resulted in 200 sets of data. For brevity, however, only the last 6 load steps are shown here in Table 2.4.1.

Table 2.4.1 Time Histories of Displacement, Curvature and Moment

TIME	10 UZ	366 UZ	10 UX	366 UX
195	-0.0798738	-0.0794411	-0.0576864	0.0594579
196	-0.0799983	-0.0795584	-0.0579294	0.059727
197	-0.0801224	-0.0796749	-0.0581716	0.0599967
198	-0.0802458	-0.0797903	-0.0584135	0.0602658
199	-0.0803705	-0.079907	-0.0586557	0.0605362
200	-0.0804939	-0.0800224	-0.0588976	0.060806

Table 2.4.1 (cont'd) Time Histories of Displacement, Curvature and Moment

673 UZ	809 UZ	673 UX	809 UX	$\Delta\alpha$	Moment
-0.566072	-0.575534	-0.207292	0.165596	0.040329641	142269
-0.568674	-0.578237	-0.208527	0.166379	0.040567341	142372
-0.571276	-0.580941	-0.209763	0.167161	0.040805082	142475
-0.573874	-0.583642	-0.211	0.167942	0.041042974	142577
-0.576475	-0.586346	-0.212238	0.168722	0.041280612	142679
-0.579075	-0.589049	-0.213478	0.1695	0.041518391	142779

Nodes 366 and 10 are referred to as Points 3 and 4 on end section 1

Nodes 809 and 673 are referred to as Points 3 and 4 on end section 2

According to the previous derivation, for Node 366 and 10 at time step 200:

$$Lz = D_{norm} + (w_3 - w_4) \approx 6.345 + (-0.0800224 + 0.0804939) \approx 6.3454715 \quad (2.4.9)$$

$$Lx = u_3 - u_4 \approx 0.060806 + 0.0588976 \approx 0.1197036 \quad (2.4.10)$$

$$\alpha_1 = \arctan\left(\frac{Lx}{Lz}\right) \approx 0.018862177 \quad (2.4.11)$$

Similarly, for Node 809 and 673:

$$Lz = D_{norm} + (w_3 - w_4) \approx 6.345 + (-0.589049 + 0.579075) \approx 6.335026 \quad (2.4.12)$$

$$Lx = u_3 - u_4 \approx 0.1695 + 0.213478 \approx 0.382978 \quad (2.4.13)$$

$$\alpha_2 = \arctan\left(\frac{Lx}{Lz}\right) \approx 0.060380567 \quad (2.4.14)$$

Therefore, the angle change is:

$$|\Delta\alpha| = |\alpha_2 - \alpha_1| \approx 0.041518391 \quad (2.4.15)$$

Repeat this process for all sets of data and plot the moment- $\Delta\alpha$ to get the graph shown in Figure 2.4.5.

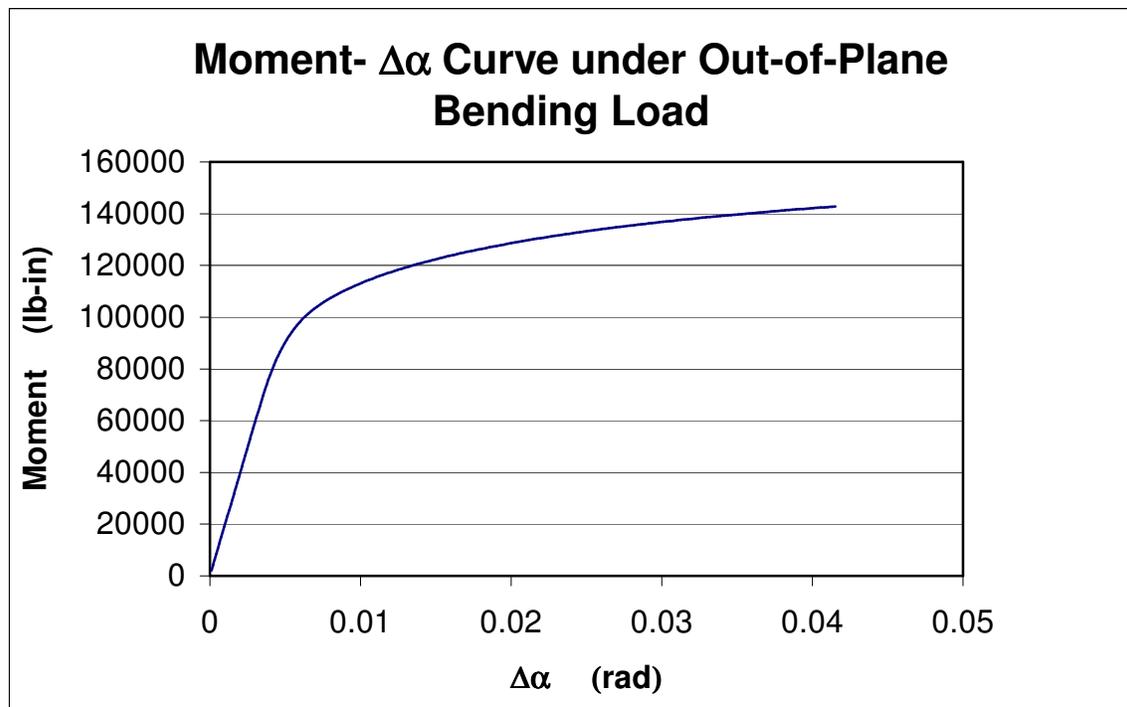


Fig. 2.4.5 Typical Moment- $\Delta\alpha$ Curve under Out-of-Plane Bending Load

2.5 Results and Discussion

The first DOE study was used to get a preliminary idea of the significance of the various parameters in the B_2 calculations. After the sensitivity analysis was conducted during the DOE post-process (discussed in Section 2.3), the sensitivity coefficient of each parameter was normalized by the sum of all coefficients and converted into percentage format for comparison purpose. The results are listed in Table 2.5.1.

Pressure and Bend angle were defined as continuous parameters and thus their sensitivity coefficients were not presented here because of a technical limitation of the iSIGHT post-processor. Since these two parameters were not included in this evaluation, they were not eliminated and so left in for the next DOE study.

Table 2.5.1 Preliminary DOE Study Result Summary

Loading Type	Schedule	Flange Location	Size	Material	Radius Type	Temperature
28%	23%	20%	13%	9%	4%	3%

As the table indicates, the effects of Temperature and Radius Type are relatively insignificant in the B_2 calculations, while Loading Type, Schedule and Flange Location were among the most influential parameters. Also, because we found that, among the four Loading Types, the in-plane closing and the out-of-plane bending always controlled, only those two loadings were included in the remaining DOE studies. Flange location, in spite of its significance on the final results, was not considered as a parameter at the next level since the most commonly used location (one in which the flange is at least 5 diameters

away from the end of the elbow) is also the least conservative (i.e. it allows the most ovalization of the elbow). Thus a tangent length of $5 \times D_o$ was used for both sides of the elbow in all studies. Furthermore, instead of using both diameter and thickness as parameters, the characteristic pipe bend parameter h was used (this is the most common size parameter used in elbow studies).

The second DOE study used the following set of parameters: Loading Type, h , Pressure, Bend Angle, Pressure, and Material. Since we had fewer factors now, the full factorial DOE technique was employed since it is the most comprehensive technique in iSIGHT. The parameter values are listed in Table 2.5.2.

Table 2.5.2 Parameter Values for Full Factorial DOE Study

Loading Type	In-Plane Closing Bending, Out-of-Plane Bending
Size and Schedule (h)	6" Sch 40, 6" Sch 10, 4" Sch 80, 8" Sch 5, 2" Sch 160
Bend Angle:	30°, 90°, 150°
Pressure:	0, 0.618, 1 (Normalized by Design Pressure)
Material:	Carbon Steel, Stainless Steel (Low & High Strength), Elastic-Perfectly Plastic

For each of these combinations, two nonlinear finite element analyses were performed to obtain the collapse moments and hence a B_2 index (one for an elbow and one for a corresponding straight pipe segment). The B_2 indices were then imported into a Microsoft Excel spreadsheet. For a given bend angle, material and set of internal pressures, these B_2 values were plotted vs. the characteristic elbow parameter h . As explained in Ref. [14] we use the EXCEL curve fitting command SOLVER to obtain a best-fit curve to the FEA data (see Appendix B for details of EXCEL worksheet). The equation has a form similar to Equation (2.1.3), but with a default value of one when the bend angle is zero, i.e. the component is a straight pipe. All the coefficients in the equation were determined by minimizing the sum of the squares of the differences

between FEA data and predictions from the equation. More reconciliation results can be found in Appendix B.

The resulting equation, also referenced as the NCSU equation, took the following form:

$$B_2 = 1 + \frac{c_1}{h^{e_1}} \frac{\left(\frac{\alpha}{\pi}\right)^{e_2}}{\left(1 + \frac{c_2 \text{Pr}}{\lambda t S_y}\right)} = 1 + \frac{0.36}{h^{1.1}} \frac{\left(\frac{\alpha}{\pi}\right)^{0.63}}{\left(1 + \frac{0.6 \text{Pr}}{\lambda t S_y}\right)} \quad (2.5.1)$$

The comparison between predictions from this equation and data points from finite element analysis for the case of $\alpha = 30^\circ$ is shown in Figure 2.5.1.

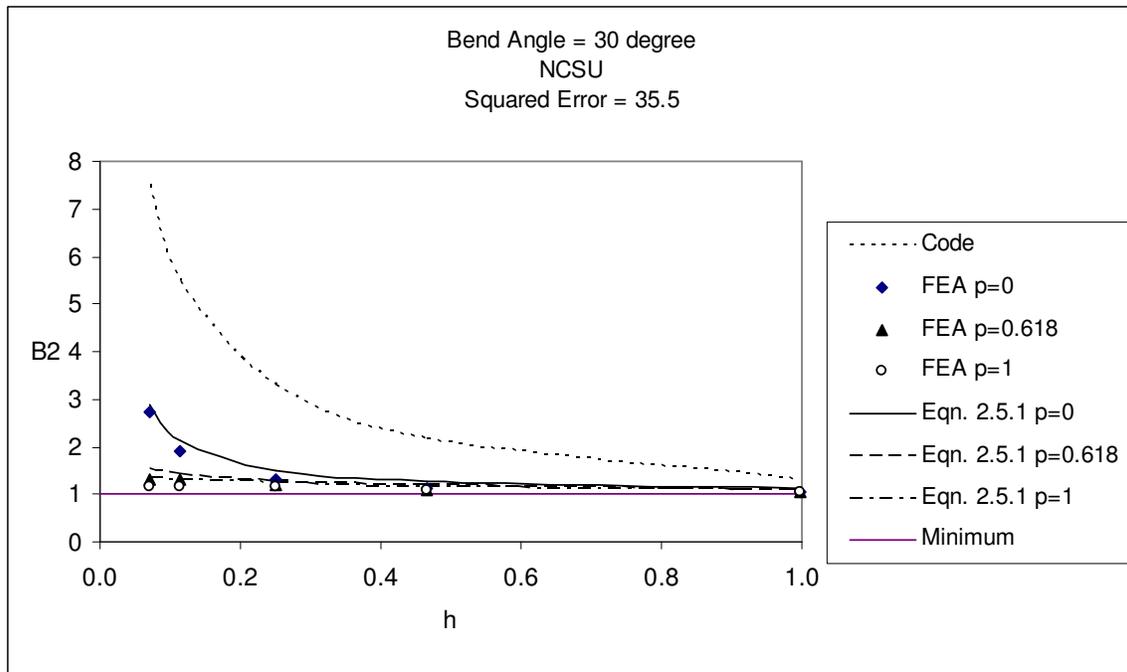


Fig. 2.5.1 Comparison of NCSU Equation and Finite Element Result

2.6 Conclusion

Both the NCSU equation (Eq. 2.5.1) and optimized Touboul's equation (Eq. 2.1.3) matched the finite element results for all different elbow components quite well (360 different combinations of elbow size and schedule, material, internal pressure, and bend angle). Since the NCSU equation has the attribute of guaranteeing that the index never falls below 1, we concluded that it is the more appropriate equation for calculating the B_2 stress index.

Chapter 3

Reliability Analysis of Straight Pipe Components

3.1 Background

The current ASME Boiler and Pressure Vessel Code Section III (the Code) was developed based on a deterministic methodology. In this methodology, a variable is defined deterministically (i.e. associated with a single value). Let f stand for allowable load and c stand for capacity. A design equation can be simply stated as:

$$c > f \quad (3.1.1)$$

In the real world, uncertainties abound. Man-made products may have defects from workmanship. Errors can also be introduced in measurements. In the deterministic methodology, a so-called worst-case scenario approach is employed to address the concern for failures. A safety factor or a safety margin is used to assure the safety of a design. Let m stand for safety margin. Eq. (3.1.1) now becomes:

$$c - m > f \quad (3.1.2)$$

Since uncertainties are almost always unavoidable, not only in the capacity but also in the load, safety margins can be inconsistent; designs based on deterministic methodology may be overly conservative in some cases but unsafe in others.

A better way to handle uncertainties is to incorporate reliability-based design methodologies such as Load and Resistance Factor Design (LRFD) into the design process. Research shows that LRFD can yield designs with consistent reliability levels [8], [9].

To migrate from a deterministic methodology to one based on reliability, the first step would be to assess the reliabilities of current design equations. This process is called “Code Calibration”.

After introducing some basics of LRFD in the next section, Section 3.3 will focus on an approach to code calibration which is based on a closed-form solution. Section 3.4 uses an approach based on the finite element method. The chapter ends with a comparison between the closed-form method and the finite element method.

3.2 Load and Resistance Factor Design Method

An LRFD based design equation is similar to a deterministic design equation. Let F and C be random variables associated with the load and capacity, respectively. The LRFD based equation can be defined as:

$$C > F \quad (3.2.1)$$

Because the random variables are not single valued variables but are associated with probability distributions, the measure of safety can be defined as:

$$R = P(C > F) \quad (3.2.2)$$

where $P()$ denotes the probability of an event and R is called reliability.

A performance function can be defined as:

$$g = C - F \quad (3.2.3)$$

The function $g()$ can be treated as a failure criterion or safety boundary. That is,

$$g() > 0 \quad \text{safe state} \quad (3.2.4a)$$

$$g() = 0 \quad \text{limit state} \quad (3.2.4b)$$

$$g() < 0 \quad \text{failure state} \quad (3.2.4c)$$

For a more general case, let X_i be the random variables (loads, capacities etc.). The reliability R is then given by the integral of joint probability distribution of X_i 's over the safe region (where $g(\cdot) > 0$)

$$R = \int_{g(\cdot) > 0} \dots \int f_{X_1 X_2 \dots X_n}(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (3.2.5)$$

where $f_{X_1 X_2 \dots X_n}(x_1, x_2, \dots, x_n)$ is the joint probability density function of the random variables X_i 's.

Usually, the joint density function is unknown, as is its integral. Alternatively an approximation method, the Advanced First-Order Reliability Method (AFORM), can be used. In AFORM, the performance function $g(\cdot)$ is expanded into a Taylor series about the most probable failure point (the point on the failure surface with the shortest distance to the origin). Only linear terms are kept to provide a first order approximation.

The reliability is then expressed in terms of the reliability index β such that $R = \Phi(\beta)$, where Φ is the cumulative distribution function of the standard normal distribution. See Figure 3.2.1 for the definition of β in the one-dimensional case,

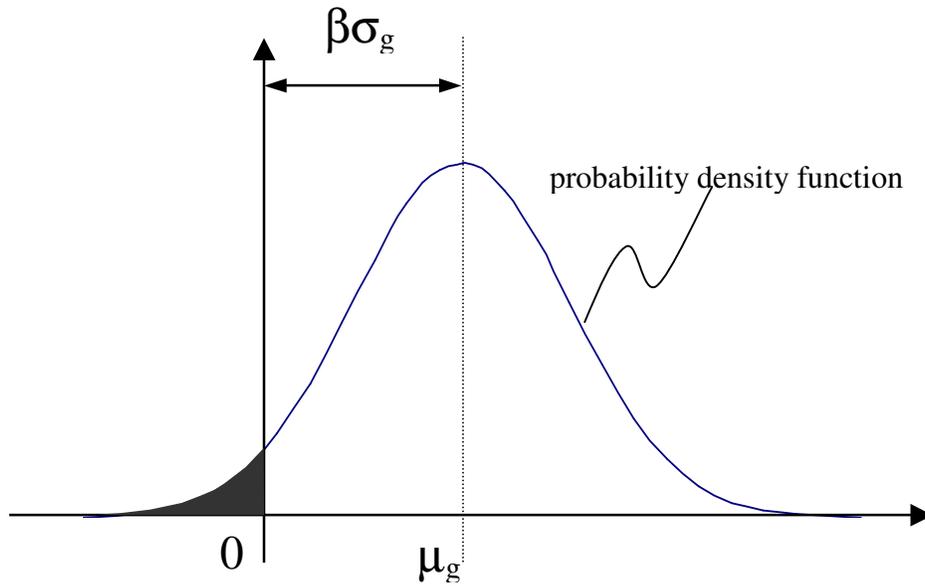


Fig 3.2.1 Reliability Index for Standard Normal Distribution

where μ_g is the mean value of performance function $g()$ and σ_g is the standard deviation of $g()$.

3.3 Saigal's Closed Form Solution [10]

In 2005, Saigal carried out an investigation on reliabilities of straight pipe components for the failure modes associated with the different service levels, assuming that ASME Boiler and Pressure Vessel Code Section III equation 9 is employed as the design equation.

In his work, the onset of yielding criterion was used for service level A. The performance function for onset of yielding is:

$$g_1(X) = X_{S_y} - B_1 \frac{X_P X_D}{2X_t} - B_2 \frac{X_M}{X_{Z_e}} \quad (3.3.1)$$

where X_P is the random variable representing internal pressure P , X_D is the random variable representing pipe diameter D , X_{Z_e} is the random variable representing elastic section modulus Z_e , X_t is the random variable representing pipe thickness t , X_{S_y} is the random variable representing yield strength, and X_M is the random variable representing bending moment M .

For service level B, formation of a single plastic hinge was used as the failure criterion. The performance function for the formation of the plastic hinge at a particular pipe cross-section for elastic-perfectly plastic material is given as:

$$g_2(X) = X_{S_y} - B_1 \frac{X_P X_D}{2X_t} - B_2 \frac{X_M}{X_{Z_p}} \quad (3.3.2)$$

The only difference between the previous equation and this one is that X_{Z_p} , the random variable representing plastic section modulus Z_p of the pipe cross-section, is used instead of X_{Z_e} .

Saigal studied three different failure criteria for service levels C and D in accordance with recommendation of ASME Working Group on Piping Design:

1. Plastic Instability

The component is considered to be in the failure state when the outermost fiber reaches the ultimate stress. Here the performance function becomes:

$$g_3(X) = X_{S_u} - B_1 \frac{X_P X_D}{2X_t} - B_2 \frac{X_M}{X_{Z_e}} \quad (3.3.3)$$

where X_{S_u} is the random variable representing ultimate strength. All other variables are the same as Eqn. 3.2.1.

2. Twice-the-elastic-slope

The component fails when the twice-the-elastic-slope line crosses the moment-curvature curve in an elastic-plastic analysis. See Figure 2.1.2 in Section 2.1 for a detailed explanation.

3. Fully plastic cross-section

When an elastic-perfectly plastic material is considered, the component fails if every point on a cross section reaches the yield stress.

For criteria 2 and 3, the performance function is formulated in terms of M_{CL} and M_{code} . M_{CL} is the collapse moment corresponding to either twice-the-elastic-slope moment or the moment when every point on a cross section reaches yield. M_{code} is the maximum code allowed moment from Code Eq. 9. It can be derived as in Eqn. 3.3.4 when internal pressure is 0.

$$M_{code} = k S_m Z_e \quad (3.3.4)$$

and k is a coefficient defined as:

$$k = 1.5 \quad \text{for service level A} \quad (3.3.5a)$$

$$k = 1.8 \quad \text{for service level B} \quad (3.3.5b)$$

$$k = 2.25 \quad \text{for service level C} \quad (3.3.5c)$$

$$k = 3.0 \quad \text{for service level D} \quad (3.3.5d)$$

Then the performance function is defined as:

$$g(X) = M_{CL} - M_{code} \quad (3.3.6)$$

Since evaluation of M_{CL} using twice-the-elastic-slope definition will require many inelastic finite element analysis executions, closed-form expressions of moment-curvature relation were developed by Saigal for straight pipes in the cases of elastic-perfectly plastic and bilinear kinematic hardening materials.

For elastic-perfectly material, his M_{CL} takes following form:

$$M = 4 t R^2 S_y \left[\left(\frac{\kappa R}{2 \varepsilon_{yld}} \right) \left(\alpha - \frac{\sin(2\alpha)}{2} \right) + \cos(\alpha) \right] \quad (3.3.7)$$

where

$$\alpha = \sin^{-1} \left(\frac{\varepsilon_{yld}}{\kappa R} \right) \quad (3.3.8)$$

and κ is curvature of the straight pipe, t is the pipe thickness, R is mean radius of the pipe cross section, S_y and ε_{yld} are yield stress and strain of the pipe material, respectively

When every point on a cross section reaches yield stress, the plastic moment takes following form:

$$M_p = Z_p S_y \quad (3.3.9)$$

where Z_p is the plastic modulus of the pipe cross-section

For bilinear kinematic hardening materials, the moment-curvature relation is defined as follows:

$$M = 4t R^2 S_y \left[\begin{array}{l} \left(\frac{\kappa R}{2\varepsilon_{yld}} \right) \left(\alpha - \frac{\sin(2\alpha)}{2} \right) + (1 - \lambda) \cos(\alpha) + \\ \left(\frac{\lambda \kappa R}{\varepsilon_{yld}} \right) \left(\frac{\pi}{4} - \frac{\alpha}{2} + \frac{\sin(2\alpha)}{2} \right) \end{array} \right] \quad (3.3.10)$$

where λ is the strain hardening parameter. All other variables are defined the same way as in Eqn. (3.3.7).

One limitation of these closed-form solutions is that the ovalization of cross sections was not considered, and ovalization can be significant in some cases.

3.4 Code Calibration Using Finite Element Method

As mentioned in the last section, the closed-form solution developed by Saigal cannot take into account geometric non-linearities such as ovalization. In the research described here, a series of finite element analyses was conducted using the commercial code ABAQUS to assess the effect of these nonlinearities on code calibration.

Since the purpose of finite element analysis was to consider geometric nonlinearity, the ABAQUS Elbow 31 element was employed to model straight pipe components. This element provides an additional degree of freedom to describe ovalization. From previous numerical study [15], this element has been shown to accommodate both geometry and material nonlinearities well, and it is much less expensive to use than the shell element models.

It is good practice to evaluate the performance of the candidate element and material models before applying them to the formal reliability analysis which involves hundreds

or thousands of finite element executions. A straight pipe with 8.5” outside diameter and 0.109” wall thickness was chosen as the component in this performance evaluation. Three different finite element models were built: one using ANSYS Shell 43, one using ABAQUS Elbow 31 with the ovalization DOF turned on, and one using ABAQUS Elbow 31 with ovalization turned off.

Figure 3.4.1 shows the comparison of the moment-curvature curves generated using the three modeling approaches and Saigal’s closed form solution using Eqn. 3.3.10. No large strain effect was considered. Results from the ANSYS shell element model and ABAQUS elbow 31 with ovalization DOF on are virtually indistinguishable. These two curves are below the one for the closed-form solutions as expected since allowing ovalization makes the pipe more flexible than the closed form equation which assumes no ovalization. However, the difference between closed-form solution and ABAQUS elbow 31 with ovalization DOF off is as large as the one between closed-form solution and ABAQUS elbow 31 with ovalization DOF on. This result is somewhat unexpected since the closed form solution and the elbow element with ovalization turned off both neglect the softening effect of ovalization. However, there are many simplifying assumptions in the closed form equation and these may account for the unexpectedly large difference in results.

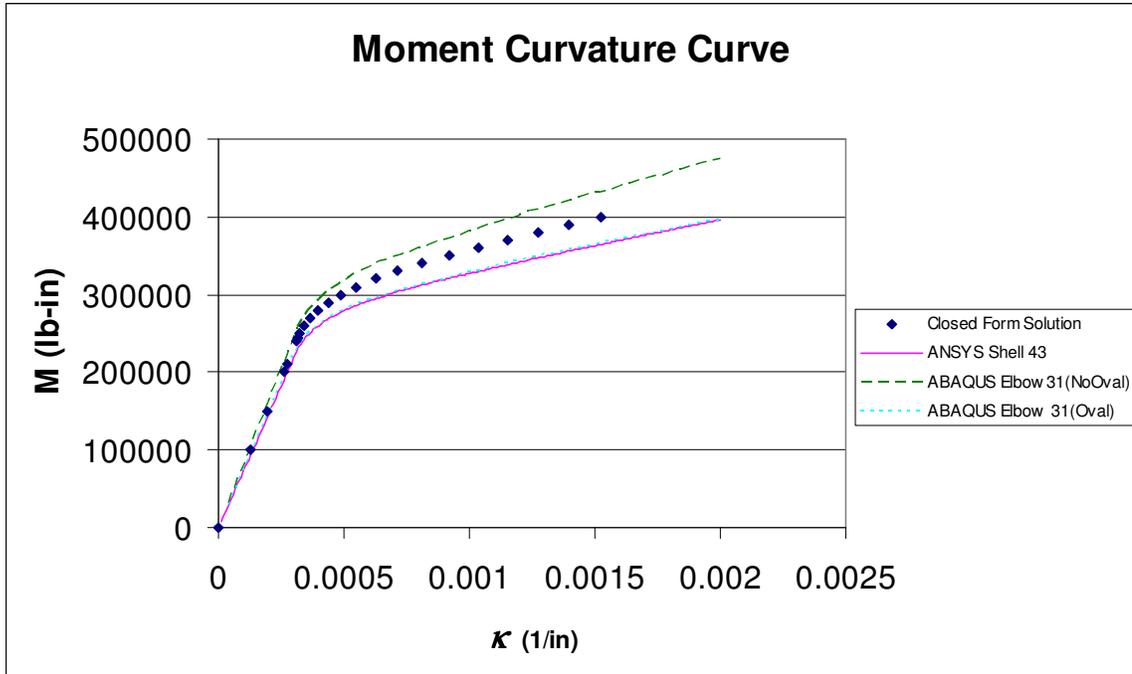


Fig. 3.4.1 Moment Curvature Curve for Bilinear Kinematic Hardening Material

Based on the modeling study described above, the ABAQUS Elbow 31 element with ovalization on was selected for use in the code calibration study described in the remaining part of this chapter.

The same pipe components and statistical properties as those used in Saigal's study were used in finite element analyses. The pipe components are described in Table 3.4.1 and the statistical properties are given in Table 3.4.2.

Table 3.4.1: Diameter and thickness of straight pipe components

Component No.	Diameter	Thickness	D/t
	D (in)	t (in)	
1	12	0.375	32.0
2	12	0.5	24.0
3	10	0.365	27.4
4	10	0.5	20.0
5	8	0.322	24.8
6	8	0.5	16.0
7	6	0.28	21.4
8	6	0.432	13.9
9	5	0.258	19.4
10	5	0.375	13.3

Table 3.4.2: Types of distributions, mean values and coefficient of variations

Variable	Distribution Type	Mean Value (μ)	Coefficient of Variation (Ω)
Su	lognormal	75	0.1
Sy	lognormal	45	0.1
D	lognormal	Different values	0.0625
t	lognormal	Different values	0.0625

3.5 iSIGHT Implementation

The Monte Carlo sampling technique was used to provide data for the reliability analysis. An experiment was conducted to check convergence of Monte Carlo sampling. Five hundred points were identified as a reasonable size for Monte Carlo sampling. This means that for each straight pipe component listed in Table 3.4.1, five hundred straight pipe models were derived by varying random variables X_D , X_t , X_{S_y} , X_{S_u} , and each derived component required one finite element code execution. To make this tedious process more affordable and error free, the iSIGHT integration and automation process used in the Chapter 2 of this dissertation was employed. This process was similar to the one used there for ANSYS, and implementation details are given in Appendix A.

Once the process was integrated into iSIGHT, a Monte Carlo Simulation was defined in the iSIGHT task plan. All five hundred finite element analysis executions were automatically carried out by the iSIGHT task manager. Each ABAQUS execution was controlled by iSIGHT. After the five hundred executions were completed, a reliability analysis using AFORM (Advanced First-Order Reliability Method) was conducted to determine the reliability index β .

This same process was repeated for all 10 straight pipe components and for both elastic-perfectly plastic and bilinear kinematic hardening materials. The results from finite element analysis and closed-form solutions are compared in the next section.

3.6 Code Calibration Results

Figure 3.6.1 - 3.6.4 show the reliability index results comparisons between finite element analysis and closed-form solutions for elastic-perfectly plastic and bilinear

kinematic hardening materials at service level C and D. In Figure 3.6.1 - 3.6.4, “ M_u ” and “TwiceElastic” denote different failure criteria. “ M_u ” stands for “The ultimate moment” and is defined as the maximum moment during a monotonic loading process.

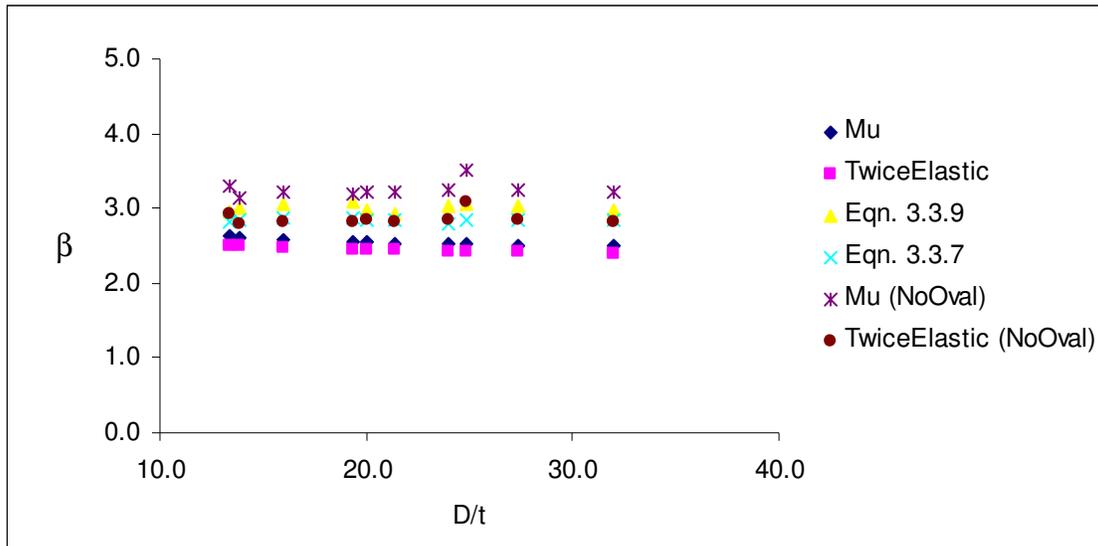


Fig. 3.6.1 Reliability Index for Elastic-Perfectly Plastic Material under Service Level C

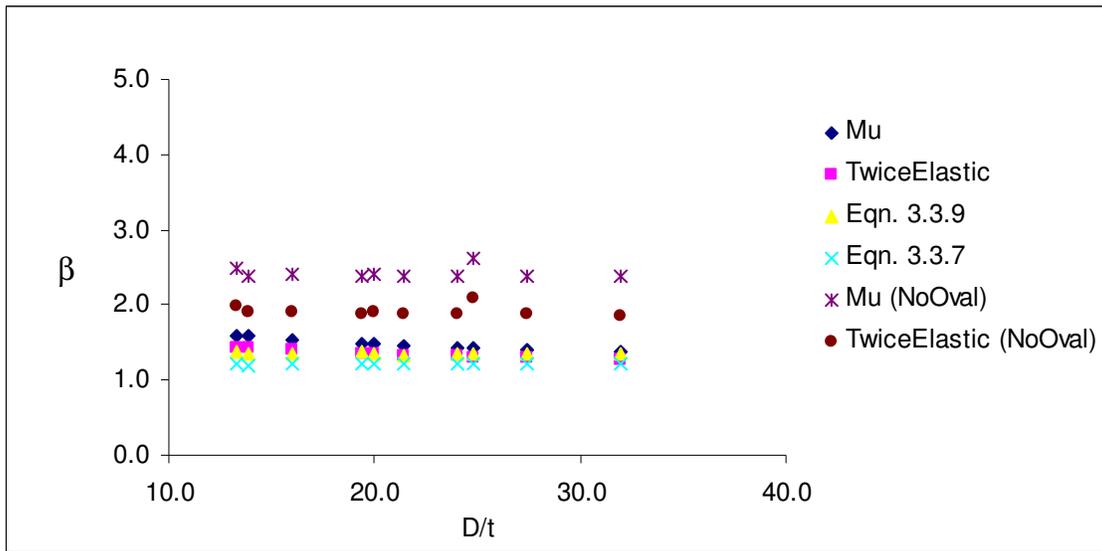


Fig. 3.6.2 Reliability Index for Elastic-Perfectly Plastic Material under Service Level D

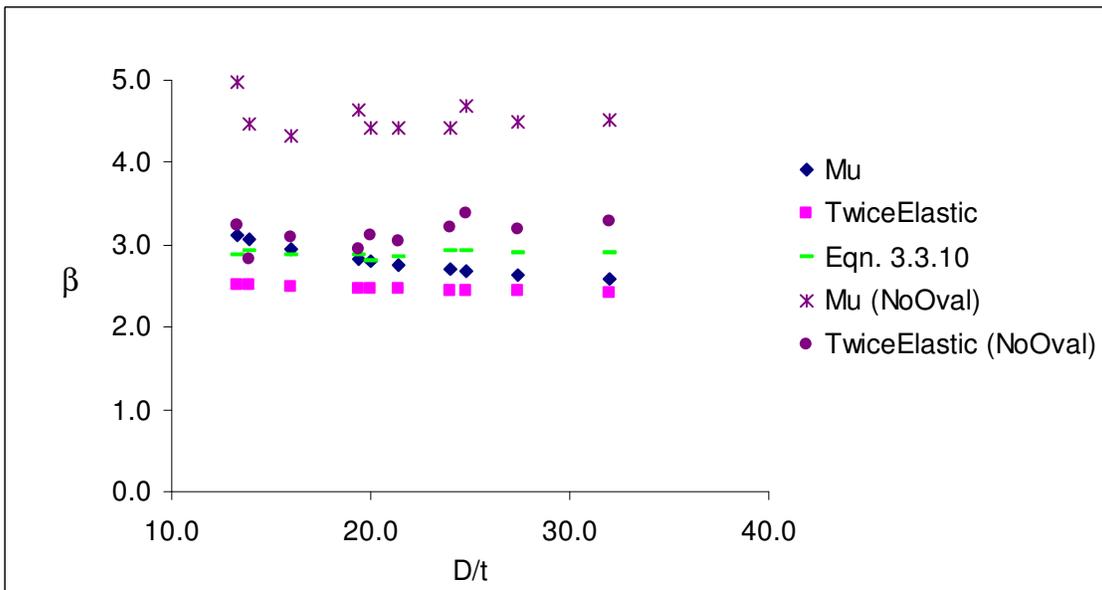


Fig. 3.6.3 Reliability Index for Bilinear Kinematic Hardening Material under Service Level C

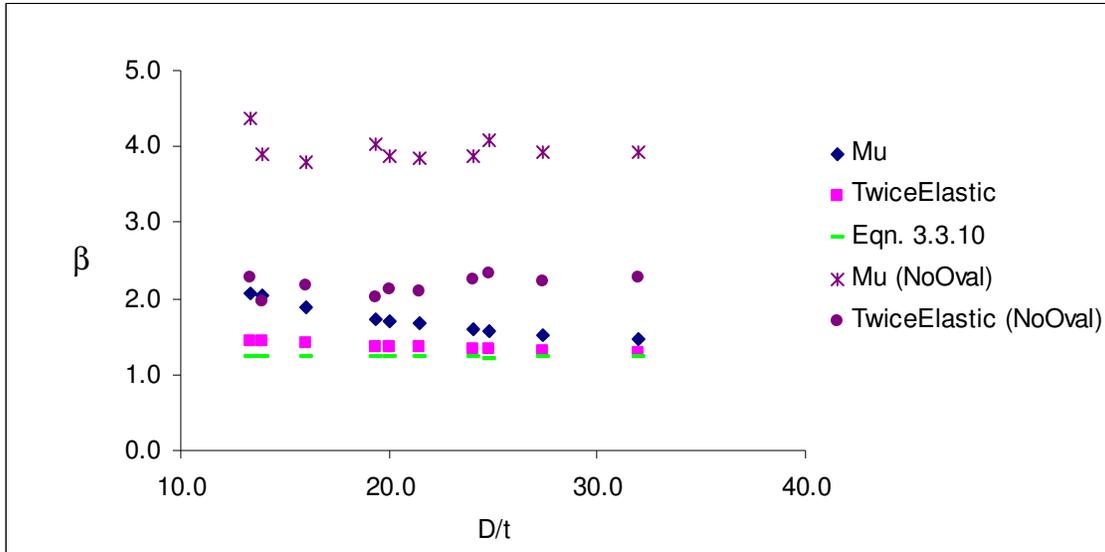


Fig. 3.6.4 Reliability Index for Bilinear Kinematic Hardening Material under Service Level D

“TwiceElastic” stands for “The collapse moment from the twice-elastic slope method” (see Section 2.1 for definition). “(NoOval)” stands for “No ovalization” and ovalization was not considered for that case. “Eqn. 3.3.7”, “Eqn. 3.3.9” and “Eqn. 3.3.10” represent results from closed-form solutions using those three equations. Figure 3.6.2 is similar to Figure 3.6.1 except it is for service level D. In these two figures, the trend of all curves is the same and reliability index levels from finite element analysis (which allows ovalization) and from closed form solutions (which do not) are reasonably close to each other. When the ovalization is not considered, the results from finite element analysis were expected to match those from the closed-form solutions, but those two sets of results do not agree well. This lack of agreement can be explained by the difference in the

moment-curvature curves between the closed-form solution and FEA using ABAQUS Elbow with no ovalization in Figure 3.4.1. Figure 3.6.3 and Figure 3.6.4 are results for bilinear kinematic hardening material. These two figures also show similar behaviors. Again, unrealistic high reliability index values are predicted when ovalization DOF of Elbow 31 is turned off.

From the β values in Figure 3.6.1 - 3.6.4, the following conclusions can be drawn about the straight pipe components that have been studied:

For service level C, the reliabilities of all components are above 99% if current ASME Code equation 9 is used.

For service level D, the reliabilities of all components are above 90% if current ASME Code equation 9 is used.

The reliability of a straight pipe component is not sensitive to D/t .

After comparing closed form solution results with finite element analysis results, the following observation can be made: closed form solutions provide an efficient approximation method for reliability index calculation for both elastic-perfectly plastic and bilinear kinematic hardening material models at service levels C and D if ovalization is not significant. Finite element analysis using ABAQUS Elbow 31 element with active ovalization degree of freedom can take into account geometric nonlinearities and can be employed in reliability analysis of thin walled straight pipes and other piping components.

Chapter 4

Summary and Future Research Recommendations

In this research, a sensitivity study was conducted to identify the most influential factors on the B_2 stress index of ASME Boiler and Pressure Vessel Code Section III equation 9 for elbow components from the initial pool of nine factors, which are Loading Type, Size, Schedule, Flange location, Material, Temperature, Pressure, Elbow bend angle, and Radius type, using formal Latin Hypercube design of experiments method. The elbow was modeled with Shell43 element of ANSYS finite element analysis code. The finite element analysis process was integrated and automated using iSIGHT.

Of those nine factors, the effects of Temperature and Radius Type were found to be less significant, thus eliminated for further consideration. The in-plane closing and the out-of-plane bending were found to be the loading types that always controlled. Therefore, these two were the only two loadings considered in the studies thereafter. To reduce the number of factors and the possible combinations of factors, the most commonly found flange location, i.e. with a tangent length of 5 times outside diameter, was used. Pipe size and schedule were combined into one factor h . A full factorial design of experiment study was then performed on Loading Type, h , Pressure, Bend Angle, Pressure, and Material. An empirical equation (also referred to as the NCSU equation) was composed from all the data of the full factorial design of experiment study. The NCSU equation matched all the finite element analysis results quite well. Compared to Touboul's equation, the NCSU equation has the advantage of being greater than one for all elbow components and being applicable to all ranges of parameter values. Thus, we conclude the NCSU equation is more suitable for the B_2 stress index calculation.

A reliability analysis using Advanced First Order Reliability Method was performed on ten straight pipe components with different size and schedule. ASME Code Eqn. 9, i.e. Eqn. (1.1.1), was the design equation investigated. The straight pipe component was modeled with Elbow 31 of ABAQUS finite element analysis code to include possible ovalization. As was the case in the sensitivity analysis, the finite element analysis process was integrated and automated using iSIGHT. Both elastic-perfectly plastic and bilinear kinematic hardening materials were considered. Twice-the-elastic-slope and ultimate moment were used as the failure criteria for service level C and D. Outside diameter, pipe wall thickness, material yield strength, and ultimate strength (if applicable) were treated as independent random variables subjected to Lognormal distribution. Monte Carlo sampling technique was employed to generate five-hundred-point samples. The results were compared to those from a previous closed form solution. Similar trends were observed from both sets of results. The reliability index was shown to be insensitive to the ratio of pipe diameter and wall thickness. For service level C, at least 99% of reliability can be achieved for all ten straight pipe components and for service level D, at least 90% of reliability can be achieved.

The automated finite element analysis process developed in this research is generic and can be easily extended to similar sensitivity or reliability studies. For instance, the reliability study process using ABAQUS and iSIGHT can be extended to a reliability study of elbow components. For this study, both B_2 definitions of current ASME Code and the one from NCSU equation can be considered. After a target reliability level is set, a reliability based the B_2 stress index can also be proposed. However, this work is beyond the scope of this research.

References:

1. ASME. *Boiler and Pressure Vessel Code*, Section III, Division 1, Subsection NB-3600 Piping Design, The American Society of Mechanical Engineers, 2001.
2. Rodabaugh, E.C. and George, H. H., (1957), "Effect of Internal Pressure on Flexibility and Stress-Intensification Factors of Curved Pipe or Welding Elbows," *Trans. ASME*, 79, pp 939-948.
3. Touboul, F. & Acker, D., (1991), "Excessive Deformation and Failure of Straight Parts and Elbows," *Proceedings of SMiRT 11*, Paper E02/2, pp 19-24, Tokyo, Japan, August 1991.
4. Dodge, W. G. and Moore, S. E., (1972), "Stress Indices and Flexibility Factors for Moment Loadings on Elbows and Curved Pipe," *WRC Bulletin 179*, December 1972.
5. Yu, L., (1999), "Elbow Stress Indices Using Finite Element Analysis," Ph.D. dissertation, North Carolina State University. [Report Number C-NPP-SEP 27/99, Center for Nuclear Power Plant Structures, Equipment, and Piping].
6. Matzen, V. C. and Tan, Y., (2002), "The History of the B_2 Stress Index.," *ASME Journal of Pressure Vessel Technology*, Vol. 124 , No. 2, 2002, pp 168-176.
7. C.F. Jeff Wu, Michael Hamada, "Experiments Planning, Analysis, and Parameter Design Optimization" New York: J. Wiley, c2000
8. Galambos, T.V., Ellingwood, B., MacGregor, J.G., and Cornell, C.A., (1982), "Probability-Based Load Criteria: Load Factors and Load Combinations", *ASCE Journal of Structural Division*, Vol. 108, No. 5, pp 978-997.
9. Rackwitz, R. and Fiessler, B., (1978), "Structural stability under combined random load sequences", *Computers and Structures*, 9, pp 489-584.
10. Saigal, R. K., "Seismic Analysis and Reliability-Based Design of Secondary Systems," Ph.D. Dissertation, North Carolina State University, 2005.
11. Swanson Analysis Systems, Version 8.0
12. Hibbitt, Karlsson & Sorensen, Inc., Version 6.5
13. Engineous Software, Inc., Version 9.0
14. Matzen, V. C. and Yuan, X., (2001), "The B_2 Stress Index as a Function of Internal Pressure, Bend Angle, Loading Type and Material," *Transactions of the 17th International Conference on Structural Mechanics in Reactor Technology*.
15. Tan, Y., "Experimental and Nonlinear FEA Investigation of Elbow Leading to a New Definition of the B_2 Stress Index," PhD Dissertation, North Carolina State University, 2001.
16. Tan, Y. and Matzen, V. C., (2002), "Correlation of In-Plane Bending Test and FEA Results for Thin-Walled Elbows," *Nuclear Engineering Design*, 217, pp 21-39.
17. Tan, Y., Wilkins, K., and Matzen, V. C., (2002), "Correlation of Test and FEA Results for Elbows Subjected to Out-of-Plane Loading," *Nuclear Engineering Design*, 217, pp 213-224.

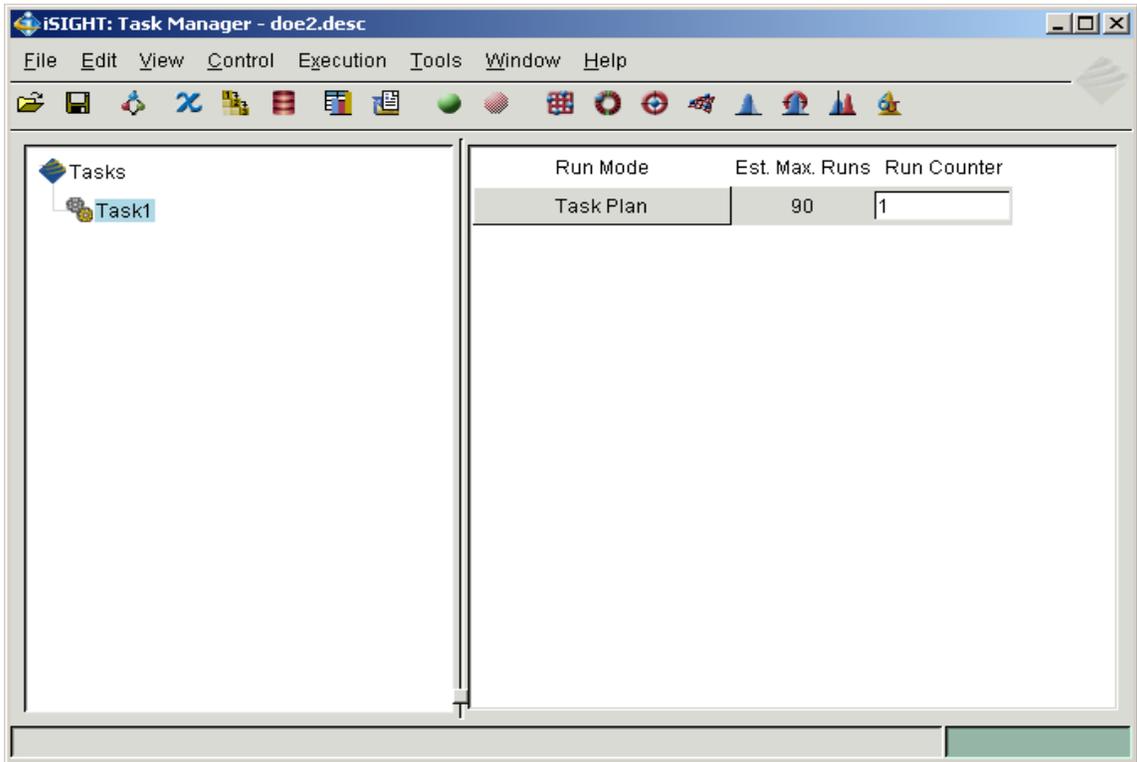
APPENDICES

APPENDIX A

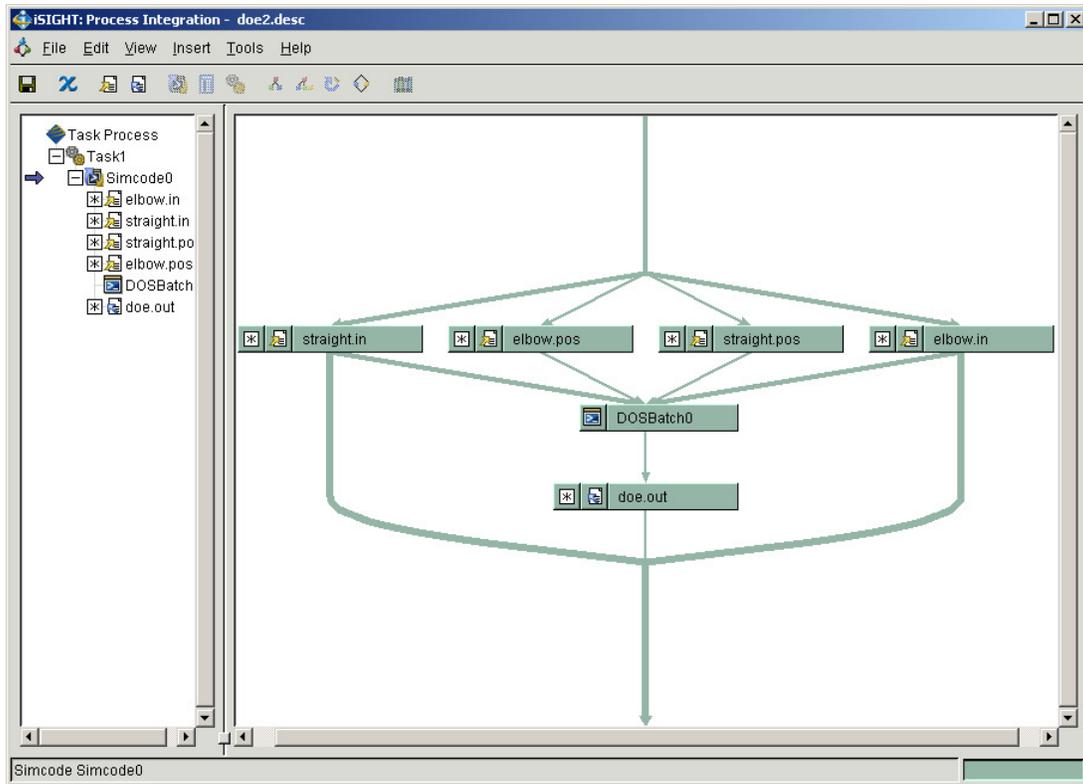
Finite Element Analysis Process Automation Using iSIGHT

In the traditional manual FEA process, the user has to either write input file line by line or utilize an input file from previous analysis as template, locate and change all the parameter values that are different from previous analysis. If error happens at one place, the entire analysis will be invalid and the user may not be aware of this. Thus, incorrect conclusion might be drawn from the problematic analysis and the error gets passed on and on. Also, a Monte Carlo Analysis may require thousands of FEA runs. Even if human-introduced error can be avoided, the time for input files preparation and output files data condensation still makes the study prohibitively expensive. To avoid human-introduced error and make mass analysis more affordable, an automated process has been developed and tested in our B₂ study.

A process integration and automation software, iSIGHT, has been employed in this process. (ANSYS is used as an example)

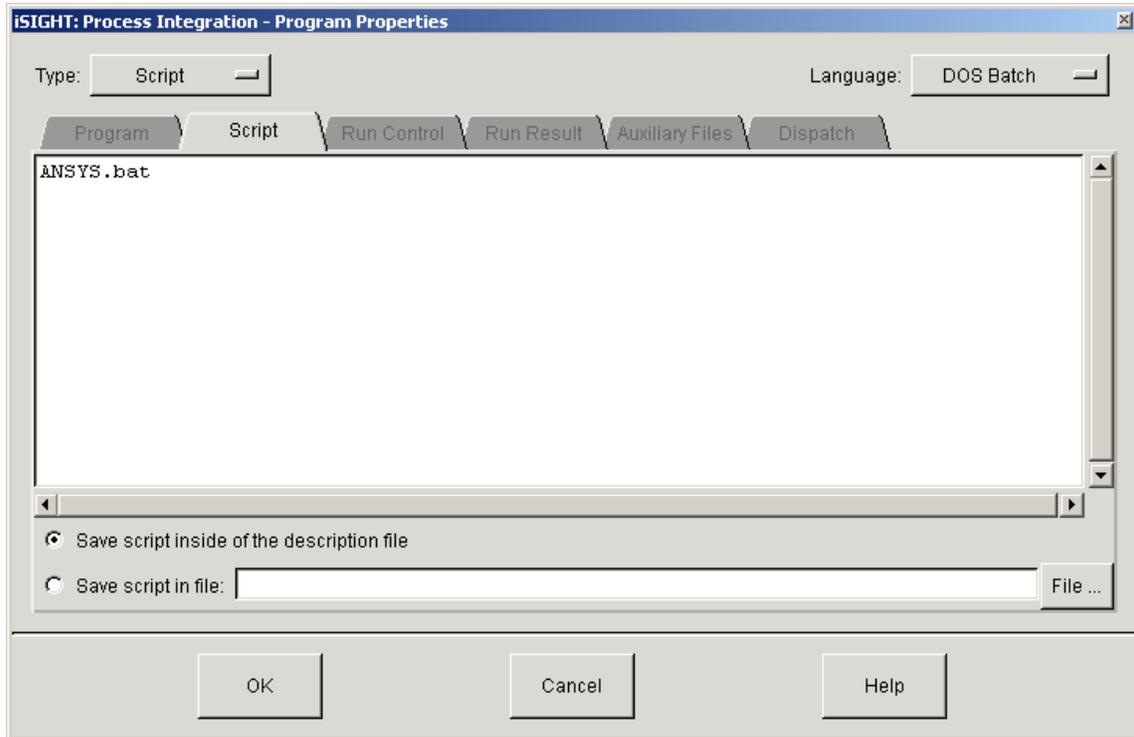


iSIGHT Main GUI



iSIGHT Process Integration GUI

First, the finite element analysis code (i.e. ANSYS or ABAQUS) needs to be integrated with iSIGHT. The simulation code (ANSYS or ABAQUS executable) has to be configured in iSIGHT simcode Program Properties GUI.



iSIGHT Program Properties GUI

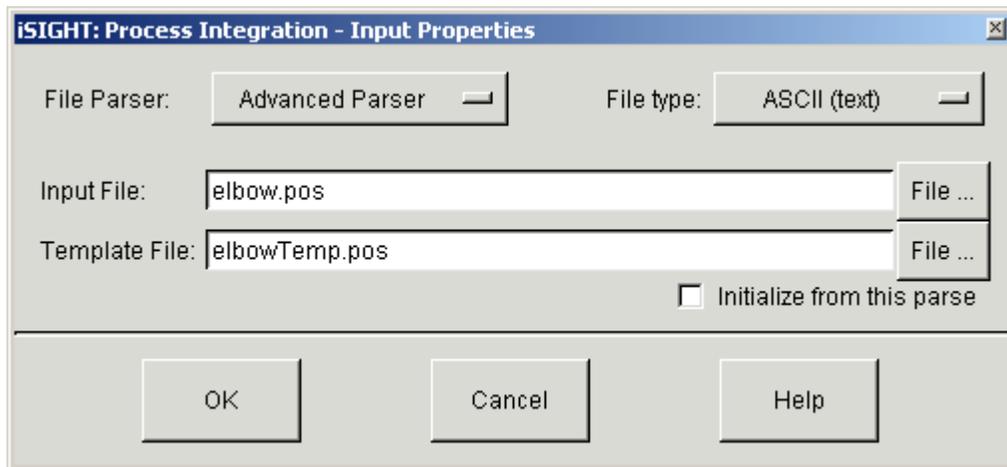
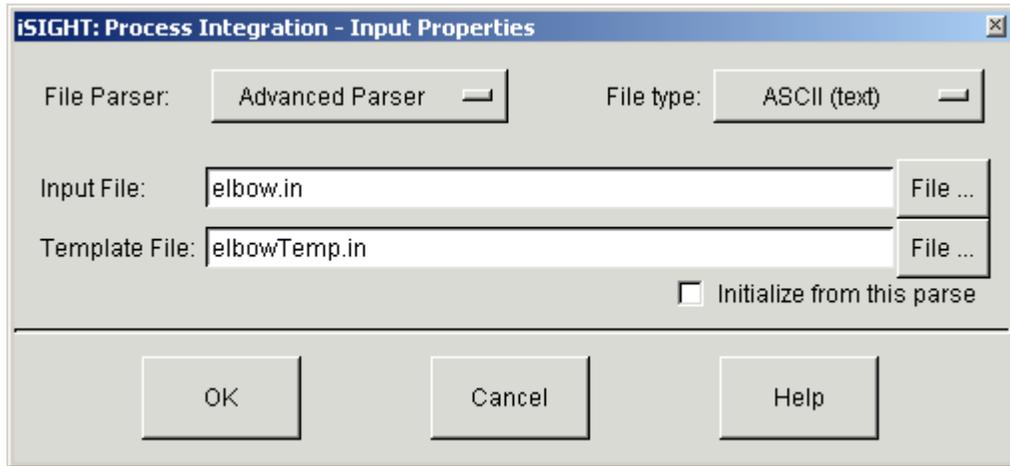
ANSYS.bat takes following form:

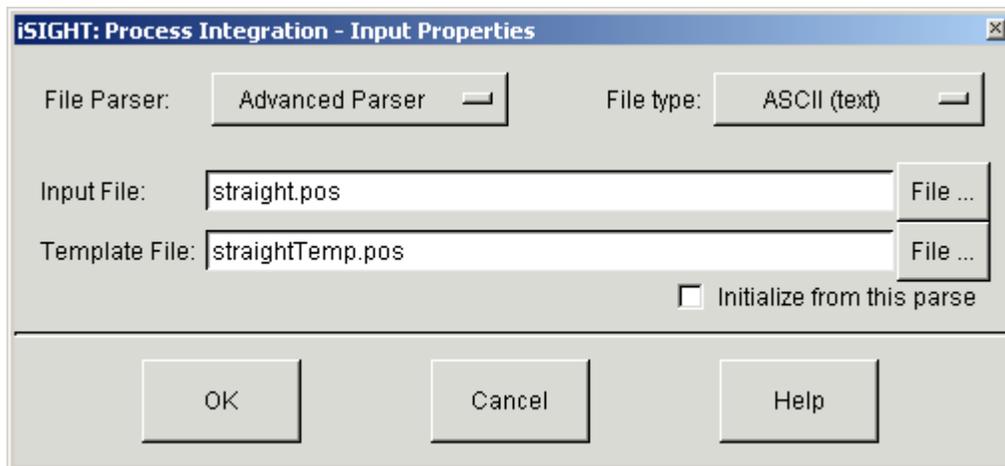
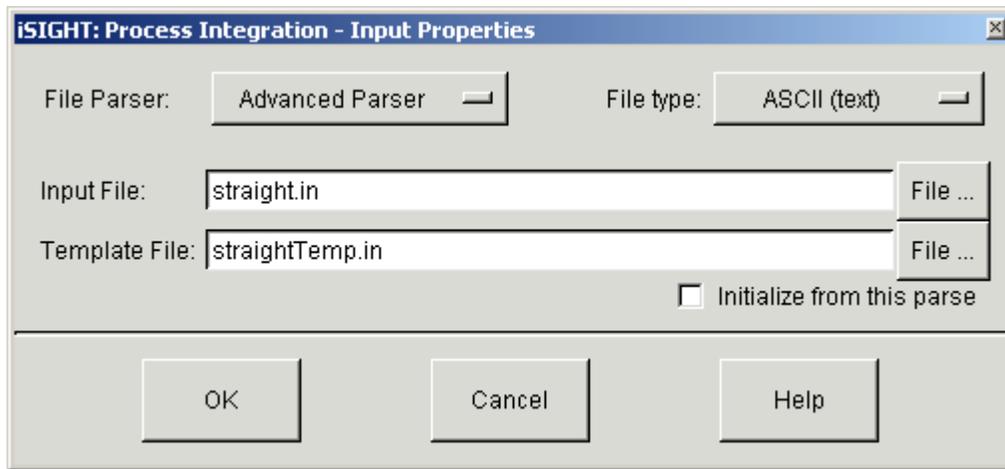
```
set ANSYS80_PRODUCT=ANSYSUL
set ANS_CONSEC=YES

ANSYS -b -i elbow.in -o junk
ANSYS -b -i elbow.pos -o junk
ANSYS -b -i straight.in -o junk
ANSYS -b -i straight.pos -o junk
```

Then input files (elbow.in, elbow.pos, straight.in, and straight.pos) will be defined in

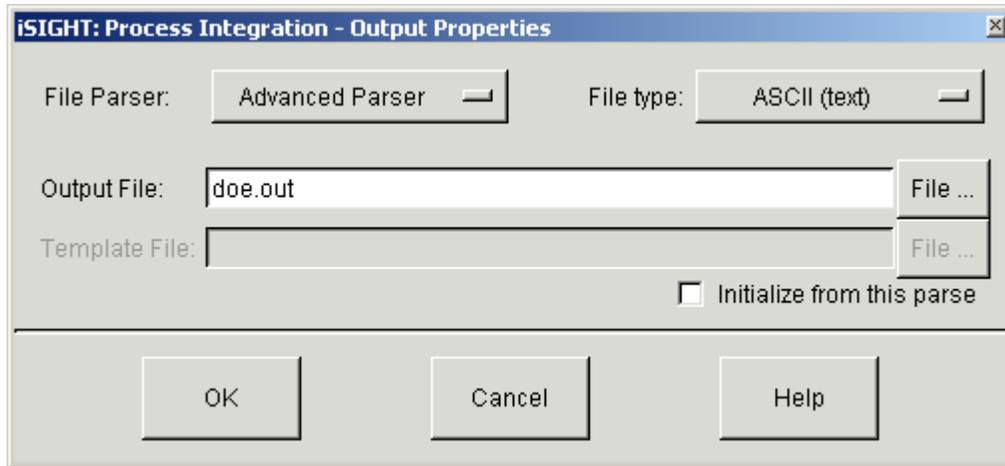
Input Properties GUI:





ISIGHT Input Properties GUI

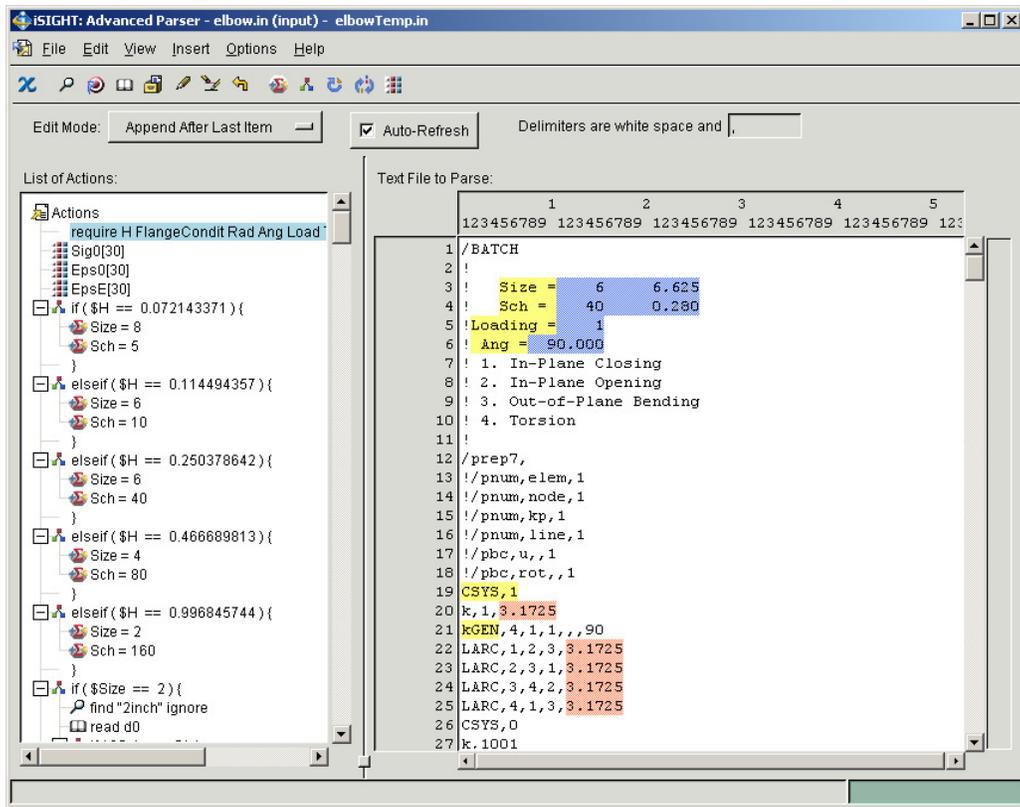
Output file (doe.out) will be defined in Output Properties GUI:



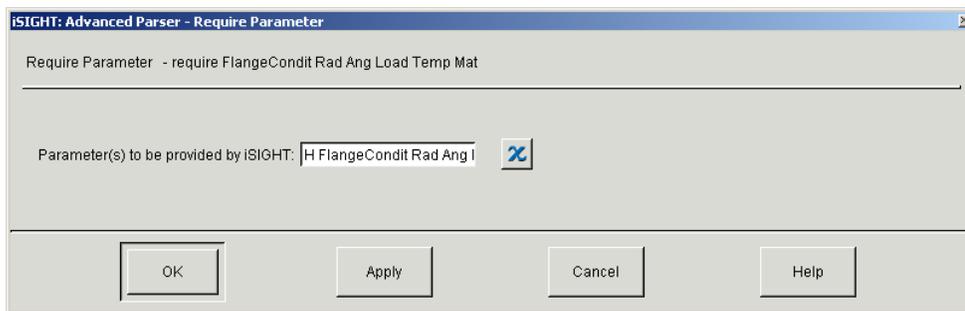
iSIGHT Output Properties GUI

Next, a series of instructions has to be created in iSIGHT file parser GUI in order to generate input files for Simcode (ANSYS or ABAQUS) at runtime from given templates (sample input files). In the B₂ sensitivity analysis, there are eight parameters: “Loading Type”, “H”, “Internal Pressure”, “Radius”, “Bend Angle”, “Temperature”, “Material”, and “Flange Location”. A finite element analysis is fully determined only if these eight parameters are decided and written into input file. Here are the steps to write these parameters into input file:

1. Define input parameters in iSIGHT file parser GUI:

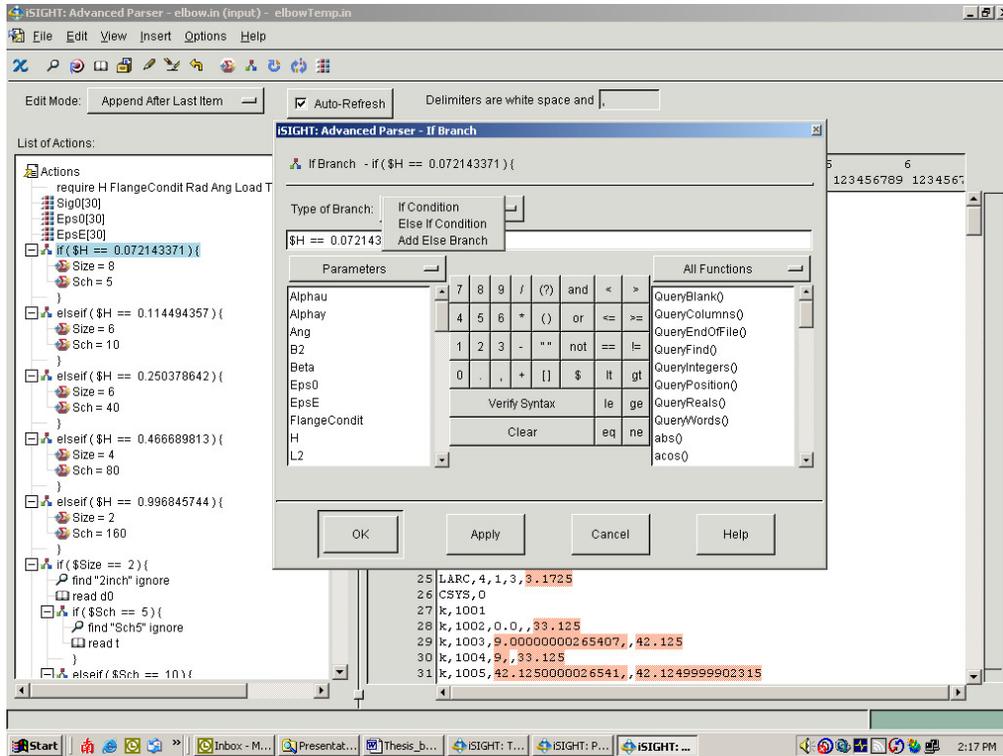


iSIGHT File Parser Main GUI



File Parser Parameter Definition GUI

2. Decide component size and schedule from “H”



File Parser If Branch

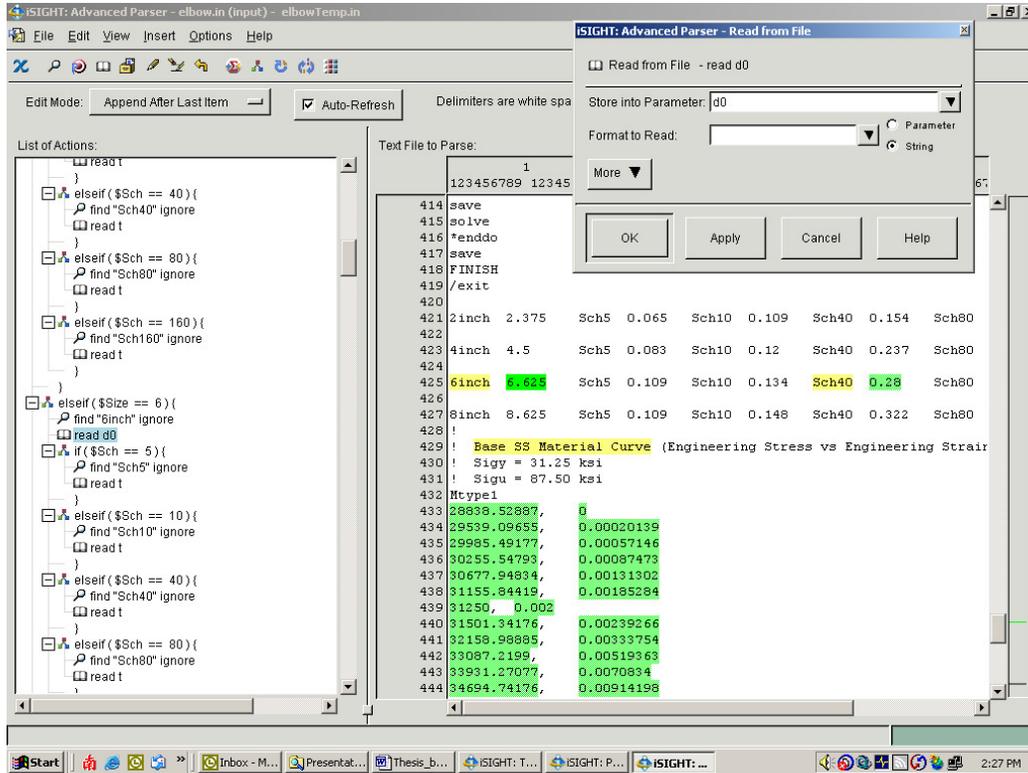
Following “if and else-if” logic is used to decide “Size” and “Schedule”:

```

if ( $H == 0.072143371 ) {
    Size = 8
    Sch = 5
} elseif ( $H == 0.114494357 ) {
    Size = 6
    Sch = 10
} elseif ( $H == 0.250378642 ) {
    Size = 6
    Sch = 40
} elseif ( $H == 0.466689813 ) {
    Size = 4
    Sch = 80
} elseif ( $H == 0.996845744 ) {
    Size = 2
    Sch = 160
}

```

3. Then outside diameter and thickness are looked up from a table attached to the input template file:



File Parser Read Function

Following “if and else-if” logic is also used with “Read” function:

```

if ( $Size == 2 ) {
  find "2inch" ignore
  read d0
  if( $Sch == 5 ) {
    find "Sch5" ignore
    read t
  } elseif ( $Sch == 10 ) {
    find "Sch10" ignore
    read t
  } elseif ( $Sch == 40 ) {
    find "Sch40" ignore
    read t
  } elseif ( $Sch == 80 ) {
    find "Sch80" ignore
    read t
  } elseif ( $Sch == 160 ) {
    find "Sch160" ignore
    read t
  }
}

```

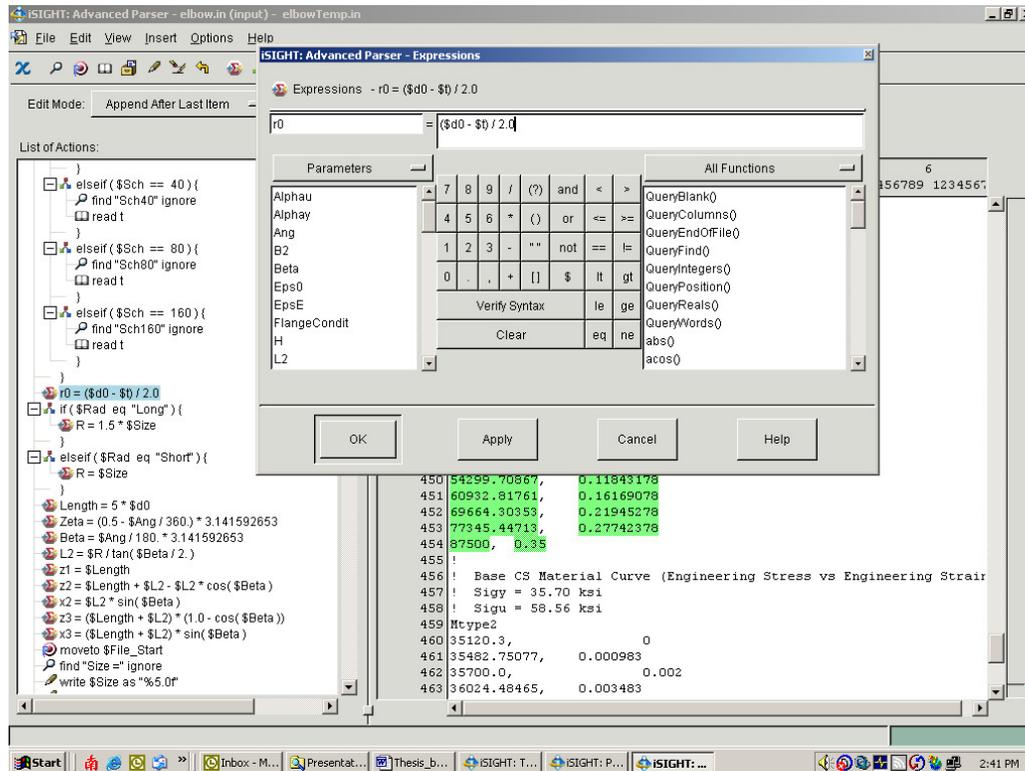
```

} elseif ( $Size == 4 ) {
    find "4inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
} elseif ( $Size == 6 ) {
    find "6inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
} elseif ( $Size == 8 ) {
    find "8inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
}

```

}

4. Component geometries are calculated from outside diameter and wall thickness:



File Parser Calculation Function

Here are all the geometry related calculations:

```

r0 = ($d0 - $t) / 2.0
if ( $Rad eq "Long" ) {
  R = 1.5 * $Size
} elseif ( $Rad eq "Short" ) {
  R = $Size
}
Length = 5 * $d0
Zeta = (0.5 - $Ang / 360.) * 3.141592653
Beta = $Ang / 180. * 3.141592653
L2 = $R / tan( $Beta / 2. )
z1 = $Length
z2 = $Length + $L2 - $L2 * cos( $Beta )
x2 = $L2 * sin( $Beta )
z3 = ($Length + $L2) * (1.0 - cos( $Beta ))
x3 = ($Length + $L2) * sin( $Beta )
moveto $File_Start
find "Size =" ignore
write $Size as "%5.0f"
write $d0 as "%10.3f"
find "Sch =" ignore

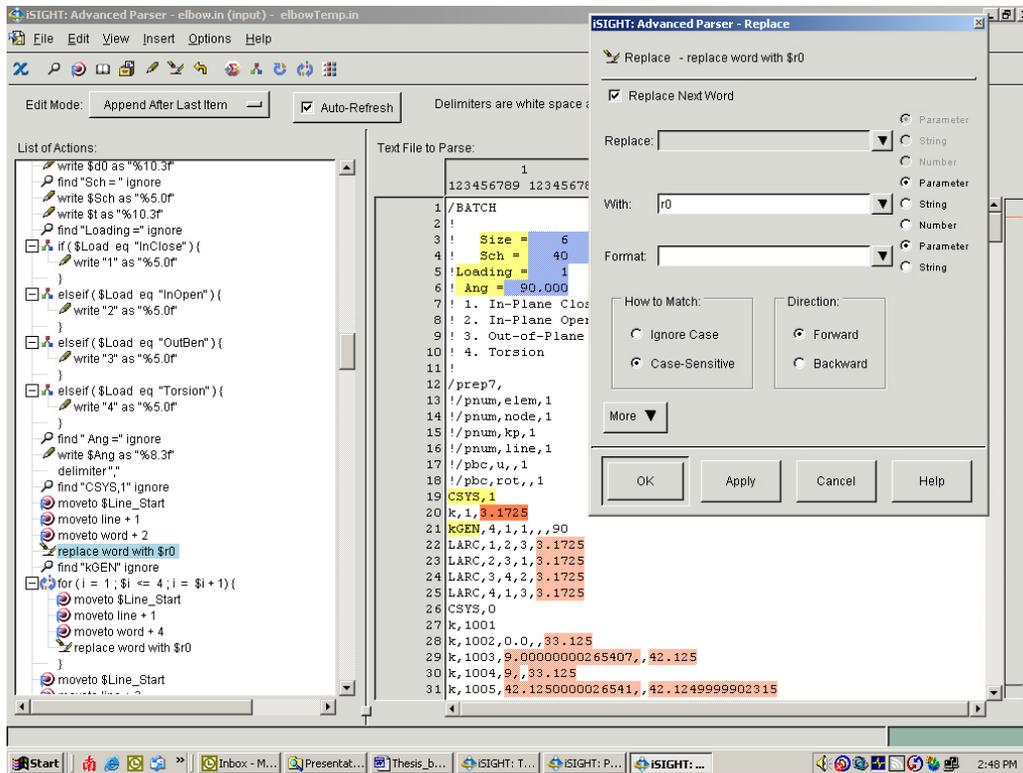
```

```

write $Sch as "%5.0f"
write $t as "%10.3f"
find "Loading =" ignore
if ( $Load eq "InClose" ) {
    write "1" as "%5.0f"
} elseif ( $Load eq "InOpen" ) {
    write "2" as "%5.0f"
} elseif ( $Load eq "OutBen" ) {
    write "3" as "%5.0f"
} elseif ( $Load eq "Torsion" ) {
    write "4" as "%5.0f"
}

```

5. Then the geometries are written into input file to define the component as a finite element model:



File Parser Write Function

Following are instructions for finite element modeling:

```

find " Ang =" ignore
write $Ang as "%8.3f"
delimiter ","
find "CSYS,1" ignore
moveto $Line_Start
moveto line + 1
moveto word + 2

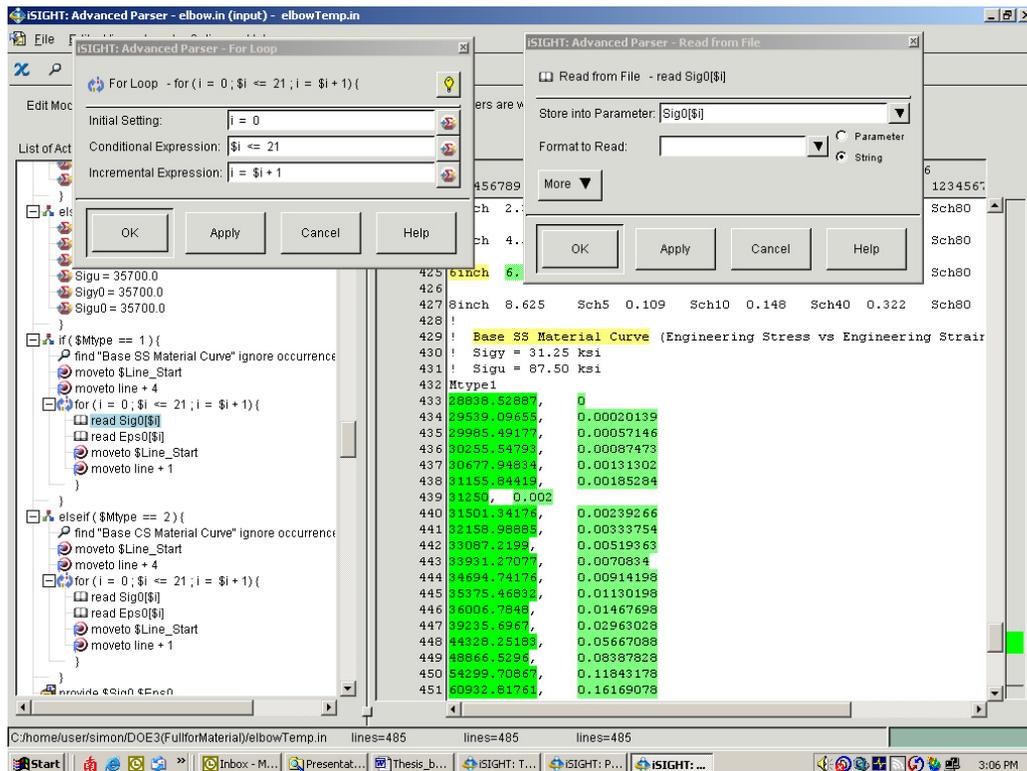
```

```

replace word with $r0
find "kGEN" ignore
for ( i = 1 ; $i <= 4 ; i = $i + 1 ) {
    moveto $Line_Start
    moveto line + 1
    moveto word + 4
    replace word with $r0
}
moveto $Line_Start
moveto line + 3
moveto word + 3
replace word with $z1
moveto $Line_Start
moveto line + 1
moveto word + 2
replace word with $x2
replace word with $z2
moveto $Line_Start
moveto line + 1
moveto word + 2
replace word with $R
replace word with $z1
moveto $Line_Start
moveto line + 1
moveto word + 2
replace word with $x3
replace word with $z3
moveto $Line_Start
moveto line + 2
moveto word + 4
replace word with $R
find "R,1," ignore
replace word with $t

```

6. For material data, a linear scaling technique introduced by Matzen and Tan is used here. First, a base-line curve will be selected based on material, and all the stress-strain points will be read into array parameters from a table attached to the input template file.



Read Stress Data into Array Using A “For Loop”

Here are the instructions:

```

if ( $Mtype == 1 ) {
  find "Base SS Material Curve" ignore occurrence all
  moveto $Line_Start
  moveto line + 4
  for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    read Sig0[$i]
  }
}

```

```

        read Eps0[$i]
        moveto $Line_Start
        moveto line + 1
    }
} elseif ( $Mtype == 2 ) {
find "Base CS Material Curve" ignore occurrence all
moveto $Line_Start
moveto line + 4
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    read Sig0[$i]
    read Eps0[$i]
    moveto $Line_Start
    moveto line + 1
}
}
provide $Sig0 $Eps0

```

7. Then the base-line stress-strain curve will be scaled using linear scaling technique. The scaled stress-strain data will be written into finite element analysis input file.

Here are the instructions:

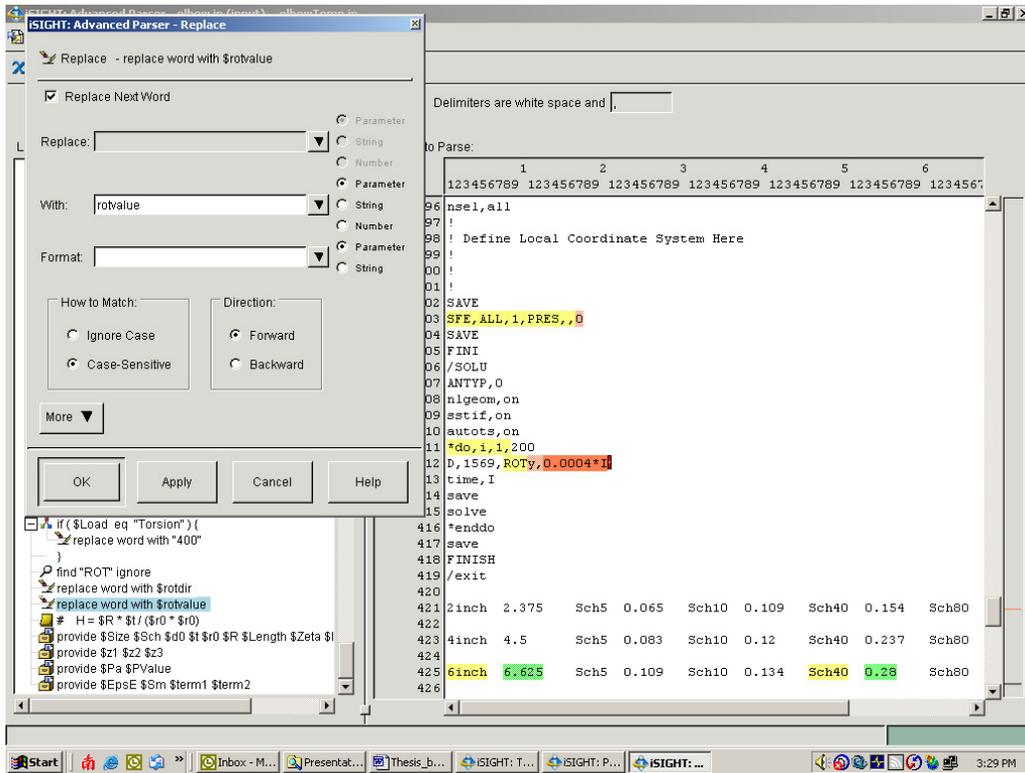
```

Alphay = $Sigy / $Sigy0
Alphau = $Sigu / $Sigu0
a = ($Alphau - $Alphay) / (0.35 - 0.002)
b = $Alphay - 0.002 * $a
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    Eps0[$i] = $Eps0[$i] - $Sig0[$i] / 28300000.0
    Sig0[$i] = $Sig0[$i] * ($a * $Eps0[$i] + $b)
    Eps0[$i] = $Eps0[$i] + $Sig0[$i] / 28300000.0
    Sig0[$i] = $Sig0[$i] * (1.0 + $Eps0[$i])
    Eps0[$i] = log( 1.0 + $Eps0[$i] )
}
find "! Material #1" ignore occurrence all
moveto $Line_Start
moveto line + 8
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    moveto word + 2
    replace word with $Eps0[$i]
    replace word with $Sig0[$i]
    moveto $Line_Start
    moveto line + 1
}
provide $Sig0
provide $Alphay $Alphau $a $b
provide $Sigy0 $Sigu0
term1 = $Sigu / 3.0
term2 = $Sigy * 2.0 / 3.0
Sm = min( $term1, $term2 )
find "ASEL,S,,,1,12" ignore
moveto $Line_Start
moveto line + 1

```

moveto word + 1
 replace word with \$Mindex

- The loading data are written into input file according to load type specified in current analysis:



Define Loading Type

Here are the instructions for loading definition:

```
Line1 = "CSKP,12,0,1001,1005,1004"
Line2 = "CSYS,12"
Line3 = "nrotat,all"
if ( $Load eq "Torsion" ) {
  rotmdir = "x"
  rotvalue = "0.0001*I"
  find "! Define Local Coordinate System Here" ignore
```

```

        moveto $Line_Start
        moveto line + 1
        write $Line1
        moveto $Line_Start
        moveto line + 1
        write $Line2
        moveto $Line_Start
        moveto line + 1
        write $Line3
    } elseif ( $Load eq "OutBen" ) {
        rotmdir = "y"
        rotvalue = "0.0004*I"
    find "! Define Local Coordinate System Here" ignore
        moveto $Line_Start
        moveto line + 1
        write $Line1
        moveto $Line_Start
        moveto line + 1
        write $Line2
        moveto $Line_Start
        moveto line + 1
        write $Line3
    } elseif ( $Load eq "InClose" ) {
        rotmdir = "y"
        rotvalue = "0.0004*I"
    } elseif ( $Load eq "InOpen" ) {
        rotmdir = "y"
        rotvalue = "-0.0004*I"
    }
    }
    Pa = 2 * $Sm * $t / ($d0 - 0.8 * $t)
    PValue = $Pres * $Pa
    find "SFE,ALL,1,PRES,, " ignore
    replace word with $PValue
    find "*do,i,1," ignore
    if ( $Load eq "Torsion" ) {
        replace word with "400"
    }
    }
    find "ROT" ignore
    replace word with $rotmdir
    replace word with $rotvalue

```

9. “elbow.pos” is compiled to perform the post process

Similar process is also used for “straight.in” and “straight.pos”, which are the input and postprocess files for straight pipe component.

After FEA simulation code execution is completed, the important analysis data will be extracted from output file by iSIGHT. This process is output file parsing. Using ABAQUS as example, the output file parsing instructions may look like the following:

```
# Define Array and Get Parameter "Mccode"
```

```

require Mccode
K[500]
M[500]
Mc = 0

# Read in Moment-Curvature Data

for ( i = 0 ; $i <= 499 ; i = $i + 1 ) {
  if ( $i == 0 ) {
    moveto line + 272
  } else {
    moveto line + 27
  }
  moveto word + 1
  read tmp
  K[$i] = abs( $tmp )
  read tmp
  M[$i] = abs( $tmp )
  if ( $Mc < abs( $tmp ) ) {
    Mc = abs( $tmp )
  }
}

# Find Ultimate Moment if It Exists

Last = $M[499]
if ( $Last < $Mc ) {
  Status = 0
} else {
  Status = 1
}

# Calculate Twice Elastic Moment

MK = ($M[1] - $M[0]) / ($K[1] - $K[0]) / 2.
tmp = $MK * $K[499]
if ( $tmp < $Last ) {
  M2 = 0
} else {
  j = 1
  get = 0
  while ( ($get == 0) and ($j < 500) ) {
    tmp1 = $MK * $K[$j]
    tmp2 = $M[$j]
    if ( $tmp1 > $tmp2 ) {
      j1 = $j - 1
      up = ($M[$j] - $M[$j1]) * ($M[$j1] - $MK * $K[$j1])
      dw = $MK * $K[$j] - $M[$j] + $M[$j1] - $MK * $K[$j1]
      M2 = $M[$j1] + $up / $dw
      get = 1
    } else {
      j = $j + 1
    }
  }
}

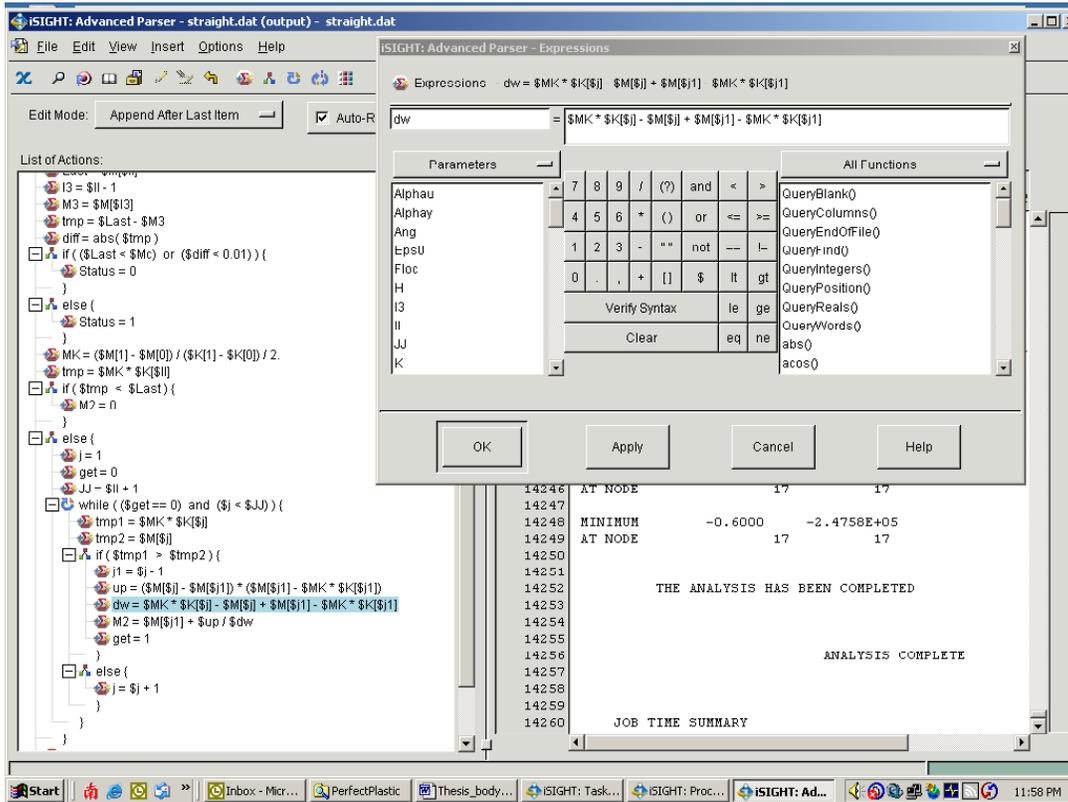
# Calculate Margins for Two Failure Criteria

```

```

Margin1 = $Mc - $Mccode
Margin2 = $M2 - $Mccode
provide $Mc $M2 $Margin1 $Margin2 $Status
provide $K $M $tmp $Last $MK $tmp1 $tmp2 $up $dw

```

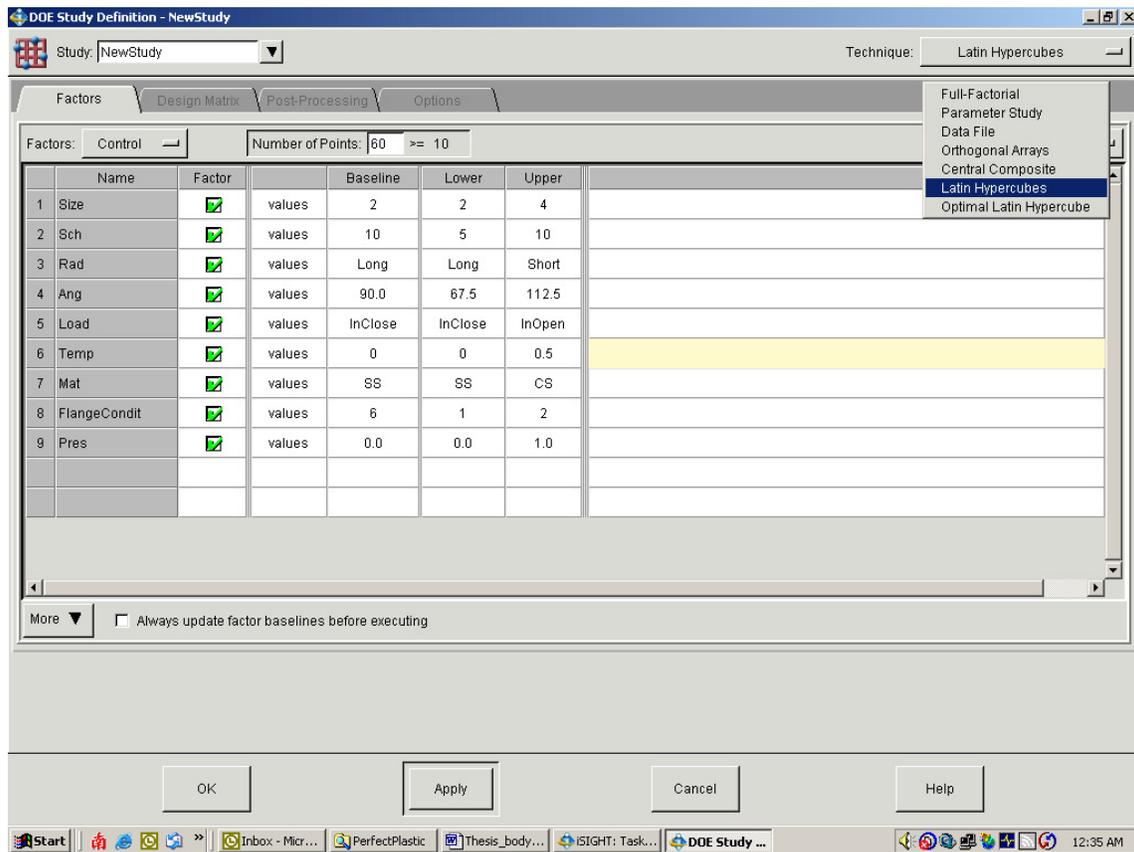


Calculate Twice-the-Elastic Moment

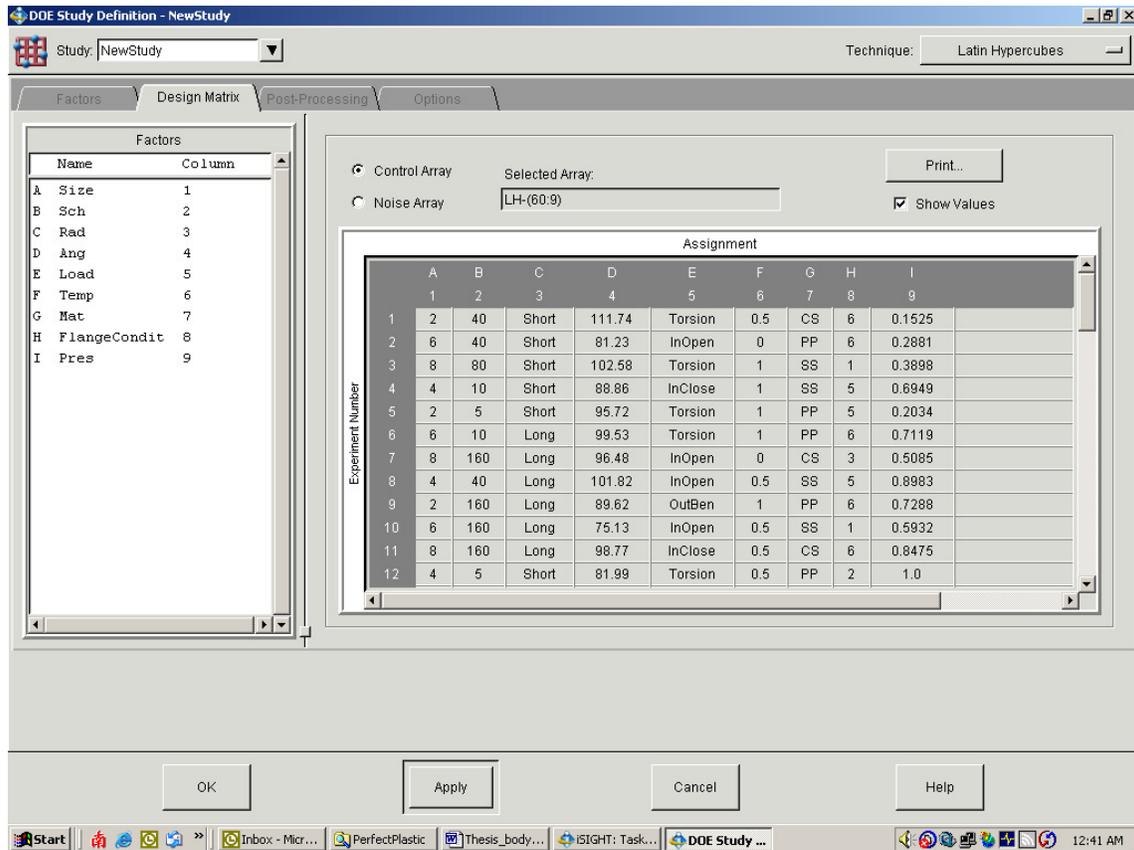
After finishing input parsing and output parsing instructions, the process integration is completed. At runtime, real input file will be created with desired input parameter values, and output parameter values will be extracted and sent back to iSIGHT for review. This process will be executed automatically by iSIGHT a number of times depending on what technique is applied to the process.

Next, a specific study such as Design of Experiments, Optimization or Monte Carlo Analysis can be selected to apply to the process. This step is also called problem definition.

For sensitivity analysis, Design of Experiments study is performed. Latin Hypercube technique is used in our initial study. In this study, there are nine factors and a total of 60 experiments is performed.

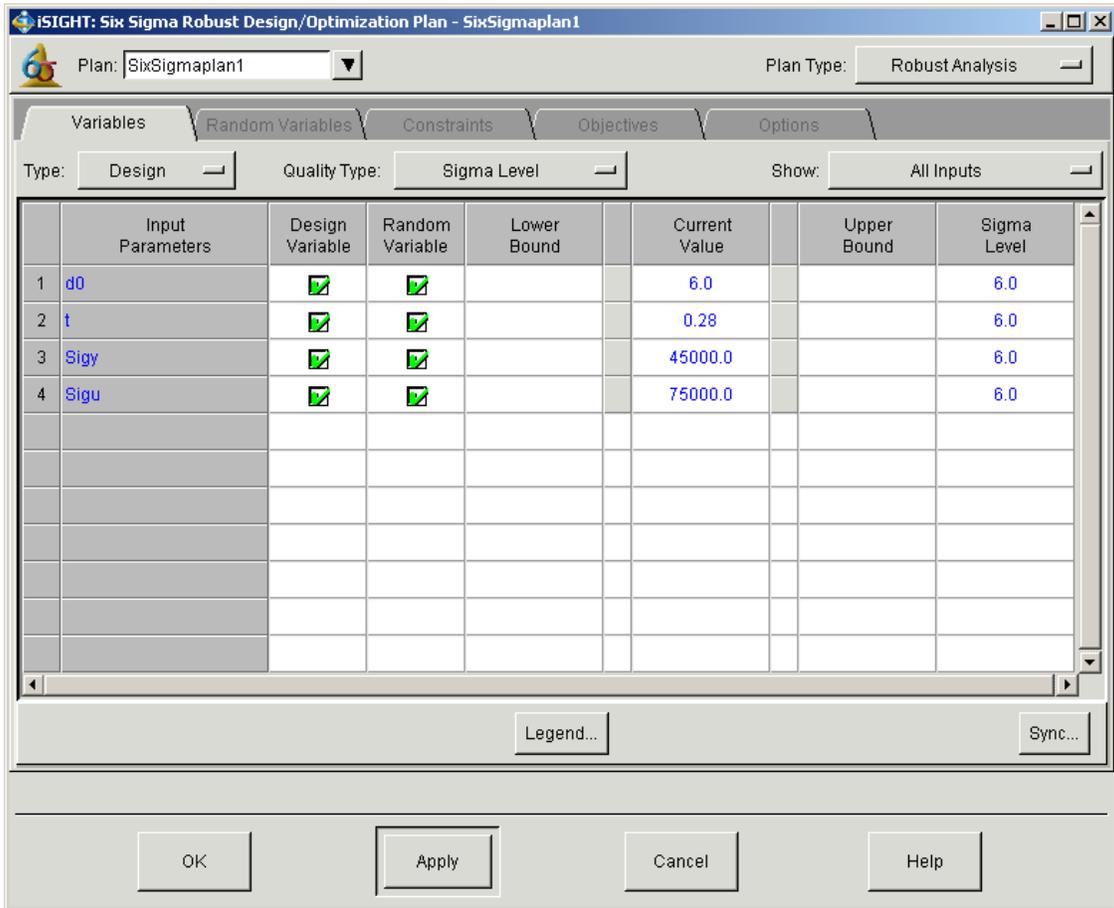


Design Of Experiment Main GUI

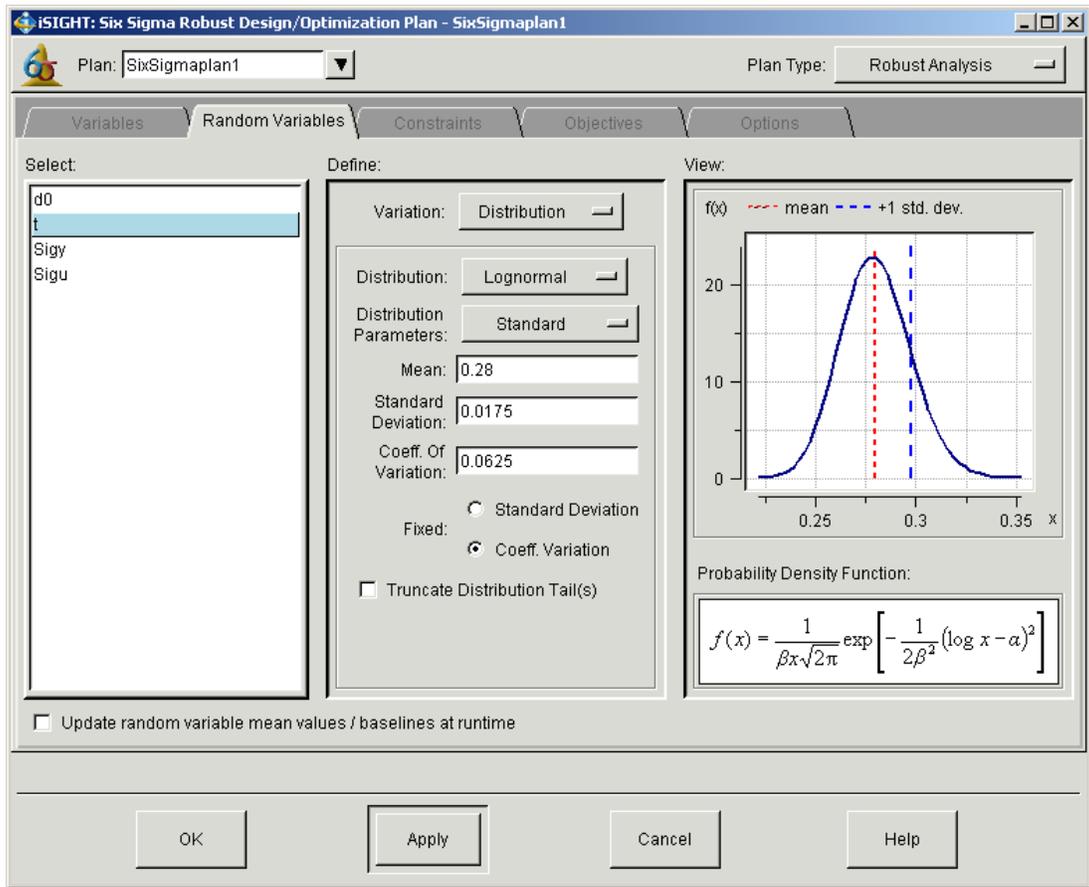


Design Matrix for Initial Latin

Reliability study is performed using Monte Carlo Sampling and First Order Reliability Method (FORM). There are four random variables all subject to Lognormal distribution.



Sixsigma Study Main GUI



Random Variables Definition


```

Version: 1.0
Evaluation: taskplan
ControlMode: user
RunCounter: 1
Audit: on Milestone Error
Audit: off Debug Info Status Normal
BoundsPolicy: adjustvalue
CheckPoint: unknown
RandGenInit: default
End TaskHeader Task1

Inputs Task1
Parameter: Load Type: discrete InitialValue: InClose
Parameter: H Type: discrete InitialValue: 0.250378642
Parameter: Pres Type: real InitialValue: 0.0
Parameter: Rad Type: discrete InitialValue: Long
Parameter: Ang Type: real InitialValue: 90.0
Parameter: Temp Type: integer InitialValue: 0
Parameter: Mat Type: discrete InitialValue: SSHigh
Parameter: FlangeCondit Type: discrete InitialValue: 6
End Inputs Task1

Outputs Task1
Parameter: Size Type: real
Parameter: Sch Type: real
Parameter: d0 Type: real
Parameter: t Type: real
Parameter: r0 Type: real
Parameter: R Type: integer
Parameter: Length Type: real
Parameter: Zeta Type: real
Parameter: Beta Type: real
Parameter: L2 Type: real
Parameter: z1 Type: real
Parameter: z2 Type: real
Parameter: x2 Type: real
Parameter: z3 Type: real
Parameter: x3 Type: real
Parameter: B2 Type: real
Parameter: Pa Type: real
Parameter: PValue Type: real
Parameter: Sigy0 Type: real
Parameter: Sigu0 Type: real
Parameter: Sig0[30] Type: real
Parameter: Eps0[30] Type: real
Parameter: EpsE[30] Type: real
Parameter: Alphay Type: real
Parameter: Alphau Type: real
Parameter: a Type: real
Parameter: b Type: real
Parameter: Sm Type: real
Parameter: term1 Type: real
Parameter: term2 Type: real
Parameter: Sigy Type: real
Parameter: Sigu Type: real
End Outputs Task1

```

SimCode Simcode0

```
InputFiles Simcode0
  FileDescription elbowin
  FileType: standard
  TemplateFile: "elbowTemp.in"
  InputFile: "elbow.in"
  InputParameters
    Rad Ang Load Temp Mat FlangeCondit Pres H
  OutputParameters
    d0 t r0 R Length Zeta Beta L2 z1 z2
    Sigy Sigu x2 z3 x3 Pa PValue Size Sch Sig0
    Eps0 Alphas Alpha a b Sigy0 Sigu0 Sm term1 term2
    EpsE
  Instructions
```

```
#Define Input Parameters
```

```
  require Pres H FlangeCondit Rad Ang Load Temp Mat
```

```
#Define Arrays for Stress-Strain Data
```

```
Sig0[30]
```

```
Eps0[30]
```

```
EpsE[30]
```

```
  if ( $H == 0.072143371 ) {
    Size = 8
    Sch = 5
  } elseif ( $H == 0.114494357 ) {
    Size = 6
    Sch = 10
  } elseif ( $H == 0.250378642 ) {
    Size = 6
    Sch = 40
  } elseif ( $H == 0.466689813 ) {
    Size = 4
    Sch = 80
  } elseif ( $H == 0.996845744 ) {
    Size = 2
    Sch = 160
  }
}
```

```
#Look-up Pipe Out-Diameter and Thickness from Attached
```

Table

```
  if ( $Size == 2 ) {
    find "2inch" ignore
    read d0
    if ( $Sch == 5 ) {
      find "Sch5" ignore
      read t
    } elseif ( $Sch == 10 ) {
      find "Sch10" ignore
      read t
    } elseif ( $Sch == 40 ) {
      find "Sch40" ignore
      read t
    }
  }
```

```

    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
} elseif ( $Size == 4 ) {
    find "4inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
} elseif ( $Size == 6 ) {
    find "6inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
} elseif ( $Size == 8 ) {
    find "8inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    }
}

```

```

        } elseif ( $Sch == 80 ) {
            find "Sch80" ignore
            read t
        } elseif ( $Sch == 160 ) {
            find "Sch160" ignore
            read t
        }
    }

#Calculate Geometric Parameters for ANSYS input file

r0 = ($d0 - $t) / 2.0
if ( $Rad eq "Long" ) {
    R = 1.5 * $Size
} elseif ( $Rad eq "Short" ) {
    R = $Size
}
Length = 5 * $d0
Zeta = (0.5 - $Ang / 360.) * 3.141592653
Beta = $Ang / 180. * 3.141592653
L2 = $R / tan( $Beta / 2. )
z1 = $Length
z2 = $Length + $L2 - $L2 * cos( $Beta )
x2 = $L2 * sin( $Beta )
z3 = ($Length + $L2) * (1.0 - cos( $Beta ))
x3 = ($Length + $L2) * sin( $Beta )
moveto $File_Start
find "Size =" ignore
write $Size as "%5.0f"
write $d0 as "%10.3f"
find "Sch =" ignore
write $Sch as "%5.0f"
write $t as "%10.3f"
find "Loading =" ignore
if ( $Load eq "InClose" ) {
    write "1" as "%5.0f"
} elseif ( $Load eq "InOpen" ) {
    write "2" as "%5.0f"
} elseif ( $Load eq "OutBen" ) {
    write "3" as "%5.0f"
} elseif ( $Load eq "Torsion" ) {
    write "4" as "%5.0f"
}
}

#Write Geometric Parameters into ANSYS input file

find " Ang =" ignore
write $Ang as "%8.3f"
delimiter ","
find "CSYS,1" ignore
moveto $Line_Start
moveto line + 1
moveto word + 2
replace word with $r0
find "kGEN" ignore
for ( i = 1 ; $i <= 4 ; i = $i + 1 ) {
    moveto $Line_Start

```

```

        moveto line + 1
        moveto word + 4
        replace word with $r0
    }
    moveto $Line_Start
    moveto line + 3
    moveto word + 3
    replace word with $z1
    moveto $Line_Start
    moveto line + 1
    moveto word + 2
    replace word with $x2
    replace word with $z2
    moveto $Line_Start
    moveto line + 1
    moveto word + 2
    replace word with $R
    replace word with $z1
    moveto $Line_Start
    moveto line + 1
    moveto word + 2
    replace word with $x3
    replace word with $z3
    moveto $Line_Start
    moveto line + 2
    moveto word + 4
    replace word with $R
    find "R,1," ignore
    replace word with $t

```

#Assign Material Properties for Given Material and Temperature

```

if ( $Temp == 0 and $Mat eq "SSCode" ) {
    Mindex = 1
    Mtype = 1
    Sigy = 25000.0
    Sigu = 70000.0
    Sigy0 = 31250.0
    Sigu0 = 87500.0
} elseif ( $Temp == 0 and $Mat eq "SSHHigh" ) {
    Mindex = 1
    Mtype = 1
    Sigy = 50000.0
    Sigu = 100000.0
    Sigy0 = 31250.0
    Sigu0 = 87500.0
} elseif ( $Temp == 0 and $Mat eq "CS" ) {
    Mindex = 1
    Mtype = 2
    Sigy = 35700.0
    Sigu = 58560.0
    Sigy0 = 35700.0
    Sigu0 = 58560.0
} elseif ( $Temp == 0 and $Mat eq "PP" ) {
    Mindex = 1
    Mtype = 3
    Sigy = 35700.0

```

```

        Sigu = 35700.0
        Sigy0 = 35700.0
        Sigu0 = 35700.0
    }

#Read in Baseline Stress-Strain Data for Given Material

    if ( $Mtype == 1 ) {
    find "Base SS Material Curve" ignore occurrence all
    moveto $Line_Start
    moveto line + 4
    for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
        read Sig0[$i]
        read Eps0[$i]
        moveto $Line_Start
        moveto line + 1
    }
    } elseif ( $Mtype == 2 ) {
    find "Base CS Material Curve" ignore occurrence all
    moveto $Line_Start
    moveto line + 4
    for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
        read Sig0[$i]
        read Eps0[$i]
        moveto $Line_Start
        moveto line + 1
    }
    }
    provide $$Sig0 $Eps0

#Construct Appropriate Stress-Strain Data from Baseline Stress-
Strain Data by Linear Scaling and Write the Data into ANSYS Input
File

    Alphay = $$Sigy / $$Sigy0
    Alphau = $$Sigu / $$Sigu0
    a = ($Alphau - $Alphay) / (0.35 - 0.002)
    b = $Alphay - 0.002 * $a
    for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
        Eps0[$i] = $Eps0[$i] - $Sig0[$i] / 28300000.0
        Sig0[$i] = $Sig0[$i] * ($a * $Eps0[$i] + $b)
        Eps0[$i] = $Eps0[$i] + $Sig0[$i] / 28300000.0
        Sig0[$i] = $Sig0[$i] * (1.0 + $Eps0[$i])
        Eps0[$i] = log( 1.0 + $Eps0[$i] )
    }
    find "! Material #1" ignore occurrence all
    moveto $Line_Start
    moveto line + 8
    for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
        moveto word + 2
        replace word with $Eps0[$i]
        replace word with $Sig0[$i]
        moveto $Line_Start
        moveto line + 1
    }
    provide $$Sig0
    provide $Alphay $Alphau $a $b
    provide $$Sigy0 $$Sigu0

```

```

term1 = $Sigu / 3.0
term2 = $Sigy * 2.0 / 3.0
Sm = min( $term1, $term2 )
find "ASEL,S,,,1,12" ignore
moveto $Line_Start
moveto line + 1
moveto word + 1
replace word with $Mindex

```

#Setup Flange Condition

```

find "!StartFlangeSetup" ignore
moveto $Line_Start
moveto line + 1
if ( $FlangeCondit == 1 ) {
    moveto $Line_Start
    moveto line + 8
    moveto word + 1
    replace word with $Mindex
    moveto $Line_Start
    moveto line + 9
    moveto word + 1
    replace word with "10"
    moveto $Line_Start
    moveto line + 9
    moveto word + 1
    replace word with "10"
    moveto $Line_Start
    moveto line + 9
    moveto word + 1
    replace word with $Mindex
} elseif ( $FlangeCondit == 2 ) {
    moveto $Line_Start
    moveto line + 8
    moveto word + 1
    replace word with $Mindex
    moveto $Line_Start
    moveto line + 9
    moveto word + 1
    replace word with "10"
    moveto $Line_Start
    moveto line + 9
    moveto word + 1
    replace word with $Mindex
    moveto $Line_Start
    moveto line + 9
    moveto word + 1
    replace word with "10"
} elseif ( $FlangeCondit == 3 ) {
    moveto $Line_Start
    moveto line + 8
    moveto word + 1
    replace word with $Mindex
    moveto $Line_Start
    moveto line + 9
    moveto word + 1
    replace word with "10"

```

```

        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
    } elseif ( $FlangeCondit == 4 ) {
        moveto $Line_Start
        moveto line + 8
        moveto word + 1
        replace word with "10"
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with "10"
    } elseif ( $FlangeCondit == 5 ) {
        moveto $Line_Start
        moveto line + 8
        moveto word + 1
        replace word with "10"
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
    } elseif ( $FlangeCondit == 6 ) {
        moveto $Line_Start
        moveto line + 8
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9
        moveto word + 1
        replace word with $Mindex
        moveto $Line_Start
        moveto line + 9

```

```

        moveto word + 1
        replace word with $Mindex
    }

#Define Loading Type

Line1 = "CSKP,12,0,1001,1005,1004"
Line2 = "CSYS,12"
Line3 = "nrotat,all"
if ( $Load eq "Torsion" ) {
    rotmdir = "x"
    rotvalue = "0.0001*I"
find "! Define Local Coordinate System Here" ignore
    moveto $Line_Start
    moveto line + 1
    write $Line1
    moveto $Line_Start
    moveto line + 1
    write $Line2
    moveto $Line_Start
    moveto line + 1
    write $Line3
} elseif ( $Load eq "OutBen" ) {
    rotmdir = "y"
    rotvalue = "0.0004*I"
find "! Define Local Coordinate System Here" ignore
    moveto $Line_Start
    moveto line + 1
    write $Line1
    moveto $Line_Start
    moveto line + 1
    write $Line2
    moveto $Line_Start
    moveto line + 1
    write $Line3
} elseif ( $Load eq "InClose" ) {
    rotmdir = "y"
    rotvalue = "0.0004*I"
} elseif ( $Load eq "InOpen" ) {
    rotmdir = "y"
    rotvalue = "-0.0004*I"
}
}
Pa = 2 * $Sm * $t / ($d0 - 0.8 * $t)
PValue = $Pres * $Pa
find "SFE,ALL,1,PRES,, " ignore
replace word with $PValue
find "*do,i,1," ignore
if ( $Load eq "Torsion" ) {
    replace word with "400"
}
}
find "ROT" ignore
replace word with $rotmdir
replace word with $rotvalue
#    H = $R * $t / ($r0 * $r0)
provide $Size $Sch $d0 $t $r0 $R $Length $Zeta $Beta

$L2 $x2 $x3

provide $z1 $z2 $z3

```

```

        provide $Pa $PValue
        provide $EpsE $Sm $term1 $term2
    End Instructions
End FileDescription elbowin

FileDescription straightin
FileType: standard
TemplateFile: "straightTemp.in"
InputFile: "straight.in"
InputParameters
    Rad Ang Load Temp Mat Pres H
OutputParameters
    Length d0 t r0 Pa PValue Size Sch Sig0 Eps0
    Sigy Sigu Alphay Alphau a b Sigy0 Sigu0 Sm term1
    term2 EpsE
Instructions
    require Pres H
    require Temp Mat
    require Load
    Sig0[30]
    Eps0[30]
    EpsE[30]
    if ( $H == 0.072143371 ) {
        Size = 8
        Sch = 5
    } elseif ( $H == 0.114494357 ) {
        Size = 6
        Sch = 10
    } elseif ( $H == 0.250378642 ) {
        Size = 6
        Sch = 40
    } elseif ( $H == 0.466689813 ) {
        Size = 4
        Sch = 80
    } elseif ( $H == 0.996845744 ) {
        Size = 2
        Sch = 160
    }
    if ( $Size == 2 ) {
        find "2inch" ignore
        read d0
        if ( $Sch == 5 ) {
            find "Sch5" ignore
            read t
        } elseif ( $Sch == 10 ) {
            find "Sch10" ignore
            read t
        } elseif ( $Sch == 40 ) {
            find "Sch40" ignore
            read t
        } elseif ( $Sch == 80 ) {
            find "Sch80" ignore
            read t
        } elseif ( $Sch == 160 ) {
            find "Sch160" ignore
            read t
        }
    }
}

```

```

} elseif ( $Size == 4 ) {
    find "4inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
}
} elseif ( $Size == 6 ) {
    find "6inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
}
} elseif ( $Size == 8 ) {
    find "8inch" ignore
    read d0
    if ( $Sch == 5 ) {
        find "Sch5" ignore
        read t
    } elseif ( $Sch == 10 ) {
        find "Sch10" ignore
        read t
    } elseif ( $Sch == 40 ) {
        find "Sch40" ignore
        read t
    } elseif ( $Sch == 80 ) {
        find "Sch80" ignore
        read t
    } elseif ( $Sch == 160 ) {
        find "Sch160" ignore
        read t
    }
}
}

```

```

}
Length = 5 * $d0
r0 = ($d0 - $t) / 2.
moveto $File_Start
find "Size =" ignore
replace word with $Size
find "Sch =" ignore
replace word with $Sch
delimiter ", "
find "CSYS,1" ignore
moveto $Line_Start
moveto line + 1
moveto word + 2
replace word with $r0
find "kGEN" ignore
for ( i = 1 ; $i <= 4 ; i = $i + 1 ) {
    moveto $Line_Start
    moveto line + 1
    moveto word + 4
    replace word with $r0
}
find "k,1002,0.,0.," ignore
replace word with $Length
find "R,1," ignore
replace word with $t
if ( $Temp == 0 and $Mat eq "SSCode" ) {
    Mindex = 1
    Mtype = 1
    Sigy = 25000.0
    Sigu = 70000.0
    Sigy0 = 31250.0
    Sigu0 = 87500.0
} elseif ( $Temp == 0 and $Mat eq "SSHHigh" ) {
    Mindex = 1
    Mtype = 1
    Sigy = 50000.0
    Sigu = 140000.0
    Sigy0 = 31250.0
    Sigu0 = 87500.0
} elseif ( $Temp == 0 and $Mat eq "CS" ) {
    Mindex = 1
    Mtype = 2
    Sigy = 35700.0
    Sigu = 58560.0
    Sigy0 = 35700.0
    Sigu0 = 58560.0
} elseif ( $Temp == 0 and $Mat eq "PP" ) {
    Mindex = 1
    Mtype = 3
    Sigy = 50000.0
    Sigu = 140000.0
    Sigy0 = 31250.0
    Sigu0 = 87500.0
}
if ( $Mtype == 1 ) {
    find "Base SS Material Curve" ignore occurrence
}

```

all

all

```
moveto $Line_Start
moveto line + 4
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    read Sig0[$i]
    read Eps0[$i]
    moveto $Line_Start
    moveto line + 1
}
} elseif ( $Mtype == 2 ) {
    find "Base CS Material Curve" ignore occurrence

    moveto $Line_Start
    moveto line + 4
    for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
        read Sig0[$i]
        read Eps0[$i]
        moveto $Line_Start
        moveto line + 1
    }
}
}
provide $Sig0 $Eps0
Alphay = $Sigy / $Sigy0
Alphau = $Sigu / $Sigu0
a = ($Alphau - $Alphay) / (0.35 - 0.002)
b = $Alphay - 0.002 * $a
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    Eps0[$i] = $Eps0[$i] - $Sig0[$i] / 28300000.0
    Sig0[$i] = $Sig0[$i] * ($a * $Eps0[$i] + $b)
    Eps0[$i] = $Eps0[$i] + $Sig0[$i] / 28300000.0
    Sig0[$i] = $Sig0[$i] * (1.0 + $Eps0[$i])
    Eps0[$i] = log( 1.0 + $Eps0[$i] )
}
}
find "! Material #1" ignore occurrence all
moveto $Line_Start
moveto line + 8
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    moveto word + 2
    replace word with $Eps0[$i]
    replace word with $Sig0[$i]
    moveto $Line_Start
    moveto line + 1
}
}
provide $Sig0
provide $Alphay $Alphau $a $b
provide $Sigy0 $Sigu0
term1 = $Sigu / 3.0
term2 = $Sigy * 2.0 / 3.0
Sm = min( $term1, $term2 )
find "ASEL,S,,1,4" ignore
moveto $Line_Start
moveto line + 1
moveto word + 1
replace word with $Mindex
Pa = 2 * $Sm * $t / ($d0 - 0.8 * $t)
PValue = $Pres * $Pa
find "SFE,ALL,1,PRES,," ignore
replace word with $PValue
```

```

        if ( $Load eq "Torsion" ) {
            rotmdir = "z"
            rotvalue = "0.0004*I"
        } else {
            rotmdir = "x"
            rotvalue = "0.0004*I"
        }
        find "ROT" ignore
        replace word with $rotmdir
        replace word with $rotvalue
        provide $Size $Sch $Length $d0 $t $r0
        provide $Pa $PValue
    End Instructions
End FileDescription straightin

FileDescription straightpos
FileType: standard
TemplateFile: "straightTemp.pos"
InputFile: "straight.pos"
Parameters
    Load
Instructions
    require Load
    delimiter ","
    if ( $Load eq "Torsion" ) {
        rotmdir = "z"
    } else {
        rotmdir = "x"
    }
    find "nsol,2,673,rot," ignore
    replace word with $rotmdir
    find "rforce,3,673,m," ignore
    replace word with $rotmdir
End Instructions
End FileDescription straightpos

FileDescription elbowpos
FileType: standard
TemplateFile: "elbowTemp.pos"
InputFile: "elbow.pos"
Parameters
    Load
Instructions
    require Load
    delimiter ","
    if ( $Load eq "OutBen" ) {
        find "nsol,2," ignore
        replace word with "366"
        find "nsol,3," ignore
        replace word with "10"
        find "nsol,4," ignore
        replace word with "366"
        moveto word + 1
        replace word with "x"
        find "nsol,5," ignore
        replace word with "10"
        moveto word + 1
    }

```

```

        replace word with "x"
        find "nsol,6," ignore
        replace word with "809"
        find "nsol,7," ignore
        replace word with "673"
        find "nsol,8," ignore
        replace word with "809"
        moveto word + 1
        replace word with "x"
        find "nsol,9," ignore
        replace word with "673"
        moveto word + 1
        replace word with "x"
        find "nsol,10,1569,rot," ignore
        replace word with "y"
        find "rforce,12,1569,m," ignore
        replace word with "y"
    }
if ( $Load eq "Torsion" ) {
    find "nsol,2," ignore
    replace word with "366"
    find "nsol,3," ignore
    replace word with "10"
    find "nsol,4," ignore
    replace word with "366"
    moveto word + 1
    replace word with "y"
    find "nsol,5," ignore
    replace word with "10"
    moveto word + 1
    replace word with "y"
    find "nsol,6," ignore
    replace word with "809"
    find "nsol,7," ignore
    replace word with "673"
    find "nsol,8," ignore
    replace word with "809"
    moveto word + 1
    replace word with "y"
    find "nsol,9," ignore
    replace word with "673"
    moveto word + 1
    replace word with "y"
    find "nsol,10,1569,rot," ignore
    replace word with "x"
    find "rforce,12,1569,m," ignore
    replace word with "x"
}
    End Instructions
End FileDescription elbowpos
End InputFiles Simcode0

OutputFiles Simcode0
    FileDescription doeout
        FileType: standard
        OutputFile: "doe.out"
        Parameters

```

```

        B2
        Instructions
            find "B2 = " ignore
            read B2
            provide $B2
        End Instructions
    End FileDescription doeout
End OutputFiles Simcode0

SimCodeProcess Simcode0
    ScriptLanguage: DOSBatch
    Script
        ANSYS.bat
    End Script
    ProcessType: transient
    Environment: unrestored
    ElapseTime: 5m
    Prologue
        WriteInputSpecs: elbowin
                        straightin
                        straightpos
                        elbowpos
    Epilogue
        ReadOutputSpecs: doeout
    Execution: "$Program"
End SimCodeProcess Simcode0

End SimCode Simcode0

TaskProcess Task1
    Control: [
        Simcode0
    ]
End TaskProcess Task1

Optimization Task1
    PotentialVariables:
        InputsGroup
    Variables:
        Load H Pres Rad Ang Temp Mat FlangeCondit
    VariableScaling
        Parameter: Pres ScaleFactor: 1.0
        Parameter: Ang ScaleFactor: 1.0
        Parameter: Temp ScaleFactor: 1.0
    InputConstraints
        Parameter: Load Member: InClose OutBen
        Parameter: H Member: 0.250378642 0.072143371 0.114494357
0.466689813 0.996845744
        Parameter: Pres LowerBound: 0.0 UpperBound: 1.0
        Parameter: Ang LowerBound: 0.0 UpperBound: 180.0
        Parameter: Mat Member: SSCode SSHigh CS
    PotentialObjectives:
        OutputsGroup InputsGroup
    OutputConstraints

    OptimizePlan PriorityRankedPlan
        DefaultUpperBound: 1E15

```

```

        UseScaling: yes
        Control: [
        ]
End Optimization Task1

DesignOfExperiments Task1
  Plan NewPlan1
    Technique: "FullFactorial"
    Factors
      ParameterList
        Type: control
        Parameters
          Ang
            Levels: values [ 30.0 90.0 150.0 ]
            BaseLine: 90.0
          H
            Levels: values [ 0.072143371 0.114494357
0.250378642 0.466689813 0.996845744 ]
            BaseLine: 0.250378642
          Load
            Levels: values [ InClose OutBen ]
            BaseLine: InClose
          Pres
            Levels: values [ 0.0 0.618 1.0 ]
            BaseLine: 0.0
        End ParameterList
      End Factors
    End Plan NewPlan1

  Study NewStudy
    Plan: NewPlan1
    Responses
      Outputs:
        B2
    End Responses
    Actions
      Objective: B2
      Direction: maximize
      ReEvaluate: no
    End Actions
    ResultsFile: "doe_Study.NewStudy"
    Prologue
      Tcl
      End Tcl
    Epilogue
      Tcl
      End Tcl
    End Study NewStudy
  End DesignOfExperiments Task1

TaskPlan Task1
  Prologue
    Tcl
      api_GenerateInputFiles Task1 {doe NewStudy} {directory
dir_inp}
    End Tcl
  Control: [

```

```

        NewStudy
    ]
End TaskPlan Task1

DataStorage Task1
  Restore: yes
  DataLog: "doe2.db" Mode: overwrite
  DataLookUp: "doe1.db"
  MatchMode: Exact
  Levels: all
  StoreGradRuns: yes
  StoreApproxRuns: yes
End DataStorage Task1

End Task Task1

```

2. iSIGHT description file for reliability analysis using ABAQUS:

```

MDOLVersion: 9.0
CompilerOptions: warn

```

```

TaskTask1

```

```

  TaskHeader Task1
    Version: 1.0
    Evaluation: sixsigma SixSigmaPlan1
    ControlMode: user
    RunCounter: 1
    BoundsPolicy: adjustvalue
    CheckPoint: unknown
  End TaskHeader Task1

```

```

  Inputs Task1
    Parameter: d0 Type: real InitialValue: 6.0
    Parameter: t Type: real InitialValue: 0.28
    Parameter: Sigy Type: real InitialValue: 45000.0
    Parameter: Sigu Type: real InitialValue: 75000.0
  End Inputs Task1

```

```

  Auxiliaries Task1
    Parameter: Size Type: integer InitialValue: 6
    Parameter: Sch Type: integer InitialValue: 40
    Parameter: Rad Type: discrete InitialValue: Long
    Parameter: Floc Type: integer InitialValue: 5
    Parameter: Ang Type: integer InitialValue: 90
    Parameter: Load Type: discrete InitialValue: InClose
    Parameter: r Type: real InitialValue: 3.0
    Parameter: R Type: integer InitialValue: 0
    Parameter: Length Type: real InitialValue: 30.0
    Parameter: Zeta Type: real InitialValue: 0.0
    Parameter: L2 Type: real InitialValue: 0.0
    Parameter: x1 Type: real InitialValue: 0.0
    Parameter: x2 Type: real InitialValue: 0.0
    Parameter: y2 Type: real InitialValue: 0.0
    Parameter: y5 Type: real InitialValue: 0.0
    Parameter: y6 Type: real InitialValue: 0.0
  End Auxiliaries Task1

```

```

Parameter: x3 Type: real InitialValue: 0.0
Parameter: x4 Type: real InitialValue: 0.0
Parameter: r0 Type: real InitialValue: 0.0
Parameter: Sigy0 Type: real InitialValue: 45000.0
Parameter: Sigu0 Type: real InitialValue: 75000.0
Parameter: Sig0[30] Type: real
      InitialValue: [ 45000.0 45017.262 45048.98229 45074.97686
45112.54457 45158.81486 45171.42857 45205.08514 45286.07486 45445.16829
      45607.14857 45783.59829 45968.74114
46258.02686 47539.73829 49857.504 52189.56686 55151.29543 58859.20971
63810.23829
      68779.18114 75000.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 ]
Parameter: Eps0[30] Type: real
      InitialValue: [ 0.0 0.00020139 0.00057146 0.00087473
0.00131302 0.00185284 0.002 0.00239266 0.00333754 0.00519363
      0.0070834 0.00914198 0.01130198
0.01467698 0.02963028 0.05667088 0.08387828 0.11843178 0.16169078
0.21945278
      0.27742378 0.35 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 ]
Parameter: Alphay Type: real InitialValue: 1.0
Parameter: Alphau Type: real InitialValue: 1.0
Parameter: a Type: real InitialValue: 0.0
Parameter: b Type: real InitialValue: 1.0
Parameter: Status Type: integer InitialValue: 0
Parameter: K[500] Type: real
      InitialValue: [ 0.001 0.002 0.003 0.004 0.005 0.006 0.007
0.008 0.009 0.01 0.011 0.012 0.013 0.014 0.015 0.016 0.017 0.018 0.019
0.02 0.021 0.022 0.023
      0.024 0.025 0.026 0.027 0.028 0.029 0.03
0.031 0.032 0.033 0.034 0.035 0.036 0.037 0.038 0.039 0.04 0.041 0.042
0.043 0.044 0.045 0.046
      0.047 0.048 0.049 0.05 0.051 0.052 0.053
0.054 0.055 0.056 0.057 0.058 0.059 0.06 0.061 0.062 0.063 0.064 0.065
0.066 0.067 0.068 0.069
      0.07 0.071 0.072 0.073 0.074 0.075 0.076
0.077 0.078 0.079 0.08 0.081 0.082 0.083 0.084 0.085 0.086 0.087 0.088
0.089 0.09 0.091 0.092
      0.093 0.094 0.095 0.096 0.097 0.098 0.099
0.1 0.101 0.102 0.103 0.104 0.105 0.106 0.107 0.108 0.109 0.11 0.111
0.112 0.113 0.114 0.115
      0.116 0.117 0.118 0.119 0.12 0.121 0.122
0.123 0.124 0.125 0.126 0.127 0.128 0.129 0.13 0.131 0.132 0.133 0.134
0.135 0.136 0.137 0.138
      0.139 0.14 0.141 0.142 0.143 0.144 0.145
0.146 0.147 0.148 0.149 0.15 0.151 0.152 0.153 0.154 0.155 0.156 0.157
0.158 0.159 0.16 0.161
      0.162 0.163 0.164 0.165 0.166 0.167 0.168
0.169 0.17 0.171 0.172 0.173 0.174 0.175 0.176 0.177 0.178 0.179 0.18
0.181 0.182 0.183 0.184
      0.185 0.186 0.187 0.188 0.189 0.19 0.191
0.192 0.193 0.194 0.195 0.196 0.197 0.198 0.199 0.2 0.201 0.202 0.203
0.204 0.205 0.206 0.207
      0.208 0.209 0.21 0.211 0.212 0.213 0.214
0.215 0.216 0.217 0.218 0.219 0.22 0.221 0.222 0.223 0.224 0.225 0.226
0.227 0.228 0.229 0.23

```

```

0.231 0.232 0.233 0.234 0.235 0.236 0.237
0.238 0.239 0.24 0.241 0.242 0.243 0.244 0.245 0.246 0.247 0.248 0.249
0.25 0.251 0.252 0.253
0.254 0.255 0.256 0.257 0.258 0.259 0.26
0.261 0.262 0.263 0.264 0.265 0.266 0.267 0.268 0.269 0.27 0.271 0.272
0.273 0.274 0.275 0.276
0.277 0.278 0.279 0.28 0.281 0.282 0.283
0.284 0.285 0.286 0.287 0.288 0.289 0.29 0.291 0.292 0.293 0.294 0.295
0.296 0.297 0.298 0.299
0.3 0.301 0.302 0.303 0.304 0.305 0.306
0.307 0.308 0.309 0.31 0.311 0.312 0.313 0.314 0.315 0.316 0.317 0.318
0.319 0.32 0.321 0.322
0.323 0.324 0.325 0.326 0.327 0.328 0.329
0.33 0.331 0.332 0.333 0.334 0.335 0.336 0.337 0.338 0.339 0.34 0.341
0.342 0.343 0.344 0.345
0.346 0.347 0.348 0.349 0.35 0.351 0.352
0.353 0.354 0.355 0.356 0.357 0.358 0.359 0.36 0.361 0.362 0.363 0.364
0.365 0.366 0.367 0.368
0.369 0.37 0.371 0.372 0.373 0.374 0.375
0.376 0.377 0.378 0.379 0.38 0.381 0.382 0.383 0.384 0.385 0.386 0.387
0.388 0.389 0.39 0.391
0.392 0.393 0.394 0.395 0.396 0.397 0.398
0.399 0.4 0.401 0.402 0.403 0.404 0.405 0.406 0.407 0.408 0.409 0.41
0.411 0.412 0.413 0.414
0.415 0.416 0.417 0.418 0.419 0.42 0.421
0.422 0.423 0.424 0.425 0.426 0.427 0.428 0.429 0.43 0.431 0.432 0.433
0.434 0.435 0.436 0.437
0.438 0.439 0.44 0.441 0.442 0.443 0.444
0.445 0.446 0.447 0.448 0.449 0.45 0.451 0.452 0.453 0.454 0.455 0.456
0.457 0.458 0.459 0.46
0.461 0.462 0.463 0.464 0.465 0.466 0.467
0.468 0.469 0.47 0.471 0.472 0.473 0.474 0.475 0.476 0.477 0.478 0.479
0.48 0.481 0.482 0.483
0.484 0.485 0.486 0.487 0.488 0.489 0.49
0.491 0.492 0.493 0.494 0.495 0.496 0.497 0.498 0.499 0.5 ]
Parameter: M[500] Type: real
InitialValue: [ 24099.0 48198.0 72296.0 96395.0 120490.0
144590.0 168690.0 192780.0 216880.0 240950.0 262130.0 276530.0 286630.0
294870.0 301620.0 306200.0 310080.0 313810.0 317360.0 320710.0 323500.0
325520.0 327160.0
328760.0 330290.0 331760.0 333160.0
334530.0 335890.0 337220.0 338510.0 339780.0 341020.0 342250.0 343450.0
344640.0 345820.0 346980.0 348060.0 349050.0 349920.0 350660.0 351360.0
352050.0 352720.0 353380.0
354010.0 354630.0 355220.0 355800.0
356370.0 356920.0 357470.0 358010.0 358540.0 359070.0 359590.0 360110.0
360630.0 361130.0 361640.0 362130.0 362620.0 363100.0 363580.0 364040.0
364500.0 364960.0 365410.0
365860.0 366310.0 366760.0 367200.0
367640.0 368080.0 368520.0 368950.0 369390.0 369820.0 370250.0 370680.0
371100.0 371520.0 371940.0 372350.0 372760.0 373170.0 373570.0 373970.0
374360.0 374750.0 375140.0
375520.0 375890.0 376270.0 376640.0
377010.0 377380.0 377740.0 378100.0 378460.0 378820.0 379170.0 379530.0
379870.0 380220.0 380570.0 380910.0 381250.0 381590.0 381920.0 382250.0
382580.0 382910.0 383240.0

```

383560.0 383880.0 384190.0 384500.0
 384810.0 385120.0 385420.0 385710.0 386010.0 386300.0 386580.0 386860.0
 387140.0 387420.0 387690.0 387960.0 388230.0 388490.0 388750.0 389010.0
 389260.0 389520.0 389760.0
 390010.0 390250.0 390490.0 390730.0
 390970.0 391210.0 391440.0 391670.0 391890.0 392120.0 392340.0 392570.0
 392790.0 393010.0 393230.0 393450.0 393670.0 393890.0 394110.0 394320.0
 394540.0 394760.0 394970.0
 395180.0 395400.0 395610.0 395820.0
 396030.0 396240.0 396440.0 396650.0 396860.0 397060.0 397270.0 397470.0
 397670.0 397880.0 398080.0 398280.0 398470.0 398670.0 398870.0 399070.0
 399270.0 399470.0 399660.0
 399860.0 400050.0 400250.0 400440.0
 400630.0 400830.0 401020.0 401210.0 401400.0 401590.0 401780.0 401970.0
 402160.0 402350.0 402540.0 402720.0 402910.0 403100.0 403280.0 403470.0
 403650.0 403840.0 404020.0
 404210.0 404390.0 404570.0 404760.0
 404940.0 405120.0 405300.0 405480.0 405670.0 405850.0 406030.0 406210.0
 406390.0 406570.0 406750.0 406920.0 407100.0 407280.0 407460.0 407630.0
 407810.0 407980.0 408160.0
 408330.0 408510.0 408680.0 408850.0
 409020.0 409190.0 409360.0 409530.0 409690.0 409860.0 410030.0 410190.0
 410360.0 410520.0 410680.0 410850.0 411010.0 411170.0 411330.0 411490.0
 411650.0 411810.0 411970.0
 412120.0 412280.0 412440.0 412590.0
 412750.0 412900.0 413060.0 413210.0 413360.0 413520.0 413670.0 413820.0
 413970.0 414120.0 414270.0 414420.0 414560.0 414710.0 414860.0 415000.0
 415150.0 415300.0 415440.0
 415580.0 415730.0 415870.0 416010.0
 416150.0 416290.0 416430.0 416570.0 416710.0 416840.0 416980.0 417120.0
 417250.0 417390.0 417520.0 417650.0 417790.0 417920.0 418050.0 418180.0
 418310.0 418440.0 418570.0
 418700.0 418820.0 418950.0 419070.0
 419200.0 419320.0 419450.0 419570.0 419690.0 419820.0 419940.0 420060.0
 420180.0 420300.0 420410.0 420530.0 420650.0 420760.0 420880.0 420990.0
 421110.0 421220.0 421330.0
 421440.0 421550.0 421660.0 421770.0
 421880.0 421980.0 422090.0 422190.0 422300.0 422400.0 422500.0 422600.0
 422700.0 422800.0 422890.0 422990.0 423090.0 423180.0 423280.0 423370.0
 423460.0 423550.0 423640.0
 423730.0 423820.0 423900.0 423990.0
 424070.0 424150.0 424230.0 424310.0 424390.0 424470.0 424540.0 424620.0
 424690.0 424760.0 424830.0 424900.0 424970.0 425040.0 425100.0 425170.0
 425230.0 425290.0 425350.0
 425410.0 425460.0 425520.0 425570.0
 425620.0 425680.0 425720.0 425770.0 425810.0 425850.0 425890.0 425930.0
 425970.0 426000.0 426040.0 426080.0 426120.0 426150.0 426180.0 426210.0
 426230.0 426260.0 426280.0
 426300.0 426320.0 426340.0 426360.0
 426370.0 426370.0 426380.0 426380.0 426390.0 426400.0 426400.0 426400.0
 426400.0 426390.0 426390.0 426380.0 426370.0 426360.0 426350.0 426340.0
 426320.0 426300.0 426280.0
 426250.0 426230.0 426200.0 426170.0
 426140.0 426110.0 426080.0 426040.0 426000.0 425960.0 425910.0 425870.0
 425820.0 425770.0 425720.0 425660.0 425600.0 425540.0 425480.0 425410.0
 425350.0 425280.0 425210.0

```

425130.0 425050.0 424980.0 424900.0
424810.0 424730.0 424640.0 424550.0 424460.0 424360.0 424260.0 424160.0
424060.0 423950.0 423850.0 423740.0 423620.0 423510.0 423380.0 423260.0
423140.0 423010.0 422880.0
422750.0 422610.0 422470.0 422330.0
422180.0 422040.0 421890.0 421740.0 421580.0 421420.0 421260.0 421100.0
420930.0 420760.0 420590.0 420410.0 420230.0 420050.0 419860.0 419670.0
419480.0 419290.0 419090.0
418890.0 418680.0 418470.0 418260.0
418050.0 417830.0 417610.0 417380.0 417160.0 416920.0 416690.0 416450.0
416210.0 415960.0 415710.0 415460.0 415200.0 ]
Parameter: tmp Type: real InitialValue: 6024750.0
Parameter: Last Type: real InitialValue: 415200.0
Parameter: MK Type: real InitialValue: 12049500.0
Parameter: tmp1 Type: real InitialValue: 337386.0
Parameter: tmp2 Type: real InitialValue: 334530.0
Parameter: up Type: real InitialValue: 10718195.0
Parameter: dw Type: real InitialValue: 10679.5
Parameter: H Type: real InitialValue: 0.0
End Auxiliaries Task1

Outputs Task1
Parameter: Mc Type: real
Parameter: M2 Type: real
Parameter: Mccode Type: real
Parameter: Margin1 Type: real
Parameter: Margin2 Type: real
Parameter: lamda Type: real
End Outputs Task1

Initialization Task1
Parameters
t d0 Mccode
Tcl
set D [api_GetParameterValue Task1 d0]
puts " D = $D "
set T [api_GetParameterValue Task1 t]
puts " T = $T "
set SY 35000.0
puts " SY = $SY "
set SU 60000.0
puts " SU = $SU "
set term1 [expr $SU/4.0]
set term2 [expr $SY*2.0/3.0]
set Sm $term1
if {$term2 < $term1} {
    set Sm $term2
}

set Di [expr $D-2.0*$T]

set I [expr 3.141592654*(pow($D,4)-pow($Di,4))/64.0]

puts " I = $I "

# Level C
set M [expr 2.25*$Sm*2*$I/$D]

```

```

# Level D
#           set M [expr 3.0*$Sm*2*$I/$D]

puts " M = $M "

    api_SetParameterValue Task1 Mccode $M
End Tcl
End Initialization Task1

SimCode Simcode0
InputFiles Simcode0
  FileDescription straightinp
  FileType: standard
  TemplateFile: "straightTemp.inp"
  InputFile: "straight.inp"
  InputParameters
    Size Sch Floc Load d0 t Sigy Sigu
  OutputParameters
    Length r Sig0 Eps0 Alphas Alpha a b Sigy0 Sigu0
    Mccode
  Instructions
    require Floc d0 t
    require Size Sch Load Sigy Sigu
    Sig0[30]
    Eps0[30]
    Sigy0 = 45000.0
    Sigu0 = 75000.0
    r = $d0 / 2.
    Length = $Floc * $d0
    moveto $File_Start
    find "size:" ignore
    replace word with $Size
    find "sch: " ignore
    replace word with $Sch
    find "Loading:" ignore
    replace word with $Load
    delimiter ","
    find "*NODE, NSET=noden1" ignore
    moveto $Line_Start
    moveto line + 1
    moveto word + 1
    replace word with $Length
    find "*BEAM SECTION" ignore
    moveto $Line_Start
    moveto line + 1
    replace word with $r
    replace word with $t
    moveto $Line_Start
    moveto line + 1
    replace word with $Length
    if ( $Load eq "Torsion" ) {
        rotmdir = 4
        rotvalue = 2.0
        Uout = "UR1"
        Fout = "RM1"
    } elseif ( $Load eq "OutBen" ) {
        rotmdir = 5

```

```

        rotvalue = 2.0
        Uout = "UR2"
        Fout = "RM2"
    } elseif ( $Load eq "InClose" ) {
        rotmdir = 6
        rotvalue = - 0.8
        Uout = "UR3"
        Fout = "RM3"
    } elseif ( $Load eq "InOpen" ) {
        rotmdir = 6
        rotvalue = 2.0
        Uout = "UR3"
        Fout = "RM3"
    }
find "*STEP,INC=10000,NLGEOM" ignore
replace word with $Load
find "*BOUNDARY" ignore
moveto $Line_Start
moveto line + 1
moveto word + 1
replace word with $rotmdir
replace word with $rotvalue
find "*NODE PRINT" ignore occurrence all
moveto $Line_Start
moveto line + 1
replace word with $Uout
replace word with $Fout
find "*** Bilinear Kinematic" ignore occurrence all
provide $Length $r
moveto $Line_Start
moveto line + 1
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    moveto column + 2
    read Sig0[$i]
    read Eps0[$i]
    moveto $Line_Start
    moveto line + 1
}
provide $Sig0 $Eps0
lamda = 1.0/(1.0 + 0.35*29000000.0/($Sigu - $Sigy))
Alphay = $Sigy / $Sigy0
Alphau = $Sigu / $Sigu0
a = ($Alphau - $Alphay) / 0.35
b = $Alphay
for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
    Sig0[$i] = $Sig0[$i] * ($a * $Eps0[$i] + $b)
    Eps0[$i] = $Eps0[$i] + $Sig0[$i] / 29000000.0
    Sig0[$i] = $Sig0[$i] * (1.0 + $Eps0[$i])
    Eps0[$i] = log( 1.0 + $Eps0[$i] )
    Eps0[$i] = $Eps0[$i] - $Sig0[$i] / 29000000.0
    if ( $Eps0[$i] < 0.0 ) {
        Eps0[$i] = 0.0
    }
}
}
find "*PLASTIC" ignore occurrence all
moveto $Line_Start
moveto line + 1

```

```

        for ( i = 0 ; $i <= 21 ; i = $i + 1 ) {
            replace word with $Sig0[$i]
            replace word with $Eps0[$i]
            moveto $Line_Start
            moveto line + 1
        }
        provide $Sig0
        provide $Alphay $Alphau $a $b
        provide $Sigy0 $Sigu0 $lamda
    End Instructions
End FileDescription straightinp
End InputFiles Simcode0

OutputFiles Simcode0
FileDescription straightdat
FileType: standard
OutputFile: "straight.dat"
InputParameters
    Mccode
OutputParameters
    Mc M2 Margin1 Margin2 Status K M tmp Last MK
    tmp1 tmp2 up dw
Instructions

#Define Array and Get Parameter "Mccode"

require Mccode
    find "STEP TIME COMPLETED" ignore occurrence

all

    read time
    II = $time * 500 - 1
    moveto $File_Start
    K[500]
    M[500]
    Mc = 0

#Read in Moment-Curvature Data

    for ( i = 0 ; $i <= $II ; i = $i + 1 ) {
        if ( $i == 0 ) {
            moveto line + 272
        } else {
            moveto line + 27
        }
        moveto word + 1
        read tmp
        K[$i] = abs( $tmp )
        read tmp
        M[$i] = abs( $tmp )
        if ( $Mc < abs( $tmp ) ) {
            Mc = abs( $tmp )
        }
    }

#Find Ultimate Moment if It Exists

```

```

Last = $M[$II]
I3 = $II - 1
M3 = $M[$I3]
tmp = $Last - $M3
diff = abs( $tmp )
if ( ($Last < $Mc) or ($diff < 0.01) ) {
    Status = 0
} else {
    Status = 1
}

#Calculate Twice Elastic Moment

MK = ($M[1] - $M[0]) / ($K[1] - $K[0]) / 2.
tmp = $MK * $K[$II]
if ( $tmp < $Last ) {
    M2 = 0
} else {
    j = 1
    get = 0
    JJ = $II + 1
    while ( ($get == 0) and ($j < $JJ) ) {
        tmp1 = $MK * $K[$j]
        tmp2 = $M[$j]
        if ( $tmp1 > $tmp2 ) {
            j1 = $j - 1
            up = ($M[$j] - $M[$j1]) * ($M[$j1]
- $MK * $K[$j1])
            dw = $MK * $K[$j] - $M[$j] +
$M[$j1] - $MK * $K[$j1]
            M2 = $M[$j1] + $up / $dw
            get = 1
        } else {
            j = $j + 1
        }
    }
}

#Calculate Margins for Two Failure Criteria

Margin1 = $Mc - $Mccode
Margin2 = $M2 - $Mccode
provide $Mc $M2 $Margin1 $Margin2 $Status
provide $K $M $tmp $Last $MK $tmp1 $tmp2 $up
$dw

provide $II $JJ
provide $I3 $M3 $diff
End Instructions
End FileDescription straightdat
End OutputFiles Simcode0

SimCodeProcess Simcode0
Program: "C:/ABAQUS/Commands/abaqus.bat"
ProcessType: transient
Environment: unrestored
ReturnCodes: 0
1

```

```

        ElapseTime: 3h 20m
        Prologue
            WriteInputSpecs: straightinp
        Epilogue
            ReadOutputSpecs: straightdat
            Execution: "$Program job=straight interactive"
        End SimCodeProcess Simcode0

End SimCode Simcode0

TaskProcess Task1
    Epilogue
        Tcl
            set st [api_GetParameterValue Task1 Status]
            if { $st > 0 } {
                api_SetTaskProcessStatus Task1 bad
            }
        End Tcl
    Control: [
        Simcode0
    ]
End TaskProcess Task1

Optimization Task1
    PotentialVariables:
        d0 t Sigy Sigu
    Variables:
        d0 t Sigy Sigu
    VariableScaling
        Parameter: d0 ScaleFactor: 1.0
        Parameter: t ScaleFactor: 1.0
        Parameter: Sigy ScaleFactor: 1.0
        Parameter: Sigu ScaleFactor: 1.0
    PotentialObjectives:
        Mc M2 Mccode Margin1 Margin2 d0 t Sigy Sigu
    Objectives
        Parameter: Margin1 Direction: maximize Weight: 1.0
ScaleFactor: 1.0
        Parameter: Margin2 Direction: maximize Weight: 1.0
ScaleFactor: 1.0

    # PLAN TO BE CONFIGURED BY ADVISOR:
    OptimizePlan PriorityRankedPlan
        Control: [
        ]
End Optimization Task1

QualityEngineeringMethods Task1
    SixSigmaRobustDesign SixSigmaPlan1
        PlanType: analysis
        DesignVariables
            QualityType: sigmalevel
            Parameter: d0
                QualityLevel: 6.0
            Parameter: t
                QualityLevel: 6.0
            Parameter: Sigy

```

```

        QualityLevel: 6.0
    Parameter: Sigu
        QualityLevel: 6.0
End DesignVariables
RandomVariables
    Parameter: d0
        DistributionType: lognormal
        MeanValue: 6.0
        CoeffVariation: 0.0625
        Fixed: CoeffVar
    Parameter: t
        DistributionType: lognormal
        MeanValue: 0.28
        CoeffVariation: 0.0625
        Fixed: CoeffVar
    Parameter: Sigy
        DistributionType: lognormal
        MeanValue: 45000.0
        CoeffVariation: 0.1
        Fixed: CoeffVar
    Parameter: Sigu
        DistributionType: lognormal
        MeanValue: 75000.0
        CoeffVariation: 0.1
        Fixed: CoeffVar
End RandomVariables
QualityConstraints
    QualityType: sigmalevel
    Parameter: Mc
        LowerBound: 232059.0
        QualityLevel: 6.0
    Parameter: M2
        LowerBound: 232059.0
        QualityLevel: 6.0
    Parameter: Margin1
        LowerBound: 0.0
        QualityLevel: 6.0
    Parameter: Margin2
        LowerBound: 0.0
        QualityLevel: 6.0
End QualityConstraints
RobustObjective
    RobustnessMetric: StdDeviation
    Parameter: Margin1
        Components: MeanAndRobustness
        MeanDirection: maximize
        MeanWeight: 1.0
        MeanScale: 1.0
        RobustnessWeight: 1.0
        RobustnessScale: 1.0
    Parameter: Margin2
        Components: MeanAndRobustness
        MeanDirection: maximize
        MeanWeight: 1.0
        MeanScale: 1.0
        RobustnessWeight: 1.0
        RobustnessScale: 1.0

```

```
End RobustObjective
SamplingMethod: MCS
Options
  SamplingTechnique: Descriptive
  NumberOfSimulations: 500
  RunMeanValuePoint: no
  ResultsFile: "SixSigmaPlan1.ss"
End SixSigmaRobustDesign SixSigmaPlan1
End QualityEngineeringMethods Task1

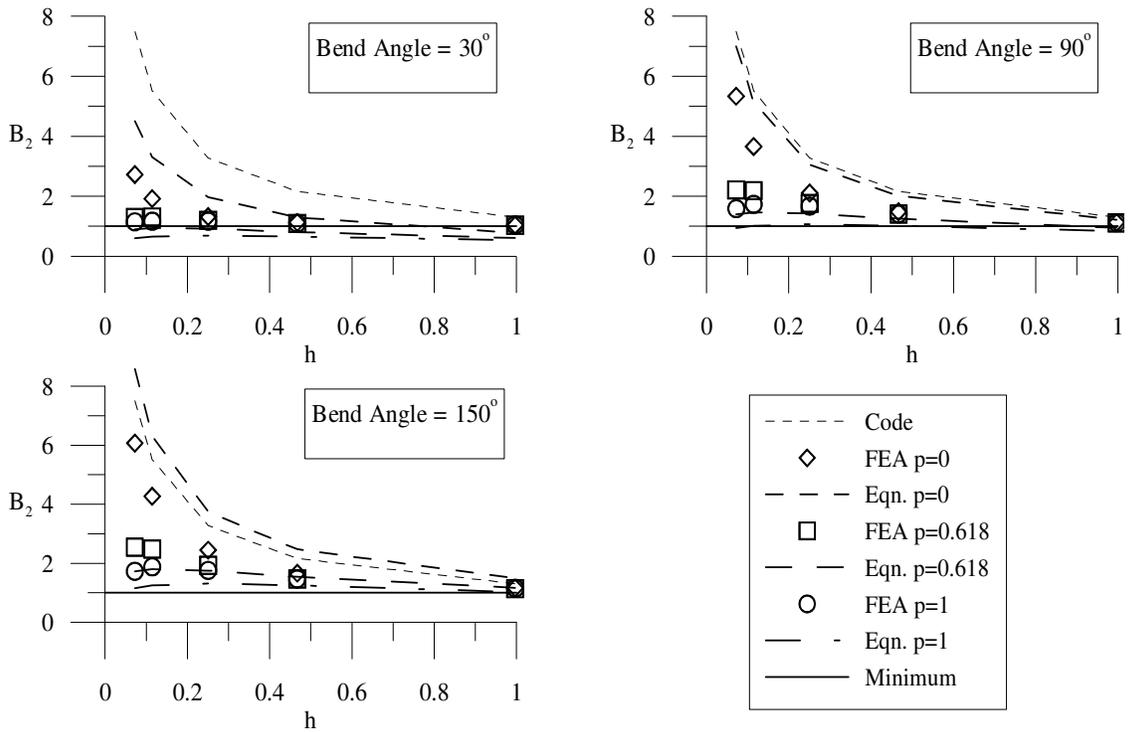
TaskPlan Task1
  StopTaskPlanOnError: no
  Control: [
    SixSigmaPlan1
  ]
End TaskPlan Task1

DataStorage Task1
  Restore: no
  DataLog: "straight.db" Mode: overwrite
  DataLookUp: "Task1.db"
  MatchMode: Exact
  Levels: all
  StoreGradRuns: yes
  StoreApproxRuns: yes
End DataStorage Task1

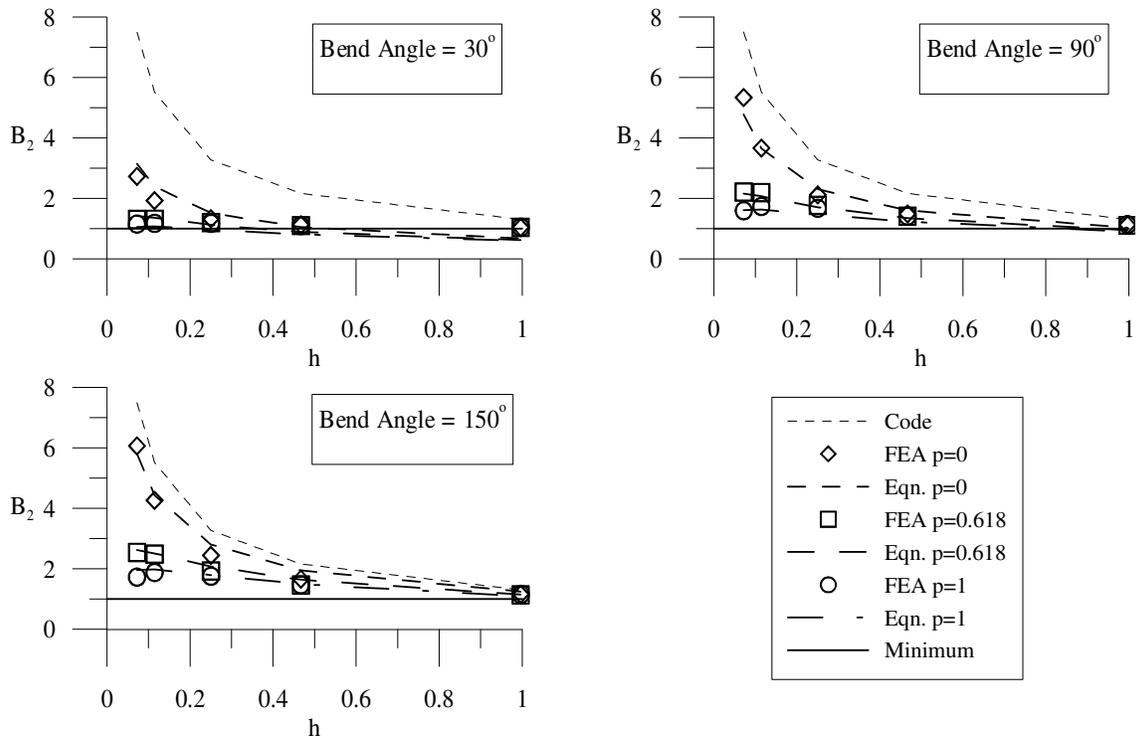
End Task Task1
```

APPENDIX B

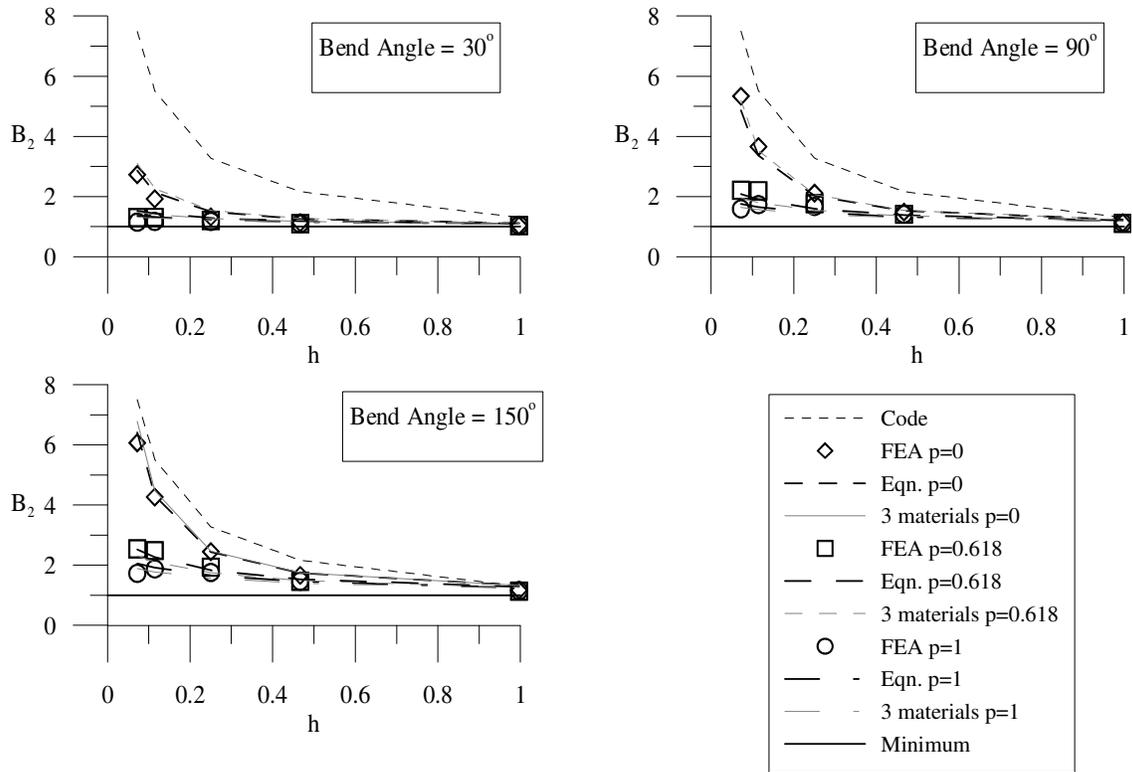
Reconciliations between Finite Element Analysis Results and B_2 Equations



Touboul and Acker equation for SS 304L



Optimized Touboul and Acker equation for SS 304L



NCSU equation for SS 304L and the combined set of materials (labeled 3 materials.)

Excel Worksheet for Curve-Fitting

NCSU							
Code Stainless				one	1		1
				c_1	0.23397327		0.36
				c_2	0.44969151		0.6
				e_1	1.06599941		1.1
				e_2	0.66310513		0.63
				Sum Sq Err	1.20084085		1.795
one+c_1/H^e_1*(alpha/90)^e_2/(1+c_2*PP*r_m/(h*t*Sy))							
				Equation			

APPENDIX C

Sample Pipe Table (unit: inch)

Schedule Size	Sch 5	Sch 10	Sch 40	Sch 80	Sch 160
2 inch	$D_o = 2.375$ $t = 0.065$	$D_o = 2.375$ $t = 0.109$	$D_o = 2.375$ $t = 0.154$	$D_o = 2.375$ $t = 0.218$	$D_o = 2.375$ $t = 0.343$
4 inch	$D_o = 4.5$ $t = 0.083$	$D_o = 4.5$ $t = 0.12$	$D_o = 4.5$ $t = 0.237$	$D_o = 4.5$ $t = 0.337$	$D_o = 4.5$ $t = 0.531$
6 inch	$D_o = 2.375$ $t = 0.109$	$D_o = 2.375$ $t = 0.134$	$D_o = 2.375$ $t = 0.28$	$D_o = 2.375$ $t = 0.432$	$D_o = 2.375$ $t = 0.718$
8 inch	$D_o = 2.375$ $t = 0.109$	$D_o = 2.375$ $t = 0.148$	$D_o = 2.375$ $t = 0.322$	$D_o = 2.375$ $t = 0.5$	$D_o = 2.375$ $t = 0.906$