**MAE 190 Shaft Design Project**

A Simple Circular Rotating

Shaft Design Problem

Author: Winston Chou

**Introduction:**

**Nature of the Problem:**

Designing a circular shaft that satisfies both strength and fatigue life requirements under dynamic loading conditions is critical in mechanical systems. This involves calculating the appropriate diameter to resist stresses due to bending moments and torsion while considering safety factors and failure criteria.

**Origin of the Equations Used:**

The equations originate from fatigue theory and static stress analysis, incorporating criteria like:

* Modified Goodman
* DE-Gerber
* DE-ASME Elliptic
* DE-Soderberg

These criteria are widely applied in fatigue design to ensure safety under varying load conditions.

**Challenges in the Project:**

* Accurate computation of the stress concentration factor based on geometry and loading conditions.
* Reliable determination of the endurance limit considering surface, temperature, size, and reliability factors.
* Ensuring convergence in iterative diameter calculations under the trial-and-error approach.

**Review of Similar Methods:**

* Traditional methods often involve manual trial-and-error or limited computational tools with rigid assumptions.
* This project introduces an automated approach to improve efficiency and accuracy.

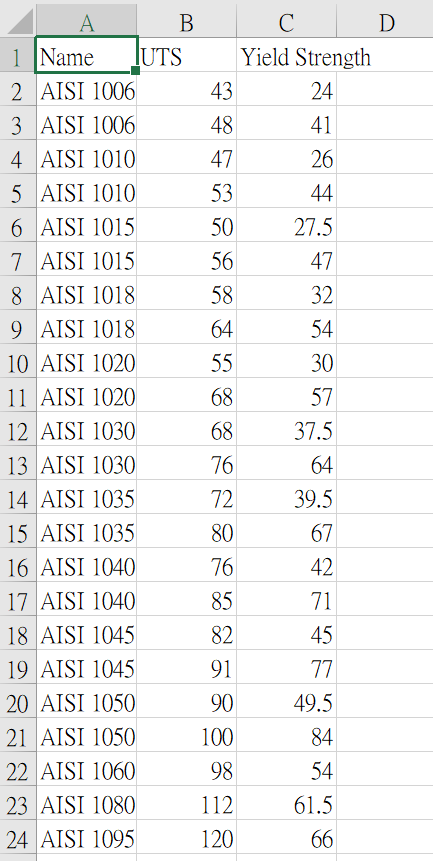
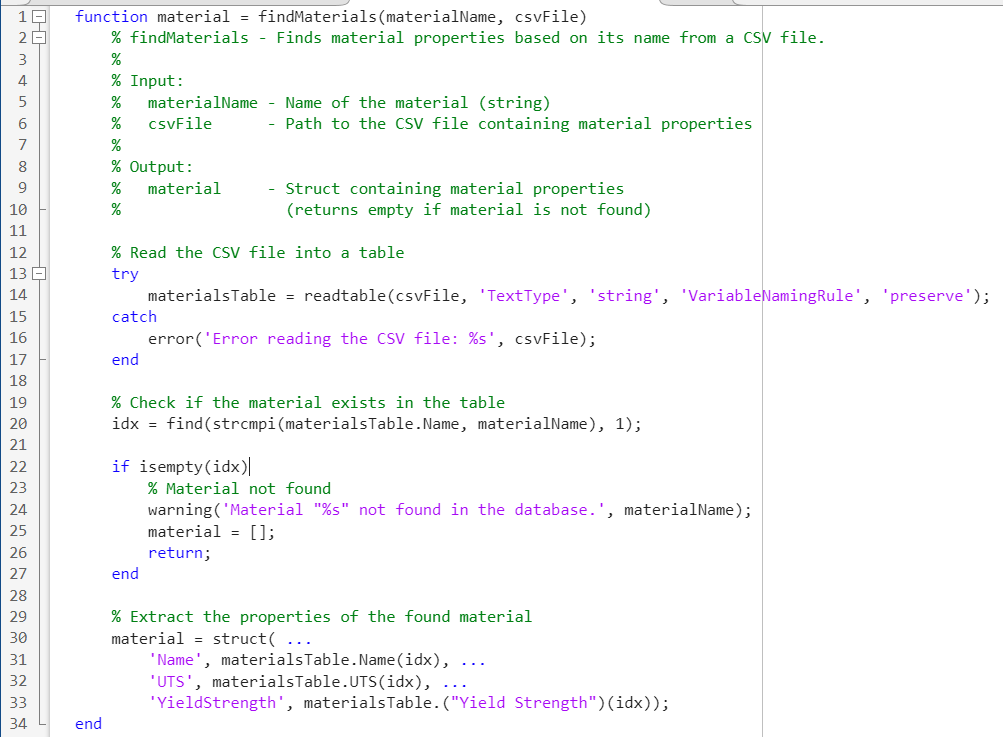
**Approach:**

The code integrates all relevant material properties and failure criteria into a single MATLAB-based framework. It employs interpolation for stress concentration factors, systematic material property retrieval, and a robust iteration loop for diameter calculation.

**Results:**

**Algorithm and Code Sections:**

* **Material Selection:** Functionality to retrieve material properties (e.g., ultimate tensile strength, yield strength) from a CSV database.

  
Figure 1: findMaterial.m (Left) and the material\_database.csv (Right, Unit: ksi)

* **Stress Concentration and Geometry Consideration:** Linear interpolation is used to determine stress concentration factors based on and ratios. Using given Figure C-2, C-3 to approximate the Stress-Concentration Factors (, ), and then derive the Fatigue Stress-Concentration Factors (, )

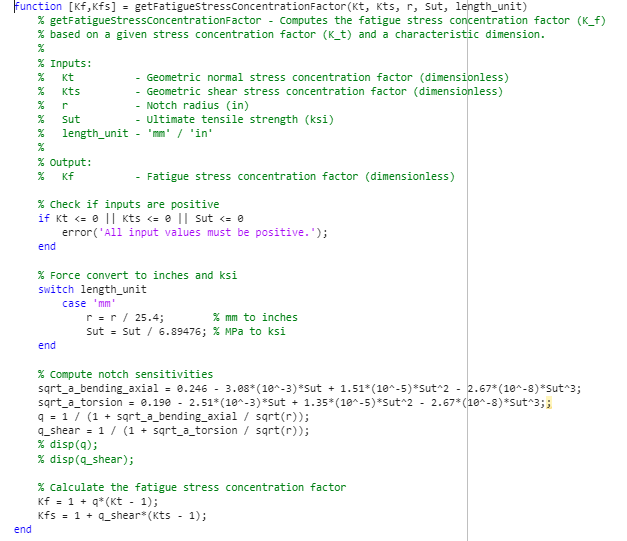
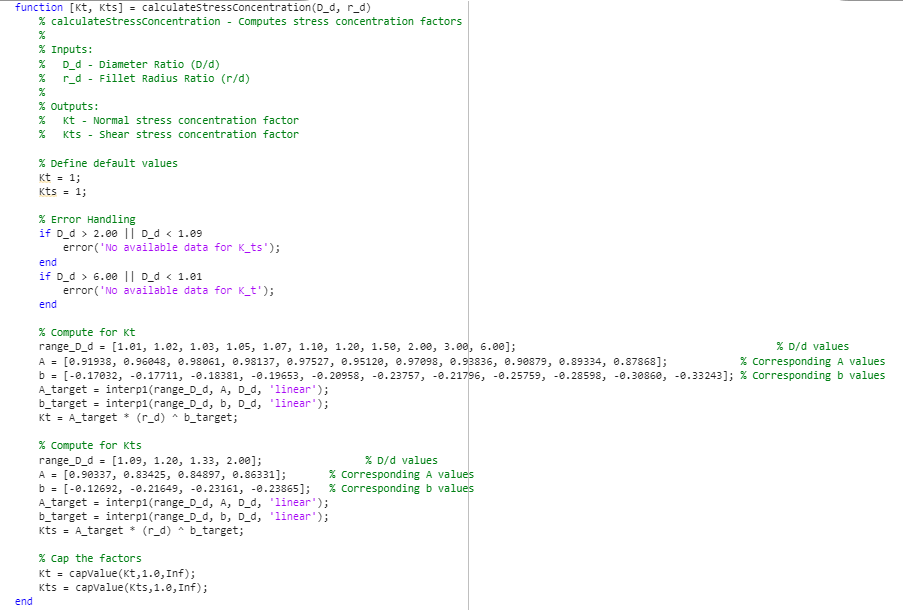


Figure 2: calculateStressConcentration.m (Left) and getFatigueStressConcentrationFactor.m (Right)

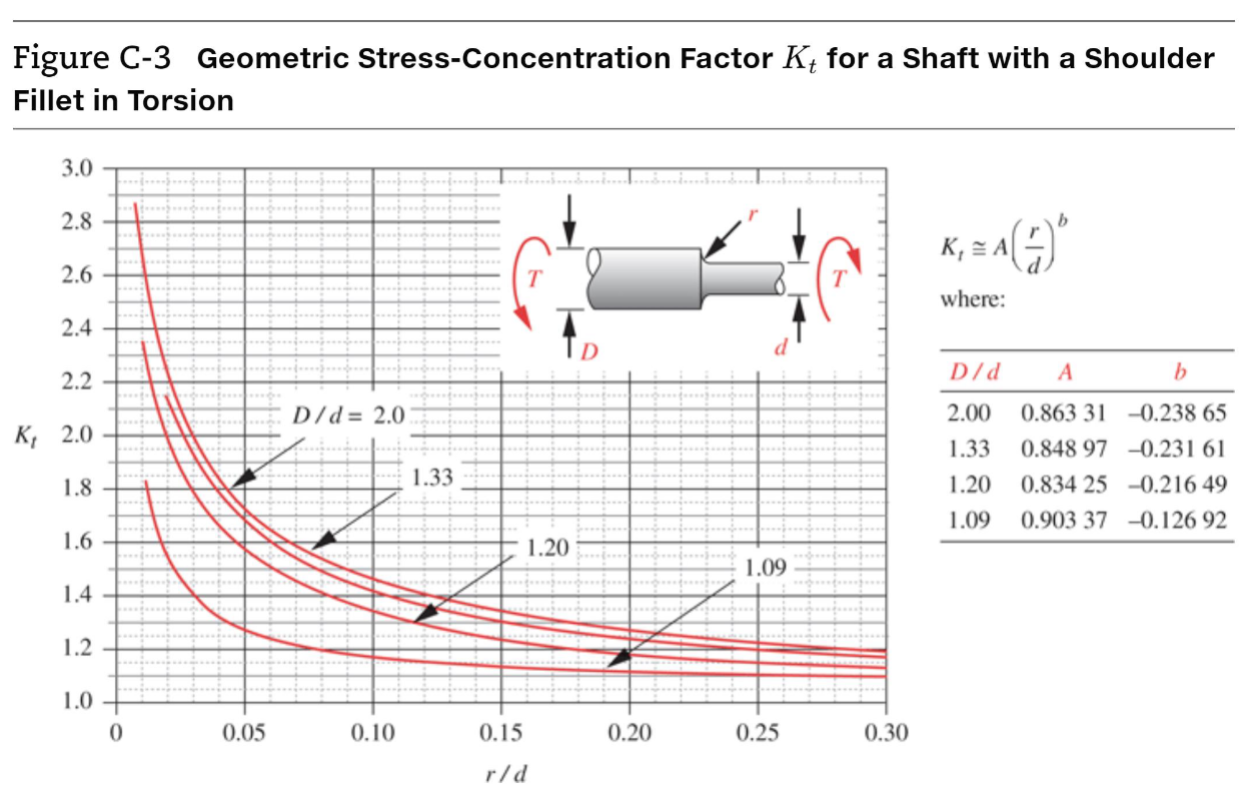
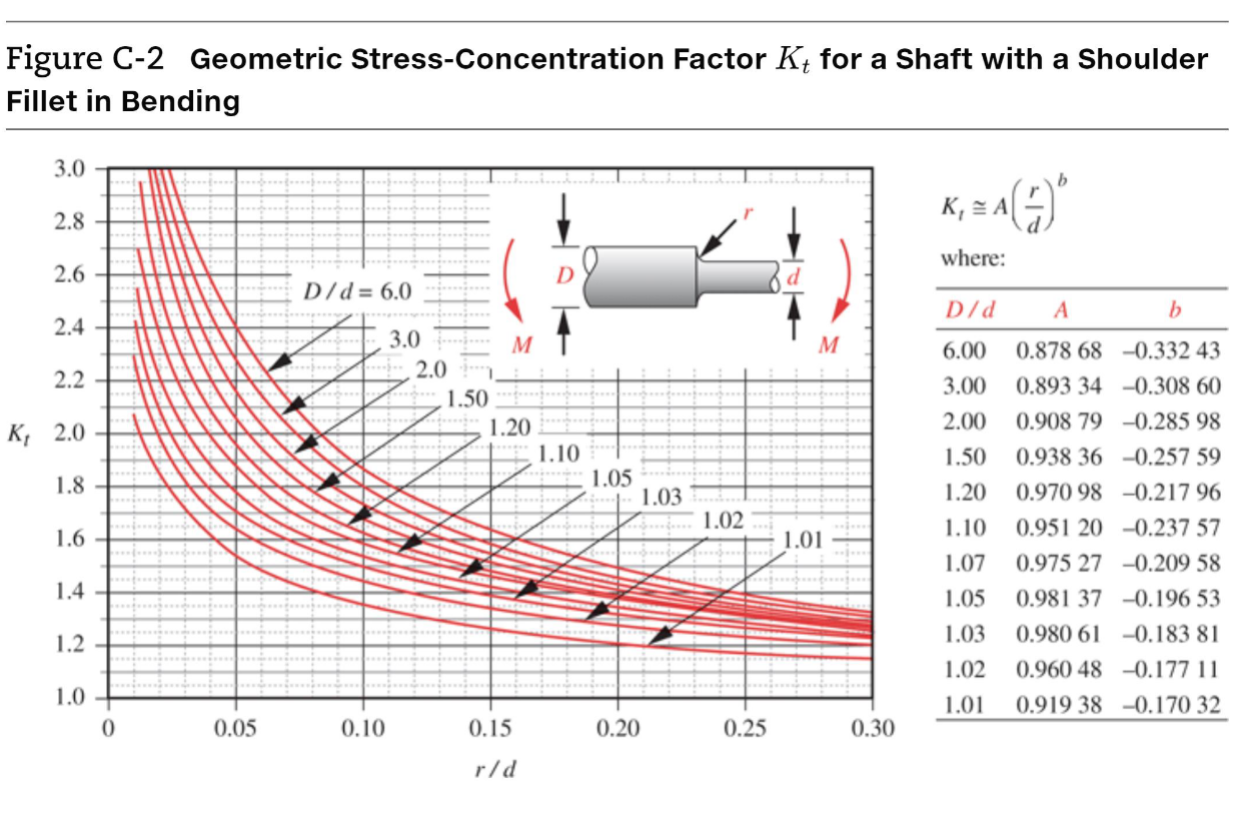


Figure 3: Geometric Stress-Concentration Factors for a Shaft (Figure C-2, C-3)

* **Endurance Limit Determination:** Incorporates surface, temperature, size, and reliability modifying factors. Loading Marin Factor, , since it would go for von-mises stresses when analysis.

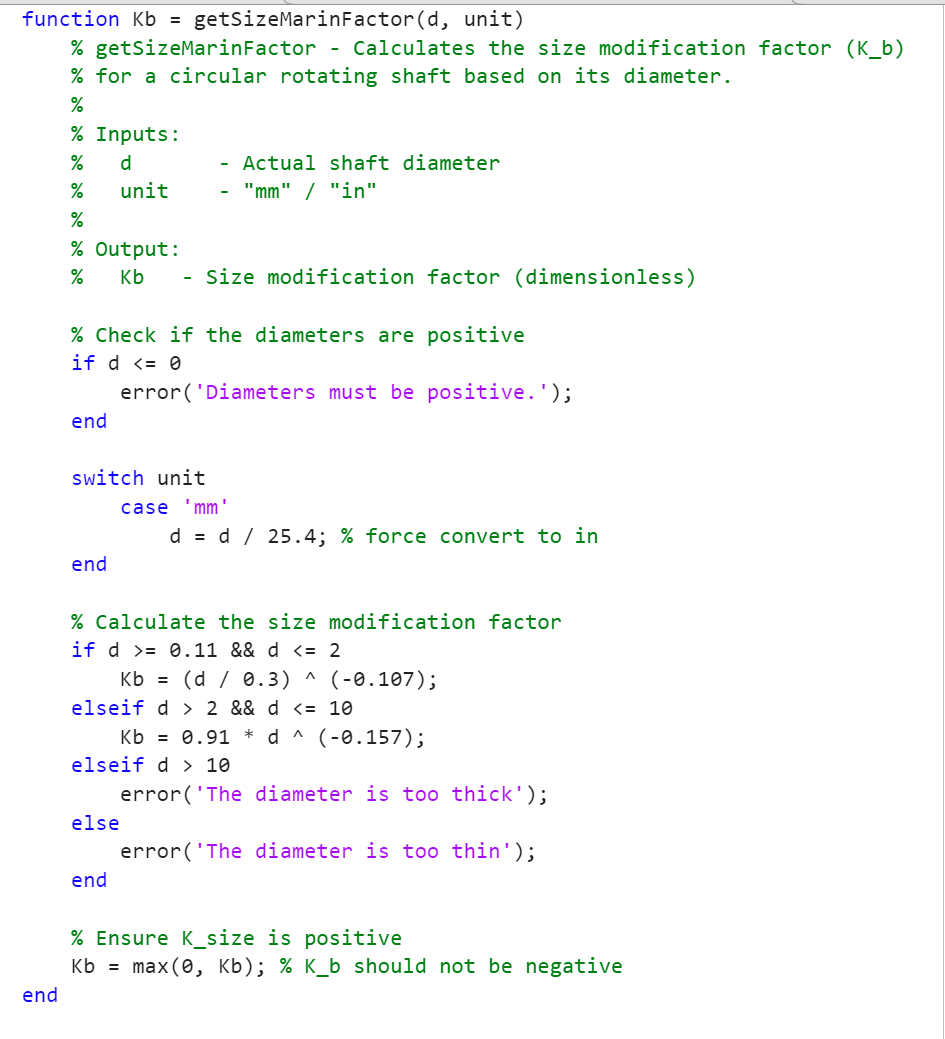
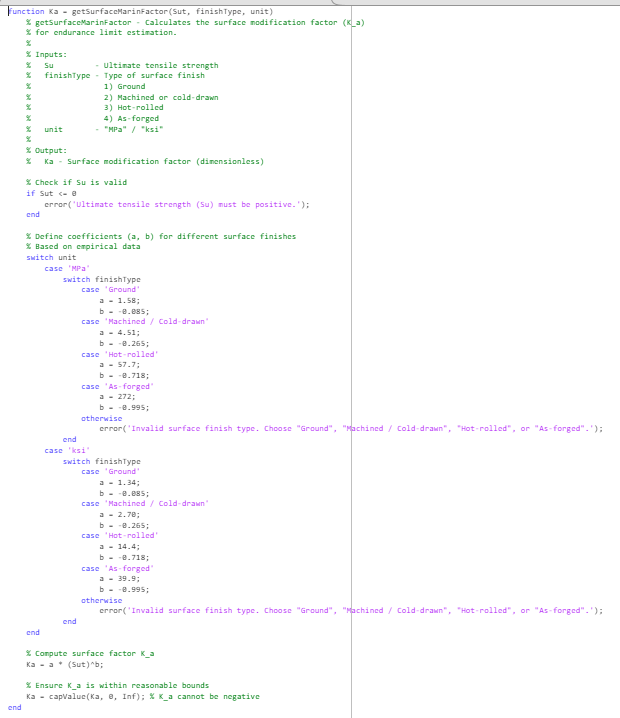


Figure 4: getSurfaceMarinFactor.m and getSizeMarinFactor.m

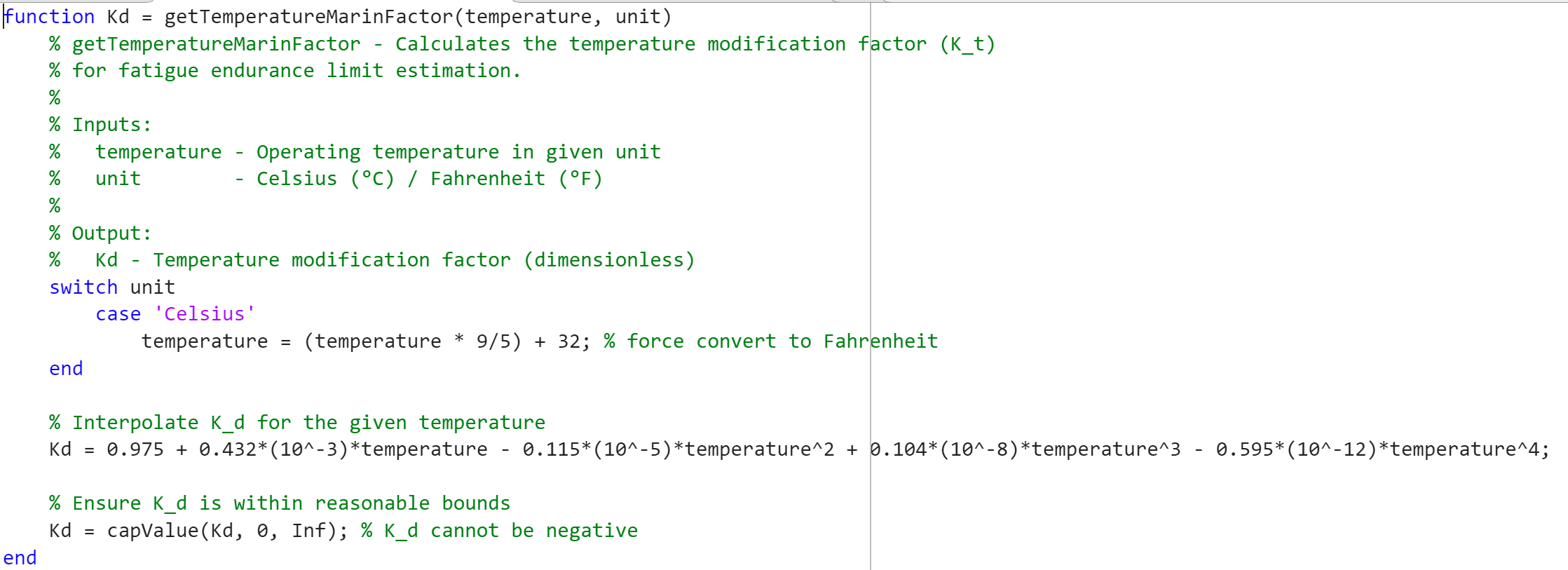


Figure 5: getTemperatureMarinFactor.m

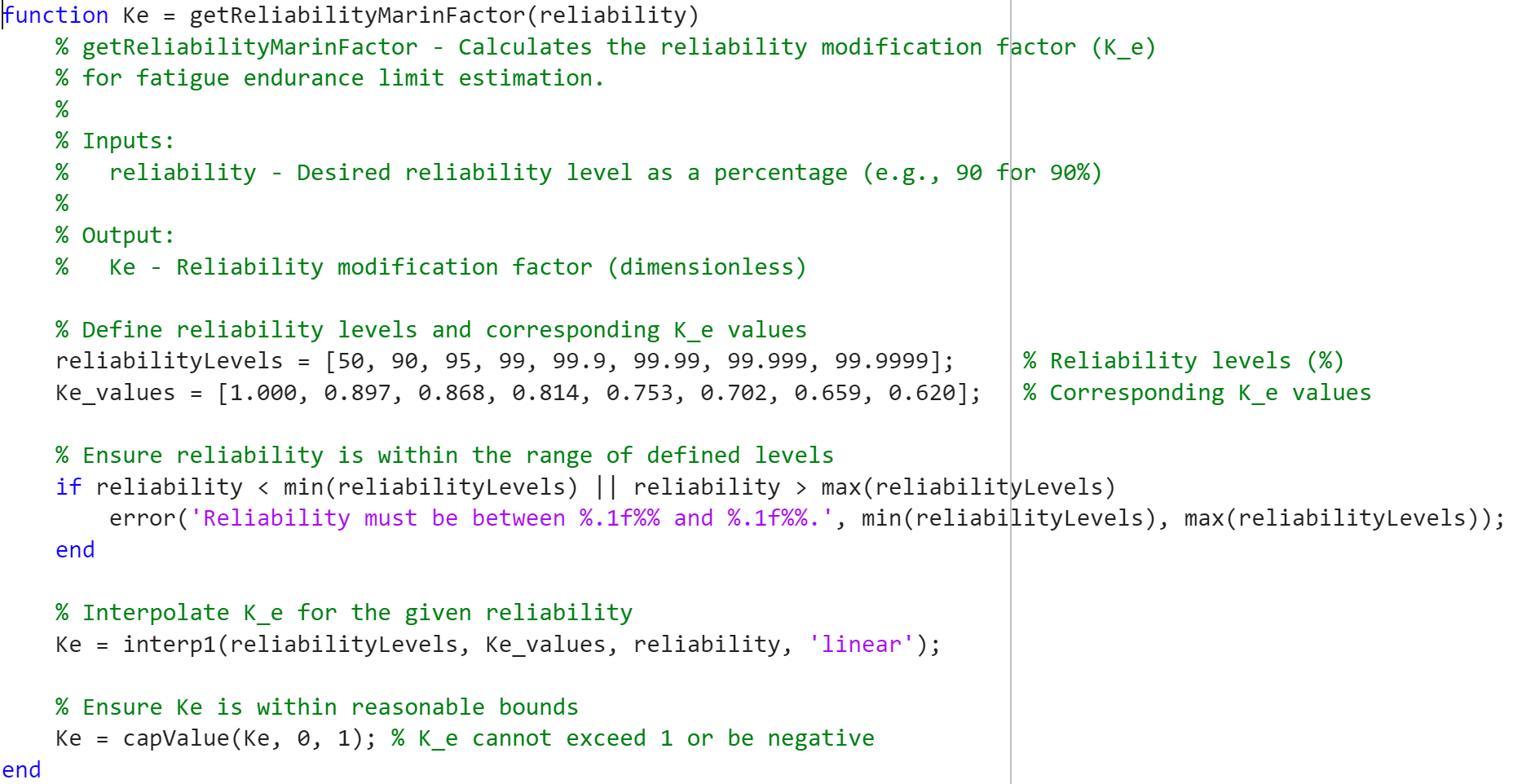


Figure 6: getReliabilityMarinFactor.m

* **Trial-and-Correction Procedure:** Iteratively refines the shaft diameter to satisfy design criteria.
  1. **Loadings:** By taking in the inputs **, , ,** of from the user, determine mean (static) and alternating (dynamic) loads .
  2. **Material/Limits**: find , and based choice of material. For 1st iteration size factor, the user can guess an approximate (), like 0.9 is a good guess (if not specify the code would use 1 as the default value).
  3. **Estimate stress concentration and from table 7-1 (**1st iteration**)**

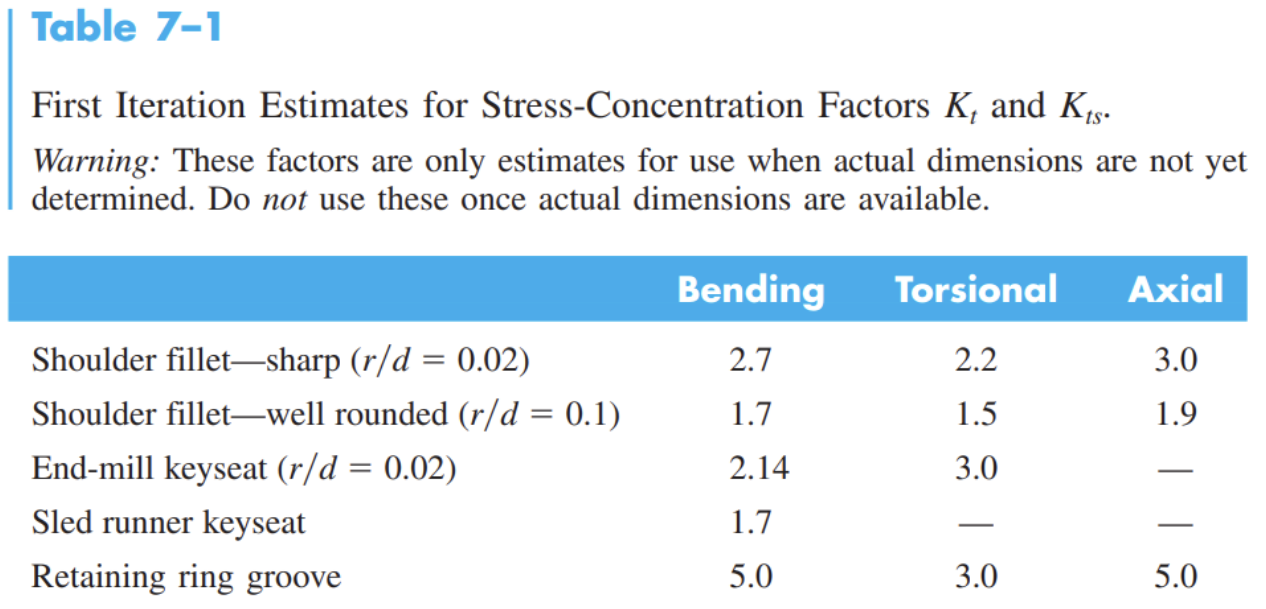
****

Figure 7: (Table 7-1) First Iteration Estimates for Stress-Concentration Factors

* 1. **Calculate the 1st diameter () with the given failure criteria:** Based on the calculated diameter () find corrected based on size factor (), as well as new stress concentration and .
  2. **Repeat the process for n iteration,   
     break out of loop if**

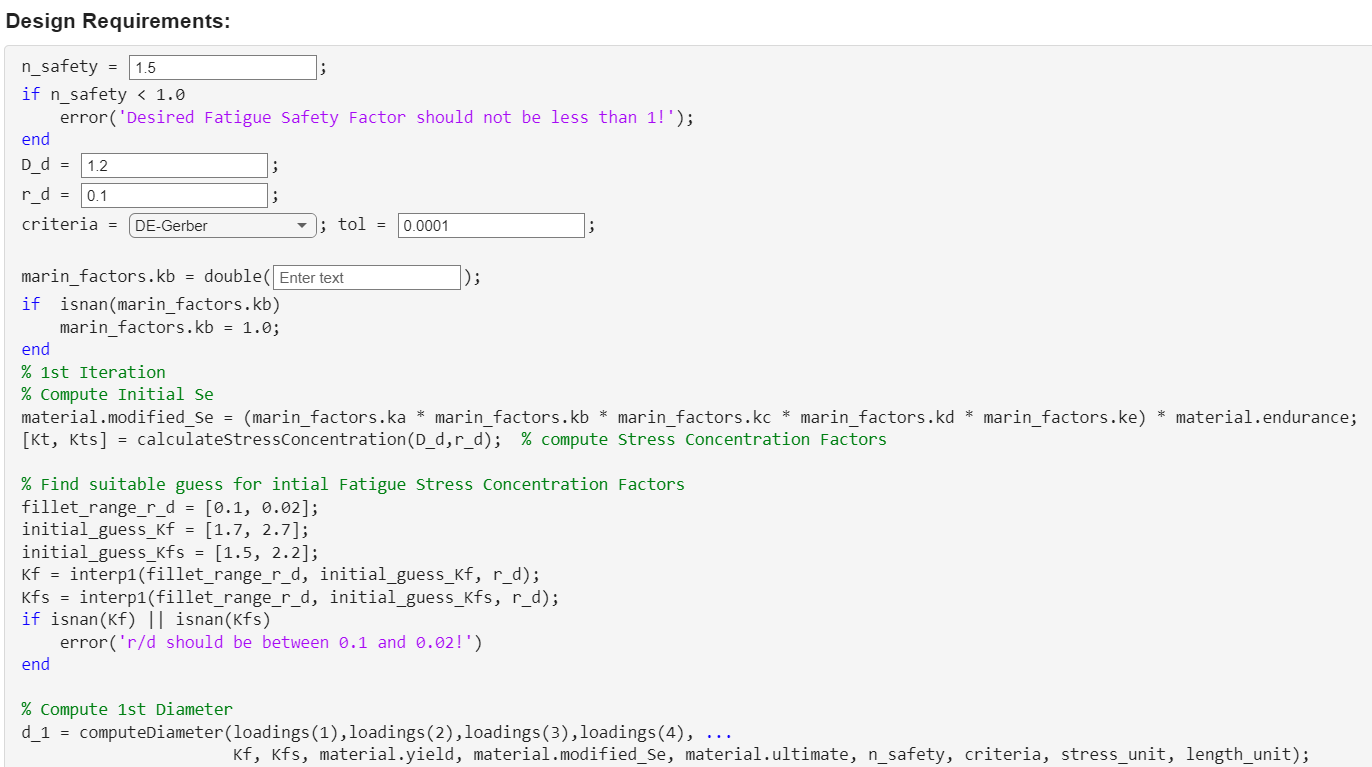
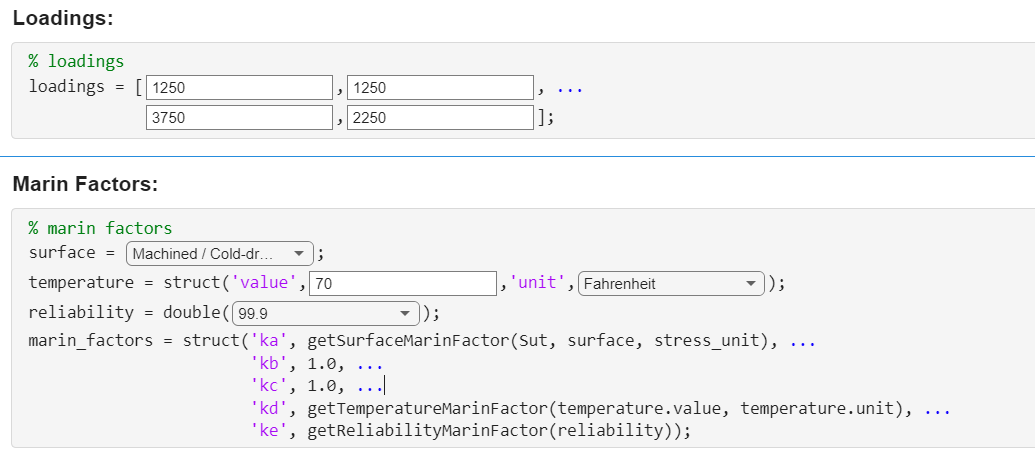
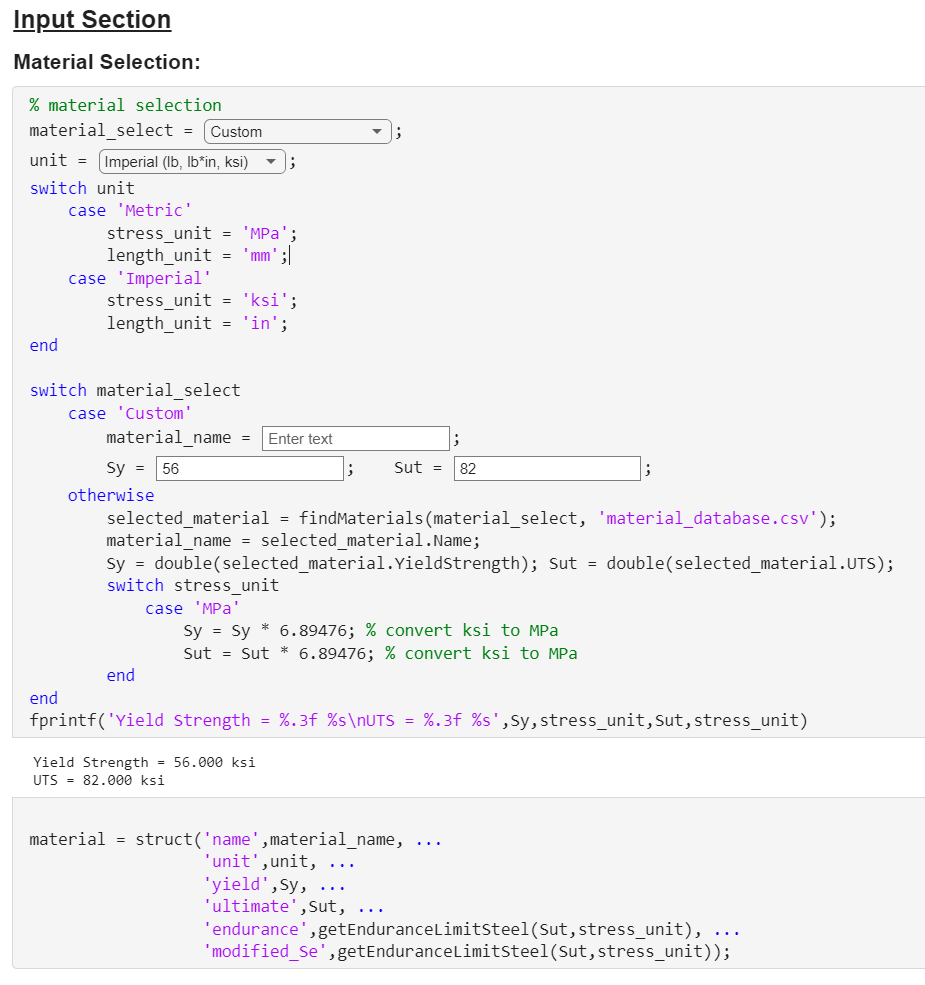
****

Figure 8: The main script, Shaft\_Design.mlx

* **Fatigue Failure Design Criteria:** Implementation of fatigue life equations (e.g., DE-Gerber) for stress analysis.

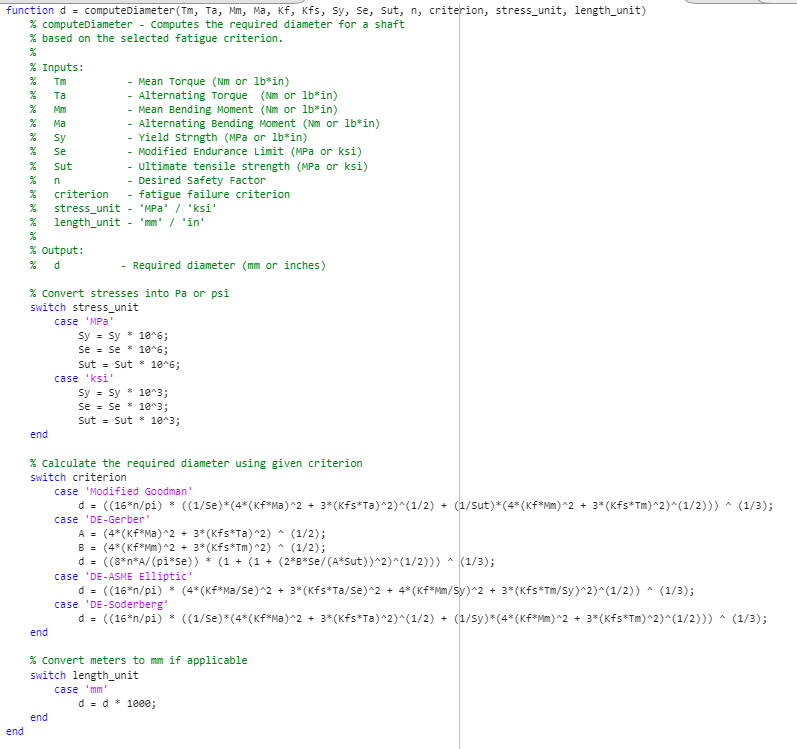


Figure 9: computeDiameter.m

* **Yielding Check:** Ensures that the computed diameter also satisfies static yield strength.

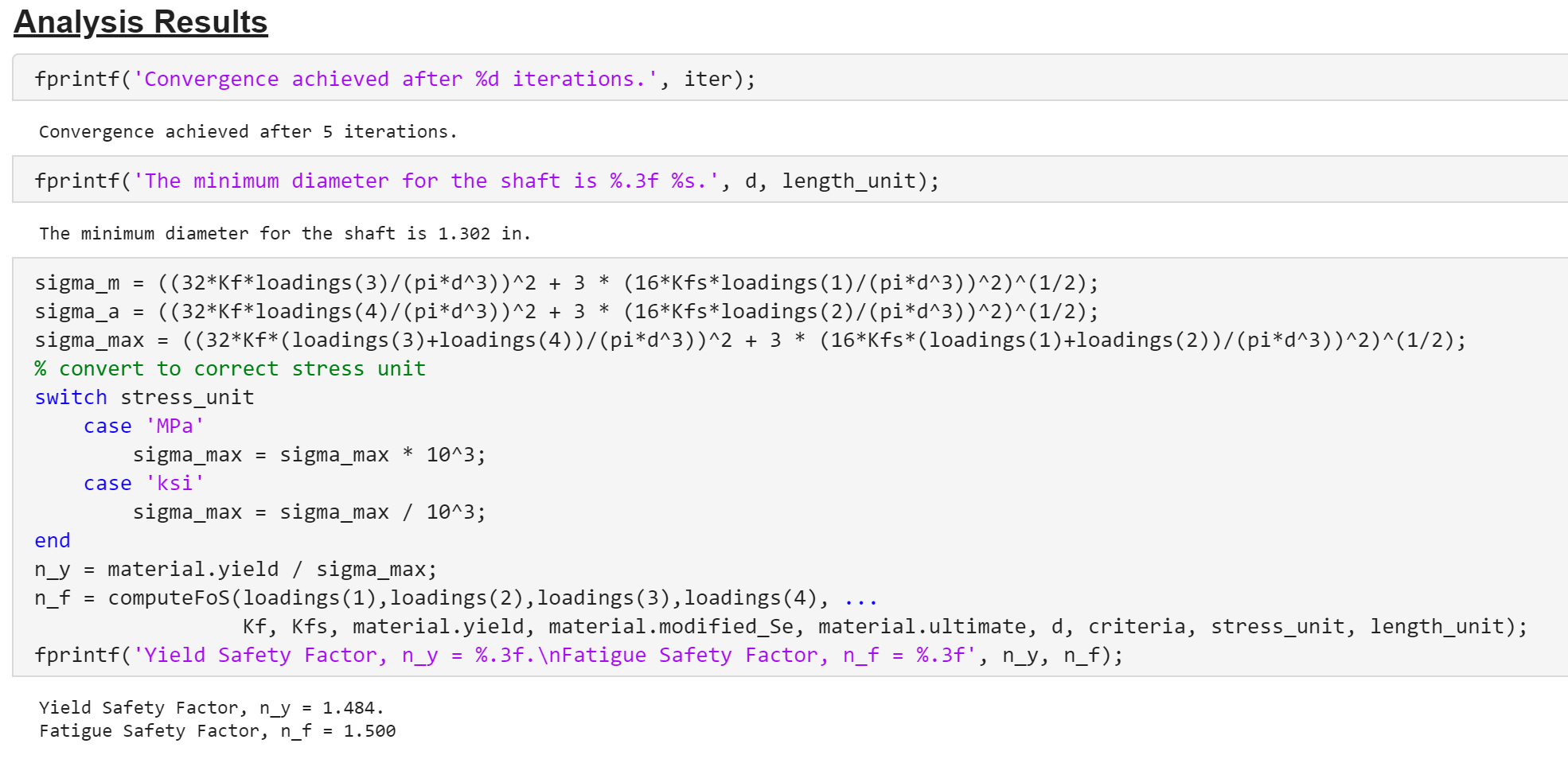
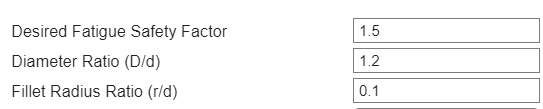


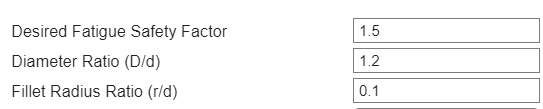
Figure 10: Analysis Section of the Main Script

**Examples 1:**

1. **Material:** AISI 1020 HR Steel  
   
2. **Design Requirements:   
   **   
   
3. **Loading:** (, , , ) = (1250, 1250, 3750, 2250)
4. **Analysis Results:**

|  |  |
| --- | --- |
| Modified Goodman | DE-Gerber |
|  |  |
| DE-ASME Elliptic | DE-Soderberg |
|  |  |

**Examples 2:**

1. **Material:** AISI 1050 HR Steel  
   
2. **Design Requirements:   
   **   
   
3. **Loading:** (, , , ) = (1250, 1250, 3750, 2250)
4. **Analysis Results:**

|  |  |
| --- | --- |
| Modified Goodman | DE-Gerber |
|  |  |
| DE-ASME Elliptic | DE-Soderberg |
|  |  |

**Discussion**

**Convergence Behavior:**The iterative approach employed in the shaft design algorithm demonstrates reliable convergence to the final diameter. The code systematically refines the shaft diameter by recalculating key parameters such as stress concentration factors, endurance limits, and Marin factors at each iteration. The convergence behavior is influenced by the initial guess for the diameter and the specified tolerance limit.

* **Convergence Plot**: A plot of iterations versus computed diameters illustrates how the algorithm approaches the final value. Typically, convergence is achieved within 3 to 6 iterations for most cases, depending on the given tolerance, as shown in Example 1 and Example 2.
* **Effect of Initial Guesses**: The speed of convergence is sensitive to the initial diameter estimate. For example, a closer initial guess reduces the number of iterations required, while a poor estimate may slightly increase computation time but still converges due to the robustness of the algorithm. But in general, it’s reaches convergence within 6 iterations with a tolerance of 0.001%.
* **Tolerance Limit**: The exit condition for the loop is based on the relative change in diameter between successive iterations . This ensures that the solution meets user-defined precision requirements, typically set at 0.0001 (0.01%).

**Input Variations:**The flexibility of the code allows users to explore how different inputs affect the output shaft diameter. Key observations include:

* **Material Properties**: Materials with higher ultimate tensile strength () and yield strength (​) result in smaller required diameters, as they can withstand greater stresses. For instance, AISI 1050 HR Steel () produces a smaller diameter compared to AISI 1020 HR Steel () under identical loading conditions.
* **Stress Concentration Factors**: Variations in geometric features such as fillet radius ratio () and diameter ratio () significantly impact stress concentration factors (, ​), which in turn affect the required diameter. Sharper fillets (lower ) lead to higher stress concentrations and larger diameters.
* **Marin Factors**: Adjustments for surface condition, temperature, size, and reliability directly influence the endurance limit (​). For example, hot-rolled surfaces or higher reliability requirements (e.g., 99.9%) increase safety margins but necessitate larger diameters.

**Material Comparison (Example 1 vs. Example 2):**

**Example 1**: AISI 1020 HR Steel ()

**Example 2**: AISI 1050 HR Steel ()

Under identical loading conditions:

* AISI 1050 HR Steel resulted in a smaller shaft diameter due to its higher strength properties. This highlights its superior performance for fatigue-critical designs where minimizing material usage is important.
* The relationship  ​ was more notable in Example 2, further demonstrating that higher-strength materials provide better margins for fatigue failure before yielding.

**Exit Conditions:**The iterative trial-and-correction loop terminates when the relative change in diameter between iterations falls below the specified tolerance. This ensures computational efficiency while maintaining accuracy. Additionally, safeguards are implemented to prevent infinite loops or unrealistic results by setting reasonable bounds on parameters such as stress concentration factors.

**Conclusion**

The developed shaft design tool successfully automates the calculation of the minimum required diameter for a circular rotating shaft under dynamic loading conditions. Using material properties, Marin factors, and user-defined design requirements, the tool integrates fatigue and static failure criteria into an iterative framework that ensures both accuracy and efficiency.

**Strength of the Code**:

The developed tool offers several advantages:

1. **Accurate Results:** For Example 2, using DE-Gerber criterion, the tool converged to a final shaft diameter of 1.554 inches after four iterations for the specified inputs. The calculated fatigue safety factor () and yield safety factor () meet the design requirements, ensuring reliability under both static and fatigue loading conditions.
2. **Flexibility**: Users can select from multiple failure criteria (e.g., Modified Goodman, DE-Gerber) and define custom inputs such as material properties or safety factors.
3. **Efficiency**: The code converges quickly and provides accurate results even for complex loading scenarios.

**Advantages for Users**:

* The intuitive interface allows users to input material properties, loadings, and design requirements with ease.
* Automated calculations eliminate manual trial-and-error processes, saving time while reducing errors.
* The inclusion of graphical outputs, such as the diameter convergence plot, provides users with valuable insights into the design process.

**Suggestions for Future Development**:

While the tool is highly effective for its intended purpose, there are opportunities for further improvement:

* Expanding the material database to include more alloys and non-ferrous materials would broaden its applicability.
* Incorporating advanced visualization tools, such as stress distribution plots or 3D models of shaft geometry, could enhance user experience.
* Exploring optimization algorithms, such as genetic algorithms or gradient-based methods, may improve computational efficiency further.

**Closing Remarks**:  
This project demonstrates the successful integration of fatigue theory, static stress analysis, and iterative computation into a cohesive shaft design framework. By automating complex calculations and providing flexibility in input parameters, this tool serves as a valuable resource for engineers designing shafts under dynamic loading conditions. With further enhancements, it has the potential to address a wider range of mechanical component design challenges.

**Appendix:**

MATLAB Code used:

<https://github.com/WinstonHChou/MAE190_Projects>