Natalie Ratzlaff
MAE 190
11/21/2024

Introduction

This code is used to iteratively determine the minimum diameter of a shaft that has a step-down in diameter with a fillet when exposed to Torsion and Bending alternating loads. This code needed to be able to reference a wide array of values, so that the user would not need to look at a table or do hand calculations when using this code. A personal goal for this code was for it to be widely applicable and easy to use, and this was achieved by using MatLab LiveScript, so all user inputs are easy to reference and modify. DE-Gerber formula was used in the iterative portion of the code to relate the smaller diameter, d, to the factor of safety, n. As will be detailed in the Results section, the SHigley book was used as a reference for several tables and equations within the code.

Results

The code itself can be separated into 3 different sections, all of which can be run separately. The first section loads in reference Excel tables, which contain reference data taken from the Shigley book, particularly for material selection (Table A20), determining ka constants A,b (Table 6-2), determining kd (Table 6-4), and determining Kt constants A,b (Figures C-2 and C-3).
The second section contains general material and situational information that does not need to be iterated on. Here, users can input:

1. the unit type (SI or Imperial),
2. the minimum and maximum bending and torsional forces being applied,
3. the type of steel being used according to ANSI number, and whether the Hot-Rolled or Cold-Drawn variant is being used,
4. the surface finish of the shaft (ka),
5. the temperature the shaft is operating under (kd),
6. the desired reliability of the shaft (ke),
7. Any additional factors impacting Se (kf),
8. the desired factor of safety on the shaft,
9. and the dimension ratios of the shaft (r/d and D/d).

The third section contains the iterative portion of the code; this is where users can input an initial guess for the smaller diameter d. This section of code contains a while loop that iterates between the 2 DE-Gerber equations while continuing to update the 3 variables that are dependent on the actual dimension d of the shaft (Kf, Kfs, and kb [and by extension Se]), until the actual factor of safety is within ±0.0001 of the desired factor of safety. This section will output the smaller diameter d (and the relative dimensions r and D), as well as the final factor of safety value (n), and the number iterative loops used to get that answer. It will also check for yielding by calculating the Von Mises max stress, and comparing it to the yield strength of the material (gives ny, the yielding factor of safety).

Note that for all user inputs, drop-down and edit fields are used, so the user is able to easily change parameters and run specific sections of the code.

Discussion

To get to the final answer, an initial guess for d is used to calculate the variables dependent on it (Kf, Kfs, Se [kb]), and then n is calculated using gerber to initialize the n variable. The while loop is then started where d is calculated using gerber with the desired factor of safety plugged in for n, and the dependent variables are then recalculated based off of this new d value. N is recalculated at the end of the while loop, and the loop breaks once n is within ±0.0001 of the intended factor of safety. Even with terrible guesses, such as a 1000in or 0.0001in diameter, the while loop converges to the desired factor of safety within 4 iterations.

Conclusion

This code is very easy to use and modify quickly thanks to the drop-down menus and the reference tables loaded into the code. It requires no hand calculations or table-referencing from the user. If this code were to be improved upon in the future, more material or situation options could be added, such as a hole being in the shaft or other potentially structurally-compromising geometrical features. The fatigue limit, rather than the endurance limit, could also be something the user could select for, in the case of less frequent alternating loading, or in the case of materials such as aluminum that do not have a fatigue limit.

Note: The zip file containing the code files has many smaller function files; the code intended for the user is named LiveCode.