Materials Selector App [CAMP] MATLAB Final Project

Oliver Carleton — Kiowa Wells — Liam Figueroa

April 2024

1 Abstract

A mechanical properties graphic comparison generator is developed to aid in the materials selection process for applications in engineering where narrowing down design options is desired. Using limited metrics that can be found across various publicly-available internet resources, the utility consolidates data to allow for the quick and easy comparison of metals. Functions are created and implemented to allow for graphical analysis of the data using Ramberg-Osgood stress-strain graphs alongside comparison charts for toughness and specific strength.

2 Introduction

Deciding which material best fits the needs of a project can be time-consuming. It is often difficult to find meaningful comparative data for the specific options you have in mind, and generating it yourself is tedious and slow. The Comparison Application for Mechanical Properties (CAMP) is a novel graphical generator tool aimed at simplifying the material selection process in engineering disciplines. It accomplishes this through the seamless integration and comparison of mechanical properties data for user-selected metals. The utility enables engineers to rapidly narrow down materials choices for applications where performance is crucial to safety and functionality.

3 Application Overview

Operating on the MATLAB platform, CAMP is a useful analytical tool that generates visual representations of metal material properties. When the program is initiated, users are asked if they would like to review a mechanical properties table of some common engineering metals. If they say yes, the table appears in the command window, and the user must hit enter to continue (Figure 1). Afterward (or if they didn't want to review the table), the user is prompted to input a handful of commonly available metrics for their chosen metals. The application then interprets the data and outputs a stress-strain graph alongside charts comparing the toughness and strength-to-weight ratios of the selected metals. These visual representations can be extremely useful to engineers, and have the added benefit of being easily understood by people with limited backgrounds in math and science.

Material	Density (g/m^3)	Yield Str (MPa)	Tensile Str (Mpa)	Young's Mod (GPa)	%EL	Fatigue Str (MPa)	HB	Cost ($/kg)
{'Al 354 T62' }	2.7	310	354	71	2.4	100	110	NaN
{'Al 2014 T6' }	2.8	420	490	72	6.8	130	140	NaN
{'Al 2024 T6' }	2.78	370	480	71	5.1	130	125	NaN
{'Al 3003 0' }	2.73	186	200	69	10	69	55	NaN
{'Al 5056 H38' }	2.64	345	414	67	14	150	100	NaN
{'Al 6061 T6' }	2.7	276	310	69	10	96	93	9
{'Al 7005 T6' }	2.78	310	380	70	12	130	94	NaN
{'Al 7075 T6' }	2.81	480	560	70	11	160	150	20
{'Brass 360 H02'}	8.5	360	470	97	18	90	100	NaN
{'304 Stainless'}	7.85	206	517	193	40	NaN	123	NaN
{'316 Stainless'}	7.85	206	517	193	35	NaN	144	NaN
{'4130 Chromoly'}	7.85	435	670	205	25.5	320	197	25
{'4140 Steel' }	7.85	655	1020	205	18	350	302	NaN
{'Ti-5Al-2.5Sn' }	4.48	820	861	120	15	290	320	NaN

Figure 1: MATLAB Materials Command Window Table

4 Methods

Mechanical properties data input by the user are stored in a structure for easy access. A stress vector S is defined with bounds based on the metals entered. The Ramberg-Osgood exponent function (ROexp.m) then uses the data to derive the strain-hardening exponent n for each material, which is needed for Equation 2:

$$n = \frac{\log\left(\frac{e_f}{0.002}\right)}{\log\left(\frac{S_u}{S_y}\right)} \tag{1}$$

- e_f is the strain at fracture.
- S_u is the ultimate tensile strength.
- S_y is the yield strength.

The elongation (elong.m) function is a modified version of the Ramberg-Osgood equation, which calculates the elongation of each material as a function of stress given its modulus of elasticity, yield strength, and strain-hardening constant:

$$\epsilon_l = \frac{S}{E} + 0.002\left(\frac{S}{S_y}\right)^n \tag{2}$$

- S is the applied stress vector.
- E is the modulus of elasticity.
- S_y is the yield strength.

- n is the material-dependent strain-hardening constant derived in Equation 1.

The specific strength (s2w.m) function calculates the strength-to-weight ratio of each material:

$$s = \frac{S_u}{\rho} \tag{3}$$

- S_u is the ultimate tensile strength.
- ρ is the density of the material.

Finally, the main script file (Final.Project.Materials.selector.m) is responsible for interacting with the user, data integration, command window interface, calling functions to perform required calculations, and plot generation.

5 Code Overview

1. **Introduction and Table Display:** After a brief introduction, the program prompts the user to decide if they want to see a materials table. If affirmative, the table is output to the command window using the following MATLAB command:

   ```
   readtable('Final Metals data.csv')
   ```

   Users are then prompted to specify the number of metals they wish to compare, which is stored in the variable 'num'. A stress vector S is initialized which will be redefined later based on the material data entered.

2. **Data Collection:** The program initiates data collection using a 'for' loop with 'num' iterations. In each iteration, it asks for six data points for the kth metal: name, density, yield strength, tensile strength, Young's modulus, and fracture elongation. These data points are stored in a structure imaginatively named "metals".

3. **Calculations:** The user-defined functions ROexp and s2w are called during each iteration of the data collection loop, right after the data has been entered by the user. As described above, ROexp derives the Ramberg-Osgood strain-hardening exponents and s2w finds the specific strength. The results of these are added to the structure for each metal.

   A new for loop is then initiated and the user-defined function elong is called, which is a formulation of the Ramberg-Osgood equation that calculates stress as a function of strain for each metal. The reason this requires a separate loop is that the vector S is changing in every iteration of the previous loop. When that loop ends, S has a fixed length and the elongations can be calculated for all of the metals. These are stored in a separate matrix EL, and all values for each metal that exceed its fracture elongation are trimmed off and replaced by NaN for plotting purposes.

Finally, toughness is calculated as follows: on each iteration, EL(k) is also stored in the structure, where its length can be shortened to match the fracture elongation for each metal (this would not work for the EL matrix, which needs to be the same length as S for plotting stress vs strain). MATLAB's built-in trapz function is then called to calculate the area under the stress/strain curve for each metal (i.e. toughness), and the values are stored in the metal's structure.

4. **Plotting:** At the end of the last for loop, specific strength, toughness, and metal names are pulled from the structure and placed into their own respective vectors for plotting purposes. Stress and strain are plotted intuitively as EL vs S.

   If the user chooses to model only one metal, only the stress-strain plot is generated, as the others are comparative bar charts. Otherwise, two figure windows are generated; one with the stress-strain graphs on a single plot, and the other with comparative bar charts for both toughness and specific strength.

6 Test Cases

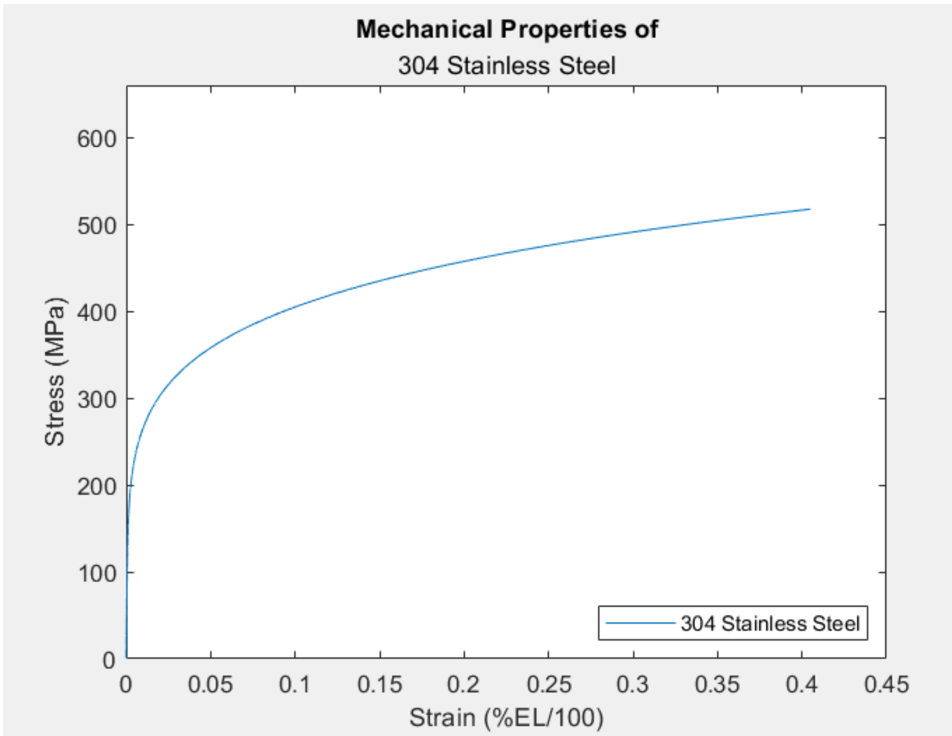

Figure 2: [Case 1] Stress-Strain Relationship [Example Case with One Metal]

The graphs in this section depict the results from three test cases. We chose to do 304 stainless for the first case, mostly because it has well-known values and makes a nice looking graph on its own. When only one metal is modeled, only the stress-strain graph is generated (Figure 2).

For case 2, we compared three metals with overlapping material similarities. This was interesting in that the toughness and strength-to-weight ratios reflect the comparative yield strengths which, as the next case demonstrates, is not as usual as one might think (Figure 3).

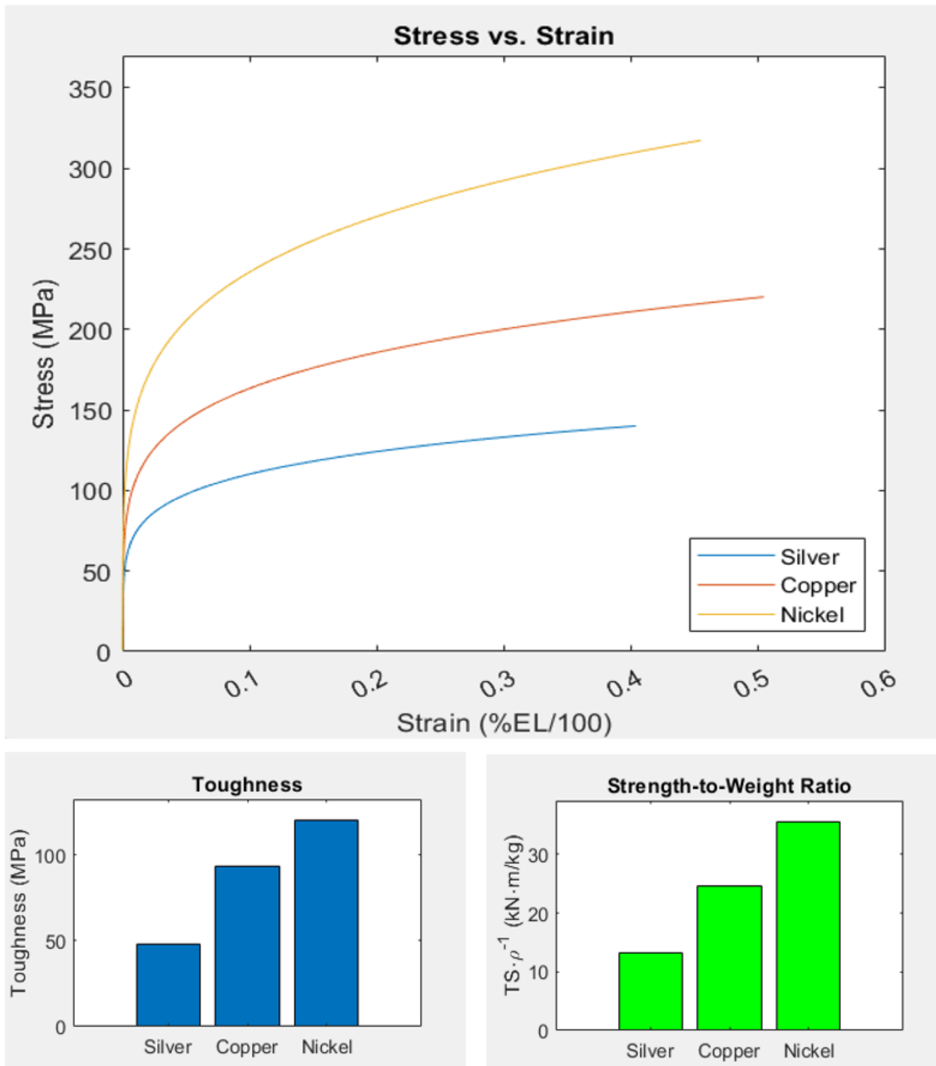

Figure 3: [Case 2] Stress-Strain Relationship, Toughness, and Strength-to-Weight Ratio for Materials

Case 3 is what inspired this project in the first place: One of our team was working on another project, scouring the internet for comparative analyses of bike frame materials and striking out. After writing up a simple program just to graph the metals of interest, we realized it could be developed into a program that was actually useful. Figures 4 and 5 show those same metals run through the current version of our program.

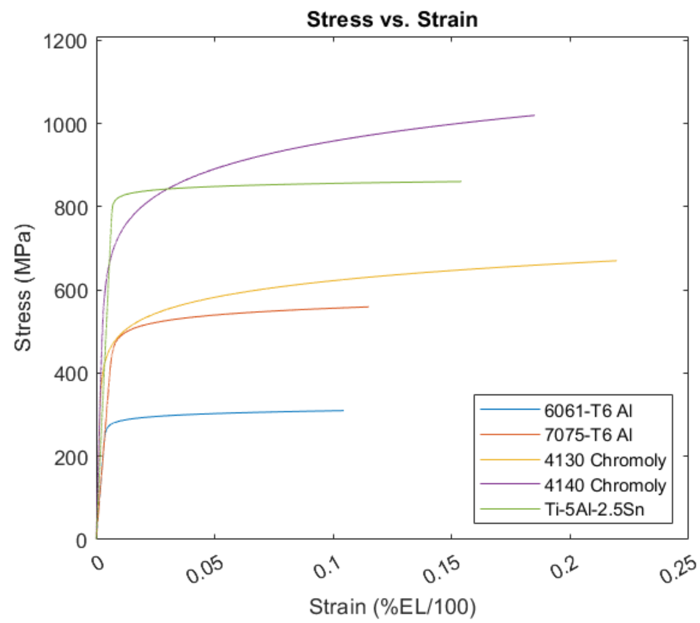

Figure 4: [Case 3] Stress-Strain Relationship for Bike Metals

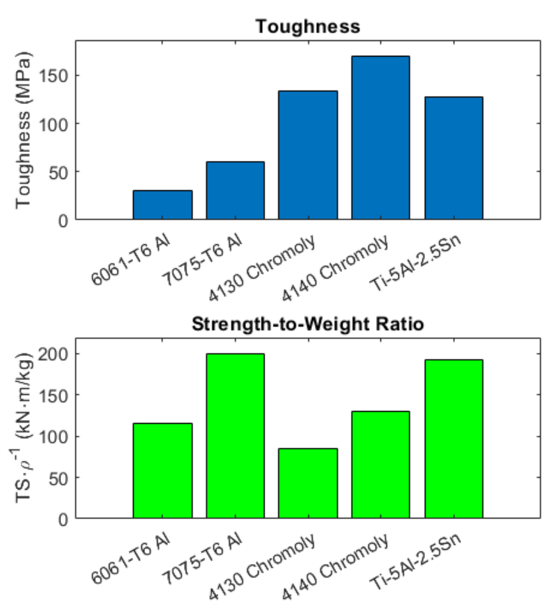

Figure 5: [Case 3] Toughness and Strength-to-Weight Ratio for Materials

7 Results/Discussion

After plenty of testing and tweaking, we are pleased with the overall performance of the program. There is room for improvement, but it is as good a foundation for future development as we could want. As shown in the previous section, CAMP generates detailed plots of valuable material property comparisons. The test cases adequately demonstrate that the utility efficiently processes and displays data, providing a fast and easy means of comparative analysis. In particular, Case 3 is a good real-life example of how much easier the program makes it to narrow things down based on specific design criteria: For light weight combined with high strength, 6061 and 4130 can be ruled out immediately just by glancing at the charts. Of the remaining 3, any slim advantage titanium might have is likely to be far outweighed by its cost, and so in minutes we are left with a clear picture that the only good options are 7075 and 4140. The user-friendly interface allows anyone to make more informed decisions about what material would be best for a given application. Its customizable nature allows for easy integration into engineering decision-making processes for streamlining selection.

One of the major drawbacks of CAMP is precision. The Ramberg-Osgood relationship is a a generalized mathematical model for predicting metal behavior, and as such should only be relied on for approximations. Where more precise modeling is desired, there's no substitute for experimental data, and those in search of more a comprehensive picture will need to look elsewhere. Another con is data dependency, which renders the tool less potent when modeling rare materials with less readily available metrics. A third limitation is that in the program's current iteration, data input is a bit tedious and annoying to correct. Finally, this program could be integrated into cloud-based services where larger datasets could be compiled to make sourcing easier, which remains the largest inefficiency of the process.

8 Conclusion

The CAMP utility has proven effective in enhancing the efficiency of the material selection processes by providing quick and accurate graphical representations of mechanical properties. It facilitates a deeper understanding of how different materials measure up, and could prove to be particularly useful in educational settings and professional design workflows. Future enhancements could include integration of a comprehensive materials database, incorporation of additional property analyses such as cost and fatigue life, and implementation of a Graphic User Interface to improve end user experience with even easier navigation and customization.