MATLAB Jump-Start with One Project

Yuxin Zhang, Mechanical Engineering, Washington State University Tri-Cities

As a Mechanical engineer, it is essential to develop computational skills due to the following reasons: (1) Many complex engineering problems with rich physics are likely to be modeled by nonlinear, coupled partial differential equations. We cannot find analytical solutions to these problems. Instead, using numerical methods is a great approach in solving complex equations. (2) With the fast development and easy access of PCs, students can learn and appreciate the strong power of solving engineering problems computationally such as simulations of fluid flows or dynamic systems. (3) Developing computational skills help students get the equivalent "hands-on" experience that they will learn from experiments. However, due to the limitation of advanced experimental facilities, not every institution or department can provide students with the ideal apparatuses such as a high-speed wind tunnel. In this case, the numerical simulation may be a good alternative.

I have been teaching various upper-level Mechanical Engineering courses in which modeling physics and solving differential equations are always the challenges for students. MATLAB turns out to be a very useful tool for computation due to its interactive environment with a large library of functions and inbuilt graphics. However, every beginning is difficult especially for students without computational experience. Although there are plenty of tutorials available, sometimes it is these overwhelming resources that get students lost at the entrance of computation. As an instructor, it is essential to help students develop their computational skills using MATLAB, even importantly, to develop a mindset that MATLAB is just a tool and it is more than a tool.

It can take a very long time for students to develop their computational skills, but at the end of the day, students want to apply the computational skills to solving engineering problems instead of just learning a tool. From this perspective, it is extremely beneficial to jump-start with a project that integrates learning computation into problem-solving. It helps students not only get fully immersed in a computational environment but also stay on track with solving real engineering problems. It also helps students quickly develop the "hands-on" experience and realize the power of computation. To realize the fast-track learning outcomes, it is essential to develop projects that combine computational training and problem-solving. For example, when I teach the convection-diffusion equation that might be covered in a Fluid Mechanics, Heat Transfer, or Computational Fluid Dynamics course, I always take a modeling-based strategy to help students learn the equation with various applications. Many physical phenomena related to the processes of convection and diffusion can be modeled by this equation such as in the transport of air pollutants, airborne infection of diseases, and semiconductor physics. In my courses, students learned the fundamentals of convection and diffusion equations as well as numerical techniques. For different projects such as the modeling of temperature distribution in a fin or backward-facing step flows in a channel, they were able to model the physics and solve the differential equations numerically by coding in MATLAB. The projects are designed in a self-explained fashion with examples and code blocks provided such that students can embark on their explorations.

Through these project-based training, students can quickly master the necessary computational skills with MATLAB such as creating functions, solving matrices, writing scripts, and debugging the codes. Some other important topics on MATLAB can be covered in a variety of projects and exercises according to the teaching requirements of different courses. For example, I designed Thermal and Fluids virtual labs using MATLAB SIMULINK modules to accommodate students' needs for remote learning. I also gave students demonstrations on how to use SIMULINK to solve ODEs. These examples provide students with more than one perspective for learning computation and implementing the skills into problem-solving. MATLAB is a tool for computation, but once we jump-start from the origin, it is more than a tool. It is a powerful platform where we can integrate, build and explore.