MATLAB PERFORMANCE TUNING

COURSE OUTLINE

1. Profiling Matlab performance
   a. When to profile and when not to bother
   b. When should we stop optimizing the code?
   c. Profiling techniques
   d. Real-time profiling limitations
   e. Using the profiler vs. tic/toc
   f. Trade-offs: performance vs. maintainability, robustness, development time, repeatability
   g. Matlab’s JIT and its effect on profiling
   h. Vertical vs. horizontal scalability

2. Standard programing techniques
   a. Perceived vs. actual performance
   b. Loop optimizations
   c. Caching
   d. Smart checks bypass
   e. Exception handling and performance
   f. Parallelization
   g. GPU

3. Data analysis techniques
   a. Selecting the right tool for the job
   b. Outliers removal
   c. Controlling the target accuracy
   d. Coordinate transformation

4. Matlab-specific techniques
   a. Effects of using different storage types
   b. Vectorization
   c. Object-orient Matlab and performance
   d. Using internal helper functions
   e. I/O aspects
   f. Graphics and GUI aspects
   g. Using mex code

5. Memory-related techniques
   a. Why memory affects performance
   b. Pre-allocation
   c. In-place data manipulations
   d. Using global and persistent variables

6. Where next? – topics and resources for further learning

Summary

A 1-or-2 day advanced Matlab course.

You will learn:

- when to profile, when not to bother
- how to determine the most effective optimization route
- trade-offs of program performance
- how to profile Matlab programs to determine where the hotspots are
- how to avoid and solve potential hotspots in your program’s execution
- how to use a variety of techniques to maximize the performance of your application at reasonable cost
- skills that will enable you to continue learning advanced topics by yourself, at your own pace

In the 2-day course, the topics will be expanded and a sample program will be analysed as a classroom exercise project.

Target audience

Matlab users with some experience using Matlab, who wish to improve their programming quality and effectiveness; Matlab users who wish to sell or deploy professional Matlab-based software where timely performance is important.

Basic familiarity with the Matlab environment, data types and functions is assumed.