This course will emphasize statistical programming in Matlab (primarily), and in a fairly in-depth manner. In addition, there will be substantial discussion about reading C (and C++), Fortran, and R code, and on translating legacy code into Matlab proper.

The class will begin with a review of basic Matlab, with an emphasis on graphics and creating graphical user interfaces. The major part of the course will be in translating mathematical/statistical approaches into implementable code. When necessary, the various routines from the Optimization Toolbox will be incorporated into the computational routines.

Class Topics Sequence:

I. Introduction to Matlab:

Nine short videos:
1) Getting started with Matlab
2) Working in the development environment
3) Writing a Matlab program
4) Importing data from files
5) Creating a basic plot interactively
6) Using basic plotting functions
7) Working with arrays
8) Introducing Matlab fundamental classes (data types)
9) Introducing structures and cell arrays

Webinar (from Mathworks) (60 minutes): Introduction to Matlab

II. Introduction to Matlab programming:
Selected slides on programming from “The Briefest of Introductions to Matlab”
Chapter 11 on Control Flow from Hanselman and Littlefield (Mastering Matlab)
Chapter 10 on Relational and Logical Operations

III. (Multiple Classes) Case study – The Dykstra-Kaczmarz method for solving linear (in)equality constrained least-squares tasks:
   a) Theory
   b) Application to linfitac.m
   c) The general M-file for solving inequality constrained least-squares tasks
   d) Application to linfitac_altcomp.m; linear_order_member.m, partitionfit_addcon.m, and dykstra.m
   e) Translating the Fortran code to Matlab

IV. (Several classes) Getting Started (Chapter 3: Graphics):
   a) Basic plotting functions (command line) (3-58 to 3-96)
   b) Working with the graphics editor (Illustrator analogue) (3-1 to 3-57)

V. (Multiple classes) Case study – The dynamic programming approach to seriation:
a) Theory
b) Fortran (90) implementation: DPSE1U.for
c) Matlab implementation: uniscaldp.m
d) Calling Fortran routines from Matlab: uniscaldpf.m, uscalfor.for, uscalforgw.for

VI. Graphical user interfaces:
   a) Matlab video(s)
   b) GUIDE (Chapters 1 and 2 in Creating Graphical User Interfaces)
   c) GUI Layout Toolbox

VII. Incorporating optimization routines from the Optimization Toolbox
   a) unifitl2nlp.m: external function is objfunl2.m; uses fmincon.m
      linfitl1.m; uses linprog.m
      lpfit.m: uses fmincon.m; internal nested function calculates Minkowski distances
   b) Section 32.12: Nonlinear curve fitting (Hanselman/Littlefield, Mastering Matlab); uses fminsearch.m (without derivatives)
   c) lsqnonlin.m (nonlinear least squares)

VIII. Desktop tools and development environment/programming style:
   a) Profiler; M-lint
   b) Matlab programming style guidelines and The Elements of Matlab Style; both by Richard Johnson

IX. R analogues of Matlab:
Matlab / R Reference (David Hiebeler)
X. Introduction to Fortran 90 programming:
Fortran 90 Tutorial by C.-K. Shene

XI. Introduction to C (C++) programming:
Part 1: Introduction to C++ Programming (pp. 7–62)
C++ for Dummies (Stephen Davis)

Use the Code::Blocks environment and the GNU GCC compiler (www.codeblocks.org)
Use the Matlab Compiler to distribute your program
Or, use the Matlab Compiler to turn your Matlab program into C++ code