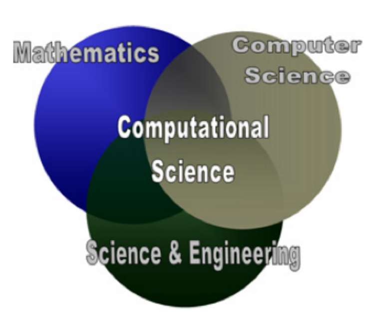
Advanced MATLAB for Scientific Computing



Course Description

The course is aimed for participants working or conducting research in scientific computing. Covered topics in scientific computing will include numerical linear algebra, numerical optimization, ODEs, and PDEs. Relevant applications areas include machine learning, electrical engineering, mechanical engineering, and aeroastro.

There will be seven interactive based lectures with application based assignments to follow. Participants will be introduced to advanced MATLAB features, syntaxes, and toolboxes not traditionally found in introductory courses. Material will be reinforced with in-lecture examples, demos, and homework assignment involving topics from scientific computing. MATLAB topics will be drawn from: advanced graphics (2D/3D plotting, graphics handles, publication quality graphics, animation), MATLAB tools (debugger, profiler), code optimization (vectorization, memory management), object-oriented programming, compiled MATLAB (MEX files and MATLAB coder), interfacing with external programs, toolboxes (optimization, parallel computing, symbolic math, PDEs).

Participants should expect to gain: ◆ exposure to the tools available in the MATLAB software ◆ knowledge of and experience with advanced MATLAB features ◆ independence as a MATLAB user. Successful completion of the course requires a cumulative score of 70% on the six assessments.

Course Outline

- Lecture 1
  - Fundamental MATLAB Language features, syntaxes, concepts [1]
    - Data types
    - Numeric arrays
    - Functions/scripts
    - Memory management
- Lecture 2
  - Additional MATLAB Desktop features
    - Publishing
    - Debugger

    - Profiler
    - Code performance and optimization

- Lecture 3
  - Graphics
    - Advanced Plotting Functions
      - Vector fields
      - Contour plots, surfaces, volumes, polygons
    - Graphics handles and objects
    - Publication-quality graphics
    - Animation

- Lecture 4
  - Numerical linear algebra [2, 3]
    - Dense vs. sparse matrices
    - Direct vs. iterative linear system solvers
    - Matrix decompositions
      - LU, Cholesky, QR factorizations
      - Eigenvalue decomposition (EVD)
      - Singular value decomposition (SVD)

- Lecture 5
  - File manipulation and system interaction
    - Text/binary file manipulation
    - System calls
    - Interfacing to spreadsheet (Excel)

- Lecture 6
  - Object-oriented programming
    - User-defined classes

- Lecture 7
  - MATLAB toolboxes & functionality
    - Solution of nonlinear systems of equations
    - Numerical optimization [4, 5, 6]
      - Optimization toolbox [6]
    - Symbolic Math Toolbox
    - Parallel Computing Toolbox
    - ODE Solvers
    - PDE Toolbox

Prerequisites

- (highly recommended) Basic programming skills in MATLAB
- (recommended) Basic knowledge of numerical analysis and numerical linear algebra

Instructor

Danielle C. Maddix, Ph.D. Candidate
Institute of Computational and Mathematical Engineering (ICME),
Stanford University

References

[1] T. A. Driscoll, "Crash course in matlab," 2003.

[2] G. H. Golub and C. F. Van Loan, Matrix computations, vol. 3. JHU Press, 2012.

[3] Y. Saad, Iterative methods for sparse linear systems. Siam, 2003.

[4] P. E. Gill, W. Murray, and M. H. Wright, "Practical optimization," 1981.

[5] J. Nocedal and S. Wright, Numerical optimization, series in operations research and financial engineering. Springer, 2006.

[6] A. Geletu, Solving Optimization Problems using the Matlab Optimization Toolbox-a Tutorial. 2007.

[7] R. Burden and J. Faires, Numerical Analysis. Cengage Learning, 2004.

[8] R. J. LeVeque, Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems, vol. 98. Siam, 2007.