Literate Scientific Computing

SORIN MITRAN

Department of Mathematics
University of North Carolina at Chapel Hill

Abstract

A project-based introduction to scientific computing that in addition to covering foundational mathematical concepts, presents the Julia language as an introductory programming environment and uses the T_EX_{MACS} scientific editing platform to refine the concept of literate programming to the requirements of reproducible computational research.

Computational modeling of natural phenomena has become a cornerstone of scientific inquiry, completing the traditional methods of theory construction and experimentation. The distinctive feature of scientific computation is exhaustive testing of our understanding of well-defined theoretical models, to an extent that is not possible without machines to rapidly carry out arithmetic operations. This seminar will introduce students to the art of successful scientific simulation. Simple models from the physical, biological, and social sciences will be introduced, given correct mathematical formulations, implemented in computer code, and analyzed. Concepts from the sciences, mathematics, and programming will be introduced as needed with no formal prerequisites beyond typical high school material. The objective will be to produce "live" computational documents that serve as virtual experiments for some field of scientific inquiry.

Relevance of computational approaches to science requires adoption of the scientific method of verification of the predictions resulting from conjectures (or hypotheses or theories). For scientific computing, the conjectures are the mathematical approach and implementation into a program executed by a computer. Predictions are obtained from program execution and verified by comparison to known results or experiments. Such computational predictions should be reproducible. This course adopts the Julia programming language, a general-purpose language with many features useful for scientific computing, as the environment to introduce the practices of reproducible computational research.

A key part of the scientific method is documentation of an investigation, clearly citing sources, approaches, hypotheses, and results. Several specialized programming languages have been developed for this purpose (TEX, LATEX, Markdown), some with an explicit focus on documenting theoretical approach and computer implementation simultaneously (web), a practice known as literate programming. The project-based approach of this course seeks to instill this practice of scholarship into all aspects of scientific computing, defining a literate programming approach based upon the T_EX_{MACS} platform in conjunction with Zotero reference management.

Table of contents

PRELIMINARIES

⟨include|L01.tm⟩

Population Models

⟨include|L02.tm⟩

⟨include|L03.tm⟩

⟨include|L04.tm⟩

⟨include|L05.tm⟩

⟨include|L06.tm⟩

CRYSTALS, QUASICRYSTALS, TILINGS

⟨include|L07.tm⟩

⟨include|L08.tm⟩

⟨include|L09.tm⟩

⟨include|L10.tm⟩

⟨include|L11.tm⟩

Image Processing

⟨include|L12.tm⟩

⟨include|L13.tm⟩

⟨include|L14.tm⟩

⟨include|L15.tm⟩

⟨include|L16.tm⟩

Random Walks and Diffusion

⟨include|L17.tm⟩
⟨include|L18.tm⟩
⟨include|L19.tm⟩
⟨include|L20.tm⟩
⟨include|L21.tm⟩

Monte Carlo

⟨include|L22.tm⟩
⟨include|L23.tm⟩
⟨include|L24.tm⟩
⟨include|L25.tm⟩
⟨include|L26.tm⟩

Graph Laplacian

⟨include|L27.tm⟩
⟨include|L28.tm⟩

FLOCKING, SCHOOLING, FOLDING