

Computing Course

CSE 380, Tools and Techniques of Computational Science

1. Hardware principles

- 1.1. Processors, caches, memory, I/O, networking principles
- 1.2. Binary representations of numbers, round-off errors
- 1.3. Integer and floating point computations

2. Programming Languages

- 2.1. Common scientific computing languages (C/C++, F90)
- 2.2. Memory allocation/management, array operations, and common data structures
- 2.3. Interoperability issues for codes using routines from multiple languages

3. Operating Systems and Unix Environments

- 3.1. Features of UNIX/Linux for scientific and technical computing
- 3.2. Languages, compilers, debuggers, performance tools
- 3.3. Make files, build systems, shell scripting
- 3.4. File management, source code control.

4. Performance Programming and Debugging

- 4.1. Best practices for developing, testing and debugging codes
- 4.2. Performance measurement and code optimization

5. Research Documentation and Simple Data Visualization

- 5.1. Tools for research and code documentation: LATEX, Doxygen
- 5.2. Plotting tools

6. Software Best Practices

- 6.1. Software design cycle
- 6.2. Regression testing, defensive programming, verification
- 6.3. Code coverage