Scientific Computing with C++

- C++ is the **most heavily used language** in science and professional programming – for large scale projects
- All major modern **operating systems** are written in C++ including Windows, Linux, Android, MacOS & iOS
- You should **always know the language of your operating system** – like you should always know the architecture & assembly language of the CPU you are using
- GPU coding (nVidia, Radeon, etc) is all done using C++
- All **major video games** are written in C++
- BNL uses C++ and Python **almost exclusively**

Useful C++ Reference Sites

- https://isocpp.org (ISO Committee)
- http://en.cppreference.com/w
- http://www.cplusplus.com/doc/tutorial
- http://c-faq.com
- http://www.cprogramming.com
- What you should know about C++11
- The standard way of converting between numbers and strings in C++11

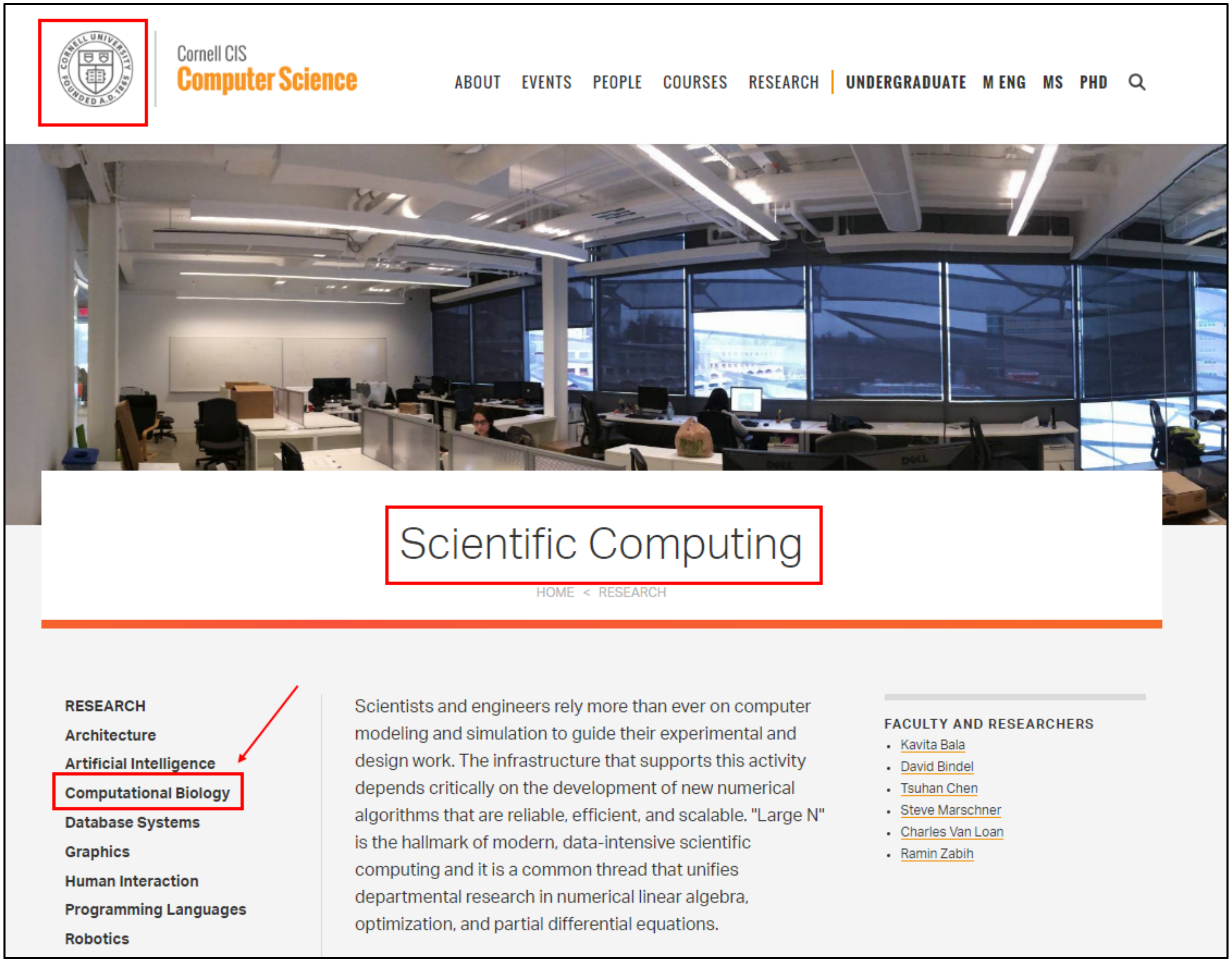
Reasonable Expectations

- It takes **500 hours** to become an **expert beginner** at anything
  - Consider playing quality: Junior High Band vs. High School Band
  - 7 & 8^{th} grade: 5 hrs/week = ~250 hrs/yr * 2 yrs = **500 hours**
  - We will spend only ~**<u>125</u> hours** together: **~$1/2^{th}$** of the required time
- We all learned how to ***read*** before we learned how to ***write***
  - A blank computer screen is too intimidating for new programmers
  - We don't recommend always "starting from scratch"
  - Many junior BNL staff inherit existing code to fix or extend
  - As a total novice, **we learn even when we just retype other's code**

Reasonable Expectations

- Programming is a **very precise science** – it metes swift frustration for the slightest inattention to details
  - There is never just one way to solve a problem – **variability** is an unsettling but **necessary** complication
  - Sometimes the only way to learn is to sit there and cry about it – **everyone** bumps their head along the way
- Lab exercises are taken directly from active research projects **at BNL**
  - Greatly simplified mathematics, but hopefully retain **the essence** of the underlying approach and motivation of the researchers
  - **Take away some of the mystery but none of the marvel**

An Degree in Scientific Computing

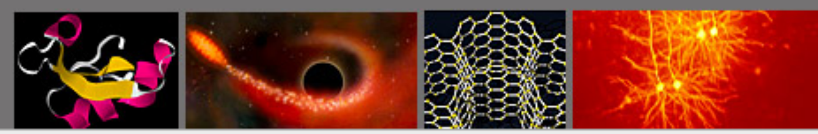


An Degree in Scientific Computing

VANDERBILT UNIVERSITY

VANDERBILT HOME | NAVIGATE VU | TOOLS | SEARCH VU

Scientific Computing
AT VANDERBILT



Home | Undergraduate Minor | Research Opportunities | People | Frequently Asked Questions | Contact

Home > Course Descriptions

Scientific Computing Courses

SC 3250 Scientific Computing Toolbox. Team taught course with topics illustrating use of computational tools in multiple science and engineering domains. Topics may include simulations of complex physical, biological, social, and engineering systems, optimization and evaluation of simulation models, Monte Carlo methods, scientific visualization, high performance computing, or data mining. Prerequisite: CS 1101 or 1103; Math 1200. [3]

SC 3260 High Performance Computing. Introduction to concepts and practice of high performance computing. Parallel computing, grid computing, GPU computing, data communication, high performance security issues, performance tuning on shared-memory-architectures. Prerequisite: CS 2201 or CS 2204. [3]

SC 3841 Directed Study in Scientific Computing. Participation in ongoing research projects under direction of a faculty sponsor. Project must combine scientific computing tools and techniques with a substantive scientific or engineering problem. Consent of both the faculty sponsor and one Director of the SC minor is required. Prerequisite: SC 3250. [1-3 each semester]

Spring 2017 Courses

- EECE 6358 Quantitative Medical Image Analysis
- MATH 3620 Introduction to Numerical Mathematics
- MATH 3630 Mathematical Modeling in Biology and Medicine
- MATH 3660 Mathematical Modeling in Economics
- MATH 4630 Nonlinear Optimization
- MATH 9601 Seminar in Applied Mathematics

An Degree in Scientific Computing

The Department of Scientific Computing (DSC) is an **interdisciplinary unit** consisting of biologists, computer scientists, engineers, geneticists, geophysicists, materials scientists, hydrologists, mathematicians, and physicists. The DSC offers an innovative undergraduate **degree** in scientific computing that imparts a **synergy between disciplines**, thus providing extensive interdisciplinary, hands on training.

An Degree in Scientific Computing

NYU | COURANT

Mathematics

PEOPLE RESEARCH COURSES PHD MASTER'S UNDERGRAD RESOURCES CALENDARS ABOUT

- People
- Research
- Courses
- PhD
- Master's
  - Overview
  - Admissions
  - M.S. in Mathematics
  - M.S. in Scientific Computing
  - M.S. in Mathematics in Finance
  - Non-Degree Study
- Undergrad
- Resources
- Calendars
- About

M.S. in Scientific Computing

The departments of mathematics and computer science at NYU's Courant Institute of Mathematical Sciences offer a master's degree in scientific computing. The program provides broad yet rigorous training in areas of mathematics and computer science related to scientific computing. It aims to prepare people with the right talents and background for a technical career doing practical computing.

The program accommodates both full-time and part-time students, with most courses meeting in the evening. The masters program focuses on computational science, which includes modeling and numerical simulation as used in engineering design, development, and optimization. While data science is an increasingly important aspect of computational science, this program is distinct and different from the recently-created Masters of Science in Data Science within the NYU Center for Data Science. Students specifically interested in data science are encouraged to apply to that program instead.

Scientific Computing: Overview

Scientific computing is an indispensable part of almost all scientific investigation and technological development at universities, government laboratories, and within the private sector. Typically a scientific computing team consists of several people trained in some branch of mathematics, science, statistics, or engineering. What is often lacking is expertise in modern computing tools such as visualization, modern programming paradigms, and high performance computing. The master's program in scientific computing aims to satisfy these needs, without omitting basic training in numerical analysis and computer science. Many graduates of this program work at technologically advanced institutions, especially in research and development, where their skills and experience complement those without interdisciplinary degrees. The program is also open to students who will go on to pursue doctoral studies in computer science, mathematics, or statistics.

The master's program in scientific computing focuses on the mathematics and computer science related to advanced computer modeling and simulation. The program is similar in structure to terminal master's programs in engineering, combining classroom training with practical experience. The coursework ranges from foundational mathematics and fundamental algorithms to such practical topics as data visualization and software tools. Electives encourage the exploration of specific application areas such as mathematical and statistical finance, applications of machine learning, fluid mechanics, finite element methods, and biomedical modeling. The program culminates in a master's project, which serves to integrate the classroom material.

An Degree in Scientific Computing

Mathematics

- ✓ Applied Linear Algebra
- ✓ Probability & Statistics
- ✓ Calculus (Integral)
- ✓ Differential Equations
- ✓ Discrete Mathematics
- ✓ Complex Analysis

Computer Science

- ✓ Internet Technologies
- ✓ Numerical Methods

- **Data Pipelining**
  - Data Formats & Translations
  - Cross Platform Application Integration
  - Live Smooth Streaming & Caching
- **Parallel Data Structures & Algorithms**
  - GPU Computing
  - Many Core (Xeon Phi)
  - Cluster / Grid Deployments

A Degree in Scientific Computing

- **Data Visualization**
  - Multidimensional & Multi-temporal
  - Machine Learning & Data Mining
  - Integrated Collaboration
- **Managing Massive Data Sets**
  - Real Time Big Data Search & Classification
  - Encoding, Compression, Transport
  - Reporting & Archiving
- **Modelling and Simulation**
  - Large-Scale Nature-Inspired & Hybrid Evolutionary
  - Highly Non-Linear Transient Systems
  - Shape-Preserving Response Prediction

Key Points

- Every **science research project** can benefit from even a touch of **scientific computing**
  - Better **statistics** & data **visualization** on posters
  - Compelling analysis from modelling & simulation
  - Novel integration of computation is a big ***differentiator!***
- Experimental Computational Mathematics (**ECM**) is a thing
  - Experiments don’t necessarily require lab equipment
  - Nowadays profound insights come only from **extended analysis**

Key Points

- It does not take thousands of lines of code to keep importance science moving right along…
  - You don’t have to be a professional programmer or know all the arcane aspects of computer languages
  - The closer you get to **cutting edge science**, the less likely you’ll be able to just “download an app” to accomplish what you need
- If you don’t know how to code…
  - You will at some point start to ***subconsciously*** limit the types of analysis you can perform because you will remain at the mercy of the available software
  - Should software shape your science, or instead, will you shape software to **advance** your science?