

CHEM 80: Numerical Methods for the Central Science

S. Kavassalis

Course Description: This course is an introduction to scientific computing in chemistry. Students will be introduced to the computational techniques used in modern chemistry research. Problems from multiple subdisciplines of chemistry will be chosen such that students will learn mathematical, computer-based problem-solving, data analysis, and data visualization relevant to the modern chemistry landscape.

Prerequisite(s): [CHEM042 HM](#) , ([CSCI005 HM](#) or [CSCI042 HM](#)), [MATH073 HM](#)

What is this all about?

Welcome to CHEM 80, an exploration of the *signals*, *software*, and *storytelling* behind modern chemistry research. This course is designed for students with some grounding in chemistry and computer science who want to delve deeper into the foundational and cutting-edge computational techniques used across chemistry, the "central science."

Chemistry is a field of signals - diverse and rich data sets that act as informational bridges, connecting chemical phenomena with the interpretable world. To make sense of measurable signals, we rely on the discipline of chemometrics. Chemometrics employs mathematical and statistical methods to derive meaningful information from chemical observations, enabling us to unravel the hidden patterns and stories in complex datasets. In this course, you will build foundational skills in signal processing, pattern recognition, and numerical modelling - powerful techniques critical to analyzing, processing, and interpreting chemical data and designing instrumentation and experiments.

In modern chemistry laboratories, software and instruments work hand-in-hand. Instrumentation coupled with commercial software often leaves much of the fundamental signal processing within a black box (you inject a sample, and out pops a crisp mass spectrum). In this course, we'll lift the lid on these hidden processes. We'll demonstrate how, at their core, modern scientific computing techniques provide the backbone for sophisticated software structures. Using Python, you'll write your own software - writing numerical recipes to turn raw signals into chemically meaningful datasets.

Beyond data and data processing lies the art of scientific storytelling - the synthesis and translation of data into meaningful narratives. In this course, we'll delve into how we can use those signals - those datasets - not just to make sense of chemical phenomena but also to communicate these discoveries effectively. Drawn from the diverse subdisciplines of our chemistry department - biochemistry, organic, inorganic, physical, and environmental - we'll engage with published papers and real datasets used in faculty research programs. Through these, you'll learn to interpret complex data and present them in coherent narratives, learning how to make publication-quality scientific figures that resonate with a larger audience.

Computing \rightleftharpoons Chemistry

While much of this course will be on developing the numerical methods that serve the chemical sciences, we will also study and contemplate the ways that fundamental chemistry has shaped computing and the potential it has to do so in the future. Selected readings will be chosen to demonstrate the power that innovation in chemistry has to shape computer science (from the Metropolis algorithm to Simulated Annealing, to chemically informed true or pseudo-random number generators, to (bio)-chemical and quantum computing).

I am not yet sold. Why take this course?

In the 2013 Nobel Prize in Chemistry announcement, the Royal Swedish Academy of Sciences wrote that "the computer is just as important a tool for chemists as the test tube" (1). Scientific computing has been called "the third pillar" of modern science (2), complementary to experiment and theory, carving a substantial place for itself in contemporary STEM education. The ability to work with data and create numerical models can significantly expand your chemical intuition and support your learning in future courses. Further, the scientific computing skills you will gain in this course apply not only to chemical research, as these foundational numerical techniques are used across the sciences and engineering within academia and industry.

1. The Nobel Prize in Chemistry 2013 - Press release - NobelPrize.org. Available from: <https://www.nobelprize.org/prizes/chemistry/2013/press-release/>
2. Skuse B. The third pillar. 2019 Mar; Available from: <https://physicsworld.com/a/the-third-pillar-of-science/>

What *numerical methods* will this course cover:

1. Basic statistics: The core statistical measures needed for chemometrics and for understanding pseudo-random number generators and stochastic processes.
2. Numerical Error and Precision: Understanding how errors can propagate in numerical calculations (chemometrics).
3. Linear Algebra: Topics include vector and matrix operations, determinant and inverse of a matrix, and eigenvalues and eigenvectors, and techniques for solving systems of linear equation. The unit will conclude with an application of these concepts in Principal Component Analysis (PCA).
4. Interpolation and Curve Fitting: How to estimate values between two known values in a series of data, introducing various methods such as polynomial and spline interpolation.
5. Data Clustering Techniques: Approaches to grouping a set of objects by shared attributes. This includes techniques such as K-means clustering and hierarchical clustering
6. Numerical Differentiation: Techniques for approximating derivatives of functions.

7. Numerical Integration: Techniques for approximating integrals of functions using methods like the trapezoid and Simpson's rule.
8. Optimization Techniques: Methods for finding roots of nonlinear equations, with a focus on algorithms to find minima and maxima of functions. We will cover classic algorithms like Gauss-Newton, Gradient Descent, and Greedy Algorithms that have a multitude of applications in chemistry.
9. Ordinary Differential Equations: Explicit (Euler and Runge-Kutta) and implicit methods for the solutions to initial-value problems.
10. Partial Differential Equations: Introduction to Finite Difference and Finite Element methods for simple PDEs.
11. Fourier Transforms: Convolution and correlation, power spectra.
12. Monte Carlo Methods: Use of random sampling for solving numerical problems, including the Metropolis Algorithm and Simulated Annealing.

We will also cover special material relevant to hot topics in chemistry research, building on the above techniques, including image processing and generation, optical character recognition (data "rescue"!), and the use of large language models for chemical problems.

What this course will not cover: In-depth chemistry beyond Core (i.e. no theory of quantum mechanical calculations). While we will discuss machine learning techniques and high-performance computing, we won't cover the math behind neural networks or fundamentals in computer vision, for example.

Goals of this course:

The goal of CHEM80, like all courses in the scientific computing/numerical methods family, is to build the mathematical and computational skills needed to solve real-world problems. Students coming out of this class will have developed strong logical reasoning, problem-solving, and programming abilities, coupled with a comprehensive understanding of mathematical principles. You will be prepared to approach and analyze scientific problems from a computational standpoint. Further, you will be equipped with a toolkit of algorithms and numerical methods that you can apply in various domains of science, technology, engineering, and mathematics. You'll also have an appreciation for the diversity of computing used in chemistry *and* for the ways that chemists have shaped the field of computing, too. Most importantly, you'll have a mindset of exploration and innovation, enabling you to undertake investigation and research, leading to new discoveries and advancements in chemistry and other fields.

Specific learning outcomes for students by the end of the course:

1. You will develop proficiency in computer programming using Python to manipulate and interpret raw chemical signals (from voltages to images).
2. You will be able to understand and apply the principles of chemometrics to derive meaningful information from diverse chemical measurements (including robust calibration and error propagation calculations).
3. You will be able to choose and apply appropriate statistical and numerical methods for data fitting, integration, ODEs solving, time series analysis, and pattern recognition.
4. You will understand the key role of scientific computing techniques in the backbone of sophisticated software structures in chemical research.
5. You will be able to communicate the interdisciplinary nature of scientific computing, recognizing the ways that chemistry and computer science intersect and influence each other.
6. You will be able to create publication-quality scientific figures in Python that effectively narrate chemical phenomena that resonate with a larger audience.