

Examples of computational math projects

Øyvind Ryan
Matematisk institutt, UIO

August 24, 2025

Contributions from UIO on Python based small math projects

- 1 Machine learning: Backpropagation and the chain rule
- 2 Condition numbers
- 3 Using SVD for image compression
- 4 Using Wavelets for image compression
- 5 Sound and Fourier analysis
- 6 Numerical eigenvalue algorithms: The companion matrix and the power method. The QR method.
- 7 Numerical eigenvalue algorithms: Jacobi's method.

These stem from compulsory projects in several courses:

- MAT1105 Linear algebra and Numerical methods
- MAT1110 Calculus and Linear algebra. Second semester. Simple matlab/python for linear algebra, 2D/3D plotting, symbolic computation).
- MAT1120 Linear algebra. Third semester (Simple matlab/python for linear algebra).

Contributions from UIO on Python based books in math courses (at UIO)

- MAT-INF2360 applications of linear algebra: [Linear Algebra, Signal Processing, and Wavelets - A Unified Approach](#). Python version
- MAT1105 Linear algebra and numerical methods. [Lineær algebra og numeriske metoder](#). Pythonversjon (in Norwegian only).

Contributions from UIO on Python based books in Physics courses

Book initiatives from Morten Hjort-Jensen and Anders Mølthe Sørensen.

- FYS1100 Mechanics and Modelling: Elementary Mechanics Using Python. A Modern Course Combining Analytical and Numerical Techniques
- The Physics321 project (Classical Mechanics I, PHY321 at MSU) [Online textbook](#). Initiative from Marcos Daniel Caballero (MSU) and Morten Hjorth-Jensen
- FYS1120 UIO, Electromagnetism: Elementary Electromagnetism Using Python
- FYS2160 UIO, Thermodynamics and Statistical Physics
Elementary Thermal and Statistical Physics Using Python (in preparation)
- FYS3150 Computational Physics

Some thoughts on why to introduce Python programming already in the first semester

Several models in Physics fall within the same class. If one implement general solvers early on (in parallel with the deduction of the theory in a math course), these can be reused in various models encountered in later courses.

Candidates for being taught early:

- Numerical differentiation/integration
- Numerical solution of differential equations
- Representation of numbers on computers
- Roundoff errors

See also [Integrating a Computational Perspective in Physics courses](#)).

Models that can be applied with general solvers

- Mechanics course: The classical pendulum

$$ml \frac{\partial^2 \theta}{\partial t^2} + \nu \frac{\partial \theta}{\partial t} + mg \sin \theta = A \cos(\omega t)$$

- Electromagnetism course: RLC circuit

$$L \frac{\partial^2 Q}{\partial t^2} + \frac{Q}{C} + R \frac{\partial Q}{\partial t} = A \cos(\omega t)$$

Both can be written as first order systems (the first with a non-linear term). General such systems can be taught in parallel with first semester programming course, and reused in the electromagnetism course later.

Two-point boundary value problems

The buckling beam:

$$R \frac{\partial^2 u(x)}{\partial x^2} = -Fu(x), \quad u(0) = u(L) = 0.$$

Discretization: $Av = \lambda v$ with

$$A = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

- There are analytical expressions for the eigenvalues/vectors.
- The Poisson equation leads to a system of the form $Av = f$.
- The Schroedinger equation: Discretization leads to tridiagonal matrices in general. Can only be solved numerically.

The CSE project at UIO

The CSE (Computing in Science Education) project started as a collaboration between professors in Mathematics, Physics, and Informatics:

- Knut Mørken (professor in Mathematics)
- Hans Petter Langtangen (professor in Informatics)
- Morten Hjort-Jensen (professor in Physics)
- Ander Malthe-Sørensen (professor in Physics)

The CSE project evolved into the Center for Computing in Science Education, a norwegian center of excellence in education. The goal of CCSE:

- to integrate computing as a natural tool in basic educations,
- to make the education research-near,
- to prepare students for an interdisciplinary workplace.

First semester course structure resulting from CSE

From 2007 CSE adapted computation and programming early on (first semester).

- **MAT1100** Calculus.
- **MAT-INF1100** Modelling and Computations.
 - Representation of numbers on a computer. Roundoff errors
 - Numerical methods: Interpolation, finding zeros of functions, numerical differentiation/integration, numerical solution of differential equations.
 - No linear algebra
- **IN1900** Introduction to Programming with Scientific Applications.

Example of synchronization across first semester courses: the trapezoidal rule, Newton's method, Euler's method.

- **MAT1100**: Defines the integral
- **MAT-INF1100**: The algorithm is deduced and discussed, with corresponding error terms.
- **IN1900**: The algorithm is implemented

Changes in recent revisions

- Some programs dropped MAT-INF1100. Taught the numerical methods themselves.
- Due to this, MAT-INF1100 was revised and adapted to math students.
- Resulting course: [MAT1105](#) Linear algebra and numerical methods. Contents:
 - Mathematical logic.
 - Introduction to linear algebra in euclidean spaces.
 - Matlab/Python.
 - Numerical methods: interpolation. Numerical differentiation/integration. Condition numbers

Strong wish to get linear algebra in the first semester for math students. They can thus get more advanced linear algebra earlier:

- [MAT1125](#) Advanced linear algebra (new third semester course)
- Linear algebra for non-math students (MAT1120) kept unchanged.

Math programs first semester: MAT1100, MAT1105, IN1900

- Bachelor program in pure and applied math: IN1900 is the only required course in informatics.
- Bachelor program in Data, Modeling and Computation: 2 or 3 informatics courses are required.