Summary
Visual computing and machine learning are characterized by their reliance on numerical algorithms to process large amounts of information such as images, shapes, and 3D volumes. This course will familiarize students with a range of essential numerical tools to solve practical problems in this area.

Content
This course provides a first introduction to the field of numerical analysis with a strong focus on visual computing and machine learning applications. Using examples from computer graphics, deep neural networks, geometry processing, computer vision, and computational photography, students will gain hands-on experience with a range of essential numerical algorithms.

The course will begin with a review of floating point arithmetic and error propagation in numerical computations. Following this, we will study and experiment with several techniques that solve systems of linear and non-linear equations and perform dimensionality reduction. Since many interesting problems cannot be solved exactly, numerical optimization techniques constitute the second major topic of this course. We will take an extensive look at automatic differentiation, the mechanism underlying popular deep learning frameworks such as PyTorch and Tensorflow. The course concludes with a review of numerical methods that introduce randomness to solve problems that would otherwise be intractable.

Students will have the opportunity to gain practical experience with the discussed methods using programming assignments based on Scientific Python.

Keywords
Visual computing, machine learning, numerical linear algebra, numerical analysis, optimization, scientific computing

Learning Prerequisites
Required courses
MATH-101 (Analysis I) and MATH-111 (Linear Algebra).

Recommended courses

Important concepts to start the course
Students are expected to have good familiarity with at least one programming language (e.g. C/C++, Java, Scala, Python, R, Ruby...). The course itself will rely on Python, but this is straightforward to learn while taking the course.
During the first weeks of the semester, there will be tutorial sessions on using Python and Scientific Python.
Learning Outcomes
By the end of the course, the student must be able to:

• Develop computer programs that use numerical linear algebra and analysis techniques to transform and visualize data.
• Reason about ways of structuring numerical computations efficiently.
• Analyze the numerical stability of programs built on top of floating point arithmetic.
• Recognize numerical problems in visual computing applications and cast them into a form that can be solved or optimized.

Teaching methods
Lectures, interactive demos, theory and programming exercises

Expected student activities
Students are expected to study the provided reading material and actively participate in class and in exercise sessions. They will be given both theoretical exercises and a set of hands-on programming assignments.

Assessment methods
1. Continuous assessment during the semester via project assignments (35%)
2. Final exam (65%)

Resources
Bibliography
Slides and other resource will be provided in class.

Ressources en bibliothèque
• Scientific Computing: An Introductory Survey / Heath
• Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics / Solomon

Prerequisite for
Although it is not a strict prerequisite, this course is highly recommended for students who wish to pursue studies in the area of Visual Computing, in particular: CS-341 (Introduction to computer graphics), CS-440 (Advanced computer graphics), CS-442 (Computer vision), CS-413 (Computational Photography), CS-444 (Virtual Reality), and CS-445 (Digital 3D geometry processing)