



**This learning unit is not open to incoming exchange students!**

Teacher(s)	Jodogne Sébastien ;
Language :	French
Place of the course	Charleroi
Prerequisites	<p>This course assumes that you have acquired the basic notions of programming (instructions, variables, loops, conditions, etc.) and the programming methodology in Python as taught in the LSINC1101, LINFO1101, or LEPL1401 courses, as well as the basics of the Java language such as taught in the LSINC1402 or LEPL1402 courses.</p> <p>This course also assumes that the basic notions of algebra and analysis covered by the LSINC1111 or LINFO1111, and LSINC1112 or LINFO1112 courses have been acquired.</p> <p><i>The prerequisite(s) for this Teaching Unit (Unité d'enseignement – UE) for the programmes/courses that offer this Teaching Unit are specified at the end of this sheet.</i></p>
Main themes	<ul style="list-style-type: none"> <li>• Representation of floating point numbers</li> <li>• Rounding error problem and error propagation (discussion for the methods below)</li> <li>• Notion of convergence and stopping criteria of iterative methods</li> <li>• Representation of matrices, efficient multiplication of matrices</li> <li>• Resolution of linear systems, including iterative methods</li> <li>• Interpolations and regressions</li> <li>• Numerical integration, numerical differentiation</li> <li>• Resolution of ordinary differential equations: problems with initial value</li> <li>• Resolution of nonlinear equations (function roots), application to simple one-dimensional optimization problems (including notion of minimum / maximum local or global)</li> </ul> <p>Since the course is intended for IT professionals, the emphasis will be on practical implementation of these methods.</p>
Learning outcomes	<p><b>At the end of this learning unit, the student is able to :</b></p> <p>Given the learning outcomes of the "Bachelor in Computer science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes:</p> <p>A.A.   •S1.G1, S1.I3  S1.G1,   •S2.2, S2.4  S1.3    •S5.1  -  A.A.   Students who have successfully completed this course will be able to:  S2.2,  S2.4    • model a simple problem using the proper mathematical notation;  -        • identify classical numerical methods suitable for solving a simple problem expressed mathematically;  -        • choose, on the basis of precise criteria, the most effective method for numerically solving such a problem,  A.A.  S6.1    • implement a numerical resolution of this simple problem;            • explain the problems related to the numerical resolution of equations and their impacts: rounding errors, convergence, stopping criteria.</p>
Evaluation methods	<p>The examination will be a written, on-site test with open-ended questions. It will cover all the material from the lectures and practical sessions. The examination grade will contribute 90% to the final evaluation, while the remaining 10% will come from continuous work and attendance during the practical sessions. The grade for continuous work and attendance is retained throughout the academic year and will not be re-evaluated during the second exam session.</p> <p>Continuous assessment is based on the practical sessions, with a single overall mark awarded after the end of the last session. Failure to comply with the methodological instructions communicated by the teacher, particularly with regard to the use of online resources or collaboration between students, in an assignment will result in an overall mark of 0 for the continuous assessment.</p> <p>In particular, the use of generative AI tools and any collaboration is strictly prohibited for the assignments. The distribution or exchange between students of (fragments of) code is not allowed by any means (GitHub, Facebook, Discord...), and this even after the deadline for submission of assignments/homeworks.</p>

Teaching methods	By presentation of the concept and by implementation.
Content	The philosophy of the course is to introduce numerical methods by describing and, above all, implementing concepts from algebra and mathematical analysis courses. The aim is to develop algorithms while observing the limits of implementing a mathematical concept: data representation (numbers, etc.) and error handling (calculation, stability, propagation, etc.).
Inline resources	<a href="https://moodle.uclouvain.be/course/view.php?id=5751">https://moodle.uclouvain.be/course/view.php?id=5751</a>
Faculty or entity in charge	SINC

<b>Programmes containing this learning unit (UE)</b>				
Program title	Acronym	Credits	Prerequisite	Learning outcomes
Bachelor in Computer Science	<a href="#">SINC1BA</a>	5	<a href="#">LSINC112</a>	