Universitat Autònoma de Barcelona

2019/2020

Numerical and Probabilistic Methods

Code: 104395
ECTS Credits: 6

Degree	Type	Year	Semester
2503740 Computational Mathematics and Data Analytics	OB	2	1

Contact

Name: Joan Carles Artés Ferragud

Email: JoanCarles.Artes@uab.cat

Use of Languages

Principal working language: catalan (cat)

Some groups entirely in English: No

Some groups entirely in Catalan: Yes

Some groups entirely in Spanish: No

Other comments on languages

Any question may be answered in english by the professor

Teachers

Salvador Borrós Cullell

Prerequisites

It is advisable to have done at least one course of analysis, linear algebra and numerical calculus.

Objectives and Contextualisation

In the analysis courses, we were taught to compute areas of functions by means of integrals, but also that not all
functions have an integral that can be expressed in a finite amount of elementary functions.
In the algebra courses we were taught that a polynomial of degree n has n roots (real or complex), but also that not all
the degree 5 or higher polynomials have necessarily to be solved by means of radicals. And also that many other non-polybromic equations cannot be explicitly resolved.
In the algebra courses we have been taught to solve systems of linear equations using the Cramer method, but do you know
that solving a 20x20 system in this way would need more time than the universe has?
In the first course of numerical calculus, some methods were introduced to solve this type of problems, not by exact way,
but by numerical approximations. This way of tackling problems presents some advantages and some drawbacks.
The main advantages are that in this way you can solve problems that would otherwise be impossible to solve. One drawback is that the exact solution is never found but a numerical approximation. This is compensated
by the advantage that we can decide a priori the degree of precision with which we want to obtain the solution and this can be
as great as we wish (and we have a computer good enough to do so with a reasonable time). Another

disadvantage
is that the numerical calculation is permanently fighting against all kinds of errors in the initial data, in the introduction
data, and in rounding up operations. These bugs also propagate as we do more and more operations with data already corrupt. Therefore, numerical calculation methods should also be able to deal with this problem.
The first course of numerical calculus ended with the resolution of integrals in numerical form. In this second yearwe
we will continue doing with new more powerful methods.
Another way of calculating integrals, for more unlikely it may seem, is by means of random methods. These methods
have traditionally been called Montecarlo methods as a paradigm of the Mecca of gambling. We will see how with very simple
(although long calculation) methods it is possible to calculate function areas in one or more dimensions, which would otherwise be impossible to calculate.
In this course we will present a new type of mathematical problems that are very common in the modelling of problems of
real life, in fact, few real life problems end up simply needing the calculation of an integral or solution of a polyomial equation. In fact, most of the problems that arise in real life end up in probems of differential equations, whether ordinary or partial. In a problem of differential equations, the goal is not to find a number to solve a problem, but to find a function.
Some problems of ordinary differential equations can be solved in exactly way and this will be done in more detail
in the second semester subject which is called ordinary differential equations. We will not enter here to discuss if it had been
more convenient to permutate these two semesters. In fact, once understood what an ordinary differential equation is,
there is little more to start solving ordinary differential equations in numerical form. However, a pair of weeks will be devoted to introduce this concept.

Competences

- Apply basic knowledge on the structure, use and programming of computers, operating systems and computer programs to solve problems in different areas.
- Calculate and reproduce certain mathematical routines and processes with ease.
- Design, develop and evaluate efficient algorithmic solutions to computational problems in accordance with the established requirements.
- Formulate hypotheses and think up strategies to confirm or refute them.
- Make effective use of bibliographical resources and electronic resources to obtain information.
- Relate new mathematical objects with other known objects and deduce their properties.
- Students must be capable of applying their knowledge to their work or vocation in a professional way and they should have building arguments and problem resolution skills within their area of study.
- Students must be capable of communicating information, ideas, problems and solutions to both specialised and non-specialised audiences.
- Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.
- Use computer applications for statistical analysis, numerical and symbolic computation, graphic visualisation, optimisation and other to experiment and solve problems.
- Using criteria of quality, critically evaluate the work carried out.

Learning Outcomes

1. Contrast, if possible, the use of calculation with the use of abstraction in solving a problem.
2. Control errors produced by machines when calculating.
3. Describe the concepts and mathematical objects pertaining to the subject.

4. Develop autonomous strategies for solving problems such as identifying the ambit of problems within the course, discriminate routine from non-routine problems, design an a priori strategy to solve a problem, evaluate this strategy.
5. Evaluate and analyse the complexity of computational algorithmic solutions in order to develop and implement that which guarantees best performance.
6. Evaluate the advantages and disadvantages of using calculation and abstraction.
7. Handle specific scientific software for the application of numerical algorithms or the automatic realisation of symbolic calculations aimed at solving particular problems.
8. Identify the essential ideas in the demonstration of ceratin basic theorems and know how to adapt these to obtain other results.
9. Make effective use of bibliographical resources and electronic resources to obtain information.
10. Programme mathematical-calculation algorithms.
11. Select and use algorithmic structures and the representation of appropriate data to solve a problem.
12. Students must be capable of applying their knowledge to their work or vocation in a professional way and they should have building arguments and problem resolution skills within their area of study.
13. Students must be capable of communicating information, ideas, problems and solutions to both specialised and non-specialised audiences.
14. Students must have and understand knowledge of an area of study built on the basis of general secondary education, and while it relies on some advanced textbooks it also includes some aspects coming from the forefront of its field of study.
15. Understand the basic concepts in the structure and programming of computers.
16. Understand the internal functioning of computers and be critical of the results that they provide.
17. Use operating systems and programmes commonly used in various fields.
18. Using criteria of quality, critically evaluate the work carried out.
19. Verify and ensure the correct operation of an algorithmic solution in accordance with the requirements of the problem to be resolved.

Content

1.- Introduction to ordinary differential equations
2.- Numerical integration. Newton-Côtes and Gaussian methods
3.- Monte Carlo methods for calculating areas
3.1- Generation of random variables
4.- Numerical integration of ordinary differential equations (one variable)
4.1- Initial value problem
4.1.1- Euler method
4.1.2- Order of consistency and convergency
4.1.3- Taylor methods
4.1.4- Multistep methods
4.1.5- Runge-Kutta methods
4.1.6- Variable Step methods
4.2- Problem of values at the border
4.2.1- Shooting method
4.2.2- Split Differences method

Methodology

The tools of mathematics, and very particularly those of numerical calculus need to be learned and practiced. Simply memorizing a formula or a theorem, if we have not applied it at any time, it is possible that it does not go to the first tries. In addition, the numerical calculation tools have been done to solve problems that need a lot of calculations and these calculations will normally be done by a computer, with a program that we have done. Even if the program is made by another person, it is convenient to know how it works in order to detect if any result can be unstable or incorrect.

But we can not make a program to apply a method if we previously have not practiced it, even if it is with a simple or even trivial problem that would not even have a need of the numerical method.

The theoretical sessions will be dedicated to the teacher's presentation of the different methods and their analysis. The exhibition of the methods will be accompanied by examples of their behavior, carried out with computers, which are aimed at both facilitating the understanding of the method and motivating their analysis.

Problems of theoretical and calculation types are resolved in the problem sessions. In the case of calculation problems, there will be some requiring the use of a calculator or even the use of a computer. In the latter case, the problems will not be computationally intensive, so the necessary algorithms may be implemented quickly in a numeric language interpreter or even in a spreadsheet (Excl). The teacher will combine the resolution of problems for the whole class, on the part of a student throughout the class and for all students at the same time, in a group, with the teacher's help.

The computer practice sessions form part of the subject dedicated to introducing scientific computing. They will be dedicated to the solution of computationally more intensive problems, which will be implemented in a compiled language. In solving these problems students will progressively construct their personal library of routines that implement basic numerical methods.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Lab exercices	15	0.6	18, 6, 5, 1, 2, 16, 15, 3, 4, 8, 7, 10, 14, 13, 12, 11, 9, 17, 19
Problem classes	8	0.32	18, 6, 5, 1, 2, 16, 15, 3, 4, 8, 7, 10, 14, 13, 12, 11, 9, 17, 19
Theoretical classes	30	1.2	18, 6, 5, 1, 2, 16, 15, 3, 4, 8, 7, 10, 14, 13, 12, 11, 9, 17, 19
Type: Autonomous			
Study, exercises and preparation of lab exercises	92	3.68	18, 6, 5, 1, 2, 16, 15, 3, 4, 8, 7, 10, 14, 13, 12, 11, 9, 17, 19

Assessment

The course evaluation will take place from three activities:

1. Partial Exam (EP): Exam of part of the course, with theoretical questions and problems.
2. Final Exam (EF): Exam of the whole subject, with theoretical questions and problems.
3. Computer Labs (PR): Delivery of code and a report.

In addition, students will be able to take a retake exam with the same characteristics as the EF exam. The practices will not be recoverable.
It is a prerequisite to overcome the course that Max (0.35 * EP + 0.65 * EF, EF, ER) > = 3.5 and PR > = 3.5.
The final grade of the course will be 0.6 * MAX (0.35 * EP + 0.65 * EF, EF, ER) + 0.4 * PR
The honour qualifications will be awarded to the first complete evaluation of the course. They will not be withdrawn if another student obtains a higher qualification after considering the ER exam.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes

Computer Program	0.4	0	0	18, 6, 5, 1, 2, 16, 15, 3, 4, 8, 7, 10, 14, 13, 12, 11, 9, 17, 19
Final exam	0.39	3	0.12	18, 6, 5, 1, 2, 16, 15, 3, 4, 8, 7, 10, 14, 13, 12, 11, 9, 17, 19
Partial exam	0.21	2	0.08	18, 6, 5, 1, 2, 16, 15, 3, 4, 8, 7, 10, 14, 13, 12, 11, 17, 19

Bibliography

Basic bibliography:

A. Aubanell, A. Benseny, A. Delshams. Eines bàsiques de càlcul numèric. Manuals de la UAB 7, Publ. UAB, 1991.

M. Grau, M. Noguera. Càlcul numèric. Edicions UPC, 1993.

J.D. Faires, R. Burden. Métodos numéricos, 3a ed. Thomson, 2004.

G. Dahlquist, A. Björk. Numerical methods. Prentice Hall, 1964.

R. Burden, J.D. Faires. Numerical analysis, 6a ed. Brooks/Cole, 1997. En castellà: Análisis numérico, 6a ed., International Thomosn, 1998.

G. Hämmerlin, K.-H. Hoffmann. Numerical mathematics. Springer, 1991.

Advanced bibliography:

E. Isaacson, H.B. Keller. Analysis of numerical methods. Wiley, 1966.

J. Stoer, R. Bulirsch. Introduction to numerical analysis, 3a ed. Springer, 2002.

A. Ralston and P. Rabinowitz. A first course in numerical analysis. McGraw-Hill, 1988.

A. Quarteroni, R. Sacco and F. Saleri. Numerical Mathematics. Springer, 2000.