

## A learner-centered approach to design a Computational Finance module in higher education

**Adamaria Perrotta**

School of Mathematics and Statistics, University College Dublin, Ireland.

---

### **Abstract**

*In this paper, we describe our design of ACM30070 “Computational Finance”, a core module in the BSc in Financial Mathematics in the School of Mathematics and Statistics. The over-arching purpose of this module is to help students to develop mathematical, statistical and coding skills, along with significant knowledge and critical thinking, that allows them to effectively construct, manipulate and visualize financial datasets and to build financial mathematical models. The use of computation and a FinTech software (FinCad Analytics) are pointed out as essential to facilitate sensemaking in computational finance. More broadly, we discuss the education-research based rationale behind the “learning by doing” and “flipped classroom” institutional models that we have chosen for ACM30070, and we show how the modern “inclusive” definition of computation has been embedded into the learning activities. An accurate description of the design principles and implementation is also presented. At the end of the paper, we briefly introduce a discipline-based education research that will follow from this module design.*

**Keywords:** *Computational finance education; financial modelling; data analysis, learning by doing.*

---

## **1. Introduction**

In the last few decades, the need for data analysis and computational modelling skills have become critical to success in both industry and academia. Workers and researchers should be able to successfully analyze and interpret datasets and use computational modelling in most of the branches of modern science. Given the importance of computation in science, the number of computational undergraduate modules within traditional science programmes is increasing. In particular there has been increased attention on computation in mathematics subjects (Lockwook *et al.*, 2019), and consequently traditional undergraduate education has been adapted in different ways. Whilst some researchers refer to a body of literature in Finance Education (Diamond & Smith, 2011), (Hoadley *et al.* 2015), (Hoadley *et al.*, 2016) only a limited portion of these specifically investigate the *Computational Finance curriculum*; thus, the area is under-researched. This is because Computational Finance is a relatively new discipline, is highly interdisciplinary and very often considered as a sub-section of the Finance curriculum. In this paper, we describe the design process and the final version of ACM30070 Computational Finance, that has been delivered in Spring '21. We focus on the enquiry-based and student-centred learning educational-research model chosen to design it, along with the importance of computation in sensemaking to the subject. What we describe here is the result of a long process started in 2016 that went through several improvements. Taking into account the experience of Covid-19 disruptions, we also show how all the designed practices can be implemented in both f-2-f and blended/online delivery.

## **2. Course Design and Implementation Plan**

### **2.1. Design Process**

The broad goals of ACM30070 “Computational Finance” come from the creation of a new BSc in Financial Mathematics in the School of Mathematics and Statistics, started in 2015/16. In 2016/2017 we have in fact been asked to design this module, ACM30070, core in stage 3 semester 2, with the aim to make it the “jewel” of the programme, providing at the same time a practitioner-oriented and deep mathematical education in implementing financial models for derivatives pricing. We jointly worked on the big picture of BSc design and the module outline. Therefore, we designed the module with the purpose of activating prior and integrative learning (Norman & Schmidt, 1992) and vertical and horizontal coherence within the programme (Blackshields, *et al.*, 2014). We closely worked with other STEM researchers, and we agreed on designing practices and activities in order to reciprocally use computational thinking to enrich the financial mathematics mastery and the financial mathematics context to enrich the computational learning. To embed real-world experience and make the study of computational finance as authentic as possible, a similar set of discussions took place with representatives of financial companies. New graduates should be able to manipulate and

analyze data to extract usable information from it, to capture the salient features of a financial system, to critically build a model, and to communicate outcomes to co-workers and supervisors. Finally, being part of the “University Discipline-Based Education Research in STEM” and of the “Community Engaged T&L” groups, we felt the need to create a module that exemplified the current best practices in the undergraduate research literature. The module was delivered on a pilot basis in Spring ’18, then it was improved after critical reflections on the first outcomes. The current redesign integrates the definition of *inclusive computation* (Caballero *et al.* 2018) and the computational practices in the *Weintrop et al. Computational Practice Taxonomy* (2016): computation in ACM30070 is no more a simple skill but it represents one of the constitutive pillars to master financial mathematics.

## 2.2. Learning Goals

The purpose of ACM30070 is to provide a practitioner-oriented education in implementing financial models and embed computational thinking in mathematics and finance contexts, with the aim of bringing current educational efforts in line with the increasing demand for problem-solving and quantitative skills. Students will learn how pure & applied mathematics and coding can be harmonically integrated to solve a real-world financial problem. The learning goals have been set-up referring to the principles of *backward course design* (Wiggins & McTighe). Upon successful completion of the module students will be able to: apply financial mathematical theory and quantitative methodologies to real-world situations; understand the industry practice; identify salient features of a financial system that can be translated into a model; judge the suitability of model, critically understanding its limitations; write computer codes to solve common problems in the financial sector; collect, create, manipulate, analyze financial datasets; understand basic numerical methods and use them to solve problems; synthesize and communicate outcomes of a scientific computing problem.

## 2.3. Theoretical Framework and Pedagogical Motivations

To design the overall course structure, we have referred to the *Seven Research-Based Principles for Smart Teaching* described in (Ambrose *et al.* 2010). Those principles focus on how people learn starting from empirical evidence; for this reason, they can be easily applied in classes where students have a modest background in a subject. This is the case of students enrolling in ACM30070, since they have little coding and financial modelling experience. Among the seven, we have selected the following principles:

**Students’ prior knowledge can help or hinder learning:** students build new concepts on robust prior knowledge, looking for links between previously acquired and new knowledge, that help them construct increasingly complex structures. For this reason, we have built the syllabus of ACM30070 referring to the ones of the two pre-requisite modules “PDE for FM” and “Stochastic Models” and of the concurrent core module “Foundations for FM”.

**To develop mastery, students must acquire component skills, practice integrating them, and know when to apply what they have learned:** since the way knowledge is organized influences how students learn, all new concepts have been presented in a scaffolded manner, and related practices have consequently been designed and integrated between each other. To drive students in this path, pre-class, in-class, post-class activities have been designed. As an example, students are asked to implement the Black-Scholes model in Python at home, taking a similar VBA code as a reference. In class, they bring their codes and work in groups, making comparisons, debugging, doing input/output data analysis. After the class, they do a homework in which more complex modelling and data analysis are involved, under scenarios where they have to take critical decisions on modelling, starting from acquired knowledge.

**Students' motivation determines, directs, and sustains what they do to learn:** the module is offered timely with students' internship summer programme. In order to match the context, map the module outcomes to external drivers and provide students with relevant experience and motivation, practices have been designed to be pertinent to their future profession (Barrett & Moore, 2011), (Schmidt *et al.* 2009). The embedding of Fincad (FAS) software in the activities has also been essential. Indeed, FAS is a financial software library widely used in the industry but also very helpful in education, since it is intuitive to use and very well documented on both theory and computational side, which is essential for self-study.

**Goal-directed practice coupled with targeted feedback enhances the quality of students' learning:** to improve student engagement and satisfaction and provide multiple options for self-regulation, each assignment addresses one or more learning objectives and have different goals. In-class formative assignments are developed in groups of 4-5 students during the labs (*knowledge as process of construction between peers*). Homework are tailored to individual reflection, critical and creative thinking and problem-solving skills. The final group project is intended to develop the "*system thinking practice*" category in the Computational Practice Taxonomy and also to stimulate long-life skills like creativity, teamwork, project management, communication, negotiation and ethical behavior. Written and oral feedback are provided for each summative and formative assessment component.

**To become self-directed learners, students must learn to monitor and adjust their approaches to learning:** in all assessment components students are encouraged to go beyond thinking about the single tasks; we propose reflective and critical questions (based on *why* instead of *what*, or *provide an example* of instead of *reproducing a proof*), practical open-ended exercises (e.g. design a strategy to hedge the risk of your portfolio), inclusive coding activities like completing a piece of code, inspecting and comparing computational outputs, check the suitability of a financial model through computation.

## 2.4. Module outline

Starting from the learning goals and theoretical framework, we referred to the available literature on *threshold concepts* in Finance for choosing the course contents and develop its structure. Diamond and Smith in (Diamond & Smith, 2011) focus on approaches to teaching threshold concepts in Quantitative Finance, with five concepts suggested. Hoadley *et al.* in (Hoadley, Wood, Kyng, & Tickle, 2015) and (Hoadley, Wood, Kyng, & Tickle, 2016) have further investigated the Quantitative Finance curriculum using threshold concepts as both a theoretical framework and a research methodology. They have identified ten concepts (including the Diamond's five) that are clearly endorsed as "*jewels in the Quantitative Finance curriculum*". Within this list we distilled the followings - Pricing and Valuation, Arbitrage, Expected Value, Change of Measure, Risk Neutrality, Ito's Lemma and Brownian Motion, Hedging. Also, we have considered the results in (Khalo & Bayaga, 2015), where the most common errors committed by learners in Quantitative Finance are identified. We have mapped the above selected threshold concepts into the Computational Practice Taxonomy and we have accordingly designed activities referring to the Kolb's learning cycle (Kolb, 1984), taking the "concrete experience" as the start and the end of the cycle. Weekly course contents & progression are shown in Table 1.

**Table 1. Course Contents and Progression.**

Week	Modelling Concept	Context/Application	Computational practices	Summative Assessment
1	Arbitrage, Martingale	Derivatives Pricing	Fincad intro	
2	Black Scholes	Vanilla Opt Pricing	Procedures & Functions	
3	Greeks and Impl Vol	Implied Volatility	Loops, if/else condition	
4	Binomial Model	Vanilla Opt Pricing	Loops and data analysis	
5	Binomial Model	American/Exotic Opt	Loops, if/else	Hw 1
6	Ito's Processes	GBM	Random Numbers generators, plotting	
7	Monte Carlo Method	Risk-Neutral Pricing	Evaluating Conditionals	
8	Monte Carlo Method	Greeks Estimate	GBM and Indicators	Mid-term
9	Monte Carlo Method	Variance Reduction	Arrays Manipulation	
10	Finite Difference	Expl/Impl Methods	Discr Approx. and grids	Hw 2
11	Finite Difference	Crank-Nicolson	Arrays Manipulation	
12	Finite Difference	Barrier Options	Arrays Manipulation	G P

### ***2.5. A typical week of class***

Students attend 4 slots of 50 minutes each per week (two lectures, one tutorial, one lab). To pre-activate individual learning and reflection, materials are uploaded on the course management system a couple of days before the classes. The materials typically include slides, notes and short videos for what concerns the lectures, and proper formative assignment like short, guided programming activities and/or working questions for the tutorials. The lectures are devoted to the financial modelling part, so they usually begin with a brief discussion on the reading materials to allow students to address questions/doubts, then the instructor moves on to the topic of the day. Tutorials and Labs are intended for several computational practices and problem-solving activities. Tutorials start with a walk-through of particular pieces of the pre-class assignment, which helps put each student on equal footing before starting the day's activities. Then, computational practices (see Table 1) are performed both individually or in small groups, so that students learn through problem-solving and comparison with peers. To further enhance the “flipped-classroom” and “learning by doing” models, Lab activities are fully student-led and the lecturer, the tutor and the teaching assistant facilitate the practices. There is no pre-class assignment, practices are entirely covered in class; students can download working questions only one hour before the class to read them. Labs start with students working in groups on modelling, pseudocoding, data analysis and other related activities. After 30 minutes, each group choose a representative to present their outcomes. The lectures and the tutor do one-two checkpoints during the first part of the lab to get students to think critically about the outcomes, then drive them in getting conclusions in the second part. The teaching assistant acts as a moderator in groups dynamics and takes field notes. Groups are constructed by the lecturer and stay the same for the term. Groups' components have similar overall capability on average; gender and possible minorities balance are taken into account. Finally, post- class annotated slides, question solutions and full working codes are provided.

### ***2.6. Course Technology, Staff Selection and Logistic (in class and blended activities)***

To reach the learning objectives and implement the pedagogical principles described in section 2.3, we had to leverage a variety of digital technologies and certain course logistics. In terms of programming languages, students learn VBA, Python and Fincad. In the financial industry, in fact, VBA and Fincad are frequently used to create complex financial spreadsheet models. Python programming is easy to learn, the code is compact and in general highly readable. When it comes to numerical or financial algorithms, the syntax is close to mathematics. There is a wealth of powerful libraries available and it is free and open source. Codes are developed in Spyder notebooks, which constitutes a critical component of the module since they allow raw code, easy access to the results of its execution throughout, plots and powerful data analysis. Beyond the technology, also the “physical” classroom space has been crucial. All activities take place in A.L.E. rooms, equipped with round tables, movable

whiteboards and plugs for laptops so that students and instructors can easily collaborate. To adapt this class format to off-campus teaching, the classes are live-streamed on Zoom and they are recorded, to take into account students' connections issues. The round tables and class discussions have been substituted with the "breakout rooms" and "pool" features in Zoom and practices are currently designed so that they can be delivered either f-2-f or online. Finally, the choice and training of the tutor and the teaching assistant have been essential to develop a collaborative learning environment. The tutor attended the module on 18/19, being one of the best-performing students. He was a teaching assistant in 19/20 and started a PhD in 20/21. He contributed to the design of lab practices, bringing in his perspective of former student and then assistant. The teaching assistant is following the same pattern, he attended the module on 19/20, and received the training to be assistant in 20/21.

### ***2.7. Assessment: summative and formative***

In addition to the various forms of formative assessments we described above, we built multiple mechanisms for summative assignment. The breakdown is: two homework (20%), midterm (20%), group project (10%), engagement grade (10%) and final exam (40%). Homework, midterm, final exam focus on reflective and critical questions, practical open-ended exercises and coding activities. The group project relies on a financial problem that is entirely new for students. Only starting references are provided, students have to conduct their own research and propose a modeling solution. The deliverables are: a python code, a report and an oral presentation. Finally, students are expected to be participative, mutually supportive and responsible for both their own and each other's learning during the labs. After each lab, they are invited to fill out a Google form survey to critically reflect on the day practice. Each completed survey is worth 1%, up to a total of 10% of engagement grade.

## **3. Conclusions**

It is well known that students can progress through computational and applied mathematics courses, sometimes with good grades, and still lack understanding of the underlying principles, relations and concepts. In this paper, we have described the ACM30070 design process. The over-arching purpose of this module is to help students to develop a robust knowledge in financial modeling and both quantitatively and qualitatively compare models' outcomes with real-world data. The use of an inclusive definition of computation, along with an enquiry-based educational model have been central to master the subject. At the moment, the practices are "module delivery" independent, since they are effective and easily adaptable in both f-2-f or live streaming format. We are currently conducting an evidence-based research study to evaluate the impact of the proposed methodologies and practices on students significant learning, engagement and sensemaking to financial mathematics. Ethical Approval for this research study was obtained in Jan '20 and data has been collected since

Jan '21, with very positive outcomes. Unfortunately, we are not presenting here the research results because data cannot be reported in any publications until final grades are officially released.

## References

- Ambrose, S.A. Bridges, M.W., DiPietro, M., Lovett, M.C., Norman, M.K. (2010), *How Learning Works: Seven Research-Based Principles for Smart Teaching*, Wiley.
- Barrett, T., & Moore, S. (2011). *An introduction to problem-based learning. In New approaches to problem-based learning: Revitalizing your practice in higher education.* New York: Routledge.
- Blackshields, D., Cronin, J., Ryan, A., Higgs, B., McCarthy, M., & Kilcummin, S. (2014). *Integrative Learning: International Research and Practice.* New York: Routledge.
- Caballero, M.D. & Hjorth-Jensen, M. (2018), Integrating a Computational Perspective in Physics Course, *New Trends in Physics Education Research*, Nova Science Publishers.
- Diamond, R., & Smith, H. (2011), Threshold Concepts: A Disciplinary Enquiry in Quantitative Finance. *Social Science Research Network*.
- Hoadley, S., Wood, L., Kyng, T., & Tickle, L. (2015), Threshold concepts in Finance: conceptualizing the curriculum. *International Journal of Mathematics Education in Science and Technology*, 46(6), 824-840.
- Hoadley, S., Wood, L., Kyng, T., & Tickle, L. (2016), Applying threshold concepts to finance education. *Education + Training*, 58(5), 476-491.
- Khalo, X., & Bayaga, A. (2015). Analysis of errors due to deficient mastery of prerequisite skills, facts and concepts: A case of Financial Mathematics. *The Independent Journal of Teaching and Learning* 10, 98-113.
- Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development (Vol. 1)*. Englewood Cliffs, NJ: Prentice-Hall
- Lockwook, E., DeJarnette, A.F., Thomas, M., (2019). Computing as mathematical disciplinary practice. *Journal of Mathematical Behavior*, 54, 100688.
- Norman, G., & Schmidt, H. (1992). The psychological basis of Problem-based Learning: A Review of Evidence. *Academic Medicine*, 67(9), 557-565.
- Schmidt, H., van der Molen, H., te Winkel, W., & Wijnen, W. (2009). Constructivist, problem-based learning does work: A meta-analysis of curricular comparisons involving a single medical school. *Education Psychologist* 44(4), 227-249.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., Wilensky, U., (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25, 127-147.
- Wiggins, G.P., & McThige, J., *Understanding by design*, Association for Supervision and Curriculum Development, Expanded 2nd ed.