An Undergraduate Mathematics Course on Networks

Mason A. Porter

1 Introduction

The study of networks incorporates tools from a diverse collection of areas—such as graph theory (of course), computational linear algebra, dynamical systems, optimization, statistical physics, probability, statistics, and more—and it is important for applications in just about any area that one can imagine [1, 2]. It is thus important to teach courses on networks in mathematics, statistics, computer science, social and organizational sciences, and other disciplines. Graph theory is an old subject, and mathematics departments have taught courses in it for decades. One can also find courses on various aspects of networks in departments such as statistics, computer science, sociology, and others. Many of them have existed for quite a while, but the notion of studying the mathematics of networks—as involving subjects like graph theory, but distinct from it in crucial ways—is relatively new, and both undergraduate and graduate mathematics curricula need to include courses with such a focus.

The importance of teaching courses on networks in mathematics departments goes far beyond the establishment of the topic of "networks" as having a distinct identity from subjects such as graph theory. The study of discrete data has undergone a revolution, and people with mathematics degrees need to be well-versed in it. Many mathematics majors go on to careers in some form of data science (in academia, industry, government, and elsewhere) [3], and mathematics curricula need to prepare them for these careers. One way to do this is to offer a suite of courses to develop a "discrete structures and data science" track through degree programs in the mathematical sciences, including through a mathematics major itself.

M. A. Porter (✉)
University of California Los Angeles, Los Angeles, CA, USA

C. B. Cramer et al. (eds.), *Network Science In Education*,
https://doi.org/10.1007/978-3-319-77237-0_1

Students who undertake such a track focus predominantly on discrete structures, and they should master elements of theoretical ("pure") mathematics, statistics, applied mathematics (including mathematical modeling), computer science, and data analysis. In addition to networks (the science of connectivity), such students should learn about subjects such as optimization, probability theory, machine learning, information theory, and complex systems.

In both teaching and research, my approach to the study of networks takes the perspective of "physical applied mathematics" [2]—focusing on modeling, with an origin and practice associated most traditionally with differential-equation models of physical phenomena—and I developed my undergraduate networks course with this philosophy in mind. I put a strong emphasis on mechanistic modeling, which contrasts both with the approaches to studying networks in courses on graph theory and with those in courses in statistics and computer-science departments. My blog associated with the University of Oxford version of my networks course [4] includes links to review articles and other online sources to supplement the lecture notes and main text.

In addition to my course, numerous other existing networks courses (with the number expanding rapidly), at multiple curricular levels, are taught in a variety of departments (e.g., statistics, computer science, physics, and so on) and emphasize different topics and approaches. For some examples, see [5–18]. See Chapter 7 for a comparison of the topics and organization in many existing courses on networks [19].

The rest of this chapter is organized as follows. In Sect. 2, I overview the topics that I cover in my networks course. In Sect. 3, I discuss how my course evolved from an informal set of lectures to a masters-level course and finally to a course for both undergraduates and masters students (including a version that is only for undergraduates). I highlight a few of the challenges in teaching my course in Sect. 4, and I conclude in Sect. 5.

2 Topics

The goal of (all versions of) my course, which I first taught in the University of Oxford's mathematics department, called the "Mathematical Institute" (MI), is to survey the study of networks from the perspective of mathematical modeling and to allow students to jump into the research literature. For example, my course's learning outcomes in the 2015 blurb in the MI's undergraduate handbook read as follows:

> Students will have developed a sound knowledge and appreciation of some of the tools, concepts, and computations used in the study of networks. The study of networks is predominantly a modern subject, so the students will also be expected to develop the ability to read and understand current (2015) research papers in the field.

In Table 1, I overview the topics that I cover in my networks course, which at University of Oxford included 16 hours of lectures and in later years—after being converted to a course that is intended primarily for undergraduates—also

Table 1 An overview of the topics in my undergraduate networks course at University of Oxford. There are 16 lectures of about 50 minutes (or so) each. I covered some of the listed topics in detail. I touched upon others briefly as generalizations of ideas, concepts, models, or methods that I discussed in detail.

Unit	Examples of topics
1. Introduction and basic concepts (1–2 lectures)	Nodes, edges, adjacencies, weighted networks, unweighted networks, degree and strength, degree distributions, other types of networks
2. Small worlds (2 lectures)	Clustering coefficients, paths and geodesic paths, Watts–Strogatz networks (focus is on modeling and heuristic calculations)
3. Toy models of network formation (2 lectures)	Preferential attachment, generalizations of preferential attachment, network optimization
4. Additional summary statistics and other useful concepts (2 lectures)	Modularity and assortativity, degree–degree correlations, centrality measures, communicability, reciprocity and structural balance
5. Random graphs (2 lectures)	Erdős–Rényi graphs, configuration model, random graphs with clustering, other models of random graphs or hypergraphs, application of generating-function methods (focus is on modeling and heuristic calculations; material in this section forms an important basis for units 6 and 7)
6. Community structure and other mesoscale structures (2 lectures)	Linkage clustering, optimization of modularity and other quality functions, overlapping communities, other methods and generalizations
7. Dynamics on networks (3–4 lectures)	General ideas, models of biological and social contagions, percolation, voter and opinion models, other topics
8. Additional topics (0–2 lectures)	Examples of possibilities: games on networks, exponential random graphs, network inference, temporal networks, multilayer networks, other topics of special interest to students (depending on how much time there is and the interests of current students)

incorporated “classes” (i.e., recitation sections) to discuss problem sheets. When I am teaching, I often have a tendency to include too much information.[1] An alternative design for an introductory networks course would be to cover fewer topics but to study them in greater depth.

For most topics, I based my presentation largely on discussions in Mark Newman’s textbook [1], although I drastically changed both presentation order and the relative emphases on topics. For more advanced topics, such as community structure and dynamical processes on networks, my course departed rather substantially from (and/or built substantially on) the discussions in [1]. For these capstone topics, I drew a lot of the material from survey, tutorial, and review articles [4, 20, 21]. I also extracted material from particularly instructive research articles (e.g., [22]), and I referred students to additional resources on my course blog [4]. The topics that I discussed in units (6) and (7) of the course (see Table 1) have varied over the years,

[1] I advocate a philosophy that students should “drink from the firehose of knowledge” (to quote a saying that I learned as an undergraduate at Caltech).

and I put some of them in homework problems only rather than in the lectures. For all units, I also discussed (at least briefly) some generalizations of ideas that are explored in [1]. For these generalizations, students (if they desire it) need to examine other sources to learn more details. In practice, covering topics (6) and (7) in a reasonable way, even at an introductory level, takes so much time that I have always chosen to spend more time on them than what is indicated in Table 1, rather than having lectures dedicated to topics from unit (8).[2]

3 Evolution of My Course

From the beginning, it was my intention to ultimately offer my networks course to undergraduates in the mathematical sciences at University of Oxford, but it started out as an informal set of lectures, which were attended by some masters students, doctoral students, and others.

3.1 Stage 1: An Informal Set of Lectures

The prehistory of my course dates to 2010. Invited by David Cai, in July 2010, I gave a set of ten lectures (of about 2.5 hours each) on “Network Dynamics” (although I covered both structure and dynamics) at an applied-mathematics summer school for masters students at Shanghai Jiao Tong University. I started adapting material from [1] and organized the material mostly as presented in Table 1 (though I added new topics to the possibilities in unit (8) as network science advanced). In practice, however, I spent way too much time on early units and ended up focusing mostly on units (1)–(4), with only a little bit of material from units (5)–(7). Using the organization that I developed for the summer school as a template, I signed a book deal to write an undergraduate textbook (which I still haven’t finished) for mathematicians and other quantitative scientists, where my choice of 8 units specifically matched the 16 one-hour (technically, 50 minutes or so) lectures in a standard MI course.

At University of Oxford, I first gave my networks course as an informal set of lectures in the spring term (“Trinity term”) in 2012. I taught one day a week, using a two-hour slot with a roughly 10-minute break at some natural point in the middle.[3]

[2] I typically mention temporal networks and multilayer networks very briefly in passing, in part because of their prominence and in part because I spend a lot of time thinking about them in my research. Additionally, unit (6) interfaces with topics like network inference, which I mention only in passing when teaching my course.

[3] At University of Oxford, it is more common to meet twice a week for “one hour” (which encompasses 50 minutes of lecturing), but my course met for one double-slot each week in most of the years that I taught it at Oxford, as I felt that this choice fit better with the 8-unit organization in Table 1.

I taught my course in the MI, and it was attended by students from the MI's Master of Science (MSc) program in Mathematical Modelling and Scientific Computation (MMSC), various doctoral students, occasional faculty members, and others. The MMSC students could use my course as a "special topic" if they wrote an extended essay on a subject that was agreed by them and me (as is standard for options courses in their program).[4] I did not assign any homework, though I pointed students to topics that they might be interested in pursuing in more detail.

3.2 Stage 2: A More Formal, Masters-Level Course

In Spring 2013, students from both the MMSC program and the MSc program in Mathematical Foundations of Computer Science (MFoCS) could take my networks course.[5] Over the years, many MFoCS students were becoming increasingly interested in applied topics, and it was desirable for my course on networks to be available for them to take as an option. To accommodate requirements for the MFoCS program, I needed to add two things: (1) homework problems for those students that went beyond what I assigned to undergraduate students (to ensure that the course was an MSc-level course) and (2) final "miniprojects" to determine student grades.

In 2013, I did not assign any homework assignment for most students, so I added a couple (three in the first year, but two in subsequent years) of homework assignments that required summarizing a research article and "refereeing" it. These assignments compel students to read papers in depth, learn how to evaluate papers and hopefully also some lessons about how to write papers, and learn good scientific citizenship (through volunteer work as referees).[6] I also met with the students to discuss each paper. The students typically did a very good job at the refereeing assignments, and paper authors to whom I showed these reports (with the students' permission) mentioned that my students' feedback was typically much more helpful than the actual referee reports that they received.

Following MFoCS rules, the students had 3 weeks at the end of a term to do their miniprojects, which are supposed to take 3–4 days of dedicated work. For a miniproject, which was required to be "double-marked" (with the grades from different people subsequently reconciled to determine a final grade) because of its open-ended nature, I asked the students to write a short paper on a specified advanced topic on networks. I changed the focal topic from year to year, and I show the miniproject that I assigned to the MFoCS students in my course in 2016 in Fig. 1. My goal was

[4] These special topics were marked both by at least one other person and by me (so-called "double marking"), and a reconciled mark from those scores became the student's grade in my course.

[5] The MFoCS program is a joint venture between the MI and the Department of Computer Science.

[6] I sometimes assigned papers that I knew well. Other times, I used the refereeing assignments as an excuse to carefully read a paper that interested me (and which, in practice, I otherwise might not read).

The University of Oxford

MSc (Mathematics and Foundations of Computer Science)

Networks (C5.4)

Hilary Term 2016

Below is listed a broad topic. Write a report on a specific subtopic within that general heading. Your report must include at least some numerical simulations (which you produce) and must include salient discussions of modeling issues, random-graph ensembles, and empirical data.

- Spatial Networks

Your report should be in the format and style of an article for the journal *Proceedings of the National Academy of Sciences*, and the main text must be no more than 6 typeset pages and must use their LaTeX style files (a template and style files will be provided). The report must include all sections (abstract, significance statement, etc.) in papers published in that journal (2016 format of papers). It is permissible to include a section of Supplemental Information that shows additional figures and calculations. In your report, indicate explicitly which ideas are new and which come from existing sources, and use appropriate and explicit attributions for all references (which must include papers reporting original research) or anything else (e.g., including code and figures) from other sources.

[*You need not submit scripts for any code you produce, but you may include them as part of Supplemental Information if you wish.*]

[*Your report need not contain original research results, though you must use some original research papers (not just review articles or books) as resources.*]

Fig. 1 The miniproject that I assigned to the MFoCS students in my networks course in "Hilary term" (winter term) in 2016. Their final grade was based on this miniproject, which was marked by at least two people (one of which was me), and then a final grade arose from a process of reconciling these grades.

for the students to have a miniature research experience (though the MFoCS program also includes a several-month dissertation as its capstone) and to cover an advanced topic in depth. Each year, I chose a focal topic that went beyond the course lecture material. Sometimes this entailed going into more detail on a capstone topic from units (6) or (7); other times, I selected a topic (e.g., "spatial networks" or "multilayer networks") from unit (8), even though I did not cover it in lectures beyond making a few cursory comments. As I discuss in Sect. 3.4, some computation (and potentially a lot of computation) is very important for the miniprojects, and ensuring that students are prepared to do them can be challenging.

3.3 Stage 3: Fourth-Year Undergraduates and Masters Students

In 2014, fourth-year ("Part C") undergraduates were able to take my networks course for the first time. Because of Oxford's end-of-year examinations, this necessitated moving my networks course from the spring term to the winter term ("Hilary term"). Unlike in the USA, most undergraduate courses in the MI are designated for students from one specific year, and these students also have to be from the mathematical sciences.[7] In the process of converting my networks course to an undergraduate course (which MSc students could also take), I also needed to formalize details such as recommended prerequisites, learning outcomes, assessment, and so on.

I indicated my learning outcomes in Sect. 2, and my course overview in the 2015 MI undergraduate course booklet read as follows:

> This course aims to provide an introduction to network science, which can be used to study complex systems of interacting agents. Networks are interesting both mathematically and computationally, and they are pervasive in physics, biology, sociology, information science, and myriad other fields. The study of networks is one of the "rising stars" of scientific endeavors, and networks have become among the most important subjects for applied mathematicians to study. Most of the topics to be considered are active modern research areas.

As I mentioned in Sect. 2, the goal of my course is to survey networks from the perspective of mathematical modeling and to teach students knowledge and skills to help them read the current research literature.

To make my course available for as wide a variety of students as possible, I did not suggest any prerequisites beyond what all undergraduates majoring in (i.e., "reading," to use UK parlance) Mathematics (and Mathematics & Statistics) are required to take anyway. For example, in the MI's official description of my 2015 networks course, I wrote the following text for recommended prerequisites:

> None [in particular, C6.2a (Statistical Mechanics) is not required], though some intuition from modules like C6.2a, the Part B graph theory course, and probability courses (at the level that everybody has to take anyway) can be useful. However, everything is self-contained, and none of these courses are required. Some computational experience is also helpful, and ideas from linear algebra will certainly be helpful.

The reason that I brought up the statistical-mechanics course, which I also developed, was that my networks course was labeled as C6.2b at the time, and the numbering could lead one to believe erroneously that material from the C6.2a course was required.

I was purposefully vague in my phrasing of "computational experience" in the recommended prerequisites, and the MI's computation requirement for first-year students was in the process of changing. As I discuss in Sect. 4, students' prior

---

[7] My UCLA version of the course (see Sect. 3.5), which I taught for the first time in spring 2017, included students from multiple majors and undergraduates in their fourth, third, and second years. A benefit of including second-year and third-year students is that some of them may desire to do an undergraduate research project on networks, and taking a networks course sufficiently early may also influence the subsequent courses that they elect to take.

experience with computation is one of the main challenges of teaching my course. The linear algebra that is required for students who are reading Computer Science (or Mathematics & Computer Science) is somewhat different than that for other undergraduates in the mathematical sciences, but in practice this issue never came up (or at least it never came to my attention) in my networks course. The students did occasionally ask questions about concepts from linear algebra and probability (e.g., generating functions show up a lot) that are important for my course, and such questions have been even more prominent in the UCLA version of my course (see Sect. 3.5).

With undergraduates now taking my course, I also needed to develop more formal homework assignments. To discuss these assignments, the lectures were supplemented with six hours of problem classes. (In practice, the total amount of time is somewhat shorter than six hours.) I arranged these as four 1.5-hour classes in 2014 and 2015 and as six one-hour classes in 2016. Problem classes are like the recitation sessions (sometimes called "discussion sessions") in US universities—although the UK problem classes are arguably structured more around homework assignments than is the case in the USA—and they normally are attended by undergraduates, MFoCS students, and students in the Mathematical and Theoretical Physics (MTP) program.[8] In problem classes, a "tutor" (who is in charge of one or more sets of classes), with some help from a teacher's assistant (TA), goes through homework problems that students find difficult, discusses reading assignments and any papers for which the students are supposed to write referee reports, walks through bits of code for computational exercises, and so on. I was a tutor for some sets of classes that were associated with my lectures, and postdocs or senior PhD students were tutors for other sets of classes.

Initially, as is standard in the MI, undergraduates received a grade for my course based entirely on one exam that they took at the end of the academic year. The fourth-year students in the mathematical sciences start having their exams in the middle of Trinity term (and hence starting around the end of May). Homework problems and other materials are meant to help undergraduate students learn and prepare for a final exam, but any "grades" on assignments are intended only for feedback; they do not affect the course grade. My homework assignments were a mix of problems that I hoped would help students prepare for the exam and longer (and occasionally open-ended) problems to encourage them to explore topics in detail in a way that is impossible in an exam question.

My course's exam lasted for 1.5–1.75 hours (it varied because of rule changes) and included three problems. The students received a grade based on their top-two marks among those problems. Because of this setup, which I inherited from MI rules, many students choose one problem to skip (sometimes based on course material that they had decided that they would not bother studying in detail) and focus their efforts on the other two problems. Because of this mechanism, people who write exams often try to make problems of equal difficulty, a very time-consuming

[8] Starting in Hilary 2016, students from Oxford's initial cohort of a new MSc program (which I helped design) in Mathematical and Theoretical Physics could also take my course.

and essentially impossible task, given that sometimes different topics have inherently different difficulty levels.

Assessment of the masters students followed the norms for their various programs. The MFoCS students were required to do the standard homework assignments in addition to their refereeing homework assignments, and they were assessed by miniproject at the end of Hilary term (see Sect. 3.2). The MMSC students could choose to do a special topic in my networks course (as one of the set of special topics that they are required to do for the program) if they wanted to receive a grade in it (see Sect. 3.1). The MTP students were required to do the same miniproject as the MFoCS students.

3.4 Stage 4: Changing from Exam Assessment to "Miniproject" Assessment

The final major change in my networks course at University of Oxford was converting the undergraduate assessment from exams to miniprojects. (See my discussion at the end of Sect. 3.3.) I taught the 2016 version of my course with miniproject-based assessment.

In my view, examination-based assessment is particularly inappropriate for a course about networks. Problems in this format are artificially short and depart substantially in both scope and time allotted from the types of problems that one actually studies in network science. Thus, although I used exam-based assessment voluntarily during the first year that undergraduates could take my course (see Sect. 3.3), I did so with the expectation of changing it shortly thereafter. After a long and (very) tedious battle, I was able to convince the MI's teaching committee to allow this change in 2015, which allowed me to implement it in 2016.[9]

Assessing the undergraduates in my course using miniprojects, which I was already doing for MFoCS students (see Fig. 1 for an example), gave them an opportunity to explore a topic in depth and provided an introduction to doing research in network science. The benefits of using miniprojects for teaching students about networks also hold at other levels, as demonstrated by the NetSci High program for teaching network science to high-school students [24].

Although the miniprojects that I used for the undergraduates in my course closely resembled the ones that I was already using for the MFoCS students, there were a couple of important differences. First, instead of picking one broad topic for the students, as I did with the MSc students, I gave undergraduate students a choice between two broad topics—"community structure and other mesoscale structures in networks," which goes predominantly with unit (6), and "spreading processes on networks," which goes predominantly with unit (7)—partly because I wanted them

[9] I believe that my course was the first lecture-based course in the MI to be approved for miniproject-based assessment. It was the first domino to fall, and at least one other course soon followed suit. I expect that there will be more.

to have some choice and partly because I wanted to give myself a bit more variety, given that I was going to be evaluating more than two-dozen reports. The other instructions in the miniproject (again see Fig. 1) were the same for both undergraduate and MSc students. Second, I purposely connected the project topics directly with capstone subjects in the course, whereas I was a bit more adventurous with the MSc students, who I felt should spend time on a topic that itself went beyond what was in the course.

Using miniproject-based assessment necessitated some tricky changes in timing. To the extent possible, the MFoCS miniprojects (which I also used for the MTP students) and undergraduate miniprojects needed to be synchronized—and both types of projects were to be undertaken during a 3-week window, with an expected commitment of 3–4 days of strenuous work—so the undergraduate miniprojects needed to occur at the end of Hilary term (as that time was fixed for the MSc students), rather than in the middle of Trinity term. For the undergraduates, we released the miniproject on Monday of the eighth and final week of Hilary term. The sixth and last problem class could thus occur no later than during the seventh week; this enforced more rigid timing at the end of my course than was the case when assessment was based on an exam to be taken a few months later. Grading so many projects (about three dozen, counting all students) was rather strenuous and time-consuming, and some of the grade reconciliation with the other markers was highly nontrivial. On the bright side, I didn't need to grade any exams or spend dozens of hours constructing an exam.

Importantly, the change from exam-based assessment to miniproject-based assessment gave me much more freedom to teach my course in the way that I wanted. I made my homework assignments "more realistic" with respect to what practitioners of network science do in their research. Even before the change, my homework assignments included several problems that allow exploration, a significant departure from the norm in the MI. After the change, I further reduced focus on problems of a style that align with exam preparation, and I increased emphasis on computation, as this is a very important aspect of network science. Practicing computational explorations also helps prepare students for undertaking a miniproject.[10] Another of my changes was to reduce the number of problems with similar calculations, such as generating-function analyses with progressively more intricate random-graph models, as I wanted students to see a small number of examples to get an idea about methods, rather than overemphasizing some topics at the expense of others. I also added some "refereeing" problems (though many of the undergrads seemed to struggle with these, or at least were perplexed by them), like the ones that I had already been assigning for several years to the MSc students (see Sect. 3.2). The MFoCS students needed to do both these refereeing problems and the ones that were designed specifically for them.

[10] Even before changing my course's mode of assessment, my homework assignments included some computational exercises, which many students tended to ignore, perhaps because they thought (mistakenly) that I couldn't test the material in these problems on timed exams without computers.

My revamped course worked much better than the examination-assessed version—which already worked much better the second time that it was offered it to undergrads, as there were many kinks to iron out from the first year—and I think that most of my students agreed with me. The MI ended up giving me a teaching award in recognition of designing and teaching my networks course.

The networks course still exists at University of Oxford, and it was taught by Heather Harrington in 2017 and by Renaud Lambiotte in 2018. Even though the course is only a few years old, it was already well-attended the first time that it was offered to undergraduate students, and in 2017 more undergraduates were enrolled in it than in any other fourth-year mathematics course at Oxford.

3.5 Stage 5: Transferring My Course to UCLA

In 2016, I moved to the Department of Mathematics at UCLA, and I taught a version of my undergraduate networks course in spring 2017. I taught it as a special-topics course, and it joined the course catalog as a regular offering starting in the 2017–2018 academic year.

For the initial UCLA version of my course, I mostly followed what I had been doing at Oxford, as I wanted to see what transpired in practice to have a better understanding of what, if any, substantive things needed changes. (I'd rather make such changes once rather than twice, and I felt that the existing form of the course was rather good.) Given that I moved back to the USA, I also went back to having formal office hours, which doesn't occur for courses at Oxford. Naturally, I am also available by appointment and answer queries by e-mail and on the course discussion board.

The extra freedom in the US system compared to Oxford allowed me to make a few formatting changes, such as in how I assess students and determine their grades. My class still has miniprojects, though I decided to make them group projects, allowing students to go further with them and making it more manageable to grade them. The miniprojects constitute 50% of the course grade, and there is both a group written report (in the format of the journal *Proceedings of the National Academy of Sciences*, as I was doing with the course at Oxford) and a group oral presentation. The homework assignments make up 25% of the grade, and quizzes and one midterm (where the midterm, which the students take during one 50-minute class period, counts as three quizzes) accounts for the final 25% of the grade.

In the spring-2017 offering of my course, there were 3–4 students in each miniproject group, and I determined the groups a few weeks into the course after soliciting ideas from the students about what topic they might want to study (though without a commitment to a specific topic) and with whom they might wish to work. In practice, with various other course commitments (such as homework assignments), the students had about three weeks to do their miniprojects. As with a PhD-student-level networks course that I taught during the winter-2017 term, I met with each group of students to help get them started in the early stages of their miniprojects (e.g., to make sure their project was doable, such as by ensuring that they were not using data that would take much longer than the three-week project timescale to clean before they could do analysis), and I was also available for consultation about their miniprojects throughout the time that they were working on them.

With 10 weeks rather than 8 weeks in a term and with three scheduled lectures each week (except for a couple of holidays), a UCLA course has a lot more lecture time than the ones that I taught at Oxford, and there is also a one-hour (technically 50 minutes) recitation section once a week. There is a single weekly discussion session for all students (in spring 2017, there were about 23 of them) in my course, in contrast to Oxford, where my course had multiple such sessions (with about 8–12 students each). I haven't added new mathematical material to the course, and I have instead used the extra time to add introductory discussions to cover the "big picture" in both complex systems and network science, go through some of the material more slowly, and interact more closely with the students in lectures. Having only 16 lecture hours at Oxford rushed things, and even student questions in lectures typically couldn't get the attention that they deserved. Moreover, when I was using exam-based assessment—and I was required to submit exam materials extremely early, entailing a strict commitment to cover the material that was being tested—I had almost no leeway to veer away from the course's intended trajectory, and not getting through the necessary material was simply not an option. At UCLA, in contrast, if one doesn't cover a topic, one can just not put it on an exam. Moreover, my networks course is an elective, and I don't need to worry about covering material that is prerequisite for other courses.

As was the case at University of Oxford, my homework assignments at UCLA include a mixture of straightforward problems that are meant to help the students learn definitions and concepts, trickier problems to stretch their knowledge about them, and computational exercises (including open-ended ones). Because I no longer have exclusive responsibility for working out detailed solutions of the homework problems (TAs help with this at UCLA), I assign several problems from [1], unlike what I did at Oxford. When I first taught my networks course at UCLA, I intended to again include paper-refereeing problems, though I didn't do so in practice; I hope to include them sometimes in future years. I also created a couple of homework "problems" of an unusual nature. For example, as part of the first homework assignment, I ask the students to take a picture of a local network (either on campus or in Westwood, which is an area next to campus), to identify the nodes and edges, and to indicate any other features that they notice (e.g., whether it is a spatial network, has multiple types of edges, and so on). My hope with this problem is to encourage my students to think about the fact that networks are everywhere.

All of my quizzes at UCLA have been "pop" quizzes (lasting about 15–20 minutes), and there were three of them in total in my spring-2017 course, though I did not fix the number before the term started. The main purpose of the quizzes—and especially of not announcing in advance when they are going to occur—is to encourage the students to attend lectures and to keep up with the homework and the reading. Similarly, the main purpose of the midterm (for which the students can use hard copies of their lecture notes, their homework assignments, and [1]) is to encourage students to spend time poring over the material to learn it better.

Unsurprisingly, the student composition of my networks course has been rather different at UCLA than it was at Oxford. At UCLA, most of the enrolled students are majoring in the mathematical sciences, though there are some exceptions, and there are now third-year and second-year students in addition to fourth-year students. Additionally, my course now includes only undergraduates. I now need to list recommended prerequisites—which are appropriate linear-algebra and probability courses, along with the desirability of some prior experience with programming—as it is no longer guaranteed that students who want to enroll in my course have previously seen certain essential topics. As was the case at University of Oxford, some relevant topics (such as generating functions) aren't covered in the prerequisite courses at UCLA, so I introduce them myself and encourage the students to look them up in detail on their own if they want more information. The level of prior programming experience (and degree of difficulty in getting started with the computational exercises) is mixed among the students, much like what I had observed at Oxford, and the command of linear algebra among my UCLA students is weaker overall than was the case for my Oxford students. I have allowed students to take my networks course even if they haven't taken courses in the prerequisite subjects. The point of specifying those subjects is to convey what knowledge I am going to assume from the first day of my course.

I expected some hiccups in my course from the institution change and the ensuing differences in its composition of students, but its UCLA debut in spring 2017 was unexpectedly smooth. My course was very popular among the students who took it—I received even more positive course evaluations than the ones from the 2016 course at Oxford—and it benefited a great deal from the extra lecture time (especially from not having to rush things and being able to interact a lot more with the students), having office hours, and other things. There were just over 20 students in my course, and learning about networks appears to have inspired them: several of them are collaborating with me on research projects, and two others contacted me to let me know that they were using skills and knowledge from my networks course in their job internships. One comment from the students that I have implemented for my course in 2018 is to start the projects earlier, which had been my intention last year. The delayed time before starting project work arose from wanting to show the students enough topics so that they would have a better idea of what they might want to work on and how to go about doing it, and presenting this amount of material took longer than I expected. I like the pace at which I can now present the course material, and I have tried to preserve that while also introducing project work earlier to give the students a bit more time on their projects.

4 Some Challenges

Teaching my networks course has been very challenging. Key challenges have included (1) finding the right balance, especially given the diverse backgrounds of my students, between mathematical rigor and models, methods, and using a physical-applied-mathematics approach; (2) computational exercises and expectations, in conjunction with diverse student backgrounds in computation and programming; and (3) exam-based assessment (until I was able to change this).

The diversity of the mathematical backgrounds of the students in my course has been a persistent challenge. For example, in the University of Oxford version, most students had one of two principal backgrounds: (1) people who had taken many applied-mathematics courses (e.g., in topics like fluid mechanics, differential equations, and mathematical biology) but who had taken few or no advanced courses in pure mathematics, where the focus is on mathematically rigorous arguments, and who were also not used to applying ideas from “physical applied mathematics” to discrete structures; and (2) people who had taken a lot of courses in discrete mathematics (in topics such as graph theory, probability, and various areas of statistics) who were more comfortable with mathematically rigorous arguments and/or statistical modeling than with doing physical-applied-mathematics modeling [25, 26] using arguments that usually are not mathematically rigorous.

My approach to studying networks—in both teaching and research—follows the tradition of physical applied mathematics [2], which emphasizes modeling and scientific rigor (and domain relevance) but typically does not focus on demanding mathematical rigor. In my networks course, I put strong emphasis on mechanistic modeling, but I usually sacrifice mathematical rigor (especially given the time constraints), and I almost never present things in a precise definition–example–theorem format. Moreover, many topics that I discuss would take a very long time to present in a mathematically rigorous way or are not yet even known at that level. I discuss much more complicated models than what one typically sees in a graph-theory course (or in a rigorous statistics course on networks),[11] and I discuss the application of (mechanistic) modeling principles that are taught much more commonly in applied-mathematics courses than in statistics (where descriptive modeling is emphasized) or pure-mathematics courses.

It is very challenging to cover formal definitions and theory, and then to discuss dynamics and modeling, and then to relate them to real data sets and numerical computations. This challenge was already present when only MSc students were taking my networks course, as MMSC students are mostly of type (1), but MFOCS students are predominantly of type (2); and it became even more prominent when I was also teaching undergraduate students. In my course, I occasionally bring up some physics jargon (much of which is used in [1]) to help make connections (e.g., to some topics in statistical mechanics), though I try to do so without overemphasizing it. Because some of the classical models in network science are not

[11] For example, in graph theory, one might spend a lot of time rigorously proving results on Erdős–Rényi (ER) random graphs, but I want to spend time on more intricate random-graph models (such as configuration models and their generalizations) that are more appropriate for studying real-world networks. I do introduce ER graphs in my course, and various homework problems are about them, but I discuss heuristic arguments for analyzing them, rather than presenting mathematically rigorous arguments (which carries the risk of drowning students in details), to demonstrate important ideas (such as a phase transition to a giant connected component).

well-defined mathematically in most of the standard presentations of them,[12] it is easy to fall into a trap in the middle of a lecture of not specifying everything that is necessary in a manner that is sufficiently precise. This was especially frustrating to Oxford students with a pure-mathematics background, as most of them are not used to this style of presentation. I try to be more precise in my presentation of network models than is often the case in research papers, but it is rather challenging to strike the right balance between precision of model specifications—as well as the level of mathematical rigor when analyzing the models, especially given that there are many features of these models for which mathematically rigorous analysis remains an open challenge—and an emphasis on modeling and discussing examples of many different types of models. Thankfully, when students examine these network models computationally on homework assignments and in their miniprojects, they get a chance to see (and, ideally, discover for themselves) exactly what information is needed to ensure a complete, precise specification. Moreover, investigating the consequences of making different choices in a given family of models is an important topic to study. (I like to include homework problems that encourage such exploration.)

A second major challenge, which occurred both at University of Oxford and at UCLA, is the diversity in students' past experiences and knowledge about doing computations. When I assign computational exercises as part of homework—note that it is very hard to make these problems equally accessible to students of widely differing computational backgrounds—I go out of my way to ensure that code (e.g., in MATLAB, for which University of Oxford has a site license) is available online, such as through the Brain Connectivity Toolbox [27] or other resources. Several of my homework problems and the course miniprojects require the use of real-world data sets, and many students stumble upon some of the famous (and infamous) data sets, such as ones that are available from Mark Newman's website [23]. Eventually, I started including some tutorial computational exercises at the beginning of my course. This helps a lot, but it has not completely removed the challenges for students with less computational experience or knowledge. Starting with release R2016a, MATLAB has included some functionality for network analysis [28], and this has been helpful for my course. Some of my students have decided that they prefer using Gephi [29] (e.g., for visualization), Python with NetworkX [30], or R with igraph [31]. I am happy with the students using whatever software they want.

My concern for my course is not whether the undergraduates can program in MATLAB or using any other language or software package, but rather that they can successfully use, understand, and interpret the output of computations. If that means running somebody else's .m file in MATLAB, so be it. Some programming experience does help, but strictly speaking it has never been something that my course has required (despite the feelings of some students to the contrary). One frustrating issue, especially for students, that sometimes arose at Oxford is that students may not know where to go if they cannot get code or a software package to work. In US

[12] For example, one needs to make choices for how one rewires or adds shortcuts in a Watts–Strogatz network [32, 33], one needs an initial ("seed") network when studying preferential-attachment models [1], and so on.

universities, such issues tend to be less problematic, as the office hours of professors and teaching assistants are great for addressing these kinds of individual queries. There were fewer opportunities at Oxford to help students sort through such technical problems (which often are not easy to address with e-mail communications), and my TAs and I encouraged students to talk to other students who had managed to get a particular package to work or to look things up online. As I mentioned previously, when my networks course was assessed by a final exam, many students ignored the computational exercises on the homework assignments, as they seemed to think that they weren't testable (despite my explicit comments to the contrary). However, that is mistaken, as an exam can include questions that describe or show output, and I can then ask the students about it.

Another issue that is worth bringing up is my course's reading assignments. In mathematics courses at Oxford, lecturers are not allowed to compel students to buy any textbooks for courses—a marked difference from the norm in US universities—so it is standard to provide students with a terse set of lecture notes. I wanted my students to read material beyond what was in my notes. In early versions of my course, in addition to a scanned version of my notes (I received complaints that they weren't typeset), I gave the students a copy of an in-progress textbook in its very rough state, and I made it clear that it was very far from polished. I also strongly encouraged my students to go through various parts of [1], as well as other resources (such as parts of some review articles), though they did not always find it clear which source they should use for a given topic. Mathematics undergrads at Oxford tend to focus on material in lecture notes, and my MI courses were unusual in expecting much more reading than what is in a short set of notes. Naturally, there is far more to the material in an advanced course than what is included in a terse set of notes. Such reading was optional, though I strongly encouraged it, through the 2014 version of my course, because I was unable to ensure that all students had easy access to [1] without forcing them or their Colleges (each Oxford student is a member of a College) to buy the book.

A few months into 2014, the first year that my course was open to undergraduates, I found out from one of my students that all Oxford students could freely access [1] online, so starting in 2015 I assigned explicit reading from [1] and elsewhere as the first "problem" on every homework sheet. I thereby informed students exactly what I required them to read, and I also suggested some optional additional reading that I felt would be helpful. This largely solved the problem of what the students should read—and it helped the 2015 version of my course to work out *a lot* better than the 2014 version, which was extremely rough—though some students complained that there was too much material to read. Others felt that [1] is fast and easy to read and that reading about 50 pages per week of it is very manageable. For the UCLA version of my course, I require my students to buy [1] (but only that book), and I continue to use sources like review articles and other online resources (as well as my lecture notes). From my experience teaching undergraduates at University of Oxford, I think that too many of them seem to prefer the boring Oxford model of lectures and exams, with "examinable" material specified in a terse set of lecture notes, and I purposely (and purposefully) taught my networks course in a more exotic way. I think that my adventuresome approach greatly benefits the students in my courses, even if some of them are not always happy about it.

5 Conclusions

When I was at University of Oxford, I developed an introductory course in network analysis that is now taken by numerous fourth-year undergraduates and masters students from several programs in the mathematical sciences. I have also translated this course into one for undergraduates (at "upper division" level) that I teach in the mathematics department at UCLA. In both variants, my networks course links ideas from applied mathematics, theoretical (i.e., "pure") mathematics, and computation through the modeling and investigation of discrete structures. An introductory mathematics course about networks is an important component of a "discrete structures and data science" pathway through an undergraduate degree in the mathematical sciences, and it is crucial for universities to include these types of pathways.

I hope that the present article will help encourage faculty—especially those in mathematics and mathematical-science departments—at other institutions to design and teach introductory courses in network analysis. My course is for advanced undergraduates, and it would also be good to develop courses in network analysis (e.g., freshman seminars) that are appropriate at an earlier stage of undergraduate education. Such courses complement existing courses in graph theory and other subjects, and they give a chance to introduce students to state-of-the-art topics that apply ideas from graph theory, probability, dynamical systems, and other important subjects in fascinating ways. I have suggested topics that are appropriate for an introductory networks course, and I have strongly advocated the use of miniprojects as a key method of assessment. As I have discussed, there are various challenges (e.g., diverse computational and coursework backgrounds among the students) in teaching an introductory course on networks, but it is a very valuable offering, and every mathematical-science department should include one. I hope that my description of my experiences will encourage the development of more undergraduate-level courses on networks in mathematics programs.

Acknowledgements I am grateful to the various "tutors" and teaching assistants who have helped me immensely when I have taught my networks course at University of Oxford and UCLA. I thank Heather Harrington, Florian Klimm, Flora Meng, Roxana Pamfil, Steve Strogatz, and Fabian Ying for helpful comments. I particularly thank Hiroki Sayama for reviewing an earlier version of this manuscript and providing very helpful feedback. (I also thank Hiroki for his support when dealing with the typesetting issues in this chapter.)

References

1. Newman, M. E. J. (2010) Networks: An Introduction, Oxford University Press.
2. Porter, M. A. and Howison, S. D. (2017) The role of network analysis in industrial and applied mathematics. ArXiv:1703.06843.
3. Society for Industrial and Applied Mathematics (2017) Careers in Applied Mathematics: Options for STEM Majors. https://www.siam.org/careers/thinking/pdf/brochure.pdf, 2017. Accessed, 1/19/18.

4. Porter, M. A. (2016) Math C5.4: Networks. http://networksoxford.blogspot.co.uk. Accessed 1/19/18.
5. Briatte, F. (2017) Awesome network analysis: Courses. https://github.com/briatte/awesome-network-analysis#courses. Accessed 1/19/18.
6. Clauset, A., (2017) Network Analysis and Modeling, CSCI 5352. http://tuvalu.santafe.edu/~aaronc/courses/5352/. Accessed 1/19/18.
7. Donner, R. and Kurths, J. (2014) Lecture "Complex Networks" — Winter term 2014/15. https://www.pik-potsdam.de/members/redonner/lecture-complex-networks-summer-term-2014. Accessed 1/19/18.
8. Dovrolis, C. (2017) CS 7280: Network Science: Methods and Applications. http://www.cc.gatech.edu/~dovrolis/Courses/NetSci/. Accessed 1/19/18.
9. D'Souza, R. (2016) ECS 253 / MAE 253: Network Theory and Applications. http://mae.engr.ucdavis.edu/dsouza/ecs253. Accessed 1/19/18.
10. Easley, D. and Kleinberg, J. , Economics 2040 / Sociology 2090 / Computer Science 2850 / Information Science 2040. https://courses.cit.cornell.edu/info2040_2014fa/. Accessed 1/19/18.
11. Giannini, M. Network Science PhD Program @ Northeastern. https://www.networkscienceinstitute.org/phd (click on 'Courses'). Accessed 1/19/18.
12. Jaillet, P. (2017) 6.268 Network Science and Models (Spring 2017). https://stellar.mit.edu/S/course/6/sp17/6.268/index.html. Accessed 1/19/18.
13. Kearns, M. (2017) Networked and Social Systems Engineering (NETS) 112: Networked Life. http://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/. Accessed 1/19/18.
14. Newman, M. E. J. (2015) Complex Systems 535/Physics 508: Network Theory. http://www-personal.umich.edu/~mejn/courses/2015/cscs535/index.html. Accessed 1/19/18.
15. Salganik, M. (2017) Sociology 543: Social Network Analysis (Spring 2017). http://www.princeton.edu/~mjs3/soc543_s2017/. Accessed 1/19/18.
16. Santa Fe Institute (2018) Complexity explorer: Courses. https://www.complexityexplorer.org/courses. Accessed 1/19/18.
17. Shalizi, C. R. (2016) 36–720, Statistical Network Models: Mini-semester I, Fall 2016. http://www.stat.cmu.edu/~cshalizi/networks/16-1/. Accessed 1/19/18.
18. Wierman, A. (2018) CS/EE/CMS 144: Networks: Structure & Economics. http://courses.cms.caltech.edu/cs144/. Accessed 1/19/18.
19. Hiroki Sayama. (2018). Mapping the curricular structure and contents of network science courses. In: Catherine B. Cramer, Mason A. Porter, Hiroki Sayama, Lori Sheetz, Stephen Miles Uzzo, eds., Network Science In Education: Transformational Approaches in Teaching and Learning. Cham, Switzerland: Springer International Publishing (arXiv:1707.09570). This article appears as Chapter 7 in this book.
20. Fortunato, S. (2010) Community detection in graphs, Physics Reports 486: 75.
21. Porter, M. A., Onnela, J.-P., and Mucha, P. J. (2009) Communities in Networks. Notices of the American Mathematical Society, 56: 1082.
22. Good, B. H., de Montjoye, Y. A., and Clauset, A. (2010) Performance of modularity maximization in practical contexts, Physical Review E, 81: 046106.
23. Newman, M. E. J. Network Data. http://www-personal.umich.edu/~mejn/netdata/. Accessed 1/19/18.
24. Cramer, C., Sheetz, L., Sayama, H., Trunfio, P., Stanley, H. E., and Uzzo, S. M. (2015) NetSci High: Bringing network science research to high schools (pp. 209–218), in Complex Networks VI: Proceedings of the 6th Workshop on Complex Networks (CompleNet 2015), G. Mangioni, F. Simini, S. M. Uzzo, and D. Wang, eds., Studies in Computational Intelligence 597, Cham, Switzerland: Springer International Publishing (arXiv:1412.3125).
25. Fowler, A. C. (1997) *Mathematical Models in the Applied Sciences*, Cambridge, UK: Cambridge University Press.
26. Lin, C. C. and Segal. L. A. (1988) *Mathematics Applied to Deterministic Problems in the Natural Sciences, Classics in Applied Mathematics*. Philadelphia, USA: Society for Industrial and Applied Mathematics.

27. Rubinov, M. and Sporns, O. (2010) Complex network measures of brain connectivity: Uses and interpretations, NeuroImage, 52: 1059. Code is available at https://sites.google.com/site/bctnet/. Accessed 1/19/18.
28. MATLAB: Graph and Network Algorithms. https://uk.mathworks.com/help/matlab/graph-and-network-algorithms.html. Accessed 1/19/18.
29. Gephi — The Open Graph Viz Platform. Available at https://gephi.org. Accessed 1/19/18.
30. NetworkX — NetworkX. Available at https://networkx.github.io. Accessed 1/19/18.
31. Igraph – Network Analysis Software. Available at http://igraph.org. Accessed 1/19/18.
32. Porter, M. A. (2012) Small-world networks, Scholarpedia, 7. 1739.
33. Watts, D. J. and Strogatz, S. H. (1998) Collective dynamics of ‘small-world’ networks, Nature, 393: 440.