Analytic Techniques for Computer Science

Paul E. Dunne

Department of Computer Science,
University of Liverpool

Part 1 – Introduction

What are we concerned with in this module?
Why is it part of a Computer Science programme?

"Saepe humanos affectus aut provocant aut mitigant amplius exempla quam verba."
Peter Abelard (1079-1142)
Historia calamitatum

["*Often the minds of people are inspired, as much as their emotions are calmed, more by examples than by words*"]

Administration & Other details

Module Coordinator

Prof. Paul E. Dunne
Rm. 2.04 (Ashton Building)
ped@csc.liv.ac.uk & P.E.Dunne@liverpool.ac.uk &
sq12@liverpool.ac.uk

Delivery

3 lectures per week & 1 scheduled tutorial class

Assessment

2 class tests (10% weighting)
Exam (80% weighting)

Module Aims

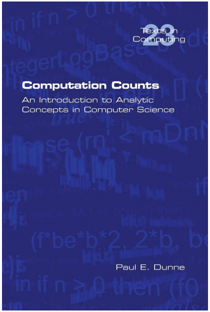
1. To equip students with an awareness of the range of methodologies that have been brought to bear in the treatment of computational issues.
2. To provide practical experience in how various formal approaches can be used to address such issues.

& Learning outcomes

By the end of this module you should have:

1. a basic appreciation of the range of techniques used to analyse and reason about computational settings.
2. the ability to solve problems involving the outcome of matrix-vector products as might arise in standard transformations.
3. the ability to apply basic rules to differentiate and integrate commonly arising functions.
4. a basic confidence in manipulating complex numbers and translating between different representations.
5. a basic appreciation of the role of Linear algebra (including eigenvalues and eigenvectors) in computational problems such as web page ranking.

Course Textbook – Computation Counts



Paul E. Dunne

College Publications, Texts in Computing (Vol. 23), 540 pages

£22.50 (from Amazon UK, Blackwells, etc.)

HCL, QA76.D92 (19 copies)

ALL Royalties to *The Walton Centre Neuroscience Fund*

In this lecture

We introduce the main topics studied.

Argue that these are central to

Science in general **AND**

Computer Science in particular.

Briefly summarize important application areas.

Details & Background

Rationale: Why?

There are areas that – like *Logic* and *Sets* and *Graphs* and *Probability* – occur again and again and again in studying and finding solutions to *computational* problems.

The topics mentioned have already been introduced within **COMP109**:

Foundations of **COMPUTER** Science

What is meant by **Analytic Techniques**?

The range of methods used to handle problems in computation whose analysis (that is as may be needed to write programs) is rather more involved than "simple" "obvious" solution cases.

For example?

Linear Algebra, Differential and Integral Calculus, Complex Analysis, etc etc

Analytic methods in Other Fields of Study

Subject Area	Example Applications	Needs
Chemistry	Molecular structure	Linear & Matrix Algebra
Physics	Planetary motion & Orbits	Calculus & ODEs
Biology	Population models	Statistics

Physics, however, has been claimed to face some worrying issues, cf

"*Ach, die Physik! Die ist ja für die Physiker viel zu schwer!*"
[attributed to the German mathematician David Hilbert (1862-1943) and, often, loosely translated as "*Physics is too hard for Physicists*", however not as negative a view as Vasily Grossman's, "*There is a terrible similarity between the principles of Fascism and those of contemporary physics*" (Life and Fate, Pt. 1, Ch. 19)]

What do these tell us?

that

Chemists **USE** Linear algebra **to do CHEMISTRY**.

Physicists **USE** Calculus **to do PHYSICS**.

Biologists **USE** Statistics **to do BIOLOGY**.

In **every single one** of these areas, the technique used is the **most suitable** for the specific application.

Important consequences

The application is **NEVER** just an "excuse" for "doing mathematics" (whatever that means).

What matters is **solving** the problem being considered.

The aim is to **apply** the tools available, not to refine how those tools are built.

For example, (qualified) electricians may **use** an ammeter to **check** electrical current; they are unconcerned with its underlying **theory**.

A very informal analogy

If Mathematics is concerned with designing better mousetraps

then Science is concerned with catching mice.

and what about Computer Science?

We find, as with the examples above, a similar **use** of a wide range of techniques in CS applications. Among such are:

Application Area	Needs
Graphics	Vector methods
Robotics	Vector methods
Web traffic analysis (eg Search engines)	Linear and Matrix algebra
Performance analysis	Calculus, Complex Numbers
Image Compression	Matrices, Complex Numbers
Debate analysis (argumentation)	Matrix algebra

Fortunately, . . .

Computer Science is still easy enough for Computer Scientists

Summary

Overview of Module Delivery

Over the remainder of the module we will consider the following topics:

1. The "traditional" concept of "*number*": different forms and motives; how simple structures are imposed and how those structures are applied to "real" problems.
2. Calculus: how differential calculus is used in understanding function behaviour; how integral calculus helps in measuring quantities.
3. An "unorthodox" view of "*number*": complex numbers and their source and properties.
4. Very brief overview of statistics and its importance in drawing conclusions from experimental data.
5. Matrices: matrix properties; graph representations; applying matrix algebra to deal with "ranking" problems.

What happens next

Throughout the module we are concerned with

HOW the various techniques are **applied** to real computational problems.

We have no interest in:

a. Proofs of correctness (these are irrelevant to our aims).
b. OR subtleties of analysis.
c. OR abstraction beyond that which is relevant to demonstrating **general** applicability.

   In the next lecture we look at *numbers and their meaning*.