

Miscada-Core IIb - *Scientific Computing*

tom.theuns@durham.ac.uk

September 2023

Scientific Computing

Lecturer: Prof Tom Theuns

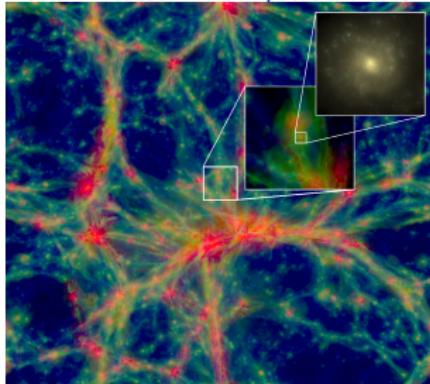
tom.theuns@durham.ac.uk

office: OCW 207

About your lecturer:



I work at the Institute for Computational Cosmology



Computer simulation of the formation of galaxies. Background blue-green-red image shows intergalactic gas, coloured according to its temperature, inset zooms into a Milky Way-like galaxy. Shown volume is 3×10^{24} m on a side. Figure from Schaye et al., '15. Want to know more, see the [Eagle web site](#).

Learning outcomes 'Scientific Computing'

- ▶ Learn how to use computer to solve complex (=realistic) problems
- ▶ Analyse physics to choose appropriate numerical algorithm
- ▶ Basic numerical methods and their implementation
 - ▶ differential equations
 - ▶ root finding: solutions for $f(x) = 0$
 - ▶ numerical integration
 - ▶ Monte Carlo methods & simulation
- ▶ Assessment via Python assignments in Jupyter notebooks,

<https://misdaca-sc-2324.notebooks.danielmaitre.phyip3.dur.ac.uk/>

The teaching assistants are



Dr Sophia Lilleengen



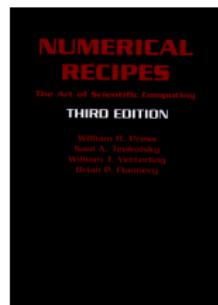
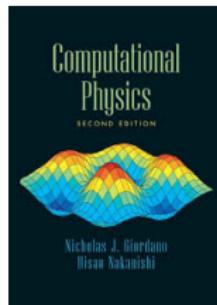
Dr Rifaa Alkhozondar



Dr Mehdi Tavakol

Material:

- ▶ Course text book: Giordano & Nakanishi, “Computational physics” (~ 10 copies in library)
- ▶ Additional reading: Press et al., “Numerical Recipes”



- ▶ Blackboard Ultra online course notes/hand-outs:

https://blackboard.durham.ac.uk/ultra/courses/_53766_1/outline

This course: Scientific computing

- ▶ Lecture on a given topic (e.g. radioactive decay), 2 topics per week
- ▶ Lab session on each topic, 2 lab sessions per week
- ▶ Course duration: 4 weeks
- ▶ Course work marked when Jupyter notebook session is submitted
- ▶ Feedback in following lecture
- ▶ Other topics in this module: statistics, machine learning, HPC

Scientific computing:

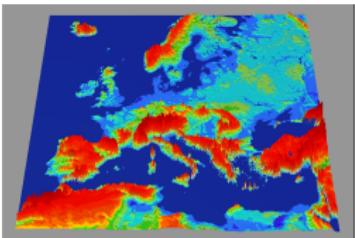
- ▶ Formulate a mathematical model for given problem
- ▶ Usually, analytical solution only possible once simplified
- ▶ **Scientific computing** to go beyond simplifications
 1. Analyse problem ('mathematical modelling')
 2. Choose and implement algorithm ('coding')
 3. Code verification ('debugging')
 4. Model validation/refinement ('experiment')
 5. Speed ('profile')
 6. Code documenting / upgrading (e.g. version control)

(This course)

Example: pendulum

- ▶ Mathematical model: $\frac{d^2\theta}{dt^2} = -\frac{g}{l} \sin \theta$. l : pendulum's length, g : gravitational acceleration, θ angle from vertical, t : time
- ▶ Analytical model (small angle approximation, $\sin(\theta) \approx \theta$) $\frac{d^2\theta}{dt^2} = -\frac{g}{l} \theta$.
Solution: simple harmonic motion
- ▶ Numerical model: solve for θ not small, include air drag on pendulum bob, etc. No known analytical solution
- ▶ Code verification: test small-angle case
- ▶ Model verification: compare to real pendulum

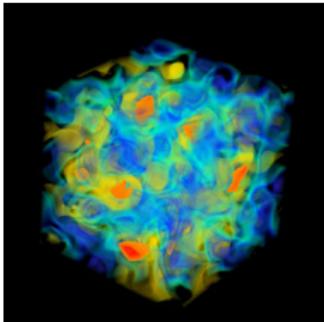
Real-world examples



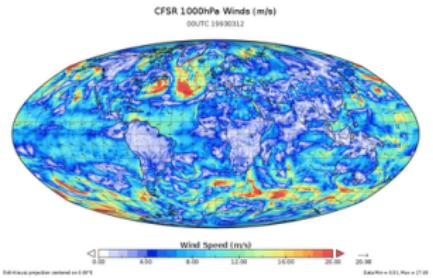
Weather forecasting



Improving efficiency of production chains



Fundamental physics



Climate modelling

Some vocabulary

- ▶ Solutions of mathematical models are **functions**

e.g. for pendulum, the function $\theta(t)$

- ▶ Functions are **maps**:

set of input values → set of output values
 domain → **range**.

for pendulum: t is input, θ is output

- ▶ An **algorithm** is a recipe for how to compute outputs given inputs
- ▶ An **implementation** codes the algorithm
- ▶ **Computations** evaluate functions
- ▶ In the digital world (computers), **the domain and range are discrete**, and **the implementation terminates after a finite number of steps**

Types of errors

- ▶ Truncation error:

Many functions are computed as a series, e.g.,

$\sin x = x - \frac{x^3}{3!} + \dots$ or similar. Evaluation is limited to a finite number of terms.

- ▶ Finite precision error:

Computer uses a finite number of bits to represent numbers.

This leads to round-off errors and breaks commutativity of mathematical operations

Example: $10^{30} + 1 - 10^{30} = 0$, and $a + b + c \neq a + c + b$

- ▶ Discretisation error:

Computer approximates a smooth function with discrete steps. Accuracy improves with decreasing step-size.

Errors can accumulate, leading to **instabilities**. Poor implementation of algorithm yields incorrect answer.

Course contents

- ▶ 1-7: formatively assessed submit final notebook 6 days after lecture
- ▶ 8: summatively assessed: **deadline November TBD**

1. Radioactive decay (TLC025)
2. Ballistic motion (CC/0007)
3. Harmonic motion (TLC025)
4. Chaos (CC/0007)
5. Root finding and integration (TLC025)
6. Random walks (CC/0007)
7. Ising model and phase transitions (TLC025)
8. Cluster growth and percolation (CC/0007) **Summatively assesed.**