ACS: Applying Computational Science

Gareth Collins

Group projects (ACS Applying Computational Science)

"Applying Computational Science and Engineering will give you a real-world experience of high-productivity problem solving."

Objectives

- To simulate applied computational science in the real world.
- To synthesise knowledge from the taught modules
- To apply techniques learned in the course to real problems
- To develop collaborative programming skills
- To reinforce best practise for software development

Project schedule:
Project 1: 21-25 November 2022
Project 2: 30 Jan. – 3 Feb. 2023
Project 3: 22-26 May 2023

Assessment:

Software	(70%)
Presentation	(20%)
Teamwork	(10%)

Group projects (ACS Applying Computational Science)

Module Learning Outcomes:

- Plan and produce software collaboratively.
- Collaboratively solve problems using software.
- Summarise work using collaborative presentations.

Group projects (ACS Applying Computational Science)

"Applying Computational Science and Engineering will give you a real-world experience of high-productivity problem solving."

- Three group projects that synthesise different elements of the MSc program.
- Each project will span five days of full-time effort, simulating a working week.
- On Monday morning, groups will meet with the project "client" to outline the problem to be solved and answer questions.
- On Friday afternoon, the groups will submit their work to the client for assessment.

ACS Project Timetable

	Monday	Tuesday	Wednesday	Thursday	Friday
09:00–12:00	Lecture / Group assignment / Q & A	Group working	Group working	Group working	Group working 12 pm Code Deadline
14:00–17:00	Group working	Group working	Reserved For Sports	Group working	Group working 4 pm Presentation Deadline 5 pm Debrief 6 pm Social

Software submission deadline: **Friday 12 pm;** Video presentation deadline: **Friday 4:00 pm**

Our expectation is that you will each spend approximately **45 hours** on these projects, over the course of the week (9 hours / day): The project should be your focus for the whole week.

ACS Project Support

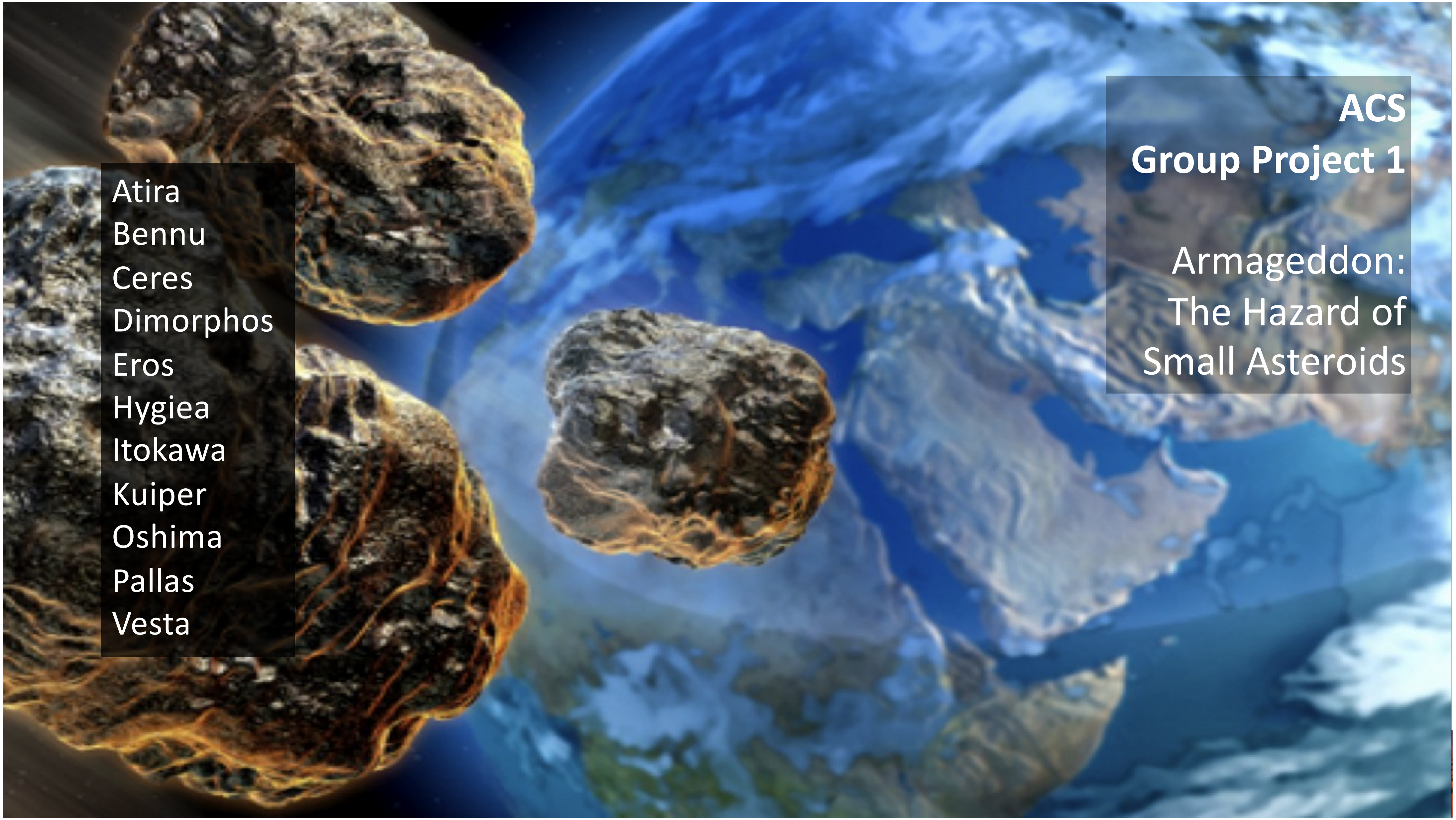
- This project is about collaboration: your first port of call for help should be each other!
- We have created a **private channel on Teams** for each group to ask us questions
- We will schedule 30 min. meetings for each group for voice/video calls or chat Tue/Wed/Thu to answer questions
- At other times we will try to answer questions via chat as soon as possible
- Please ask questions of **clarification** in the **General** channel
- We aim to provide indicative scores of functionality and performance during week

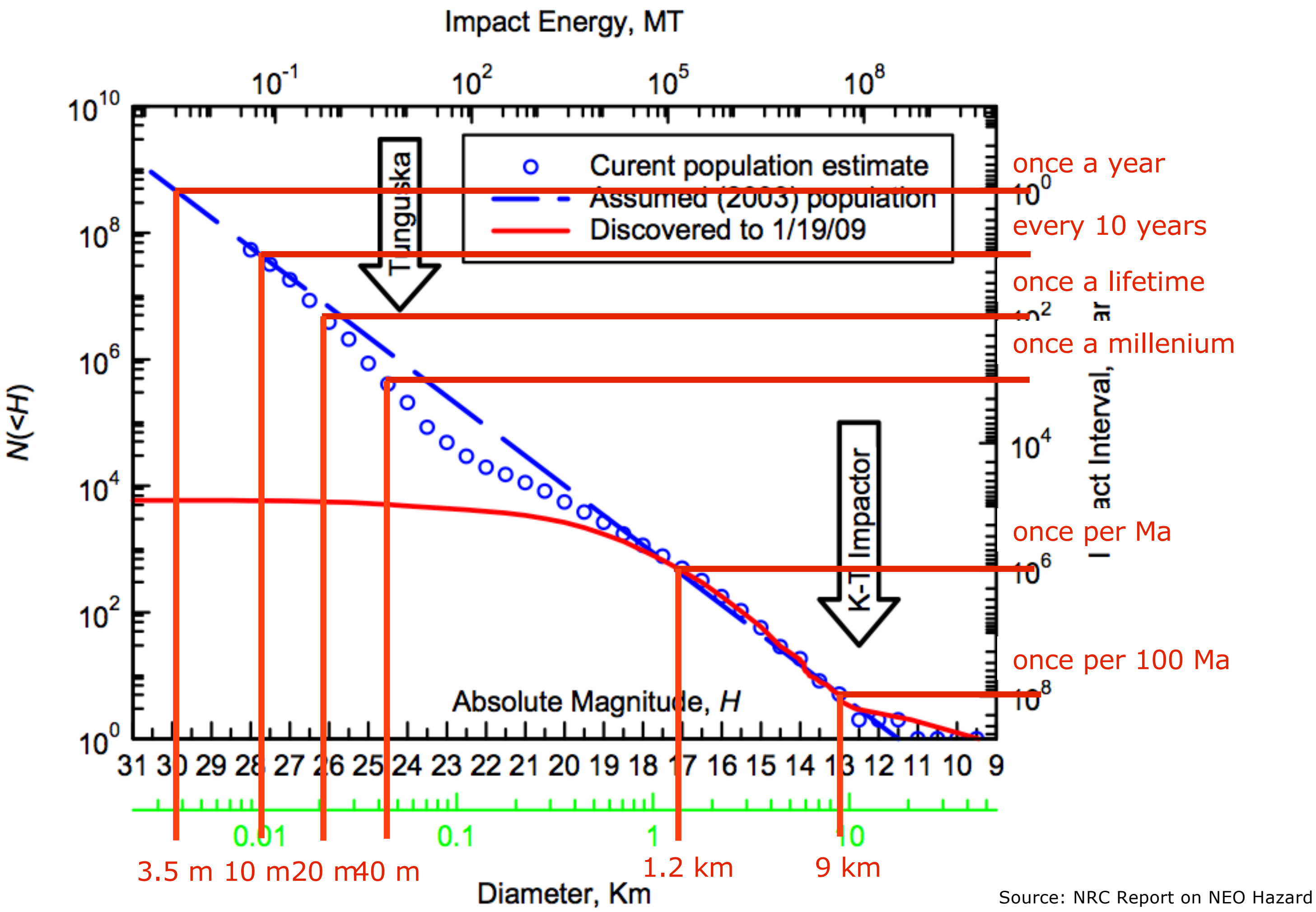
A welcoming environment

- We encourage and promote diversity in science
- Whoever you are, and whatever your background, we welcome you.
- We hope that everyone finds their experience welcoming, encouraging and rewarding.
- We want to foster a community based on mutual respect, tolerance, and encouragement and we kindly ask that you respect these principles.
- If you experience or witness any unwelcome behaviour we encourage you to challenge the behaviour or report it, in confidence, to the module coordinator or the course director.
- If you have questions about support or pastoral concerns, talk to PGT Senior Tutor: James Percival (j.percival@imperial.ac.uk)

A word about academic integrity

- **Academic integrity** is fundamental to learning, teaching and research
- **Academic misconduct** is the attempt to gain an academic advantage, whether intentionally or unintentionally, in any piece of assessment submitted to the College
- Each project will provide specific guidance about which aspects of the software must be written from scratch and which aspects can make use of packages or code written by others.
- **Plagiarism must be avoided**. If you use code sourced outside your group, you must include clear and proper attribution of credit.
- In group projects, **collusion constitutes the sharing of work between groups**. You are therefore strongly discouraged from discussing the group project with peers outside your project group.
- For group projects, **you are actively encouraged to work collaboratively as a team**—sharing ideas and code within your team is exactly what you should be doing and is **not collusion**.
- **As a guiding principle**: always acknowledge the contributions of others in your work, and do not leave yourself open to allegations that you have supplied answers to enable another student or group to commit academic misconduct.


ACS
Group Project 1
Armageddon:
The Hazard of
Small Asteroids
Atira
Bennu
Ceres
Dimorphos
Eros
Hygiea
Itokawa
Kuiper
Oshima
Pallas
Vesta

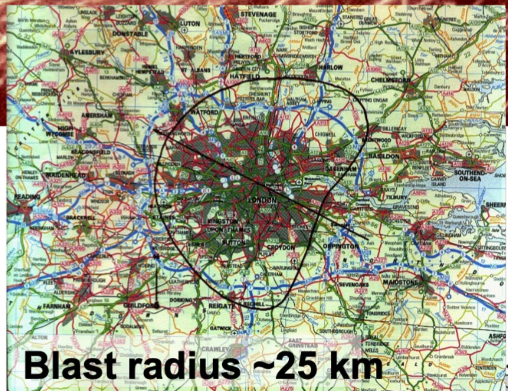


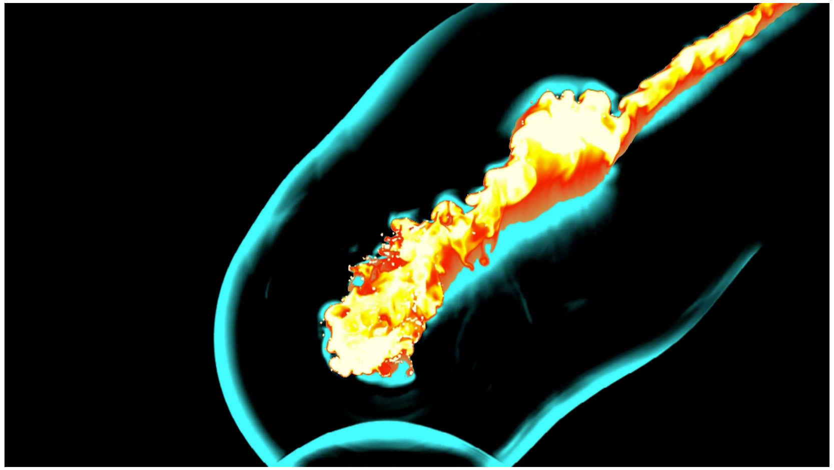
Source: NRC Report on NEO Hazard

Imperial College London

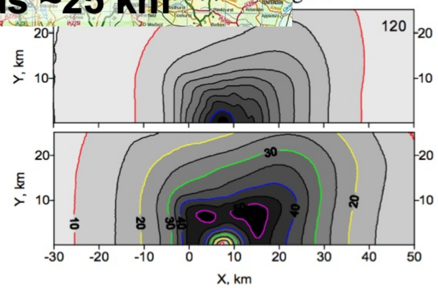
DEPARTMENT OF EARTH SCIENCE AND ENGINEERING

Famous airbursts: 1908 - Tunguska, Siberia.

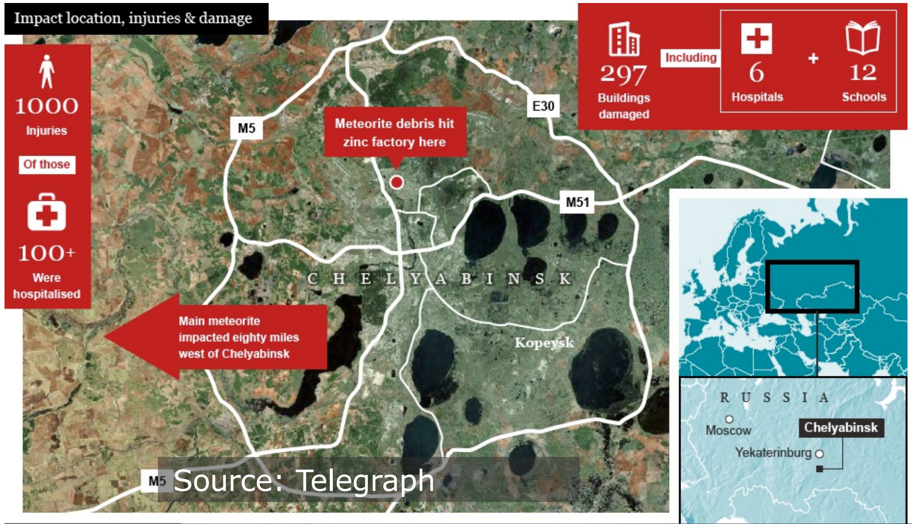




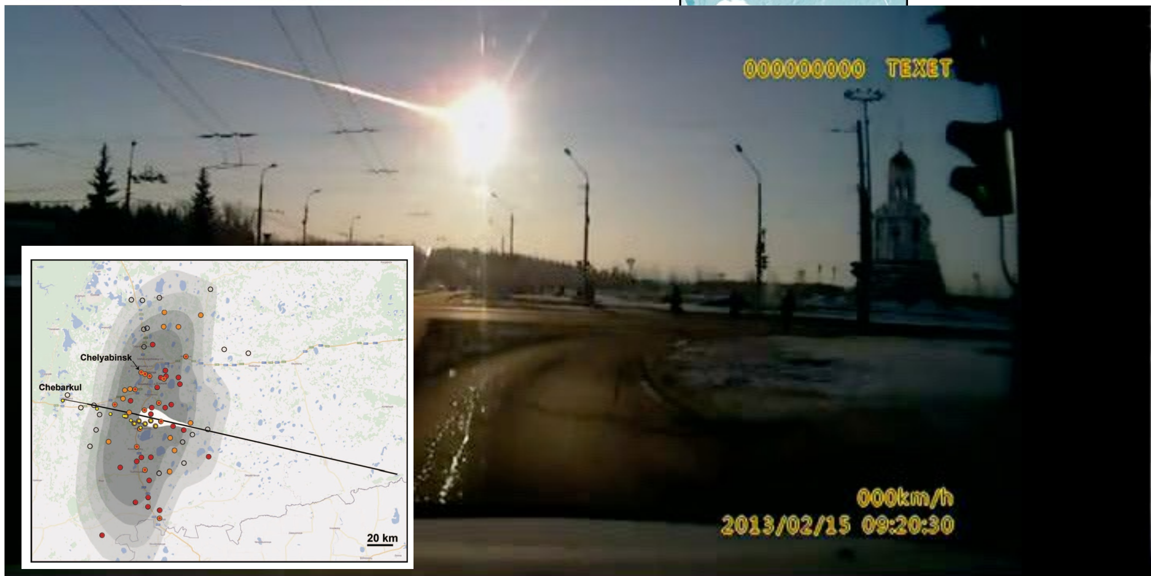
Plan view of blast contours (Artemieva & Shuvalov 2007)



Famous airbursts: 15th Feb. 2013 - Chelyabinsk, Russia



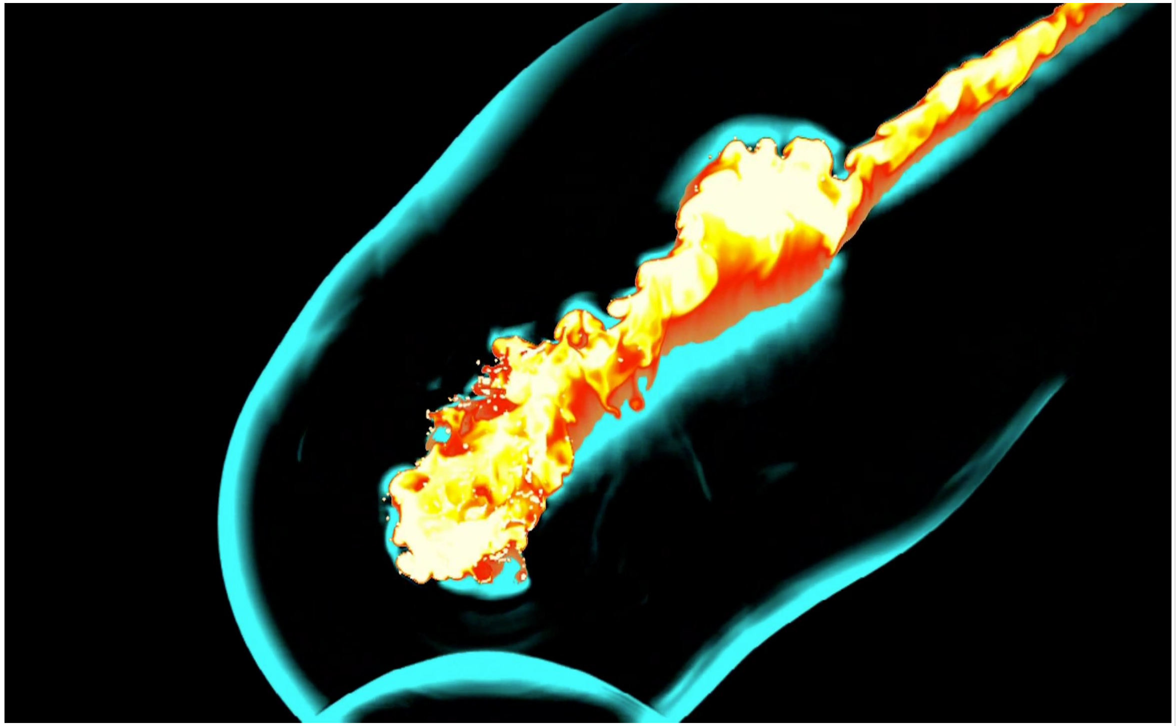
- Estimated mass: ~10,000 Tons (~20 m diameter; 19 km/s; 18 degrees).
- ~500 kTon energy
- Disrupted at 26-30 km altitude
- Blastwave shattered windows over a large area, injuring >1000 people



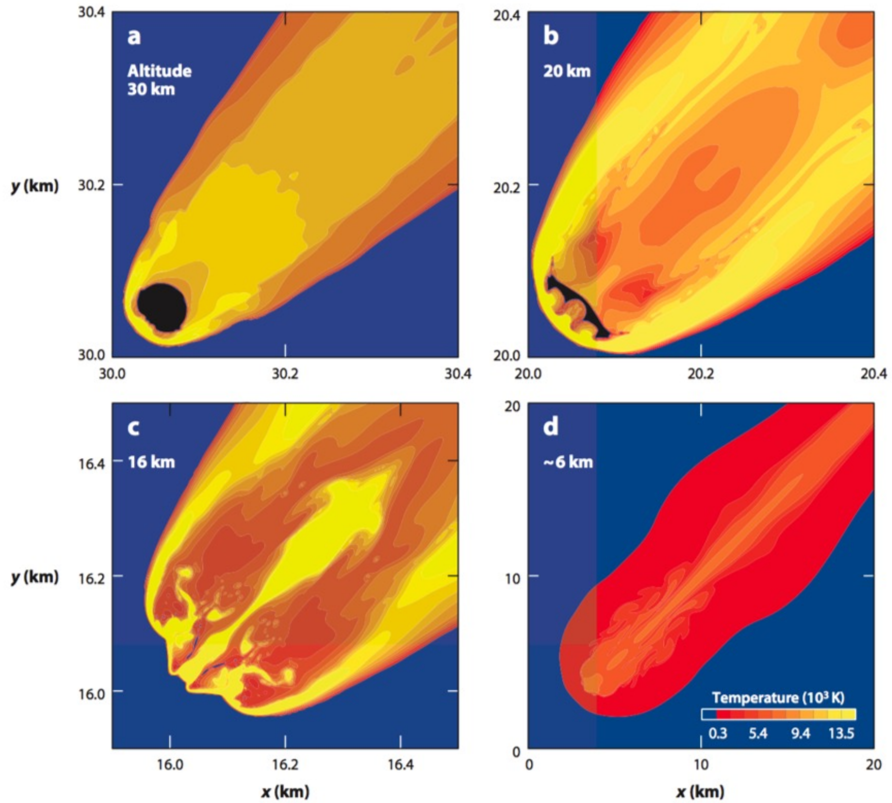

The “Carancas“ impact, Lake Titicaca, Peru, Sep. 15, 2007

Dynamic modelling of airbursts is a huge computational challenge

Artemieva & Shuvalov, 2016, *Ann. Rev. Earth Planet. Sci.*



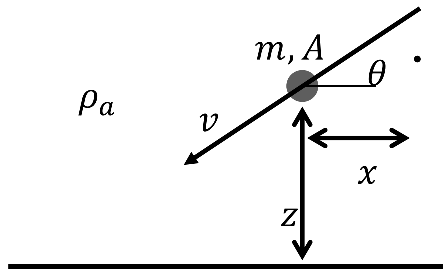
Boslough, 2013, pers. comm.



…there is a need for fast simulators that solve approximate equations

Meteor dynamics: no ablation; no gravity; no lift; flat planet

$$\frac{dv}{dt} = \frac{-C_D \rho_a A v^2}{2m}$$

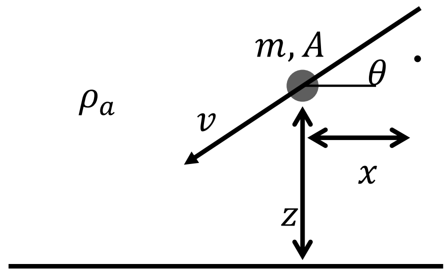


$$\frac{dz}{dt} = -v\sin\theta \qquad \frac{dx}{dt} = v\cos\theta$$

Meteor dynamics: with ablation; no gravity; no lift; flat planet

$$\frac{dv}{dt} = \frac{-C_D \rho_a A v^2}{2m}$$

$$\frac{dm}{dt} = \frac{-C_H \rho_a A v^3}{2Q}$$



$$\frac{dz}{dt} = -v \sin\theta \qquad \frac{dx}{dt} = v \cos\theta$$

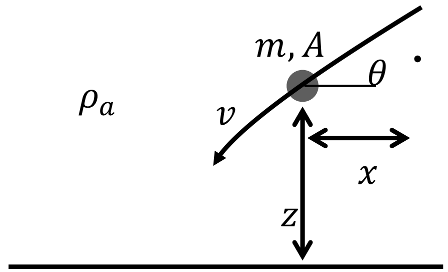
Meteor dynamics: with ablation and gravity; no lift; flat planet

$$\frac{dv}{dt} = \frac{-C_D \rho_a A v^2}{2m} + g \sin\theta$$

$$\frac{dm}{dt} = \frac{-C_H \rho_a A v^3}{2Q}$$

$$\frac{d\theta}{dt} = \frac{g\cos\theta}{v}$$

$$\frac{dz}{dt} = -v\sin\theta \qquad \frac{dx}{dt} = v\cos\theta$$



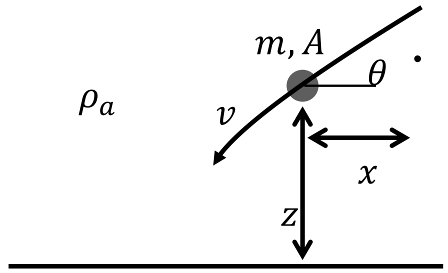
Meteor dynamics: with ablation, gravity and lift; flat planet

$$\frac{dv}{dt} = \frac{-C_D \rho_a A v^2}{2m} + g \sin\theta$$

$$\frac{dm}{dt} = \frac{-C_H \rho_a A v^3}{2Q}$$

$$\frac{d\theta}{dt} = \frac{g\cos\theta}{v} - \frac{C_L \rho_a A v}{2m}$$

$$\frac{dz}{dt} = -v\sin\theta \qquad \frac{dx}{dt} = v\cos\theta$$



Meteor dynamics: with ablation, gravity, lift and planet curvature

$$\frac{dv}{dt} = \frac{-C_D \rho_a A v^2}{2m} + g \sin\theta$$

$$\frac{dm}{dt} = \frac{-C_H \rho_a A v^3}{2Q}$$

$$\frac{d\theta}{dt} = \frac{g \cos\theta}{v} - \frac{C_L \rho_a A v}{2m} - \frac{v \cos\theta}{R_P + z}$$

$$\frac{dz}{dt} = -v \sin\theta \qquad \frac{dx}{dt} = \frac{v \cos\theta}{1 + z/R_P}$$

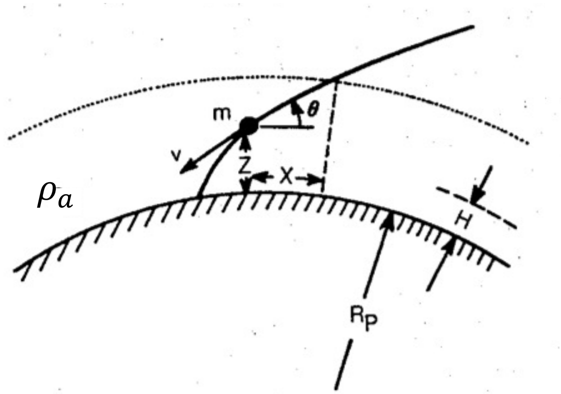


Fig. 11.1 Variables used to describe the path of a meteoroid of mass *m* as it descends through the atmosphere of a planet with radius R_P. The atmospheric density at any given altitude is ρ_g and its scale height is *H*. The meteoroid's altitude is *Z*, its instantaneous speed is *v*, and its trajectory forms an instantaneous angle θ with the local horizontal. The meteoroid's distance downrange from an arbitrary initial position is *X*. In this figure the initial position is determined by the intersection between the meteoroid's trajectory (solid line) and an arbitrary shell located well outside the planet's atmosphere (dotted line).

Meteoroids break-up under drag forces:

density of atmosphere

impactor speed

$$Y \approx \underline{\rho_a(z) v_i^2}$$

Meteoroid strength

Ram pressure on meteoroid

Break-up altitudes on Earth are typically 30-70 km, implying stony meteoroid strengths of 0.1-10 MPa

After break-up fragments spread, which increases drag, leading to rapid deceleration and extreme heating of atmosphere

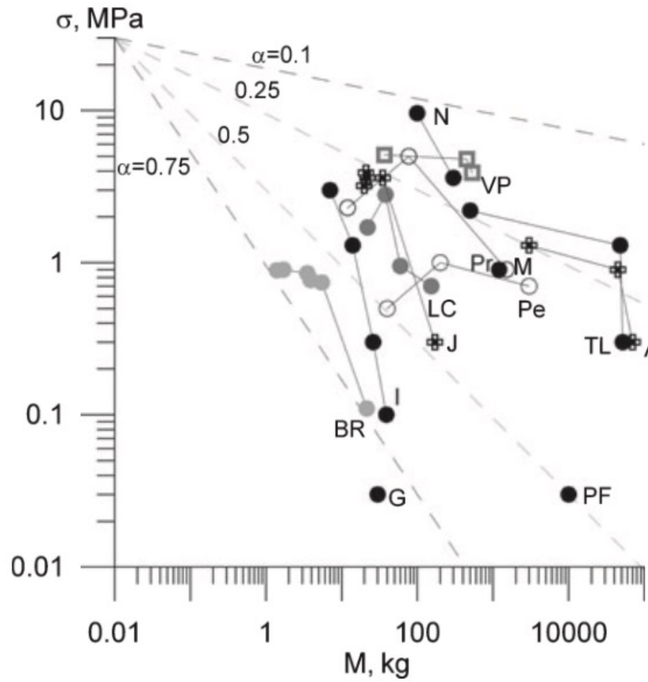


Fig. 4. Estimated apparent bulk strength (=ram pressure in Table 4) at first, second, and third breakups as a function of mass for our 13 cases (Pr—Příbram; LC—Lost City; I—Innisfree; Pe—Peekskill; TL—Tagish Lake; M—Morávka; N—Neuschwanstein; PF—Park Forest; VP—Villalbeto de la Pena; BR—Bunburra Rockhole; AS—Almahata Sitta; J—Jesenice; G—Grimsby). In some cases, other breakups predated the first listed breakup. In the case of Peekskill, for example, there is evidence of breakup before our first observed event. The dashed lines represent power law function, which relates laboratory strength data to bulk rock strength data.

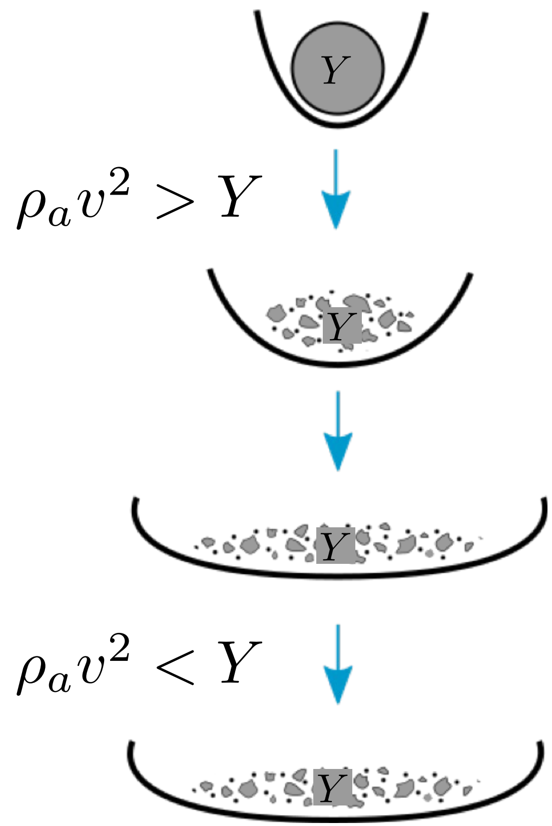
Debris Cloud Model of Meteoroid Disruption

- Treats disrupted meteoroid as an impermeable debris cloud (with circular cross-section).
- Cloud expands ("pancaking") while the ram pressure at it's front exceeds the meteoroid strength
- Spreading equation:

$$\frac{dr}{dt} = \left[\frac{7}{2}\alpha\frac{\rho_a}{\rho_m}\right]^{1/2} v$$

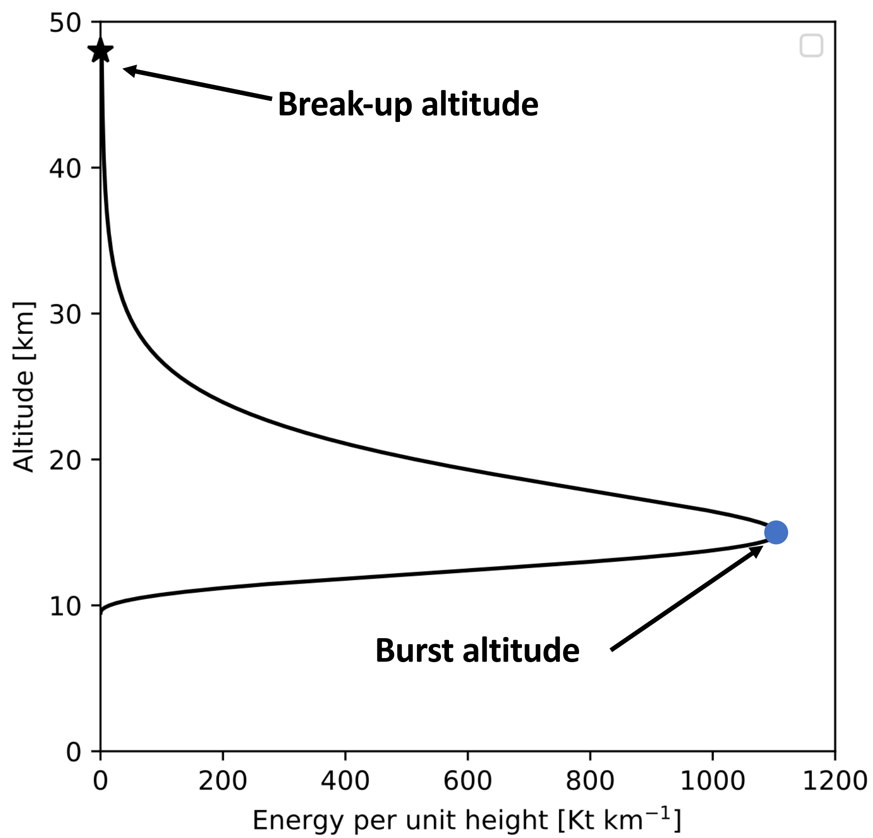
r is radius, α is dispersion coefficient.

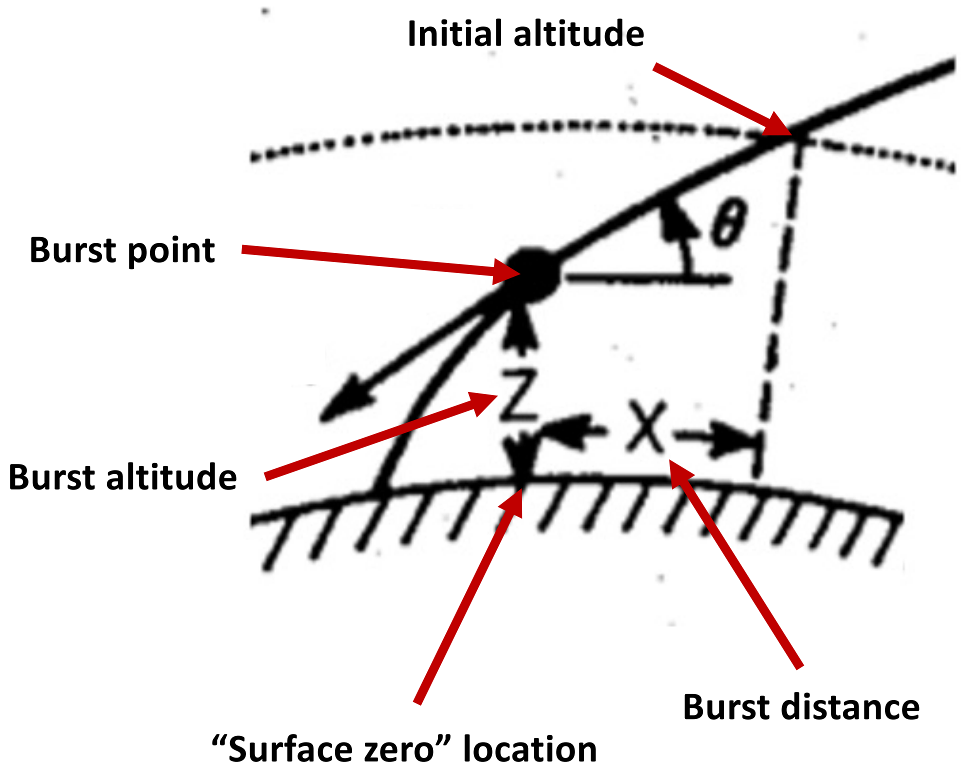
- Fragmentation ceases when ram pressure drops below strength of the meteoroid



Hills and Goda (1993) http://doi.org/10.1086/116499

An example solution



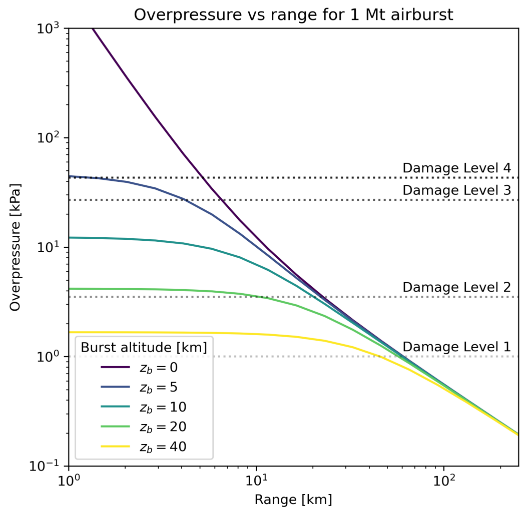


Burst energy is the **larger** of the **kinetic energy lost at burst** and the **residual kinetic energy at burst**

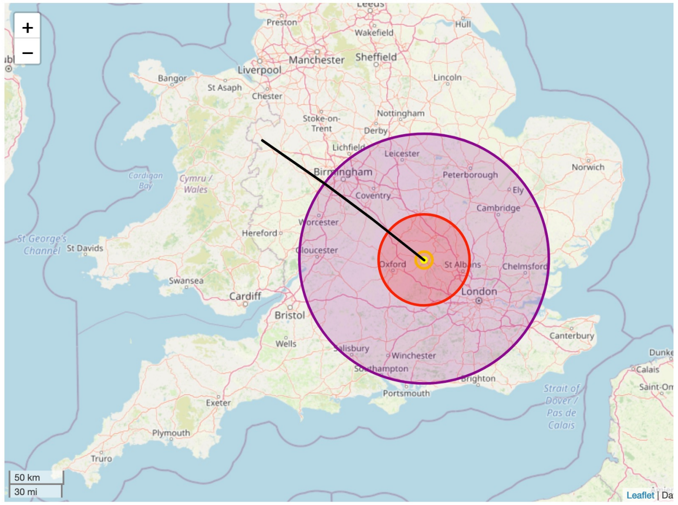
Airblast damage

Damage Level	Description	Pressure (kPa)
1	~10% glass windows shatter	1.0
2	~90% glass windows shatter	3.5
3	Wood frame buildings collapse	27
4	Multistory brick buildings collapse	43

Table 1: Pressure thresholds (in kPa) for airblast damage



The rapid deposition of energy in the atmosphere is analogous to an explosion and so the environmental consequences of the airburst can be estimated using empirical data from atmospheric explosion experiments (Glasstone and Dolan, 1977)



Challenge

(1) To develop a **Python program** with two features

- **a) Airburst solver** to predict burst energy and location
- **b) Airblast damage mapper** to plot damage zone & determine high-risk postcodes

(2) To prepare a **short video presentation**:

- a) Describe your approach and methods
- b) Verify your tool against an analytical solution and quantify accuracy
- c) Find asteroid radius and strength that gives best match to Chelyabinsk observation
- d) Demonstrate your tool for a specified scenario

Airburst solver: core functionality

The program should take as inputs:

- Asteroid radius
- Asteroid speed
- Asteroid density
- Asteroid strength
- Asteroid trajectory angle

For the core task you can assume an exponential (isothermal) atmosphere:

$$\rho_a = \rho_0 \exp(-z/H)$$

The program should be return a dataframe including the following variables:

- time, altitude, (horiz.) distance, velocity, mass, radius, **kinetic energy loss per km**

It should also return the following **five key results**:

- **Airburst** or a **cratering** event
- The **peak kinetic energy loss per km**
- The **burst altitude**
- The **burst energy**
- The **burst distance** (horizontal distance from entry point to burst point)

Airblast damage mapper: core functionality

The program should take as inputs:

- Entry latitude
- Entry longitude
- Entry bearing (degrees from north)
- Entry altitude

Plus the outputs from the airburst solver:

- Burst energy
- Burst altitude
- Burst distance

And return the following information, in a series of functions:

- The surface zero location of the airburst in latitude and longitude
- The airblast damage radii for four different damage thresholds
- The postcodes (or post-code sectors) inside the radius of each airblast damage level.
- The population of each of these postcodes (or post-code sectors).

UK Postcodes

	Postcode	LAU218CD	Latitude	Longitude
0	AB101AB	S31000932	57.149606	-2.096916
1	AB101AF	S31000932	57.148707	-2.097806
2	AB101AG	S31000932	57.149051	-2.097004
3	AB101AH	S31000932	57.148080	-2.094664
4	AB101AL	S31000932	57.150058	-2.095916

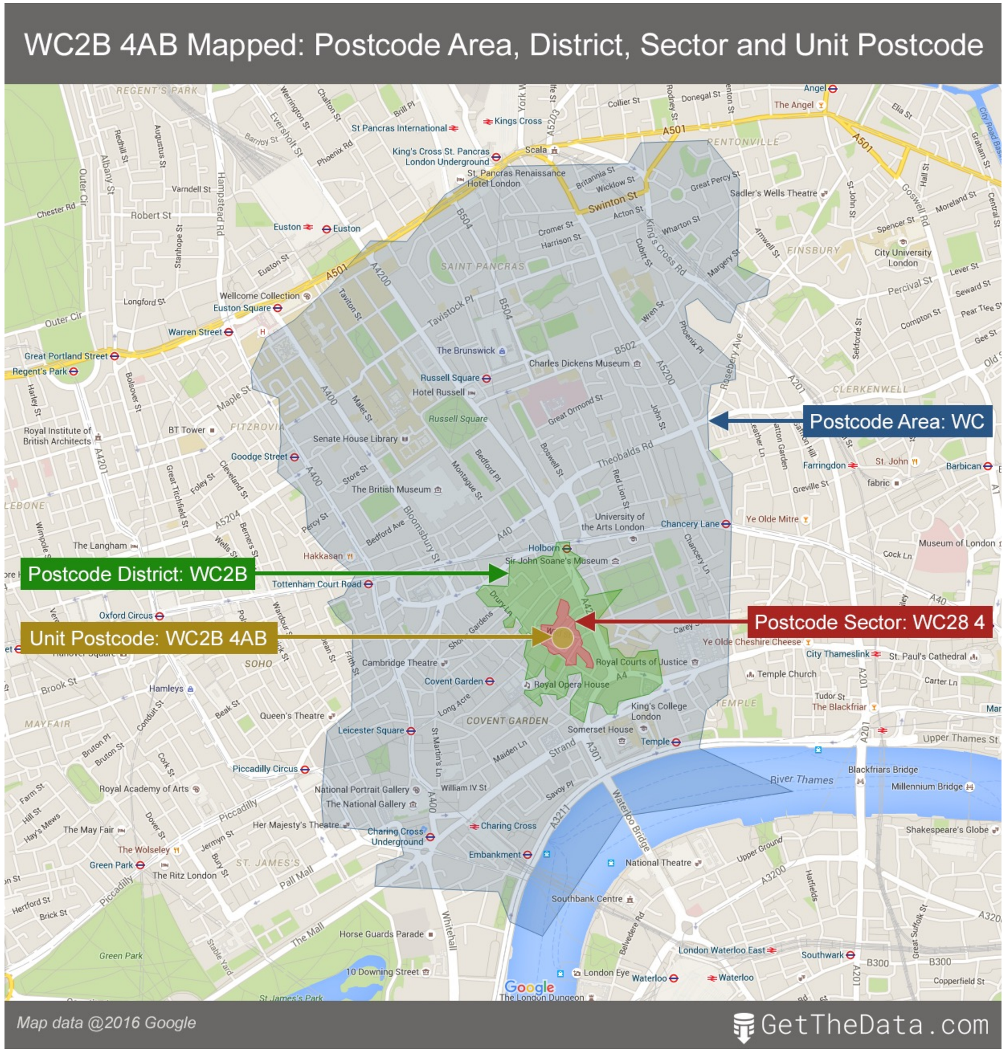
Postcode unit

Location of Postcode unit

	date	geography	geography code	Rural Urban	Variable: All usual residents; measures: Value
0	2011	AL1 1	AL1 1	Total	5453
1	2011	AL1 2	AL1 2	Total	6523
2	2011	AL1 3	AL1 3	Total	4179
3	2011	AL1 4	AL1 4	Total	9799
4	2011	AL1 5	AL1 5	Total	10226

Postcode sector

Population of postcode sector



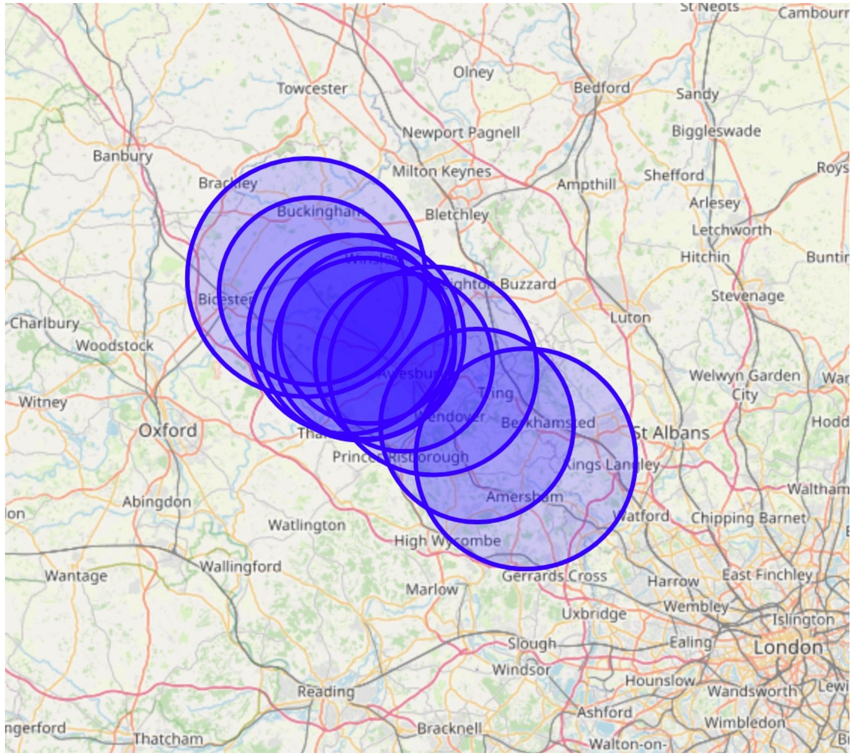
Imperial College London

DEPARTMENT OF EARTH SCIENCE AND ENGINEERING

Challenge: extended functionality

Additional tasks in increasing order of difficulty:

1. Implement a **tabular atmosphere** for Earth
2. Develop an **optimisation method** for finding the asteroid radius and strength that gives the best fit to an observed energy deposition curve for the Chelyabinsk meteor
3. Implement a way to **plot your results on a map**
4. Expand your tool to run multiple scenarios to quantify uncertainty in outcome and calculate **risk**



Risk = Probability × Cost

Likelihood of damage × Consequence of damage = Damage Risk

Probability that postcode sector is damaged × Number of people in postcode sector = People in postcode sector affected

Assessment

Scoring algorithm (50%)
(functionality & performance)

Airburst solver / outcomes
Analytical solution verification
Damage mapping
Postcode locations
Risk calculator

Presentation (20%)
(understanding & creativity)

ODE solver & searching
Accuracy / convergence
Optimisation method
Damage mapping
Risk calculation

Sustainability (20%)
Testing
Documentation
GitHub usage

Teamwork (10%)

Peer-evaluation

Video Presentations:

- Short (15-min) video presentations to summarise your results:
  - Describe your airburst solver solution algorithm, including ODE solving routine.
  - Demonstrate the accuracy of your numerical solution (e.g., analysis of error vs. timestep size or error vs. solver tolerance).
  - Demonstrate how to use your software to determine the optimal parameters (asteroid size & strength) for the Chelyabinsk airburst.
  - Describe your algorithm for finding postcodes within each damage zone
  - Describe your uncertainty quantification algorithm (risk calculation).
  - Demonstrate your software for a specified scenario that will be provided on Friday.
- Record with Teams and upload to MS Stream

Learning Outcomes

At the end of this exercise, you should have learned:

- To develop software collaboratively
  - The value of automated testing
  - Effective teamwork
- New technical skills:
  - Solve a coupled ODE system
  - Geospatial search algorithms
  - Simple uncertainty quantification
  - Manipulate data using Pandas
- Reinforced knowledge from MPM:
  - Programming with Python
  - Using GitHub for collaborative software maintenance
  - GitHub Actions for automated testing
  - Using Pandas for data manipulation
  - Using Sphinx for automated documentation

Final thoughts…

- This module is about collaboration: talk to each other; be kind to each other
- Choose a leader
- Make a sensible division of labour:
  - Airburst ODE Solver & Damage Mapper
  - Some smaller subtasks (e.g., analysis of results, outcome calculator, plotting)
  - Core vs Extension tasks
  - User-interface, plotting, documentation
  - Testing, verification and validation
- Work in pairs (effective way to program and more fun!)
- Questions we will answer:
  - Clarification about problem/requirements/assessment
  - Help with Python / git / GitHub / GitHub Actions / Sphinx
- Questions we won’t answer:
  - Choice of algorithm / approach. -> CM Lecture 7 is the place to start!
  - Help with debugging your code

Questions?