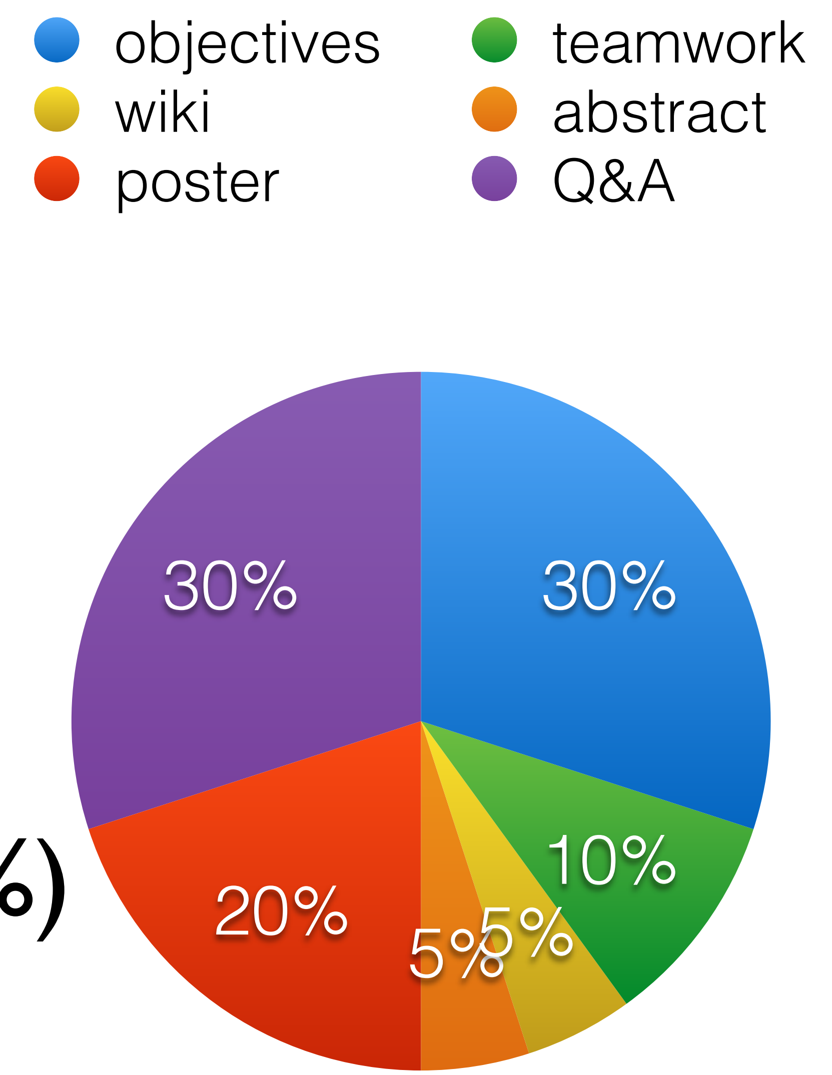
Final Project

PHY494 Computational Methods in Physics (2017)

Requirements

- team (3)
- GitHub repository
- keep “lab notebook” on wiki page
- final poster presentation
  - abstract, poster as PDF
  - description of contributions (PDF)
  - individual Q&A

Grading

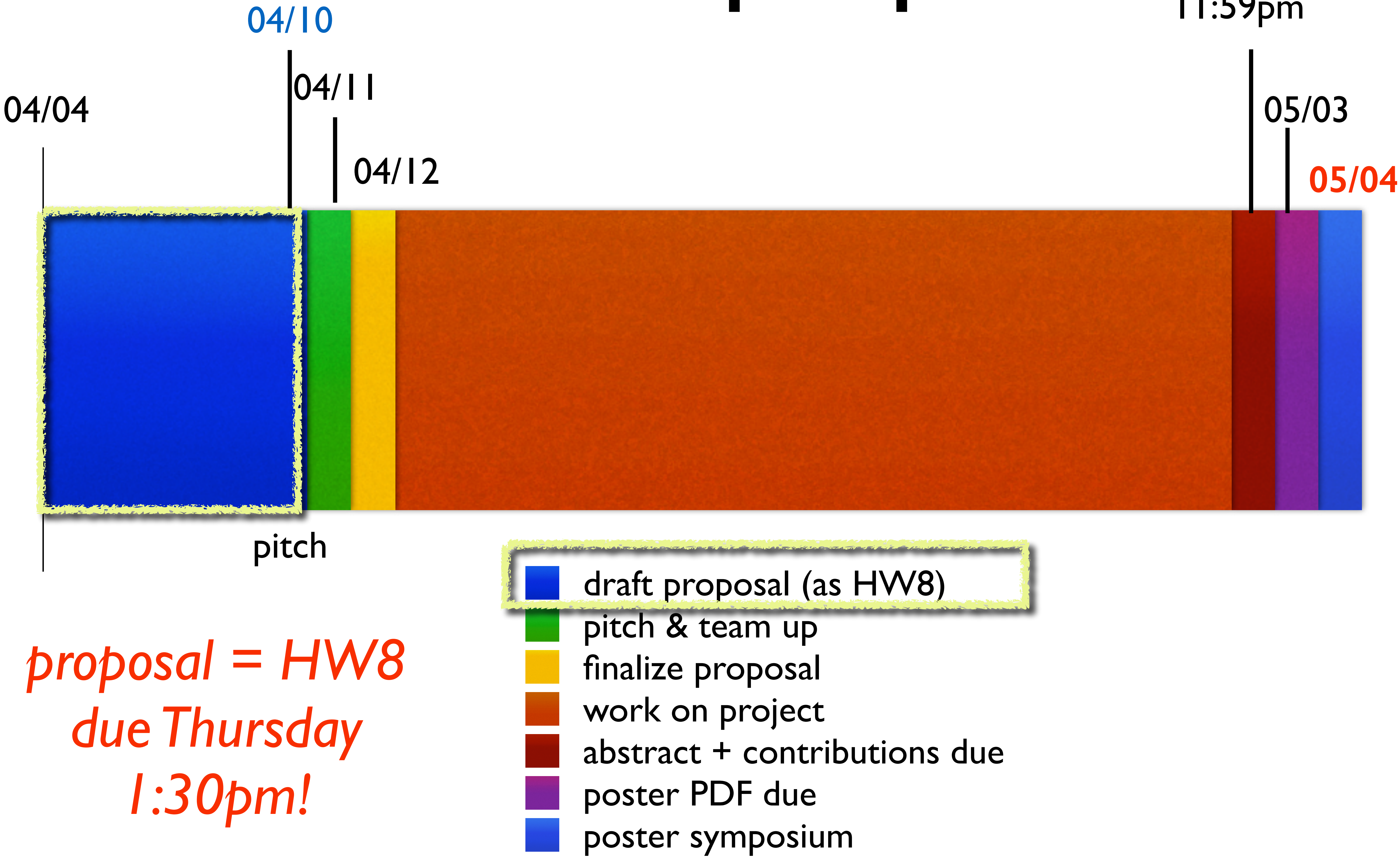


- code, achieves objectives (30%)
- teamwork (contributions, commits, evidence of communication) (10%)
- keep "lab notebook" on wiki page (5%)
- final poster presentation
    - abstract (5%)
    - poster (20%)
    - individual Q&A (30%)

Projects

- Select from list (see HW8)
- Propose your own.

Draft proposal



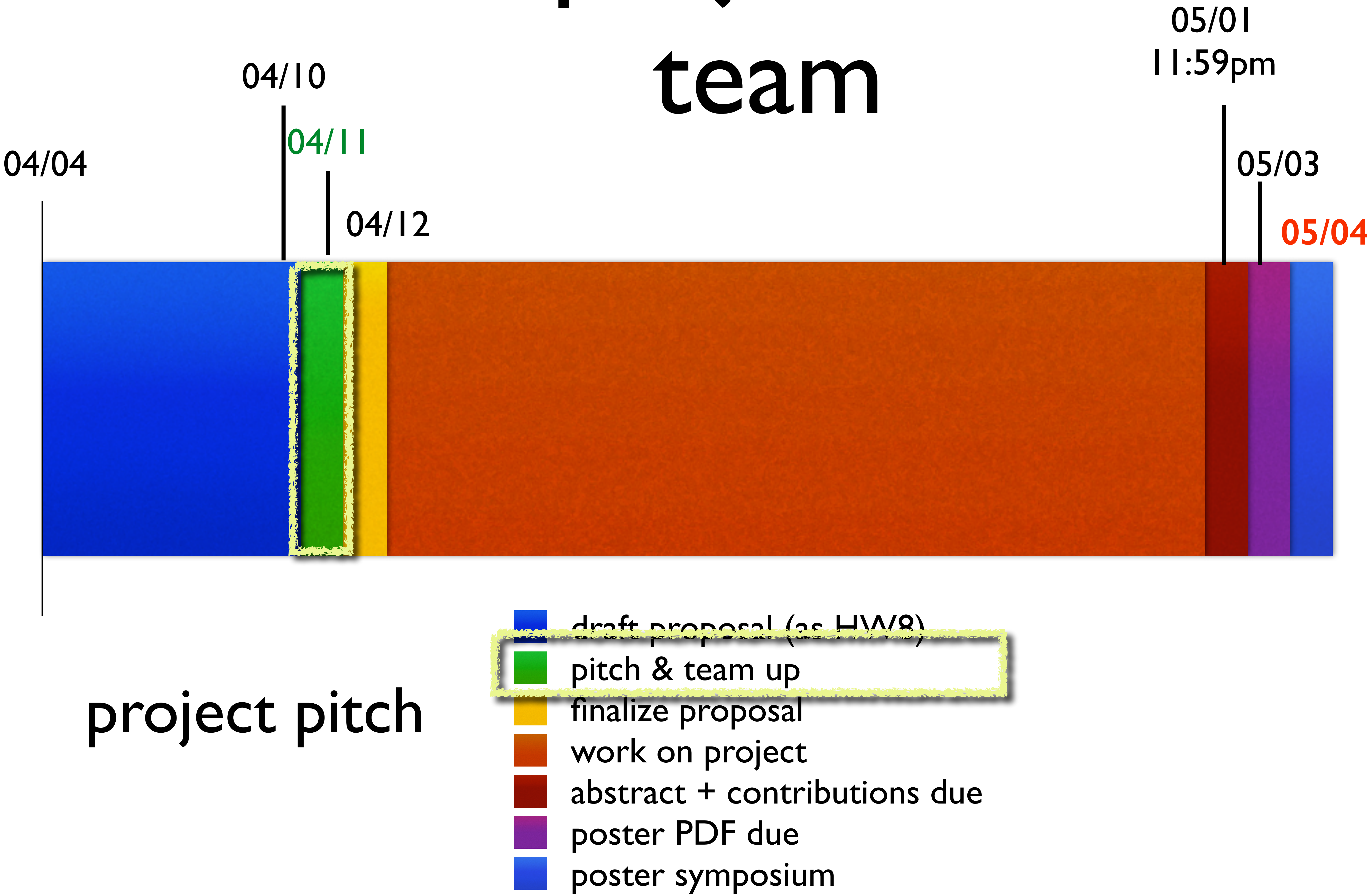
proposal = HW8
due Thursday
1:30pm!

Project Proposal

as HW8: due Monday 1:30pm!

- 1 page
- title
- *Problem:* Describe the problem to be solved.
- *Approach:* Algorithms, outline of how you will solve the problem, requirements (e.g. what data needs to be collected?)
- *Objectives*: List 3–6 measurable non-trivial outcomes that you want to achieve; your grade will depend on achieving these objectives

Pitch project & build team

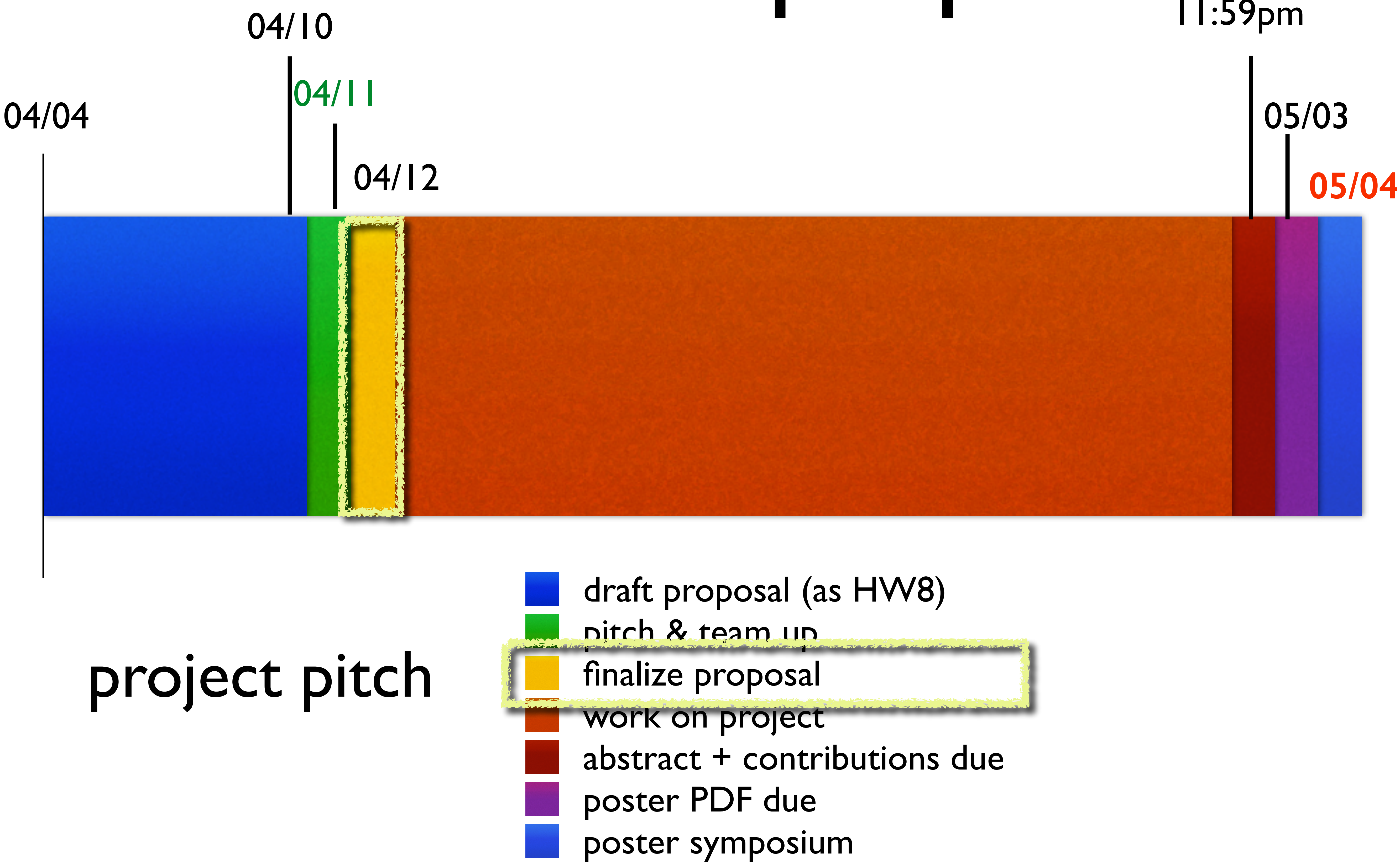


project pitch

Project Pitch

- ~5 min presentation (next Tuesday)
  - email any slides as PDF in advance
- introduce project (*Problem*, *Approach*, *Objectives*)
- attract a team (3 students)!
- You can only *choose* a project that was pitched… so be prepared to pitch yourself!

Finalize proposal

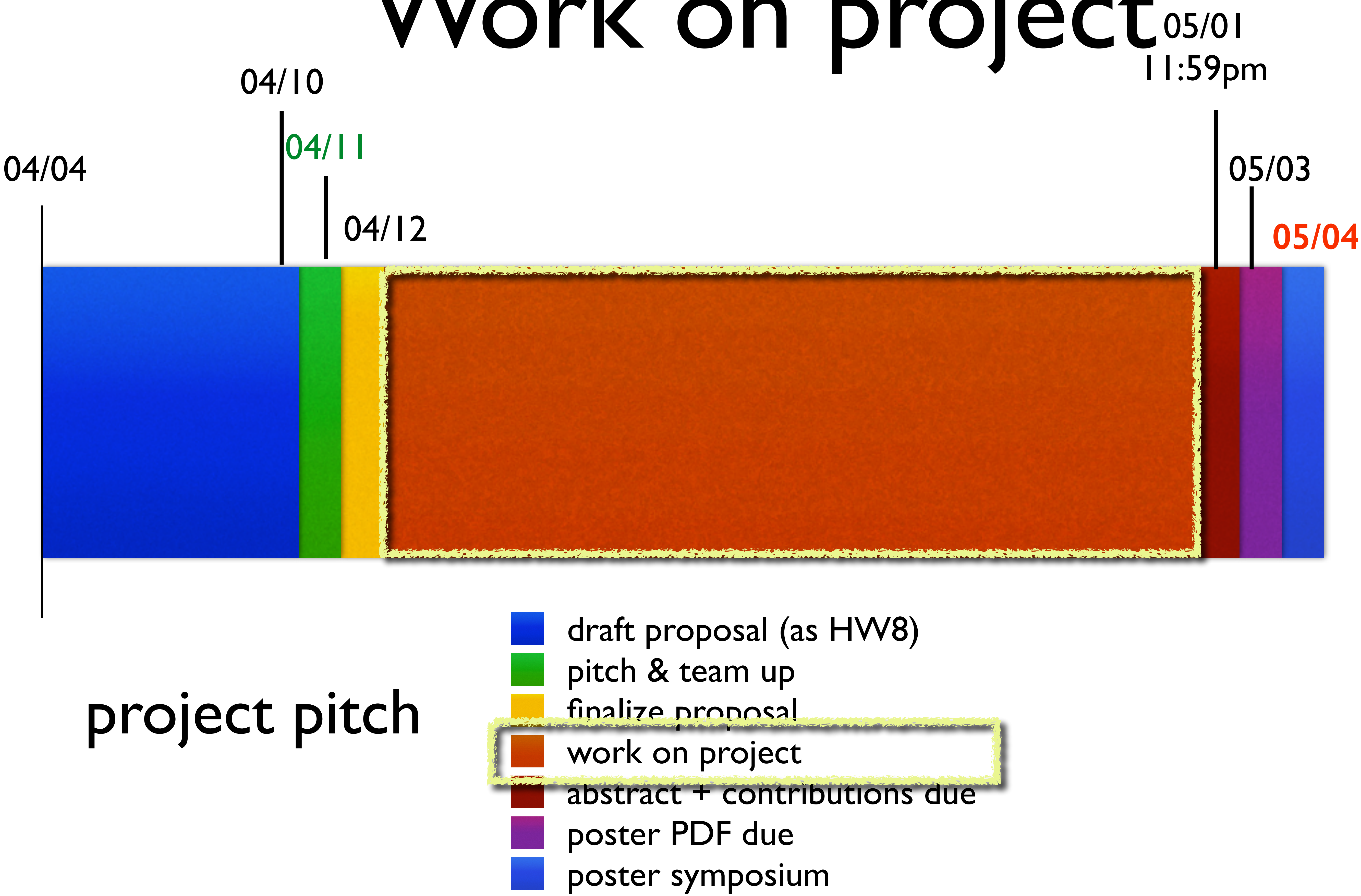


project pitch

Finalize proposal

1. work as a team to write the final proposal
2. submit to team repository
3. final proposals will be evaluated by instructor and requested changes will be posted as issues: need to be addressed

Work on project



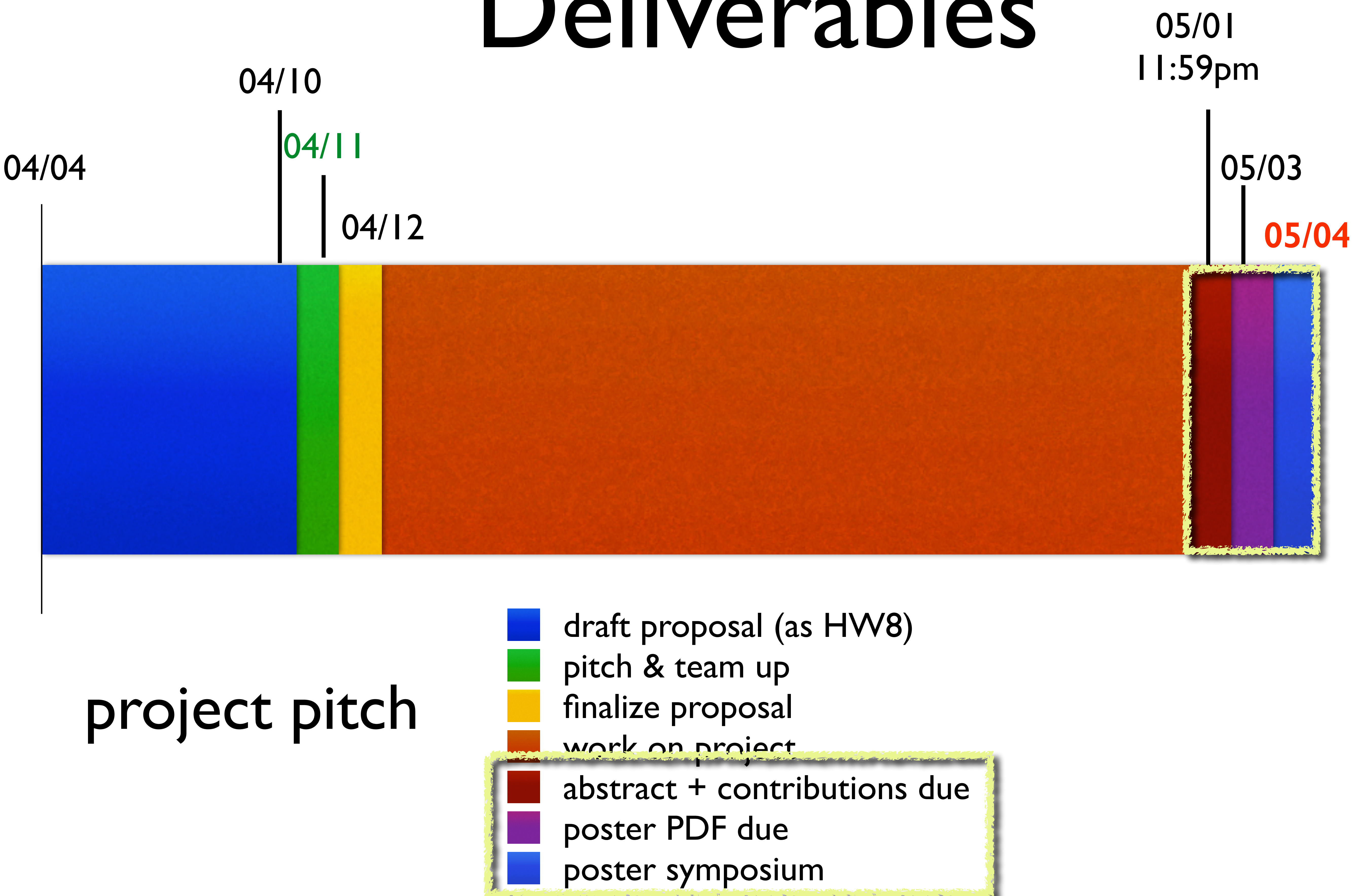
project pitch

Work on project

- use GitHub **repository** for code
- use GitHub **Wiki** for organization, notes on algorithms, preliminary results
- *optional*: use GitHub issue tracker for assigning tasks and keeping track of bugs and objectives
- Create **abstract** and **poster** together.

Start early!

Deliverables



project pitch

Project ideas

- rough outline: expand in your proposal
- see also *Computational Physics* and *Computational modelling and visualisation of physical systems with Python*

1. Monte Carlo Simulation of liquid Argon

- difficulty: 1
- implement basic MC for liquid Argon in *NVT*
  - analyze at different temperatures (phase transition?)
  - calculate equation of state $P(T, \rho)$
- visualize
- extra: implement *NPT* (volume moves)

2. Molecular Dynamics Simulation of liquid Argon

- difficulty: 2
- implement basic MD for liquid Argon in *NVE*
  - energy conservation, time step
  - liquid structure (radial distribution function)
- visualize

3. Monte Carlo simulation of the Ising magnet

- difficulty: 2–3
- implement MC for Ising spin system in 1D, 2D, possibly 3D
- compute magnetization as function of T (phase transition?)
- visualize
- compare to 2D exact result

4. Classical chaotic scattering

- difficulty: 1
- Problem 9.4 in *Computational Physics*
- integrate equations of motions of particles scattering from “4-peak” potential (e.g. with RK4)
- vary parameters
- analyze cross section
- visualize

5. Quantum Mechanics: Wave packet propagation

- difficulty: 3
- See 22.2 and 22.3 in *Computational Physics*
- solve the time-dependent Schrödinger equation for a Gaussian wave packet in different potentials (Visscher or Maestri/Askar&Cakmak algorithm)
- 1D: step barrier, harmonic well, square well; 2D: slit
- calculate transmission/reflection, wave velocity, ...
- visualize

6. Fluid mechanics: 2D Navier-Stokes

- difficulty: 2 (3 without *CP*…)
- See Ch 25 in *Computational Physics*
- solve Navier-Stokes for incompressible 2D flow (finite difference with SOR algorithm)
- velocity (flow) field and vorticity for submerged beam, cylinder, drop shape
- visualize
- vary parameters (e.g. Reynolds number, velocities, boundary conditions)

7. Energy efficient houses

- difficulty: 4
- solve the heat equation for a 2D house
- implement house plan with realistic parameters for walls, windows, doors and typical shapes and sizes
- How much energy is needed to maintain interior at constant T throughout a Phoenix year?
- What is the effect of increasing insulation, shading windows, changing target temperature?

8. Solar system

- difficulty: 1
- simulate the solar system (planets + sun + various comets such as Halley's comet) using classical Newtonian mechanics
- obtain realistic parameters from NASA
- stability over time?
- effect of planets on comets?

9. Stock market forecasting with wavelets

- difficulty: 3
- See e.g. Yousefi et al (2005), doi:10.1016/j.chaos.2004.11.015
- implement a wavelet analysis (Daubechies wavelets) for stock market and commodity prices timeseries
- test how well you can forecast (correlation between "predicted" data and data not used for the wavelet analysis)
- vary forecast range

10. Agent based modelling of self-driving cars

- difficulty: 2 (?)
- “Once cars can talk to each other, we will not need any traffic lights or even restrictions on which side of the road they can drive, it will look very organic, e.g. like water flowing.”
- Use agent-based modelling to simulate large numbers of cars on a small street network.
- Analyse traffic flow as function of car density (and rules)

11. Analysis of natural motion

- difficulty: 3 (?)
- motion (walking, swimming, flying) can be decomposed into very few components using SVD
- example: Girdhar K, Gruebele M, Chemla YR (2015) The Behavioral Space of Zebrafish Locomotion and Its Neural Network Analog. PLoS ONE 10(7): e0128668. doi:10.1371/journal.pone.0128668
- use videos and image processing + SVD
- gait analysis: is how you walk like a finger print?
- What are the most complex motions?