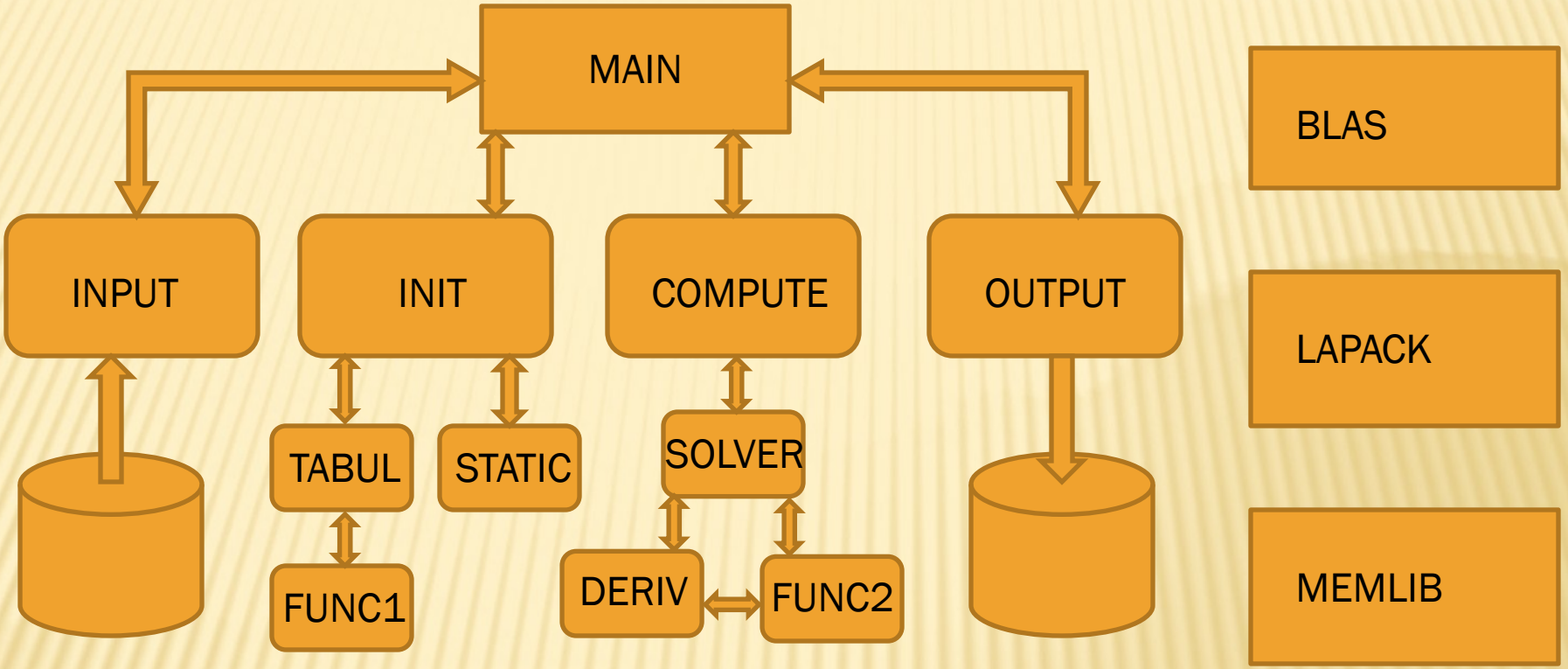


Scientific Programming

LECTURE 5: STYLE AND STRUCTURE

HOW TO DECIDE ON THE STRUCTURE?

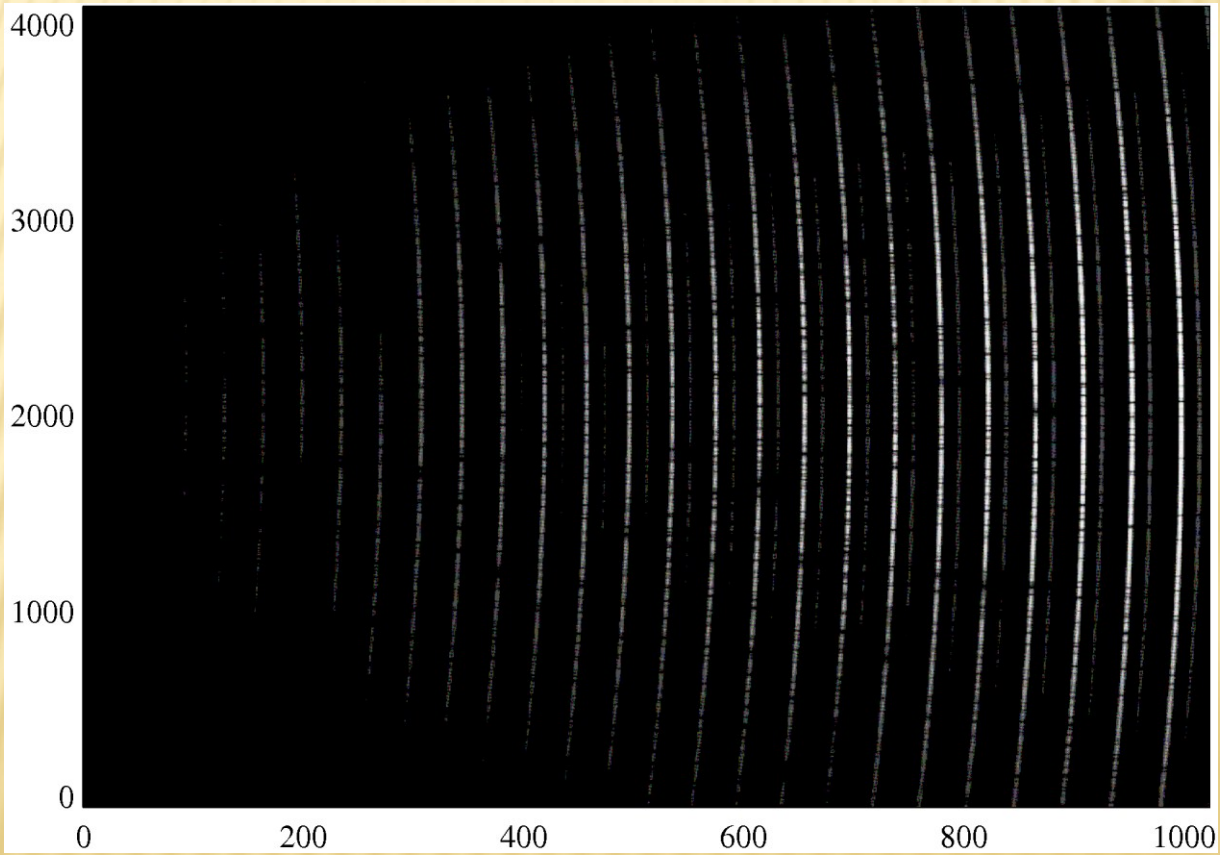
- How to split the code in modules and subroutines?
- How to select the interface?
- Many runs may require automatic documentation
- Long runs may need interrupt/continuation possibility
- How the results are going to be presented?

WHAT COULD BE THE GOALS OF A SERIOUS CODE? (EXAMPLES)

- Artificial Intelligence: object detection/classification
- Data modelling
- Dynamic models

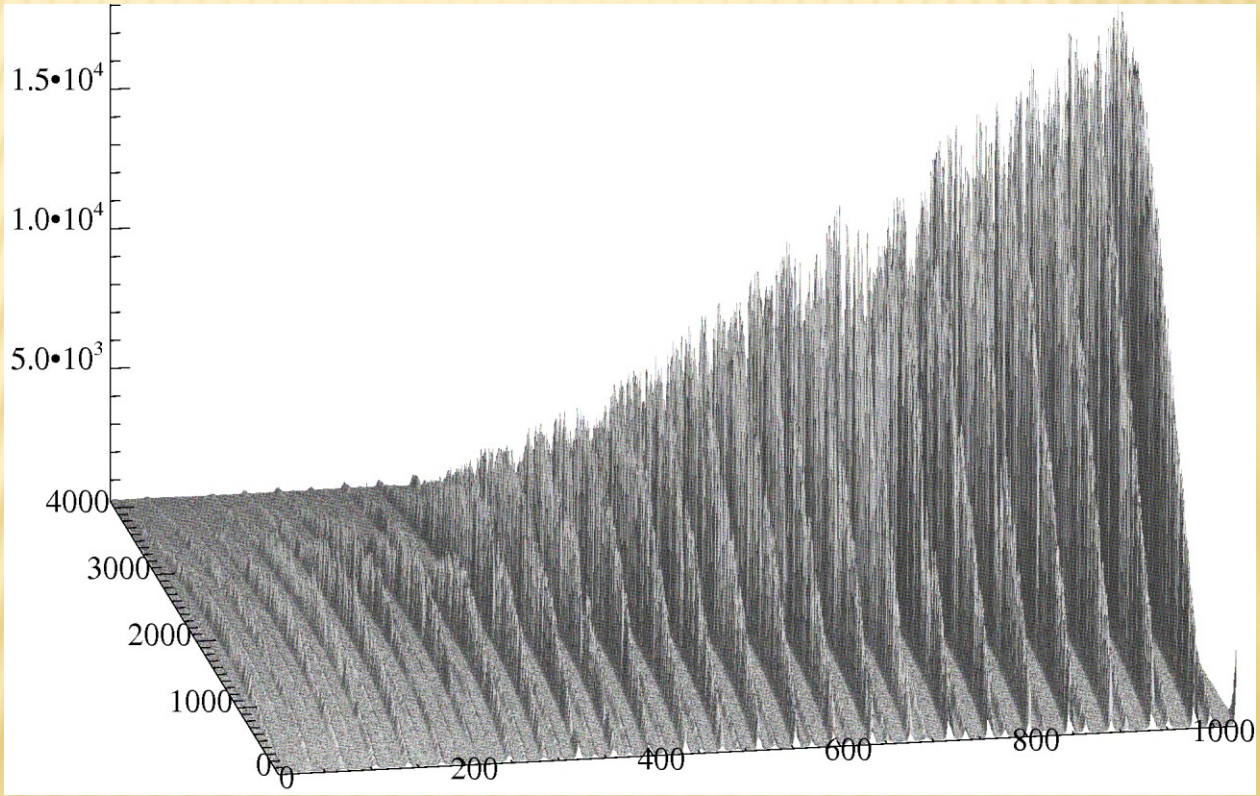
AI: OBJECT DETECTION/CLASSIFICATION

Do you see structures on this image?



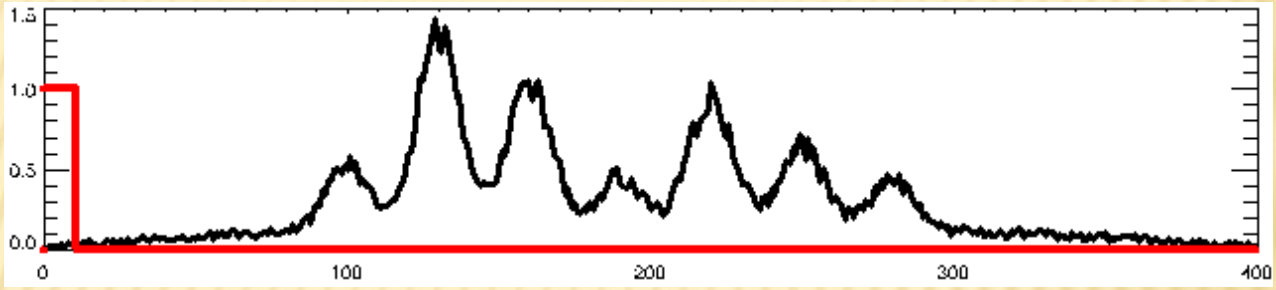
AI: OBJECT DETECTION/CLASSIFICATION

... but your computer code has to deal with this:



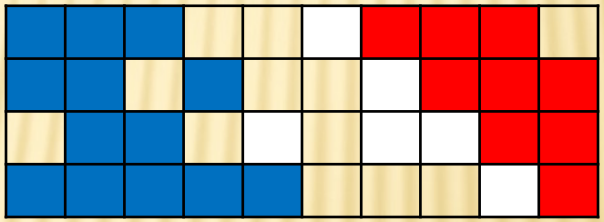
AI: ONE POSSIBLE ALGORITHM

- Defining pixels that contain signal



- Identifying clusters of pixels

$$\left\{\vec{x}, \vec{y}\right\} + \vec{c}$$



- Fitting some analytical shape function to the clusters ⟶ polynomial coefficients for each cluster
- Merging clusters ⟶ re-doing polynomials

HOW WILL YOU WRITE SUCH CODE?

- What do you need from this code?
- How would you check that the code does the right thing?
- What programming language will you use?

DATA MODELLING

- You have measurements and physical model controlled by some parameters.
- You want to find the set of parameters that realizes the best fit to your data
- For example, fitting stellar spectrum
  - Equations: $$\frac{dI_\lambda}{dx_\lambda} = j_\lambda - \alpha_\lambda(\vec{Z})I_\lambda$$

    $$\sum_\lambda \omega_\lambda \left[I_\lambda - O_\lambda \right]^2 = \min$$
  - Parameters: chemical composition Z
  - Given chemical composition we solve for intensity
  - Then we compare with the observations and adjust chemical composition

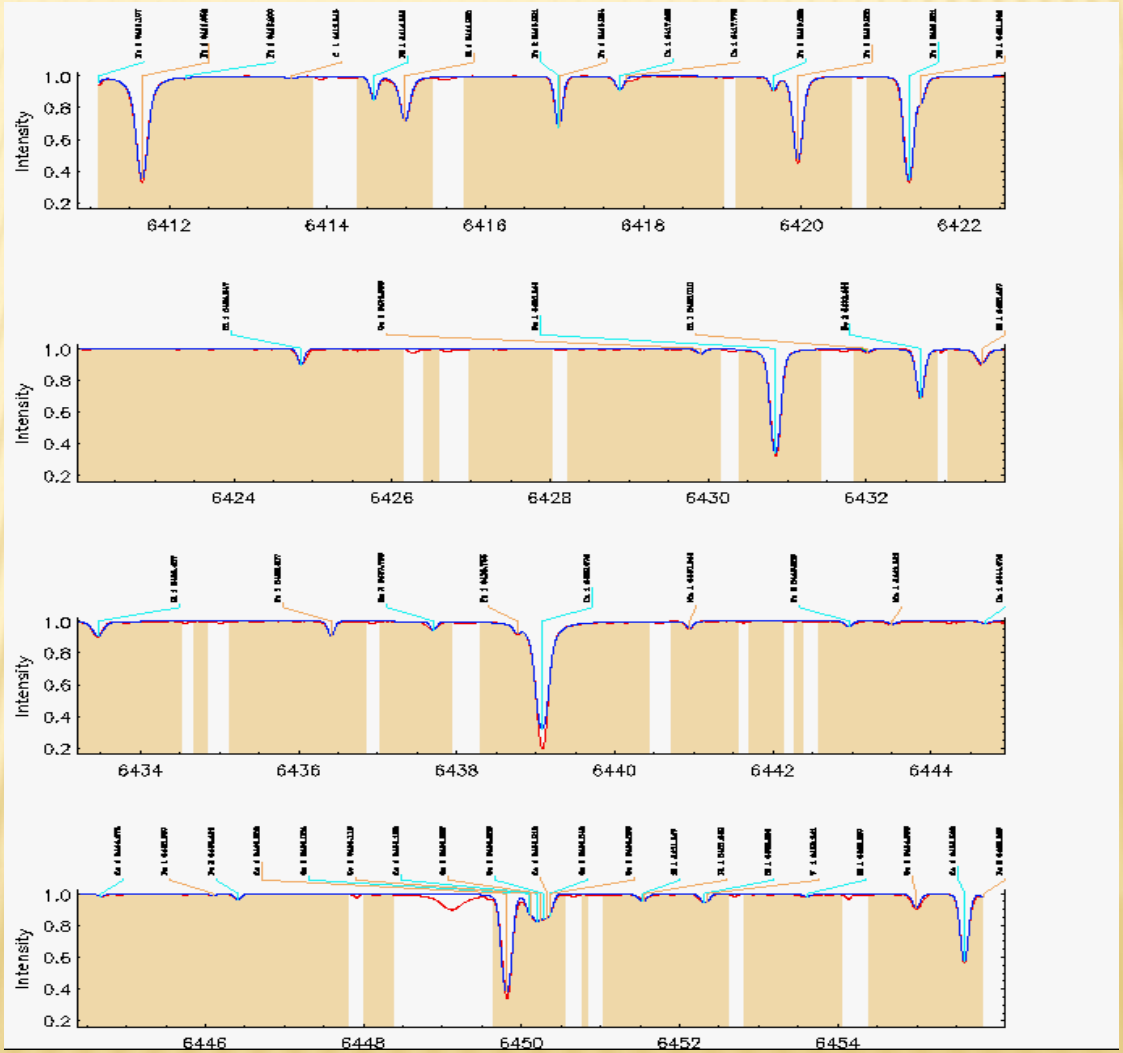
DATA MODELLING

Outstanding issues:

- Do we search for abundance Z of one element at a time?
- Can we associate certain elements of Z and wavelength intervals?
- Is there a clever way to wind optimal Z?
  E.g. Marquardt-Levenberg algorithm, but this requires 1st derivatives over Z.
- Subroutines:
  `input` – reads in observations and line data
  `init` – computes
  `process` – does the optimization, needs evaluation of intensities and its 1st derivatives over Z
  `output` – reports the result

DATA MODELLING

Here is how the result of such optimization may look like:



COMPONENTS

- Input stellar model and spectral line parameters
- Pre-compute parts of the equations not affected by chemical abundances
- Select initial guess

Loop:

- Compute synthetic spectrum and its 1st derivatives
- Adjust abundances using one of the optimization techniques
- Loop until a good fit is found
- Save the results

HOW WOULD YOU STRUCTURE SUCH CODE?

- Where will the code spend most of the time?
- How do you know the optimization algorithm is working?
- The relation between abundances and synthetic spectrum is highly non-linear. How will you compute your derivatives?
- What do you want to save besides the optimal set of abundances?

EVOLUTIONARY MODELS

3D hydrodynamic simulations with radiative energy transport

Hydro

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &+ \boldsymbol{\nabla}\cdot(\rho\,\boldsymbol{v}) &&= 0 \\
\frac{\partial \rho \boldsymbol{v}}{\partial t} &+ \boldsymbol{\nabla}\cdot\left(\rho\,\boldsymbol{v}\!:\!\boldsymbol{v} + P\,\bar{\bar{\mathsf{I}}} + \bar{\bar{\mathsf{P}}}_{\mathrm{rad}}\right) &&= \mathbf{0} \\
\frac{\partial \rho e_{\mathrm{ik}}}{\partial t} &+ \boldsymbol{\nabla}\cdot\left(\left[\rho\,e_{\mathrm{ik}} + P\right]\boldsymbol{v} + \boldsymbol{F}_{\mathrm{rad}}\right) &&= 0
\end{aligned}$$

Radiation

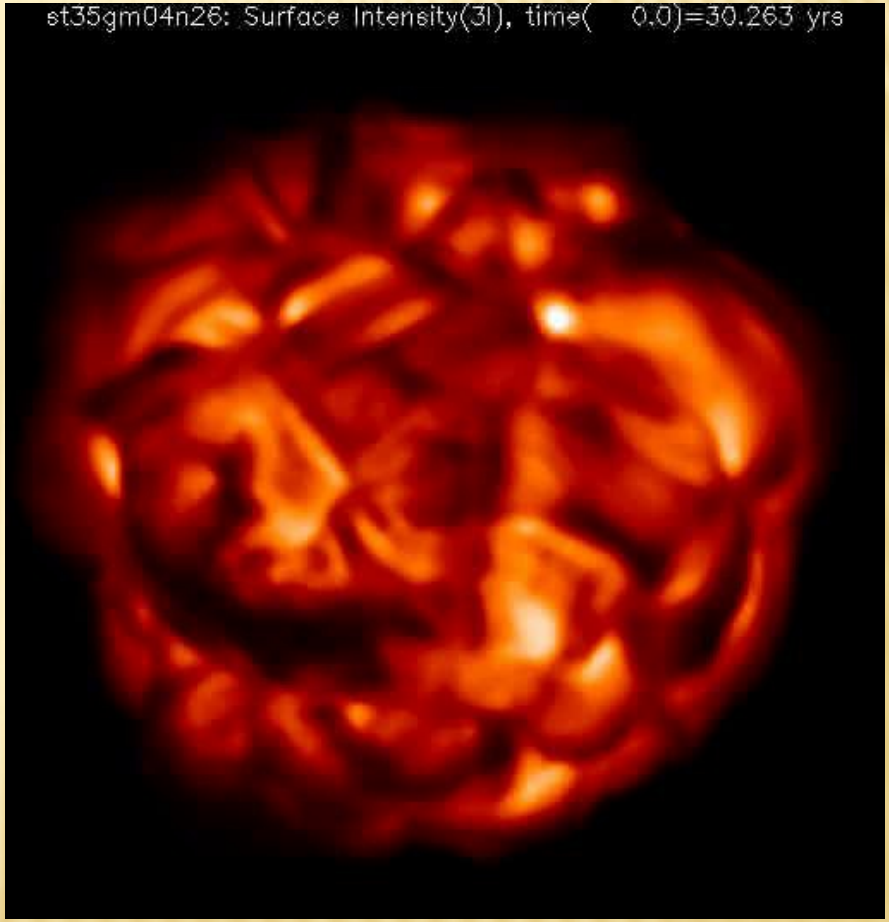
$$\boldsymbol{F}_{\mathrm{rad}}(\boldsymbol{x}) = \int_0^\infty \oint_\Omega \hat{\boldsymbol{n}}\, I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}, \nu)\, d\omega\, d\nu$$

$$\bar{\bar{\mathsf{P}}}_{\mathrm{rad}}(\boldsymbol{x}) = \frac{1}{c_{\mathrm{l}}} \int_0^\infty \oint_\Omega \hat{\boldsymbol{n}}\!:\!\hat{\boldsymbol{n}}\, I_\nu(\boldsymbol{x}, \hat{\boldsymbol{n}}, \nu)\, d\omega\, d\nu \quad .$$

EVOLUTIONARY MODELS

Evolutionary models cannot be compared with observations directly.

Therefore, post processing is crucial (analysis of various statistical, spectral and integral properties).



HOW IS THIS CODE DIFFERENT FROM PREVIOUS EXAMPLES?

- Implicit/explicit scheme
- Selection of the PDE solver
- Advancing in time the hydro part
- Adjusting the radiation part
- Main issues: performance and resolution
- Parallelization?

CONCLUSIONS

1. Many AI-type algorithms are hard to make robust. In same cases it is simply not possible. This requires a comfortable user interface and, perhaps, built-in graphics. Select the programming language wisely.
2. Data models and dynamic models: selected algorithm dictates code structure.
3. Think about the parameters that you are going to change. Make sure you do not need to modify the code to vary those parameters (e.g. grid size).
4. Complex numerical algorithms often require fine tuning and studying large parameter space. This requires lots of numerical experiments. Incorporate in the code the ability to propagate to the output the parameters of each experiment (self-documentation).

CONCLUSIONS

5. Evolutionary codes have to run for long time to “forget” the initial conditions. In fact, it is often better to start from the results of a run with different set of parameters and than start from scratch. Make sure all the necessary results are saved to file(s) reasonably often so you can re-start you calculations.
6. Think in advance what you may need to interrupt your code, so make sure all information is saved regularly.
7. Think about “easy to grasp” ways of presenting your results.

NEXT LECTURE: OPTIMIZATION

The lecture is on October 3rd (Monday).

Tomorrow is the last chance to present the 1st homework

Start preparing for the 2nd homework round (see the course web page).