

Homework info and requirements

NLA 2018 team

October 29, 2018

1. General info

Problem sets are distributed in Jupyter Notebooks and will be uploaded in both Canvas and [GitHub repository](#). We recommend using Anaconda Python Distribution which allows easy installation of all packages required for the course.

Problem sets include both theoretical problems and coding. Each problem in a problem set should be done in a separate cell. Theoretical problems should be solved inside Jupyter Notebooks using [Markdown-mode](#), which partially supports \LaTeX . Equations in your solutions have to be written in \LaTeX -mode. Coding problems should be solved in Python. Students who are not experienced with Python may find useful tutorials [here](#).

Besides standard problems you will find “bonus problems”. They are challenging and non-obligatory. However, they may help you if, for instance, your grade is slightly below higher grade threshold.

This year we will use Telegram bot¹ to automatically check coding parts of your HW solutions. We will be very glad to receive feedback on our bot, so feel free to report any bugs!

Each problem set may have its own number of points. Total score for all problem sets will be then scaled to be 50% of the total grade.

2. Submission details

1. Submission should be a single Jupyter Notebook named in the following fashion:
`surname_name_pset-number.ipynb`, e.g. `oseledets_ivan_1.ipynb`
2. You can submit solutions many times, but only the last one will be graded
3. After deadline you can not submit your solution

3. How you should code in Python

1. Use `numpy` arrays instead of built-in lists.
2. Avoid `import *`!!! Namespaces is a feature.
3. Try not to use loops in your code as far as it is possible. Vectorized code with `numpy` arrays works much more efficiently!
4. For those of you, who are familiar with MATLAB, we provide a [link to manual](#) with correspondence between MATLAB and `numpy` commands.
5. We recommend that you follow [PEP8 style guide](#).
6. Define variables and functions with meaningful names.
7. Avoid dynamic memory allocation.

¹As soon as it's operational we will share the link

4. How to plot figures and not make TAs angry

Here we list requirements for plots that MUST be satisfied.

1. Every axis should have its own label.
2. Axis labels should have appropriate font size to be readable (12pt is ok).
3. Every figure should have explaining legend.
4. Every figure should have its own title. It should help TAs to understand what a figure is about.
5. Every figure should have appropriate scale. Particularly, you have to choose log-scale or linear scale depending on lines you plot. See Fig. 1 to compare these scales. In this example the log-scale is preferable and plot in linear scale will be penalized.

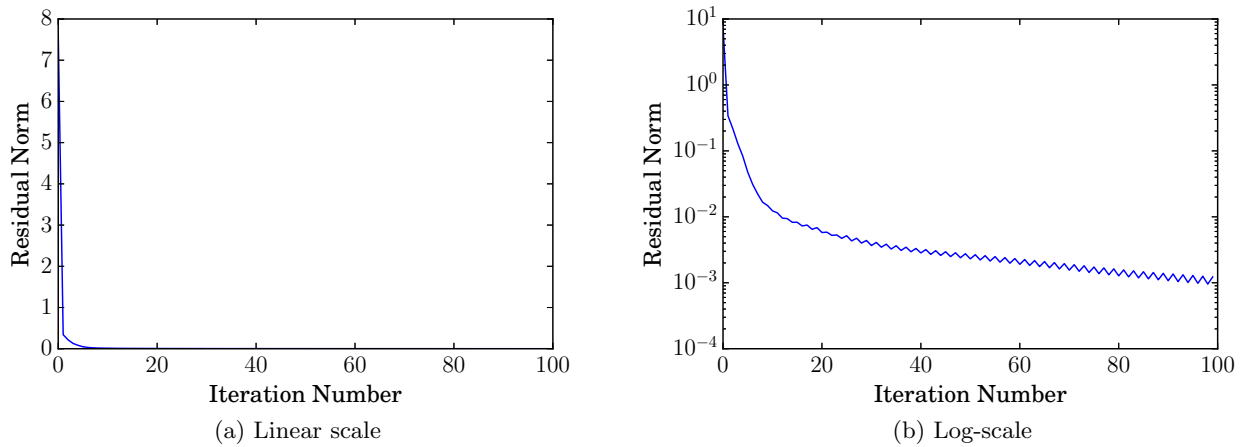


Figure 1: Comparison scales of plots

5. Cite your sources

Copying someone else's homework or copying solutions from the internet or elsewhere and presenting them as your own is strongly prohibited. If you use other sources (including other students) you must cite them explicitly. However, if your work is fully copied (up to renamed variables) or you did not mention a source you will be penalized down to 0 pts for that task.