

Exercise sessions

Welcome to the course “*Introduction to Optimization and Operations Research*”.

At each exercise session, you will be expected to have a computer available, and you will solve exercises using Jupyter notebooks. Jupyter notebook files will be available at the github link on the moodle home page. Each session will have one notebook with the questions and another notebook with solutions. We encourage you to try to solve the questions without looking at the solutions and to use Chatgpt or your preferred LLM to assist you with coding questions.

There are different software tools that you can use to run Jupyter notebooks (e.g., conda) but EPFL offers a tool called “noto” which requires no installation, and you can access with your gaspar account.

You can run notebook files with noto.

To get started click on this link Noto exercise in moodle, sign in with your gaspar account and make sure that you select python3 in the kernel.

Once in noto, you can browse and access all the exercises from github from noto.

Lab Sessions – Guidelines and Schedule

General Guidelines

- All exercises are provided in **pairs of Jupyter notebooks**:
 - The **questions notebook** contains the problem statement and skeleton code.
 - Your task is to complete the missing code.
 - The **solutions notebook** contains the fully worked-out answers.
- If you get blocked, **consult the solutions**.
 - If you are new to Python, you will likely need to do this regularly at first.
 - This is perfectly normal and a good way to learn.
- You may also consult **online resources** and **AI agents**.
 - Use them carefully: the goal is to understand your work.
 - Solving the exercises without comprehension is not useful.
- Below you will find a **suggested organization of topics** throughout the semester.
 - This is **indicative** only.

- In the first weeks, you may not have time to complete all the suggested material.
- This is fine — you can return to it later.

Weekly Breakdown

September 12

Python primer / refresher

- `topic_00_python/primer.ipynb`
- `topic_00_python/lab01_python_questions.ipynb`

Modeling (to be continued next week)

- `topic_01_modeling/lab02_projectile_questions.ipynb`
- `topic_01_modeling/lab03_projectile_2_questions.ipynb`
- *(if time)* `topic_01_modeling/lab01_coola_questions.ipynb`

September 19

Modeling

- `topic_01_modeling/lab04_kkt_questions.ipynb`
- `topic_01_modeling/lab05_standard_form_questions.ipynb`
- `topic_01_modeling/lab06_existence_questions.ipynb`

Linear constraints (if time)

- `topic_02_linear_optimization/lab01_feasibility_questions.ipynb`

September 26 – Linear constraints (to be continued next week)

- `topic_02_linear_optimization/lab02_feasible_directions_questions.ipynb`
- `topic_02_linear_optimization/lab03_bases_questions.ipynb`

- [topic_02_linear_optimization/lab04_basic_directions_questions.ipynb](#)

October 3

Linear constraints

- [topic_02_linear_optimization/lab05_reduced_costs_questions.ipynb](#)
- [topic_02_linear_optimization/lab06_active_constraints_questions.ipynb](#)
- [topic_02_linear_optimization/lab07_redundant_constraints_questions.ipynb](#)

Simplex algorithm (to be continued next week)

- [topic_03_simplex/lab01_enumeration_questions.ipynb](#)
- [topic_03_simplex/lab02_graphical_questions.ipynb](#)

October 10 (4 periods) – Simplex algorithm

- [topic_03_simplex/lab03_simplex_questions.ipynb](#)
- [topic_03_simplex/lab04_tableau_questions.ipynb](#)
- [topic_03_simplex/lab05_pivoting_questions.ipynb](#)
- [topic_03_simplex/lab07_phase_one_questions.ipynb](#)

Additional exercises

- [topic_03_simplex/lab08_multiple_choice_questions_questions.ipynb](#)
- [topic_03_simplex/lab06_simplex_tableau_questions.ipynb](#)

October 17 – Duality

- [topic_04_duality/lab01_feasibility_questions.ipynb](#)
- [topic_04_duality/lab02_dual_problem_questions.ipynb](#)
- [topic_04_duality/lab03_complementarity_slackness_questions.ipynb](#)

October 24

Mid-term break

October 31

Mock exam

November 7 – Networks

- [topic_05_networks/lab01_network_representation_questions.ipynb](#)
- [topic_05_networks/lab02_flows_divergences_questions.ipynb](#)
- [topic_05_networks/lab03_trees_questions.ipynb](#)

November 14 – Transshipment (to be continued next week)

- [topic_06_transshipment/lab01_total_unimodularity_questions.ipynb](#)
- [topic_06_transshipment/lab02_shortest_path_questions.ipynb](#)
- [topic_06_transshipment/lab03_standard_form_questions.ipynb](#)
- [topic_06_transshipment/lab04_optimality_conditions_questions.ipynb](#)

November 21

Transshipment

- [topic_06_transshipment/lab05_maximum_flow_questions.ipynb](#)
- [topic_06_transshipment/lab06_transportation_questions.ipynb](#)

Additional exercise

- [topic_06_transshipment/lab07_train_tickets_questions.ipynb](#)

Shortest paths (to be continued next week)

- [topic_07_shortest_paths/lab01_generic_algorithm_questions.ipynb](#)
- [topic_07_shortest_paths/lab02_bellman_questions.ipynb](#)

November 28 – Shortest paths

- [topic_07_shortest_paths/lab03_dijkstra_algorithm_questions.ipynb](#)
- [topic_07_shortest_paths/lab04_pert_questions.ipynb](#)

December 5 – Discrete optimization (to be continued next week)

- [topic_08_discrete/lab01_branch_and_bound_questions.ipynb](#)
- [topic_08_discrete/lab02_modeling_questions.ipynb](#)
- [topic_08_discrete/lab03_set_covering_questions.ipynb](#)
- [topic_08_discrete/lab04_tsp_questions.ipynb](#)

December 12 (4 periods)

Discrete optimization

- [topic_08_discrete/lab05_relaxation_questions.ipynb](#)

Nonlinear optimization

- [topic_09_nonlinear/lab01_first_wolfe_questions.ipynb](#)
- [topic_09_nonlinear/lab02_second_wolfe_questions.ipynb](#)
- [topic_09_nonlinear/lab03_newton_local_questions.ipynb](#)
- [topic_09_nonlinear/lab04_preconditioning_questions.ipynb](#)
- [topic_09_nonlinear/lab05_validity_questions.ipynb](#)
- [topic_09_nonlinear/lab06_linesearch_questions.ipynb](#)
- [topic_09_nonlinear/lab07_multiple_choice_questions.ipynb](#)

December 19

No class – Catch up at home on exercises you did not complete.

Example of ChatGPT answer:



How do I merge two lists together in a numpy array in python ?



To merge two lists together into a NumPy array in Python, you can follow these steps:

1. **Import the NumPy library:** Make sure you have NumPy installed and then import it.
2. **Convert the lists to NumPy arrays:** If the lists are not already NumPy arrays, you can convert them.
3. **Use NumPy's concatenation function:** You can use `numpy.concatenate` to merge the arrays.

Here's a simple example:

python

Copy code

```
import numpy as np

# Example lists
list1 = [1, 2, 3]
list2 = [4, 5, 6]

# Convert lists to numpy arrays
array1 = np.array(list1)
array2 = np.array(list2)

# Merge the arrays
merged_array = np.concatenate((array1, array2))

print(merged_array)
```

This will output:

csharp

Copy code

```
[1 2 3 4 5 6]
```

Explanation:

- `np.array(list1)` and `np.array(list2)` convert the Python lists into NumPy arrays.
- `np.concatenate((array1, array2))` merges the two arrays into a single NumPy array.

If you want to merge them along a different axis (e.g., stacking them vertically or horizontally), you can use `np.vstack`, `np.hstack`, or similar functions depending on your needs.



Message ChatGPT

