



NATIONAL
ACCELERATOR
LABORATORY



THE UNIVERSITY OF
CHICAGO

Optimization and Machine Learning for Particle Accelerators: organization of the course

Presenter: R. Lehe

Day 1

Instructors:



Auralee Edelen
(SLAC)



Adi Hanuka
(prev. SLAC, now
Eikon Therapeutics)



Remi Lehe
(LBNL)



Christopher Mayes
(SLAC)



Ryan Roussel
(U. Chicago)

Graders:



Jorge Diaz Cruz
(U. New Mexico)



Mauricio Ayllon Unzueta
(U.C. Berkeley)



- Course website:
https://slaclab.github.io/USPAS_ML
- Gathers resources for the course:
 - Agenda
 - Lecture slides
(posted just before each lecture)
 - Lecture recordings
(posted after each lecture)
 - Lab solutions (posted after
corresponding labs)
 - Slack workspace



Overview

This page gathers the class material for the 2021 U.S. Particle Accelerator School course on [Optimization and Machine Learning for Accelerators](#).

Agenda

[Download](#)

Lecture slides and recordings

- Introduction [slides](#)
- Optimization 1: Introduction and local methods
- Optimization 2: More advanced methods
- Introduction to machine learning [slides](#)
- Gaussian processes



Agenda

Time (CST time zone)	Monday June 21	Tuesday June 22	Wednesday June 23	Thursday June 24	Friday June 25
			Return homework from lab 1		Return homework from lab 3
10:00 AM	Lecture 1: Optimization 1	Lecture 2: Optimization 2	Lecture 3: Intro to machine learning	Lecture 4: Gaussian processes	Lecture 5: Bayesian optimization
11:00 AM	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5
12:00 PM	Break	Break	Break	Break	Break
1:00 PM					
2:00 PM	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5

Homework from lab 1

Homework from lab 3

Homework from lab 5

Monday June 28	Tuesday June 29	Wednesday June 30	Thursday July 1	Friday July 2
Return homework from lab 5		Return homework from lab 6		Return homework from lab 8
Lecture 6: Modern neural networks	Lecture 7: Uncertainty quantification	Lecture 8: Unsupervised learning	Lecture 9: Reinforcement learning	Exam
Lab 6	Lab 7	Lab 8	Lab 9	
Break	Break	Break	Break	Break
				Lecture 10: Current challenges
Lab 6	Lab 7	Lab 8	Lab 9	

Homework from lab 6

Homework from lab 8

Connect to Zoom during the colored areas of the planning (see Google Calendar invitations)



Labs and homework

- Labs, homework and exam are in **Jupyter notebook** format.
- We will use RadiaSoft's cloud platform "Sirepo" (www.sirepo.com/jupyter) to:
 - Run the labs in a controlled environment
 - Gather the returned homework
- When connecting to Sirepo for the first time: use **the email address that you provided to USPAS**
- **Blue questions:** done live, during lab sessions
Green questions: homework

Note: When using Sirepo: no GPU access.

For (free) GPU resources, you can run the notebooks on <https://colab.research.google.com/> after this course.
(Not supported during this course.)

lab_01.ipynb Python 3

Gradient Descent on the Accelerator Example

Task: Now apply gradient descent to optimizing the quadrupole strengths to minimize the total final beam size using the above function. Write the corresponding code below. (Feel free to reuse the function `gradient_descent` and to tune `alpha` and `n_iterations` to reach convergence - which you can assess by plotting the returned history ; do not hesitate to use a large value for `alpha`.)

What is the value of the beamsize that is obtained at the end of the optimization?

```
[ ]: K = torch.tensor([1.0, -1.0, 0.5], requires_grad = True)
# Your code here
```

Your answer here: (What is the value of the beamsize that is obtained at the end of the optimization?)

Gradient descent with numerical differentiation

Homework: Try to minimize again the beam size, but using numerical differentiation instead of autograd. In this case, you will **not** call the `.backward` and `.grad` functions, but instead you will need to calculate the gradient numerically: vary in the input `K` by `h=1.e-4` in each direction, in order to compute the gradient of `beam_size`.

(Note that, because we do not call `.backward`, we also do not need to set `requires_grad = True`.)

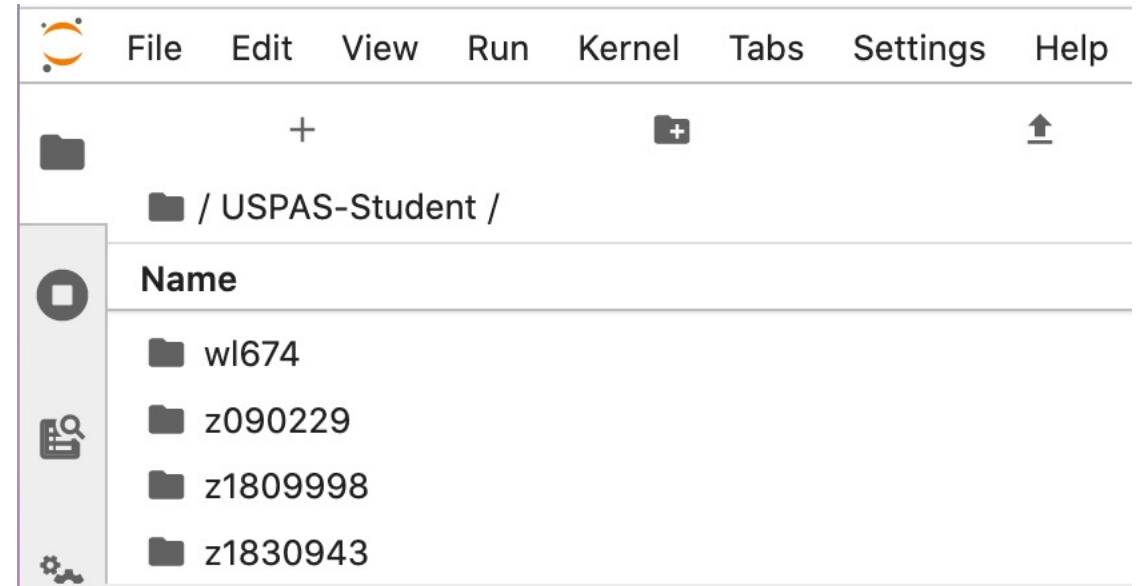
Does the algorithm reach the same final value, for the beamsize?

```
[ ]: def gradient_descent_nd(X0, function, alpha=0.1, n_iterations=100):
    """
    Performs n_iterations iterations of gradient descent,
```



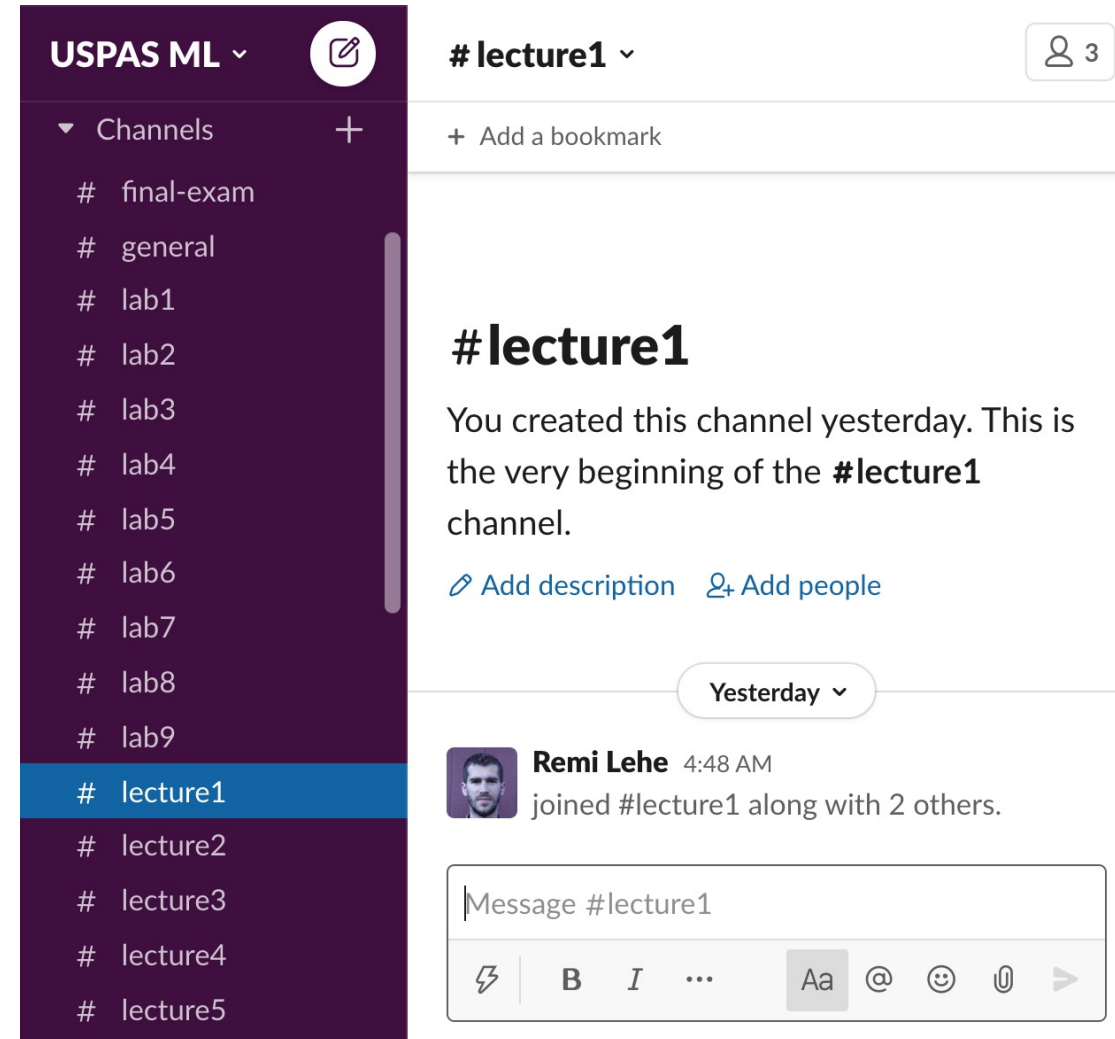
Grading

- Homework and final exam will be graded.
Overall grade = 60% homework + 40% exam
- Need A or B to pass this course and get academic credits
- When done with your homework/exam, **copy** your notebook to the folder that corresponds to your **email address**, within the USPAS–Student folder, on www.sirepo.com/jupyter
- **Audit students:** no need to return homework or participate in exam.



- Lectures, labs and exam will take place through Zoom.
Lectures will be **recorded** and later posted to the website.
- Having **cameras on** is encouraged!
Mute unless you are specifically asked to unmute (e.g. for questions)
- **Feel free to ask questions at any time during lab/lecture!**
by either:
 - Raising your hand in Zoom
 - Or typing the question in the chatSomeone will be monitoring the chat/raised hand and will warn the speaker.
- For questions outside of the lab/lecture sessions, use Slack (next slide)
- Reminder: do not post Zoom link publicly ; link is only for registered students.

- Our slack workspace: uspas-ml.slack.com
- You should have been invited last week. If not: let us know **now** via Zoom chat (along with your email address)!
- Purpose:
 - interaction outside of lecture/lab hours
 - any questions (on course content or organization)
 - one-on-one help (esp. debugging your environment)
 - sharing interesting resources, etc.



The screenshot shows a Slack interface. On the left is a dark sidebar for the workspace 'USPAS ML'. It lists several channels: # final-exam, # general, # lab1 through # lab9, # lecture1 (highlighted in blue), # lecture2 through # lecture5. On the right is the main view for the '# lecture1' channel. At the top right of this view, it says '# lecture1' and '3' people are in the channel. Below that is a '+ Add a bookmark' button. The main content area shows the channel name '#lecture1' in large bold text, followed by a message: 'You created this channel yesterday. This is the very beginning of the #lecture1 channel.' Below the message are links for 'Add description' and 'Add people'. A separator bar indicates the message is from 'Yesterday'. The message itself is from 'Remi Lehe' at '4:48 AM', stating 'joined #lecture1 along with 2 others.' At the bottom is a text input field with the placeholder 'Message #lecture1' and a rich text toolbar with icons for link, bold, italic, font size, mention, emoji, and upload.



Questions?

Any question at this point?