

High Performance Python Lab

Lecture 2

Skoltech

ZHOIRES
SUPERCOMPUTER
HIGH PERFORMANCE COMPUTING AND BIG DATA
SKOLTECH CDISE

Prof. Sergey Rykovanov
Head of the HPC & Big Data Lab
CDISE Skoltech
Term 2 / 2020

Course instructor



Associate Professor
Sergey Rykovanov

2000-2006 Lomonosov Moscow State University, Physics Faculty

2006-2009 PhD from Ludwig-Maximilians University Munich, Germany (Theoretical and computational plasma physics)

2009-2012 Postdoc at Ludwig-Maximilians University Munich

2012-2014 Postdoc at Lawrence Berkeley National Laboratory USA

2014-2018 Group Leader at Helmholtz Institute Jena, Germany

July 2018- ... Skoltech

In Skoltech:

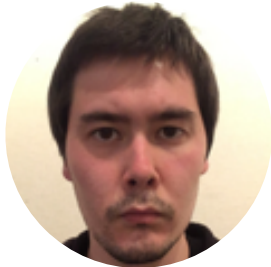
Head of the AI&Supercomputing laboratory (AIS lab)

Head of the HPC track within the CDISE IST program

Instructor for:

- High Performance Computing and Modern Architectures (6 credits)
- High Performance Python Lab (3 credits)
- Parallel Computing in Mathematical Modeling and Data-intensive applications (3 credits)

Course Teaching Assistant (TA)



2012-2016 Higher School of Economics, BSc.

2016-2018 Moscow State University, MSc.

2019-now Skoltech (CDISE), PhD student

PhD student
Rishat Zagidullin



2014-2018 Moscow State University, BSc

2020-now Skoltech, MSc student

since 2021 Intern at Intel

MSc-2 student
Elmira Volkova

Course Teaching Assistants (TAs)



2014-2018 Moscow Institute of Physics and Technology, BSc.
2018-2020 Skoltech, MSc.
2020-now Skoltech (CDISE), Junior Research Engineer

Junior Research
Engineer
Anna Anikina



2015-2019 Moscow Institute of Physics and Technology, BSc.
2019-2021 Skoltech CDISE, MSc.
2021-now Skoltech CDISE, PhD

PhD Student
Egor Konyagin

Survey and “Zhores” account (2-5 minutes)

Please fill out survey:

<https://bit.ly/35tSKUR>

You will need an account on “Zhores”, please fill in the form:

<https://bit.ly/3ksvXir>

**Please join Telegram channel for lectures and for HW discussion
(link in Zoom chat)**




```
this.state = {  
  page_type: !1  
}, this.views = [], this.overlays = [];  
var t = this;  
this.pages = {}, this.app_settings = {}, this.dynamic  
return {  
  name: "",  
  rendered: !1,  
  initialized: !1,  
  filters_options: [],  
  filters: []  
}
```

Python bootcamp continued

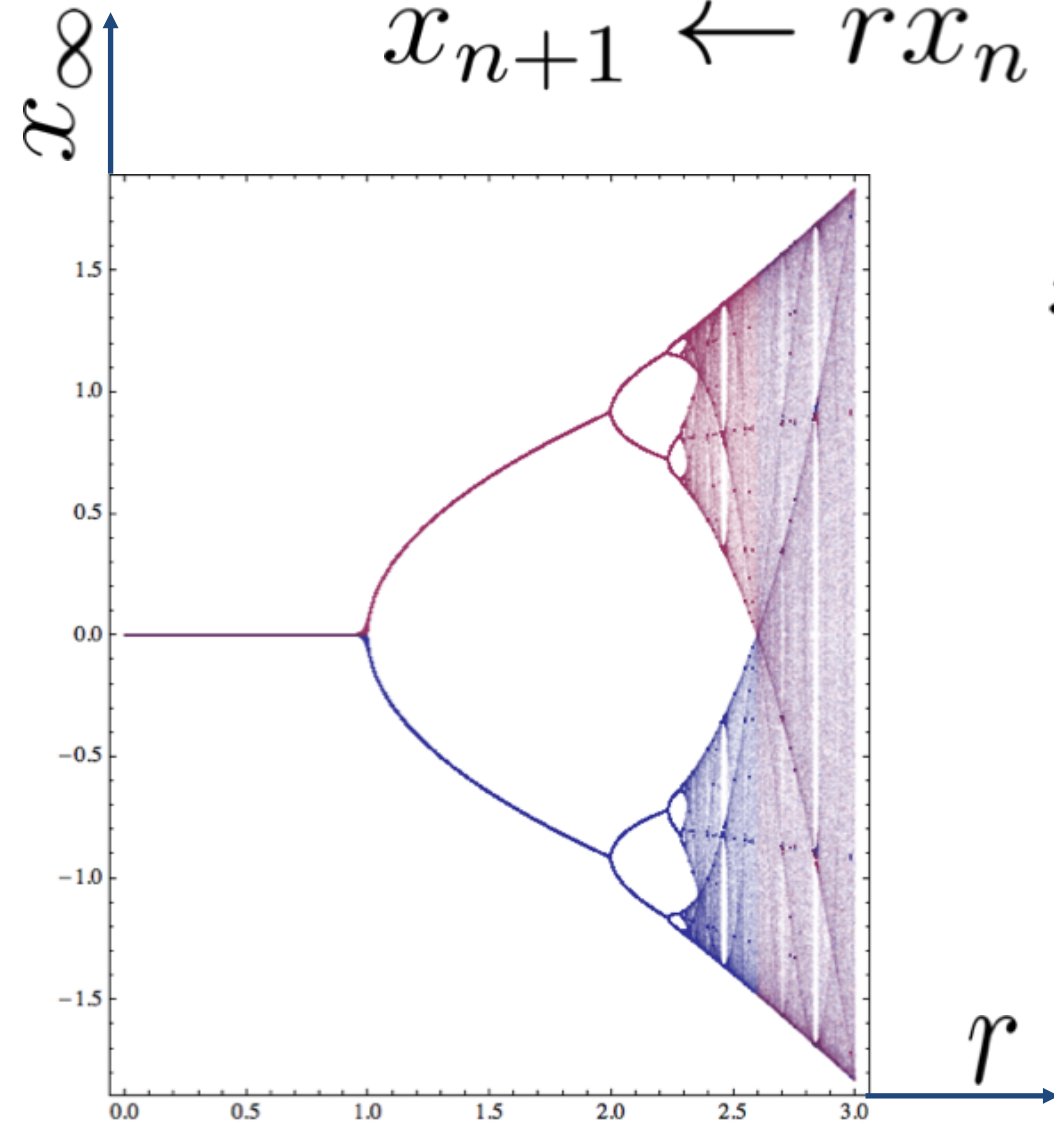
```
}, this.inl = function(a) {  
  var r = this;  
  r.app_settings = $.extend(  
    api_url: "",  
    ROOT_: "",  
    trigger: "#13251231",  
    "#13251231",  
    "#13251231",  
    "#13251231"
```

Task 1. Bifurcation map

$$x_{n+1} \leftarrow r x_n (1 - x_n)$$

$$x_0 \leftarrow \text{rand}(0, 1)$$

$$r \leftarrow \text{const}$$



Task 1. Bifurcation map

Step-by-step guide:

- implement the map, plot the evolution of x
- play around with values of r , see the change of evolution
- then create a linspace of r 's, for every r save last “ m ” values of x after first “ n ” values (can be $m=200$, $n=200$), play around with values
- Get the bifurcation map
- visualize the evolution (play around)

$$x_{n+1} \leftarrow r x_n (1 - x_n)$$

$$x_0 \leftarrow \text{rand}(0, 1)$$

$$r \leftarrow \text{const}$$

if you are interested - describe real physical systems that exhibit bifurcation

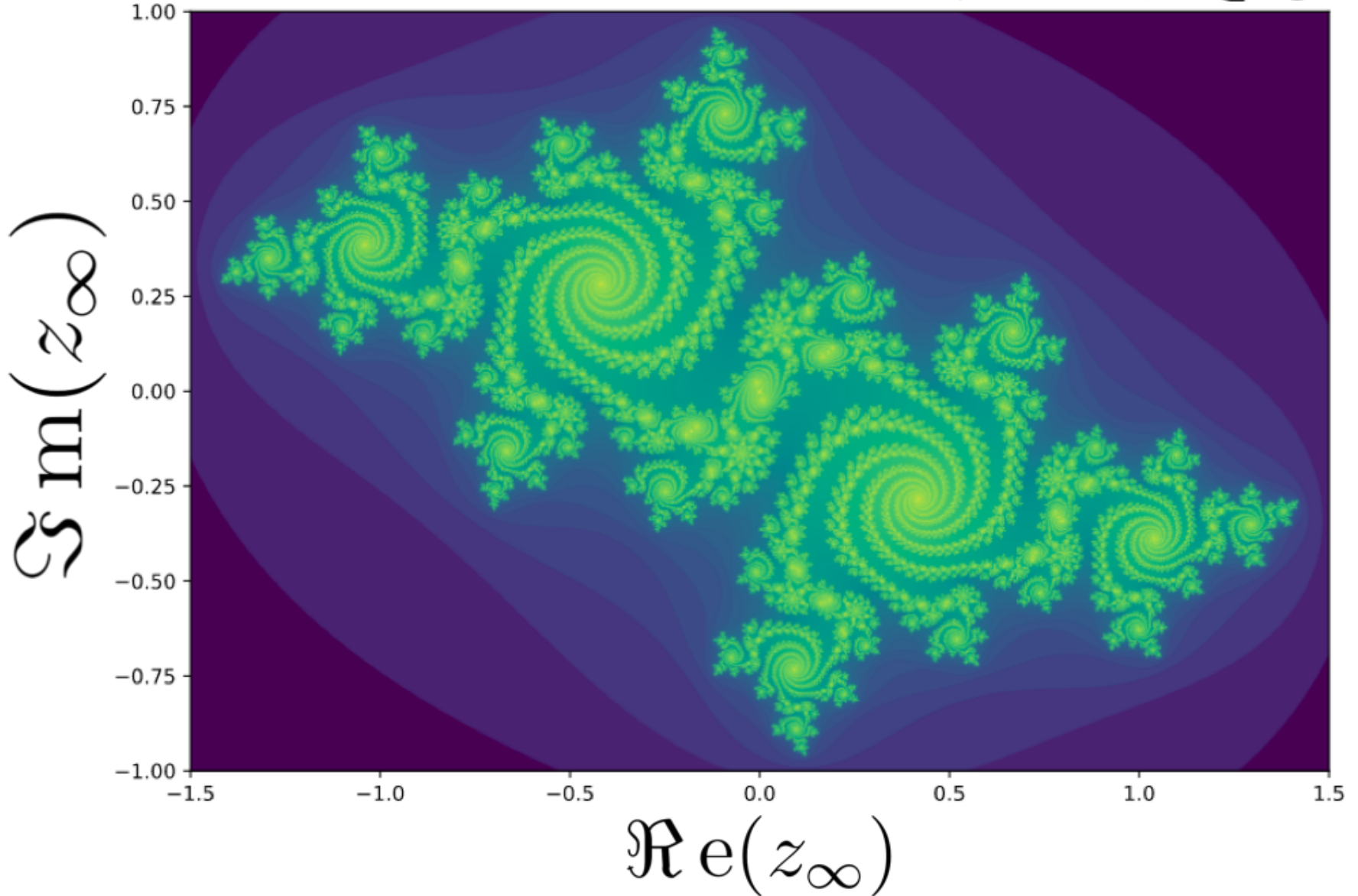
Grading for task 1 (Bifurcations)

Subtask	Score
1.Implement the map, plot the evolution of x	1
2.Create a linspace of r's, for every r save the last "m" values of x after the first "n" values (can be m=200, x=200), play around with values	1
3.Plot the bifurcation map	1
4.Parallel computation of bifurcation map	2
5.Plot speedup vs number of processes	2

Task 2. Julia set

$$z_{n+1} \leftarrow z_n^2 + c$$

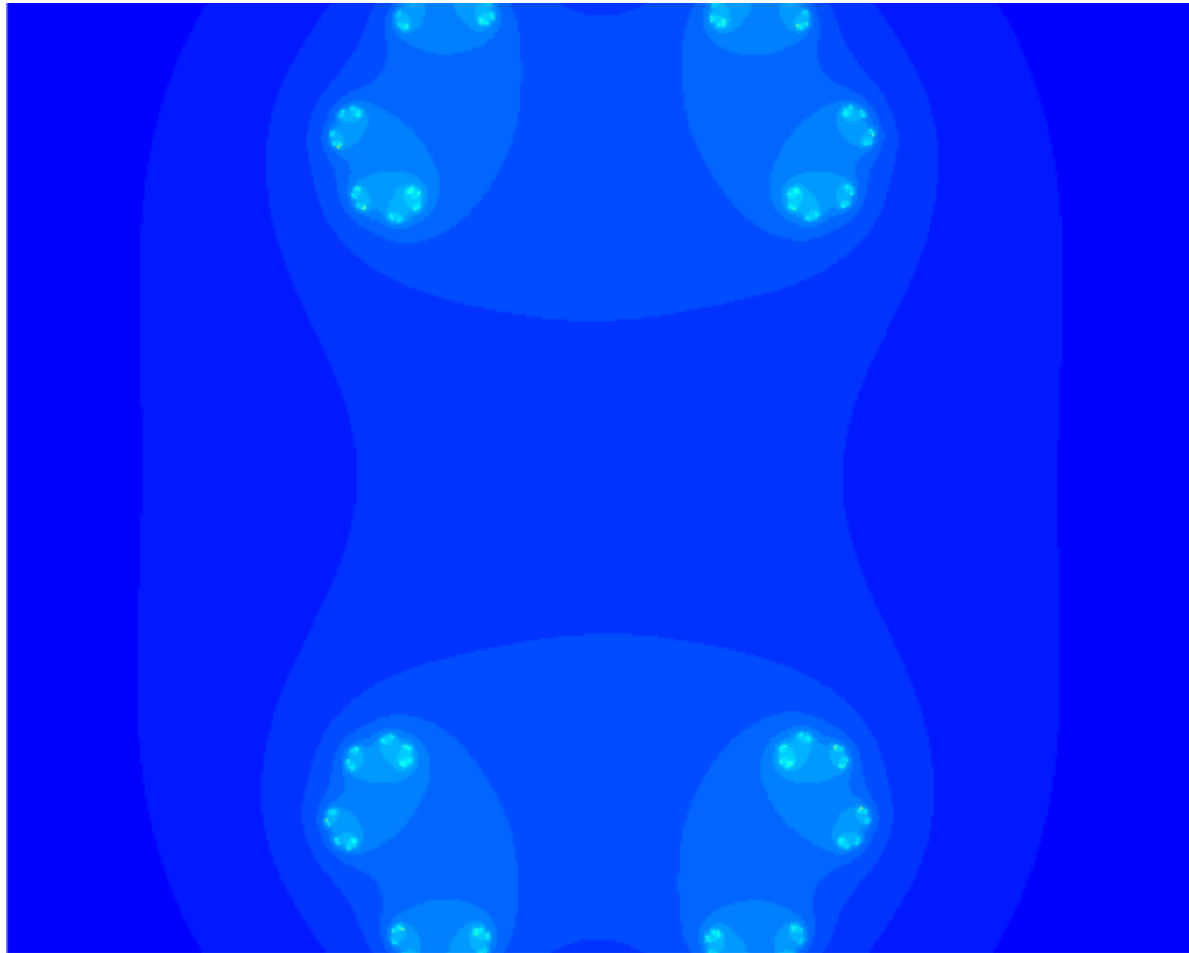
$$c \leftarrow \text{const} \in \mathbb{C}$$



Task 2. Julia set – 2

$$c \leftarrow 0.7885 e^{ia} \quad a \leftarrow \text{range}(0, 2\pi)$$

$\Im m(z_\infty)$



$\Re e(z_\infty)$

Task 2. Julia set

Step-by-step guide:

$$z_{n+1} \leftarrow z_n^2 + c$$

- fix a value of C (can be 0)
- implement the map, plot the evolution of z (Re(z), Im(z))
- play around with values of z₀, see the change of evolution
- for a given z₀ if the sequence converges use black color, if it exponentially diverges use white color, if it starts jumping between “n” values use different colors
- plot the Julia set
- C=0 - Mandelbrot set

Mandelbrot set is $z_n = 0$, c - arbitrary

Grading for task 2 (Fractals)

Subtask	Score
1.Black and white colors of pixels are correct	1
2.Different colors for bifurcation points	1
3.Generate figure of Julia set ($c = 1-r$) where r is the golden ratio. Label the axes ($\text{Re}(z_0)$, $\text{Im}(z_0)$), fontsize should be 20, figsize = (14,11)	2
4.Plot figures for $c=\exp(ia)$, $a = \text{range}(0,2\pi)$ & write down axes like in subtask 3, create animation of these figures slowly changing the a - value. hint: use, e. g., one of these examples: https://stackoverflow.com/questions/753190/programmatically-generate-video-or-animated-gif-in-python & title should include values of a & gif should have longitude 1 minute (<i>max score - 3</i>)	3

Task 3. Schelling's Model

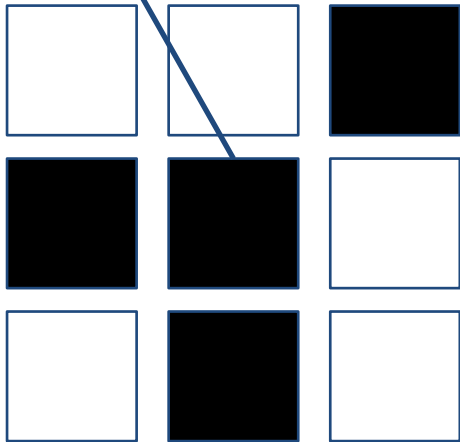
- 1) Suppose there are two types of agents: X and O. Two populations of the two agent types are initially placed into random locations of a neighborhood represented by a grid. After placing all the agents in the grid, each cell is either occupied by an agent or is empty.
- 2) Now we must determine if each agent is satisfied with its current location. A satisfied agent is one that is surrounded by at least t percent of agents that are like itself. This threshold t is one that will apply to all agents in the model.
- 3) When an agent is not satisfied, it can be moved to any vacant location in the grid. Any algorithm can be used to choose this new location. For example, a randomly selected cell may be chosen, or the agent could move to the nearest available location.
- 4) All dissatisfied agents must be moved in the same *round*. After the round is complete, a new round begins, and dissatisfied agents are once again moved to new locations in the grid.

Task 3. Schelling's Model

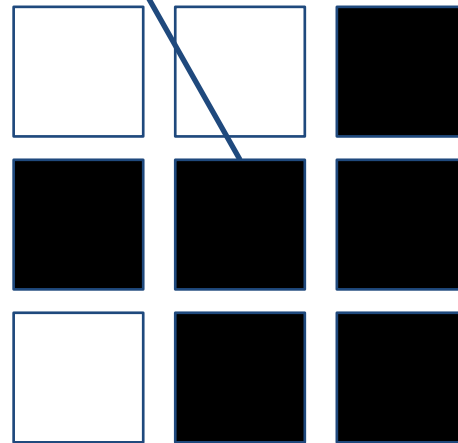
Segregation model (black and white households)

- map is large (i.e. 100x100 with periodic boundaries)
- each household on a map has 8 neighbours
- the agent decides to randomly move if less than R (you can change that) neighbours are of the same color

I want to move



I stay



Task 3. Schelling's Model

Segregation model (black and white households)

- map is large (i.e. 100x100 with periodic boundaries)
- each household on a map has 8 neighbours
- the agent decides to randomly move if less than $R \cdot 8$ (you can change that) neighbours are of the same color

Steps:

Randomize the map with half white/half black

define the value of R (0, $\frac{1}{8}$, $\frac{2}{8}$, $\frac{3}{8}$, $\frac{4}{8}$, $\frac{5}{8}$, $\frac{6}{8}$, $\frac{7}{8}$, 1)

start the game

on each step for each cell find out if the cell wants to move - this cell is now on the market (considered free for moving in)

after you finished with the whole map - cells that want to move can move into free cells

repeat the whole procedure

Grading of task 3 (Schelling model)

Subtask	Score
1.Create 9 gifs of map evolution for 9 values of R	5
2.Plot number of households that want to move versus time for 9 values of R on one graph, label 9 curves, label the axes and title the graph	2

Task 4. Spectrogram

<https://bit.ly/3kGqCEn>

Grading for Task 4 (Spectrogram)

Subtask	Score
1.Add 4th wave packet (frequency = 4 and time_shift = 7 cycles). Demonstrate the effect on the plot of the FFT spectrum	1
2.Implement the spectrogram, show the effect of 1 on the spectrogram. Don't forget to label the axes	2
3.Change the number of time steps in your signal to the power of 2 (i.e. 2^{14}) and then slightly change the number of timesteps. Measure the timing, can you explain the difference?	2
4.Parallel version of spectrogram implemented	2
5.Plot speedup vs the number of processors	2