

# Jupyter notebooks

Vojtěch Nikl

---

**Seminář gridového počítání 2023**  
**MetaCentrum**

- Skupina webových aplikací/IDE umožňující intraktivně vytvářet a sdílet zdrojové kódy a dokumenty v mnoha jazycích
- Interpretovaný kód, matematika, vizualizace, simulace, ...
- Samostatný vývoj i výuka
- Cloud comp. Microsoft Azure, Google Colab, Amazon's SageMaker, ...



- Server běžící na výpočetním uzlu v Metacentru
- Prostředí pro samotný vývoj
- Zdrojové soubory `.ipynb` (kód, popis, obrázky, grafy, koment.)
- **JupyterHub**
  - Přihlašování přes Kerberos
  - Spawnování notebooků na PBS uzly (s uživatelským kerberos ticketem)
  - Běží v Kubernetes
  - <https://jupyter.cloud.metacentrum.cz>
- [https://wiki.metacentrum.cz/wiki/Jupyter\\_for\\_MetaCentrum\\_users](https://wiki.metacentrum.cz/wiki/Jupyter_for_MetaCentrum_users)
- **OnDemand**





1

Příprava `.jupyter_bashrc`

Lokální knihovny

4

Files Running Clusters

Select items to perform actions on them.

- 0 / notebooks
- ..
- src
- cpp17-test.ipynb
- mandelbrot.ipynb
- Untitled.ipynb

2

Sign in

Username:

Password:

Sign In

5

```
File Edit View Insert Cell Kernel Help
+ %< > < > < > < > Run C < > Code
In [2]: import numpy as np
        from pylab import imshow, show
        from timeit import default_timer as timer

        def mandel(x, y, max_iters):
            """
            Given the real and imaginary parts of a complex number,
```

3

## Server Options

Spawn notebook server on a PBS node (recommended) or locally on a cloud

PBS

PBS Queue (aka qsub -q X)

default

PBS Number of Processor Cores (aka qsub -l nprocs=X)

1

PBS Number of GPUs (aka qsub -l ngpus=X, increase only when a gpu queue is selected)

0

PBS Memory (aka qsub -l mem=X)

4gb

PBS Walltime (aka qsub -l walltime=X), use 10 hours max, after that the MetaCentrum kerberos login ticket expires

01:00:00

Path and name of your `.jupyter_bashrc` file (see wiki to properly prepare your jupyter notebooks' environment)

/storage/bmo2/home/\$PBS\_O\_LOGNAME/.jupyter\_bashrc

Directory where PBS saves job's stdout and stderr files (aka qsub -o X -e X)

skirit.ics.muni.cz/tmp

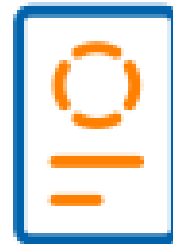
[Optional] Commands executed just before the notebook server starts (equivalent to using `.jupyter_bashrc`)

[Optional] Commands executed just after the notebook server ends unexpectedly

Start

- Kernely – Python, C++ (Cling), R, Go a další
  - <https://gist.github.com/chronitis/682c4e0d9f663e85e3d87e97cd7d1624>
- K dispozici všechny předinstalované knihovny a moduly
- CUDA (python nvcc plugin)
  - Uzly s GPU – fronty `gpu_jupyter` a obecná `gpu`
  - aktivace `%load_ext nvcc_plugin`
  - Použití `%%cu`
- Vytvoření a spuštění binder images
- 1 uživatel – N notebook serverů
- Potencialní problémy
  - Vypršení ticketu po 10 hodinách
  - Zamrznutí při inicializaci – výjimečné, řešení – start nového notebooku

- Oddělená instance JupyterLabu zaměřená na mezinárodní uživatele
  - Pod ní infrastruktura Kubernetes, notebooky v dockeru
  - Určené pro větší skupiny, mezinárodní spolupráce apod.
- Vč. podpory pro Binder (EGI Replay)
- Integrace s prostředím EGI, potažmo EOSC
  - AAI (EGI Check-in)
  - Úložiště a datové zdroje
    - CVMFS (distribuce větších SW balíčků)
    - EGI DataHub (OneData)
    - Dirac
    - Federace zdrojů družicových dat (C-SCALE – Copernicus)
    - B2DROP (Sync'n'share řešení EUDAT)
    - vznikají další ...
- Pro přístup je třeba vyplnit přihlášku



```
def create_fractal(min_x, max_x, min_y, max_y, image, iters):
    height = image.shape[0]
    width = image.shape[1]

    pixel_size_x = (max_x - min_x) / width
    pixel_size_y = (max_y - min_y) / height

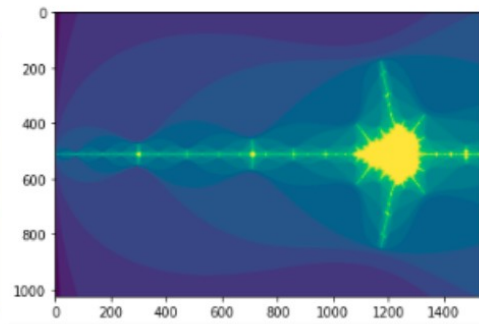
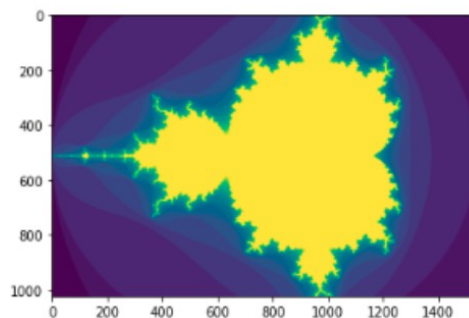
    for x in range(width):
        real = min_x + x * pixel_size_x
        for y in range(height):
            imag = min_y + y * pixel_size_y
            color = mandel(real, imag, iters)
            image[y, x] = color

image = np.zeros((1024, 1536), dtype = np.uint8)
start = timer()
create_fractal(-2.0, 1.0, -1.0, 1.0, image, 20)
dt = timer() - start

print ("Mandelbrot created in %f s" % dt)
imshow(image)
show()

create_fractal(-2.0, -1.7, -0.1, 0.1, image, 20)
imshow(image)
show()
```

mandelbrot created in 6.406227 s





Jupyter Untitled1 Last Checkpoint: a day ago (autosaved)

File Edit View Insert Cell Kernel Help

Run Code

```
In [9]: int a = 5;
        int b = 6;
        int c = a+b;

Out[9]:
```

```
In [10]: printf("%d",c);

11

Out[10]: (int) 2
```

```
In [11]: using namespace std;

Out[11]:
```

```
In [12]: cout << c << endl;

11

Out[12]: (std::basic_ostream<char, std::char_traits<char> >::__ostream_type &) @0x7fb635fc5480
```

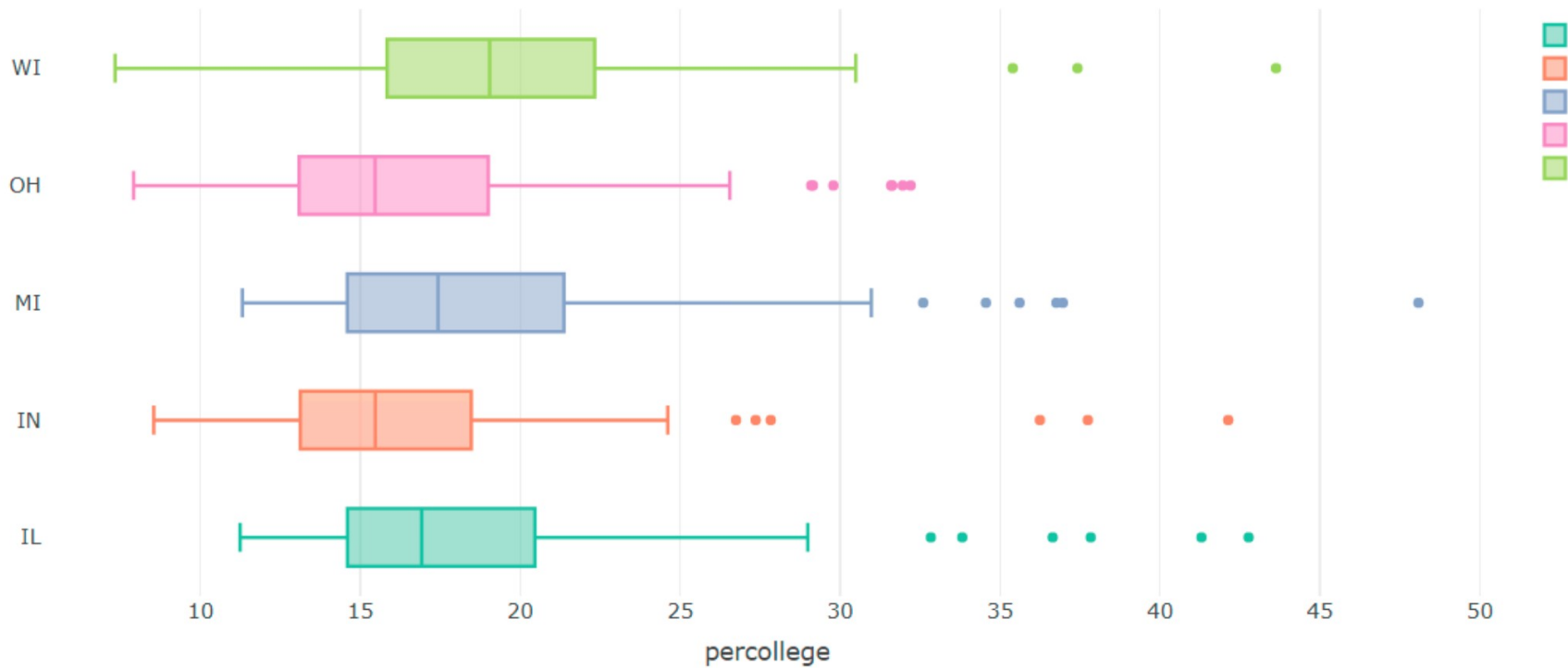
```
In [13]: printf("%d %d %d",c, a ,b);

11 5 6

Out[13]: (int) 6
```



```
In [8]: library(plotly)  
p <- plot_ly(midwest, x = ~percollege, color = ~state, type = "box")  
p
```



jupyter spectrogram (autosaved) Python 3

File Edit View Insert Cell Kernel Help

Markdown CellToolbar

### Simple spectral analysis

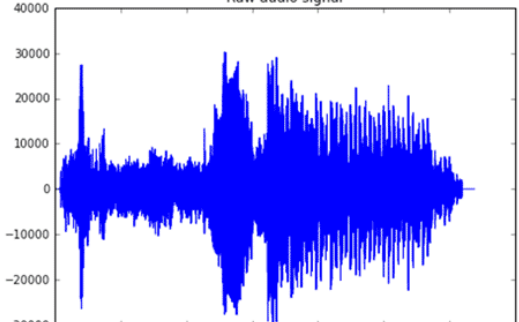
An illustration of the [Discrete Fourier Transform](#)

$$X_k = \sum_{n=0}^{N-1} x_n \exp\left(\frac{-2\pi i}{N} kn\right) \quad k = 0, \dots, N-1$$

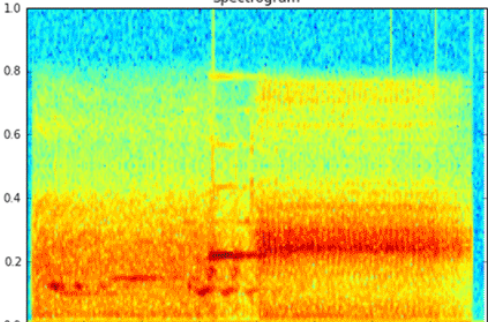
```
In [2]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

```
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```



Raw audio signal



Spectrogram

- Řešení problémů a otázek
  - Nejdříve přečíst FAQ na wiki
  - Primárně dotaz do RT systému [rt.cesnet.cz](http://rt.cesnet.cz) do fronty *meta*
  - Sekundárně na můj email [nikl@cesnet.cz](mailto:nikl@cesnet.cz)
- Pomáháme s přenesením a přípravou prostředí
- **Děkuji za pozornost!**