



*University of Silesia, WP2 & WP4*

---

# Demonstrators, 3d in notebook

**Marcin Kostur**  
Jerzy Łuczka  
Jan Aksamit









# interactive books

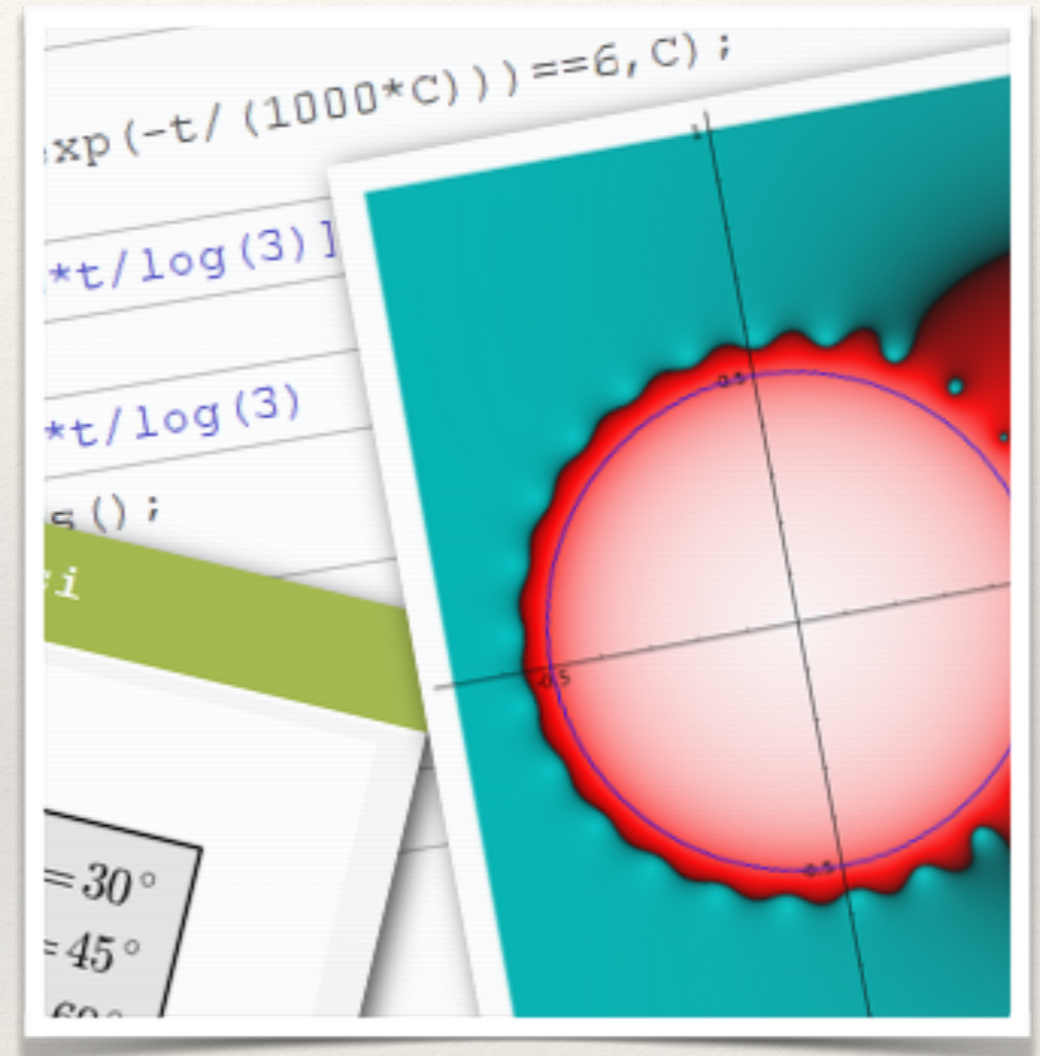
*Task 2.9 Deliverables: 2.8, 2.9, and 2.14*

UŚląski delivers

- Nonlinear Processes in Biology (D2.9)
- Linear Algebra (D2.8)
- Problems in Physics with Sage/Python (D2.14)

SOUTHAMPTON:

Computational Mathematics for  
Engineering (D2.14)





# We will (try to) answer following questions:

- ❖ When is a fully interactive worksheet required and when is a textbook with executable code cells sufficient?
- ❖ How to assemble a classical monograph by reusing independently working building block of text and code?
- ❖ What are best tools and practices for using a single source for producing printed and electronic (interactive) textbooks?
- ❖ How to collaboratively write reusable course material?
- ❖ How can we facilitate automatic testing of all code examples, plots, etc?
- ❖ How can students can benefit from using VRE?

Menu school » previous next

## Parrondo's paradox

**Abstract**

Parrondo's paradox is a combination of games which behaves quite counterintuitively. Beyond its curious behavior, it provides a simple illustration of a physical system, the so-called flashing ratchet. After giving some background information on Parrondo's paradox, we will explore several approaches to a numerical implementation. An object-oriented approach will finally allow us to study in some detail various aspects of Parrondo's paradox. Readers less interested in the programming aspects can skip that part and after having read the introduction can go directly to the last section.

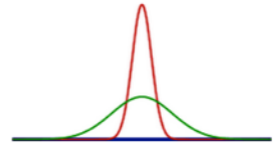
Literature:

- [J. M. R. Parrondo and Luis Dinís, Contemp. Phys. 45, 147 \(2004\)](#)
- [Parrondo's paradox - Wikipedia, The Free Encyclopedia](#)

In Parrondo's paradox, a flashing ratchet is simulated by two games representing the situations with the potential switched on and off. In both cases, the player will loose on average, thereby mimicking a tilted potential with an average motion to the left as explained above. The question to be explored in the following is how the situation changes for the player when one switches between the two games.


Parrondo's paradox serves as an illustration of Brownian ratchets by means of appropriately combining two games where tossing a coin represents the random Brownian motion.

Brownian motion of particles leads to a broadening of their distribution. An initially sharp distribution (red curve) will be significantly broader (green curve) after some time.



If the potential is slightly tilted to the left, the distribution will not only broaden but also move to the left in the course of time.

### What is Parrondo's paradox?





## Tossing a coin

Let us try out the creation of random numbers. The numbers are generated in a way that they are equally distributed between 0 and 1. For the simplest fair game, we would define a threshold of 0.5 where the player wins if the random number is below the threshold and loses otherwise.

```
1 print random()
```

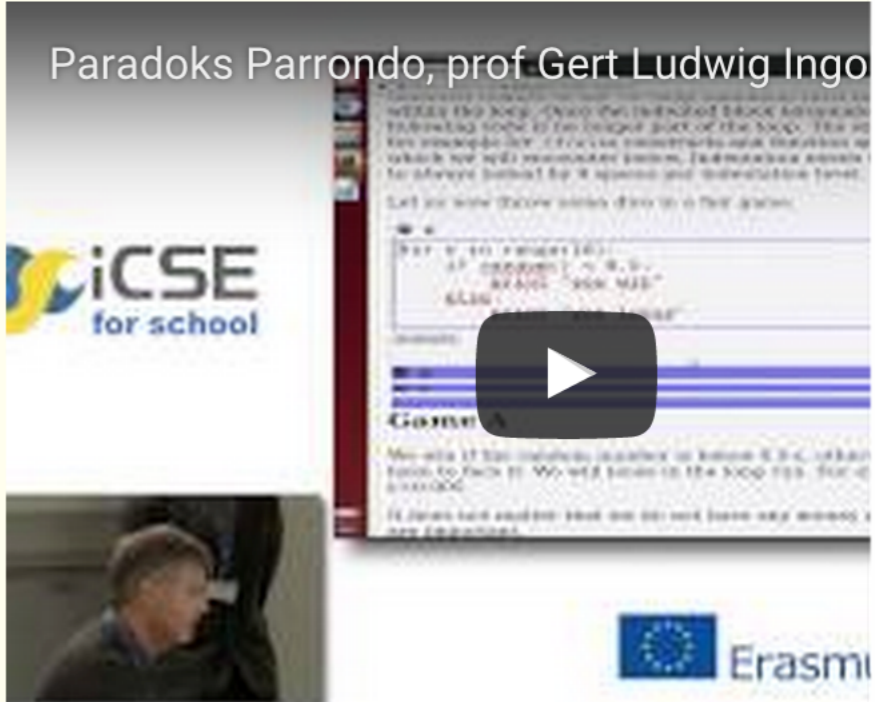
Wykonaj

```
0.306631890114
```

[Help](#) | Powered by SageMath

Let's play a game!

Paradoks Parrondo, prof Gert Ludwig Ingo



ICSE for school

Erasmus

*Experience with structured documents containing code*

# Sphinx & sagecell

<https://github.com/marcinofulus/sphinx-sagecell>



*User interfaces, T4.8, T4.9, D4.12.*

---

## 3d in Jupyter

---

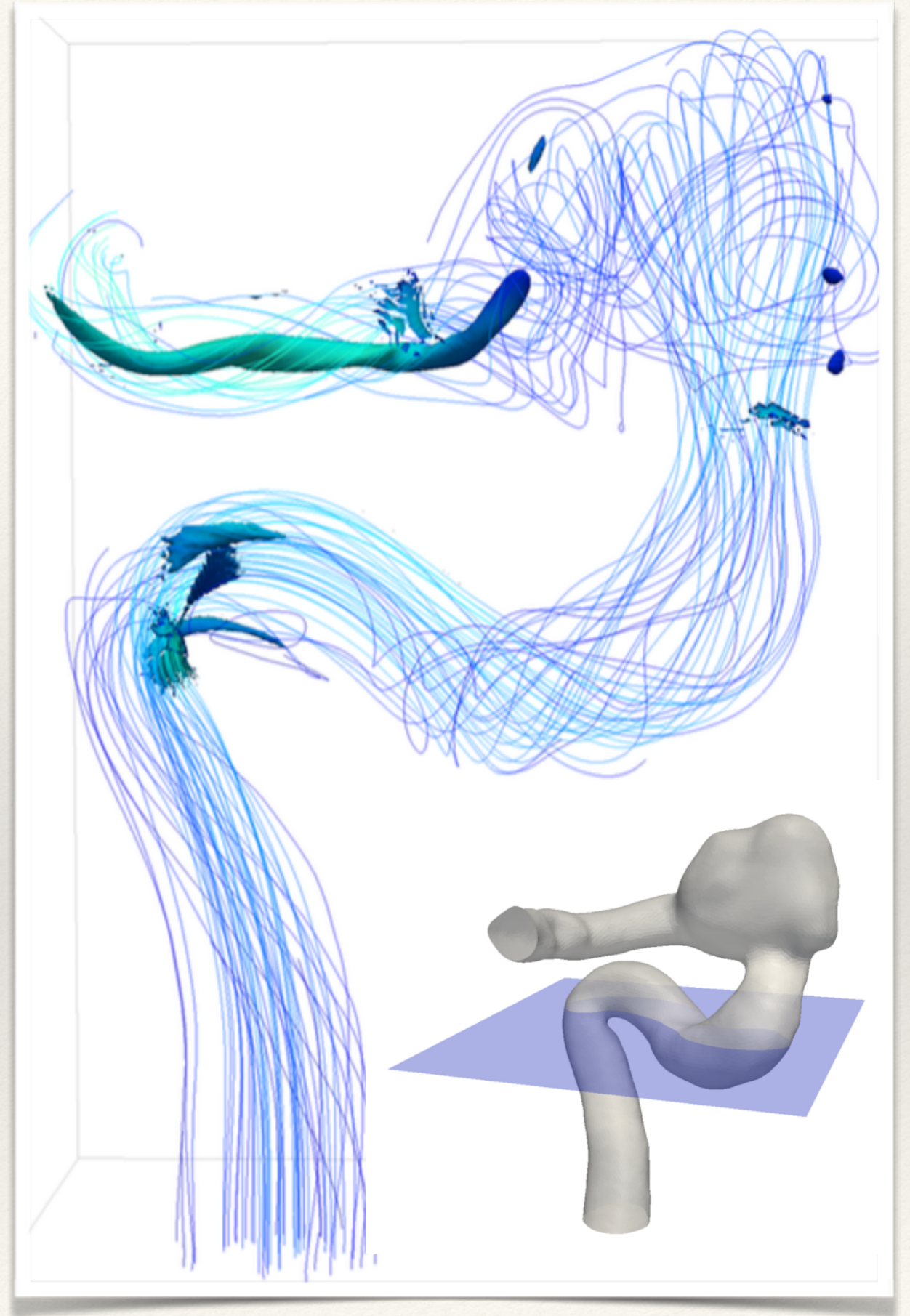
Motivation:

*Work with HPC fluid dynamics in notebook interface across large cluster installations.*

**Sailfish** - Lattice Boltzmann solver,  
natively written for GPU

@UŚląski

- subcontracting
- already have a team of javascript coders





---

# 3d - ongoing effort

---

- ❖ Data types:
- ❖ STL files (STereoLithography)
- ❖ Text labels
- ❖ Vectors
- ❖ Vector fields in 2d (cut-plane of a scene) or 3d (with optional label)
- ❖ Points (with optional label)
- ❖ Lines made of n-segments (with optional label)
- ❖ Texture on rectangle placed in 3d
- ❖ Surface from a function given on a regular mesh
- ❖ Unstructural meshes
- ❖ Voxels, displayed as solid cubes in 3d, based on data as boolean numpy array (3 index)