

# Computer Modelling with Single Prompts (lecture)

Maciej Matyka

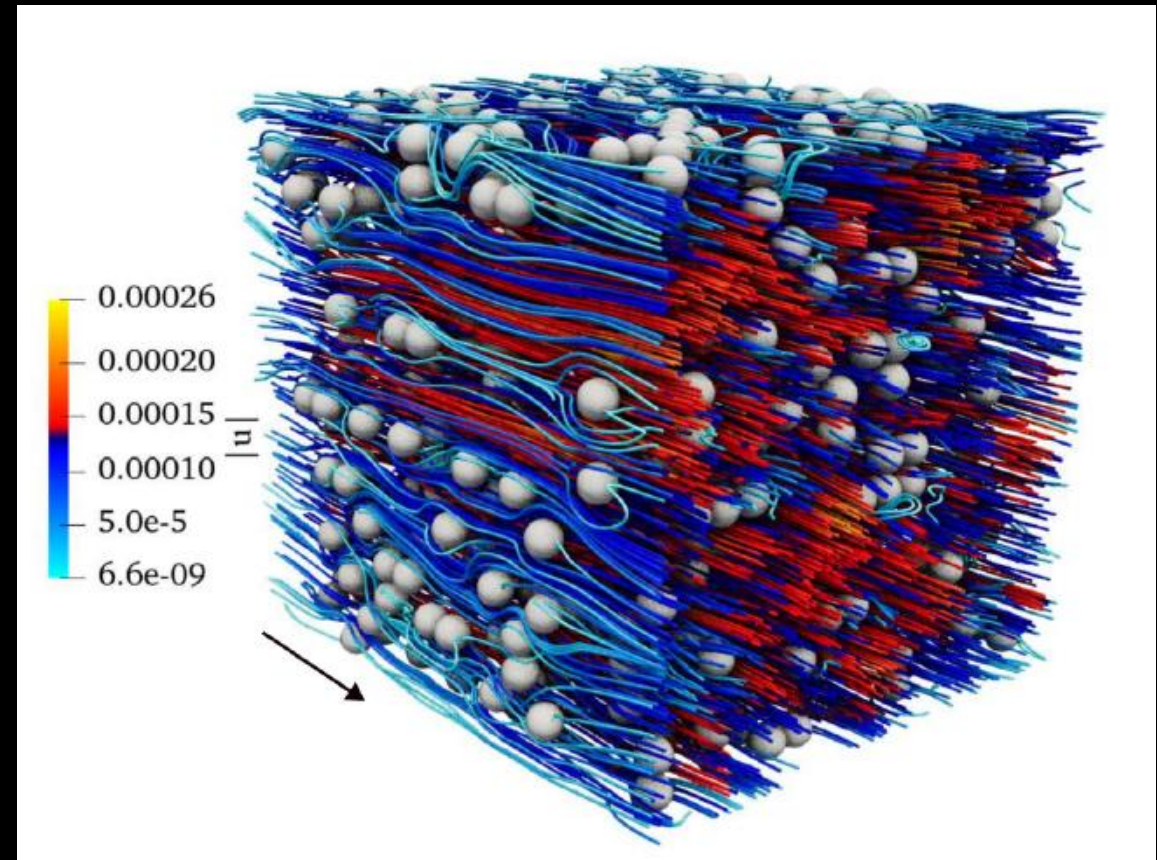
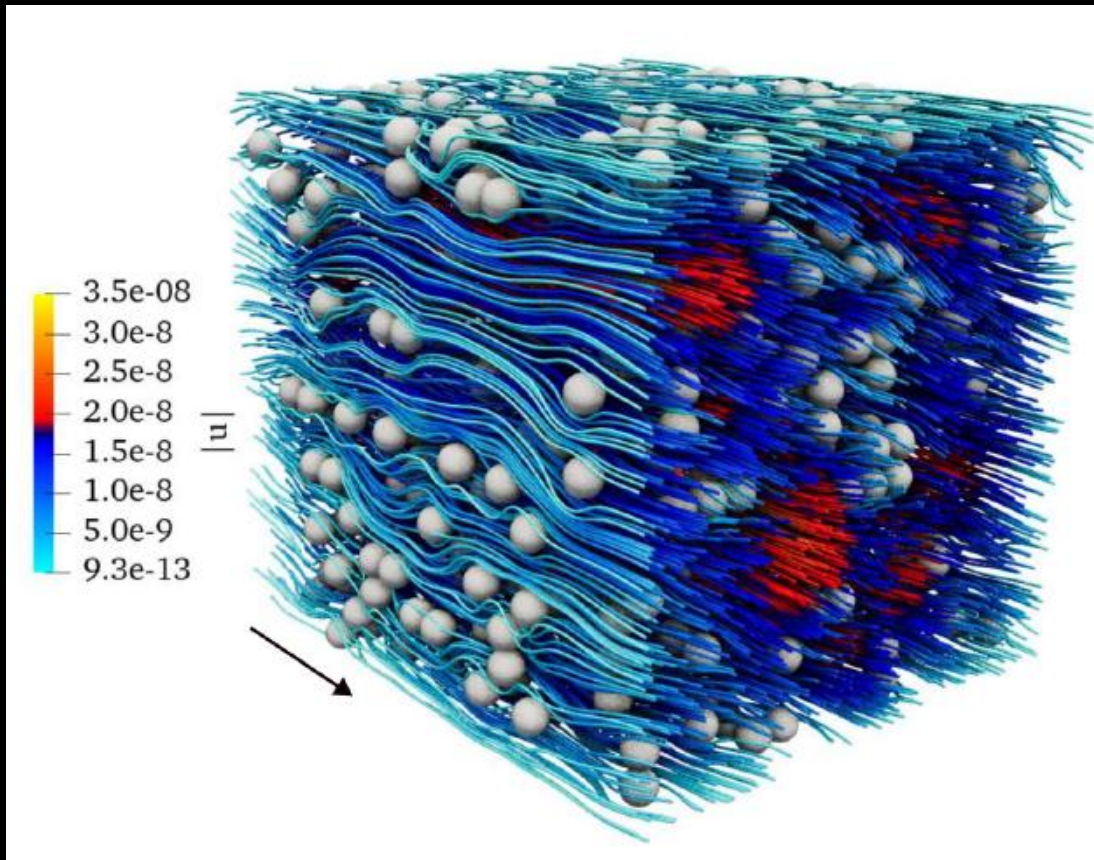
Faculty of Physics and Astronomy, University of Wrocław  
Josef Stefan Institute, Ljubljana

# Faculty of Physics and Astronomy University of Wrocław

- Experimental Physics
- Theoretical Physics
- Computational Physics



# Fluid Flow in porous 3D media



David Potter

**metody  
obliczeniowe fizyki**

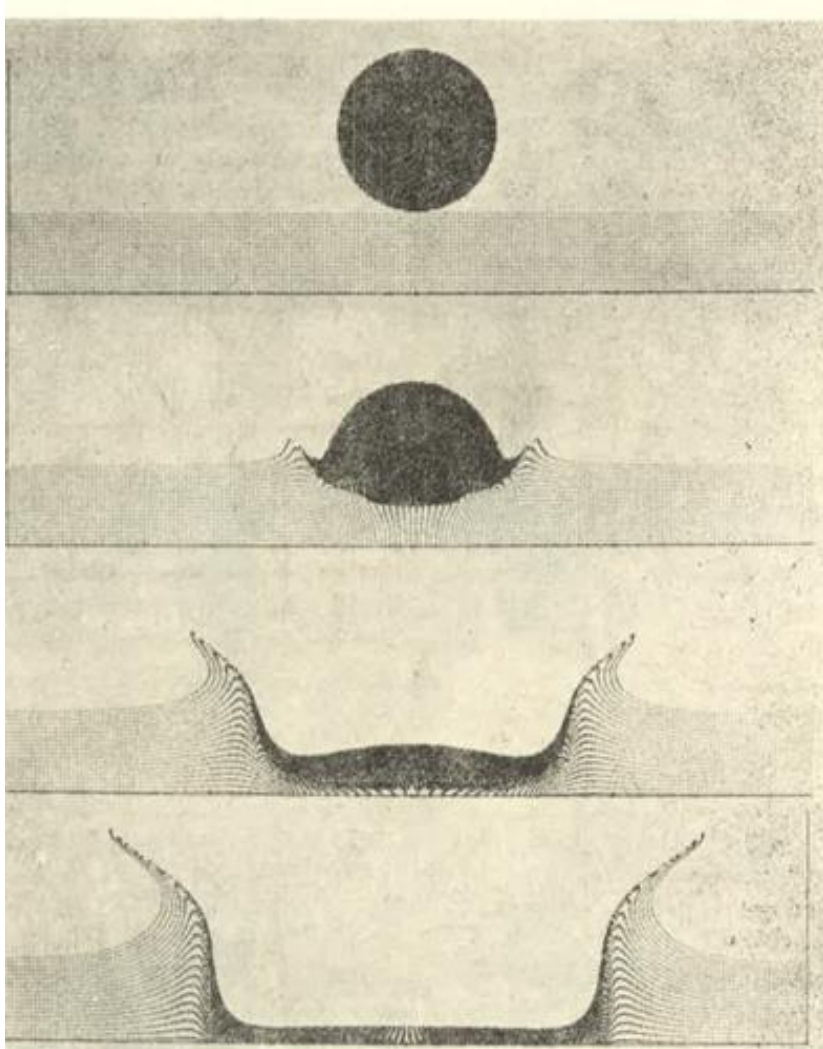
fizyka komputerowa

David Potter  
"Computational  
Physic", PWN 1977

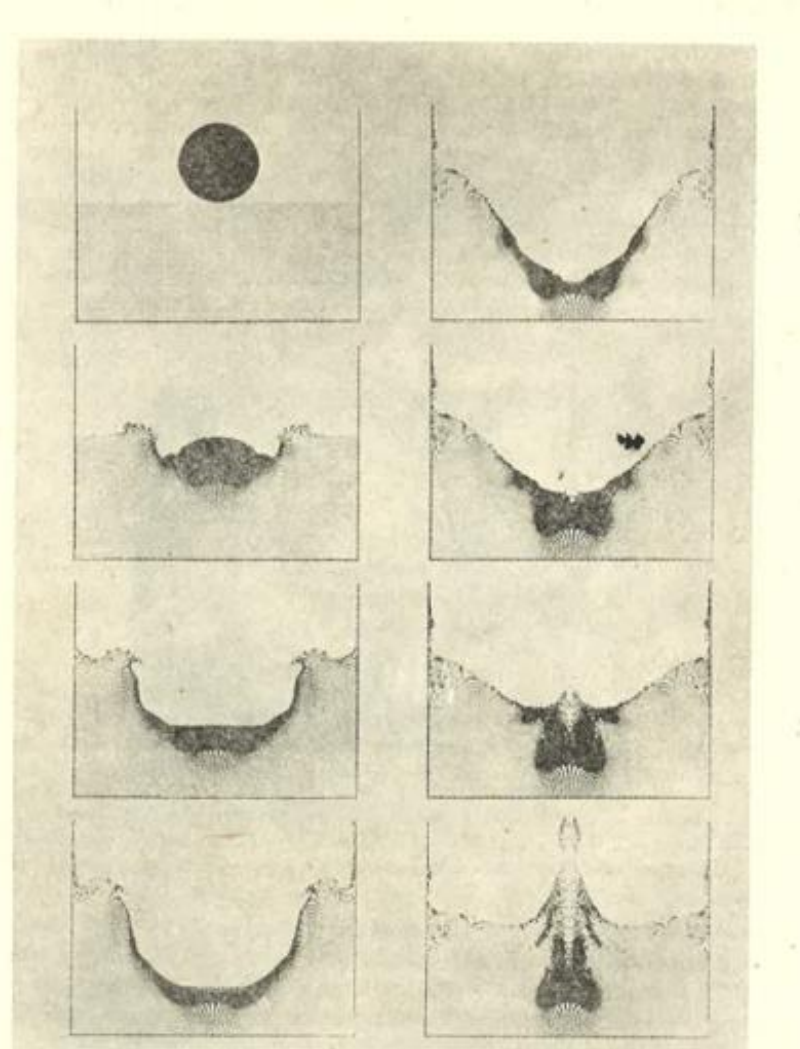
**And I found  
these images**

From 1967,  
Harlow &  
Shannon, The  
Splash of  
Liquid Drop  
paper\*

**"I want to see  
it animated..."**



Rozprysk kropli cieczy spadającej do płytkiego zbiornika. Symulacje przeprowadzono przy użyciu cząstek znaczących w dwóch wymiarach (z pracy: Harlow F. H., Shannon, J. P., Science, 157, 457 (1967)). Copyright 1967 by the American Association for the Advancement of Science).



14. Rozprysk kropli w głębokim zbiorniku z cieczą otrzymany w metodzie cząstek znaczących (z pracy: Harlow, F. H., Shannon, J. P., Science, 157, 457 (1967)). Copyright 1967 by the American Association for the Advancement of Science)

Paper:

[https://uniwroc-my.sharepoint.com/:b:/g/personal/maciej\\_matyka\\_uwr\\_edu\\_pl/EZwNukQ7W7dHuWyL5WrQ1E0B9dqSQ7uil\\_8JQwRPXnxM7A?e=xWzsO3](https://uniwroc-my.sharepoint.com/:b:/g/personal/maciej_matyka_uwr_edu_pl/EZwNukQ7W7dHuWyL5WrQ1E0B9dqSQ7uil_8JQwRPXnxM7A?e=xWzsO3)

# Marker and cell method

- **Particles** on grid (PDE + interface tracking)
- Full, explicit **Navier-Stokes** solver
- Developed at **Los Alamos T4 group**  
(The Physics of Condensed Matter and Complex Systems group)
- The report was not easily accessible (public release at that time)

# The MAC report LANL

Welch, J. E., Harlow, F. H., Shannon, J. P., & Daly, B. J. (1965). *The MAC method-a computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces* (No. LA-3425). Los Alamos Scientific Lab., Univ. of California, N. Mex..

e. 3  
LA-3425

CIC-14 REPORT COLLECTION  
REPRODUCTION  
COPY

LOS ALAMOS SCIENTIFIC LABORATORY  
of the  
University of California  
LOS ALAMOS • NEW MEXICO

## THE MAC METHOD

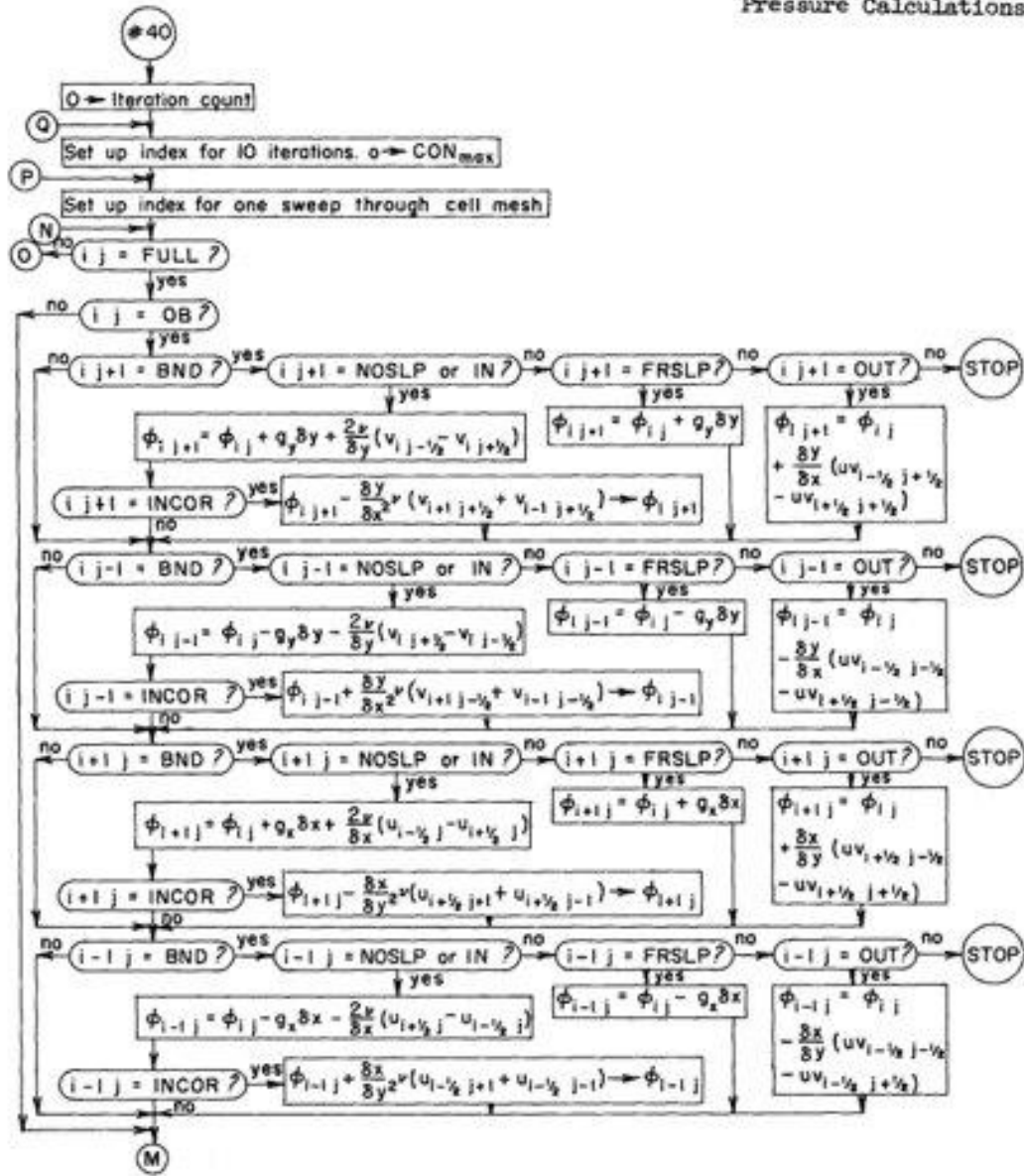
A Computing Technique for Solving Viscous,  
Incompressible, Transient Fluid-Flow Problems  
Involving Free Surfaces



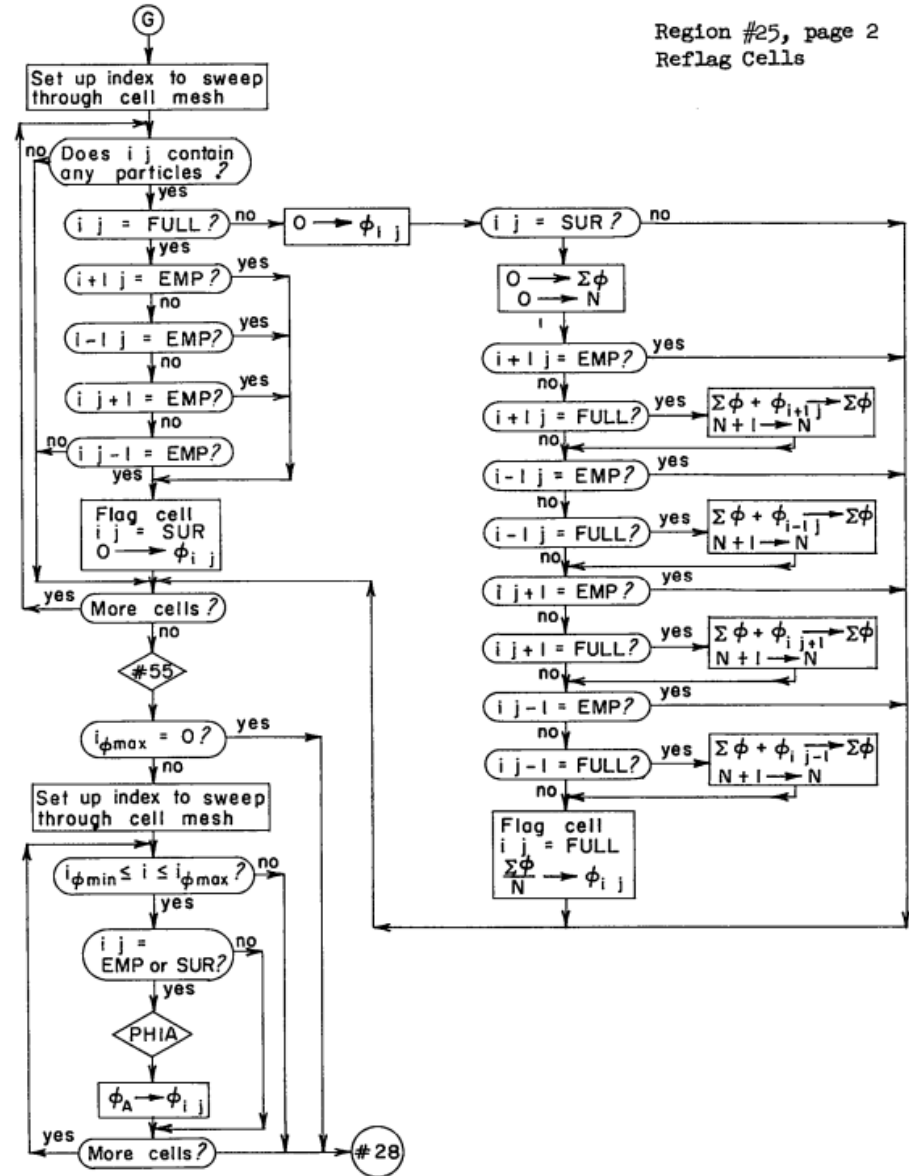
UNITED STATES  
ATOMIC ENERGY COMMISSION  
CONTRACT W-7405-ENG. 36

# This is how the code was released

Region #40, page 1  
Pressure Calculations



Region #25, page 2  
Reflag Cells



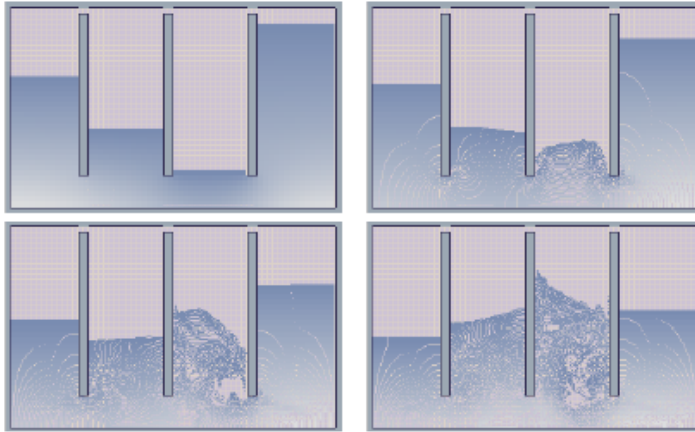
**12 months later...**

(1st year student,  
**thousands of lines of C++**,  
hours in the lab afternoons)

# Fluid v 1.0 – The MAC Method implementation

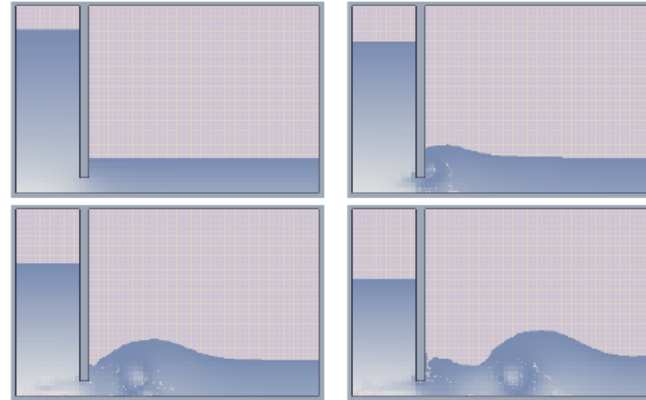
## kłady

nych stron to po kilka klatek z gotowych symulacji. Animacje w komputerze SGI Octane z dwoma procesorami MIPS R10000 300Mhz. JW. Kolory (lub odcienie szarości) pokazują rozkład ciśnienia.

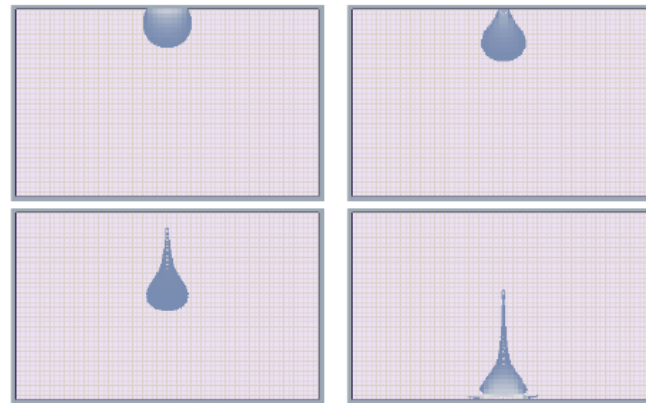


kości słupa wody w poszczególnych zbiornikach wypełnionych cieczą. Grawitacja  $GY = -1.0$ . Kolory pokazują chwilowy rozkład ciśnienia w układzie.

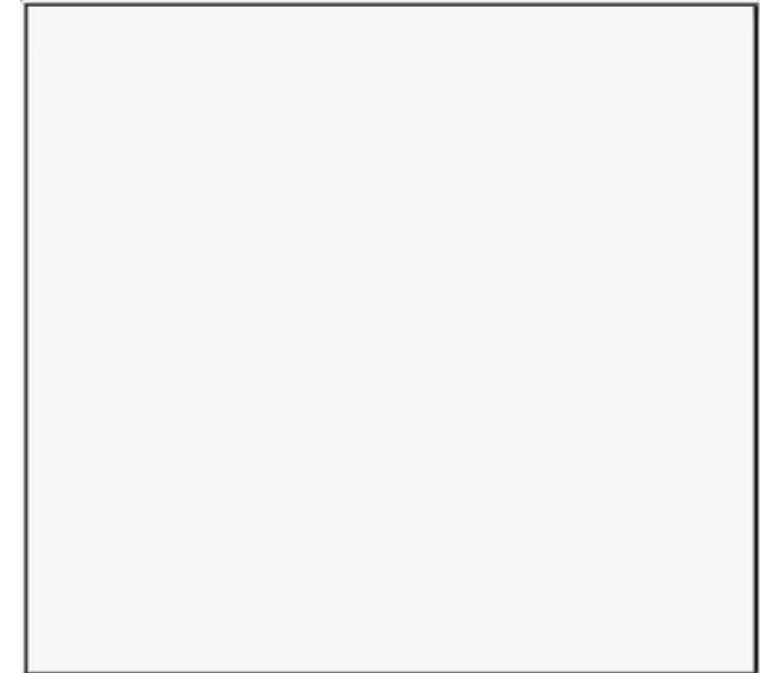
Original fluid C code.



otwierania śluzy. Zapora zwolniona w chwili  $t = 0$ .  $GY = -0.1$ , lepkość



ormowania się kropli. Grawitacja  $GY = -1.0$ , lepkość 0.01, ściany FRSLP. efekt ten uzyskałem przypadkowo, przez własny błąd (ułożenia kuli wodnej w rysunkach widać, że równania Naviera-Stokesa "potrafią" wiele.



Msc: (2 years later - C++ implementation of free-surface flow)

# Why am I here?

- >25 years of experience in programming (**demoscene**)
- **Assembler (MC68020)**, C, C++, Python, JavaScript
- Teaching programming for years
- Book on **Computer Simulations** (2 editions)
- Book "**C++ Combinations**" - for lectures
- Everything went smoothly, then...



nature > news > article

NEWS | 30 January 2025

# How China created AI model DeepSeek and shocked the world

Government policies, generous funding and a pipeline of AI graduates have helped Chinese firms create advanced LLMs.

By [Gemma Conroy](#) & [Smriti Mallapaty](#)



Chinese technology [start-up DeepSeek](#) has taken the tech world by storm with the release of two large language models (LLMs) that rival the performance of the dominant tools developed by US tech giants – but built with a fraction of the cost and computing power.

## Access options

Access through your library

Buy or subscribe

## Related Articles

**Scientists flock to DeepSeek as they're using the blockading model**

**China's cheap, open AI model DeepSeek thrills scientists**

**'In awe': scientists improve ChatGPT model on AI**

**We are witnessing  
paradigm shift in programming:**

Language Models

# Bigram – simple language model

- Build **probability table** from large text
- „What letter is after given letter”
- Step by step prediction:

$$P(„M”|”y”) = 70%$$

$$P(„M”|”e”) = 65%$$

...

$$P(„M”|”x”) = 0.0%$$

Examples:

[1-bigram](#)

[K-bigram](#)

# What makes it Large Language Model (LLM's)?

- Tokenizer
- Neural networks (RNN, LSTM, Transformer)
- Large context, abstraction levels, multi abstraction connections, agents...

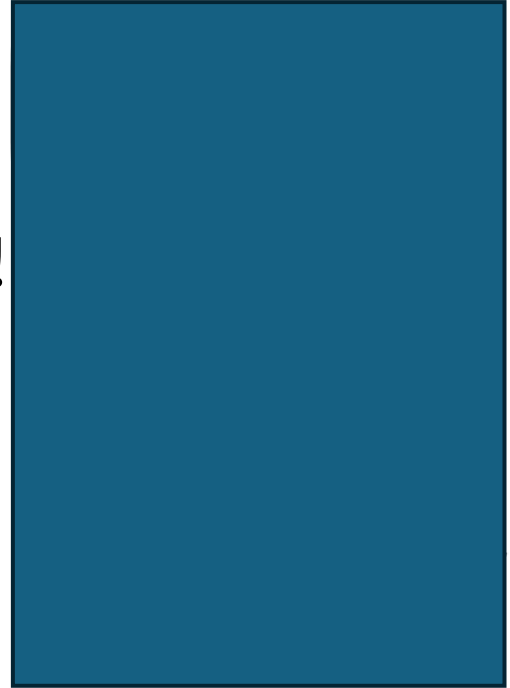
# What is prompting? Game example.

- Look on the picture, remember it.
- Describe it to someone else.
- He/she draws it.
- Compare them.
- This game is super-complex – it just looks simple!



# Why is it hard?

- Imagine you didn't see what you describe.
- Usually we want to describe picture that we don't know!  
(problem)
- **Hallucinations?**



# My first day with DeepSeek

Write a **JavaScript program** that simulates **Poiseuille flow** in a channel **using the LBM method** and presents the results in the form of a velocity field with white where the velocity is highest and black where the velocity is 0. Let the simulation take place **in a periodic system on the x-axis** with **no-slip conditions** on the top and bottom walls. Let gravity be the force driving the flow.

(\*originally in polish)

# Deep Seek

- A process of "deep thinking" has been introduced.

<https://chat.deepseek.com/a/chat/s/7d6d12bc-13db-4ed4-936f-037e4e46d468>

LBM - Krętość

Napisz program w JavaScript, który symuluje metodą LBM Poiseuille flow w kanale i prezentuje wyniki w postaci pola prędkości z kolorem białym tam, gdzie prędkość jest największa i kolorem czarnym tam, gdzie prędkość jest równa 0. Niech symulacja odbywa się w układzie periodycznym w osi x z warunkami bez poślizgu na ścianach górnej i dolnej. Niech siłą wymuszającą przepływ będzie grawitacja.

Thought for 160 seconds

- Okay, I need to write a JavaScript program that simulates Poiseuille flow using the Lattice Boltzmann Method (LBM). The results should show a velocity field with white where velocity is highest and black where it's zero. The simulation should have periodic boundaries in the x-direction and no-slip conditions on the top and bottom walls. The driving force is gravity.

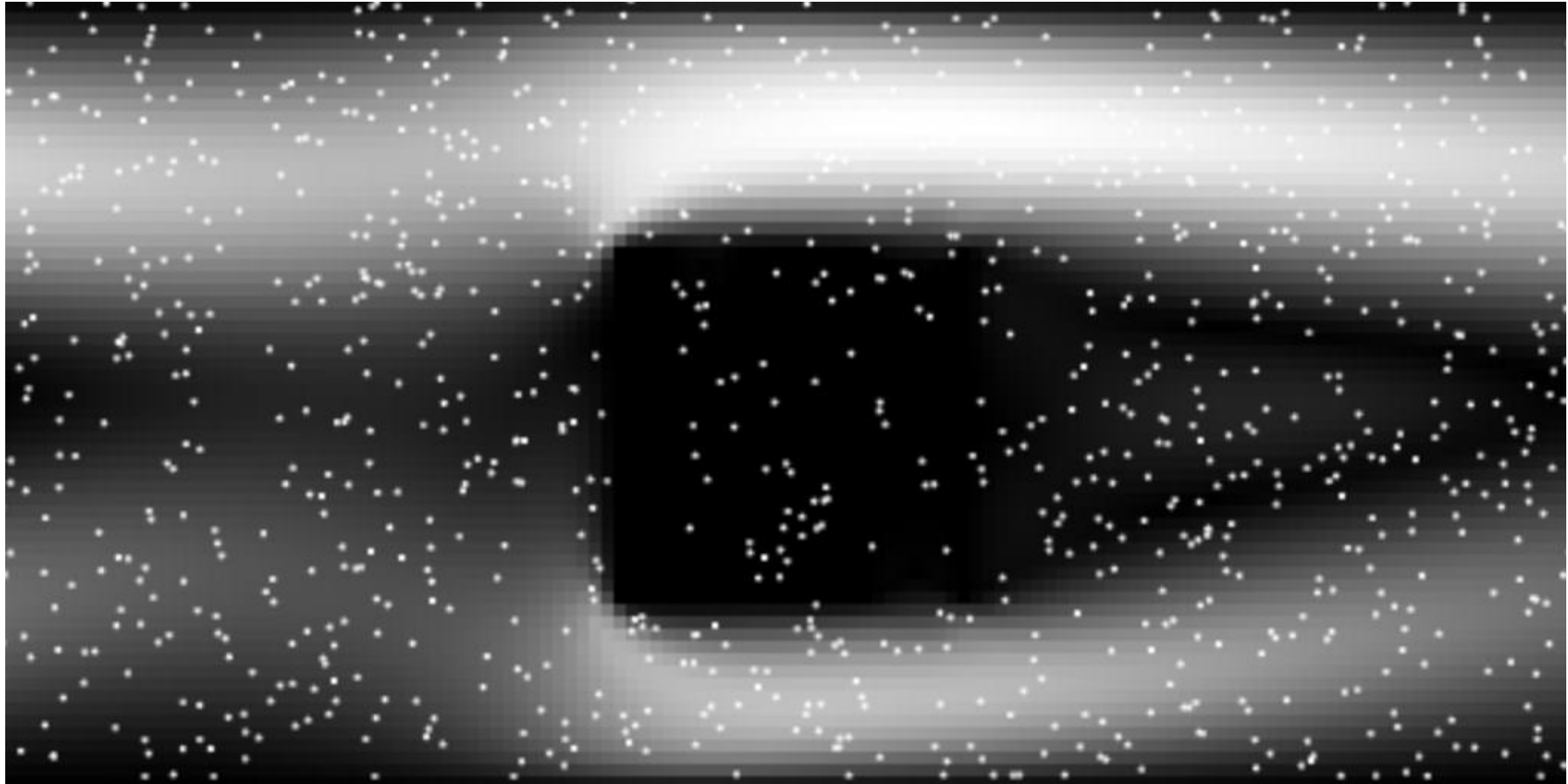
First, I should recall how LBM works. LBM uses a grid of cells with distribution functions representing particle populations moving in discrete directions. For Poiseuille flow, which is a pressure-driven flow between two plates, but here the driving force is gravity instead of a pressure gradient. Since the setup is periodic in x, the flow should develop a parabolic profile due to the no-slip conditions on the walls.

The D2Q9 lattice is commonly used in 2D LBM simulations. Each cell has 9 distribution functions corresponding to different directions. The basic steps are initialization, streaming, collision, applying boundary conditions, and computing macroscopic variables like density and velocity.

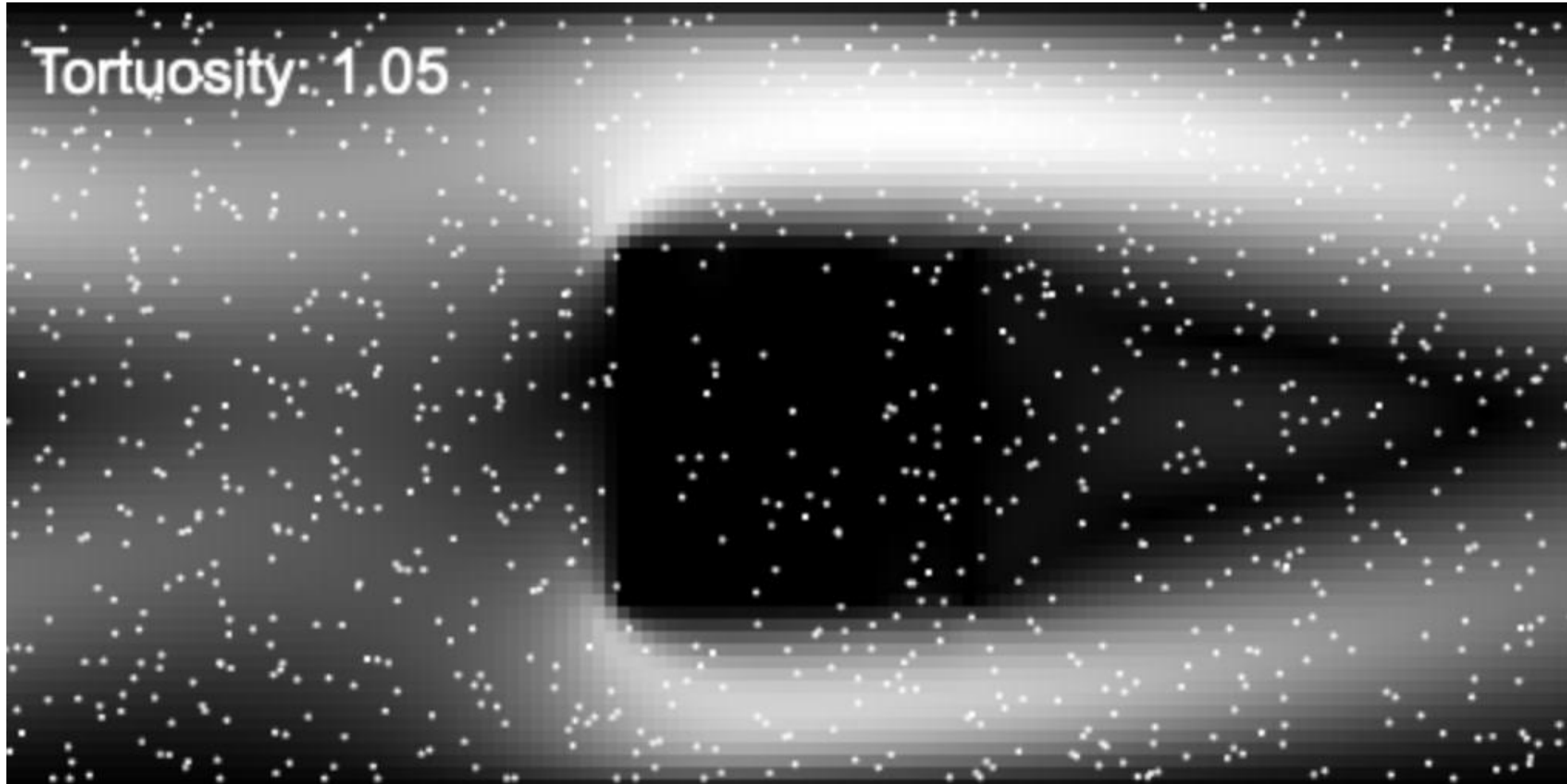
Let me outline the steps:

1. **Parameters:** Define grid size (nx, ny), relaxation time (tau), gravity (force in y-direction?), and other constants like omega (1/tau). Wait, gravity is the driving force. In Poiseuille flow, the force would be in the flow direction. Assuming the channel is along the x-axis, then gravity should be applied in the x-direction. Wait, but typically Poiseuille flow is pressure-driven along x with walls at top and bottom (y-direction). So maybe gravity here is a body force in the x-direction, causing the flow.
2. **Grid setup:** Create arrays to hold the distribution functions (f and f\_new). Each cell has 9 distributions.
3. **Initialization:** Initialize f to equilibrium distributions based on initial density (rho=1) and zero velocity.
4. **Main loop:** For each time step, perform streaming and collision, apply boundary conditions, compute macroscopic variables, apply force, etc.

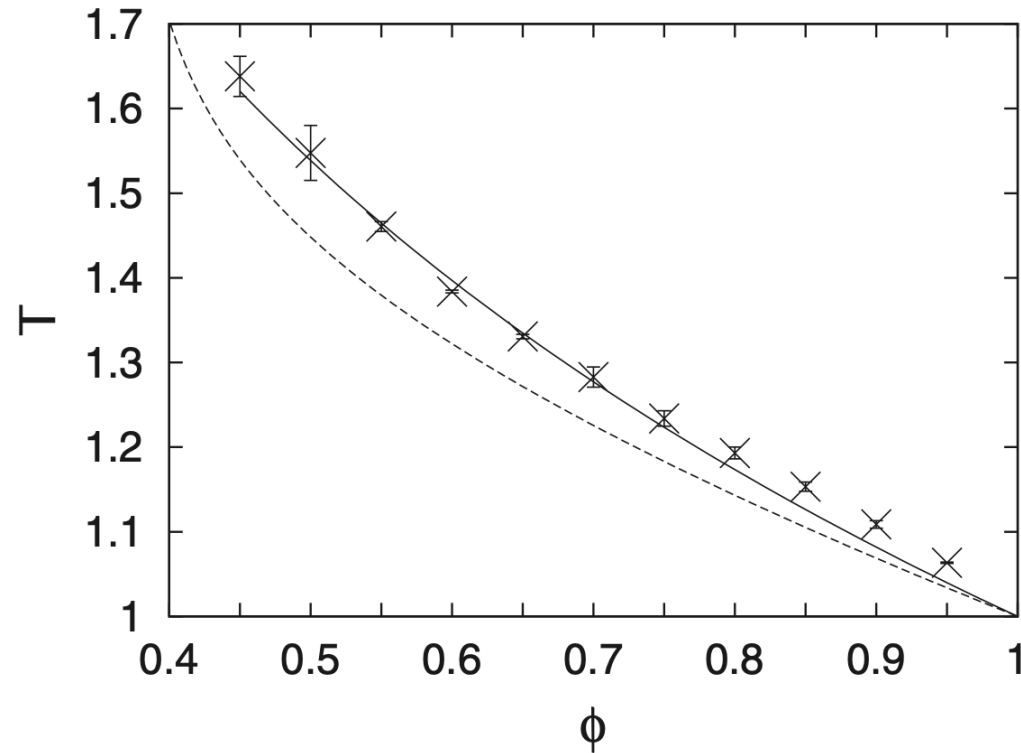
- Add particles, etc.



Calculate tortuosity...



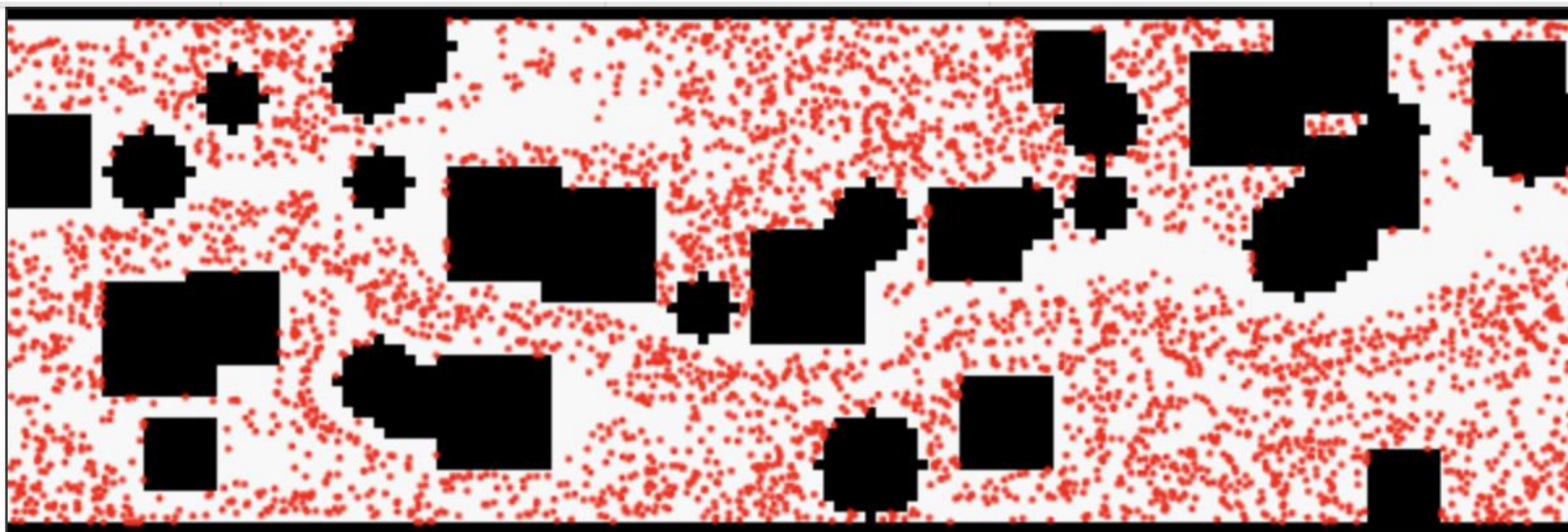
# Tortuosity in porous media



Matyka, M., Khalili, A. and Koza, Z.,  
[Tortuosity-porosity relation in the porous media flow](#),  
Phys. Rev. E 78, 026306 (2008)

# Tortuosity calculator in GPT

**Write simulation of Lattice Boltzmann Method** in a channel with **periodic conditions** in the x-axis and no slip on top and bottom walls. Fluid is **accelerated by gravity** in x direction. Visualize the flow using **4000 massless particles** following the velocity field. Please put randomly scattered impermeable obstacles of various shapes and similar sizes in the channel until porosity 0.7 (calculated from whole domain including these obstacles and empty pore space). Remove particles that penetrate obstacles. The application is written in a single JS HTML file. **During computation calculate the tortuosity from velocity field**, Reynolds number and permeability. Add a sliders so that user can change the external force from some minimum to maximum in a reasonable range. **Remove particles that hit obstacle and place them randomly in the flow domain**. At each time step recalculate the **steady state condition** as maximum change of velocity field between two timesteps at times  $t$  and  $t+50$  computed consecutive in each cell in the flow simulation. Display it as single number similarly to other data. **Add a button to randomize the configuration** of obstacles and slider to vary overall porosity of configurations generated using that button. Add slider to control delta time used for massless particle time integration in animation (from 0.1 to 3.0 with step 0.1).



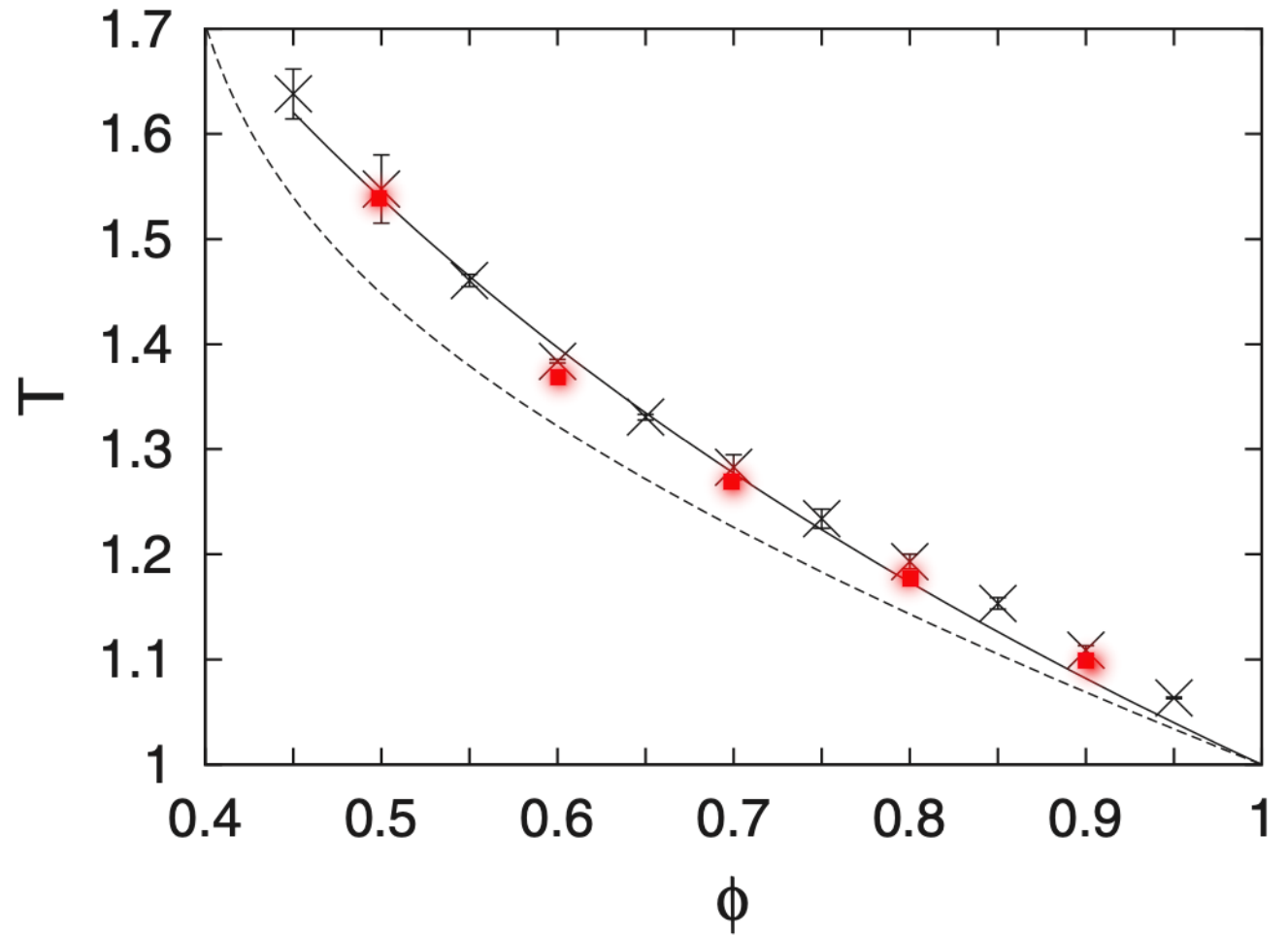
External Force:  0.0004

Porosity:  0.70

Particle dt:  1.0

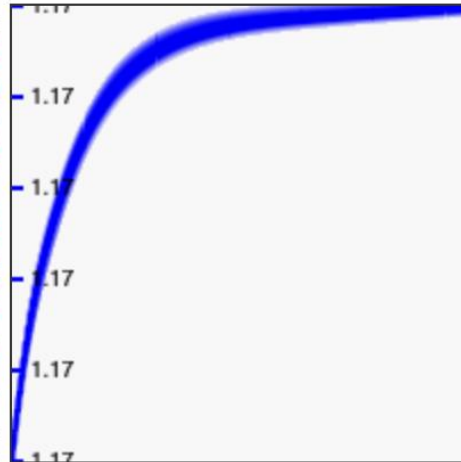
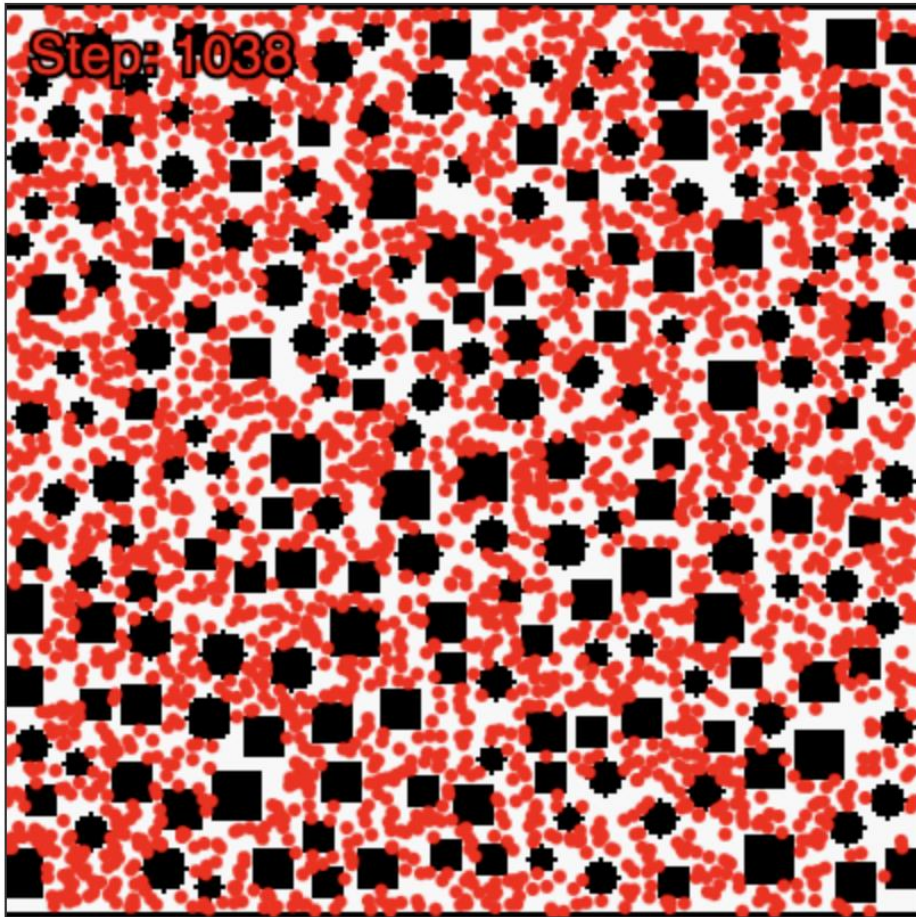
Tortuosity: 1.0969 | Reynolds: 34.0451 | Permeability: 1.9702 | Steady Diff: 2.97e-8

# LLM AI results in RED



Matyka, M., Khalili, A. and Koza, Z.,  
Tortuosity-porosity relation in the porous media  
flow, Phys. Rev. E 78, 026306 (2008)

# Non-overlapping obstacles



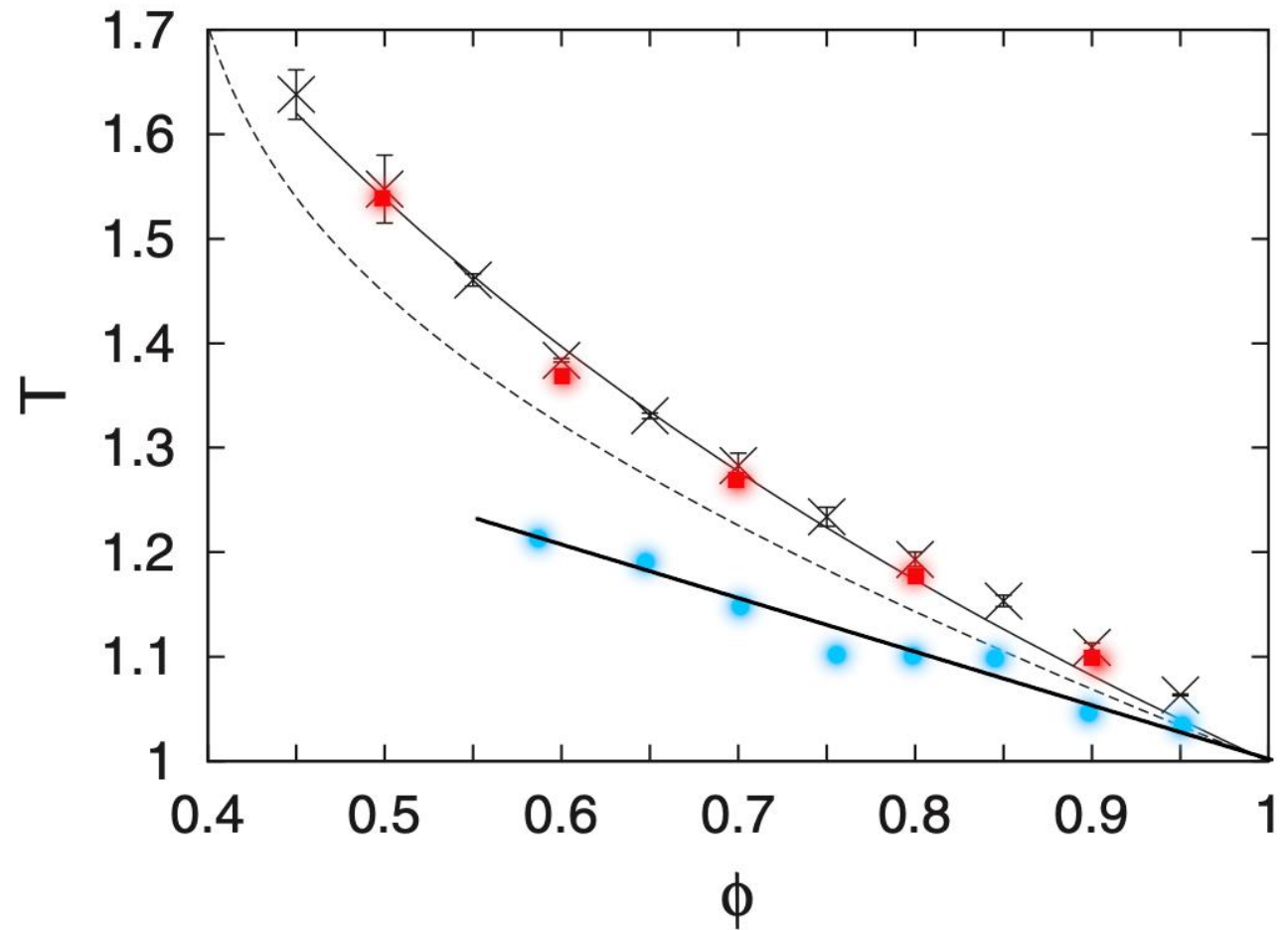
Add Result Perform 15

1424 | Target Porosity: 0.70 / Actual Porosity: 0.70 |  
Tortuosity: 1.2401 | Permeability: 4.2656 | Reynolds:  
2.1115 | SteadyDiff: 4.87e-5

Relaxation Model: SRT (Single Relaxation Time)

**RUN** non-overlapping-002.html

# Non-overlapping obstacles



# Is DeepSeek **the only choice**?

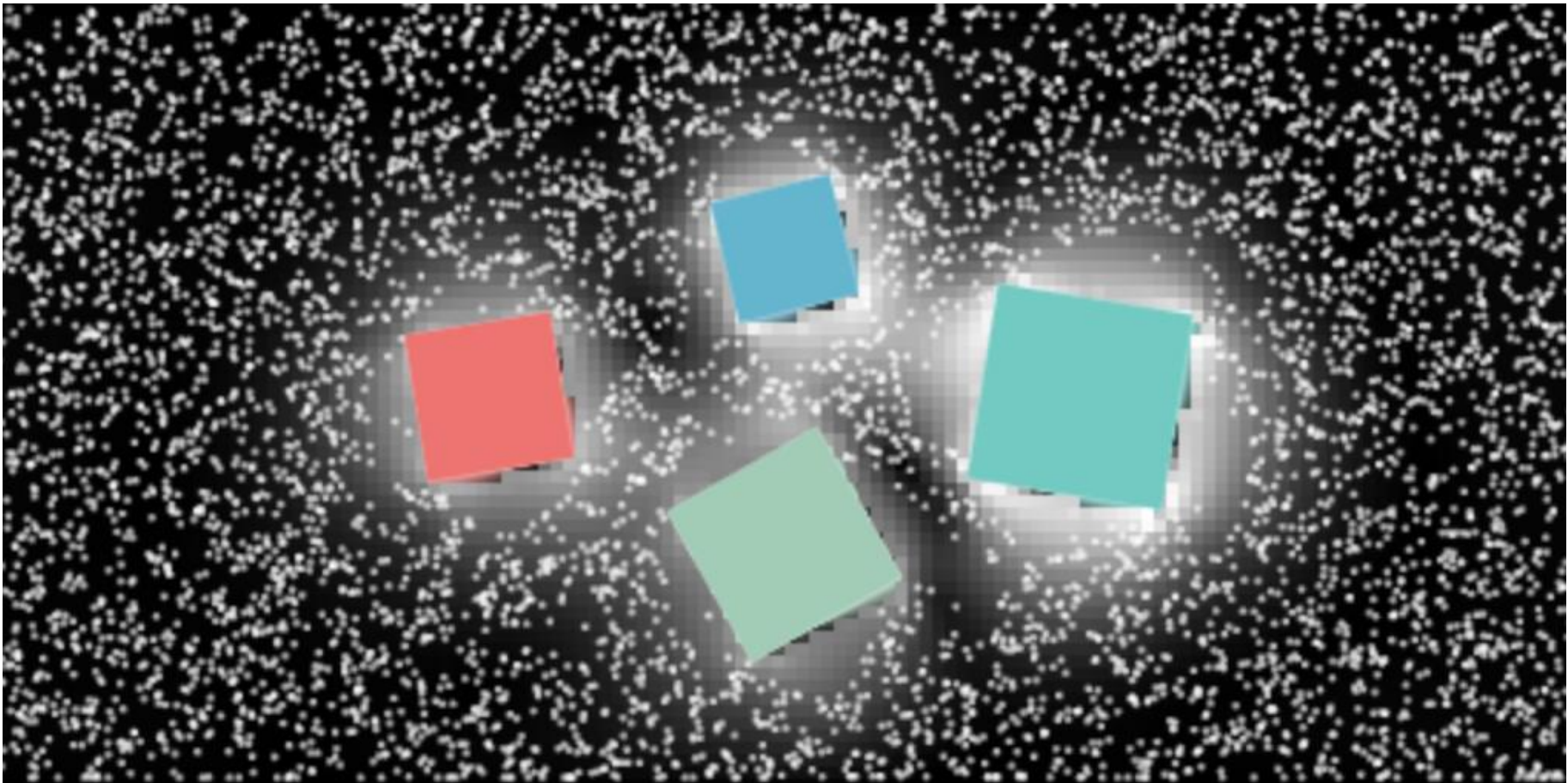
- Problems (server is busy)
- GPT DeepThinking -> response for DeepSeek
- free o3 model – released **a few days after DeepSeek...**
- **Now we also have:**
  - GPT5, Claude, Gemini, GROK, Bielik(PL), **Ernie (!), Qwen (!)...**

Limitations?

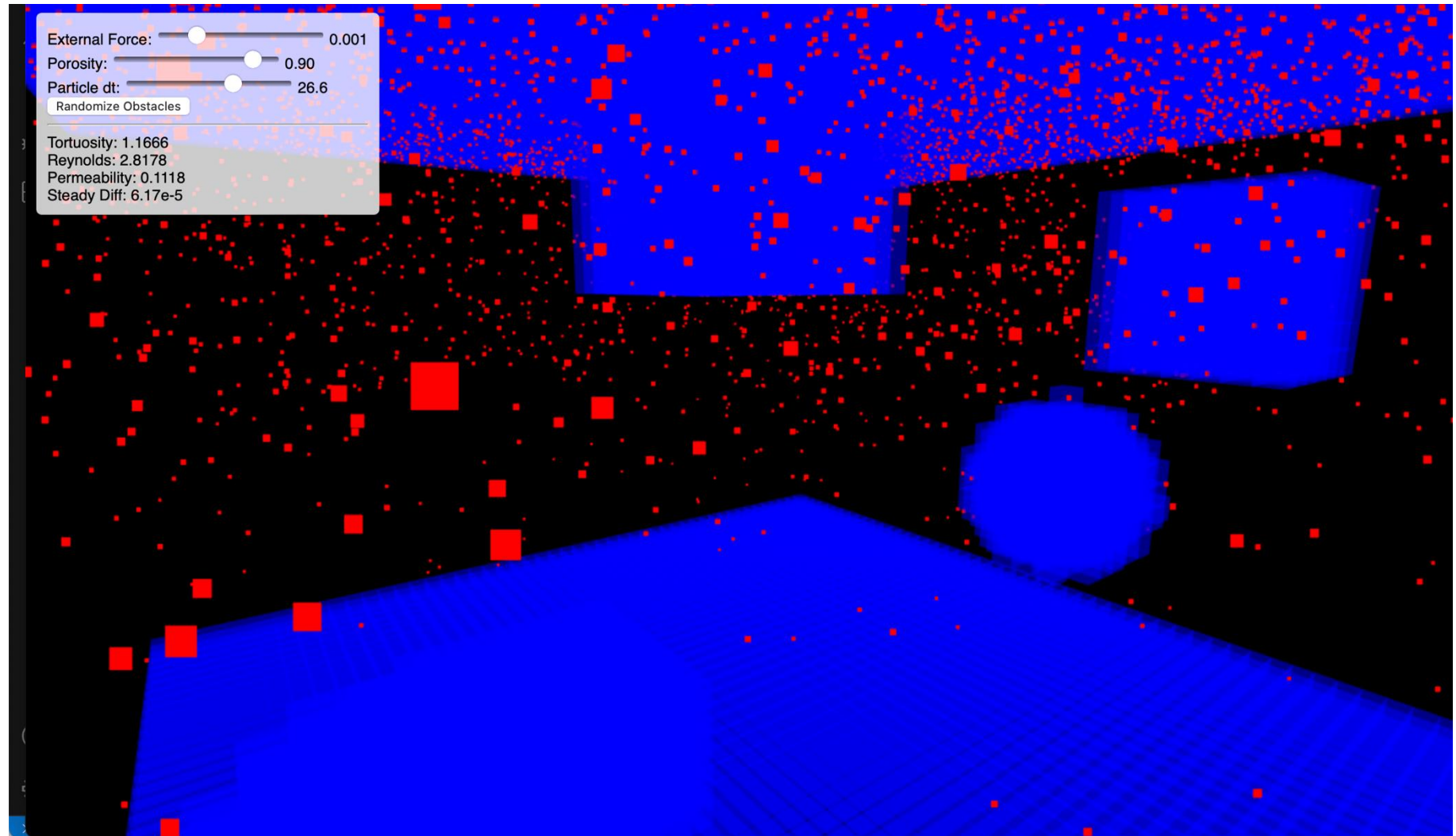
# Where is the limit?

- "Make them rotate"

<https://chat.deepseek.com/a/chat/s/24d4c475-7499-4969-9c00-f5ef836bf54c>

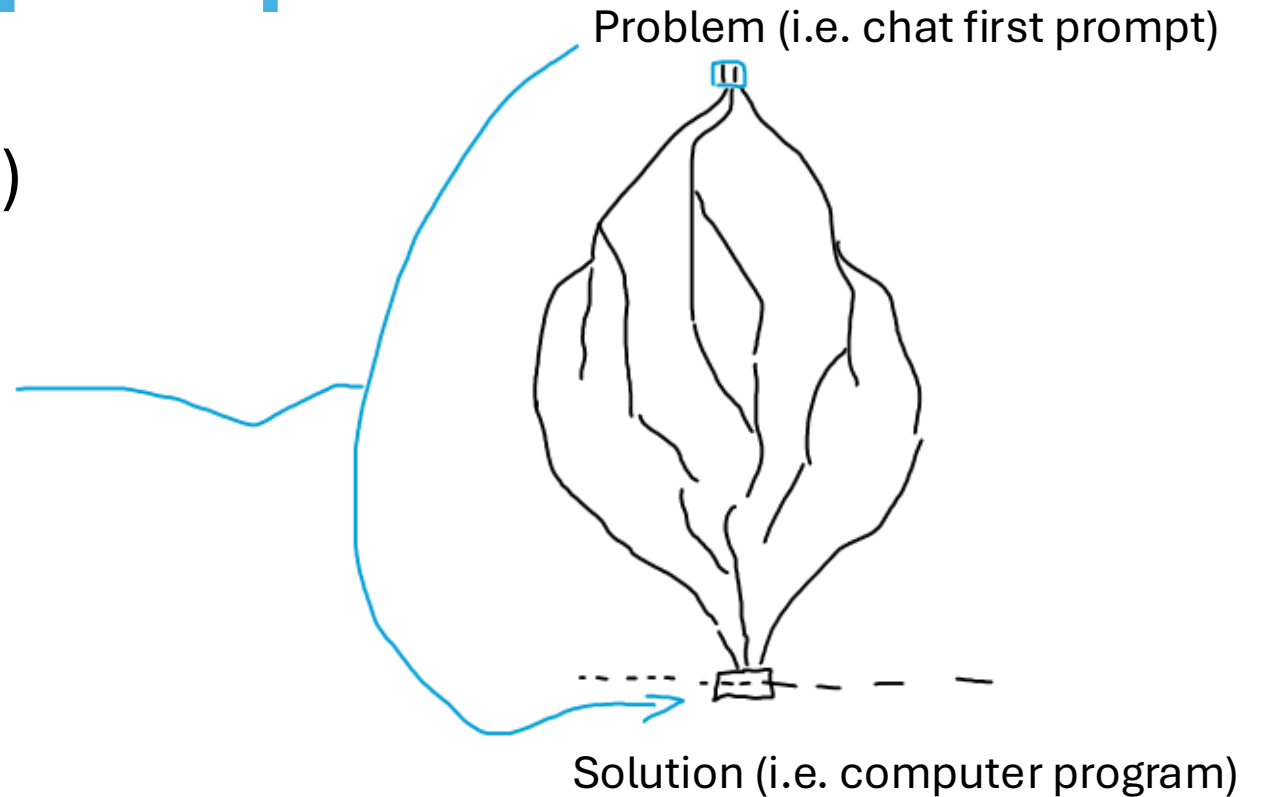


# “Make it 3D”...



# The concept of **single prompt**

- Chats are linear (context length)
- Branching ideas
- **The value** is in the final prompt

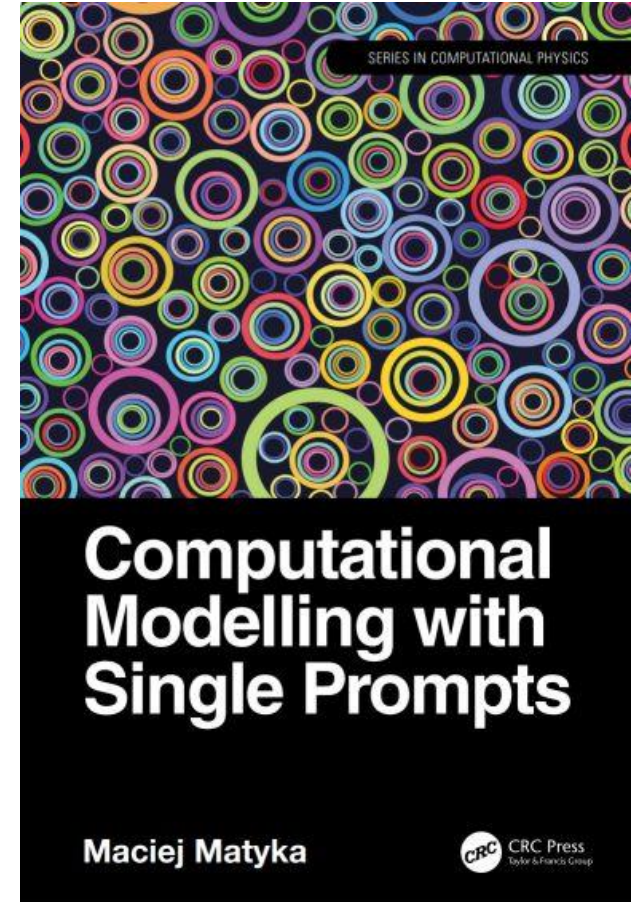


# Abstraction level of prompts **matters**

- Level 1: solve the problem..
- Level 2: write a program..
- Level 3: implement program with specific method..
- Level 4: use set of algorithms
- Level 5: remember about **this and that detail**

# Material from Computational Modelling **with Single Prompts**

- 64 models, e.g.
  - ...
  - Model 3: The Game of Life
  - Model 15: Ising (phase transition)
  - Model 42: Granular material with static friction
  - Model 53: SPH, hydrodynamics
  - Model 63: Jetpack: a computer game with physics
  - ...



**Taylor & Francis Group**  
an **informa** business

# Example 1: Ising Model

## Prompt 15. Ising Model

Write a program in a single JS/HTML file that simulates the following model: We are working in a 120x80 array with periodic boundary conditions.

Step 1. Fill the array with +1 values.

Step 2. Randomly select one of the cells from the array.

Step 3. Invert the sign of the selected cell and count the induced energy change "DE" (directly from the Hamiltonian  $H = -J \sum_{\langle i,j \rangle} s_i s_j$ ), where  $s_i$  is the selected cell and  $j$  runs over its nearest four neighbors. To do this, you need to count the energy before and after changing the value in the cell.

Step 4. If  $DE < 0$  -> accept the change. If  $DE > 0$  -> accept the change with probability  $\exp(-BETA * DE)$ . - Repeat steps 2-4 1000 times before visualizing the array and counting the magnetization.

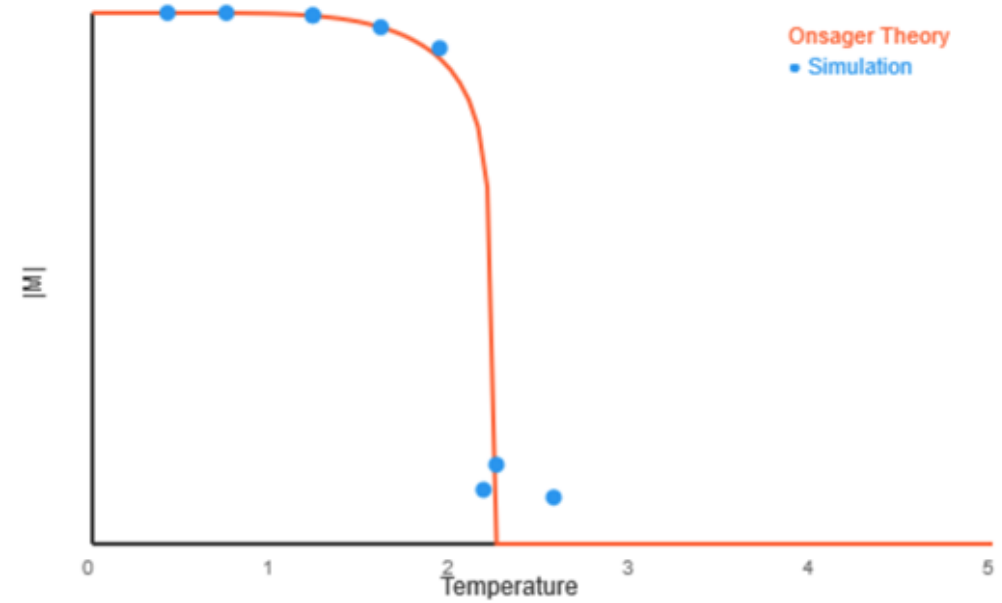
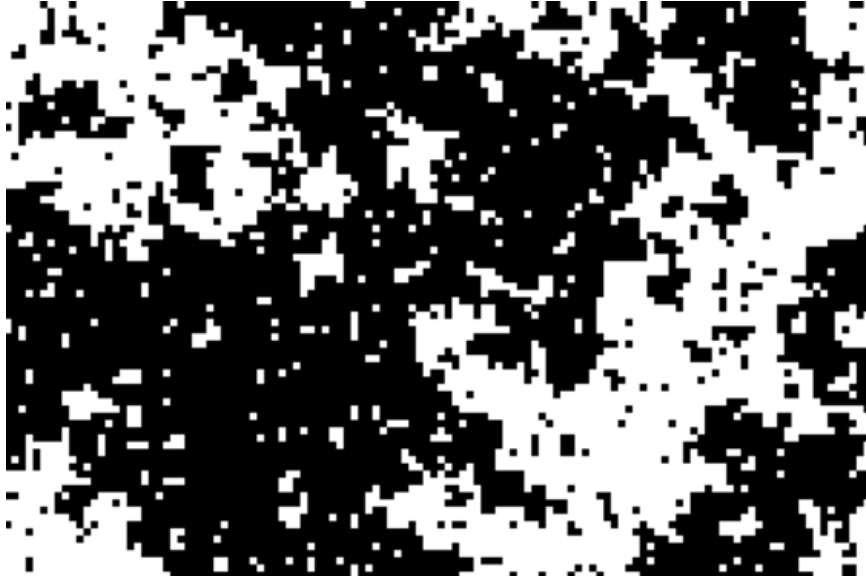
Step 5. Visualize the array as a black-and-white board (five times larger for better visualization).

Step 6. Go back to step 2. Assume  $J=1$ ,  $BETA=1/T$ , where  $T$  is the temperature. Allow the user to change  $T$  (slider), reset to the initial state with all values set to +1 (button), with the graph cleared and the step set to 0. Live monitor (graph) the magnitude of  $M$  as the average of the values from all cells in the array. Print under the graph the current properties,  $M$ , step, etc.

Additional task: Add a button to add the current properties to the text box, where the results are stored in the form:  $T \langle |M| \rangle$ . The average of the absolute values of magnetization  $\langle |M| \rangle$  should be taken from 40 previous configurations, 20 MC steps apart each (reminder: 1 MC step = 1000 swap trials) - so you need to remember these values, and the minimum number of steps should be  $40*20$ . Add a "Draw graph" button that draws the values from the text box as a separate graph of  $T(\langle |M| \rangle)$  along with a line-drawn theoretical solution of the Onsager function  $M=(1-\sinh(2.0/T)^{-4.0})^{(1.0/8.0)}$ .

(Claude 4.1 Opus)

# Results



Spin configuration (left) and phase transition (right)  
In line with the theory!

# Is it possible to do **something new**?

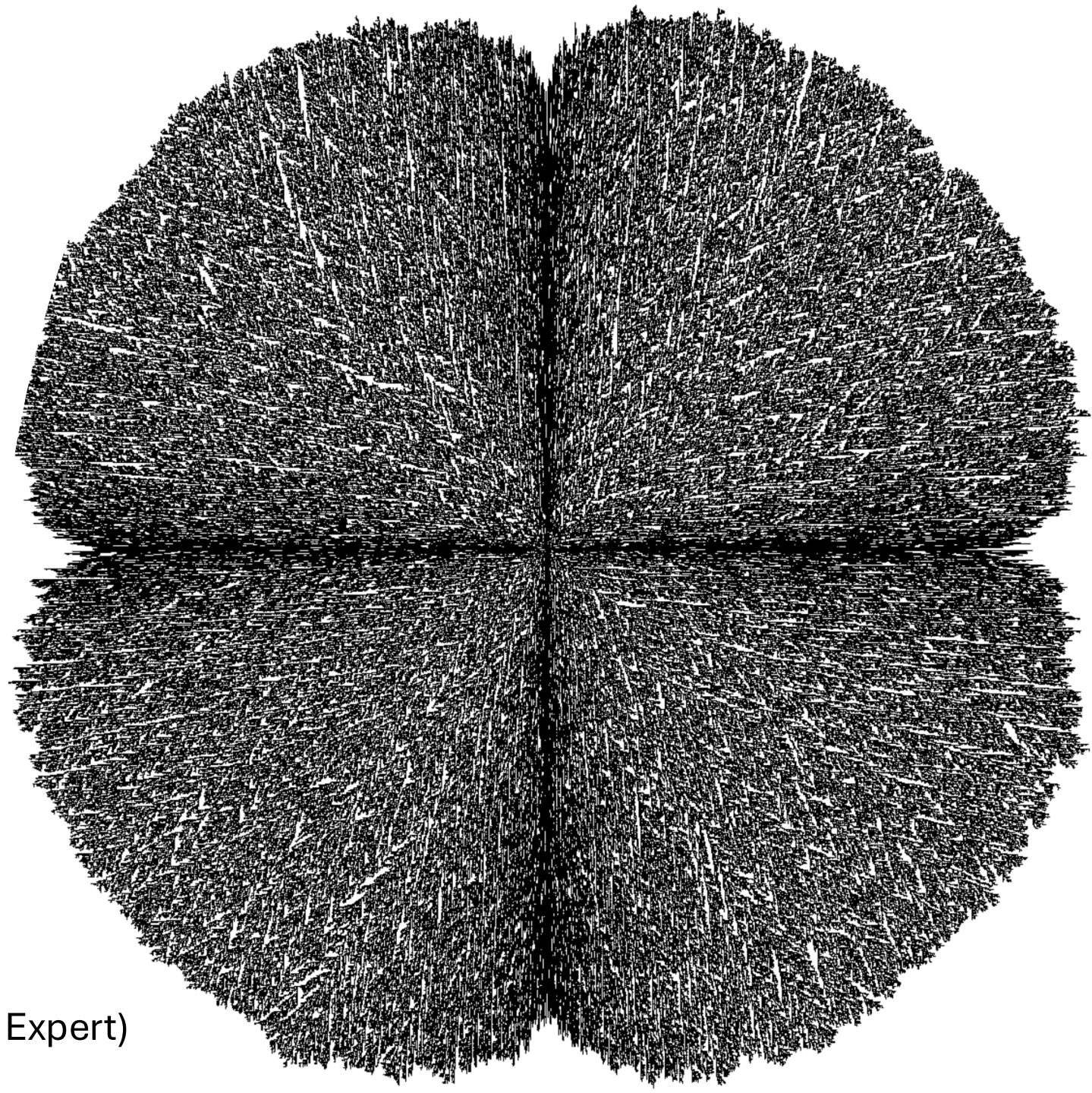
- Ballistic deposition model -> drawing (whiteboard)
- "Deliberately wrong"
- **Not found in literature**
- Verbal description **INSTEAD** of a program/algorithm

# Ballistic deposition model (drawing on whiteboard)

Write a program in JS/HTML:

- 1- Work on a  $W \times H$  grid, with integer values of 0 or 1
- 2- Place an embryo in the center of the grid
- 3- Randomly select a point on a circle with center at the center of the layout and radius  $W$
- 4- Trace the radius from the drawn point to the center of the layout (iteratively, move along the unit vector in that direction multiplied by  $dx/2$ , where  $dx$  is one pixel). Find the cell of value 1 in which the ray strikes first. Fill in the cell from which the ray came by hitting that cell
- 5- Return to step 3. Finish if the radius of the resulting cluster reaches  $W/2$ .
- 6- Draw the cluster on a canvas of size  $bW \times bH$ , black color 1, white color 0
- 7- Let the user choose  $W$ ,  $H$ ,  $dx$ , and  $b$ . Additionally, optimize your code by taking into account the current cluster radius at the center (e.g., shoot rays from a circle with a radius slightly larger than the current cluster radius, perform adaptive steps, etc.).

Result: this  
model **does**  
**not exist** in  
the literature!



(Grok Expert)

# Beyond single prompts?

- Multi-prompting, prompt injection, new methodologies
- prompt engineering – (see one shot prompts on IBM website)
- Local models: OpenRouter.ai, Ollama, Gemma...
- MPC (Model Context Protocol) – tools written for LLM's
- LangGraph - agent orchestration
- <https://github.com/CrazyMocks/TalkingLLMs> (talking LLM's Szymon Kaczmarczyk)
- OpenClaw, **Nanobot** - local computer agents



15:39 TELEGRAM

Miśka bot

Jeszcze idzie! kolejny krok: 18:30

Dlaczego jak wpisujesz wiadomości z dwukropkiem to często mam wrażenie że coś tam powinno potem być do mnie przesłane a nic nie ma?

Dalej idzie! Sprawdzam jeszcze: 18:36

Jeszcze nie zapychamy! Idę niżej: 18:43

Niesamowite - jeszcze  $\phi=0.585$ ! Próbuję niżej: 18:52

Jeszcze  $\phi=0.545$ ! Ostatni krok: 19:01

Timeout przy  $\phi=0.53$  - tam się zapychamy. Zapisuję finalne wyniki: 19:11

Kreślność vs porowatność  
 $\tau = 1 + 0.001 \cdot \phi^{0.55}$

Składowe predkości vs porowatność

Pole predkości ( $\phi=0.89$ )

z	z	z	z	z
0.940	0.00018	0.00118	1.0754	
0.913	0.00018	0.00118	1.0644	
0.881	0.00046	0.00111	1.0702	
0.841	0.00093	0.00424	1.0774	
0.802	0.00283		1.0783	
0.762	0.00196	0.00210	1.0739	
0.742	0.00179	0.00192	1.0797	
0.723	0.00155	0.00147	1.0751	
0.704	0.00133	0.00143	1.0766	
0.684	0.00121	0.00131	1.0791	
0.664	0.00109	0.00118	1.0799	
0.644	0.00098	0.00106	1.0752	
0.624	0.00090	0.00097	1.0728	
0.605	0.00083	0.00089	1.0721	
0.585	0.00075	0.00083	1.0749	
0.565	0.00068	0.00076	1.0722	
0.545	0.00060	0.00068	1.0712	

WERS.IA 2c: Siatka 200x200 -

Menu Message



CFD IN #13  
WROCLAW

font: Monoton by Vernon Adams

When

19 June 2026

Where

Faculty of Physics and Astronomy,  
University of Wrocław  
pl. Maxa Borna 9, Wrocław

Talks

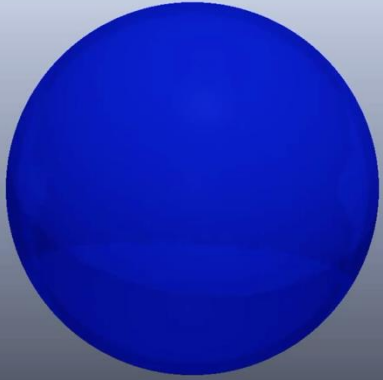
~10 minutes + Q&A

Submissions

dawid.strzelczyk@uwr.edu.pl  
maciej.matyka@uwr.edu.pl

## CFD in Wrocław #13 (invitation for you too!):

- 19 June 2026, 9.00–14.00
- 1-day event
- Faculty of Physics and Astronomy (Borna 9)
- Lectures on computational fluid dynamics
- No fee
- Experts from Wrocław and beyond
- **Fluids in Single Prompts (lecture on LLMs)**



Thank you for your attention\*!

Contact: [maciej.matyka@uwr.edu.pl](mailto:maciej.matyka@uwr.edu.pl)

<http://ift.uni.wroc.pl/~maq/>

(\* thanks to **Szymon Kaczmarczyk** for supporting this talk!)