

# Opis in arhitektura: vizualizacija toka vode (Python in C++)

Projekt: water\_flow\_visualization

September 22, 2025

## Povzetek

Ta dokument opisuje implementacijo vizualizacije toka vode v dveh izvedbah: Python (`water_flow_visualization.py`) in C++ (`water_flow_visualization.cpp`). Algoritem temelji na divergencno prostem hitrostnem polju (izpeljanem iz časovno spremenljive tokovne funkcije), polju barvila ("dye"), pol-lagrangevem prenašanju (advekciji) in blagem dušenju, kar daje vodnat, turbulenten videz. Program omogoča živ prikaz (okno) in izvoz animiranega GIF-a.

## 1 Pregled algoritma

- **Tokovna funkcija**  $\psi(x, y, t)$ : kombinacija sinusnih/cosinusnih vzorcev, ki se s časom spreminjajo.
- **Hitrostno polje**  $\vec{v} = (u, v)$ : izračunano kot rotacija tokovne funkcije,  $u = \partial\psi/\partial y$ ,  $v = -\partial\psi/\partial x$ , skalirano in rahlo zglajeno (Gauss).
- **Polje barvila** (RGB): inicializirano z modrim odtenkom, šumom in vinjetiranjem za globinski učinek.
- **Advekcija** (pol-lagrangevska): za vsak piksel sledimo poti nazaj po  $\vec{v}$  in vzorčimo z bilinearno interpolacijo.
- **Dušenje in barvno ravnotežje**: mešanica trenutnega barvila z osnovnim (0.5%).
- **Izhod**: posodabljanje okna (neobvezno) in sestavljanje GIF-a s podano hitrostjo sličic (FPS).

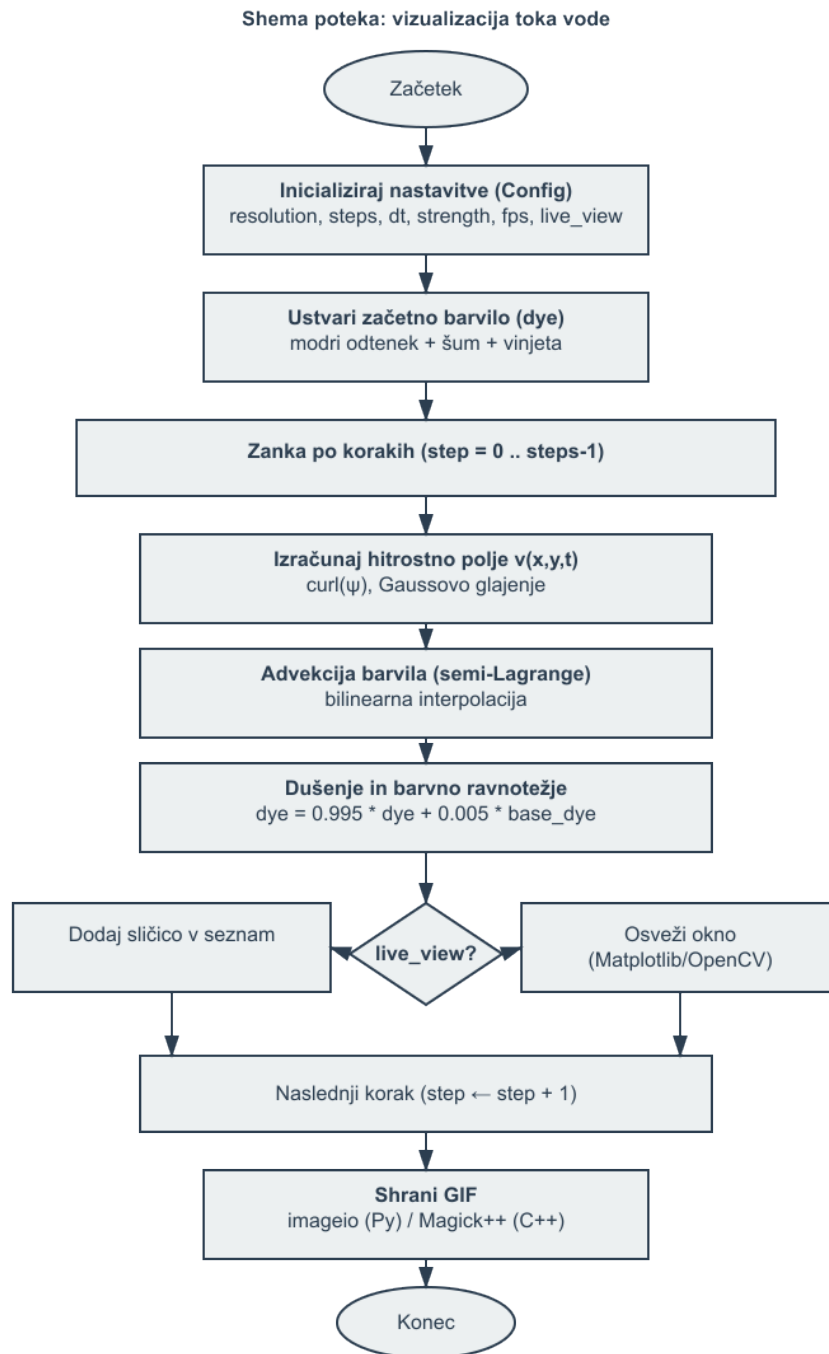


Figure 1: Shema poteka algoritma (flowchart).

## 2 Python: `water_flow_visualization.py`

### Ključne komponente

- `SimulationConfig`: nastavitve (`resolution`, `steps`, `dt`, `strength`, `fps`, `live_view`, `gif_name`, `output_dir`).

- **Tokovna funkcija in hitrostno polje:** `stream_function`, `velocity_field`, `gaussian_blur`.
- **Vzorčenje in advekcija:** `bilinear_sample`, `advect`.
- **Inicializacija barvila:** `create_initial_dye` (modri toni + Gaussov šum + vinjeta).
- **Zanka simulacije:** izračun  $\vec{v}$ , advekcija, dušenje, osvežitev okna (če je omogočeno), zbiranje sličic.
- **Izvoz:** `imageio.mimsave` za GIF; `matplotlib` za živ prikaz.

## Zagon

```
python water_flow_visualization.py --fps=90 --strength=1.2 --no-live-view
```

Podprti so tudi preprosti preglasi nastavitve v obliki `-key=value`.

## 3 C++: `water_flow_visualization.cpp`

### Ključne komponente

- **Config:** enake nastavitve kot v Pythonu.
- **Tok in hitrost:** `streamFunction`, `buildVelocityField`, `gaussianBlur` (ločljivi 1D filtri).
- **Advekcija:** `advect` z bilinearno interpolacijo.
- **Inicializacija:** `createInitialDye` (modra baza + šum + vinjeta).
- **Živ prikaz (neobvezno):** OpenCV okno (če je `USE_OPENCV`); posodobitev vsako sličico.
- **GIF izvoz:** prek `Magick++` (`ImageMagick`) se sličice zapišejo v animacijo z zelenim FPS.

### Gradnja (CMake)

```
cmake -S . -B build -DWATER_FLOW_USE_OPENCV=ON \
-DOpenCV_DIR="C:\\Libraries\\opencv-4.10.0\\Build" \
-DImageMagick_INCLUDE_DIRS="C:\\Program Files\\ImageMagick-7.1.2-Q16-HDRI\\include" \
-DImageMagick_LIBRARIES="C:\\Program Files\\ImageMagick-7.1.2-Q16-HDRI\\lib\\CORE_RL_MagickCore.lib; \\
C:\\Program Files\\ImageMagick-7.1.2-Q16-HDRI\\lib\\CORE_RL_MagickCore.lib; \\
C:\\Program Files\\ImageMagick-7.1.2-Q16-HDRI\\lib\\CORE_RL_MagickWand.lib"
cmake --build build --config Release
```

Zagon: `build`

`Release`

`water_flow_cpp.exe -fps=90` (dodajte `-no-live-view` za "headless").

## 4 Parametri in priporočila

- **resolution:** velikost slike (privzeto 512). Višje vrednosti so počasnejše.
- **steps:** število sličic (trajanje GIF-a).
- **dt:** korak časa za advekcijo (stabilnost/ostrost vzorca).
- **strength:** skala hitrosti; višje pomeni izrazitejši vrtinci.
- **fps:** hitrost GIF-a in predogleda.
- **live\_view:** omogoči/izključi okno za predogled (OpenCV/Python Matplotlib).

## 5 Razlike med izvedbama

- **Odvisnosti:** Python uporablja `numpy/matplotlib/imageio`, C++ pa `Magick++` in po želji `OpenCV`.

- **Hitrost:** C++ je običajno hitrejši; Python je preprostejši za prilagoditve.
- **Vizualno:** algoritmi so usklajeni; zaradi numeričnih razlik so možne manjše razlike v teksturah.

## 6 Izhod kot PDF

Za vključitev SVG uporabite `-shell-escape` in nameščen Inkscape:

```
pdflatex -shell-escape docs/report.tex
```

Če SVG pretvorba ni na voljo, zamenjajte `\includesvg{flowchart}` z `\includegraphics` na PNG/PDF različico diagrame (ali uporabite TikZ).