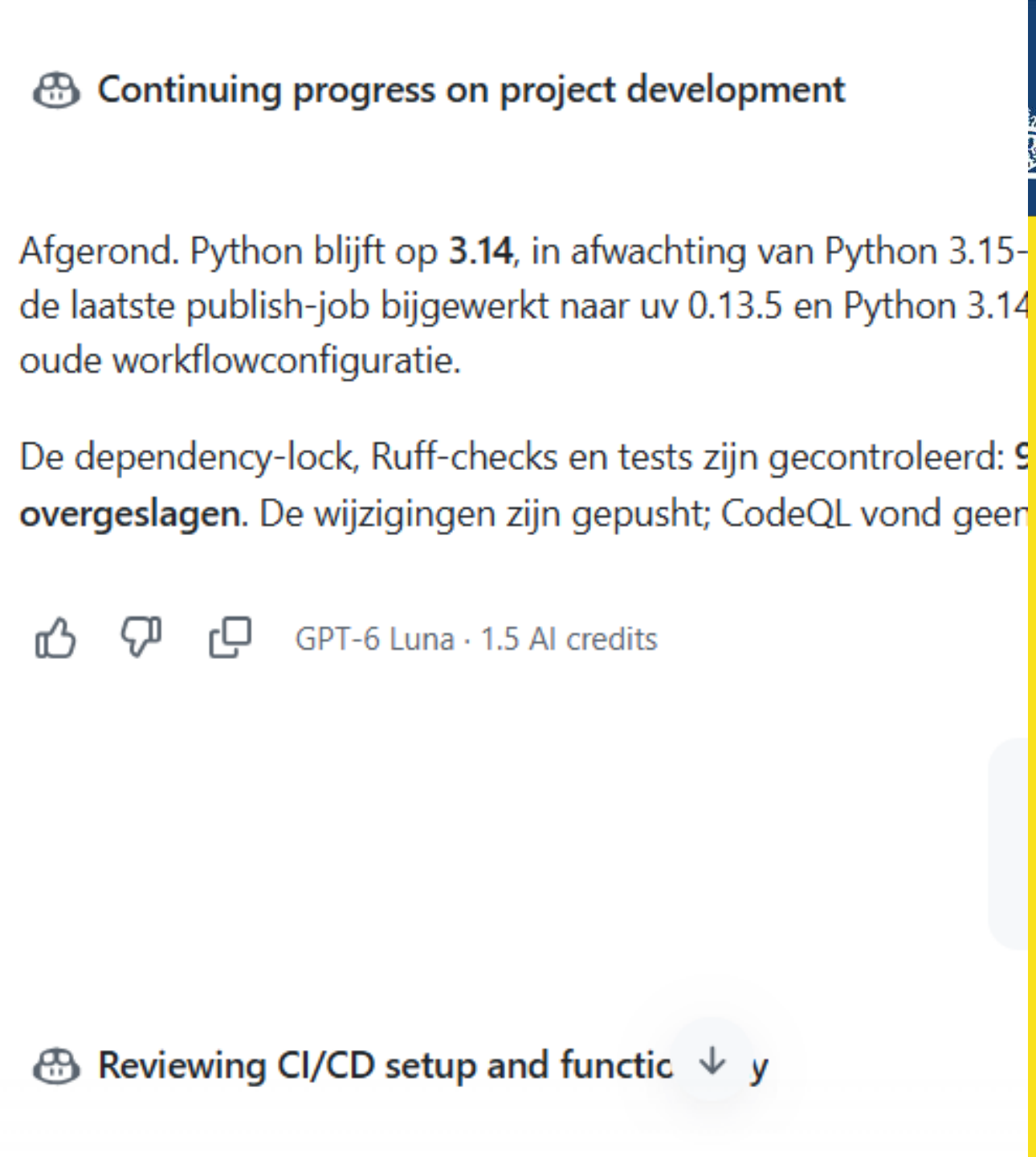


Rijkswaterstaat
Ministry of Infrastructure and Water Management

NMDC / Digishape Vibe Coding sessie

Gert-Jan Schotmeijer
Fedor Baart

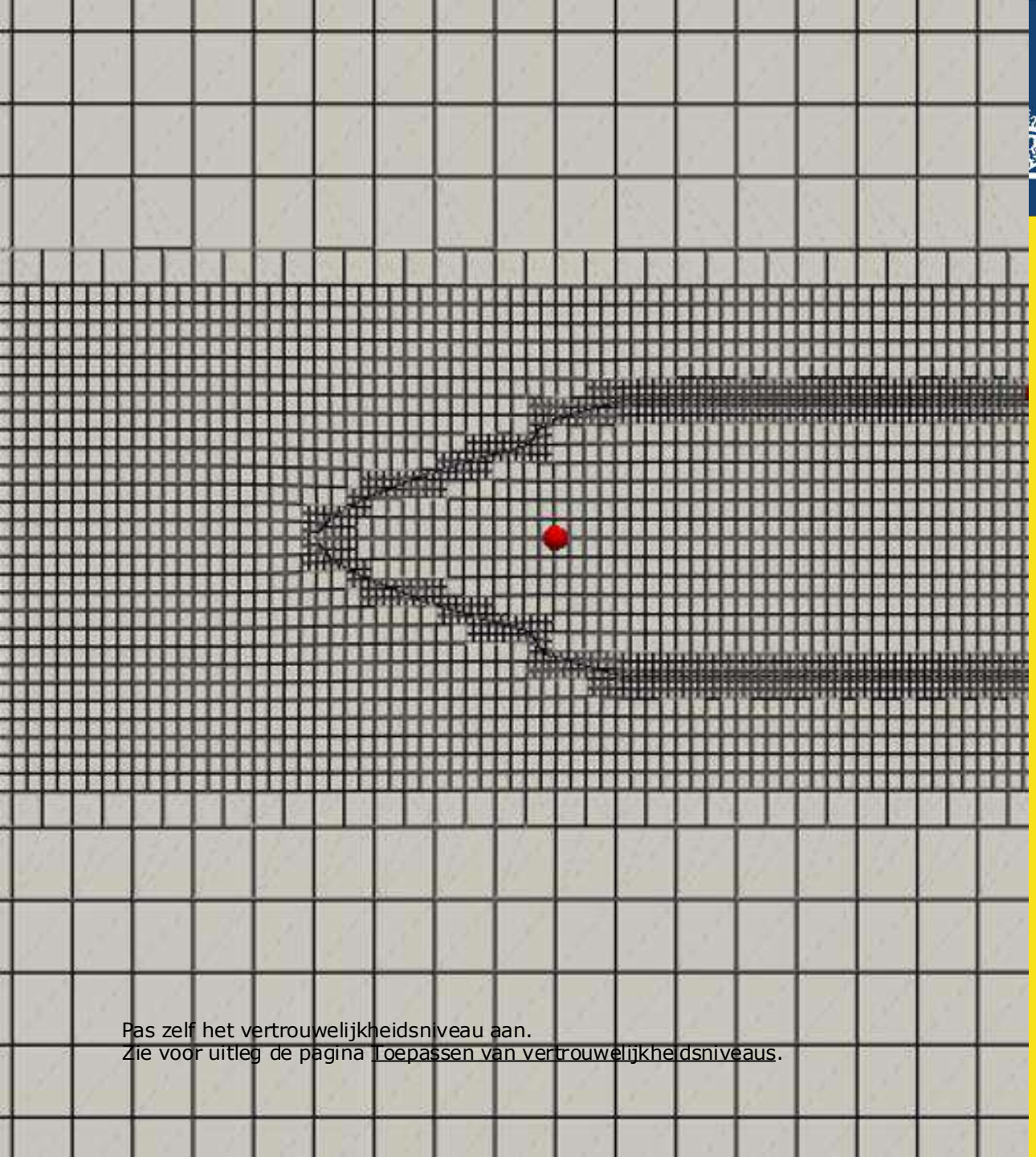
2026-10-30

Agenda

- 09:00 - 09:30 Inloop
- 09:30 - 10:15 Ruimte voor presentaties ~15 minuten
- 10:15 - 10:50 Praktijkverhalen, ruimte voor demos, home projects, ervaringen
- 10:50 - 11:10 Koffiepauze
- 11:15 - 12:00 Discussie met stellingen

Stellingen

- *"Vroeger noemden we code waarvan niemand meer wist hoe het werkte 'legacy code'. Met AI produceren we vanaf dag één alleen nog maar legacy code."*
- *"Samenwerken met een bot werkt. Samenwerken met collega's werkt. Met allebei tegelijk werkt niet."*
- *“Ik heb geen zin om de AI gegenereerde slob van anderen te beoordelen."*
- *“De drempel om AI resultaten zonder controle te gebruiken ben ik overgestapt."*



Rijkswaterstaat
Ministry of Infrastructure and Water Management

Vibe Coding for Numerical Model Generation

Towards guidelines for AI-Assisted Numerical Modeling

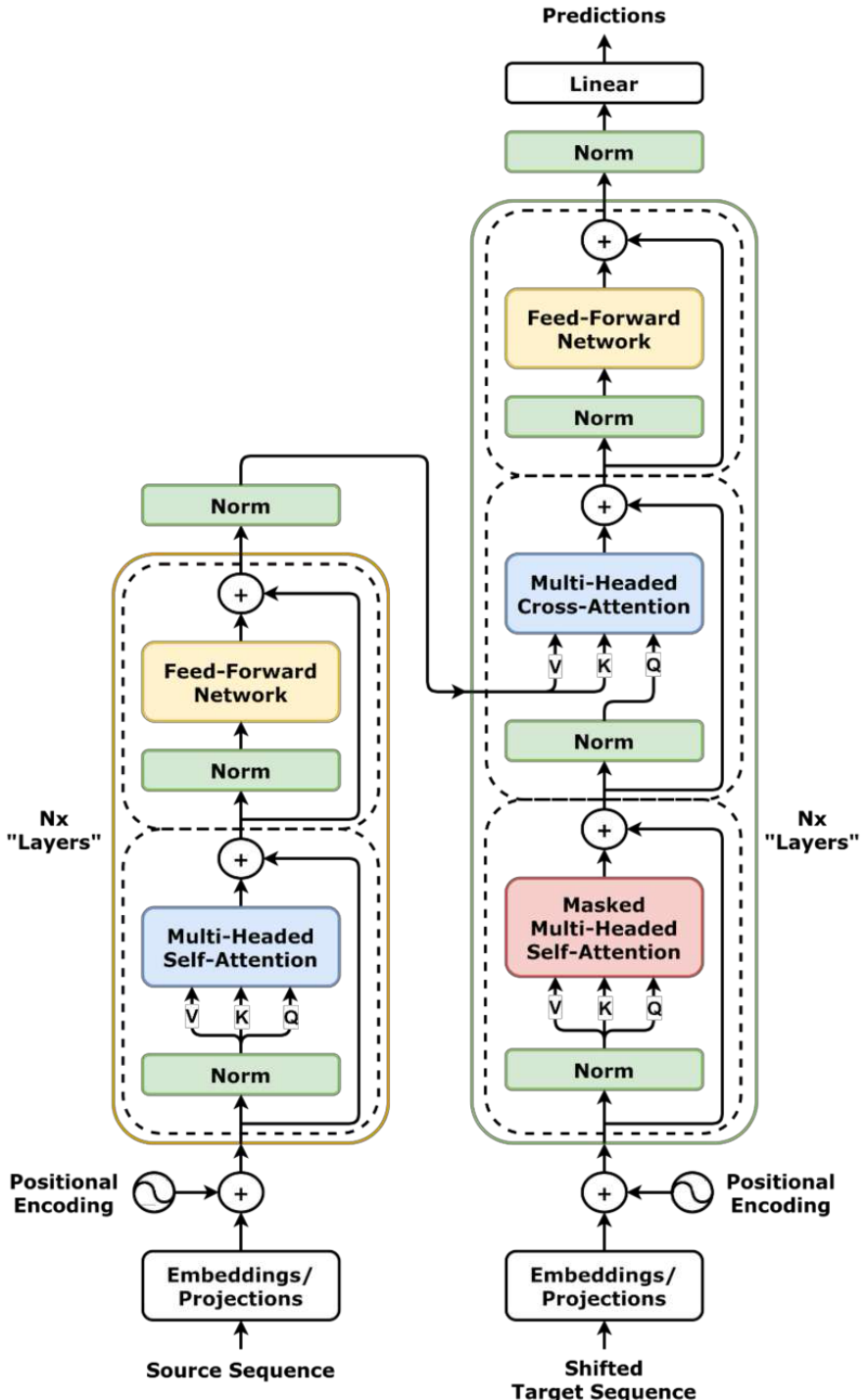
Fedor Baart

2026-10-30

Pas zelf het vertrouwelijkheidsniveau aan.
Zie voor uitleg de pagina Toepassen van vertrouwelijkheidsniveaus.

Topics

- Vibe coding what / why?
- Simple demo
- Goal
- Fitting in with AI policy
- Three experiments
- Guidelines
- Discussion



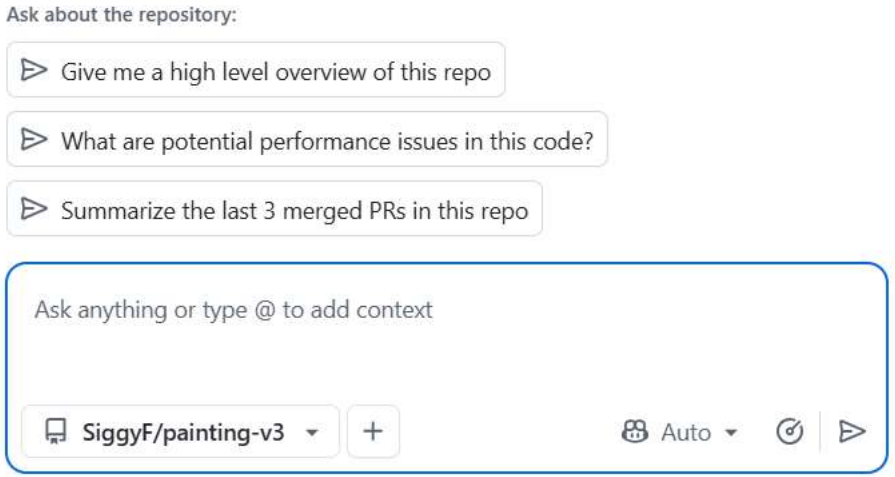
What is vibe coding



Chatbot

Typical input:
Chat, documents

Typical output:
Text, documents, images



Vibe coding

Typical input:
Chat, source code

Typical output:
Source code, datasets, configuration, documentation

Chatbot: "How do I calculate this flood?"



AI-overzicht

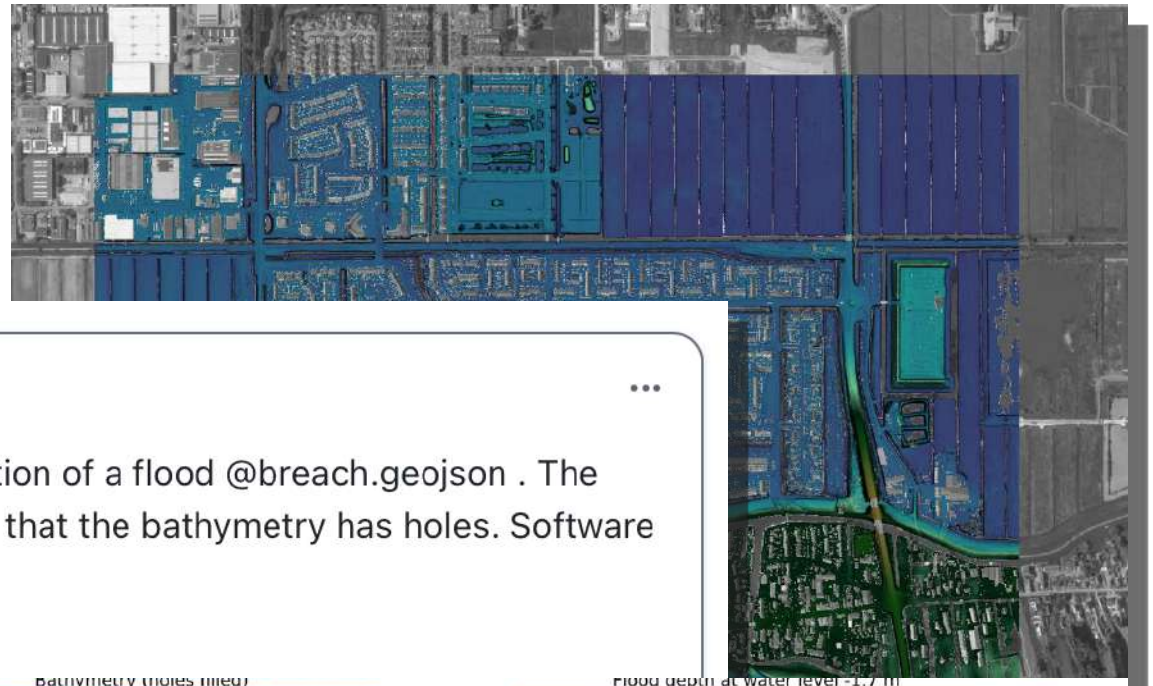
To calculate a flood, scientists and engineers **combine hydrological analysis (estimating the volume and peak flow of water from rainfall) with hydraulic modelling** (calculating how that water moves across land and changes depth based on terrain elevation). Floodinfo.ie +1



Core Steps to Calculate a Flood

- **Gather data:** Collect historical streamflow records, local rainfall intensity data, and high-resolution elevation maps (Digital Elevation Models) of the area. GOV.UK +3
- **Calculate peak discharge (Q):** Use statistical flood frequency analysis (like the USGS peak flow calculation methods) or Manning's Equation to determine how much water per second flows through a channel based on channel slope, wetted perimeter, and cross-sectional area. GOV.UK +3

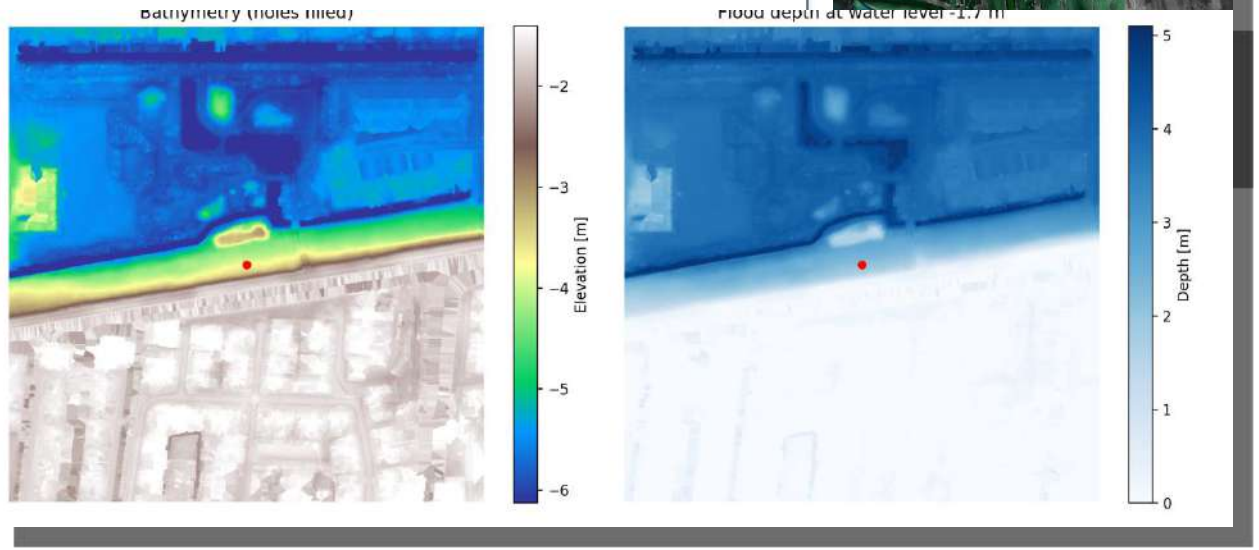
Chat + data



demo-copilot × wilnis-dtm.tif × breach.geojson ×

We have a @wilnis-dtm.tif file with a local bathymetry and a breach location of a flood @breach.geojson . The breach is about 60m. Can you compute a simple flood simulation. Notice that the bathymetry has holes. Software is installed (numpy, uv, rasterio).
You can visualize the flood in matplotlib.

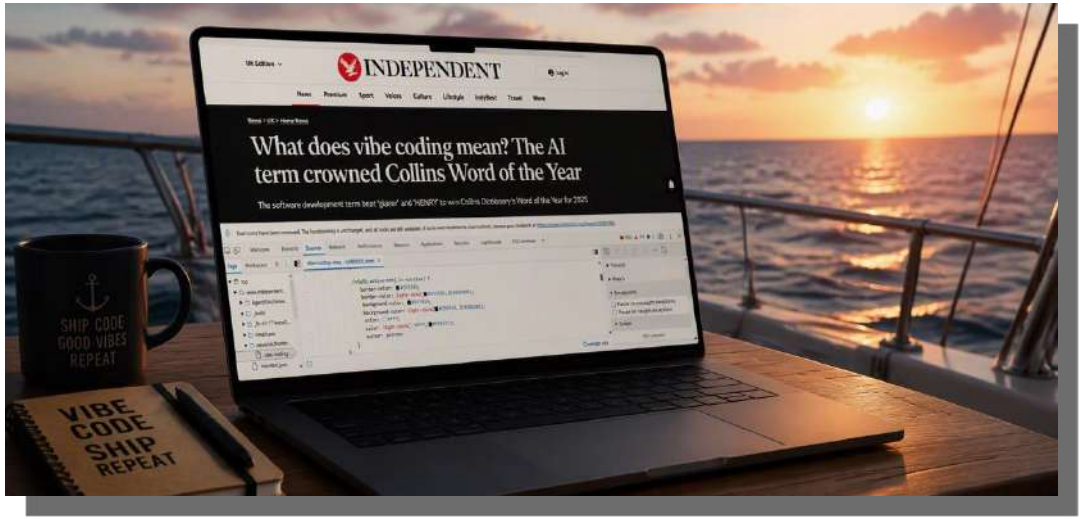
\+ Interactive | MAI-Code-1.1-Flash Low



Src: AHN5

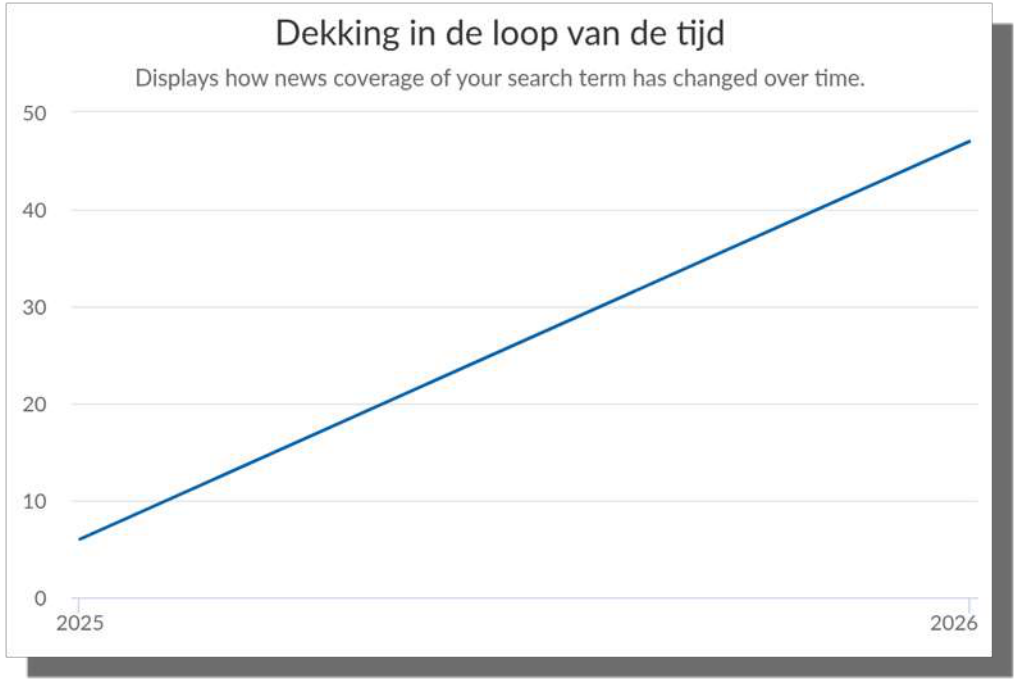
Vibe coding: terminology





Goal: prepare

- Guidelines for suppliers and knowledge partners
- Feeling for strong and weak points
- Estimates of production increase / decrease
- Qualify new risks



Emerging technology

Challenge: government guidelines

2024: AI disallowed

General LLM ban

2026: RWS: Only contracted

Only chatbot contracted



Experimenting encouraged

2025: Digital Affairs

Except for Chat Copilot

2026: RWS: All AI blocked

Wat is het standpunt?

Als overheid stimuleren wij innovatie ten behoeve van betere dienstverlening met behulp van generatieve AI. Daarnaast worden overheidsorganisaties aangemoedigd om deze technologie te betrekken in het zoeken naar oplossingen voor problemen. Generatieve AI kan worden ingekocht, ontwikkeld of gebruikt wanneer:

Approach

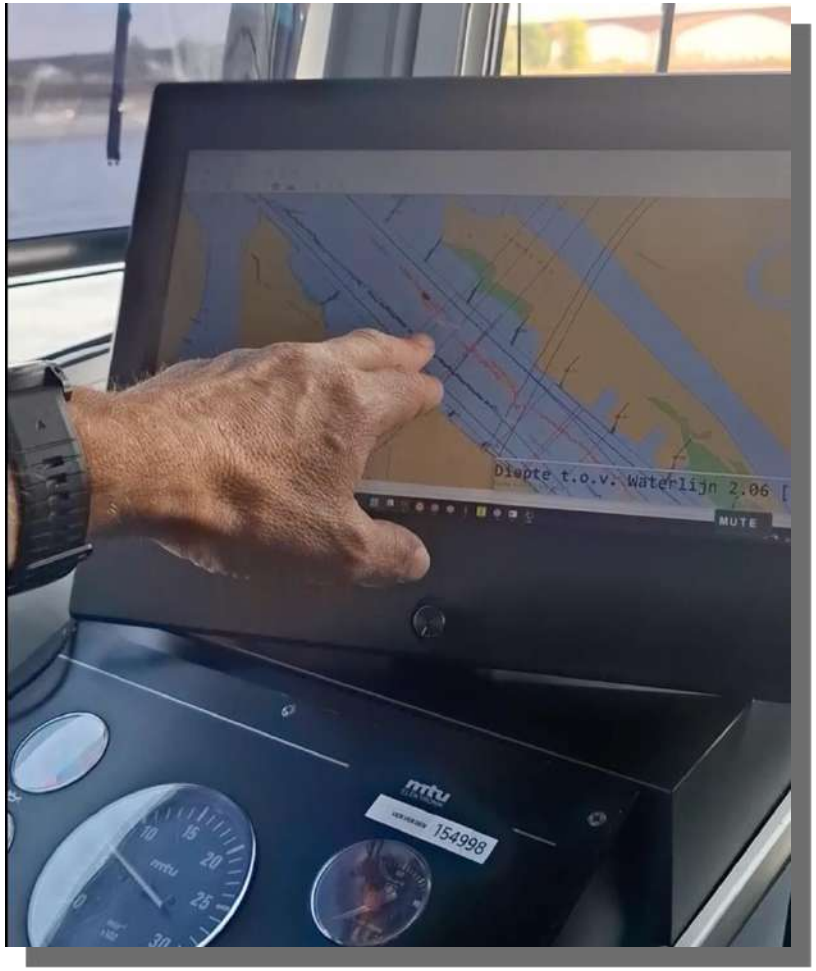
- Three experiments
  - Preprocessing:
    - Workflow of simulations
    - Simplified derivative model
    - Input generation
  - Numerical core:
    - New solver
    - Reuse and extend solver
    - Combine different numerical approaches
  - Postprocessing
    - Rebuild old visualization
    - New programming language

- Tools used:
  - Gemini app / CLI / agy
  - Claude Code / CLI
  - Github Copilot / CLI
  - OpenCode CLI
- Models used:
  - Gemini Pro / Flash
  - Claude Opus / Sonnet
  - GPT OSS 120B
  - GPT Luna / Sol / Codex
- Tools:
  - MCP: Git / Numerical Skills

Experiment 1: Virtual Towing Tank

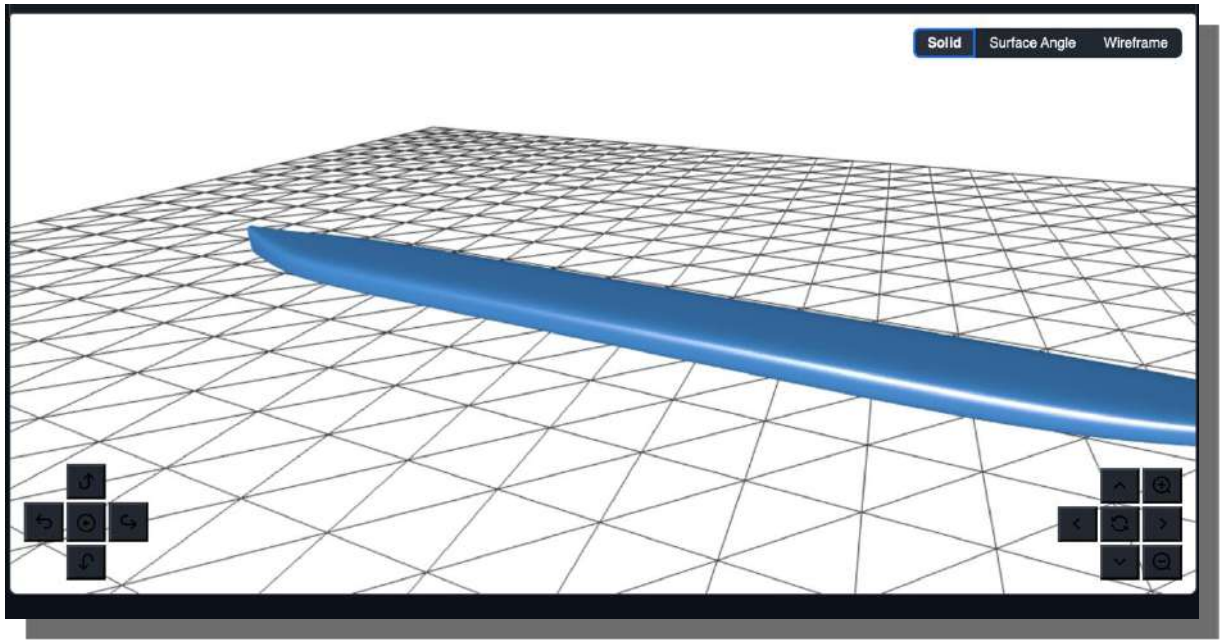
Goals:

- Towing tank model for undeep water
- Open-source solver
- Automated workflow to generate emulator input
- Support local and HPC development
- Visualizations in paraview



src: ad.nl

Hull design: Blender




```
# Target Grid Size: 9 U stations x 5 V points
# We will create a 4x4 primitive and subdivide to get >= 9x5
# Subdivide (2 cuts) -> 10x10 grid.
# We will use the first 9 rows and 5 columns, or resample?
# Resample logic: Map our 9 defined stations to the 10 available U-rows.
# Or just use the 10 rows!

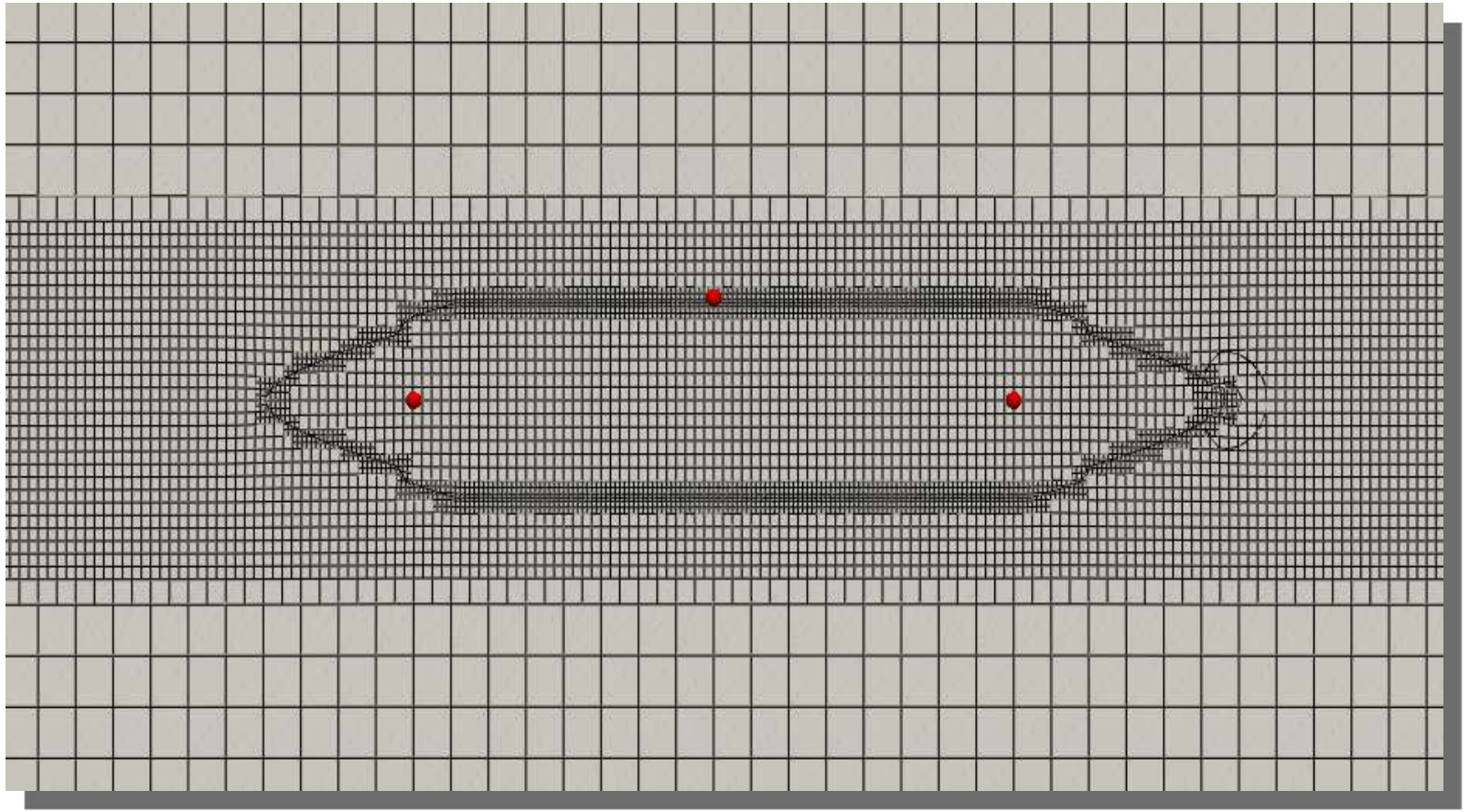
# 1. Create Primitive
# Location 0,0,0. Radius 1.
bpy.ops.surface.primitive_nurbs_surface_surface_add(radius=1, location=(0,0,0))
obj = bpy.context.active_object
obj.name = "Barge_Surface"

# 2. Subdivide
bpy.ops.object.mode_set(mode='EDIT')
bpy.ops.curve.select_all(action='SELECT')
# 2 cuts -> 10x10 grid (4 points -> 3 segs -> 2 cuts/seg -> 4 + 6 = 10)
bpy.ops.curve.subdivide(number_cuts=2)
bpy.ops.object.mode_set(mode='OBJECT')

spline = obj.data.splines[0]

# CRITICAL: Enable endpoints so surface reaches the control points
spline.use_endpoint_u = True
```

Schematization



OpenFOAM template system (Jinja)

jax-vessels / templates / floating_hull / system / include / **dynamicMesh.6dof.empty**

SiggyF feat: Implement hydrostatic verification safeguard and stabilize 6DoF...

Code | Blame | 38 lines (31 loc) · 743 Bytes

```
dynamicFvMesh   dynamicMotionSolverFvMesh;

motionSolverLibs ( "libsixDoFRigidBodyMotion.so" );
solver          sixDoFRigidBodyMotion;

sixDoFRigidBodyMotionCoeffs
{
    patches         (hull);
    innerDistance   5;
    outerDistance   20;

    mass            2010000;
    centreOfMass    (67.5 0 2);
    momentOfInertia (25000000 3000000000 3000000000);

    accelerationRelaxation 0.05;
    accelerationDamping    0.8;

    solver
    {
        type symplectic;
    }
```

```
{% set foam_class = 'dictionary' %}
{% set foam_object = 'snappyHexMeshDict' %}
{% include 'header.j2' %}

castellatedMesh true;
snap            true;
addLayers       false;

geometry
{
    "{{ case_name }}.stl"
    {
        type triSurfaceMesh;
        file "{{ case_name }}.stl";
        name hull;
    }
}
```

HPC + local workflow: docker + apptainer

```
if [ -z "$CASE_DIR" ]; then
    echo "Usage: sbatch submit_job.sh <case_dir>"
    exit 1
fi

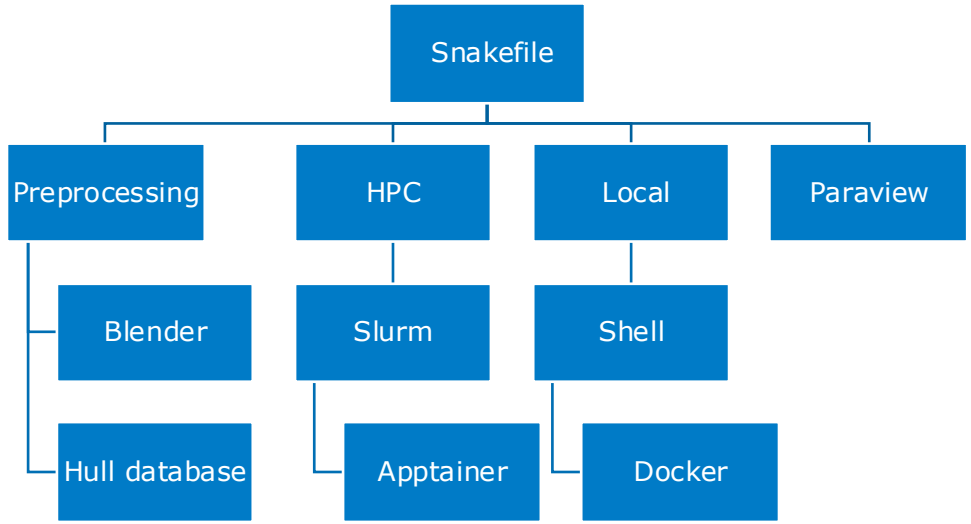
echo "Running case $CASE_DIR on $(hostname)"

# Ensure log directory exists
mkdir -p logs

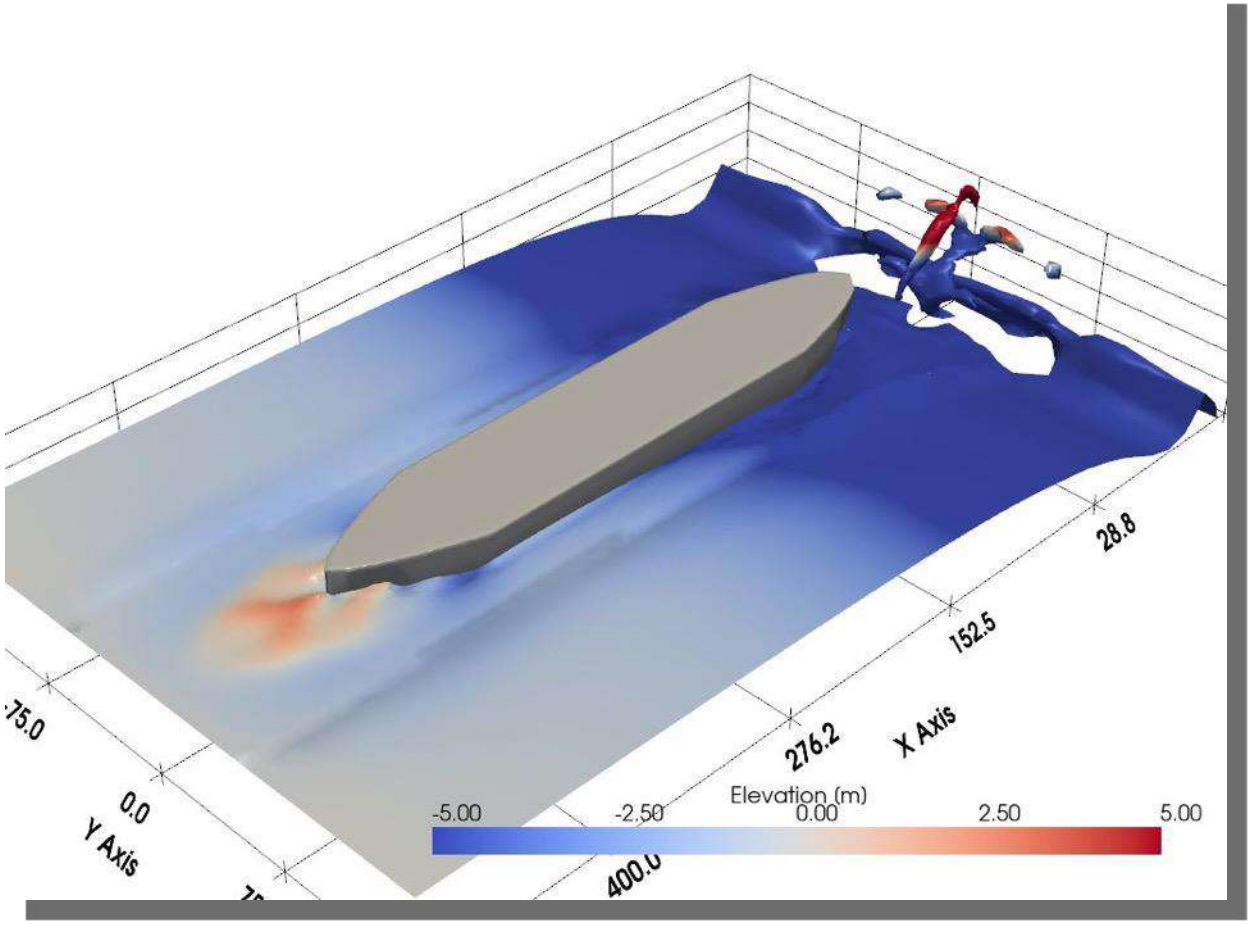
# Run OpenFOAM Solver
echo "Starting Solver..."
apptainer exec --bind /scratch:/scratch "$SOLVER_IMAGE" /bin/bash -c "sou

# Run Data Extraction
# We use the python-extract image to generate CSVs
echo "Starting Data Extraction..."
# We assume the project root contains scripts/extract_results.py
# Since we cd into $CASE_DIR (e.g. cases/dtc), the script is at ../../scr
apptainer exec --bind /scratch:/scratch "$EXTRACT_IMAGE" /bin/bash -c "cd

echo "Job Complete."
```



Paraview visualisations



Emulator workflow

