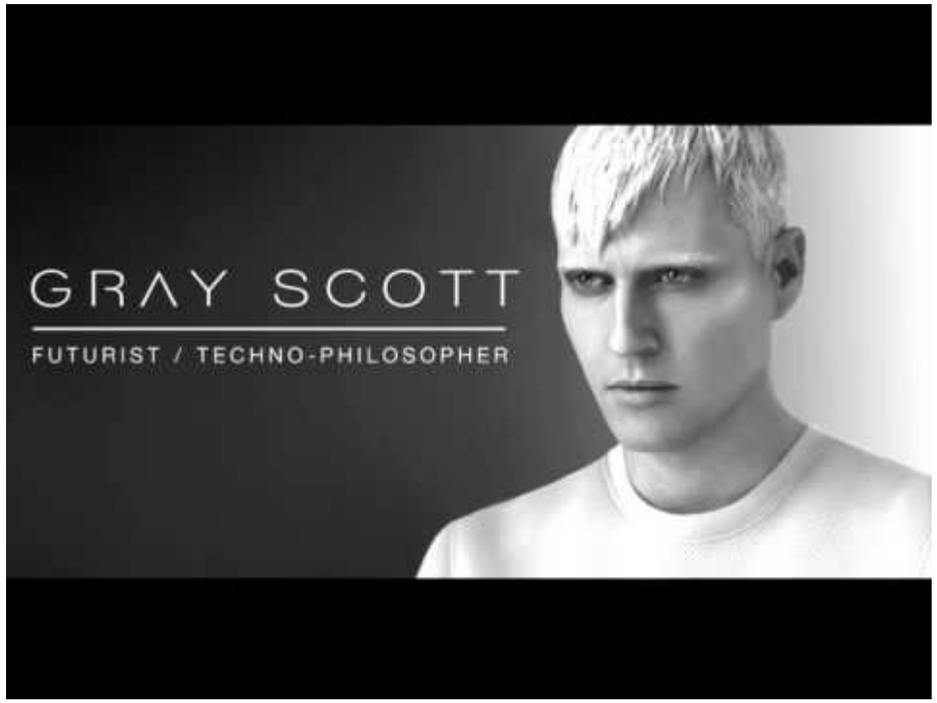
Scaling up: Gray-Scott model

Hadrien Grasland 2024-11-26

Previously…

- Introduced Rust for numerical computing
  - C++ as the main competition, how they compare
  - Rust basics for numerics, from « let » to « dot »
- Not bad at this small scale
  - …but we already have dot() at home
  - Can Rust handle more realistic computations?

Enter Gray-Scott



...whoops, not that one!

Gray-Scott reaction-diffusion model

- Basic idea: Chemistry as a cellular automaton (CaaCA?)
  - If U is surrounded by two V, then U becomes V
- To this U + 2V ⟶ 3V **reaction**, we add…
  - A **source** term that adds more U (else we'll run out of it)
  - A **drain** term that removes some V* (else we'll drown in it)
  - **Diffusion** terms for U and V (nature hates inhomogeneity)
- …and go to the **continuum** limit (~10^{23} particles is too much)

* Technically turns it into a non-reactive species P that we do not study.

Wild differential equations appear!

$$\frac{\partial u}{\partial t} = -uv^2 + F(1-u) + D_u \Delta u$$

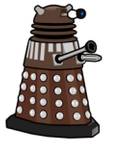
$$\frac{\partial v}{\partial t} = uv^2 - (F+k)v + D_v \Delta v$$

where

u, v = concentrations of U and V
F = Feed rate (replenish U)
k = Kill rate (destroy V)
D_u, D_v = diffusion rate of U and V

How do we solve for u(t) and v(t)?

- Formal analysis: No idea lol.
- Numerical analysis: **Approximate! Approximate!**

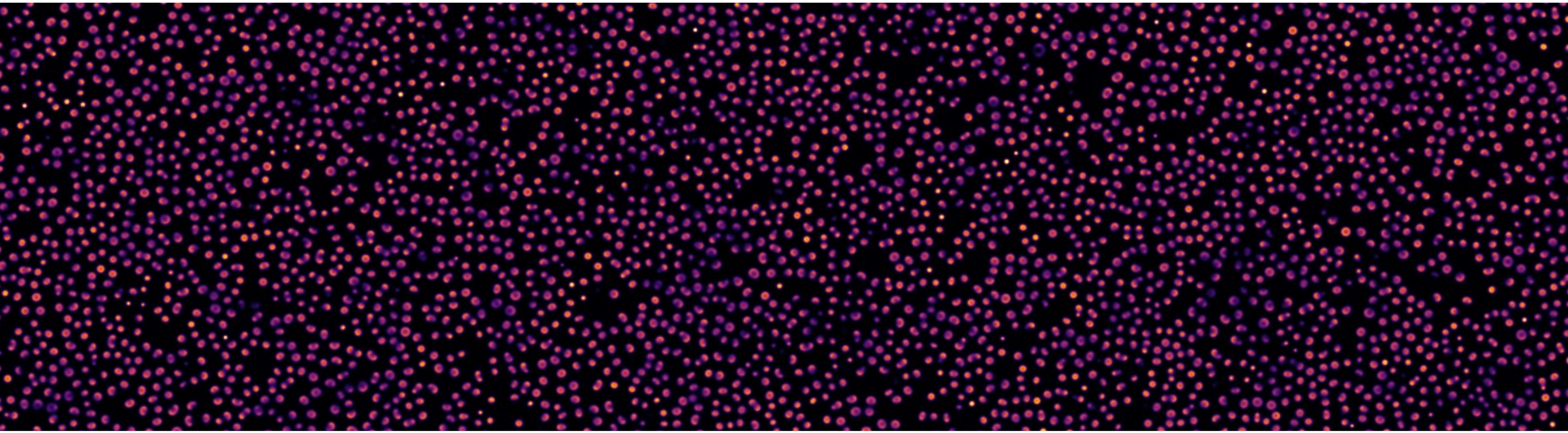


Not a numerical analysis course…

- …so we use the simplest scheme that could possibly work
- Discretize space via **finite differences:**
  - Consider a regular grid (2D for simplicity)
  - Only compute u or v values on points of that grid
  - Spatial derivative ~ difference of grid neighbors / grid step
- Discretize time via **Euler's method:** $u(t+\Delta t) \simeq u(t) + \Delta t \frac{\partial u}{\partial t}(t)$

Main goal: Pretty pictures

- Save precise v(t) data into an HDF5 data cube
- Post-process HDF5 cube 2D slices into nice-looking PNGs



Secondary goals

- Control some parameters via CLI arguments
- Report progress as we go through the time steps
- Get as close as possible to peak CPU performance

Let’s get started!