1 User Manual for Simulator_GUI

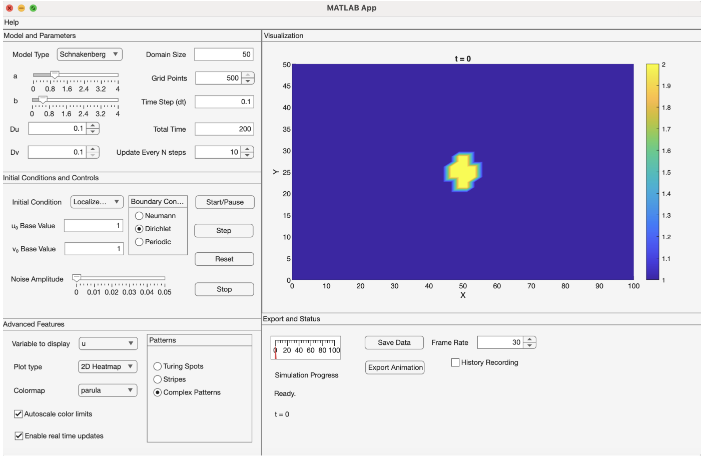

Figure 1: Screenshot of the Simulator_GUI during a typical Schnakenberg simulation.

1. What the app simulates (theory in brief)

This app solves two–species reaction–diffusion (RD) systems on a 2D rectangular domain Ω:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D_u \Delta u + f(u, v; \theta), \\ \frac{\partial v}{\partial t} &= D_v \Delta v + g(u, v; \theta),\end{aligned} \tag{1}$$

where $u(\mathbf{x}, t), v(\mathbf{x}, t)$ are concentrations, $D_u, D_v > 0$ are diffusion coefficients, and Δ is the Laplacian. The **same solver pipeline** is used for all models; only (f, g) changes.

Schnakenberg model

$$f = a - u + u^2 v, \qquad g = b - u^2 v \tag{2}$$

with parameters $a > 0, b > 0$. A homogeneous steady state is

$$u_0 = a + b, \qquad v_0 = \frac{b}{(a+b)^2}.$$

For suitable (a, b, D_u, D_v), diffusion can destabilize the uniform state and create **Turing patterns**.

Brusselator model

$$f = A - (B+1)u + u^2 v, \qquad g = Bu - u^2 v \tag{3}$$

with parameters $A > 0, B > 0$. Homogeneous steady state:

$$u_0 = A, \qquad v_0 = \frac{B}{A}.$$

Depending on (A, B) and diffusion contrast, the system can exhibit stationary patterns and oscillatory regimes.

Heat test (verification mode)

To validate diffusion and boundary handling, set reactions to zero:

$$f = 0, \quad g = 0 \quad \Rightarrow \quad \frac{\partial u}{\partial t} = D_u \Delta u, \quad \frac{\partial v}{\partial t} = D_v \Delta v.$$

With a localized initial bump, you should observe smoothing/spreading over time.

What patterns you should expect

- **Spots:** isolated high/low regions (activator peaks).
- **Stripes/Labyrinths:** elongated bands.
- **Complex/Mixed:** combination of spots and stripes; sensitive to parameters and noise.

A large diffusion contrast (often $D_v \gg D_u$) commonly favors Turing-like structures.

2. Running the simulation (workflow)

1. Choose **Model Type**: `Schnakenberg`, `Brusselator`, or `Heat`.
2. Set parameters: **a,b** (or A, B), **Du,Dv**, domain and discretization, and time settings.
3. Choose **Initial Condition** and **Noise Amplitude**.
4. Choose **Boundary Conditions**.
5. Choose **Variable to display**, **Plot type**, and **Colormap**.
6. Click **Start/Pause** to run (timer-based), or **Step** for manual stepping.

3. Controls and what they do

Model and Parameters panel

- **Model Type**: selects (f, g) in Eq. (1). `Heat` disables reactions for diffusion-only testing.
- **a, b sliders**: reaction parameters (Schnakenberg/Brusselator). Changing these shifts the regime (uniform vs patterns vs oscillations).
- **Du, Dv**: diffusion coefficients. Increasing D smooths gradients faster. Strong contrast D_v/D_u can trigger pattern formation.
- **Domain Size**: physical size of the rectangle. Larger domains allow more wavelengths (more spots/stripes).
- **Grid Points**: spatial resolution. Higher improves detail but increases runtime/memory.

- **dt**: time step. Too large can reduce accuracy or stability; too small slows simulation.
- **Total Time**: simulation end time.
- **Update Every N steps**: rendering/refresh interval (performance vs smoothness).

Initial Conditions and Controls panel

- **Initial Condition**:
  - `Uniform`: starts from constant u_0, v_0 (often add noise to trigger patterns).
  - `Random`: adds random perturbations; helps seed pattern growth.
  - `Localized Spot`: seeds a single region (useful to see spreading / initiation).
- **$\mathbf{u_0}$ Base Value, $\mathbf{v_0}$ Base Value**: baseline initial values (often set near steady state).
- **Noise Amplitude**: magnitude of random perturbation; larger noise seeds faster but may distort fine structures.
- **Boundary Conditions**:
  - **Neumann (no-flux)**: $\partial u/\partial n = \partial v/\partial n = 0$; patterns reflect at walls.
  - **Dirichlet**: fixed boundary values (here typically 0); can damp patterns near boundaries.
  - **Periodic**: opposite boundaries are tied; good for "infinite tiling" behavior and cleaner pattern statistics.
- **Buttons**:
  - **Start/Pause**: rebuilds and runs with a timer; pause stops the timer.
  - **Step**: advances by one batch and updates the plot.
  - **Reset**: restores defaults and rebuilds.
  - **Stop**: stops the simulation timer immediately.

Advanced Features panel

- **Pattern presets** (`Turing Spots`, `Stripes`, `Complex Patterns`): auto-sets parameter combinations (and may lock/unlock inputs) to reproduce typical regimes.
- **Enable real time updates**: updates the plot every step (smoother, but slower).
- **Autoscale color limits**: automatically rescales color range; turning off keeps a fixed scale (better for comparing time frames).
- **Variable to display**:

$$z = \begin{cases} u & \text{display activator} \\ v & \text{display inhibitor} \\ u+v & \text{combined field} \\ \sqrt{u^2+v^2} & \text{magnitude} \end{cases}$$

- **Plot type** (how your choice changes the visualization):
  - **2D Heatmap**: fastest; color-coded scalar field $z(\mathbf{x})$ (best for patterns).
  - **3D Surface**: height $= z$; visually intuitive but heavier to render.
  - **Contour**: isolines / filled contours; highlights wavelengths and boundaries of spots/stripes.
  - **2D Quiver (if enabled)**: vector arrows (typically needs a vector field; if used with scalar z, define what arrows represent).
- **Colormap**: affects contrast and readability; does not change physics.

Export and Status panel

- **Simulation Progress**: percentage of t/Total Time.
- **Status + Current time**: run state and current t.
- **History Recording**: must be enabled to save/export; stores frames and solution snapshots.
- **Save Data**: saves current (x, y, u, v) as CSV or full MAT (with parameters/history).
- **Export Animation**: writes AVI/MP4 from recorded frames. **Frame Rate** controls playback speed.

4. Practical tips (so it "looks right")

- For Schnakenberg, start near (u_0, v_0) and add small noise; patterns usually emerge gradually.
- If nothing happens: increase diffusion contrast, increase Total Time, or add small noise.
- If plot flickers: enable autoscale (or disable it for consistent comparisons).
- If performance is slow: reduce Grid Points, increase Update Every N steps, or disable real-time updates.
- For verification: choose `Heat` and ensure the bump smooths out without oscillations or patterning.

5. Quick reference (one table)

Setting	Effect on result / plot
Model Type	selects kinetics (f, g); `Heat` = diffusion-only
a, b (or A, B)	changes reaction balance; controls regime (uniform/pattern/oscillation)
D_u, D_v	smoothing speed; contrast can induce Turing patterns
Domain Size	more space ⇒ more pattern wavelengths
Grid Points	resolution/detail vs runtime
dt	accuracy/stability vs speed
Initial Condition + Noise	seeds instability; larger noise = faster onset
Boundary Conditions	affects edge behavior and global symmetry
Variable to display	what scalar field is rendered
Plot Type	heatmap/surface/contour change how z is visualized
Autoscale	dynamic color range vs fixed comparisons
History Recording	enables Save/Export