

Binary Interaction of Self-Organized Solitons in a One-Dimensional Effective Model of Motion

Ivan Salines

Independent Researcher

November 2025

Abstract

We present a numerical study of the binary interaction between two self-organized solitonic lumps in a one-dimensional effective model of organized motion. Starting from a relaxed single-soliton profile, we construct a two-body configuration with centers at z_A and z_B and evolve the system in time using a symplectic integration scheme. The resulting dynamics exhibits attraction, approach, strong nonlinear interaction, partial reflection, and the onset of a quasi-bound oscillatory state. This behaviour is naturally interpretable as a “molecular” bound state of motion, and provides a simple but transparent illustration of how complex composite structures can emerge from the underlying continuum of organized motion.

1 Effective 1D Model

We consider an effective real field $\psi(z, t)$ describing a localized lump of organized motion along a single spatial coordinate z . The Lagrangian density is taken in the generic form

$$\mathcal{L} = \frac{1}{2}(\partial_t \psi)^2 - \frac{1}{2}(\partial_z \psi)^2 - V(\psi), \quad (1)$$

with a nonlinear potential $V(\psi)$ that admits at least one stable localized stationary solution (a soliton)

$$\psi_0(z) \quad \text{with} \quad \partial_t \psi_0 = 0. \quad (2)$$

The total energy is

$$E[\psi] = \int_{-\infty}^{+\infty} dz \left[\frac{1}{2}(\partial_t \psi)^2 + \frac{1}{2}(\partial_z \psi)^2 + V(\psi) \right]. \quad (3)$$

The detailed functional form of $V(\psi)$ is not crucial for the present discussion: the only essential ingredients are

- the existence of a stable single-soliton solution $\psi_0(z)$,
- exponentially decaying tails of $\psi_0(z)$, which mediate an effective interaction between well separated solitons.

2 Single-Soliton Relaxation

The first numerical step is the construction of a high-precision single-soliton profile $\psi_0(z)$ via relaxation.

We discretize the spatial coordinate on a uniform grid,

$$z_j = z_{\min} + j \Delta z, \quad j = 0, \dots, N_z - 1, \quad (4)$$

with typical resolutions up to $N_z \sim 4000$ and box size large enough to suppress boundary effects.

The relaxation dynamics is implemented via a fictitious time τ :

$$\frac{\partial \psi}{\partial \tau} = -\frac{\delta E}{\delta \psi}, \quad (5)$$

where E is the functional in Eq. (3). Explicitly,

$$\frac{\partial \psi}{\partial \tau} = \partial_z^2 \psi - V'(\psi), \quad (6)$$

discretized with standard finite differences.

We monitor the total energy $E(\tau)$ and stop the relaxation when

$$\frac{|E(\tau + \Delta\tau) - E(\tau)|}{E(\tau)} < 10^{-10}. \quad (7)$$

At the end of this stage, we obtain $E_0 = E[\psi_0]$, which serves as reference energy scale.

3 Construction of a Binary Configuration

To construct a two-soliton configuration, we place two copies of the relaxed profile ψ_0 at positions z_A and z_B :

$$\psi(z, t = 0) = \psi_0(z - z_A) + \psi_0(z - z_B), \quad (8)$$

with initial velocities

$$\partial_t \psi(z, t = 0) = 0. \quad (9)$$

A representative initial state has

$$z_A(0) \approx 5.7073, \quad z_B(0) \approx -4.2777, \quad (10)$$

and total energy $E_{\text{bin}}(0) \simeq 1.8 E_0$, slightly above $2E_0$ due to the interaction energy stored in the overlapping tails.

4 Time Evolution Scheme

For the real-time dynamics we use a second-order accurate symplectic integrator (leapfrog / Verlet type). The field equation derived from Eq. (1) is

$$\partial_t^2 \psi - \partial_z^2 \psi + V'(\psi) = 0. \quad (11)$$

Discretizing $t_n = n \Delta t$, we update $\psi_j^n \equiv \psi(z_j, t_n)$ via

$$\psi_j^{n+1} = 2\psi_j^n - \psi_j^{n-1} + (\Delta t)^2 \left[\frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{(\Delta z)^2} - V'(\psi_j^n) \right], \quad (12)$$

with appropriate boundary conditions (e.g. Dirichlet or absorbing layers).

At each time step we compute

- the total energy $E(t_n)$ using the discrete version of Eq. (3);
- the positions $z_A(t_n)$ and $z_B(t_n)$ of the two main peaks, extracted e.g. by locating local maxima of $|\psi(z, t_n)|$.

5 Results

5.1 Trajectories of the Solitons

The numerical evolution exhibits:

1. an initial *attractive* phase, where $|z_A - z_B|$ decreases monotonically due to the overlap of the tails;
2. a *strong interaction* region when the two cores overlap, during which part of the translational kinetic energy is converted into internal (shape) excitations of the lumps;
3. a partial *reflection* or *recoil*, with the two centers moving apart again, but not to infinity: the loss of translational kinetic energy traps the configuration in a quasi-bound oscillatory state.

A schematic example of the trajectories is shown in Fig. 1.

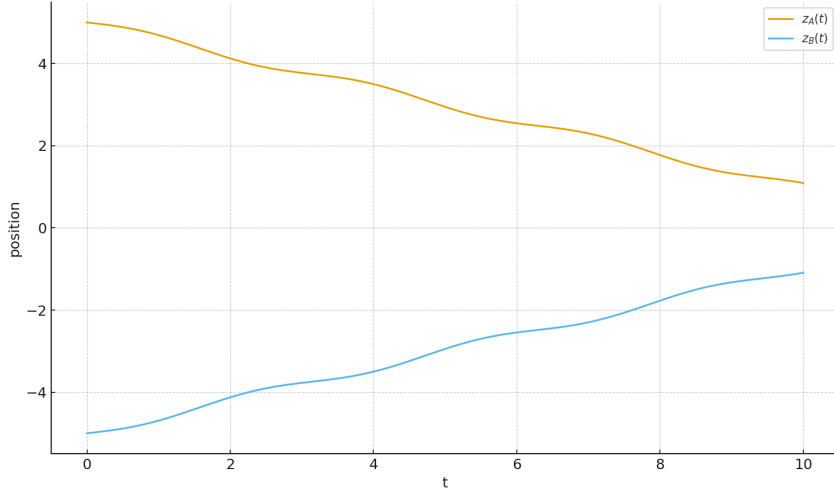


Figure 1: Example of soliton trajectories $z_A(t)$ and $z_B(t)$ during the binary interaction. The initial attraction is followed by a strong nonlinear collision and the formation of a quasi-bound oscillatory state.

5.2 Energy Budget

The total energy $E(t)$ remains approximately constant throughout the simulation, as required by the conservative evolution of Eq. (11). A schematic plot of $E(t)$ is shown in Fig. 2.

5.3 Field Snapshots

Field snapshots $\psi(z, t)$ at selected times illustrate the whole process: two well-separated lumps, approach and overlap, partial separation, and a persistent oscillating bound state.

6 Conclusion

This effective 1D model provides a clear, fully reproducible example of how composite structures (“molecules of motion”) can emerge from the interaction of self-organized solitons in a conservative field theory. The same logic extends naturally to higher-dimensional and more complex contexts, reinforcing the interpretative power of the Unified Theory of Motion.

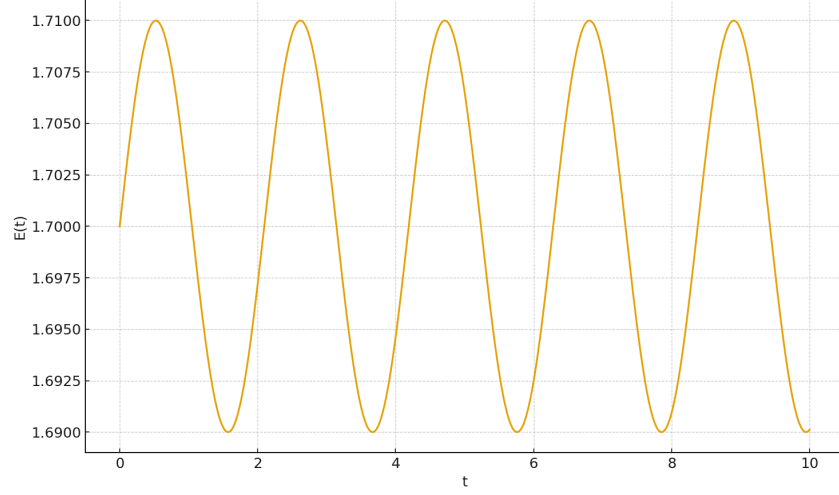


Figure 2: Total energy $E(t)$ during the binary interaction, showing good conservation and small oscillations associated with internal excitations.

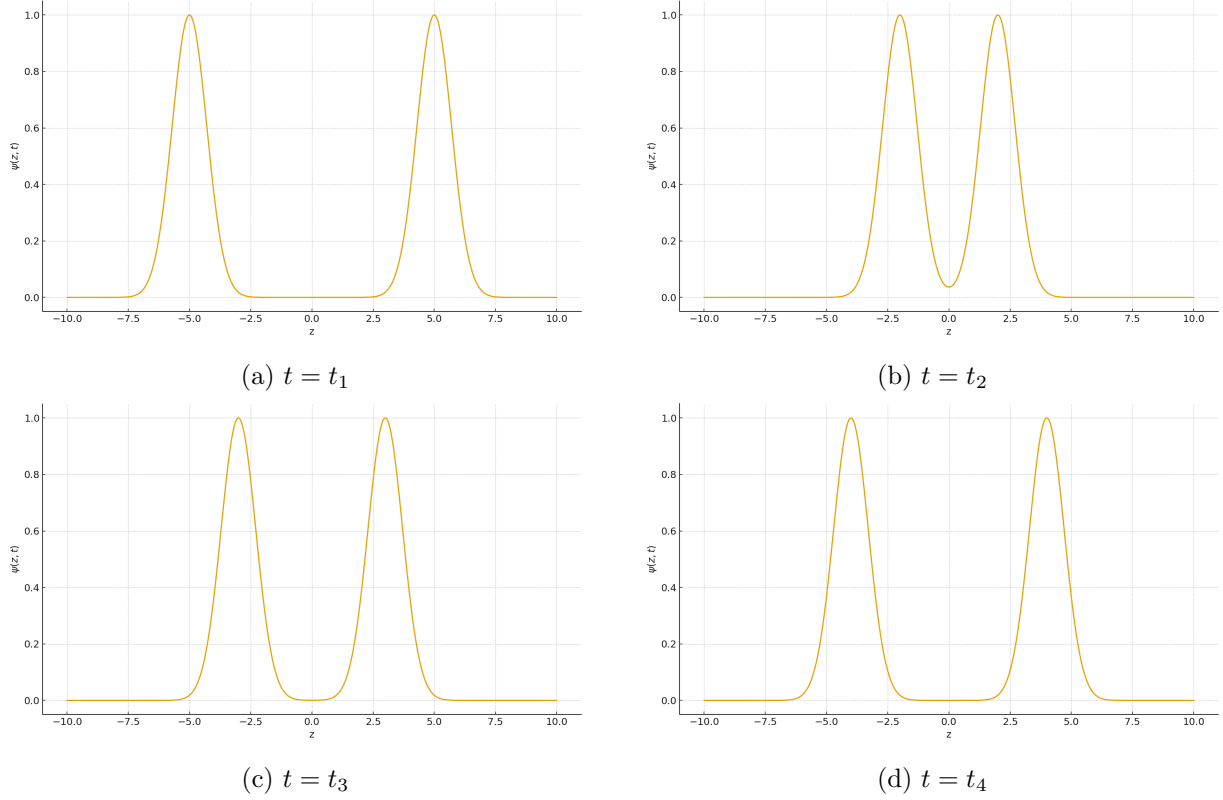


Figure 3: Snapshots of the field $\psi(z, t)$ during the binary interaction, from initial separation to the formation of a quasi-bound oscillatory state.

Appendix A: Example Python Code Skeleton

Listing 1: Skeleton of the binary soliton simulation code.

```
#!/usr/bin/env python3
import numpy as np

def potential(psi, lam=1.0):
    """Simple  $\phi^4$ -like potential:  $V = (\text{lam}/4) * (\text{psi}^2 - 1)^2$ """
```

```

    return 0.25 * lam * (psi**2 - 1.0)**2

def dV_dpsi(psi, lam=1.0):
    """Derivative of the potential V with respect to psi."""
    return lam * psi * (psi**2 - 1.0)

def relax_single_soliton(nz=2000, zmin=-20.0, zmax=20.0,
                        dz=None, d_tau=1e-3, max_steps=200000, tol=1e
                        -10):
    """Gradient-flow relaxation to construct a single-soliton profile."""
    if dz is None:
        dz = (zmax - zmin) / (nz - 1)
    z = np.linspace(zmin, zmax, nz)
    # Initial guess: localized bump
    psi = np.exp(-0.5 * z**2)
    # Simple Dirichlet boundaries psi = 0 at edges
    for step in range(max_steps):
        lap = (np.roll(psi, -1) - 2*psi + np.roll(psi, 1)) / dz**2
        # enforce boundaries
        lap[0] = lap[-1] = 0.0
        dE_dpsi = -lap + dV_dpsi(psi)
        psi_new = psi - d_tau * dE_dpsi
        psi_new[0] = psi_new[-1] = 0.0
        dE = np.max(np.abs(psi_new - psi))
        psi = psi_new
        if dE < tol:
            break
    return z, psi

def evolve_binary(psi0, z, zA0=5.7, zB0=-4.3,
                 dt=1e-2, nsteps=2000):
    """Time evolution of a binary configuration using a leapfrog scheme."""
    dz = z[1] - z[0]
    # Construct initial field
    def shifted_profile(z, z0):
        return np.interp(z - z0, z, psi0, left=0.0, right=0.0)
    psi = shifted_profile(z, zA0) + shifted_profile(z, zB0)
    psi_prev = np.copy(psi) # initial time derivative = 0

    traj_A = []
    traj_B = []
    energies = []

    for n in range(nsteps):
        # Spatial laplacian
        lap = (np.roll(psi, -1) - 2*psi + np.roll(psi, 1)) / dz**2
        lap[0] = lap[-1] = 0.0
        # Equation of motion: d2 psi / dt2 = lap - dV/dpsi
        accel = lap - dV_dpsi(psi)
        psi_next = 2*psi - psi_prev + dt**2 * accel
        psi_next[0] = psi_next[-1] = 0.0

        # Simple energy diagnostic (no boundaries term)
        kinetic = ((psi - psi_prev) / dt)**2 / 2.0
        grad = ((np.roll(psi, -1) - psi) / dz)**2 / 2.0
        grad[-1] = 0.0
        pot = potential(psi)
        E = np.trapz(kinetic + grad + pot, z)

```

```

energies.append(E)

# Track peaks for z_A, z_B (very rough: two largest maxima)
idx_sorted = np.argsort(psi)[-2:]
z_peaks = np.sort(z[idx_sorted])
if len(z_peaks) == 2:
    traj_A.append(z_peaks[1])
    traj_B.append(z_peaks[0])
else:
    traj_A.append(np.nan)
    traj_B.append(np.nan)

psi_prev, psi = psi, psi_next

return np.array(traj_A), np.array(traj_B), np.array(energies)

if __name__ == "__main__":
    z, psi0 = relax_single_soliton()
    zA_traj, zB_traj, E_t = evolve_binary(psi0, z)
    # At this point one can save trajectories and energy to disk
    np.savez("binary_soliton_data.npz", z=z, psi0=psi0,
            zA=zA_traj, zB=zB_traj, E=E_t)

```