SFV–dSB: First-Principles Two-Field Bounce (Reproducible Benchmark)

This repository contains code and minimal configurations to reproduce the **O(4)** Coleman bounce in the **SFV/dSB** model on the **negative-ϕ false-vacuum branch**, including the exact PNG figures used in the manuscript.

> **Benchmark:** $v = 9.0 \times 10^{-5} M_{\rm Pl}$, $\lambda = 1.3 \times 10^{-4}$, $\lambda_\Phi = 0.1$, $g_{\rm portal} = 2.0$ $\Rightarrow$ $S_E \approx 1424$.
>
> **Repository placeholder:** *[URL/DOI to be inserted upon posting]*.

Contents

```
SFV-dSB-bounce/
├─ README.md                  # this file
├─ CITATION.cff               # citation metadata (fill in once DOI is known)
├─ LICENSE                    # choose one (e.g., MIT/BSD-3-Clause)
├─ env/
│  ├─ requirements.txt        # pip environment (NumPy/SciPy/Matplotlib)
│  └─ conda-environment.yml   # optional conda environment
├─ configs/
│  ├─ config_benchmark.yaml        # single-run (SE ≈ 1424) config
│  ├─ config_scan.yaml             # corridor/parameter scan config
│  ├─ config_fieldspace_plot.yaml  # field-space contour + path figure
│  └─ schema.md                    # short spec of all config keys
├─ scripts/
│  ├─ goldenRunDetails_v4f.py      # benchmark runner (SciPy solve_bvp)
│  ├─ bounceScan_v3.py             # 1D scan (e.g., in ρ = vΦ/v)
│  ├─ fieldspace_plot.py           # contours of V(Φ,φ) + bounce path
│  └─ utils.py                     # shared helpers (I/O, Jacobians, etc.)
└─ runs/
   ├─ benchmark_YYYYMMDD_HHMM/     # auto-created on runs
   └─ scans/                       # auto-created for scan outputs
```

Quick start

1) Create the environment

Option A: pip

```
python -m venv .venv
. .venv/Scripts/activate   # Windows
# source .venv/bin/activate   # macOS/Linux
pip install -r env/requirements.txt
```

Option B: conda (optional)

```
conda env create -f env/conda-environment.yml
conda activate sfv-dsb-bounce
```

Pinned tools (minimal):
- Python 3.10
- numpy==1.26.
- *scipy==1.14.*
- matplotlib==3.8.*

> We print Python/NumPy/SciPy versions at runtime into `solver_log.txt` for provenance.

Reproduce the benchmark (SE ≈ 1424)

Runs the solver with the **negative-ϕ branch** enforced, writes outputs to a timestamped folder in `runs/`.

```
python scripts/goldenRunDetails_v4f.py --config configs/config_benchmark.yaml
```

Outputs (in `runs/benchmark_YYYYMMDD_HHMM/`):

- `solver_log.txt` — iteration log (residuals, node counts, rmax, etc.)
- `params_resolved.json` — fully resolved parameters actually used (after any derived quantities)
- `results.csv` — summary row(s): SE, R0, w, nodes, tol, etc.
- `profile_phiPhi.png` — Φ(r) and ϕ(r) profiles (paper figure)
- `action_density.png` — $I(\tilde{r}) = 2\pi^2 \tilde{r}^3 [\cdots]$ (paper figure)
- `fieldspace_contours_path.png` — contours of $V(\Phi, \phi)$ + path (paper figure)

The three PNG names match the manuscript.

Reproduce the field-space contour + path figure only

If you just want the field-space plot (e.g., to regenerate the figure quickly):

```
python scripts/fieldspace_plot.py --config configs/config_fieldspace_plot.yaml
```

Writes `fieldspace_contours_path.png` to the configured output path.

Run a parameter scan (ρ corridor)

Scans in $\rho \equiv v_\Phi/v$ (or another parameter you set in the config), classifies outcomes (converged nontrivial / converged trivial / fail), and records $S_E(\rho)$ when converged.

```
python scripts/bounceScan_v3.py --config configs/config_scan.yaml
```

Outputs (default): - `runs/scans/rho_scan_results.csv` — one row per point (ρ, converged flag, SE, diagnostics) - Optional per-point logs/PNGs if `record_per_point: true` is enabled in the config.

Configuration files

All runs are driven by human-readable **YAML** configs located in `configs/`. This separates *what you ran* from code.

- `config_benchmark.yaml` — parameters and solver knobs for the **SE ≈ 1424** point.
- `config_scan.yaml` — defines the scanned parameter and grid, plus per-point solver constraints (tol, retries).
- `config_fieldspace_plot.yaml` — grid ranges, normalization, and styling for $V(\Phi, \phi)$ contours; overlays the path if available.
- `schema.md` — a 2–3 page key/value reference (types, defaults, units, and how each key is used in the code).

  Branch enforcement: set `branch_selection.false_vacuum_branch: "negative_phi"` and `branch_selection.enforce_branch: true`
  The scripts will keep the initial path on the negative-ϕ half-plane and apply a tiny offset/clip to prevent flips.

Reproducibility checklist

- Deterministic configs checked into `configs/`.

- Scripts print **Python/NumPy/SciPy/OS** versions into `solver_log.txt`.
- All derived numbers (e.g., bias from the rule in the paper) are recorded in `params_resolved.json`.
- Mesh/box/tolerance stability notes:
- Default `tol=1e-5`, `rmax=100`, `max_nodes=5e5`.
- We validated S_E stability at the benchmark under modest mesh and $r_{\max}$ changes ($\leq$1% variation).

Paper figures (PNG)

The manuscript expects the following PNG names (these are produced by the benchmark run):

- `profile_phiPhi.png`
- `action_density.png`
- `fieldspace_contours_path.png`

If you prefer a different folder, adjust `outputs.outdir` in `config_benchmark.yaml`.

Troubleshooting

- **No convergence / many nodes added**
  Increase `rmax` (e.g., 150–200), relax `tol` slightly (e.g., 2e-5), or widen the seed wall (`R0`, `w`) in `initial_guess`.
  Thick-wall cases may benefit from smaller continuation steps in `solver.continuation.schedule`.

- **Branch flips to positive-ϕ**
  Ensure `branch_selection.enforce_branch: true` and that `initial_path` logic in `utils.py` clamps small negative ϕ for the first segment.

- **Different SciPy version**
  We pin SciPy 1.14.* in `env/requirements.txt`. Other versions may work but are not guaranteed identical.

- **Long runs / memory**
  Reduce `max_nodes`, or increase step size in the continuation schedule. For scans, set `scan.per_point.max_time_s`.

How to cite

Once the preprint DOI is available, we'll add it here and to `CITATION.cff`. For now, please cite the preprint:

Hoffmann, S. *First-Principles Two-Field Bounce in the SFV/dSB Model and a Quantitative Hierarchy of Origins* (preprints.org, 2025). DOI: **TBD**.

License

Choose a permissive license unless you have constraints (MIT or BSD-3-Clause are common). Put it in `LICENSE` and reference it here.

Contact

Questions, suggestions, or reproduction issues: open a GitHub issue or email **[your contact here]**.