Vesicle Shape Simulation Research Project — Comprehensive Summary and Task Checklist

---

Project Overview

This research aims to develop and analyze a computational framework for simulating axisymmetric vesicle shapes governed by bending energy and boundary constraints. The simulation platform serves as both a **physics-based model** and a **data generation pipeline** for downstream machine learning and shape analysis tasks.

Core Goals

1. **Shape Simulation:** Accurately solve vesicle equilibrium shapes across parameter regimes defined by bending modulus, spontaneous curvature, and area/volume constraints.
2. **Dataset Creation:** Generate a labeled, high-resolution dataset of stable and unstable shape configurations.
3. **Energy Landscape Exploration:** Characterize energy minima, bifurcations, and stability regions.
4. **ML Integration:** Develop surrogate models and classifiers for rapid shape prediction and regime identification.
5. **Scientific Insights:** Quantify transitions between shape classes (oblate, prolate, pear-shaped, stomatocyte) and their dependence on mechanical parameters.

Current Status (Simulation Framework)

- **Solver:** Robust `bvp6c`-based ODE/BVP solver with adaptive mesh and multi-stage retry.
- **Residual Control:** Implements BC, DE, and geometry-based acceptance gates.
- **Warm Starts:** Uses neighbor continuation and disk-based cached solutions for acceleration.
- **Persistence:** SHA-256–based caching (`simpleDataHash`) with catalog indexing.
- **Debug Mode:** Short-run option (`sim.debug.short`) for quick reproducibility and smoke tests.

System Architecture Overview

Core Components: - `sim_driver_quad_tree`: Main entry point for parameter exploration. - `solveAtParams`: Adaptive multi-attempt solver (residual gated, cached, warm-start capable). - `resultFileFor` + `simpleDataHash`: Unique ID system for reproducible experiments. - `writeCatalogRow`: Global metadata archive for analysis and visualization.

Directory Structure:

```
/project_root
├── results/            # cached .mat results by SHA-256 hash
├── catalog.csv         # simulation catalog index
```

```
├── logs/                # runtime logs and residual diagnostics
└── src/                 # solver, helpers, and analysis scripts
```

Dataset and Shape Representation

Shape Variables: Axisymmetric state variables including curvature, slope angle, and radius.
Parameter Space: (A, V, κ_A, H_0) — area, volume, bending rigidity, spontaneous curvature.
Representation: Each shape is stored as an ordered tuple of fields `(s, y(s))` sampled along arc length.
Labels: Discrete morphology label (oblate, prolate, stomatocyte, pear, etc.) inferred via geometric criteria.

ML Pipeline (Planned)

Objective: Train surrogate models to predict vesicle shape and energy from parameter tuples.

Pipeline Stages: 1. **Data Ingestion:** Read cached simulation data and normalize parameter scales. 2. **Feature Engineering:** Extract curvature profiles, aspect ratios, and symmetry metrics. 3. **Modeling:** Train regression (energy, pressure) and classification (shape label) models. 4. **Evaluation:** Compare surrogate predictions with high-fidelity BVP outputs. 5. **Interpretation:** Map learned latent spaces to physical bifurcations.

Planned Analyses

- **Phase Diagrams:** Visualize morphological regions in H_0, V or A, V space.
- **Energy Landscape Analysis:** Compute and compare local minima and bifurcation points.
- **Stability Metrics:** Quantify DE and BC residuals across branches.
- **Comparative Study:** Validate against theoretical limits (e.g., Helfrich energy minimizers).

Research Milestones and Task Checklist

Stage 1 – Validation & Calibration

☐ Benchmark solver against known analytical results.
☐ Quantify effect of mesh density on residual reduction.
☐ Verify reproducibility of cached vs. fresh runs.

Stage 2 – Dataset Curation

☐ Define canonical parameter grid for data generation.
☐ Generate and store first dataset snapshot (~1000 simulations).
☐ Develop utility for dataset slicing and visualization.

Stage 3 – Data Management and Provenance

☐ Replace per-node `.mat` files with unified archive (e.g., HDF5 or SQLite).
☐ Add JSON metadata snapshots for easier inspection.
☐ Implement automatic indexing in `catalog.csv`.

Stage 4 – Post-Processing and Visualization

☐ Compute derived metrics (area, curvature integrals, energies).
☐ Plot DE/BC residuals vs. parameter space.
☐ Generate phase-line visualization overlays.

Stage 5 – ML Pipeline Integration

☐ Build shape embedding model (e.g., PCA, autoencoder).
☐ Train regression model for total energy prediction.
☐ Train classification model for morphology type.
☐ Evaluate cross-validation and generalization performance.

Stage 6 – Analysis and Reporting

☐ Correlate energy minima with observed shape classes.
☐ Identify bifurcation curves via shape label transitions.
☐ Prepare figures and tables for publication.
☐ Draft methods section describing solver and numerical validation.

Optional Future Enhancements

- **cells.json Snapshot:** Store quadtree states for quick restarts and visualization.
- **Web-Based Dashboard:** Interactive shape viewer and residual heatmap browser.
- **High-Performance Computing (HPC):** Adapt for multi-node execution or port to Julia/Python (JAX backend).

Immediate Action Items

1. **Finalize simulation gate:** Verify DE and BC thresholds enforce true convergence.
2. **Run debug sweep:** Use `sim.debug.short` to validate cache logic and catalog logging.
3. **Generate summary plots:** Plot DE residual vs. BC residual colored by morphology.
4. **Document current solver:** Write `README.md` summarizing configuration and usage.

Maintained by: [Your Research Team]
Last Updated: October 2025

Document Purpose: Serves as the primary navigation and task-tracking summary for the vesicle simulation research workflow.