HybridShockWaveSim

HybridShockWaveSim implements a **hybrid computational framework** for simulating **shock waves in non-ideal gases**. It combines **extended Rankine–Hugoniot** models (using **Peng–Robinson** and **Redlich–Kwong** equations of state), **high-fidelity** finite-volume (or finite-difference) simulations with **adaptive mesh refinement (AMR)**, and an optional **machine learning** module for fast predictions in real-gas shock scenarios.

Key Features

- **Extended Rankine–Hugoniot**:

  - Incorporates **Peng–Robinson** and **Redlich–Kwong** real-gas EOS for accurate post-shock states, especially near critical points or dense-gas regimes.
  - Supports perturbation-based solutions for rankine–hugoniot jump conditions, reducing error in shock front locations and thermodynamic predictions.

- **Adaptive Finite-Volume / Finite-Difference Solver**:

  - Captures shock fronts with minimal numerical diffusion.
  - Employs **AMR** to refine cells where steep gradients occur, ensuring high resolution while limiting computational cost.

- **Machine Learning Integration (InProgress)**:

    - Trains neural networks or other ML models on solver outputs to provide **rapid** approximate predictions for real-gas shock states.
    - Helps speed up parametric sweeps or repeated simulations.

- **Case Studies**:

    - Real-gas shock tubes at various temperatures/pressures, demonstrating <15% error in shock front dynamics, entropy production, and post-shock relaxation.
    - Benchmarks showing ~30% runtime reduction over purely uniform grids.

Repository Structure

```
HybridShockWaveSim/
  ├─ solver/
  │    ├─ analytics.py      # Real-gas EOS, extended Rankine–Hugoniot
  │    ├─ numerics.py       # FVM/FDM solver, AMR routines, time-integration
  ├─ visualization/
  │    └─ plot_bokeh.py     # Bokeh-based interactive shock profile viewer
  ├─ main.py                # Coordinates entire pipeline (analytic ->
numeric -> ML -> visualize)
  ├─ requirements.txt       # NumPy, SciPy, Bokeh, etc.
  ├─ shock_experiment.csv   # Example synthetic data for validation
  └─ README.md              # This readme
```

- **`solver/analytics.py`**: Real-gas equations of state (e.g. Peng–Robinson), extended R–H formulas.
- **`solver/numerics.py`**: Core solver with AMR, controlling time steps and handling boundary conditions.
- **`solver/ml_model.py`**: Trains or loads an ML model to approximate shock states (if used).
- **`visualization/plot_bokeh.py`**: Interactive plots to explore how shock profiles change under parameter tweaks.
- **`main.py`**: Brings all modules together for a full simulation run or interactive session.

Installation

1. Clone this repository:

```
git clone https://github.com/Jayyp1234/HybridShockWaveSim.git
```

2. Install Python dependencies:

```
cd HybridShockWaveSim
pip install -r requirements.txt
```

This ensures NumPy, SciPy, Bokeh, etc. are installed.

Usage

1) Command-Line "Main" Run

```
python main.py
```

- Loads initial states from `analytics.py`,
- Runs the solver from `numerics.py` with AMR,
- Optionally trains/uses ML from `ml_model.py`,
- Exports or prints final shock profiles.

2) Interactive Visualization

```
bokeh serve --show visualization/plot_bokeh.py
```

- Opens a browser where you can **tune real-gas parameters** (e.g. acentric factor (\omega)) or AMR thresholds,
- Then re-runs or reloads data to show updated pressure/density curves.

3) Validation with Synthetic Experimental Data

- A sample file `shock_experiment.csv` is included.

- Compare your solver results or ML predictions to these “experimental” values.
- Evaluate error metrics to confirm <15% discrepancy in shock front location or density.

Contributing

- **Pull requests** or suggestions are welcome.
- For major changes, please open an **issue** first to discuss what you would like to change.

References

- **Peng–Robinson** EOS: Peng, D.-Y. and Robinson, D. B. (1976), *A new two-constant equation of state*, *I&EC Fundamentals*, 15(1), 59–64.
- **Redlich–Kwong** EOS: Redlich, O., & Kwong, J. N. S. (1949). *On the thermodynamics of solutions*.
- **AMR for Shock**: Berger, M. J., & Colella, P. (1989). *Local adaptive mesh refinement for shock hydrodynamics*.
- Additional references in the code docs or inline comments.

Enjoy your exploration of **shock waves** in **non-ideal gases** with this synergy of analytics, numerics, and machine learning. If you have questions or suggestions, please reach out or file an issue!