AE 370 – Project 2

A Numerical Method for an Initial–Boundary Value Problem (IBVP)

Overview

This repository contains the full, reproducible codebase used for **AE 370 Project 2**, which focuses on the development, verification, and convergence analysis of a numerical method for solving an **initial–boundary value problem (IBVP)**.

The goals of this repository are to: - Clearly separate **numerical method implementation** from **analysis and post-processing** - Enable **full reproducibility** of all figures, tables, and convergence results presented in the report - Provide readable documentation explaining **how the code maps to the reported results**

This repository is structured to go *beyond a homework-style submission* and meet the expectations for **significant supporting documentation**, as outlined in the project rubric.

Repository Structure

```
AE370-Project2/
│
├── src/
│   ├── ibvp_solver.py          # Core numerical method (time marching solver)
│   ├── grid.py                 # Spatial grid generation and utilities
│   ├── initial_conditions.py   # Initial condition definitions
│   ├── boundary_conditions.py  # Boundary condition implementations
│   └── exact_solution.py       # Exact / manufactured solutions (if applicable)
│
├── analysis/
│   ├── convergence_study.py    # Grid and time-step refinement studies
│   ├── error_metrics.py        # L1, L2, and Linf error calculations
│   └── stability_checks.py     # CFL / stability verification
│
├── notebooks/
│   ├── Verification_ConvergenceStudies.ipynb
│   └── AE370_Project2_FigureCode.ipynb
│
├── results/
│   ├── figures/                # All generated plots used in the report
│   └── data/                   # Saved error and convergence data
│
```

```
├── README.md                   # This file
└── acknowledgments.md          # Use of ChatGPT and other tools
```

> **Note:** File names may differ slightly depending on the final repository version, but the logical separation between *solver*, *analysis*, and *figure generation* is preserved throughout.

Numerical Method Implementation

The numerical method for solving the IBVP is implemented in the `src/` directory.
The **primary solver** is written as a reusable function that:

1. Initializes the spatial grid and time step
2. Applies initial conditions
3. Advances the solution in time using the chosen numerical scheme
4. Enforces boundary conditions at each time step
5. Returns the full time history or final solution

This design allows the solver to be: - Called by convergence studies - Reused for multiple grid resolutions - Easily modified or extended

Key Design Principle

> **The solver never generates plots or performs convergence analysis.**
> All verification and analysis are handled by separate scripts to ensure clarity and reproducibility.

Verification and Convergence Studies

Convergence and verification are handled in the `analysis/` directory and supporting Jupyter notebooks.

What is Verified

- **Order of accuracy** in space and/or time
- **Consistency** with exact or manufactured solutions
- **Grid convergence** through systematic refinement

How Convergence is Computed

1. Run the solver at multiple grid resolutions
2. Compute error norms (L1, L2, Linf)
3. Estimate observed order of accuracy using log–log slopes
4. Store results for plotting and reporting

All convergence plots included in the report are generated directly from these scripts or notebooks.

Figure Generation and Mapping to the Report

All figures used in the written report are generated using code in: - `notebooks/Verification_ConvergenceStudies.ipynb` - `notebooks/AE370_Project2_FigureCode.ipynb`

Each notebook: - Loads solver outputs or recomputes results - Produces labeled, publication-quality plots - Matches figures **one-to-one** with those appearing in the report

This ensures a clear and traceable mapping between:

> **Code → Numerical results → Report figures**

Reproducibility Instructions

To reproduce all results from scratch:

1. **Clone the repository**

```
git clone <repository-url>
cd AE370-Project2
```

2. **Install dependencies**

```
pip install numpy matplotlib scipy jupyter
```

3. **Run convergence studies**

```
python analysis/convergence_study.py
```

4. **Generate figures**

```
jupyter notebook notebooks/Verification_ConvergenceStudies.ipynb
```

All generated figures will appear in the `results/figures/` directory.

Expected Background

This repository assumes familiarity with: - Numerical methods for PDEs - Initial–boundary value problems - Consistency, stability, and convergence concepts - Python and NumPy-based scientific computing

---

Acknowledgments

This project made use of **ChatGPT** as a support tool for: - Code refactoring and readability improvements - Documentation drafting (including this README) - Debugging assistance and conceptual checks

All numerical methods, modeling decisions, and verification strategies were **driven and validated by the project group**, with ChatGPT serving as an assistive tool rather than an authoritative source.

Summary

This repository is intentionally designed to: - Meet **full reproducibility** expectations - Provide **clear mapping between code and results** - Exceed the baseline requirements for project documentation

It is suitable for independent verification, grading review, and future extension of the numerical method.