# Introduction

## Overview

This is an example project that demonstrates the generic repository structure for tested code, manuscript editing, and PDF rendering. The work presents a novel optimization framework with comprehensive theoretical analysis and experimental validation, building upon foundational optimization theory [**?**, **?**] and recent advances in adaptive methods [**?**, **?**].

## Project Structure

The project follows a standardized structure:

- `src/` - Source code with comprehensive test coverage
- `tests/` - Test files ensuring 100% coverage
- `scripts/` - Project-specific scripts for generating figures and data
- `manuscript/` - Markdown source files for the manuscript
- `output/` - Generated outputs (PDFs, figures, data)
- `repo_utilities/` - Generic utility scripts for any project

## Key Features

### Test-Driven Development

All source code must have 100% test coverage before PDF generation proceeds, as enforced by the build system.

### Automated Script Execution

Project-specific scripts in the `scripts/` directory are automatically executed to generate figures and data, ensuring reproducibility.

### Markdown to PDF Pipeline

Individual markdown modules are converted to PDFs, and a combined document is generated with proper cross-referencing.

### Generic and Reusable

The utility scripts can be used with any project that follows this structure, making it easy to adopt for new research projects.

## Manuscript Organization

The manuscript is organized into several key sections:

1. **Abstract** (Section **??**): Research overview and key contributions
2. **Introduction** (Section ): Overview and project structure

3. **Methodology** (Section **??**): Mathematical framework and algorithms
4. **Experimental Results** (Section **??**): Performance evaluation and validation
5. **Discussion** (Section **??**): Theoretical implications and comparisons
6. **Conclusion** (Section **??**): Summary and future directions
7. **References** (Section **??**): Bibliography and cited works

## Example Figure

The following figure was generated by the example script:

[width=0.8]../output/figures/example$_{f}igure.png$

Figure 1: Example project figure showing a mathematical function

This demonstrates how figures are automatically integrated into the manuscript with proper cross-referencing capabilities. The figure shows a mathematical function that demonstrates the project's capabilities. As shown in Figure , the system generates high-quality visualizations that are automatically integrated into the manuscript.

## Data Availability

All generated data is saved alongside figures for reproducibility:

- **Figures**: PNG format in `output/figures/`
- **Data**: NPZ and CSV formats in `output/data/`
- **PDFs**: Individual and combined documents in `output/pdf/`
- **LaTeX**: Source files in `output/tex/`

## Usage

To generate the complete manuscript:

```
# Clean previous outputs
./repo_utilities/clean_output.sh

# Generate everything (tests + scripts + PDFs)
./repo_utilities/render_pdf.sh
```

The system will automatically: 1. Run all tests with 100% coverage requirement 2. Execute project-specific scripts to generate figures and data 3. Validate markdown references and images 4. Generate individual and combined PDFs 5. Export LaTeX source files

## Customization

This template can be customized for any project by:

1. Adding project-specific scripts to `scripts/`
2. Modifying markdown files in `markdown/`
3. Setting environment variables for author information
4. Adjusting LaTeX preamble in `preamble.md`
5. Adding new sections with proper cross-references

## Cross-Referencing System

The manuscript demonstrates comprehensive cross-referencing:

- **Section References**: Use `\ref{sec:section_name}` to reference sections
- **Equation References**: Use `\eqref{eq:objective}` to reference equations (see Section **??**)
- **Figure References**: Use `\ref{fig:figure_name}` to reference figures
- **Table References**: Use `\ref{tab:table_name}` to reference tables

All references are automatically numbered and updated when the document is regenerated. For example, the main objective function (**??**) is defined in the methodology section.