# Plato

*Release 0.0*

Plato Team

Jul 22, 2021

# CONTENTS

plato

OPTIMIZATION-BASED DESIGN

**Note:** These pages are a pre-release version. They are under active development.

These Github pages are meant to serve as the documentation for **Plato Engine** and its associated repositories. These repositories can be found at the Plato Engine GitHub account.

These pages are built using the Plato Docs GitHub repository.

CHAPTER

# ONE

# CONTENTS

## 1.1 Description

The **PLATO Engine** computer program serves as a collaborative testbed rich in light-weight synthesis tools for optimization-based design. PLATO Engine is a research code designed to facilitate collaboration with academia, labs and industries by providing interfaces for plug-n-play insertion of synthesis technologies in the areas of modeling, analysis and optimization. Currently, PLATO Engine offers a set of light-weight tools for finite element analysis, linear- and nonlinear-programming, and non-gradient based optimization. The PLATO Engine program is designed to run on high-performance computers.

### 1.1.1 Application

The PLATO Engine testbed is designed to support research in the area of optimization-based design on high-performance computing systems. The PLATO Engine testbed is used to explore interoperability with several analysis, modeling and optimization tools. The testbed is also used to test the viability of these analysis, modeling and optimization tools for the solution of optimization-based design problems.

### 1.1.2 Approach

PLATO Engine is intended to serve as a collaborative testbed. It is designed to enable intercommunication of modeling, analysis and optimization data using a Multiple Program, Multiple Data (MPMD) parallel programming model. The MPMD model allows multiple, independent programs/executables to share data in-memory. The optimization algorithms orchestrate the execution and communication between multiple analysis codes and aggregates their contributions to create designs that meet multiple performance criteria.

### 1.1.3 High Performance Computing

PLATO Engine has been designed for MPMD parallel executions. It also targets Single Program, Multiple Data (SPMD) parallel programming model; however, the SPMD model is not heavily used by the targeted PLATO applications. PLATO Engine is also performance-protable, which allow it to optimally perform in current and next-generation computing architectures.

### 1.1.4 Required Libraries

Trilinos library (provides Epetra, Seacas and STK): https://github.com/trilinos/trilinos

Omega_h library (provides mesh metadata): https://github.com/SNLComputation/omega_h

Netcdf library (provides I/O libraries): https://www.unidata.ucar.edu/software/netcdf/

AMGX library (provides GPU linear solver): https://github.com/NVIDIA/AMGX

Lapack library (provides linear algebra libraries): http://www.netlib.org/lapack/

Boost library (provides C++ source libraries): https://www.boost.org/

### 1.1.5 Hardware Requirements

Tested compilers are `g++ 4.7.2`, `g++ 5.4.0`, and `g++ 7.2.0` compilers. Tested OS include Linux and Mac. RAM requirements are problem size dependent.

**TODO:** is the following still true?

**Note: Currently, PLATO Analyze only runs on Graphics Processing Units (GPUs).**

### 1.1.6 Contributing

Please open a GitHub issue to ask a question, report a bug, request features, etc. If you'd like to contribute, please fork the repository and use a feature branch. Make sure to follow the team's coding style policies. Pull requests are welcomed.

**TODO:** create coding style policies page

### 1.1.7 User Support

Users are welcomed to submit questions via email to plato3D-help@sandia.gov.

## 1.2 Plato Repositories

These repositories can be found at the Plato Engine GitHub account.

| GitHub Repository | Description |
|---|---|
| AMGX | Distributed multigrid linear solver library on GPU |
| exo2obj | Convert from an exodus volume mesh to an obj surface mesh |
| nvcc_wrapper | Wrapper shell script for NVIDIA nvcc to better conform to host compiler command line arguments |
| Plato3D | |
| Plato3D_backend | |
| platoanalyze | Fast physics and gradients for multidisciplinary analysis and optimization |
| platodocs | Documentation for Plato |
| platoengine | Plato Engine - Optimization-Based Design Ecosystem |
| spack | The spack configuration repository for building Plato. |
| tests | |
| use_cases | Collection of Plato Engine use cases |
| verification | verification and validation problems |

## 1.3 Documentation Process

This section describes the process for cloning, editing, and commiting to these GitHub pages, i.e., these web pages.

### 1.3.1 Cloning Plato Docs

Firstly, navigate to a directory where the user can clone Plato Docs.

```
git clone https://github.com/platoengine/platodocs.git

mkdir platodocs-build

cd platodocs-build

git clone https://github.com/platoengine/platodocs.git html

cd html

git checkout -b gh-pages remotes/origin/gh-pages
```

The directory structure should look like this:

```
topfolder\
|-- platodocs\
|   |-- docs\
|   |   |-- source\
|   |   |-- make.bat
|   |   |-- Makefile
|   |-- platodocs.pdf
|   `-- README.md
|
`-- platodocs-build\
    |-- doctrees\
```

(continues on next page)

(continued from previous page)

```
    `-- html\
        |-- _images\
        |-- _source\
        |-- _images\
        |-- index.html
        |-- .nojekyll
        `-- other files
```

# CHAPTER TWO

## INDICES AND TABLES

- genindex
- modindex
- search