

---

# TAO for Python

*Release 2.1*

**Lisandro Dalcin**

September 12, 2012

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	Features . . . . .	2
1.2	Components . . . . .	2
<b>2</b>	<b>Tutorial</b>	<b>2</b>
<b>3</b>	<b>Installation</b>	<b>2</b>
3.1	Requirements . . . . .	2
3.2	Using <b>pip</b> or <b>easy_install</b> . . . . .	3
3.3	Using <b>distutils</b> . . . . .	3

## Abstract

This document describes `tao4py`, a `Python` port to the `TAO` libraries.

`TAO` is a software package for the parallel solution of large-scale optimization problems.

`TAO` relies on `PETSc` for basic functionality such as the representation of matrices and vectors, and the solution of linear systems of equations. Thus, `tao4py` must be used together with its companion `petsc4py`.

# 1 Overview

*TAO for Python* (`tao4py`) is a `Python` package that provides convenient access to the functionality of `TAO`.

`TAO`<sup>1</sup> implements algorithms and tools for the numerical solution of large, sparse ...

## 1.1 Features

`TAO` is intended for use in large-scale application projects ...

## 1.2 Components

`TAO` components provide the functionality required for the parallel solutions of large-scale optimization problems.

**TaoSolver** Provides ...

**TaoApp** Provides ...

# 2 Tutorial

XXX To be written ... Any contribution welcome!

# 3 Installation

## 3.1 Requirements

You need to have the following software properly installed in order to build *TAO for Python*:

- Any `MPI` implementation<sup>2</sup> (e.g., `MPICH` or `Open MPI`), built with shared libraries.
- `PETSc` 3.3 or 3.2 release, built with shared libraries.
- `TAO` 2.1 or 2.0 release, built with shared libraries.
- `Python` 2.4 to 2.7 or 3.1 to 3.2<sup>3</sup>.
- `NumPy` package.
- `petsc4py` package.

---

<sup>1</sup> Lois Curfman McInnes, Jorge Moré, Todd Munson and Jason Sarich. *TAO User Manual*. ANL/MCS-TM-242 - Revision 1.10. Mathematics and Computer Science Division, Argonne National Laboratory. 2007

<sup>2</sup> Unless you have appropriately configured and built `TAO` and `PETSc` without `MPI` (configure option `--with-mpi=0`).

<sup>3</sup> You may need to use a parallelized version of the `Python` interpreter with some `MPI-1` implementations (e.g. `MPICH1`).

## 3.2 Using pip or easy\_install

You can use **pip** to install `tao4py` and its dependencies (`mpi4py` is optional but highly recommended):

```
$ pip install [--user] numpy mpi4py
$ pip install [--user] petsc petsc4py
$ pip install [--user] tao tao4py
```

Alternatively, you can use **easy\_install** (deprecated):

```
$ easy_install [--user] tao4py
```

If you already have working PETSc and TAO builds, set environment variables `TAO_DIR` and `PETSC_DIR` (and perhaps `PETSC_ARCH` for prefix installs) to appropriate values and next use **pip**:

```
$ export TAO_DIR=/path/to/tao
$ export PETSC_DIR=/path/to/petsc
$ export PETSC_ARCH=arch-linux2-c-opt
$ pip install [--user] petsc4py tao4py
```

## 3.3 Using distutils

### Downloading

The *TAO for Python* package is available for download at the project website generously hosted by Google Code. You can get a release tarball **curl** or **wget**.

- Using **curl**:

```
$ curl -O http://tao4py.googlecode.com/files/tao4py-X.Y.tar.gz
```

- Using **wget**:

```
$ wget http://tao4py.googlecode.com/files/tao4py-X.Y.tar.gz
```

### Building

After unpacking the release tarball:

```
$ tar -zxf tao4py-X.X.X.tar.gz
$ cd tao4py-X.X.X
```

the distribution is ready for building.

Some environmental configuration is needed to inform the location of PETSc and TAO. You can set (using **setenv**, **export** or what applies to you shell or system) the environmental variables `TAO_DIR`, `PETSC_DIR`, and `PETSC_ARCH` indicating where you have built/installed TAO and PETSc:

```
$ export TAO_DIR=/usr/local/tao
$ export PETSC_DIR=/usr/local/petsc
$ export PETSC_ARCH=arch-linux2-c-opt
```

Alternatively, you can edit the file `setup.cfg` and provide the required information below the `[config]` section:

```
[config]
tao_dir    = /usr/local/tao
petsc_dir  = /usr/local/petsc
```

```
petsc_arch = arch-linux2-c-opt
...
```

Finally, you can build the distribution by typing:

```
$ python setup.py build
```

## Installing

After building, the distribution is ready for installation.

You can do a site-install type:

```
$ python setup.py install
```

or, in case you need root privileges:

```
$ su -c 'python setup.py install'
```

This will install the `tao4py` package in the standard location `prefix/lib64/pythonX.Y/site-packages`.

You can also do a user-install type. There are two options depending on the target Python version.

- For Python 2.6 and up:

```
$ python setup.py install --user
```

- For Python 2.5 and below (assuming your home directory is available through the `HOME` environment variable):

```
$ python setup.py install --home=$HOME
```

and then add `$HOME/lib/python` or `$HOME/lib64/python` to your `PYTHONPATH` environment variable.