ASAP™ Documentation | Release 0.0.1

March 30th, 2017

LICENSE

CONTENTS

- Prerequisites
- Installing Anaconda Python
- Installing Dependencies
- Installing ASAP
- Examples

PREREQUISITES

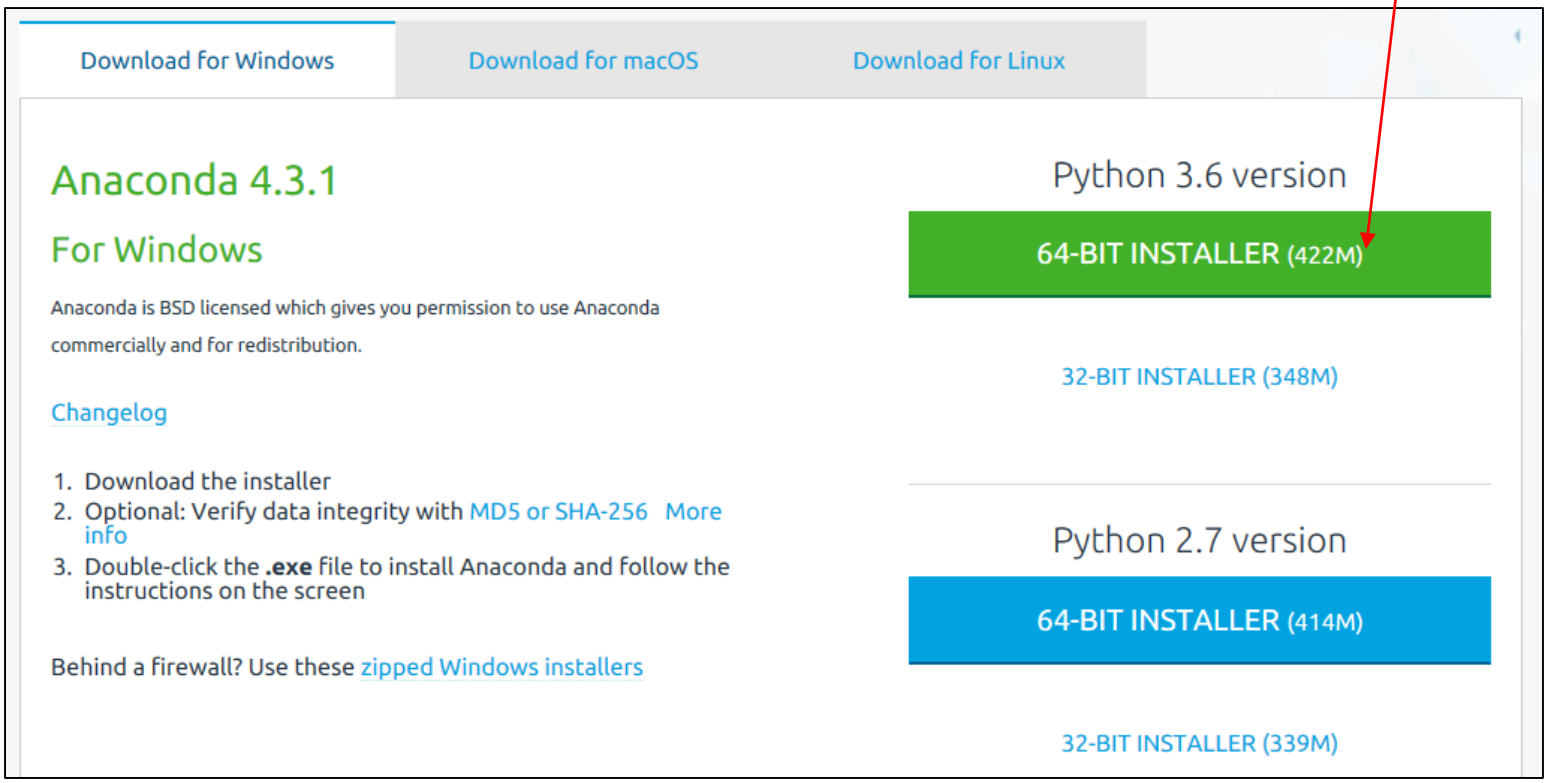
- ASAP is currently developed for Windows 64-bit Python 3.5
  - Pre-built binaries for geometry kernel and meshing tools are available
  - Other platforms and Python versions will be available in the future

- Anaconda Python is **strongly** recommended
  - Simple and powerful package management system
  - Pre-built binaries will self-install from Anaconda cloud

INSTALLING ANACONDA PYTHON

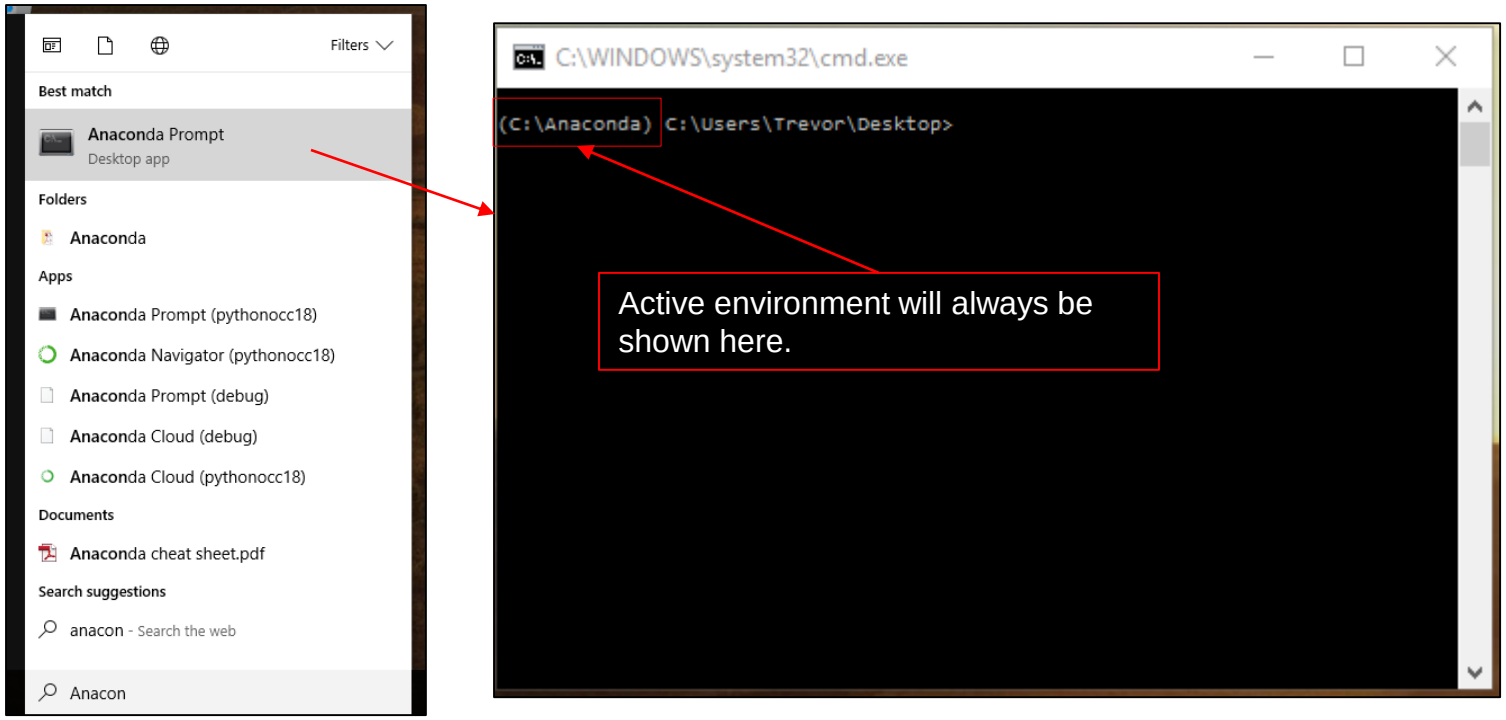
- Anaconda Python installers can be found here:
  - https://www.continuum.io/downloads
  - Recommended install directory is something like "C:\Anaconda" (avoid spaces)

Install Python 3.6 64-bit as root environment. Python 3.5 environment specifically for ASAP will be created afterwards.



INSTALLING ANACONDA PYTHON

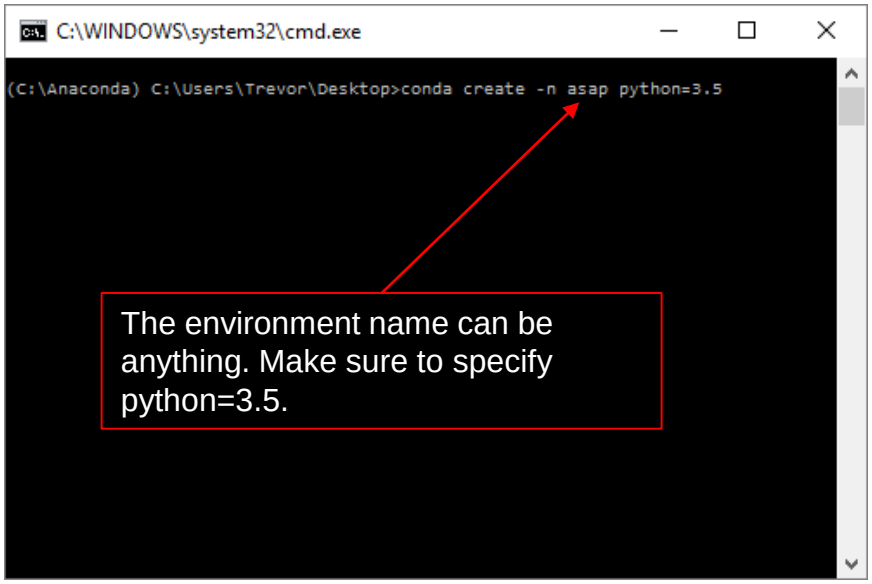
- The “Anaconda Prompt” should now be available
  - Use “conda” command to create environments, manage packages, etc.

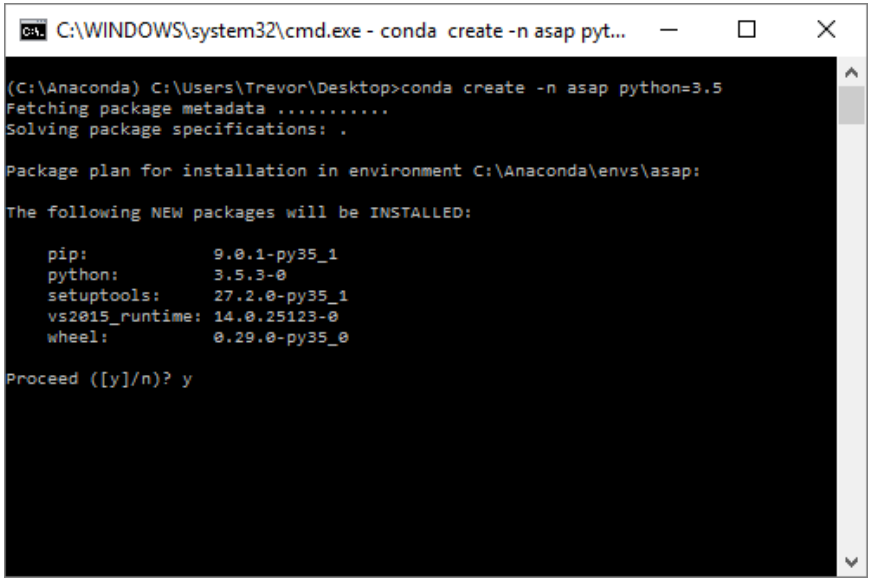


INSTALLING DEPENDENCIES

- Create a designated conda environment for ASAP
  - This will create a clean separation between ASAP required packages and other Python installations and packages

conda create –n *name* python=3.5



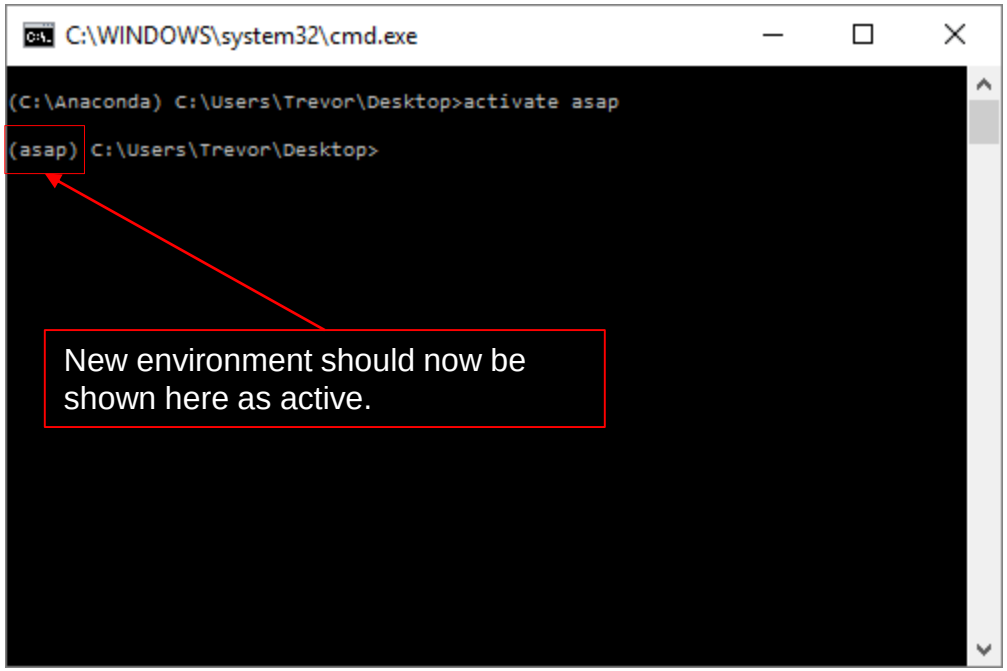


Note: A lot of junk may show up after installation in the command prompt. Ignore this. It appears to be a bug in conda.

INSTALLING DEPENDENCIES

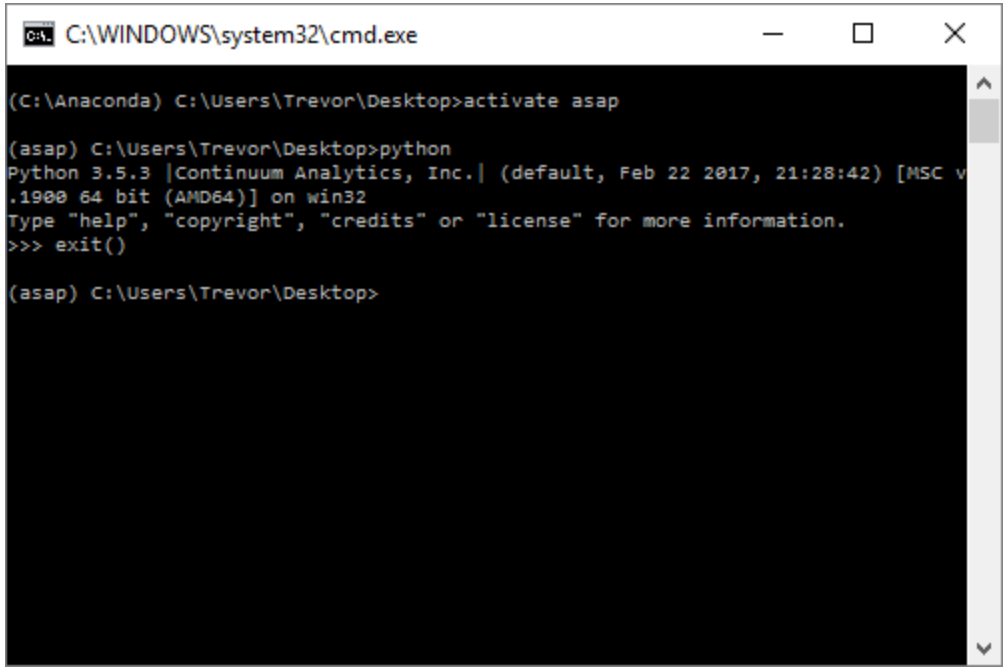
- Activate the new environment

activate *name*



Installing Dependencies

- Activating an environment sets the systems paths such that they point to the environment Python interpreter and packages
  - Launching “python” from the activated environment should show Python 3.5
  - By default packages will be installed to active environment unless a name is given



INSTALLING DEPENDENCIES

- First install OpenCascade Community Edition, PythonOCC, SMESH, and Netgen from the Anaconda cloud using an Anaconda Prompt with the previously created environment activated
- All of this will be done with one command:

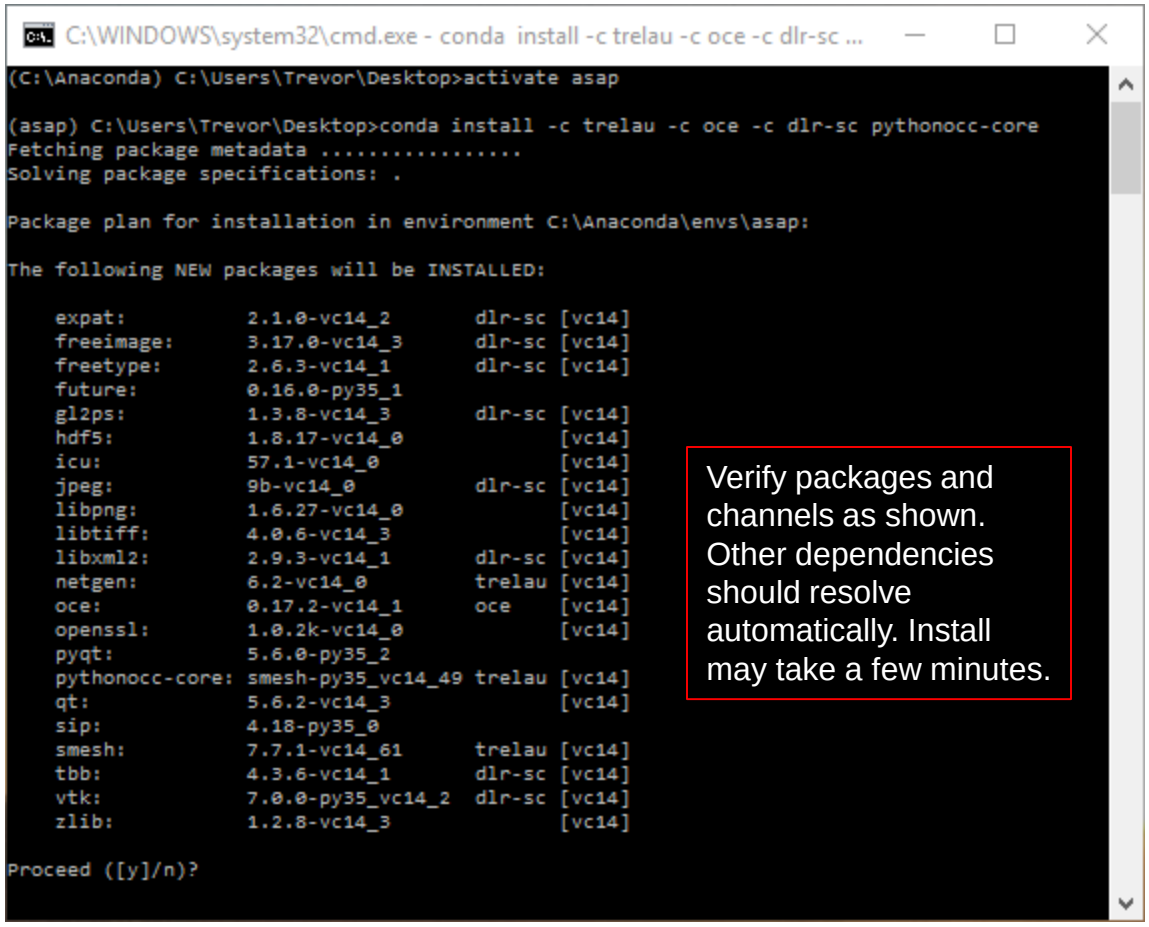
conda install –c trelau –c oce –c dlr-sc pythonocc-core

install command will install the packages in the active environment

-c command adds channels to search for dependencies on the Anaconda cloud

pythonocc-core is the name of package we are searching for and installing

INSTALLING DEPENDENCIES



Verify packages and channels as shown. Other dependencies should resolve automatically. Install may take a few minutes.

INSTALLING DEPENDENCIES

- The conda directory can be cleaned using the command “conda clean –a”

INSTALLING DEPENDENCIES

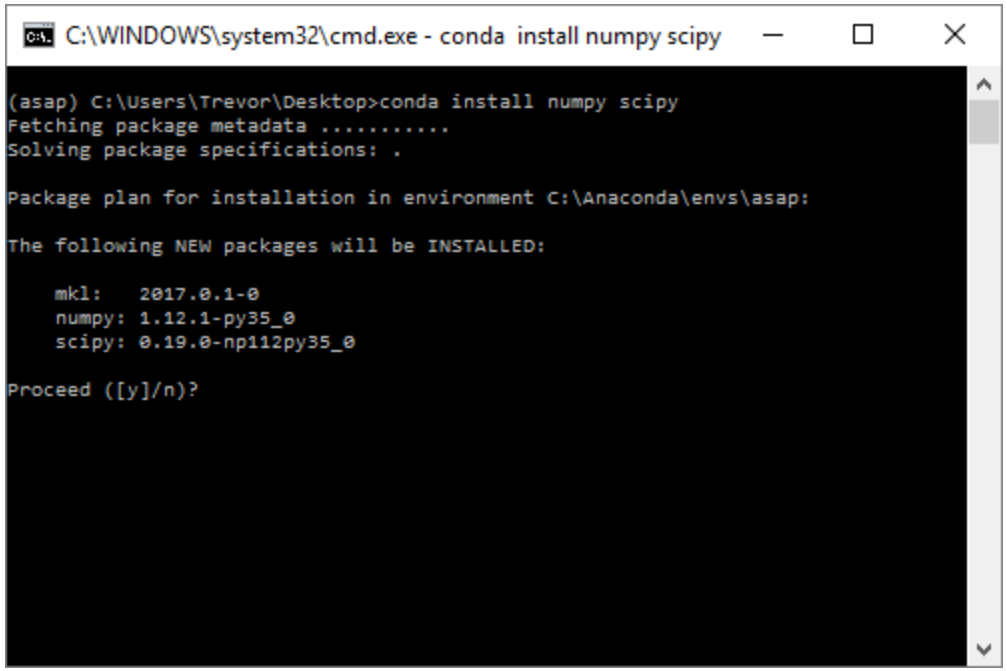
- ASAP is currently optimized to work with a modified version of OpenVSP and can be downloaded here:

OpenVSP-3.5.0-metadata-py35-win64

- Changes include exporting metadata to the STEP file and additional reference surfaces for wing components

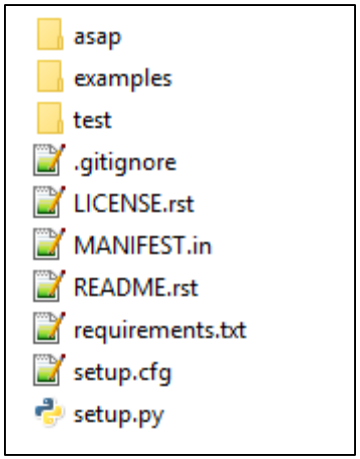
INSTALLING DEPENDENCIES

- Install numpy and scipy
  - "conda install numpy scipy"



INSTALLING ASAP

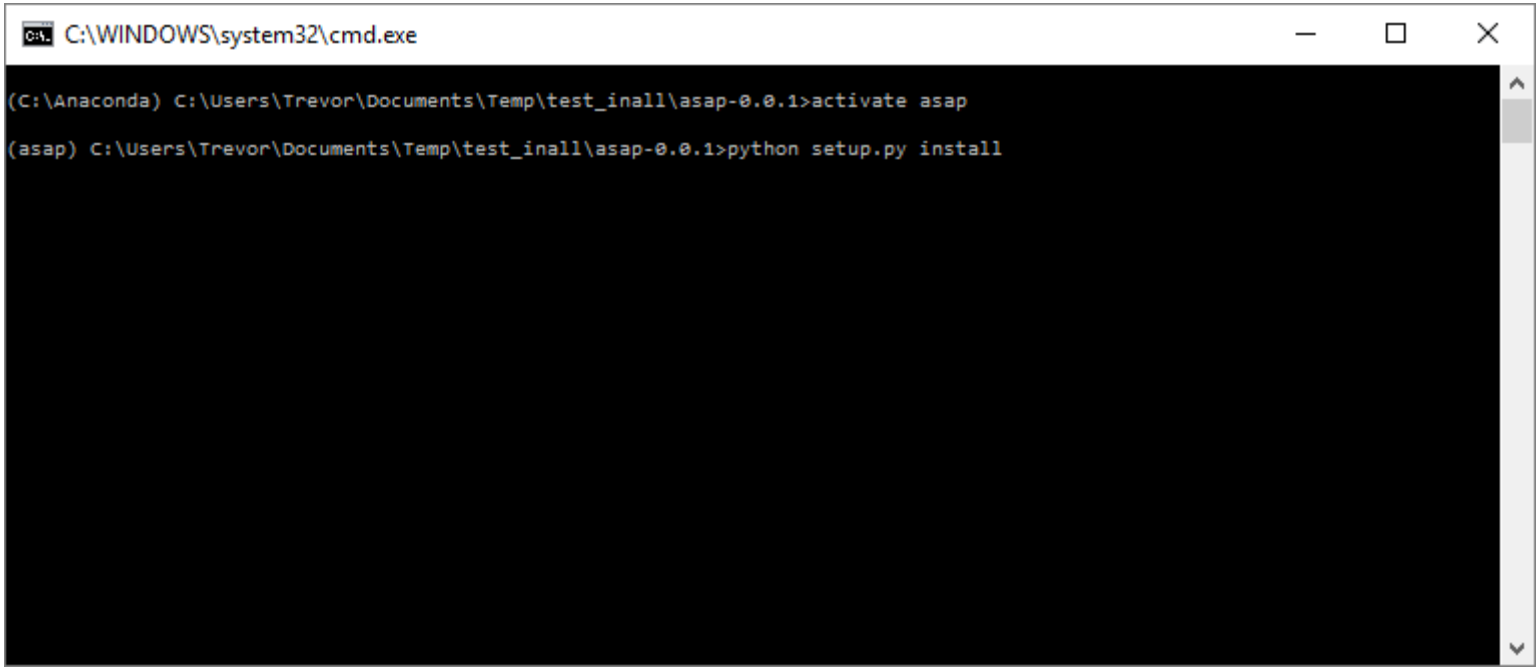
- ASAP is currently distributed as a “source distribution”
- Place “ASAP-0.0.1” in desired directory
- Contents should look like:



INSTALLING ASAP

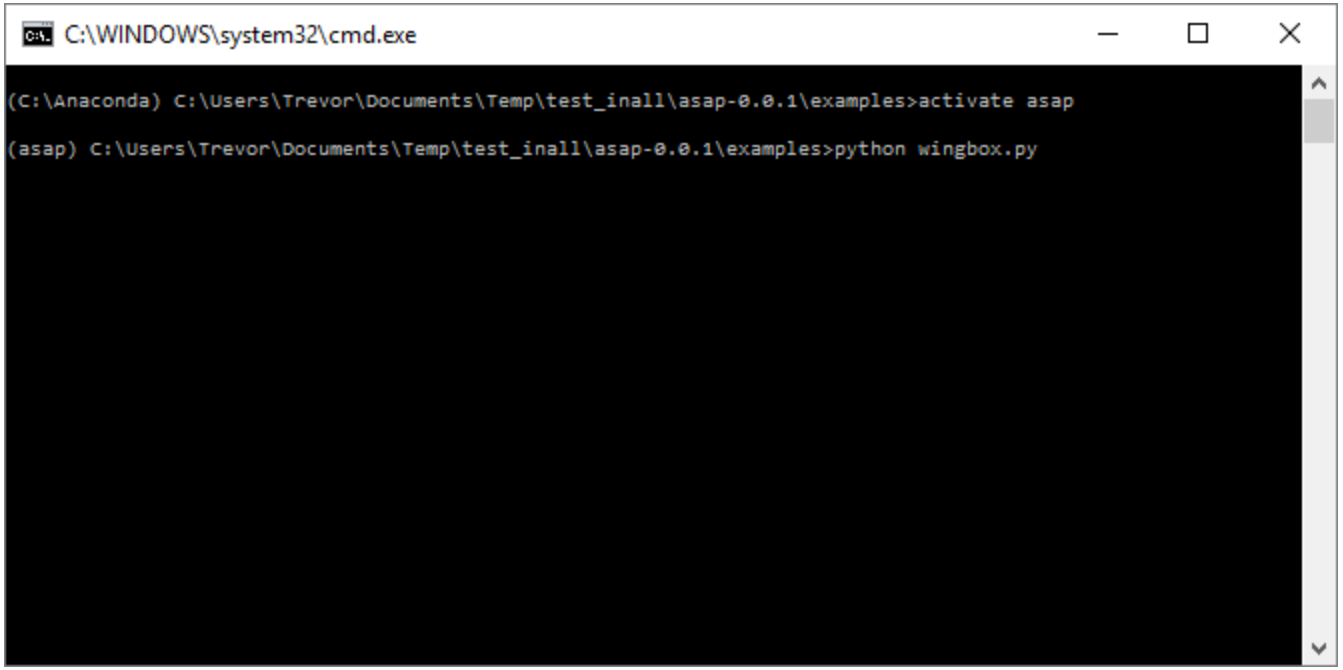
- Launch an Anaconda Prompt from the ASAP-0.0.1 folder, activate the previously created environment, and install using the command

python setup.py install

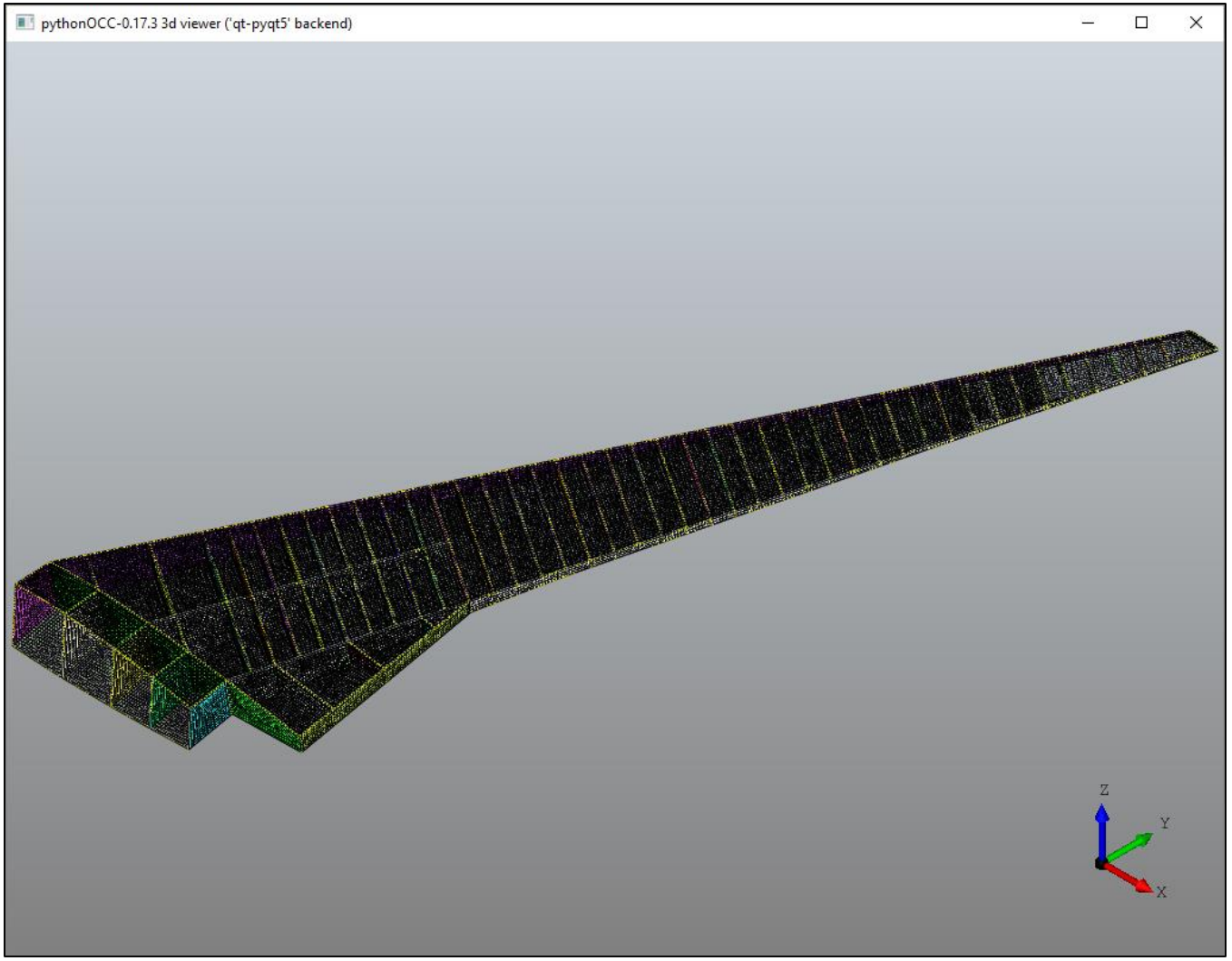


EXAMPLES

- Navigate to the “examples” folder and run the wingbox.py script from an Anaconda Prompt with the appropriate environment activated

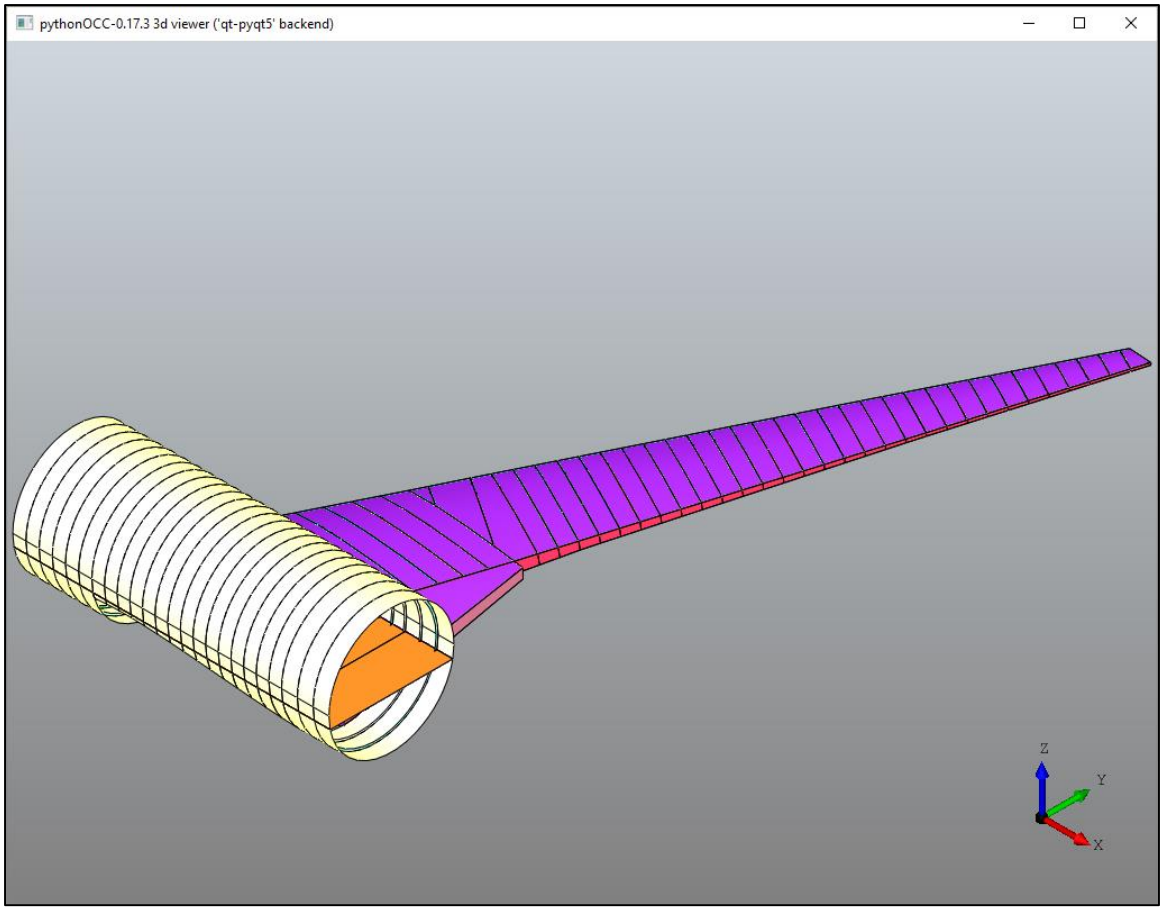


EXAMPLES



EXAMPLES

python wing-body.py



laughlinresearch