Shock & Detonation Toolbox Installation Instructions

Joel Lawson and Joseph Shepherd*
Graduate Aerospace Laboratories
California Institute of Technology
Pasadena, CA 91011 USA

February 16, 2020

These instructions are for installing the 2018 version of the Shock and Detonation Toolbox. The toolbox documentation and download files are available on the SDToolbox website. In order to use these scripts, the reader must install the Cantera software and Python or MATLAB. The documentation and software for Cantera is open source and available at www.cantera.org.

1 MATLAB

1.1 Prerequisites

- MATLAB R2015 or later
- Cantera 2.3 or 2.4

1.2 Toolbox Installation

Download the zip file from the SDToolbox website and unpack this into a temporary directory. Copy the entire SDToolbox/MATLAB/SDToolbox directory to a desired location. This could be MATLAB's Toolbox directory (e.g. C:/Program Files/MATLAB/toolbox/ on Windows) but need not be. Open MATLAB and use "Set Path" to add the SDToolbox directory and subdirectories to the MATLAB path. All the SDToolbox functions should now be available from the command line in MATLAB. You may need to execute the rehash and rehash toolbox commands to get the new functions to be loaded correctly.

1.3 Demo Scripts & Custom Thermodynamic Data Files

The MATLAB versions of the demo scripts are all found in SDToolbox/MATLAB/Demo. They can be left here when the toolbox is moved to its desired location, or can be separately moved to another location. Several of the scripts create output files, so be sure to run them in a location where you have write permissions.

The toolbox also comes with a large number of cti reaction mechanism, transport and thermodynamic data files in Cantera cti format, located in SDToolbox/cti. Several of these files are needed for the mechanisms in the demo scripts. In order to run the demo scripts, these custom cti files need to be in Cantera's search path. The descriptions of these data files and references to the sources can be found on the SDToolbox cti webpage. Information about thermodynamic data sources, formats and some utility programs are on the thermo webpage.

There are several options here, depending on user preference. The files can be put in the same directory as Cantera's default data files (e.g. C:/Program Files/Cantera/data/ on Windows), but one may prefer to keep the new files separate. Cantera also checks the current working directory, so they can also be placed in the same directory as the demo scripts themselves.

Alternatively, the CANTERA_DATA environmental variable can be set to the path where the files are located. In Windows, this is done via the Environmental Variables dialog via the Control Panel. In Unix-like systems, there are multiple sets of environmental variables (e.g. if using Bash for an interactive shell, environmental variables are defined in .bashrc)—so in this case, you would need to determine which shell MATLAB is being run from, and define CANTERA_DATA in the corresponding configuration file.

Finally, if you only want to temporarily add these files to Cantera's search path, you can use the Cantera function adddir() during a MATLAB session.

*joseph.e.shepherd@caltech.edu

1.4 Toolbox manifest

- SDToolbox
  - SDTconfig.m
  - CV
    - CV.txt
    - cvsolve.m
    - cvsys.m
  - PostShock
    - CJspeed.m
    - CJ_calc.m
    - FHFP.m
    - hug_eq.m
    - hug_fr.m
    - PostShock.txt
    - PostShock_eq.m
    - PostShock_fr.m
    - shk_calc.m
    - shk_eq_calc.m
  - Reflections
    - FHFP_reflected_fr.m
    - PostReflectedShock_eq.m
    - PostReflectedShock_fr.m
    - reflected_eq.m
    - reflected_fr.m
    - Reflections.txt
  - Stagnation
    - Stagnation.txt
    - stgsolve.m
    - stgsys.m
  - Thermo
    - eq_state.m
    - gruneisen_eq.m
    - gruneisen_fr.m
    - soundspeed_eq.m
    - soundspeed_fr.m
    - state.m
    - Thermo.txt
  - Utilities
    - CJspeed_plot.m
    - cv_plot.m
    - Utilities.txt
    - znd_fileout.m
    - znd_plot.m
  - ZND
    - ZND.txt
    - zndsolve.m
    - zndsys.m

2 Python

2.1 Prerequisites

- Python 3
- Cantera 2.3 or 2.4
- The most recent versions of the following Python packages:
  - NumPy
  - SciPy
  - matplotlib
- In addition to the above, the following packages are used by some demo scripts (only cycler is not part of the Python Standard Library, being a package from the wider matplotlib project):
  - datetime
  - sys
  - pickle
  - cycler

2.2 Toolbox Installation

Copy the entire SDToolbox/Python3/sdtoolbox directory to a location on your Python's path. Typically, this would be the site-packages directory (or dist-packages under Debian Linux). For example, under a default Anaconda installation on Windows, this might be found at C:/Users/<user>/Anaconda3/Lib/site-packages, under Debian's native Python 3 environment, at /usr/local/lib/python3.x/dist-packages.

For Windows with a plain Python installation and
PYTHONPATH = c:/users/<user>/AppData/Local/Programs/Python/Python35/,
the toolbox should be located in PYTHONPATH/Lib/site-packages. After installation, go up one directory to PYTHONPATH/Lib/site-packages and create a text file named sdtoolbox.pth containing the single line sdtoolbox; then go up one more directory to PYTHONPATH/Lib and run the python script site.py.

Once the package is placed in the proper location, it should be available via import sdtoolbox in an interactive Python session.

2.3 Demo Scripts & Custom Thermodynamic Data Files

The Python versions of the demo scripts are found in the directory SDToolbox/Python3/demo. These can be left in place, or moved to another location as desired. Again, some demos require write permissions, which may not be available without sudo in the default Python package directories.

Once the toolbox itself is functional as described above, you should be able to run demo scripts from the terminal (e.g. python demo_CJstate.py) or in an IDE like Spyder or the idle application on Windows. Remember to call python3 rather than python if your system has both Python 2 and 3 installed.

As with MATLAB, the demo scripts require some of the custom cti files found in the downloaded archive, and these also need to be placed in a location visible to Cantera. Analogously to MATLAB, if you do not wish to mix these files in with the default Cantera data files, you can either set the CANTERA_DATA environmental variable to permanently add a new path, or use the cantera.add_directory() function to temporarily add a path during an interactive Python session.

2.4 Toolbox manifest

```
└──sdtoolbox
       config.py
       cv.py
       postshock.py
       reflections.py
       stagnation.py
       thermo.py
```

utilities.py
znd.py
___init___.py

3 Demo Manifest

The manifest is given for the MATLAB directory, the Python directory file naming convention is identical except for the file name extensions and some functions that are included within the demo scripts rather than as separate files.

Demo
- adiasys.m
- demo_CJ.m
- demo_CJstate.m
- demo_CJstate_isentrope.m
- demo_CJ_and_shock_state.m
- demo_cvCJ.m
- demo_cvshk.m
- demo_cv_comp.m
- demo_detonation_pu.m
- demo_equil.m
- demo_EquivalenceRatioSeries.m
- demo_ExplosionSeries.m
- demo_exp_state.m
- demo_g.m
- demo_GasPropAll.m
- demo_oblique.m
- demo_overdriven.m
- demo_OverdriveSeries.m
- demo_PrandtlMeyer.m
- demo_PrandtlMeyerDetn.m
- demo_PrandtlMeyerLayer.m
- demo_PrandtlMeyer_CJ.m
- demo_precompression_detonation.m
- demo_PressureSeries.m
- demo_PSeq.m
- demo_PSfr.m
- demo_quasi1d_eq.m
- demo_reflected_eq.m
- demo_reflected_fr.m
- demo_RH.m
- demo_RH_air.m
- demo_RH_air_eq.m
- demo_RH_air_isentropes.m
- demo_RH_CJ_isentropes.m
- demo_rocket_impulse.m
- demo_RZshk.m
- demo_ShockTube.m
- demo_shock_adiabat.m
- demo_shock_point.m
- demo_shock_state_isentrope.m
- demo_STGshk.m
- demo_STG_RZ.m
- demo_TP.m
- demo_TransientCompression.m
- demo_vN_state.m
- demo_ZNDCJ.m
- demo_ZNDshk.m

```
    demo_ZND_CJ_cell.m
    README.txt
    tpsys.m
```

4 CTI Manifest

```
cti
    airNASA9ions.cti
    airNASA9noions.cti
    aramco2.cti
    Blanquart2018.cti
    Burke2012.cti
    ck2cti.txt
    Davis2005.cti
    ffcm1.cti
    gri30_highT.cti
    h2br2.cti
    hexaneFull.cti
    hexanePartial.cti
    hexaneReduced.cti
    Hong2011.cti
    JetSurf2.cti
    Keromnes2013.cti
    Li2015.cti
    Mevel2015.cti
    Mevel2017.cti
    Mevel2018.cti
    OH-IUPAC-NASA9.cti
    OH-Ruscic-Burcat.cti
    PG14.cti
    README.txt
    sandiego20161214.cti
    sandiego20161214_H2only.cti
    tree

├──2-Butenal
    CH3CHCHCHO.cti
    CH3CHCHCHO_NASA7.dat
    CH3CHCHCHO_NASA9.dat

├──Blanquart
    │   BadSpecies-orig.txt
    │   chem.inp
    │   therm.dat
    │   tran.dat
    │
    └──chem
            C10H20.dat
            C12H23.dat
            C12H24.dat
            C12H24O.dat
            C12H24O3.dat
            C12H24OOH.dat
            C12H25.dat
            C12H25O.dat
            C12H25O2.dat
            C6H11.dat
```

```
        C6H12.dat
        C6H12O.dat
        C6H12OOH.dat
        C6H13O.dat
        C6H13O2.dat
        C6H13O4.dat
        C8H15.dat
        C8H16O3.dat
        C8H16OOH.dat
        C8H17O2.dat
        C8H17O4.dat
        C9H17.dat
        C9H18.dat
        N-C10H21.dat
        N-C12H26.dat
        N-C6H14.dat
        N-C8H17.dat
        N-C9H19.dat
        refit_log.txt

├──hexaneFull
    chem.inp
    therm.dat
    tran.dat

├──hydroxyl
    OH(A)-partition.csv
    OH(A)-partition.cti
    OH(X)-partition.csv
    OH(X)-partition.cti
    OH-cantera.csv
    OH-comparison.pdf
    OH-partition.csv
    OH-partition.cti
    plotter.m

├──JetSurf
    therm.dat

├──methylidyne
    CH(A)-cantera.csv
    CH(A)-partition.csv
    CH(A)-partition.cti
    CH(X)-cantera.csv
    CH(X)-partition.csv
    CH(X)-partition.cti
    CH-cantera.csv
    CH-comparison.pdf
    CH-partition.csv
    CH-partition.cti
    plotter.m

├──Mevel2015
    chem.inp
    OH-refit.cti
    therm.dat
    tran.dat
```

- Mevel2017
  - chem.cti
  - chem.inp
  - chem7.cti
  - chem9.cti
  - therm7.dat
  - therm9.dat
  - tran.dat
- Mevel2018
  - chem.inp
  - therm.dat
  - tran.dat
- NASA7
  - GOOS_BURCAT_RUSCIC_THERM.DAT
  - nasa7.dat
  - nasa7mod.dat
- NASA9
  - nasa9.dat
  - therm-ions.dat
  - therm-noions.dat
  - therm.dat
  - tran.dat
- nitric_oxide
  - NO(A)-partition.csv
  - NO(A)-partition.cti
  - NO(X)-partition.csv
  - NO(X)-partition.cti
  - NO-cantera.csv
  - NO-partition.csv
  - NO-partition.cti
  - plotter.m
- SanDiego
  - therm.dat
- utilities
  - CH_rotvib.m
  - diatomic.cti
  - NO_rotvib.m
  - OH_rotvib.m
  - partition_rotvib.m
  - poly_cp.m
  - thermo_check.py
  - thermo_fit.m
  - thermo_refit.m
  - thermo_replace.m
  - twobutenal.m

5 Testing

5.1 MATLAB

Load and run the script demo_CJState in MATLAB. The output in the Command Window should be:

demo_CJstate
CJ computation for Mevel2017.cti with composition H2:2 O2:1 N2:3.76
Initial state P1 = 100000 Pa & T1 = 295 K
CJ speed 1968.4786 (m/s)
CJ State
  Pressure 1570490.3075 (Pa)
  Temperature 2939.2441 (K)
  Density 1.5365 (kg/m3)
  Entropy 10641.4045 (J/kg-K)
  w2 (wave frame) 1092.2785 (m/s)
  u2 (lab frame) 876.2 (m/s)
  c2 (frozen) 1126.6344 (m/s)
  c2 (equilibrium) 1090.4271 (m/s)
  gamma2 (frozen) 1.2418
  gamma2 (equilibrium) 1.1633

5.2 Python

Load the script demo_CJstate.py in IDLE and run it. The output should be:

========== RESTART: C:\JES\18\cantera\test\Python\demo_CJstate.py ==========
CJ computation for Mevel2017.cti with composition H2:2 O2:1 N2:3.76
Initial conditions: P1 = 1.000e+05 Pa & T1 = 295.00 K
CJ Speed 1968.5 m/s
CJ State
  Pressure 1.573e+06 Pa
  Temperature 2940.0 K
  Density 1.538 kg/m3
  Entropy 1.064e+04 J/K
  w2 (wave frame) 1091.0 m/s
  u2 (lab frame) 877.5 m/s
  c2 frozen 1126.7 m/s
  c2 equilbrium 1090.4 m/s
  gamma2 frozen 1.242
  gamma2 equilbrium 1.163