Installation

1) Copy source code to a local drive
2) Install Miniconda for Python 3.x (if you already have Anaconda or Miniconda installed skip to 4)
3) Update conda

   Command: conda update -n base -c default conda

4)

   Windows: Open Anaconda Prompt

   Linux/Mac: Open Terminal

5)

   Windows: Type: conda env create -f "[frhodo_environment.yml path]"

   Example: conda env create -f "C:\Frhodo\frhodo_environment.yml"

   Linux/Mac: Type: conda env create -f [frhodo_environment.yml path]

   Example: conda env create -f /Users/TSikes/Documents/Frhodo/frhodo_environment.yml

Run

General

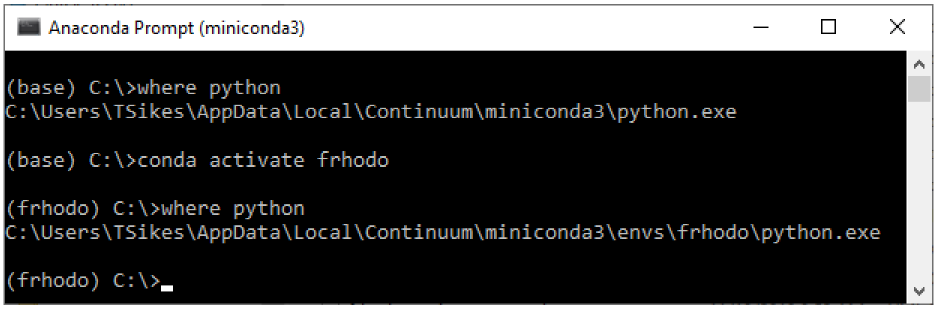
1) Activate Frhodo environment
2) Run main.py from the Frhodo environment

If you have any errors at startup about imports and you have similar packages installed elsewhere, ensure you are pointing to the correct packages in the Frhodo environment. This might require you delete your homebrew version of Python/Cantera/etc.

Windows

1) If using Notepad++ as your IDE continue to a) else refer to b). If you are uncertain where the paths you need you can type “where python” in the Anaconda Prompt and it will give you the location of the activated environment. You can change the activated environment by typing “conda activate [env]” where [env] could be base, frhodo, etc.

   Ex:



```
(base) C:\>where python
C:\Users\TSikes\AppData\Local\Continuum\miniconda3\python.exe

(base) C:\>conda activate frhodo

(frhodo) C:\>where python
C:\Users\TSikes\AppData\Local\Continuum\miniconda3\envs\frhodo\python.exe

(frhodo) C:\>_
```

   My [Main conda path] is C:\Users\TSikes\AppData\Local\Continuum\miniconda3

   My [Conda env path] is C:\Users\TSikes\AppData\Local\Continuum\miniconda3\envs\frhodo

   [main.py path] is main.py’s full file path

a) If using Notepad++ as your IDE then the NppExec (install this through the plugin manager) Execute window should have the following:

```
NPP_SAVE
cd "$(FULL_CURRENT_PATH)"
CMD /C ""[Main conda path]\Scripts\activate.bat" "[Conda env path]" && python -u "[main.py path]""
```

b) If you would like to use another IDE you can call:

```
CMD /C ""[Main conda path]\Scripts\activate.bat" "[Conda env path]" && python -u "[main.py path]""
```

c) If you would like to run from anaconda prompt:

1. Open anaconda prompt
2. Type: conda activate frhodo
3. Type: python "[main.py path]"

File Structure

Experiment Directory Format

Experiment Directory → Shock1.exp, shock1.rho, etc. are within 1 subdirectory of specified directory. If multiple shocks of the same number exist, the highest level one is used.

- Search specified directory and 1st level subdirectories for ".exp" file
- Use ".exp" file's root directory as directory for .rho, and raw signal files

Experimental File Format

The experimental files holds experimental information in a configuration file format. Required fields are:

Parameter	Shorthand Name	Units
Driven section temperature	T1	°C
Driven section pressure	P1	Torr
Driver section pressure	P4	Psi
Pressure transducer spacing	PT Spacing	mm
Avg time between transducers	tOpt	µs
Sample Rate	SampRate	Hz

One solution if your experiment is not a shock tube is to set the mixture within Frhodo and then alter T5 and P5. Then copy the conditions and make that into an .exp file. This the experiment will load as expected whenever you change the shock number.

The incident shock velocity is set by PT Spacing/tOpt. If this does not fit your experiment, you could set tOpt to be 1/conversion factor between the expected units [mm/µs] and your experimental units. Then you can simply set PT Spacing to be your shock velocity.

Below is an example of what the configuration file would look like. Species can be extended beyond 1.

```
[Mixture]
Mol_0_Name="Krypton"
Mol_0_Formula="Kr"
Mol_0_Mol frc=0.960
Mol_1_Name="Cycloheptane"
Mol_1_Formula="cC7H14"
Mol_1_Mol frc=0.040

[Expt Params]
T1=21.000000
P1=5.010000
P4=30.000000
tOpt=116.557292
PT Spacing=120.000000
SampRate=50000000.000000
```

Experimental Data Format

Data is expected to be in CSV format with no headers. Column 1 should be time in µs and column 2 is the experimental observable. Below is an example of the experimental data file.

```
1.493735E-1,3.551242E-4
1.693735E-1,3.529244E-4
1.893735E-1,3.501086E-4
2.093735E-1,3.450050E-4
2.293735E-1,3.379656E-4
2.493735E-1,3.284624E-4
2.693735E-1,3.185193E-4
```

Mechanism Directory Format

Mechanism Directory → [file with “mech” keyword in chemkin format], [file with “thermo” keyword]

File extension	File
.therm	Chemkin format thermodynamics file
.mech or .ck	Chemkin format kinetics file
.tran	Chemkin format transport file (not currently used)
.cti	Cantera format mechanism file

- Chemkin format kinetics file can contain thermo data or it can be a separate file
    - If the thermo data is in mech then the therm file is not needed
- Currently only NASA7 is supported for thermodynamic expressions
- Due to how some of the user interface is currently generated, large mechanisms will take a long time to process, but they will work. The current design is developed for mechanisms < 200 reactions.

Simulation Directory Format

Simulation Directory (program will create if it doesn't exist) → Experimental Set Name → Shock x

- If multiple simulations are saved there will be an additional folder with Sim x and a log file of comments made by the user
- The mechanism can be changed based on a pulldown box which contains all files in the folder with the "mech" keyword

Shock Solver

Coupled shock solver based on FROSH. Assumes frozen chemistry across shockwave. Values for both incident shock and reflected shock are calculated. The incident shock values are used for the incident shock reactor and the reflected shock values are used for the 0D reactors.

Shock properties are automatically populated based on the shock.exp file. It assumes the units from this file to be: P_1 [Torr], T_1 [°C], shock trigger time [µs], and shock trigger distances [mm].

The shock solver works in reverse by assuming the last state changed is set. In other words, if T_2 changed P_2 is held constant.

Mixture

Exp names are autopopulated from shock.exp. Thermo names are autopopulated from the mechanism provided.

The simulation will not run if an unknown exp name exists

Simulation Settings

Incident Shock Reactor

Ode solver can be selected as Radau or BDF. Radau is slightly slower but more accurate at extreme rates, its sample factor must be increased for a smooth curve. BDF is faster otherwise. The difference between the two is on the order of 10s of milliseconds, so probably indistinguishable to the human eye.

The ODE solver uses an adaptive step, initial value problem solver. The sample factor is actually increasing the number of points between the solver's chosen points to interpolate between.

Tables

The mechanism is shown in two formats: Bilbo (log(A)) and Chemkin format. Reactions can be expanded to edit them directly in the program and within memory. Only reactions with a modified Arrhenius or Arrhenius third body definition can be edited. This is a deliberate choice.

Right- click:

General: opens menu to automatically copy the rate coefficients of all expanded reactions, reset all reactions, expand/collapse all reactions, or set dependent reactions.

Coefficients: opens menu to reset box value, copy, paste, or set new reset value

Dependent Reactions

Dependent reactions can be set by mathematical equations which relate various pre exponential factors or other variables. For example: =A2*2+A3/4 would set the variable to (reaction 2's pre exponential factor) *2 + (reaction 3's pre exponential factor)/4

Plot

Draggable items on the plots are: weight dots, weight arrows, and simulation line. The changes made can be found in either the optimization tab or the time offset in the experiment settings tab. Right click on plot with no tools selected for additional options.

For Pan/Zoom, left click pans, right click zooms

For Zoom-to-rect, left click zooms in, right click zooms out.

Shortcuts:

Command	**Shortcut(s)**
Home/Reset	**double right-click plot** or **h** or **r**
Back	**c** or **left arrow** or **backspace**
Forward	**v** or **right arrow**
Pan/Zoom	**p**
Zoom-to-rect	**o**
Cancel Pan/Zoom tools	**esc** while active
Constrain pan/zoom to x axis	hold **x** when panning/zooming with mouse
Constrain pan/zoom to y axis	hold **y** when panning/zooming with mouse
Preserve aspect ratio	hold **ctrl** when panning/zooming with mouse
Toggle major grids	**g** when mouse is over an axes
Toggle minor grids	**G** when mouse is over an axes

The Raw Signal plot shows the schlieren raw signal, if provided, and the averages. The markers are draggable.

Saving

Upon saving a comment box will pop up. This comment box has shortcuts of Ctrl+Enter and Ctrl+S, in addition to hitting Save, to save. This appends to a log of the simulations run. The output files are put into CGS units. If the Output Save Times box is empty, all ODE time steps will be saved. All selected variables will be saved in their own text files. The default is to save for all species and reactions, but this can be changed.

Sim Explorer

The Sim Explorer is for investigating variables beyond just the experimental observable for a given experiment. Multiple variables can be shown Y1 and Y2 (the left y-axis and right y-axis, respectively). Multiple species or reactions can be shown by selecting them with the check box. Legends will be created for multiple species/reactions and can be moved by the user.

Optimization

To begin using the optimization routines, first enable "Load Full Series Into Memory" under the Files tab. Then you can add the experimental series that you are accessing through your choice of Experimental Directory by going to Tables -> drop down menu -> Series Viewer -> Add Series. This should generate a minimized table for you to select which experiments you would like to include in the optimization. The final requirement is that at least one reaction have rate coefficient uncertainties. This is done in one of the mechanism tabs. If a reaction turns purple, then it will be optimized. Additional constraints on the rate expression coefficients can be applied, but are not required. The optimization can be run either through the optimization tab's Run Optimization or through the toolbar Run. An optimization can be aborted through the toolbar Abort.

Additional options on the optimization can be found in the Optimization tab. When computing the loss function, the weight function as shown for each experiment will be used. Modify this to suit your needs on each experiment. Multiprocessing is enabled and works by splitting up simulations to each processor within a given loop. This is to maximize responsiveness to the user.

Bugs/Crashes

If a bug or crash occurs, please copy the text that is shown. Report it and give as much detail about what you were doing within the program for the best chance of us being able to find it and patch it.