Chapter 15

Hints and Tips

The routines provided in the Toolbox are reasonably robust but do not always yield the correct answer or even result in errors that cause the programs halt with error messages. Usually these issues can be resolved by adjusting the parameters that control the routine algorithms. In some cases, you may have to investigate the innards of the toolbox and do some diagnostic work. The routines are not particularly sophisticated and should be considered "research" grade software which does not attempt to prevent user errors or provide particularly verbose error messages. On the other hand, these have been used by successfully by many generations of students and professional researchers. Based on these experiences here are some observations about possible problems and tips for solutions.

Jump Conditions

For certain cases, the jump conditions will not converge. In those instances, it may be necessary to adjust the error bounds and convergence parameters.

Three parameters control the convergence and bounds on the specific volume for the Newton-Raphson iteration used to solve the jump conditions. These are specified in files located in the SDToolbox directory:

MATLAB Function - SDTconfig.m
Python Function - config.py

The default values of these parameters are:

```
ERRFT = 1e-4;
ERRFV = 1e-4;
volumeBoundRatio = 5;
```

The values of the error parameters represent the maximum relative errors allowed for convergence of shock and detonation jump condition computations, see the discussion in Section 8.1. Iteration ceases and the solution is returned when the conditions $\Delta T/T <$ `ERRFT` and $\Delta v/v <$ `ERRFV` are both met.

The value of `volumeBoundRatio` is the lower bound on specific volume ratio v_1/v_2 used as a starting point for the iteration. For shock waves in gases with a high specific heat, higher values of `volumeBoundRatio` may be required in order to get solutions but care must be taken not to select `volumeBoundRatio` larger than the maximum value possible on the Hugoniot. The perfect gas analytical solution for strong shock is a useful estimate if the ratio of specific heats γ is known.

$$\frac{v_1}{v_{2,min}} \geq \frac{\gamma+1}{\gamma-1} \tag{15.1}$$

In rare instances, Cantera may fail to converge to an equilibrium composition. The equilibrium solvers are fairly robust but you may find that there are particular combinations of stoichiometry and thermodynamic state, particularly for exothermic mixtures, that halt with error messages. This can often be solved by

using compositions are slightly displaced from equivalence ratios of precisely unity and for sufficiently rich mixtures, the stoichiometries that determine the possible oxidation states of H (H2 vs H2O) and C (CO2 vs CO vs C) can be problematic. Displacing the compositions slightly from the precise values that define those boundaries may be helpful in obtaining convergence.

ODE Solvers

The functions **cvsolve**, **cpsolve**, **zndsolve** as well as other programs that use ordinary differential equations solvers will require some adjustment of input parameters when used with reaction mechanisms and compositions other than those selected in the demo programs.

Time trouble

If the `t_end` parameter is too small, a peak in the reaction zone energy release will not be found. The solution is simply to increase the value of `t_end`. However, excessively large values of `t_end` can result in a lengthy simulation and a frustrated user. For ZND solutions, if a sonic point is reached within the reaction zone, the solution will be singular and the ode solver will halt with an error message. This can be avoided by either reducing `t_end` or using an events function to halt the ode solver gracefully when the sonic point is approached. This approach is needed to compute so called "eigenvalue" solutions or models of reaction zone structure with area change, friction or thermal energy losses.

Convergence issues

If you have trouble getting a converged solution with the ode solver, this is usually associated with large mechanisms for hydrocarbons. There are often species that are present in very small amounts at the end of the reaction zone and change (decrease) rapidly in the energy release portion of the reaction zone. Although these are usually not significant to resolve in the post-energy release zone, if the solver takes too large a time step, negative species amounts will result in the solver halting with an error message. Cantera will report an error but the difficulty is fundamentally with the ode solver.

The issue is created by the solver automatically adjusting the time step based on the state of the solution and derivatives. This is usually not an issue but can be a problem if there is a sudden change in conditions that the time step algorithm cannot handle properly. This happens within energy release zone for compositions and conditions with long induction zone and short energy release zone. The time step will be increased within the induction zone to sufficiently large values so that rapid decreases in minor species at the end of induction can create problems in the form of negative concentrations, which are an anathema to the thermodynamic state.

There are three approaches to dealing with these problems.

1. Switch solvers.

    a. For python programs, use `LSODA` or `BDF`, these are more robust alternatives to the `Radau` solver that was used in previous versions of the toolbox. A method parameter has been added to the calls and the default is `LSODA`.

    b. For MATLAB programs, try `ode23tb` instead of the `ode15s` that is the default. However, it is often necessary to reduce the maximum time step and tolerance parameters.

2. Reduce the tolerance paramters, `absTol`, `relTol`

3. Reduce the `max_step` parameter

Examine the species (particularly the minor species) near the energy release region to determine what sort of `abs_tol` and `max_step` are needed. The values can be surprisingly small in order to avoid oscillations in species concentrations.

Underdriven detonations

An underdriven or sub-CJ detonation is shock wave with $U < U_{\mathrm{CJ}}$. A ZND reaction zone simulation of an underdriven case will always terminate in a sonic singularity and the solver will halt with an error message. The solution is valid up to this point but if will be necessary to reduce `t_end` or add a events function to the ode solver to enable the solver to halt normally and output the solution. If reaction zone length or time scale estimates are needed for sub-CJ cases, constant pressure or constant volume simulations should be used.

If the `postshock_eq` function is called with $U < U_{\mathrm{CJ}}$, a solution may be returned that is not valid. Always check the CJ speed and only use results from equilibrium postshock computations for $U \geq U_{\mathrm{CJ}}$.

Weak Shocks

The shock jump conditions only have solutions for $U > a$ where a is the sound speeds. Attempts to solve the jump conditions with U close to or smaller than a will either fail with an error or result in an invalid solution. It is good practice to compare the magnitude of the shock speed with the sound speed before computing shock jump conditions.