

# CANTERA

Object-Oriented Software for Reacting Flows

## Cantera is a suite of software tools for reacting flow problems

- Thermodynamic and transport properties
- Non-ideal equations of state
- Chemical equilibrium
- Reactor networks
- Steady 1D flames
- Reaction path diagrams
- Heterogeneous chemistry
- Open source
- Object-oriented
- Multi-Platform
- Available at <http://www.cantera.org>

## Cantera is multilingual

- Cantera can be used from several popular programming / problem-solving environments
- Interactive / scripting environments (MATLAB, Python) for rapid problem solution and software prototyping
- Fortran 90 and C++ for large-scale computation
- Additional interfaces can be developed for any application that can call C functions (Excel, Mathematica, ...)



MATLAB Toolbox

Python Package

Fortran 90 module

C++ Class Library

# Users new to Cantera should begin with Python or MATLAB

## ■ MATLAB

- Commercial product of The MathWorks, Inc.
- Object-oriented scripting language
- Many 'toolboxes' available for control, digital signal processing, numerical mathematics, graphics, etc.



## ■ Python

- An easy-to-use, elegant object-oriented scripting language
- Free, open-source, multiplatform
- Very good for rapid prototyping
- Many modules available for graphics, numerical mathematics, image processing, encryption, etc.
- <http://www.python.org>



# Cantera has a similar 'look and feel' in all environments

## MATLAB

```
gas = IdealGasMix('mech.xml');  
setState_TPX(gas,300.0,OneAtm,'CH4:1,O2:2,N2:7.52');  
equilibrate(gas,'HP');  
disp(gas)
```

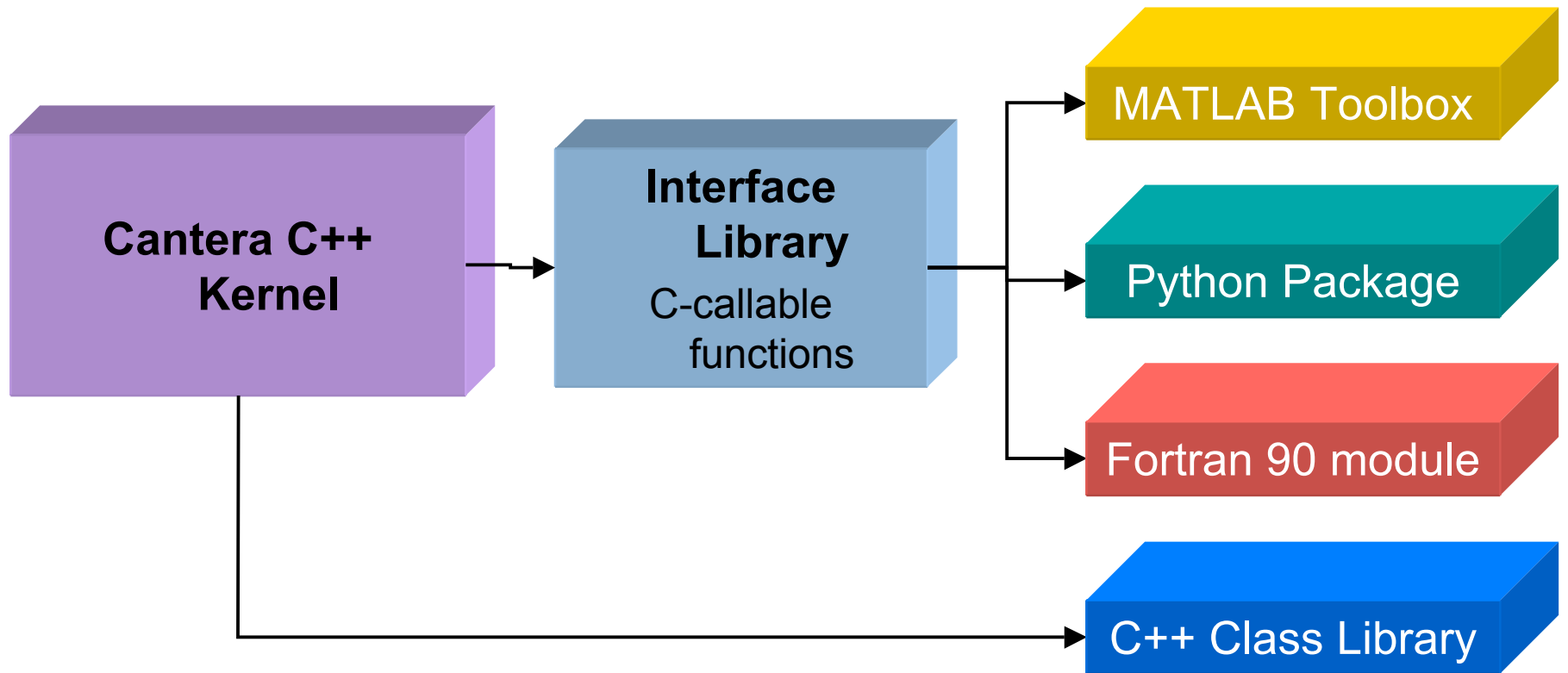
## Python

```
from Cantera import *  
gas = IdealGasMix('mech.xml')  
gas.setState_TPX(300.0,OneAtm,'CH4:1,O2:2,N2:7.52')  
gas.equilibrate('HP')  
print gas
```

## Fortran 90

```
program equil  
use cantera  
type(gas_t) gas  
gas = IdealGasMix('mech.xml')  
call setState_TPX(gas, 300.0,OneAtm,'CH4:1,O2:2,N2:7.52')  
call equilibrate(gas,'HP')  
call printSummary(gas)  
end
```

# All interfaces use a common C++ kernel



# The Kernel

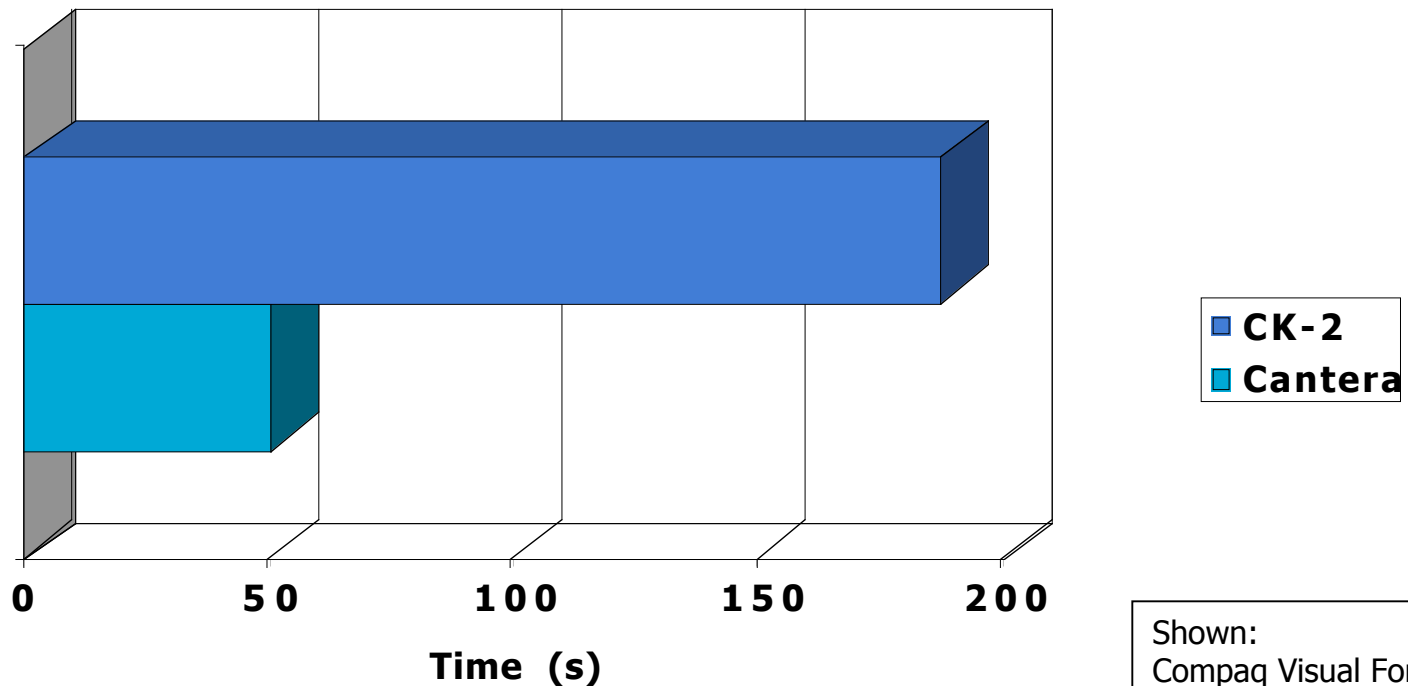
- A C++ class library
- Designed for performance
  - Property caching
  - Virtual methods used sparingly
  - Templates used to allow inlining at compile time
  - Standard Template Library container classes used
  - CPU-intensive code hand-optimized
- Uses standard open-source numerical libraries
  - BLAS, LAPACK, CVODE

# Performance and Validation



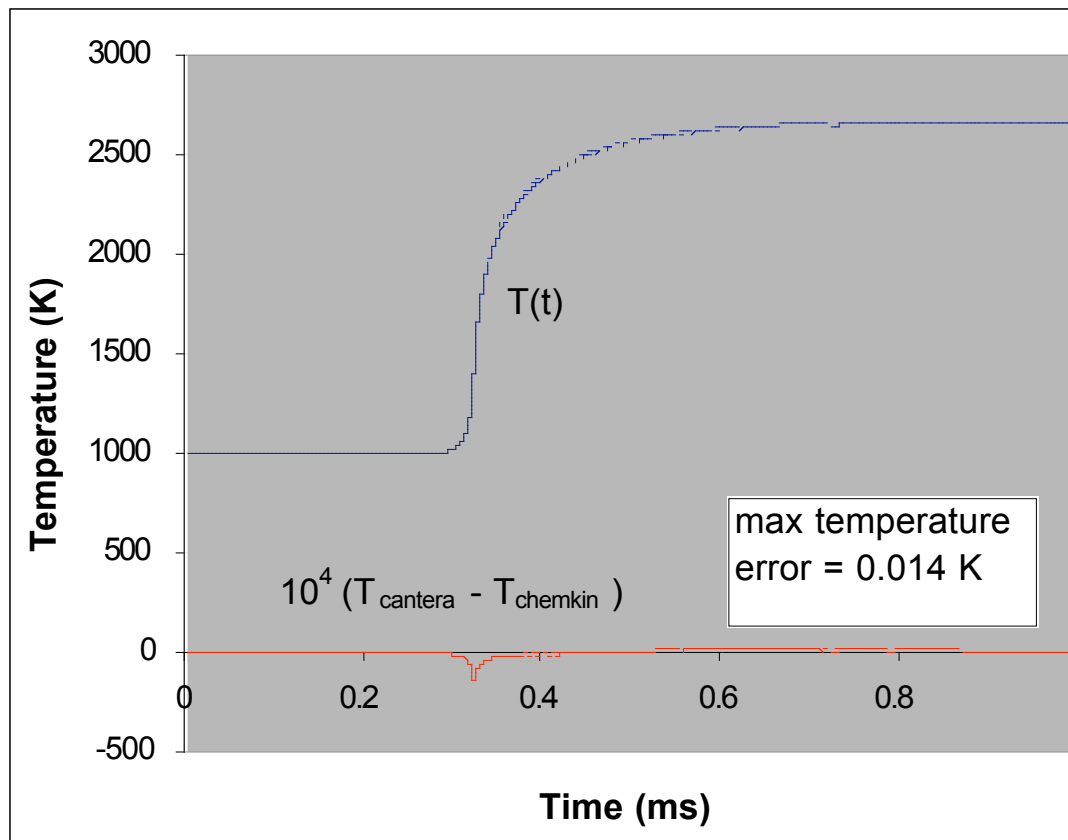
# Benchmark kinetics performance vs. Chemkin-II

Constant P, H Problem with Sensitivity Analysis  
Similar performance gain on all platforms  
(Linux, Windows, OSF/1, IRIX)



Shown:  
Compaq Visual Fortran  
MS Visual C++  
750 MHz Pii

## Validation: Cantera and Chemkin-II produce essentially identical results



Cantera and Chemkin-2 are in excellent agreement for all solution variables and sensitivity coefficients.

Typical relative difference in any component at any time is one part in  $10^6$  or  $10^7$

## Kinetics performance due to efficient rate of progress computation

### ■ Property caching

- Expensive reaction rates only recomputed when temperature actually changes
- Saves time in Jacobian evaluation

### ■ Concentration products

- Reactions with one, two, or three reactants handled explicitly
- Example: a three-reactant reaction of species  $k_1$ ,  $k_2$ , and  $k_3$ :

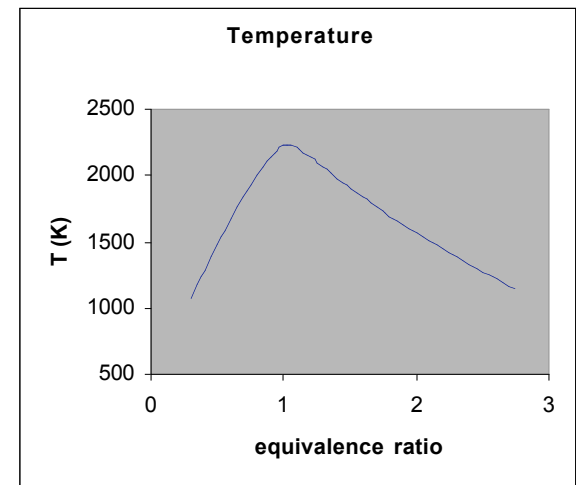
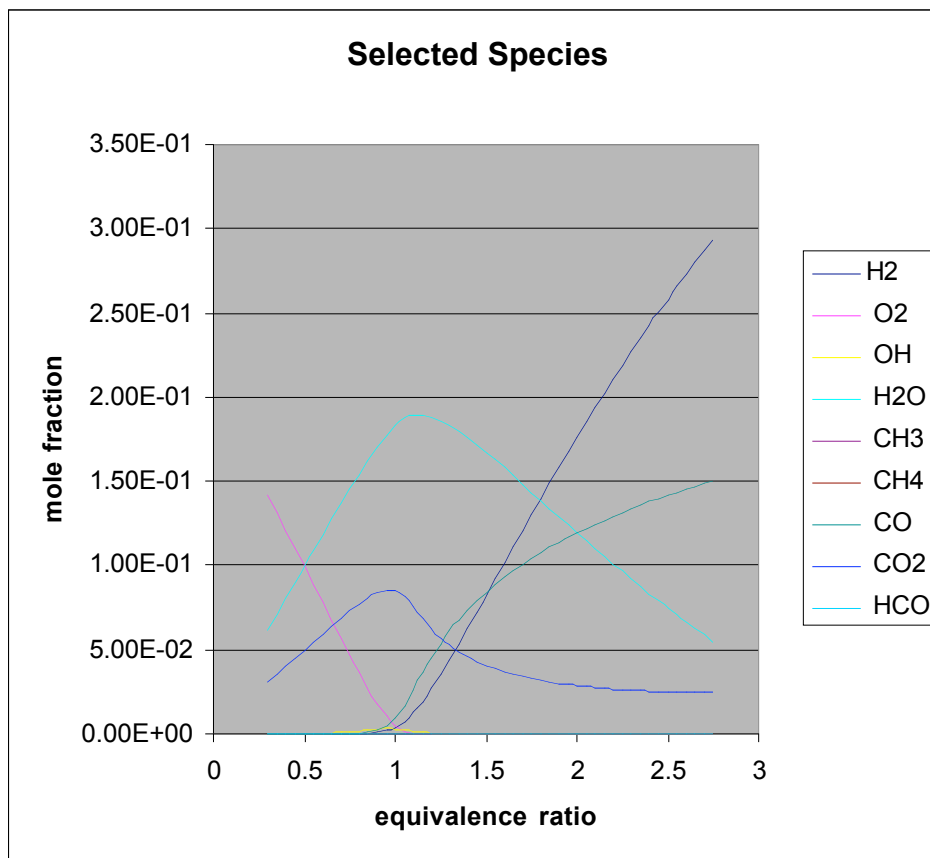
general form  $\prod_k C_k^{\nu_{ki}^{(r)}}$  replaced by  $C_{k_1} C_{k_2} C_{k_3}$

# Chemical Equilibrium

# Chemical Equilibrium

```
g = IdealGasMix('mech.xml')  
g.setState_TPX(300.0, OneAtm, 'CH4:1,O2:2,N2:7.52')  
g.equilibrate('HP')
```

The 'equilibrate' method sets the gas to a state of chemical equilibrium



Example: equilibrium methane/air adiabatic combustion at 1 atm

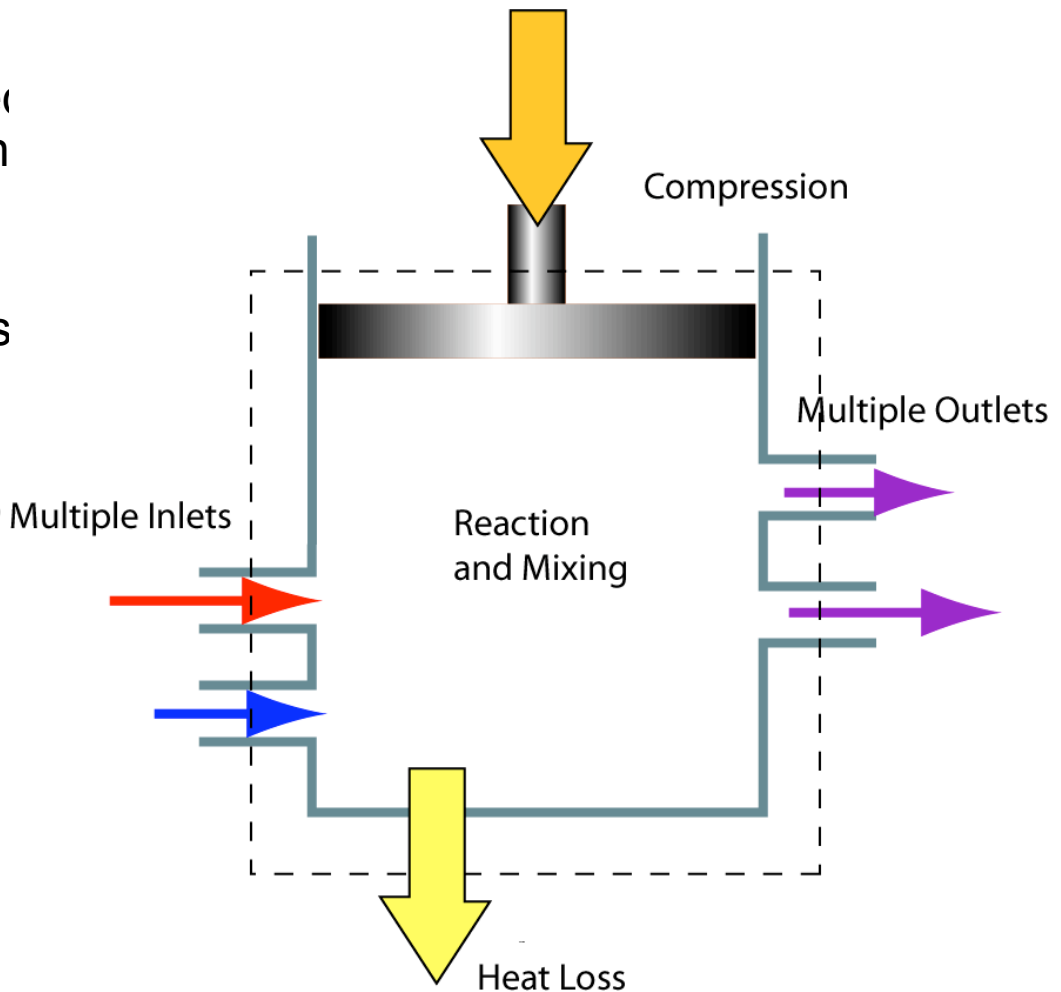
# Chemical Equilibrium Details

- Element potential method used
  - One of several equivalent ‘non-stoichiometric’ algorithms (Smith and Missen)
  - Idea dates back to ~1959, and used in NASA equilibrium program in early 60’s
  - Popularized in combustion community by STANJAN code of Reynolds in 70’s
- Element potentials are the chemical potentials of the atomic vapor species
  - Given these, all other chemical potentials can be computed from equation of reaction equilibrium for the atomization reactions
  - Choose element potentials; compute partial pressures, total pressure
  - Adjust until P and elemental composition have desired values
- Requires solving nonlinear system of M algebraic equations
  - M is usually  $< 5$
  - Variations on Newton’s method work well, if initial estimate is close enough
  - Get initial estimate by solving linear programming problem

# Zero-Dimensional Kinetics

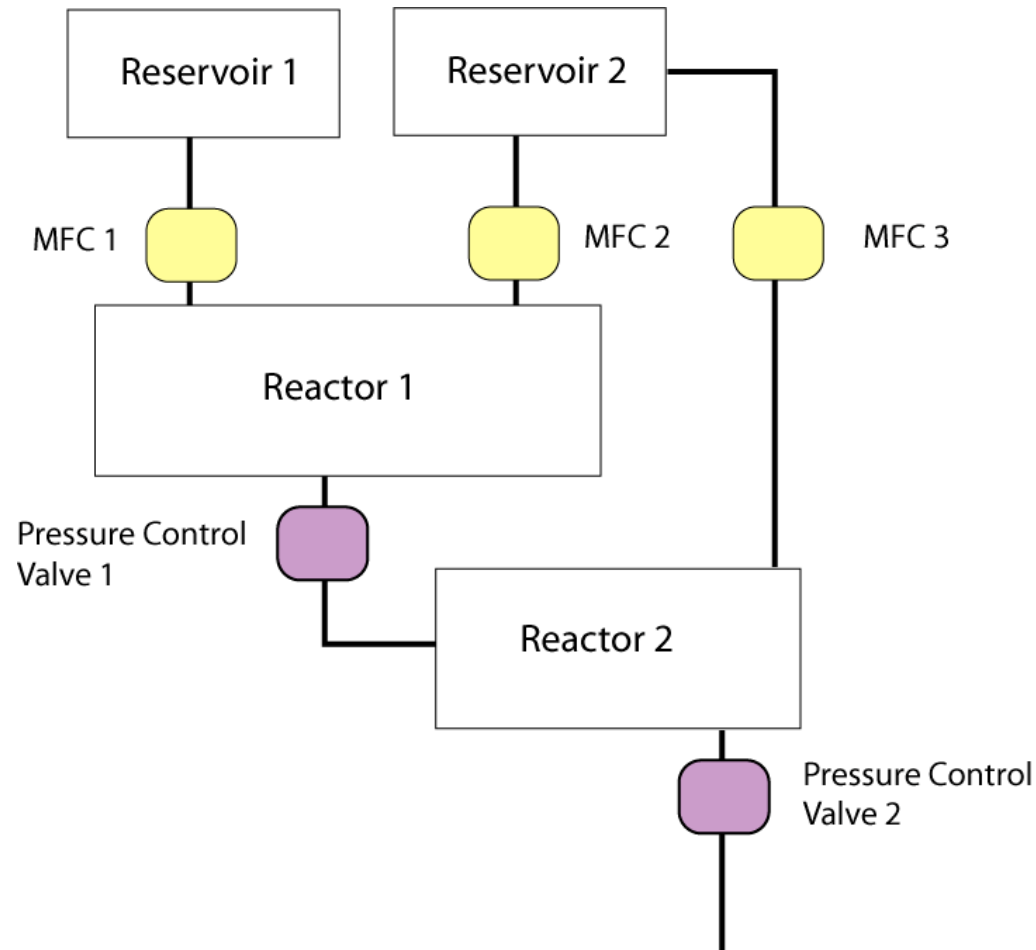
# Stirred Reactor Models

- Generic transient stirred reactor model can be used to build many different batch and continuous reactors
- Mass flow rates, heat loss volume may all be varied
- Reactors can be linked to create complex process models





## Reactors may be connected in arbitrary networks



Reservoirs provide specified inputs

Each reactor may use a different mixture model or reaction mechanism

Can assemble many different processes from a small set of components

All inputs may be time-dependent

## Reactor Details

- Primitive conservation equations are integrated, with no assumption of equation of state or boundary conditions
- **Same** reactor objects may be used to build constant-volume reactors, constant-pressure ones, CSTRs, etc.

$$\frac{dM_k}{dt} = \sum_{\text{inlets}} \dot{m}_i Y_{k,i} - Y_k \sum_{\text{outlets}} \dot{m}_o + V \hat{m}_k \dot{\omega}_k$$

$$\frac{dU}{dt} = \sum_{\text{walls}} A_i \dot{q}_i - P\dot{V} + \sum_{\text{inlets}} \dot{m}_i h_i - h \sum_{\text{outlets}} \dot{m}_o$$

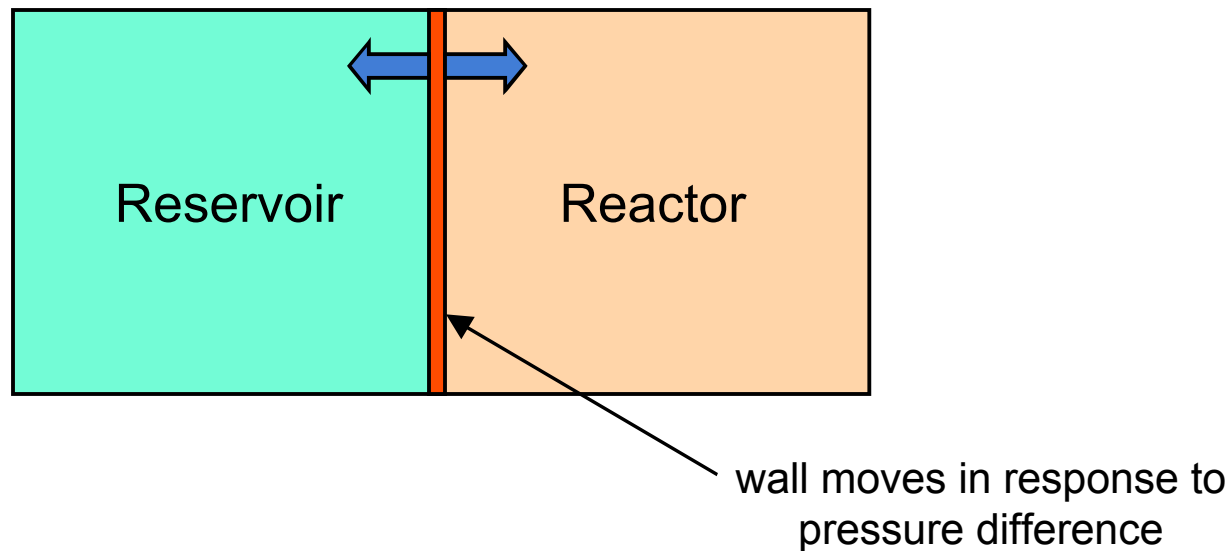
$$Y_k = M_k/M$$

$$T = T(U/M, Y_1, \dots, Y_K)$$

Temperature is determined by Newton iteration

## Building a constant-pressure reactor

- Define two objects – a reactor, and a reservoir
- Connect them with a wall, and set the wall expansion parameter to a large value
- Reactor pressure will be held to reservoir value for sufficiently large  $K$



$$\dot{V} = K A (P_1 - P_2)$$

## code to create a constant-pressure reactor

### Python

```
gas = GRI30()
gas.setState_TPX(1001.0, OneAtm,
                 'H2:2,O2:1,N2:4')
r = Reactor(gri3)

env = Reservoir( Air() )
w = Wall(r,env)
w.set(K = 1.0e6)
w.set(A = 1.0)
```

### MATLAB

```
gas = GRI30;
set(gas,'T',1001.0,'P',oneatm,'X','H2:2,O2:1,N2:4');

r = Reactor
insert(r, gas);

env = Reservoir;
a = IdealGasMix('air.xml');
insert(env, a);

w = Wall;
install(w, r, env)
setExpansionRateCoeff(w, 1.0e6);
setArea(w, 1.0);
```

# Running the reactor

## Python

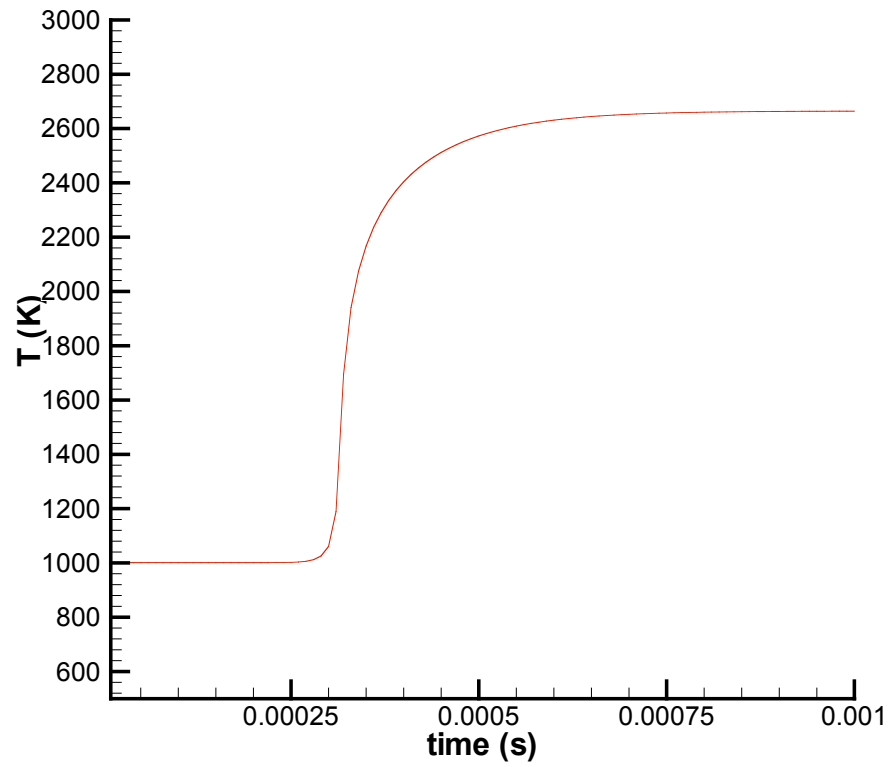
```
time = 0.0
dt = 1.0e-5
for n in range(100):
    time += dt
    r.advance(time)
    print r.time(), r.temperature()
```

## MATLAB

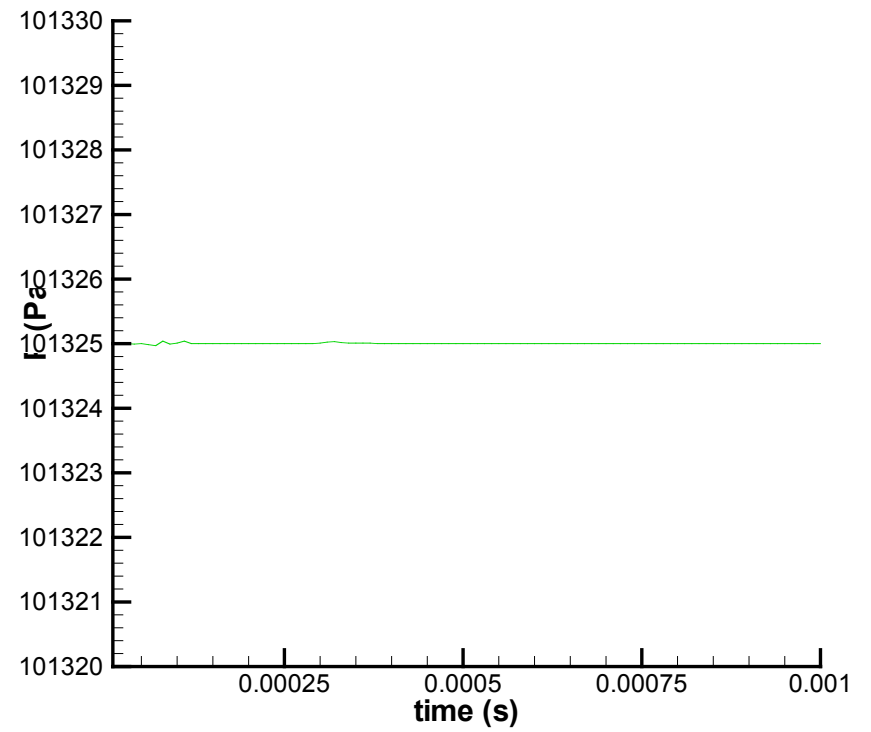
```
dt = 1.0e-5;
for n = 1:100
    t = t + dt;
    advance(r, t);
    disp([time(r) temperature(r)]);
end
```

# Results

## Temperature



## Pressure



# Transport Properties

# Transport Models

- Interchangeable transport property 'managers' handle all transport-related tasks
- All have a common interface
- May be swapped dynamically during a simulation
- Allows adaptive transport property evaluation
  - In regions of small gradients, use simple, fast models
  - In regions of high gradients (boundary layers, flamefronts), use accurate models



## MultiTransport: A multicomponent transport model for ideal gas mixtures

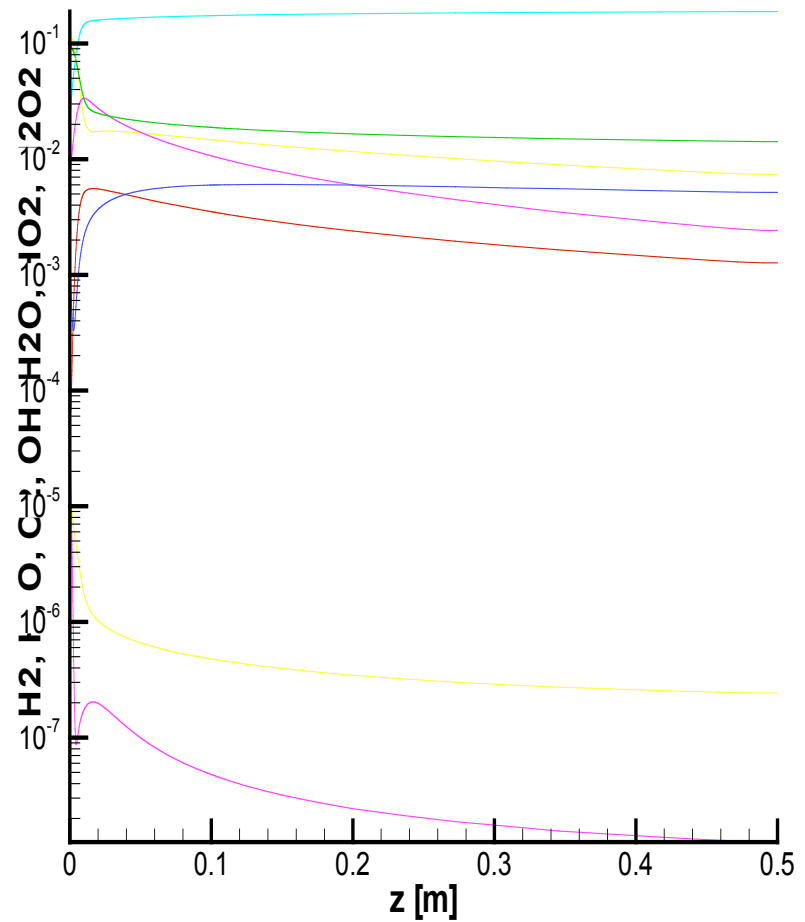
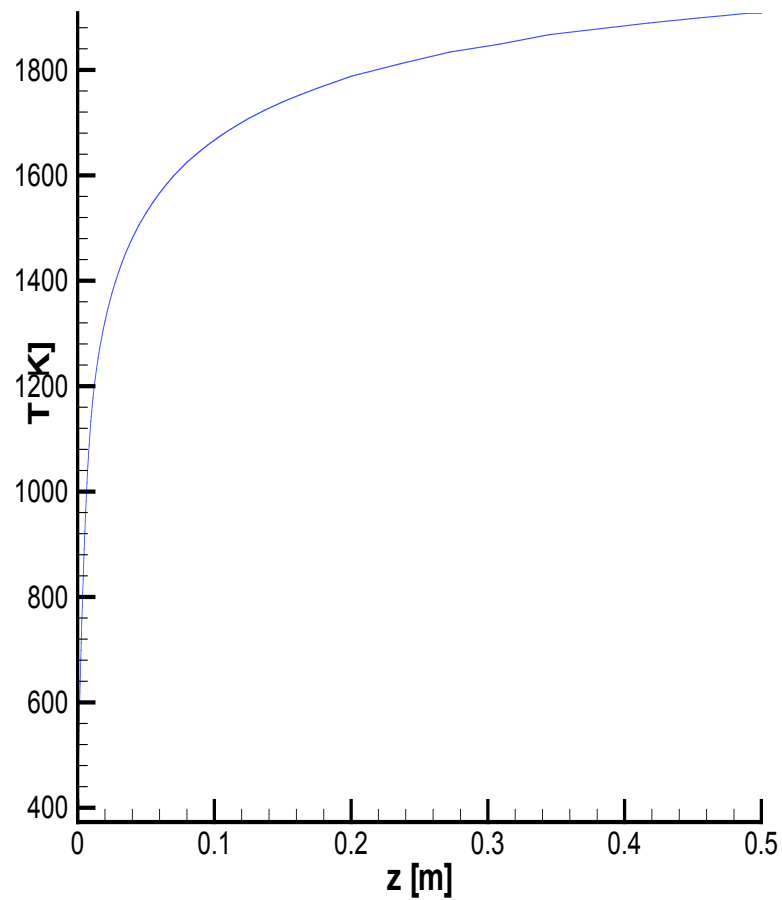
- Implements a multicomponent model based on that of Kee, Dixon-Lewis, Warnatz, Coltrin and Miller (1986).
- Computes:
  - Viscosity
  - Binary diffusion coefficients
  - Multicomponent diffusion coefficients
  - Thermal conductivity
  - Thermal diffusion coefficients
- Enhancements over Kee et al. implementation:
  - Optional use of GMRES cuts time to solve L-matrix by 50%
  - Slightly more accurate and faster collision integral fits
  - Direct computation of fluxes given gradients avoids L-matrix inversion

# One-Dimensional Flames

## One-Dimensional Flames

- Burner-stabilized and axisymmetric stagnation-point flames implemented
- At present, flame simulations run only from Python interface
- Solution technique is hybrid Newton / time-stepping algorithm similar to that of TWOPNT (Grcar)
- Adaptive grid refinement
- Fast evaluation of Jacobian
  - Only steady-state Jacobian computed
  - Transient Jacobian formed by modifying diagonal elements

# Example: A low-pressure, burner-stabilized hydrogen / oxygen / argon flame



## A complete Python flame simulation script

```
from Cantera import units
from Cantera.flame import *
gas = IdealGasMix(src = 'h2o2.xml', transport='Mix')
flame = BurnerFlame(domain = (0, 0.4), fuel = 'H2:1',
                    oxidizer = 'O2:1, AR:7', gas = gas,
                    grid = [0, 0.02, 0.04, 0.06, 0.08,
                            0.1, 0.15, 0.2, 0.49, 0.5])
flame.set(mdot = 0.04, equiv_ratio = 0.9, T_burner = 373.0,
         pressure = 0.05 * units.atm, tol = (1.e-5, 1.e-12),
         timesteps = ([1,2,5,10,20], 1.e-5),
         refine = (2.0, 0.8, 0.9), jac_age = (20, 10))
flame.set(energy = 'off')
flame.solve(1)
flame.set(energy = 'on', refine = (2.0, 0.05, 0.1))
flame.solve(1)
flame.save('energy', 'solution with the energy equation enabled',
         'h2o2_flame1.xml')
```

**(for a version with comments, see ‘Cantera/python/examples/flame1.py’)**

# Installing Cantera

## Installation CD

- An installation CD is available that contains everything needed to install Cantera on a PC running Windows.
- The CD contents can also be downloaded from <http://www.cantera.org>



# Installing Cantera on a Windows PC



setup.bat

- If you are installing Cantera for the first time...
  - Run 'setup.bat' on the CD.
  - This will install Cantera and a few third-party packages needed by Cantera
  
- If you are updating a previous installation
  - uninstall Cantera first from the Control Panel
  - run update.bat on the CD

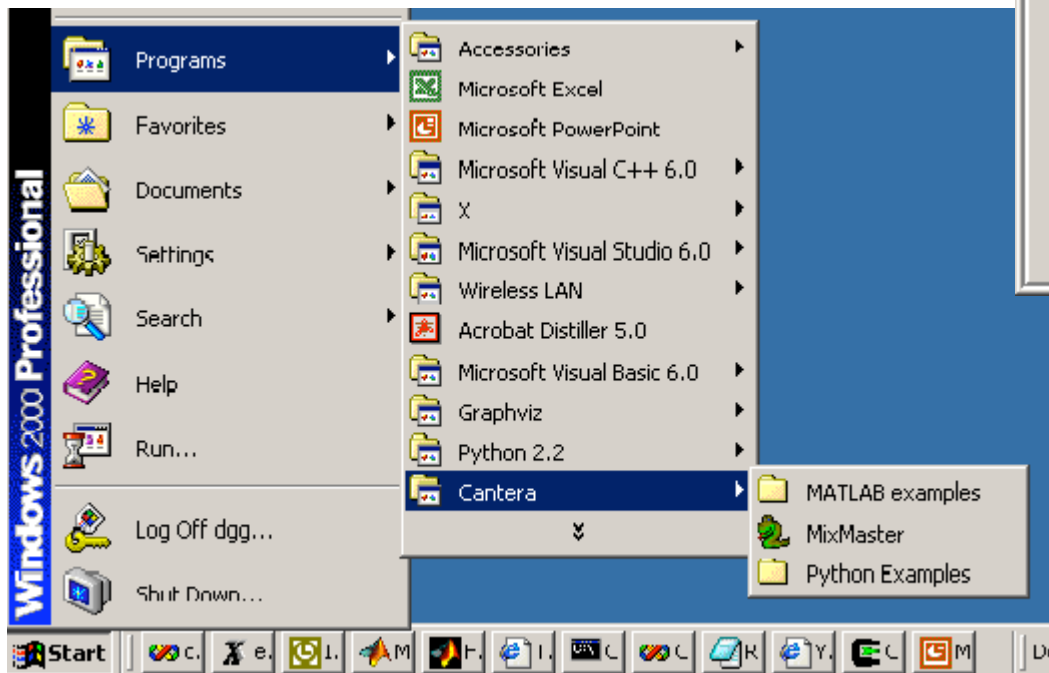
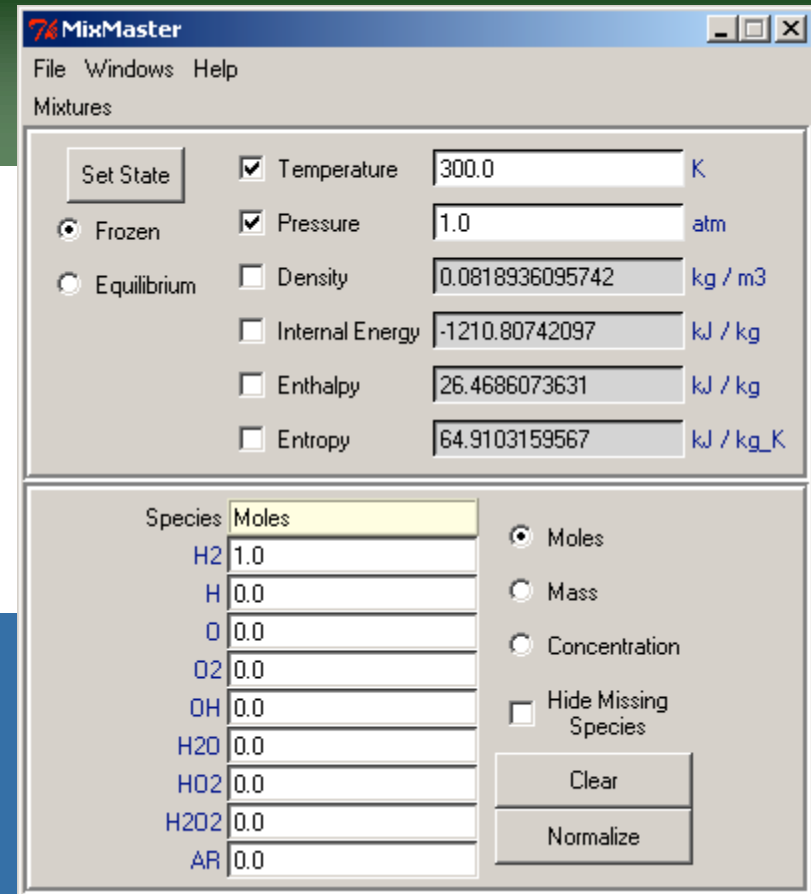


## Third-Party Software Packages

- **Python 2.2.** An easy-to-use object-oriented scripting language
- **Numeric extensions for Python.** Adds efficient array operations to Python.
- **GraphViz** Used by Cantera to draw reaction path diagrams
- These packages are also available on the web

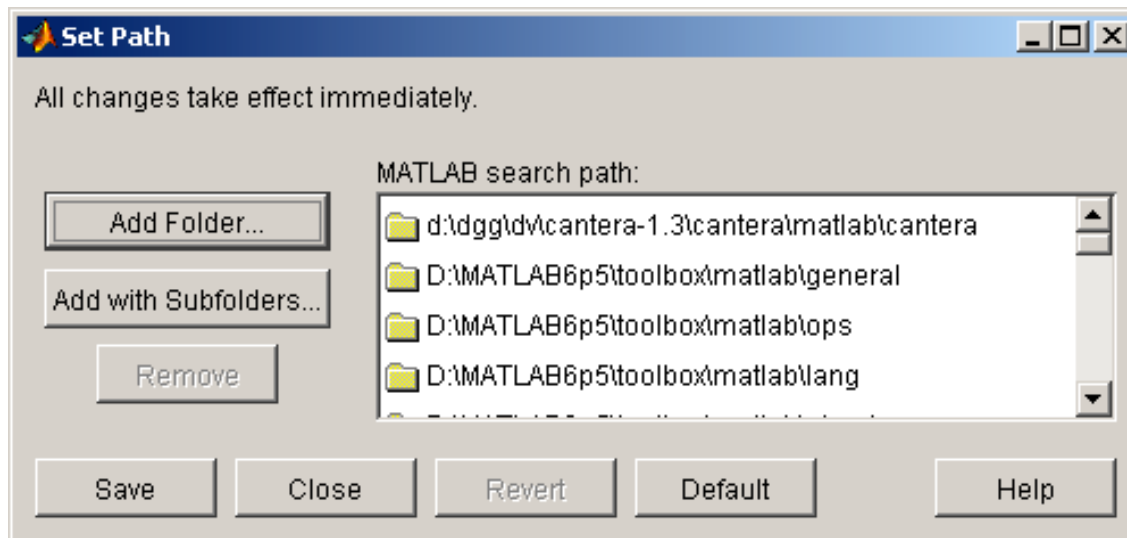
# Testing the Installation

- On the Start menu, select **Programs/Cantera/MixMaster**
- If a graphical window like that at the right appears, you're all set!



# Installing the Cantera MATLAB Toolbox

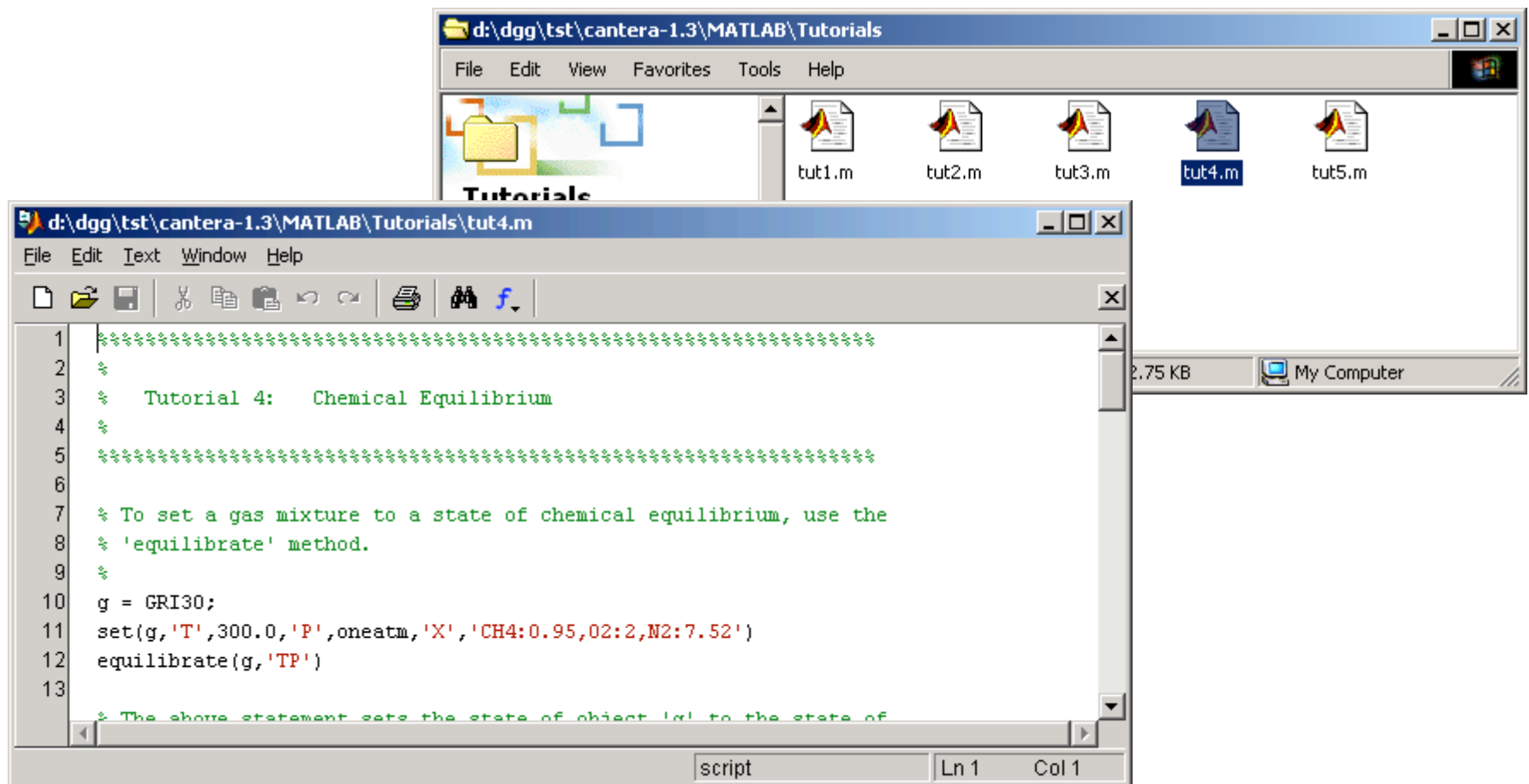
- In the installation directory, go into folder MATLAB and unzip file Cantera-matlab.zip.
- This file expands to a folder named 'cantera', which is the Cantera MATLAB toolbox. You can move it to any convenient place on your disk.



Now start MATLAB, and on the File menu select 'Set Path', and add this folder to the MATLAB path.

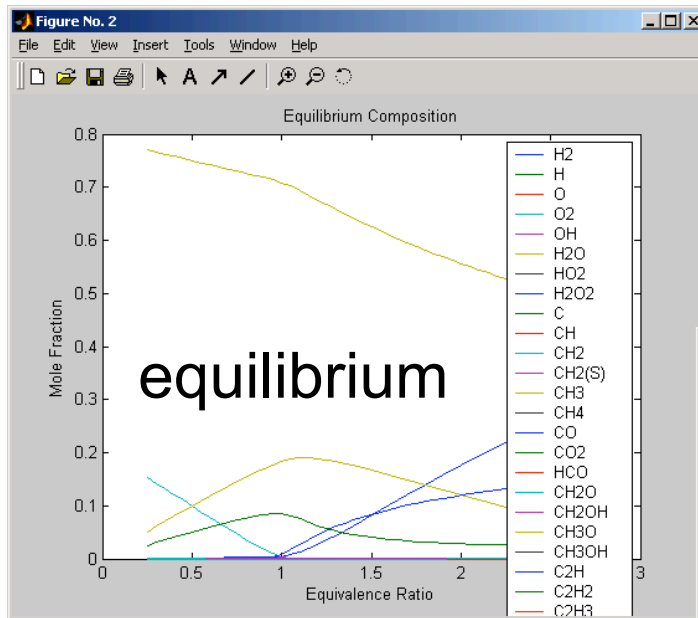
# A set of tutorials covers the basics of using Cantera in MATLAB

- Each tutorial is an m-file
- For each one, read it first, then run it

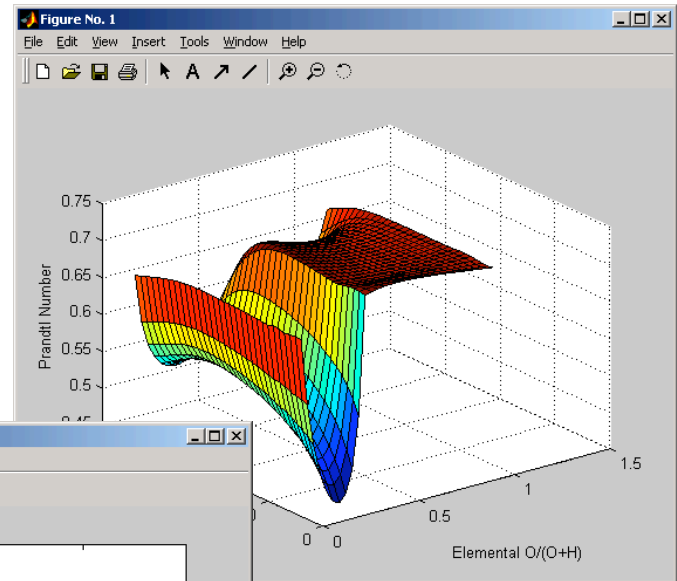
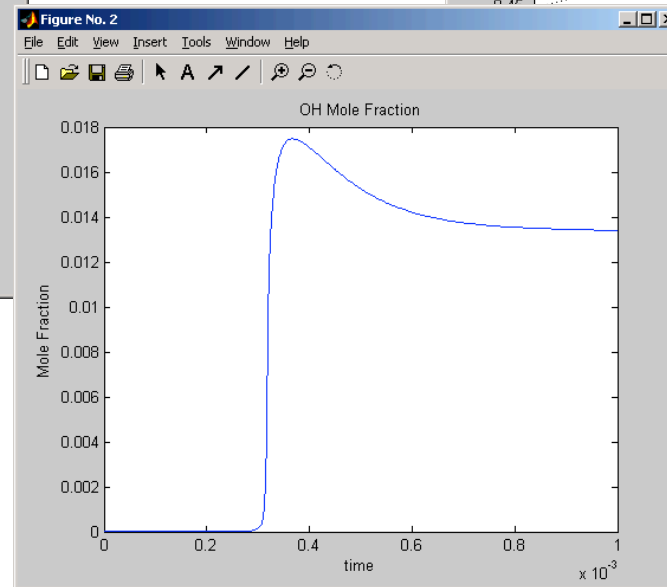


# A set of examples provide starting points for writing your own m-files

- Running file 'run\_examples.m' in the Examples folder will run all examples.



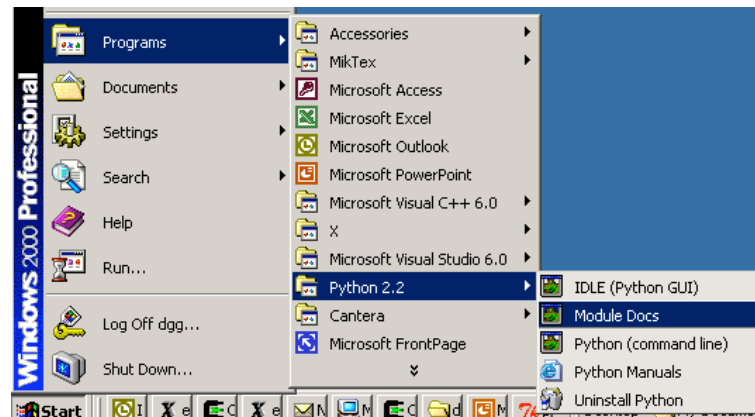
zero-dimensional  
kinetics



transport  
properties

# Getting started with Cantera in Python

- The installation procedure installs Python for you
- Try running the example Python scripts
  - zero-D kinetics
  - flames
- To get help, select 'Module Docs' on the Start menu under 'Python 2.2'
  - press 'Open Browser'
  - browse to Cantera, and select any module to view its documentation



# Reaction Mechanism File Formats

- Cantera supports two formats to specify reaction mechanisms
  - **CTML format**
    - an XML-based markup language
    - Python syntax to create an object from a specification in CTML format:

```
gas = IdealGasMix('mech.xml')
```
    - CTML file contains all required thermo and transport parameters
  - **Chemkin format**
    - Widely used
    - Python syntax:

```
gas = IdealGasMix('mech.inp', thermo = 'therm.dat',  
                  trandb = 'tran.dat',  
                  transport = 'Mix')
```
    - Transport and thermo parameters optional
    - call to IdealGasMix also creates a CTML representation of the mechanism

## Carrying on

- See the Cantera web site <http://www.cantera.org>
- Join the Cantera User's Group to receive information about new releases, bugs/bug fixes, etc.
- Frequent updates will be available that expand the capabilities of Cantera – stay tuned!