# FlameMaster Flame Calculator

Combustion Summer School

2018

Prof. Dr.-Ing. Heinz Pitsch

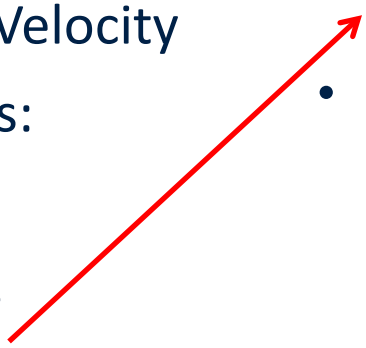
Institut für Technische Verbrennung

RWTH AACHEN UNIVERSITY

# Course Overview

## Part I: Fundamentals and Laminar Flames

- Introduction
- Fundamentals and mass balances of combustion systems
- Thermodynamics, flame temperature, and equilibrium
- Governing equations
- Laminar premixed flames: Kinematics and Burning Velocity
- Laminar premixed flames: Flame structure
- Laminar diffusion flames
- FlameMaster flame calculator



- Introduction
- Counterflow diffusion flame
- Flamelet structure of diffusion flames
- FlameMaster flame calculator
- Single droplet combustion

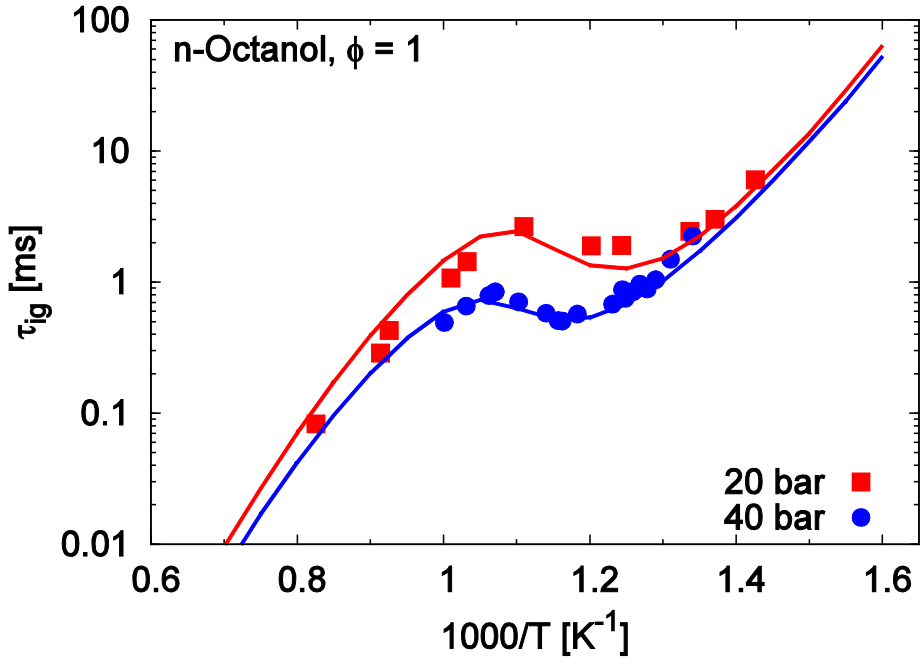
# FlameMaster Flame Calculator

- FlameMaster: A C++ Computer Program for 0D Combustion and 1D Laminar Flame Calculations
  - Premixed and non-premixed
  - Steady and unsteady
  - Emphasis on pre- and post-processing
    - Sensitivity analysis
    - Reaction flux analysis
- At request, available online at https://itv.rwth-aachen.de/index.php?id=13

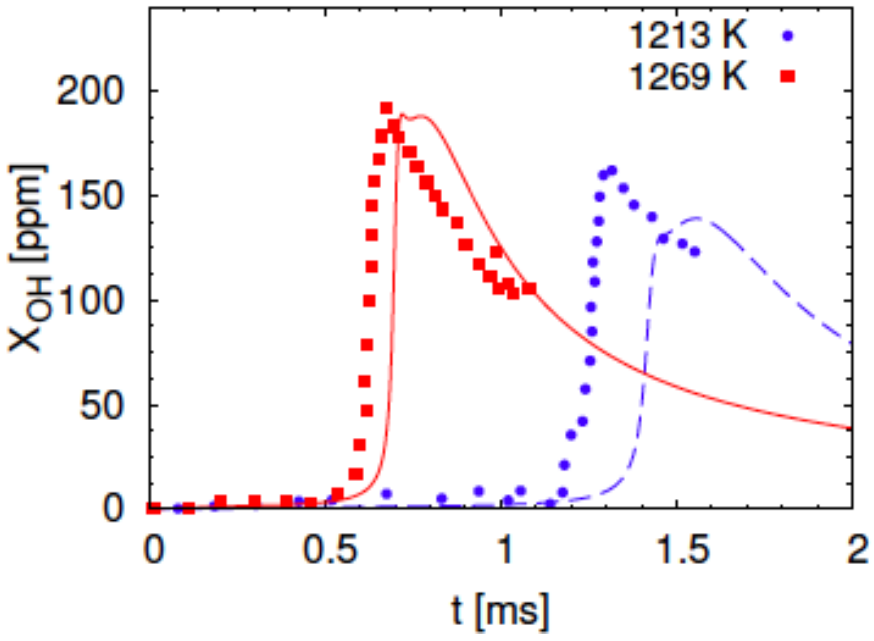
# FlameMaster Flame Calculator

- Example: Shock tube, homogeneous reactor

N-Octanol Ignition Delay Times

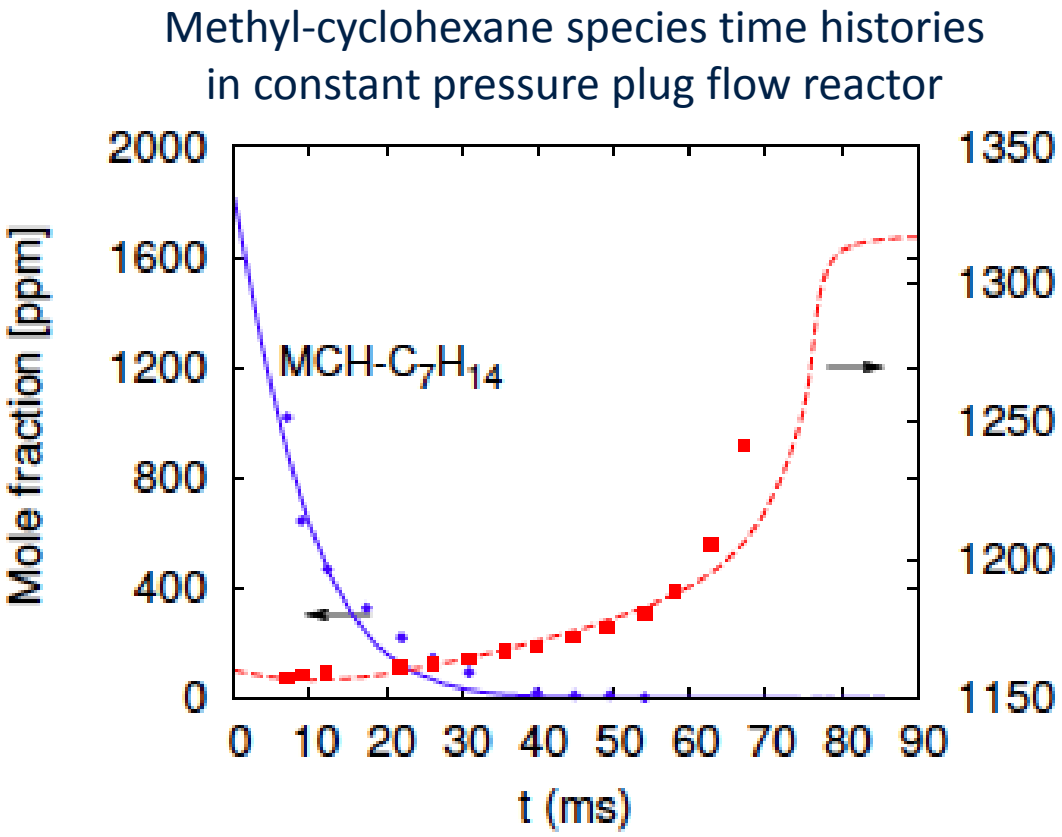



Methyl-cyclohexane species time histories in shock tube



# FlameMaster Flame Calculator

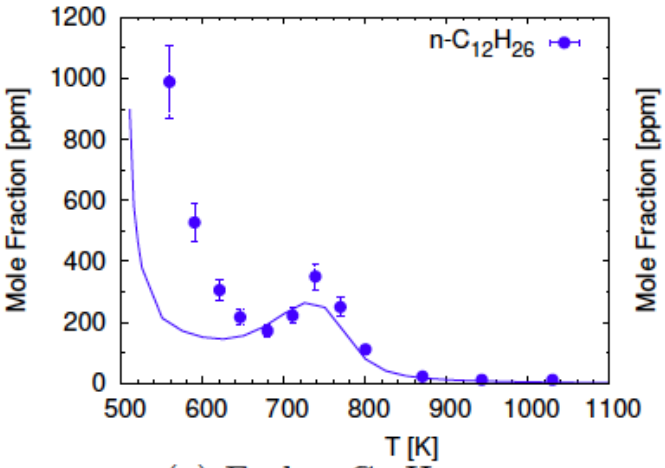
- Example: Flow reactor



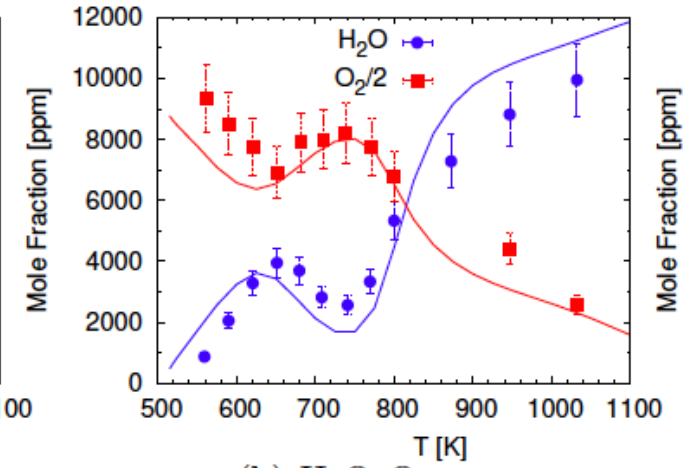
# FlameMaster Flame Calculator

- Example: Jet stirred reactor

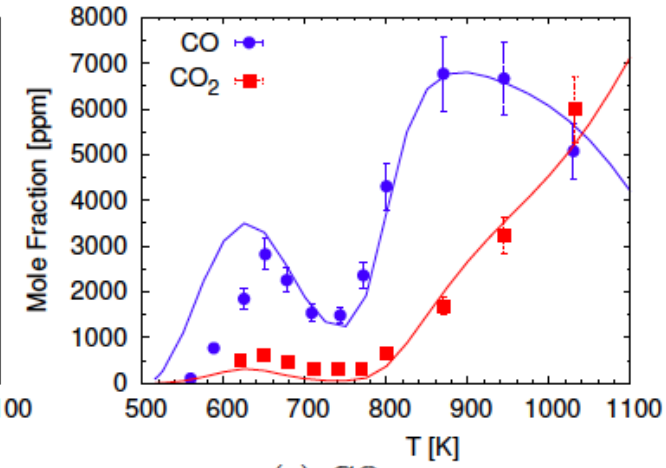
N-Dodecane oxidation in jet stirred reactor




(a) Fuel, $n$-$C_{12}H_{26}$



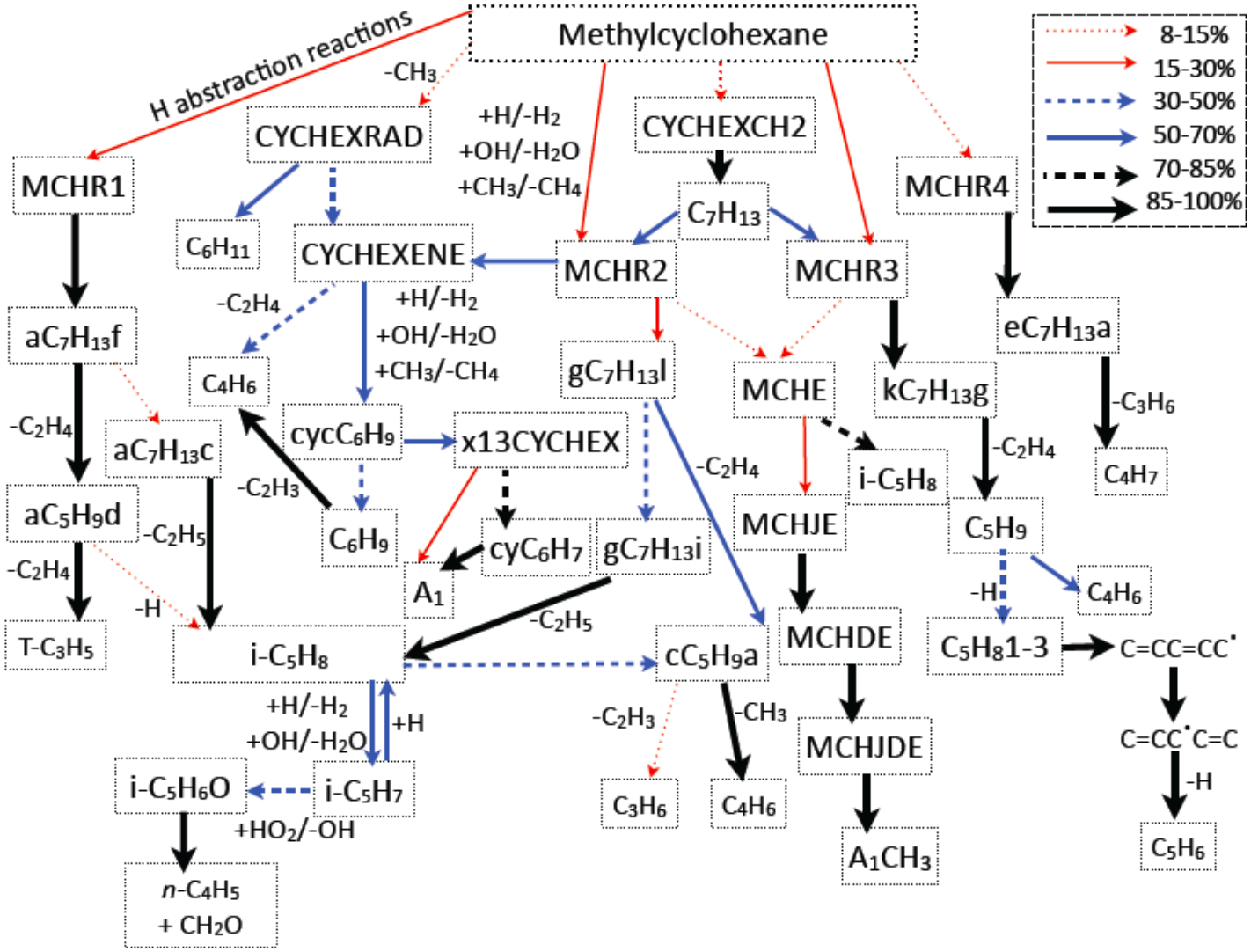

(b) $H_2O, O_2$




(c) $CO_x$

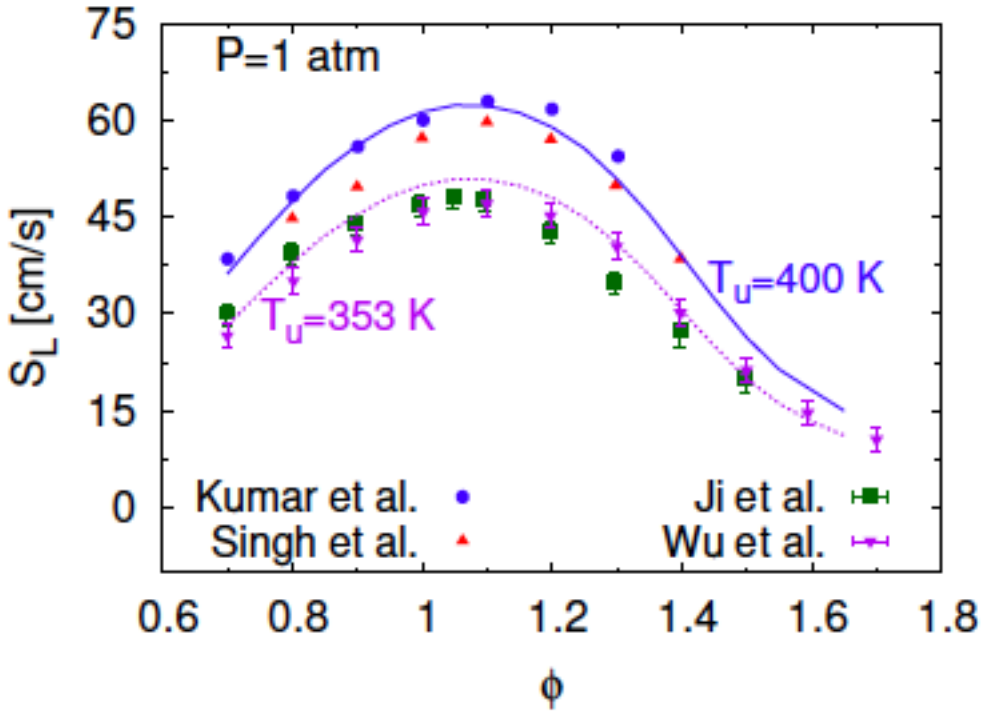
# FlameMaster Flame Calculator

- Example: Reaction flux analysis



# FlameMaster Flame Calculator

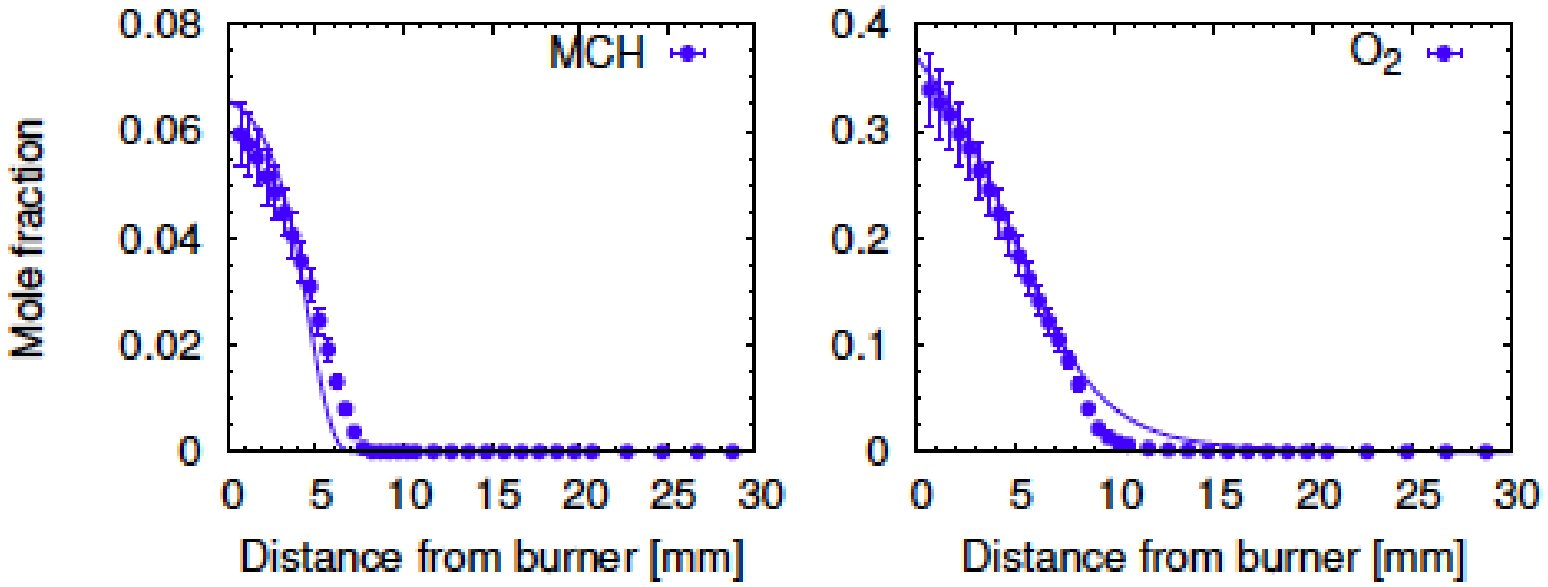
- Example: Laminar burning velocities



# FlameMaster Flame Calculator

- Example: Premixed flame structure

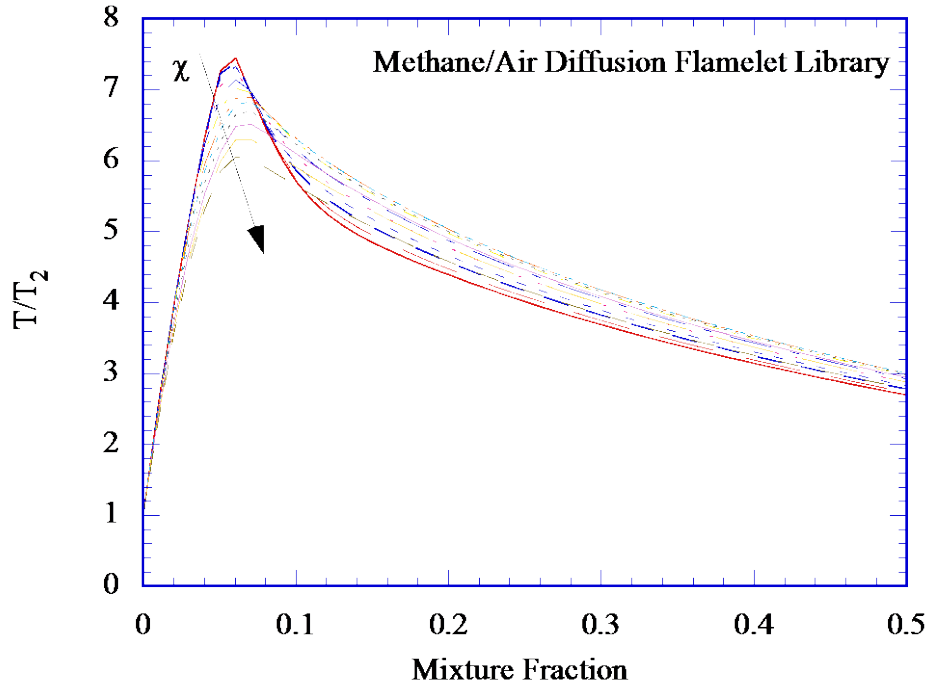
Methyl-cyclohexane species profiles in
premixed burner stabilized flame

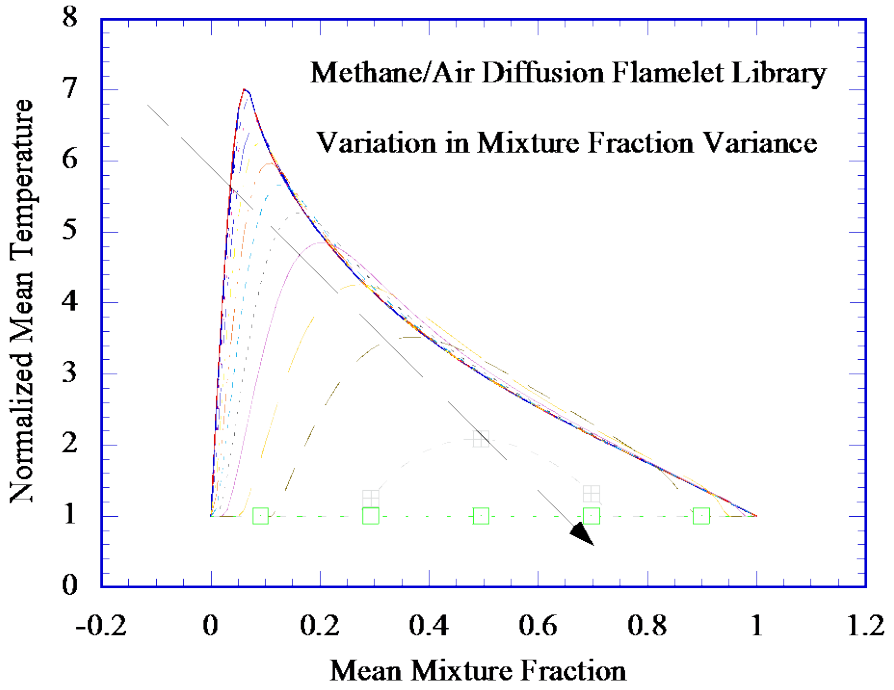


# FlameMaster Flame Calculator

- Example: Flamelet libraries

Flamelet library for methane/air non-premixed combustion



# FlameMaster Tutorial

- Distribution and installation
- Code structure
- Pre-processing
- Examples:
  - Homogeneous reactor
  - Premixed flames
  - Diffusion flames
  - Unsteady flamelet

# FlameMaster Tutorial: Distribution and Installation

- Code available at request, available online at
  https://itv.rwth-aachen.de/index.php?id=13
- Distribution for Linux, MacOS, and Windows systems
- Prerequisites
  1. C and C++ compiler (C11/C++11 compliant)
  2. CMake
  3. Flex, Bison
  4. Sundials (optionally installed by FlameMaster)
- Installation on Linux/MacOS

```
tar -xf FlameMaster.zip
cd FlameMaster
mkdir Build && cd Build
cmake ../Repository -DCMAKE_BUILD_TYPE=Release -DINSTALL_SUNDIALS=ON
make install -j4
```

- Set up environment

```
cd .. && source Bin/bin/Source.bash
```

- Create Mechanisms

```
cd Run/ScanMan && bash CreateAllMechanisms.bash
```

**Further information:**
`Repository/README.md`

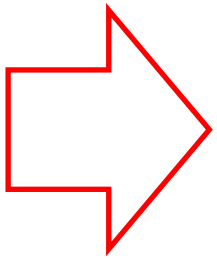
**Operating System specific instructions:**
`Repository/doc/`

**Consider using your package manager**

# FlameMaster Tutorial: Installation on Windows I

- Multiple options
- Avoid spaces or special characters in directories and file names

- Option 1: Install Cygwin including all prerequisites (http://cygwin.org/)
- Option 2: Install Linux in a Virtual Box (https://www.virtualbox.org/)



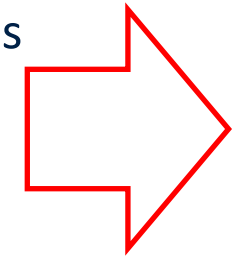
Use Linux-instructions

**Option 1/2 are the recommended options**

# FlameMaster Tutorial: Installation on Windows II

- Option 3: Install FlameMaster natively without Cygwin
  - Install only Compilers, CMake, and Python
  - FlameMaster installs flex, bison, and sundials
  - During the installation of CMake and Python
    - Choose to add the executables to the PATH
  - Extract FlameMaster to `C:\FlameMaster`
  - Commands for the installation

```
cd C:\FlameMaster
mkdir Build && cd Build
cmake ../Repository -DCMAKE_BUILD_TYPE=Release
-DINSTALL_SUNDIALS=ON
cmake --build . --config Release --target
install
```



**Option 3/4 are in beta**

**Possible issues:**
Fixing requires solid understanding of C++ and CMake

**Further instructions:**
`Repository/doc/Windows.md`

- Option 4: Install FlameMaster from a binary installer
  - Install FlameMaster to `C:\FlameMaster`
- Set up environment
  - Double click: `C:\FlameMaster\Bin\bin\Source.bat`
- Create Mechanisms

```
cd Run\ScanMan && CreateAllMechanisms.bat
```

# FlameMaster Tutorial: Directory layout

| Directory | Purpose |
|---|---|
| Repository/ | Permanent directory for the source code. Do all changes here. |
| Build/* | Intermediate directory containing configuration information ("CMake cache") and object files |
| Bin/* | Installation directory containing binaries and scripts that set up your environment (for bash, csh, zsh, bat) |
| Run/* | Directory containing examples |
| Data/* | Directory checked by ScanMan and FlameMan via the shell variable FM_DATA to find binary input files |

***Do not modify automatically generated files. Changes are overwritten if you rerun the installation. Instead, modify the respective template files in** `Repository/`

# FlameMaster Tutorial: Code Structure

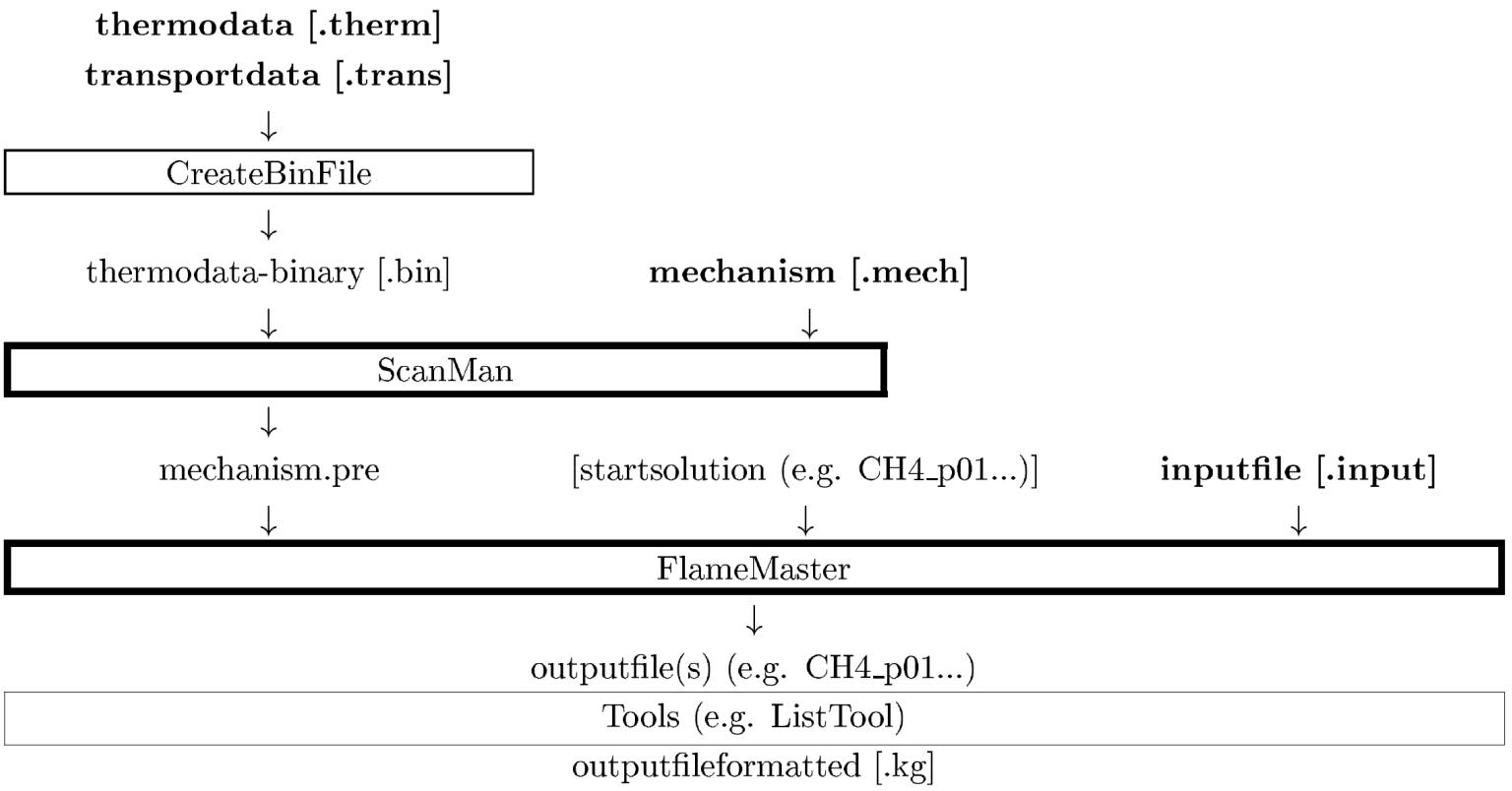
- Four different parts (main executables):
  1. Preprocessing of thermo- and transport data: `CreateBinFile`
  2. Preprocessing of chemical mechanism: `ScanMan`
  3. Simulation: `FlameMan`
  4. Post-processing tools: e.g. `ListTool`

- All tools have help function: Type <Executable> -h

- Main input files
  1. Mechanism
     - Thermodata file
     - Transport data file
     - Mechanism file
  2. FlameMaster input file
     - Information about configuration, boundary and initial conditions, post-processing options, …

# FlameMaster Tutorial: Code Structure

- Four different parts (main executables):
  1. Preprocessing of thermo- and transport data: `CreateBinFile`
  2. Preprocessing of chemical mechanism: `ScanMan`
  3. Simulation: `FlameMan`
  4. Post-processing tools: e.g. `ListTool`



# FlameMaster Tutorial: Proproc. Thermo/Trans

- `CreateBinFile`
  - Creates a binary file `thermotransout` to be used by `ScanMan` including all thermodata and transport data
  - Input files are thermodata file and transport data file
  - Reads Chemkin II format

Example: Thermodata file

```
THERMO
       300      1000      5000
HE                120186HE  1               G  0300.00   5000.00  1000.00      1
 0.02500000e+02 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00    2
-0.07453750e+04 0.09153489e+01 0.02500000e+02 0.00000000e+00 0.00000000e+00    3
 0.00000000e+00 0.00000000e+00-0.07453750e+04 0.09153488e+01                   4
CH3CHCHO          000000H   5O   1C   3     G    300.00   5000.00 1000.00      1
 3.71333831E+00 2.33955799E-02-1.07842958E-05 1.82368810E-09 0.00000000E+00    2
-4.55057438E+03 7.42946144E+00 1.54196914E+00 3.03297391E-02-1.88922690E-05    3
 5.92263358E-09-7.53762316E-13-4.01761098E+03 1.83707011E+01                   4
AR                120186AR  1               G   300.000  5000.000  1000.000    1
 2.50000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00    2
-7.45375000E+02 4.36600000E+00 2.50000000E+00 0.00000000E+00 0.00000000E+00    3
 0.00000000E+00 0.00000000E+00-7.45375000E+02 4.36600000E+00                   4
N2                121286N   2               G   300.000  5000.000  1000.000    1
 2.92664000E+00 1.48797680E-03-5.68476000E-07 1.00970380E-10-6.75335100E-15    2
-9.22797700E+02 5.98052800E+00 3.29867700E+00 1.40824040E-03-3.96322200E-06    3
 5.64151500E-09-2.44485400E-12-1.02089990E+03 3.95037200E+00                   4
H                 L 7/88H   1   00   00   00G   200.000  3500.000   1000.00    1
 2.50000001E+00-2.30842973E-11 1.61561948E-14-4.73515235E-18 4.98197357E-22    2
 2.54736599E+04-4.46682914E-01 2.50000000E+00 7.05332819E-13-1.99591964E-15    3
 2.30081632E-18-9.27732332E-22 2.54736599E+04-4.46682853E-01                   4
H2                TPIS78H   2   00   00   00G   200.000  3500.000   1000.00    1
 3.33727920E+00-4.94024731E-05 4.99456778E-07-1.79566394E-10 2.00255376E-14    2
-9.50158922E+02-3.20502331E+00 2.34433112E+00 7.98052075E-03-1.94781510E-05    3
 2.01572094E-08-7.37611761E-12-9.17935173E+02 6.83010238E-01                   4
```

- Syntax:

```
CreateBinFile -i <thermo> -m <transport> -o <thermotransout>
```

- Example:

```
cd Run/ScanMan/GRI3.0
CreateBinFile -i gri.30.therm -m gri.30.trans -o gri.30.bin
```

# FlameMaster Tutorial: Proproc. Mechanism File

- ScanMan
  - Creates a binary file to be used by `FlameMan` including all thermodata, transport data, and reaction mechanism data <mechanism>.mech
  - Input files are binary file `thermotransout` created by `CreateBinFile` and reaction mechanism file
  - ScanMan has its own format for mechanism file, but also reads Chemkin II format
  - Resulting output file should typically be copied to directory `Data`

```
Let allowed atoms be C, H, O, N, He, AR.
Let additional species be N2,CO,CO2.
Let temperature exponent be n_k.
Let order of reaction be n.
Let units for A be [ cm^(3(n-1)) / ( s * mole^(n-1) * K^n_k ) ].
Let units for E be [ kJ / mole ].

#=============================================================================#
#                                                                             #
#                          H2/CO Chemistry from                               #
#                                                                             #
#            "An optimized kinetic model of H2/CO combustion"                 #
#                  Davis, Joshi, Wang, and Egolfopoulos                       #
#                  Proc. Comb. Inst. 30 (2005) 1283-1292                      #
#                                                                             #
#=============================================================================#

# Reactions of H2/O2

1f: H + O2 -> O + OH     { a = 1.040E+14 n = 0.000 E =  63.957 }
2f: O + H2 -> H + OH     { a = 3.818E+12 n = 0.000 E =  33.254 }
3f: O + H2 -> H + OH     { a = 8.792E+14 n = 0.000 E =  80.207 }

4f: H2 + OH -> H2O + H   { a = 2.160E+08 n = 1.510 E =  14.351 }
5f: OH + OH -> O + H2O   { a = 3.340E+04 n = 2.420 E =  -8.075 }

6f: H2 + M41 -> H + H + M41 { a = 4.577E+19 n = -1.400 E = 436.726 }
7f: H2 + AR -> H + H + AR   { a = 5.840E+18 n = -1.100 E = 436.726 }
8f: H2 + HE -> H + H + HE   { a = 5.840E+18 n = -1.100 E = 436.726 }
```

Example: Mechanism file

# FlameMaster Tutorial: Proproc. Mechanism File

- ScanMan (Mechanism file in FlameMaster format)
  - Syntax:

    ```
    ScanMan -i <Mechanismfile>.mech -t <thermotransportoutput>
    <mechanismbinfile>.pre
    ```

  - Example:

    ```
    cd Run/ScanMan/GRI3.0
    ScanMan -i gri.30.mech -t gri.30.bin -rs3 > gri.out
    mv gri.30.pre ../../../Data
    ```

- ScanMan (Mechanism file in Chemkin format)
  - `CreateBinFile` not required
  - Syntax:

    ```
    ScanMan -i <Mechanismfile>.chmech -t <thermodata> -m
    <transportdata> -f chemkin
    ```

  - Example:

    ```
    cd Run/ScanMan/GRI3.0_CK
    ScanMan -i grimech30.dat -t thermo30.dat -m transport.dat -f
    chemkin -rs3 > gri.out
    ```

# FlameMaster Tutorial: FlameMaster Simulation

- `FlameMan` performs simulations for different configurations
- Main input file called `FlameMaster.input` (default name) or name specified on command line using `-i` option
- All parameters and inputs are provided in `FlameMaster.input` file
- Some parameters can be specified by command line options, which overwrite input file values

```
#######
# I/O #
#######

WriteEverySolution is TRUE
OutputPath is ./Output
NOutputs = 50

#############
# Chemistry #
#############

MechanismFile is Base.pre
globalReaction is 2H2 + O2 == 2H2O ;

fuel is H2
oxidizer is o2

#########
# Flame #
#########

Flame is Isochor Homo Reactor
#Flame is Isobar Homo Reactor

#phi = 0.5
phi = 1.0
#phi = 2.0

Pressure = 13e5
Pressure = 1e5

######################
# Boundary conditions #
######################

ContInc = -0.1
ContType is Temperature
ContBound = 0.7

InitialCond {
    t = 1.7
    X->N2 = 0.79
    X->O2 = 0.21
}
```

Example: FlameMaster input file

## Typical input file (part 1)

- Comments start with #
- Typical statement is
  <Keyword> = <value>
  <Keyword> is <value>
- Most <Keywords> have default values, usually `FALSE`, if applicable

- I/O
  - Outputpath

- Chemistry
  - Mechanism input file
  - Global reaction (for equivalence ratio)
  - Fuel and oxidizer (for equivalence ratio)

```
#######
# I/O #
#######

WriteEverySolution is TRUE
OutputPath is ./Output
NOutputs = 50

##############
# Chemistry #
##############

MechanismFile is Base.pre
globalReaction is 2H2 + O2 == 2H2O ;

fuel is H2
oxidizer is o2
```

Example: FlameMaster input file

## Typical input file (part 2)

- Configuration
  - Flame is …
  - `phi` is equivalence ratio
  - Several `phi` and `pressure` can be specified
  - If no `phi` is specified, it will be computed from initial composition

```
#########
# Flame #
#########

Flame is Isochor Homo Reactor
#Flame is Isobar Homo Reactor

#phi = 0.5
phi = 1.0
#phi = 2.0

Pressure = 13e5
Pressure = 1e5

########################
# Boundary conditions #
########################

ContInc = -0.1
ContType is Temperature
ContBound = 0.7

InitialCond {
    t = 1.7
    X->N2 = 0.79
    X->O2 = 0.21
}
```

Example: FlameMaster input file

# FlameMaster Tutorial: FlameMaster Simulation

## Typical input file (part 2)

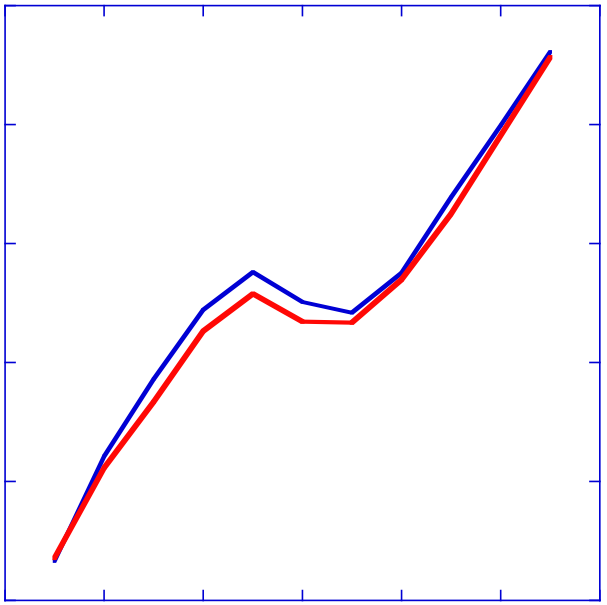
- Boundary and initial conditions
  (here initial condition for homogeneous reactor)
  - Initial composition either from phi and global reaction or specification in `InitialCond`
  - `X->N2` is mole fraction
  - `Y->N2` is mass fraction
  - T or t is temperature
    - If T < 10, then value is 1000/T
  - Continuation in temperature, if `ContInc` is non-zero
    - `ContType is Temperature` means continuation in temperature
    - Continuation ends, if `ContBound` is reached
    - ContInc is increment between two simulations. If ContInc < 10, then increment is applied to 1000/T
    - In the example, temperature is varied from 1000/T = 1/1.7 to 1/0.7 corresponding to 588 to 1000

```
#########
# Flame #
#########

Flame is Isochor Homo Reactor
#Flame is Isobar Homo Reactor

#phi = 0.5
phi = 1.0
#phi = 2.0

Pressure = 13e5
Pressure = 1e5

########################
# Boundary conditions #
########################

ContInc = -0.1
ContType is Temperature
ContBound = 0.7

InitialCond {
    t = 1.7
    X->N2 = 0.79
    X->O2 = 0.21
}
```

Example: FlameMaster input file

# FlameMaster Tutorial: Homogeneous Reactor

Constant volume homogeneous reactor computing ignition delay times for *n*-heptane/air mixtures



- `Example:`
  `cd Run/FlameMan/Prem/0D/nHeptane`
  `FlameMan`

- This will create a file `Output/N-C7H16_IgniDelTimes.dout` which has the values of ignition delay times for all computed conditions as tab-separated list

- If the flag `WriteEverySolution is TRUE` is added to the input file, then mass fractions and temperatures as function of time are given in files like `Y1_N-C7H16_p13phi100to0833.dout`
- File names include information on fuel, pressure, equivalence ratio, and initial temperature

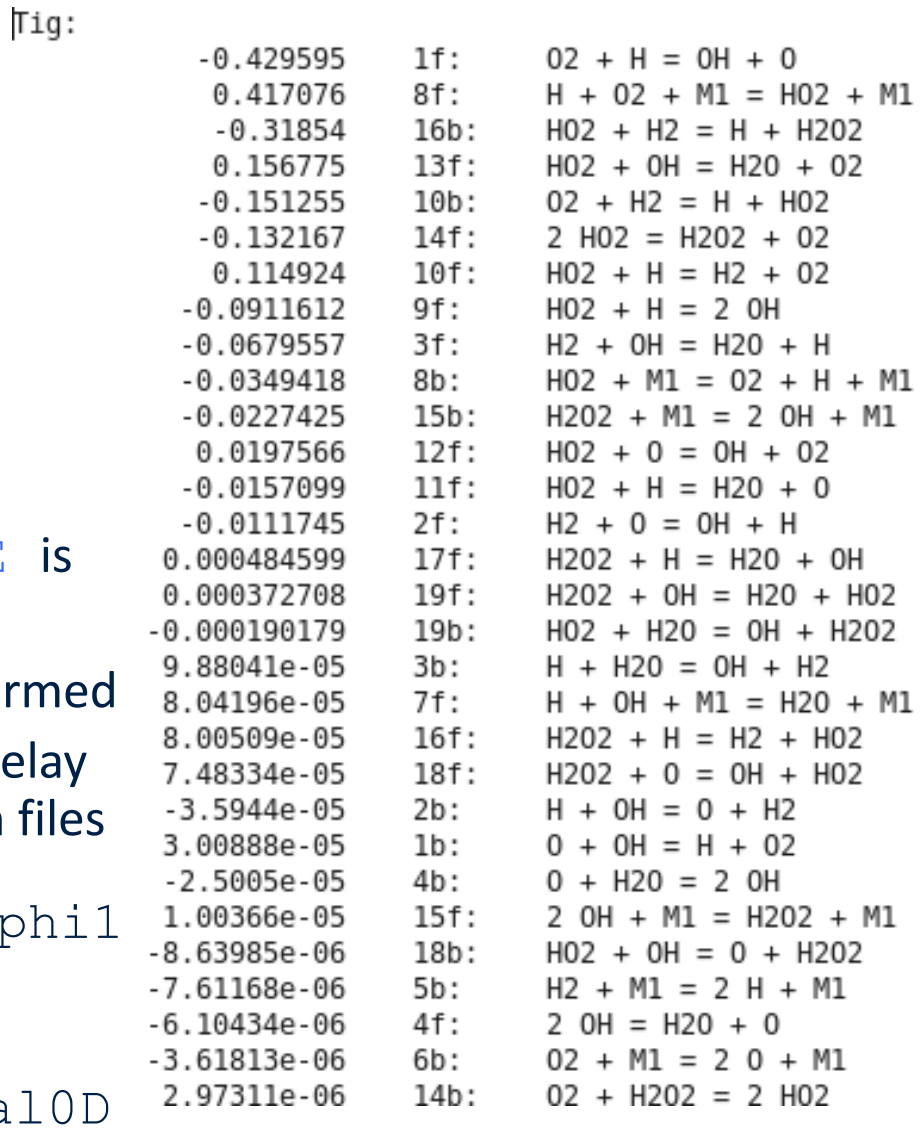
# FlameMaster Tutorial: Homogeneous Reactor

Constant volume homogeneous reactor performing sensitivity analysis for hydrogen/air mixtures

- `Example:`
  `cd Run/FlameMan/Prem/0D/H2`
  `FlameMan`

- If the flag `SensAnalReac is TRUE` is added to the input file, then
  - Reaction sensitivity analysis is performed
  - Sensitivity coefficients for ignition delay times or chosen species are given in files like `MAXSorted_Reac_SC_H2_p01phi100to1000.tout`

- More details are found in `Run/FlameManPrem/0D/SensAnal0D.pdf`



```
Tig:
    -0.429595    1f:    O2 + H = OH + O
     0.417076    8f:    H + O2 + M1 = HO2 + M1
     -0.31854   16b:    HO2 + H2 = H + H2O2
     0.156775   13f:    HO2 + OH = H2O + O2
    -0.151255   10b:    O2 + H2 = H + HO2
    -0.132167   14f:    2 HO2 = H2O2 + O2
     0.114924   10f:    HO2 + H = H2 + O2
   -0.0911612    9f:    HO2 + H = 2 OH
   -0.0679557    3f:    H2 + OH = H2O + H
   -0.0349418    8b:    HO2 + M1 = O2 + H + M1
   -0.0227425   15b:    H2O2 + M1 = 2 OH + M1
    0.0197566   12f:    HO2 + O = OH + O2
   -0.0157099   11f:    HO2 + H = H2O + O
   -0.0111745    2f:    H2 + O = OH + H
 0.000484599   17f:    H2O2 + H = H2O + OH
 0.000372708   19f:    H2O2 + OH = H2O + HO2
-0.000190179   19b:    HO2 + H2O = OH + H2O2
 9.88041e-05    3b:    H + H2O = OH + H2
 8.04196e-05    7f:    H + OH + M1 = H2O + M1
 8.00509e-05   16f:    H2O2 + H = H2 + HO2
 7.48334e-05   18f:    H2O2 + O = OH + HO2
-3.5944e-05    2b:    H + OH = O + H2
 3.00888e-05    1b:    O + OH = H + O2
-2.5005e-05    4b:    O + H2O = 2 OH
 1.00366e-05   15f:    2 OH + M1 = H2O2 + M1
-8.63985e-06   18b:    HO2 + OH = O + H2O2
-7.61168e-06    5b:    H2 + M1 = 2 H + M1
-6.10434e-06    4f:    2 OH = H2O + O
-3.61813e-06    6b:    O2 + M1 = 2 O + M1
 2.97311e-06   14b:    O2 + H2O2 = 2 HO2
```

# FlameMaster Tutorial: Homogeneous Reactor

Constant volume homogeneous reactor performing reaction flux analysis for hydrogen/air mixtures

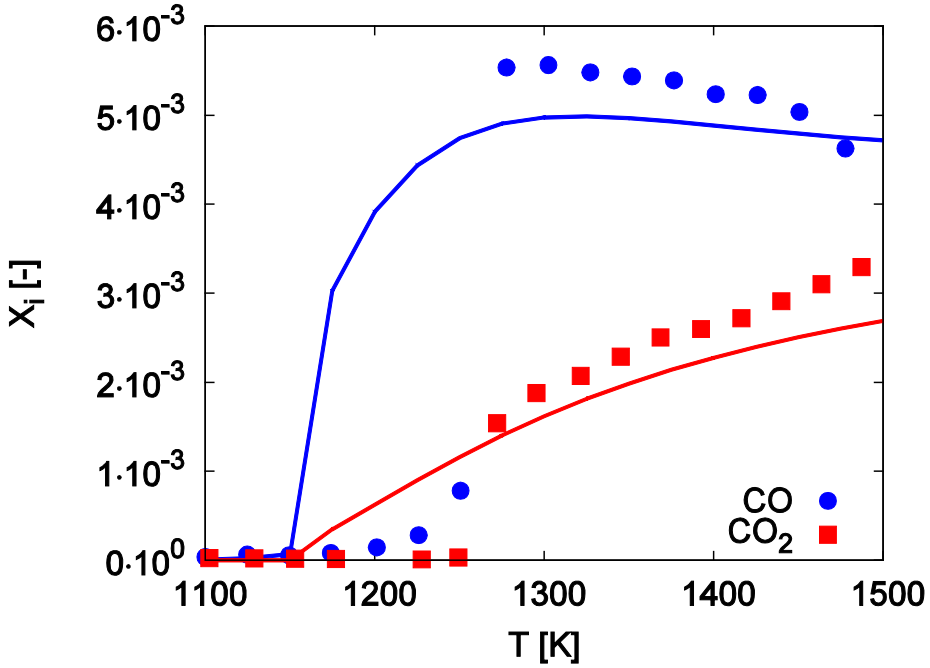
- Example:
  `cd Run/FlameMan/Prem/0D/H2`
  `FlameMan`

- If the flag `ReactionFluxAnal is TRUE` is added to the input file, then reaction pathway analysis is performed, and production and consumption channels of species are given in files like `Output/IntProRa_H2_p30phi100to1000.tout` (requires also `AdditionalOutput is TRUE`)

```
H:
          0.776747    3f:     H2 + OH = H2O + H
         -0.377077    8f:     H + O2 + M1 = HO2 + M1
         -0.271011    1f:     O2 + H = OH + O
          0.253203    2f:     H2 + O = OH + H
         -0.163056    9f:     HO2 + H = 2 OH
        -0.0965657    7f:     H + OH + M1 = H2O + M1
        -0.0352961    5f:     2 H + M1 = H2 + M1
        -0.0293398    10f:    HO2 + H = H2 + O2
        -0.0268207    11f:    HO2 + H = H2O + O
         0.00500216   16f:    H2O2 + H = H2 + HO2
       -0.000833782   17f:    H2O2 + H = H2O + OH
OH:
         -0.834354    3f:     H2 + OH = H2O + H
          0.350298    9f:     HO2 + H = 2 OH
           0.29111    1f:     O2 + H = OH + O
          0.271982    2f:     H2 + O = OH + H
         -0.103727    7f:     H + OH + M1 = H2O + M1
          0.103051    15f:    2 OH + M1 = H2O2 + M1
         -0.0611793   13f:    HO2 + OH = H2O + O2
          0.053952    4f:     2 OH = H2O + O
         0.00727006   12f:    HO2 + O = OH + O2
        0.000895619   17f:    H2O2 + H = H2O + OH
       -0.000739134   19f:    H2O2 + OH = H2O + HO2
        0.000400465   18f:    H2O2 + O = OH + HO2
O:
          0.947228    1f:     O2 + H = OH + O
         -0.884988    2f:     H2 + O = OH + H
          0.0937429   11f:    HO2 + H = H2O + O
         -0.0877758   4f:     2 OH = H2O + O
         -0.0236557   12f:    HO2 + O = OH + O2
        -0.00227764   6f:     2 O + M1 = O2 + M1
        -0.00130305   18f:    H2O2 + O = OH + HO2
```

# FlameMaster Tutorial: Perfectly Stirred Reactor

Perfectly stirred reactor (PSR)
computing stable species profiles
for methane/air mixtures



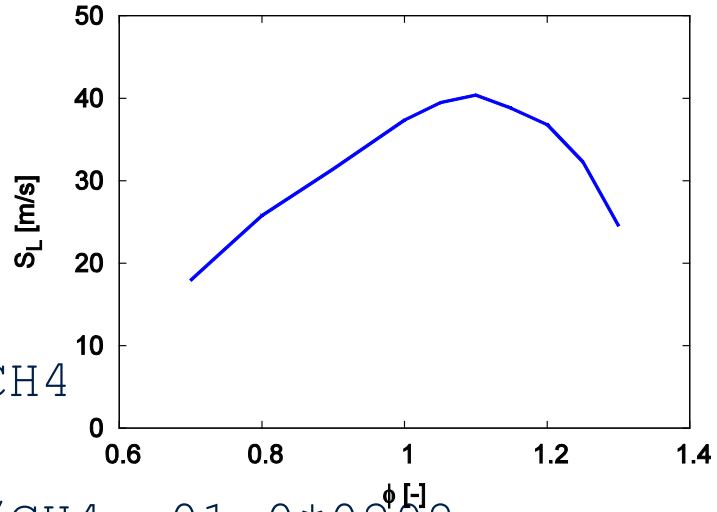

- `Example:`
  `cd Run/FlameMan/Prem/PSR`
  `FlameMan`

- This will create a file
  `Output/PSR_X_CH4_p01phi128tau00.14.dout`
  which has mole fractions of species
  for all computed initial temperatures as tab-separated list

- More details are found in `Run/Prem/PSR/README.md`

# FlameMaster Tutorial: Unstretched Premixed Flame

## Unstretched premixed flame computing laminar burning velocities for methane/air mixtures



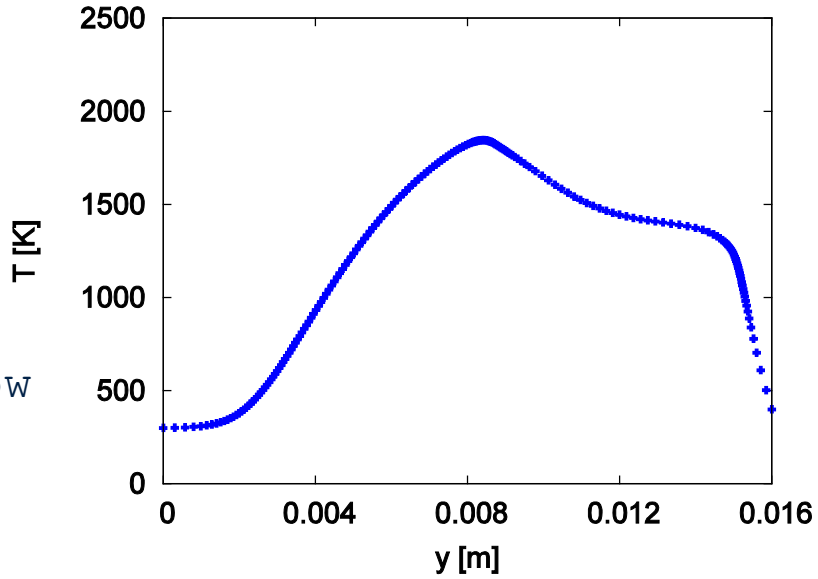

- `Example:`
  `cd Run/FlameMan/Prem/Unstretched/CH4`
  `FlameMan`
  `ListTool -s syms -r r.dout Output/CH4_p01_0*0298`

- `FlameMan` will create files like `Output/CH4_p01_0phi1_0000tu0298` containing the spatial distribution of all scalar fields at all computed conditions
  - With `PrintMoleFractions is TRUE`, also mole fractions are printed

- `ListTool` generates a file `r.dout`, which has the values of burning velocities or other scalar quantities specified in `syms` and maximum values of all vector quantities for all computed conditions as tab-separated list
  - Specified symbols are from the header of FlameMaster output files

# FlameMaster Tutorial: Counterflow Diffusion Flame

Counterflow diffusion flame with plug flow boundary condition on physical coordinate for acetylene/air mixtures



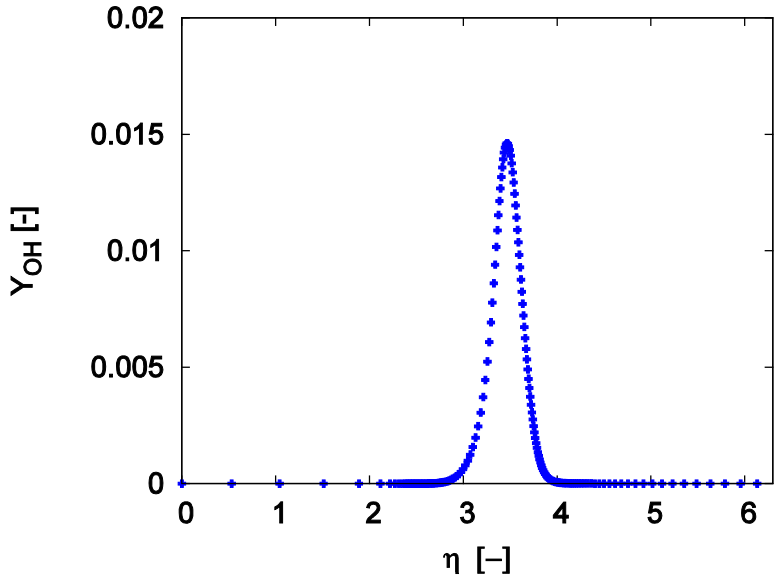

- `Example:`
  `cd Run/FlameMan/Diff/SteadyPlugFlow`
  `FlameMan`

- This will create a file
  `Output/C2H2_p01a00018tf0300to0400`
  containing the spatial distribution of all scalar fields at all computed conditions

- Different forms of continuation can be used
  - E.g., if the flag `ContType is velocity` is added to the input file, then critical conditions (e.g. strain rates) of extinction or autoignition can be determined.

# FlameMaster Tutorial: Counterflow Diffusion Flame

Counterflow diffusion flame with potential flow boundary condition on similarity coordinate for hydrogen/air mixtures



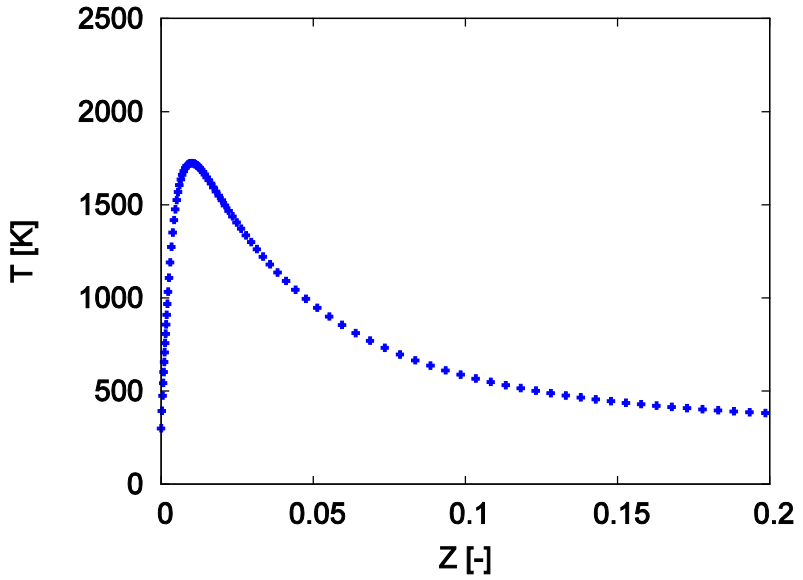

- Example:
  ```
  cd Run/FlameMan/Diff/Steady/H2
  FlameMan -i H2.count.in
  ```

- This will create a file
  `OutCount/H2_p01_0a00100_0tf0300to0300`
  containing the spatial distribution of all scalar fields at all computed conditions

- More details are found in
  `Repository/doc/FlameManMan.pdf`
  `Repository/doc/MThesisGer.pdf`

# FlameMaster Tutorial: Flamelet Equations

Diffusion flames solving
flamelet equations
using mixture fraction coordinate
for hydrogen/air mixtures

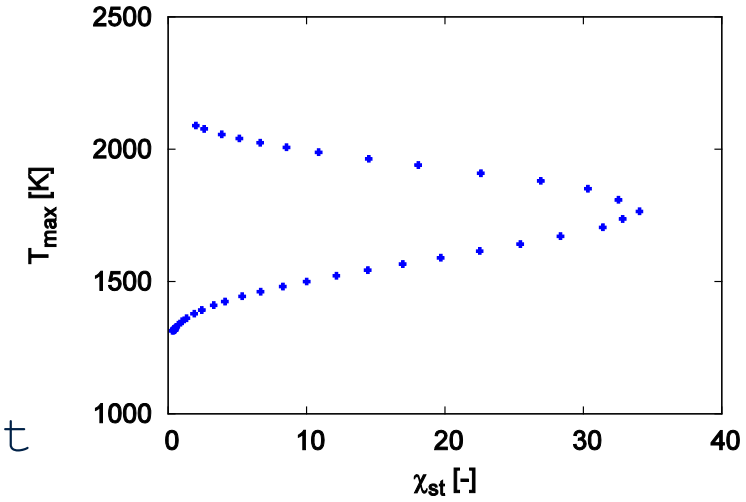



- Example:
  ```
  cd Run/FlameMan/Diff/Steady/H2
  FlameMan -i H2.mixfrac.in
  ```

- This will create a file
  `OutMixFrac/H2_p01_0chi1.734tf0300to0300`
  containing the spatial distribution of all scalar fields at all computed conditions

# FlameMaster Tutorial: Flamelet Equations

Solving flamelet equations
with continuation
for methane/air mixtures




- `Example:`
  `cd Run/FlameMan/Diff/Steady/CH4`
  `FlameMan -i FlameMasterCont.input`

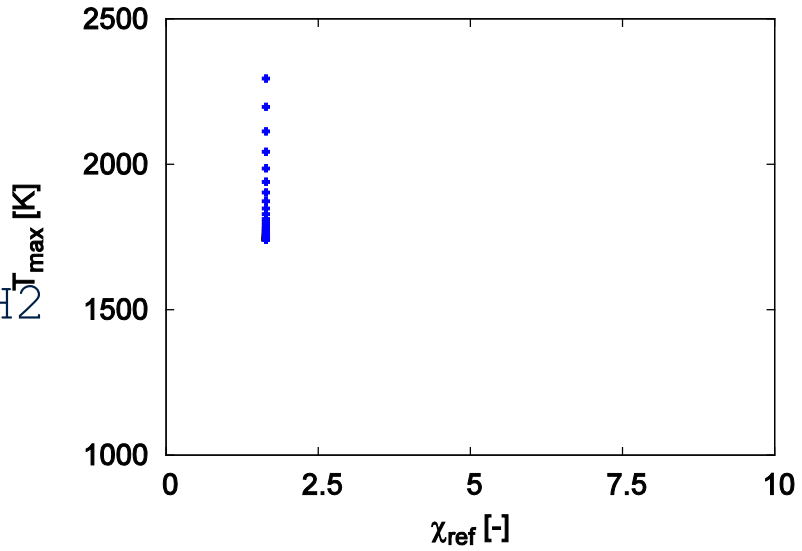
- If the flag `ArclengthCont is TRUE` is added to the input file, `FlameMaster` computes solutions along S-shaped curve

- `ListTool -s syms -r r.dout OutMethaneCont/CH4_p01_0chi*st*`
  creates a file `r.out` containing data for scalar quantities specified in `syms` and maximum values of all vector quantities for all computed flamelets

- More details are found in
  `Run/FlameMan/Diff/Steady/CH4/README.md`

# FlameMaster Tutorial: Transient flamelet

## Transient flamelet
## for hydrogen/air mixtures




- Example:
  ```
  cd Run/FlameMan/Diff/Unsteady/H2
  FlameMan
  ```

- This will create output files
  ```
  Lib_Chi01.64_T1742t00014.tout
  Lib_Chi01.64_T1742t00015.tout
  Lib_Chi01.64_T1742t00016.tout
  ```
  …

  which has the values of temperature, species mass and mole fractions, velocity, heat release, density … as function of mixture fraction at the computed condition

# FlameMaster Flame Calculator

- FlameMaster: A C++ Computer Program for 0D Combustion and 1D Laminar Flame Calculations
  - Premixed and non-premixed
  - Steady and unsteady
  - Emphasis on pre- and post-processing
    - Sensitivity analysis
    - Reaction flux analysis
- At request, available online at https://itv.rwth-aachen.de/index.php?id=13

# Summary

## Part I: Fundamentals and Laminar Flames

- Introduction
- Fundamentals and mass balances of combustion systems
- Thermodynamics, flame temperature, and equilibrium
- Governing equations
- Laminar premixed flames: Kinematics and Burning Velocity
- Laminar premixed flames: Flame structure
- Laminar diffusion flames
- FlameMaster flame calculator

- Introduction
- Counterflow diffusion flame
- Flamelet structure of diffusion flames
- FlameMaster flame calculator
- Single droplet combustion