A Guide to FlameMaster

H. Pitsch, R. Seiser, B. Varatharajan

Contents

Chapter 1

Introduction

1.1 The application FlameMaster

The application FlameMaster is a comprehensive tool for numerical modeling of combustion processes. Multiple configurations are supported as well as the use of complex chemical kinetic mechanisms. In this introduction, an overview is given about the different configurations, the included modeling options and the software programs.

1.1.1 Supported Configurations

- 0-dimensional
  - 1. Isochoric homogeneous reactor
    2. Isobaric homogeneous reactor
- 1-dimensional
  - Non-premixed combustion (counterflow diffusion flame)
    3. Plug-flow boundary conditions
    4. Potential-flow boundary conditions
    5. Mixture fraction formulation ("Flamelet")
  - Non-premixed unsteady combustion
    6. Mixture fraction formulation ("Flamelet")
  - Premixed combustion
    7. Unstretched configuration

1.1.2 Additional included, but unsupported configurations

- 0-dimensional
  - 1. Perfectly stirred reactor (PSR)
- 1-dimensional

  - Premixed combustion

    2. Unsteady formulation
    3. Counterflow configuration (potential-flow)

1.1.3 Modeling options

- Effects included in the formulation
  - Radiation
  - Differential diffusion effects
  - Thermal diffusion
  - Complex mixing model
  - Unsteady formulation
- Additional modeling tools
  - Sensitivity analysis

1.1.4 Programs

The application consists of two main programs, ScanMan and FlameMaster. ScanMan is a program for preprocessing a chemical kinetic mechanism, connecting it with necessary thermo data, and performing some checks of consistency. FlameMaster is the main program that formulates the conservation equations for the selected configuration, and solves them using appropriate solving algorithms. Additional programs (tools) are used to process a supplied thermo data file (program "CreateBinFile") and to process the ouput of FlameMaster (program "ListTool").
The so far described programs are written in C++ and supplied with their source codes. Some implemented libraries are written in FORTRAN77. For scanning and parsing input files, two programs "Flex" and "Bison" are used. These two programs were developed by the Free Software Foundation (FSF) as extensions of the UNIX programs "Lex" and "Yacc".

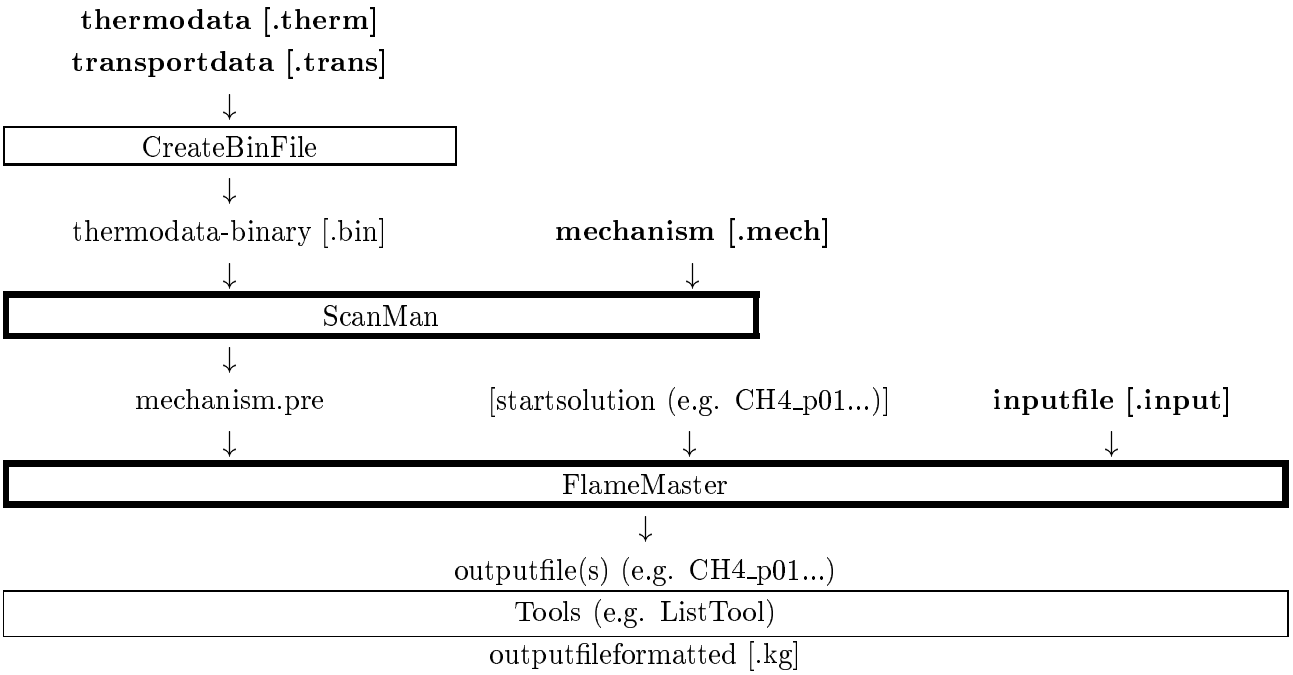
- Required system software
  - GNU C++ compiler (g++)
  - Fortran77 compiler (F77)
  - Flex (lexical analizer generater), alternatively a version of Flex is included in the installation package.
  - DDD Debugger (highly recommended)
- Systems and Operating Systems
  - Sun Workstation (Sun OS and Solaris)
  - HP Workstation
  - Intel PC (Linux)
  - Apple Macintosh (Linux)

Chapter 2

Running FlameMaster

2.1 Overview

The typical procedure of running FlameMaster is shown in the flow diagram below.



The steps are the following: First, given thermodynamic data and transport properties have to be converted into a binary file using CreateBinFile. Second, a chemical-kinetic mechanism has to be compiled using ScanMan. the third and main step is to run FlameMaster using an inputfile which contains all parameters necessary for the specific configuration. In the one-dimensional cases a startsolution is needed, which is a resultfile from a previous run. It can be necessary to repeat this step several times when changing from one given solution to a desired set of conditions in small steps. Basically, the created outputfiles contain all necessary information. For formatting or plotting the results, however, some additional tools (e.g. ListTool) are available.

Chapter 3

Programs

The programs CreateBinfile, ScanMan, and FlameMaster will now be described in more detail. The following items will be listed and discussed.

1. Purpose
   A short description of what the program is intended to do.

2. Input files
   Which input files are necessary, what they stand for.

3. Command Options
   The specific program call from the command line and additional options will be described.

4. Output
   Which outputfiles are created, what information they contain, which syntax they have.

5. Appendix
   If applicable, an extensive listing of commands and an example inputfile is given here.

3.1 CreateBinFile

3.1.1 Purpose

CreateBinFile converts ASCII files containing thermodynamic data and transport properties into a binary thermodata file. NASA and FlameMaster styles are supported.

3.1.2 Input files

NASA style

Two files are necessary. One, containing thermodynamic data in the NASA type format, the other, containing transport properties. Remark: Species names are limited to 18 characters.

FlameMaster style

One single file contains all necessary thermodynamic data and transport properties. Remark: Certain restrictions apply for species names.

3.1.3 Command and options

In the command line type:
CreateBinFile -i thermodatainputfile[.txt] -m transportdatainputfile -o thermodataoutputfile[.bin] [-h] [-p] [-d]

-i thermodatainputfile Ascii thermodynamic data file in NASA or FlameMaster style

-m transportdatainputfile Ascii transport property file in NASA style. If omitted then thermodatainputfile is to be assumed in FlameMaster style.

-o thermodataoutputfile Binary thermodynamic data and transport property file, the extension .bin is recommended.

-h help

-p

-d

3.1.4 Output

Standard Output

All error messages are sent to the standard output device (usually the computer screen).
Typical error messages

 # "CH3OH": molar mass unknown The molar mass is not given in the thermodata file. This has no further effect, because ScanMan will calculate the molar mass from the atom information.

Output Files

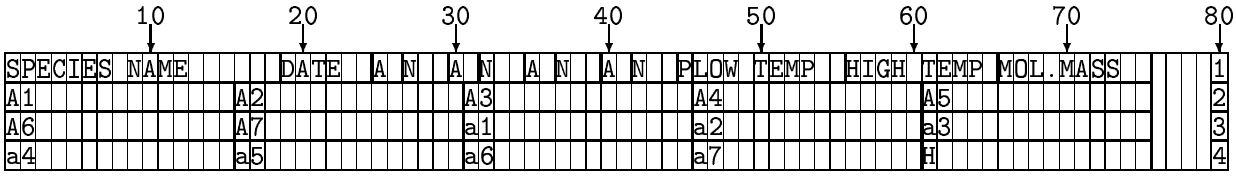
The only output file is the binary thermodata file that was specified with the -o option

3.1.5 Syntax Information

NASA style

Two files are necessary. One, containing thermodynamic data in the NASA type format, the other, containing transport properties.

The thermodynamic data file contains four lines per species with fixed positions for the entries



Line	Column	Contents	Format	Letter
1	1-18	Species name (must start in column 1)	18A1	
1	19-24	Origin and Date (not used in the code)	6A1	
1	25-44	Atomic symbols and number	4(2A1,I3)	A,N
1	45	Phase of species (S, L, or G for solid, liquid, or gas, respectively)	A1	P
1	46-55	Low temperature boundary	E10.0	
1	56-65	High temperature boundary	E10.0	
1	66-73	Molecular weight in kg/kmol (or empty) In the old NASA tables this used to be the common temperature	E8.0	
1	80	The Number "1"	I1	1
2	1-75	Coefficients $a_1 - a_5$ in eqs. 3.1-3.3, for the upper temperature interval	5(E15.0)	A1-A5
2	80	The Number "2"	I1	2
3	1-30	Coefficients $a_6 - a_7$ in eqs. 3.1-3.3, for the upper temperature interval	2(E15.0)	A6-A7
3	31-75	Coefficients $a_1 - a_3$ in eqs. 3.1-3.3, for the lower temperature interval	3(E15.0)	a1-a3
3	80	The Number "3"	I1	3
4	1-60	Coefficients $a_4 - a_7$ in eqs. 3.1-3.3, for the lower temperature interval	4(E15.0)	a4-a7
4	61-75	(H298-H_0)/R	E15.0	H
4	80	The Number "4"	I1	4

Example

```
        10        20        30        40        50        60        70        80
CH4                 L 8/88C  1.H  4.   0.    0.G200.000  6000.000  16.042761      1
 1.63552643E+00 1.00842795E-02-3.36916254E-06 5.34958667E-10-3.15518833E-14      2
-1.00056455E+04 9.99313326E+00 5.14987613E+00-1.36709788E-02 4.91800599E-05      3
-4.84743026E-08 1.66693956E-11-1.02466476E+04-4.64130376E+00-8.97226656E+03      4
```

The transport data file contains one lines per species with fixed positions for the entries

```
SPECIESNAME A EPS/K-- SIGMA- ?????  ?????  ?????  !  COMMENT
```

Example line

```
CH4 2 141.400 3.746 0.000 2.600 13.000 !  OIS
```

FlameMaster style

The thermodatainputfile consists of lines with the following syntax

```
SPECIESFORMULA A1 A2 A3 A4 A5 A6 A7 a1 a2 a3 a4 a5 a6 a7 MOL.MASS EPS/K SIGMA
```

Example line

```
CH4 .16834788E+01 .10237236E-01 -.38751286E-05 .67855849E-09 -.45034231E-13 -.1008
0787E+05 .96233950E+01 .77874148E+00 .17476684E-01 -.27834090E-04 .30497080E-07 -.
12239307E-10 -.98252285E+04 .13722195E+02 .16043E+02 .14140E+03 .37460E+01
```

The thermodynamic parameters

A1-A7, a1-a7 [-] Coefficients that are used in following thermodynamic formulas.

$$\frac{c_p}{R} = a_1 + a_2 T + a_3 T^2 + a_4 T^3 + a_5 T^4 \tag{3.1}$$

$$\frac{H^0}{RT} = a_1 + \frac{a_2}{2} T + \frac{a_3}{3} T^2 + \frac{a_4}{4} T^3 + \frac{a_5}{5} T^4 + \frac{a_6}{T} \tag{3.2}$$

$$\frac{S^0}{R} = a_1 \ln T + a_2 T + \frac{a_3}{2} T^2 + \frac{a_4}{3} T^3 + \frac{a_5}{4} T^4 + a_7 \tag{3.3}$$

$a_1 - a_7$ are the coefficients A1-A7 and a1-a7 for the high and low temperature interval, respectively.

MOL.MASS	M	[kg/kmol]	Molecular weight of species
EPS/K	ϵ/k	[K]	Characteristic Lennard-Jones energy divided by the Boltzmann constant
SIGMA	σ	[Å]	Hard-sphere diameter (Characteristic Lennard-Jones length)

3.2 ScanMan

3.2.1 Purpose

ScanMan processes the chemical mechanism file, converts it to a binary file and adds the thermodynamic data and transport properties.

3.2.2 Input files

`Mechanismfile[.mech]` contains the chemical mechanism including some definitions and third body coefficients. See syntax below. `thermodataoutputfile[.bin]`, which was generated from CreateBinfile will be linked together with mechanismfile.

3.2.3 Command and options

In the command line type:
ScanMan -i mechanism[.mech] -t thermodata[.bin] [-o output[.pre]] [-abcdhprsuLS3]

-i mechanism[.mech] Ascii input file that contains the chemical mechanism.

-t thermodata[.bin] Name of the binary thermodata file.

-o output[.pre] Optional name of the output file. If this arguments is not given, the name of the input file followed by '.pre' will be used. The ending '.mech' will be replaced.

-a A list of declared and actually used atoms will be generated.

-b A list of the empirical function F_c will be generated. F_c is used to calculate the additional factor in the Lindemann-Hinshelwood equation.

-c The check of forward versus backward reactions will not be performed

-d [scanner, parser] All scanner and/or parser actions will be shown.

-h Help. The possible options will be displayed.

-p The progress of the scanner wil be displayed.

-r A list of the reactions will be displayed

-s A list of species will be displayed.

-u Shows the allowed units and the units that are used.

-3 A list of Third-body coefficients will be displayed.

-L A LaTeX table [.tex] will be generated.

-S A lexicalical match is used to find species in the thermodata file.

3.2.4 Output

Standard Output

The additional output specified by the options above will be sent to the standard output device. If errors occur while parsing the mechanism file, error-messages will be sent to the computer-terminal. Typical error messages

Warning: set NASA polynomial coefficients of 'C7H16' to zero: The species C7H16 was not found in the thermodata file.

Result Files

The outputfile[.pre] is the primary output and can be processed further by FlameMaster. In addition, the following files are generated:

mechanism.h

mechanism.chmech The mechanism in CHEMKIN format.

mechanism.thermo The thermodata in CHEMKIN format.

mechanism.trans The transportdata in CHEMKIN format.

3.3 FlameMaster

3.3.1 Purpose

3.3.2 Input files

`Mechanismfile[.mech]` contains the chemical mechanism including some definitions and third body coefficients. See syntax below. `thermodataoutputfile[.bin]`, which was generated from CreateBinfile will be linked together with mechanismfile.

3.3.3 Command and options

In the command line type:
FlameMaster -i inputfile -s s -r r -o o -p p -d -h FlameMaster -r ... ScanMan -i mechanism[.mech] -t thermodata[.bin] [-o output[.pre]] [-abcdhprsuLS3]

-i inputfile[.input] Ascii input file that contains the initial/boundary conditions and various control sequences. If not specified the default file FlameMaster.input is used.

-s restartfile Name of the restart file. This is usually an output file from a previous run.

-r mechanism[.pre] If the name of the precompiled mechanism is given here it overrides the mechanism given in the input file.

-o outputpath The directory where output files will be written to.

-p pressure If the pressure is given here, it overrides the pressure given in the inputfile.

-d Debug the scanner.

3.3.4 Output

Standard Output

All error messages are sent to the standard output device (usually the computer screen).
Typical error messages

Output Files

3.3.5 Syntax Information

The following table summarizes all input file keywords. Their detailed explaination is given under the corresponding number on the following pages.

Table 3.1: Comand Overview

nr.	Command	0.D.		1.D.				
		Iso-chor	Iso-bar	Plug-flow	Pot.-flow	Flame-let	Unst. Flame-let	Pre-mixed
1	Author	2	2	2	2	2	2	2
2	MechanismFile	46	46	46	46	46	46	46
3	OutputPath	53	53	53	53	53	53	53
4	Pressure	58	58	58	58	58	58	58
5	FlameType	25	25	25	25	25	25	25
6	MaxIter	45	45	45	45	45	45	45
7	StartProfilesFile			73	73	73	73	73
8	Outputfile	52	52	52	52	52	52	52
9	StrainRate			?	75	75	75	
10	RightBC			66	66	66	66	66
11	LeftBC			40	40	40	40	40
12	Dirichlet			21	21	21	21	21
13	Gradient			30	30	30	30	30
14	X,Y	97	97	98	98	98	98	98
15	MassFlux			43	?	?	?	**43?**
16	f			**22?**	22	22	22	?
17	T	79	79	80	80	80	80	80
18	Fuel	**27?**	**27?**	27	27	27	27	27

Table 3.1: Comand Overview

nr.	Command	0.D.		1.D.				
		Iso-chor	Iso-bar	Plug-flow	Pot.-flow	Flame-let	Unst. Flame-let	Pre-mixed
19	ToSpecies	?	?	87	87	87	87	87
20	DampFlag	16	16	16	16	16	16	16
21	ContinFlag	?	?	10	10	10	10	10
22	R	63	63	63	63	63	63	63
23	Q	62	62	62	62	62	62	62
24	GridCorrectionstart							
25	SootRadiation							
26	SootUpdateProdRates							
27	SizeDependentDiffusion							
28	SurfaceDependentCoagulation							
29	Coagulation							
30	SurfaceGrowth							
31	SurfaceOxidation							
32	ThermoPhoresis							
33	PAHOHOxidation							
34	PAHO2Oxidation							
35	CoagulationFactor							
36	NSootMoments							
37	FlameIsAxiSymmetric			24	24	24	24	24
38	NoDiffusivityCorrection							
39	ClipNegativeConcentrations							
40	WriteBT							

Table 3.1: Comand Overview

nr.	Command	0.D.		1.D.				
		Iso-chor	Iso-bar	Plug-flow	Pot.-flow	Flame-let	Unst. Flame-let	Pre-mixed
41	WriteResiduum							
42	WatchGridding							
43	WriteEverySolution	94	94	94	94	94	94	94
44	WriteFullResiduals							
45	ComputeUnphysicalChain							
46	NOfUnphysicalChain							
47	UseModifiedNewton	89	89	89	89	89	89	89
48	UseNumericalJac	?	?	91	91	91	91	91
49	UseNumericalDM							
50	SensitivityObject							
51	ReactionFluxAnalysis							
52	PrintRHSSpecies							
53	PrintRHSTemperature							
54	LiquidPool			42				
55	Phi	57	57	57	57	57	57	57
56	ScalarDissipationRate							
57	MinStrainRate							
58	DeltaSContinuation							
59	InitialEquations			35	35	35	35	35
60	MaxGridPoints			**44!**	**44!**	**44!**	**44!**	**44!**
61	InitialGridPoints			**36!**	**36!**	**36!**	**36!**	**36!**
62	DampFlag	16	16	16	16	16	16	16

Table 3.1: Comand Overview

nr.	Command	0.D.		1.D.				
		Iso-chor	Iso-bar	Plug-flow	Pot.-flow	Flame-let	Unst. Flame-let	Pre-mixed
63	ContinFlag	10	10	10	10	10	10	10
64	TimeDepFlag			84	84	84	84	84
65	DeltaNewGrid			17	17	17	17	17
66	TolRes	86	86	86	86	86	86	86
67	TolDy	85	85	85	85	85	85	85
68	TStart	**88!**	**88!**	!	!	!	!	!
69	TEnd	**82!**	**82!**	!	!	!	!	!
70	Left	!	!	**39!**	**39!**	**39!**	**39!**	**39!**
71	Right	!	!	**65!**	**65!**	**65!**	**65!**	**65!**
72	Gamma							
73	Kappa							
74	Tau							
75	OneSolutionOneGrid			51	51	51	51	51
76	AdjustComputationalDomain							
77	PrintMolarFractions	59	59	59	59	59	59	59
78	SensitivityAnalysis							
79	SteadyStatesNumerical							
80	R			63	63	63	63	63
81	Q			62	62	62	62	62
82	GridCorrectionstart							
83	GridCorrectionEnd							
84	GridCorrectionAlpha							

Table 3.1: Comand Overview

nr.	Command	0.D.		1.D.				
		Iso-chor	Iso-bar	Plug-flow	Pot.-flow	Flame-let	Unst. Flame-let	Pre-mixed
85	LambdaMin	38	38	38	38	38	38	38
86	DeltaTMax							
87	DeltaTStart							
88	ContSteps	?	?	15	15	15	15	15
89	FromSpecies							
90	ContinuationBound							
91	ContinuationInc							
92	Oxidizer	**54?**	**54?**	54	54	54	54	54
93	CAInFile							
94	LewisNumberFile	41	41	41	41	41	41	41
95	ContinuationType							
96	ContinuationSide							
97	GlobalReaction	29	29	29	29	29	29	29
98	InitialCond	34	34					

Command Description

1. AdjustComputationalDomain
   Syntax:

   Configurations:
   Example:

2. Author
   Syntax: Author = string[50]
   This line is copied to the output file. It can hold the author's name or any comment.
   Configurations: All
   Example: Author = Frank SMITH - for heptane project

3. CAInFile
   Syntax:

   Configurations:
   Example:

4. ClipNegativeConcentrations
   Syntax:

   Configurations:
   Example:

5. Coagulation
   Syntax:

   Configurations:
   Example:

6. CoagulationFactor
   Syntax:

   Configurations:
   Example:

7. ComputeUnphysicalChain
   Syntax:

   Configurations:
   Example:

8. ComputeWithRadiation
   Syntax: ComputeWithRadiation = boolean
   If TRUE then the radiation terms are included in the energy equations.
   Configurations: 3,4,5,6,7
   Example: ComputeWithRadiation = TRUE

9. ConstantLewisNumber
   Syntax: ConstantLewisNumber = boolean
   Calculate with constant Lewis numbers. If no LewisNuberFile is specified, they are calculated from the startsolution at the temperature maximum. The diffusion coefficient of the species i will then be $D_i = \lambda/\rho c_p Le_i$.
   Configurations: All
   Example: ConstantLewisNumber = TRUE

10. ContinFlag
    Syntax: ContinFlag = boolean

With this option a continuation method with controlled steps is activated. A parameter $0 \leq \phi \leq 1$ is used within the program that terminates when ϕ equals to 1.
Configurations: 3,4,5,6,7
Example: ContinFlag = TRUE

11. ContinuationBound
Syntax:

Configurations:
Example:

12. ContinuationInc
Syntax:

Configurations:
Example:

13. ContinuationSide
Syntax:

Configurations:
Example:

14. ContinuationType
Syntax:

Configurations:
Example:

15. ContSteps
Syntax: ContSteps = int
If ContinFlag is TRUE then this argument gives the number of steps if constant steps are used
Configurations: 3,4,5,6,7
Example: ContSteps = xxx

16. DampFlag
Syntax: DampFlag = boolean
A damped Newton iteration will be used. Seexxx
Configurations: All
Example: DampFlag = TRUE

17. DeltaNewGrid
Syntax: DeltaNewGrid = int
This arguments sets the number of Newton iterations before a new grid is generated
Configurations: 3,4,5,6,7
Example: DeltaNewGrid = int

18. DeltaSContinuation
    Syntax:

    Configurations:
    Example:

19. DeltaTMax
    Syntax:

    Configurations:
    Example:

20. DeltaTStart
    Syntax:

    Configurations:
    Example:

21. Dirichlet
    Syntax: Dirichlet { ... }
    Defines the section for a Dirichlet boundary conditions. A Dirichlet boundary has absolut values for temperature and concentrations. The command is used within a LeftBC or RightBC Section.
    Configurations: 3,4,5,6,7
    Example: LeftBC { Dirichlet { T = 298.0 Y->CH4 = 0.2 } }

22. f
    Syntax: f = float
    Defines the stream function at a boundary condition. The command is used within a Dirichlet or Gradientxxx Section.
    Configurations: 3,4,5,6
    Example: LeftBC { Dirichlet { f = -6.0 T = 298.0 Y->CH4 = 1.0 } }

23. f'
    Syntax: f' = float
    xxx
    Configurations: 4,5,6
    Example: xxx

24. FlameIsAxiSymmetric
    Syntax: FlameIsAxiSymmetric = boolean
    Axisymmetric represents a counterflow burner with round ducts, otherwise the flow field will be assumed two-dimensional (slot burner)
    Configurations: 3,4,5,6,7
    Example: FlameIsAxiSymmetric = TRUE

25. FlameType
    Syntax: FlameType = string

Type of Configuration. The argument can take the following values
1. Isochor Homogeneous Reactor
2. Isobar Homogeneous Reactor
3. EigenValueDiffusion with Physical Coordinate
4. CounterFlowDiffusion
5. CounterFlowDiffusion in Mixture Fraction Space
6. xxxunsteady flamelet
7. UnstrechedPremixed
8. FlameSheet
9. Transient Flamelet

Configurations: All
Example: FlameType = CounterFlowDiffusion

26. FromSpecies
Syntax:

Configurations:
Example:

27. Fuel
Syntax: Fuel = string
This is only used for the Burke-Schumann solution. String must match the fuel as defined in the mechanism.
Configurations: All
Example: Fuel = C3H8

28. Gamma
Syntax:

Configurations:
Example:

29. GlobalReaction
Syntax: GlobalReaction = string;
This defines the global reaction of the chemical mechanism. The string must be terminated with a semicolon. Blanks, tabs and newlines are not significant.
Configurations: All
Example: GlobalReaction = C3H8 + 5O2 == 3CO2 + 4H2O;

30. Gradient
Syntax: Gradient { ... }
Defines the section for a Gradient boundary conditions. A Gradient boundary sets a gradient for temperature and concentrations. The command is used within a LeftBC or RightBC Section. Units xxx
Configurations: 3,4,5,6,7
Example: LeftBC { Gradient { xxx } }

31. GridCorrectionAlpha
    Syntax:

    Configurations:
    Example:

32. GridCorrectionEnd
    Syntax:

    Configurations:
    Example:

33. GridCorrectionstart
    Syntax:

    Configurations:
    Example:

34. InitialCond
    Syntax: InitialCond { ... }
    Defines the section to specify the initial conditions
    Configurations: 1,2
    Example: InitialCond { T = 1250 Y->CH4 = 1.0 }

35. InitialEquations
    Syntax: InitialEquations = int
    This option can be used to get to a start profile. In the beginning only int equations will be solved, the other values will be held constant. The temperature is calculated from the Burke-Schumann solution. If converged, then the other equations are added one by one until the complete system is solved. The default value is the number of all equations.
    Configurations: 3,4,5,6,7
    Example: InitialEquations = 4

36. InitialGridPoints
    Syntax: InitialGridPoints = int
    Determines whether or not an aequidistant initial grid is generated. If InitialGridPoints < grid points in the startsolution <= MaxGridPoints then the grid of the startsolution is used. This allows to go to a larger number of grid points while starting with the grid of the startsolution.
    Configurations: 3,4,5,6,7
    Example: InitialGridPoints = 30

37. Kappa
    Syntax:

    Configurations:
    Example:

38. LambdaMin
    Syntax: LambdaMin = float
    If DampFlag is TRUE then this argument sets the smallest allowed damping factor
    Configurations: All
    Example: LambdaMin = 0.02

39. Left
    Syntax: Left = int
    Position of left boundary in the coordinates of the startsolution. This line is only effective if an (aequidistant) grid is generated at the beginning. See InitialGridPoints. Units: [m]
    Configurations: 3,4,5,6,7
    Example: Left = 0.0

40. LeftBC
    Syntax: LeftBC { ... }
    Defines the section for the boundary conditions on the fuel side.
    Configurations: 3,4,5,6,7
    Example: LeftBC { Dirichlet { T = 298.0 Y->CH4 = 0.2 } }

41. LewisNumberFile
    Syntax: LewisNumberFile = string
    This specifies the file where Lewis numbers are read from, when ConstantLewisNumber is TRUE. If the file does not exist, (e.g. LewisNumberOne) the Lewis numbers of all species are set equal to 1. xxxsyntax
    Configurations: All
    Example: LewisNumberFile = LewisNumberOne #does not exist

42. LiquidPool
    Syntax: LiquidPool = boolean
    The fuel side will be modeled as liquid pool boundary. See xxx
    Configurations: 3
    Example: LiquidPool = TRUE

43. MassFlux
    Syntax: MassFlux->string = float
    Defines the mass flux fraction of species string at a boundary condition. The command is used within a Dirichlet or Gradientxxx Section.
    Configurations: 3,7
    Example: LeftBC { Dirichlet { T = 298.0 MassFlux->CH4 = 0.2 } }

44. MaxGridPoints
    Syntax: MaxGridPoints = int (odd number)
    Maximum number of grid points when generating a new grid. If the number of grid points in the startsolution is larger, an aequidistant grid will be generated right at the beginning.
    Configurations: 3,4,5,6,7
    Example: MaxGridPoints = 151

45. MaxIter
    Syntax: MaxIter = int
    Maximum number of Newton iterations.
    Configurations: All
    Example: MaxIter = 30

46. MechanismFile
    Syntax: Mechanismfile = string[50]
    File that contains the chemical kinetic mechanism that was compiled with ScanMan.
    Configurations: All
    Example: MechanismFile = ~/FlameMaster/Mech/nHeptane-short.pre

47. MinStrainRate
    Syntax:

    Configurations:
    Example:

48. NoDiffusivityCorrection
    Syntax:

    Configurations:
    Example:

49. NOfUnphysicalChain
    Syntax:

    Configurations:
    Example:

50. NSootMoments
    Syntax:

    Configurations:
    Example:

51. OneSolutionOneGrid
    Syntax: OneSolutionOneGrid = boolean
    If this option is TRUE an adaptive grid is generated at least once even if the solution would converge before DeltaNewGrid
    Configurations: 3,4,5,6,7
    Example: OneSolutionOneGrid = TRUE

52. Outputfile
    Syntax:

Configurations: All
Example:

53. OutputPath
Syntax: OutputPath = string
Gives an alternative output path where the results are written to.
Configurations: All
Example: OutputPath = ./Propane

54. Oxidizer
Syntax: Oxidizer = string
See Fuel.
Configurations: All
Example: Oxidizer = O2

55. PAHO2Oxidation
Syntax:

Configurations:
Example:

56. PAHOHOxidation
Syntax:

Configurations:
Example:

57. Phi
Syntax:

Configurations: All
Example:

58. Pressure
Syntax: Pressure = float
System pressure in [Pa]. This line can appear multiple times, then several outputs are created. If omitted then equals the pressure in the startsolution.
Configurations: All
Example: Pressure = 1.013e5

59. PrintMolarFractions
Syntax: PrintMolarFractions = boolean
The results are printed also in terms of molar fractions
Configurations: All
Example: PrintMolarFractions = TRUE

60. PrintRHSSpecies
Syntax:

Configurations:
Example:

61. PrintRHSTemperature
Syntax:

Configurations:
Example:

62. Q
Syntax: Q = float
If Q is positive it describes the maximum allowed error, on basis of that the number of grid points are selected. If Q is negative, the number of gridpoints is kept unchanged when generating an adaptive grid.
Configurations: All
Example: Q = -0.25

63. R
Syntax: R = float
This is a grid parameter that describes the ratio $\Delta\eta_{max}/\Delta\eta_{min}$.
Configurations: All
Example: R = 60

64. ReactionFluxAnalysis
Syntax:

Configurations:
Example:

65. Right
Syntax: Right = int
Position of right boundary. See Left! Units: [m]
Configurations: 3,4,5,6,7
Example: Right = 0.01

66. RightBC
Syntax: RightBC { ... }
Defines the section for the boundary conditions on the oxidizer side.
Configurations: 3,4,5,6,7
Example: RightBC { Dirichlet { T = 298.0 Y->O2 = 0.233 } }

67. ScalarDissipationRate
Syntax:

Configurations:
Example:

68. SensitivityAnalysis
Syntax:

Configurations:
Example:

69. SensitivityObject
Syntax:

Configurations:
Example:

70. SizeDependentDiffusion
Syntax:

Configurations:
Example:

71. SootRadiation
Syntax:

Configurations:
Example:

72. SootUpdateProdRates
Syntax:

Configurations:
Example:

73. StartProfilesFile
Syntax: StartProfilesFile = string[50]
File that contains the chemical kinetic mechanism that was compiled with ScanMan.
Configurations: 3,4,5,6,7
Example: StartProfilesFile = ~/FlameMaster/heptane/N-C7H16_p01a00100tf0343to0298

74. SteadyStatesNumerical
Syntax:

Configurations:
Example:

75. StrainRate
Syntax: StrainRate = float
Defined as the horizontal gradient of the vertical velocity at the stagnation plane. Units: [1/s]. This line can appear multiple times, then several outputs are created. The strain rates are chosen in the order they appear. If the Iteration does not converge the program stops. If omitted then equals the strain rate in the startsolution.
Configurations: 4,5,6
Example: StrainRate = 100

76. SurfaceDependentCoagulation
Syntax:

Configurations:
Example:

77. SurfaceGrowth
Syntax:

Configurations:
Example:

78. SurfaceOxidation
Syntax:

Configurations:
Example:

79. T (1)
Syntax: T = float
Specifies the initial temperature. Units: [K], if T¡10 then in [1000/K]. This line can be used multiple times to calculate different cases and also together with Continuation xxx
Configurations: 1,2
Example: InitialCond { T = 0.8 Y->CH4 = 1.0 } # = 1250 K

80. T (2)
Syntax: T = float
Specifies the temperature at a boundary. Units: [K]. The command is used within a Dirichlet or Gradientxxx Section.
Configurations: 3,4,5,6,7
Example: LeftBC { Dirichlet { T = 298.0 Y->CH4 = 1.0 } }

81. Tau
Syntax:

Configurations:
Example:

82. TEnd
Syntax:

Configurations: 1,2
Example:

83. ThermoPhoresis
Syntax:

Configurations:
Example:

84. TimeDepFlag
Syntax: TimeDepFlag = boolean
The calculation will be instationary. Starting with the startsolution the profiles will approach the steady-state solution of the new boundary conditions.
Configurations: 3,4,5,6,7
Example: TimeDepFlag = TRUE

85. TolDy
Syntax: TolDy= float
Sets the tolerance for the norm of the residuum. If a smaller value is obtained during the calculation it will converge.
Configurations: All
Example: TolDy = 1.0e-5

86. TolRes
Syntax: TolRes = float
Sets the tolerance for the norm of the increment during the Newton iteration. If a smaller value is obtained during the calculation it will converge.
Configurations: All
Example: TolRes = 1.0e-15

87. ToSpecies
Syntax: ToSpecies = string
This option can be used to change to a new species. In a continuation method 5 w% are added after each converged solution. If the calculation does not converge, the program will stop. string must match the species in the mechanism file.
Configurations: 3,4,5,6,7
Example: ToSpecies = C2H4

88. TStart
Syntax: TStart = float
xxx
Configurations: 1,2
Example: TStart = 100

89. UseModifiedNewton
    Syntax: UseModifiedNewton = boolean
    If TRUE then the Newton solver allows modified steps (in general faster).
    Configurations: All
    Example: UseModifiedNewton = TRUE

90. UseNumericalDM
    Syntax:

    Configurations:
    Example:

91. UseNumericalJac
    Syntax: UseNumericalJac = boolean
    Allows the Jacobian matrix to be numerically evaluated. This is recommended.
    Configurations: 3,4,5,6,7
    Example: UseNumericalJac = TRUE

92. WatchGridding
    Syntax:

    Configurations:
    Example:

93. WriteBT
    Syntax:

    Configurations:
    Example:

94. WriteEverySolution
    Syntax:

    Configurations: All
    Example:

95. WriteFullResiduals
    Syntax:

    Configurations:
    Example:

96. WriteResiduum
    Syntax:

    Configurations:
    Example:

97. X,Y (1)
    Syntax: X->string = float, Y->string = float
    X defines the initial mole fraction, Y defines the initial mass fraction of species string. The command is used within the InitialCond Section.
    Configurations: 1,2
    Example: InitialCond { T = 1250 X->CH4 = 0.2 }

98. X,Y (2)
    Syntax: X->string = float, Y->string = float
    X defines the mole fraction, Y defines the mass fraction of species string at a boundary. The command is used within a Dirichlet or Gradientxxx Section.
    Configurations: 3,4,5,6,7
    Example: LeftBC { Dirichlet { T = 298.0 Y->CH4 = 1.0 } }

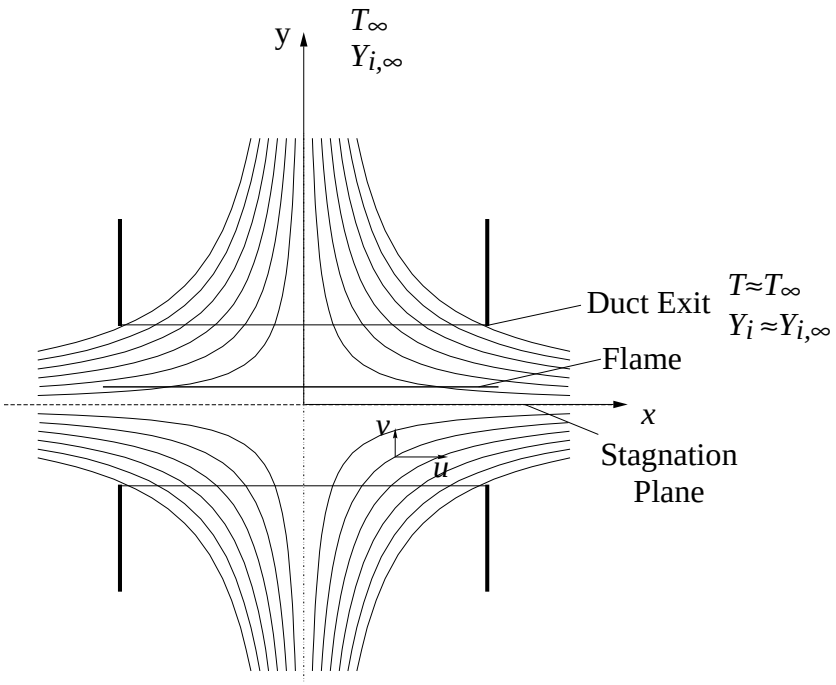
Chapter 4

Modeling

4.1 COUNTERFLOW DIFFUSION (POTENTIAL-FLOW)

The flow field consist of two opposing streams and is inviscid and irrotationaloutside the boundary layer around the stagnation plane. The boundaries for fuel and oxidizer are $y_F \to \infty$ und $y_{Ox} \to -\infty$. The position of the flame is close to the stagnation plane, for hydrocarbon fuels typically on the oxidizer side.

Figure 4.1: Potential Flow Configuration, shown together with a possible burner position.



The flow field within the boundary layer is described by the Navier-Stokes equations. With a similarity transformation a one-dimensional formulation can be obtained. This will reduce the problem to solve a system of ordinary differential equations. This procedure is summarized in the following pages. Here, the equations are derived for the planar case, which physically reflects two streams coming out of slots with infinite depth in z-direction. The other typical case, the axisymmetric case (round ducts),

also used by FlameMaster, is formulated with conservation equations in polar coordinates z,r.

Conservation equations

Continuity:

$$\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \tag{4.1}$$

Momentum equation in x-direction:

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \frac{\partial}{\partial x}\left[\mu\left(2\frac{\partial u}{\partial x} - \frac{2}{3}\nabla \cdot \vec{v}\right)\right] - \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] = 0 \tag{4.2}$$

Momentum equation in y-direction:

$$\rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] - \frac{\partial}{\partial y}\left[\mu\left(2\frac{\partial v}{\partial y} - \frac{2}{3}\nabla \cdot \vec{v}\right)\right] = 0 \tag{4.3}$$

Here, x und y are the independent coordinates with u und v as the velocities. ρ denotes the density, p the pressure, and μ the dynamic viscosity.

The divergence of the velocity can be expressed

$$\nabla \cdot \vec{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\,. \tag{4.4}$$

Buoyancy is neglected in the momentum equations.

For the similarity transformation the coordinates are given with

$$\eta = \left[\frac{a}{(\rho\mu)_{\mathrm{ref}}}\right]^{1/2} \int_0^y \rho\, dy \tag{4.5}$$

and

$$\xi = x \tag{4.6}$$

Here a, the strain rate, is defined as the horizontal velocity gradient (of the horizontal velocity) in the oxidizer stream (at $y_F \to \infty$):

$$a = \frac{du_\infty}{dx}\,. \tag{4.7}$$

The reference values ρ_{ref} und μ_{ref} also refer to conditions in the oxidizer stream. The transformation of the coordinates is done by

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi} \tag{4.8}$$

and

$$\frac{\partial}{\partial y} = \left[\frac{a}{(\rho\mu)_{\mathrm{ref}}}\right]^{1/2} \rho \frac{\partial}{\partial \eta}\,. \tag{4.9}$$

With introduction of the stream function f, defined by

$$\rho u = \sqrt{(\rho\mu)_{\text{ref}}\, a}\, x \frac{\partial f}{\partial y} \tag{4.10}$$

and

$$\rho v = -\sqrt{(\rho\mu)_{\text{ref}}\, a}\, \frac{\partial (x f)}{\partial x} \tag{4.11}$$

the continuity equation is satisfied.

The resulting velocities are

$$u = a\xi \frac{\partial f}{\partial \eta} \tag{4.12}$$

and

$$\rho v = -\sqrt{(\rho\mu)_{\text{ref}}\, a}\, f(\eta)\,. \tag{4.13}$$

The pressure p is calculated by

$$p = p_0 - \frac{1}{2}\left(a^2 \rho_\infty \xi^2 + a\mu_\infty F(\eta)\right)\,. \tag{4.14}$$

Here, p_0 is the stagnation pressure. The second term describes the decreas in pressure due to acceleration in ξ-direction and the third term the change of pressure due to expansion. The transformed momentum equations are: x-direction:

$$f\frac{\mathrm{d}^2 f}{\mathrm{d}\eta^2} + \frac{\rho_\infty}{\rho} - \left(\frac{\mathrm{d}f}{\mathrm{d}\eta}\right)^2 + \frac{\mathrm{d}}{\mathrm{d}\eta}\left(C\frac{\mathrm{d}^2 f}{\mathrm{d}\eta^2}\right) = 0 \tag{4.15}$$

y-direction:

$$f\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\frac{f}{\rho}\right) + \frac{1}{2\rho_\infty}\frac{\mathrm{d}F}{\mathrm{d}\eta} + \frac{C}{\rho}\frac{\mathrm{d}^2 f}{\mathrm{d}\eta^2} - 2\frac{\mathrm{d}}{\mathrm{d}\eta}\left(C\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\frac{f}{\rho}\right)\right) f\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\frac{f}{\rho}\right) - \frac{2}{3}\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\frac{C}{\rho^2} f \frac{\mathrm{d}\rho}{\mathrm{d}\eta}\right) = 0\,. \tag{4.16}$$

Here, the Chapman-Rubesin parameter

$$C = \frac{\rho\mu}{(\rho\mu)_{\text{ref}}} \tag{4.17}$$

was used

The momentum equations in x and y-direction can be solved independently. The equation in x-direction will be used when the pressure is of interest.

To calculate the Chapman-Rubesin parameter in eq. (4.15) the species and energy conservation equations are formulated using the similarity transformation.

Species conservation:

$$f\frac{\mathrm{d}Y_i}{\mathrm{d}\eta} - \frac{1}{\sqrt{(\rho\mu)_{\mathrm{ref}}}a}\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\rho Y_i V_{iy}\right) + \frac{\dot{m}_i}{\rho a} = 0\,, \qquad i = 1, 2, ..., n \tag{4.18}$$

Energy conservation:

$$f\frac{\mathrm{d}T}{\mathrm{d}\eta} + \frac{1}{c_p(\rho\mu)_{\mathrm{ref}}}\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\rho\lambda\frac{\mathrm{d}T}{\mathrm{d}\eta}\right) - \frac{1}{\sqrt{(\rho\mu)_{\mathrm{ref}}}a}\frac{\mathrm{d}T}{\mathrm{d}\eta}\sum_{i=1}^{n}\frac{c_{pi}}{c_p}\rho Y_i V_{iy}$$

$$- \frac{1}{c_p\rho a}\sum_{i=1}^{n}\dot{m}_i h_i - \frac{1}{c_p\rho a}\sum_{i=1}^{n} q_{Ri} = 0\,. \tag{4.19}$$

Here, Y_i denotes the mass fraction of species i, n the total number of species, T the temperature, V_{iy} the diffusion velocity of species i in y-direction, $\dot{m}_i$ the volume based production rate of species i, c_{pi} the specific heat of species i-ten at constant pressure, h_i the specific enthalpy of species i, q_{Ri} the power of radiation of species i, λ the thermal conductivity of the mixture, M the average molecular weight of the mixture, andd c_p the specific heat of the mixture at constant pressure.

To complete the system the equation of state for the ideal gas is emplyed.

$$\rho = \frac{p\,M}{\mathcal{R}\,T} \tag{4.20}$$

where $\mathcal{R}$ ist the universal gas constant.

For the radiation source term in eq. (4.19) a model for optical thin gases is used. Here, all terms other than CO_2 and H_2O are neglected. The source term then can be described as

$$\sum_{i=1}^{n} q_{Ri} = -2\sigma_S\rho\mathcal{R}T^5\left[\alpha_{P,\mathrm{CO_2}}\left(\frac{Y}{M}\right)_{\mathrm{CO_2}} + \alpha_{P,\mathrm{H_2O}}\left(\frac{Y}{M}\right)_{\mathrm{H_2O}}\right]. \tag{4.21}$$

Here, σ_S is the Stefan-Boltzmann constant, $\alpha_{P,\mathrm{CO_2}}$ and $\alpha_{P,\mathrm{H_2O}}$ are the radiation coefficients. For the latter the following empirical equations are used

$$\alpha_{P,\mathrm{CO_2}} = 46.241\ \exp\left(-8.888\,10^{-4}\,T\right) \tag{4.22}$$

and

$$\alpha_{P,\mathrm{H_2O}} = 22.6\ \exp\left(-1.546\,10^{-3}\,T\right) \tag{4.23}$$

.

The boundary conditions for the present system are:

$\eta \to -\infty$:

$$\frac{\mathrm{d}f}{\mathrm{d}\eta} = \sqrt{\frac{\rho_\infty}{\rho_{-\infty}}} \tag{4.24}$$

$$T = T_{-\infty} \tag{4.25}$$

$$Y_i = Y_{i,-\infty} \tag{4.26}$$

$\eta = 0$:

$$f = 0 \tag{4.27}$$

$\eta \to \infty$:

$$\frac{\mathrm{d}f}{\mathrm{d}\eta} = 1 \tag{4.28}$$

$$T = T_\infty \tag{4.29}$$

$$Y_i = Y_{i,\infty} \tag{4.30}$$

4.2 Molecular and Thermodynamic Transport Coefficients

Nach The diffusion velocity $\vec{V_i}$ consists of three parts:

$$\vec{V_i} = \vec{V_i}^D + \vec{V_i}^T + \vec{V_c}\,, \qquad i = 1, 2, ..., n \tag{4.31}$$

Here, $\vec{V_i}^D$ is the pressure diffusion, $\vec{V_i}^T$ the thermal diffusion and $\vec{V_c}$ a correction velocity, which is the same for all species. The pressure diffusion is described by the Curtiss-Hirschfelder approximation

$$\vec{V_i}^D = -\frac{D_i}{X_i}\nabla X_i\,, \qquad i = 1, 2, ..., n\,. \tag{4.32}$$

with X_i representing the mole fraction of spezies i. In the present one-dimensional case ∇X_i can be written as dX_i/dy. D_i is the average diffusion coefficient of species i in the mixture and is given wih:

$$D_i = \frac{1 - \sum\limits_{j=1}^{n} Y_i}{\sum\limits_{\substack{j=1 \\ j\neq i}}^{n} \frac{X_j}{\mathcal{D}_{ij}}}\,, \qquad i = 1, 2, ..., n \tag{4.33}$$

The binary diffusion coefficients $\mathcal{D}_{ij}$ are

$$\mathcal{D}_{ij} = 0.0753206 \frac{\sqrt{T^3 \left(\frac{1}{M_i} + \frac{1}{M_j}\right)}}{p\left(\sigma_i + \sigma_j\right)^2 \Omega_{ij}^{(1,1)}} \tag{4.34}$$

σ is the hard-sphere diameter of each species in [Å] and $\Omega_{ij}^{(1,1)}$ the collision integral. The molecular weight is used in [kg/kmol] everything else in SI-units.
The thermal diffusion is neglected.

Because the Curtiss-Hirschfelder approximation is only exact for binary mixtures, but not for more component mixtures, the correction velocity is chosen that mass conservation

$$\sum_{i=1}^{n} \vec{V}_i Y_i = 0 \tag{4.35}$$

is satisfied. With eq. (4.31) it results in:

$$\vec{V}_c = -\sum_{i=1}^{n} \vec{V}_i^D Y_i \,. \tag{4.36}$$

Dynamic viscosity and thermal conductivity of the mixture are calculated with empiric mixture rules on basis of the properties of the components

$$\mu = \sum_{i=1}^{n} \frac{Y_i \mu_i}{\Delta_i} \,, \tag{4.37}$$

$$\lambda = \sum_{i=1}^{n} \frac{Y_i \lambda_i}{\Delta_i} \,, \tag{4.38}$$

$$\Delta_i = \sum_{j=1}^{n} G_{ij} \frac{M_i}{M_j} Y_j \,, \tag{4.39}$$

$$G_{ij} = \frac{1}{\sqrt{8}} \left(1 + \frac{M_i}{M_j}\right)^{-1/2} \left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{-1/2} \left(\frac{M_j}{M_i}\right)^{1/4}\right]^2 \,. \tag{4.40}$$

Dynamic viscosity and thermal conductivity of the single components are, using the elemantary gas model:

$$\mu_i = 2.6693 \cdot 10^{-5} \frac{\sqrt{M_i T}}{\sigma_i^2 \Omega_{ij}^{(2,2)}} \tag{4.41}$$

$$\lambda_i = \mu_i \left(cp_i + \frac{5}{4} \frac{\mathcal{R}}{M_i}\right) \,. \tag{4.42}$$

Heat capacities and enthalpy of all species are calculated using the NASA polynomials

4.3 Chemical Kinetics

The source terms in equations (4.18) and (4.19) are represented by the chemical production rate $\dot{m}_i$, which is formed by the rates of the participating reactions:

$$\dot{m}_i = M_i \sum_{k=1}^{r} \nu_{ik} w_k \ , \tag{4.43}$$

Here r is the number of reactions, ν_{ik} the stoichiometric coefficient of species i in reaction k and w_k its reaction rate. For each reaction the reaction rate is computed from the rate coefficients k_k and the concentrations of the appearing species:

$$w_k = k_{fk}(T) \prod_{j=1}^{n} \left(\frac{\rho Y_j}{M_j} \right)^{\nu'_{kj}} - k_{bk}(T) \prod_{j=1}^{n} \left(\frac{\rho Y_j}{M_j} \right)^{\nu''_{kj}} \ . \tag{4.44}$$

For the rate coefficients a modified Arrhenius form is used, which includes the temperature dependency:

$$k_k = A_k T^{n_k} \exp\left(-\frac{E_k}{\mathcal{R}T} \right) \ . \tag{4.45}$$

A_k is the frequency factor, n_k the pre-exponental temperature coefficient, and E_k the activation energy of reaction k. Note, that the units for A_k are depending on the order of the reaction p and on the temperature coefficient n_k.

$$[A_k] = \left[\frac{\mathrm{m}^{3(p-1)}}{\mathrm{s\ kmol}^{p-1}\mathrm{K}^{n_k}} \right] \ . \tag{4.46}$$

4.3.1 Third Body Efficiencies

Reactions can include third-body reactans, as for example in: $C_2H_4 + M -> C_2H_2 + H_2 + M$
The reaction rate will dependend on the concentration of the third-body M which can represent selected species with their specific third-body-efficiencies. The effective concentration of M will the be calculated from the concentration of the single species weighted by their efficiencies:

$$C_M = \sum_{i=1}^{n} z_i C_i. \tag{4.47}$$

As an example one could choose:
$z_{CH_4} = 6.5$
$z_{H_2O} = 6.5$
$z_{CO_2} = 1.5$
$z_{CO} = 0.75$
$z_{O_2} = 0.4$
$z_{N_2} = 0.4$
$z_{other} = 1.0$

4.3.2 Pressure Dependend Reactions

For some reactions the rate coefficients will depend on temperature and pressure. For these reactions the following formulation is used:

$$k = F k_\infty k_L \,. \tag{4.48}$$

Here,the Lindemann-Hinshelwood form is used:

$$k_L = \frac{k_0 \left[M\right] / k_\infty}{1 + k_0 \left[M\right] / k_\infty} \tag{4.49}$$

k_0 and k_∞ are calculated with the regular expression (4.45) by using the seperatively given kinetic parameters, A_0, n_0, E_0 and $A_\infty, n_\infty, E_\infty$, respectively. $[M] = p/RT$ is the molar density of the gas mixture.

The additional factor F is given by

$$\log_{10} F = \frac{1}{1 + \left(\log_{10}\left[k_0 \left[M\right] / k_\infty\right] \frac{1}{N}\right)^2} \log_{10} F_c \,, \tag{4.50}$$

$$N = 0.75 - 1.27 \log_{10} F_c. \tag{4.51}$$

For the pressure dependend term F_c usually empirical expressions are used. These can look like the following example:

$$F_c = 0.411 \, \exp\left(-\frac{73.4}{T}\right) + \exp\left(-\frac{T}{422.8}\right) \,. \tag{4.52}$$

Note: If one defines forward and backward reaction, k_{0b} and $k_{\infty b}$ k_{0f} and $k_{\infty f}$ will not be independent from each other. The following relation is valid, basically reflecting the equilibrium constant:

$$\frac{k_{0f}}{k_{0b}} = \frac{k_{\infty f}}{k_{\infty b}} \,. \tag{4.53}$$

Missing kinetic parameters (e.g. for k_{0b}) can be calculated from this equation.