Creating a Flamelet Library for the Steady Flamelet Model or the Flamelet/Progress Variable Approach

Heinz Pitsch

September 3, 2006

Contents

1 Background

1.1 Overview

In non-premixed combustion, fuel and oxidizer are initially separated. Chemical reactions occur only because of diffusive molecular mixing of these components. If the chemistry is fast enough, a reaction layer forms at about stoichiometric conditions. In this layer, fuel and oxygen are consumed and reaction products are formed. For hydrogen and hydrocarbon chemistry in technical devices, combustion is typically controlled by the rate of molecular mixing, although the chemistry becomes important if the chemical time-scale compares with the time-scale of the flow. In that case, local flame extinction might occur. Also the chemistry of pollutant formation is often governed by slow chemical reactions.

It has long been realized that the direct closure of the mean chemical source term in the Reynolds averaged species transport equations or even in the LES filtered equations can hardly be accomplished, and conserved scalar methods have been used in many applications. Using so called coupling functions, the rate of mixing of fuel and oxidizer can be described by a non-reactive scalar, the mixture fraction. Different definitions have been used for the mixture fraction [1, 2], but essentially the mixture fraction is a measure of the local equivalence ratio. Hence, the mixture fraction is a conserved scalar, independent of the chemistry. This leads to the so called conserved scalar method, which forms the basis for most of the combustion models for non-premixed turbulent combustion. Considering the simplest case of infinitely fast chemistry, all species mass fractions and the temperature are a function of mixture fraction only. In the example of LES, if the sub-filter probability density distribution of the mixture fraction is known, the Favre-filtered mass fractions $\widetilde{Y}_i$, for instance, can then be obtained as

$$\widetilde{Y}_i = \int\limits_0^1 Y_i(Z) P(Z) dZ \,, \tag{1}$$

where Z is the mixture fraction and $P(Z)$ is the marginal density weighted filter probability density function (FPDF) of the mixture fraction.

1.2 FPDF of the Mixture Fraction

Different methods have been proposed to determine the FPDF. In the commonly used presumed FPDF model, a presumed shape of the FPDF is provided. Similar to RANS models, a beta-function distribution is usually assumed for the marginal FPDF of the mixture fraction in LES, and parameterized by the first two moments of the mixture fraction.

In RANS models typically a transport equation is solved for the scalar variance $\left\langle Z''^2 \right\rangle$, in which the Favre averaged scalar dissipation rate $\langle \chi \rangle$ appears as an unclosed sink term that requires modeling. The additional assumption of a constant ratio of the integral time-scale of the velocity fluctuations τ_t and the scalar fields leads to the expression

$$\langle \chi \rangle = c_\phi \frac{1}{\tau_t} \left\langle Z'^2 \right\rangle \,, \tag{2}$$

where c_ϕ is the so called time-scale ratio.

In the models most commonly used in LES [3, 4], the scalar variance transport equation and the time-scale ratio assumption are actually used in the opposite sense. Instead of solving the sub-filter variance equation, the assumption that the scalar variance production appearing in that equation equals the dissipation term leads to an algebraic model for the dissipation rate of the form

$$\widetilde{\chi} = 2D_t \left(\nabla \widetilde{Z} \right)^2 \,, \tag{3}$$

where an eddy diffusivity model was used for the sub-filter scalar flux in the production term. In Eq. (3), $D_t = (c_Z \Delta)^2 \widetilde{\mathcal{S}}$ is the eddy diffusivity, where c_Z can be determined using a dynamic model and $\widetilde{\mathcal{S}} = |2\widetilde{S}_{ij}\widetilde{S}_{ij}|^{1/2}$ is the characteristic Favre-filtered rate of strain. Writing Eq. (2) for the sub-filter scales and combining it with Eq. (3) then leads to the model for the scalar variance

$$\widetilde{Z'^2} = c_V \Delta^2 \left(\nabla \widetilde{Z} \right)^2 \,, \tag{4}$$

where $\tau_{t,\Delta} \sim 1/\widetilde{\mathcal{S}}$ has been assumed, and a new coefficient c_V has been introduced, which can be determined dynamiclly following [4].

From this, the mean and variance of mixture fraction can be computed, and the beta-distribution as a model for the FPDF of mixture fraction can be evaluated.

1.3 Steady Flamelet Model

Flamelet models for non-premixed combustion have been introduced by [5, 6]. The basic assumption is that the chemical time-scales are short enough so that reactions occur in a thin layer around stoichiometric mixture on a scale smaller than the small scales of the turbulence. This has two consequences: the structure of the reaction zone remains laminar, and diffusive transport occurs essentially in the direction normal to the surface of stoichiometric mixture. Then, the scalar transport equations can be transformed to a system where the mixture fraction is an independent coordinate. A subsequent asymptotic approximation leads to the flamelet equations,

$$\frac{\partial Y_i}{\partial \tau} - \rho \frac{\chi}{2} \frac{\partial^2 Y_i}{\partial Z^2} - \dot{m}_i = 0 \,, \tag{5}$$

where τ is the time, ρ is the density, and $\dot{m}_i$ are the chemical production rates. Similar equations can be derived for other scalars such as temperature or enthalpy. The steady laminar flamelet model is developed by assuming that the flame structure is in steady state.

Then, the time derivative in Eq. (5) can be neglected, which leads to the steady flamelet equations

$$\rho \frac{\chi}{2} \frac{\partial^2 Y_i}{\partial Z^2} + \dot{m}_i = 0\,, \tag{6}$$

The solution is then only a function of the scalar dissipation rate and the boundary conditions. If, for a given case, the boundary conditions are assumed to be fixed and the scalar dissipation rate, which is a function of mixture fraction, can be modeled such that it is parameterized by a scalar value, then the flamelets can be pre-computed and tabulated in terms of these quantities.

For the scalar dissipation rate typically a functional form is presumed, which corresponds to an idealized configuration. Peters [6] has proposed to use the distribution of the scalar dissipation rate in a counterflow diffusion flame or an unsteady mixing layer, which lead to the same functional form given by an inverse error function. Other expressions have been proposed [7, 8] and are available in the code, but as a default, the inverse error function expression by Peters [6] is used. This functional form can be written as

$$\chi(Z) = \chi_{\mathrm{ref}} \frac{f(Z)}{f(Z_{\mathrm{ref}})}\,, \tag{7}$$

where χ_{ref} is the scalar dissipation rate at a reference value of the mixture fraction Z_{ref}. Typically, stoichiometric conditions are used to define the reference value, which is also used in the present context.

Using Eq. (7), the solutions of the steady state form of the flamelet equations can be written as $\phi(Z, \chi_{\mathrm{st}})$, where ϕ stands for any scalar that is given from the solutions of the flamelet equations, such as temperature, species mass fractions, or chemical source terms. The dependence of the solution on two parameters has to be considered in the computation of the filtered values, which are then given as

$$\widetilde{\phi} = \int\limits_0^\infty \int\limits_0^1 \phi(Z, \chi_{\mathrm{st}}) P(Z, \chi_{\mathrm{st}}) dZ d\chi_{\mathrm{st}}\,. \tag{8}$$

Since χ_{st} and Z are independent, this can be written as

$$\widetilde{\phi} = \int\limits_0^\infty \int\limits_0^1 \phi(Z, \chi_{\mathrm{st}}) P(Z) P(\chi_{\mathrm{st}}) dZ d\chi_{\mathrm{st}}\,, \tag{9}$$

Typically, the distribution of χ_{st} is neglected, which is equivalent to assuming a delta-function for the distribution. It follows for $\widetilde{\phi}$

$$\widetilde{\phi} = \int\limits_0^1 \phi(Z, \chi_{\mathrm{st}}) P(Z) dZ\,. \tag{10}$$

Introducing the assumption of a beta-function sub-filter distribution of the mixture fraction leads to

$$\widetilde{\phi} = \int\limits_0^1 \phi(Z, \chi_{\mathrm{st}}) \beta(Z; \widetilde{Z}, \widetilde{Z''^2}) dZ \;, \tag{11}$$

where the semicolon indicates that the beta-distribution is a function of Z and determined by the two parameters $\widetilde{Z}$ and $\widetilde{Z''^2}$. With this form, all quantities of interest $\widetilde{\phi}$ can be pre-computed and tabulated as function of $\widetilde{Z}$, $\widetilde{Z'^2}$, and χ_{st}. Then, at each time step in an LES computation, in each cell, these quantities are determined from the LES solver, and the density, temperature, and other filtered quantities can be retrieved from the lookup table. The determination of χ_{st} is described in Pitsch and Peters [8], but it is not without problems, since the inversion of Eq. (7) becomes singular if the mixture fraction tends towards zero or unity. Here we use a different method. For each given set of $\widetilde{Z}$, $\widetilde{Z'^2}$, and χ_{st}, according to Eq. (11), also the filtered scalar dissipation rate $\widetilde{\chi}$ can be determined. This means that for given values of $\widetilde{Z}$ and $\widetilde{Z'^2}$, a certain value of χ_{st} implies a value of $\widetilde{\chi}$ and vice versa. It can be easily shown that this is a unique relationship. Hence in the representation of the flamelet library, χ_{st} can be replaced by $\widetilde{\chi}$. The library is then given in the form

$$\widetilde{\phi} = \widetilde{\phi}(\widetilde{Z}, \widetilde{Z'^2}, \widetilde{\chi}) \,. \tag{12}$$

Note that in deriving the flamelet equations as shown in Eq. (5), the assumption of unity Lewis numbers for all species has been invoked. Formulations for non-unity Lewis numbers or full diffusion models have also been formulated [2], but these formulations should only be used for laminar flames. For turbulent flames, it has often been shown that the unity Lewis number assumption leads to substantially improved results compared with the detailed transport model. The reason is that the transport of fuel and oxygen to the reaction zone has to occur over large distances in space. This transport is predominantly caused by turbulent transport rather than molecular diffusion.

1.4 Flamelet/Progress Variable Model

The steady flamelet model is easy to use and does not really increase computational cost. Also, the model is a substantial improvement over the fast chemistry assumption and typically leads to very good results, if the assumptions are not violated. However, one problem with the steady flamelet model is that, by itself, it cannot predict flame lift-off, which is a typical feature in many technical combustion devices. The incorrect representation of flame lift can lead to large discrepancies in flow field predictions, as shown by Pierce and Moin [9, 10] and Pitsch et al. [11], which would certainly translate into inaccuracies in the prediction of pollutants. Also, local extinction and re-ignition events cannot be described properly. The reason is that the solution space used in the steady flamelet model is very restrictive. The solutions of the steady flamelet equations are shown in Fig. 1 for the example of methane/air combustion at standard conditions. Here, the maximum flamelet temperature is shown as function of the stoichiometric scalar dissipation rate. The upper branch describes the fully burning solutions and the lower branch the non-burning state. The intermediate branch comes from solutions of the steady flamelet equations, but the solutions are unstable. In the steady flamelet model, only the upper branch is used. Although also the lower branch describes physical solutions to the equations, incorporating these solutions in the model, would lead to large jumps in temperature and density for dissipation rates around the extinction limit, and hence to numerical instabilities. It was also seen from direct numerical simulations, that the fully extinguished solution at dissipation rates larger than the quenching limit, are never reached [12].

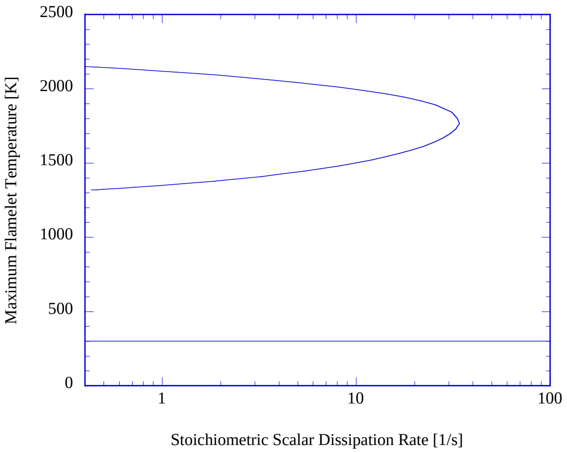

Figure 1: Computed S-shaped curve for methan/air at standard conditions.

The flamelet/progress variable model (FPV) model was developed by Pierce and Moin [9, 10] specifically for LES to address these problems. The model is constructed as follows. All filtered scalar quantities are evaluated from a flamelet library and a presumed joint FPDF. This is similar to the steady flamelet model, but here, the two quantities for parameterizing the flamelet solutions are the mixture fraction Z and a reaction progress parameter λ, which is related to the reaction progress variable C. Hence, also the presumed joint FPDF that is used in the model has to be formulated for these two quantities. A transport equation is solved for the filtered reaction progress variable. The reaction progress variable can be defined in different ways. Here, the sum of the mass fractions of CO_2, CO, H_2O, and H_2 is used. Hence,

$$C = Y_{\mathrm{CO2}} + Y_{\mathrm{H2O}} + Y_{\mathrm{CO}} + Y_{\mathrm{H2}}. \tag{13}$$

The filtered chemical source term appearing in the C-transport equation is determined from the flamelet library together with the presumed FPDF of Z and λ. This model is similar to the Intrinsic Low Dimensional Manifold (ILDM) model [13], where the ILDM is constructed from the solution of the flamelet equations. The great advantage of this formulation is that the use of the reaction progress parameter as a parameter in the flamelet model allows for the description of local extinction and reignition, and, in the absence of premixed flame fronts, also partially premixed combustion. In addition, it is relatively straightforward to implement the model in an LES solver, and it does not add substantially to the computational cost.

The reaction progress parameter is defined from the progress variable in a way that it is independent of the mixture fraction. Details of the definition can be found in Ihme et al. [14]. Then, the FPDF of Z and λ are independent and the assumption of a delta-function for the reaction progress parameter leads to

$$\widetilde{\phi} = \int\limits_0^1 \phi(Z,\lambda)\beta(Z;\widetilde{Z},\widetilde{Z''^2})dZ\,, \tag{14}$$

In a way similar to the replacement of χ_{st} by $\widetilde{\chi}$ in the steady flamelet model, also here, for the tabulation, the reaction progress parameter λ can be replaced by the reaction progress variable $\widetilde{C}$, which is directly known from the solution of a transport equation. The tabulation is then given in the form

$$\widetilde{\phi} = \widetilde{\phi}(\widetilde{Z},\widetilde{Z''^2},\widetilde{C})\,. \tag{15}$$

The transport equation for the reaction progress variable can be derived from the definition of C and the assumption of unity Lewis numbers for the species involved in its definition as

$$\frac{\partial \rho C}{\partial t} + \nabla\cdot(\rho \boldsymbol{u} C) = \nabla\cdot(\rho D\nabla C) + \dot{m}_C\,. \tag{16}$$

From this, the modeled filtered reaction progress variable equation can be written as

$$\frac{\partial \overline{\rho}\widetilde{C}}{\partial t} + \nabla\cdot(\overline{\rho}\widetilde{\boldsymbol{u}}\widetilde{C}) = \nabla\cdot\Big(\overline{\rho}(D+D_t)\nabla\widetilde{C}\Big) + W_C\,, \tag{17}$$

with

$$W_C = \overline{\rho}\widetilde{\left(\frac{\dot{m}_C}{\rho}\right)}\,. \tag{18}$$

Here, the source term has been written such that the Favre average of the chemical source term appears. This term is closed using the flamelet assumption. The term W_C is provided from the flamelet table in the form of Eq. (15), which will be described below.

2 Flamelet Models in Large-Eddy Simulations

The implementation of flamelet models in large-eddy simulations can be sub-divided into four different parts:

1. Computation of flamelet solutions, stored in so called flamelet files. These files then describe $\phi(Z, \chi_{\mathrm{st}})$ for the steady flamelet model and $\phi(Z, \lambda)$ for the FPV model. The files are generated with the FlameMaster code that will be described below and are stored in the so called flamelet format defined by Rogg and Peters [15].

2. Converting the flamelet files into a flamelet library using Eqs. (11) or (14). The resulting file contains the filtered quantities parameterized according to Eqs. (12) or (15) depending on the model employed.

3. Once the library is generated, it can be accessed from an LES code. For this, the library will have to be read and access functions for reading data from the tabulated library are required. This is done using an interface, which is also described below

4. To access the flamelet library, certain parameters need to be computed. For this, additional equations need to be implemented in the LES code. For both the steady flamelet model and the FPV model, the filtered mixture fraction and the sub-filter mixture fraction variance need to be computed. Different methods are possible, one of which is described in Section 1.2. For the steady flamelet model, the filtered scalar dissipation rate is also required, and a model for this is also given in Section 1.2. For the FPV model, the filtered progress variable needs to be computed by solving a transport equation, which is given in Pierce and Moin [9].

2.1 Generating Flamelet Solutions Using the FlameMaster Code

The FlameMaster code is used to solve the flamelet equations. Several inputs are required. As indicated in Section 1.3, boundary conditions for the flamelet equations need to be provided and several solutions have to be generated for varying values of χ_{st}. In addition, a chemical mechanism along with the thermodynamic data for all species need to be provided. Mechanisms are available in several different formats. The most common format is the CHEMKIN format. Here we will use the FlameMaster format, and a conversion of the file formats needs to be performed if the mechanism and thermo data are only available in CHEMKIN format. The tools performing this conversion are a part of the FlameMaster package and are described in the tutorial, which is a part of the code distribution. Once the chemical mechanism is in FlameMaster format, it needs to be pre-processed. This is done using the ScanMan code, which reads in the chemical mechanism and the thermodynamic data, and creates a binary mechanism file that is used as input for FlameMaster. For the solution of the flamelet equations with FlameMaster, an input file needs to be provided. The default name for the input file is `FlameMaster.input`. Other file names can be specified with the command line option '-i'. The preparation of the input files and the general use of FlameMaster are described in the manual that is a part of the code distribution. Here, we only want to describe two specific tasks. First, the generation of a start solution, and second, the computation of the S-shaped curve.

2.1.1 Generating a Start Profile

The FlameMaster code uses a Newton solver to solve the non-linear flamelet equations. The Newton solver is guaranteed to converge only if the initial guess is close to the solution. Therefore, a start profiles file is always specified. This start profiles file has the solution to the flamelet equations in the flamelet format [15] for conditions close to the conditions that will be computed. The question that is addressed here is, what to do if a new fuel or a new chemical mechanism is considered, for which such a solution does not exist.

The FlameMaster code also has a solver for the unsteady flamelet equations. The solution procedure is very similar to using the steady solver. To use the unsteady solver, only a few changes need to be made to the input file. Here we will describe how to use the unsteady solver to generate a file containing the species and temperature start profiles for a fully burning case that can be used for the steady solver. The idea is that it is easy to generate a non-burning solution. It is obvious from the flamelet equations that this solution can be obtained in good approximation from a linear interpolation between the boundary conditions at $Z = 0$ and $Z = 1$. If temperature and pressure are high enough the unsteady solution can be computed until auto-ignition occurs and the unsteady solution comes to a steady state. Since this can take an enormous time for standard conditions, the strategy here is to start from an non-burning solution at high oxidizer temperature, say $T_{\mathrm{ox}} = 1000\,\mathrm{K}$, and a high pressure, say $p = 50\,\mathrm{bar}$. The simulation is then done, until auto-ignition has occurred, then pressure and oxidizer temperature are gradually reduced within a certain time window until the required conditions are reached, and then, the solution is continued in time until steady state conditions are reached. The solution is written to a file, which can then be used as a start profiles file for the steady solver.

In order to facilitate the changes in the conditions, an input file can be used for the unsteady flamelet solver that specifies the boundary conditions and parameters as function of time. This file name is arbitrary, but is typically called `CA.in`. This file has to be specified in the FlameMaster input file with, for example,

```
CAinFile = ./CA.in
```

The format of the CAinFile is clear from the following example.

```
RPM = -1
VarsIn = 6
Time(s) Pressure(Pa)    TOx(K)  TFuel(K)  Sci(1/s)  ZR
0       5000000         1000    300       1.0       1
```

```
0.01    5000000           1000    300         1.0         1
0.02     100000            300    300         1.0         1
0.03     100000            300    300         1.0         1
```

The first three lines should not be changed, but additional rows of the same format can be added. All numbers are just whitespace separated and do not need to appear in colums. The first number in a line is the time in seconds. The second is the pressure in Pascal. The third and fourth are the oxidizer and fuel temperature in Kelvin, the fifth column is the stoichiometric scalar dissipation rate in inverse seconds. The last number should be present, but should not be changed. It denotes the maximum value of the mixture fraction in the computation, which here should always be unity.

This specific example starts the simulation with an air temperature of 1000 K and a pressure of 50 bar. Between 10 ms and 20 ms, the air temperature is linearly decreased to 300 K and the pressure to 1 bar. Then the simulation proceeds with constant conditions to a time of 30 ms. The scalar dissipation rate during the entire simulation is $1\,\mathrm{s}^{-1}$.

An alternative good strategy for the dissipation rate is to set it to a low value initially, since this promotes auto-ignition, and change it then to a higher value, since this promotes mixing, which accelerates reaching a steady state. However, if the dissipation rate is too high, extinction might occur and the steady state will be the non-burning solution.

Solutions are written to disk at different points in time. The file name contains the time, and the file for $t = 30$ ms should be used as start profiles file for the steady flamelet solver.

Another approach to make large changes in the computed conditions is given in the steady solver itself by using a continuation method, which is a successive change in parameters or boundary conditions. For this, several commands for the FlameMaster input file are available and will be explained here using a few examples. To change the oxidizer temperature from a low to a very high value, for instance, the following sequence of commands can be used:

```
ContType is Temperature
ContSide is left
ContInc = 25
ContBound = 800
```

If these commands are added to the FlameMaster input file, the simulation will start with the conditions prescribed otherwise. Once these converge, a continuation is performed to increase the oxidizer temperature. The temperature continuation is specified with the command `ContType`. Several other options for `ContType` are available in FlameMaster. For this particular flame type, however, the only other available option is `Pressure`. The command `ContSide` specifies where the temperature continuation should be performed `left` here refers to the $Z = 0$, which is the oxidizer side. Two other available options are `right` and `bothsides`. If `ContType` is `Pressure`, naturally `ContSide` does not need to be specified. The command ContInc specifies the increment to be applied at each step. Note that this is additive for the temperature and multiplicative for the pressure. The continuation will stop when the value specified by ContBound is reached.

This method can also be used to change the boundary species compositions. Then, instead of ContType is replaced by two commands specifying the species that are to be changed. An example is

```
FromSpecies is CH4
ToSpecies is C2H4
ContSide is right
ContInc = 0.02
ContBound = 1.0
```

Here, on the fuel side denoted by right, CH_4 is replaced by C_2H_4 in steps of 0.02. This procedure will also automatically revert to smaller increments in case the simulation does not converge.

2.1.2 Computation of a Sequence of Solutions Along the S-shaped curve

Conventional Method - A steady flamelet library consists of a series of solutions along the S-shaped curve. There are several ways such a series of solutions can be computed with the FlameMaster code. The most straightforward method is to simply specify a set of scalar dissipation rates in the FlameMaster input file. For instance, if a start solution is used for a scalar dissipation rate close to $\chi_{\mathrm{st}} = 2\,\mathrm{s}^{-1}$, the following set of lines in the input line produces the solutions for the specified dissipation rates, starting with $\chi_{\mathrm{st}} = 2\,\mathrm{s}^{-1}$.

```
Scalar DissipationRate = 2
Scalar DissipationRate = 3
Scalar DissipationRate = 5
Scalar DissipationRate = 7
Scalar DissipationRate = 10
Scalar DissipationRate = 15
Scalar DissipationRate = 20
Scalar DissipationRate = 30
Scalar DissipationRate = 50
Scalar DissipationRate = 70
Scalar DissipationRate = 100
Scalar DissipationRate = 150
Scalar DissipationRate = 200
Scalar DissipationRate = 300
Scalar DissipationRate = 500
Scalar DissipationRate = 700
Scalar DissipationRate = 1000
```

This example can be found in the FlameMaster distribution in the file

`FlameMaster/FlameManRun/Diff/Steady/CH4/FlameMaster.input`

For the example shown in Fig. 1, the extinction value of the stoichiometric scalar dissipation rate is about $\chi_{\mathrm{st,q}} = 34.9\,\mathrm{s}^{-1}$. Therefore, the simulations for $\chi_{\mathrm{st}} = 50\,\mathrm{s}^{-1}$ is expected not to converge. In the input file, one typically specifies a maximum number of iterations of about 25 or 50. If after this number of Newton iterations the solver still has not converged, the solver will start again with the last converged solution and attempt an intermediate value of the scalar dissipation rate. In the example here, a value of $\chi_{\mathrm{st}} = 35\,\mathrm{s}^{-1}$ will be attempted. This will also not converge and the solver will attempt $\chi_{\mathrm{st}} = 31.25\,\mathrm{s}^{-1}$. After this converges, the solver will return to the last non-converged attempt and so on. This iterative procedure continues until a certain minimum increment of scalar dissipation rate is reached. The last solution will therefore be for a scalar dissipation rate very close to the extinction limit.

After the simulation has been proceeded and completed from the initial value of the scalar dissipation rate towards the quenching limit, a new simulation can be performed starting from the same start solution, but towards smaller scalar dissipation rates. The input file for this is in the same directory in file `FlameMasterDown.input`. The input for dissipation rates could then be

```
Scalar DissipationRate = 1.5
Scalar DissipationRate = 1
Scalar DissipationRate = 0.7
Scalar DissipationRate = 0.5
Scalar DissipationRate = 0.3
Scalar DissipationRate = 0.2
Scalar DissipationRate = 0.15
Scalar DissipationRate = 0.1
Scalar DissipationRate = 0.07
Scalar DissipationRate = 0.05
Scalar DissipationRate = 0.03
Scalar DissipationRate = 0.02
Scalar DissipationRate = 0.015
Scalar DissipationRate = 0.01
Scalar DissipationRate = 0.007
Scalar DissipationRate = 0.005
Scalar DissipationRate = 0.003
Scalar DissipationRate = 0.002
Scalar DissipationRate = 0.0015
Scalar DissipationRate = 0.001
```

All these simulations will typically converge. Note that if too small values of the scalar dissipation rate are attempted, the solver will not converge anymore, since the flamelet equations become singular. However, since the dissipation rate is proportional to the inverse of a residence time, values that are such small that the code cannot converge also do not make physical sense.

Note also that for the computation of steady flamelets for application in a turbulent combustion model, effects due to radiation should be neglected. It was found by Pitsch and Peters [8] that the interaction of the radiation time with the residence times becomes important for low scalar dissipation rate. Since this interaction is neglected in the steady flamelet equations, the results are substantially improved by not considering radiative heat losses.

The resulting flamelet files are stored in the sub-directory `OutMethan` and the corrresponding S-curve is shown in Fig. 2, where each solution is marked by a symbol. The data for the S-curve can be extracted from the solution files by using the ListTool program that comes with the FlameMaster distribution. The command used here is

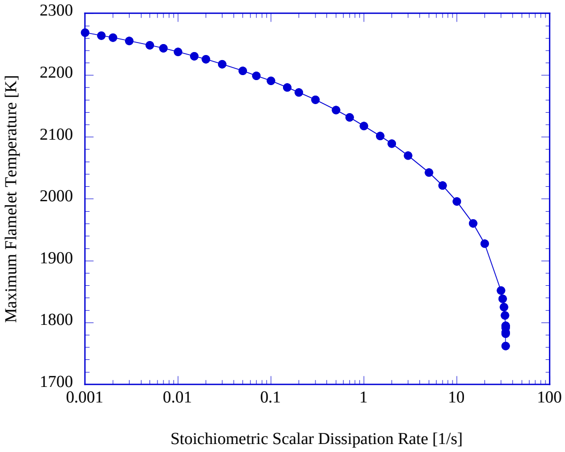

Figure 2: Computed upper burning branch of the S-shaped curve for methane/air flames at standard conditions.

```
ListTool -s syms -r rCH4.dout OutMethan/CH4_p01_0chi*300
```

Since the computation becomes singular at the extinction limit, the middle branch of the S-shaped curve cannot easily be computed using the method described in the previous section. The problem is in finding an initial solution on the middle branch that can be used for the continuation procedure described above. Since the solutions are unstable, these cannot be found with the unsteady solver as described in Section 2.1.1. This procedure will always lead to solutions on the stable branches. A closer inspection of the turning point from Fig. 2 is shown in Fig. 3. There is some scatter in the points very close to the turning point, which is caused by the effect remeshing in this very sensitive regime. The smallest computed dissipation rate interval has been chosen small enough so that the solver performs many simulations very close to the turning point. There is then a good chance, although this is not guaranteed, that the solver will at some point jump to the middle branch. It should then be possible to use this solution as start profile for a simulation towards smaller values of the scalar dissipation rate which then computes the middle branch of the S-shaped curve. The results of such a simulation are shown in Fig. 4

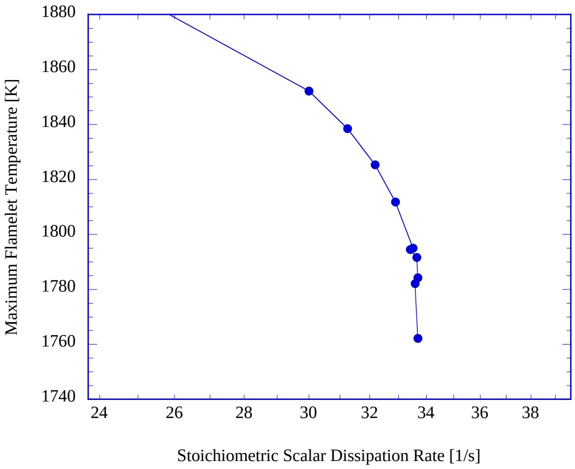

Figure 3: Close up of the turning point in the S-shaped curve shown in Fig. 2

together with the data for the upper branch. The input values for scalar dissipation rates used in this simulation were

```
Scalar DissipationRate = 33
Scalar DissipationRate = 30
Scalar DissipationRate = 25
Scalar DissipationRate = 20
...
```

and are given in file `FlameManDownMiddleBranch.input`. The simulation was run using the command

```
FlameMaster -i FlameManDownMiddleBranch.input -s OutMethan/CH4_p01_0chi33.6883tf0300to0300
```

which uses the flamelet profile with the smallest computed maximum temperature as start profile for the new simulation.

Note that the actual value of the dissipation rate from that start profile is $\chi_{\mathrm{st}} = 33.6883\,\mathrm{s}^{-1}$, while the first computed value is $\chi_{\mathrm{st}} = 33\,\mathrm{s}^{-1}$. This is to make it easier for the code to converge.

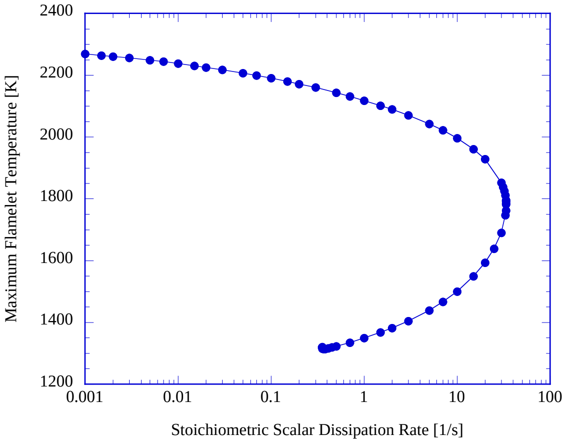

Figure 4: Computed upper and middle branch of the S-shaped curve for methan/air at standard conditions

A typical filename of a flamelet file produced using this method is `CH4_p01_0chi00001tf0300to0300`, which means the fuel is methane, pressure is 1.0 bar, stoichiometric scalar dissipation rate is $1\,\mathrm{s}^{-1}$, fuel temperature is 300 K and oxidizer temperature is 300 K. The file name here does not provide any information whether the solution is part of the upper or middle branch. Therefore it is important here to write the solutions for the upper branch and middle branch into different directories to avoid that files are overwritten. The output directory can be specified, for instance, by using the following line in the input file

```
OutputPath is ./OutputUpperBranch
```

In the example shown here, the output for the middle branch was written into the directory `OutMethanMiddleBranch`. The ListTool command can just be expanded by these files.

Arclength Continuation Method - All solutions of the S-shaped curve, including those along the unstable path, are used in the FPV model. Although the method described in the previous section works in most cases, a more reliable method is desirable. Another way of computing around the turning point is to use the so called arclength continuation method [16]. In this method, the length of a curve such as that shown in Fig. 4 is the parameter, and the scalar dissipation rate is computed as an eigenvalue of the problem. This method is implemented in FlameMaster V3.3 and can be invoked by the following line in the input file

```
ArclengthCont = TRUE
```

If this method is used, only one or two scalar dissipation rates need to be specified in the input file. The solver will start with the first specified scalar dissipation rate and continue until the end of the curve is reached, which is when the solver cannot converge, because the scalar dissipation rates become too small. If only one value for the scalar dissipation rate is specified, the solver will continue after computing the solution for that value towards higher scalar dissipation rates. The direction can be specified by adding a second value for the scalar dissipation rate to the input file. If this second value is larger than the first, the code will compute towards higher values, if the second value is lower than the first, the solver will compute towards lower values. Typically, a solution path would first be computed towards larger values, and then a second simulation towards lower values. As an example, the input values

```
Scalar DissipationRate = 2
Scalar DissipationRate = 1
```

would first compute the solution for $\chi_{\mathrm{st}} = 2\,\mathrm{s}^{-1}$ and then just continue towards lower values of the scalar dissipation rate. The current settings in the code are such that the change in temperature from one to the next point on the curve is about 20 K or less, and the change in scalar dissipation rate between two points is about a factor of 1.3 or less. The input files for these simulations are included in the FlameMaster distribution in the directory

```
FlameMaster/FlameManRun/Diff/Steady/CH4/
```

and the file names are

```
FlameMasterCont.input
FlameMasterContDown.input
```

Results from simulations where the upper and the middle branch of the S-shaped curve were computed with the arclength continuation method are shown in Fig. 1. Although it was mentioned before that radiation should be neglected in these simulations, it is interesting to remark that a closed loop can be obtained when radiation is considered. This is because radiative heat losses at very low scalar dissipation rates lead to extinction. This curve is shown in Fig. 5. This loop can be computed in either direction and it can be started from any point except the vertical turning points. For the present case, cannot compute around the point of lowest temperature, because of the stiffness encountered for low scalar dissipation rates at simultaneously small temperatures. Therefore, here the solver was started from $\chi_{\mathrm{st}} = 2\,\mathrm{s}^{-1}$ first towards larger and then towards smaller values of the scalar dissipation rate. For slightly different conditions though, the solver will continue computing the loop and never stop, except for possible convergence problems at the lower left turning point. The maximum number of files computed with this method is therefore set to 50. If necessary, this can be changed by increasing the value of the variable `fMaxFlamelets` in file `FlameManSource/TCountDiffFlameMix.C`. The advantages of this method are that the computed points are approximately equidistant in this space.

A typical file name using this method is `CH4_p01_0chi7.34531tf0295to0295Tst1906`, where the stoichiometric temperature is added at the end of the file name to avoid overriding files with the same scalar dissipation rate on the other branch.

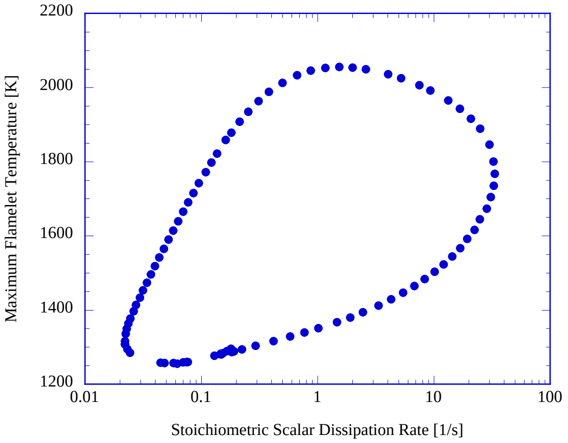

Figure 5: Steady flamelet solutions with radiation. Because of the long residence times, radiation causes extinction at low scalar dissipation rate.

References

[1] R. W. Bilger. The stucture of diffusion flames. *Combust. Sci. Technol.*, 13:155, 1976.

[2] H. Pitsch and N. Peters. A consistent flamelet formulation for non-premixed combustion considering differential diffusion effects. *Comb. Flame*, 114:26–40, 1998.

[3] S. S. Girimaji and Y. Zhou. Analysis and modeling of subgrid scalar mixing using numerical data. *Phys. Fluids*, 8:1224–1236, 1996.

[4] C. D. Pierce and P. Moin. A dynamic model for subgrid-scale variance and dissipation rate of a conserved scalar. *Phys. Fluids*, 10:3041–3044, 1998.

[5] N. Peters. Local quenching due to flame strech and non-premixed turbulent combustion. *Combust. Sci. Technol.*, 30:1–17, 1983.

[6] N. Peters. Laminar diffusion flamelet models in non-premixed turbulent combustion. *Prog. Energy Combust. Sci.*, 10:319–339, 1984.

[7] J. S. Kim and F. A. Williams. Structures of flow and mixture-fraction fields for counterflow diffusion flames with small stoichiometric mixture fractions. *SIAM J. Appl. Math.*, 53:1551–1566, 1993.

[8] H. Pitsch, M. Chen, and N. Peters. Unsteady flamelet modeling of turbulent hydrogen/air diffusion flames. *Proc. Combust. Inst.*, 27:1057–1064, 1998.

[9] C. D. Pierce and P. Moin. Progress-variable approach for large eddy simulation of non-premixed turbulent combustion. *J. Fluid Mech*, 504:73–97, 2004.

[10] C. D. Pierce and P. Moin. *Progress-Variable Approach for Large Eddy Simulation of Turbulent Combustion.* PhD thesis, Stanford University, 2001.

[11] H. Pitsch, P. Trouillet, C. D. Pierce, E. Tribbett, C. M. Sipperley, C. F. Edwards, and C. T. Bowman. A joint experimental/large-eddy simulation study of a model gas turbine combustor. *Western States Section Meeting of the Combustion Institute, 2002, San Diego, CA*, pages WSSCI 02S–59, 2002.

[12] P. Sripakagorn, S. Mitarai, G. Kosaly, and H. Pitsch. Extinction and reignition in a diffusion flame (a direct numerical simulation study). *J. Fluid Mech*, 518:231–259, 2004.

[13] U.A. Maas and S.B. Pope. Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. *Comb. Flame*, 88:239–264, 1992.

[14] M. Ihme, C. M. Cha, and H Pitsch. Prediction of local extinction and re-ignition effects in non-premixed turbulent combustion by a flamelet/progress variable approach. *Proc. Combust. Inst.*, 30:793–800, 2004.

[15] N. Peters and B. Rogg, editors. *Reduced Kinetic Mechanisms for Applications in Combustion Systems.* Springer Verlag, 1993.

[16] V. Giovangigli and M. D. Smooke. Extinction of strained premixed laminar flames with complex chemistry. *Comb. Sci. Technol.*, 53(1):23–49, 1987.