

MASTER

Automatic design optimization of a Bunsen burner

Tol, L.T.C.M.

Award date:
2020

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain



EINDHOVEN UNIVERSITY OF TECHNOLOGY

GRADUATION PROJECT

AUTOMATIC DESIGN OPTIMIZATION OF A BUNSEN BURNER

DEPARTMENT OF MECHANICAL ENGINEERING

RESEARCH GROUP: POWER & FLOW

By:

L.T.C.M. Tol

Student Number:

0894988

Supervisors:

Dr. Ir. J.A. van Oijen
Dr. Ir. N. Beishuizen

TU Eindhoven
Bosch Thermotechnology

October 23, 2020

Contents

1	Introduction	4
1.1	Background	4
1.2	Project goal	6
1.3	Report structure	7
2	Governing equations	8
2.1	CFD simulation	8
2.2	Adjoint optimization problem	11
2.3	Grid deformation	13
3	CFD simulation	16
3.1	Model description	16
3.2	Results	17
4	Unconstrained shape optimization	20
4.1	Model description	20
4.2	Objective function definition	22
4.3	Simulation results	23
5	Prevention of invalid deformations	26
5.1	FFD self-intersection prevention	26
5.2	Grid intersection prevention	30
6	Parameter study	34
6.1	Model setup	34
6.2	Boundary conditions	35
6.3	Simulation results	36
7	Improved optimization procedure	40
7.1	Model description	40
7.2	Simulation results	41
8	Conclusion	47
A	Governing equations	52
A.1	Preconditioning matrix construction	52
A.2	Flamelet generated manifold method	53
A.3	Adjoint equation derivation	56
A.4	Algorithmic Differentiation	57
A.5	Free form deformation basis functions	59
B	CFD simulation	60
B.1	Detailed chemistry simulation	60
C	Unconstrained optimization	62
C.1	Boundary conditions for nonphysical deformations	62
D	Prevention of nonvalid deformations	64
D.1	Partial derivatives of Bézier surface	64

Abstract

The introduction of adjoint equations into the field of aerodynamic shape optimization has greatly simplified the design process allowing development of automatic optimization procedures. In the open-source shape optimization software SU2, a combustion model using the Flamelet Generated Manifold method can be used to automatically optimize combustion problems.

In this research a Bunsen-type burner model is optimized with respect to the uniformity of the velocity at the burner outflow surface using SU2 to assess and improve the capabilities of the software. After a validation of the combustion model it is found that the burner outflow velocity is highly similar to that without combustion. While an initial adjoint-based optimization on this flow without combustion has shown an improvement of 55% towards the goal of velocity uniformity, it is determined that a more robust mesh deformation algorithm is required as the deformations can be so large that divergence of the simulation results.

The deformation algorithm is improved by implementation of an iterative procedure in which intersections between boundary surfaces are prevented based on the value of the determinant of the Jacobian matrix in Free Form Deformation. Furthermore, an additional routine is implemented that ensures the validity of the mesh after deformation based on the convexity of the elements. By controlling the deformation magnitude of the Free Form Deformation control points, results are achieved that ensure a valid deformation. This is found to increase the robustness of the deformation.

A parameter study has been done to improve the burner model with respect to the uniformity of the burner outflow velocity, which further improves the robustness by decreasing the required deformation magnitude to reach the optimized burner geometry. These measures are then used in another adjoint-based optimization procedure, which did show an increase in the robustness of the deformation algorithm but did not show an additional improvement with respect to the burner outflow velocity on the shape after the parameter study. In total, an improvement of 73% with respect to the velocity uniformity at the burner outflow surface is achieved.

1 Introduction

1.1 Background

The introduction of adjoint equations into the field of aerodynamic shape optimization has made it possible to develop automatic design optimization procedures for Computational Fluid Dynamics (CFD) problems [1, 2]. In this type of optimization, an objective is expressed in the form of a minimization problem. Using the results of a CFD simulation, the gradient of the objective function with respect to changes in geometry is calculated. This information is then used to quantify the sensitivity of the objective function with respect to the changes in geometry, after which the geometry can be changed.

This change in geometry decreases the value of the objective function, after which the same procedure can be carried out to further decrease the objective function value. This process can then be repeated until a minimum in the value of the objective function is found. A schematic visualization of this procedure can be seen in Figure 1.1.

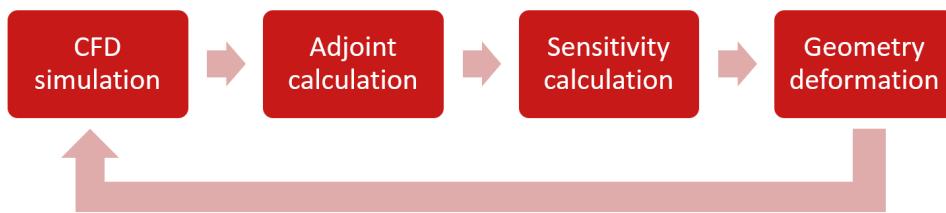


Figure 1.1: Adjoint-based shape optimization procedure

Deformation of the geometry can be done in a variety of ways. In general, this is done using parametric curves such as Bézier polynomials [3]. Through this parameterization, the curves are defined as a function of a smaller amount of points that control the deformation of the geometry. Through a change in the position of these so-called control points, the curve is deformed while ensuring smoothness of the surface.

An extension of the parametric curve technique is the Free Form Deformation (FFD) method [4]. Instead of using the parameterization to control curves, a two- or three-dimensional grid can be used to parameterize multi-dimensional objects. This parameterization can then be used to deform the entire object by changing the position of the control points, analogous to the deformations of parametric curves. An example of this through deformation of the TU/e logo as shown in Figure 1.2, where the colored dots represent the control point positions.

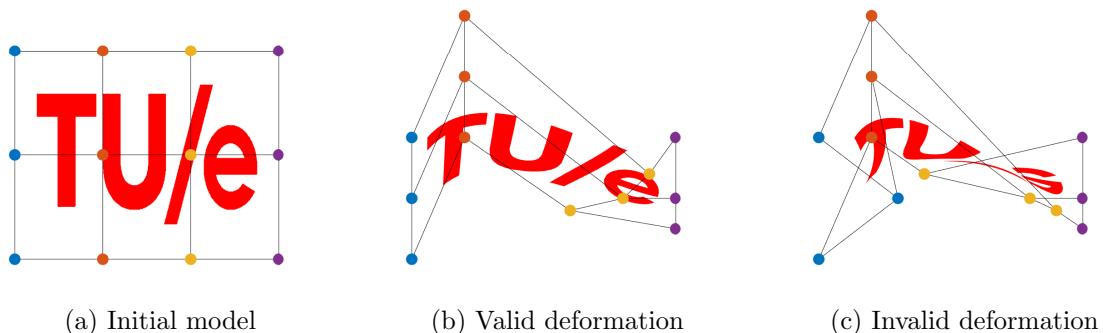


Figure 1.2: TU/e logo before and after Free Form Deformation

While FFD is a powerful technique that can be used to smoothly deform all kinds of objects, in general this does not ensure the validity of the geometry after deformation. Especially in case of large deformations, the control points can be placed in such a way that the deformed geometry is not similar to the initial model. This is illustrated in Figure 1.2c, where the TU/e logo has lost most of the resemblance to its original design.

Adjoint-based optimization software

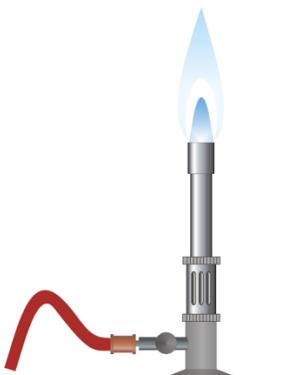
In 2012, researchers of Stanford University created the open-source software SU2 specifically for solving adjoint-based optimization problems [5]. This software can be used to perform CFD simulations, calculate the solution of the adjoint equations and perform the required deformations using the FFD technique. Furthermore, a fully automated routine is implemented that can perform the iterative adjoint-based optimization procedure for a wide variety of objective functions. Over the course of the last decade, this software has evolved and is currently being developed by researchers from various countries [5, 6].

As part of this ongoing development engineers from Bosch Thermotechnology have recently implemented a combustion model based on the Flamelet Generated Manifold (FGM) method into the software, which enables users to combine the automatic shape optimization capabilities in combination with flame modeling problems. This company, as part of the globally operating Bosch group, develops innovative solutions for both residential and commercial heating with a focus on efficiency and sustainability.

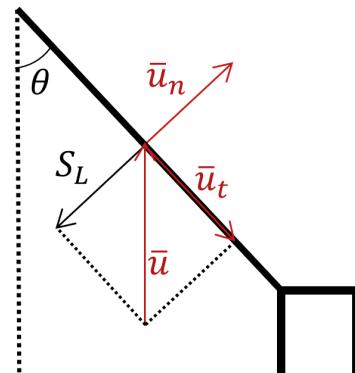
In order to aid in the design process of the next generation of heating equipment, Bosch Thermotechnology is exploring the capabilities of adjoint-based optimization through further development of the SU2 software. By using simple test cases the existing software can be validated and new limitations to the efficiency and effectiveness of the optimization procedure can be identified and tackled.

Bunsen-type burner

The test case used in this project is a Bunsen-type burner. This burner, as shown in Figure 1.3a, is characterized by its straight cylindrical tube through which the gases flow. Besides allowing for a detailed analysis on the results of simulations done on this type of burner due to the simple geometry of the tube, the results have a practical relevance since the Bunsen burner is used widely in research facilities all over the world.



(a) Bunsen burner [7]



(b) Laminar burning velocity: $S_L = \bar{u} \sin(\theta)$

Figure 1.3: Bunsen burner geometry and laminar burning velocity

An application in which the Bunsen burner is used is in the calculation of the laminar burning velocity, which can be seen as one of the fundamental properties for premixed flames. One method for doing this is based on the angle of a conical flame with respect to the burner outflow velocity [8]. In this method, the flame is assumed to have a stable conical profile above the outflow surface of the burner as represented in Figure 1.3b. For stability of the flame cone, the component of the burner outflow velocity normal to the flame surface \bar{u}_n is required to be equal to the laminar burning velocity S_L . Through decomposition of the outflow velocity vector \bar{u} , the laminar burning velocity can be calculated: $S_L = \bar{u} \sin(\theta)$.

Although this method could be used to calculate the laminar burning velocity of a Bunsen flame, the usability is limited as a consequence of the velocity profile on the burner outflow surface. Due to the influence of the wall, the flow will develop as a laminar pipe flow and the velocity profile at the outflow surface of the burner will become parabolic. This deviation in velocity leads a curve in the flame, especially above the part of the surface near the wall.

In Figure 1.4 this method is used to calculate the burning velocity of a typical flame created on a Bunsen-type burner, where the straight line is placed through a visual assessment of the flame. From this figure it can be seen that there is a range of angles around the current value in which equally acceptable results would be retrieved because the flame is not straight.

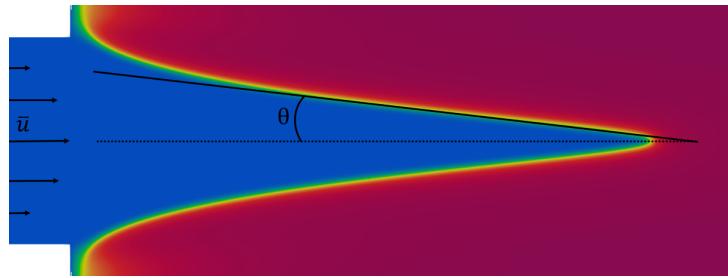


Figure 1.4: Bunsen flame profile

Research has indicated that the straightness of the flame cone can be increased by changing the geometry of the burner tube [9, 10]. Through decreasing the outflow radius with respect to the inflow radius, a more uniform velocity over the outflow surface results. However, no indications are given as to whether the used geometry is the optimal design for this purpose.

1.2 Project goal

The goal of this project is to assess and improve the capabilities of the SU2 software. Using a Bunsen burner model, the newly-implemented FGM model is validated. An adjoint-based optimization procedure is then done based on the results of the flame simulation, with the goal to optimize the burner geometry with respect to the uniformity of the velocity on the burner outflow surface.

After the initial optimization, an iterative procedure is implemented that ensures the validity of the geometry after deformation. Furthermore, the results from the adjoint-based optimization are used to identify the important parameters with respect to the uniformity of the outflow velocity, which are then varied in a parameter study. The best result of this parameter study is then used in combination with the algorithm that ensures valid deformation results in an improved adjoint-based optimization procedure to see whether the fully optimized shape of the burner can be achieved.

1.3 Report structure

In Section 2, the equations governing the adjoint-based shape optimization procedure using the FGM combustion model and FFD-based shape deformation is described. Using this theory, a CFD simulation of a Bunsen-type burner is presented in Section 3.

The objective for velocity uniformity on the burner outflow surface is described in the form of a minimization problem and an adjoint-based optimization procedure is done in Section 4 on the burner tube using the results of the CFD simulation from Section 3.

An iterative algorithm that ensures the physical validity of the geometry after deformation is elaborated in Section 5. Then, in Section 6 the results of the burner optimization as presented in Section 4 are used to vary the most important parameters that influence the magnitude of the velocity on the burner outflow surface.

An improved optimization procedure on the burner tube is presented in Section 7, using the best result from the parameter study and the algorithm that ensures the physical validity of the model after deformation. The conclusions from this research are provided in Section 8. Finally, possible approaches for further research on this topic are elaborated.

2 Governing equations

The process of adjoint-based optimization requires the solution of four procedures carried out sequentially, as described in Section 1. First, a CFD simulation of the flow problem, also called primal, is done to retrieve the flow state. Then, the solution of the adjoint equations can be calculated and used to derive the sensitivity required to perform the desired geometric deformations. This information finally is used to perform the actual shape deformation. Once this is done, the process can be restarted using the deformed geometry and the optimized solution can be retrieved by iteratively performing this four-step routine as shown in Figure 1.1.

In this section, the process will be described analogous to the way it is carried out in the optimization procedure. First, the solution of the Navier-Stokes equations for a viscous flow is described and the combustion modeling using the Flamelet Generated Manifold (FGM) method is elaborated. After that the adjoint equations are presented and finally the deformation algorithm is explained.

2.1 CFD simulation

The Navier-Stokes equations are based around solving expressions for conservation of mass, momentum and energy. Here, the variables $U = [\rho, \rho\bar{v}, \rho c_p T]^T$ are used to represent these equations in matrix form, in which ρ is the density, \bar{v} the velocity vector, c_p the heat capacity under constant pressure and T the temperature.

A problem of compressible solvers for the Navier-Stokes equations is that a poor convergence behavior is seen at low Mach number [11]. For the flow within a Bunsen tube, where the velocities are generally low, this could become a problem. Therefore, two measures have been taken to improve the convergence behavior: the flow is assumed to be incompressible and a preconditioning approach is followed. The incompressible Navier-Stokes equations can be written as [11]:

$$R(U) = \frac{\partial U}{\partial t} + \nabla \cdot \bar{F}^c(U) - \nabla \cdot \bar{F}^v(U, \nabla U) - Q = 0 \quad (2.1)$$

In this equation, the required condition on the residual is given as $R(U) = 0$. The variables $\bar{F}^c(U)$ and $\bar{F}^v(U, \nabla U)$ represent respectively the convective and viscous fluxes given by:

$$\bar{F}^c = \begin{bmatrix} \rho\bar{v} \\ \rho\bar{v} \otimes \bar{v} + \bar{I}p \\ \rho c_p T \bar{v} \end{bmatrix} \quad \bar{F}^v = \begin{bmatrix} \cdot \\ \bar{\tau} \\ \kappa \nabla T \end{bmatrix}$$

Here, ρ_T represents the derivative of the density with respect to temperature T , p the pressure, κ the thermal conductivity and $\bar{\tau}$ the viscous stress tensor. With μ representing the viscosity, the viscous stress tensor can be written as:

$$\bar{\tau} = \mu(\nabla \bar{v} + \nabla \bar{v}^T) - \mu \frac{2}{3} \bar{I}(\nabla \cdot \bar{v})$$

The last term in Equation 2.2, source term Q , can be used to include various additional terms to the equations corresponding to the required conditions. For an axisymmetric simulation, this additional term is defined as:

$$Q = \begin{bmatrix} -\rho \frac{v}{y} \\ -\rho u \frac{v}{y} + \frac{\mu}{y} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) - \frac{2}{3} \frac{\partial}{\partial x} \left(\mu \frac{v}{y} \right) \\ -\rho u \frac{v}{y} + 2 \frac{\mu}{y} \left(\frac{\partial v}{\partial y} + \frac{v}{y} \right) - \frac{2}{3} \frac{\partial}{\partial y} \left(\mu \frac{v}{y} \right) \\ -\rho c_p T \frac{v}{y} + \frac{1}{y} \kappa \frac{\partial T}{\partial y} \end{bmatrix}$$

The variables x and u represent respectively the coordinate and velocity component in the direction tangential to the axis of symmetry and y and v represent the radial components of the coordinate and velocity.

Through use of a preconditioning method, the problem is transformed into an expression that can be solved more efficiently and robustly. This preconditioning leads to the equations being solved in terms of the primitive variables $V = [\rho, \bar{v}, T]^T$ instead of the variables $U = [\rho, \rho \bar{v}, \rho c_p T]$ [11]. Using this preconditioning method and the primitive variables to describe the flow state, the resulting problem for the incompressible CFD simulation is shown in Equation 2.2. In Appendix A.1, the construction of the preconditioning matrix and the derivation of this equation is provided in more detail.

$$R(V) = \Gamma \frac{\partial V}{\partial t} + \nabla \cdot \bar{F}^c(V) - \nabla \cdot \bar{F}^v(V, \nabla V) - Q = 0 \quad (2.2)$$

The required condition on the residual is now given as $R(V) = 0$. The convective and viscous fluxes and the source term can be written in the same way as given earlier. Variable Γ represents the preconditioning matrix and $\bar{F}^c(V)$, given by:

$$\Gamma = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & \rho_T \\ \frac{u}{\beta^2} & \rho & 0 & \rho_T u \\ \frac{v}{\beta^2} & 0 & \rho & \rho_T v \\ \frac{c_p T}{\beta^2} & 0 & 0 & \rho_T c_p T + \rho c_p \end{bmatrix}$$

Here, β represents the preconditioning variable that is added for an increased robustness, where $\beta^2 = \epsilon^2 (\bar{v} \cdot \bar{v})_{max}$ with $\epsilon = 2$.

Flamelet Generated Manifold

Combustion is modeled using the Flamelet Generated Manifold method [12, 13]. In this method, a flame is characterized by one or more controlling variables that are a measure for the progression of the flame at a certain position in space. These variables, the so-called progress variables, are defined in terms of a one-dimensional flame. The specific expression used for the modeling of the flame can be chosen in various ways. However, in order to be able to give information about the progression of the flame, these variables are required to have a unique value from the unburnt to the burnt side of the flame [14]. In Appendix A.2, this method is described in more detail.

The manifold used for this project is two-dimensional, with as progress variables the enthalpy h and a combination of species mass fractions \mathcal{Y} . During runtime of the simulation, the equations for the two progress variables are solved. In Equations 2.3 and 2.4, these equations are shown for respectively the progress variable \mathcal{Y} and enthalpy h [14].

$$\frac{\partial(\rho\mathcal{Y})}{\partial t} + \nabla \cdot (\rho\bar{u}\mathcal{Y}) - \nabla \cdot \left(\frac{\kappa}{Leyc_p} \nabla \mathcal{Y} \right) = \dot{\omega}_{\mathcal{Y}} \quad (2.3)$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho\bar{u}h) - \nabla \cdot \left(\frac{\kappa}{c_p} \nabla h \right) = \nabla \cdot \mathcal{H} \quad (2.4)$$

In these equations, κ represents the thermal conductivity, Ley the Lewis number, $\dot{\omega}_{\mathcal{Y}}$ the source term of the progress variable per unit time and volume and \mathcal{H} the enthalpy flux due to differential diffusion. In the implementation of this problem, a unity Lewis number is assumed as a consequence of using methane and the effects of differential diffusion are neglected. This gives $Ley = 1$ and $\nabla \cdot \mathcal{H} = 0$.

Implementation of this method into the governing flow equations is done through rewriting the conservation equations in such a way that the conservation equations for both progress variables can be written analogously. In Equation 2.5, the resulting equation for progress variable \mathcal{Y} is shown.

$$\frac{\partial(\rho\mathcal{Y})}{\partial t} + \nabla \cdot (\rho\bar{u}\mathcal{Y}) - \nabla \cdot (\rho\mathcal{D}\nabla\mathcal{Y}) - Q = 0 \quad (2.5)$$

In this equation, $\dot{\omega}_{\mathcal{Y}}$ is included in the source term and $\rho\mathcal{D} = \kappa/c_p$ due to the assumption of unity Lewis numbers. The use of the FGM method allows the variables ρ , μ , c_p , \mathcal{D} and $\dot{\omega}_{\mathcal{Y}}$ to be stored in a lookup table as a function of \mathcal{Y} and h . The variable ρ_T can then be calculated by dividing the density by the temperature. The values in this table can be calculated and stored prior to simulation using one-dimensional flame simulation software like CHEM1D [15]. Since the variables now only have to be retrieved from a lookup table instead of calculated during runtime, the required computational effort is reduced.

Through calculation of the enthalpy in a separate equation in the FGM method as presented in Equation 2.4 and the relation between enthalpy and temperature, the temperature is essentially calculated twice: once in the incompressible Navier-Stokes equations and once in the FGM equations. However, the calculations as done in the Navier-Stokes equations do not take all properties of the added combustion into account. Therefore, the values as calculated using FGM are used in subsequent iterations for the variables that are calculated in both methods.

The progress variable \mathcal{Y} as used in this project is defined as:

$$\mathcal{Y} = Y_{CO_2} + Y_{CO} + Y_{H_2O} + Y_{H_2} + Y_{NO} + Y_{NO_2}$$

In Figure 2.1a, the combined mass fraction of this variable is shown using a one-dimensional flame simulation. Since it increases smoothly over the entire length of the flame, this can be seen as a suitable progress variable. Furthermore, the source term of the progress variable as a function of the progress variable value shows a smooth profile as can be seen in Figure 2.1b. As a consequence of the unity Lewis number assumption, the enthalpy stays constant throughout the one-dimensional flame.

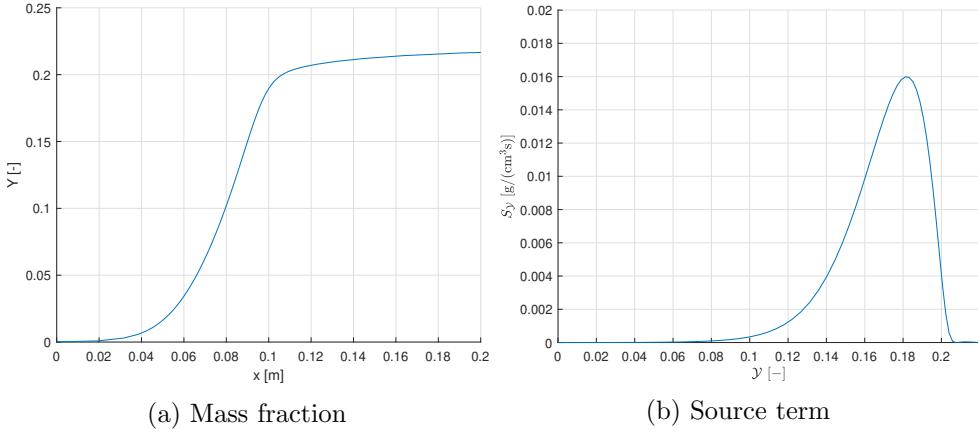


Figure 2.1: Progress variable mass fraction and source in one-dimensional flame

2.2 Adjoint optimization problem

From the solution of the CFD simulation, the flow state variables contained in vector V are retrieved. In the solution of the adjoint equations, the values of these variables are used to calculate the desired deformations of the geometry.

The adjoint equations can be retrieved if an objective J is defined in the form of a minimization problem. For this problem, the exact definition is dependent on the specific problem that has to be solved and can therefore be defined in many different ways. The general problem can be defined as [16]:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & J(V(\alpha), \alpha) \\ \text{subject to} \quad & R(V(\alpha), \alpha) = 0 \\ & g(\alpha) \geq 0 \\ & h(V(\alpha)) \geq 0 \end{aligned} \tag{2.6}$$

In this function, J is the objective function that has to be minimized, which is dependent on flow state V and design variables α . The design variables are used to control the shape of the model and can either be coupled directly to the positions of the surface nodes or indirectly through parameterization of the surface. The choices for design variables α as used in this project are described in more detail in Section 2.3. The constraint $R = 0$ is the solution of the CFD simulation, as shown in Equation 2.2. The variables g and h represent respectively geometric and physical constraints, which can be added optionally if they are required in the specific problem being modeled.

The desired movement of the design variables in the adjoint-based optimized optimization procedure is retrieved by calculation of the sensitivity of objective J with respect to α : $\frac{dJ^T}{d\alpha} = \frac{\partial J^T}{\partial \alpha} + \frac{\partial J^T}{\partial V} \frac{\partial V}{\partial \alpha}$. The solution of this expression provides the direction in which each design variable has to be changed in order to decrease the value of the objective function.

Although this expression could be solved on its own, this would require calculation of flow state V for every change in design variable α . Specifically, this means that for every design variable an entire CFD simulation has to be performed. Especially when there are a lot of design variables, this would take up a lot of computational time.

In the adjoint equations, this inconvenience is solved by introducing a so-called adjoint state Ψ . This is a Lagrange multiplier, which allows the calculation of $\frac{dJ}{d\alpha}$ to be done without calculation of the partial derivative with respect to flow state V : $\frac{dJ^T}{d\alpha} = \frac{\partial J^T}{\partial \alpha} - \Psi^T \frac{\partial R}{\partial \alpha}$ with $\frac{\partial R^T}{\partial V} \Psi = \frac{\partial J}{\partial V}$. Here, R represents the solution of the governing flow equations as presented in Equation 2.2. More details on the derivation of this expression is provided in Appendix A.3.

Within the field of adjoint-based optimization, this equation is solved in one of two ways: using either the continuous or the discrete adjoint approach. In the continuous adjoint approach the gradient $\frac{dJ}{d\alpha}$ is calculated from the analytical expressions for the governing equations and then discretized to provide a numerical solution. In the discrete adjoint approach the gradient is calculated based on the discretized governing equations. This difference can be seen as either calculating “the inexact gradient of the exact cost function” [17, p. 2] in the continuous approach or “the exact gradient of the inexact cost function” [17, p. 2] in the discrete approach.

The choice for using either the continuous or discrete approach is dependent on multiple factors, including the complexity of the derivation of the required gradients and the computational effort required for obtaining the solution to the problem. In the continuous approach the derivative of the flow and additional transport equations has to be calculated and discretized by hand [18], which is not necessary in the discrete approach [19]. The drawback of this is that the discrete adjoint approach generally requires more memory [19, 20]. Other advantages of the discrete adjoint approach are that the consistency of the solution is independent of the coarseness of the grid [19] and it is more robust than the continuous adjoint approach [18]. However, in case the computational grid is made finer, the discrete and continuous adjoint solutions should give the same results [17, 20].

In this project, the discrete adjoint approach is used in the adjoint-based optimization procedure. Calculation of the derivatives is done using Algorithmic Differentiation (AD) [21]. In this method, the discretized objective function is decomposed into simple operations and then differentiated using a repeated application of the chain rule. The calculation of these simplified operations can either be done from the input to the output side in the forward AD mode or from the output to the input side in the reverse mode, called the adjoint mode. In the forward mode the derivatives of all outputs with respect to one input variable can be calculated in one pass, whereas in the reverse mode the derivatives of one output with respect to all input variables can be calculated [22].

The goal of this project is to optimize the tube of a Bunsen burner with respect to the uniformity of the outflow velocity by changing the burner geometry. This can therefore be seen as a problem with multiple inputs and a single output, for which the reverse AD mode is more efficient. Whereas the computational cost would increase linearly for the amount of input variables in the forward mode, this can be done in the reverse mode at a cost of less than five times the computational cost of a solution of the primal problem [22]. In Appendix A.4, an example of derivative computations using the forward and reverse AD mode is shown. Implementation of AD within the SU2 software is done by use of the CoDiPack software, specifically designed for this purpose [23].

2.3 Grid deformation

A straightforward way to perform the grid deformation after calculation of the adjoint equations is to view each grid point as a design variable and changing its position according to the gradient information as calculated in the adjoint solution. However, this does not preserve the smoothness of the surfaces that are deformed. In order to ensure the smoothness of the surface boundary, deformation of the surface is done using Free Form Deformation (FFD) [4].

Although FFD can be used to deform all points in the surface in a smooth and continuous manner, this method does not take into account any specific properties with respect to element size or shape. As a consequence, especially in case of large deformations this could lead to robustness issues leading to nonvalid deformations. In order to prevent this from happening, only the surfaces on the boundary are deformed using FFD. The deformation of the internal grid points is then done using linear elasticity equations. This method is shown to perform robustly even when large deformations are required [24].

Free form deformation

The process of FFD deformations starts with the definition of a box around the model that is going to be deformed. This so-called FFD box is defined by its four corner points in a two-dimensional model. Deformation of the surface is then done by changing the position of the points in the FFD box, the control points, through which all points inside the FFD box are deformed smoothly. This deformation can be seen as similar to modeling a lump of clay [4]. In 2.2a shows an example of a rectangular model with an FFD box defined around the model.

After the definition of the FFD box, the surfaces that are going to be deformed are parameterized. In two-dimensional models, these surfaces are generally one or more of the edges on the boundary of the model. Here, parameters $0 \leq u \leq 1$ and $0 \leq v \leq 1$ are assigned using a linear interpolation between the four control point coordinates in respectively the x and y direction. For three-dimensional problems, parametric coordinate w is added analogously for the z direction. For the model of Figure 2.2a this parameterization is shown in Figure 2.2b.

The smoothness of the deformation is controlled by the amount of control points used to deform the surface. By adding extra points along the edges of the FFD box in one or more directions, the local deformations of the surface can be controlled with greater detail. This behavior is explained through use of the mathematical definition of the FFD problem, as presented in Equation 2.7 [25].

$$\bar{F}(u, v, w) = \sum_{i=0}^L \sum_{j=0}^M \sum_{k=0}^N B_i(u) B_j(v) B_k(w) \bar{P}_{ijk} \quad (2.7)$$

In this equation, function \bar{F} is used to calculate the cartesian coordinates of the surface points. These coordinates are shown to be dependent on the values of all cartesian control point positions \bar{P}_{ijk} and so-called basis functions B , which are a function of one parametric coordinate.

The basis functions B define the specific dependence of the points on the surface with respect to the control point positions. In this project the blending is done using Bézier basis functions. Using this type of blending, a smaller amount of control points can be used to perform deformations with respect to other deformation methods using a similar smoothness. In Appendix A.5 a more detailed explanation of this type of basis functions is provided. In Equation 2.8, the generalized expression for the Bézier basis functions is shown [26].

$$B_i^p(u) = \frac{p!}{i!(p-i)!} u^i (1-u)^{p-i} \quad (2.8)$$

In this equation, i represents the index of the basis function corresponding to the amount of control points in each direction as shown in Equation 2.7. Through this definition, the surface can be described through a polynomial of degree p , where $p+1$ control points are used.

Figure 2.2c shows deformed results of the FFD procedure. All four edges on the boundary of the model are deformed, using a grid of 6x2 control points. As a consequence, the surface is deformed using a polynomial of degree 5 in the x direction and linearly in the y direction.

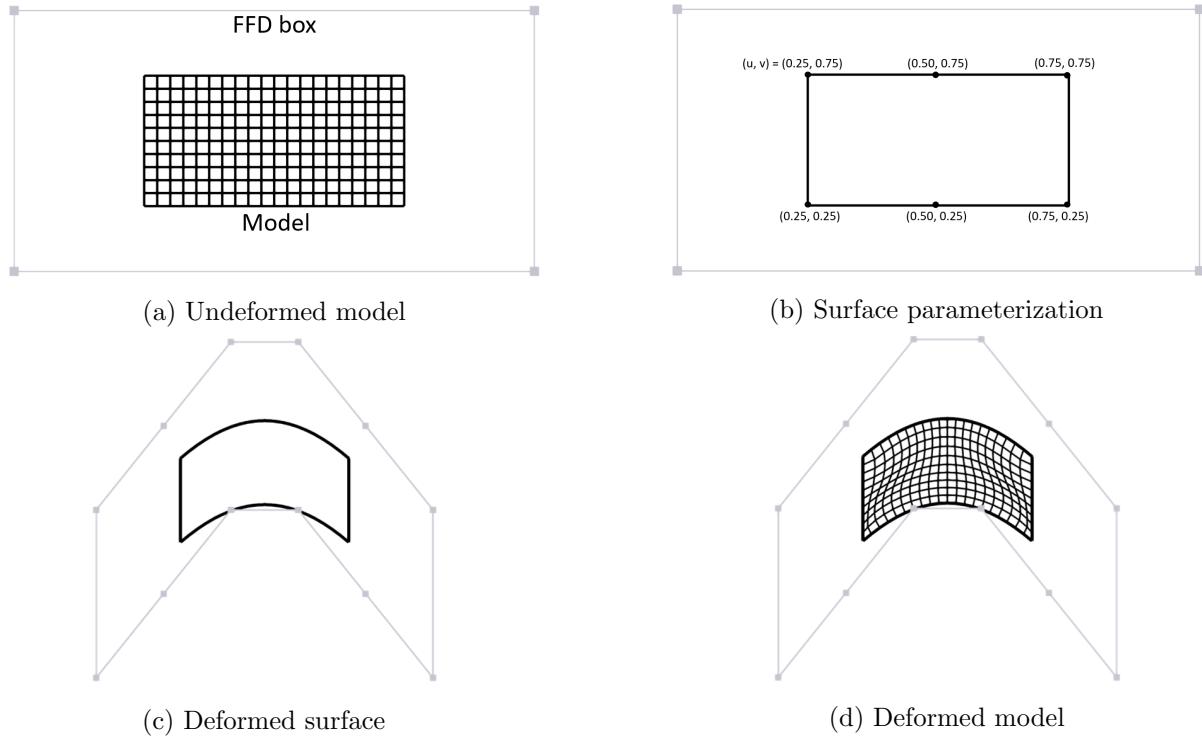


Figure 2.2: Deformation procedure

The sensitivity $\frac{dJ}{d\alpha}$ as calculated in the adjoint equations gives the gradient of the objective function with respect to a change in the design variables. However, the order of magnitude of these deformations is dependent on the specific expression used for the definition of the objective function. Applying the calculated sensitivity values directly to a change in control point positions may therefore be either too large or too small, resulting in respectively extremely deformed models or no noticeable change. This issue is solved by applying a scaling factor to the calculated sensitivity values, which directly relates the sensitivity to the maximum deformation magnitude:

$$\text{Maximum deformation} = \text{Maximum sensitivity} \times \text{Scaling factor}$$

Linear elasticity equations

After deformation of the points on the surface using FFD, the remaining points within the grid are deformed based on linear elasticity equations. This linear elasticity problem can be written as: [24]

$$\nabla \cdot \sigma = f \quad (2.9)$$

In this equation, σ represents the stress tensor and f the external forces. This equation has to hold on the entire domain Ω . The stress tensor is given in terms of the strain tensor ε by:

$$\sigma = \lambda \text{Tr}(\varepsilon) I + 2\mu\varepsilon \quad (2.10)$$

In this equation $\text{Tr}(\varepsilon)$ is the trace of the strain tensor ε . λ and μ are properties of the elastic material, defined in terms of Young modulus E and Poisson ratio ν as:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \quad \mu = \frac{E}{2(1+\nu)}$$

In solving these equations, the distribution of E is linearly dependent on either the distance to the wall or the inverse volume of a cell, where large elements can deform more than smaller cells. The stress tensor can be written as a function of the element displacements u as:

$$\epsilon = \frac{1}{2}(\nabla u + \nabla u^T)$$

This problem for finding displacements u is solved using a Galerkin method. For this problem, Dirichlet boundary conditions are used since the displacements of the surface boundary are either known through the definition as stated in the configuration for the initial simulation or through the FFD deformation as performed earlier. The result of this deformation on the model described earlier is presented in Figure 2.2d.

3 CFD simulation

In this section, the CFD simulation of a Bunsen-type burner will be described. The main properties of such a burner will be explained and the setup of the burner model is elaborated. Furthermore, the reasoning behind the mesh size used for the simulations is provided and finally, the results as acquired using the FGM model in SU2 are validated.

3.1 Model description

The Bunsen-type burner is characterized by its straight cylindrical tube. This type of burner comes in various shapes and sizes, with the defining shape parameters being the length, radius and wall thickness of the tube. The burner used in this project is modeled approximately according to the dimensions of commercially available burners. From this, a length of 100 mm, an inner radius of 6 mm and a wall thickness of 2 mm result.

Although a small section of the burner could be modeled in order to perform the flame simulation, the goal is to adapt the burner geometry in a later stage of the process. Therefore, the entire length of the burner tube is modeled. Furthermore, an additional section with a length of 60 mm has been added downstream of the burner tube. This is the section where the flame will develop. In Figure 3.1 this modeled geometry is schematically shown, with the outflow surface of the burner indicated with a gray dotted line.

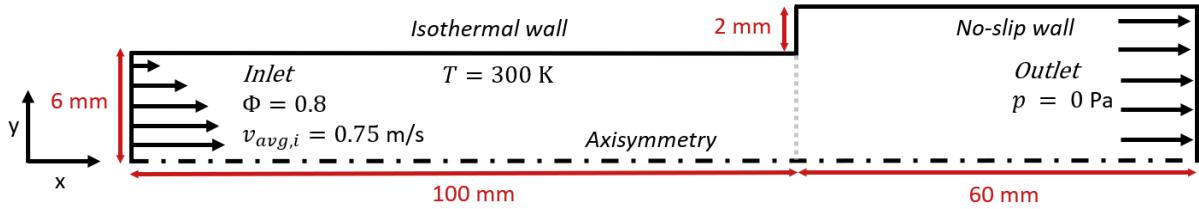


Figure 3.1: Burner model setup

Boundary conditions

The inlet velocity of 0.75 m/s has been chosen such that the flow will remain in the laminar regime. This velocity corresponds to a Reynolds number of approximately 600. This remains well below the value of $Re = 2000$, which is approximately where the transitional regime starts for pipe flows. A Poiseuille velocity profile is imposed on the inlet of the tube, in order to aid in the convergence of the problem while adding to the physical realism by using the profile of a fully developed flow. At the outlet, an ambient pressure is imposed.

For the inlet conditions of the combustion products, a methane/air mixture with a equivalence ratio of $\Phi = 0.8$ has been modeled. This value, which corresponds to a volumetric air/fuel ratio of 11.9, is used as a typical value in residential heating equipment.

Isothermal conditions have been imposed on the burner walls and a no-slip wall condition in which no heat flux can occur is selected for the boundary on the top. Finally, an axisymmetry condition has been imposed on the central axis of the burner.

Mesh size

A structured mesh consisting of quadrilateral elements is used. At the burner outflow surface the elements are square, whereas away from this surface in both directions parallel to the burner outflow surface the element size is increased in order to decrease the time required for the simulation with respect to a uniform element size. Along the direction tangential to the axis of symmetry the mesh size is uniform, since the flame will develop over this entire surface.

The size of the elements at the burner outflow surface is based on the size required to resolve the structure of the flame. The source term of progress variable \mathcal{Y} is taken as the representative variable for this required size, where its increase and decrease has to span over multiple elements. In Figure 3.2 the mesh size at 2 mm away from the burner outflow surface is shown. At this surface, the element size is approximately 0.04 mm.

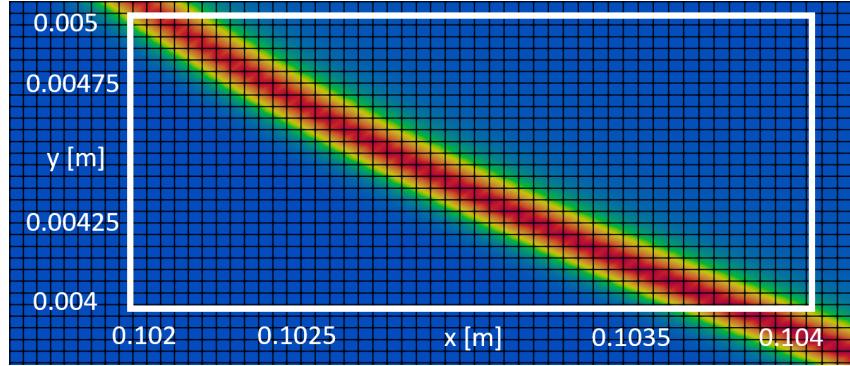


Figure 3.2: Progress variable source around the burner outflow surface

Due to the Poiseuille profile imposed at the inlet it is expected that if the flow would change at all along the length of the burner tube, this would only occur at its tip under the influence of the developing flame. Therefore, the progression of the element size has been chosen such that a maximum element length of 1 mm results at the inlet. The progression for the area downstream of the flame has been chosen such that the change in the progress variable source term would occur over at least two elements. This gives a maximum element length downstream of the flame of 0.2 mm.

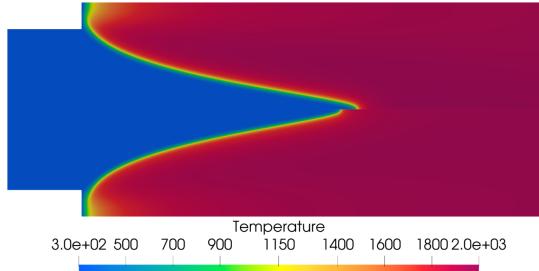
3.2 Results

Validation of the SU2 simulations is done by a comparison with the results of a simulation using detailed chemistry equations in the commercial software ANSYS Fluent. This simulation is done using a finite-rate chemistry model and is solved using a stiff chemistry solver. For this model a reduced chemical mechanism based on the GRI-1.2 reaction mechanism is used, which consists of the 17 species and 73 reactions that are the most important for lean premixed methane/air flames [27]. In Appendix B.1, the calculation of the detailed chemistry equations and a validation of the results obtained from the Fluent simulation is provided.

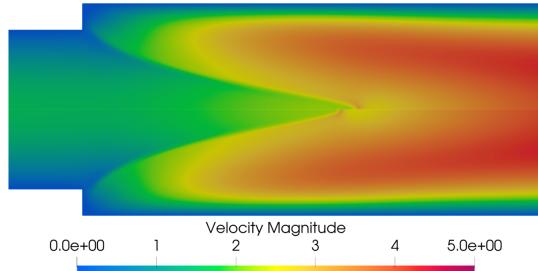
Figure 3.3 shows the comparison between the temperature and velocity results of this comparison between the SU2 and Fluent simulation, where the SU2 results are displayed in the top half of the figures and the Fluent results in the bottom half.



(a) Temperature results



(b) Close-up of temperature results



(c) Close-up of velocity results

Figure 3.3: Simulation comparison between SU2 and ANSYS Fluent

The results of both simulations seem to provide results that are in relatively close agreement, although the flame as calculated using FGM is slightly longer than the flame as calculated using detailed chemistry. In order to quantify the difference between the simulations as done using FGM and detailed chemistry, the temperature on the symmetry axis between the two simulations is compared in Figure 3.4. A close-up contour of this profile is shown in Figure 3.5.

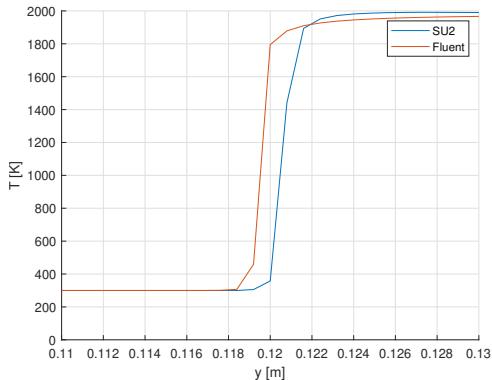


Figure 3.4: Temperature profile on symmetry axis

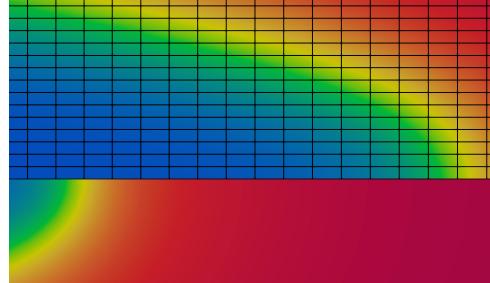


Figure 3.5: Close-up of flame tip

The difference in flame length is approximately 2 mm, which corresponds to approximately 15 element lengths in the direction parallel to the axis of symmetry. The maximum flame temperatures are 1992 K and 1974 K for respectively the SU2 and Fluent simulation.

Although the comparison as shown above shows that there is a relatively large difference in the length of the flame with respect to the thickness of the flame, the objective of this project is to optimize the burner with respect to the velocity on the burner outflow surface. This comparison between the two results and the analytical Poiseuille profile is presented in Figure 3.6.

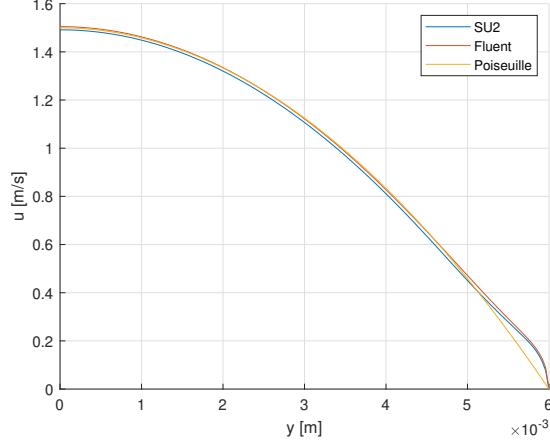


Figure 3.6: Outflow velocity comparison

The velocity on the burner outflow surface as calculated using SU2 is in close agreement with the results as calculated using detailed chemistry equations, with a difference in velocity of 1%. Another thing that can be noted from Figure 3.6 is that the outflow velocity profile resembles a parabolic profile, except near the wall. This difference in velocity is due to the influence of the flame, which develops close to that region. This is illustrated in Figure 3.7, where the temperature increase leads to a decrease in density, through which in turn the velocity is increased.

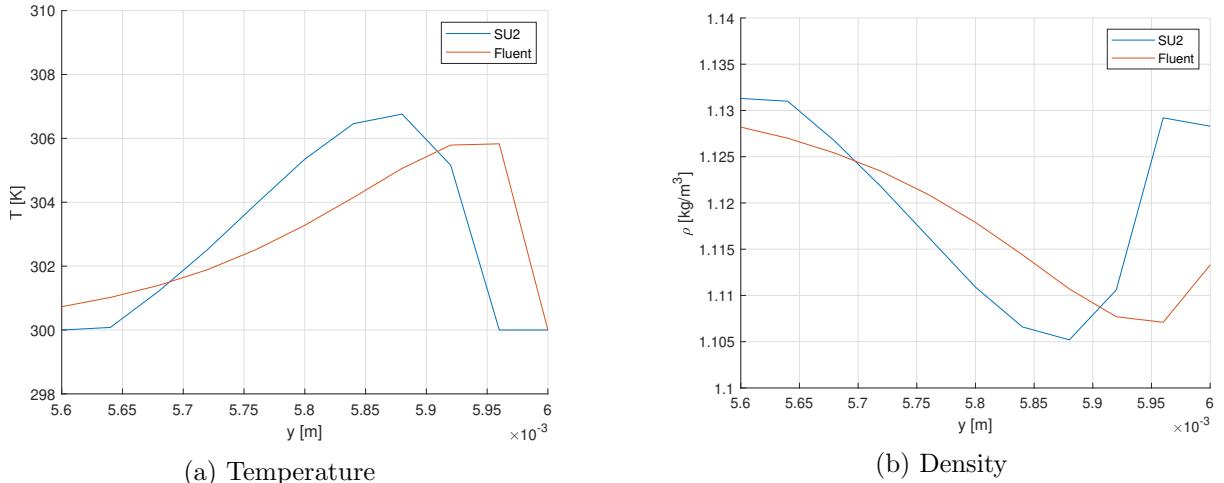


Figure 3.7: Temperature and density near the wall

As a consequence of the large similarity between the velocity on the burner outflow surface and the analytical Poiseuille profile, it is decided that the optimization of the burner can be done on a burner tube without combustion. This allows for the computations to be done using larger elements, since the small flame thickness does not have to be resolved in the simulations.

4 Unconstrained shape optimization

In the results of the CFD simulation, it has been indicated that the velocity on the burner outflow surface strongly resembles a parabolic Poiseuille profile. Therefore, the optimization with respect to the velocity on the outflow of the burner can be done using a model without combustion, which allows for a coarser grid to be used in the optimization procedure.

In this section, an unconstrained optimization procedure is described using a model without combustion. First, the characteristics of the model are described and the FFD box definition is elaborated. Then, the objective function which is used for the optimization is explained and finally, the results are presented.

4.1 Model description

The goal of this procedure is to gain more insight in the deformation of the model in case the entire tube is allowed to move freely, which has been the basis for modeling the entire length of the burner tube in the CFD simulation of Section 3. However, in case of large deformations at the end of the burner tip that would require the outflow radius to become larger, there is only limited movement allowed because of the small wall thickness.

In this section, this limitation is overcome by extending the domain to include the outer wall of the burner tube and some extra space in which the burner radius could increase. This model is shown in Figure 4.1.

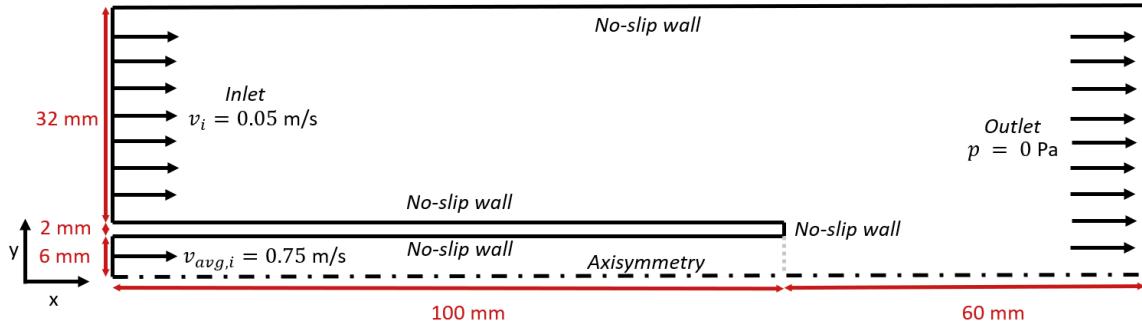


Figure 4.1: Boundary conditions for the unconstrained optimization procedure

The geometry of this model is roughly similar to that in the CFD simulation of Section 3. The entire burner tube is still modeled with a length of 100 mm, an inner radius of 6 mm and a wall thickness of 2 mm and an extra section of 60 mm is modeled downstream of the burner tube. For this model however, the outer boundary of the domain is now extended to 40 mm from the axis of symmetry. The position of this boundary, which corresponds to 5 outer burner radii from the axis of symmetry, is expected to be far enough away to let the burner tube deform without moving through this boundary.

Boundary conditions

Most of the boundary conditions of the burner without combustion have been taken similar to those modeled in the previous simulation. Specifically, this means that the burner is modeled with an axisymmetric axis, an imposed Poiseuille profile with an average velocity of 0.75 m/s at the inlet of the burner tube and ambient conditions on the outlet. As in the simulation as done earlier, the edge of the domain has been modeled using a no-slip wall condition.

A difference between the CFD simulation and the current simulation using a cold flow is that the walls are now modeled as no-slip walls instead of isothermal. Furthermore, an external flow with a small velocity of 0.05 m/s has been used in the area outside of the burner. This aids in the convergence properties of the model, without influencing the velocity on the burner outflow surface as displayed with a gray dotted line.

Mesh size

The considerations for the mesh size as used in this simulation are similar to those in the initial CFD simulation. The element size within the burner tube tangential to the axis of symmetry is kept constant at a value of 0.1 mm. Along the direction parallel to the axis of symmetry, a larger mesh size is chosen since no large changes in the flow are expected along this direction.

A minimum element length of 0.3 mm is chosen at the burner outflow surface. This length progresses in the burner tube, with a size of approximately 3 mm at the inlet. Downstream of the outflow surface, the element length progresses up to 0.4 mm. Since the flow outside of the burner tube does not have any influence on the velocity on the burner outflow surface, this element size is also increasing in length further away from the burner tube in the direction tangential to the axis of symmetry. This size increases to 1 mm at the outer surface of the domain.

FFD box definition

The FFD box has been defined in such a way that it encloses the entire burner wall. The length of the box corresponds to the length of the burner tube and is 10 cm long. The width of the box is modeled at 6 mm, which corresponds to three times the wall thickness. This is modeled symmetrically around the burner wall. A grid of 50x2 control points has been used, which gives a spacing of 2 mm between the control points along the length of the wall. This allows for a smooth deformation. In Figure 4.2, this FFD box is displayed in gray.



Figure 4.2: FFD box for unconstrained optimization

As a consequence of the Poiseuille profile modeled on the inlet of the burner tube one limitation on the deformation of the tube has been imposed. Since a deformation of the points on the inlet would result in a changed velocity profile, this would lead to a difference in the mass flow between the results before and after deformation. By fixing the points on the inlet, the Poiseuille velocity profile is retained. As this is the only constraint imposed on the burner tube, this does not limit the validity of the results since all other points on the burner wall can be deformed. This means that conclusions can be made based on the direction in which the deformations are moving and in which location along the burner wall they occur.

A maximum deformation magnitude of the surface points of 0.2 mm is used for the first iteration. This is expected to give a small but noticeable difference in the burner geometry.

4.2 Objective function definition

It has been described earlier that a uniform flow is characterized by the velocity being equal over the entire outflow surface. Furthermore, in the optimization routine the value of the objective function is minimized. This requires the objective function to be defined in terms of a cost function. For the uniformity objective function, the following requirements can be placed on the objective function:

- Velocities both higher and lower than the average velocity should increase the value of the objective function;
- The contribution of the velocity acting on an element should be dependent on the surface area of that element;
- The behavior should be independent of the magnitude of the average velocity.

From these considerations, the objective function for this problem is defined as:

$$J = \sqrt{\frac{\sum_{i=0}^n \left(\frac{v_i - v_{avg}}{v_{avg}}\right)^2 A_i}{A}} \quad (4.1)$$

In this equation, the deviation of the normal outflow velocity v_i on element i from the average velocity v_{avg} is normalized to remove the dependence on the value of the average velocity. This value is then squared in order to provide only positive contributions. A surface-averaged value is then retrieved by summing all velocity contributions multiplied by area A_i of element i and dividing by total surface area A , where $A = \sum_{i=0}^n A_i$. The square root of this value is then taken on the sum of all contributions. The average velocity is calculated by dividing the total mass flow by the total surface area and the density:

$$v_{avg} = \frac{\dot{m}}{\rho A}$$

By using this expression for the objective function, the normal outflow velocity will converge to the average velocity for a decreasing objective function value: $v_i \rightarrow v_{avg}$ if $J \rightarrow 0$. Furthermore, the tangential velocity will converge to zero as a consequence of mass conservation. This gives a fully uniform outflow normal to the outflow surface if and only if $J = 0$.

4.3 Simulation results

Using the model and objective function as stated above, the adjoint-based optimization procedure is performed. The velocity result of this optimization procedure after 10 iterations is compared to that of the model before optimization, with the deformed results shown in the top half of the figure and the initial results in the bottom half.

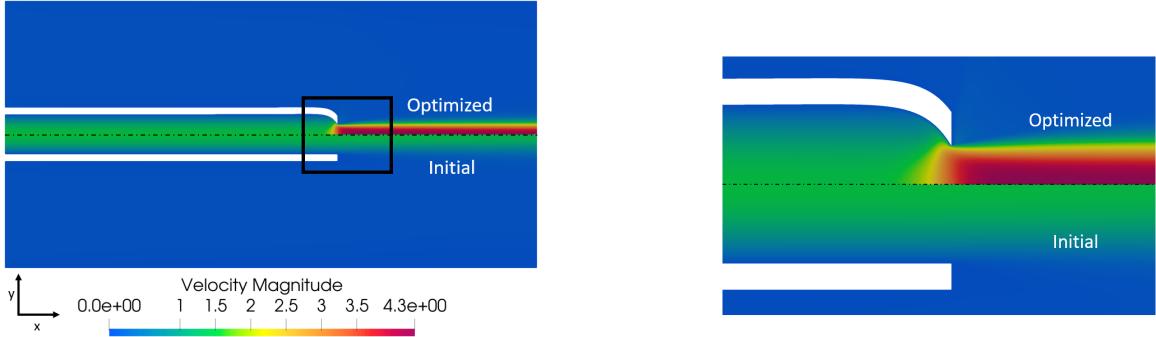


Figure 4.3: Velocity after optimization (top) compared to initial model (bottom)

It can be seen that the burner tube is only deformed in a small region around the tip, which is moved inward. This deformation leads to a 55% decrease in the value of the objective function, which is shown in Figure 4.4a.

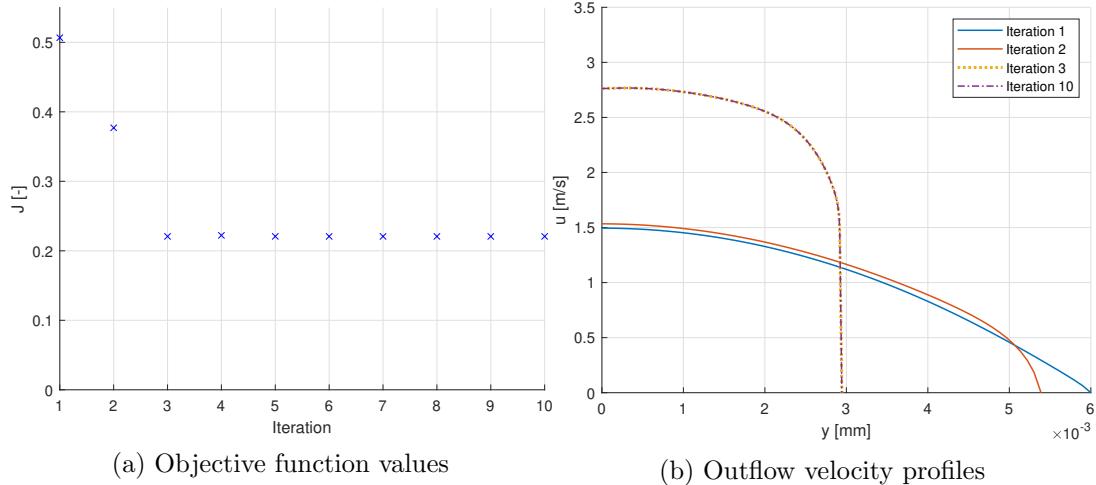


Figure 4.4: Objective function and velocity results

An interesting thing that can be seen is that the value of the objective function is only decreasing for two iterations after which it stays approximately constant. This is also represented in Figure 4.4b, where the outflow velocity profiles for iteration 1, 2 and 3 differ and the outflow velocity for iteration 10 is indistinguishable from that of iteration 3. In Figure 4.5 the outlines of the burner tips are shown for iterations 1, 2 and 3-10.

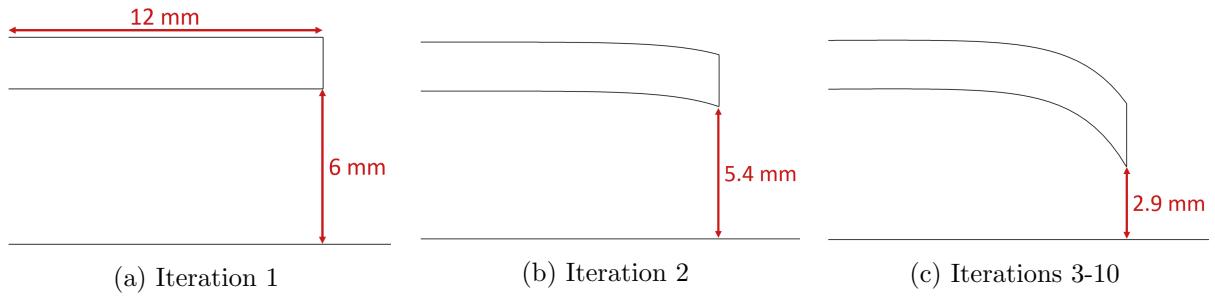


Figure 4.5: Burner tip surfaces

Although the results show a considerable improvement based on the value of the objective function, the usability of the results is limited because of the way the deformations have affected the burner outflow surface. In Figure 4.6 a close-up of the deformed mesh around the burner tip is displayed. Here, the red line represents the deformed burner outflow surface that was initially a straight line.

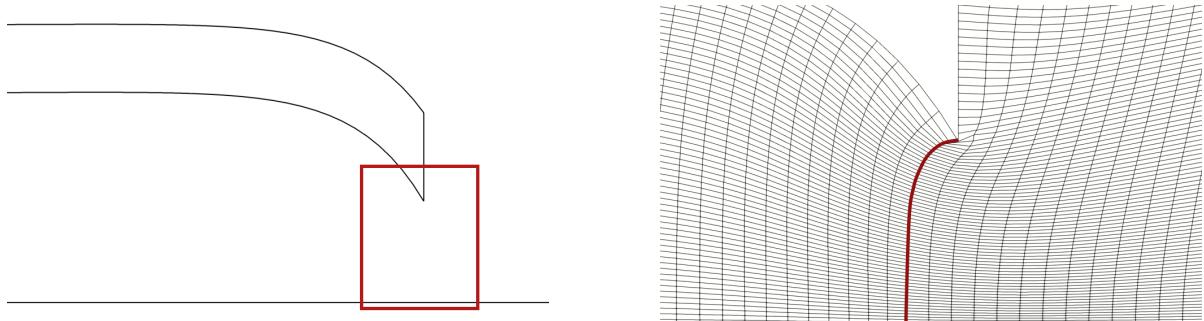


Figure 4.6: Close-up of burner outflow surface

Although this problem can be solved either through redefining the mesh after deformation or using a different definition of the FFD box, another problem occurred during subsequent optimization procedures that limited the effectiveness and efficiency of the shape optimization procedure. As a consequence of a deformation magnitude that was too large, either deformations occurred that did converge but did not provide results that could be physically realized or that led to divergence of the results. Examples of these deformations are presented in respectively Figure 4.7a, in which grid cells overlap and 4.7c where elements are both overlapping and reversed in orientation. In Appendix C.1, more information on the settings used for these deformations is given.

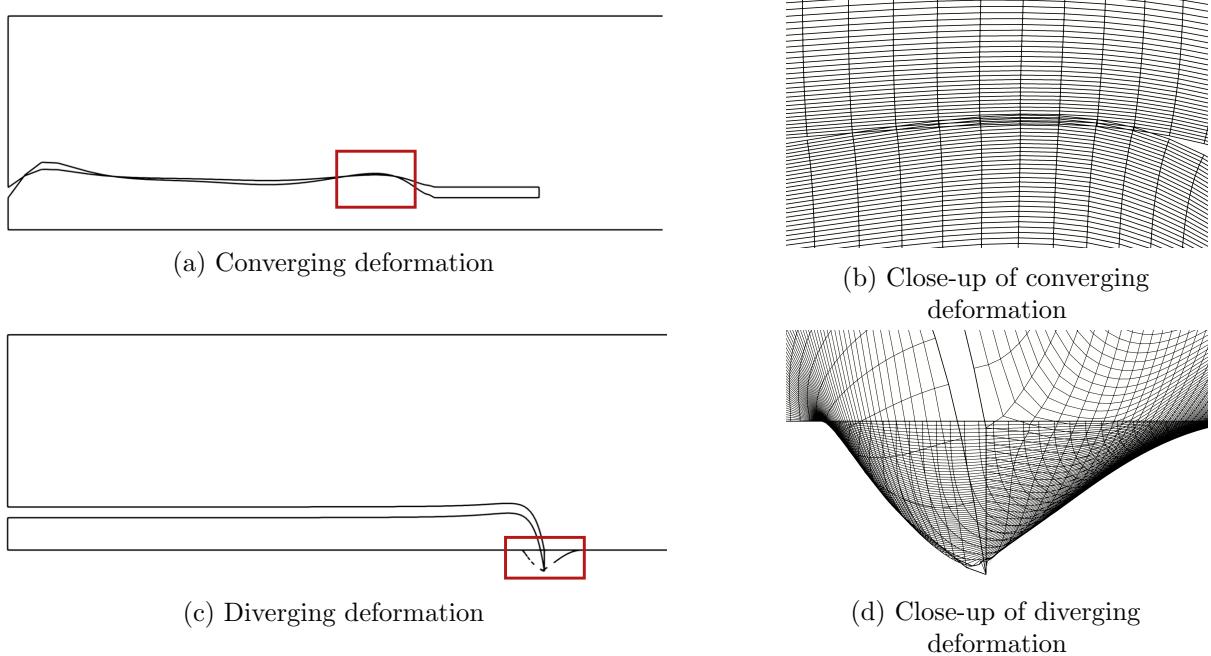


Figure 4.7: Examples of nonphysical deformation results

As both nonvalid deformations are the result of deformations that are too large for the model to handle, this could in principle be prevented by limiting the maximum deformation magnitude. By decreasing this magnitude, the point at which the nonphysical deformation occurs would not be met. However, as a consequence of the geometric deformation the sensitivity of the objective function with respect to a change in control point coordinates is changed. In case of an increase in sensitivity, the subsequent deformation magnitude is increased. This can then lead to nonvalid deformations in later iterations.

Another limitation of decreasing the magnitude of the control point deformation is that in general the amount of iterations required to achieve the optimal deformation is increased. This leads to more time being required to perform the deformations. Especially since there is no guarantee that this will lead to a valid deformation, the problem in the validity of the deformation is seen as one that needs to be solved before moving on to a new optimization routine.

Besides decreasing the magnitude of deformation of the control points, the chances of nonvalid deformations occurring can be decreased by improving the burner geometry before deformation. Through this change in geometry, the magnitude of deformations required to achieve the optimized shape are decreased. As a result the maximum deformation magnitude can be decreased, which in turn decreases the chances of the deformations being so large that a nonvalid deformation results.

In Section 5 an iterative procedure in which nonvalid deformations can be avoided is elaborated. Furthermore, in Section 6 a parameter study is elaborated in which a better shape of the burner model is found that can be used as an initial model. After that, in Section 7 a new optimization procedure is presented using the improved initial geometry and the procedure that prevents nonvalid elements from occurring in the deformed burner model. Furthermore, in that section a solution to the curvedness of the burner outflow surface is presented.

5 Prevention of invalid deformations

The results from the unconstrained burner simulation as presented in Section 4 have indicated that the adjoint-based optimization procedure can be a powerful tool for deforming a geometry in order to improve the model with respect to a given objective, through the decrease of 55% in the objective function with respect to velocity uniformity. However, from these results it has also become clear that this method is far from perfect: large deformations can result in either undesired geometries as a consequence of deformation of the burner outflow surface straightness or as a consequence of the physically impossible results leading to self-intersections of the grid points.

In this section, two measures taken to prevent self-intersections of grid points from occurring in the geometry after deformation are presented. As described in Section 2.3, the deformation of the geometry is done using a two-step process through a deformation of the boundary points using FFD and a subsequent deformation of the internal grid points using linear elasticity equations. In both of these procedures however, invalid deformations can occur.

The steps as taken in this section are designed around the two procedures separately. The earlier a nonvalid deformation can be detected and prevented, the more efficiently the process can be performed. The procedures as added in this project are described in the order they are performed in the simulations: first the prevention of self-intersections within the FFD deformation is explained and subsequently the prevention of invalid elements occurring after the internal grid point deformation is elaborated.

5.1 FFD self-intersection prevention

The occurrence of nonvalid grid points in the results of Figure 4.7a is a consequence of the deformation using FFD. The procedure implemented to prevent this from happening is a two-step process: first, it is established whether there is a self-intersection within the deformed surfaces. After that, an algorithm is implemented that deforms the surface in such a way that the intersections disappear. These procedures are now described in the order in which they are performed.

Detection of self-intersections

The detection of occurring self-intersections within a FFD surface can be done in multiple ways. One method is to look at a set of conditions to find whether the FFD surface is well-posed by means of its control points [28]. Using this method, the surface is either seen as valid or invalid. Another method used to assess FFD surfaces is based on an approximation of the curve in straight line segments through subdivision of the control polygon [29]. The possible self-intersection are then calculated using the approximated straight lines.

The method used in this project is selected because of the combination between its effectivity and the information it can provide to the modeler. In this technique, the determinant of the Jacobian matrix after deformation is calculated for every point on the surface that is deformed using FFD. In case there are one or more points that have a negative determinant, there is a self-intersection within the deformed surface [25].

Using the Jacobian-based method, the amount of points on which the surface would self-intersect can be returned. The main advantage of this method is that the amount of points that have a negative determinant can then be used to provide the user with information about the quality of the deformation: if many points are seen to be invalid, the quality of the model is further from the desired solution than when there are only a few points at which the deformed surface would be invalid. Furthermore, the self-intersections can be calculated in an exact manner.

The validity of the deformation in the Jacobian-based method is ensured if the following two conditions apply [25]:

- The surface is at least C^1 continuous on the entire domain;
- $\det(J) > 0$ for every point on the surface.

In this set of requirements, the surface continuity is a measure for the smoothness of the curve. For C^1 continuity specifically, this means that the entire FFD surface is required to have a continuous first derivative. In Appendix D.1, the derivation of the partial derivatives of a surface defined by Bézier basis functions as used in this project is provided. There, it is shown that the partial derivative with respect to a certain parametric coordinate is only dependent on the derivative with respect to that parametric coordinate. Furthermore, since the derivative of a basis function is a linear combination of two other basis functions and every basis function is continuous, it can be concluded that the C^1 continuity is always satisfied.

The determinant of the Jacobian matrix is a measure for the mapping between a point in parametric and cartesian coordinates. A negative Jacobian determinant implies that a certain positive volume in parametric coordinates is mapped onto a negative volume in cartesian coordinates. This means that a certain amount of volume is inverted at the points with negative Jacobian determinants, from which it can be concluded that there is a self-intersection within the FFD box. In Figure 5.1, an example of a deformed geometry with a self-intersection in the deformed surface is shown. Based on the negative values of the Jacobian on the position of certain points on the surface, it can be seen that this self-intersection can be detected. In this figure, the outside surface in the Jacobian plot corresponds to the surface that is deformed using FFD.

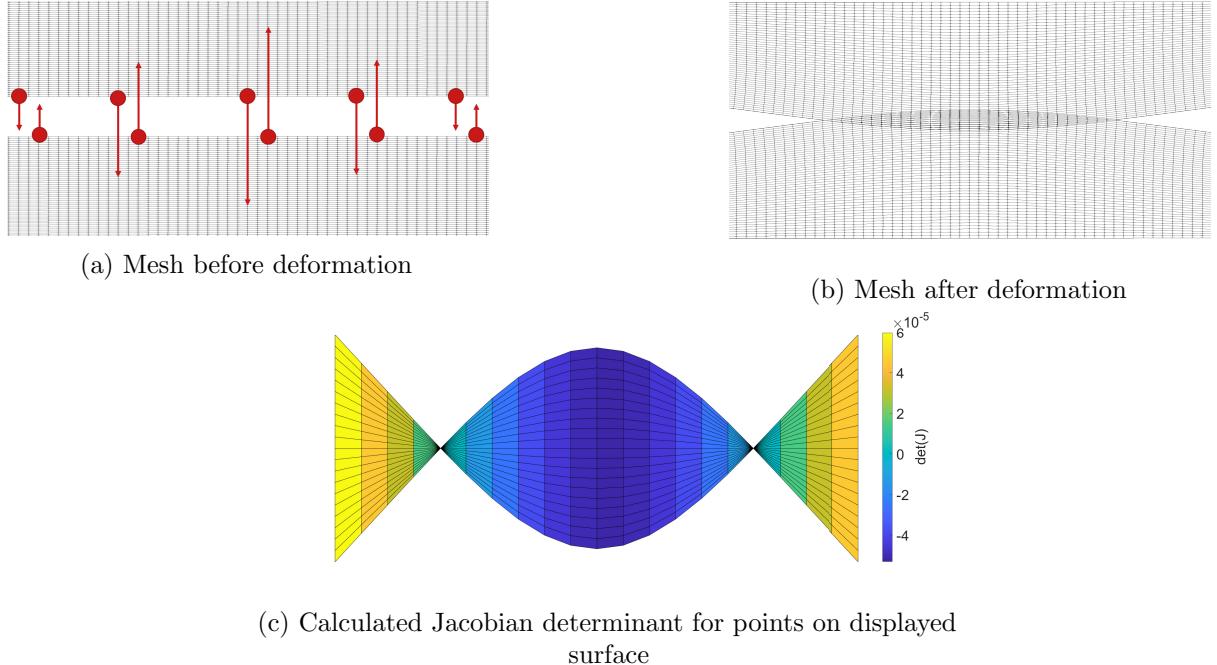


Figure 5.1: Deformation before recursive intersection prevention procedure

In Equation 5.1, the calculation of the Jacobian determinant can be seen. In this equation, J is the Jacobian matrix. Using the notation of Section 2, F_x , F_y and F_z represent the functions for finding the cartesian coordinates of the surface points after deformation and u , v and w are the parametric coordinates within the FFD box.

$$\det(J) = \begin{vmatrix} \frac{\partial F_x}{\partial u} & \frac{\partial F_x}{\partial v} & \frac{\partial F_x}{\partial w} \\ \frac{\partial F_y}{\partial u} & \frac{\partial F_y}{\partial v} & \frac{\partial F_y}{\partial w} \\ \frac{\partial F_z}{\partial u} & \frac{\partial F_z}{\partial v} & \frac{\partial F_z}{\partial w} \end{vmatrix} \quad (5.1)$$

The partial derivative of function F with respect to parametric coordinate v is presented in Equation 5.2 [26]. For the derivation of this equation, the properties that the cartesian control point coordinates are independent of the parametric coordinates and the basis functions are only dependent on one parametric coordinate are used. This is described in more detail in Appendix D.1.

$$\frac{\partial F}{\partial v} = \sum_{i=0}^L \sum_{j=0}^M \sum_{k=0}^N B_i(u) B_j'(v) B_k(w) P_{ijk} \quad (5.2)$$

In this equation, $B_j'(v)$ represents the derivative of the Bézier basis function. For basis function j of degree p , this is defined as presented in Equation 5.3 as derived in Appendix D.2. Here, $B_{-1}^{p-1}(v) \equiv 0$ and $B_p^{p-1}(v) \equiv 0$ since the values of these basis functions are not defined.

$$B_j'(v) = n(B_{j-1}^{p-1}(v) - B_j^{p-1}(v)) \quad (5.3)$$

The calculation of the Jacobian determinant as described above is done for all points on the surface after deformation. This information is then provided as feedback to the modeler in order to give an indication of the quality of the model with respect to the desired deformation as mentioned earlier. Furthermore, if one or more negative determinants are detected, the procedure to prevent this from occurring in the deformed geometry is started.

Prevention of FFD self-intersection

The prevention of self-intersections within the FFD surface is done by using a recursive procedure in which the positions of the control points are adjusted. Based on the values of the Jacobian determinants, the deformation magnitudes are either decreased or increased until a user-specified condition to end the procedure is met [25]. This can be either a maximum amount of iterations or a certain amount of iterations without points that have a negative determinant, the recursion depth. In Figure 5.2 a schematic representation of this procedure is shown, where everything within the black box is added with respect to the previous deformation procedure.

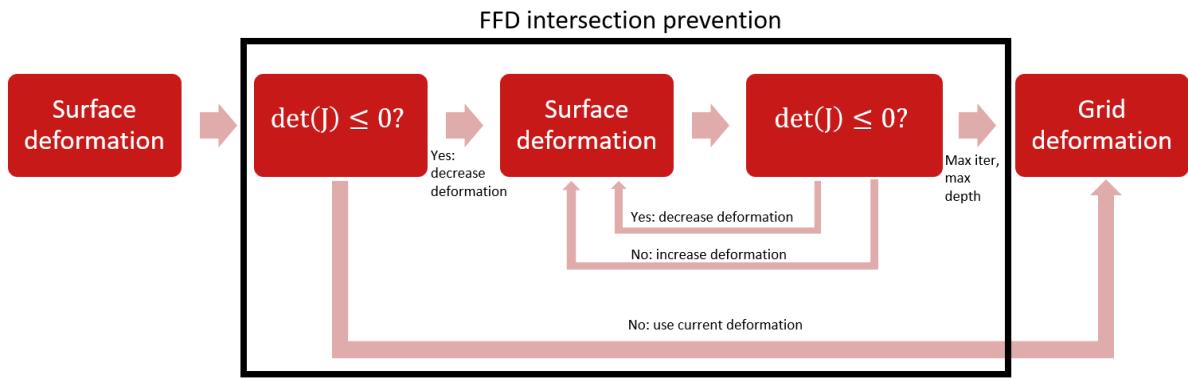


Figure 5.2: Deformation procedure after addition of FFD intersection prevention

After deformation of the surface, a first assessment of the Jacobian determinants on all points on the surface is done. If there are no points with a negative determinant, the recursive procedure is bypassed and the deformation of the internal grid points is carried out. However, if there are one or more points that have a negative determinant, the magnitude of the deformation is decreased. A new surface deformation is then performed using the adjusted control point positions and another Jacobian assessment is done. Based on the minimum value of the determinants the magnitude of deformation is adjusted and a new evaluation is done.

The adjustment of the deformation magnitude is done through a simultaneous adjustment of all control point positions, due to the indirect mapping between the points on the surface and the positions of the control points. This adjustment is implemented in such a way that the resulting deformation approaches the physically achievable maximum. If no negative determinants are present, the deformation magnitude is increased. Conversely, if there are negative determinants, the deformation magnitude is decreased. In this procedure, the amount of displacement that is either added or subtracted to the location of the control points is decreased in each iteration using the knowledge of the previous deformations. In Figure 5.3 the deformation magnitude is shown schematically.

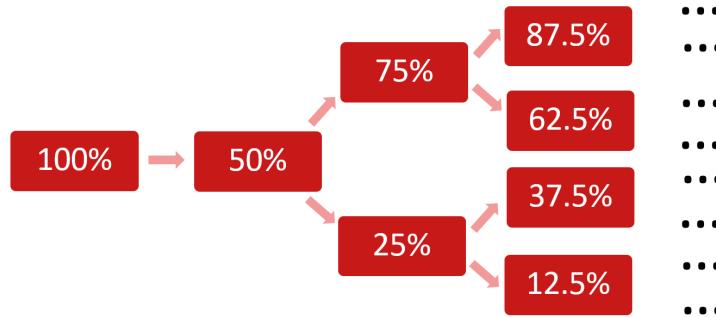


Figure 5.3: Recursive procedure for adjusting deformation magnitude

The deformation magnitude is always lowered after the first iteration. This is a consequence of the procedure only being started if there are self-intersections in the first iteration, which means that the deformation magnitude is too high and a lower value is required. After this, the difference in deformation magnitude is halved after each iteration regardless of whether it is added or subtracted. This way, the deformations can get close to the previously evaluated situation but can never exceed it. As a consequence, intersections can be prevented from occurring in the deformed geometry. In Figure 5.4 the results from this procedure on the geometry of Figure 5.1 are shown. Here, it becomes clear that the self-intersections have disappeared and the Jacobian determinant is positive on all points.

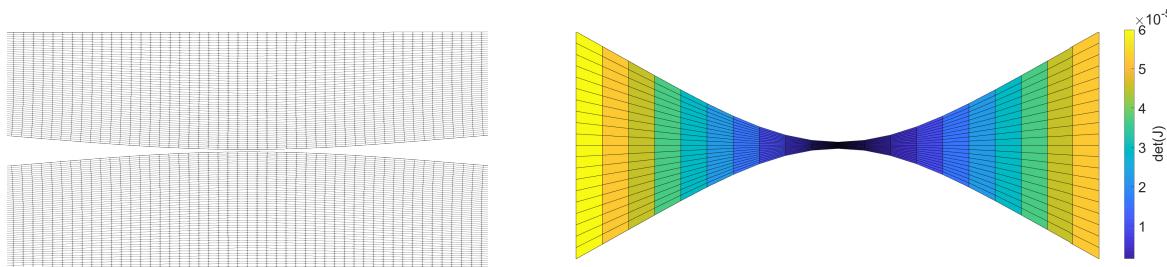


Figure 5.4: Deformation results after recursive intersection prevention procedure

At the moment, no constraints are placed on the minimum thickness of the surface, which means that in principle a point with zero thickness can be the result of using this algorithm. In case such a behavior is desired, an additional constraint could be added besides the requirement on the value of the Jacobian determinant. Then, if this thickness is lower than the minimum required thickness, the deformation of the control points can be decreased using the process of Figure 5.2.

5.2 Grid intersection prevention

After the completion of the FFD intersection prevention procedure, the grid deformation using linear elasticity equations is carried out. In case of large deformations this can lead to nonphysical deformations in the model. Therefore, another procedure to prevent this has been implemented that uses the same approach as the prevention of self-intersections within the FFD surface. First the nonphysical points are detected based on an assessment of the mesh quality and after that a procedure is carried out to prevent this from occurring in the deformed geometry.

Mesh quality assessment

The assessment of the quality of all mesh elements can be done in a variety of ways, for all different types of elements that can be present in a mesh. These mesh quality metrics all differ from each other on multiple aspects: the information they can provide to the modeler, the amount of input they require and the usability for different types of elements. Examples of methods used are based on the condition number of the Jacobian matrix of the elements and comparing it to a certain 'ideal' element of that type [30, 31] or by assessing whether the elements are valid with respect to the characteristics of the respective type [32]. However, there is no set of quality metrics available that directly links the quality of the mesh to the quality of the achieved simulation results since the simulation results can only be retrieved after the generation of the mesh.

In the methods that compare all elements to an ideal element, the definition of this ideal element is required. Furthermore, the evaluation of the extent to which a certain element corresponds to this ideal element varies between different methods and can be a measure of shape, size or a combination of both. These factors can make it difficult to evaluate all elements in a similar manner, especially when there are multiple element types present in one mesh.

The method of evaluating elements based on the validity, also referred to as convexity, of their shape can be seen as a more general way of looking at the elements. Through use of this method, all elements can be evaluated in a similar manner based on the ordering of the element vertices in the software implementation, which follows a standard format [33, 34]. This is the method used for assessing the element quality in this project.

In Figure 5.5, the ordering of points for triangular and quadrilateral elements in the current implementation can be seen. Furthermore, angle θ is shown, which represents the angle between edges $0 \rightarrow 1$ and $0 \rightarrow 2$ for the triangle and between edges $0 \rightarrow 1$ and $0 \rightarrow 3$ for the quadrilateral.

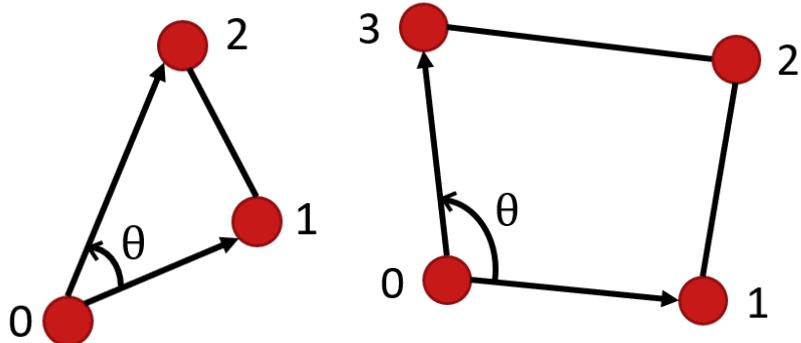


Figure 5.5: Valid triangular and quadrilateral element

In this method, angle θ is calculated for every set of adjacent edges in an element using the cross product as shown in Equation 5.4. The element is nonconvex if angle θ is larger than 180° [35]. Specifically, this means that the value of $\sin(\theta)$ has to be negative. However, since the norm of a vector is always larger than 0, the convexity of the element can be evaluated based on the value of the entire cross product. This can only be negative if $\theta > 180^\circ$.

$$\vec{a} \times \vec{b} = \| \vec{a} \| \| \vec{b} \| \sin(\theta) \quad (5.4)$$

Although in general the angle between two edges in a triangular element can never be larger than 180° , the ordering of the nodes in every elements enables the evaluation of the orientation of the element. Since in the cross product the ordering of vectors \vec{a} and \vec{b} is of importance in the calculated angle, the resulting angle calculated in an inverted element is $360^\circ - \theta$ and therefore larger than 180° . An example of this is provided in Figure 5.6, where the triangular element of Figure 5.5 is inverted and an invalid quadrilateral element are displayed.

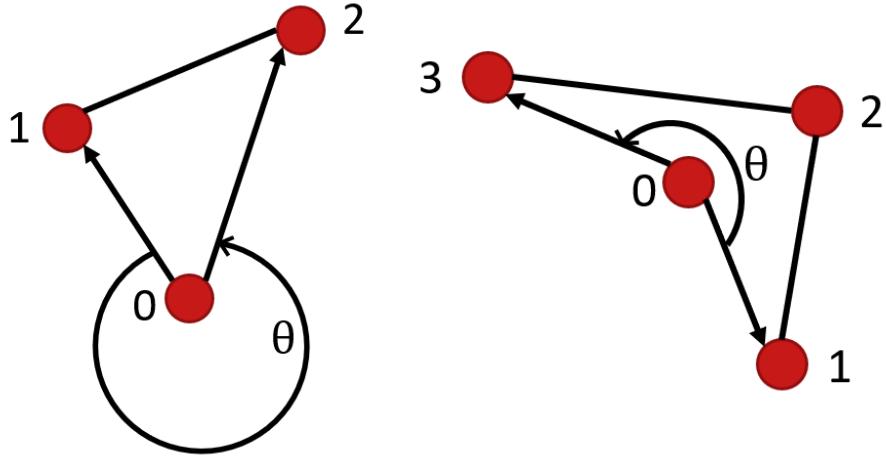


Figure 5.6: Invalid triangular and quadrilateral element

This evaluation on the validity of the elements is done for all elements. After this is completed, the amount of nonvalid elements is presented as output for the modeler. This can then be used to evaluate the general quality of the model with respect to the required deformation: when there is a large amount of nonvalid elements, the initial mesh is further from the desired geometry than when there are only a few nonvalid elements.

Prevention of nonvalid grid elements

The procedure implemented to prevent nonvalid elements from occurring in the mesh after deformation is done in a recursive procedure similar to that used in the prevention of self-intersections within the FFD deformation. This procedure is only started if one or more nonvalid elements are present in the mesh and the magnitude of the deformations is adjusted as presented in Figure 5.3. In Figure 5.7, the implemented procedure can be seen.

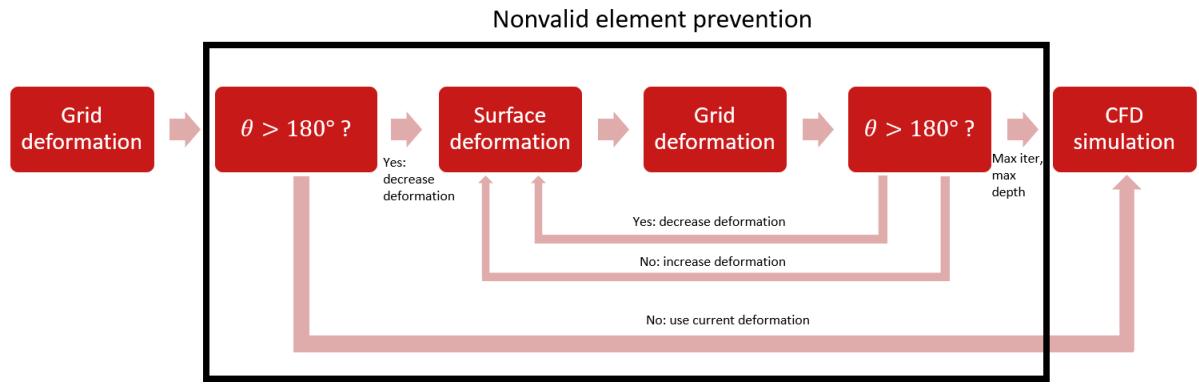


Figure 5.7: Deformation procedure after addition of nonvalid element prevention

When there are nonvalid elements after the first deformation, a new FFD-based surface deformation is done using a lower deformation magnitude of the control points, in order to ensure a geometric deformation that is similar to the desired deformation magnitude.

With the process implemented to prevent invalid elements from occurring in the mesh after deformation described, a valid deformation is ensured through the entire deformation process. In Figure 5.8, the implementation of the procedures as described in this section are shown within the entire process of adjoint-based optimization.

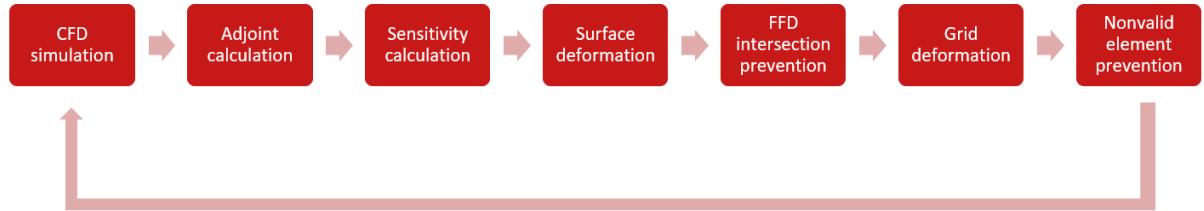


Figure 5.8: Improved adjoint-based optimization procedure

6 Parameter study

The results of the unconstrained optimization as described in Section 4 show that the deformation of the burner tube only occurs at the tip of the burner. Furthermore, the magnitude of the deformations required to achieve the optimal burner dimensions is too large for the straight burner tube and nonphysical elements are generated in the deformed mesh.

In this section, a parameter study is presented in which a better shape of the burner geometry is researched to be used as the initial design. Through this parameter study, the objective function is decreased in comparison to using a straight tube. Since this can be seen as a partial optimization, the required deformation magnitude is decreased to achieve the fully optimal geometry with respect to the value of the objective function as described in Section 4.

6.1 Model setup

From the unconstrained burner optimization it became clear that the desired deformation of the tube consisted of a decrease in radius over a small length around the burner tip. These deformations can be used for a simplification of the burner model. For this model, three different geometries are compared in order to find a simple geometry that provides accurate results.

Burner outflow comparison

Figure 6.1 shows the three different models that can be used to represent the burner to be used in this parameter study. In this comparison, the focus is on the velocity profile at the outflow surface of the burner tube, as indicated with a gray dotted line.



Figure 6.1: Burner geometries compared for use in parameter study

In this comparison, the three models represent the same burner tip with three different types of modeled outlet: the end of the model being the outlet of the burner tube, the wall thickness of the model simulated as a part of the model and an intermediate model, where a section is included after the burner outflow surface that is continued using the burner outflow radius. Figure 6.2 shows the velocity profile of the three different geometries on the outflow radius numbered from left to right.

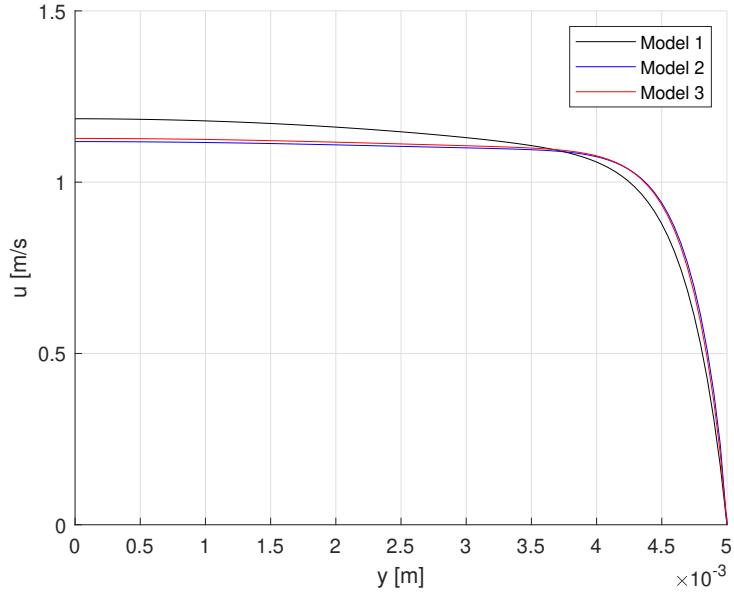


Figure 6.2: Velocity results of three different burner geometries

An extra section added to the domain downstream of the burner tube is required to provide reliable results with respect to the outflow velocity since the results differ visibly between the simplest model 1 and the most realistic model 3. The results of the simulation with and without an added wall thickness are very similar. Since the extra elements added in the model to allow the wall thickness to be modeled require more computational resources without providing additional accuracy to the flow results, the intermediate model is considered the most suitable for use in the parameter study.

6.2 Boundary conditions

The burner model is shown in Figure 6.3. In this model, two parameters are assumed to be of importance in the quality of the result with respect to the objective function as presented in Section 4.2: burner outflow radius R and length L over which the transition between the larger and smaller radius is made. These two parameters are varied in order to find a more suitable geometry to be used in future burner simulations.

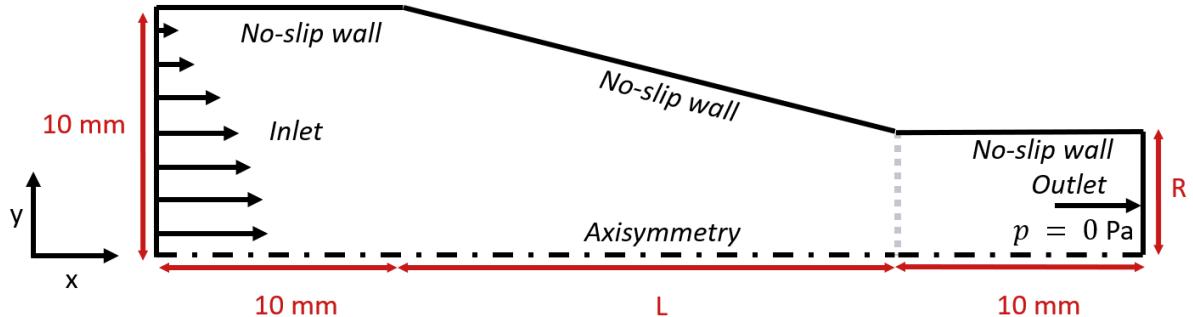


Figure 6.3: Model used for parameter study

The burner model as presented can be characterized by three separate sections. At the inlet of this model, a straight part of 10 mm in length is modeled which represents the burner tube. This section is included in the model to provide a good basis for the flow to develop. A difference between this burner model and the one used in the unconstrained optimization is that the inlet radius is taken to be 10 mm instead of the 6 mm as used previously. Although this is partly done as a consequence of the increasing radius around the inlet in the unconstrained optimization, this increase also provides a better basis for the physical relevance of the problem. If this inlet radius would have been kept constant at 6 mm, the decrease in outflow radius would require the outlet radius of the burner to be extremely small. Therefore, an outflow radius of 10 mm is selected to let the results be in better accordance with realistic situations.

The second section, of length L , represents the tip of the burner. This part is tapered to represent the desired deformations as resulting from Section 4. At the end of this section a dotted gray line is displayed, which represents the burner outflow surface. Along this line, the value of the objective function will be evaluated. The third section in the model, which is 10 mm in length and has radius R , represents the area downstream of the burner tube.

The boundary conditions in the model used for this parameter study are similar to those used in the unconstrained optimization procedure in order to provide results that can be linked directly to those simulations. Specifically, an axisymmetric boundary condition is imposed on the axis of symmetry of the burner and the walls are subject to a no-slip condition. Furthermore, an ambient outlet pressure and an inlet with an imposed Poiseuille profile are defined as the boundary conditions for this problem.

The inlet velocity is scaled to provide similar results at the burner outflow surface. This is done using a constant Reynolds number for all simulations. The baseline for this scaling is set for an outlet radius of 5 mm and an average velocity of 1 m/s. With a Reynolds number of 658 for an air flow in ambient conditions, this is well below the transitional Reynolds number of approximately 2000:

$$\text{Re} = \frac{\rho v D}{\mu} = \frac{1.204 \cdot 1 \cdot 0.01}{1.825 \cdot 10^{-5}} \approx 658$$

6.3 Simulation results

The burner tube as shown in Figure 6.3 has been used in simulations for parameters L and R with values between respectively 1 – 40 mm and 1 – 10 mm. In this section only the most important observations resulting from these simulations will be discussed based on a selection of the results.

Figure 6.4 shows the value of the objective function as a function of burner outflow radius R for four different values L . The first observation that can be made is that the value of the objective function decreases for both a decreasing transition length L and outflow radius R . Furthermore, the rate at which the value of the objective function decreases is lower for the geometries with smaller lengths L . Below $R = 5$ mm, the objective function values for transition lengths of 1 and 5 mm are almost constant.

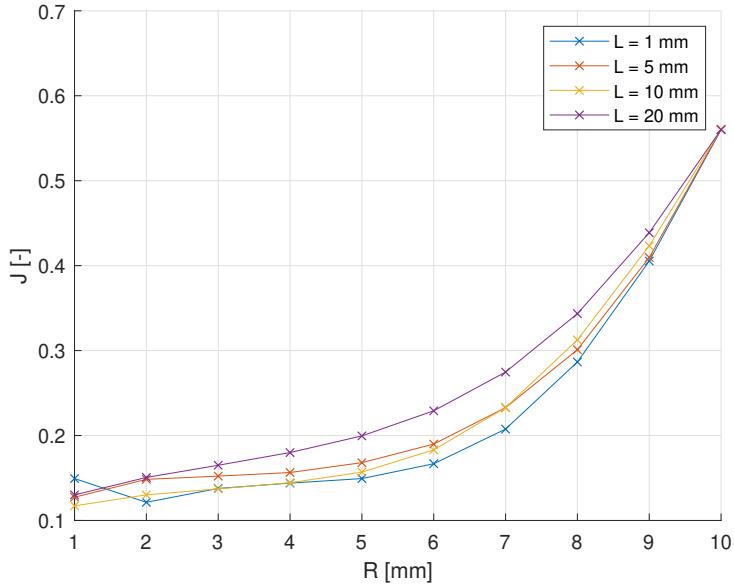


Figure 6.4: Objective function values for $L = 1, 5, 10, 20$ mm

In Figure 6.5 three of the geometries resulting to Figure 6.4 are shown for an outflow radius of $R = 5$ mm. Since the results using $L = 10$ mm show similar results to $L = 5$ mm, this model is dropped from the comparison.

The slope of the lines in Figure 6.4 can be explained by looking at the steepness of the tapered sections of the three geometries. The transition for $L = 20$ mm is fairly smooth, while the transition for a length of 1 mm almost resembles a step profile. This smoothness of the wall allows the flow to develop a little more in the direction of a parabolic profile, which in turn increases the value of the objective function.

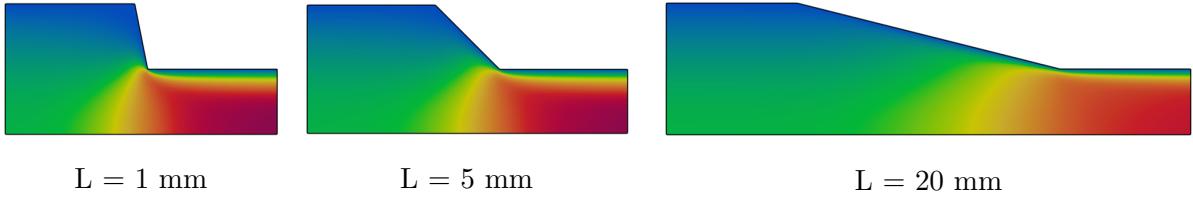


Figure 6.5: Burner tube geometries for $R = 5$ mm

The objective function values for $R = 5$ mm differs considerably for the three different lines shown in Figure 6.4. However, even for the geometry with $L = 20$ mm the objective function value has decreased by approximately 65% with respect to the initial situation. This decrease in objective function value is represented in Figure 6.6a, where the velocity profiles of the three curves as shown earlier can be seen. Although the objective function values differ, all three profiles are relatively uniform from the axis of symmetry until approximately 1 mm from the wall and only minimal differences occur between the curves.

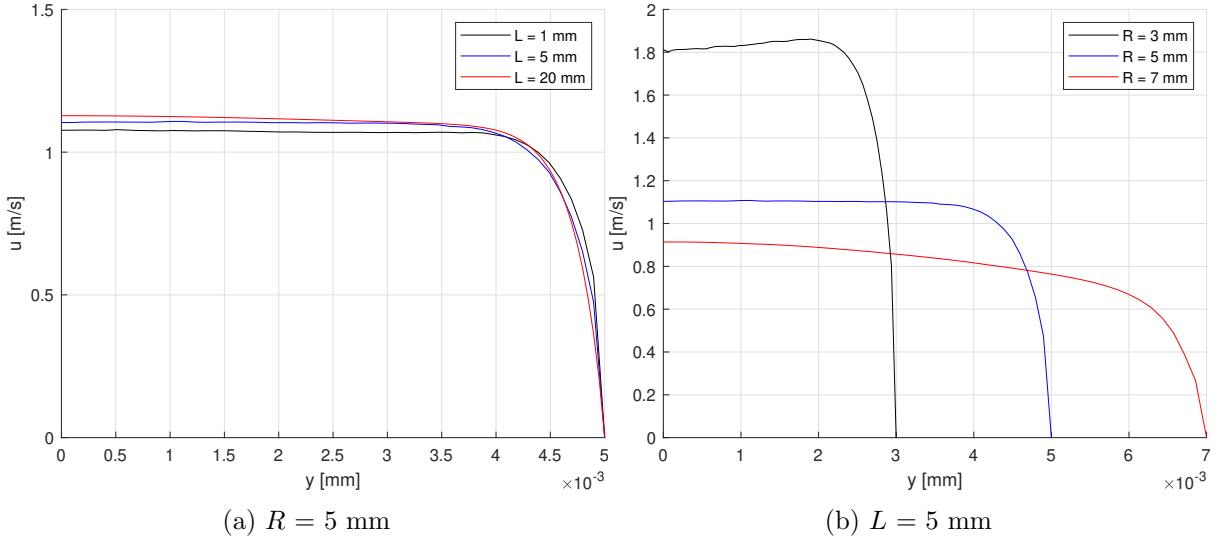


Figure 6.6: Outflow velocity profiles for $R = 5 \text{ mm}$ and $L = 5 \text{ mm}$

The decrease in the slope of the objective function values for outflow radii below 5 mm for the simulations with a small length L can be explained through the lines presented in Figure 6.6b. Here, the velocity profile of $R = 7 \text{ mm}$ is increasing over the entire burner radius. If the outflow radius is made smaller, the profile of $R = 5 \text{ mm}$ results. In this profile, the velocity rises quickly close to the wall and is uniform over the remaining part of the outflow radius. Then, if the outflow radius is made even smaller, the slope becomes too steep and the flow cannot develop enough to retrieve a uniform profile. This is shown in the peak in the velocity for $R = 3 \text{ mm}$. Although this peak occurs at a shorter distance from the wall than the maximum velocity for $R = 5 \text{ mm}$, the peak rises higher above the average velocity and therefore leads to a positive contribution to the value of the objective function. In Figure 6.7 this is indicated through a display of the normalized velocity as a function of the normalized radial coordinate.

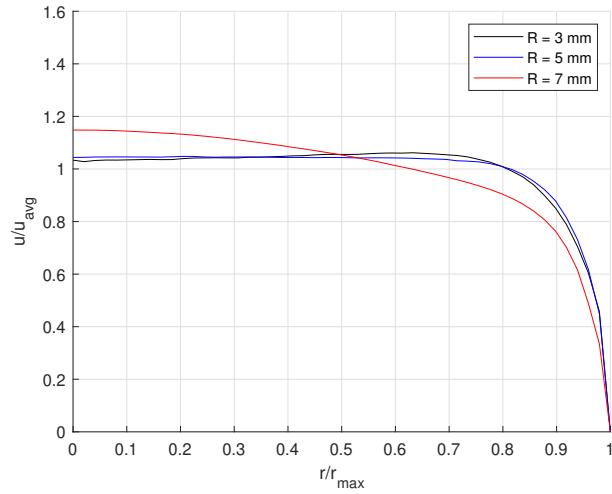


Figure 6.7: Normalized velocity profiles for $L = 5 \text{ mm}$

In this study, it is shown that the value of transition length L does not have a large influence on the value of the objective function. Furthermore, although the outflow radius could in principle be continuously decreased to lower the value of the objective function, the influence is small in case length L is sufficiently small. Based on these considerations, a transition length $L = 5$ mm and a burner outflow radius of $R = 5$ mm is seen as the most suitable geometry for further research. This geometry provides both a low value of the objective function and an outflow radius that can be physically realized. Furthermore, there is some room left for further improvement in an adjoint-based optimization procedure.

7 Improved optimization procedure

The results from Section 5 have shown that nonvalid elements can be prevented from occurring in the deformed mesh. After that, Section 6 has provided a more suitable model than the straight tube to be used for an additional adjoint-based optimization procedure. This reduces the magnitude of deformations required to achieve the fully optimized solution.

In this section, the improved initial model will be used in combination with the invalid grid prevention algorithms to ensure valid and deformation results. This will first be done on a model without combustion, similar to the optimization procedure as presented in Section 4. As a further improvement to the evaluation, conditions are imposed on the burner outflow surface that ensure its straightness after deformation.

After the optimization procedure without combustion, the results are used in a simulation on a model with combustion enabled, which will be compared to the results of the initial CFD simulation as presented in Section 3.

7.1 Model description

From the results of Section 6, it is concluded that the length of the tapered section does not have a large effect on the outflow velocity profile. Furthermore, it has been shown that the objective function value did not decrease much for an outflow radius smaller than approximately half the size of the inflow radius.

In this section, the results of the improved optimization procedure will be compared to those of the initial CFD simulation of Section 3. Therefore, the conditions on the burner outflow surface are taken similar to those of the CFD simulation to allow for a direct comparison. This has led to a burner outflow radius of 6 mm and a wall thickness of 2 mm. The inflow radius is selected to be 12 mm, which should give some room for additional optimization based on the results of the parameter study.

In order to allow for a smooth transition between the large and small radius, the transition length has been set at 6 mm. In Figure 7.1 the schematic representation of this model is shown. Here, the burner outflow surface is represented by a gray dotted line.

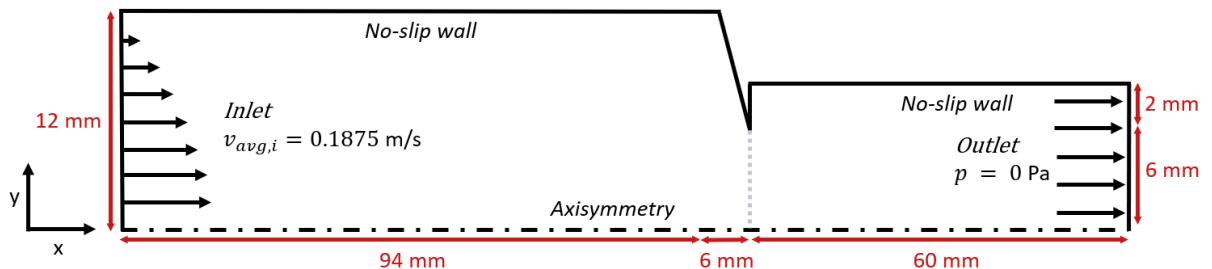


Figure 7.1: Burner model setup

The boundary conditions for this simulation are similar to those in the initial optimization procedure as presented in Section 4 with the no-slip walls, the axisymmetry condition on the axis of symmetry, the ambient outlet conditions and a Poiseuille profile applied at the inlet to aid in the convergence of the model. The only change is that the average inflow velocity is now 0.1875 m/s, a quarter of that of the initial burner model. Due to doubling the inlet radius, the inlet velocity has to be taken four times as low in order to provide the same conditions on the burner outflow surface. The mesh size used for this model is taken similar to that in the optimization of Section 4.

FFD box definition

The dimensions of the FFD box have been chosen based on the results of the simulations as described in the previous sections. Specifically, this means that the points on the burner outflow surface are now fixed. This avoids the warping of the outflow surface as shown in Section 4. Now, through deformations of the tapered section and a part of the burner tube upstream of the tapered section, the optimized geometry can be achieved. In Figure 7.2 the FFD box used for this deformation is shown.



Figure 7.2: FFD box definition

The right edge of the FFD box is defined on the outflow surface of the burner. As a result of this choice in combination with fixing the two points that define this edge, all points on the burner outflow surface will remain in the same location regardless of the deformation of all other control points.

The width of the FFD box is taken as 12 mm. As a result, the tapered section with a length of 6 mm and an additional 6 mm upstream of the burner wall can be deformed. Using an FFD box of 24x2 control points, a spacing between the points of 0.5 mm results. Furthermore, the height of the FFD box is set at 12 mm, the same height as the inflow radius. This way, all points on the burner wall that are inside the FFD box can be parameterized.

7.2 Simulation results

In Figure 7.3 the velocity results for the optimization as described above are presented. The deformed results are displayed in the top half of the figure, whereas the original model is shown in the bottom half. Although the surface is definitely deformed after the optimization, the velocity results do not show obvious differences. This is also represented in Figure 7.3c, where the initial results are shown in the left half of the figure and the results after optimization in the right half. Although the deformation has made the profile a little better in the tapered section, this effect is not visible at the outflow surface.

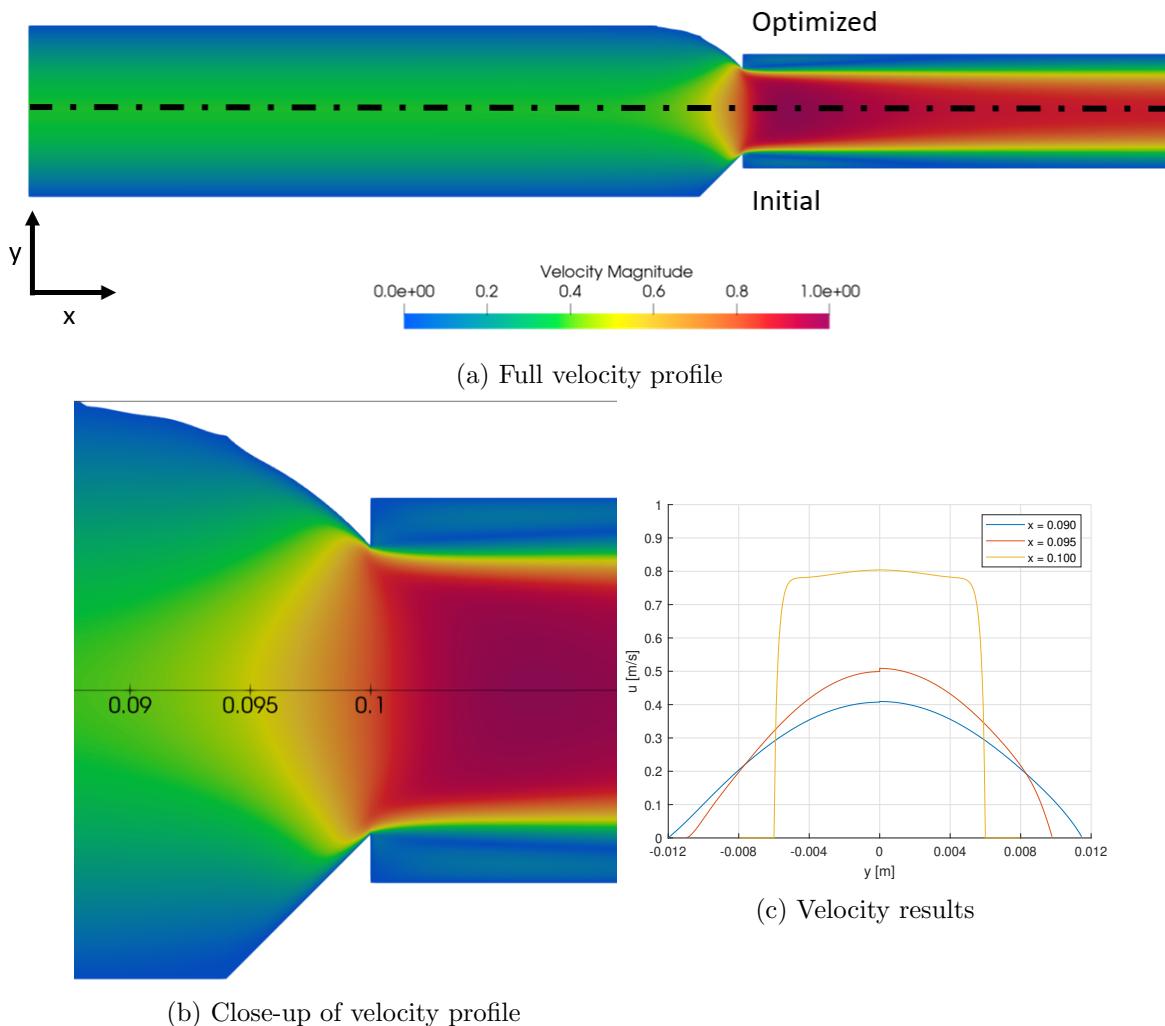


Figure 7.3: Comparison between optimized burner (top) and initial model (bottom)

Figure 7.4 shows a close-up of the deformed section of the burner surface for the initial situation and after 10, 30 and 55 iterations. Between these iterations, the edge of the slope is gradually becoming more rounded. This allows for a smoother transition in the flow velocity.

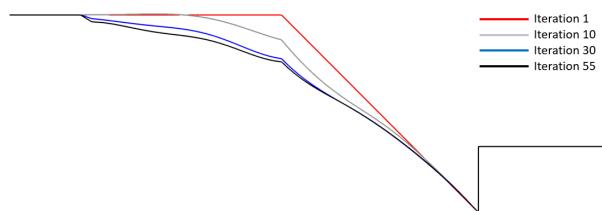


Figure 7.4: Burner wall for iteration 1, 10, 30 and 55

Although the surface has deformed noticeably, this large effect is not represented in the value of the objective function as shown for 55 iterations in Figure 7.5a. Another thing that can be noted is the small discontinuity in the deformed surface. This originates from the edge of the initial model where the sloped section starts. While the FFD smoothens the surface, the large discontinuity in the initial model has not fully disappeared in the deformed result. For this model it is assumed that this discontinuity does not considerably influence the results. However, if this would be considered a problem, the discontinuity could be removed by refining the mesh in this area and increasing the number of control points. This increases the amount of control over the deformation, which allows the surface to be deformed more smoothly.

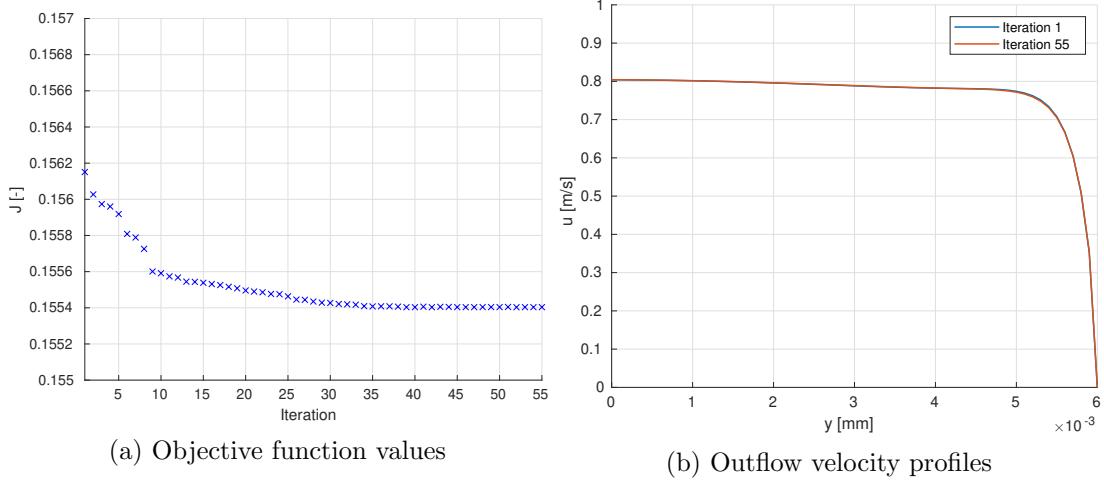


Figure 7.5: Results of optimization procedure

The value of the objective function does not decrease much during the 55 iterations, with a decrease in the objective function value of 0.5%. In Figure 7.5b the velocity profile on the burner outflow surface is presented. In accordance with the objective function results there is practically no difference between the results of the first iteration and after 55 iterations. However, the results as obtained using this model are not in accordance with those of the parameter study. The outflow velocity profile keeps rising from the wall to the axis of symmetry, which implies that a further improvement can be made by either decreasing the outflow surface or increasing the upstream radius. Three other simulations are done in order to investigate whether another shape of the FFD box, an increased amount of control points or another inlet radius would lead to improved results. The FFD boxes for these simulations are shown in Figure 7.6.



Figure 7.6: Other FFD boxes used in optimization procedure

In the FFD box in Figure 7.6a, the size of the burner radius upstream of the tapered edge is kept constant and the burner outflow surface is allowed to be deformed. In this representation, this is done using a grid of 12x2 and 24x2 control points for respectively model 1 and 2. Model 3 represents the same model as used initially in this section. Instead of an inlet radius of 12 mm, an inlet radius of 11 mm is taken and the inlet velocity is scaled accordingly to provide equal burner outflow conditions. This is expected to have a less uniform flow profile, which leaves more room for improvement. The results of optimizations using these FFD boxes are shown in Figure 7.7.

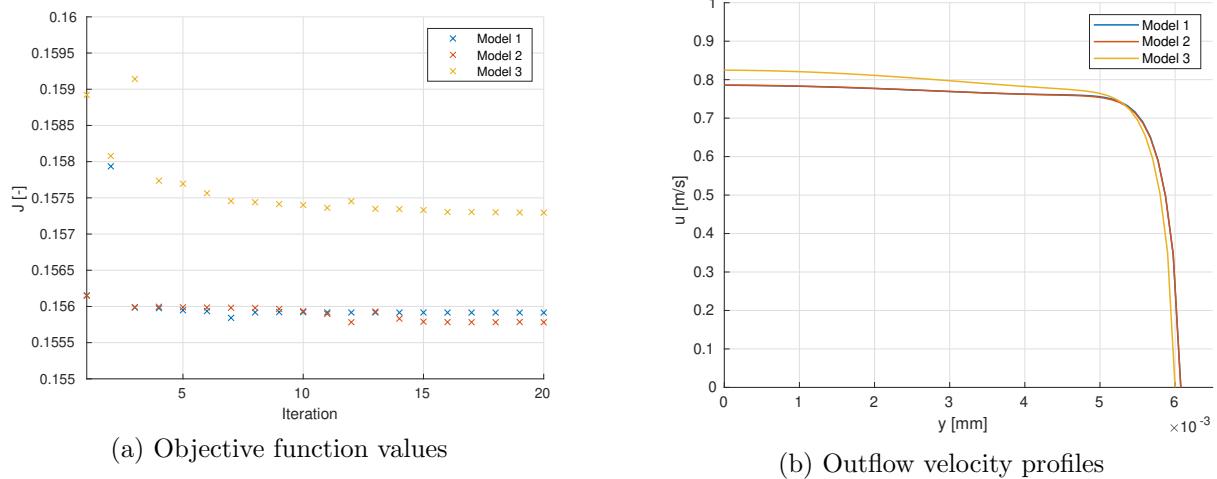


Figure 7.7: Results of optimization procedures using different FFD boxes

The results from all three models show a change in objective function value similar to the model as used initially. As expected, the model with an inlet radius of 11 mm performs worse than the models with an inlet radius of 12 mm and does not lead to better results after optimization.

Prevention of nonvalid deformations

Because of the improvement in the initial shape and the corresponding decrease in the magnitude of the deformations, the deformations were small enough that during this procedure no nonvalid elements occurred in the mesh after deformation. As a test, model 1 from figure 7.6 is deformed using a larger deformation magnitude. An added advantage of this large deformation is that this procedure can be used to see whether a further improvement in the objective function value is achieved. In Figure 7.8 the initial deformed mesh and the mesh after the recursive lowering of the control point deformations are displayed around the tip of the burner.

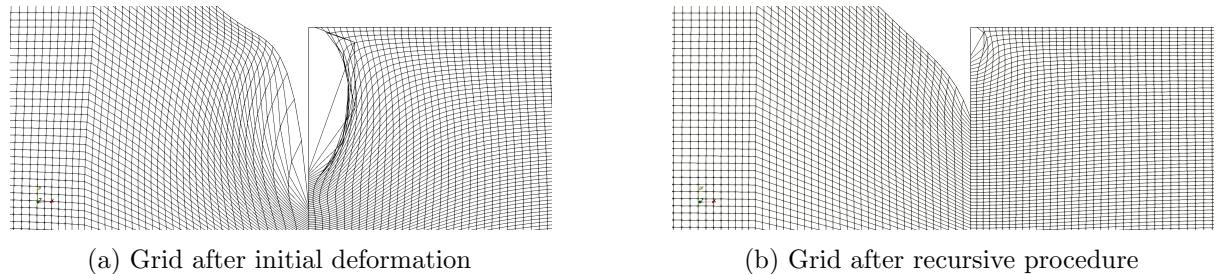


Figure 7.8: Deformed grid before and after prevention of nonvalid elements

In this deformation the points on the burner outflow surface and top wall are moved towards the axis of symmetry. This decreases the area of the burner outflow surface, as would be expected from the results of the parameter study. However, since the top node of the burner wall is kept fixed to keep the outer boundary of the model straight, the top edge of the burner wall is stretched. This large stretch gives rise to nonvalid elements occurring in this area.

Through the recursive lowering of the deformation magnitude, the nonvalid points are removed from the grid. After this deformation, the simulation continues and similar results to those shown earlier in this section result. It can be concluded that the mesh quality has improved as a result of this procedure. This means that the implementation of this procedure has made the automatic optimization procedure more robust.

The multiple optimization procedures have not led to a considerable improvement in the objective function value, while it should be still possible to achieve a more uniform velocity profile. A possible solution to this is to gain more control over the surface through a further refinement of the grid and using more control points in the FFD box. However, this is left as an opportunity for further research.

Combustion results

As a consequence of the lack of improvement in the burner outflow velocity profile in the adjoint-based optimization, a new simulation involving combustion has been done on the model resulting from the parameter study. This has the advantage that a good mesh quality can be ensured since no extra deformation is done after construction of the mesh.

As the flow does not change in most of the burner tube, only the tapered section of 6 mm and a straight section of 6 mm upstream of this section is modeled for this problem. Downstream of the burner outflow area the same mesh as in the CFD simulation of Section 3 is used, upstream an element size of 0.04 mm is used along the entire surface parallel to the axis of symmetry. Due to the structured nature of the mesh, the element size in the direction tangential to the axis of symmetry increases gradually to 0.08 mm. In Figure 7.9 the result of this simulation is shown in the top half of the figure, with the results from Section 3 in the bottom half.

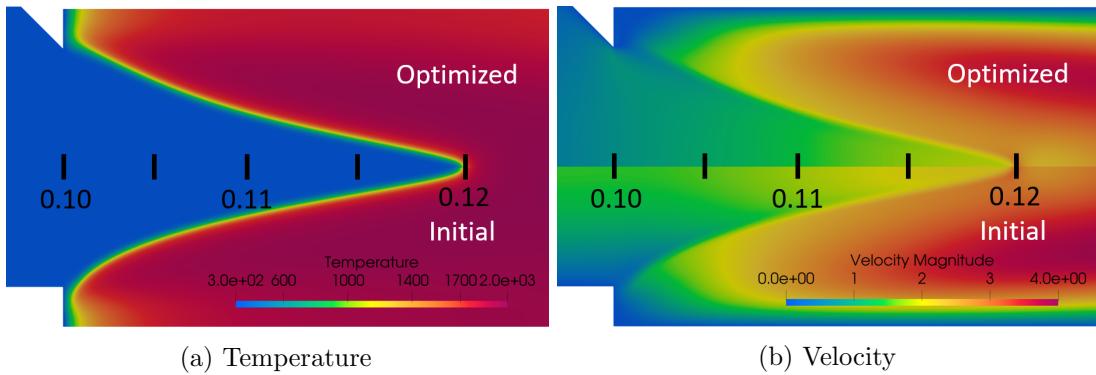


Figure 7.9: Comparison of results using combustion: improved (top) and initial (bottom)

The temperature plot shows that the straightness of the flame surface has increased, especially close to the burner outflow surface. The reason for this is illustrated through the velocity results, where the velocity profile of the optimized results is uniform on the entire outflow surface and shows a large difference in the initial results. In Figure 7.10 the velocity results are plotted for the five locations indicated in Figure 7.9, where the y coordinate is negative for the initial results and positive for the optimized results.

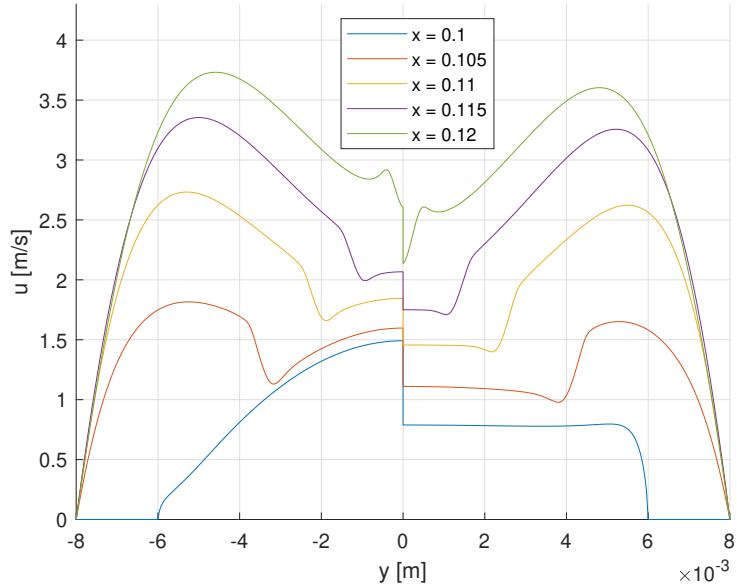


Figure 7.10: Velocity tangential to axis of symmetry: initial (left) and optimized (right)

These results show that the velocity uniformity on the burner outflow surface has an effect on the velocity profile in the entire unburnt area. Whereas the parabolic profile of the initial burner tube leads to a curved profile in the unburnt area downstream, the uniform profile in the optimized model stays preserved.

Another thing that can be noted is that the profile at the burner outflow surface is more uniform than in the flow without combustion. This might point to an inaccuracy in the results of one of the simulations or the effect the added combustion has on the outflow velocity. However, the exact reason for this difference is left for further research into this subject.

The objective function for this simulation has a value of 0.143, whereas the initial value as calculated in the initial CFD simulation was 0.538. This corresponds to an improvement of 73%. Furthermore, by far the largest contribution to this value is due to the influence of the wall, which cannot be completely removed. Therefore, it can be said that a geometry close to the optimal shape has been designed with respect to uniformity of the velocity on the burner outflow surface.

8 Conclusion

In this research, the open-source shape optimization software SU2 is used to improve the uniformity of the velocity profile at the outflow surface of a Bunsen-type burner using an adjoint-based optimization procedure. The Flamelet Generated Manifold combustion model as implemented in the software is validated on this model in an initial CFD simulation. Furthermore, it is found that the outflow velocity of the burner is similar to that in a burner tube without combustion, which has a Poiseuille profile. Therefore, optimization is done using a cold flow in similar conditions to allow for a more efficient optimization process.

Through an initial adjoint-based optimization procedure it is found that while the burner geometry can be improved 55% using the straight tube of the Bunsen burner, the required deformations in combination with the characteristics of the deformation software can lead to invalid elements occurring in the mesh after deformation. This can lead to either divergence of the results or convergence to a state that cannot be physically realized.

The convergence issues resulting from the deformation are avoided through use of two methods: improving the initial geometry in a parameter study that uses the information gained from the initial deformation and the implementation of an iterative procedure in which the validity of the deformed mesh is ensured within both the FFD surface deformation and the deformation of the internal elements. In this procedure the deformation magnitude of the control points is adjusted up to the point just under that in which the nonvalid deformations will result, which is shown to provide valid deformations.

The parameter study has shown that the objective function value is decreased if the tip of the burner is tapered towards a smaller outflow radius. It is found that for outflow radii lower than half of the inlet radius the objective function does not considerably decrease further and that lowering the length of the transition does not have a large effect, as long as it is sufficiently small.

In an additional adjoint-based optimization procedure using a model with an outflow radius that is half of the inlet radius and an equal length of the transition between the large and small radius no significant improvement is achieved, while this should still be possible. Through gaining more control of the deformations by refining the mesh and adding more control points, the fully optimized model might be obtained in further research.

In a comparison between the original CFD simulation of the burner tube and the optimized results a clear difference is observed in the straightness of the flame resulting due to a more uniform velocity profile at the outlet. The value of the objective function has decreased with 73% after the optimization, with the greatest remaining part resulting from the behavior near the wall. Since the velocity will always be zero at the wall, it can be concluded that the achieved solution approaches the optimal design for the burner tube.

Future research

The algorithms added that ensure the validity of the mesh after both the surface deformation using FFD and the deformation using the linear elasticity equations are shown to work well in combination with the adjoint-based optimization procedure. However, the iterative procedure converges to the point where the geometry is at the edge of becoming invalid. This can lead to walls that have zero thickness, which could be seen as an undesirable result. By imposing a constraint on the minimum wall thickness in the surface deformation algorithm, the desired thickness can be achieved.

In the algorithm that prevents nonvalid elements from occurring in the deformed mesh, the elements are deformed in a similar way, up to the point where the element of the worst quality is on the edge of becoming nonvalid. If the desired movement is then in the direction that would make this element nonvalid, no further deformations can be achieved. In further a procedure could be implemented that improves the quality of the mesh after deformation, through which further deformations can be achieved in the desired direction.

While the optimization on the burner tube after the parameter study has shown deformations, only a small decrease in the value of the objective function is achieved while a further improvement could be made. In this research no definite answer is given as to why this deformation did not result in the fully optimized geometry. Through further refinement of the mesh and using an FFD box with more control points, a greater level of control can be gained on the geometry. This might give the fully optimized results and a definite answer to the question what the optimal shape is for uniformity of the velocity profile on the outflow surface of a Bunsen burner.

References

- [1] Pironneau, O. (1973). On optimum profiles in Stokes flow. *Journal of Fluid Mechanics* 59(1), pp. 117-128. DOI: 10.1017/S002211207300145X
- [2] Jameson, A. (1988). Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3), 233–260. DOI: 10.1007/BF01061285
- [3] Arens, K., Rentrop, P., Stoll, S.O., Wever, U. (2005). An adjoint approach to optimal design of turbine blades. *Applied Numerical Mathematics* 53, pp. 93-105. DOI: 10.1016/j.apnum.2004.11.003
- [4] Sederberg, T. W., Parry, S. R., (1986). Free-Form Deformation of Solid Geometric Models. *Computer Graphics* 20(4), pp. 151-160. DOI: 10.1145/15886.15903
- [5] Economou, T.D., Palacios, F., Copeland, S.R., Lukaczyk, T.W., Alonso, J.J. (2016) SU2: An Open-Source Suite for Multiphysics Simulation and Design. *AIAA Journal*, 54(3). DOI: 10.2514/1J053813
- [6] SU2 foundation board (2020). Welcome & SU2 Year in Review. 1st Annual SU2 Conference. <https://su2foundation.org/wp-content/uploads/2020/06/economou.pdf>.
- [7] Generalic, E. "Bunsen burner." Croatian-English Chemistry Dictionary & Glossary. 20 Oct. 2018. KTF-Split. 22 Oct. 2020. [<https://glossary.periodni.com>].
- [8] Hu, S., Gao, J., Gong, C., Zhou, Y., Bai, X.S., Li, Z.S., Alden, M. (2018). Assessment of uncertainties of laminar flame speed of premixed flames as determined using a Bunsen burner at varying pressures. *Applied Energy* 227, pp. 149-158. DOI: 10.1016/j.apenergy.2017.09.083
- [9] Mazas, A.N., Fiorina, B., Lacoste, D.A., Schuller, T. (2011). Effects of water vapor addition on the laminar burning velocity of oxygen-enriched methane flames. *Combustion and Flame* 158, pp. 2428-2440. DOI: 10.1016/j.combustflame.2011.05.014
- [10] Mache, H., Hebra, A. (1941). Zur Messung der Verbrennungsgeschwindigkeit explosiver Gasgemische. *Akademie der Wissenschaften in Wien, Sitzungsberichte* vol. 150, pp. 157-174.
- [11] Economou, T. (2018) Simulation and Adjoint-based Design for Variable Density Incompressible Flows with Heat Transfer, pp. 1-7. Presented at AIAA AVIATION Forum, Atlanta, Georgia. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2018-3111
- [12] Oijen, van, J.A., Lammers, F.A., Goey, de, L.P.H. (2001). Modeling of Complex Premixed Burner Systems by Using Flamelet-Generated Manifolds. *Combustion and Flame* 127(3), pp. 2124-2134. DOI: 10.1016/S0010-2180(01)00316-9
- [13] Oijen, van, J.A., Donini, A., Bastiaans, R.J.M., Thije Boonkkamp, ten, J.H.M., Goey, de, L.P.H. (2016). State-of-the-art in premixed combustion modeling using flamelet generated manifolds. *Progress in Energy and Combustion Science* 57, pp. 30-74. DOI: 10.1016/j.pecs.2016.07.001
- [14] Oijen, van, J. A. (2002). Flamelet-generated manifolds : development and application to premixed laminar flames, pp. 3-30. Eindhoven: Technische Universiteit Eindhoven. DOI: 10.6100/IR557848
- [15] CHEM1D: A One-Dimensional Laminar Flame Code. Eindhoven University of Technology. <https://www.swmath.org/software/22208>. Accessed 22 September 2020

- [16] Soto, O., Löhner, R., Yang, C. (2004). An Adjoint-Based Design Methodology for CFD Optimization Problems. *International Journal of Numerical Methods for Heat and Fluid Flow* 16(4), pp. 734-759. DOI: 10.1108/09615530410544292
- [17] Nadarajah, S.K., Jameson, A. (1999). A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *38th Aerospace Sciences Meeting and Exhibit*. DOI: 10.2514/6.2000-667
- [18] Hu, G., Kozlowski, T. (2020) Assessment of continuous and discrete adjoint method for sensitivity analysis in two-phase flow simulations. *Annals of Nuclear Energy*, 147. DOI: 10.1016/j.anucene.2020.107731
- [19] Kenway, G.K.W., Mader, C.A., He, P., Martins, J.R.R.A. (2019). Efficient Adjoint Approaches for Computational Fluid Dynamics, *Progress in Aerospace Science* 110. DOI: 10.1016/j.paerosci.2019.05.002
- [20] Giles, M.B., Pierce, N.A. (2000). An Introduction to the Adjoint Approach to Design. *Flow, Turbulence and Combustion* 65, pp. 393-415. DOI: 10.1023/A:1011430410075
- [21] Zhou, B.Y., Albring, T., Gauger, R., Ilario da Silva, C.R., Economou, T.D., Alonso, J.J. (2016). An efficient Unsteady Aerodynamic and Aeroacoustic Design Framework using Discrete Adjoint. *The 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. DOI: 10.2514/6.2016-3369
- [22] Homescu, C. (2011). Adjoints and automatic (algorithmic) differentiation in computational finance. arXiv:1107.1831v1
- [23] Sagebaum, M., Albring, T., Gauger, R. (2019). High-Performance Derivative Computations using CoDiPack. *ACM Transactions on Mathematical Software (TOMS)*, 45 (4). DOI: 10.1145/3356900
- [24] Dwight, R. (2009). Robust Mesh Deformation using the Linear Elasticity Equations. DOI: 10.1007/978-3-540-92779-2_62
- [25] Gain, J.E., Dodgson, N.A. (2001). Preventing Self-Intersection under Free-Form Deformation. *IEEE Transactions on Visualization and Computer Graphics* 7(4), pp. 289-298. DOI: 10.1109/2945.965344
- [26] Piegl, L., Tiller, W. (1997). *The NURBS Book*, pp. 9 - 25. Berlin: Springer-Verlag Berlin Heidelberg. DOI: 10.1007/978-3-642-59223-2
- [27] Sankaran, R., Hawkes, E.R., Chen, J.H., Lu, T., Law, C.K. (2007). Structure of a spatially developing turbulent lean methane-air Bunsen flame. *Proceedings of the Combustion Institute* 31, pp. 1291-1298. DOI: 10.1016/j.proci.2006.08.025
- [28] Zhao, X.Y., Zhu, C.G. (2015). Injectivity conditions of rational Bézier surfaces. *Computers & Graphics* 51, pp. 15-25. DOI: 10.1016/j.cag.2015.05.017
- [29] Lasser, D. (1989) Calculating the self-intersections of Bézier curves. *Computers in Industry* 12(3), pp. 259-268. DOI: 10.1016/0166-3615(89)90072-9
- [30] Knupp, P.M. (2003). Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design* 39, pp. 217-241. DOI: 10.1016/S0168-874X(02)00070-7
- [31] Munson, T. (2007). Mesh shape-quality optimization using the inverse mean-ratio metric.
- [32] Joe, B. (2008). Shape Measures for Quadrilaterals, Pyramids, Wedges, and Hexahedra.

- [33] Mesh File (n.d.). Retrieved from: https://su2code.github.io/docs_v7/Mesh-File/
- [34] VTK Users Guide (n.d.), pp. 480-481. Retrieved from: <https://www.kitware.com/products/books/VTKUsersGuide.pdf>
- [35] Alexandrov, A.D. (2005). Convex Polyhedra, pp. 7-26. Berlin: Springer-Verlag Berlin Heidelberg. DOI: 10.1007/b137434
- [36] Albring, T., Sagebaum, M., Gauger, N.R. (2016). Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2. 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. DOI: 10.2514/6.2016-3518
- [37] Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M. (2018). Automatic Differentiation in Machine Learning: a Survey. Journal of Machine Learning Research 18, pp. 1-43. DOI: 10.5555/3122009.3242010
- [38] Law, C. K. (2006). Combustion Physics, pp. 51-58. Cambridge University Press. <https://doi.org/10.1017/CBO9780511754517>
- [39] ANSYS FLUENT 12.0 Theory Guide - 7.1.2 The Generalized Finite-Rate Formulation for Reaction Modeling (n.d.), viewed 14 May 2020, <https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node129.htm>

A Governing equations

A.1 Preconditioning matrix construction

In Section 2 it has been described that the equations for conservation of mass, momentum and energy are transformed from using the variables $U = [\rho, \rho\bar{v}, \rho c_p T]^T$ to primitive variables $V = [\rho, \bar{v}, T]^T$ through use of a preconditioning matrix.

This preconditioning matrix is used to aid in the convergence at low Mach numbers by introducing a coupling between the pressure and velocity. In this section, the transformation is described in more detail, as elaborated by Economou [11].

Equation A.1 shows the incompressible Navier-Stokes equations as a function of variables U .

$$R(U) = \frac{\partial U}{\partial t} + \nabla \cdot \bar{F}^c(U) - \nabla \cdot \bar{F}^v(U, \nabla U) - Q = 0 \quad (\text{A.1})$$

In this equation, the convective and viscous fluxes and the stress tensor are respectively written as:

$$\bar{F}^c = \begin{Bmatrix} \rho\bar{v} \\ \rho\bar{v} \otimes \bar{v} + \bar{I}p \\ \rho c_p T \bar{v} \end{Bmatrix} \quad \bar{F}^v = \begin{Bmatrix} \cdot \\ \bar{\tau} \\ \kappa \nabla T \end{Bmatrix} \quad \bar{\tau} = \mu(\nabla \bar{v} + \nabla \bar{v}^T) - \mu \frac{2}{3} \bar{I}(\nabla \cdot \bar{v})$$

For the transformation from $R(U)$ to $R(V)$, first $\frac{\partial U}{\partial t}$ is rewritten as:

$$\frac{\partial U}{\partial t} = \frac{\partial U}{\partial V} \frac{\partial V}{\partial t} = M \frac{\partial V}{\partial t}$$

In this equation, M is the derivative of U with respect to V . With ρ_p as the derivative of density ρ with respect to pressure p and ρ_T as derivative with respect to temperature T , this gives:

$$M = \begin{bmatrix} \rho_p & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & \rho_T u \\ \rho_p v & 0 & \rho & \rho_T v \\ \rho_p c_p T & 0 & 0 & \rho_T c_p T + \rho c_p \end{bmatrix}$$

After this, the system is simplified by introducing a matrix K and premultiplying all terms by this matrix which converts the system to a nonconservative form. Here, K is chosen as:

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 \\ -v & 0 & 1 & 0 \\ -c_p T & 0 & 0 & 1 \end{bmatrix}$$

This gives for the nonconservative system:

$$KM \frac{\partial V}{\partial t} + \nabla \cdot K \bar{F}^c(V) - K \nabla \cdot \bar{F}^v(V, \nabla V) - KQ = 0$$

With

$$KM = \begin{bmatrix} \rho_p & 0 & 0 & \rho_T \\ 0 & \rho & 0 & 0 \\ 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & \rho c_p \end{bmatrix}$$

Incompressibility is introduced in this system by stating that in an incompressible system, ρ is only dependent on the temperature: $\rho = \rho(T)$, which would mean that $\rho_p = 0$. However, in this preconditioning method the coupling of pressure and velocity is introduced by taking $\rho_p = \frac{1}{\beta^2}$ with $\beta^2 = \epsilon^2(\bar{v} \cdot \bar{v})_{max}$ where the value of ϵ can be defined by the user and has a default value of 2.0 in this case. This gives the nonconservative preconditioning matrix Γ_{nc} :

$$\Gamma_{nc} = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & \rho_T \\ 0 & \rho & 0 & 0 \\ 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & \rho c_p \end{bmatrix}$$

A conservative form is retrieved back by premultiplying the entire system by K^{-1} . This gives for Γ :

$$\Gamma = \begin{bmatrix} \frac{1}{\beta^2} & 0 & 0 & \rho_T \\ \frac{u}{\beta^2} & \rho & 0 & \rho_T u \\ \frac{v}{\beta^2} & 0 & \rho & \rho_T v \\ \frac{c_p T}{\beta^2} & 0 & 0 & \rho_T c_p T + \rho c_p \end{bmatrix}$$

The final form of the governing equations for the flow as a function of the primitive variables is now retrieved and is written as:

$$R(V) = \Gamma \frac{\partial V}{\partial t} + \nabla \cdot \bar{F}^c(V) - \nabla \cdot \bar{F}^v(V, \nabla V) - Q = 0 \quad (\text{A.2})$$

A.2 Flamelet generated manifold method

In Section 1, it has been described that the Flamelet Generated Manifold (FGM) method is used in this project to model combustion in the simulations. In Section 2, the most important choices for using this method are shown to the reader that is familiar with this technique. In this section, a more detailed explanation of the method is provided and the choices made for the variables used to control the combustion problem are elaborated.

Perhaps the most important assumption behind this method is that a multi-dimensional flame can be seen as a set of one-dimensional curves, called flamelets. Through this assumption, combustion can be modeled using one-dimensional flame characteristics [14].

One-dimensional flame simulation

The first step in the FGM process is the simulation of a one-dimensional, premixed flat flame. For this calculation, the species mass fractions as used in the multi-dimensional case are used. This procedure can be carried out using specialized combustion software that calculates the flame using detailed chemistry equations, such as CHEM1D [15]. From this simulation, the characteristics of a one-dimensional flame can be retrieved. In Figure A.1, the species mass fractions of O₂, CO, CO₂ and H₂O are shown for the simulation of a stoichiometric methane/air flame.

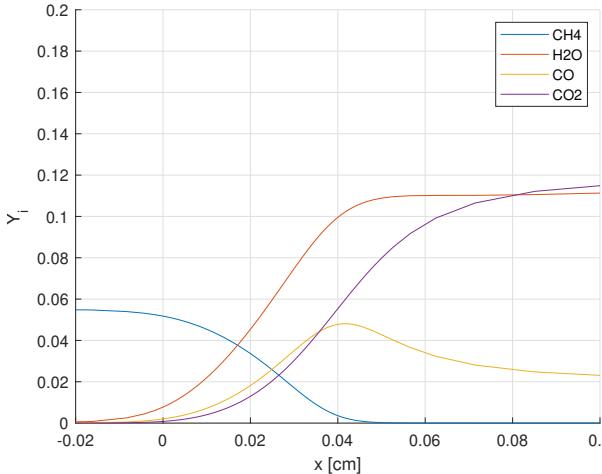


Figure A.1: Stoichiometric flame characteristics

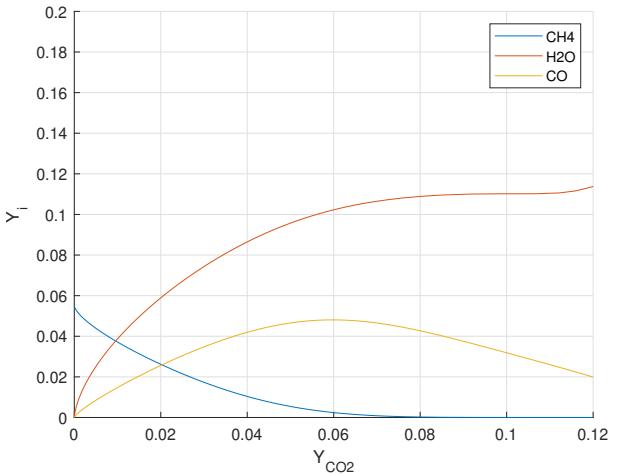


Figure A.2: Species mass fractions as function of Y_{CO₂}

From this figure, it can be seen that the species mass fractions of O₂, CO₂ and H₂O have a unique value within the entire flame area, while the mass fraction CO first increases to a maximum and then decreases. This unique mapping between the species mass fractions and the progression within the flame can be exploited within the FGM method to derive the entire flame characteristics based on the value of the mass fraction of one species. This is illustrated in Figure A.2, for the mass fractions of the same species as a function of the mass fraction of CO₂.

Progress variable selection

The variable used to characterize the progression of the flame between the unburnt and the burnt side is called the progress variable, indicated with \mathcal{Y} . In principle, this can be any linear combination of species mass fractions [14]. However, the extent to which this variable is useful to define the entire flame characteristics is dependent on a couple of factors. Besides the uniqueness of the mapping between the mass fraction and the progression of the flame, the slope of the mass fraction increase or decrease is important.

In Figure A.1, the importance of this increase can be seen in a comparison between the mass fraction of CH₄ and CO₂. For the former, the difference in mass fraction value is a lot smaller than that of the latter after $x = 0.05$ cm. In practice, due to the limited accuracy of the computational method, this could mean that a small error in the mass fraction of CH₄ as the progress variable have a much larger effect on the value of the mass fraction of CO₂. This behavior is especially large for calculation of the source terms of the different species. In Figure A.3, this is illustrated for the source terms of the species used as a progress variable. Here, profiles of the source terms are a lot smoother as a function of CO₂ than those of the other species.

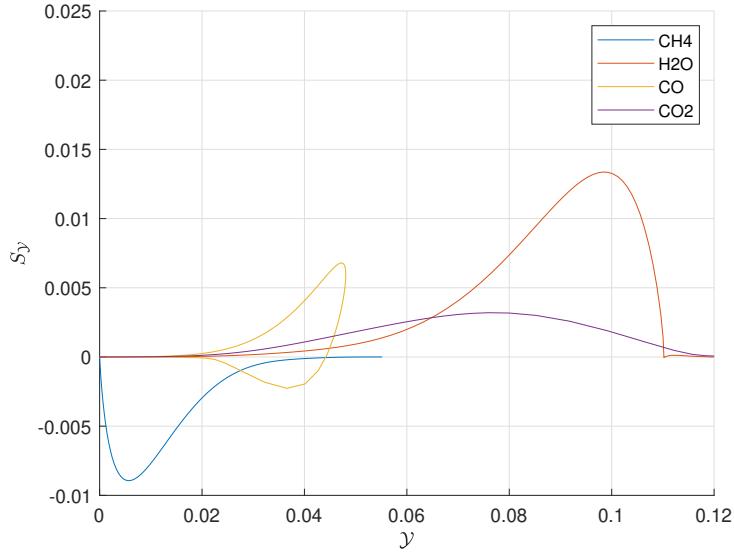


Figure A.3: Source terms as a function of mass fractions

Tabulation of characteristics

During runtime of the two-dimensional simulation, the following conservation is solved for this variable, as shown in Section 2:

$$\frac{\partial(\rho\mathcal{Y})}{\partial t} + \nabla \cdot (\rho\bar{u}\mathcal{Y}) - \nabla \cdot (\rho\mathcal{D}\nabla\mathcal{Y}) - Q = 0 \quad (\text{A.3})$$

In these equations, ρ represents the density, \bar{u} the velocity vector, κ the thermal conductivity, \mathcal{D} the diffusivity and Q the source term, which includes for instance the chemical source term $\omega_\mathcal{Y}$ and the contributions due to axisymmetry of the problem.

In this method, the variables that are not solved using the governing equations as explained in Section 2 are stored in a lookup table. This table, calculated prior to calculation, contains all relevant flow parameters calculated through detailed chemistry equations as a function of the progress variable. During runtime of the simulation, the values of these variables are retrieved from the table and used to solve Equation A.3.

Multi-dimensional manifold

In order to provide a better accuracy for the FGM method, more progress variables can be used to characterize the position within the flame. The addition of these extra variables can decrease the inaccuracies as described earlier in this section at the expense of a longer computational time due to having to calculate an extra conservation equation for the additional progress variable.

One method of adding this extra variable to the lookup table is to first do an initial simulation of the one-dimensional flame. From this simulation, the characteristics of the entire flame and therefore both progress variables are known. Then, if two other one-dimensional simulations are done while keeping one of the variables constant in every simulation, a triangle-like structure is constructed. This can then be done for the entire path from the unburnt to the burnt side, which gives the full two-dimensional manifold. Analogously, more additional progress variables can be added to the simulations.

A.3 Adjoint equation derivation

In Section 2.2 the equation for the sensitivity of the user-defined objective function with respect to changes in the vector of design variables α is given. This is written as [11]:

$$\frac{dJ^T}{d\alpha} = \frac{\partial J^T}{\partial \alpha} + \frac{\partial^T}{\partial V} \frac{\partial V}{\partial \alpha} \quad (\text{A.4})$$

In order to arrive at the expression for the adjoint equations, it is assumed that the equations governing the flow is dependent on the design variables α and the flow state V , which itself is also dependent on α : $R = R(\alpha, V(\alpha))$. In Section 2.2 it has been described that the governing equations for the flow are required to satisfy $R = 0$, which also implies that the derivative of R is with respect to α is equal to zero:

$$\frac{dR}{d\alpha} = \frac{\partial R}{\partial \alpha} + \frac{\partial R}{\partial V} \frac{\partial V}{\partial \alpha} = 0 \quad (\text{A.5})$$

Now, if the solution of Equation A.5 is multiplied by the arbitrary Lagrange multiplier Ψ and subtracted from Equation A.4, this gives:

$$\frac{dJ^T}{d\alpha} = \left(\frac{\partial J^T}{\partial V} - \Psi^T \frac{\partial R}{\partial V} \right) \frac{\partial V}{\partial \alpha} + \frac{\partial J^T}{\partial \alpha} - \Psi^T \frac{\partial R}{\partial \alpha}$$

If the so-called adjoint state Ψ is now chosen such that the term $\frac{\partial J^T}{\partial V} - \Psi^T \frac{\partial R}{\partial V} = 0$, the final equations as presented in Equation 2.2 result. For completeness, these equations are written as:

$$\frac{dJ^T}{d\alpha} = \frac{\partial J^T}{\partial \alpha} - \Psi^T \frac{\partial R}{\partial \alpha} \quad \text{with} \quad \frac{\partial R^T}{\partial V} \Psi = \frac{\partial J}{\partial V} \quad (\text{A.6})$$

A.4 Algorithmic Differentiation

In Section 2.2 it has been described that Algorithmic Differentiation (AD) can be used to efficiently calculate the derivative of either multiple output functions with respect to a single input variable in the forward mode or the derivative of a single output function with respect to multiple input variables in the reverse or adjoint mode. In this section, the calculations used in both modes will be described in more detail through use of an example.

The calculation of derivatives in AD is done using a repeated application of the chain rule and by simplification of the equations into single operations. The resulting structure, the computational graph, can then be visualized to show the dependence between the input and output variables. [36]. In Figure A.4 this is shown for the equation $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ where the operations as done on the input values are displayed within each node [37].

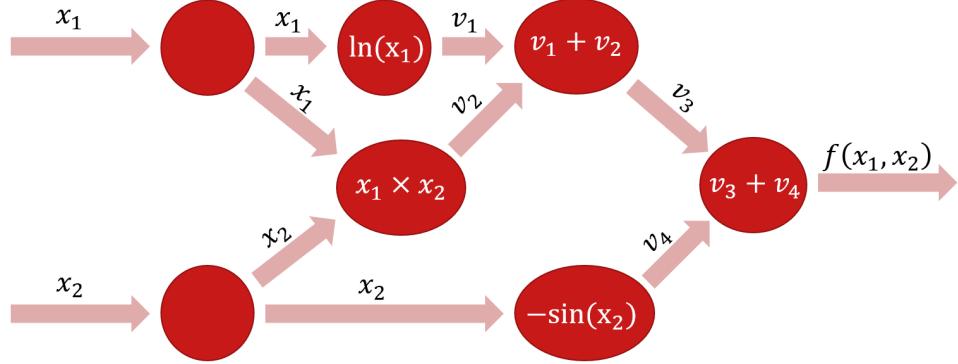


Figure A.4: Computational graph for $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$

Using this graph, the values v_i that are calculated can be used in the differentiation of the solution. In the forward mode, the process is followed from the input to the output. Here, the input values x_i and their derivatives \dot{x}_i are provided. These variables then traverse the graph from left to right, passing through the nodes where the derivative operations from Figure A.4 are applied to the input variables. This process is shown in Figure A.5.

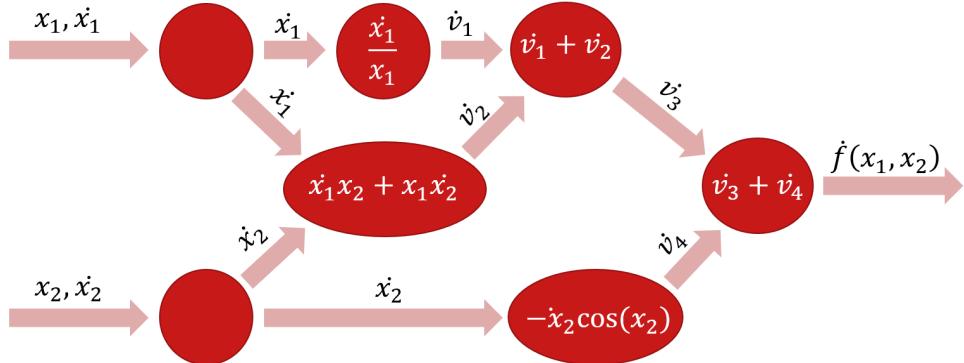


Figure A.5: Forward mode of AD

In this method, the input variables x_i are dependent on the derivative represented by \dot{f} . Here, if the derivative with respect to x_1 is calculated, $(\dot{x}_1, \dot{x}_2) = (\frac{\partial x_1}{\partial x_1}, \frac{\partial x_2}{\partial x_1}) = (1, 0)$. Similarly, if the derivative with respect to x_2 is calculated, $(\dot{x}_1, \dot{x}_2) = (0, 1)$. This way, in each pass through the graph the derivative of the output function with respect to a single input variable can be calculated. Consequently, the amount of passes through this graph is linearly dependent on the amount of input variables.

In the reverse, or adjoint mode of AD, the derivatives are calculated by traversing the graph in the opposite direction of the primal. Here, the input values are the value of the objective function $f(x_1, x_2)$ and its derivative with respect to itself. This gives $\bar{f} = 1$. In Figure A.6, the operations as done are displayed.

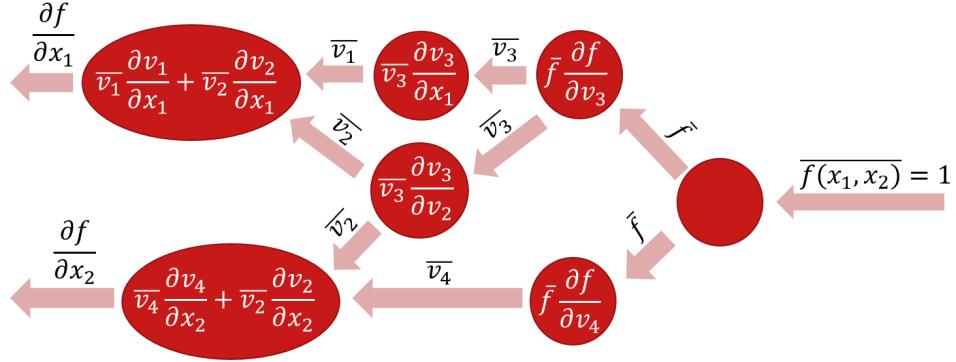


Figure A.6: Adjoint mode of AD

In order to illustrate what is calculated in this figure, the combination of this graph with the primal trace as shown in Figure A.4 is made. Using this method, the derivative of f with respect to both x_1 and x_2 can be calculated by traversing the graph only once. Analogous to the case in the forward mode where the derivatives of x_i are either 1 or 0, the derivatives of additional objective functions in the adjoint mode would be either 1 or 0 based on which partial derivative is taken. Therefore, the amount of passes through this graph increases linearly with the amount of objective functions.

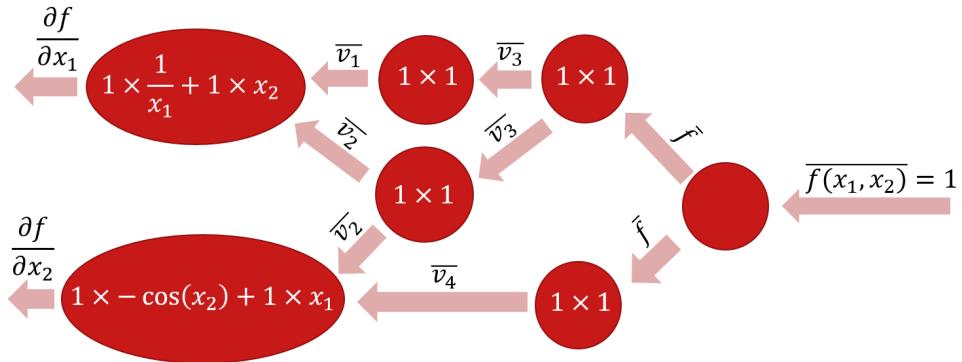


Figure A.7: Adjoint mode of AD applied to $f(x_1, x_2) = \ln(x_1) + x_1x_2 - \sin(x_2)$

A.5 Free form deformation basis functions

The procedure of Free Form Deformation (FFD) has been described briefly in Section 2.3. There, the equation that provides the mapping between the control point positions and the cartesian coordinates of the deformed surface points has been presented in Equation 2.7. In this equation, it has been explained that the specific interpolation between control points and deformed object point coordinates is dependent on the value of so-called basis functions. Although more representations exist, in this section these basis functions will be described in more detail for blending using Bézier curves.

Bézier blending functions

The basis functions of Bézier polynomials are more commonly referred to as Bernstein polynomials [4, 26]. The value of this polynomial is dependent on three parameters, the parametric coordinate u , degree p and index i . Using these parameters, basis function i of degree p can be written as [26]:

$$B_i^p(u) = \frac{p!}{i!(p-i)!} u^i (1-u)^{p-i} \quad (\text{A.7})$$

In this equation, parametric coordinate represents the position of the point under consideration with respect to the FFD box as described in Section 2.3 and index i represents the number of the control point under consideration.

The value of degree p is a value that can be chosen by the modeler and represents the 'smoothness' of the curve. Here, an increased degree of the curve leads to a smoother curve. In Figure A.8, this is represented for the basis functions of degree 1, 2 and 4.

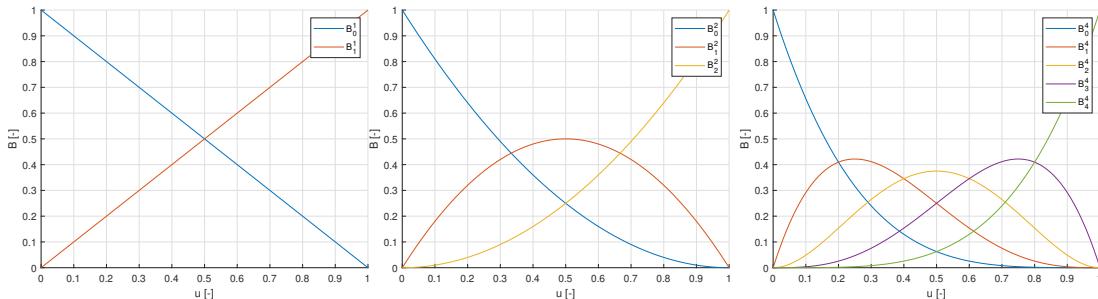


Figure A.8: Bézier blending basis functions

Another method for calculation of the value of basis function B_i^p for a fixed value of u can be achieved through the recursive property of this type of curve [26]:

$$B_i^p(u) = (1-u)B_i^{p-1}(u) + uB_{i-1}^{p-1}(u)$$

Here, $B_i^p = 0$ if $i < 0$ or $i > n$. Using the property that Bézier curves of degree 0 are either 0 or 1, this method can be more efficient than the direct calculation of the value through use of the Bernstein polynomial.

B CFD simulation

B.1 Detailed chemistry simulation

The results of the CFD simulation are validated using a laminar finite-rate chemistry model in the commercial software ANSYS Fluent as described in Section 3. In this section the chemistry model is explained and the choice for the mesh size is shown.

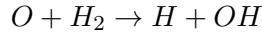
Finite-rate chemistry model

At the core of the equations governing the finite-rate chemistry model are the equations for continuity of species, solved in addition to the Navier-Stokes equations. In these equations, a separate differential equation is solved for every species available in the mixture. In Equation B.1, this equation is shown [14].

$$\frac{\partial(\rho Y_i)}{\partial t} + \nabla \cdot (\rho \bar{u}_i Y_i) = \dot{\omega}_i \quad (\text{B.1})$$

In this equation, ρ represents the density of the entire mixture, which is the sum of the density of all individual species. Y_i represents the mass fraction of species i . Furthermore, \bar{u}_i represents the specific velocity of species i .

The final term in Equation B.1, $\dot{\omega}_i$, represents the mass reaction rate. This is the mass of species i produced or consumed per unit volume and time. The value of the mass reaction rate $\dot{\omega}_i$ is based on the chemical reactions that take place in the mixture. The calculation of this variable is shown through use of the reaction between atomic oxygen (O) and hydrogen (H_2) that forms atomic hydrogen (H) and hydroxyl (OH) [38]:



This reaction can be written in a more general form by summing the contributions over the total number of species N_s :

$$\sum_{i=1}^{N_s} \nu'_i \mathcal{M}_i \xrightarrow{k_f} \sum_{i=1}^{N_s} \nu''_i \mathcal{M}_i$$

In this expression, ν'_i and ν''_i represent respectively the stoichiometric coefficients before and after the reaction of species i . Here, species i is represented by \mathcal{M}_i . The rates at which these reactions occur are proportional to the concentrations of the species present before the reaction, with k_f as the forward reaction rate coefficient defined by the so-called Arrhenius expression [39]:

$$r_f = k_f [O][H_2] \quad \text{with} \quad k_f = AT^\beta \exp(-E_a/(RT))$$

In this description, r_f represents the reaction rate of the forward reaction. $[O]$ and $[H_2]$ are the concentrations of respectively molecular oxygen and hydrogen. A and β are reaction constants and E_a is the activation energy. The values of the variables contained in k_f are included in the input file for the reaction mechanism. Using the same notation and the fact that every reaction can take place in both directions, the general expression for the reaction is extended using backward reaction rate k_b to [38]:

$$\sum_{i=1}^{N_s} \nu'_i \mathcal{M}_i \xrightleftharpoons[k_b]{k_f} \sum_{i=1}^{N_s} \nu''_i \mathcal{M}_i$$

Now, the expression for the reaction rate can be extended to include the backward reaction rate. Using the same expression of the equations involving the reaction itself, with $[\mathcal{M}_i]$ the concentration of species i , this gives:

$$r = k_f \prod_{i=1}^{N_s} [\mathcal{M}_i]^{\nu'_i} - k_b \prod_{i=1}^{N_s} [\mathcal{M}_i]^{\nu''_i}$$

Now, by summing the contributions of the reaction rate over all reactions R as used in the mechanism, the total source term of species i can be calculated [14]:

$$\omega_i = M_i \sum_{j=1}^R (\nu''_{ij} - \nu'_{ij}) r_j \quad (\text{B.2})$$

As a consequence of the species interacting with each other and each species having interactions with multiple other species, the conservation equations for all species are coupled through the source terms.

Mesh size

The model setup for the simulations in Fluent is taken similar to that in the SU2 simulations, to allow for a direct comparison. This means that the boundary conditions are taken as presented in Figure 3.1. The heat of reaction has been used to base the mesh size on, where it is required to increase and decrease over multiple elements. Using the same mesh as in the SU2 simulation, this is shown in Figure B.1. Based on these results, the mesh size is assumed to be fine enough to accurately represent the flame.

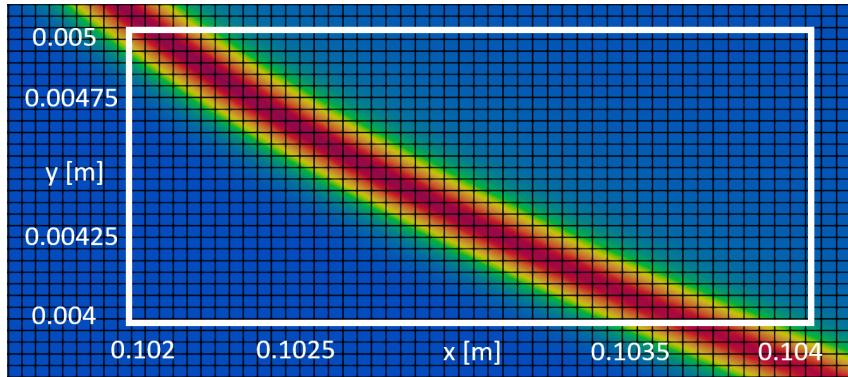


Figure B.1: Heat of reaction around the burner outflow surface

C Unconstrained optimization

C.1 Boundary conditions for nonphysical deformations

In Section 4, two examples are presented of deformations that lead to nonphysical results after deformations. In one of these examples, the simulation did converge. However, the resulting geometry could never be achieved in a real model. This is shown in Figure C.1a. The other model, as shown in Figure C.1b, led to divergence of the results and thereby to the optimization procedure being stopped. In this section, the setups for these problems are described in more detail and the origin of the nonphysical deformations are elaborated for each of the examples.

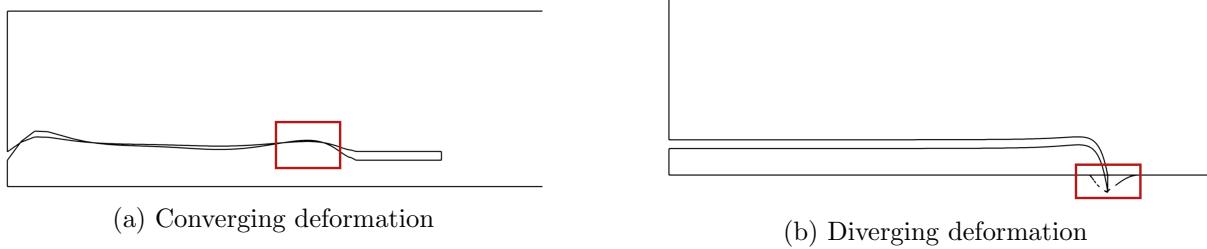


Figure C.1: Examples of nonphysical results after deformation

Converging deformation

This model is setup as a response to the deformation of the burner outflow surface as presented in Section 4. In this model, an effort is done to keep the burner outflow surface straight by decreasing the amount of deformation close to the outflow surface. This is done by deforming only a part of the burner tube, instead of the entire tube as done before. In this FFD box a grid of 40x2 control points is used as opposed to the 50x2 in the original model. This is done to achieve the same control point spacing of 2 mm. In Figure C.2, this FFD box is shown.

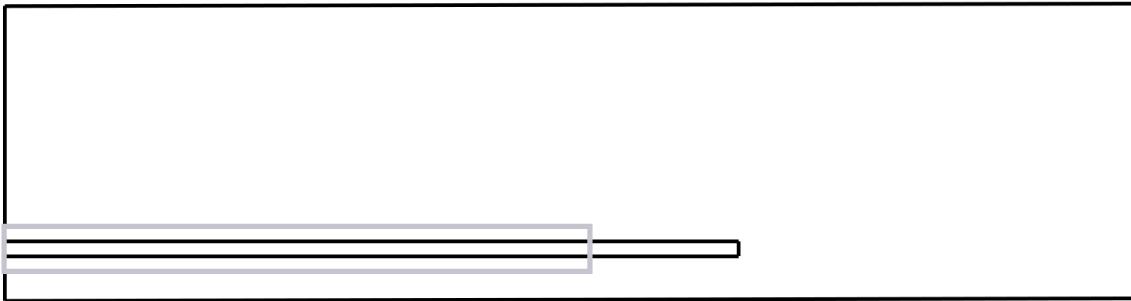


Figure C.2: FFD box definition for converging model

Besides allowing deformations of only a part of the burner tube, an additional constraint has been imposed on the FFD box: the position of the upper row of control points is fixed. Through this constraint, the location of the inner burner wall cannot be influenced by the possible deformations of the top row. This provides a direct relation between the deformation of the bottom row of control points and the inner burner wall.

As a consequence of the way deformations are performed in FFD as described in Section 2, deformation of the bottom row of control points has a larger effect on the inner burner wall than on the top burner wall. Therefore, a deformation of the lower row of control points in the direction of the upper row will lead to the two burner walls being moved closer together.

In case the required deformations are too large, the control points on the lower row can cross those on the upper row. As a consequence, the two burner walls can intersect. This has led to the nonphysical deformation occurring. As an illustration, this is shown in Figure C.3. Here, the undeformed FFD box is shown in gray, the deformed FFD box in red.

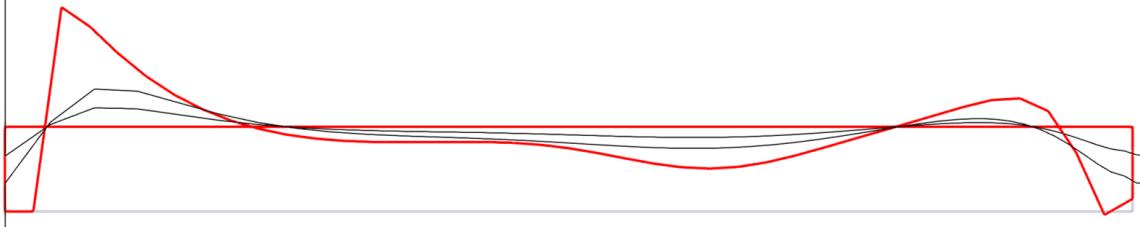


Figure C.3: FFD box definition for converging model

Diverging deformation

For this deformation, the setup of the problem is equal to the one shown in Section 4 except for the maximum deformation of the control points. This magnitude is set in such a way that the deformation is large enough to provide a noticeable change in the geometry, but small enough to perform deformations that are physically valid.

Although this method can be used to directly influence the magnitude of the control point deformations in the first iteration to provide the desired results, this does not provide fully reliable results in consecutive iterations. As a consequence of the deformation of the model, the sensitivity of the objective function with respect to the control point positions is changed. In case of an increased sensitivity, the magnitude of the change in control point positions is also increased. If this is not taken into account by the modeler, this can lead to the deformations being too large for the model to handle and the deformation as shown in Figure C.1b.

D Prevention of nonvalid deformations

D.1 Partial derivatives of Bézier surface

In Section 2.3 the construction of a surface using Bézier basis functions has been described. The function that defines this surface, also described in Equation 2.7, is given as [4]:

$$\bar{F}(u, v, w) = \sum_{i=0}^L \sum_{j=0}^M \sum_{k=0}^N B_i(u) B_j(v) B_k(w) \bar{P}_{ijk}$$

In this equation, \bar{F} represents function used to retrieve the cartesian coordinates of a point on the surface. The location of this point is then defined as a function of parametric coordinates (u, v, w) that can assume a continuous value between 0 and 1, through Bézier basis functions B and control points \bar{P} with index i, j and k depending on the amount of control points as defined in every direction x, y and z .

Calculation of the derivative of surface F is done using the chain rule. In this calculation, it is first noted that the control point positions are fixed, which means that their derivatives are 0. Furthermore, the parametric coordinates u, v and w are by definition independent of each other, as described in Section 2.3. Specifically, this means that the derivative of a basis function of one parametric coordinate with respect to another is always 0. Using this knowledge, the derivative of function F with respect to parametric coordinate u is defined as:

$$\frac{\partial \bar{F}}{\partial u} = \sum_{i=0}^L \sum_{j=0}^M \sum_{k=0}^N B'_i(u) B_j(v) B_k(w) \bar{P}_{ijk} \quad (\text{D.1})$$

In this equation, $B'_i(u)$ represents the derivative of basis function B with respect to parametric coordinate u .

Basis function derivative

The definition of the basis functions using Bézier blending is described in Section 2.3. For the function with index i and degree p , this is defined as, as also presented in Equation 2.8 [26]:

$$B_i^p(u) = \frac{p!}{i!(p-i)!} u^i (1-u)^{p-i}$$

Using this definition and the chain rule, the definition of its derivative with respect to parametric coordinate u can be written as:

$$\frac{dB_i^p(u)}{u} = i \left(\frac{p!}{i!(p-i)!} u^{i-1} (1-u)^{p-i} \right) - (p-i) \left(\frac{p!}{i!(p-i)!} u^i (1-u)^{p-i-1} \right)$$

Now, the first part of this equation can be rewritten in terms of $i-1$ and the second part in terms of both $i-1$ and $p-1$ by taking apart the factorials:

$$\frac{dB_i^p(u)}{u} = i \frac{p}{i} \left(\frac{(p-1)!}{(i-1)!(p-i)!} u^{i-1} (1-u)^{p-i} \right) - (p-i) \frac{p}{p-i} \left(\frac{(p-1)!}{i!(p-i-1)!} u^i (1-u)^{p-i-1} \right)$$

Rewriting this expression gives:

$$\frac{dB_i^p(u)}{u} = p \left(\frac{(p-1)!}{(i-1)!(p-i)!} u^{i-1} (1-u)^{p-i} - \frac{(p-1)!}{i!(p-i-1)!} u^i (1-u)^{p-i-1} \right)$$

In this expression, the two terms within the brackets can now be written as Bézier basis functions of degree $p-1$ as shown in Equation D.2. In this equation, since the basis functions B_{-1}^{p-1} and B_p^{p-1} are not defined, their values are taken to be 0 by definition [26].

$$\frac{dB_i^p(u)}{u} = p \left(B_{i-1}^{p-1} - B_i^{p-1} \right) \quad (\text{D.2})$$

This expression for the derivative of the basis functions can now be used for calculating the partial derivatives of the deformed points in the FFD surface with respect to the parametric coordinates as presented in Section 5. Through use of these partial derivatives, the determinant of the Jacobian matrix for the FFD deformation and subsequently possible self-intersections within the deformed surface can be calculated.

E TU/e scientific code of conduct



Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conductⁱ.

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

23-10-2020

Name

L.J.C.M. Tel

ID-number

0894988

Signature

Lemarst

Submit the signed declaration to the student administration of your department.

ⁱ See: <https://www.tue.nl/en/our-university/about-the-university/organization/integrity/scientific-integrity/>

The Netherlands Code of Conduct for Scientific Integrity, endorsed by 6 umbrella organizations, including the VSNU, can be found here also. More information about scientific integrity is published on the websites of TU/e and VSNU