A multi-level local defect correction technique for laminar flame simulation

A multi-level local defect correction technique for laminar flame simulation

J.H.M. ten Thije Boonkkamp, R. Rook and R.M.M. Mattheij
Department of Mathematics and Computer Science, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract

We present a numerical algorithm to solve the low-Mach number approximation of the conservation laws for laminar flames. Elements of our algorithm are a pressure correction (PC) method to decouple the computation of velocity and pressure, and a multi-level local defect correction (LDC) method to solve the resulting set of (non)linear boundary value problems. The PC method is based on a constraint equation rather than the continuity equation, describing expansion of the gas mixture due to combustion. Boundary value problems for laminar flames are characterised by a high activity region, the so-called flame front, where the solution varies very rapidly, whereas outside this region the solution is very smooth. The basic idea of the LDC method is to compute a global coarse grid solution, that is accurate enough to represent the solution outside the flame front, and a sequence of local fine grid solutions to capture all the details in the flame front. Moreover, these fine grid solutions are subsequently used to improve the coarse grid solution by a defect correction technique. We have applied our PC-LDC algorithm to simulate a flame on a slit burner.

Keywords. Laminar premixed flames, low-Mach number approximation, expansion equation, pressure correction, pressure equation, (multi-level) local defect correction, nonlinear boundary value problem.
AMS subject classifications. 35Q30, 65N22, 76V05.

1 Introduction

Numerical simulation codes for laminar flames are a useful tool to investigate the behaviour of flames and/or design burning devices. Nowadays, advanced numerical methods for partial differential equations (PDEs) and powerful computers enable the numerical simulation of complex combustion phenomenae. However, numerical flame simulation is still a demanding task, for several reasons. First, in flames many chemical reactions take place involving many different species, and for each species a conservation equation has to be solved. Typically, GRI-Mech 3.0 contains 325 reactions and 53 species [12]. Moreover, these conservation equations have to be coupled with the flow equations. Consequently, the mathematical model consists of a large set of partial differential equations in two- or even three dimensional space. Second, the governing equations are extremely stiff, meaning that they allow for solutions with largely varying time and space scales. More specifically, very rapid variations occur in the flame front, which is a very thin zone between the unburned and burned gases, where all the reactions and heat production take place; outside the flame front the solution is almost constant. A third difficulty is that the chemical source terms are extremely nonlinear in the temperature. Finally, transport models, e.g. for heat flux or mass diffusion, are often very complicated and result in many coupling terms in the governing equations [11].

A numerical method for laminar flame simulation should at least satisfy the following requirements. First, a refined grid near the flame front is absolutely manadatory to capture its detailed structure and to keep the storage requirements and computing time to a minimum. Second, space discretisation methods should be accurate in all regimes of the problem and should preserve certain properties of the PDE formulation, e.g. conservation or positivity of the species mass fractions. For time dependent flames, a time integrator is needed that can handle stiff equations. This implies that only implicit methods are suitable. As a result of space discretisation, and possibly time integration, we obtain large systems of algebraic equations. Iterative solution methods for these systems should be fast, and above all, robust. Finally, a projection method, also referred to as pressure correction method, to decouple the pressure and velocity computation is needed.

A lot of research on all these aspects of flame simulation is going on, although not always within the context of combustion simulation. With regard to grid generation there are basically two options, i.e., structured grids versus unstructured grids. A representative of the first approach is the multi-level adaptive mesh refinement (AMR) method; see e.g. [1]. On the other hand, unstructured grids are often combined with finite element methods (FEM). FEM simulations of laminar flames on unstructured grids are presented in e.g. [7, 9]. A finite difference method on unstructured grids for flame simulation is the local rectangular refinement (LRR) method [5, 6]. Next, splitting methods are often used for time integration of conservation laws [21]. The basic idea is to split the set of governing equations into several subsytems, such that each subsystem can be integrated efficiently. For example, for laminar flames it is very benificial to apply different time integration methods to the advection, diffusion and reaction terms, respectively. Explicit methods are suitable for the advection terms, whereas implicit method are required for the other two terms; see e.g. [24]. Finally, pressure correction methods were originally introduced for incompressible flow, and are nowadays also widely used in the numerical simulation of compressible flows and flames; see e.g. [1, 10, 25, 20].

In this paper, we focus on one aspect of laminar flame simulation, i.e., a multi-level method to solve boundary value problems that are characterised by a high activity region, where the solution varies extremely fast compared to its relatively smooth behaviour outside the region. Obviously the flame front is the high activity region. Our method of choice is the local defect correction (LDC) method. The basic idea of the method is as follows. First, we compute the solution on a global, uniform coarse grid that is fine enough to represent the solution outside the flame front. However, this coarse grid is certainly not fine enough to capture the detailed structure of the flame inside the flame front. Therefore, we cover the flame front with a sequence of local, uniform fine grids, and recompute the solution there using Dirichlet boundary data from the next coarser grid solution. Finally we use the fine grid solutions to improve the coarse grid solution by a defect correction technique. The use of uniform grids in LDC is very advantageous: they can be represented by simple data structures, allow accurate discretisation methods and there exist efficient iterative solution methods for the resulting algebraic systems. The LDC method has been developed and analysed for elliptic boundary value problems in [2, 3, 13, 15]. Applications to flame simulation are presented in [16], where the method is used to solve the thermodiffusive model, and in [4], to solve the streamfunction-vorticity formulation of a premixed Bunsen flame.

We have organized our paper as follows. In Section 2 we briefly summarize the governing equations of laminar flames. We describe the PC method for these equations in Section 3. Next, in Section 4 and Section 5 we describe the two-level and multi-level versions of the LDC method, respectively. The combined PC-LDC method applied to laminar flames is given in Section 6. Finally, in Section 7, we apply the method to simulate a flame on a slit burner.

2 Governing equations

In this section we summarize a mathematical model for laminar, premixed flames under atmospheric conditions. Typically, we consider the combustion of hydrocarbons in air. The species in the flame are numbered 1 through N. Species N is nitrogen and is present in abundance, while the other species are considered trace species. The governing equations for such flames are the conservation equations of mass, momentum and energy of the gas mixture and the balance equations of mass for the trace species. These equations can be written in the following form [26, 29]

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho \boldsymbol{v}) = 0, \tag{2.1a}$$

$$\frac{\partial}{\partial t}(\rho \boldsymbol{v}) + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{v}\boldsymbol{v}) = -\nabla p + \boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{T}} + \rho \boldsymbol{g}, \tag{2.1b}$$

$$\frac{\partial}{\partial t}(\rho h) + \nabla\cdot(\rho \boldsymbol{v} h) - \nabla\cdot\Big(\frac{\lambda}{c_p}\nabla h\Big) = \nabla\cdot\boldsymbol{J}_h, \tag{2.1c}$$

$$\frac{\partial}{\partial t}(\rho Y_i) + \nabla\cdot(\rho \boldsymbol{v} Y_i) - \nabla\cdot\Big(\frac{1}{\mathrm{Le}_i}\frac{\lambda}{c_p}\nabla Y_i\Big) = \omega_i, \quad i = 1, 2, \ldots, N-1. \tag{2.1d}$$

The independent variables in (2.1) are the density ρ, the flow velocity $\boldsymbol{v}$, the hydrostatic pressure p, the specific enthalpy h and the species mass fractions Y_i. Other variables/constants in (2.1) are the viscous stress tensor $\boldsymbol{\mathcal{T}}$, the gravitational acceleration vector $\boldsymbol{g}$, the thermal conductivity λ, the specific heat at constant pressure c_p, the enthalpy flux $\boldsymbol{J}_h$, the Lewis numbers Le_i, and the reaction rates ω_i. The first two equations in (2.1) are referred to as the *flow equations* and the latter two as the *combustion equations*. Note, that we only have to solve the conservation equations for the first $N-1$ species and that the mass fraction of the last species follows from the constraint $\sum_{i=1}^{N} Y_i = 1$.

The conservation equations (2.1) have to be completed with the caloric equation of state

$$h = \int_{T_{\mathrm{ref}}}^{T} c_p(T', Y_i)\,\mathrm{d}T' + \sum_{i=1}^{N} h_{i,\mathrm{ref}} Y_i, \tag{2.2a}$$

which defines h as a function of the temperature T and the species mass fractions Y_i, and the thermal equation of state

$$p_{\mathrm{amb}} = \frac{\rho R T}{M}, \quad \frac{1}{M} = \sum_{i=1}^{N} \frac{Y_i}{M_i}. \tag{2.2b}$$

In (2.2) $h_{i,\mathrm{ref}}$ are the specific enthalpies of formation at reference temperature T_{ref}, p_{amb} is the ambient pressure, R is the universal gas constant, M is the average molar mass of the gas mixture and M_i are the species molar masses.

The main assumptions underlying the governing equations in (2.1) and (2.2) are the following. First, the gas mixture is an ideal, Newtonian fluid. Second, the diffusion velocities $\boldsymbol{V}_i$ are given by the relations [27]

$$\rho Y_i \boldsymbol{V}_i = -\frac{1}{\mathrm{Le}_i}\frac{\lambda}{c_p}\nabla Y_i, \quad \mathrm{Le}_i := \frac{\lambda/c_p}{\rho D_{i,\mathrm{m}}}, \quad i = 1, 2, \ldots, N-1, \tag{2.3a}$$

$$\sum_{i=1}^{N} Y_i \boldsymbol{V}_i = \boldsymbol{0}, \tag{2.3b}$$

where $D_{i,\mathrm{m}}\,(i = 1, 2, \ldots, N-1)$ are the mixture averaged diffusion coefficients of the trace species. Moreover, the heat flux vector $\boldsymbol{q}$ is given by

$$\boldsymbol{q} = -\lambda \nabla T + \rho \sum_{i=1}^{N} h_i Y_i \boldsymbol{V}_i, \tag{2.4}$$

i.e., only conduction and enthalpy transport by diffusion are taken into account. Combining the relations in (2.3) and (2.4) we obtain the following expression for the enthalpy flux

$$\boldsymbol{J}_h = \frac{\lambda}{c_p} \sum_{i=1}^{N} \Big(\frac{1}{\mathrm{Le}_i} - 1 \Big) h_i \nabla Y_i. \tag{2.5}$$

Finally, the low-Mach number approximation holds, i.e., p is set to a constant in the equation of state (2.2b) and the term $\mathrm{D}p/\mathrm{D}t$ is neglected in the enthalpy equation (2.1c); see [8].

Using the thermal equation of state (2.2b), we can express the density ρ as a function of the combustion variables T and Y_i. Applying the material derivative $\mathrm{D}/\mathrm{D}t := \partial/\partial t + \boldsymbol{v}\cdot\nabla$ to this relation and eliminating $\mathrm{D}\rho/\mathrm{D}t$ from the continuity equation (2.1a), we obtain a constraint equation of the form

$$\nabla\cdot\boldsymbol{v} = s, \tag{2.6a}$$

where s describes expansion of the gas mixture due to conduction, enthalpy transport, diffusion and chemical reactions. It is given by

$$\begin{aligned} s :&= \frac{1}{T}\frac{\mathrm{D}T}{\mathrm{D}t} - \frac{1}{M}\frac{\mathrm{D}M}{\mathrm{D}t} \\ &= \frac{1}{c_p T}\frac{\mathrm{D}h}{\mathrm{D}t} + \sum_{i=1}^{N} \Big(\frac{M}{M_i} - \frac{h_i}{c_p T} \Big) \frac{\mathrm{D}Y_i}{\mathrm{D}t} \\ &= \frac{1}{\rho c_p T} \Big[\nabla\cdot\Big(\frac{\lambda}{c_p} \nabla h \Big) + \nabla\cdot\boldsymbol{J}_h \Big] + \frac{1}{\rho} \sum_{i=1}^{N} \Big(\frac{M}{M_i} - \frac{h_i}{c_p T} \Big) \Big[\nabla\cdot\Big(\frac{1}{\mathrm{Le}_i} \frac{\lambda}{c_p} \nabla Y_i \Big) + \omega_i \Big]. \end{aligned} \tag{2.6b}$$

We have determined the terms in the third line of (2.6b) by replacing the material derivatives by expressions obtained from the combustion equations (2.1c), (2.1d) and the caloric equation of state (2.2a). Note that expansion of the mixture due to pressure variations is not taken into account by virtue of the assumption that $p = p_{\mathrm{amb}}$ in (2.2b). Equation (2.6a) is referred to as the *expansion equation* and will replace the continuity equation.

Summarizing, our model for laminar flames consists of the momentum equations (2.1b), the combustion equations (2.1c) and (2.1d), the expansion equation (2.6a) and the equations of state (2.2).

3 Semi-discrete formulation of the combustion problem

We apply the transverse method of lines [23], i.e., discretise the time derivatives first, to derive a set of boundary value problems for velocity, pressure and all combustion variables. In order to decouple the velocity and pressure computation, we employ the pressure correction method. We restrict ourselves to steady flames.

Consider the governing equations (2.1b)-(2.1d) in convective form, together with the expansion equation (2.6a). Introducing the species mass fraction vector $\boldsymbol{Y} := (Y_1, \ldots, Y_{N-1})^{\mathrm{T}}$ and the combustion variables vector $\boldsymbol{\psi} := (T, Y_1, \ldots, Y_{N-1})^{\mathrm{T}}$, these equations can be written in the symbolic form

$$\frac{\partial \boldsymbol{v}}{\partial t} = \boldsymbol{\mathcal{F}}_1[\boldsymbol{v}, \boldsymbol{\psi}] - \rho^{-1}(\boldsymbol{\psi})\nabla p, \tag{3.1a}$$

$$\frac{\partial h(\boldsymbol{\psi})}{\partial t} = \mathcal{F}_2[\boldsymbol{v}, \boldsymbol{\psi}], \tag{3.1b}$$

$$\frac{\partial \boldsymbol{Y}}{\partial t} = \boldsymbol{\mathcal{F}}_3[\boldsymbol{v}, \boldsymbol{\psi}], \tag{3.1c}$$

$$\nabla \cdot \boldsymbol{v} = s(\boldsymbol{\psi}), \tag{3.1d}$$

where $\boldsymbol{\mathcal{F}}_1$, $\mathcal{F}_2$ and $\boldsymbol{\mathcal{F}}_3$ are spatial differential operators. Note that we have explicitly denoted the dependence of ρ, h and s on the combustion variables $\boldsymbol{\psi}$. Obviously, these equations should be coupled to the equations of state (2.2). Since we are not interested in time accurate solutions, we apply the implicit Euler time integration method to (3.1) to find

$$\frac{1}{\Delta t}\big(\boldsymbol{v}^{n+1} - \boldsymbol{v}^n\big) - \boldsymbol{\mathcal{F}}_1[\boldsymbol{v}^{n+1}, \boldsymbol{\psi}^{n+1}] + \rho^{-1}(\boldsymbol{\psi}^{n+1})\nabla p^{n+1} = \boldsymbol{0}, \tag{3.2a}$$

$$\frac{1}{\Delta t}\big(h(\boldsymbol{\psi}^{n+1}) - h(\boldsymbol{\psi}^n)\big) - \mathcal{F}_2[\boldsymbol{v}^{n+1}, \boldsymbol{\psi}^{n+1}] = 0, \tag{3.2b}$$

$$\frac{1}{\Delta t}\big(\boldsymbol{Y}^{n+1} - \boldsymbol{Y}^n\big) - \boldsymbol{\mathcal{F}}_3[\boldsymbol{v}^{n+1}, \boldsymbol{\psi}^{n+1}] = \boldsymbol{0}, \tag{3.2c}$$

$$\nabla \cdot \boldsymbol{v}^{n+1} = s(\boldsymbol{\psi}^{n+1}), \tag{3.2d}$$

where $\boldsymbol{v}^n = \boldsymbol{v}^n(\boldsymbol{x})$ denotes the semi-discrete approximation of $\boldsymbol{v}(\boldsymbol{x}, t^n)$ at time level $t^n := n\Delta t$, with $\Delta t > 0$ the time step; etc. This system is difficult to solve because of the coupling between the velocity field $\boldsymbol{v}^{n+1}$ and the pressure field p^{n+1} in equation (3.2a). One way to deal with this problem is to apply the pressure correction method; see e.g. [17, 28], i.e., solve system (3.2) in a predictor-corrector fashion in the following way. First, we define a predictor $\boldsymbol{v}^*$ for the velocity field $\boldsymbol{v}$ by replacing p^{n+1} in (3.2a) by p^n. This way we obtain for $\boldsymbol{v}^*$ and $\boldsymbol{\psi}^{n+1}$ the following system

$$\frac{1}{\Delta t}\big(\boldsymbol{v}^* - \boldsymbol{v}^n\big) - \boldsymbol{\mathcal{F}}_1[\boldsymbol{v}^*, \boldsymbol{\psi}^{n+1}] + \rho^{-1}(\boldsymbol{\psi}^{n+1})\nabla p^n = \boldsymbol{0}, \tag{3.3a}$$

$$\frac{1}{\Delta t}\big(h(\boldsymbol{\psi}^{n+1}) - h(\boldsymbol{\psi}^n)\big) - \mathcal{F}_2[\boldsymbol{v}^*, \boldsymbol{\psi}^{n+1}] = 0, \tag{3.3b}$$

$$\frac{1}{\Delta t}\big(\boldsymbol{Y}^{n+1} - \boldsymbol{Y}^n\big) - \boldsymbol{\mathcal{F}}_3[\boldsymbol{v}^*, \boldsymbol{\psi}^{n+1}] = \boldsymbol{0}. \tag{3.3c}$$

Next, we replace the term $\boldsymbol{\mathcal{F}}_1[\boldsymbol{v}^{n+1}, \boldsymbol{\psi}^{n+1}]$ in (3.2a) by its predictor $\boldsymbol{\mathcal{F}}_1[\boldsymbol{v}^*, \boldsymbol{\psi}^{n+1}]$, defining the corrector for $\boldsymbol{v}^{n+1}$. Denoting this by $\boldsymbol{v}^{n+1}$ as well, we have

$$\frac{1}{\Delta t}\big(\boldsymbol{v}^{n+1} - \boldsymbol{v}^n\big) - \boldsymbol{\mathcal{F}}_1[\boldsymbol{v}^*, \boldsymbol{\psi}^{n+1}] + \rho^{-1}(\boldsymbol{\psi}^{n+1})\nabla p^{n+1} = \boldsymbol{0}. \tag{3.4}$$

Subtracting equation (3.3a) from (3.4) we find the relation

$$\frac{1}{\Delta t}\big(\boldsymbol{v}^{n+1} - \boldsymbol{v}^*\big) + \rho^{-1}(\boldsymbol{\psi}^{n+1})\nabla q^n = \boldsymbol{0}, \quad q^n := p^{n+1} - p^n. \tag{3.5}$$

Finally, applying the divergence operator to (3.5) and using the constraint (3.2d), we obtain the following elliptic equation for the pressure increment q^n

$$\nabla\cdot\big(\rho^{-1}(\boldsymbol{\psi}^{n+1})\nabla q^n\big) = \frac{1}{\Delta t}\big(\nabla\cdot\boldsymbol{v}^* - s(\boldsymbol{\psi}^{n+1})\big). \tag{3.6}$$

We will refer to this equation as the *pressure equation*.

Summarizing, we have the following algorithm.

PC-algorithm

1. Solve system (3.3) for the predictor velocity $\boldsymbol{v}^*$ and the combustion variables $\boldsymbol{\psi}^{n+1}$.
2. Solve equation (3.6) for the pressure update q^n.
3. Update the pressure and compute the corrector velocity $\boldsymbol{v}^{n+1}$ from (3.5)

To conclude we note the following. First, the pressure correction method is embedded in the method of false transients to compute the steady solution of (3.1). The time step Δt should be chosen as large as possible to reach the steady solution in as little time steps as possible. However, if Δt is chosen too large, the (iterative) solution procedure for (3.3) might break down because a good initial guess is lacking. Second, for time dependent problems we have to replace the implicit Euler method by a high order implicit time integrator, but otherwise the PC algorithm remains the same. Finally, integrating the pressure equation over the domain Ω and applying Gauss' theorem, we obtain the following constraint on the pressure update

$$\oint_{\partial\Omega}\rho^{-1}\big(\boldsymbol{\psi}^{n+1}\big)\frac{\partial q^n}{\partial \boldsymbol{n}}\,\mathrm{d}S = \frac{1}{\Delta t}\Big(\oint_{\partial\Omega}\boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}S - \int_\Omega s\big(\boldsymbol{\psi}^{n+1}\big)\,\mathrm{d}V\Big), \tag{3.7}$$

where $\boldsymbol{n}$ is the outward unit normal on the boundary $\partial\Omega$ and where we have set $\boldsymbol{v}^* = \boldsymbol{v}$ at the boundary.

4 Local defect correction

In this section we present a brief outline of LDC; for a more detailed account see e.g. [2, 3, 13]. Thus, on a simply connected domain $\Omega \subset \mathbb{R}^d\,(d = 2, 3)$, consider the boundary value problem (BVP)

$$\mathcal{N}[u] = 0, \quad \boldsymbol{x} \in \Omega, \tag{4.1a}$$

$$\mathcal{B}[u] = g, \quad \boldsymbol{x} \in \partial\Omega, \tag{4.1b}$$

where $\mathcal{N}$ is a nonlinear elliptic differential operator and $\mathcal{B}$ a boundary operator, either of Dirichlet or Neumann type. We cover Ω with a uniform coarse grid Ω_H, of grid size H, and apply some discretisation method, resulting in the nonlinear system

$$\mathcal{N}_H[u_H] = f_H, \quad \boldsymbol{x} \in \Omega_H, \tag{4.2}$$

where u_H is the grid function representing all unknowns on Ω_H and where the right-hand side f_H contains the boundary data g. We will not specify the discrete operator $\mathcal{N}_H$; either a finite difference or finite volume approximation is suitable. We assume that (4.2) has a solution. For the iterative solution of (4.2) we have to use (damped) Newton iteration or one of its variants, resulting in an approximate solution u_H^0.

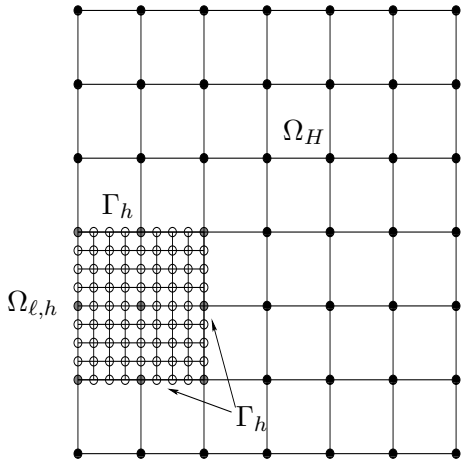

Figure 1: A global coarse and a local fine grid, with a refinement factor $\sigma = 0.25$.

Suppose, the solution u of (4.1) changes very rapidly in a small subdomain of Ω, the so-called *high activity region*, whereas outside this region u is very smooth. In this high activity region, the grid size H is definitely too large to capture the behaviour of u. Therefore we introduce a local subdomain $\Omega_\ell \subset \Omega$ enclosing the high activity region, and cover Ω_ℓ with a local uniform fine grid $\Omega_{\ell,h}$ of grid size $h < H$; see Figure 1. The refinement factor $\sigma := h/H < 1$, but is otherwise arbitrary. We will formulate a discrete BVP on $\Omega_{\ell,h}$ analogous to (4.2), which gives a much more accurate approximation of u in Ω_ℓ. However, we first need the following concepts. Let Γ be the interface between Ω_ℓ and Ω, i.e., $\Gamma := \partial\Omega_\ell \setminus (\partial\Omega_\ell \cap \partial\Omega)$. This implies that $\partial\Omega_\ell$ consists of two parts, viz. Γ and $\partial\Omega_\ell \cap \partial\Omega$. The latter may be empty. On the interface we have the grid $\Gamma_h := \Gamma \cap \Omega_{\ell,h}$. Next we introduce the vector space $\mathcal{F}(\Omega_H)$ of grid functions on Ω_H, and likewise $\mathcal{F}(\Gamma_h)$. Finally, let $\mathcal{P}^{h,H} : \mathcal{F}(\Omega_H) \longrightarrow \mathcal{F}(\Gamma_h)$ be an interpolation operator that maps grid functions on Ω_H onto grid functions on Γ_h. The discrete BVP on $\Omega_{\ell,h}$ consists of the discretisation of the partial differential equation (4.1a) in Ω_ℓ, possibly the boundary condition (4.1b) on $\partial\Omega_\ell \cap \partial\Omega$ and the (discrete) interface condition

$$u = b_{\ell,h} := \mathcal{P}^{h,H}[u^0_H], \quad \boldsymbol{x} \in \Gamma_h. \tag{4.3}$$

It is obvious to use the same discretisation method as in (4.2), however, it is not strictly necessary. In any case, the discrete BVP on $\Omega_{\ell,h}$ reads

$$\mathcal{N}_{\ell,h}\big[u_{\ell,h}; b_{\ell,h}\big] = f_{\ell,h}, \quad \boldsymbol{x} \in \Omega_{\ell,h}, \tag{4.4}$$

where $u_{\ell,h}$ is the grid function of unknowns on $\Omega_{\ell,h}$ and $f_{\ell,h}$ the grid function containing the boundary data g on $\partial\Omega_\ell \cap \partial\Omega$. Note that the dependence of $u_{\ell,h}$ on the coarse grid solution u^0_H is explicitly denoted by the term $b_{\ell,h}$. We assume that (4.4) has a solution, which we compute using (damped) Newton iteration. We denote the fine grid solution thus computed by $u^0_{\ell,h}$.

We will use the fine grid solution $u^0_{\ell,h}$ to improve u^0_H. To this end we have to estimate the local discretisation error of the coarse grid scheme (4.2), which is defined as follows:

$$d^*_H := \mathcal{N}_H[u] - f_H, \tag{4.5}$$

i.e., it is the residual of (4.2) after substitution of the exact solution u. Replacing u in (4.5) by the grid function w_H^i, $i = 0$, defined by

$$w_H^i(\boldsymbol{x}) := \begin{cases} u_H^i(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \Omega_H \setminus \Omega_{\ell,H}, \\ \mathcal{R}^{H,h}[u_{\ell,h}^i](\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \Omega_{\ell,H}, \end{cases} \tag{4.6}$$

where $\Omega_{\ell,H} := \Omega_\ell \cap \Omega_H$ and $\mathcal{R}^{H,h} : \mathcal{F}(\Omega_{\ell,h}) \longrightarrow \mathcal{F}(\Omega_{\ell,H})$ is a restriction operator that maps grid functions on $\Omega_{\ell,h}$ onto grid functions on $\Omega_{\ell,H}$, we obtain the following approximation of the defect

$$d_H^i(\boldsymbol{x}) := \begin{cases} 0 & \text{if } \boldsymbol{x} \in \Omega_H \setminus \Omega_{\ell,H}, \\ \big(\mathcal{N}_H[w_H^i] - f_H\big)(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \Omega_{\ell,H}. \end{cases} \tag{4.7}$$

Once we have computed d_H^i, $i = 0$, we can add it to the right hand side of (4.2), resulting in the system

$$\mathcal{N}_H[u_H] = f_H + d_H^i, \quad \boldsymbol{x} \in \Omega_H. \tag{4.8}$$

We have to solve (4.8) iteratively, to find the approximation u_H^1, which is presumably a better approximation than u_H^0. The coarse grid solution u_H^1 can again be used to determine the interface condition needed to compute a new fine grid solution $u_{\ell,h}^1$; etc.

Summarizing, we have the following iterative method.

two-level LDC algorithm

a. Initialization

   - Solve the coarse grid problem (4.2) for u_H^0.
   - Compute the interface condition (4.3).
   - Solve the fine grid problem (4.4) for $u_{\ell,h}^0$.

b. Iteration, $i = 1, 2, \ldots$

   - Compute the defect d_H^i from (4.6) and (4.7).
   - Solve the updated coarse grid problem (4.8) for u_H^i, i.e., $\mathcal{N}_H[u_H^i] = f_H + d_H^{i-1}$.
   - Compute the interface condition (4.3).
   - Solve the fine grid problem (4.4) for $u_{\ell,h}^i$, i.e., $\mathcal{N}_{\ell,h}\big[u_{\ell,h}^i; \mathcal{P}^{h,H}[u_H^i]\big] = f_{\ell,h}$.

Convergence of this algorithm is usually very fast; typically one or two iterations are needed for convergence; see [3]. If the algorithm has converged, after say j iterations, we take the composite grid function

$$u_{H,h}(\boldsymbol{x}) := \begin{cases} u_H^j(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \Omega_H \setminus \Omega_{\ell,H}, \\ u_{\ell,h}^j(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \Omega_{\ell,h}, \end{cases} \tag{4.9}$$

as solution of (4.1), i.e. $u_{H,h}$ is the fine grid solution in Ω_ℓ and the coarse grid solution outside Ω_ℓ.

To demonstrate the feasibility of LDC, consider the following two-point BVP

$$u'' + f(u) = 0, \quad 0 < x < 1, \tag{4.10a}$$

$$u(0) = u_\ell, \quad u(1) = u_\mathrm{r}, \tag{4.10b}$$

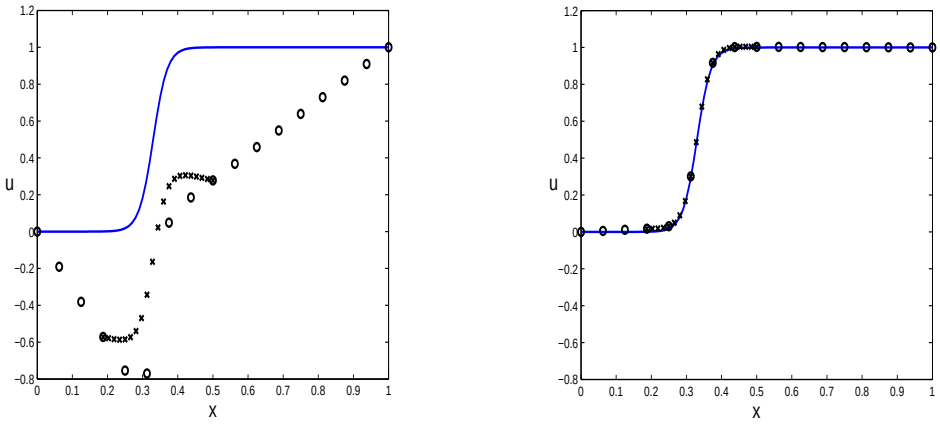

Figure 2: Numerical solution of the two-point BVP (4.10) before (left) and after (right) coarse grid correction.

where the source term $f(u)$ and the boundary data u_ℓ and u_r are chosen such that (4.10) has the exact solution

$$u(x) = \tfrac{1}{2}\big(1 + \tanh(\alpha(x - x_0))\big).$$

We have computed a numerical solution of (4.10) for the parameter values $x_0 = 0.33$ and $\alpha = 25$. In this case the exact solution is characterised by a steep interior layer near $x = x_0$, which requires a fine grid to be resolved. For space discretisation we use the standard central difference scheme. We have computed a global coarse grid solution with $H = 1/16$ and a local fine grid solution on $\Omega_\ell := (0.2, 0.5)$ with $h = 1/64$, the results of which are shown in Figure 2. The left figure shows the results before coarse grid correction. Clearly, the coarse grid solution is poor in the entire domain, and consequently, the interface conditions for the fine grid solution are very inaccurate. The fine grid solution does not really improve the accuracy of the coarse grid solution. However, in Ω_ℓ the *structure* of the fine grid solution is a fair approximation of the interior layer, although the absolute errors are still large. The right figure shows the composite grid solution after one coarse grid correction. Obviously, this solution is much better than the one before coarse grid correction. More LDC-iteration are not needed. Actually, we can show in this case that further LDC-iterations do not improve the solution [14].

5 Multi-level LDC

It is often benificial to introduce several levels of refinement if the BVP to be solved involves many different scales, like the combustion problem in Section 2. Therefore we generalize the LDC algorithm by introducing several nested and increasingly finer grids, covering (part of) the high activity region. Let h_0 denote the grid size of the global coarse grid Ω_{h_0} and $h_1 > h_2 > \cdots > h_{k_\mathrm{max}}$ the grid sizes of the local fine grids Ω_{k,h_k}, $(k = 1, 2, \cdots, k_\mathrm{max})$. Thus, we have to solve the global coarse grid problem

$$\mathcal{N}_{h_0}[u_{h_0}] = f_{h_0}, \quad \boldsymbol{x} \in \Omega_{h_0}, \tag{5.1}$$

and the sequence of fine grid problems

$$\mathcal{N}_{k,h_k}\big[u_{k,h_k}; b_{k,h_k}\big] = f_{k,h_k}, \quad \boldsymbol{x} \in \Omega_{k,h_k}, \quad k = 1, 2, \ldots, k_\mathrm{max}. \tag{5.2}$$

Here u_{k,h_k} is the numerical approximation of u on the kth local fine grid Ω_{k,h_k} of grid h_k and b_{k,h_k} is the value of u_{k,h_k} on the interface between $\Omega_{k-1,h_{k-1}}$ and Ω_{k,h_k}. All fine grid functions and discretisations have a double subscript similar to u_{k,h_k}. In the following we will suppress the subsript k and just write $\mathcal{N}_{h_k}$, u_{h_k}, etc. We denote the initially computed solutions of (5.1) and (5.2) by $u^0_{h_k}$ $(k = 0, 1, \ldots, k_{\max})$. The interface function b_{h_k} has to be computed by some interpolation procedure from the solution one level higher, i.e.,

$$b_{h_k} := \mathcal{P}^{h_k,h_{k-1}}\big[u^0_{h_{k-1}}\big], \quad k = 1, 2, \ldots, k_{\max}. \tag{5.3}$$

Similar to the two-level LDC algorithm, we solve (5.1) and (5.2) on increasingly finer grids and compute the defects d_{h_k} on the grids Ω_{h_k} $(k = 0, 1, \ldots, k_{\max} - 1)$. The defect d_{h_k} depends on the grid functions u_{h_k} and $u_{h_{k+1}}$ and has to be computed analogously to (4.6) and (4.7). After having computed the defects, we apply the defect correction, given by

$$\mathcal{N}_{h_k}[u_{h_k}] = f_{h_k} + d_{h_k}, \quad \boldsymbol{x} \in \Omega_{h_k}, \quad k = k_{\max} - 1, k_{\max} - 2, \ldots, 0. \tag{5.4}$$

The updated solution u_{h_k} $(k = 0, 1, \ldots, k_{\max} - 1)$ can again be used to define interface conditions, compute fine grid solutions, etc.

This procedure gives rise to the following algorithm; see also Figure 5.

Multi-level LDC algorithm

a. Initialization, $i = 0$

   - Solve the basic coarse grid problem (5.1) for u_{h_0}.
   - For $k = 1, 2, \ldots, k_{\max}$ do
     - Compute the interface condition (5.3) for u_{h_k}.
     - Solve the fine grid problem (5.2) for u_{h_k}.

b. Iteration, $i = 1, 2, \ldots$

   - For $k = k_{\max} - 1, k_{\max} - 2, \ldots, 0$ do
     - Compute the defect $d^i_{h_k}$.
     - Solve the updated problem (5.4) for u_{h_k}.
   - For $k = 1, 2, \ldots, k_{\max}$ do
     - Compute the interface condition (5.3) for $u^i_{h_k}$.
     - Solve the fine grid problem (5.2) for $u^i_{h_k}$.

After convergence, we combine the numerical solutions u_{h_k} $(k = 0, 1, \ldots, k_{\max})$ into the composite grid solution, i.e., we take the numerical solution on the finest grid available.

To get an impression of the efficiency of the multi-level LDC algorithm, we estimate the number of control volumes needed to discretise the BVP (4.1) on the domain $\Omega = (0, 1) \times (0, 1)$. The global coarse grid Ω_{h_0} contains $1/h_0^2$ control volumes. Assume that the high activity region is contained in $\Omega_\ell = (0, \ell) \times (0, \ell)$ $(0 < \ell < 1)$. We cover Ω_ℓ with three local fine grids Ω_{h_k} of dimension $\ell_k \times \ell_k$ and of grid size h_k. Clearly, the number of control volumes of the fine grids are $\big(\ell_k/h_k\big)^2$. The total number of control volumes N_{LDC} is thus given by

$$N_{\mathrm{LDC}} = \frac{1}{h_0^2} + \Big(\frac{\ell_1}{h_1}\Big)^2 + \Big(\frac{\ell_2}{h_2}\Big)^2 + \Big(\frac{\ell_3}{h_3}\Big)^2.$$

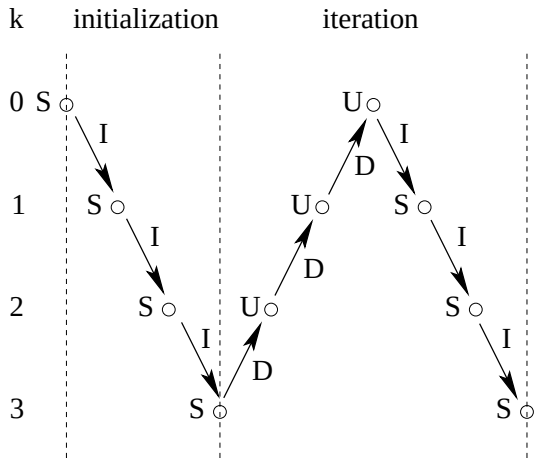

Figure 3: schematic representation of the initialization procedure and one iteration step of the four-level LDC method. The meaning of the identifiers is as follows: S stands for solve discrete system, I for computation interface conditions, D for defect computation and U for solve updated system.

We choose $\ell_3 = \ell$ and $\ell_k = (1+\alpha_k)\ell\,(k=1,2)$ with $0 < \alpha_2 < \alpha_1$, i.e., the fine grids $\Omega_{h_k}\,(k=1,2)$ are slightly larger than Ω_ℓ. Moreover, we adopt the following refinement strategy: $\sigma := h_1/h_0 \leq 0.5$ and $h_k/h_{k-1} = 0.5\,(k=2,3)$. For a global uniform grid having the smallest grid size h_3 we would need $N_{\text{uniform}} = 1/h_3^2$ control volumes. The gain $G := N_{\text{uniform}}/N_{\text{LDC}}$ is then given by

$$\frac{1}{G} = \Big(\frac{\sigma}{4}\Big)^2 + \Big[\Big(\frac{1+\alpha_1}{4}\Big)^2 + \Big(\frac{1+\alpha_2}{2}\Big)^2 + 1\Big]\ell^2.$$

A plot of $G = G(\sigma,\ell)$ as a function of σ and ℓ is given in Figure 5. From this figure we conclude that LDC is especially efficient when $\ell \ll 1$, i.e., when the high activity region is only a small fraction of the entire domain. It is also clear that $G(\sigma,\ell)$ is only weakly dependent on the refinement factor σ.

6 LDC for laminar flames

In this section we will combine the PC-algorithm from Section 3 with the multi-level LDC-algorithm from Section 5. In the PC-algorithm we have to solve two BVPs, i.e., one corresponding to system (3.3), to compute $\boldsymbol{v}^*$, $\boldsymbol{\psi}^{n+1}$, and one corresponding to the elliptic equation (3.6), to compute q^n. We will successively solve both BVPs using the LDC-algorithm.

For space discretisation of (3.3) and (3.6) we use the cell-centred finite volume method on a staggered grid in combination with an exponential scheme for the computation of the numerical fluxes; for more details see [18, 19]. Moreover, for the pressure gradient at the cell boundaries we take the central difference scheme. We apply these discretisation schemes on a global coarse grid Ω_{h_0}, of typical grid size h_0, and on a sequence of local fine grids Ω_{h_k}, of typical grid sizes h_k. This way, we obtain for (3.3) a nonlinear algebraic system on Ω_{h_0}, which can be symbolically written as

$$\boldsymbol{\mathcal{N}}_{h_0}\big[\boldsymbol{v}^*_{h_0}, \boldsymbol{\psi}^{n+1}_{h_0}\big] = \boldsymbol{f}_{h_0}, \quad \boldsymbol{x} \in \Omega_{h_0}, \tag{6.1a}$$

where $\boldsymbol{v}^*_{h_0}$, $\boldsymbol{\psi}^{n+1}_{h_0}$ and $\boldsymbol{f}_{h_0}$ are (vector-valued) grid functions on Ω_{h_0}. Likewise we have on Ω_{h_k} the fine

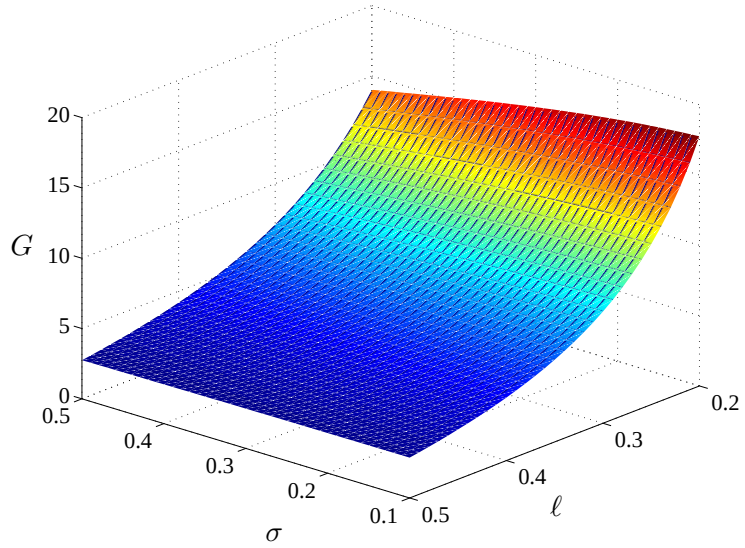

Figure 4: The gain $G = G(\sigma, \ell)$ as a function of σ and ℓ. Parameter values are: $\alpha_1 = 0.2$ and $\alpha_2 = 0.1$

grid discretisations

$$\boldsymbol{\mathcal{N}}_{h_k}\big[\boldsymbol{v}^*_{h_k}, \boldsymbol{\psi}^{n+1}_{h_k}; \boldsymbol{v}^*_{h_{k-1}}, \boldsymbol{\psi}^{n+1}_{h_{k-1}}\big] = \boldsymbol{f}_{h_k}, \quad \boldsymbol{x} \in \Omega_{h_k}, \quad k = 1, 2, \ldots, k_{\max}. \tag{6.1b}$$

Note that we have explicitly denoted the dependence of $\boldsymbol{\mathcal{N}}_{h_k}$ on the grid functions $\boldsymbol{v}^*_{h_{k-1}}$, $\boldsymbol{\psi}^{n+1}_{h_{k-1}}$ at one level lower, without being specific about the interpolation procedure used. After solving the nonlinear systems (6.1) we can compute the defects, which we denote by $\boldsymbol{d}\big[\boldsymbol{v}^*_{h_k}, \boldsymbol{\psi}^{n+1}_{h_k}; \boldsymbol{v}^*_{h_{k+1}}, \boldsymbol{\psi}^{n+1}_{h_{k+1}}\big]$ $(k = 0, 1, \ldots, k_{\max} - 1)$ to indicate the dependence on the numerical solutions on both Ω_{h_k} and $\Omega_{h_{k+1}}$. Discretisation of (3.6) results in $k_{\max} + 1$ linear algebraic systems, which we can write as

$$\mathcal{L}_{h_0}(\boldsymbol{\psi}^{n+1}_{h_0})\big[q^n_{h_0}\big] = g_{h_0}, \quad \boldsymbol{x} \in \Omega_{h_0}, \tag{6.2a}$$

$$\mathcal{L}_{h_k}(\boldsymbol{\psi}^{n+1}_{h_k})\big[q^n_{h_k}; q^n_{h_{k-1}}\big] = g_{h_k}, \quad \boldsymbol{x} \in \Omega_{h_k}, \quad k = 1, 2, \ldots, k_{\max}, \tag{6.2b}$$

where the notation $\mathcal{L}_{h_0}(\boldsymbol{\psi}^{n+1}_{h_0})$ indicates that the linear differential operator $\mathcal{L}_{h_0}$ depends on the combustion variables $\boldsymbol{\psi}^{n+1}_{h_0}$; likewise for $\mathcal{L}_{h_k}$. Next, we compute the defects of the grid functions $q^n_{h_k}$ $(k = 0, 1, \ldots, k_{\max} - 1)$, denoted by $d\big[q^n_{h_k}; q^n_{h_{k+1}}\big]$. Adding all defects to the corresponding systems we obtain

$$\boldsymbol{\mathcal{N}}_{h_k}\big[\boldsymbol{v}^*_{h_k}, \boldsymbol{\psi}^{n+1}_{h_k}; \boldsymbol{v}^*_{h_{k-1}}, \boldsymbol{\psi}^{n+1}_{h_{k-1}}\big] = \boldsymbol{f}_{h_k} + \boldsymbol{d}\big[\boldsymbol{v}^*_{h_k}, \boldsymbol{\psi}^{n+1}_{h_k}; \boldsymbol{v}^*_{h_{k+1}}, \boldsymbol{\psi}^{n+1}_{h_{k+1}}\big], \quad \boldsymbol{x} \in \Omega_{h_k}, \tag{6.3a}$$

$$\mathcal{L}_{h_k}(\boldsymbol{\psi}^{n+1}_{h_k})\big[q^n_{h_k}; q^n_{h_{k-1}}\big] = g_{h_k} + d\big[q^n_{h_k}; q^n_{h_{k+1}}\big], \quad \boldsymbol{x} \in \Omega_{h_k}, \tag{6.3b}$$

from which we can successively compute the corrected solutions at the levels $k = k_{\max} - 1, k_{\max} - 2, \ldots, 0$. These updated solutions are then used to determine interface conditions, compute fine grid solutions, etc.

In the following we supress the superscripts n, $*$ and $n+1$ indicating a time level, in order to keep the notation easy to handle. Thus we have $\boldsymbol{v} = \boldsymbol{v}^*$, $\boldsymbol{\psi} = \boldsymbol{\psi}^{n+1}$ and $q = q^n$. Instead, we add as superscript the iteration level. Combining the PC-algorithm and the LDC-algorithm and putting everything together, we obtain the following algorithm.

Multi-level PC-LDC algorithm

1. Compute the predictor velocity $\boldsymbol{v}$ and the combustion variables $\boldsymbol{\psi}$.

   a. Initialization, $i = 0$
   - Solve the coarse grid problem $\boldsymbol{\mathcal{N}}_{h_0}\big[\boldsymbol{v}^i_{h_0}, \boldsymbol{\psi}^i_{h_0}\big] = \boldsymbol{f}_{h_0}$.
   - For $k = 1, 2, \ldots, k_{\max}$ do
     - Compute the interface conditions for $\boldsymbol{v}^i_{h_k}$ and $\boldsymbol{\psi}^i_{h_k}$.
     - Solve the fine grid problems $\boldsymbol{\mathcal{N}}_{h_k}\big[\boldsymbol{v}^i_{h_k}, \boldsymbol{\psi}^i_{h_k}; \boldsymbol{v}^i_{h_{k-1}}, \boldsymbol{\psi}^i_{h_{k-1}}\big] = \boldsymbol{f}_{h_k}$.

   b. Iteration, $i = 1, 2, \ldots$
   - For $k = k_{\max} - 1, k_{\max} - 2, \ldots, 0$ do
     - Compute the defect $\boldsymbol{d}\big[\boldsymbol{v}^{i-1}_{h_k}, \boldsymbol{\psi}^{i-1}_{h_k}; \boldsymbol{v}^{i-1}_{h_{k+1}}, \boldsymbol{\psi}^{i-1}_{h_{k+1}}\big]$
     - Solve the updated problem $\boldsymbol{\mathcal{N}}_{h_k}\big[\boldsymbol{v}^i_{h_k}, \boldsymbol{\psi}^i_{h_k}; \boldsymbol{v}^i_{h_{k-1}}, \boldsymbol{\psi}^i_{h_{k-1}}\big] = \boldsymbol{f}_{h_k} + \boldsymbol{d}\big[\boldsymbol{v}^{i-1}_{h_k}, \boldsymbol{\psi}^{i-1}_{h_k}; \boldsymbol{v}^{i-1}_{h_{k+1}}, \boldsymbol{\psi}^{i-1}_{h_{k+1}}\big]$.
   - For $k = 1, 2, \ldots, k_{\max}$ do
     - Compute the interface conditions for $\boldsymbol{v}^i_{h_k}$ and $\boldsymbol{\psi}^i_{h_k}$.
     - Solve the fine grid problem $\boldsymbol{\mathcal{N}}_{h_k}\big[\boldsymbol{v}^i_{h_k}, \boldsymbol{\psi}^i_{h_k}; \boldsymbol{v}^i_{h_{k-1}}, \boldsymbol{\psi}^i_{h_{k-1}}\big] = \boldsymbol{f}_{h_k}$.

   c. Converged solution: $\boldsymbol{v}_{h_k}$, $\boldsymbol{\psi}_{h_k}$ $(k = 0, 1, \ldots, k_{\max})$.

2. Compute the pressure increment q.

   a. Initialization, $i = 0$
   - Solve the coarse grid problem $\mathcal{L}_{h_0}(\boldsymbol{\psi}_{h_0})\big[q^i_{h_0}\big] = g_{h_0}$.
   - For $k = 1, 2, \ldots, k_{\max}$ do
     - Compute the interface conditions for $q^i_{h_k}$.
     - Solve the fine grid problem $\mathcal{L}_{h_k}(\boldsymbol{\psi}_{h_k})\big[q^i_{h_k}; q^i_{h_{k-1}}\big] = g_{h_k}$.

   b. Iteration, $i = 1, 2, \ldots$
   - For $k = k_{\max} - 1, k_{\max} - 2, \ldots, 0$ do
     - Compute the defect $d\big[q^{i-1}_{h_k}; q^{i-1}_{h_{k+1}}\big]$
     - Solve the updated problem $\mathcal{L}_{h_k}(\boldsymbol{\psi}_{h_k})\big[q^i_{h_k}\big] = g_{h_k} + d\big[q^{i-1}_{h_k}; q^{i-1}_{h_{k+1}}\big]$.
   - For $k = 1, 2, \ldots, k_{\max}$ do
     - Compute the interface condition for $q^i_{h_k}$.
     - Solve the fine grid problem $\mathcal{L}_{h_k}(\boldsymbol{\psi}_{h_k})\big[q^i_{h_k}; q^i_{h_{k-1}}\big] = g_{h_k}$.

   c. Converged solution: q_{h_k} $(k = 0, 1, \ldots, k_{\max})$.

3. Compute the new pressure and velocity from (3.5).

For the solution of all nonlinear systems we use block-Gauss-Seidel iteration (outer iteration) in combination with quasi-Newton iteration (inner iteration) to compute the variables $\boldsymbol{v}$ and ψ. We take the solution at the previous time level as initial guess for the outer iteration. Moreover, for the inner iteration, we approximate the Jacobi matrices by finite differences and use the solution from the previous Gauss-Seidel sweep as initial guess. The time step is chosen adaptively, such that the solution methods for all algebraic systems do converge. For an appropriate time step, convergence of quasi-Newton iteration is very fast. Typically, one to five iterations are required for convergence, and obviously, most iterations are needed in the flame front. The PC-LDC algorithm is carried out each time step until all (discrete) time derivatives are smaller than 10^{-6}. Finally, to compute interface conditions and defects, we use bilinear interpolation.

7 Numerical results and discussion

We have applied the multi-level PC-LDC algorithm to simulate a two-dimensional methane/air flame on (a section of) a slit burner, the domain of which is shown in Figure 5. The boundary conditions for this problem are as follows:

$$
\begin{array}{lll}
\text{inflow (I)} & u = 0, \quad v = v_0\Big(1 - \Big(\frac{x}{d}\Big)^2\Big), & T = T_{\text{in}}, \quad Y_j = Y_{j,\text{in}}, \\
\text{outflow (O)} & \dfrac{\partial f}{\partial \boldsymbol{n}} = 0 \quad (f = u, v, T, Y_j), & p = p_{\text{amb}}, \\
\text{symmetry (S)} & u = 0, \quad \dfrac{\partial f}{\partial \boldsymbol{n}} = 0 \quad (f = v, T, Y_j), & \\
\text{wall (W)} & u = v = 0, \quad T = T_{\text{wall}}, & \dfrac{\partial Y_j}{\partial \boldsymbol{n}} = 0,
\end{array}
$$

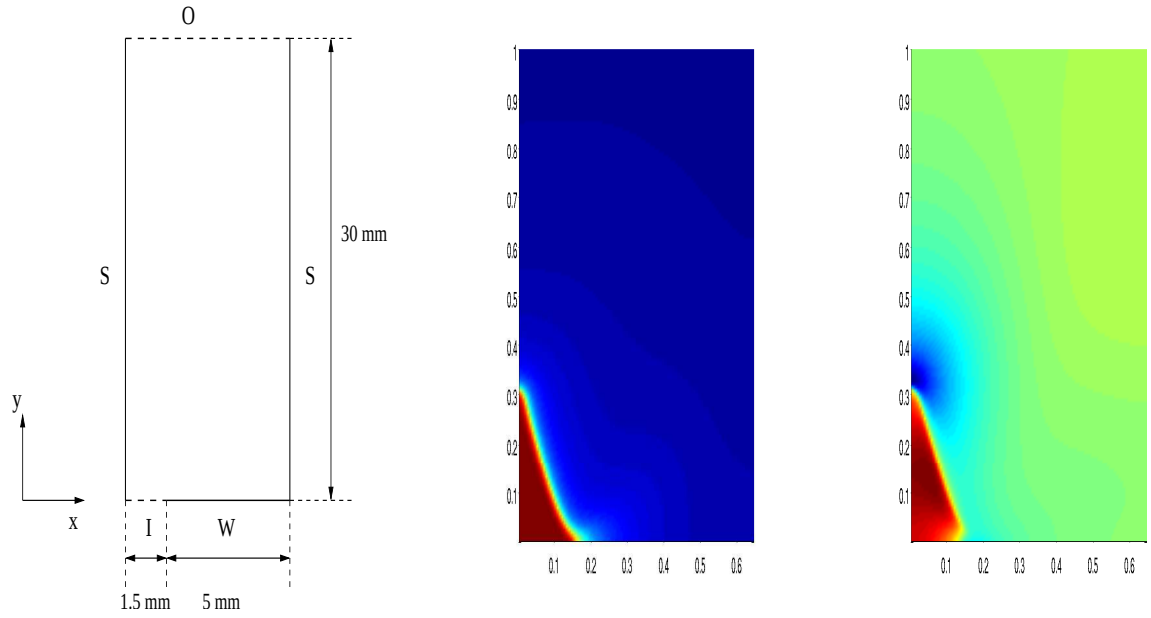

Figure 5: Section of a slit burner: computation domain (left), the reference solution for the methane mass fraction (middle) and the pressure (right), computed on a uniform fine grid.

where $v_0 = 2.6\,\mathrm{m/s}$, $d = 1.5\,\mathrm{mm}$ (width of the inflow opening), $T_{\mathrm{in}} = T_{\mathrm{wall}} = 400\,\mathrm{K}$, $p_{\mathrm{amb}} 1\,\mathrm{atm}$ and where the inlet species mass fractions $Y_{j,\mathrm{in}}$ correspond to an equivalence ratio $\varphi = 0.95$. The one-step model for methane/air combustion is used [22]. Reference solutions for the methane mass fraction and the pressure, computed on a very fine uniform grid of approximately $50,000$ control volumes, are also shown in Figure 5. Note that only the lower part of the computational domain is shown. Clearly, the flame displays features of a Bunsen flame, i.e., a very thin flame front (approximately $0.2\,\mathrm{mm}$ thick) attached to the burner wall and a pressure minimum just behind the flame tip in the burned gas mixture.

Our multi-level LDC method uses a global uniform coarse grid (level 0) of 780 control volumes and three nested local uniform fine grids of 240 (level 1), 720 (level 2) and 2048 (level 3) control volumes, respectively, that tightly enclose the flame front in the lower left corner of the domain. The grid size of the finest local grid (level 3) is the same as the grid size of the global fine grid for the reference solutions. The local fine grids all together only cover approximately 3% of the computational domain. We carried out one LDC-iteration. The computation of the numerical solution on the finest grid is the most time consuming part of the algorithm, since this grid contains the largest number of control volumes. The numerically computed methane mass fraction is shown in Figure 6. From this figure we conclude the following. The global coarse grid solution is fairly accurate in the burned gas mixture, where all variables are slowly varying, however, it clearly does not capture the detailed structure of the flame front. We observe that the flame front is not smooth and it is too thick. Also the flame height is too small. On the other hand, the solution on the finest grid gives an accurate representation of the flame in the flame front. Finally, piecing together all solutions we obtain the composite grid solution, which closely resembles the reference solution. The same conclusions hold for the numerically computed pressure, shown in Figure 7. The composite grid solution is computed on a grid having approximately 13 times less control volumes than the grid for the reference solution. Moreover, the size of the algebraic systems in the LDC algorithm is at least a factor 25 smaller than the corresponding algebraic systems for the

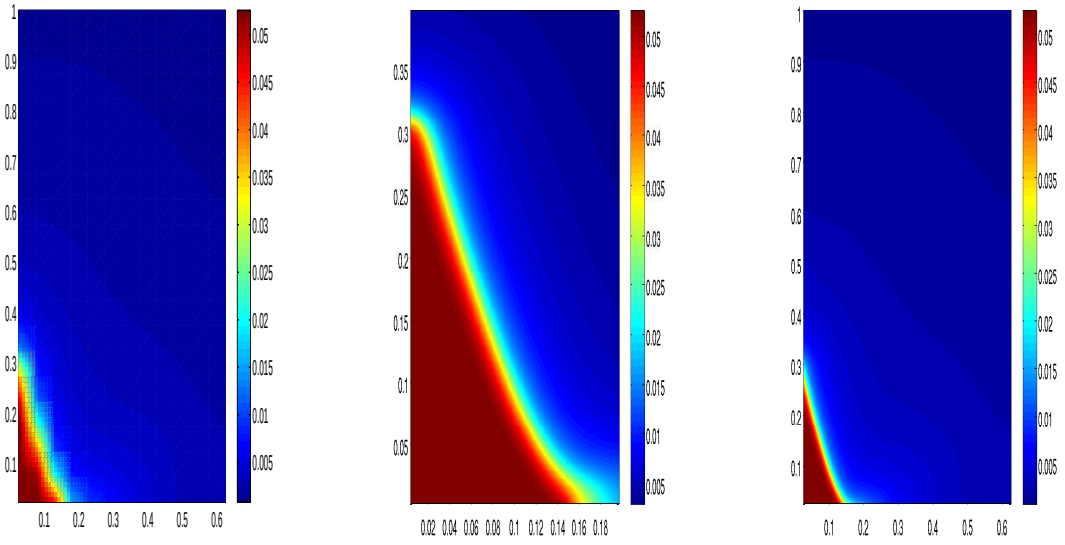

Figure 6: The methane mass fraction on the global coarse grid (left), the local finest grid (middle) and the composite grid (right).

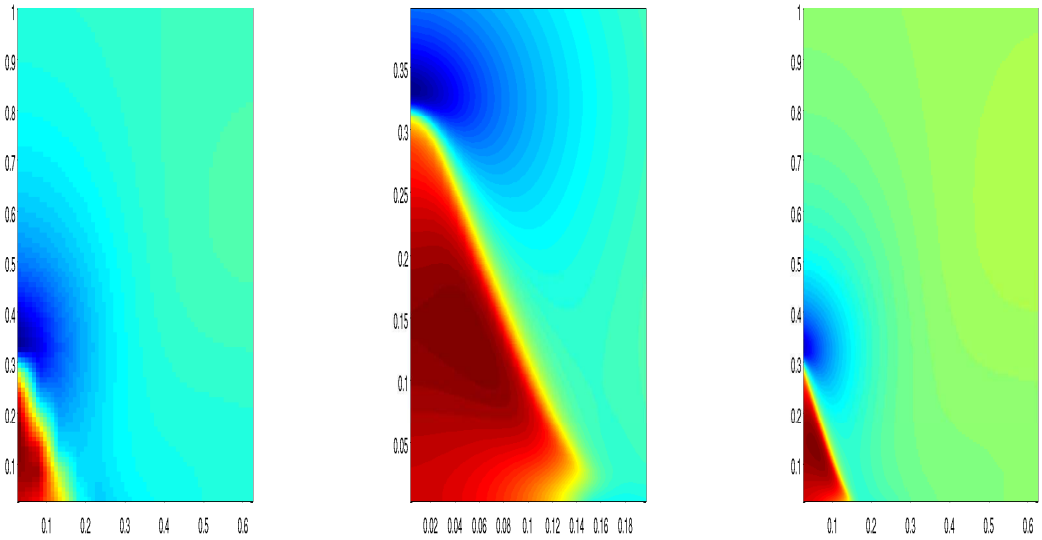

Figure 7: The pressure on the global coarse grid (left), the local finest grid (middle) and the composite grid (right).

reference solution solution, which makes the linear algebra much easier and much faster.

The performance of LDC iteration depends critically on the quality of both the global coarse grid solution and the local fine grid solutions. In a proper application of LDC, the coarse grid size h_0 should be small enough to resolve all scales outside the flame front, however, it is definitely too large for an accurate approximation of the detailed solution in the flame front (preheat zone, inner layer and oxydation layer). This is taken care of by the local fine grids, which should tightly enclose the flame front. If the initial coarse grid solution provides the correct interface condition on Γ_{h_0}, then defect correction is in fact not necessary anymore. This can occur if we choose the refinement region far too wide, which is of course not very wise to do, or if we have an accurate initial solution. In fact, we can prove that LDC iteration applied to the Poisson equation has converged once the interface conditions have converged [2]. On the other hand, when the initial interface condition on Γ_{h_0} is not correct, a defect correction is needed. This might happen, for example, if we have a poor initial guess or when we want to compute solutions close to quenching or blow off. In our computations, we observed that the interface conditions were slightly changed, less than 10% of the coarse grid interface condition, after one defect correction step. Therefore, we believe that one LDC-iteration step is enough. On the other hand, omitting the correction step all together would lead to a slightly inaccurate solution.

To conclude, LDC is a general solution strategy for laminar flames, applicable to different models (simple/complex chemistry and/or transport). It can be combined with any space discretisation method and (iterative) solution method for the resulting algebraic systems. Simple structured grids should be used, allowing accurate discretisations and efficient (iterative) solution methods.

8 Summary and conclusions

In this paper we have combined the PC method, to decouple the velocity and pressure computation, with the LDC method, to solve BVPs characterised by a high activity region. The PC method is based on the expansion equation as constraint, rather than the continuity equation. The basic idea of LDC is to compute a global coarse grid solution and a local fine grid solution, and subsequently use the latter to improve the coarse grid solution in a defect correction manner. The method is extended recursively to include several levels of refinement. Moreover, this procedure can be repeated iteratively, however, usually one defect correction step is sufficient for convergence. The main advantage of this approach is that we can use structured grids, possibly even uniform, allowing accurate discretisation methods resulting in algebraic systems for which efficient iterative solution methods are available.

We have applied a four-level LDC method to compute a methane/air flame on a slit burner. Our LDC solution has the same accuracy as a reference solution computed on a much finer grid, containing 13 times more control volumes.

References

[1] A.S. Algrem, J.B. Bell, P. Colella, L.H. Howell and M. Welcome, *A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations*, J. Comput. Phys., 142 (1998), pp. 1-46.

[2] M.J.H. Anthonissen, *Local Defect Correction Techniques: Analysis and Application to Combustion*, Ph.D. thesis, Eindhoven University of Technology, 2001.

[3] M.J.H. Anthonissen, R.M.M. Mattheij, and J.H.M. ten Thije Boonkkamp, *Convergence analysis of the local defect correction method for diffusion equations*, Numer. Math., 95 (2003), pp. 401-425.

[4] M.J.H. Anthonissen, B.A.V. Bennett, and M.D. Smooke, *An adaptive multilevel local defect correction technique with application to combustion*, Combust. Theory and Modelling, 9 (2005), pp. 273-299.

[5] B.A.V. Bennett and M.D. Smooke, *Local rectangular refinement with application to axisymmetric laminar flames*, Combust. Theory and Modelling, 2 (1998), pp. 221-258.

[6] B.A.V. Bennett and M.D. Smooke, *Local rectangular refinement with application to nonreacting and reacting fluid flow problems*, J. Comput. Phys., 151 (1999), pp. 684-727.

[7] M. Braack and A. Ern, *Coupling multimodeling with local mesh refinement for the numerical computation of laminar flames*, Combust. Theory and Modelling, 8 (2004), pp. 771-788.

[8] J.D. Buckmaster, *An introduction to combustion theory*, in: The Mathematics of Combustion, J.D. Buckmaster ed., SIAM, Philadelphia, 1985, pp. 3-46.

[9] E. Burman, A. Ern and V. Giovangigli, *Bunsen flame simulation by finite elements on adaptively refined, unstructured triangulations*, Combust. Theory and Modelling, 8 (2004), pp. 65-84.

[10] M.S. Day and J.B. bell, *Numerical simulation of laminar reacting flows with complex chemistry*, Combust. Theory and Modelling, 4 (2000), pp.535-556.

[11] A. Ern and V. Giovangigli, *Multicomponent Transport Algorithms*, Springer, Heidelberg, 1994.

[12] GRI Mech 3.0: `www.me.berkley.edu/gri_mech/version30/text30.html`.

[13] P.J.J. Ferket, *Solving Boundary Value Problems on Composite Grids with an Application to Combustion*, Ph.D. thesis, Eindhoven University of Technology, 1996.

[14] P.J.J. Ferket, *Coupling of a global coarse grid discretization and local fine grid discretizations*, in: Numerical Treatment of Coupled Systems, Notes on Numerical Fluid Mechanics, Vol. 51, W. Hackbusch and G. Wittum eds., Vieweg, braunschweig, 1995

[15] M.Graziadei, R.M.M. Mattheij, and J.H.M. ten Thije Boonkkamp, *Local defect correction with slanting grids*, Numer. Meth. for Part. Diff. Eq., 20 (2004), pp. 1-17.

[16] M.Graziadei, and J.H.M. ten Thije Boonkkamp, *Local defect correction for laminar flame simulation*, in: Progress in Industrial Mathematics at ECMI 2004, A.Di Bucchianico, R.M.M. Mattheij and M.A. Peletier eds., Spinger, Berlin, 2006.

[17] J. van Kan, *A second-order accurate pressure correction scheme for viscous incompressible flow*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 870-891.

[18] B. van 't Hof, J.H.M. ten Thije Boonkkamp and R.M.M. Mattheij, *Discretization of the stationary convection-diffusion-reaction equation*, Numer. Meth. for Part. Diff. Eq., 14 (1998), pp. 607-625.

[19] B. van 't Hof, *Numerical Aspects of Laminar Flame Simulation*, Ph.D. thesis, Eindhoven University of Technology, 1998.

[20] B. van 't Hof, J.H.M. ten Thije Boonkkamp and R.M.M. Mattheij, *Pressure correction for laminar combustion simulation*, Combust. Sci. and Tech., 149 (1999), pp. 201-223.

[21] W. Hundsdorfer, and J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer, Berlin, 2003

[22] R. de Lange, *Modelling of Premixed Laminar Flames*, Ph.D. thesis, Eindhoven University of Technology, 1992.

[23] R.M.M. Mattheij, S.W. Rienstra and J.H.M. ten Thije Boonkkamp, *Partial Differential Equations, Modeling, Analysis, Computation*, SIAM, Philadelphia, 2005

[24] H.N. Najm and O.M. Knio, *Modeling low-Mach number reacting flow with detailed chemistry and transport*, J. Sci. Comput., 25 (2005), pp. 263-287.

[25] R.B. Pember, L.H. Howell, J.B. Bell, P. Colella, W.Y. Crutchfield, W.A. Fiveland and J.P. Jessee, *An adaptive projection method for unsteady, low-Mach number combustion*, Combust. Sci and Tech., 140 (1998), pp. 123-168.

[26] T. Poinsot and D. Veynante, *Theoretical and Numerical Combustion, Second Edition*, Edwards, Philadelphia, 2005.

[27] A. Majda and K.G. Lamb, *Simplified equations for low-Mach number combustion with strong heat release*, in: Dynamical Issues in Combustion Theory, Eds. P.C. Fife, A. Liñan and F.A. Williams, Springer, New York, 1991, pp. 167-211.

[28] P. Wesseling, *Principles of Computational Fluid Dynamics*, Springer, Berlin, 2000.

[29] F.A. Williams, *Combustion Theory. The Fundamental Theory of Chemically Reacting Flow Systems*, Addison-Weseley, Redwood City, 1985.