# Numerical simulation of cooling gas injection using adaptive multiscale techniques

Wolfgang Dahmen, Thomas Gotzen*, Silvia Sorana Melian-Flamand,
Siegfried Müller

*Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55,
52056 Aachen*

---

## Abstract

The interaction of a jet of cooling gas injected through single boreholes or slots with a supersonic flow field plays an essential role in the design of innovative cooling systems. In order to appropriately resolve the physically relevant effects an efficient and reliable solver is needed. For this purpose, a fully adaptive and parallel multiresolution finite volume scheme is used. Here grid adaptation is based on a multiscale decomposition of the discrete flow data in terms of cell averages at hand. The resulting multiscale data are compressed by means of hard thresholding. From the remaining data a locally refined grid is constructed.

For validation purposes, first a simplified two-dimensional configuration mimicking the gas injection through a slot of infinite length in a plate is investigated. These numerical results are thoroughly validated by van Driest's self-similar solution for laminar boundary layers and computations performed with non-adaptive codes. In the following, three-dimensional simulations of slots of finite length are carried out and compared with experimental data. The influence of varying rate of cooling gas, injection angle etc. on the cooling efficiency is investigated.

*Keywords:* Finite Volume Method, Film cooling, Cooling gas injection, Multiscale techniques, Grid adaptation

---

## 1. Introduction

In combustion chambers of rocket engines the walls are exposed to very high temperatures. In order to avoid material damage one has to resort to well-designed thermal protection system. Another application area for such systems are supersonic reentry processes. One widely-used cooling concept for problems of this type is *radiation cooling* in combination with materials with very low thermal conductivity. One disadvantage of such passive techniques is that they

---

*Corresponding author
*Email address:* `gotzen@igpm.rwth-aachen.de` (Thomas Gotzen)

cannot be adjusted during the flight and therefore must be designed to withstand the highest thermal load occurring in the course of the flight trajectory. Instead active cooling strategies such as film cooling might offer a promising more economical alternative to reducing the heat load of the wall. The basic idea of film cooling is to inject cooling fluid, commonly air, through boreholes or slots in the wall surface such that a thin film develops at the wall. Thus, the high temperature gas does not come into direct contact with the surface and the heat load at the wall is reduced. So far these concepts have been used primarily for subsonic applications. For example, film cooling is widely used in subsonic flow regimes for cooling turbine blades. An overview of previous experimental studies and theoretical models is given by Goldstein [1]. These investigations deal mostly with free stream Mach numbers in the range of 0.02 to 0.2]. The interaction of the injected cooling gas with the main flow field has been studied for slot injection, cf. Taslim [2] and Fitt [3] as well as for shaped holes, cf. Thole [4]. According to the characteristics of turbine flows, these studies address subsonic, turbulent flow conditions. As a typical adverse effect, the cooling gas injection leads to three-dimensional vortex structures, which in general reduce the cooling efficiency locally. The studies mentioned above focus on reducing this effects.

Only very recently, active cooling strategies have been developed also for *supersonic* flows. Film cooling for supersonic flows is, for instance, considered in scramjet applications. Here the flows are also turbulent and tangential injection is the preferred choice, cf. Konopka [5].

Gollnick [6] focuses on tangential injection into a supersonic, *laminar* flow field with different types of cooling gases. Near the leading edge of a reentry vehicle, for example, supersonic laminar flow conditions are encountered in the region of the critical thermal loads. In these applications, the blowing ratio, i.e., the ratio of the specific mass flux of the cooling gas compared to the specific free stream mass flux, is lower than for turbine blade cooling. The Mach number is typically between 2 and 3.

Non-tangential cooling gas injection into laminar, supersonic flows has been recently investigated experimentally by Heufer & Olivier [8] focusing on slot injection in a Mach-2.6 boundary layer with different blowing ratios and angles. These experiments play on one hand an important role for the validation of the simulation tools presented in this paper. On the other hand, our simulations will be seen to refine the information obtained from these experiments. Furthermore, Linn & Kloker [9] performed numerical simulations of injection through rows of holes with the same Mach number. The latter also investigated the injection through aligned and staggered rows of holes into a Mach-6 boundary layer [10]. However, in summary, it is fair to say that one still faces a severe lack of understanding concerning the fundamentals of the flow field and the thermal interaction.

In the present work we focus on the numerical simulation of cooling gas injected into a laminar, supersonic flow field through a finite single slot in a plate. For this configuration experimental data are available. The main purpose is to resolve the three-dimensional flow field generated by the finite slot and understand the behavior of the flow, in particular, the effects triggered at the ends of the slot. More specifically, the vortex system generated at both ends of the slot are not sufficiently resolved by experiments, primarily because of its small extent in spanwise direction. We shall show that numerical simulations provide more detailed information on these vortices.

We employ the adaptive and parallel solver Quadflow [11] for our simulations. This solver has been designed as an integrated tool in a way that each of its constituents, namely the type of discretization, grid generation and handling, as well as grid adaptation support each other to a possibly large extent. Specifically, the core ingredients are: (i) the flow solver concept based on a finite volume discretization, (ii) the grid adaptation concept based on wavelet techniques, and (iii) the grid generator based on B-spline mappings. In particular, the three constituents are not just viewed as black boxes communicating only via interfaces.

In this context, the mathematical concept of multiresolution-based grid adaption plays a central role in that it reliably detects all physical relevant effects and resolves them reasonably. First work in this regard has been published by Harten [12]. The basic idea is to perform a multiscale analysis of a sequence of cell averages associated with any given finite volume discretization on a given highest level of resolution (reference mesh). This results in cell averages on some coarse level and the fine scale information is encoded in arrays of detail coefficients of ascending resolution. In Harten's original approach, the multiscale analysis is used to steer a hybrid flux computation that resorts to cheap finite differences whenever the detail coefficients indicate that it is save to do so. This results in *quantitative* savings of CPU time. However, this strategy does *not* reduce the overall *complexity rate* because the computational work still stays proportional to the number of cells of the uniform high resolution reference mesh, ultimately rendering this approach insufficient for complex 3D computations. In contrast to this strategy, in [13, 14] threshold techniques are applied to the multiresolution decomposition where detail coefficients below a threshold value are discarded. By means of the remaining significant details, a locally refined mesh is determined whose complexity is substantially reduced in comparison to the underlying reference mesh.

The fully adaptive concept has proven to be highly efficient and reliable. So far, it has been employed with great success to different application scenarios, e.g., 2D/3D-steady and unsteady computations of compressible fluids around airfoils modeled by the Euler and Navier-Stokes equations, respectively, on block-structured curvilinear grid patches [11], backward-facing step flows on 2D triangulations [15], and simulation of a flame ball modeled by reaction-diffusion equations on 3D Cartesian grids [16]. By now, there is an increasing number of groups working on this subject: Postel et al. [17], Schneider et al. [18], Burger et al. [19], Domingues et al. [20], Duarte et al. [21], and Koumoutsakos et al. [22]. For a more thorough discussion on adaptive multiresolution finite volume schemes we refer to the recent review [23] and the articles in [24].

The present paper is structured as follows: First, the basic concepts of the flow solver Quadflow concerning grid adaptation, grid generation, finite volume discretization and parallelization are briefly summarized in Section 2. In Section 3 the solver is validated by means of a supersonic flow over a flat plate without injection. This allows us to compare the numerical results with van Driest's [27] self-similar solution. Here the reliability and efficiency of the grid adaptation are of in particular interest. Finally, in Section 4 we apply the solver to the problem of cooling gas injection where a gas is injected through (i) one infinite slot in the plate into a two-dimensional flow field and (ii) through a finite slot into a three-dimensional flow field. Specifically, we investigate the interaction of the cooling gas jet with the laminar boundary layer and the supersonic flow field. We conclude with a summary of the main results and give an outlook to future work.

## 2. Numerical Method

For all computations reported in this article we have employed the adaptive flow solver Quadflow, cf. [11]. This solver has been developed over a period of more than one decade within the collaborative research center SFB 401 *Modulation of Flow and Fluid-Structure Interaction at Airplane Wings* [28], [29]. It solves the compressible Euler and Navier-Stokes equations around complex aerodynamic configurations by a cell-centered finite volume method on locally refined grids. Mesh adaptation is based on multiscale analysis as opposed to of classical gradient- or residual-based error estimators. To deal with complex geometries the computational grids are represented by block-structured parametric B-Spline patches. In order to reduce the computational load in terms of CPU time and memory to an acceptable amount, these tools are complemented by parallelization techniques based on space-filling curves which allows us to run the simulations on distributed memory architectures.

The central objective is to realize adaptively generated discretizations that are able to resolve the physically relevant phenomena at the expense of possibly few degrees of freedom and correspondingly reduced storage demands. This requires a careful coordination of the core ingredients namely the discretization of the underlying system of partial differential equations, the generation and management of suitable meshes and the adaptation mechanism. In the following we briefly summarize the underlying equations and the basic ingredients of the Quadflow solver.

### *2.1. Governing Equations*

Laminar viscous fluid flow is described by the Navier–Stokes equations for a compressible gas. Neglecting body forces and volume supply of energy, the conservation laws for any control volume $V$ with boundary $\partial V$ and outward unit normal vector $\mathbf{n}$ on the surface element $dS \subset \partial V$ can be written in integral form as:

$$\frac{\partial}{\partial t}\int_V \mathbf{u}\,dV + \oint_{\partial V} \left(\mathbf{F}^c(\mathbf{u}) - \mathbf{F}^d(\mathbf{u})\right)\mathbf{n}\,dS = \mathbf{0}\,. \tag{1}$$

To complete the posed problem initial values $\mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x})$, $\mathbf{x} \in V \subset \mathbb{R}^d, d \in \{2, 3\}$, and boundary conditions $\mathbf{u}(\mathbf{x}, t)|_{\Gamma(t)} = B(\mathbf{x}, t)$, $\mathbf{x} \in \Gamma(t)$ are to be supplemented, where $\Gamma(t)$ is a subset of $\partial V$, determined by the flow characteristics at time $t$
Here $\mathbf{u} = (\rho,\, \rho\mathbf{v},\, \rho E)^T$ denotes the vector of the unknown conserved quantities. Moreover, $\mathbf{F}^c$ and $\mathbf{F}^d$ represent the convective flux including pressure and the diffusive flux function, respectively:

$$\mathbf{F}^c = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \circ \mathbf{v} + p\,\mathbf{I} \\ \rho\mathbf{v}(E + p/\rho) \end{pmatrix}, \quad \mathbf{F}^d = \begin{pmatrix} 0 \\ \tau^v \\ \mathbf{v}\tau^v - \mathbf{q} \end{pmatrix}, \tag{2}$$

where $\rho$ denotes the density, $p$ the static pressure, $\mathbf{v}$ the velocity vector of the fluid, $E$ the total energy, and the symbol $\circ$ indicates the dyadic product. The viscous stress tensor $\tau^v$ for an isentropic Newtonian fluid is defined as

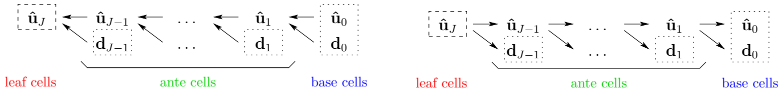
$$\tau^v = \mu\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right) - \frac{2}{3}\mu\,(\nabla \cdot \mathbf{v})\,\mathbf{I}\,. \tag{3}$$

Heat conduction is modeled by Fourier's law $\mathbf{q} = -\kappa\nabla T$, where the thermal conductivity is assumed as $\kappa = c_p\mu/Pr$, with Prandtl number $Pr = 0.72$. The variation of the dynamic viscosity $\mu$ as a function of temperature is determined by Sutherland's law. The static pressure is related to the specific internal energy according to the equation of state for a perfect gas $p = \rho(\gamma - 1)\left(E - 1/2\,|\mathbf{v}|^2\right)$, where $\gamma$ is the ratio of specific heats, which is 1.4 for air.

### *2.2. Adaptation*

The main distinction from previous work regarding adaptation lies in the fact that we employ here recent *multiresolution techniques* based on biorthogonal wavelets, cf. [12, 30]. The starting point is to transform the arrays of cell averages associated with any given finite volume discretization into a different format that reveals insight into the characteristic contributions of the solution to different length scales. The cell averages on a given highest level of resolution $l = L$ are represented as cell averages on some coarse level $l = 0$, while the intermediate fine scale information is encoded in arrays of *detail coefficients* of ascending resolution $l = 0, \ldots, L-1$. In Figure 1(a) the multiscale transformation is sketched. This requires a *hierarchy of meshes* as exemplified by Figure 2(a) for a Cartesian grid hierarchy.

The multiscale representation is used to create *locally refined meshes* proceeding in the following three steps, cf. [13]:
Step 1. Due to the cancellation property of biorthogonal wavelets the details may become small, if the underlying data are locally smooth. Therefore, quite in the spirit of image compression, we may compress the vector of details by means of hard thresholding, i.e., we discard all detail coefficients whose absolute values fall below a level-dependent threshold $\varepsilon_l = 2^{(l-L)d}\varepsilon$ for a suitable parameter $\varepsilon$.



(a) (b)

Figure 1: Pyramid scheme of multiscale (a) and inverse multiscale (b) transformation.

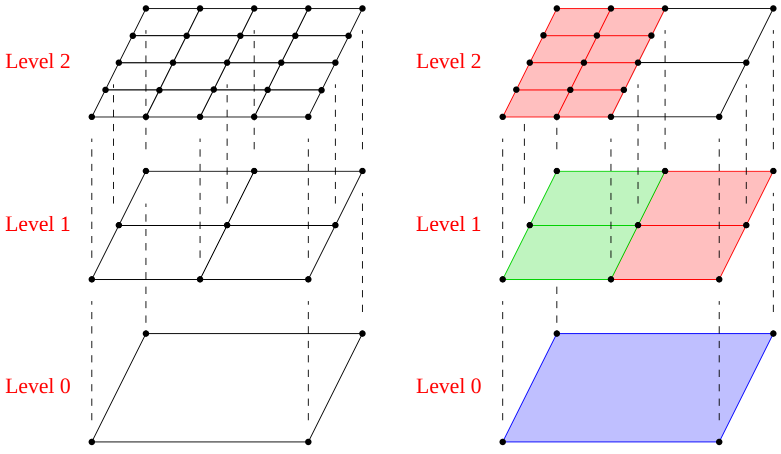
Step 2. In order to account for the dynamics of a flow field, due to the time evolution, and to appropriately resolve all physical effects on the new time level, this set is to be inflated somewhat such that the resulting *prediction set* contains all $\varepsilon$-significant details of the old *and* the new time level. The prediction strategy depends on the underlying system of evolution equations to be approximated. Here we use Harten's heuristic prediction strategy [12].
Step 3. From the significant details we finally construct the locally refined grid and corresponding cell averages. For this purpose, we proceed levelwise from coarse to fine, see Figure 2(b), checking for all cells of a level whether there exists a significant detail. If there is one, we refine the respective cell, i.e., we replace the average of this cell by the averages of its children by locally applying the inverse multiscale transformation, see Figure 1(b).

A few comments on some distinctions from pure Cauchy problems with compactly supported initial data are in order. Since the dynamics of the flow are triggered by the free stream conditions the first time step is executed on a uniform refinement of the initial grid on level $l = 1$ and not just on level $l = 0$ itself, in order to resolve the relevant details. We postpone more details about subsequent refinement strategies to the numerical experiments below.

From a mathematical point of view a principal objective is to extract the inherent complexity of the problem by placing as few degrees of freedom so as to still capture the features of the searched for solution within a given tolerance. A central mathematical problem is then to show that the essential information to be propagated in time is still kept with sufficient accuracy when working on locally coarser meshes, cf. [14, 31]. A detailed derivation and analysis of the multiscale-based grid adaptation concept can be found in the monograph [13]. An overview on recent developments and an extended list of related work is given in [23].

### *2.3. Mesh Generation*

The adaptation strategy gives rise to locally refined meshes of quadtree respectively octree type. The second important ingredient is the generation of such meshes along with the information needed by the flow solver at any stage of a dynamical calculation. A key idea is to represent such meshes through *parametric mappings* with as few parameters - control points - as possible so that further successive refinements just boil down to evaluating these mappings and thus can be efficiently computed based on the knowledge of the few control points. This is of vital importance with regard to (geometrically) non-stationary



(a) (b)

Figure 2: Hierarchy of nested Cartesian grids (a) and grid adaptation procedure in case of Cartesian grids (b)

processes. Roughly speaking, the mesh in each block results from evaluating a parametric mapping from the computational domain into the physical domain, see Figure 3. Such mappings can be based on B-spline representations in combination with well established concepts from CAGD (computer aided geometric design). The quantities to be updated in time are the relatively few control parameters in those parametric representations, while mesh points on any level of resolution can be efficiently computed due to the locality of the B-spline representation. The fact that one needs indeed only relatively few control points in order to generate meshes of good quality is partly due to the variation diminishing property of B-splines. To deal with complex geometries, we embed this concept in a multi-block topology, where in each block we generate a B-spline mapping, see Figure 4. For the realization of B-spline mesh generation on multiblock topologies we refer to the thesis of Lamby [32]. Some recent developments are presented in [33].

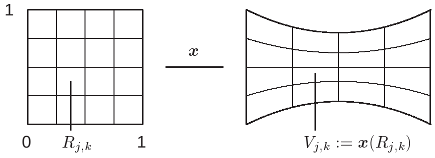


Figure 3: Parametric Mappings

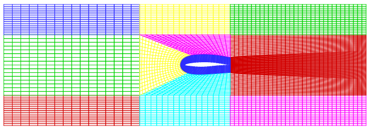


Figure 4: Multi-block topology

### *2.4. Flow Solver*

The discretization scheme has to meet the requirements of the adaptation concept and has to fit well with the mesh generation. This requires the development of a finite volume scheme for fairly general cell partitions that can cope, in particular, with hanging nodes and possible unstructured parts in complicated regions of the flow domain. For this purpose, the locally adapted grid is treated as a fully unstructured mesh with arbitrary polygonal/polyhedral control volumes in two and three space dimensions, respectively. The convective fluxes are determined by solving quasi–onedimensional Riemann problems at the cell interfaces. Several approximate Riemann solvers (Roe, HLLC, AUSMDV) and upwind schemes (van Leer) have been incorporated, cf. [35]. A linear, multidimensional reconstruction of the conservative variables is applied to increase the spatial accuracy. In order to avoid oscillations in the vicinity of local extrema and discontinuities, limiters with TVD property are used such as the Venkatakrishnan limiter. Concerning the computation of the viscous fluxes, the gradients of the variables at cell interfaces are determined using the divergence theorem. Finally, the time–integration is executed by an explicit multistage Runge–Kutta scheme and a fully implicit Newton–Krylov type method in case of unsteady and steady state computations, respectively. Details on the implementation can be found in the thesis of Bramkamp [34]. An overview of the latest developments is given in [36].

### *2.5. Parallelization*

Although multiscale-based grid adaptation leads to a significant reduction of the computational complexity (CPU time and memory) in comparison to computing on uniform meshes, this by itself is ultimately not sufficient to warrant an acceptable efficiency when dealing with realistic 3D computations for complex geometries. In addition, parallelization techniques are indispensable for further reducing the computational time to an affordable order of magnitude.

The performance of a parallelized code crucially depends on load-balancing and minimal interprocessor communication. Since due to hanging nodes, the underlying adaptive grids are unstructured this task is by no means trivial. Instead of employing graph-partitioning methods, we use *space-filling curves*, see e.g. [37]. Here the basic idea is to map level-dependent multiindices identifying the cells in a grid hierarchy to a one-dimensional line. The interval is then split into different parts each containing approximately the same number of entries. An example is presented in Figure 5. For more details on the parallelization concept and its realization we refer to [38].

## 3. Validation: Supersonic flow over a flat plate

The flow solver Quadflow has been primarily developed for the simulation of transonic flow fields around airfoils, cf. [36]. In order to perform numerical simulations of cooling gas injection into a supersonic flow field, we need to validate the solver thoroughly by means of classical benchmark problems. For

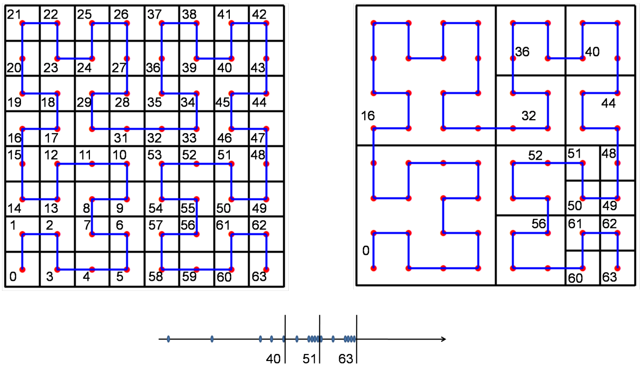


Figure 5: Encoding of the Hilbert order for a three-level adaptive grid with the corresponding split of the unit interval to three processors.

this purpose, we first consider the supersonic flow over a 2D flat plate and compare the results with van Driest's self-similar solution [27].

### *3.1. Computational Setup*

First of all, we consider a supersonic flow over a flat plate of length $200\,mm$, where the plate thickness in the computation is assumed to be zero. The free stream conditions at the leading edge of the plate are characterized by the Mach number $Ma_e = 2.6$, the Reynolds number $Re_e = 4.3 \cdot 10^6$ and the temperature $T_e = 488\,K$. These will also be used later on in Section 4, where in addition a cooling gas is injected through a slot in the plate. Since the flow field is supposed to be laminar, the computations are presented in two space dimensions only. Note that these 2D results coincide with slices of 3D computations, that have been carried out as well for consistency reasons.

The flow field is characterized by a laminar boundary layer and an isotropic compression wave emanating at the leading edge of the plate. These physical effects have to be adequately resolved by locally anisotropic and isotropic grids, respectively. To account for both opposing requirements, we therefore split the computational domain into basically two blocks $\Omega_B = [-0.01\,m, 0.2\,m] \times [0\,m, \delta]$ and $\Omega_F = [-0.01\;m, 0.2\;m] \times [\delta, 0.15\;m]$ for the offset at the plate and the far field, respectively. Here the offset $\delta = 0.005\,m$ is about four times the analytical prediction, according to van Driest [27], of the boundary layer thickness at the end of the plate.

Due to the high Reynolds number, the boundary layer is very thin. To resolve this thin layer by dyadic, isotropic grid refinement would require a very high number of refinement levels resulting in a tremendously large number of cells. This can be avoided to some extent by concentrating grid lines towards the wall resulting in stretched cells. On the other hand, the leading edge of the boundary layer needs to be resolved locally by an isotropic grid to capture the compression wave emanating at this point. To compensate for the stretching in wall-normal direction, we also have to narrow grid lines in tangential direction at the leading edge. In addition, it 5 out that the flow field in front of the plate has to be discretized as well to resolve the physical effects at the leading edge. Therefore the offset block $\Omega_B$ is split into two sub-blocks $\Omega_{B,1} = [-0.01\,m, 0\,m] \times [0\,m, \delta]$ in front of the plate and $\Omega_{B,2} = [0\;m, 0.2\;m] \times [0\;m, \delta]$ at the plate, where in each block a B-spline mapping is constructed. In order to concentrate grid lines at the wall and at the leading edge of the plate ($x = 0\,m$), we apply stretching functions to the parameter spaces of the B-Spline functions corresponding to the tangential and wall-normal direction, respectively. In the far field, i.e., block $\Omega_F$, a Cartesian mesh is used.

In front of the flat plate, we impose symmetric boundary conditions, i.e.,

$$\frac{\partial \rho}{\partial n} = 0,\ \frac{\partial u}{\partial n} = 0,\ v = 0,\ \frac{\partial p}{\partial n} = 0, \tag{4}$$

for the density $\rho$, the velocity components $u$ and $v$ in $x$- and $y$-direction, respectively, and for the pressure $p$. On the surface of the plate, we prescribe no-slip conditions and a constant wall temperature of $T_w = 293\,K$ modeling an impermeable and isothermal wall.

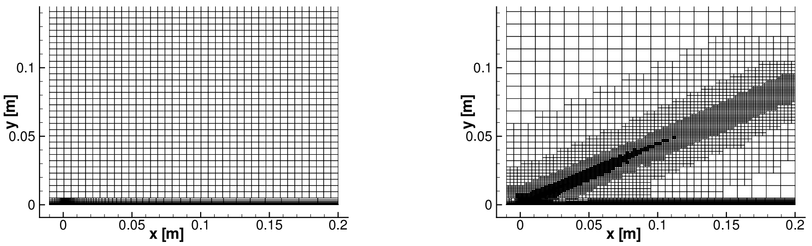
The coarse grid discretization corresponding to level $l = 0$ is comprised of 605 cells. Note that the current implementation of the no-slip conditions requires resolving all cells attached to the wall by the highest refinement level in order to avoid the occurrence of hanging nodes, i.e., grid adaptation is suspended in these cells. Due to grading, the resolution may decrease gradually in wall-normal direction. This can be seen in Fig. 6(a). As mentioned earlier, since the flow field is initialized by the homogeneous free stream conditions, the computation is started on the uniformly refined mesh corresponding to level $l = 1$, see Fig. 6(a). In the course of the computation grid adaptation is applied whenever the averaged density residual has dropped by 4 orders of magnitude for five adaptations. Afterwards, computation continues until a residual drop of $10^{-5}$ is reached. Note that the first four adaptation steps are only used to generate an initial guess for the iteration on the final adaptive grid. Therefore these have not to be fully converged in time. Since in each adaptation step an additional refinement level can be introduced, the final adaptive grid corresponds to $L = 5$. Due to the steady state character of the solution, we employ an implicit time discretization by a backward Euler step. Moreover, the time step is chosen locally with respect to a varying global CFL number determined by the following CFL evolution strategy

$$CFL_{k+1} = min(CFL_{min} \cdot 1.05^k, CFL_{max}), \tag{5}$$

where $CFL_{min}$ and $CFL_{max}$ are set to 1 and 10000, respectively. Here the index $k$ enumerates the number of time steps since the last adaptation, i.e., after each adaptation the CFL number is again set to $CFL_{min}$. For solving the Riemann problem at the cell interfaces we use the van Leer solver. The viscosity is calculated by Sutherland's law

$$\mu(T) = \mu_e \left(\frac{T}{T_e}\right)^{3/2} \frac{T_e + S}{T + S} \tag{6}$$

with the Sutherland constant $S = 110\,K$. The wall-temperature ratio is given as $T_w/T_e = 1.665$.



(a) (b)

Figure 6: Initial grid (uniform 1st refinement level) (a) and final locally refined grid (b).

*3.2. Numerical Results*

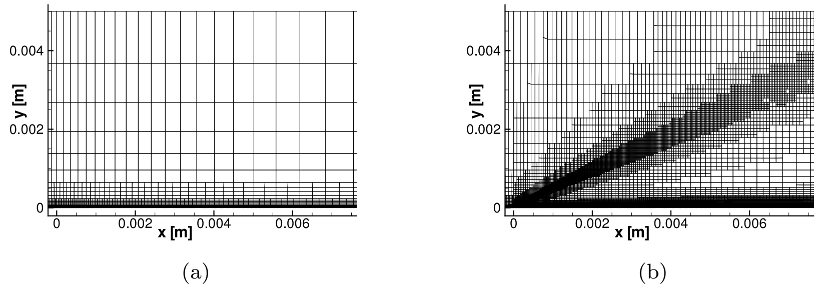
As explained above, starting from a homogeneous flow field, the initial grid, see Fig. 6(a), is locally refined during the computation by means of multiscale-based grid adaptation. The final grid corresponding to the steady state solution consists of about 45000 cells distributed over 5 levels of refinement, see Fig. 6(b). For comparison, the uniformly refined mesh corresponding to $L = 5$ consists of about 620000 cells. Hence the computational complexity is reduced by a factor of about 14.

As can be concluded from Fig. 6(b), the compression wave has been automatically detected during runtime through grid adaptation. At the trailing edge, see Fig. 7(b), the compression wave is adequately resolved by cells on the highest refinement level. In the far field, since the compression becomes weaker, it no longer needs to be resolved on the highest refinement level.

For the boundary layer we have similar conclusions. Although the grid is a priorily refined in this layer, due to the stretching applied to the parameter spaces of the B-Spline mappings, the grid is further refined. This is verified by Fig. 8, where the resolution of the boundary layer at the end of the plate is shown for the initial and the final grid, respectively. The numerical boundary layer thickness at the end of the plate $\delta_{num} = 1.12\,mm$ that can be seen from Fig. 8(b), fits very well with the analytical prediction $\delta = 1.14mm$ of van Driest's self-similar solution. At the end of the plate the boundary layer is resolved by about 50 cells in wall-normal direction.

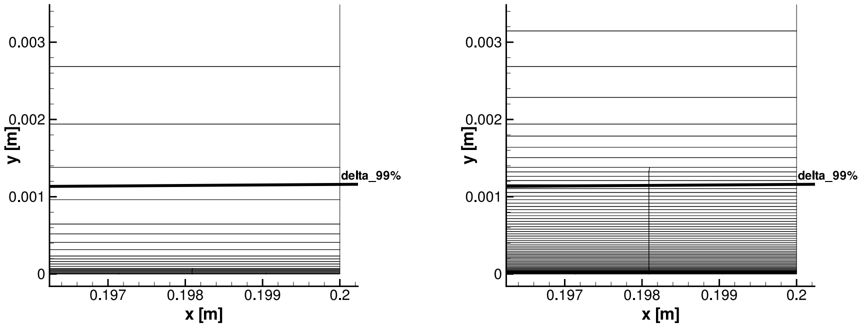
It has to be emphasized that the concept of multiscale-based grid adaptation reliably and automatically detects all physical relevant effects, namely, the compression wave and the boundary layer.

The results obtained are validated by van Driest self-similar solution [27]. As can be seen in Fig. 9, the numerical results for the skin friction agree very well with its theoretical prediction. The results for the Stanton number, describing the heat flux at the wall, show slight differences, but the quality of agreement is still satisfactory.



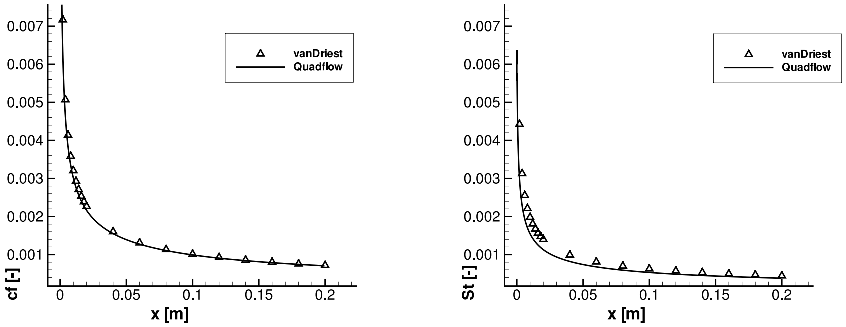
(a) (b)

Figure 7: Resolution of the leading edge of the plate: initial grid (uniform 1st refinement level) (a) and final adaptive grid (b).



(a) (b)

Figure 8: Resolution of the boundary layer: initial grid (uniform 1st refinement level) (a) and final adaptive grid (b).



(a) (b)

Figure 9: Supersonic flow over a flat plate compared according to van Driest's self-similar solution [27]: skin friction (a) and Stanton number (b).

## 4. Application

The validated solver is now applied to the simulation of cooling gas injection through a slot that is formally infinite in spanwise direction for a 2D simulation but has finite length in 3D. The results are compared with experiments that were carried out in the shock tunnel TH2 at the Shock Wave Laboratory of the RWTH Aachen as well as with numerical simulations for the formally infinite slot provided by a structured two-dimensional finite volume solver [8] and a three-dimensional mixed finite difference / spectral method of higher order [26], both non-adaptive. Note that due to the use of the spectral method in spanwise direction, the latter code cannot handle the finite slot.

### *4.1. Experimental Setup*

The experiments have been conducted at supersonic inflow conditions, where the main flow field hits a wedge with edge length 150 mm at an angle of attack $\alpha = 30°$, see Figure 10. Typically an oblique shock forms at the leading edge of the wedge. Cooling gas, here air, is injected through a slot in the surface that is located 55 $mm$ from the leading edge of the wedge, see Fig. 11. Characteristic parameters of the experimental setup are the slot width $L_s$, the injection angle $\Theta$ and the blowing ratio

$$F = \frac{\rho_c |\mathbf{v_c}|}{\rho_e |\mathbf{v_e}|}, \tag{7}$$

where the subscripts $e$ and $c$ denote the respective values behind the oblique shock wave and the condition of the injected cooling gas. These parameters can be varied as follows: $L_s \in [0.5\, mm, 1\, mm]$, $\Theta \in [30°, 90°]$ and $F \in [0.0151, 0.13]$ where for the latter range the flow is expected to remain laminar. Note that $\Theta = 90°$ corresponds to an injection orthogonal to the wall. For a slot width in spanwise direction larger than 90 mm Heufer [8] found no measurable influence of the vortices generated at the ends of the slot on the wall heat flux on the symmetry line of the plate. In agreement with the above setup we compare the two-dimensional simulations of the formally infinite slot with experiments with 120 mm slot width, see Section 4.2, and three-dimensional computations with 60 mm slot width, see Section 4.4, respectively.

### *4.2. Computational Setup (2D)*

The flow field in the experiment is essentially laminar and, as stated in the previous section, two-dimensional on the symmetry axis for large enough spanwise slot widths. In order to reduce the computational complexity, we therefore simplify the 3D experimental setup by a two-dimensional configuration of a flat plate, see Fig. 11, where we impose inflow boundary conditions determined by the flow conditions behind the oblique shock. These post-shock conditions correspond to the free stream condition specified in Section 3. For the injection, we consider five configurations with varying slot width $L_s$, injection angle $\Theta$ and maximum blowing ratio $F$ as summarized in Table 1. For all cases the cooling fluid is air at a temperature equal to the wall temperature $T_w = 293\, K$. Note

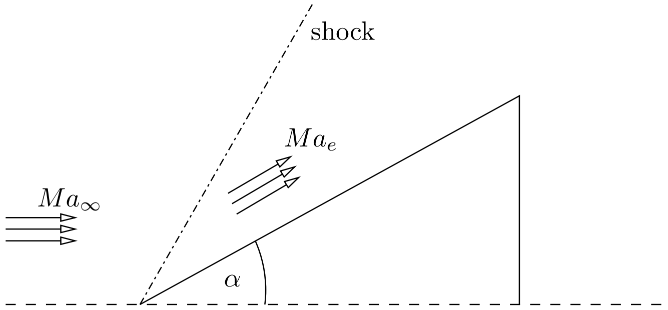


Figure 10: Side view of experimental setup: upper part of wedge.

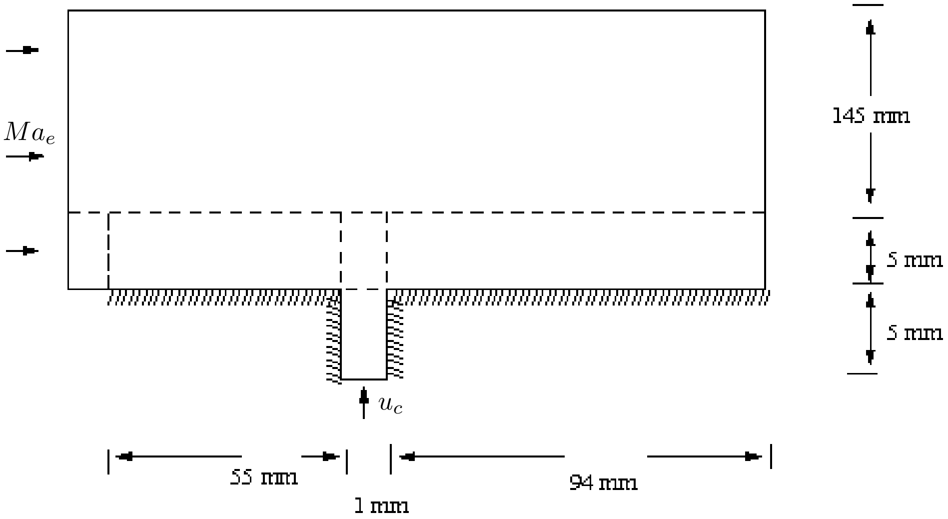


Figure 11: Computational setup: plate with a slot where the dashed lines indicate the boundaries of the grid blocks (not to scale).

that cond. 1 and 3 only differ in the smaller injection angle. Since the effective slot width varies with the injection angle according to

$$L_{s,eff} = L_s \sin(\theta). \tag{8}$$

the effective mass flux is given by

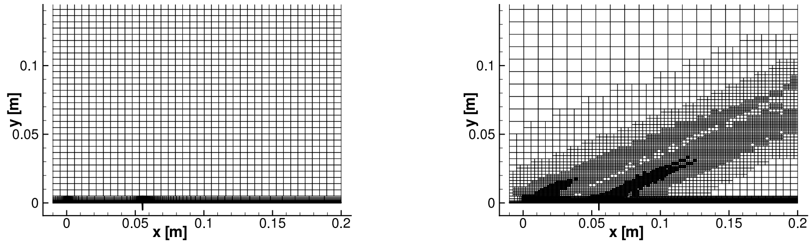
$$F_{eff} = F\, L_s \sin(\theta) \tag{9}$$

and hence is the same in both cases. In our computations, the mass flow of injection gas is either prescribed by imposing a top hat profile via boundary conditions at the slot, i.e.,

$$\frac{\partial p}{\partial n} = 0,\ (\rho\, u, \rho\, v) = F\, \rho_e\, Ma_e\, c_e\, (cos(\Theta), sin(\Theta)),\ T = T_w, \tag{10}$$

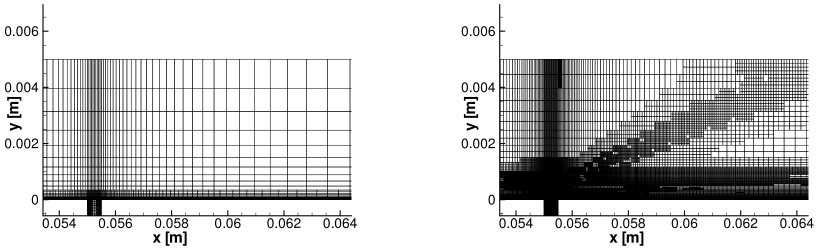
|  |  | cond. 1 | cond. 2 | cond. 3 | cond. 4 | cond. 5 | cond. 6 |
|---|---|---|---|---|---|---|---|
| $L_s$ | [mm] | 0.5 | 1 | 1 | 1 | 1 | 0.5 |
| $\Theta$ | [°] | 90 | 90 | 30 | 90 | 60 | 90 |
| $F$ | [ - ] | 0.065 | 0.0151 | 0.065 | 0.065 | 0.065 | 0.13 |
| $L_{s,eff}$ | [mm] | 0.5 | 1 | 0.5 | 1 | 0.87 | 0.5 |
| $F_{eff}$ | [mm] | 0.0325 | 0.0151 | 0.0325 | 0.065 | 0.0563 | 0.065 |

Table 1: List of injection parameters for different configurations.



(a) (b)

Figure 12: Cond. 1: Initial grid (uniform 1st refinement level) (a) and final adaptive grid (b) for simulated injection.

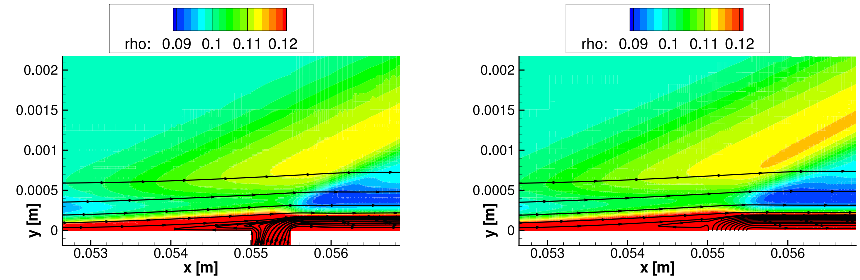


(a) (b)

Figure 13: Cond. 1: Zoom into the initial grid (uniform 1st refinement level) (a) and final adaptive grid (b) near the injection slot in case of simulated injection.

or we fully simulate the flow in the injection channel. In the following we refer to the results of the different approaches by either *simulated* or *modeled* injection, respectively.

The computations have been carried out either on a 5-block (modeled injection) or 6-block (one extra block for the simulated injection) grid, refined by stretching either towards the solid wall or, in the other parameter direction, towards the leading edge of the plate and to the slot, respectively. The computational domain extends over $\Omega = [-0.01\,m, 0.15\,m] \times [0\,m, 0.15\,m]$. In total the coarse 5-block grid consists of 796 grid cells and the coarse 6-block grid of 846 cells, respectively, and 4 levels of refinement are used, see Fig. 12.



(a) (b)

Figure 14: Cond. 1: Density distribution and streamlines for simulated (a) and modeled (b) injection.

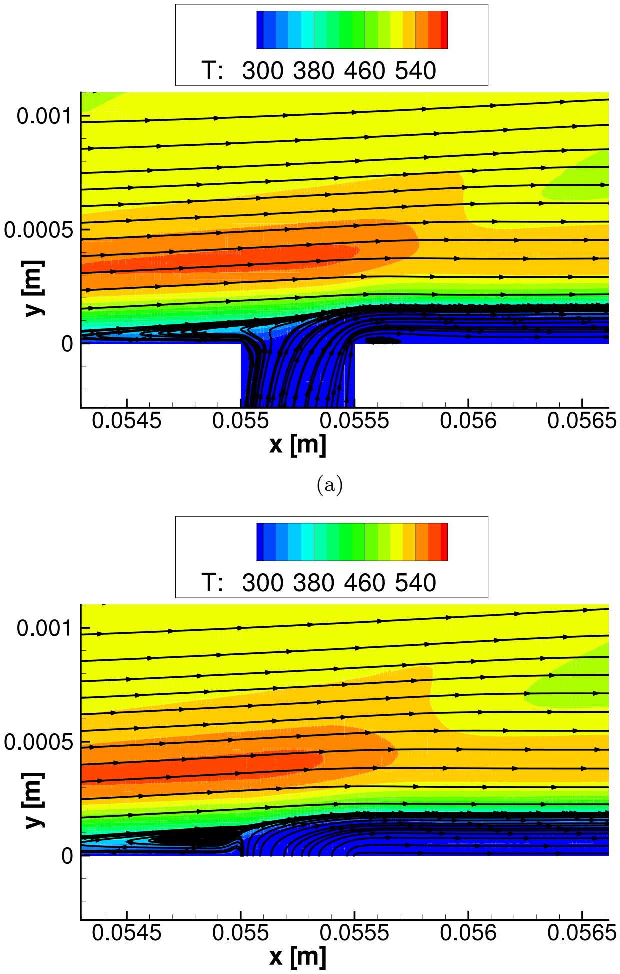
### *4.3. Numerical Results (2D)*

First, we discuss the performance of the grid adaptation exemplifying the computations corresponding to cond. 1 in case of simulated injection. For this configuration, the final adaptive grid corresponding to the steady state solution consists of about 58000 grid cells. For comparison, the non-adaptive reference computations by Heufer [25] are based on a grid for a smaller computational domain that consists of 74000 grid cells. As can be seen in Figs. 12 and 13 the grid is locally refined in the boundary layer and at the compression waves. Note that there are two isotropic compression waves, one emanating at the leading edge of the plate and another one on top of the injection jet that is shown in Fig. 13(b). The second compression wave results from the interaction of the injection jet with the laminar boundary layer. Due to the injection, the boundary layer becomes thicker and causes a local compression of the attached supersonic flow field which in turn gives rise to the development of a compression wave. This is displayed in Fig. 14 for the density field.

Next we discuss the influence of modeling and simulating the injection. For this purpose, we present in Fig. 15 the local temperature fields in the neighborhood of the slot produced by the numerical simulations corresponding to simulated and modeled injection, respectively. A few millimeters in front of the slot, the main flow deflects from the plate and the boundary layer becomes thicker. Furthermore, there is a separation bubble located in the subsonic area in front of the slot, which arises from a large vortex located there. This clockwise rotating vortex seems to be more developed in case of modeled injection. For cond. 6, with doubled mass flux, two vortices form upstream of the modeled injection, see Fig. 16(b). The second one closer to the wall rotates counterclockwise. For simulated injection, the cooling gas jet already separates in the injection channel and therefore the second vortex is located there, see Fig. 16(a). In addition, for the simulated injection we observe another small vortex directly behind the slot.

Although there are visible differences in the flow field, the effect on the cooling efficiency

$$\eta = 1 - \dot{q}_c/\dot{q}_{nc} \tag{11}$$

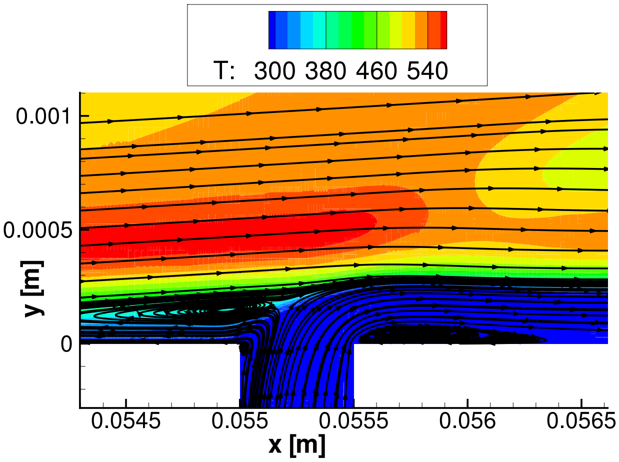


(a)

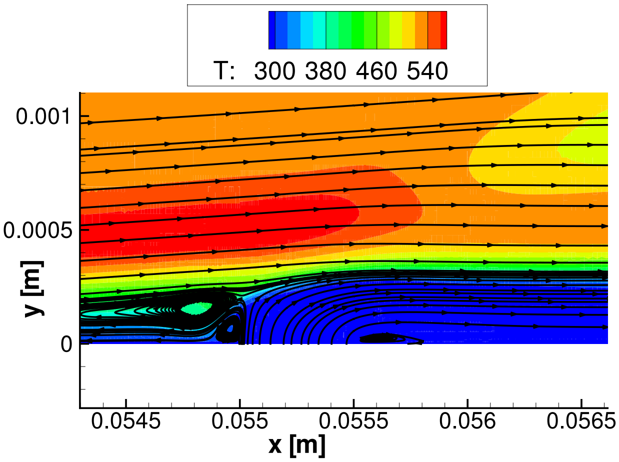
(b)

Figure 15: Cond. 1: Temperature distribution and streamlines for simulated (a) and modeled (b) injection.

is small as can be seen in Fig. 17(b). Here the cooling efficiency is defined by the specific wall heat fluxes $\dot{q}_c$ and $\dot{q}_{nc}$ corresponding to simulations with and without cooling gas injection, respectively. This was already found by Heufer [25]. Far upstream the slot the cooling efficiency is 0, until it starts to increase in the separation bubble. In the injection area it reaches 1 and then slowly decreases. The only qualitative difference for the simulated injection can be found directly behind the slot, where a cooling efficiency larger than 1 appears. This results from the expanding flow around the corner, where the temperature next to the wall becomes smaller than the wall temperature at this position and the wall heat flux changes its sign. Furthermore, the cooling efficiency seems to decrease



(a)



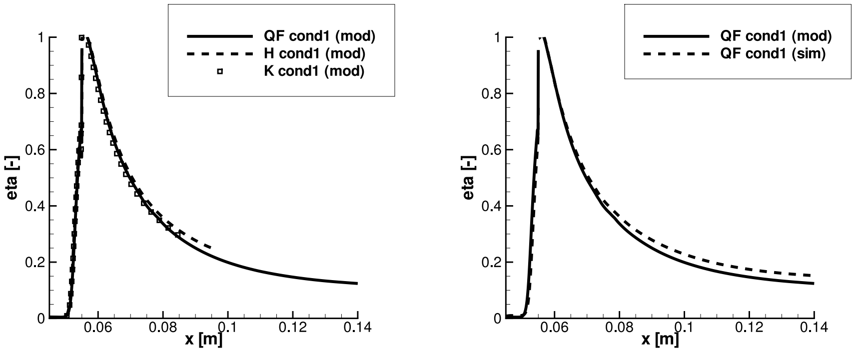
(b)

Figure 16: Cond. 6: Temperature distribution and streamlines for simulated (a) and modeled (b) injection.

faster in case of modeled injection, but this phenomenon was not observed for other conditions.

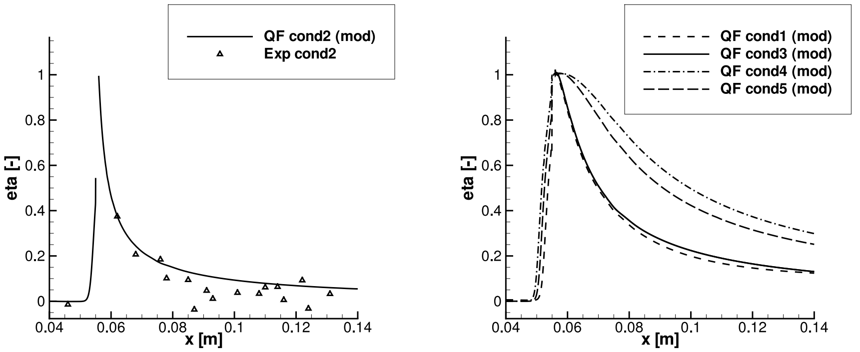
To validate our results, we compare them with experimental measurements conducted in the shock tunnel TH2 and non-adaptive computations with the 2D finite volume code and the 3D mixed finite difference / spectral method, provided by Heufer [25] and Keller [26], respectively. As can be concluded from Figs. 17(a) and 17(c), the results for the cooling efficiency are in good agreement and, hence, confirm our computations.

In addition we arrange parameter studies for varying injection parameters. As can be seen in Fig. 17(e), the lower blowing ratio (cond. 2 compared to 4)



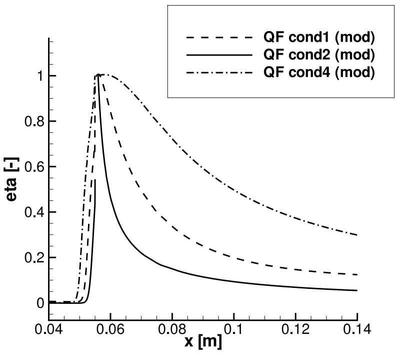
(a) Cond. 1: Comparison with results from Heufer (H) and Keller (K).

(b) Cond. 1: Comparison of modeled (mod) and simulated (sim) injection.



(c) Cond. 2: Comparison of modeled (mod) injection and experimental results [25]

(d) Comparison of different injection angles.



(e) Comparison of different slot widths (cond. 1, 4) and blowing ratios (cond. 2, 4).

Figure 17: Cooling efficiency $\eta$.

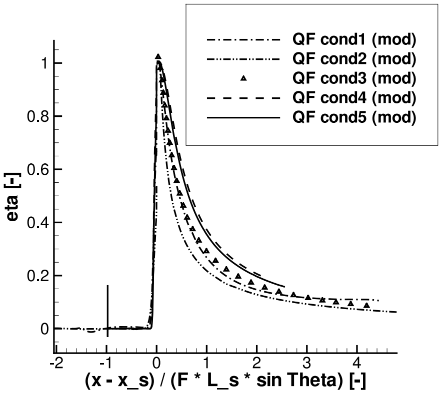


Figure 18: Cooling efficiency $\eta$ over $x$ normalized by effective mass flux.

results in a much faster decreasing cooling efficiency. Decreasing the slot width (cond. 1 compared to 4) has the same effect. If a critical blowing ratio is reached, the cooling efficiency suddenly becomes negative, because the high mass flux results in the transition of the boundary layer from laminar to turbulent. The blowing ratios considered here are all below the critical ones found by Heufer [8].

In the above cases the direction of injection is orthogonal to the flow field. Now we discuss the influence of injecting the cooling gas at a different angle where we confine ourselves to the approach of modeled injection. The results for the cooling efficiency are displayed in Fig. 17(d). Note that the injection angle does not change the specific mass flux. Nevertheless, it influences the cooling efficiency, because the effective slot width (8) changes by the factor of $sin(\Theta)$. Therefore cond. 1 and 3 lead to almost the same results, because the factor 2 in the slot width is compensated by the factor $sin(30°) = 1/2$ in the effective slot width $L_{s,eff}$. Keeping the other parameters fixed and decreasing only the injection angle from 90° (cond. 4) over 60° (cond. 5) to 30° (cond. 3) results in a decreasing cooling efficiency, because of the decreasing effective slot width.

In order to compare the results of all different configurations, we finally scale the $x$-axis with the different effective mass fluxes, see Table 1. This allows us to compare the cooling efficiencies independently from the mass flux. As we may conclude from Fig. 18, the results seem to be more or less independent of the geometric parameters. At least this holds for the parameter range under consideration, but might be different for small injection angles or higher blowing ratios. There remains a significant dependence only on the effective mass flux itself, i.e., the cooling efficiency in these regions scales nonlinearly with the effective mass flux.

### *4.4. Computational Setup (3D)*

After having validated the numerical method in 2D, we now turn to the three-dimensional application. Here we investigate the injection of a cooling

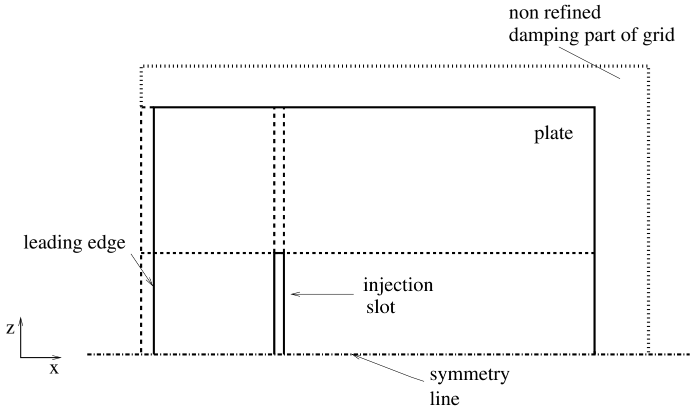


Figure 19: Projection of computational setup (bottom): flat plate with slot where the lines indicate the plate (solid line), the block boundaries (dashed and solid lines), the non-refined region (dotted line) and the symmetry line (dashed-dotted line) (not to scale).

gas through a finite slot into a supersonic laminar flow field and the interaction between the main flow field and the jet. The computational configuration mimics the experimental setup from Section 4.1, where we consider a flat plate of length 150 $mm$ and width 100 $mm$. The leading edge of the injection slot is located at 55 $mm$ from the plate's leading edge and is centered at the symmetry line $z = 0\,m$. The slot measures 60 $mm$ in spanwise direction. The computational setup is sketched in Figure 19.

Because of the symmetry of the configuration in spanwise direction the computational domain is reduced to one half of a flat plate with symmetric boundary conditions. The grids are similar to those in the two-dimensional case, where we uniformly extend the grids in spanwise direction. Thus the computational domain is $\Omega = [-0.01\,m, 0\,m, 0\,m] \times [0.4\,m, 0.15\,m, 0.15\,m]$ and, for simulated injection, by another block for the injection channel. In total the coarse grid for the modeled injection consists of 16585 grid cells and 5 levels of refinement are used. The final adaptive grid after 5 adaptation steps has about 40 million cells. In comparison to the uniformly refined reference mesh, we save a factor of about 13. In order to avoid problems due to reflecting waves that might occur at the outflow boundary and at the non-symmetric side boundaries, we ŭse an extended far field around the plate. Introducing a swamp region in the far field, see Figure 19, where we do not apply grid adaptation, the computational load can be significantly reduced, because waves emitting from the leading edge and the injection are resolved only on a coarse grid in the far field. This results in numerical damping of these waves in the swamp region. However, this does not affect the accuracy near the injection slot as confirmed by reference computations without swamp region.

In the following we investigate different configurations, varying the injection angle $\Theta$ and the blowing ratio $F$ according to Tab. 2. Note that a blowing ratio of $F = 0.13$ is close to the limit, where stable laminar solutions can be generated

| | | cond. 1a | cond. 1b | cond. 2a | cond. 2b |
|---|---|---|---|---|---|
| $L_s$ | [mm] | 0.5 | 1 | 0.5 | 1 |
| $\Theta$ | [°] | 90 | 30 | 90 | 30 |
| $F$ | [ - ] | 0.065 | 0.0065 | 0.13 | 0.13 |
| $L_{s,eff}$ | [mm] | 0.5 | 0.5 | 0.5 | 0.5 |
| $F_{eff}$ | [mm] | 0.0325 | 0.0325 | 0.65 | 0.65 |

Table 2: List of injection parameters for different configurations.

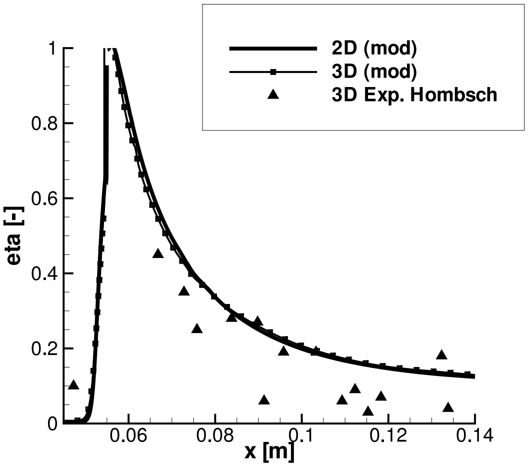


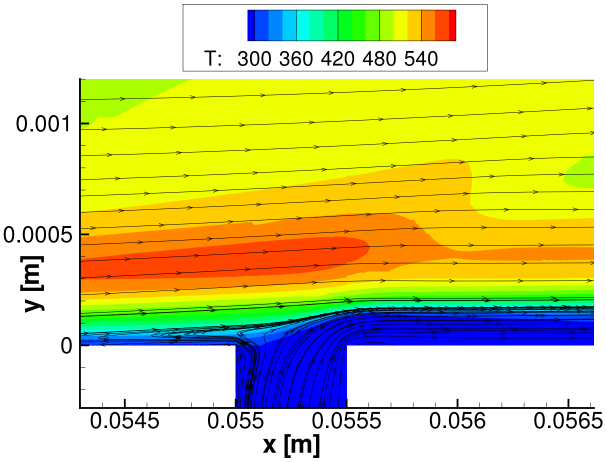
Figure 20: Cond. 1a: Three-dimensional results on symmetry line compared with experimental data and two-dimensional results.

in experiments. Recall that the effective slot width denotes the width of the injection channel. This value is the same for all configurations which ensures that the same amount of cooling gas is injected for different injection angles and constant blowing ratio.

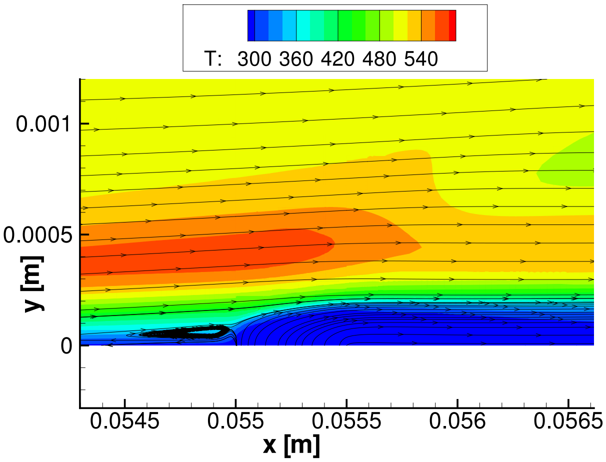
### *4.5. Numerical results (3D)*

First, we compare the two-dimensional with three-dimensional results for cond. 1a, the latter taken directly from the symmetry line, see Fig. 20. For validation purposes, we invoke experimental data provided by Hombsch [39]. As can be seen, the cooling efficiencies derived from the two-dimensional computation and from the symmetry line of the three-dimensional one match very well.

Comparing the temperature distribution and the streamlines around the injection at $z = 10\,mm$, see Fig. 21(b), with the two-dimensional results, see Fig. 15(b), the similarity is obvious. For higher blowing ratios and modeled injection, see Fig. 22(b), a second vortex can be found upstream of the injection, generated by the injected cooling gas. Therefore, it rotates in the opposite



(a) Cond. 1a - Simulated injection



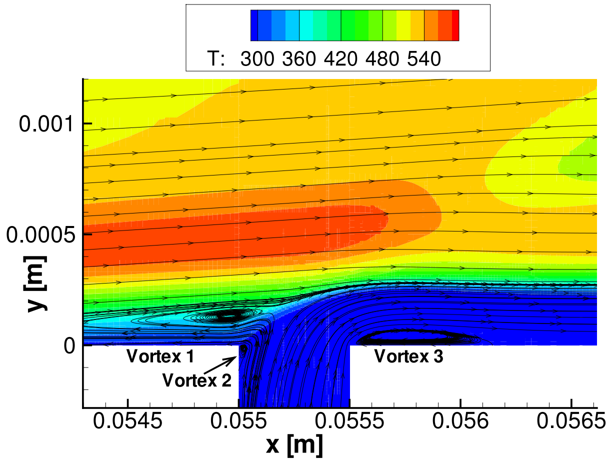
(b) Cond. 1a - Modeled injection

Figure 21: Temperature distribution and tangential streamlines on slice $z = 0.01m$.

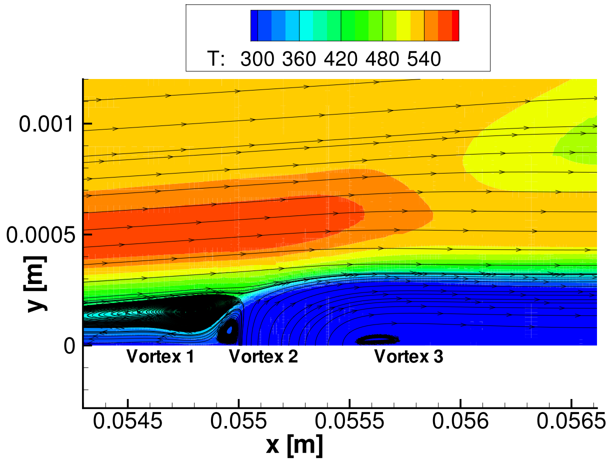
direction to vortex 1 in the separation area, fed from the main flow. There is also a third vortex downstream of the injection slot. For simulated injection, vortex 2 is not located upstream of the injection, but in the injection slot itself, see Fig. 22(a). These observations are again similar to the two-dimensional results, see Fig. 16.

The differences in the cooling efficiency in spanwise direction can be seen in Fig. 23. The distribution of the cooling efficiency is shown over the simulated half-plate for different configurations. The symmetry line is located at $z = 0\,m$ and the slot width is again $60\,mm$, that means it extents to $z = 0.3\,m$.

The three figures on the left show the two conditions with the lower blowing



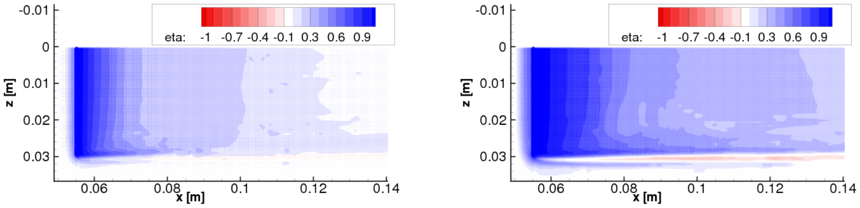
(a) Cond. 2a - Simulated injection



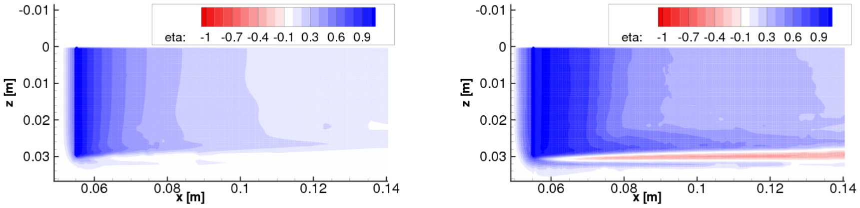
(b) Cond. 2a - Modeled injection

Figure 22: Temperature distribution and tangential streamlines on slice $z = 0.01m$.

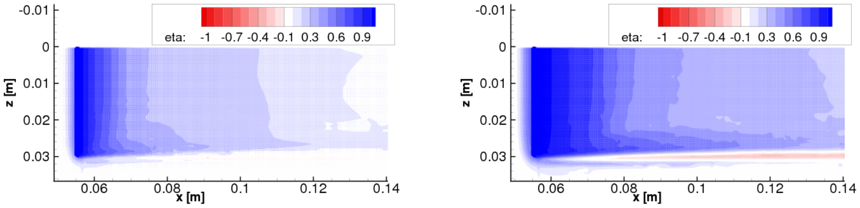
ratio (cond. 1a and 1b). Comparing, from top to bottom, fully simulated with modeled orthogonal injection, see Fig. 23(a) and 23(c), and modeled injection with an injection angle of 30°, see Fig. 23(e), there is just a small difference in the distribution of the cooling efficiency between the first one and the other ones. Next to the symmetry line, the isolines of the cooling efficiency are parallel to the injection slot for all three configurations. Further downstream, the cooling efficiency rises a little bit and then drops down to zero, before $z = 0.03m$. Due to the injection jet of the cooling gas, the hot gas of the main flow field is deflected and bends around the injection slot and narrows the regime of the cooling fluid. This effect shows different intensities for the three approaches, that will become



(a) Cond. 1a: Simulated injection (b) Cond. 2a: Simulated injection



(c) Cond. 1a: Modeled injection (d) Cond. 2a: Modeled injection



(e) Cond. 1b: Modeled injection (f) Cond. 2b: Modeled injection

Figure 23: Cooling efficiency $\eta$.

clear for higher blowing ratios. In Fig. 23(d), for the modeled injection, there is a quite large region with a negative cooling efficiency downstream the edge of the injection slot. The wall heat flux in this region is larger than for no cooling at all, i.e., the wall is not cooled but heated. Clearly, this effect is the opposite of what one would like to achieve. As we have seen for the two-dimensional simulations before, the cooling gas injection is an obstacle for the main flow field. In front of this obstacle, a system of three vortices is generated, as can be seen in Fig. 22(b). These three vortices influence the vortex structure downstream from the edge of the injection slot. Fig. 24 shows this three vortices and the vortex system in front of the injection slot and behind. Downstream from the edge of the injection slot, two counterrotating vortices appear. In between these two vortices, hot gas from the main flow is transported directly towards the wall. The temperature boundary layer is very thin in the region between these two vortices.

For the simulated injection, see Fig. 23(b), this effect is less pronounced. The region with negative cooling efficiency is smaller. The simulated injection poses

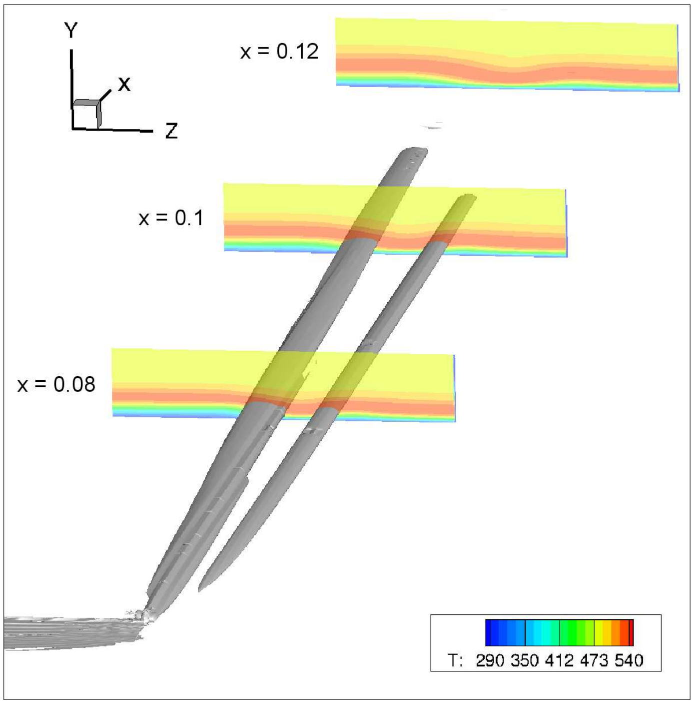
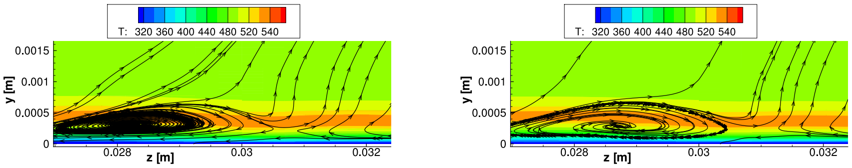


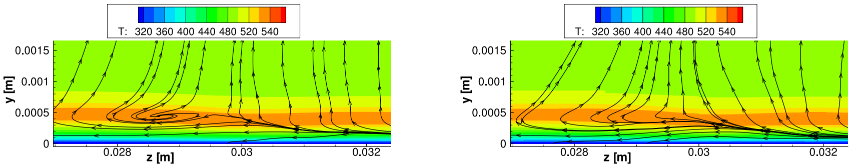
Figure 24: Cond. 2a, modeled injection: vorticity magnitude and temperature slices at $x = 0.08\,m$, $x = 0.1\,m$ and $x = 0.12\,m$.

a much smaller obstacle for the main flow, because, in contrast to the modeled case, the jet is not perfectly orthogonal to the wall but bent in flow direction. Therefore, vortex 1 is less strong. Injecting the cooling gas with an angle of 30°, see Fig. 23(f), has the same effect, vortex 1 is not that strong, due to the lower momentum of the injected mass flux in wall-normal direction. Therefore, the wall is not heated in as large a region as in the modeled case. Although the developement of the downstream vortices has not yet been investigated in detail, we can clearly state that there is a direct impact on their size and strength from the spanwise vortices around the injection slot and due to this, an influence on the existence and size of the heated region.

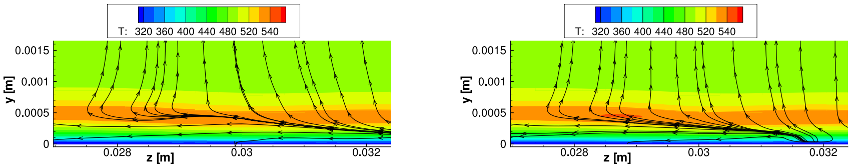
The streamwise vortices generated at the end of the injection slot and their effect on the temperature distribution can be seen in more detail in Fig. 25 for cond. 1a and 1b, modeled injection, and in Fig. 26 for cond. 2a, simulated and modeled injection. For a smaller blowing ratio the streamlines just form one vortex, close to the injection, see Fig. 25(a). Further downstream, this vortex vanishes, see Fig. 25(c) and Fig. 25(e). There is no significant influence of this vortex on the temperature boundary layer. Changing the injection angle to 30°, the streamlines and the temperature distributions are almost the same, see Fig. 25(b), 25(d) and 25(f)).



(a) Cond. 1a, $x = 0.08\,m$ (b) Cond. 1b, $x = 0.08\,m$



(c) Cond. 1a, $x = 0.1\,m$ (d) Cond. 1b, $x = 0.1\,m$



(e) Cond. 1a, $x = 0.12\,m$ (f) Cond. 1b, $x = 0.12\,m$

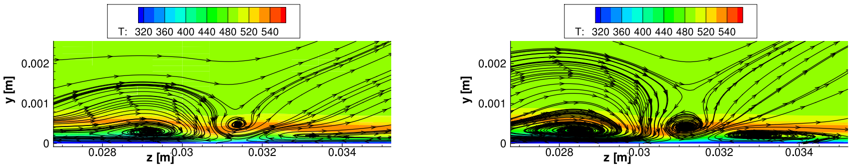
Figure 25: Temperature slices with tangential streamlines at different positions for modeled 90° and 30° injection.

For a higher blowing ratio, more significant differences can be observed. First of all, the influence of the injection on the temperature boundary layer is larger. In particular, in the first cross section (Fig. 26(a) and 26(b)) the layer of low temperature located directly at the wall is thicker on the left. As mentioned before, there are two counterrotating vortices for this condition. Between them, the streamlines are directed towards the wall. This leads to a very thin temperature boundary layer between the two vortices. For modeled injection, further downstream, the vortex on the right moves into the cooled regime, see Fig. 26(d) and 26(f). This increases the area that is not cooled at all when compared with the simulated injection where the vortex more or less keeps its spanwise position, see Fig. 26(c) and 26(e), also compare Fig. 23(d) and 23(b).

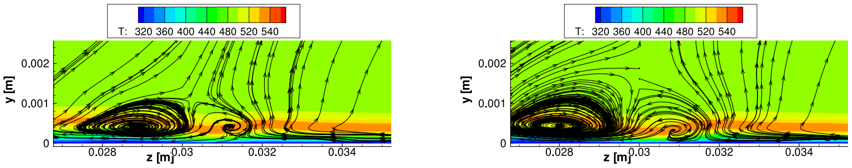
Due to their small extent in spanwise direction, the vortex systems that are resolved by our simulations could not be found in the experiments for the following simple reason. The vortex reaches a diameter of about $4\,mm$, while the thermal elements have a diameter of $2\,mm$ and can therefore not be placed within a mutual distance of less than $6\,mm$. Furthermore, the effect on the wall heat flux is not strong enough to be seen in infrared pictures.

## 5. Conclusion and Outlook

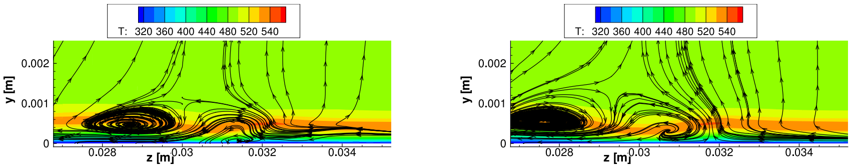
The numerical investigations confirm that the Quadflow solver is an appropriate tool for simulating film cooling offering insight that could not be obtained



(a) Simulated injection, $x = 0.08\,m$ (b) Modeled injection, $x = 0.08\,m$



(c) Simulated injection, $x = 0.1\,m$ (d) Modeled injection, $x = 0.1\,m$



(e) Simulated injection, $x = 0.12\,m$ (f) Modeled injection, $x = 0.12\,m$

Figure 26: Cond. 2a: Temperature slices with tangential streamlines at different positions for both modeled and simulated injection.

from experiments. In particular, the multiscale-based grid adaptation concept turns out to be a reliable tool by which all physically relevant effects are automatically detected and appropriately resolved. Since the grid is only refined in regions, where the flow field exhibits local variations, this results in substantial savings of cells in comparison to a fully refined grid and hence the computational costs both in terms of CPU time and memory using are significantly reduced, whereas the accuracy of the uniform mesh corresponding to the highest refinement level is maintained. Because of this, we have been able to locally resolve physical effects on small scales, that are hard to detect by experiments, due to size limitations of measuring devices.

Film cooling using slots is a promising strategy also in supersonic flows. Slots generate a very homogeneous cooling film which entails a good cooling effect over large homogeneous regions. Of special interest is the region downstream from the edges of the slot where, in particular, for higher momentum ratios in wall-normal direction vortices appear transporting hot gas directly to the wall. The presented simulations show that this effect is rather sensitive since modeling or simulating the injection yields quantitatively varying results. More accurate ways of modeling, such as prescribing a velocity profile on the injection boundary, should be investigated in the future. Depending on the geometry of the underlying technical problem, staggered rows of finite slots or passive cooling solutions in this region could be possible. Normal injection facilitates injecting a large amount of cooling gas through narrow slots, but boosts the separation of the main flow and stronger vortices may form. Injection with a smaller injection angle needs wider slots, but the momentum ratio is smaller. The technical feasibility depends again on the concrete problem and its geometry. Tangential injection into the main flow, for instance, behind a step is a topic of current research [5].

In the present work we consider only air as cooling gas. With regard to an optimal cooling efficiency, we have started to investigate different coolants. This can have a major impact on the thickness of the cooling layer and the resulting cooling efficiency were already investigated in experiments by Hombsch [43]. In particular, the influence of various coolants such as Air, Argon, Helium, Sulfur(VI)fluoride and Carbon Dioxide on the wall heat fluxes and the corresponding cooling efficiencies were measured. In order to perform numerical simulations the air flow over the flat plate and the injected coolants have to be modeled as a mixture of perfect gases. For this purpose, the QUADFLOW solver is currently being extended, see Windisch [44].

Our future interest will be concerned also with the interaction of a cooling gas jet with a turbulent flow field. Since a DNS is not feasible, we have to model the effect of the unresolved fluctuations on the resolved coarse scales. Up to now, turbulent flow fields have been investigated by means of the Reynolds Averaged Navier-Stokes (RANS) equations, where the averaged governing equations are solved for the mean variables. For the turbulent closure, classical one- and two-equation models such as Spalart-Allmaras and Menter-SST, respectively, are used. Recently a differential Reynolds stress model, the SSG/LRR-$\omega$ model, has been implemented into Quadflow [40]. In addition to these models, we will use the Variational Multiscale Method (VMS). This method can be considered as an advanced LES. It has already been implemented into Quadflow according to Koobus and Farhat [41] and Dahmen et al. [42] and is currently being validated.

An even more challenging question concerns the use of porous media materials for the injection. Appropriate materials are currently being investigated [45] and experiments have already been conducted [46]. In order to perform efficient numerical simulations we have to avoid the resolution of corresponding fine scale geometric structures using upscaling strategies, for instance, based on local fine scale problems. For this purpose, the detailed knowledge of the injection process through single holes or slots will be helpful in deriving effective boundary conditions for a homogenized mass flux of cooling gas through porous media.

## Acknowledgments

Financial support has been provided by the German Research Council (Deutsche Forschungsgemeinschaft - DFG) in the framework of the Sonderforschungsbereich Transregio 40.

The authors gratefully acknowledge the data from computations and experiments by K.A. Heufer and M. Hombsch at the Shock Wave Laboratory of the RWTH Aachen and computations by M. Keller (University of Stuttgart).

## References

[1] R.J. Goldstein, Film cooling, Adv. Heat Transf. 7 (1971) 321–379.

[2] M.E. Taslim, S.D. Spring, Experimental investigations of film cooling effectiveness for slots of various exit geometry, J. Thermophys. and Heat Transf. 6 (2) (1992) 302–307.

[3] A.D. Fitt, J.R. Ockendon, T.V. Jones, Aerodynamics of slot film cooling: theory and experiment, J. Fluid Mech. 160 (1985) 15–27.

[4] M. Thole, M. Gritsch, A. Schulz, S. Wittig, Transonic film cooling investigations: effects of hole shapes and orientations, Am. Soc. Mech. Eng. 66-GT-222 (1996).

[5] M. Konopka, M. Meinke, W. Schröder, Large-Eddy Simulation of supersonic film cooling at finite pressure gradients, 8th EUROMECH Fluid Mechanics Conference, Bad Reichenhall, Germany, September 13-16, 2010.

[6] A. F. Gollnick, Jr., Thermal effects on transpiration cooled hemisphere, J. Aerosp. Sci. 29 (1962) 583–595.

[7] D. M. Peterson, G. V. Candler, Hybrid Reynolds-Averaged and Large-Eddy Simulation of normal injection into a supersonic crossflow, J. Propuls. and Pow. 26(3) (2010) 533–544.

[8] K.A. Heufer, H. Olivier, Experimental and numerical study of cooling gas injection in laminar, supersonic flow, AIAA-J. 46 (2008) 2741–2751.

[9] J. Linn, M.J. Kloker, Effects of wall-temperature conditions on effusion cooling in a Mach-2.67 boundary layer, AIAA-J. 49 (2) (2011) 299–307.

[10] J. Linn, M.J. Kloker, DNS of film cooling in hypersonic boundary-layer flow, High Performance Computing in Science and Engineering 08, Transactions of the HLRS 2008, 171– 190, Springer (2008).

[11] F. Bramkamp, Ph. Lamby, S. Müller, An adaptive multiscale finite volume solver for unsteady and steady state flow computations, J. Comp. Phys. 197 (2) (2004) 460–490.

[12] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, Comm. Pure Appl. Math. 48 (12) (1995) 1305–1342.

[13] S. Müller, Adaptive Multiscale Schemes for Conservation Laws, Lecture Notes on Computational Science and Engineering 27, first ed., Springer Verlag, 2003.

[14] A. Cohen, S.M. Kaber, S. Müller, M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws, Math. Comp. 72 (241) (2003) 183–225.

[15] A. Cohen, S.M. Kaber, M. Postel, Multiresolution analysis on triangles: application to gas dynamics, Hyperbolic Problems: Theory, Numerics, Applications, G. Warnecke, H. Freistühler (eds.), 257–266, Birkhäuser, Basel (2002).

[16] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic PDEs, J. Comp. Phys. 188 (2) (2003) 493–523.

[17] F. Coquel, M. Postel, N. Poussineau, Q.H. Tran, Multiresolution technique and explicit-implicit scheme for multicomponent flows, J. Numer. Math. 14 (2006) 187–216.

[18] O. Roussel, K. Schneider, Adaptive multiresolution method for combustion problems: Application to flame ball-vortex interaction, Comput. & Fluids 34 (7) (2005) 817–831.

[19] R. Bürger, R. Ruiz, K. Schneider, Fully adaptive multiresolution schemes for strongly degenerate parabolic equations with discontinuous flux, J. Eng. Math. 60 (3-4) (2008) 365–385.

[20] M. Domingues, O. Roussel, K. Schneider, On space-time adaptive schemes for the numerical solution of partial differential equations, ESAIM Proc. 16 (2007) 181–194.

[21] M. Duarte, M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, F Laurent, New resolution strategy for multi-scale reaction waves using time operator splitting, space adaptive multiresolution and dedicated high order implicit/explicit time integrators, HAL-00457731, version 1, `http://hal.archives-ouvertes.fr/hal-00457731/en/`, 2010.

[22] B. Hejazialhosseini, D. Rossinelli, M. Bergdorf, P. Koumoutsakos, High order finite volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions, J. Comp. Phys. 229 (2010) 8364–8383.

[23] S. Müller, Multiresolution Schemes for Conservation Laws, in Multiscale, Nonlinear and Adaptive Approximation, R. DeVore and A. Kunoth (eds.), Springer Verlag, 2009, pp. 379–408.

[24] F. Coquel, Y. Maday, S. Müller, M. Postel, Q.H. Tran (eds.), Multiresolution and Adaptive Methods for Convection-Dominated Problems, ESAIM: Proceedings, Vol. 29, 2009.

[25] K.A. Heufer, Investigations of Film Cooling in Supersonic Flows (german), first ed., Shaker Verlag, 2008.

[26] M. Keller, M.J. Kloker, Influence of a favorable streamwise pressure gradient on laminar film cooling at Mach 2.67, in: Proceedings of 4th European Conference for Aerospace Sciences (EUCASS), Sankt Petersburg, Russia, July 4-8 2011.

[27] J.D. Anderson, Hypersonic and High Temperature Gas Dynamics. McGraw-Hill, New York, 1989, 228–259.

[28] J. Ballmann, Flow Modulation and Fluid-Structure Interaction at Airplane Wings, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 84, Springer Verlag, 2003.

[29] W. Schröder, Flow Modulation and Fluid-Structure Interaction Findings, Numerical Fluid Mechanics and Multidisciplinary Design (NNFM) 109, Springer Verlag, 2010.

[30] J.M. Carnicer, W. Dahmen, J.M. Peña, Local decomposition of refinable spaces and wavelets, Appl. Comput. Harmon. Anal. 3 (1996) 127–153.

[31] N. Hovhannisyan, S. Müller, On the stability of fully adaptive multiscale schemes for conservation laws using approximate flux and source reconstruction strategies, IMA Journal of Numerical Analysis 30 (2010) 1256–1295.

[32] Ph. Lamby, Parametric Multi-Block Grid Generation and Application to Adaptive Flow Simulations, Diss. RWTH Aachen, `http://darwin.bth.rwth-aachen.de/opus3/volltexte/2007/1999/pdf/Lamby_Philipp.pdf`, 2007.

[33] K.-H. Brakhage, W. Dahmen, Ph. Lamby, A unified approach to the modeling of airplane wings and numerical grid generation using B-spline representations, in Flow Modulation and Fluid-Structure-Interaction Findings, W. Schröder (ed.), Numerical Fluid Mechanics and Multidisciplinary Design (NNFM) 109, Springer Verlag, 2010, pp. 239–264.

[34] F. Bramkamp, Unstructured *h*-Adaptive Finite-Volume Schemes for Compressible Viscous Fluid Flow, Diss. RWTH Aachen, `http://darwin.bth.rwth-aachen.de/opus3/volltexte/2003/725/03_255.pdf` (2003).

[35] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer Verlag, 1997.

[36] G. Schieffer, S. Ray, F. Bramkamp, M. Behr, J. Ballmann, An adaptive implicit finite volume scheme for compressible turbulent flows about elastic configurations, in Flow Modulation and Fluid-Structure-Interaction Findings, W. Schröder (ed.), Numerical Fluid Mechanics and Multidisciplinary Design (NNFM) 109, Springer Verlag, 2010, pp. 25–51.

[37] G. Zumbusch, Parallel multilevel methods. Adaptive mesh refinement and loadbalancing, Advances in Numerical Mathematics, Teubner, Wiesbaden, 2003.

[38] K. Brix, S. Mogosan, S. Müller, G. Schieffer, Parallelization of multiscale-based grid adaptation using space-filling curves, in Multiresolution and

Adaptive Methods for Convection-Dominated Problems, F. Coquel, Y. Maday, S. Müller, M. Postel, Q.H. Tran (eds.), ESAIM: Proceedings, Vol. 29, 2009, pp. 108–129.

[39] M. Hombsch, private communication (2011).

[40] A. Bosco, B. Reinartz, S. Müller, Differential Reynolds stress model and grid adaptation for hypersonic double wedge simulations, in: Turbulence, Heat and Mass Transfer 6, K. Hanjalić, Y. Nagano, S. Jakirlić (eds.), Proceedings of 6th Int. Symposium on Turbulence, Heat and Mass Transfer, Rome, Italy, 14-18 September 2009, CD Proceedings (2009).

[41] B. Koobus, Ch. Farhat, A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes – application to vortex shedding, Comp. Meth. Appl. Mech. and Eng. 193 (2004) 1367–1383.

[42] W. Dahmen, T. Gotzen, S. Müller, R. Schäfer, Adaptive multiresolution finite volume discretization of the Variational Multiscale Method. General Framework, in: N.A. Adams, R. Radespiel, T. Sattelmayer, W. Schröder, B. Weigand (Eds.), Annual Report SFB TRR40, München, 2010, pp. 9–25.

[43] M. Hombsch, H. Olivier, Flow condition and cooling gas variation for film cooling studies in hypersonic flow, in: N.A. Adams, R. Radespiel, T. Sattelmayer, W. Schröder, B. Weigand (Eds.), Annual Report SFB TRR40, München, 2010, pp. 27–39.

[44] Ch. Windisch, B. Reinartz, S. Müller, Numerical Simulation of Coolant Variation in Laminar Supersonic Film Cooling, accepted for 50th AIAA Aerospace Sciences Meeting, 9-12 January 2012, Nashville, Tennessee.

[45] T. Langener, M. Selzer, J. von Wolfersdorf, H. Hald, Measurements of transpiration cooling in subsonic flows applied to C/C material, in: N.A. Adams, R. Radespiel, T. Sattelmayer, W. Schröder, B. Weigand (Eds.), Annual Report SFB TRR40, München, 2010, pp. 69–82.

[46] A. Gülhan, S. Braun, An experimental study on the efficiency of transpiration cooling in laminar and turbulent hypersonic flows, Exp. Fluids 50 (3) (2011), pp. 509–525.