Efficient Simulation and Optimization of Rotationally Symmetric, Converging-Diverging de Laval Nozzles for Twin Wire Arc Spraying

S. Turek, M. Möller, M. Razzaq, L. Rivkind

Chair of Applied Mathematics & Numerics (LS 3), Department of Mathematics

Abstract

Recent developments in the design of rotationally symmetric, converging-diverging de Laval nozzles for the use in twin wire arc spraying processes are discussed. Various aspects of an efficient implementation of the proposed gas dynamics solution algorithm on modern multi- and many-core architectures are addressed. Finally, a general framework for the application of massively parallel, derivative-free optimization methods on cluster systems is presented.

Keywords: *Converging-diverging nozzle, shape optimization, parallelization, high-resolution scheme, equations of gas dynamics, boundary treatment*

1 Introduction

Thermal spraying techniques such as HVOF, plasma and flame spraying are widely used in industrial applications to produce metal coatings on structural materials to protect them against high temperatures, erosion, and wear or to impose a special surface structure which is particularly desirable in sheet metal forming. Amongst the alternative approaches, twin wire arc spraying is quite attractive due to its high deposition rates, the low process temperatures which cause less stress in the coating and the substrate and, from an economical point of view, its moderate operation costs. On the other hand, the resulting coatings usually contain numerous pores due to low particle velocities as compared to other spraying processes with higher kinetic energy.

Encouraged by the measurements and observations pursued by subproject A1 of the SFB 708, the common aim is to modify parts of the commercial spraying system so as to achieve higher particle velocities and to obtain a well focused spray jet at the same time. To reach this goal, different strategies are

available: (a) use of higher pressures for the primary gas supply, (b) installation of a secondary gas supply which can be used to control and improve the focus of the particle-laden free stream, and (c) acceleration of the primary gas to supersonic speed using a converging-diverging de Laval nozzle both upstream and/or downstream. Admittedly, improving the existing spraying system making use of all available strategies amounts to solving a huge optimization problem which is infeasible with the tools at hand.

In this report, we address one particular aspect of the optimization process, namely the design of a replacement for the existing converging nozzle inside the commercial burner which by construction cannot reach flow velocities higher than unit Mach. Given the inner dimensions of the burner, the conditions of operation at the primary inlet and the ambient pressure, the task is to design a converging-diverging de Laval nozzle which accelerates the primary gas to supersonic speed. As a second requirement, the nozzle should be correctly expanded to produce a focused free stream at the exit. For simplicity, the nozzle is assumed to be rotationally symmetric which makes it possible to devise prototypic nozzles based on an idealized one-dimensional analysis as described in Section 2. This model cannot predict the external flow downstream the nozzle exit so that corresponding CFD simulations are in order. In Section 3, an efficient solution algorithm for the numerical simulation of axi-symmetric compressible gas flows is discussed with special emphasis on its parallel implementation on multi-core architectures. Finally, a general framework for using massively parallel, derivative-free optimization methods on cluster systems is addressed in Section 4. The final goal is to combine all these ingredients into a simulation-based shape optimization of the nozzle.

2 One-dimensional analytical model

2.1 Quasi-one-dimensional model

In this part we examine the gas flow through converging-diverging nozzles using quasi-one-dimensional model. This model allows to calculate cross-sectional averaged properties as function in x along a nozzle and, as it has often been reported in literature, provides already qualitatively good results to nozzle flow behavior. Moreover, it predicts proper criteria for the nozzle design w.r.t. desired nozzle requirements. The following assumptions are used for the construction of the mathematical model:

The gas is inviscid, compressible, frictionless, adiabatic, isentropic and obeys the ideal (perfect) gas laws. The flow field is steady and varies only along the x

-axis. One of the important characteristic parameters in compressible fluids is the Mach number $M = v/a$, which is based on the local velocity v and the speed of sound v defined by $a^2 = \gamma RT$, where T the temperature, R the gas constant, and γ is the adiabatic index. When $M > 1$ the flow is supersonic and when $M < 1$ the flow is subsonic. The sonic velocity $M = 1$ can be achieved only at the minimum cross-sectional area of a nozzle, called the throat. The governing equations for steady quasi-one-dimensional flows are obtained by conservation principles of mass, momentum and energy (e.g. [And01]). The nozzle properties in the equations bellow are specified as follows:

subscript * specifies the properties at the throat of the nozzle (when M=1),
subscript o specifies the stagnation properties at the entrance of the nozzle,
subscript e specifies the properties at the exit of the nozzle,
subscript a specifies the properties of the ambient flow.

The ideal gas law takes the form $P = \rho RT$. Isentropic flow obeys the relations:

$$\frac{P_0}{P} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{(\gamma-1)}}, \frac{T_0}{T} = \left(1 + \frac{\gamma - 1}{2} M^2\right) \quad , \frac{\rho_0}{\rho} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{1}{(\gamma-1)}}$$

The area ratio $\frac{A}{A_*}$ is a function of the local Mach number along flow direction x:

$$\left(\frac{A}{A_*}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma + 1}\left(1 + \frac{\gamma - 1}{2} M^2\right)\right]^{\frac{\gamma+1}{\gamma-1}}$$

where A is the local cross-sectional area and A_* is the cross-sectional area at the throat of the nozzle. The above listed equations govern the nozzle flow field and nozzle design once the nozzle constraints are specified in order to determine a unique solution. For example, for a given nozzle profile $A(x)$ and stagnation conditions (P_0, T_0), the nozzle flow field is determined uniquely.

2.2 The correctly expanded nozzle

The structure of the supersonic flow leaving the nozzle is characterized by different regimes. If the pressure P_e at the nozzle exit equals the surrounding pressure P_a, that is $P_e = P_a$, the nozzle is called correctly expanded. The flow in the ideally expanded jet is uniform and supersonic without choking. As quoted in literature (e.g., [Heb01]), the expanded jet may provide the optimal regime for wire-arc spraying process. As an example, we consider the performance of a supersonic axisymmetric nozzle which is designed to produce an ideally expanded jet with $P_e = P_a = 1\ bar$. The design parameters of the nozzle are: the exit radius $r_e = 1.5mm$, the entrance radius $r_o = 15mm$ and the nozzle length $L = 51mm$. The stagnation conditions at the entrance

are: the total pressure $P_o = 4.1bar$ and the total temperature $T_o = 293K$. Assuming that sonic conditions are attained at the throat of the nozzle and using $\gamma = 1.4$ for air flow, the nozzle flow field is defined completely by the quasi-one-dimensional model. The resulting flow properties at the nozzle exit are as follows: the temperature $T_e = 192K$, the density $\rho_e = 1.78kg/m^3$, the Mach number $M_e = 1.576$, the velocity $V_e = 442m/s$.

In order to fabricate the nozzle design, the throat radius r_* has to be defined. It refers to the state of knowledge before $3D$ numerical simulations have been performed. In our case, the critical radius is $r_* = 1.35mm$. Although the quasi-one-dimensional model is unsuitable for processing nozzle profiles, we can construct axisymmetric expanded nozzles of variable geometry applying knowledge from experimental tests and theoretical analysis of the compressible flows, see, e.g., [Bay01,Ale01]. These geometries can be used as first geometry approximation for the optimization process.

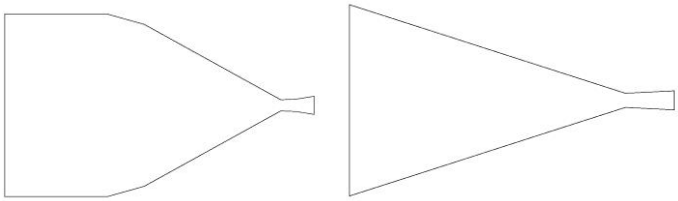

Fig. 1*: Schematics of two axisymmetric nozzle geometries for the above mentioned nozzle constraints: (left) the nozzle with expander half-angle* $\theta = 6$*; (right) the nozzle with expander half-angle* $\theta = 3$.

Variation of the Mach number, M_e, is possible via the total pressure P_o. There are different expanded nozzles ($P_e = P_a = 1\ bar$) with an exit radius $r_e = 1.5mm$ to achieve different Mach numbers at the nozzle exit.

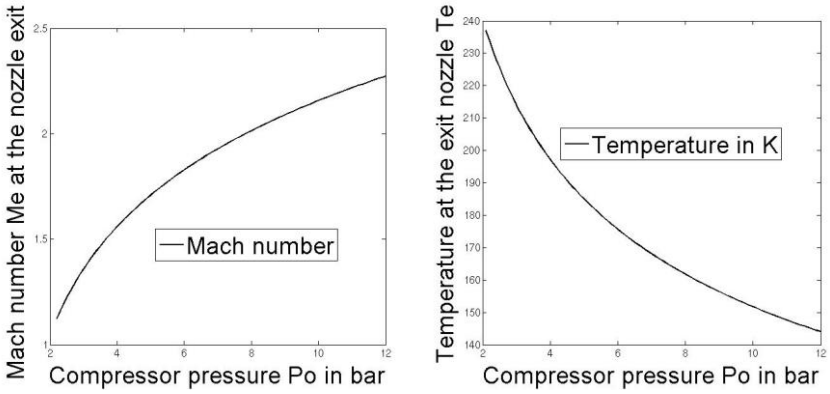

Fig. 2*: Variation of the Mach number,* M_e *(left), and temperature, T (right), at the exit of the expanded nozzle as a function of compressor pressure,* P_o*, with the total temperature,* $T_0 = 293K$.

3 Compressible Flow Solver

3.1 Governing equations

A commonly accepted mathematical model to describe high speed gas flows through nozzles are the inviscid equations of gas dynamics which result from the more general compressible Navier-Stokes equations by neglecting heat conduction and viscous effects. In case of rotationally symmetric nozzles it is even possible to adopt the two-dimensional Euler equations and equip them with a geometric source term to account for the third direction [Tor99]

$$\frac{\partial U}{\partial t} + \frac{\partial F_r(U)}{\partial r} + \frac{\partial F_z(U)}{\partial z} = S(U)$$

Here, the symmetry of the nozzle is assumed around the z-axis so that the second coordinate r represents a measure for the distance from the axis of symmetry z. The vector of conservative variables U, the tuple of inviscid fluxes $\boldsymbol{F} = (F_r, F_z)$ in axial and radial direction and the source term are given by

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, F_r(U) = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ u(\rho E + P) \end{bmatrix}, F_z(U) = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + P \\ v(\rho E + P) \end{bmatrix}, S(U) = -\frac{1}{r}\begin{bmatrix} \rho u \\ \rho u^2 \\ \rho uv \\ u(\rho E + P) \end{bmatrix}$$

where ρ, $\boldsymbol{v} = (u, v)$, and ρE are the density, velocity, and total energy per unit mass, respectively. The pressure P is related to these quantities by the equation of state for an ideal polytropic gas (with adiabatic index $\gamma = 1.4$)

$$P = (\gamma - 1)\rho(E - 0.5|\boldsymbol{v}|^2)$$

The problem is complemented by suitable boundary conditions and compatible initial data $U(r, z, 0) = U_0(r, z)$. Here, we consider boundary conditions of the form $\boldsymbol{n} \cdot \boldsymbol{F} = F_n(U, U_\infty)$ which typically depend on the computed solution values U in the interior and the external state vector U_∞ prescribed at the inlet and outlet as well as the free-slip condition $\boldsymbol{n} \cdot \boldsymbol{v} = 0$ to hold along the walls. In both cases, $\boldsymbol{n} = (n_r, n_z)$ denotes the outward unit vector normal to the boundary.

3.2 Finite element method

In continuous Galerkin finite element methods, the first-order PDE system is cast into its so-called weak form which allows to impose all sorts of boundary conditions in weak sense by using integration by parts. That is, the solution U has to satisfy the following relation for all admissible test functions w

$$\int_\Omega w\frac{dU}{dt} - \frac{dw}{dr}F_r(U) - \frac{dw}{dz}F_z(U)d\Omega + \int_\Gamma wF_n(U,U_\infty)d\Gamma = \int_\Omega wS(U)d\Omega$$

A general framework for the design of high-resolution positivity-preserving schemes for problems if this type is proposed in [Kuz10,Kuz12]. In essence, it consists of the following two steps which are performed in each time step:

1. Compute a provisional solution using a non-oscillatory low-order method
2. Remove excess artificial diffusion from the provisional solution using a mass-conserving flux correction post-processing procedure

3.3 Low-order method

Let U and the inviscid fluxes be approximated by the same set of (bi-)linear finite element basis functions $\{\varphi_i\}_{i=1}^N$. Then the semi-discrete form of the non-oscillatory low-order method for an individual degree of freedom i reads

$$m_i\frac{dU_i^L}{dt} = \sum_{j\neq i}\boldsymbol{c}_{ji}\cdot\boldsymbol{F}_j - \boldsymbol{c}_{ij}\cdot\boldsymbol{F}_i + D_{ij}\left(U_j^L - U_i^L\right) - \int_\Gamma \varphi_i F_n(U^L,U_\infty)d\Gamma + m_i S(U_i^L)$$

Here, the lumped mass matrix entries and the coefficients $\boldsymbol{c}_{ij}$ are given by

$$m_i = \sum_j m_{ij}, \quad m_{ij} = \int_\Omega \varphi_i\varphi_j d\Omega, \quad \boldsymbol{c}_{ij} = \int_\Omega \varphi_i\nabla\varphi_j d\Omega$$

where the gradient in the last integral term refers to derivatives in r and z-direction. It should be noted that the implementation of highly efficient numerical integration procedures within an unstructured finite element code is a nontrivial task due to cache misses and uncoalesced memory accesses stemming from indirect addressing. The need for proper synchronization of memory transactions imposes an additional burden if it comes to parallel assembly. Noteworthy, the above coefficients remain unchanged as long as the mesh is fixed, and hence, they can be precomputed once and for all and stored in optimal order for later use. This feature is very attractive for modern streaming processors which achieve highest performance if the same instruction has to be applied to multiple data delivered as contiguous streams.

Different options exist for choosing the artificial viscosity operator D_{ij}. A very simple but robust choice is to use scalar dissipation of Rusanov-type [Kuz10]

$$D_{ij} = \max(d_{ij},d_{ji})I, \quad d_{ij} = \left|\boldsymbol{e}_{ij}\cdot\boldsymbol{v}_j\right| + \left|\boldsymbol{e}_{ij}\right|c_j$$

where $c_i = \sqrt{\gamma P_i/\rho_i}$ is the speed of sound at node i and $\boldsymbol{e}_{ij} = 0.5(\boldsymbol{c}_{ji} - \boldsymbol{c}_{ij})$. In our experience, choosing artificial viscosities based on Roe-type linearization tends to produce non-positive pressure values at the nozzle exit, and hence, this strategy cannot be used within a positivity-preserving numerical schemes.

A similar behavior was observed at the nozzle exit if the normal flux $F_n(U_i^L, U_\infty)$ was calculated by Roe's approximate Riemann solver. As a remedy, scalar dissipation of Rusanov-type is used for calculating the normal flux as follows

$$F_n(U_i^L, U_\infty) = \boldsymbol{n} \cdot \frac{\boldsymbol{F}_i + \boldsymbol{F}(U_\infty)}{2} - \frac{1}{2} D_{i\infty}(U_i^L - U_\infty)$$

adopting $D_{i\infty} = \max(|\boldsymbol{n} \cdot \boldsymbol{v}_i| + c_i, |\boldsymbol{n} \cdot \boldsymbol{v}_\infty| + c_\infty) I$ as artificial diffusion operator.

For the discretization in time, either the Crank-Nicolson time stepping scheme ($\theta = 0.5$) or the fully implicit backward Euler method ($\theta = 1.0$) is used

$$[M - \theta \Delta t(K + D)]U^{n+1} = [M + (1 - \theta)\Delta t(K + D)]U^n + S^{n+\theta}$$

depending on whether the temporal evolution of the flow and the formation of the free stream during the startup phase is of interest or not. In either case, the entries of operator K are defined making use of the homogeneity property $\boldsymbol{c} \cdot \boldsymbol{F} = \boldsymbol{c} \cdot \boldsymbol{A}U$ of the Euler equations, where $\boldsymbol{A}$ denotes the flux Jacobian matrix.

3.4 Flux correction algorithm

Once the low-order predictor U^L has been computed the loss of accuracy incurred by mass lumping and the addition of artificial dissipation can be reduced by adding a limited amount of compensating anti-diffusion

$$m_i U_i = m_i U_i^L + \Delta t \sum_{j \neq i} \alpha_{ij} \left[m_{ij} \left(\frac{dU_i^L}{dt} - \frac{dU_j^L}{dt} \right) + D_{ij}(U_i^L - U_j^L) \right]$$

The correction factors α_{ij} which vary between zero and one are determined by Zalesak's limiting algorithm. The details are given in [Kuz10,Kuz12].

3.5 Parallel implementation

The summations present in the low-order scheme and the flux-correction procedure extend over all edges ij connecting nodes i and j. Moreover, Zalesak's limiting algorithm consists of one iteration over all nodes and two edge-loops which require a more sophisticated parallelization strategy.

A viable approach to implement edge-based assembly procedures in parallel is as follows: A classical edge-coloring algorithm is applied to the finite element sparsity graph and the edges are reordered accordingly. As a result,

edges within the same color group share no common nodes and can therefore be processed in parallel without the need to synchronize memory transactions. Once the starting position of each color group is stored, edge-based loops can be implemented using a sequential outer loop over the color groups and a fully parallelized internal loop. Figure 3 illustrates the speedup in CPU time obtained with an OpenMP implementation of the edge-based inviscid flux assembly procedure. The computation was performed on an Intel Xeon X5680 with two 6-core CPUs running at 3.33GHz. Although the precomputed coefficients and auxiliary integer data are stored contiguously, this algorithm is memory-bound due to the indirect addressing occurring in gathering and scattering of (solution) data from nodal vectors. This might be improved by using cache-efficient renumbering strategies proposed in [Löh10]. The overall speedup of a complete simulation run is demonstrated in Figure 4.

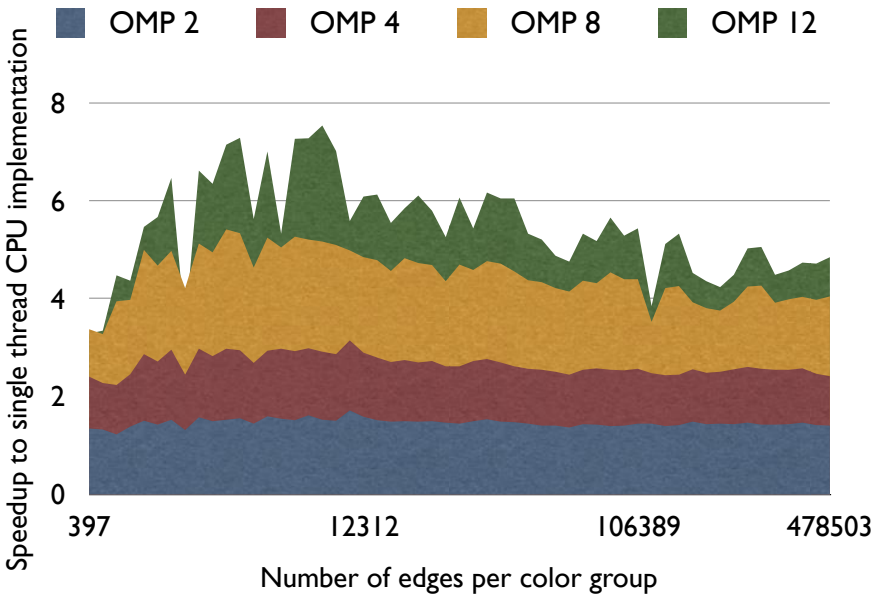

Fig. 3: *Speedup in CPU time for edge-based inviscid flux assembly.*

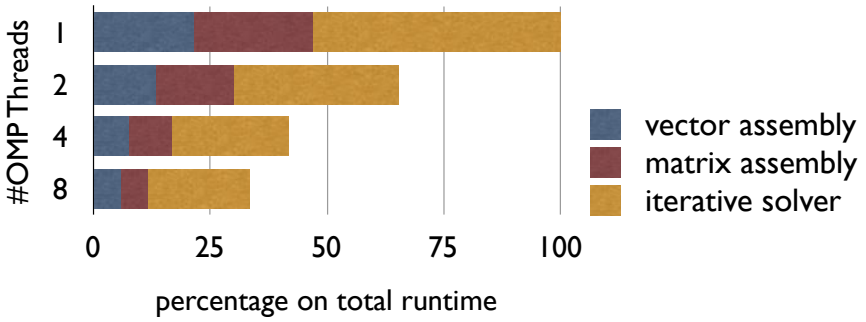

Fig. 4: *Speedup in CPU time for the complete simulation run.*

4 Parallel derivative-free optimization methods

Another aim of subproject B7 is to apply an optimization solver to the described compressible flow solver in rotationally symmetric nozzle configurations. It should be applicable to other numerical simulation tasks inside of the SFB 708, for instance for the simulation of grinding processes as well as the simulation of springback effects during metal forming. Hence, the underlying optimization method should be derivative-free, applicable together with a (more or less) arbitrary simulation code, and it should exploit the full potential of the LiDo Cluster of the TU Dortmund university (as an example for Torque/Maui based cluster systems) by supporting parallel optimization.

First of all, we checked and analyzed different freely available optimization codes in how far they can be used for our purposes, namely:

- **IMFIL, MCS, NEWUOA, SNOBFIT:** These are research codes written in MATLAB which are available as source code. However, there is no support for parallel processing, *http://ab-initio.mit.edu/wiki/index.php/NLopt_Algorithms*.
- **PSwarm:** This is a special global optimization code based on pattern search and particle swarm. It supports parallel processing, although it is basically a research code and is not intended for this purpose, *http://www.norg.uminho.pt/aivaz/pswarm*.
- **NLopt:** NLopt is a library of different codes for nonlinear optimization plus global optimization. It supports, e.g., the global optimization approach DIRECT and a number of local optimization algorithms like COBYLA, BOBYQA, Nelder Mead and others. Derivative-free and derivative-based optimization approaches based on reconstructed gradients are supported, however not in a parallel fashion, *http://ab-initio.mit.edu/nlopt.*
- **CONDOR:** CONDOR is a parallel Trust-Region algorithm based on the UOBYQA algorithm of Powell. The algorithm itself is available via a published paper, but neither the source code nor binaries of the software are available. Therefore, little is known about the algorithm and thus, it can hardly be used in this project, *http://dx.doi.org/10.1016/j.cam.2004.11.029*.
- **NOMAD:** NOMAD is a C++ optimization package for Black-Box optimization which is based on the MADS algorithm (for which a

convergence analysis is available). It is available for most platforms and also supports parallel processing via MPI, *http://www.gerad.ca/nomad*.

- **TOMLAB:** TOMLAB is a collection of optimization algorithms provided by TomOpt. The algorithms in the package belong to the most powerful of todays derivative-free and derivative-based optimization packages. Unfortunately, the package is commercial and thus, no source code is freely available *http://tomopt.com/tomlab*.
- **APPSPACK:** The APPSPACK package has been succeeded by the HOPSPACK package *https://software.sandia.gov/appspack/version5.0/index.html.*
- **HOPSPACK:** HOPSPACK is a parallel pattern search package for nonlinear programs developed by SANDIA National Laboratories. It is freely available. However, the development has been stopped and it includes only a very basic compass search algorithm. The package has been included in the DAKOTA package *https://software.sandia.gov/trac/hopspack.*
- **DAKOTA:** DAKOTA is a kind of meta-package from the SANDIA National Lab. for optimization and parameter studies. It does not provide algorithms itself but realizes a common interface to a set of optimization packages of other vendors like, e.g., HOPSPACK, CONMIN, COLINY and others. The package supports linear and nonlinear constraints, surrogate-based minimization, standard nonlinear optimization, Pareto Set optimization, nonlinear least squares parameter fitting and optimization under uncertainty. For derivative-based optimization algorithms, derivatives can automatically be computed by reconstructed gradients only using function evaluations. The package can be used as standalone package or as library, allowing parallel processing *http://dakota.sandia.gov/*.

To draw an overall conclusion, there are basically only three freely available software packages that support parallel processing, namely NOMAD, HOPSPACK and DAKOTA. It turned out that from this list, the DAKOTA package is the most powerful package of all analyzed packages. It supports parallel processing out-of-the-box and provides a rather simple interface to any simulation code developed in the SFB 708. It also provides a couple of very powerful state-of-the-art optimization algorithms including surrogate models and uncertainty.

4.1 Current Developments

Based on the DAKOTA package as our currently preferred optimization code, we performed preliminary optimization runs for various nozzle-like designs. Results for a low Mach number simulation of a 2D nozzle are depicted in Figure 5. As a proof-of-concept, we combined DAKOTA with our FEATFLOW software. Here, we carried out a 2-parameter optimization for a stationary flow configuration (with (non-dimensional) inflow speed 0.3 as parabolic profile and domain length 0.3; the height of the nozzle at inlet is 1). The optimization process starts initially at y=(0,0.75). The aim was to find the two y-positions (red) on the outflow section of the nozzle-like domain such that the velocity in the center of the outflow (the green vector) is as close as possible to (1,0).

After 53 iterations, the coordinates y=(-0.15,0.14) have been obtained, and the outflow vector is sufficiently close to (1,0), as expected. Due to the proof-of-concept character of this test case, this optimization was done in serial. Currently, we are going to use DAKOTA in parallel on the LiDo cluster with more advanced optimization strategies than it was used in the shown test example. Moreover, the next step will be to apply DAKOTA for the design of the LiBo nozzles in combination with the compressible flow solver described in Section 3. To check the quality of the corresponding results, and also for validation purposes, we will compare in a first step the obtained results with the described analytical 1D models which however are based on very special assumptions. Then, in the next step, we will perform optimization runs for the full gas flow solver including a much wider variety of physical effects.

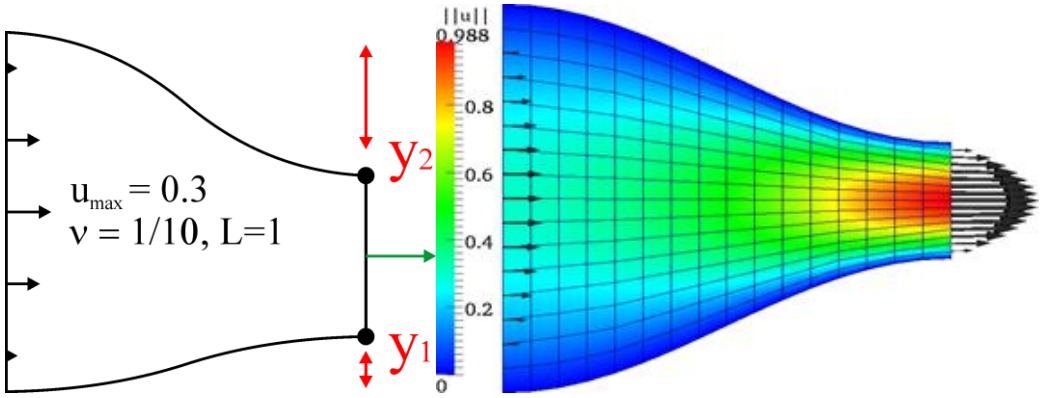

Fig. 5: *2D nozzle domain and preliminary optimization results.*

5 Acknowledgement

The authors gratefully acknowledge the financial support of the DFG (German Research Foundation) within the Collaborative Research Centre SFB 708, project B7.

References

[Ale01] Louisos, W. F.; Alexeenko, A. A.; Hitt, D. L.; Zilic, A.: Design considerations for supersonic micronozzles International Journal of Manufacturing Research (2008), 3(1), 80-113

[And01] Anderson, J. D.: Modern Compressible Flow with Historical Perspective, 3rd ed., (2003), McGraw-Hill, New York, NY.

[Bay01] Bayt, R. L.; Breuer, K. S.: System Design and Performance of Hot and Cold Supersonic Micro-jets, (2001), AIAA Paper 2001-0721.

[Heb01] Wang, X.; Russ, S.; Heberlein, J.; Gerberich, W.; Pfender, E.: Arc Spray by a Supersonic Atomizing Gas Stream, INTERNATIONAL SYMPOSIUM ON PLASMA CHEMISTRY 1993, 1, 133-138.

[Kuz10] Kuzmin, D. Möller, M.; Shadid, J.N.; Shashkov, M.: Failsafe flux limiting and constrained data projections for equations of gas. J. Comput. Phys. 229 (2010) no. 23, 8766-8779.

[Kuz12] Kuzmin, D.; Möller, M.; Gurris, M.; Algebraic Flux Correction II. Compressible Flow Problems. Chapter 6 in: Kuzmin, D.; Löhner, R.; Turek, S. (Eds.) Flux-Corrected Transport: Principles, Algorithms, and Applications. Springer, 2nd ed, 2012.

[Löh10] Löhner, R.: Cache-efficient renumbering for vectorization. Int. J. Numer. Method Biomed Eng. 26 (2010) no.5, 628-636.