V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

EFFICIENCY OF GEOMETRIC MULTIGRID METHODS FOR SOLVING THE SENSITIVITY EQUATIONS WITHIN GRADIENT BASED FLOW OPTIMIZATION PROBLEMS

J. Michaelis*, J. Siegmann, G. Becker, M. Schäfer

Institute of Numerical Methods in Mechanical Engineering
Technische Universtität Darmstadt
Dolivostr. 15, 64293 Darmstadt, Germany
*e-mail: michaelis@fnb.tu-darmstadt.de

Key words: Geometric multigrid methods, sensitivity equations, gradient-based flow optimization, finite-volume method, shape optimization

Abstract. *This paper presents a geometric multigrid approach for solving the sensitivity equations derived by differentiating the Navier-Stokes equations with respect to control parameters. The considered control variables can be arbitrary flow parameters like the inflow velocity or the control points of a boundary surface represented by NURBS. The SIMPLE method with an embedded SIP solver (ILU) is employed as smoother in the framework of V-cycles. The multigrid approach is examined with respect to the computational costs for generic test cases, i.e., a channel flow over a bump with variable inlet velocity and a channel with a parameter controlled NURBS surface.*

1 INTRODUCTION

Numerical flow simulation is a well-established field in modern engineering. Numerical algorithms support practical experiments by providing cost-efficient and risk-free methodologies for flow engineers. Due to continuously increasing CPU power and the development of more sophisticated algorithms a new research field got into the spotlight of interest. The automated flow optimization process is an ongoing topic in todays research.

The major challenge is to provide algorithms which are able to deal with arbitrary objective functionals depending on predefined design parameters (e.g., geometry deformation, wall temperature, inlet velocity) and state variables (e.g., velocity, pressure, and temperature distribution). The algorithms are also required to be fast and robust enough to find an optimal solution within a short period of time, even if a large number of different control parameters are considered. An example for such a challenging optimization problem is the redesign of a technical component within an engine. In this case the components geometry might be variable within certain bounds, as well as the temperature at the surface or the velocity of the fluid before entering the engine. The objective of the redesign could be, for instance, to minimize the pressure loss within the component, while assuring an acceptable fluid temperature distribution.

Derivative-based methods are one approach to solve these optimization problems. Therefore, it is necessary to compute the gradient of the objective functional. An efficient way to compute this gradient is to solve the continuous sensitivity equations (CSE). These equations arise from differentiating the Navier-Stokes equations with respect to each design parameter. Solving these equations provides knowledge about the influence of the different control parameters and allows to compute the functionals gradient. Every control parameter generates an own set of CSE. Therefore, fast solving strategies are necessary.

In this paper we present the theoretical background of the CSE. Thereafter, we introduce the geometric multigrid concept applied to solve the CSE. The efficiency of the solving strategy is examined for two generic design parameter test cases.

2 GRADIENT BASED OPTIMIZATION

A general optimization problem consists of a given objective functional $\mathcal{J}$ depending on a set of state variables $\boldsymbol{\phi}$ (e.g., velocity, pressure) and some control variables $\mathbf{a}$ (e.g., shape parameters, inlet velocity) and probably some additional constraints $\mathbf{F}$. In mathematical notation this can be written as

$$\min_{\mathbf{a}} \ \mathcal{J}\left(\boldsymbol{\phi}\left(\mathbf{a}\right), \mathbf{a}\right) \quad \text{subject to} \quad \mathbf{F}\left(\boldsymbol{\phi}\left(\mathbf{a}\right), \mathbf{a}\right) = \mathbf{0}. \tag{1}$$

In a flow optimization problem the main constraints are the partial differential equations which describe the flow field presented in Section 3. Other side constraints are also conceivable, like restrictions to the inlet velocity to prevent transition to turbulence or limitations to the maximal deformation of the fluid area due to geometrical reasons.

Gradient-based optimization algorithms utilize functional evaluations as well as information about the gradient of the functional, whose i-th component can be written as

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}a_i} = \frac{\partial \mathcal{J}}{\partial \phi_j}\frac{\partial \phi_j}{\partial a_i} + \frac{\partial \mathcal{J}}{\partial a_i}, \tag{2}$$

where $\mathrm{d}/\mathrm{d}a_i$ denotes the total derivative with respect to the control component a_i. In this formulation and below we use Einstein's summation notation.

A sophisticated approach for computing the gradient is to calculate $\partial \boldsymbol{\phi}/\partial a_i$ from the sensitivity equations (described in Section 4) and to assemble the gradient from equation (2). However, it is necessary to solve for each design parameter one sensitivity equation system, hence the development of fast computation algorithms is of special interest.

3 FLOW EQUATIONS

The incompressible and steady flows of interest are modeled by the Navier-Stokes equations. The conservation equations of mass and momentum can be written as

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{3}$$

$$\frac{\partial\left(\rho u_i u_j\right)}{\partial x_j} - \frac{\partial\left(\mu \tau_{ij}\right)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \rho f_i \tag{4}$$

with

$$\tau_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}, \tag{5}$$

where ρ and μ represent the fluid density and dynamic viscosity, respectively, u_i is the i-th velocity component, p is the pressure, and f_i is the i-th component of a body force. The steady equation system is closed with proper boundary conditions – both Dirichlet and Neumann conditions can be applied. In this paper we focus on Dirichlet boundary conditions

$$u_i = \bar{u}_i \quad \text{on } \Gamma_D, \tag{6}$$

where a prescribed velocity value $\bar{u}_i$ is set along the boundary. Solely at an outlet we assume a zero gradient Neumann boundary condition, which is sufficient when the outflow region is far enough from the main region of interest.

4 CONTINUOUS SENSITIVITY EQUATIONS

Two general approaches for computing the flow sensitivities are known and described in detail in [6]. One method is discretizing the Navier-Stokes equations first and differentiating the discretized system. Literature about this strategy is widley released, e.g., [8, 10, 14]. This approach is called discretize-then-differentiate and is often realized in connection with an automatic differentiating software, working on the source code of the

flow solver. We focus on the second method, the differentiate-then-discretize approach. Hence the underlying state equations are differentiated first in order to obtain a continous set of equations for the sensitivities. These equations are discretized and solved by any numerical method.

The continuous sensitivity equations (CSE) according to an arbitrary control parameter a can be obtained by implicit differentiation of the state equations (3) and (4). For sake of simplicity we introduce following notation

$$\frac{\partial u_i}{\partial a} = s^a_{u_i}, \qquad \frac{\partial p}{\partial a} = s^a_p, \qquad \frac{\partial \mu}{\partial a} = s^a_\mu, \qquad \frac{\partial \rho}{\partial a} = s^a_\rho, \qquad \frac{\partial f_i}{\partial a} = s^a_{f_i}. \tag{7}$$

Using (7) the CSE become

$$\frac{\partial s^a_{u_j}}{\partial x_j} = 0, \tag{8}$$

$$\frac{\partial}{\partial x_j}\left(s^a_\rho u_i u_j + \rho s^a_{u_i} u_j + \rho u_i s^a_{u_j}\right) = -\frac{\partial s^a_p}{\partial x_i} + s^a_\rho f_i + \rho s^a_{f_i} + \frac{\partial}{\partial x_j}\left(s^a_\mu\, \tau_{ij} + \mu s^a_{\tau_{ij}}\right) \tag{9}$$

with

$$s^a_{\tau_{ij}} = \frac{\partial s^a_{u_i}}{\partial x_j} + \frac{\partial s^a_{u_j}}{\partial x_i}. \tag{10}$$

If we assume that the parameter a has no influence on the fluid properties ρ and μ, as well as on the body force f_i, the equations (8) and (9) simplify to

$$\frac{\partial s^a_{u_j}}{\partial x_j} = 0, \tag{11}$$

$$\frac{\partial}{\partial x_j}\left(\rho s^a_{u_i} u_j\right) - \frac{\partial}{\partial x_j}\left(\mu s^a_{\tau_{ij}}\right) = -\frac{\partial s^a_p}{\partial x_i} - \frac{\partial}{\partial x_j}\left(\rho u_i s^a_{u_j}\right). \tag{12}$$

It is necessary to close the CSE system. For this the boundary conditions need to be differentiated with respect to the control parameter a as well. Hence, the zero gradient condition is retained at the outlet, and the Dirichlet boundary conditions are implicitly differentiated as done with the state equations. However, if a is a shape parameter, the boundary geometry itself is affected by changes of the control parameter. For this the Dirichlet boundary condition (6) must be stated more precisely as $u_i(x(a), y(a), z(a), a) = \bar{u}_i(x(a), y(a), z(a), a)$ on $\Gamma_D(a)$. Implicit differentiation of this condition, as described in [4], yields

$$\frac{\mathrm{d}u_i}{\mathrm{d}a} = \frac{\mathrm{d}\bar{u}_i}{\mathrm{d}a} \quad \text{on } \Gamma_D(a), \tag{13}$$

$$\Leftrightarrow \quad \frac{\partial u_i}{\partial a} + \frac{\partial u_i}{\partial x_j}\frac{\mathrm{d}x_j}{\mathrm{d}a} = \frac{\mathrm{d}\bar{u}_i}{\mathrm{d}a} \quad \text{on } \Gamma_D(a), \tag{14}$$

where $\mathrm{d}/\mathrm{d}a$ denotes the total derivative with respect to the shape parameter a. Using equation (14) leads to the boundary condition for the sensitivity at the manipulable boundary

$$s^a_{u_i} = \frac{\mathrm{d}\bar{u}_i}{\mathrm{d}a} - \frac{\partial u_i}{\partial x_j}\frac{\mathrm{d}x_j}{\mathrm{d}a} \quad \text{on } \Gamma_D(a). \tag{15}$$

The sensitivity of the shape position $\mathrm{d}x_j/\mathrm{d}a$ appears, as well as the gradient of the flow velocities at the boundary $\partial u_i/\partial x_j$.

The required parametrization of the control variable dependent geometry surface can, for instance, be realized by using NURBS surfaces whose control point positions depend on the design parameter a, as described in [1]. Thus, the surface coordinate vector can be written as

$$\mathbf{x}(s,t,a) = \sum_{i=1}^{\hat{n}}\sum_{j=1}^{\hat{m}} \mathcal{N}_{i,p}(s)\,\mathcal{N}_{j,q}(t)\,\mathbf{P}_{i,j}(a) \quad s,t \in [0,1], \tag{16}$$

where s,t are the normalized parametrization variables along the surface directions. $\hat{n}$ and $\hat{m}$ denotes the number of control points in each direction, p and q are the order of the B-Spline basis functions $\mathcal{N}$ in the respective direction, and $\mathbf{P}_{i,j}(a)$ denotes the coordinates of the (i,j)-th control point which is depending on design parameters. Utilizing this parametrization towards the computation of $\mathrm{d}x_j/\mathrm{d}a$, the boundary condition for the sensitivity (15) can be transformed into

$$s^a_{\mathbf{u}} = \frac{\mathrm{d}\bar{\mathbf{u}}}{\mathrm{d}a} - \nabla\mathbf{u}\cdot\left(\sum_{i=1}^{\hat{n}}\sum_{j=1}^{\hat{m}} \mathcal{N}_{i,p}(s)\,\mathcal{N}_{j,q}(t)\,\frac{\mathrm{d}}{\mathrm{d}a}\mathbf{P}_{i,j}(a)\right) \quad \text{on } \Gamma_D(a) \tag{17}$$

as described in [13].

5 NUMERICAL SOLVING STRATEGY AND DISCRETIZATION

Before computing the flow sensitivities, the state equations (3) and (4) need to be solved in every iteration. This is done using our in-house flow solver FASTEST. It applies a fully conservative finite-volume approach to solve the incompressible Navier-Stokes equations on a non-staggered, block-structured and cell-centered grid, see [3]. The implemented solver for the sensitivity equations system (11) and (12) is embedded in the existing flow solver.

Analogous to the Navier-Stokes equations the spatial discretization of the sensitivity equation utilizes a finite-volume method with a collocated variable arrangement, in contrast to the commonly used finite element approaches [9, 12, 16]. The solution domain is discretized into hexahedral (in general non-orthogonal) control volumes (CVs). Skipping further details of the discretization (see, e.g., [5]), this procedure leads for the n-th iteration to a linear equation system of the form

$$\mathbf{A}^h\mathbf{s}^{n,h}_{\mathbf{u}} = \mathbf{0}, \tag{18}$$

$$\mathbf{B}^h\left(\mathbf{u}^{N,h}\right)\mathbf{s}^{n,h}_{\mathbf{u}} + \mathbf{C}^h\mathbf{s}^{n,h}_p = \mathbf{D}^h\left(\mathbf{u}^{N,h}, \mathbf{s}^{n-1,h}_{\mathbf{u}}, \mathbf{s}^{n-1,h}_p\right), \tag{19}$$

where $\mathbf{A}^h, \mathbf{B}^h, \mathbf{C}^h$ and $\mathbf{D}^h$ are discrete operators resulting from the discretization procedure on a grid with characteristic grid length h (e.g. maximum CV width). The operator $\mathbf{B}^h$ depends on the state variables $\mathbf{u}^{N,h}$, which are predetermined in N iterations. Equation (19) is linear in $\mathbf{s}_\mathbf{u}^{n,h}$, the vector of unknown velocity sensitivities in the n-th iteration. The unknown pressure sensitivities are represented by the vector $\mathbf{s}_p^{n,h}$. The source term operator $\mathbf{D}^h$ on the right hand side depends on the predetermined flow velocities and on the sensitivities from the previous iteration $(\mathbf{s}_\mathbf{u}^{n-1,h}, \mathbf{s}_p^{n-1,h})$.

Due to the internal coupling of the pressure sensitivity and the sensitivities of the velocity in equation (19), but the absence of the pressure sensitivity in the differentiated continuity equation (18), a pressure correction algorithm of the SIMPLE type [5] with an embedded SIP solver (ILU) is used to solve the equation system. This strategy is analogous to the procedure solving the incompressible Navier-Stokes equations. This algorithm can serve as a smoother in a geometric multigrid concept.

6 MULTIGRID METHODS

Conventional iterative solvers usually show a drastic increase of the computational effort with decreasing grid spacing. In particular, for 3-dimensional problems the cost swells rapidly. Using a Fourier series expansion of the error it is possible to show that a solver as described in the previous section, removes efficiently errors with a wavelength of about the characteristic control volume size h (see, e.g., [11]). Error components with a smaller frequency, i.e., a longer wavelength, are only slowly eliminated.

Multigrid algorithms are based on the idea to solve an equation system involving a hierarchy of meshes (see, e.g., [2, 7]). We employ a geometric multigrid algorithm where the different grid levels are generated by coarsening steps from the finest grid.

The linear equation system (18) and (19) can be written shortly as

$$\mathbf{L}^h \mathbf{s}^h = \mathbf{b}^h, \tag{20}$$

where $\mathbf{s}$ summarizes the unknown sensitivities and the superscript h emphasizes the spatial discretization. Although this is a linear system we employ the full approximation scheme (FAS), a non-linear multigrid algorithm. For linear problems the linear multigrid algorithm and the FAS are analytically equivalent. However, in contrast to linear multigrid algorithms, FAS does not approximate the error on different grid levels, but improves directly the solution of the sensitivities on the different grids. In future studies, this can be used for adaptive optimization methods, wherein the gradient is computed simultaneously with low computational effort on the coarse grid levels. Utilizing an error estimation it is possible to extend the multigrid cycle to finer grids during the optimization process. Thereby, more accurate gradient approximations can be obtained near the optimum. Furthermore some advanced techniques like τ-extrapolation base on the FAS approach, see [15]. These reasons motivate the usage of the non-linear algorithm on a linear problem as well. Hence, we write equation (20) in non-linear notation

$$\mathbf{L}^h \left(\mathbf{s}^h\right) = \mathbf{b}^h. \tag{21}$$

The SIMPLE algorithm yields after m iterations towards an approximation of the solution, which satisfies equation (21) up to a residual vector $\mathbf{r}^h$. The non-linear defect equation reads:

$$\begin{aligned} & \mathbf{L}^h\left(\mathbf{s}^h\right) && = \mathbf{b}^h \\ \Leftrightarrow \quad & \mathbf{L}^h\left(\mathbf{s}^{m,h}+\mathbf{s}^h-\mathbf{s}^{m,h}\right)-\mathbf{L}^h\left(\mathbf{s}^{m,h}\right) && = \mathbf{b}^h-\mathbf{L}^h\left(\mathbf{s}^{m,h}\right) \\ \Leftrightarrow \quad & \mathbf{L}^h\left(\mathbf{s}^{m,h}+\mathbf{e}^h\right)-\mathbf{L}^h\left(\mathbf{s}^{m,h}\right) && = \mathbf{r}^h, \end{aligned} \tag{22}$$

where $\mathbf{e}^h = \mathbf{s}^h - \mathbf{s}^{m,h}$ is the current error on grid level h. The idea is to solve the defect equation (22) on the coarser grid levels. Hence, the computed fine grid values are restricted by a restriction operator $\mathcal{R}_h^{2h}$ to the next coarser grid level $(2h)$. For building up the new discretization $\mathbf{L}^{2h}$ on the coarser grid we can make use of the conservativity of the finite volume method, e.g., the flux over the faces of the assembled coarse grid CV is equal to the fluxes over the faces of the eight underlying fine grid CVs. After the restriction we transform equation (22) into

$$\mathbf{L}^{2h}\left(\mathcal{R}_h^{2h}\left(\mathbf{s}^{m,h}\right)+\mathbf{e}^{2h}\right)=\mathcal{R}_h^{2h}\left(\mathbf{r}^h\right)+\mathbf{L}^{2h}\left(\mathcal{R}_h^{2h}\left(\mathbf{s}^{m,h}\right)\right), \tag{23}$$

wherein the coarse grid variable $\mathbf{s}^{2h} = \mathcal{R}_h^{2h}(\mathbf{s}^{m,h}) + \mathbf{e}^{2h}$ is defined and the known terms on the right hand side are merged. We obtain the new coarse grid defect equation

$$\mathbf{L}^{2h}\left(\mathbf{s}^{2h}\right)=\mathbf{b}^{2h}, \tag{24}$$

which is similar to equation (21) except for the grid spacing. Thus, it is possible to use in turn coarser meshes to solve this defect equation. After finding an approximative solution $\tilde{\mathbf{s}}^{2h}$ for equation (24), by solving directly or employing further recursions using more grid levels, it is necessary to adapt the error correction back to the fine grid. An interpolation operator $\mathcal{I}_{2h}^h$ prolongates the found coarse grid defect $\mathbf{e}^{2h}$ to the fine grid

$$\mathbf{e}^h=\mathcal{I}_{2h}^h\left(\mathbf{e}^{2h}\right)\approx\mathcal{I}_{2h}^h\left(\tilde{\mathbf{s}}^{2h}-\mathcal{R}_h^{2h}\left(\mathbf{s}^{m,h}\right)\right), \tag{25}$$

which can be used iteratively to update the previously computed fine grid solution

$$\mathbf{s}^h=\mathbf{s}^{m,h}+\mathbf{e}^h. \tag{26}$$

This updated solution is iterated again by the SIMPLE algorithm to smooth high-frequency errors appearing from the interpolations between the different grid levels. We use V-cycles to move through the grid hierarchy.

The utilization of the above procedure causes an enormous acceleration in the computation of the sensitivities, as we will show in the next section.

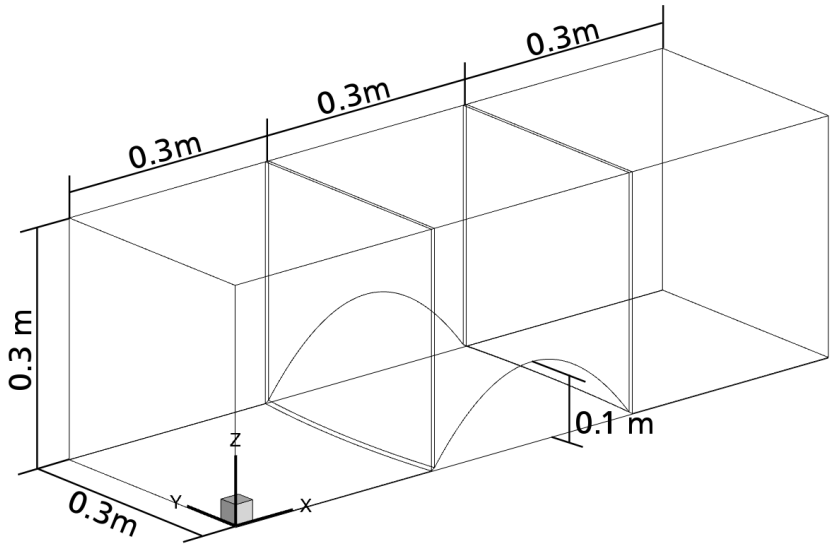

Figure 1: Geometric setup for 3-dimensional flow over NURBS-generated bump

7 VERIFICATION AND MULTIGRID ANALYSIS

For the verification of the multigrid implementation, a 3-dimensional channel flow over a NURBS-generated bump in the middle section is analyzed. The channel geometry is given in Figure 1. Its length is 0.9 m and the cross-section is quadratic with a height and depth of 0.3 m each. The inlet is located at $x = 0.0$ m and the outlet is situated at $x = 0.9$ m. All the others surrounding channel faces are impermeable walls. The bump ranges from $x = 0.3$ m to $x = 0.6$ m, its peak reaches $z = 0.1$ m. A viscous fluid with $\rho = 1400\,\mathrm{kg/m^3}$ and $\mu = 10\,\mathrm{kg/ms}$ is used to assure a Reynolds number in the range of 10 for inlet velocities of about 0.1 m/s.

To generate a block-structured mesh the geometry is divided into three blocks, each block has 64 CVs in each spatial direction. This allows to employ six grid levels for the computation of the flow variables and the sensitivity values. For each grid coarsening step we halve the number of nodes in each direction, ending up with only eight CVs per block on the coarsest mesh (see Table 1).

Grid level	1	2	3	4	5	6
# CV	24	192	1536	12288	98304	786432

Table 1: Number of control volumes on the different grid levels

From literature (e.g., [3]) it is well-known that the computational cost scales approximately linear with the number of CVs while applying a multigrid algorithm. The com-

putation time for solving the sensitivity equation system is measured on a Intel Core2 Duo E8400 (3.00 GHz) with 4096 MB main memory. The measured time is averaged over several runs to minimize temporal fluctuation. Another characteristic quantity is the number of necessary fine grid iterations, which usually remain almost constant with grid refinement (e.g.,[11]).

Two different control settings are examined to verify this typical multigrid behavior. In the first test case the inlet velocity is set by a design parameter. In the second one the surface of the bump is controlled by a shape variable.

7.1 Variable Inlet Parameter

We apply the following parabolic x-velocity profile at the inlet

$$u = 0.1\,\alpha\,\frac{y\,(0.3-y)}{0.15^2}\,\frac{z\,(0.3-z)}{0.15^2}\,\frac{\mathrm{m}}{\mathrm{s}}\,,\ \ \alpha \geq 0, \tag{27}$$

with the control parameter α. The y- and z-component are set to zero.

For the sensitivity computation V-cycles of maximum depth are performed, e.g., on the second grid level a V-cycle over two grids is used, while on the finest grid level a V-cycle over all six meshes is employed. On each grid level we perform 5 SIMPLE iterations and restrict the obtained defect equation to the next coarser grid. Only on the coarsest level 10 SIMPLE iterations are performed, before interpolating the error correction back to the finer grids. On the way back to the finest grid 3 SIMPLE iterations are used on each grid to smooth the high frequent errors resulting from the interpolations between the hierarchical levels. For better comparability of the different sized V-cycles the underrelaxation factors within the SIMPLE algorithm are identically chosen on same grids. The sensitivity calculation is converged when the maximum residual value on the finest mesh is smaller then 10^{-6}.

The results are given in Table 2. Although the number of fine grid iterations is not perfectly constant, the computational time increases almost linearly with the number of CVs. This correlation is shown in Figure 2 where the computational time is plotted against the number of control volumes in a double logarithmic diagramm. The continous line shows the theoretically expected linear increase of the computational costs.

Scheme (# CVs)	SG1 (24)	MG2 (192)	MG3 (1536)	MG4 (12288)	MG5 (98304)	MG6 (786432)
Fine grid iterations	132	105	83	94	118	176
Time (sec)	0.05	0.13	0.43	3.50	41.18	510.66

Table 2: Multigrid behavior for variable inlet parameter

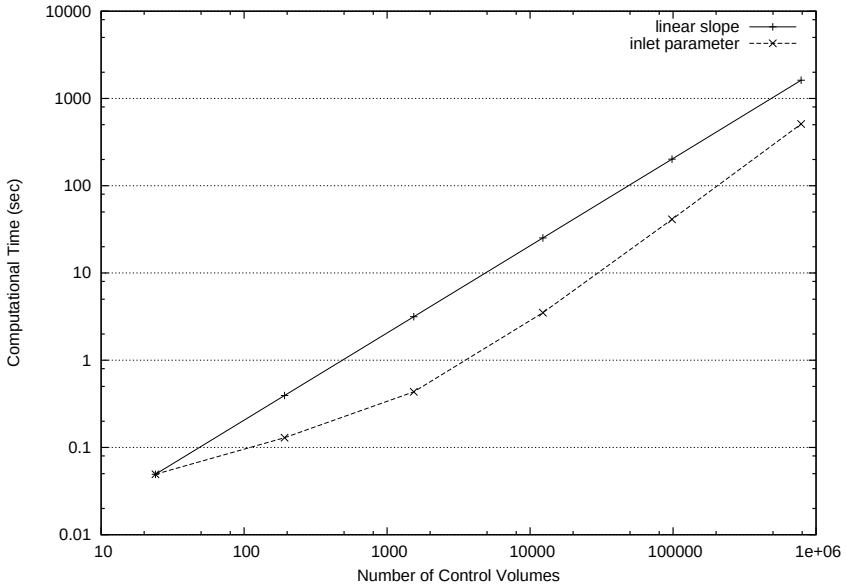

Figure 2: Computational time over the number of control volumes (inlet sensitivity)

7.2 Variable Shape Parameter

In the second test case the design parameter β moves the positions of the control points (CPs) of the NURBS surface, wherein the bump shape can be changed directly. This test case is well known from literature [6, 12]. 9 CPs are defined to modify the bump, see Figure 3. The four corner points are fixed to guarantee geometric continuity at the channel walls, whereas the three CPs at $x = 0.45\,\mathrm{m}$ can be translated in z-direction, i.e.,

$$\mathbf{P}_{i,2}\left(\beta\right) = \mathbf{P}_{i,2}^{(0)} + \beta \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad i = 1, \ldots, 3, \tag{28}$$

where $\mathbf{P}_{i,2}^{(0)}$ is the initial position of the i-th CP at $x = 0.45\,\mathrm{m}$. Thus the bump can be raised or lowered by varying β.

The V-cycles setup is the same as used for the inlet sensitivity test case, as well as the underrelaxation parameters and the convergence criterion. We examine as before the behavior of the number of fine grid iterations and the necessary time for the calculations as the main indicators for computational costs. The results can be found in Table 3 and Figure 4. The number of fine grid iterations remain almost constant on the finer grids in agreement with the multigrid theory. Also the (only) linear increase of the computing time with the number of CVs can be seen.

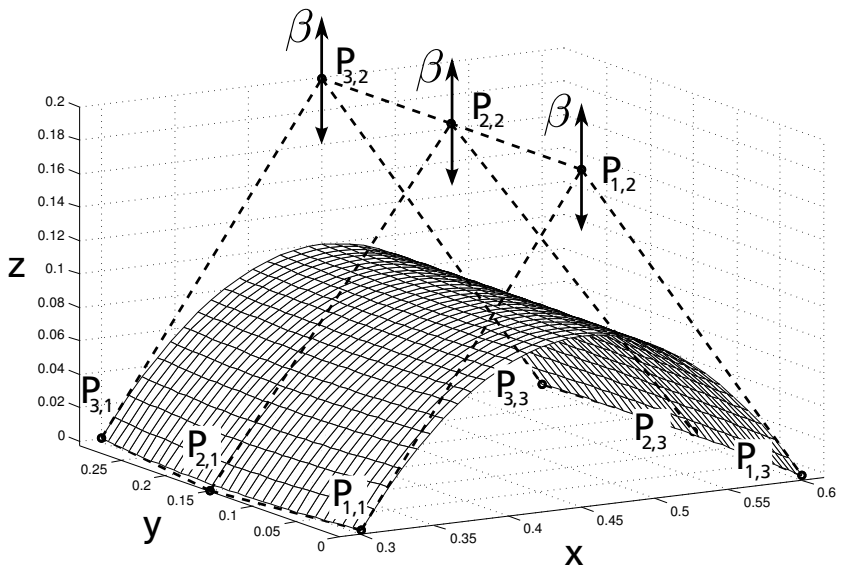

Figure 3: Parameter controlled NURBS surface on the 5th grid level

Scheme (# CVs)	SG1 (24)	MG2 (192)	MG3 (1536)	MG4 (12288)	MG5 (98304)	MG6 (786432)
Fine grid iterations	165	100	86	102	126	127
Time (sec)	0.05	0.14	0.48	3.82	43.95	369.15

Table 3: Multigrid behavior for controllable bump surface design

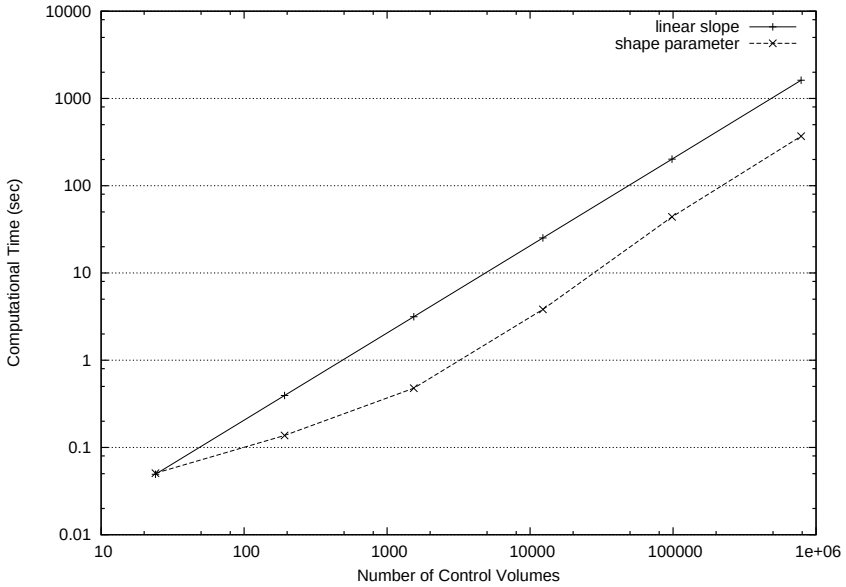

Figure 4: Computational time over the number of control volumes (shape sensitivity)

8 CONCLUSIONS

A geometric multigrid algorithm for solving the continuous sensitivity equations has been presented in this paper. We introduced a general gradient based optimization problem and derived the continous sensitivity equations from the Navier-Stokes equations. Furthermore, we presented a finite-volume discretization scheme and a geometric multigrid algorithm for solving the sensitivity equations.

The implemented multigrid method has been investigated for two different test cases, i.e., a channel flow over a bump with variable inlet velocity and a channel with a parameter controlled NURBS surface. For both examples the flow sensitivities were calculated with different spatial discretizations. Furthermore, time meassurements were realized, which showed very good agreement with multigrid theory. We have shown that the computational cost increases only linearly with the number of control volumes for decreasing grid spacing. This allows fast and robust flow sensitivity computations for arbitrary design parameters, which enables the employment of efficient gradient-based minimization algorithms in the context of flow control and flow optimization problems.

REFERENCES

[1] G. Becker, U. Falk, and M. Schäfer, Shape optimization with higher-order surfaces in consideration of fluid-structure interaction. *Fluid-Structure Interaction: Theory, Numerics and Applications*. Kassel University Press, Kassel, 2009.

[2] W. L. Briggs, V. E. Henson, and B. Kaspar, A multigrid tutorial. Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[3] F. Durst and M. Schäfer, A parallel block-structured multigrid method for the prediction of incompressible flows. *International Journal for Numerical Methods in Fluids*, Vol. 22, 549-565, 1996.

[4] R. Duvigneau and D. Pelletier. On accurate boundary conditions for a shape sensitivity equation method. *International Journal for Numerical Methods in Fluids*, Vol. 50, 147-164, 2006.

[5] J. H. Ferziger and M. Perić, Computational Methods for Fluid Dynamics. Springer Berlin, 2002.

[6] M. D. Gunzburger, Perspectives in flow control and optimization. Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[7] W. Hackbusch, Multi-grid methods and applications. Springer Berlin, 1985.

[8] H. Hu, Application of an automatic differentiation method to a 2D Navier-Stokes CFD code. *Computer Methods in Applied Mechanics and Engineering*, Vol. 156, 179-183, 1998.

[9] F. Ilinca, D. Pelletier, and A. Hay, First and second order sensitivity equation methods for value and shape parameters. *International Journal for Numerical Methods in Fluids*, Vol. 57, 1349-1370, 2008.

[10] E. Laporte and P. LeTallec, Numerical Methods in Sensitivity Analysis and Shape Optimization. Birkhäuser Verlag, Basel, 2002.

[11] M. Schäfer, Computational Engineering – Introduction to Numerical Methods. Springer Berlin, 2006.

[12] L.G. Stanley and D.L. Stewart. Design sensitivity analysis. Society for Industrial and Applied Mathematics Philadelphia, 2002.

[13] J. Siegmann, G. Becker, J. Michaelis, and M. Schäfer, A general sensitivity based shape optimization approach for arbitrary surfaces. In proceedings of the *8th World Congress on Structural and Multidisciplinary Optimization*, WCSMO, Lisbon, 2009.

[14] D. Thévenin and G. Janiga, Optimization and computational fluid dynamics. Springer Berlin, 2008.

[15] U. Trottenberg, C. W. Oosterlee, A Schüller, Multigrid. Elsevier Academic Press, San Diego, 2001.

[16] É. Turgeon, D. Pelletier, and J. Borggaard, A continuous sensitivity equation approach to optimal design in mixed convection. *Numerical Heat Transfer*, Part A: Applications, 38: 8, 869-885, 2000