No. 611 **August 2019**

A monolithic operator-adaptive Newton-Multigrid solver for Navier-Stokes Equations in 3D

R. Jendrny, O. Mierka, R. Münster, S. Turek

ISSN: 2190-1767

A monolithic operator-adaptive Newton-Multigrid solver for Navier-Stokes Equations in 3D

R. Jendrny*, O. Mierka, R. Münster, S. Turek

Institute of Applied Mathematics (LS III), TU Dortmund University, Vogelpothsweg 87, D-44227 Dortmund, Germany

Abstract

The aim of this paper is to describe a new, fast and robust solver for 3D flow problems which are described by the incompressible Navier-Stokes equations.
The correspondig simulations are done by a monolithic 3D flow solver, i.e. velocity and pressure are solved at the same time. During these simulations the convective part is linearized using two different methods: Fixpoint method and Newton method. The Fixpoint method is working in a quite robust way, but it has a slow convergence depending on the Reynolds number. In contrast, if the Newton method does not fail, the simulations which are done by this linearization converge typically much faster. In the case of the Newton method quadratical convergence is obtained. The challenging part is to find a method which unites the stability of the Fixpoint method and the fast convergence of the Newton method.
For the resulting operator-adaptive Newton method, several numerical examples are considered: The flow around a sphere and a cylinder is simulated to analyze the behaviour of the used methods. Since the behaviour of the linearization types is different between each of them, the results caused by varying Reynolds numbers and the arised equations are analyzed concerning the efficiency of each method.

Keywords: Navier-Stokes equations, Fixpoint method, Newton method, adaptive Newton, flow around a sphere, flow around a cylinder

1. Motivation

In this paper we focus on 3D flow problems, which can be described by the non-stationary Navier-Stokes equations:

$$\begin{aligned} u_t + u \cdot \nabla u + \nabla p &= \nu \Delta u && \text{in } \Omega, \\ \nabla \cdot u &= 0 && \text{in } \Omega, \end{aligned}$$

with prescribed boundary values and initial solutions.
Later in this paper we will study the flow around a cylinder and the flow around a sphere which is, for example, already studied in [5] in a detailed way. As motivation the important result of that paper is that there is a range in which the flow has a stationary behaviour: Until Reynolds number 270 the flow has a stationary structure. Therefore, in the first step we only focus on this stationary region. Then, our

*Corresponding author, robert.jendrny@math.tu-dortmund.de

model reduces to:

$$u \cdot \nabla u + \nabla p = \nu \Delta u,$$
$$\nabla \cdot u = 0.$$

Compared to the unsteady flow simulations until steady state in [5] it seems to be a more efficient way to solve this system in a coupled way, that means by a monolithic approach.
The most difficult part of solving these stationary equations is to realize a good handling of the non-linear part. For small Reynolds numbers the influence of this part has less influence. So, the treated Reynolds number influences the characteristic of the equations: If the Reynolds number is small we can neglect the convective part and the equations are just linear ones. But this region is only a small one and for a stable and robust 3D solver we cannot neglect the non-linear part.

2. Numerical approach

To solve the stationary NSE we first of all have to discretize the equations.

2.1. Discretization of the occuring equations

This is done by the finite element method using the LBB-stable FEM pair Q2-P1. For details we refer to [2]. For the stationary NSE it results in the well known system:

$$\begin{pmatrix} \tilde{S}(u_h) & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix} \text{ with } \tilde{S}(u_h) = L + K(u_h),$$

where L represents the diffusive part $-\nu\Delta u$ and $K(u_h)$ replaces the convective part $u \cdot \nabla u$.
For unsteady flow problems the non-stationary Navier-Stokes equations are considered: Then the momentum equation reads as $u_t + u \cdot \nabla u + \nabla p = \nu \Delta u$ and we additionaly get the mass matrix in the upper left part $\tilde{S}(u_h) = M + \theta \Delta t (L + K(u_h))$ with θ chosen due to the used time discretization.
To assemble the matrices L, $K(u_h)$ and B we can use two concepts: We can take a linear mapping between the physical and the reference element or a mapping using the isoparametric concept. The benefit of the isoparametric concept can be visualized as follows (For simplicity the 2D case is presented.).

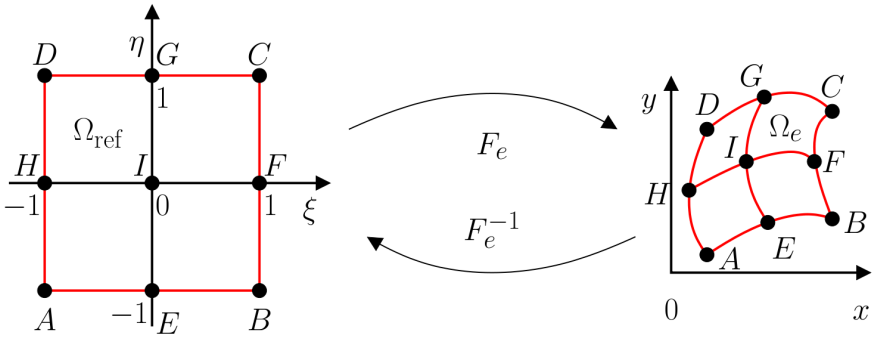

Figure 1: 2D Visualization of isoparametric mapping [7]

Since the domains of our numerical tests and applications do not have straight boundarys the advantage of the isoparametric concept should be significant since only then, the full convergence order can be observed.

2.2. Derivation of the used solver

Now we have a discrete system of our problem which ends up in a (non-linear) algebraic system. In this section we discuss how to solve this system [3]. Therefore we rewrite our equations:

- Continous problem:

$$\begin{aligned} -\nu\Delta u + u\cdot\nabla u + \nabla p &= f, \\ \nabla\cdot u &= 0, \end{aligned}$$

- discrete problem:

$$\begin{pmatrix} L+K(u_h) & B \\ B^T & 0 \end{pmatrix}\begin{pmatrix} u_h \\ p_h \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix},$$

$$\text{with } K(v)w \sim v\cdot\nabla w,$$

$$\text{i.e. } [K_{i,i}(u_h^{n-1})]_{m,n} = \int_\Omega [u_1^{n-1}\partial_x\varphi_m + u_2^{n-1}\partial_y\varphi_m + u_3^{n-1}\partial_z\varphi_m]\varphi_n dx$$

$$\text{and } K_{i,j} = 0, \text{ if } i \neq j.$$

In [5] we have used a an operator splitting based variant which is a good choice for non-stationary flows [9]. However, since we concentrate on stationary flows, we can use the gain of a monolithic solver.

2.2.1. Fixpoint method

We apply the Fixpoint method to the discrete problem and write it in a defect-correction procedure. For simplicity we take a full update, so we can neglect the step size ω.

$$\begin{bmatrix} u_h^n \\ p_h^n \end{bmatrix} = \begin{bmatrix} u_h^{n-1} \\ p_h^{n-1} \end{bmatrix} + \begin{bmatrix} L+K(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} g \\ 0 \end{bmatrix} - \begin{bmatrix} L+K(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u_h^{n-1} \\ p_h^{n-1} \end{bmatrix} \right).$$

The non-linear term, which is represented by the convective part, is defined as:

$$K := [K]_{m,n} := [K_{i,i}(u_h^{n-1})]_{m,n} \text{ and } K_{i,j} = 0, \text{ if } i \neq j.$$

However, we have further parts which we have to mention: On the one hand, we have an outer loop which iterates over n until the defect

$$\begin{bmatrix} g \\ 0 \end{bmatrix} - \begin{bmatrix} L+K(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u_h^{n-1} \\ p_h^{n-1} \end{bmatrix}$$

is small enough: This is the non-linear loop. On the other hand, we have the preconditioning step, i.e. apply $\begin{bmatrix} L+K(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix}^{-1}$ to the defect. Instead of inverting this matrix we solve these linear "defect" equations using a direct linear solver (or in most cases an iterative one) and update the solution $n-1$.

2.3. Newton method

To accelerate the convergence speed we also apply a Newton solver. From theory we know that we need a Fréchet derivative of our functional. For details we refer to [8]. In the undamped case, we get the following equations:

$$\begin{aligned} -\nu\Delta u^n + u^{n-1}\cdot\nabla u^n + u^n\cdot\nabla u^{n-1} + \nabla p^n &= f - u^{n-1}\cdot\nabla u^{n-1} \\ \nabla\cdot u^n &= 0. \end{aligned}$$

Again we write this in a defect-correction procedure and get:

$$\begin{bmatrix} u_h^n \\ p_h^n \end{bmatrix} = \begin{bmatrix} u_h^{n-1} \\ p_h^{n-1} \end{bmatrix} + \begin{bmatrix} L + R(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} g \\ 0 \end{bmatrix} - \begin{bmatrix} L + K(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u_h^{n-1} \\ p_h^{n-1} \end{bmatrix} \right)$$
with $R(u_h^{n-1}) = K(u_h^{n-1}) + \alpha\bar{M}(u_h^{n-1}), \alpha \in \{0, 1\}$.

Anew, we have neglected the step size ω.
To get a better understanding of the matrix R we summarize the assembly of this matrix:

- First we have to mention that the "convective" part is defined as before.
- The "reactive" part reads as

$$\bar{M}_{i,j}(u_h^{n-1}) = M(\partial_j u_i^{n-1}), \text{ where } M_{m,n}(v) = \int_\Omega v\varphi_m\varphi_n dx.$$

- Summarizing, the matrix R can be assembled by

$$R = \begin{pmatrix} K + \alpha\bar{M}_{1,1} & \alpha\bar{M}_{1,2} & \alpha\bar{M}_{1,3} \\ \alpha\bar{M}_{2,1} & K + \alpha\bar{M}_{2,2} & \alpha\bar{M}_{2,3} \\ \alpha\bar{M}_{3,1} & \alpha\bar{M}_{3,2} & K + \alpha\bar{M}_{1,3} \end{pmatrix}.$$

Remark 2.1. *(*Connection of Newton and Fixpoint method*)*

- *If the scalar α is set to 0 the Newton method and the Fixpoint method match.*
- *We have a "full"/pure Newton method if we take $\alpha = 1$.*

3. Convergence behaviour of the solver

In this section we want to analyze the derived solvers. Since we are, at this time, only interested in the convergence of the nonlinear solvers, the linear (sub-)problems are solved directly [12], i.e. we do not use an iterative scheme.
In the literature one can find several benchmark computations in which nonlinear flow problems are analyzed. To check the accuracy of our solver we start with focusing on the 2D flow around a cylinder benchmark [10]. The simulation setting can be sketched as follows:

- Domain: The simulation domain is given by a two dimensional channel and a cylindrical obstacle in the first part of the fluid domain (The detailed area size can be found in [10].).

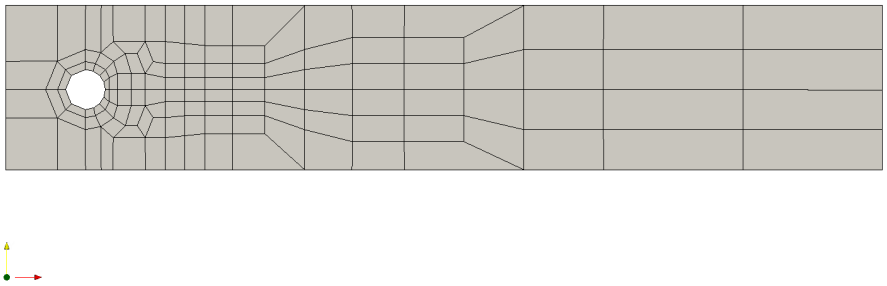

Figure 2: Fluid domain and coarse grid

- Boundary conditions: A parabolic inflow boundary condition with mean velocity 0.2 is prescribed. At the outflow part there is the "do nothing" boundary condition and at the other boundaries no slip conditions are prescribed. Together with the viscosity value 1E-3 the flow can be characterized as Re = 20,

Since our described approach has been designed to simulate 3D flow problems, we need to adjust the geometry: We **extrude** the 2D area in the third dimension to have a three dimensional object.
Then, the biggest difference between the Fixpoint and the Newton method is the convergence speed: For the Fixpoint method there should be linear convergence and the Newton method should converge quadratically. Doing a simulation of the benchmark configuration we get the defect behaviour as visualized in figure 3.

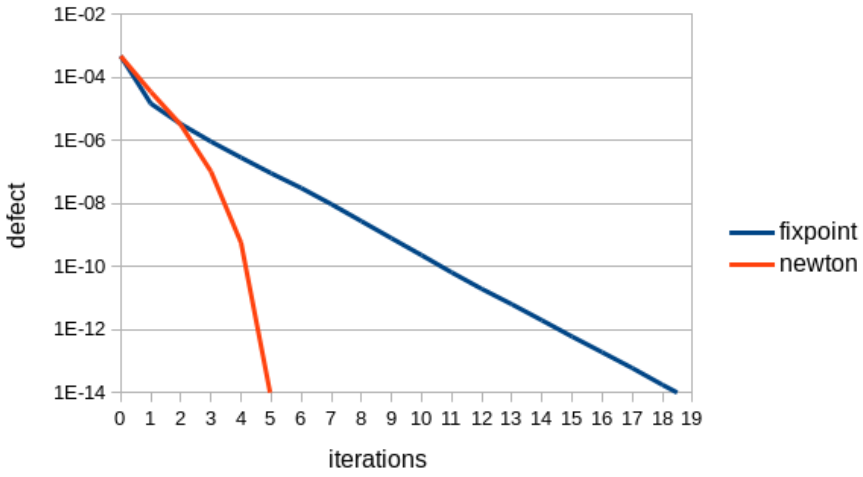

Figure 3: Validation of convergence speed

As one can see the Fixpoint method needs more iterations than the Newton method to achieve the same accuracy in the defect. In this case it needs four times more iterations than the Newton method.
The described benchmark configuration (see also the FEATFLOW[1] webpage for the details) has only a small range in which the flow has a stationary structure. This region has an upper bound at Reynolds number lower than 100. Several test simulations have shown that in this range the Fixpoint and the Netwon method

[1] `www.featflow.de`

converge with their expected convergence order. If we want to increase the nonlinear effect, meaning increasing the Reynolds number, we have to take another setting: One option is the flow around a sphere which has been analyzed in [5].
Taking the flow around a sphere problem, the range of stationary flow problems has an upper bound at Reynolds number 270. In the next test cases we are interested in the flow with Reynolds numbers 150 and 250. We know that for higher order of nonlinearity, meaning larger Reynolds numbers, the Fixpoint method needs a lot of nonlinear iterations. The advantage of the Newton method should stand out, if the starting solution for the Newton method is close enough to the final solution. Otherwise the Newton method might diverge.

Re	Starting	Fixpoint	Newton
150	initial zero	29	7
150	150 (level-1)	28	4
250	initial zero	53	div
250	250 (level-1)	45	div

Table 1: Flow around a sphere: High nonlinearity examples

As long as the Newton method converges, it converges much faster than the Fixpoint method. But if the nonlinearity increases the Newton method diverges, even if the starting solution should be accurate enough. Consequently, we have to improve the numerical behaviour of the described Newton solver.

4. Improvement of the solver

4.1. Operator-adaptive Newton method

We are interested in a nonlinear solver which unites the stability of the Fixpoint method and the fast convergence of the Newton method. We have seen that there can be situations in which the Newton method fails (see last table 1). To get a first idea we remind the formulation of the Newton method:

$$\begin{bmatrix} u_h^n \\ p_h^n \end{bmatrix} = \begin{bmatrix} u_h^{n-1} \\ p_h^{n-1} \end{bmatrix} + \begin{bmatrix} L + K(u_h^{n-1}) + \alpha \bar{M}(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} g \\ 0 \end{bmatrix} - \begin{bmatrix} L + K(u_h^{n-1}) & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u_h^{n-1} \\ p_h^{n-1} \end{bmatrix} \right).$$

We point out the parameter α which identifies if we use the Fixpoint method or the Newton method. The above example has shown that the Newton does not converge all the time whereas the Fixpoint method converges. In a next step we expand the upper test cases: We do not only solve the problem with each pure method; additionaly, we take a mixture of these methods, e.g. $\alpha = 0.5$, and get the results:

Re	Starting	Fixpoint ($\alpha = 0$)	mixture ($\alpha = 0.5$)	Newton ($\alpha = 1$)
250	initial zero	53	40	div
250	250 (level-1)	45	36	div

Table 2: Mixture of both methods

The results show that the choice of α seems to be important: The mixture converges, too, and it has a faster convergence than the Fixpoint method. If the pure Newton method and a large Reynolds number are combined there is no chance of convergence

because of the "bad starting" solution. Here, we can directly identify the influence of the non-linearity: Up to a certain Renolds number the Newton method converges very fast; after this limit the Newton method gets into trouble to attain convergence. The goal of the new method is obvious: There is a need for a new method that unites the benefits of each method and that switches between them or, even better, changes the influence of the methods in an adaptive way. This method will be based on [6]. Here, we need a criterion how to set α which can be coupled with the nonlinear defect:

- How to increase the value α?
  If the nonlinear defect decreases from iteration $n-1$ to n, the solution comes closer to the final one. This can be the chance that the solver can get a speed up by a bigger part of the Newton method. So, α can increase.

- Can α also decrease?
  In case of an increasing nonlinear defect, the solver may diverge with the current setting. To enhance the behaviour of the defect, α should go to 0 because of the stability of the Fixpoint method.

If the defect does not change in two following iterations the solver can end up in an infinite loop. To control this, α should also decrease in that case.
Thus, we need a function which updates the value α, i.e. $\alpha_{n+1} = \mathcal{F}(x)\ \alpha_n$. The argument x is obviously defined by the change of the last two defects: $x := \frac{||def_n||}{||def_{n-1}||}$. The definition of the function $\mathcal{F}$ is given in [6]: It is described as $\mathcal{F}(x) = a + \frac{b}{c+\exp(d\,x)}$.

Remark 4.1. *(*Update of α*)*
The value α controls the influence of the Newton method part. Therefore, α is bounded between 0 and 1: If the above update is used, the lower bound can only be crossed if the image of the function $\mathcal{F}$ can be negative. With the previous definition the lower bound cannot be crossed. But the upper bound can be crossed. Then we need to add the condition that ***if*** *$\alpha >$* ***1, then set*** *$\alpha :=$* ***1****.*

The function $\mathcal{F}$ has 4 free parameters which have to be defined. For example, we assume the following conditions:

- The maxmimal reduction of α should be bounded, here: $a = 0.2$.

- $\mathcal{F}$ should be "big" if x is small: $\mathcal{F}(0) = 3$.

- If there is no change in two subsequent defects, α should be decreased: $\mathcal{F}(1) = 0.9$.

- If $||def_n||$ is one half of the previous one, increase α by 50%: $\mathcal{F}(0.5) = 1.5$.

Remark 4.2. *(*Conditions to fit the function*)*
In the current code version the user can set the values $\mathcal{F}(0)$ and $\lim_{x\to\infty} \mathcal{F}(x)$ by its own. So the user is able to adapt this method to the given problem.

Corollary 4.3. *(Function $\mathcal{F}$)*
With the above assumptions the function $\mathcal{F}$ for the following tests is defined as:

$$\mathcal{F}(x) = 0.20 + \frac{1.43}{-0.48 + \exp(0.94x)}.$$

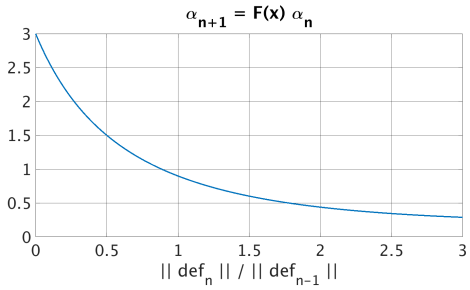

Figure 4: Function $\mathcal{F}$

4.2. Adaptive stopping criterion for the linear solver

The next topic that we have to focus on is the solving of the linear subproblems. In general we solve these problems using iterative schemes, i.e., the linear equations are solved until a certain tolerance is reached. Since the linear problems are not solved exactly, it can influence the convergence order of the nonlinear solver. This order is dependent on the accuracy of the linear solver: There is a necessary number of digits that have to be gained by this solver. If this solver does not gain enough digits, the convergence order can get lost. But there is also a limitation of the convergence order of the nonlinear solver: If the necessary accuracy is reached there is no need of further linear solver iterations, because the order of the nonlinear solver cannot be improved.
We know that the Fixpoint method has a convergence of order 1. So, if we have a reduction of the nonlinear defect of 0.1 we will only have a reduction of 0.1 in the next nonlinear iteration, although the linear problems are solved exactly. In this case we would work more than needed. It is enough to solve these linear problems with the same precision as the given order of the nonlinear solver. We can do the analogue argumentation for the Newton method: Here we have a convergence of order 2, i.e., if we have a reduction of 10^{-2} we can have a reduction of 10^{-4} in the next and 10^{-8} in the over-next nonlinear iteration because of the given order. This also holds if we solve the linear problems exactly. Again we would do more work than needed.
Therefore, we have implemented an "adaptive stopping criterion" for our linear solver which depends on the order of convergence. It can be written as:

1. Compute (or read) the nonlinear defect from last iteration: $||def_{n-1}||$.
2. Compute the non-linear defect from current iteration: $||def_n||$.
3. Calculate the digital criterion: $p = 2^\alpha$ and $asc = \left(\frac{||def_n||}{||def_{n-1}||}\right)^p$, where $\alpha = 0$ stands for the Fixpoint scheme and $\alpha = 1$ for the Newton one.
4. In the coming iteration the linear solver needs to gain asc digits.

It should be clear that this adaptive stopping criterion helps us to get the "optimal" convergence order of each nonlinear method. We underline this with the following testproblem: Again we simulate the (pseudo) 3D flow around a cylinder configuration [10]. On the one hand we use the Fixpoint method and on the other hand the Newton method. In both studies we take 4 different values which the linear solver should to gain: 1 digit, 2 digits, our new strategy and an exact linear solution. On the first axis you see the nonlinear iterations and on the second one the nonlinear defect.

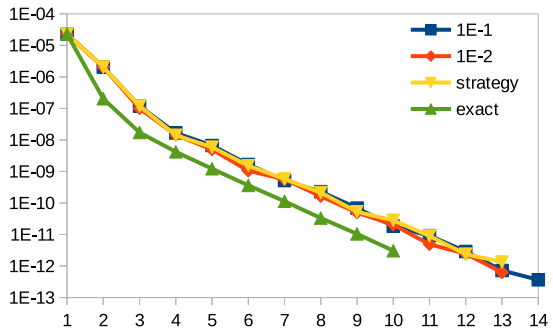

Figure 5: Fixpoint convergence

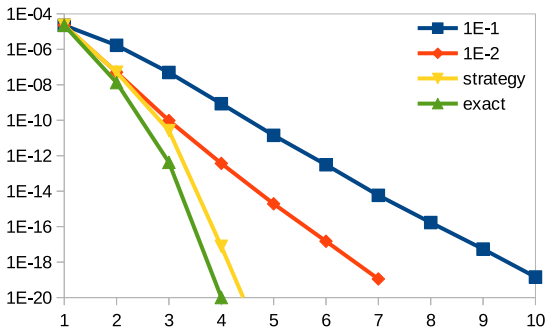

Figure 6: Newton convergence

Firstly, the convergence behaviour simulated by the exact solver confirms the general considerations from above, e.g. the Fixpoint method can only converge linear although the subproblems are solved exactly, but the exact solution has the "best" convergence speed. However, our new strategy is close to that and much better than the other ones. If we take a look at the nonlinear defect, our strategy needs approximately the same number of nonlinear iterations as the exact linear one. We stop the linear solver much earlier: The exact solver would always gain machine accuracy. So, the benefit of this strategy leaps out.

4.3. Combination of both aspects

Next, we sum up the two improvement topics and combine them into one approach. This can be described as follows.

- INPUT: α_0, asc
- COMPUTE initial defect def_0
- Nonlinear iteration ($n = 1, 2, ...$)
  1. SOLVE linear sub-problem gaining asc digits.
  2. COMPUTE def_n.
  3. READ def_{n-1}.
  4. SET $\alpha_{n+1} = \mathcal{F}(x)\alpha_n$
     - IF ($\alpha_{n+1} > 1$) THEN $\alpha_{n+1} = 1$
  5. SET $p = 2^{\alpha_{n+1}}$ and $asc = \left(\frac{||def_n||}{||def_{n-1}||}\right)^p$
     - IF ($asc > 10^{-1-\alpha}$) THEN $asc = 10^{-1-\alpha}$
  6. UPDATE $n \to n + 1$
  7. CONVERGENCE? or GO BACK TO 1.

Remark 4.4.

- *To guarantee convergence the asc-value should not increase to 1 in step* 5. *Usually, we take 0.1 for the Fixpoint and 0.2 for the Newton method as minimal reduction of the linear solver.*

- *The adaptive method should be bounded by the pure Fixpoint and Newton method, so α has an upper bound in step 4.*

Using the (pseudo) 3D flow around cylinder problem we want to demonstrate how this algorithm is working.
In the following, we compare two initial values for α: $\alpha = 0.1$ and $\alpha = 0.5$. We set the *asc* to $10^{-1-\alpha}$ so that the following convergence behaviour results.

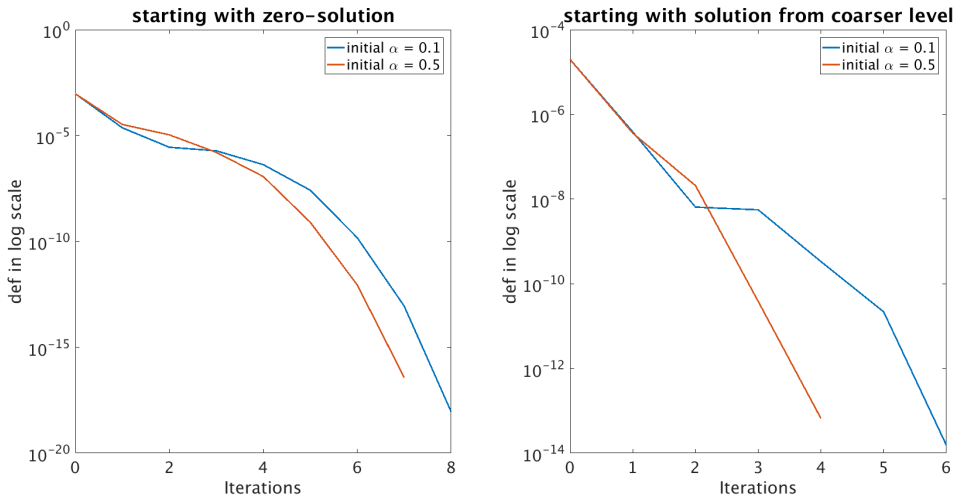

Figure 7: Adaptive convergence

On the left hand side we see the simulation results for Reynolds number 20 on level 2; on the right one there are the results on level 3 starting with the final results from level 2. In both cases we see that a "big" initial α-value accelerates the total convergence. This should be clear since the pure Newton method is also working fine. But it is also true that a small value α can be good for the first one or two iterations since the current solution is "far" away from the exact solution which can result in "convergence problems".
Now we want to compare our adaptive startegy with the two "pure" methods. The representation is the same as above: On the left you can find the level 2 results, on the right the results of level 3.

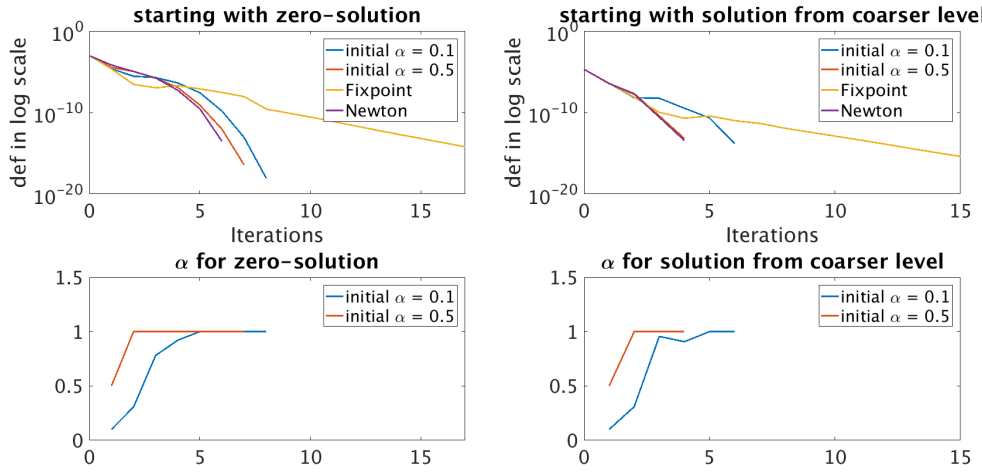

Figure 8: Comparison to Fixpoint and Newton

It seems to be clear that the adaptive method with an initial value 0.5 has the same convergence speed as the pure Newton method. But this method should be more stable, because it still uses some benefits from the Fixpoint method. We can also see that the α series is nearly monotone increasing.

4.4. Application: Flow around a sphere

4.4.1. Solver settings

Coming back to the motivation of this paper we want to see if our code is able to simulate the flow around a sphere that can be subdivided into different ranges: The range of axis-symmetric flows ends at Reynolds number approximated 210; until Reynolds number 270 there is a so called planar-symmetric flow which is a stationary one, too [5].
Before we start to simulate these flows and validate our approach, we want to summarize the solver configuration.

- **Nonlinear solvers:**
  - Three different linearization schemes are used: The pure Fixpoint, the pure Newton and the operator-adaptive Newton method.
  - To get suitable simulation results all simulations are started using the one level coarser grid solution (except for Level 2 in which we start from scratch).
  - Since we only focus on stationary flows we take an absolute stopping criterion, i.e. the absolute defect should be smaller than 1E-12.
  - We limit the number of nonlinear iterations at 100.
- **Linear solver - MG:**
  - The linear problems are solved using the F-cycle and our new adaptive stopping criterion.
  - On each discretization level we prescribe 20 smoothing (pre- and postsmoothing) steps that are done by a damped Vanka method with damping factor 0.2.
  - The number of linear solving steps is bounded by 10 MG iterations.

4.4.2. Numerical validation and comparison of the methods

We start with a relative small Reynolds number and perform the test configuration and the boundary conditions from [5]:

Figure 9: Extract of the simulation area

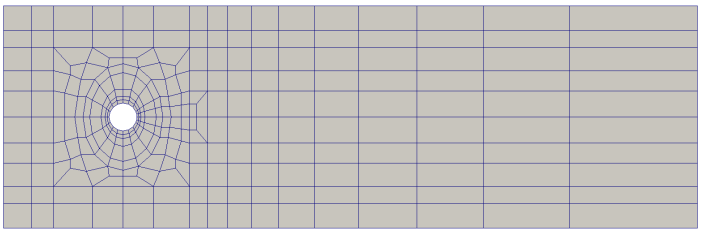

Figure 10: Coarse mesh cutplane

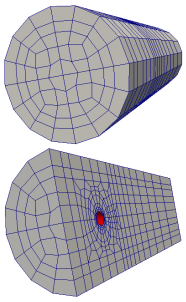

Figure 11: 3D sketch

We have a constant inflow at the inflow part and Dirichlet data at the lateral surface of the pipe in flow direction of value 1. We set the viscosity to 0.1, since we want to calculate the flow for a small Reynolds number, here Re = 10. As a reference we take the results that arise from a very fine discretized simulation done by our 2.5D axis-symmetric code. There we have got a drag value of 4.52982 and a zero lift by definition.

lvl	NEL	NEQ	**linear mapping** drag	F_y	F_z	e_{drag}
2	14176	408994	4.49788	1.50E-6	3.11E-12	0.71%
3	113408	3222974	4.52148	3.90E-7	1.51E-12	0.18%
4	907264	25.5E6	4.52769	6.13E-8	1.53E-11	0.05%
lvl	NEL	NEQ	**isoparametric mapping** drag	F_y	F_z	e_{drag}
2	14176	408994	4.53041	1.44E-6	3.85E-10	0.01%
3	113408	3222974	4.52985	3.92E-7	3.95E-12	0.00%
4	907264	25.5E6	4.52981	6.17E-8	4.03E-11	0.00%

Table 3: Level convergence

We discern level convergence and, of course, the version using a isoparametric mapping gives more precise results.
Because we solve the flow reagrding a small Reynolds number, the benefit of the Newton solver should not be significant. Smaller Reynolds number means that the (nonlinear) convective part in the Navier Stokes equations should have less influence. To underline this we tabulate the nonlinear iterations (NL) and also the required total multigrid steps (TOT MG). (Remark: The solver iterations of both mapping strategies match.)

lvl	**Fixpoint** NL	TOT MG	**Newton** NL	TOT MG
2	12	18	5	15
3	13	20	3	14
4	14	20	3	14

Table 4: Fixpoint vs. Newton

The nonlinearity is not high enough to see the advantage of the Newton solver compared to the Fixpoint method, because the linear (sub-)problems are not solved

exactly - we still use our new strategy. If these problems would be solved exactly we would see that we need only a few nonlinear iterations since the nonlinearity gets weaker and the Newton method would almost become an exact solver. Nevertheless, the Newton configuration still solves the problem faster (66% of MG-steps compared to Fixpoint); but, if we consider the numerical assembly of this solver, the Newton method is not advantageous for small Reynolds numbers.
Therefore, we increase the Reynolds number from 10 to 100, i.e., the viscosity has to decrease by a factor of 10. Now, the nonlinearity should get a bigger influence. (Again the reference values are taken from our 2.5D axis-symmetric code: Drag = 1.112266, lift = 0.)

lvl	NEL	NEQ	drag	lift	f_z	e_{drag}
2	14176	408.994	1.11466	1.46E-4	1.02E-14	0.22%
3	113408	3.222.974	1.11361	2.19E-5	1.11E-14	0.12%
4	907264	25.592.182	1.11275	2.88E-6	1.09E-11	0.04%

Table 5: FAS (Re 100): Level convergence for isoparametric mapping

The level convergence is obvious.

lvl	**Fixpoint**		**adaptive** 0.1		**adaptive** 0.5		**Newton**	
	NL	TOT MG	NL	TOT MG	NL	TOT MG	NL	TOT MG
2	25	94	7	43	7	40	8	46
3	16	31	6	22	4	20	3	18
4	14	25	6	21	4	19	3	16

Table 6: FAS (Re 100): Fixpoint vs. Newton

Since we have increased the Reynolds number the (nonlinear) convective part is the more dominant one. For the case that we start with a zero solution (level 2) the adaptive test cases are more competetive: Here we have the benefit of the good convergence at the beginning of the Fixpoint part and there is the possibility to increase the Newton part to get better convergence in each iteration. The Fixpoint method can only convince in the first iteration; the Newton method has its advantage if the solution is more and more close to the exact one. Since the simulations on level 3 and 4 start with the one level coarser solutions, the start value for the actual simulation is close enough to the current one and the Newton method reaches the fastest convergence.
In our simulations we could also see that we get a more stable level independent convergence if we use the Newton method instead of the Fixpoint method. This stable convergence is underlined by the number of total multigrid steps that are nearly all the same for each problem. Using the adaptive method there is nearly the same stable convergence, but the convergence speed decreases because there are still some left-overs of the Fixpoint method.

4.4.3. Multigrid convergence behaviour
In the last section we have seen that there are some benefits regarding convergence speed, if we use the (adaptive) Newton method. In this section we want to take a closer look at the multigrid convergence behaviour. Therefore, we concentrate on the last test case: Flow around a sphere with Reynolds number 100 and space discretization level 3.

The first topic which we want to analyze is the decrease of the defect for all 4 methods.

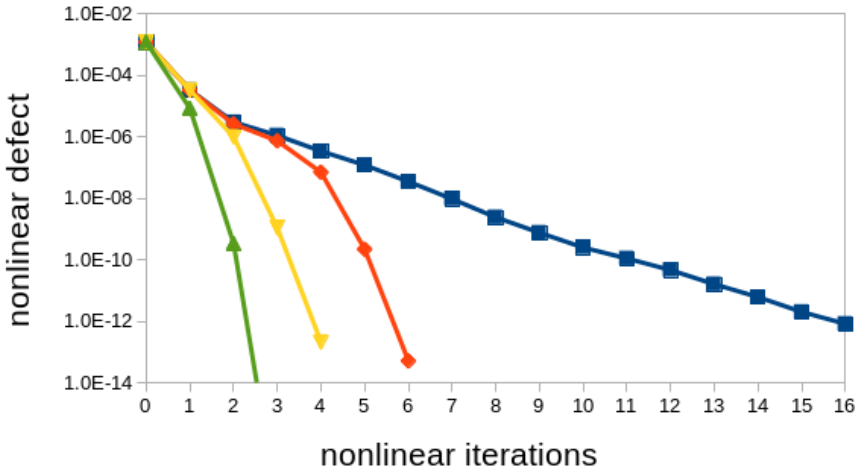

Figure 12: Defect: Fixpoint (blue), adaptive 0.1 (red), adaptive 0.5 (yellow) and Newton (green)

With this analysis we cannot see how the multigrid solver works but we can identify the convergence speed: As we have seen before, the Fixpoint method has a linear and the Newton method a quadratical convergence speed. The two adaptive methods start with linear convergence and end up close to quadratical one.
Since we have implemented a new adaptive stopping criterion we also want to check how this criterion has influenced the global defect convergence.

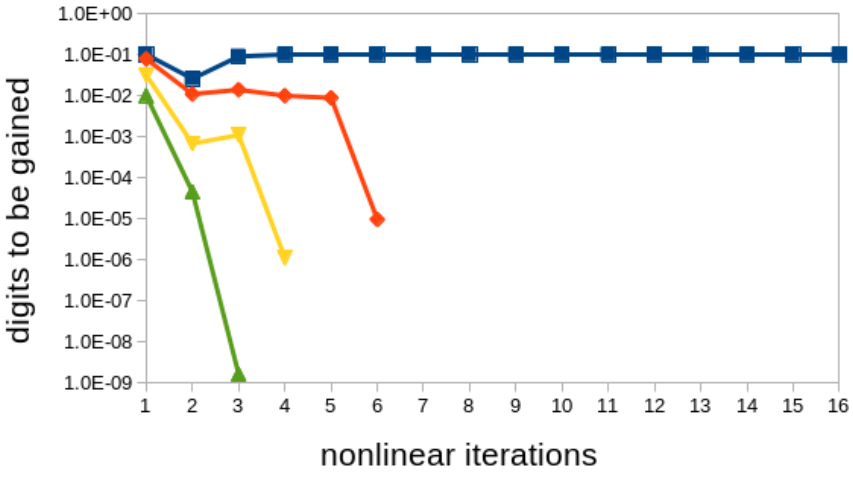

Figure 13: Adaptive stopping criterion: Fixpoint (blue), adaptive 0.1 (red), adaptive 0.5 (yellow) and Newton (green)

If we ignore the Fixpoint method, in which the adaptive stopping criterion is constant by construction, we clearly obtain that a smaller criterion results in faster convergence (see Fig. 13). It is also obvious that a stopping criterion that is less than 10^{-2} may result in quadratical convergence. These stopping criterions, which are connected to the digits gained by the multigrid solver, have effects to the used multigrid steps in each nonlinear iteration.

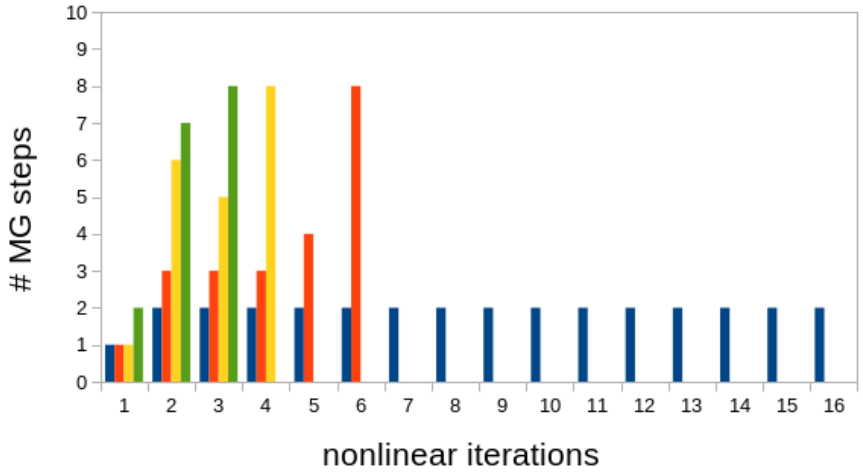

Figure 14: MG steps: Fixpoint (blue), adaptive 0.1 (red), adaptive 0.5 (yellow) and Newton (green)

Because one multigrid step can only reach a certain accuracy, the number of multigrid steps increases if the stopping criterion is relative small. For the Newton and the adaptive methods there is a need of 8 multigrid steps in the final stage of the simulation to get the desired accuracy. Compared to that a constant stopping criterion ensues in constant multigrid steps.
To conclude this multigrid analysis we receive the following chain:

small stopping criterion $\Rightarrow$ more multigrid steps $\Rightarrow$ less non-linear ierations.

4.4.4. Higher order of nonlinearity
If the nonlinearity increases, for example by increasing the Reynolds number, a robust and stable (linear) solver is needed. At the moment we are only interested to see if the adaptive Newton method can improve the Fixpoint method or if it can help the pure Newton method to converge. Therefore, we only concentrate on the resulting nonlinear ierations. Hence, we will ignore the multigrid solver and use the direct solver: *MUMPS* [12], which is a parallel solver based on UMFPACK [4],[11]. But because of the parallelism we are not memory limited by one CPU. MUMPS (**MU**ltifrontal **M**assively **P**arallel sparse direct **S**olver) is built on a LU decomposition acting in the multifrontal method, which is a version of Gaussian elimination for large sparse systems of equations, especially those arising from FEM.

Numerical tests.
We start with the simulation of the flow around a sphere on discretization level 1 (NEL: 1772, NEQ: 59048). Our goal is to find a test configuration which cannot be solved by the Newton method directly. Consequently, we will vary our current simulation parameters and we will also vary the starting solution. Here, we only focus on the number of nonlinear iterations.

Re	Starting	Fixpoint	Newton	adaptive		
				0.1	0.5	1
140	Re 100	28	6	x	x	x
150	Re 140	26	7	x	x	x
200	Re 150	29	7	x	x	x
200	Re 100	29	7	x	x	x
200	initial zero	32	div	10	10	div

Table 7: NL with exact linear solver (level 1)

If a simulation result from smaller Reynolds number is used as an initial solution, the Newton method is working perfectly. So, we are not interested in the adaptive methods (marked by "x"). Only if the simulation is starting from a zero solution, the Newton method is not working and we need the adaptive strategy. With this method, irrespective of the initial starting value α, we reduce the nonlinear iterations by a factor of 3. If we start the adaptive Newton method with $\alpha = 1$ there will be no convergence because the error in the first iteration cannot be repaired (decreasing α) in the coming iterations. Surprisingly, it does not matter which old solution has been used.
As we have seen, level 1 simulations are not interesting enough, because we can only start from a zero solution or from a solution of the same discretization level. We have more choices of starting solutions if we simulate on finer grids. Here, we have an increased number of unknowns and consequently there is a need of more memory. The goal of our simulation is still the same: Can we find further configurations that cannot be simulated by the Newton method directly? For completenes we also add some results from level 1 simulations.

		Re 200				Re 250			
Starting	Level	Fixpoint	Newton	adaptive		Fixpoint	Newton	adaptive	
				0.1	0.5			0.1	0.5
0		32	div	10	10	60	div	9	8
Stokes	L1	31	7	9	7	55	7	9	7
Re 100	L1	29	7	9	7	53	6	10	6

Table 8: Nonlinear iterations (level 1)

The more interesting part is presented by the following tables for the referred levels 2 and 3.

		Re 200				Re 250			
Starting	Level	Fixpoint	Newton	adaptive		Fixpoint	Newton	adaptive	
				0.1	0.5			0.1	0.5
0		28	div	10	8	53	div	11	9
Stokes	L2	26	6	9	6	42	7	9	7
Re 100	L2	25	5	8	5	40	6	8	6
Stokes	L1	28	div	9	6	48	div	11	7
Re 100	L1	27	div	9	6	46	div	9	7
Re 200/250	L1	27	div	9	6	45	div	9	7

Table 9: Nonlinear iterations (level 2)

		Re 200				Re 250			
Starting	Level	Fixpoint	Newton	adaptive		Fixpoint	Newton	adaptive	
				0.1	0.5			0.1	0.5
0		30	div	9	7	52	div	8	7
Stokes	L3	29	6	8	6	50	7	11	7
Re 100	L3	29	5	8	5	49	5	11	5
Stokes	L2	30	div	9	6	51	div	11	7
Re 100	L2	30	div	9	5	50	div	11	7
Re 200/250	L2	29	5	8	5	49	6	10	6

Table 10: Nonlinear iterations (level 3)

Evidently, it seems to be better to start from a solution at the same discretization level. Even the one level coarser solution for the same Reynolds number does not allow the Newton method to converge. It does not matter which coarser solution is prolongated and set as a starting solution. But if the discretization in space becomes really fine, the Newton method can also converge using a prolongated coarser starting solution, because the differences between the occuring solutions "vanish" for finer and finer discretization levels.

Corollary 4.5. *(*ADAPTIVE NEWTON*)*

- *If the Newton method can solve a certain simulation setting, it typically converges faster than the other presented solvers.*
- *If the Newton method fails, the adaptive methods can handle these simulation.*
- *The adaptive Newton method can be taken as an acceleration of the Fixpoint method.*
- *A good starting solution for the Newton method is given by the results that are computed by the Stokes equations, because possible steep gradients, which arise in the Newton method, will be smoothed.*

To guarantee convergence which can be accelerated by the adaptive Newton method it is useful to start with a relative small value α.

5. Stabilization by artificial diffusion

In the last section we have seen that the code can only converge for high Reynolds numbers if a direct solver and the Fixpoint method are used. To get results that are computed in a quick way it is also possible to use the adaptive Newton method; but also in this case there is a need of a direct solver. At the moment the code can solve problems with the iterative linear solver until Reynolds number 120. But as we have seen in [5] there are stationary flow structures until Reynolds number 250. The goal of the following sections is to find a scheme which allows solving problems for really high Reynolds numbers by our multigrid solver. One approach can be the usage of artificial diffusion.

5.1. General idea

Before we describe a working approach based on artificial diffusion stabilization, we mention an observation: During the solving process of our iterative multigrid solver the local Reynolds numbers increase on coarser meshes. The first idea is that we

use a stronger damping of the multigrid smoother which is based on a Vanka-like approach. This typically leads to an increasement of solver iterations.
Another idea is to add more diffusion on the coarser meshes on which the local Reynolds numbers increase.
Instead of adding diffusion only on the coarser meshes we could add artificial diffusion to all mesh levels. The benefit is obvious: There should be no problem in the transfer from level to level.

5.2. Numerical derivation of a working formula

The goal now is to find a setting in which the added (artificial) diffusion is as small as possible to get convergence and accurate results. The considered equations will look as follows:

$$\begin{pmatrix} \tilde{S}(u_h) & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix} \text{ with } \tilde{S}(u_h) = \tilde{L} + K(u_h), \tilde{L} = L(\mu) \text{ and } \mu = \nu + C \cdot h^\gamma.$$

The idea is based on the approach that we add a mesh dependent artificial diffusion in each element of our finite element approach.
To get a working formula we have tried different parameters for C and γ, i.e. $C \in \{0.1, 0.05\}$ and $\gamma \in \{1, 2, 3\}$. As first test cases, if this idea can work, we have analyzed the flow around a sphere problem with Reynolds number 200 on mesh level 2. We set $\gamma = 2$ and $C = 0.1$.

	FP	Adap 0.1	Newton
nonlinear iter	63	9	6
linear iter	411	80	60
∅ linear iter	6.524	8.889	10
f_x (drag)	7.939E-01	7.939E-01	7.939E-01
f_y (lift)	-3.290E-06	-3.289E-06	-3.289E-06
$\lvert f_z \rvert$	2.719E-11	5.036E-16	1.705E-15

Table 11: Differences between nonlinear solvers

We see that all 3 solver combinations converge to the same solution as expected. If we take a closer look at the table we observe that, again, the Newton method needs less linear and nonlinear solver iterations than the other ones.

5.2.1. Determination of parameter C

One important part of the artificial diffusion approach is the parameter C. We compare the influence of this parameter by taking different values for γ. Again we perform simulations for the flow around a sphere with Renolds number 200, but this time we consider discretization level 3. For $C = 0.1$ we get the following results.

γ	method	nonlin	lin	f_x (drag)	f_y (lift)	$\lvert f_z \rvert$
1	FP	23	54	8.513E-1	6.221E-5	-.954E-11
	Adap 0.1	7	29	8.513E-1	6.221E-5	3.154E-14
2	FP	53	196	7.929E-1	2.597E-6	4.874E-12
	Adap 0.1	7	41	7.929E-1	2.598E-6	3.181E-16
3	FP	62	233	7.882E-1	-3.147E-6	5.050E-11
	Adap 0.1	8	70	7.882E-1	-3.146E-6	1.926E-15
reference values				7.864E-1	≈ 0	≈ 0

Table 12: Parameter $C = 0.1$

If we halve the parameter the results look as.

γ	method	nonlin	lin	f_x (drag)	f_y (lift)	$\lvert f_z \rvert$
1	FP	29	76	8.230E-1	4.143E-5	2.959E-11
	Adap 0.1	7	29	8.230E-1	4.143E-5	7.333E-16
2	FP	57	203	7.904E-1	-1.305E-7	6.238E-12
	Adap 0.1	8	53	7.904E-1	-1.305E-7	1.475E-15
3	FP	62	265	7.881E-1	-3.218E-6	2.796E-11
	Adap 0.1	9	82	7.881E-1	-3.218E-6	1.634E-15
reference values				7.864E-1	≈ 0	≈ 0

Table 13: Parameter $C = 0.05$

There are two observations for the smaller choice of C:

- The values for the acting forces are closer to the reference ones [5]. This is clear since the influence of the added artificial diffusion is not as significant as before.
- The solver iterations, both nonlinear and linear ones, increase, because the nonlinearity increases with less artificial diffusion.

If we compare the results for the two different parameters we conclude that a choice of parameter $C = 0.1$ should be working fine since there is a chance to get accurate results if we adapt the parameter γ. This choice can also be underlined by the solver iterations which are smaller than for $C = 0.05$.

5.2.2. Determination of γ and damping parameter ω for smoothing
In this part we fix the parameter C to 0.1. The aim is now to find a suitable combination of γ and the damping parameter of the smoother: We use the two usual damping parameters 0.5 and 0.7. Since we know that there are convergence problems because of the nonlinearity we also test the damping 0.1. For γ we take 1, 2 and 3. The reference drag value for the flow around a sphere with Reynolds number 200 is given by 7.8710E-1 computed by our code with the exact linear solver. First we start with simulations on discretization level 2.

ω	0.1			0.5			0.7		
α	nonlin	lin	drag	nonlin	lin	drag	nonlin	lin	drag
1	7	30	9.0338E-1	7	21	9.0338E-1	6	16	9.0338E-1
2	8	52	8.0781E-1	div	div	div	div	div	div
3	div	div	div	div	div	div	div	div	div

Table 14: Level 2

There are a lot of combinations that do not work, i.e. these configurations lead to divergence. The best choice seems to be $\omega = 0.1$ and $\gamma = 2$, because there convergence is obtained and the drag value is the closest one compared to the reference. But there are also bad news: Small damping parameters and a larger value γ force more solver iterations. With the same test configurations we simulate the problem on discretization level 3.

ω	0.1			0.5			0.7		
α	nonlin	lin	drag	nonlin	lin	drag	nonlin	lin	drag
1	7	29	8.5129E-1	7	21	8.5129E-1	7	20	8.5129E-1
2	7	41	7.9286E-1	6	20	7.9286E-1	6	21	7.9286E-1
3	8	70	7.8823E-1	div	div	div	div	div	div

Table 15: Level 3

On level 3 there are more configurations which lead to convergence. Only a barely damping in combination with $\gamma = 3$ leads to divergence. For level 3 the best choice regarding efficiency of the solver - meaning only a few solver iterations - and accurate results is given by $\omega = 0.5$ and $\gamma = 2$.
Lastly, we also take a look at the results for level 4.

ω	0.1			0.5			0.7		
α	nonlin	lin	drag	nonlin	lin	drag	nonlin	lin	drag
1	7	31	8.1566E-1	7	20	8.1566E-1	7	19	8.1566E-1
2	7	30	7.8894E-1	7	15	7.8894E-1	7	20	7.8894E-1
3	7	30	7.8787E-1	7	15	7.8787E-1	7	20	7.8787E-1

Table 16: Level 4

On level 4 we see that there are no differences in the solver statistics for each damping parameter. Again $\omega = 0.5$ gives the combination with the lowest expenditure of the solver.

Corollary 5.1. *(Coice of C, α and ω)*
An operating combination for the artificial diffusion parameters is given by $C = 0.1$, $\gamma = 2$ and $\omega = 0.5$.

For a stable configuration, if the upper choice is not working, the damping parameter can be reduced, e.g. $\omega = 0.1$.

5.3. Application: Time dependent 3D benchmark

We finish this section by giving results of the corresponding time dependent 3D benchmark simulations. The configuration is described in [1]. Here we only give a sketch of it:

- Mesh and domain: Flow around a cylinder 3D configuration, discretization level 2,

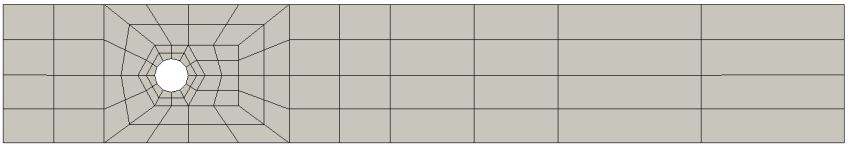

Figure 15: 2D cutplane of the coarse mesh

- Inflow boundary condition: Double-parabolic inflow, $u(y,z,t) = \{\frac{9}{4} \cdot 0.205^{-4}\} \cdot \sin(\frac{\pi}{8}t)y(0.41-y)z(0.41-z)$, $v = w = 0$,
- Physical parameter: $\nu = 10^{-3}$, i.e. Re = 100 at t = 4,

- Simulation time: $t \in [0, 8]$.

Of course, we need an extension of the previous approach which can solve time dependent problems. The initial version is able to solve such problems using the Crank-Nicolson method with all the above described techniques. So we have a strong solver combination that can be written in a defect correction procedure:

$$\begin{bmatrix} u_h^{n,k} \\ p_h^{n,k} \end{bmatrix} = \begin{bmatrix} u_h^{n,k-1} \\ p_h^{n,k-1} \end{bmatrix} + \begin{bmatrix} M + \theta\Delta t\{L + K(u_h^{n,k-1}) + \alpha\bar{M}(u_h^{n,k-1})\} & \Delta t B \\ B^T & 0 \end{bmatrix}^{-1}$$
$$\left(\begin{bmatrix} Mu_h^{n-1} - (1-\theta)\Delta t\{L + K(u_h^{n-1})\} \\ 0 \end{bmatrix} - \begin{bmatrix} M + \theta\Delta t\{L + K(u_h^{n,k-1})\} & \Delta t B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u_h^{n,k-1} \\ p_h^{n,k-1} \end{bmatrix} \right),$$

where n denotes the current time and k the iteration of the nonlinear solver.
The computed drag and lift values which occur at the cylinder can be visualized over time: In the top there is the drag value and in the bottom you can see the lift value. Additionally to the global time interval we have picked up some regions in which we have zoomed in: For the drag there is a zoom around the middle of the time interval in which the drag value has its maximum; for the lift there is a zoom to the middle region, too. But there is also a zoom in the more interesting region at the beginning of the time intervall where a "saddle point" of the curve can be found.
We compare the results that are stabilized by our artificial diffusion and the results without any stabilization with the reference solution given in [1].

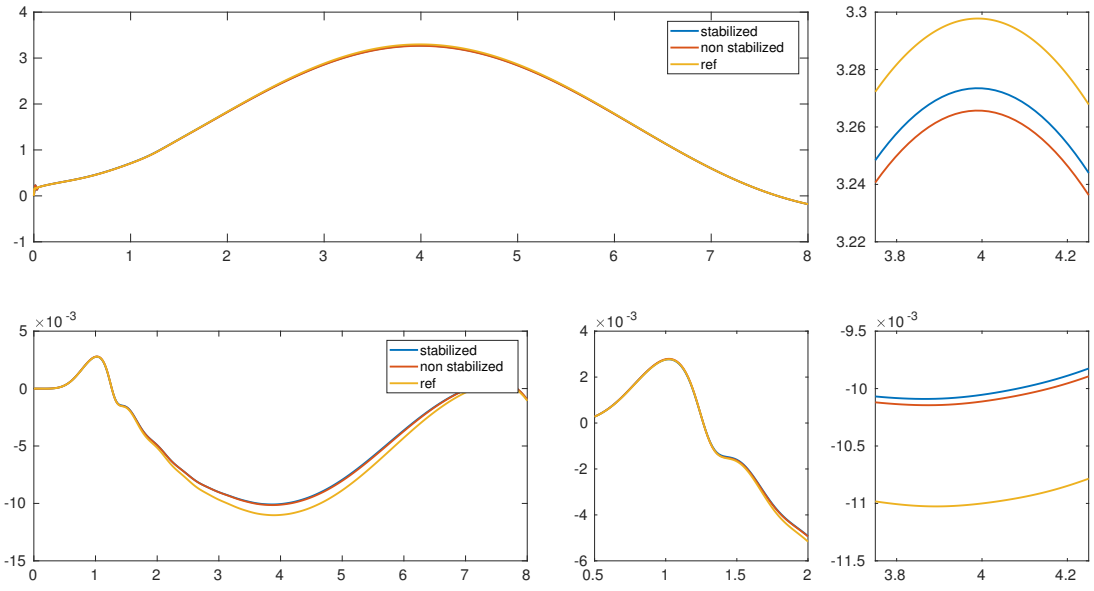

Figure 16: 3D Benchmark: Level 2

We observe that the drag value of the stabilized version is closer to the reference one. This should be intuitively correct, because a lower Reynolds number leads to larger drag values. An analogue observation is seen in the case of the lift value: Here the stabilized version is closer to the zero-lift value, since simulations with small Reynolds numbers lead into axis-symmetric flow structures that have a zero

lift. The interesting part is the zoom around T = 1.5: In this region there is nearly no difference between the considered configurations. For finer meshes we have similar results except for that we observe level convergence which means that all three configurations become closer to each other. The level convergence can be underlined by the Benchmark values.

Level 2	max drag	max lift	min lift	∅	T=1	T=4	T=6
normal	3.2657	0.002794	-0.010145	14.2|3.7	14|3	13|5	15|4
stabilized	3.2735	0.002773	-0.010090	13.9|3.8	13|3	11|4	15|4
Level 3	max drag	max lift	min lift	∅	T=1	T=4	T=6
normal	3.2944	0.002776	-0.010960	15.1|3.7	14|3	15|4	15|4
stabilized	3.2963	0.002771	-0.010946	15.0|3.7	13|3	14|4	15|4
Level 4	max drag	max lift	min lift	∅	T=1	T=4	T=6
normal	3.2977	0.002774	-0.011016	10.9|3.7	10|3	12|5	12|4
stabilized	3.2982	0.002773	-0.011013	10.8|3.7	10|3	11|5	11|4
reference	3.2978	0.002775	-0.010999				

Table 17: Benchmark values

The level convergence to the given reference values of [1] leaps out. The unstabilized version has a faster convergence, since the problem can be solved in an unstabilzed configuration and the stabilization error is standing out.
We have also tabulated the used linear and nonlinear solver iterations: In ∅ and T $\in \{1, 4, 6\}$ there are the total linear iterations on the left and the nonlinear ones on the right side. And again we can see that for finer meshes there is a need of less iterations for both solvers, because we have started the simulations with the solution of the one level coarser mesh and the difference of both solutions becomes smaller for finer meshes.

6. Conclusions and Outlook

In this work we have improved our monolithic three dimensional flow solver to have more accurate results. We have used the isoparametric concept inasmuch as we have worked with triquadratic functions and curved boundaries. From theory we know that the Fixpoint method does not have a fast convergence speed, contrary the Newton method can converge quadratically in a region around the exact solution. That is the reason why we have implemented the Newton solver: We are interested in fast calculated results needing less multigrid or nonlinear iterations. In our simulations we have seen that we can achieve the reference results and if we use the Newton method we do not need a lot of nonlinear iterations and we economize multigrid iterations. Since the convergence radius of the Newton method is local we can guarantee global convergence with our adaptive strategy. This strategy can also be seen as an acceleration of the Fixpoint method.
Finally, we are able to solve (stationary) flow problems in three dimensions described by nonlinear equations in an efficient way.

We are able to switch between Fixpoint and Newton method. But we still use a full update. It can be interesting if we also implement a line search technique to guarantee global convergence.
Regarding the flow around a sphere configuration we obtain unsteady flow problems

if we increase the Reynolds number. Since the velocity is discretized by order 3 in space, a high order discretization in time could also be useful.
All above techniques will be important, if we use other equations to describe flows, e.g. non-Newtonian fluids depicted by the (non-)stationary (Navier-)Stokes equations which are real life problems.

References

[1] E. Bayraktar, O. Mierka & S. Turek, *Benchmark Computations of 3D Laminar Flow Around a Cylinder with CFX, OpenFOAM and FeatFlow*. International Journal of Computational Science and Engineering, 7, 3, 253-266, 2012.

[2] D. Boffi & L. Gastaldi, *On the quadrilateral Q2-P1 element for Stokes problem*. International Journal for numerical methods in fluid, pp 1001-1011, 2002.

[3] H. Damanik, *FEM Simulation of Non-isothermal Viscoelastic Fluids*. PhD thesis, Department of Mathematics, TU Dortmund University, 2011

[4] T. A. Davis, *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia, Sept. 2006.

[5] R. Jendrny, H. Damanik, O. Mierka & S. Turek, *Numerical studies for Flow Around a Sphere regarding different flow regimes caused by various Reynolds numbers*. Ergebnisberichte des Instituts für Angewandte Mathematik Nummer 531, Department of Mathematics, TU Dortmund University, 2015.

[6] S. Mandal, A. Ouazzi & S. Turek, *Modified Newton Solver for Yield Stress Fluids*. Numerical Mathematics and Advanced Applications ENUMATH 2015 (pp.481-490), Springer International Publishing, Cham, 2016.

[7] M. Möller, *Mathematical and Practical Aspects of Finite Elements*. Lecture Notes, Department of Mathematics, TU Dortmund University, 2010.

[8] R. Schmachtel, *Robuste lineare und nichtlineare Lösungsverfahren für die inkompressiblen Navier-Stokes-Gleichungen*. PhD thesis, Department of Mathematics, TU Dortmund University, 2003

[9] S. Turek, *Efficient solvers for incompressible flow problems: An Algorithmic and Computational Approach*. Springer, 3-540-65433-X, 1999.

[10] `http://www.featflow.de/en/benchmarks/cfdbenchmarking/flow/dfg_benchmark1_re20.html`, 2017

[11] `http://faculty.cse.tamu.edu/davis/suitesparse.html`, version 5.7.6, 2016

[12] `http://mumps.enseeiht.fr`, version 5.1.1, 2017