Efficient Newton-multigrid solution techniques for higher order space-time Galerkin discretizations of incompressible flow[☆]

S. Hussain[a,*], F. Schieweck[b], S. Turek[c]

[a] *Department of Mathematics, Mohammad Ali Jinnah University, Islamabad, Pakistan*
[b] *Institut für Analysis und Numerik, Otto-von-Guericke-Universität Magdeburg, Germany*
[c] *Institut für Angewandte Mathematik (LS III), TU Dortmund, Germany*

Abstract

In this paper, we discuss solution techniques of Newton-multigrid type for the resulting nonlinear saddle-point block-systems if higher order *continuous Galerkin-Petrov* (cGP(k)) and *discontinuous Galerkin* (dG(k)) time discretizations are applied to the nonstationary incompressible Navier-Stokes equations. In particular for the cGP(2) method with quadratic ansatz functions in time, which lead to 3rd order accuracy in the L^2-norm and even to 4th order superconvergence in the endpoints of the time intervals, together with the finite element pair Q_2/P_1^{disc} for the spatial approximation of velocity and pressure leading to a globally 3rd order scheme, we explain the algorithmic details as well as implementation aspects. All presented solvers are analyzed with respect to their numerical costs for two prototypical flow configurations.

Keywords: Continuous Galerkin-Petrov method, discontinuous Galerkin method, incompressible Navier-Stokes equations, Newton-multigrid solver

1. Introduction

To perform nonstationary simulations of incompressible flows, one of the main aims is to find time stepping schemes which allow relatively large time step sizes to gain accurate results at minimum numerical cost. Here, higher order methods are often essential in order to achieve highly accurate results on computationally feasible grids. A well-known approach to solve time dependent problems with higher order accuracy is the Galerkin method, see for instance [1]. In [2, 3, 4], we have analyzed a special class of time discretizations of variational type, called *continuous* Galerkin-Petrov (cGP) schemes, which show highly accurate results, particularly in comparison to standard one-step schemes like Crank-Nicolson and also discontinuous Galerkin (dG) time discretizations. To be precise: The cGP(2) approach with quadratic ansatz functions in time leads to 3rd order accurate

[☆]The authors thank the German Research Association (DFG: SCHI 576/2-1, TU 102/35-1) and the Higher Education Commission (HEC) of Pakistan (LC06052) for financial support.

[*]Corresponding author
Email address: shafqat.hussain@jinnah.edu.pk (S. Hussain)

results in each time point and even to 4th order accuracy in the endpoints of the time intervals. This fully implicit A-stable method requires the solution of a nonlinear saddle-point block-system in each time step which consists of two coupled generalized Navier-Stokes equations. Therefore, fourth order in time can be obtained while the number of equations is only doubled in comparison to classical second order one-step schemes. However, to realize an efficient scheme which requires less CPU time for achieving comparable accuracy, also solvers of optimal complexity, that means $O(N)$ with N denoting the total number of unknowns in space, are necessary. Candidates for such solvers are Newton-like methods with superlinear convergence behaviour for treating the nonlinear problems while multigrid-Krylov solvers are preferred choices for solving the auxiliary linear subproblems.

In this paper, we present corresponding solution techniques which are based on the classical Newton method for solving the resulting nonlinear systems while the associated linear subproblems are treated by using a monolithic (geometrical) multigrid solver which exploits the characteristics of the underlying Q_2/P_1^{disc} finite element spaces (for the spatial discretization) for the construction of the grid transfer routines. As smoothing operator, we extend the classical ideas of local pressure Schur complement techniques, resp., Vanka-like methods, which treat all flow variables simultaneously, but locally on smaller patches as for instance the mesh cells. In the final step, these simple schemes are used as preconditioners in a corresponding Krylov-space method (here: GMRES) which acts as a more robust smoother in the presented multigrid framework. We explain the algorithmic details and implementation aspects and analyze the resulting efficiency w.r.t. the convergence behavior as well as the observed temporal accuracy, for different mesh sizes in space and time, and for two prototypical flow configurations with benchmarking character. The corresponding test cases are the classical *flow around cylinder* problem and the *flow through a Venturi pipe*. These test cases show the high potential of the presented methodology. On the one hand, the described Newton-multigrid solvers lead to computational costs which are approximately three times higher per time step compared to a standard scheme like Crank-Nicolson. On the other hand, much larger time steps (a factor of 5-10 in our test configurations) can be used to guarantee the same temporal accuracy as the second order Crank-Nicolson scheme. As a main result, the resulting $cGP(2) - Q_2/P_1^{disc}$ scheme is optimal for 2D simulations in that sense that one global refinement step, which leads to appr. 8 times more unknowns in space and time, requires appr. 8 times more CPU time while at the same time the accuracy in space and time is improved by a factor of appr. 8, too (remark: in 3D, the corresponding 'optimal' approach would need 4th order accurate results in space and time).

The paper is organized as follows: Section 2 describes the discretization of the Navier-Stokes equations in space and time by using Galerkin methods. The resulting discretized block-systems for different time discretization schemes are presented in Section 3. In Section 4, we explain the solution techniques for the resulting discrete problems. Besides the (classical) Newton scheme, we propose a multigrid method for the solution of the linear subproblems in Section 5. Finally, Section 6 presents the numerical results for a couple of test problems to analyze the temporal accuracy, resp., the efficiency of the discussed solution methods. In Section 7, the paper is concluded with a discussion of the results.

2. FEM discretization in space and time

For a domain $\Omega \subset \mathbb{R}^d$, we consider the nonstationary incompressible Navier-Stokes equations, i.e. we want to find for each time $t \in [0,T]$ a velocity field $\mathbf{u}(t) : \Omega \to \mathbb{R}^d$ and a pressure field $p(t) : \Omega \to \mathbb{R}$ *such that*

$$\begin{aligned} \partial_t \mathbf{u} - \nu \triangle \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p &= f, && \text{in} \quad \Omega \times (0,T], \\ \operatorname{div} \mathbf{u} &= 0 && \text{in} \quad \Omega \times (0,T], \\ \mathbf{u} &= g && \text{on} \quad \partial\Omega \times (0,T], \\ \mathbf{u}(x,0) &= \mathbf{u}_0(x) && \text{in} \quad \Omega \quad \text{for} \quad t = 0, \end{aligned} \tag{1}$$

where ν denotes the viscosity, f the body force and $\mathbf{u}_0$ the initial velocity field at time $t = 0$. For simplicity, we restrict to the case $d = 2$ and we assume homogeneous Dirichlet conditions at the boundary $\partial\Omega$ of a polygonal domain Ω (for other choices see [5]). To make this problem well-posed in the case of pure Dirichlet boundary conditions, we have to look for the field $p(t)$ at each time t in the subspace $L^2_0(\Omega) \subset L^2(\Omega)$ of functions with zero integral mean value.

For the time discretization, we decompose the time interval $I = (0,T]$ into N disjoint subintervals $I_n := (t_{n-1}, t_n]$, where $n = 1, \ldots, N$ and $0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T$. Thus, the value of the time-discrete approximation $\mathbf{u}_\tau$ at time t_n is always defined as the I_n-value (i.e. the left-sided value in case of discontinuous approximation) $\mathbf{u}_\tau(t_n) := \mathbf{u}^- := \mathbf{u}_\tau|_{I_n}(t_n)$. The symbol $\boldsymbol{\tau}$ denotes the *time discretization parameter* and is also used as the maximum time step size $\tau := \max_{1 \le n \le N} \tau_n$, where $\tau_n := t_n - t_{n-1}$. Then, for the subsequent continuous and discontinuous Galerkin time stepping schemes, we approximate the solution $\mathbf{u}$ by means of a function $\mathbf{u}_\tau$ which is piecewise polynomial with respect to time. In case of the cGP(k)-method, we are looking for $\mathbf{u}_\tau$ in the discrete time-continuous space (with $\mathbf{V} = (H^1_0(\Omega))^2$)

$$\mathbf{X}^k_\tau := \{\mathbf{u} \in C(I, \mathbf{V}) : \ \mathbf{u}\big|_{I_n} \in \mathbb{P}_k(I_n, \mathbf{V}) \quad \forall\, n = 1, \ldots, N\}, \tag{2}$$

where

$$\mathbb{P}_k(I_n, \mathbf{V}) := \Big\{ \mathbf{u} : I_n \to \mathbf{V} \ : \ \mathbf{u}(t) = \sum_{j=0}^{k} \mathbf{U}^j t^j \,, \ \forall\, t \in I_n, \ \mathbf{U}^j \in \mathbf{V}, \ \forall\, j \Big\}. \tag{3}$$

Moreover, we introduce the discrete time-discontinuous test space

$$\mathbf{Y}^{k-1}_\tau := \{\mathbf{v} \in L^2(I, \mathbf{V}) : \ \mathbf{v}\big|_{I_n} \in \mathbb{P}_{k-1}(I_n, \mathbf{V}) \quad \forall\, n = 1, \ldots, N\} \tag{4}$$

consisting of piecewise polynomials of order $k-1$ which are (globally) discontinuous at the end points of the time intervals. Similarly, we will use for the time-discrete pressure p_τ an analogous ansatz space $\tilde{X}^k_\tau$, where the vector valued space $\mathbf{V}$ is replaced by the scalar valued space $Q = L^2_0(\Omega)$, and an analogous discontinuous test space $\tilde{Y}^{k-1}_\tau$.

In case of the dG($k-1$)-method, we are looking for $\mathbf{u}_\tau$ in the time-discontinuous discrete space $\mathbf{Y}^{k-1}_\tau$. Next, we describe separately the cGP(k) and dG($k-1$)-method.

2.1. cGP(k)-method

In order to derive the time discretization, we multiply the equations in (1) with some suitable I_n-supported test functions and integrate over $\Omega \times I_n$. To determine $\mathbf{u}_\tau|_{I_n}$ and $p_\tau|_{I_n}$ we represent them by the polynomial ansatz

$$\mathbf{u}_\tau|_{I_n}(t) := \sum_{j=0}^{k} \mathbf{U}_n^j \phi_{n,j}(t), \qquad p_\tau|_{I_n}(t) := \sum_{j=0}^{k} P_n^j \phi_{n,j}(t), \tag{5}$$

where the "coefficients" $(\mathbf{U}_n^j, P_n^j)$ are elements of the function spaces $\mathbf{V} \times Q$ and the polynomial functions $\phi_{n,j} \in \mathbb{P}_k(I_n)$ are the Lagrange basis functions with respect to the $k+1$ nodal points $t_{n,j} \in I_n$ satisfying the conditions

$$\phi_{n,j}(t_{n,i}) = \delta_{i,j}, \qquad i,j = 0,\ldots,k \tag{6}$$

with the Kronecker symbol $\delta_{i,j}$. For an easy treatment of the initial condition, we set $t_{n,0} = t_{n-1}$. Then, the initial condition is equivalent to the condition

$$\mathbf{U}_n^0 = \mathbf{u}_\tau|_{I_{n-1}}(t_{n-1}) \quad \text{if} \quad n \geq 2 \qquad \text{or} \qquad \mathbf{U}_n^0 = \mathbf{u}_0 \quad \text{if} \quad n = 1. \tag{7}$$

The other points $t_{n,1},\ldots,t_{n,k}$ are chosen as the quadrature points of the k-point Gaussian formula on I_n which is exact if the function to be integrated is a polynomial of degree less or equal to $2k-1$. We define the basis functions $\phi_{n,j} \in \mathbb{P}_k(I_n)$ of (5) via affine reference transformations (see [4, 3] for more details). Now, we can describe the *time discrete I_n-problem of the cGP(k)-method* [4, 6]:

Find on the interval $I_n = (t_{n-1}, t_n]$ the k unknown pairs of "coefficients" $(\mathbf{U}_n^j, P_n^j) \in \mathbf{V} \times Q$, $j = 1,\ldots,k$, such that for all $i = 1,\ldots,k$, it holds for all $(\mathbf{v}, q) \in \mathbf{V} \times Q$

$$\begin{aligned} \sum_{j=0}^{k} \alpha_{i,j} \left(\mathbf{U}_n^j, \mathbf{v}\right)_\Omega + \frac{\tau_n}{2} a(\mathbf{U}_n^i, \mathbf{v}) + \frac{\tau_n}{2} n(\mathbf{U}_n^i, \mathbf{U}_n^i, \mathbf{v}) + \frac{\tau_n}{2} b(\mathbf{v}, P_n^i) &= \tfrac{\tau_n}{2} \left(f(t_{n,i}), \mathbf{v}\right)_\Omega \\ b(\mathbf{U}_n^i, q) &= 0 \end{aligned} \tag{8}$$

with $\mathbf{U}_n^0 := \mathbf{u}_\tau(t_{n-1})$ for $n > 1$, $\mathbf{U}_1^0 := \mathbf{u}_0$ and $(\cdot,\cdot)_\Omega$ denotes the usual inner product in $(L^2(\Omega))^d$. The bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ on $\mathbf{V}\times\mathbf{V}$ and $\mathbf{V}\times Q$, respectively, are defined as

$$a(\mathbf{u}, \mathbf{v}) := \int_\Omega \nabla \mathbf{u} \cdot \nabla \mathbf{v}\, dx \quad \forall\, \mathbf{u}, \mathbf{v} \in \mathbf{V}, \qquad b(\mathbf{v}, p) := -\int_\Omega \nabla \cdot \mathbf{v}\, p\, dx, \tag{9}$$

and the trilinear form $n(\cdot,\cdot,\cdot)$ on $\mathbf{V}\times\mathbf{V}\times\mathbf{V}$ is given as $n(\mathbf{w},\mathbf{u},\mathbf{v}) := \sum_{i=1}^d n_s(\mathbf{w}, u_i, v_i)$ where

$$n_s(\mathbf{w}, u_i, v_i) := \int_\Omega (\mathbf{w} \cdot \nabla u_i) v_i\, dx \quad \forall\, \mathbf{w} \in \mathbf{V}, \quad u_i, v_i \in H_0^1(\Omega). \tag{10}$$

A typical property of this cGP(k)-variant is that the initial pressure P_n^0 of the ansatz (5) does not occur in this formulation. In order to achieve superconvergence for the pressure approximation at the discrete time levels t_n, special interpolation techniques using two neighbored time intervals can be applied (see [3]).

In the following subsections, we specify the constants $\alpha_{i,j}$ of the cGP(k)-method for the cases $k = 1$ and $k = 2$, and for comparison we describe explicitly the well-known dG(1) approach (see [3] for more details).

2.1.1. cGP(1)-method.

We use the one-point Gaussian quadrature formula with $\hat{t}_1 = 0$ and $t_{n,1} = t_{n-1} + \frac{\tau_n}{2}$. Then, we get $\alpha_{1,0} = -1$ and $\alpha_{1,1} = 1$ (see [4, 3]). Thus, problem (8) leads to the following problem for the "one" pair of unknowns $\mathbf{U}_n^1 = \mathbf{u}_\tau(t_{n-1} + \frac{\tau_n}{2})$ and $P_n^1 = p_\tau(t_{n-1} + \frac{\tau_n}{2})$: Find $(\mathbf{U}_n^1, P_n^1) \in \mathbf{V} \times Q$ such that for all $(\mathbf{v}, q) \in \mathbf{V} \times Q$ it holds

$$\begin{aligned} \left(\mathbf{U}_n^1, \mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2} a(\mathbf{U}_n^1, \mathbf{v}) + \tfrac{\tau_n}{2} n(\mathbf{U}_n^1, \mathbf{U}_n^1, \mathbf{v}) + \tfrac{\tau_n}{2} b(\mathbf{v}, P_n^1) &= \tfrac{\tau_n}{2} \left(f(t_{n,1}), \mathbf{v}\right)_\Omega + \left(\mathbf{U}_n^0, \mathbf{v}\right)_\Omega \\ b(\mathbf{U}_n^1, q) &= 0. \end{aligned} \tag{11}$$

Once we have determined the solution $\mathbf{U}_n^1$ at the midpoint $t_{n,1}$ of the time interval I_n, we get the solution at the next discrete time point t_n simply by linear extrapolation based on the ansatz (5), i.e.,

$$\mathbf{u}_\tau(t_n) = 2\mathbf{U}_n^1 - \mathbf{U}_n^0, \tag{12}$$

where $\mathbf{U}_n^0$ is the initial value at the time interval $(t_{n-1}, t_n]$ coming from the previous time interval I_{n-1} or the initial value $\mathbf{u}_0$.

If we would replace $f(t_{n,1})$ by the mean value $(f(t_{n-1}) + f(t_n))/2$, which means that we replace the one-point Gaussian quadrature of the right hand side by the trapezoidal rule, the resulting cGP(1)-method is equivalent to the well-known *Crank-Nicolson scheme*. The cGP(1)-method is accurate of order 2 in the whole time interval as it is known for the Crank-Nicolson scheme. Concerning the pressure approximation, one observes that the second order accuracy holds only in the midpoints of the time intervals. By means of linear interpolation between the midpoints of two neighboring time intervals we get second order accuracy also at the discrete time levels t_n.

2.1.2. cGP(2)-method.

Here, we use the 2-point Gaussian quadrature formula with the points $\hat{t}_1 = -\frac{1}{\sqrt{3}}$ and $\hat{t}_2 = \frac{1}{\sqrt{3}}$. Then, we obtain the coefficients

$$(\alpha_{i,j}) = \begin{pmatrix} -\sqrt{3} & \frac{3}{2} & \frac{2\sqrt{3}-3}{2} \\ \sqrt{3} & \frac{-2\sqrt{3}-3}{2} & \frac{3}{2} \end{pmatrix} \qquad i = 1, 2,\ j = 0, 1, 2. \tag{13}$$

Consequently, on the time interval I_n, we have to solve for the two "unknowns"

$$(\mathbf{U}_n^j, P_n^j) = \left(\mathbf{u}_\tau(t_{n,j}), p_\tau(t_{n,j})\right) \in \mathbf{V} \times Q \qquad \text{with} \qquad t_{n,j} := (t_{n-1} + t_n + \tau_n \hat{t}_j)/2 \quad \text{for} \quad j = 1, 2.$$

The corresponding coupled system reads

$$\begin{aligned} \alpha_{1,1} \left(\mathbf{U}_n^1, \mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2} a(\mathbf{U}_n^1, \mathbf{v}) + \tfrac{\tau_n}{2} n(\mathbf{U}_n^1, \mathbf{U}_n^1, \mathbf{v}) + \alpha_{1,2} \left(\mathbf{U}_n^2, \mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2} b(\mathbf{v}, P_n^1) &= \ell_1(\mathbf{v}) \\ \alpha_{2,1} \left(\mathbf{U}_n^1, \mathbf{v}\right)_\Omega + \alpha_{2,2} \left(\mathbf{U}_n^2, \mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2} a(\mathbf{U}_n^2, \mathbf{v}) + \tfrac{\tau_n}{2} n(\mathbf{U}_n^2, \mathbf{U}_n^2, \mathbf{v}) + \tfrac{\tau_n}{2} b(\mathbf{v}, P_n^2) &= \ell_2(\mathbf{v}) \\ b(\mathbf{U}_n^1, q) &= 0 \\ b(\mathbf{U}_n^2, q) &= 0, \end{aligned} \tag{14}$$

which has to be satisfied for all $(\mathbf{v}, q) \in \mathbf{V} \times Q$ with

$$\ell_i(\mathbf{v}) = \frac{\tau_n}{2} \left(f(t_{n,i}), \mathbf{v}\right)_\Omega - \alpha_{i,0} \left(\mathbf{U}_n^0, \mathbf{v}\right)_\Omega \qquad i = 1, 2. \tag{15}$$

Once we have determined the solutions $\mathbf{U}_n^1, \mathbf{U}_n^2$ at the Gaussian points in the interior of the interval I_n, we get the solution at the right boundary t_n of I_n by means of quadratic extrapolation from the ansatz (5), i.e.,

$$\mathbf{u}_\tau(t_n) = \mathbf{U}_n^0 + \sqrt{3}(\mathbf{U}_n^2 - \mathbf{U}_n^1), \tag{16}$$

where $\mathbf{U}_n^0$ is the initial value at the time interval I_n. The cGP(2)-method is accurate of order 3 in the whole time interval and superconvergent of order 4 in the discrete time points (see [4, 3]).

2.2. dG(k − 1)-method

Here, the time-discrete velocity and pressure solution is determined in the solution space $(\mathbf{u}_\tau, p_\tau) \in \mathbf{Y}_\tau^{k-1} \times \tilde{Y}_\tau^{k-1}$, where $k \geq 1$. The ansatz for $(\mathbf{u}_\tau, p_\tau)$ on interval I_n is then analog to (5) with the difference that the sum starts with $j = 1$ and the scalar basis functions $\phi_{n,j}$ are polynomials of order $k - 1$. In this paper, we will concentrate only on the case $k = 2$, i.e. on the well-known dG(1)-method. We can derive the following constants for $i, j \in \{1, 2\}$ (see again [4] and [3])

$$(\alpha_{i,j}) = \begin{pmatrix} 1 & \frac{\sqrt{3}-1}{2} \\ \frac{-\sqrt{3}-1}{2} & 1 \end{pmatrix}, \quad (d_i) = \begin{pmatrix} \frac{\sqrt{3}+1}{2} \\ \frac{-\sqrt{3}+1}{2} \end{pmatrix}. \tag{17}$$

Then, on the time interval I_n, one has to determine the two "unknowns" $(\mathbf{U}_n^j, P_n^j) \in \mathbf{V} \times Q$ as the solution of the following coupled system

$$\begin{aligned} \alpha_{1,1}\left(\mathbf{U}_n^1, \mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2} a(\mathbf{U}_n^1, \mathbf{v}) + \tfrac{\tau_n}{2} n(\mathbf{U}_n^1, \mathbf{U}_n^1, \mathbf{v}) + \tfrac{\tau_n}{2} b(\mathbf{v}, P_n^1) + \alpha_{1,2}\left(\mathbf{U}_n^2, \mathbf{v}\right)_\Omega &= \ell_1(\mathbf{v}) \\ \alpha_{2,1}\left(\mathbf{U}_n^1, \mathbf{v}\right)_\Omega + \alpha_{2,2}\left(\mathbf{U}_n^2, \mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2} a(\mathbf{U}_n^2, \mathbf{v}) + \tfrac{\tau_n}{2} n(\mathbf{U}_n^2, \mathbf{U}_n^2, \mathbf{v}) + \tfrac{\tau_n}{2} b(\mathbf{v}, P_n^2) &= \ell_2(\mathbf{v}) \\ b(\mathbf{U}_n^1, q) &= 0 \\ b(\mathbf{U}_n^2, q) &= 0 \end{aligned} \tag{18}$$

which has to be satisfied for all $(\mathbf{v}, q) \in \mathbf{V} \times Q$ with $\ell_i(\cdot)$ defined by

$$\ell_i(\mathbf{v}) = \frac{\tau_n}{2}\left(f(t_{n,i}), \mathbf{v}\right)_\Omega + d_i\left(\mathbf{U}_n^0, \mathbf{v}\right)_\Omega \qquad i = 1, 2. \tag{19}$$

After solving the above system, we obtain $\mathbf{u}_\tau$ and p_τ at time t_n by means of the following linear extrapolation

$$\mathbf{u}_\tau(t_n) = \frac{\sqrt{3}+1}{2}\mathbf{U}_n^2 - \frac{\sqrt{3}-1}{2}\mathbf{U}_n^1 \qquad \text{and} \qquad p_\tau(t_n) = \frac{\sqrt{3}+1}{2}P_n^2 - \frac{\sqrt{3}-1}{2}P_n^1. \tag{20}$$

The dG(1)-method is of order 2 in the whole time interval and superconvergent of order 3 in the discrete time points (see [4, 3]).

After discretizing the Navier-Stokes equations (1) in time (see [2]), we now discretize the resulting "I_n-problems" in space by using the finite element method [1, 7, 8, 9]. In our numerical experiments, the finite element spaces $\mathbf{V}_h \subset \mathbf{V}$ and $Q_h \subset Q$ are defined by biquadratic and discontinuous linear finite elements, respectively, on a quadrilateral

mesh T_h covering the computational domain Ω. Each "I_n-problem" for the cGP(k) or the dG($k-1$)-approach has the structure:

For given $\mathbf{U}_n^0 \in \mathbf{V}$, *find* $(\mathbf{U}_n^j, P_n^j) \in \mathbf{V} \times Q$, $j = 1, \ldots, k$, *such that*

$$\begin{aligned} \sum_{j=1}^{k} \alpha_{i,j} \left(\mathbf{U}_n^j, \mathbf{v}\right)_\Omega + \frac{\tau_n}{2} a(\mathbf{U}_n^i, \mathbf{v}) + \frac{\tau_n}{2} n(\mathbf{U}_n^i, \mathbf{U}_n^i, \mathbf{v}) + \frac{\tau_n}{2} b(\mathbf{v}, P_n^i) &= \ell_i(\mathbf{v}) \\ b(\mathbf{U}_n^i, q) &= 0, \end{aligned} \tag{21}$$

which has to be satisfied for all $i = 1, \ldots, k$ *and all* $(\mathbf{v}, q) \in \mathbf{V} \times Q$ *with*

$$\ell_i(\mathbf{v}) := \frac{\tau_n}{2} \left(f(t_{n,i}), \mathbf{v}\right)_\Omega + d_i \left(\mathbf{U}_n^0, \mathbf{v}\right)_\Omega \tag{22}$$

where $\alpha_{i,j}$ *and* d_i *are the constants described above and* $d_i = -\alpha_{i,0}$ *in case of cGP(k) .*

For the space discretization, all $(\mathbf{U}_n^j, P_n^j) \in \mathbf{V} \times Q$ are approximated by finite element functions $(\mathbf{U}_{n,h}^j, P_{n,h}^j) \in \mathbf{V}_h \times Q_h$, respectively, and the fully discrete "I_n-problem" reads:

For given $\mathbf{U}_{n,h}^0 \in \mathbf{V}_h$, *find* $(\mathbf{U}_{n,h}^j, P_{n,h}^j) \in \mathbf{V}_h \times Q_h$, $j = 1, \ldots, k$, *such that it holds*

$$\begin{aligned} \sum_{j=1}^{k} \alpha_{i,j} \left(\mathbf{U}_{n,h}^j, \mathbf{v}_h\right)_\Omega + \frac{\tau_n}{2} a(\mathbf{U}_{n,h}^i, \mathbf{v}_h) + \frac{\tau_n}{2} n(\mathbf{U}_{n,h}^i, \mathbf{U}_{n,h}^i, \mathbf{v}_h) + \frac{\tau_n}{2} b(\mathbf{v}_h, P_{n,h}^i) &= \ell_i(\mathbf{v}_h) \\ b(\mathbf{U}_{n,h}^i, q_h) &= 0 \end{aligned} \tag{23}$$

for all $(\mathbf{v}_h, q_h) \in \mathbf{V}_h \times Q_h$ *and all* $i = 1, \ldots, k$.

Once we have solved this system, we have computed for each time $t \in I_n$ a finite element approximation $u_{\tau,h}(t) \in \mathbf{V}_h$ of the time discrete solution $\mathbf{u}_\tau(t) \in \mathbf{V}$ which is defined by an analogous ansatz to (5) where the $\mathbf{U}_n^j \in \mathbf{V}$ are replaced by the discrete functions $\mathbf{U}_{n,h}^j \in \mathbf{V}_h$.

In the following, we will write problem (23) as a nonlinear algebraic block system. Let $S_h \subset H_0^1(\Omega)$ denote the scalar finite element space for the velocity components of $\mathbf{U}_{n,h}^j = (U_{n,h}^j, V_{n,h}^j) \in \mathbf{V}_h = S_h^2$ and let $\phi_\mu \in S_h$, $\mu = 1, \ldots, m_h$, denote the scalar finite element basis functions of S_h. Then, we define the nodal vector $\underline{\mathbf{U}}_n^j = (\underline{U}_n^j, \underline{V}_n^j) \in \mathbb{R}^{2m_h}$ of $\mathbf{U}_{n,h}^j = (U_{n,h}^j, V_{n,h}^j) \in \mathbf{V}_h$ such that

$$U_{n,h}^j(x) = \sum_{\mu=1}^{m_h} (\underline{U}_n^j)_\mu \phi_\mu(x), \qquad V_{n,h}^j(x) = \sum_{\mu=1}^{m_h} (\underline{V}_n^j)_\mu \phi_\mu(x) \qquad \forall\, x \in \Omega. \tag{24}$$

Similarly for the pressure, let $\psi_\mu \in Q_h$, $\mu = 1, \ldots, n_h$, denote the finite element basis functions and $\underline{P}_n^j \in \mathbb{R}^{n_h}$ the nodal vector of $P_{n,h}^j \in Q_h$ such that

$$P_{n,h}^j(x) = \sum_{\mu=1}^{n_h} (\underline{P}_n^j)_\mu \psi_\mu(x) \qquad \forall\, x \in \Omega. \tag{25}$$

Furthermore, we introduce the mass matrix $M \in \mathbb{R}^{m_h \times m_h}$, the discrete Laplacian matrix $L \in \mathbb{R}^{m_h \times m_h}$, the gradient matrices $B_i \in \mathbb{R}^{n_h \times m_h}$, $i = 1, 2$, as

$$M_{\nu,\mu} := (\phi_\mu, \phi_\nu)_\Omega\,, \quad L_{\nu,\mu} := a(\phi_\mu, \phi_\nu), \quad (B_i)_{\nu,\mu} := b(\phi_\mu \mathbf{e}^i, \psi_\nu), \tag{26}$$

and the right hand side vectors $F_n^i, G_n^i \in \mathbb{R}^{m_h}$, $i = 1, \ldots, k$, with the components

$$(F_n^i)_\nu := \big(f(t_{n,i}), \phi_\nu \mathbf{e}^1\big)_\Omega\,, \quad (G_n^i)_\nu := \big(f(t_{n,i}), \phi_\nu \mathbf{e}^2\big)_\Omega\,. \tag{27}$$

Next, for a given nodal vector $\underline{\mathbf{w}} = (\underline{\mathbf{w}}^1, \underline{\mathbf{w}}^2) \in \mathbb{R}^{2m_h}$, which generates an associated discrete velocity field $\mathbf{w}_h(\underline{\mathbf{w}}) \in \mathbf{V}_h$, we define the matrix $N(\underline{\mathbf{w}}) \in \mathbb{R}^{m_h \times m_h}$ as

$$N(\underline{\mathbf{w}})_{\nu,\mu} := n_s(\mathbf{w}_h(\underline{\mathbf{w}}), \phi_\mu, \phi_\nu), \qquad \text{with} \qquad \mathbf{w}_h(\underline{\mathbf{w}}) := \sum_{j=1}^{m_h} \begin{pmatrix} \underline{\mathbf{w}}_j^1 \\ \underline{\mathbf{w}}_j^2 \end{pmatrix} \phi_j. \tag{28}$$

Using the block-matrices and block-vectors

$$\mathbf{M} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix}, \quad \mathbf{N}(\underline{\mathbf{w}}) = \begin{bmatrix} N(\underline{\mathbf{w}}) & 0 \\ 0 & N(\underline{\mathbf{w}}) \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \mathbf{F}_n^i = \begin{bmatrix} F_n^i \\ G_n^i \end{bmatrix}, \tag{29}$$

the fully discrete "I_n-problem" is equivalent to the following nonlinear $k \times k$ block system:

For given $\underline{\mathbf{U}}_n^0 \in \mathbb{R}^{2m_h}$, find $\underline{\mathbf{U}}_n^j \in \mathbb{R}^{2m_h}$ and $\underline{P}_n^j \in \mathbb{R}^{n_h}$, $j = 1, \ldots, k$, such that for all $i = 1, \ldots, k$, it holds

$$\begin{aligned} \sum_{j=1}^{k} \alpha_{i,j} \mathbf{M}\underline{\mathbf{U}}_n^j + \frac{\tau_n}{2}\mathbf{L}\underline{\mathbf{U}}_n^i + \frac{\tau_n}{2}\mathbf{N}(\underline{\mathbf{U}}_n^i)\underline{\mathbf{U}}_n^i + \frac{\tau_n}{2}\mathbf{B}\underline{P}_n^i &= d_i \mathbf{M}\underline{\mathbf{U}}_n^0 + \tfrac{\tau_n}{2}\mathbf{F}_n^i, \\ \mathbf{B}^T\underline{\mathbf{U}}_n^i &= 0. \end{aligned} \tag{30}$$

The vector $\underline{\mathbf{U}}_n^0$ is defined as the finite element nodal vector of the fully discrete solution $\mathbf{u}_{\tau,h}(t_{n-1})$ computed from the previous time interval $[t_{n-2}, t_{n-1}]$ if $n \geq 2$ or from a finite element interpolation of the initial data $\mathbf{u}_0$ if $n = 1$. In the case of higher Reynolds numbers, we apply additionally an edge oriented FEM stabilization (EOFEM) [10] for the convective term. This means that we replace the trilinear form $n(\mathbf{w}, \cdot, \cdot)$ by a modified form $n_h(\mathbf{w}, \cdot, \cdot)$ such that, in (30), differences will appear only in the nonlinear matrix $\mathbf{N}(\underline{\mathbf{w}})$.

In the following, we will present the resulting block systems for the cGP(1), cGP(2) and dG(1) method which are used in our numerical experiments.

3. Block-systems for the cGP and dG-methods

Here, we present the corresponding block systems for the cGP(1), cGP(2) and dG(1) methods, respectively.

3.1. cGP(1)-method

The 3×3 block system on the time interval I_n reads:

For given initial velocity $\underline{\mathbf{U}}_n^0 = (\underline{U}_n^0, \underline{V}_n^0)$, *find* $\underline{\mathbf{U}}_n^1 = (\underline{U}_n^1, \underline{V}_n^1)$ *and a pressure* $\underline{P}_n^1$ *such that*

$$\begin{bmatrix} A(u,v) & 0 & B_u \\ 0 & A(u,v) & B_v \\ B_u^T & B_v^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = \begin{bmatrix} R_u \\ R_v \\ 0 \end{bmatrix} \tag{31}$$

where

$$A(u,v) = M + \frac{\tau_n}{2} L + \frac{\tau_n}{2} N(u,v), \qquad B_u = B_1, \qquad B_v = B_2, \tag{32}$$

with the abbreviations

$$u = \underline{U}_n^1, \quad v = \underline{V}_n^1, \quad p = \frac{\tau_n}{2} \underline{P}_n^1 \tag{33}$$

and the convection matrix $N(u,v)$ denoting the matrix $N(\underline{\mathbf{w}})$ defined in (28) with the nodal vector $\underline{\mathbf{w}} := (u,v) \in \mathbb{R}^{2m_h}$. The right hand side vectors R_u and R_v are given by

$$R_u = \tfrac{\tau_n}{2} F_n^1 + M \underline{U}_n^0, \qquad R_v = \tfrac{\tau_n}{2} G_n^1 + M \underline{V}_n^0. \tag{34}$$

Once we have determined the solution $\underline{\mathbf{U}}_n^1 = (\underline{U}_n^1, \underline{V}_n^1)$ we compute the nodal vector $\underline{\mathbf{U}}_{n+1}^0 = (\underline{U}_{n+1}^0,\ \underline{V}_{n+1}^0)$ of the fully discrete solution $u_{\tau,h}$ at the time t_n by using the following linear extrapolation

$$u_{\tau,h}(t_n) \sim \underline{U}_{n+1}^0 = 2\underline{U}_n^1 - \underline{U}_n^0, \qquad v_{\tau,h}(t_n) \sim \underline{V}_{n+1}^0 = 2\underline{V}_n^1 - \underline{V}_n^0. \tag{35}$$

3.2. cGP(2)-method

The 6×6 block system on the time interval I_n reads:

For given initial velocity $\underline{\mathbf{U}}_n^0 = (\underline{U}_n^0, \underline{V}_n^0)$, *find* $\underline{U}_n^1, \underline{U}_n^2, \underline{V}_n^1, \underline{V}_n^2$ *and* $\underline{P}_n^1, \underline{P}_n^2$ *such that*

$$\begin{bmatrix} A(u,v) & 0 & B_u \\ 0 & A(u,v) & B_v \\ B_u^T & B_v^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = \begin{bmatrix} R_u \\ R_v \\ 0 \end{bmatrix} \tag{36}$$

where

$$A(u,v) = \begin{bmatrix} 3M + \tau_n L + \tau_n N(u^1, v^1) & \left(2\sqrt{3} - 3\right) M \\ \left(-2\sqrt{3} - 3\right) M & 3M + \tau_n L + \tau_n N(u^2, v^2) \end{bmatrix}, \tag{37}$$

$$B_u = \begin{bmatrix} B_1 & 0 \\ 0 & B_1 \end{bmatrix}, \qquad B_v = \begin{bmatrix} B_2 & 0 \\ 0 & B_2 \end{bmatrix} \tag{38}$$

with the abbreviations

$$u = \begin{bmatrix} u^1 \\ u^2 \end{bmatrix} = \begin{bmatrix} \underline{U}_n^1 \\ \underline{U}_n^2 \end{bmatrix}, \quad v = \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} = \begin{bmatrix} \underline{V}_n^1 \\ \underline{V}_n^2 \end{bmatrix}, \quad p = \begin{bmatrix} p^1 \\ p^2 \end{bmatrix} = \begin{bmatrix} \tau_n \underline{P}_n^1 \\ \tau_n \underline{P}_n^2 \end{bmatrix} \tag{39}$$

The right hand side vectors R_u and R_v are given by

$$R_u = \begin{bmatrix} R_u^1 \\ R_u^2 \end{bmatrix} = \begin{bmatrix} \tau_n F_n^1 + 2\sqrt{3} M \underline{U}_n^0 \\ \tau_n F_n^2 - 2\sqrt{3} M \underline{U}_n^0 \end{bmatrix}, \qquad R_v = \begin{bmatrix} R_v^1 \\ R_v^2 \end{bmatrix} = \begin{bmatrix} \tau_n G_n^1 + 2\sqrt{3} M \underline{V}_n^0 \\ \tau_n G_n^2 - 2\sqrt{3} M \underline{V}_n^0 \end{bmatrix}. \tag{40}$$

The nodal vectors $\underline{U}_n^i$ and $\underline{V}_n^i$, $i = 1, 2$, are associated with the finite element approximations $u_{\tau,h}(t_{n,i})$ and $v_{\tau,h}(t_{n,i})$, respectively, where $t_{n,i}$ denotes the i-th integration point of the 2-point Gaussian quadrature rule on the time interval I_n. Once they have been computed, we get the nodal vector $\underline{\mathbf{U}}_{n+1}^0 = (\underline{U}_{n+1}^0, \underline{V}_{n+1}^0)$ of the fully discrete solution $u_{\tau,h}$ at the time t_n by using the following quadratic extrapolation

$$u_{\tau,h}(t_n) \sim \underline{U}_{n+1}^0 = \underline{U}_n^0 + \sqrt{3}(\underline{U}_n^2 - \underline{U}_n^1), \qquad v_{\tau,h}(t_n) \sim \underline{V}_{n+1}^0 = \underline{V}_n^0 + \sqrt{3}(\underline{V}_n^2 - \underline{V}_n^1). \tag{41}$$

3.3. dG(1)-method

The analogous 6×6 block system on the time interval I_n reads:

For given initial velocity $\underline{\mathbf{U}}_n^0 = (\underline{U}_n^0, \underline{V}_n^0)$, *find* $\underline{U}_n^1, \underline{U}_n^2, \underline{V}_n^1, \underline{V}_n^2$ *and* $\underline{P}_n^1, \underline{P}_n^2$ *such that*

$$\begin{bmatrix} A(u,v) & 0 & B_u \\ 0 & A(u,v) & B_v \\ B_u^T & B_v^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = \begin{bmatrix} R_u \\ R_v \\ 0 \end{bmatrix} \tag{42}$$

where

$$A(u,v) = \begin{bmatrix} 2M + \tau_n L + \tau_n N(u^1, v^1) & \left(\sqrt{3} - 1\right) M \\ \left(-\sqrt{3} - 1\right) M & 2M + \tau_n L + \tau_n N(u^2, v^2) \end{bmatrix} \tag{43}$$

and B_u, B_v, u, v, p are defined as in (38) and (39). The right hand side vectors R_u and R_v are given by

$$R_u = \begin{bmatrix} R_u^1 \\ R_u^2 \end{bmatrix} = \begin{bmatrix} \tau_n F_n^1 + \left(\sqrt{3} + 1\right) M \underline{U}_n^0 \\ \tau_n F_n^2 + \left(-\sqrt{3} + 1\right) M \underline{U}_n^0 \end{bmatrix}, \quad R_v = \begin{bmatrix} R_v^1 \\ R_v^2 \end{bmatrix} = \begin{bmatrix} \tau_n G_n^1 + \left(\sqrt{3} + 1\right) M \underline{V}_n^0 \\ \tau_n G_n^2 + \left(-\sqrt{3} + 1\right) M \underline{V}_n^0 \end{bmatrix}. \tag{44}$$

Again the nodal vectors $\underline{U}_n^i$ and $\underline{V}_n^i$, $i = 1, 2$, are associated with the finite element approximations $u_{\tau,h}(t_{n,i})$ and $v_{\tau,h}(t_{n,i})$, respectively, where $t_{n,i}$ denotes the i-th integration point of the 2-point Gaussian quadrature rule on the time interval I_n. Once they have been computed, we get the nodal vector $\underline{\mathbf{U}}_{n+1}^0 = (\underline{U}_{n+1}^0, \underline{V}_{n+1}^0)$ and $\underline{P}_{n+1}^0$ of the left side limit of the fully discrete solution $u_{\tau,h}$ at the time t_n by using the following linear extrapolation

$$\begin{aligned} u_{\tau,h}^-(t_n) \sim \underline{U}_{n+1}^0 &= \tfrac{\sqrt{3}+1}{2} U_n^2 - \tfrac{\sqrt{3}-1}{2} U_n^1, \\ v_{\tau,h}^-(t_n) \sim \underline{V}_{n+1}^0 &= \tfrac{\sqrt{3}+1}{2} V_n^2 - \tfrac{\sqrt{3}-1}{2} V_n^1. \end{aligned} \tag{45}$$

4. Nonlinear Solver

For all introduced time-space discretization schemes described before, a nonlinear system of algebraic equations of the following type has to be solved for each time interval:

$$\begin{bmatrix} A(u,v) & 0 & B_u \\ 0 & A(u,v) & B_v \\ B_u^T & B_v^T & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = \begin{bmatrix} R_u \\ R_v \\ 0 \end{bmatrix} \tag{46}$$

The nonlinear system (46) can be characterized as a saddle point problem which will be solved either by means of a standard fixed point iteration or by the Newton method. We will denote this solution approach as the outer nonlinear iteration. In each outer iteration step, a coupled linear system has to be solved. The linear subproblems are treated by using a monolithic geometrical multigrid solver with a smoother based on blocking of all cell unknowns and with canonical FEM grid transfer operators. In addition, the coarse grid problem can be solved by using a direct solver or by a preconditioned GMRES method.

4.1. General nonlinear outer iteration

On an actual time interval $I_n = (t_{n-1}, t_n]$, we define the start iterate (u_0, v_0, p_0) of the nonlinear iteration by means of the known solution at time t_{n-1} from the previous time interval for $n > 1$ or from the space-discrete initial solution for $n = 1$. In the case of cGP(2) or dG(1), we define also the start iterates of the second components u^2, v^2 and p^2 in (39) as the solution at time t_{n-1}. For a given old iterate (u_ℓ, v_ℓ, p_ℓ), we perform the following three steps to compute the new iterate $(u_{\ell+1}, v_{\ell+1}, p_{\ell+1})$:

1. Compute the defect vector containing the nonlinear residual for (u_ℓ, v_ℓ, p_ℓ)

$$\begin{bmatrix} d_u \\ d_v \\ d_p \end{bmatrix} = \begin{bmatrix} R_u \\ R_v \\ 0 \end{bmatrix} - \begin{bmatrix} A(u_\ell, v_\ell) & 0 & B_u \\ 0 & A(u_\ell, v_\ell) & B_v \\ B_u^T & B_v^T & 0 \end{bmatrix} \begin{bmatrix} u_\ell \\ v_\ell \\ p_\ell \end{bmatrix}. \tag{47}$$

2. Solve an auxiliary (linear) subproblem with the defect vector as right hand side

$$\begin{bmatrix} A_{uu}(u_\ell, v_\ell) & A_{uv}(u_\ell, v_\ell) & B_u \\ A_{vu}(u_\ell, v_\ell) & A_{vv}(u_\ell, v_\ell) & B_v \\ B_u^T & B_v^T & 0 \end{bmatrix} \begin{bmatrix} \Delta u_\ell \\ \Delta v_\ell \\ \Delta p_\ell \end{bmatrix} = \begin{bmatrix} d_u \\ d_v \\ d_p \end{bmatrix}, \tag{48}$$

   where $A_{uu}, A_{uv}, A_{vu}, A_{vv}$ are chosen due to the fixed-point iteration or the Newton method.
3. Update the iterate to obtain $(u_{\ell+1}, v_{\ell+1}, p_{\ell+1})$

$$\begin{bmatrix} u_{\ell+1} \\ v_{\ell+1} \\ p_{\ell+1} \end{bmatrix} = \begin{bmatrix} u_\ell \\ v_\ell \\ p_\ell \end{bmatrix} + \omega_\ell \begin{bmatrix} \Delta u_\ell \\ \Delta v_\ell \\ \Delta p_\ell \end{bmatrix}, \tag{49}$$

where ω_ℓ is a damping parameter which is in most of the cases set to 1 (see [11] for adaptive strategies). The last iterate $(u_{\ell+1}, v_{\ell+1}, p_{\ell+1})$ for the nonlinear iteration, depending on the stopping criterion, is accepted for the solution. In our numerical simulations, the nonlinear iteration is stopped if the L^2-norm of the nonlinear residual drops down below a given tolerance (here: 10^{-10}).

4.2. Fixed-point iteration

For the fixed-point iteration, the matrices $A_{uu}, A_{uv}, A_{vu}, A_{vv}$ in (48) are given by

$$A_{uu}(u_\ell, v_\ell) := A_{vv}(u_\ell, v_\ell) := A(u_\ell, v_\ell), \tag{50}$$
$$A_{uv}(u_\ell, v_\ell) := A_{vu}(u_\ell, v_\ell) := 0. \tag{51}$$

4.3. Newton method

Applying the standard Newton method yields the following block matrices

$$A_{uu}(u_\ell, v_\ell) := A(u_\ell, v_\ell) + S^{(k)}_{uu}(u_\ell, v_\ell), \tag{52}$$
$$A_{vv}(u_\ell, v_\ell) := A(u_\ell, v_\ell) + S^{(k)}_{vv}(u_\ell, v_\ell), \tag{53}$$
$$A_{uv}(u_\ell, v_\ell) := S^{(k)}_{uv}(u_\ell, v_\ell), \tag{54}$$
$$A_{vu}(u_\ell, v_\ell) := S^{(k)}_{vu}(u_\ell, v_\ell), \tag{55}$$

where $k \in \{1, 2\}$ denotes the order of the underlying time discretization cGP(k) or dG($k-1$), respectively, and $S^{(k)}_{\alpha\beta}$ correspond to the additional terms from the linearization of the nonlinear convection term $(\mathbf{u}_\ell \cdot \nabla \mathbf{u}_\ell)$. In the following, we present explicitly for each time discretization the blocks of the full Newton matrix $\mathcal{N}$ in (48), i.e.,

$$\mathcal{N} := \begin{bmatrix} A(u_\ell, v_\ell) + S^{(k)}_{uu}(u_\ell, v_\ell) & S^{(k)}_{uv}(u_\ell, v_\ell) & B_u \\ S^{(k)}_{vu}(u_\ell, v_\ell) & A(u_\ell, v_\ell) + S^{(k)}_{vv}(u_\ell, v_\ell) & B_v \\ B_u^T & B_v^T & 0 \end{bmatrix}. \tag{56}$$

In the case of the cGP(1)-method, we have

$$A(u, v) = M + \frac{\tau_n}{2} L + \frac{\tau_n}{2} N(u, v), \qquad B_u = B_1, \qquad B_v = B_2, \tag{57}$$

and

$$\left(S^{(1)}_{uu}(\tilde{u}, \tilde{v})\right)_{\nu,\mu} = \frac{\tau_n}{2} \left((\partial_x \tilde{u}_h)\phi_\mu, \phi_\nu\right)_\Omega, \qquad \left(S^{(1)}_{uv}(\tilde{u}, \tilde{v})\right)_{\nu,\mu} = \frac{\tau_n}{2} \left((\partial_y \tilde{u}_h)\phi_\mu, \phi_\nu\right)_\Omega, \tag{58}$$
$$\left(S^{(1)}_{vu}(\tilde{u}, \tilde{v})\right)_{\nu,\mu} = \frac{\tau_n}{2} \left((\partial_x \tilde{v}_h)\phi_\mu, \phi_\nu\right)_\Omega, \qquad \left(S^{(1)}_{vv}(\tilde{u}, \tilde{v})\right)_{\nu,\mu} = \frac{\tau_n}{2} \left((\partial_y \tilde{v}_h)\phi_\mu, \phi_\nu\right)_\Omega, \tag{59}$$

where, $\tilde{u}_h \in S_h$ and $\tilde{v}_h \in S_h$ denote the finite element functions associated with the nodal vectors $\tilde{u}$ and $\tilde{v}$, respectively. For the cGP(2)-method, we have

$$A(u, v) = \begin{bmatrix} 3M + \tau_n L + \tau_n N(u^1, v^1) & \left(2\sqrt{3} - 3\right) M \\ \left(-2\sqrt{3} - 3\right) M & 3M + \tau_n L + \tau_n N(u^2, v^2) \end{bmatrix}, \tag{60}$$

$$B_u = \begin{bmatrix} B_1 & 0 \\ 0 & B_1 \end{bmatrix}, \qquad B_v = \begin{bmatrix} B_2 & 0 \\ 0 & B_2 \end{bmatrix}, \tag{61}$$

$$S^{(2)}_{\alpha\beta}(\tilde{u}, \tilde{v}) = \begin{bmatrix} 2S^{(1)}_{\alpha\beta}(\tilde{u}, \tilde{v}) & 0 \\ 0 & 2S^{(1)}_{\alpha\beta}(\tilde{u}, \tilde{v}) \end{bmatrix} \qquad \forall\ \alpha, \beta \in \{u, v\} \tag{62}$$

with $S^{(1)}_{\alpha\beta}(\tilde{u},\tilde{v})$ defined in (58) and (59). In the case of the dG(1)-method, the matrix block $A(u,v)$ is

$$A(u,v) = \begin{bmatrix} 2M + \tau_n L + \tau_n N(u^1, v^1) & \left(\sqrt{3}-1\right) M \\ \left(-\sqrt{3}-1\right) M & 2M + \tau_n L + \tau_n N(u^2, v^2) \end{bmatrix} \tag{63}$$

and the other matrix blocks of $\mathcal{N}$ in (56) are the same as for the cGP(2)-method.

Remark 1. *If the edge oriented jump stabilization (see [10]) is used, then the corresponding velocity matrix block $A(u,v)$ in (46) is updated as follows*

$$\tilde{A}(u_\ell, v_\ell) = A(u_\ell, v_\ell) + J, \tag{64}$$

where the matrix J corresponds to additional "jump terms" in the weak formulation. This matrix J has the following form for the cGP(1) and cGP(2)-method (and analogously for the dG(1)-method), respectively,

$$\textit{cGP(1):} \qquad J = (J_{\lambda,\mu}), \qquad J_{\lambda,\mu} = \sum_E max\{\gamma^* \nu h_E, \gamma h_E^2\} \int_E [\nabla \phi_\mu]_E [\nabla \phi_\lambda]_E \, d\sigma, \tag{65}$$

$$\textit{cGP(2):} \qquad J = \begin{bmatrix} (J_{\lambda,\mu}) & 0 \\ 0 & (J_{\lambda,\mu}) \end{bmatrix}, \tag{66}$$

where the sum in (65) is taken over all inner edges E of the mesh. ν, h_E denote the viscosity and the length of edge E. The parameters γ and γ^* have no significant influence on the accuracy of the results and the solution is stable and accurate for a large range of parameters. In our case, these parameters are set to 0.1 and 0, respectively. The jump $[.]_E$ is defined as $[\psi]_E = \psi^+ - \psi^-$, where ψ^+, ψ^- indicate the values of the discontinuous function ψ coming from the elements K^+, K^- sharing the interior edge E.

5. Linear solver

The resulting linear systems (48) on each time interval $[t_{n-1}, t_n]$, which are 6×6 block systems in the case of the cGP(2) and dG(1) approach and 3×3 block systems for the cGP(1)-method, are treated by using a geometrical multigrid solver with a smoother based on an element loop where, for each element, simultaneously all unknowns are updated that belong to this element. This type of smoother is also called "local pressure Schur complement smoother" in [11] or "Vanka-type smoother" in [12] (which can be additionally applied as preconditioner in a GMRES method to make this method more robust).

5.1. Geometric Multigrid Methods

The fundamental concept of multigrid techniques is to exploit different mesh levels of the underlying problem in order to reduce different frequencies of the error by applying a relatively cheap smoother on each grid level. For switching between successive mesh levels, one applies a restriction and prolongation operator with computational costs that are even cheaper than for the smoother and that are only proportional to the number of unknowns. A typical feature of multigrid methods is that they have a convergence rate which is nearly independent of the mesh size and that the computational costs per iteration are only proportional to the number of unknowns.

In the following, we present the standard multigrid algorithm that we have used in our computations. Independently from the previous notation we denote in the sequel the global system matrix assembled on mesh level $k \in \{1, \dots, L\}$ (and consisting of several block-matrices) by $A_k \in \mathbb{R}^{N_k \times N_k}$ where N_k denotes the total number of all degrees of freedom on level k. Similarly, we denote by $u_k, f_k \in \mathbb{R}^{N_k}$ the global solution and right hand side vector such that the problem that occurs on mesh level k has the form

$$A_k u_k = f_k. \tag{67}$$

Furthermore, let

$$I_{k-1}^k : \mathbb{R}^{N_{k-1}} \to \mathbb{R}^{N_k} \qquad \text{and} \qquad I_k^{k-1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k-1}}$$

denote the prolongation and restriction operator, respectively, acting between level $k-1$ and k. Then, the k-level geometric multigrid algorithm for solving (67) is described recursively in Algorithm 1.

Algorithm 1 Geometric multigrid algorithm.

The k-level iteration: $u_k^{\mathbf{new}} = MG(k, A_k, u_k^{\mathbf{old}}, f_k, \nu_1, \nu_2, \mu)$

case $k = 1$: u_1^{new} is the exact solution

$$u_1^{\text{new}} = A_1^{-1} f_1$$

case $k > 1$: Perform the following four steps:

1. Pre-smoothing: Compute $u_k^{\nu_1}$ by applying ν_1 smoothing iterations, i.e.,

$$u_k^i = S_k u_k^{i-1} \qquad \forall\, i = 1, \dots, \nu_1 \qquad \text{with} \qquad u_k^0 = u_k^{\text{old}}.$$

2. Restriction and correction: Compute the restricted residual

$$f_{k-1} = I_k^{k-1}(f_k - A_k u_k^{\nu_1})$$

and compute the correction u_{k-1}^{μ} by applying μ cycles on level $k-1$ starting with $u_{k-1}^0 = 0$, i.e.,

$$u_{k-1}^i = MG(k-1, A_{k-1}, u_{k-1}^{i-1}, f_{k-1}, \nu_1, \nu_2, \mu) \qquad \forall\, i = 1, \dots \mu.$$

3. Prolongation and step length control: Interpolate the correction onto grid level k to obtain $u_k^{\nu_1+1}$, i.e.,

$$u_k^{\nu_1+1} = u_k^{\nu_1} + \alpha_k I_{k-1}^k u_{k-1}^{\mu},$$

where α_k may be a fixed value or chosen adaptively to minimize the error $u_k^{\nu_1+1} - u_k$ in an appropriate norm [11].

4. Post-smoothing: Compute $u_k^{\text{new}} = u_k^{\nu_1+\nu_2+1}$ by applying ν_2 smoothing iterations to $u_k^{\nu_1+1}$, i.e.,

$$u_k^{\nu_1+i+1} = S_k u_k^{\nu_1+i} \qquad \forall\, i = 1, \dots, \nu_2.$$

Each application of MG(k, ...) is called a cycle on level k. Sufficiently many cycles on the finest grid level L are required to obtain a good approximate solution of a linear subproblem

$A_L u_L = f_L$ within the outer nonlinear iteration. The multigrid iteration is terminated if a prescribed convergence criterion is reached. In our numerical tests, the multigrid iteration is stopped if the L^2-norm of the relative residual is smaller than 10^{-6} or the absolute residual drops down below 10^{-15}. In the case when the number of coarse grid cycles is chosen as $\mu = 1$, we will speak of a V-cycle and in the case $\mu = 2$ of a so-called W-cycle. Another interesting case lies in between, the so called F-cycle, which we use in our application [11]. The numbers $\nu_1, \nu_2 \in \mathbb{N}$ of pre-smoothing and post-smoothing steps, respectively, is typically small, for instance between 1 and 4. Moreover, the coarsest grid should be coarse enough in order to keep the numerical costs for the exact solver small. The key ingredients in the multigrid methods are the smoother as well as the restriction and prolongation operators, which we will sketch in the following sections for the resulting 6×6, resp., 3×3 block-systems.

5.2. Smoother based on blocked cell unknowns

The efficiency and robustness of the multigrid method is essentially influenced by the choice of the smoother. In our numerical study, we employ a block-Gauß-Seidel type smoother where in an element loop, for each element, simultaneously the block of all unknowns that belong to this element cell is updated. Thus, the information which has been updated in previous elements, influences immediately all velocity and pressure degrees of freedom (for all required time points of the time interval) in the neighbored mesh cells. This type of smoother was originally proposed by Vanka [13] for the solution of Navier-Stokes equations discretized with the finite difference method and is therefore often called "Vanka-type smoother" [12, 14]. In [11] it has been analyzed as "local Multilevel Pressure Schur Complement smoother", and it can be additionally accelerated by embedding this approach as preconditioner in an outer GMRES method.

To describe this smoother, we have to introduce some notation. Let $\mathcal{T}_h$ denote the set of all element cells K on the actual mesh level, I_h the index set of all global degrees of freedom in the coupled system to be solved, $I_h(K)$ the index set of all global degrees of freedom that belong to mesh cell $K \in \mathcal{T}_h$ and $\hat{I}$ the index set of all local degrees of freedom that belong to the reference element $\hat{K}$. In any implementation of a finite element method, one has to store information about the following mapping

$$\mathrm{dof} : \mathcal{T}_h \times \hat{I} \to I_h, \qquad \mu = \mathrm{dof}(K, \hat{\mu}) \in I_h(K) \quad \forall\, \hat{\mu} \in \hat{I}, \tag{68}$$

where, for a given element $K \in \mathcal{T}_h$ and a given local degree of freedom belonging to K with the number $\hat{\mu} \in \hat{I}$, the uniquely assigned global number μ of this degree of freedom is given by $\mathrm{dof}(K, \hat{\mu})$. Let us assume that the coupled global linear system on the actual mesh level to which a smoothing iteration has to be applied is given in the form

$$Au = f. \tag{69}$$

Then, for a given element $K \in \mathcal{T}_h$, we define the local matrix $A_K \in \mathbb{R}^{\hat{n} \times \hat{n}}$ as

$$(A_K)_{\hat{\nu},\hat{\mu}} := A_{\mathrm{dof}(K,\hat{\nu}),\mathrm{dof}(K,\hat{\mu})} \qquad \forall\, \hat{\nu}, \hat{\mu} \in \hat{I} = \{1, \ldots, \hat{n}\}, \tag{70}$$

where $\hat{n}$ denotes the number of all local degrees of freedom on the reference element (velocity and pressure for all required time points on the actual time interval). Furthermore, we need a K-local restriction operator

$$R_K : \mathbb{R}^N \to \mathbb{R}^{\hat{n}}, \qquad (R_K d)_{\hat{\mu}} := d_{\mathrm{dof}(K,\hat{\mu})} \qquad \forall\, \hat{\mu} \in \hat{I} = \{1, \ldots, \hat{n}\}, \tag{71}$$

which assigns to a given global defect vector $d \in \mathbb{R}^N$ the local block-vector $R_K d \in \mathbb{R}^{\hat{n}}$ that contains all components of d that are associated with all degrees of freedom that belong to element K. Finally, we need a K-dependent extension operator

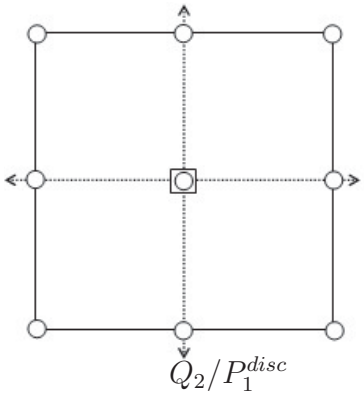
$$E_K : \mathbb{R}^{\hat{n}} \to \mathbb{R}^N, \qquad (E_K v_K)_\mu := \begin{cases} (v_K)_{\hat{\mu}} & \text{if} \quad \exists\, \hat{\mu} \in \hat{I} : \ \mu = \mathrm{dof}(K, \hat{\mu}), \\ 0 & \text{if} \quad \mu \notin I_h(K), \end{cases} \tag{72}$$

where, to a local correction vector $v_K \in \mathbb{R}^{\hat{n}}$ a global correction vector $v = E_K v_K \in \mathbb{R}^N$ is assigned which would update only those components in a global vector that are associated with unknowns belonging to element K.

Now, we are in the position to define in an exact mathematical way the action of the smoothing operator $S : \mathbb{R}^N \to \mathbb{R}^N$ applied to an approximation $u^0 \in \mathbb{R}^N$ of the solution u of the linear system (69). We assume a certain numbering of mesh cells, i.e., let $\mathcal{T}_h = \{K_i,\ i = 1, \ldots, \mathrm{NEL}\}$. Then, we define $Su^0 := u^{\mathrm{NEL}}$, where u^{NEL} is successively computed by means of the following iteration over all elements K_i of the mesh:

$$u^i = u^{i-1} + \omega E_{K_i} \left(A_{K_i}\right)^{-1} R_{K_i} \left(f - A u^{i-1}\right) \qquad \forall\, i = 1, \ldots, \mathrm{NEL}. \tag{73}$$

Here, $\omega > 0$ is a relaxation parameter and is set to $\omega = 1$ in our numerical tests. The computation of the local correction vector $v_K = (A_K)^{-1} d_K$ in (73) is implemented by applying an optimized direct solver (e.g., from the BLAS routines) to the local system $A_K v_K = d_K$. Thus, the high performance of modern hardware architectures can be fully exploited within the smoothing steps. Concerning the size of the local problems, we have on each element K, in case of the used conforming LBB-stable Q_2/P_1^{disc} element pair, 18 degrees of freedom for the velocity and 3 for the pressure per required time point on the current time interval. Therefore, the dimension $\hat{n}$ of the local system is 21 if the cGP(1) or Crank-Nicolson method is applied and 42 in the case of the cGP(2) or dG(1)-method. Figure 1 shows the location of the local degrees of freedom of the Q_2/P_1^{disc} element pair on the reference element for the time discretizations cGP(1) and cGP(2).



Q_2/P_1^{disc}

$$\text{cGP(1)} = \begin{cases} \circ & \underline{U}_n^1, \underline{V}_n^1 \\ \square & \underline{P}_n^1, \frac{\partial \underline{P}_n^1}{\partial x}, \frac{\partial \underline{P}_n^1}{\partial y} \end{cases}$$

$$\text{cGP(2)} = \begin{cases} \circ & \underline{U}_n^1, \underline{V}_n^1, \underline{U}_n^2, \underline{V}_n^2 \\ \square & \underline{P}_n^1, \frac{\partial \underline{P}_n^1}{\partial x}, \frac{\partial \underline{P}_n^1}{\partial y}, \underline{P}_n^2, \frac{\partial \underline{P}_n^2}{\partial x}, \frac{\partial \underline{P}_n^2}{\partial y} \end{cases}$$

Figure 1: Location of the degrees of freedom for the Q_2/P_1^{disc} element pair.

5.3. Restriction and prolongation

In the following, we describe in more detail the prolongation and restriction operator $I_{k-1}^k : \mathbb{R}^{N_{k-1}} \to \mathbb{R}^{N_k}$ and $I_k^{k-1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k-1}}$, respectively, which transfer information between the

coarse grid level $k-1$ and the fine level k in Algorithm 1. We use the so-called canonical grid transfer routines based on the FEM space which treat all solution components separately. In the case of conforming Q_2 finite elements, the prolongation operator is constructed by using a biquadratic interpolation. Let v_i, $i = 1, \ldots, 9$, denote the nodal values of a velocity component on a coarse grid cell K_H of level $k-1$ and w_i the corresponding values on a refined son-element $K_h \subset K_H$ of grid level k as shown in Figure 2 (left) (see [15, 16] for the details). Then, the nodal values w_1, w_2 and w_9 are computed as (see also Figure 2 (left))

$$w_1 := v_1, \qquad w_2 := \frac{1}{8}(3v_1 + 6v_2 - v_3), \tag{74}$$

$$w_9 := \frac{1}{64}(9v_1 + 18v_2 - 3v_3 - 6v4 + v5 - 6v_6 - 3v_7 + 18v_8 + 36v_9). \tag{75}$$

The remaining nodal values w_i are computed similarly. The restriction is then set up as the adjoint of the prolongation operator, i.e., the matrices associated to I_{k-1}^k and I_k^{k-1} are exactly transposed to each other. Next, we describe the construction of the prolongation operator for

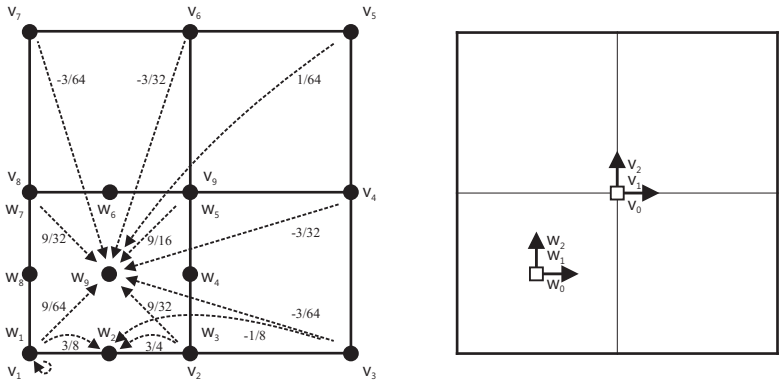


Figure 2: Prolongation for Q_2 with biquadratic interpolation (left) and P_1 with linear interpolation (right).

the discontinuous P_1 element approximating the pressure which is done by linear interpolation. To this end, we suppose that v_0, v_1, v_2 denote the 3 nodal values (consisting of the value of the function and its derivatives w.r.t. x and y, respectively, at the barycenter) that determine the P_1-pressure function on a coarse grid cell K_H at mesh level $k-1$. Furthermore, let w_0, w_1, w_2 denote the corresponding values for a son-cell $K_h \subset K_H$ on the fine grid level k as shown in Figure 2 (right). Then, the nodal values w_i are computed as

$$w_0 := v_0 - \frac{1}{2}v_1 - \frac{1}{2}v_2, \qquad w_1 := v_1 \qquad w_2 := v_2. \tag{76}$$

Here, the values of w_1, w_2 will remain the same v_1, v_2, respectively, since the derivative of linear function is always constant. Again, the restriction operator is defined as the adjoint of the prolongation operator.

6. Numerical Results

In [2], we performed nonstationary simulations for two special flow configurations, namely, *flow around cylinder* and *flow through a Venturi pipe*, to demonstrate the temporal accuracy

and efficiency of the presented higher order time discretization schemes for the incompressible Navier-Stokes equations. Thus, we have continued the work started in [17], where different time stepping schemes were analyzed for these classes of problems. The test problem *flow around cylinder* corresponds to the classical benchmark in [18]. Here, the main difficulty is to compute the right nonstationary behavior of the flow pattern with periodic oscillations and examine the ability of different time discretization schemes to capture the dynamics of the flow. As a second test case, we consider the nonstationary flow for a high Reynolds number through a Venturi pipe which has many real life and industrial applications. In this section, we mainly analyze the behavior of the nonlinear and linear solvers for the presented problems.

6.1. Accuracy Results

In this subsection, we demonstrate the accuracy and the numerical cost of our proposed higher order space-time discretization scheme cGP(2)-Q_2/P_1^{disc}. To this end, we consider the Stokes problem on the domain $\Omega := (0,1)^2$ and with $\nu = 1$. The prescribed velocity field $\mathbf{u} = (u_1, u_2)$ is

$$\begin{aligned} u_1(x,y,t) &:= x^2(1-x)^2\left[2y(1-y)^2 - 2y^2(1-y)\right]\sin(10\pi t), \\ u_2(x,y,t) &:= -\left[2x(1-x)^2 - 2x^2(1-x)\right]y^2(1-y)^2\sin(10\pi t), \end{aligned}$$

and the pressure distribution is

$$p(x,y,t) := -(x^3 + y^3 - 0.5)(1.5 + 0.5\sin(10\pi t)).$$

The initial data is $\mathbf{u}_0(x,y) = \mathbf{u}(x,y,0)$.

We apply the time discretization scheme cGP(2) with an equidistant time step size $\tau = T/N$. To measure the error (in time), we use the standard L^2-norm $\|\cdot\|_2 := \|\cdot\|_{L^2(I,L^2(\Omega))}$ of the time discretization error $u(t) - u_{h,\tau}(t)$ for the velocity over the time interval $I = [0,1]$. Here, the error consists of both temporal and spatial discretization errors due to the chosen finite element space as none of the finite element spaces fully captures the corresponding solutions. In order to analyze the errors explicitly, we first consider the fixed space mesh size h_ℓ and reduce the time step size, respectively, the time step size τ is fixed and we reduce the spatial mesh size. We present in Table 1, 2 and 3, for different space mesh levels, the full discretization error $u - u_{h,\tau}$ of the velocity in the L^2-norm and the total number "#DOFs" together with the required CPU-time. The CPU-times are measured in seconds for the nonlinear solver implemented within the solver package FEAT2 (www.featflow.de) and the simulations are performed on a Dual-Core AMD Opteron 8220 with eight CPUs at 2.8GHz.

From Table 1 to 3, we observe that for a fixed mesh size $h_\ell = 2^{-(\ell-1)}$ and $\tau \to 0$, the space error becomes dominant for sufficiently small time step sizes $\tau < \tau_0(h_\ell)$. We indicate by means of an underline that row in the column block of grid level ℓ which corresponds to the last suitable time step size $\tau_0(h_\ell)$.

We see that in the standard L^2-norm $\|\cdot\|_{2,L}$ the error of the cGP(2)-method behaves, for fixed mesh size h_ℓ, like $\mathcal{O}(\tau^3)$ as long as $\tau \geq \tau_0(h_\ell)$ whereas it starts to stagnate for $\tau < \tau_0(h_\ell)$. If we look at the error norms for the sequence of space-time meshes with $(\tau, h) = (\tau_0(h_\ell), h_\ell)$, $\ell = 4, 6, 8$, we observe that the error decreases by a factor of about 8 if we increase the level ℓ by one. This indicates an asymptotic behaviour of the form

$$\|u - u_{\tau,h}\|_L \leq C(\tau^3 + h^3)$$

ℓ	$1/\tau$	$\|u - u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
4	10	4.40E-04		15400		14.58	
4	20	1.12E-04	3.94	30800	2.00	24.47	1.68
4	40	1.99E-05	5.63	61600	2.00	45.73	1.87
4	80	1.51E-05	1.31	123200	2.00	85.66	1.87
4	160	1.51E-05	1.00	246400	2.00	167.38	1.95
4	320	1.51E-05	1.00	492800	2.00	326.81	1.95

Table 1: Full discretization error for fixed space mesh level ($\ell = 4$), total number of unknowns and CPU-time for the cGP(2)-Q_2/P_1^{disc} scheme.

ℓ	$1/\tau$	$\|u - u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
6	10	4.40E-04		230440		241.84	
6	20	1.11E-04	3.97	460880	2.00	424.03	1.75
6	40	1.31E-05	8.49	921760	2.00	796.51	1.88
6	80	1.62E-06	8.07	1843520	2.00	1509.06	1.89
6	160	3.11E-07	5.21	3687040	2.00	2875.87	1.91
6	320	2.39E-07	1.30	7374080	2.00	5414.28	1.88
6	640	2.37E-07	1.01	14748160	2.00	10760.68	1.99

Table 2: Full discretization error for fixed space mesh level ($\ell = 6$), total number of unknowns and CPU-time for the cGP(2)-Q_2/P_1^{disc} scheme.

where $\|\cdot\|_L$ stands for $\|\cdot\|_{2,L}$. This asymptotic behaviour is optimal with respect to the quadratic polynomial ansatz of the cGP(2)-method in time and the quadratic ansatz of the Q_2-element in space.

ℓ	$1/\tau$	$\|u - u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
8	10	4.40E-04		3625000		3576.66	
8	20	1.11E-04	3.97	7250000	2.00	6040.72	1.69
8	40	1.31E-05	8.49	14500000	2.00	11217.90	1.86
8	80	1.60E-06	8.15	29000000	2.00	22530.31	2.01
8	160	2.01E-07	7.97	58000000	2.00	44870.56	1.99
8	320	2.53E-08	7.93	116000000	2.00	29567.10	1.46
8	640	4.85E-09	5.22	232000000	2.00	159293.34	1.77
8	1280	3.73E-09	1.30	464000000	2.00	272482.16	1.71

Table 3: Full discretization error for fixed space mesh level ($\ell = 8$), total number of unknowns and CPU-time for the cGP(2)-Q_2/P_1^{disc} scheme.

Next, we repeat the tests for the different space mesh levels with fixed time step size. Table 4 and 5 show the results for the fixed time step size $\tau = 1/40$ and $\tau = 1/160$, respectively.

ℓ	$1/\tau$	$\|u - u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
2	40	8.33E-04		4960		2.84	
3	40	1.18E-04	7.07	16800	3.39	11.25	3.96
4	40	1.99E-05	5.94	61600	3.67	45.73	4.06
5	40	1.32E-05	1.51	235680	3.83	189.67	4.15
6	40	1.31E-05	1.01	921760	3.91	796.51	4.20
7	40	1.31E-05	1.00	3645600	3.96	2806.51	3.52
8	40	1.31E-05	1.00	14500000	3.98	11217.90	4.00
9	40	1.31E-05	1.00	57835680	3.99	43985.14	3.92

Table 4: Full discretization error for fixed time step size ($\tau = 1/40$), total number of unknowns and CPU-time for the cGP(2)-Q_2/P_1^{disc} scheme.

ℓ	$1/\tau$	$\|u - u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
2	160	8.37E-04		19840		7.73	
3	160	1.18E-04	7.11	67200	3.39	39.29	5.08
4	160	1.51E-05	7.81	246400	3.67	167.38	4.26
5	160	1.91E-06	7.91	942720	3.83	682.75	4.08
6	160	3.11E-07	6.13	3687040	3.91	2875.87	4.21
7	160	2.03E-07	1.53	14582400	3.96	10993.84	3.82
8	160	2.01E-07	1.01	58000000	3.98	44870.56	4.08
9	160	2.01E-07	1.00	231342720	3.99	169389.32	3.78

Table 5: Full discretization error for fixed time step size ($\tau = 1/160$), total number of unknowns and CPU-time for the cGP(2)-Q_2/P_1^{disc} scheme.

From Table 4, we see that for a fixed time step size $\tau = 1/40$ and $h_\ell = 2^{-(\ell-1)} \to 0$, the time error becomes dominant for sufficiently small mesh sizes. Table 5 shows a similar behavior for $\tau = 1/160$. We indicate again by means of an underline that row in the column block of grid level ℓ which corresponds to the last suitable mesh size.

Next, we present in Table 6 the full discretization error $u - u_{h,\tau}$ of the velocity for the corresponding method in the L^2-norm, the total number "#DOFs" of all unknowns occurring on the space mesh (i.e., on each time interval) and the required CPU-time in seconds where the mesh and time step size have been chosen as

$$h = 2^{-(\ell-1)}, \qquad \tau = \frac{1}{5}\, 2^{-(\ell-2)}), \qquad \ell = 2, 3, \ldots.$$

It can be seen from Table 6 that the full (space-time) discretization error is reduced by a factor of 8 if we increase the level ℓ by one (leading to eight times more space-time unknowns), whereas the numerical cost in terms of the CPU-time increase only by a factor of 8, too, which due to the optimal complexity of the multigrid solver is a big advantage for this higher order method.

ℓ	$1/\tau$	$\|u-u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	# MG	CPU	Factor
4	20	1.12E-04		30800		5.0	24.47	
5	40	1.32E-05	8.48	235680	7.65	6.0	189.39	7.74
6	80	1.62E-06	8.15	1843520	7.82	5.5	1509.06	7.97
7	160	2.03E-07	7.97	14582400	7.91	4.5	11526.30	7.64
8	320	2.53E-08	8.02	116000000	7.95	6.0	89967.44	7.81

Table 6: Full discretization error, total number of unknowns, number of multigrid iterations and CPU-time for the cGP(2)-Q_2/P_1^{disc} scheme.

6.2. Nonstationary flow around cylinder

The flow configuration related to the 'flow around cylinder' configuration [18], which is considered here, can be found at `www.featflow.de/en/benchmarks/cfdbenchmarking.html`. The examined accuracy of the benchmark crucially depends on the following quantities

$$F_D = \int_S (\rho\nu \frac{\partial u_t}{\partial n} n_y - p n_x) dS, \quad \text{and particularly} \quad F_L = -\int_S (\rho\nu \frac{\partial u_t}{\partial n} n_x + p n_y) dS$$

representing the total forces in the horizontal and vertical directions, respectively. Figure 3

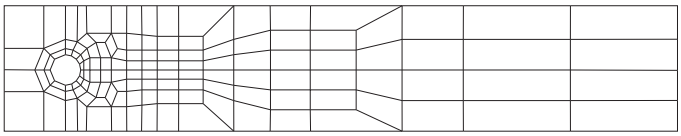


Figure 3: Coarse mesh for *flow around cylinder*.

Lev.	#EL	#DOF(total)
2	520	5 928
3	2 080	23 296
4	8 320	92 352

Figure 4: Size of the different systems in space.

shows the initial coarse mesh (level 1), which will be uniformly refined, and Figure 4 presents for different space mesh levels the number '#EL' of elements and the total number '#DOF' of all space degrees of freedom which are needed to represent the discrete velocity and pressure solution based on the Q_2/P_1^{disc} element pair at one fixed time point.

In order to compare the accuracy of the higher order time discretization, the flow is started in each computation from the same developed solution at time t_0, and the simulation is performed until T=10 for various uniform time step sizes $\tau_n := \tau$. After T=10, all quantities of interest have been plotted and analyzed in detail in [2]. In Table 7, we recall from [2], for different given space levels and time discretization schemes, the maximum allowed time step sizes that guarantee comparable accurate results with an error of approx. 0.3% per time period at a given space level (see [2] for the details). We can see that the corresponding time step size required for the cGP(2)-method is 1.5 times larger than for the dG(1)-method and 5 times larger than for the cGP(1)-method. The numerical costs for the cGP(2)-method per time step are nearly the same as for the dG(1)-method. So, we can roughly expect that the cGP(2)-method is 1.5 times faster than the dG(1)-method to gain a certain accuracy if the corresponding solvers show a convergence behaviour which is more or less independent of the time step size. Concerning the comparison to the cGP(1)-method, it depends on the question how much more computing time is needed to solve the (nonlinear) 6x6 block systems for one time step of the cGP(2)-method compared to the time for solving the (nonlinear) 3x3 block systems for the cGP(1)-method. In the following tables

Lev	**cGP(1)**	**dG(1)**	**cGP(2)**
2	1/100	1/30	1/20
3	1/100	1/30	1/20
4	1/100	1/30	1/20
factor	5	1.5	1

Table 7: Maximum allowed time step sizes with deviation of approx. 0.3% per time period.

which demonstrate the associated solution behaviour, we always underline, for each column of a time discretization method, the one element which corresponds to the maximal possible time step size that is necessary to ensure the required accuracy with respect to Table 7.

In order to measure and compare the efficiency of the nonlinear solvers for the presented time discretization schemes, we show the averaged number '#NL' of nonlinear iterations per time step for the fixed-point and Newton method for different (space) mesh levels. We stop the nonlinear iteration if the L^2-norm of the nonlinear residual drops down below 10^{-10}. Table 8 shows that, for

τ	Fixed-point			Newton		
	cGP(1) #NL	**dG(1)** #NL	**cGP(2)** #NL	**cGP(1)** #NL	**dG(1)** #NL	**cGP(2)** #NL
1/10	11.82	13.18	11.18	4.18	4.73	5.00
1/15	9.00	9.88	8.62	3.94	4.06	4.06
1/20	8.00	8.24	<u>7.14</u>	3.81	4.00	<u>4.00</u>
1/25	7.00	7.04	7.00	3.04	3.69	4.00
1/30	6.03	<u>7.00</u>	6.03	3.03	<u>3.03</u>	3.03
1/50	5.02	5.02	5.02	3.00	3.02	3.02
1/100	<u>4.01</u>	4.01	4.01	<u>2.01</u>	2.01	2.01
1/200	3.00	3.00	3.00	2.00	2.00	2.00

Table 8: Averaged number of nonlinear iterations per time step at space level=3.

the fixed-point iteration as well as for the Newton method, almost the same number of nonlinear iterations are required for the different time discretization schemes.

Moreover, as expected, the number of iterations decreases only slightly if we reduce the time step size. Furthermore, the Newton method converges 2-3 time faster as compared to the fixed-point method due to its superlinear convergence. Table 9 and 10 demonstrate the same behavior for the space level 4 and 5, respectively.

Summarizing, the number of nonlinear steps associated with the maximum allowed time step sizes to gain the accuracy with an error of approx. 0.3% per time period for the cGP(2) and dG(1)-method (see Table 7) require only approx. 2 times more than the cGP(1)-method, respectively. Moreover, this factor is eventually improved by using the Newton method.

Next, we analyze the behavior of the multigrid solver for the solution of the linear subproblems within the different time discretization schemes. To this end, we present the averaged number '#MG' of multigrid iterations per nonlinear step. Here, our multigrid solver uses for smoothing $\nu_1 = \nu_2 = 4$ pre- and post-smoothing steps of a preconditioned GMRES method where the

τ	Fixed-point			Newton		
	cGP(1) #NL	**dG(1)** #NL	**cGP(2)** #NL	**cGP(1)** #NL	**dG(1)** #NL	**cGP(2)** #NL
1/10	10.09	11.64	10.18	4.00	4.09	4.91
1/15	8.00	8.69	7.94	3.94	4.06	4.00
1/20	7.00	7.05	7.00	3.05	4.00	4.00
1/25	6.04	6.04	6.00	3.04	3.04	3.04
1/30	5.65	6.00	5.48	3.03	3.03	3.03
1/50	4.02	5.00	4.24	3.00	3.00	3.00
1/100	3.01	4.00	3.98	2.01	2.01	2.01
1/200	3.00	3.00	3.00	2.00	2.00	2.00

Table 9: Averaged number of nonlinear iterations per time step at space level=4.

τ	Fixed-point			Newton		
	cGP(1) #NL	**dG(1)** #NL	**cGP(2)** #NL	**cGP(1)** #NL	**dG(1)** #NL	**cGP(2)** #NL
1/10	9.00	10.27	9.00	4.00	4.09	4.55
1/15	7.00	7.69	7.00	3.56	4.00	4.00
1/20	6.00	6.05	6.00	3.05	3.38	3.90
1/25	5.04	6.00	5.04	3.04	3.04	3.04
1/30	5.00	5.03	5.00	3.00	3.03	3.03
1/50	4.02	4.02	4.02	2.02	3.00	3.00
1/100	3.01	3.01	3.01	2.01	2.01	2.01
1/200	2.00	3.00	3.00	2.00	2.00	2.00

Table 10: Averaged number of nonlinear iterations per time step at space level=5.

preconditioner is the application of one step of the operator S_k ("Vanka") on the actual grid level k explained in Section 5.2. The multigrid solver stops if the L^2-norm of the relative residual is smaller than 10^{-6} or the absolute residual drops down below 10^{-15}.

τ	Fixed-point-multigrid			Newton-multigrid		
	cGP(1) #MG	**dG(1)** #MG	**cGP(2)** #MG	**cGP(1)** #MG	**dG(1)** #MG	**cGP(2)** #MG
1/10	11.92	11.75	10.91	10.80	10.80	10.60
1/15	11.67	11.40	10.89	11.25	10.80	10.20
1/20	11.62	11.11	11.00	11.25	11.25	10.75
1/25	11.71	11.38	11.43	11.25	11.00	11.00
1/30	11.71	11.86	11.57	11.00	11.00	11.00
1/50	12.17	12.00	11.83	11.67	11.25	11.00
1/100	12.40	12.00	11.60	11.67	11.67	11.00
1/200	12.25	12.00	12.00	11.67	11.33	11.00

Table 11: Averaged number of multigrid iterations per nonlinear step at space level=3.

τ	Fixed-point-multigrid			Newton-multigrid		
	cGP(1) #MG	**dG(1)** #MG	**cGP(2)** #MG	**cGP(1)** #MG	**dG(1)** #MG	**cGP(2)** #MG
1/10	10.80	10.64	10.20	10.50	10.00	9.60
1/15	10.75	10.67	10.50	10.25	9.80	10.00
1/20	10.71	10.75	10.71	10.00	10.25	10.25
1/25	10.57	11.00	11.00	10.00	10.25	10.25
1/30	11.00	11.00	11.00	9.75	10.25	10.25
1/50	11.00	11.20	11.60	10.33	10.67	10.67
1/100	12.00	12.00	12.00	11.33	11.33	11.33
1/200	12.50	12.50	12.50	11.67	12.00	12.00

Table 12: Averaged number of multigrid iterations per nonlinear step at space level=4.

From the Table 11 to 13, we see that the multigrid solver requires almost the same number of iterations for the different presented time discretization schemes. Moreover, the number of multigrid iterations remains fairly constant if we increase the refinement level of the space mesh. There is also no noticeable increase in the number of iterations if we decrease the time step which is due to the incompressibility constraint. This means that the behavior of the multigrid solver is very robust and almost independent of the mesh size, the time step size and the used time discretization method.

Finally, we summarize in Table 14 the corresponding numerical costs that are necessary for each method in order to obtain the accuracy with an error of approx. 0.3% per time period as described in Table 7. These numerical costs include the time step size, the averaged number #NL of nonlinear steps per time step, the averaged number #MG of multigrid iterations per nonlinear iteration and the total CPU-time in seconds.

τ	Fixed-point-multigrid			Newton-multigrid		
	cGP(1) #MG	**dG(1)** #MG	**cGP(2)** #MG	**cGP(1)** #MG	**dG(1)** #MG	**cGP(2)** #MG
1/10	9.56	9.36	9.11	9.79	9.20	9.00
1/15	10.14	9.75	9.43	9.50	9.50	9.25
1/20	10.00	9.86	9.67	9.50	9.50	9.25
1/25	10.00	10.00	9.50	9.25	9.50	9.25
1/30	10.20	10.00	9.80	9.67	9.50	9.00
1/50	10.00	10.00	10.00	9.67	9.67	9.67
1/100	9.75	10.50	11.00	9.67	10.00	10.33
1/200	11.00	11.67	12.00	9.50	11.00	11.33

Table 13: Averaged number of multigrid iterations per nonlinear step at space level=5.

quantity	Fixed-point			Newton		
	cGP(1)	**dG(1)**	**cGP(2)**	**cGP(1)**	**dG(1)**	**cGP(2)**
$\triangle t$	1/100	1/30	1/20	1/100	1/30	1/20
averaged #NL	3.0	5.0	6.0	2.0	3.0	3.9
averaged #MG	9.7	9.8	9.6	9.6	9.5	9.2
total CPU-time	582790	729574	464168	407315	464307	356868

Table 14: Summary of numerical costs at space level=5 which are necessary for cGP(1), dG(1) and cGP(2) to achieve the comparable accuracy (0.3% per period).

6.3. Nonstationary flow through a Venturi pipe

The test configuration for the flow through a Venturi pipe which is considered in this section, has been already used in [11, 17] (see [2] for more details). The aim of this simulation is to control the flux through the upper channel. Beside this interesting flow quantity, we have also compared the accuracy of all the presented time discretization schemes by computing the velocity and the pressure at various points (see [2]). Figure 6 gives an overview of the size of the problem on different space mesh levels. Since we have a high Reynolds number flow here, we employ as a stabilization method the edge oriented jump FEM approach (see [10] and Remark 1) with the parameters $\gamma = 0.1$ and $\gamma^* = 0.0$. In order to compare the accuracy of different time

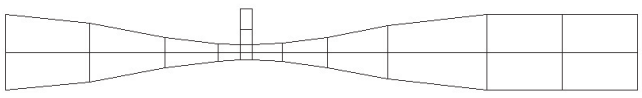


Figure 5: Coarse mesh for the Venturi pipe flow

Lev.	#EL	#DOF(total)
3	384	4 466
4	1 536	17 378
5	6 144	68 546
6	24 576	272 258

Figure 6: Size of the different systems in space.

discretizations, the flow is started on each mesh level from the corresponding Stokes solution at time $t = 0$, and the simulation is performed until T=30 using different time discretization methods for different time step sizes τ. After T=30, all the quantities of interest have been

plotted and analyzed in detail in [2]. A detailed analysis regarding the temporal accuracy in [2] has shown that the cGP(2)-method captures the dynamics of the flow at quite large time step sizes as expected. The results on different mesh levels look somewhat more different due to the higher Reynolds number. As in the test case before, we have determined in the same way the maximum allowed time step sizes which lead to very similar results in the "picture norm". Table 15 shows these time step sizes for different space mesh levels (see again [2] for more details).

Lev	**cGP(1)**	**dG(1)**	**cGP(2)**
3	1/50	1/20	1/6
4	1/50	1/20	1/8
5	1/100	1/20	1/15
factor	≈ 6	$\approx$1.5	1

Table 15: Maximum allowed time step sizes which lead (almost) to same results (from [2]).

Now we analyze the solver for the Venturi pipe flow similar to the example "flow around cylinder". Again, we show the averaged number of nonlinear iterations per time step for the fixed-point and Newton method for different space mesh levels. We apply the same stopping criterion as in the previous example for the nonlinear and linear solver, respectively,

τ	Fixed-point			Newton		
	cGP(1) #NL	**dG(1)** #NL	**cGP(2)** #NL	**cGP(1)** #NL	**dG(1)** #NL	**cGP(2)** #NL
1/10	10.36	10.91	9.64	3.09	3.36	3.45
1/15	8.56	9.19	8.50	3.06	3.06	3.06
1/20	7.67	8.14	7.62	3.05	3.05	3.05
1/25	7.04	7.58	7.00	2.77	3.04	3.04
1/50	5.80	5.90	5.75	2.08	2.53	2.55
1/100	4.72	4.88	4.77	2.01	2.01	2.01
1/200	4.00	4.00	4.00	2.00	2.00	2.00

Table 16: Averaged number of nonlinear iterations per time step at space level=3.

From Table 16 to 18, we see again that, for the fixed-point iteration as well as for the Newton method, almost the same number of iterations is required for every time discretization scheme. Moreover, for a fixed space mesh level, the number of nonlinear iterations slightly decreases if the time step size is reduced, as expected. Concerning the number of nonlinear iterations, it is noticed that the Newton method is more efficient than the fixed point iteration which shows almost the same behavior as in case of the flow around cylinder example. These results indicate that the nonlinear solver is also robust with respect to the underlying flow configuration.

Next, we analyze the behavior of the multigrid solver for the solution of the linear subproblems. Here, the multigrid solver is using the same settings as in the example with the flow around a cylinder. In Table 19 to 21, we present the averaged number of multigrid iterations per nonlinear step for solving the corresponding linear block systems for the space mesh levels 3-5. We observe again that the multigrid solver requires almost the same number of iterations for the

τ	Fixed-point			Newton		
	cGP(1) #NL	**dG(1)** #NL	**cGP(2)** #NL	**cGP(1)** #NL	**dG(1)** #NL	**cGP(2)** #NL
1/10	10.27	11.09	9.91	3.27	3.73	3.73
1/15	8.56	9.25	8.75	3.06	3.06	3.06
1/20	7.76	8.52	7.95	3.05	3.05	3.05
1/25	7.12	7.73	7.31	3.04	3.04	3.04
1/50	5.63	5.80	5.57	2.12	2.47	2.49
1/100	4.33	4.50	4.38	2.01	2.01	2.01
1/200	3.60	3.81	3.76	2.00	2.00	2.00

Table 17: Averaged number of nonlinear iterations per time step at space level=4.

τ	Fixed-point			Newton		
	cGP(1) #NL	**dG(1)** #NL	**cGP(2)** #NL	**cGP(1)** #NL	**dG(1)** #NL	**cGP(2)** #NL
1/10	9.64	10.64	9.73	3.27	3.73	3.73
1/15	8.31	9.00	8.62	3.06	3.06	3.06
1/20	7.52	8.24	7.76	3.05	3.05	3.05
1/25	6.96	7.58	7.23	3.04	3.04	3.04
1/50	5.39	5.55	5.41	2.06	2.24	2.25
1/100	4.01	4.24	4.01	2.01	2.01	2.01
1/200	3.00	3.27	3.16	2.00	2.00	2.00

Table 18: Averaged number of nonlinear iterations per time step at space level=5.

τ	Fixed-point-multigrid			Newton-multigrid		
	cGP(1) #MG	**dG(1)** #MG	**cGP(2)** #MG	**cGP(1)** #MG	**dG(1)** #MG	**cGP(2)** #MG
1/10	9.45	10.45	10.80	10.00	11.25	11.50
1/15	9.89	10.70	10.33	10.25	11.00	11.00
1/20	9.89	10.56	10.38	10.25	11.00	11.00
1/25	10.38	10.38	10.00	10.75	11.00	11.00
1/50	9.83	10.50	10.33	10.67	10.25	10.50
1/100	10.20	10.60	10.20	10.67	10.67	10.33
1/200	10.40	10.40	10.00	10.67	10.67	10.33

Table 19: Averaged number of multigrid iterations per nonlinear step at space level=3.

τ	Fixed-point-multigrid			Newton-multigrid		
	cGP(1) #MG	**dG(1)** #MG	**cGP(2)** #MG	**cGP(1)** #MG	**dG(1)** #MG	**cGP(2)** #MG
1/10	9.50	9.91	10.00	10.50	10.00	10.40
1/15	10.11	10.30	10.33	10.00	11.25	11.00
1/20	10.25	10.56	10.38	10.50	11.50	10.75
1/25	10.12	10.50	10.43	10.75	11.25	11.00
1/50	10.83	10.50	10.17	11.00	10.50	10.25
1/100	11.00	10.80	11.20	11.00	11.33	11.00
1/200	10.80	10.20	10.40	11.00	11.00	10.67

Table 20: Averaged number of multigrid iterations per nonlinear step at space level=4.

τ	Fixed-point-multigrid			Newton-multigrid		
	cGP(1) #MG	**dG(1)** #MG	**cGP(2)** #MG	**cGP(1)** #MG	**dG(1)** #MG	**cGP(2)** #MG
1/10	10.67	11.20	10.20	12.00	11.80	10.60
1/15	10.38	11.11	10.62	11.00	12.00	11.75
1/20	10.00	11.38	11.29	10.75	12.00	11.50
1/25	10.43	12.00	11.14	10.75	12.00	12.00
1/50	11.00	12.17	11.17	11.33	12.00	11.00
1/100	11.80	11.20	11.00	11.00	11.33	10.67
1/200	12.25	11.50	11.25	12.00	11.00	11.00

Table 21: Averaged number of multigrid iterations per nonlinear step at space level=5.

presented time discretization schemes. Moreover, the number of multigrid iterations remains almost constant for increasing space mesh level. There is also no noticeable increase in the number of iterations if we decrease the time step. Comparing with the results in the flow around cylinder example we observe a similar solver behavior as for the Venturi pipe flow. This indicates that our multigrid solver is also robust with respect to the underlying flow configurations. At the end, similar to the flow around cylinder example, we summarize in Table 22 the corresponding computational costs in terms of CPU time that are needed for each method to obtain the required accuracy as described in Table 15.

quantity	Fixed-point			Newton		
	cGP(1)	**dG(1)**	**cGP(2)**	**cGP(1)**	**dG(1)**	**cGP(2)**
$\triangle t$	1/100	1/20	1/15	1/100	1/20	1/15
averaged #NL	4.0	8.2	8.6	2.0	3.0	3.0
averaged #MG	11.8	11.3	10.6	11.0	12.0	11.7
total CPU-time	1501952	1645490	1153457	724233	633672	433118

Table 22: Summary of numerical costs at space level=5 which are necessary for cGP(1), dG(1) and cGP(2) to achieve the required accuracy.

7. Conclusion

We have presented the details of efficient solution techniques for solving the nonlinear block-systems resulting from the *continuous* Galerkin-Petrov and *discontinuous* Galerkin time discretization schemes of higher order for the nonstationary incompressible Navier-Stokes equations (in 2D). The spatial discretization is carried out by using biquadratic finite elements for velocity and discontinuous linear elements for the pressure on a (general) quadrilateral mesh. The resulting discretized block-systems of nonlinear equations which can be characterized as saddle point problems are treated by using the fixed-point iteration or particularly the Newton method as outer iteration. The associated linear subproblems are solved by means of a monolithic multigrid method with a GMRES smoother which is preconditioned by an elementwise block Gauß-Seidel (Vanka-like) iteration where, for each element cell, simultaneously all unknowns are updated that belong to this cell.

We have analyzed the behaviour of the nonlinear and linear solvers for the presented time discretization schemes cGP(1), cGP(2) and dG(1) applied to two prototypical CFD test problems. The numerical results have shown that, for both nonlinear solution variants, almost the same number of nonlinear iterations is required for the different time discretization schemes. Moreover, the number of nonlinear iterations decreases slightly (for the Newton method) by reducing the time step size as expected. Furthermore, the Newton method converges 2-3 times faster compared to the fixed-point iteration due to its superlinear convergence behaviour. In an analogous way, we have also analyzed the behaviour of the proposed multigrid method for solving the linear subproblems arising from the different time discretization schemes. The results show that the multigrid convergence is almost independent of the spatial mesh size (as expected) and of the time step (due to the incompressibility). So, combining the superlinear behaviour of the Newton method with the robust and efficient solution behaviour of the multigrid solvers, the resulting total computational costs to achieve a certain accuracy are essentially smaller for the higher order

time discretizations compared to the classical Crank-Nicolson scheme (which is nearly equivalent to cGP(1)), leading in 2D to an almost optimal computational behaviour since (uniform) grid refinement in space and time leads not only to 8 times more unknowns, but also to 8 times higher computational costs while at the same time also 8 times more accurate solutions can be obtained.

References

[1] V. Thomée, Galerkin finite element methods for parabolic problems, 2nd Edition, Vol. 25 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2006.

[2] S. Hussain, F. Schieweck, S. Turek, An efficient and stable finite element solver of higher order in space and time for nonstationary incompressible flow, International Journal for Numerical Methods in Fluids, published: Doi: 10.1002/fld.

[3] S. Hussain, F. Schieweck, S. Turek, A note on accurate and efficient higher order Galerkin time stepping schemes for the nonstationary Stokes equations, The Open Numerical Methods Journal 4 (2012) 35–45.

[4] S. Hussain, F. Schieweck, S. Turek, Higher order Galerkin time discretizations and fast multigrid solvers for the heat equation, Journal of Numerical Mathematics 19 (1) (2011) 41–61.

[5] J. Heywood, R. Rannacher, S. Turek, Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations, International Journal for Numerical Methods in Fluids 22 (1996) 325–352.

[6] F. Schieweck, A-stable discontinuous Galerkin-Petrov time discretization of higher order, J. Numer. Math. 18 (1) (2010) 25 – 57.

[7] C. Cuvelier, A. Segal, A. A. Steenhoven, Finite element methods and Navier-Stokes equations, D. Reidel Publishing Company, 1986.

[8] C. Johnson, Numerical Solution of Partial Differential Equations by the Finite Element Method, Cambridge University Press New York, 1988.

[9] J. Donea, A. Huerta, Finite element methods for flow problems, JohnWiley and Sons, New York, 2003.

[10] S. Turek, A. Ouazzi, Unified edge–oriented stabilization of nonconforming FEM for incompressible flow problems: Numerical investigations, Journal of Numerical Mathematics 15 (4) (2007) 299–322.

[11] S. Turek, Efficient solvers for incompressible flow problems. An algorithmic and computational approach, Vol. 6 of Lecture Notes in Computational Science and Engineering, Springer, 1999.

[12] H. Wobker, S. Turek, Numerical studies of Vanka-type smoothers in Computational Solid Mechanics, Advances in Applied Mathematics and Mechanics 1 (1) (2009) 29–55.

[13] S. P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables, Journal of Computational Physics 65 (1) (1986) 138 – 158.

[14] H. Damanik, J. Hron, A. Ouazzi, S. Turek, Monolithic Newton–multigrid solution techniques for incompressible nonlinear flow models, Int. J. Numer. Meth. Fluids, 2012 (accepted).

[15] M. Köster, S. Turek, The influence of higher order FEM discretisations on multigrid convergence, Computational Methods in Applied Mathematics 6 (2) (2006) 221–232.

[16] W. Hackbusch, Multi-Grid Methods and Applications., Springer-Verlag, Berlin, 1985.

[17] S. Turek, A comparative study of some time-stepping techniques for the incompressible navier-stokes equations: from fully implicit nonlinear schemes to semi-implicit projection methods, Int. J. Numer. Meth. Fluids 22 (1996) 987 – 1011.

[18] S. Turek, M. Schäfer, Benchmark computations of laminar flow around cylinder, in: E. Hirschel (Ed.), Flow Simulation with High-Performance Computers II, Vol. 52 of Notes on Numerical Fluid Mechanics, Vieweg, 1996, pp. 547–566.