A note on accurate and efficient higher order Galerkin time stepping schemes for the nonstationary Stokes equations

S. Hussain*, F. Schieweck†, S. Turek*

Abstract

In this note, we extend our recent work for the heat equation in [1] and compare numerically *continuous* Galerkin-Petrov (cGP) and *discontinuous* Galerkin (dG) time discretizations for the nonstationary Stokes equations in two dimensions. For the space discretization, we use the LBB-stable finite element pair Q_2/P_1^{disc} and we discuss implementation aspects as well as methods for solving the resulting block systems which are treated by using monolithic multigrid solvers with Vanka-type smoothers. By means of numerical experiments we compare the different time discretizations w.r.t. accuracy and computational costs and we show that the convergence behavior of the multigrid method is almost independent of mesh size and time step leading to an efficient solution process.

Keywords: Discontinuous Galerkin method, continuous Galerkin-Petrov method, Stokes equations, multigrid solver

2000 Mathematics Subject Classification (MSC): 65M12, 65M55, 65M60

1 The cGP- and dG-methods for the Stokes equations

We consider the nonstationary Stokes equations, i.e. we want to find a velocity $\mathbf{u} : \Omega \times [0,T] \to \mathbb{R}^2$ and a pressure $p : \Omega \times [0,T] \to \mathbb{R}$ *such that*

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

where ν denotes the viscosity, $f : \Omega \times (0,T) \to \mathbb{R}^2$ is the body force and $\mathbf{u}_0 : \Omega \to \mathbb{R}^2$ the initial velocity field at time $t = 0$. For simplicity, we assume homogeneous Dirichlet conditions at the boundary $\partial\Omega$ of a polygonal domain $\Omega \subset \mathbb{R}^2$. To make this problem well-posed, one needs to impose an additional condition on p, i.e., $\int_\Omega p d\Omega = 0$.

*Institut für Angewandte Mathematik, TU-Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany

†Institut für Analysis und Numerik, Otto-von-Guericke-Universität Magdeburg, Postfach 4120, D-39016 Magdeburg, Germany

The proposed time discretization schemes are accurate of higher order and have been studied for the heat equation in [1]. In particular, the cGP(1) and cGP(2)-method are accurate of order 2 and 3, respectively, in the whole time interval. Moreover, the cGP(2)-method shows a superconvergent behavior of order 4 in the discrete time points. The dG(1)-method is of order 2 in the whole time interval and superconvergent of order 3 in the discrete time points. The purpose of this paper is to perform numerical tests to illustrate that we can get for these time discretization schemes the same accuracy as for the heat equation also in the case of the Stokes equations. To this end, we only concentrate on the algorithmic and numerical aspects. A rigorous analysis regarding theoretical aspects will be subject of a forthcoming paper.

We start with the time discretization of problem (1) which is of variational type. In the following, let $I = [0,T]$ denote the time interval with some positive final time T. For a function $\mathbf{u} : \Omega \times I \to \mathbb{R}^2$ and a fixed $t \in I$, we will denote by $\mathbf{u}(t) := \mathbf{u}(\cdot,t)$ the associated velocity function at time t which is an element of a suitable function space $\mathbf{V}$. In case of the Stokes equations, this space is the Sobolev space $\mathbf{V} = (H_0^1(\Omega))^2$. Similarly, we denote by $p(t) := p(\cdot,t)$ the associated pressure function at time t which is an element of the function space $Q = L_0^2(\Omega)$ where

$$L_0^2(\Omega) = \Big\{ q \in L^2(\Omega) : \int_\Omega q\,dx = 0 \Big\}.$$

In the time discretization, we decompose the time interval I into N subintervals $I_n := [t_{n-1}, t_n)$, where $n = 1,\dots,N$ and $0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T$. The symbol τ will denote the *time discretization parameter* and will also be 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 cGP(k)-method, we approximate the solution $\mathbf{u} : I \to \mathbf{V}$ by means of a function $\mathbf{u}_\tau : I \to \mathbf{V}$ which is piecewise polynomial of order k with respect to time, i.e., we are looking for $\mathbf{u}_\tau$ in the discrete time space

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

where

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

We introduce the time discrete test space

$$\mathbf{Y}_\tau^{k-1} := \{ \mathbf{v} \in L^2(I,\mathbf{V}) : \ \mathbf{v}\big|_{I_n} \in \mathbb{P}_{k-1}(I_n,\mathbf{V}) \quad \forall\, n = 1,\dots,N \} \tag{3}$$

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 X_τ^k, where the vector valued space $\mathbf{V}$ is replaced by the scalar valued space Q, and an analogous discontinuous test space Y_τ^{k-1}.

Now, in order to derive the time discretization, we multiply the momentum equation in (1) with some suitable I_n-supported test functions $\mathbf{v}_\tau \in \mathbf{Y}_\tau^{k-1}$, integrate over $\Omega \times I_n$, use Fubini's Theorem and partial space integration of the terms $\triangle \mathbf{u}$ and ∇p and apply the k-point Gaussian quadrature rule for the evaluation of the time integrals. To determine $\mathbf{u}_\tau|_{I_n}$ and $p_\tau|_{I_n}$ we represent them by the polynomial ansatz

$$\mathbf{u}_\tau(t) := \sum_{j=0}^{k} \mathbf{U}_n^j \phi_{n,j}(t), \qquad p_\tau(t) := \sum_{j=0}^{k} P_n^j \phi_{n,j}(t), \tag{4}$$

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

$$\phi_{n,j}(t_{n,i}) = \delta_{i,j}, \qquad i,j = 0,\ldots,k \tag{5}$$

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

$$\mathbf{U}_n^0 = \mathbf{u}_\tau|_{I_{n-1}}(t_{n-1}) \quad \text{if} \quad n \geq 2 \qquad \text{or} \qquad \mathbf{U}_n^0 = \mathbf{u}_0 \quad \text{if} \quad n = 1. \tag{6}$$

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. This formula 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 (4) via the affine reference transformation $T_n : \hat{I} \to I_n$ where $\hat{I} := [-1,1]$ and

$$t = T_n(\hat{t}) := \frac{t_{n-1}+t_n}{2} + \frac{\tau_n}{2}\hat{t} \in I_n \qquad \forall\, \hat{t} \in \hat{I},\ n = 1,\ldots,N. \tag{7}$$

Let $\hat{\phi}_j \in \mathbb{P}_k(\hat{I})$, $j = 0,\ldots,k$, denote the basis functions satisfying the conditions

$$\hat{\phi}_j(\hat{t}_i) = \delta_{i,j}, \qquad i,j = 0,\ldots,k, \tag{8}$$

where $\hat{t}_0 = -1$ and $\hat{t}_i$, $i = 1,\ldots,k$, are the standard *Gaussian quadrature points* for the reference interval $\hat{I}$. Then, we define the basis functions on the original time interval I_n by

$$\phi_{n,j}(t) := \hat{\phi}_j(\hat{t}) \qquad \text{with} \qquad \hat{t} := T_n^{-1}(t) = \frac{2}{\tau_n}\left(t - \frac{t_n - t_{n-1}}{2}\right) \in \hat{I}. \tag{9}$$

At the end, we obtain the following *time discrete I_n-problem of the cGP(k)-method* [1, 3]: Find on 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

$$\begin{aligned} \textstyle\sum_{j=0}^{k} \alpha_{i,j} \left(\mathbf{U}_n^j, \mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2} a(\mathbf{U}_n^i, \mathbf{v}) + \tfrac{\tau_n}{2} b(\mathbf{v}, P_n^i) &= \tfrac{\tau_n}{2}\left(f(t_{n,i}), \mathbf{v}\right)_\Omega \qquad \forall\, \mathbf{v} \in \mathbf{V}, \\ b(\mathbf{U}_n^i, q) &= 0 \qquad \forall\, q \in Q, \end{aligned} \tag{10}$$

where τ_n denotes the length of the time interval I_n, $\mathbf{U}_n^0 := \mathbf{u}_\tau(t_{n-1})$ for $n > 1$, $\mathbf{U}_1^0 := \mathbf{u}_0$ and $(\cdot,\cdot)_\Omega$ the usual inner product in $L^2(\Omega)$. 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

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

A typical property of this cGP(k)-variant is that the initial pressure P_n^0 of the ansatz (4) does not occur in this formulation. This will be the reason for some problems to achieve superconvergence for the pressure approximation at the discrete time levels t_n.

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 we describe explicitly the well-known dG(1) approach.

1.1 cGP(1)-method

We use the one-point Gaussian quadrature formula with the point $\hat{t}_1 = 0$ and $t_{n,1} = t_{n-1} + \frac{\tau_n}{2}$. Then, we get $\alpha_{1,0} = -1$ and $\alpha_{1,1} = 1$. Thus, equation (10) leads to the following equation for the "one" unknown $\mathbf{U}_n^1 = \mathbf{u}_\tau(t_{n-1}+\frac{\tau_n}{2}) \in \mathbf{V}$ and $P_n^1 = p_\tau(t_{n-1}+\frac{\tau_n}{2}) \in Q$

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

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 polynomial interpolation with the ansatz (4), 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*.

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

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

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

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

The corresponding coupled system reads:

$$\begin{aligned}
\alpha_{1,1}\left(\mathbf{U}_n^1,\mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2}a(\mathbf{U}_n^1,\mathbf{v}) + \alpha_{1,2}\left(\mathbf{U}_n^2,\mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2}b(\mathbf{v},P_n^1) &= \tfrac{\tau_n}{2}\left(f(t_{n,1}),\mathbf{v}\right)_\Omega - \alpha_{1,0}\left(\mathbf{U}_n^0,\mathbf{v}\right)_\Omega \\
\alpha_{2,1}\left(\mathbf{U}_n^1,\mathbf{v}\right)_\Omega + \alpha_{2,2}\left(\mathbf{U}_n^2,\mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2}a(\mathbf{U}_n^2,\mathbf{v}) + \tfrac{\tau_n}{2}b(\mathbf{v},P_n^2) &= \tfrac{\tau_n}{2}\left(f(t_{n,2}),\mathbf{v}\right)_\Omega - \alpha_{2,0}\left(\mathbf{U}_n^0,\mathbf{v}\right)_\Omega \\
b(\mathbf{U}_n^1,q) &= 0 \\
b(\mathbf{U}_n^2,q) &= 0,
\end{aligned} \tag{13}$$

which has to be satisfied for all $\mathbf{v}\in\mathbf{V}$ and $q\in Q$. Once we have determined the solutions $(\mathbf{U}_n^j,P_n^j)$ 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 again by means of polynomial interpolation from the ansatz (4), i.e.,

$$\mathbf{u}_\tau(t_n) = \mathbf{U}_n^0 + \sqrt{3}(\mathbf{U}_n^2 - \mathbf{U}_n^1), \tag{14}$$

where $\mathbf{U}_n^0$ is the initial value at the time interval I_n.

1.3 dG(1)-method

In the dG(1)-method, velocity and pressure are approximated by a discontinuous piecewise linear ansatz space, i.e. $(\mathbf{u}_\tau,p_\tau)\in\mathbf{Y}_\tau^1\times Y_\tau^1$. On time interval I_n we use the polynomial representation

$$\mathbf{u}_\tau(t) := \sum_{j=1}^{2}\mathbf{U}_n^j\phi_{n,j}(t), \qquad p_\tau(t) := \sum_{j=1}^{2}P_n^j\phi_{n,j}(t), \tag{15}$$

with the two "coefficients" $(\mathbf{U}_n^j,P_n^j)\in\mathbf{V}\times Q$, $j=1,2$, which are the values of $\mathbf{u}_\tau$ and p_τ, respectively, at the points $t_{n,j}\in I_n$ of the 2 point Gaussian formula. The real functions $\phi_{n,j}\in\mathbb{P}_1(I_n)$ are the linear Lagrange basis functions with respect to these two Gaussian points.

In order to present the method, we use the following constants for $i,j\in\{1,2\}$

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

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:

$$\begin{aligned}
\gamma_{1,1}\left(\mathbf{U}_n^1,\mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2}a(\mathbf{U}_n^1,\mathbf{v}) + \tfrac{\tau_n}{2}b(\mathbf{v},P_n^1) + \gamma_{1,2}\left(\mathbf{U}_n^2,\mathbf{v}\right)_\Omega &= d_1\left(\mathbf{U}_n^0,\mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2}\left(f(t_{n,1}),\mathbf{v}\right)_\Omega, \\
\gamma_{2,1}\left(\mathbf{U}_n^1,\mathbf{v}\right)_\Omega + \gamma_{2,2}\left(\mathbf{U}_n^2,\mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2}a(\mathbf{U}_n^2,\mathbf{v}) + \tfrac{\tau_n}{2}b(\mathbf{v},P_n^2) &= d_2\left(\mathbf{U}_n^0,\mathbf{v}\right)_\Omega + \tfrac{\tau_n}{2}\left(f(t_{n,2}),\mathbf{v}\right)_\Omega, \\
b(\mathbf{U}_n^1,q) &= 0 \\
b(\mathbf{U}_n^2,q) &= 0
\end{aligned} \tag{16}$$

which has to be satisfied for all $\mathbf{v} \in \mathbf{V}$ and $q \in Q$. Once we have solved the above system, we obtain $\mathbf{u}_\tau$ and p_τ at the time t_n by means of the following linear interpolation

$$\mathbf{u}_\tau(t_n) = \frac{\sqrt{3}+1}{2}\mathbf{U}_n^2 - \frac{\sqrt{3}-1}{2}\mathbf{U}_n^1 \qquad \text{and} \qquad p_\tau(t_n) = \frac{\sqrt{3}+1}{2}P_n^2 - \frac{\sqrt{3}-1}{2}P_n^1. \tag{17}$$

2 Space Discretization by FEM

Next, in each time step, we apply a standard Galerkin finite element discretization with the so-called Q_2/P_1^{disc} Stokes element, i.e., with biquadratic finite elements for the velocity and discontinuous piecewise linear elements for the pressure. This LBB-stable element pair leads to an L_2-approximation order of $O(h^3)$ for the velocity and $O(h^2)$ for the pressure where h denotes the mesh size of the space grid. 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.

2.1 cGP(1)-method

The corresponding 3×3 block system on each time interval I_n reads: *For given initial velocity coefficient vectors* $\underline{\mathbf{U}}_n^0 = (\underline{U}_n^0, \underline{V}_n^0)$, *find* $\underline{U}_n^1, \underline{V}_n^1$ *and a scaled pressure vector* $\underline{\tilde{P}}_n^1$ *such that*

$$\begin{aligned} \left(M + \tfrac{\tau_n}{2}A\right)\underline{U}_n^1 + B_1\underline{\tilde{P}}_n^1 &= \tfrac{\tau_n}{2}F_n^1 + M\underline{U}_n^0 \\ \left(M + \tfrac{\tau_n}{2}A\right)\underline{V}_n^1 + B_2\underline{\tilde{P}}_n^1 &= \tfrac{\tau_n}{2}F_n^2 + M\underline{V}_n^0 \\ B_1^T\underline{U}_n^1 + B_2^T\underline{V}_n^1 &= 0 \end{aligned}$$

where $\underline{\tilde{P}}_n^1 := \frac{\tau_n}{2}\underline{P}_n^1$, and M, A and B denote the mass, Laplacian and gradient matrices, respectively. Once we have determined the solution $\underline{U}_n^1$, $\underline{V}_n^1$ we compute the nodal vector $\underline{U}_{n+1}^0$, $\underline{V}_{n+1}^0$ of the discrete solution $u_{\tau,h}$ at the time t_n by using the following linear extrapolation

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

2.2 cGP(2)-method

The 6×6 block system on each time interval I_n reads: *For given initial velocity vectors* $\underline{\mathbf{U}}_n^0 = (\underline{U}_n^0, \underline{V}_n^0)$, *find* $\underline{U}_n^1, \underline{U}_n^2, \underline{V}_n^1, \underline{V}_n^2$ *and scaled pressure vectors* $\underline{\tilde{P}}_n^1, \underline{\tilde{P}}_n^2$ *such that*

$$\begin{pmatrix} 3M+\tau_nA & (2\sqrt{3}-3)M & 0 & 0 & B_1 & 0 \\ (-2\sqrt{3}-3)M & 3M+\tau_nA & 0 & 0 & 0 & B_1 \\ 0 & 0 & 3M+\tau_nA & (2\sqrt{3}-3)M & B_2 & 0 \\ 0 & 0 & (-2\sqrt{3}-3)M & 3M+\tau_nA & 0 & B_2 \\ B_1^T & 0 & B_2^T & 0 & 0 & 0 \\ 0 & B_1^T & 0 & B_2^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \underline{U}_n^1 \\ \underline{U}_n^2 \\ \underline{V}_n^1 \\ \underline{V}_n^2 \\ \underline{\tilde{P}}_n^1 \\ \underline{\tilde{P}}_n^2 \end{pmatrix} = \begin{pmatrix} R_n^1 \\ R_n^2 \\ R_n^3 \\ R_n^4 \\ 0 \\ 0 \end{pmatrix}$$

where $\underline{\tilde{P}}_n^i := \tau_n \underline{P}_n^i$ and

$$R_n^1 = \tau_n F_n^1 + 2\sqrt{3} M \underline{U}_n^0, \qquad R_n^2 = \tau_n F_n^2 - 2\sqrt{3} M \underline{U}_n^0,$$
$$R_n^3 = \tau_n G_n^1 + 2\sqrt{3} M \underline{V}_n^0, \qquad R_n^4 = \tau_n G_n^2 - 2\sqrt{3} M \underline{V}_n^0.$$

Here, we compute the nodal vector $\underline{U}_{n+1}^0$ and $\underline{V}_{n+1}^0$ of the fully discrete solution $u_{\tau,h}$ at the time t_n by using the following quadratic extrapolation

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

2.3 dG(1)-method

The analogues 6×6 block system on the time interval I_n reads: *For given initial velocity vector* $\underline{\mathbf{U}}_n^0 = (\underline{U}_n^0, \underline{V}_n^0)$, *find* $\underline{U}_n^1, \underline{U}_n^2, \underline{V}_n^1, \underline{V}_n^2$ *and scaled pressure coefficient vectors* $\underline{\tilde{P}}_n^1, \underline{\tilde{P}}_n^2$ *such that*

$$\begin{pmatrix} 2M + \tau_n A & \left(\sqrt{3}-1\right) M & 0 & 0 & B_1 & 0 \\ \left(-\sqrt{3}-1\right) M & 2M + \tau_n A & 0 & 0 & 0 & B_1 \\ 0 & 0 & 2M + \tau_n A & \left(\sqrt{3}-1\right) M & B_2 & 0 \\ 0 & 0 & \left(-\sqrt{3}-1\right) M & 2M + \tau_n A & 0 & B_2 \\ B_1^T & 0 & B_2^T & 0 & 0 & 0 \\ 0 & B_1^T & 0 & B_2^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \underline{U}_n^1 \\ \underline{U}_n^2 \\ \underline{V}_n^1 \\ \underline{V}_n^2 \\ \underline{\tilde{P}}_n^1 \\ \underline{\tilde{P}}_n^2 \end{pmatrix} = \begin{pmatrix} R_n^1 \\ R_n^2 \\ R_n^3 \\ R_n^4 \\ 0 \\ 0 \end{pmatrix}$$

where $\underline{\tilde{P}}_n^i := \tau_n \underline{P}_n^i$ and

$$R_n^1 = \tau_n F_n^1 + \left(\sqrt{3}+1\right) M \underline{U}_n^0, \quad R_n^2 = \tau_n F_n^2 + \left(-\sqrt{3}+1\right) M \underline{U}_n^0,$$
$$R_n^3 = \tau_n G_n^1 + \left(\sqrt{3}+1\right) M \underline{V}_n^0, \quad R_n^4 = \tau_n G_n^2 + \left(-\sqrt{3}+1\right) M \underline{V}_n^0.$$

In this case, we compute the nodal vector $\underline{U}_{n+1}^0$,$\underline{V}_{n+1}^0$ 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

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

One can obtain the pressure at the discrete time points t_n by using the same extrapolation

$$p_{\tau,h}^-(t_n) \sim \underline{P}_{n+1}^0 = \frac{\sqrt{3}+1}{2} \underline{P}_n^2 - \frac{\sqrt{3}-1}{2} \underline{P}_n^1.$$

3 Postprocessing for high order pressure

In many flow problems, often the hydrodynamic forces such as drag, lift etc, have to be calculated. These forces consist of functionals for velocity and pressure at the same discrete

time points. Now, since we have superconvergence results for the velocity only at the discrete time points t_n, it is desirable to get a high order pressure at the same points. In contrast to the dG(1)-method, we cannot obtain the pressure in cGP-methods at the discrete time points by using the same extrapolation as for velocity since this would involve the initial pressure which we do not have. In this section, we explain how to get higher order accuracy for the pressure in the cGP-methods at the discrete time points t_n from the obtained pressure at the intermediate k Gaussian points in the subinterval $[t_{n-1}, t_n]$. The same technique is then also applied for the dG(1)-method which gives better results than the associated extrapolation. To do this, we use the Lagrangian interpolation polynomials to get the solution at time t_n which we explain in the following for the cGP(1), cGP(2) and dG(1)-method, respectively.

3.1 cGP(1)-method

We consider the left and right subinterval at time t_n. Let t_0 and t_1 be the intermediate Gaussian points in these subintervals where the solution is already known. We can construct the corresponding linear Lagrangian interpolation polynomial $L(t)$ such that $L(t_i) = p_\tau(t_i)$ for $i = 0, 1$. Once this linear polynomial $L(t)$ is obtained we can get the solution at the discrete time point t_n. In this way, one extra time step would be required to compute the solution at the end point $t = T$ of the simulation. The corresponding interpolation is visualized in Figure 1.

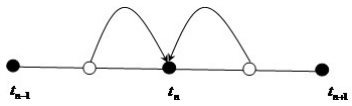

Figure 1: Lagrange interpolation for pressure at the discrete time point t_n.

3.2 cGP(2) and dG(1)-method

In case of cGP(2) or dG(1)-method, we have two Gaussian points in each subinterval I_n. Let t_0, t_1, t_2 and t_3 be the four points in the neighboring subintervals at t_n. Now we construct the cubic Lagrangian polynomial passing through these four points. Once we have determined this polynomial $L(t)$ we obtain the solution at the next discrete time point t_n (see Figure 2). As in case of cGP(1), one more time step is needed to find the solution at the end point.

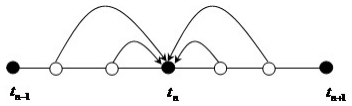

Figure 2: Lagrange interpolation for pressure at the discrete time point t_n.

4 Solution of the linear systems

The resulting linear systems in 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 local pressure Schur complement smoother (see [4, 5]). Multigrid methods are known as the most efficient iterative methods for the solution of large linear systems arising from the discretization of partial differential equations, particularly of elliptic type. In this paper, we use the standard refinement scheme (see [4]) for the grid hierarchies, and for the smoothing operator, a cell centered Vanka like smoother is employed. Moreover, we use the canonical grid transfer routines regarding the chosen FEM space which treat both solution components separately in the case of the cGP(2) and dG(1) approaches (see [2] for the details, particularly regarding the grid transfer for the biquadratic finite elements). Finally, the coarse grid problem is solved by a direct solver.

5 Numerical results

In this section, we perform several numerical tests in order to compare the accuracy of the proposed time discretization schemes. As a test problem we consider the Stokes problem (1) with the domain $\Omega := (0,1)^2$ and $\nu = 1$. 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 $p(x,y,t) := -(x^3 + y^3 - 0.5)(1.5 + 0.5\sin(10\pi t))$. The initial data is $\mathbf{u}_0(x,y) = \mathbf{u}(x,y,0)$.

We apply the time discretization schemes cGP(1), cGP(2) and dG(1) with an equidistant time step size $\tau = T/N$. To measure the error (in time), the following discrete L^∞-norm of a function $v : I \to L^2(\Omega)$ is used

$$\|v\|_\infty := \max_{1\le n\le N} \|v^-(t_n)\|_{L^2(\Omega)}, \qquad v^-(t_n) := \lim_{t\to t_n - 0} v(t), \qquad t_n := n\tau.$$

The behavior of the standard L^2-norm $\|\cdot\|_2 := \|\cdot\|_{L^2(I,L^2(\Omega))}$ and the discrete L^∞-norm of the time discretization error $u(t) - u_{h,\tau}(t)$ for the velocity over the time interval $I = [0,1]$

can be seen in Table 1 and 2, respectively. The estimated value of the experimental order of convergence (EOC) is also calculated and compared with the theoretical order of convergence for both velocity and pressure. All our numerical tests, where we compared the accuracy of our time discretization schemes are related to space mesh level 7 with equidistant $h = 2^{-6}$.

	cGP(1)		cGP(2)		dG(1)	
$1/\tau$	$\|u - u_{h,\tau}\|_2$	EOC	$\|u - u_{h,\tau}\|_2$	EOC	$\|u - u_{h,\tau}\|_2$	EOC
10	5.56E-03		4.40E-04		2.91E-03	
20	1.53E-03	1.86	1.11E-04	1.99	6.51E-04	2.16
40	3.93E-04	1.97	1.33E-05	3.06	1.82E-04	1.84
80	9.87E-05	1.99	1.62E-06	3.04	4.83E-05	1.91
160	2.47E-05	2.00	2.03E-07	3.00	1.25E-05	1.95
320	6.18E-06	2.00			3.17E-06	1.98
640	1.55E-06	2.00			8.00E-07	1.99
1280	3.88E-07	2.00			2.03E-07	1.98
2560	1.01E-07	1.94				

Table 1: Error norms $\|u - u_{h,\tau}\|_2$ for velocity.

	cGP(1)		cGP(2)		dG(1)	
$1/\tau$	$\|u - u_{h,\tau}\|_\infty$	EOC	$\|u - u_{h,\tau}\|_\infty$	EOC	$\|u - u_{h,\tau}\|_\infty$	EOC
10	2.38E-15		6.74E-04		2.18E-03	
20	8.17E-04	-38.32	1.38E-04	2.29	3.73E-04	2.55
40	2.10E-04	1.96	1.03E-05	3.75	5.98E-05	2.64
80	5.13E-05	2.03	6.88E-07	3.90	8.86E-06	2.75
160	1.28E-05	2.00	4.75E-08	3.86	1.19E-06	2.90
320	3.20E-06	2.00			1.60E-07	2.90
640	8.01E-07	2.00				
1280	2.01E-07	1.99				
2560	5.72E-08	1.82				

Table 2: Error norms $\|u - u_{h,\tau}\|_\infty$ for velocity.

We see that the cGP(2)-method is of order 3 in the L^2-norm and superconvergent of order 4 at the discrete time points t_n, while the dG(1)-method is of order 2 in the L^2-norm and superconvergent of order 3 at the end points of the time intervals as expected from the theory. The cGP(1)-method is of order 2 everywhere which is the same behavior as that of the well-known Crank-Nicolson scheme.

Now we show the accuracy of our time discretization schemes for the pressure. Here, we also illustrate the behavior of the L^2-norm $\|\cdot\|_2 := \|\cdot\|_{L^2(I,L^2(\Omega))}$ and the discrete L^∞-norm of the error in the pressure, respectively. From Table 3, we observe that the experimental orders

	cGP(1)		cGP(2)		dG(1)	
$1/\tau$	$\|p-p_{h,\tau}\|_2$	EOC	$\|p-p_{h,\tau}\|_2$	EOC	$\|p-p_{h,\tau}\|_2$	EOC
10	2.10E-01		7.89E-04		2.60E-03	
20	4.25E-02	2.30	3.03E-04	1.38	8.31E-04	1.64
40	1.08E-02	1.97	4.18E-05	2.86	2.53E-04	1.72
80	2.73E-03	1.99	5.31E-06	2.98	7.13E-05	1.82
160	6.83E-04	2.00	6.87E-07	2.95	1.88E-05	1.93
320	1.71E-04	2.00	1.91E-07	1.85	4.86E-06	1.95
640	4.39E-05	1.96			1.25E-06	1.96
1280	1.10E-05	2.00			3.57E-07	1.81
2560	2.75E-06	2.00			1.37E-07	1.38
5120	6.86E-07	2.00				
10240	1.74E-07	1.98				

Table 3: Error norms $\|p-p_{h,\tau}\|_2$ at the Gaussian points for pressure.

of convergence (EOC) coincide with the theoretical orders of convergence for corresponding time discretization schemes. Next, we want to analyze the behavior of L^∞-error for the pressure. As we have already discussed, one can achieve the high order pressure at the discrete time points t_n by using the Lagrangian interpolation polynomials symmetric at t_n. The behavior of the discrete L^∞-norm of the error for pressure can be seen in Table 4.

	cGP(1)		cGP(2)		dG(1)	
$1/\tau$	$\|p-p_{h,\tau}\|_\infty$	EOC	$\|p-p_{h,\tau}\|_\infty$	EOC	$\|p-p_{h,\tau}\|_\infty$	EOC
10	9.97E-06		7.55E-04		2.25E-03	
20	1.00E-01	-13.30	1.35E-03	-0.84	1.36E-03	0.73
40	2.94E-02	1.77	8.86E-05	3.93	1.14E-04	3.57
80	7.63E-03	1.94	5.60E-06	3.98	1.91E-05	2.58
160	1.93E-03	1.99	4.13E-07	3.76	2.43E-06	2.98
320	4.83E-04	2.00			3.84E-07	2.66
640	1.21E-04	2.00				
1280	3.02E-05	2.00				
2560	7.55E-06	2.00				
5120	1.90E-06	1.99				
10240	4.74E-07	2.00				

Table 4: Error norms $\|p-p_{h,\tau}\|_\infty$ for the pressure using Lagrange interpolation.

We observe that the cGP(2)-method has superconvergent results of order 4 for pressure at the discrete time points t_n, while both the cGP(1) and dG(1)-method are of order 2 and 3, respectively, at the end points of the time intervals as expected.

Since the error norms we compared so far contain both the spatial and time error, after a certain stage the space error becomes dominant. To see the accuracy for the time error more clearly, we now compute the norm $\|u_h - u_{h,\tau}\|_\infty \sim \|u_{h,\tau^*} - u_{h,\tau}\|_\infty$ by considering the reference time step size $\tau^* = 1/2560$ for velocity and pressure for the cGP(2) and dG(1)-method.

	cGP(2)				dG(1)			
$1/\tau$	$\|u_h - u_{h,\tau}\|_\infty$	EOC	$\|p_h - p_{h,\tau}\|_\infty$	EOC	$\|u_h - u_{h,\tau}\|_\infty$	EOC	$\|p_h - p_{h,\tau}\|_\infty$	EOC
10	6.74E-04		7.55E-04		2.18E-03		2.25E-03	
20	1.38E-04	2.29	1.35E-03	-0.84	3.73E-04	2.55	1.36E-03	0.73
40	1.03E-05	3.75	8.86E-05	3.93	5.98E-05	2.64	1.14E-04	3.57
80	6.88E-07	3.90	5.60E-06	3.98	8.86E-06	2.75	1.50E-05	2.93
160	4.39E-08	3.97	3.51E-07	4.00	1.19E-06	2.90	2.42E-06	2.63
320	2.75E-09	4.00	2.19E-08	4.00	1.55E-07	2.94	3.46E-07	2.80
640	1.71E-10	4.01	1.37E-09	4.01	1.96E-08	2.98	4.57E-08	2.92

Table 5: Temporal errors for velocity and pressure.

One can see from Table 5 that the experimental orders of convergence for the cGP(2) and dG(1)-methods are much more visible in the absence of spatial discretization errors.

Next, we perform numerical tests to analyze the corresponding behavior of the multigrid solver for the different time discretization schemes. As explained before, the solver uses a cell oriented Vanka type smoother and applies four pre- and post-smoothing steps. We present the averaged number of multigrid iterations per time step for solving the corresponding systems in Table 6. 'Lev' denotes the refinement level of the space mesh.

Lev	$\tau = 1/20$	$\tau = 1/80$	$\tau = 1/320$	$\tau = 1/1280$
3	6-7-7	8-9-8	9-10-10	10-11-10
4	9-8-9	8-8-8	8-10-9	10-11-7
5	9-9-9	8-8-8	8-9-8	9-10-9
6	10-10-9	10-10-8	8-8-8	7-8-8
7	10-10-9	10-10-10	9-9-10	8-8-8

Table 6: Averaged multigrid iterations per time step for cGP(1) - cGP(2) - dG(1).

From Table 6, 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 (due to the non-diagonal mass matrix of Q_2). This means that the behavior of the multigrid solver is almost independent of the spatial mesh size and the time step.

Next, in order to measure and compare the efficiency of the multigrid solver for our time

discretizations, we present in Table 7 the averaged CPU-time required for one solver iteration on a given space mesh level.

	Lev=5			Lev=6			Lev=7		
$1/\tau$	cGP(1)	cGP(2)	dG(1)	cGP(1)	cGP(2)	dG(1)	cGP(1)	cGP(2)	dG(1)
10	0.10	0.33	0.35	0.43	1.40	1.39	1.86	5.98	5.86
20	0.10	0.33	0.35	0.43	1.41	1.40	1.83	5.88	5.84
40	0.11	0.33	0.33	0.47	1.40	1.43	2.03	6.02	6.14
80	0.10	0.34	0.35	0.44	1.41	1.40	2.03	6.12	6.15
160	0.10	0.33	0.36	0.43	1.44	1.41	1.95	6.19	6.11
320	0.10	0.33	0.36	0.53	1.45	1.55	1.94	6.23	5.88
640	0.11	0.34	0.34	0.53	1.40	1.40	2.11	6.22	6.09
1280	0.10	0.36	0.36	0.46	1.48	1.40	1.91	6.23	6.27

Table 7: CPU-time per solver iteration for space mesh level=5,6,7, respectively.

In these numerical tests, the multigrid solver has been implemented in our solver package FEAT2 (www.featflow.de). The CPU-times have been measured on an AMD Opteron 250 at 2.4GHz. In Table 7, we observe that the CPU-time in case of cGP(2) or dG(1) is almost 3 times the CPU-time of cGP(1) for the multigrid solver. We also note that the CPU-time grows approximately by a factor of 4 as expected if we increase the space mesh level. These factors are nearly optimal since the number of space unknowns is increased by a factor of 4 if the level is increased by one.

Finally, we compare the time discretization schemes with respect to accuracy and numerical costs. Here, the multigrid solver uses four Vanka iterations in the pre- and post-smoothing step. The space discretization was done on mesh level 7. Table 8 shows, for different sizes of the time step τ and different time discretization schemes, the discrete L^∞-norm and the total CPU-time for the computation in all time intervals. Due to its superconvergence of order 3 in the discrete time points, the dG(1)-method is faster than cGP(1) which is only of order 2. One can see that, in order to achieve the accuracy of 10^{-7}, we need the very small time step $\tau = 1/2560$ for the cGP(1) while this accuracy can be already achieved with $\tau = 1/160$ and $\tau = 1/320$ in cGP(2) and dG(1)-schemes, respectively. To compare the numerical costs per time step let us note that the number of multigrid iterations to solve one linear block system is approximately the same (about 8) for the three time discretization schemes. However, the costs of one multigrid iteration in the cGP(2) or dG(1) method is almost 3 times higher than in cGP(1). Nevertheless, for a desired accuracy of 10^{-7}, the cGP(2) scheme is about 5 times faster than cGP(1) due to the much larger time step size required for cGP(2).

Next, we also compare our presented time discretization schemes with respect to accuracy and numerical costs for the pressure. To this end, we will only compare the accuracy measured

	cGP(1)		cGP(2)		dG(1)	
$1/\tau$	$\|u-u_{h,\tau}\|_\infty$	CPU	$\|u-u_{h,\tau}\|_\infty$	CPU	$\|u-u_{h,\tau}\|_\infty$	CPU
10	2.38E-15	40	6.74E-04	183	2.18E-03	189
20	8.17E-04	80	1.38E-04	364	3.73E-04	365
40	2.10E-04	160	1.03E-05	713	5.98E-05	764
80	5.13E-05	342	6.88E-07	1401	8.86E-06	1445
160	1.28E-05	645	4.75E-08	2866	1.19E-06	2884
320	3.20E-06	1337			1.60E-07	6145
640	8.01E-07	2833				
1280	2.01E-07	5564				
2560	5.72E-08	13441				

Table 8: Error norms $\|u-u_{h,\tau}\|_\infty$ and total CPU-time to achieve the accuracy of 10^{-7} for velocity field.

in the discrete L^∞-norm. Here, the pressure is obtained at the discrete time points t_n by using the Lagrangian interpolation procedure. From Table 9, it can be seen that to achieve

	cGP(1)		cGP(2)		dG(1)	
$1/\tau$	$\|p-p_{h,\tau}\|_\infty$	CPU	$\|p-p_{h,\tau}\|_\infty$	CPU	$\|p-p_{h,\tau}\|_\infty$	CPU
10	9.97E-06	40	7.55E-04	183	2.25E-03	189
20	1.00E-01	80	1.35E-03	364	1.36E-03	365
40	2.94E-02	160	8.86E-05	713	1.14E-04	764
80	7.63E-03	342	5.60E-06	1401	1.91E-05	1445
160	1.93E-03	645	4.13E-07	2866	2.43E-06	2884
320	4.83E-04	1337			3.84E-07	6145
640	1.21E-04	2833				
1280	3.02E-05	5564				
2560	7.55E-06	13441				
5120	1.90E-06	27427				
10240	4.74E-07	53692				

Table 9: Error norms $\|p-p_{h,\tau}\|_\infty$ and total CPU-time to achieve the accuracy of 10^{-7} for the pressure using interpolation.

the accuracy of 10^{-7}, the cGP(1) and dG(1)-methods need very small time step sizes, i.e., $\tau = 1/10240$ and $\tau = 1/320$, while this accuracy has been already achieved with $\tau = 1/160$ for the cGP(2) scheme. Hence, the cGP(2)-method always gives the accurate results for velocity and pressure in a much more efficient way.

At the end, to show that the proposed time discretization schemes can also efficiently handle the case when the solution approaches a steady state, we provide numerical tests

with very large time steps. We consider problem (1) for $\Omega = (0,1)^2$ and the prescribed (time-independent) velocity

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

and the pressure distribution $p(x,y,t) := -(x^3 + y^3 - 0.5)$. For these analytical solutions for $\mathbf{u}$ and p, we compute the corresponding right hand sides. As initial data we take $\mathbf{u_0} = 0$. Table 10 indicates for the multigrid method the number of solver iterations required for one

Lev	$\tau = 10^{-6}$	$\tau = 10^{-3}$	$\tau = 1$	$\tau = 10^{3}$	$\tau = 10^{6}$
3	5-5-5	5-5-5	4-4-4	4-4-4	4-4-4
4	7-7-7	5-6-6	6-6-6	6-6-6	6-6-6
5	8-8-8	6-7-7	6-7-7	7-7-7	7-7-7
6	8-8-8	6-7-7	9-9-9	9-9-9	9-9-9
7	8-8-8	7-7-7	9-9-9	9-9-9	9-9-9
8	8-8-8	8-8-8	9-9-9	9-9-9	9-9-9

Table 10: Averaged multigrid iterations per time step for cGP(1) - cGP(2) - dG(1).

time step which shows that there is no big difference in the number of solver iterations for time step size $\tau = 10^{-6}$ up to $\tau = 10^{6}$. This means that the behavior of the multigrid convergence is pretty robust with respect to very small as well as very large time steps.

6 Conclusion

We have implemented the *continuous* Galerkin-Petrov and *discontinuous* Galerkin time discretization schemes for the nonstationary Stokes equations. The spatial discretization is carried out by using biquadratic finite elements for velocity and discontinuous linear pressure. The associated linear systems have been solved using a coupled geometrical multigrid method. From the numerical studies, we observe that the estimated experimental orders of convergence confirm the expected theoretical orders. Furthermore, the tests show that the cGP(2)-scheme provides significantly more accurate numerical solutions for both velocity and pressure than the other presented schemes cGP(1) and dG(1) which means that quite large time step sizes are allowed to gain highly accurate results. Secondly, all the presented time discretization schemes are also compared with respect to their numerical costs.

In our recent work, we plan to extend these time discretization schemes to the Navier-Stokes equations to simulate complex time dependent flow problems in a very efficient way together with special Newton-multigrid techniques for the corresponding saddle point problems.

Acknowledgements

The authors want to express their gratitude to the German Research Association (DFG) and the Higher Education Commission (HEC) of Pakistan for their financial support of the study.

References

[1] S. Hussain, F Schieweck, and S. Turek. Higher order Galerkin time discretizations and fast multigrid solvers for the heat equation. *Journal of Numerical Mathematics*. accepted, 2011.

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

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

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

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