On a Galerkin discretization of 4th order in space and time applied to the heat equation

S. Hussain*, F. Schieweck†, S. Turek*, P. Zajac*

Abstract

We present a new time discretization scheme based on the continuous Galerkin Petrov method of polynomial order 3 (cGP(3)-method) which is combined with a reduced numerical time integration (3-point Gauß-Lobatto formula). The solution of the new approach can be computed from the solution of the lower order cGP(2)-method, which requires to solve a coupled 2×2 block system on each time interval, followed by a simple post-processing step, such that we get the higher accuracy of 4th order in time in the standard L^2-norm with nearly the cost of the cGP(2)-method. Moreover, the difference of both solutions can be used as an indicator for the approximation error in time. For the approximation in space we use the nonparametric $\widetilde{Q}_3$-element which belongs to a family of recently derived higher order nonconforming finite element spaces and leads to an approximation error in space of order 4, too, in the L^2-norm. The expected optimal accuracy of the full discretization error in the L^2-norm of 4th order in space and time is confirmed by several numerical tests. We discuss implementation aspects of the time discretization as well as efficient multigrid methods for solving the resulting block systems which lead to convergence rates being almost independent of the mesh size and the time step. In our numerical experiments we compare different higher order spatial and temporal discretization approaches with respect to accuracy and computational cost.

Keywords: continuous Galerkin-Petrov method, nonconforming FEM, heat equation, multigrid method

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

*Institut für Angewandte Mathematik, Technische Universität Dortmund, Vogelpothsweg 87, 44227 Dortmund, Germany

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

1 Introduction

Regarding the 'optimal complexity' of time-dependent PDE simulations, as for instance for the heat equation as simplest example, discretization methods of order 4 in space and time seem to be excellent candidates, particularly in 3D. Since during grid refinement each regular refinement step in time (factor 2) and in space (factor $8 = 2^3$) leads to 16 times more unknowns, a 4th order scheme is necessary to balance this increase in numerical complexity (while for 2D problems, a refinement in space-time leads to 8 times more unknowns such that a 3rd order space-time scheme is already sufficient). Hereby we assume that a specifically adapted multigrid solver is available which exhibits linear complexity w.r.t. the problem size.

A candidate for such a higher order temporal discretization scheme will be proposed in this paper. This scheme arises if we apply in the usual cGP(3)-method [4] (with a cubic polynomial ansatz in time) as a reduced numerical time integration the 3-point Gauß-Lobatto formula. The idea has been previously published in the preprint [8]. We call the resulting scheme the "cGP-C1(3)-method" since it turns out that the numerical solution is a C^1-function in time. In numerical experiments we show that the discretization error in the L^2-norm is of order 4 with respect to the time step size in the whole time interval whereas the standard cGP(2)-method (see [4, 10]) is only superconvergent of order 4 at the endpoints of the time intervals. A theoretical proof of the 4th order accuracy in the case of a nonlinear ODE system has been presented in [8]. Concerning the numerical cost, it turns out that the numerical solution of the cGP-C1(3)-method can be computed on each time interval by solving the (same) coupled 2×2 block system in space of the cGP(2)-method followed by a simple post-processing step which requires to solve one linear system with the mass matrix. Moreover, we demonstrate numerically that the L^2-norm of the difference between the cGP(2) and cGP-C1(3)-solution can be used as indicator for the approximation error in time of the cGP(2)-solution.

As corresponding 4th order space discretization, we use the standard Galerkin Finite Element Method (FEM) with higher order nonconforming (nonparametric) quadrilateral $\widetilde{Q}_3$-elements (see [5]). We use such nonconforming elements since they show an advantageous numerical behaviour for saddle-point problems, particularly for incompressible flow problems together with discontinuous pressure approximations, and they are preferable for parallel computing due to the fact that they only require edge- or face-oriented communication which simplifies the parallel data exchange. To solve the associated linear (block) systems, we propose a geometrical multigrid solver with canonical grid transfer operators due to the FEM space $\widetilde{Q}_3$. The numerical experiments confirm that such multigrid methods [2, 5, 6, 11] are very efficient solvers since their rate of convergence is almost independent of the space mesh size and the size of the time step on structured as well as on semi-structured meshes.

We compare in numerical experiments our new space-time discretization of order 4 with two other discretizations of order 3 and 2 concerning the achieved accuracy in relation to the required CPU-time. The results clearly show the big advantage of the developed high order methodology and the superior computational complexity during grid refinement.

2 The cGP(k)-method for the heat equation

As a model problem we consider the heat equation: *Find* $u : \Omega \times [0,T] \to \mathbb{R}$ *such that*

$$\begin{aligned} d_t u - \triangle u &= f \qquad \text{in} \quad \Omega \times (0,T), \\ u &= 0 \qquad \text{on} \quad \partial\Omega \times [0,T], \\ u(x,0) &= u_0(x) \quad \text{for} \quad x \in \Omega, \end{aligned} \tag{1}$$

where $u(x,t)$ denotes the temperature in the point $x \in \Omega$ at time $t \in [0,T]$, $f : \Omega \times (0,T) \to \mathbb{R}$ a given source term and $u_0 : \Omega \to \mathbb{R}$ the initial temperature field at time $t = 0$. For simplicity, we assume a polygonal domain $\Omega \subset \mathbb{R}^2$ and homogeneous Dirichlet boundary conditions. Then, problem (1) can be considered as an evolution problem in the Hilbert space $V := H_0^1(\Omega)$. Let $(\cdot,\cdot)_\Omega$ denote the inner product in $L^2(\Omega)$ and $a(\cdot,\cdot)$ the following bilinear form

$$a(u,v) := (\nabla u, \nabla v)_\Omega \qquad \forall\, u, v \in V.$$

For the time discretization, we decompose the time interval $I = [0,T]$ into subintervals $I_n := [t_{n-1}, t_n]$, $n = 1, \ldots, N$. Applying the *exact cGP(k)-method* (see [4] for details) we get a time marching process with the following "I_n-problem": *Find* $u_\tau|_{I_n} \in \mathbb{P}_k(I_n, V)$ *such that*

$$\int_{I_n} \Big\{ (d_t u_\tau(t), v)_\Omega + a(u_\tau(t), v) \Big\} \psi_{n,i}(t)\, dt = \int_{I_n} (f(t), v)_\Omega\, \psi_{n,i}(t)\, dt \qquad \forall\, v \in V \tag{2}$$

for $i = 1, \ldots, k$, with the "initial condition" $u_\tau|_{I_n}(t_{n-1}) = u_\tau|_{I_{n-1}}(t_{n-1})$ for $n \geq 2$ or $u_\tau|_{I_1}(t_0) = u_0$ which ensures the continuity of the time discrete solution $u_\tau : I \to V$. The functions $\psi_{n,i}$ denote real-valued basis functions of the polynomial space $\mathbb{P}_{k-1}(I_n)$ and the notation $u_\tau|_{I_n} \in \mathbb{P}_k(I_n, V)$ means that there exist V-valued coefficients $U_n^j \in V$ such that $u_\tau|_{I_n}$ has the representation

$$u_\tau|_{I_n}(t) := \sum_{j=0}^{k} U_n^j \phi_{n,j}(t) \qquad \forall\, t \in I_n, \tag{3}$$

where 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 usual conditions (with $\delta_{i,j}$ denoting the Kronecker symbol)

$$\phi_{n,j}(t_{n,i}) = \delta_{i,j}, \quad i,j = 0, \ldots, k, \qquad \text{such that} \qquad U_n^j = u_\tau|_{I_n}(t_{n,j}) \quad \forall\, j. \tag{4}$$

In [4], we have chosen the time point $t_{n,0} = t_{n-1}$ and the other points $t_{n,1}, \ldots, t_{n,k}$ as the quadrature points of the k-point Gauß formula on I_n. In this case, one has to compute the value $u_\tau|_{I_n}(t_n)$, which is the initial value for the next interval I_{n+1}, by means of (3) as a linear combination of the values $U_n^0, \ldots, U_n^k$. An alternative, which has been applied in [10], is to choose the $t_{n,j}$ as the quadrature points of the $(k+1)$-point Gauß-Lobatto formula. In this paper, we use the Gauß-Lobatto approach of [10] since it enables us to get profit from a relationship between the cGP(2)-method and our new proposed cGP-C1(3)-method which will be explained in Section 3.

For the Gauß-Lobatto points, it holds $t_{n,0} = t_{n-1}$ which implies that the initial condition is equivalent to the condition

$$U_n^0 = u_\tau|_{I_{n-1}}(t_{n-1}) \quad \text{if} \quad n \geq 2 \qquad \text{or} \qquad U_n^0 = u_0 \quad \text{if} \quad n = 1. \tag{5}$$

In order to derive a numerical scheme from the exact cGP(k)-method (2) we transform all I_n-integrals into integrals over the reference interval $\hat{I} := [-1, 1]$ by maens of the affine transformation $T_n : \hat{I} \to I_n$ given by

$$t = T_n(\hat{t}) := \frac{t_{n-1} + t_n}{2} + \frac{\tau_n}{2}\hat{t} \in I_n \qquad \forall\, \hat{t} \in \hat{I},\ n = 1, \ldots, N, \tag{6}$$

and apply for each $\hat{I}$-integral over a given function $g : \hat{I} \to \mathbb{R}$ a suitable m-point quadrature formula of the form

$$\int_{\hat{I}} g(\hat{t})\, d\hat{t} \approx \widehat{Q}_m(g) := \sum_{\mu=0}^{m-1} \widehat{w}_\mu g(\hat{q}_\mu) \tag{7}$$

where $\hat{q}_\mu \in \hat{I}$ denote the quadrature points and $\widehat{w}_\mu$ the weights. This leads us to the *numerically integrated cGP(k)-method*. In particular, if we apply for the quadrature formula $\widehat{Q}_m$ the $(k+1)$-point Gauß-Lobatto formula we will call the corresponding method shortly the cGP(k)-GL($k+1$)-method.

Let us assume that the basis functions $\phi_{n,j}$ and $\psi_{n,i}$ are generated form reference basis functions $\hat{\phi}_j \in \mathbb{P}_k(\hat{I})$ and $\hat{\psi}_i \in \mathbb{P}_{k-1}(\hat{I})$ via the transformation $T_n : \hat{I} \to I_n$ in the way

$$\phi_{n,j} = \hat{\phi}_j \circ T_n^{-1}, \qquad \psi_{n,i} = \hat{\psi}_i \circ T_n^{-1}.$$

Then, inserting the ansatz (3) into (2) and applying the quadrature formula (7), we get the following scheme for the I_n-problem of the numerically integrated cGP(k)-method:

For the given initial value $U_n^0 \in V$, determine the "coefficients" $U_n^1, \ldots, U_n^k \in V$ by solving the coupled system of weak equations:

$$\sum_{j=0}^{k} \Big\{ \alpha_{i,j} \left(U_n^j, v\right)_\Omega + \frac{\tau_n}{2} \beta_{i,j} a(U_n^j, v) \Big\} = \frac{\tau_n}{2} \sum_{\mu=0}^{m-1} \gamma_{i,\mu} \left(f\big(T_n(\hat{q}_\mu)\big), v \right)_\Omega \qquad \forall\, v \in V, \tag{8}$$

where $i = 1, \dots, k$ *and*

$$\alpha_{i,j} := \widehat{Q}_m(\hat{\phi}_j' \hat{\psi}_i), \qquad \beta_{i,j} := \widehat{Q}_m(\hat{\phi}_j \hat{\psi}_i), \qquad \gamma_{i,\mu} := \widehat{w}_\mu \hat{\psi}_i(\hat{q}_\mu). \tag{9}$$

The initial value is defined by $U_n^0 = u_\tau(t_{n-1})$ for $n > 1$ and $U_1^0 = u_0$.

In the following, we specify the cGP(k)-GL($k+1$)-method, which we call simply the "cGP(k)-method", for the cases $k = 1$ and $k = 2$.

2.1 cGP(1)-method

We use the 2-point Gauß-Lobatto formula (trapezoidal rule) with $m = 2$, $\hat{w}_0 = \hat{w}_1 = 1$ and $\hat{q}_0 = -1$, $\hat{q}_1 = 1$. Then, we obtain with $\hat{\psi}_1(\hat{t}) \equiv 1$

$$\alpha_{1,0} = -1, \quad \alpha_{1,1} = 1, \quad \beta_{1,0} = \beta_{1,1} = \gamma_{1,0} = \gamma_{1,1} = 1 \tag{10}$$

and the I_n-problem reads:

For given inital value $U_n^0 \in V$*, find the "unknown"* $U_n^1 = u_\tau(t_n) \in V$ *such that*

$$\left(U_n^1, v\right)_\Omega + \frac{\tau_n}{2} a(U_n^1, v) = \frac{\tau_n}{2} \Big\{ \left(f(t_n), v\right)_\Omega + \left(f(t_{n-1}), v\right)_\Omega \Big\} + \left(U_n^0, v\right)_\Omega - \frac{\tau_n}{2} a(U_n^0, v) \tag{11}$$

holds for all $v \in V$*.* This scheme is the well-known *Crank-Nicolson method*.

2.2 cGP(2)-method

We use the 3-point Gauß-Lobatto formula with $m = 3$, $\hat{w}_0 = \hat{w}_2 = 1/3$, $\hat{w}_1 = 4/3$ and $\hat{q}_0 = -1$, $\hat{q}_1 = 0$, $\hat{q}_2 = 1$, which is the Simpsons rule. For the test functions $\hat{\psi}_i \in \mathbb{P}_1(\hat{I})$ with

$$\hat{\psi}_1(\hat{t}) = \frac{3}{4}(1 - \hat{t}), \qquad \hat{\psi}_2(\hat{t}) = 3\hat{t}$$

we get

$$(\alpha_{i,j}) = \begin{pmatrix} -5/4 & 1 & 1/4 \\ 2 & -4 & 2 \end{pmatrix}, \quad (\beta_{i,j}) = (\gamma_{i,j}) = \begin{pmatrix} 1/2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \tag{12}$$

and the I_n-problem reads:

For given inital value $U_n^0 \in V$ *find the two "unknowns"* $U_n^1, U_n^2 \in V$ *such that*

$$\begin{aligned} \Big\{ \left(U_n^1, v\right)_\Omega + \tfrac{\tau_n}{2} a(U_n^1, v) \Big\} + \tfrac{1}{4} \left(U_n^2, v\right)_\Omega &= \ell_1(v), \\ -4 \left(U_n^1, v\right)_\Omega + \Big\{ 2 \left(U_n^2, v\right)_\Omega + \tfrac{\tau_n}{2} a(U_n^2, v) \Big\} &= \ell_2(v), \end{aligned} \tag{13}$$

holds for all $v \in V$*.* Here, the $\ell_i(\cdot)$ are defined by

$$\begin{aligned} \ell_1(v) &= \tfrac{\tau_n}{2} \Big\{ \left(f(t_{n,1}), v\right)_\Omega + \tfrac{1}{2} \left(f(t_{n-1}), v\right)_\Omega \Big\} + \tfrac{5}{4} \left(U_n^0, v\right)_\Omega - \tfrac{\tau_n}{4} a(U_n^0, v), \\ \ell_2(v) &= \tfrac{\tau_n}{2} \Big\{ \left(f(t_{n,2}), v\right)_\Omega - \left(f(t_{n-1}), v\right)_\Omega \Big\} - 2 \left(U_n^0, v\right)_\Omega + \tfrac{\tau_n}{2} a(U_n^0, v). \end{aligned} \tag{14}$$

3 The cGP-C1(3)-method for the heat equation

Now we modify the cGP(3)-method in the way that we replace for the quadrature formula $\widehat{Q}_m$ in (7) the usual 4-point Gauß-Lobatto formula by the 3-point Gauß-Lobatto formula. We will call the arising method as the "cGP-C1(3)-method" instead of cGP(3)-GL(3)-method since it turns out that the time discrete solution $u_\tau(t)$ is C^1 in time.

Lemma 1 *Let $u_\tau \in C(I,V)$ with $u_\tau|_{I_n} \in \mathbb{P}_3(I_n,V)$ for all $n = 1,\dots,N$, be on each time interval $I_n = [t_{n-1},t_n]$ the solution having the representation (3) with $k = 3$ where the coefficients $U_n^0,\dots,U_n^3 \in V$ satisfy the coupled system (8) with $m = 3$ and the quadrature formula $\widehat{Q}_m$ being the 3-point Gauß-Lobatto formula. Then, it holds $u_\tau \in C^1(I,V)$, i.e.*

$$d_t u_\tau|_{I_n}(t_n) = d_t u_\tau|_{I_{n+1}}(t_n) \qquad \forall\, n = 1,\dots,N-1. \tag{15}$$

Proof. For a time point t in the interior of a time interval I_n, the C^1-property is obvious. Thus, it remains to prove (15). Note that the integration points $\hat{t}_q$ of the 3-point Gauß-Lobatto formula are $\hat{t}_0 = -1$, $\hat{t}_1 = 0$, $\hat{t}_2 = 1$. Since the choice of the basis functions $\hat{\psi}_i \in \mathbb{P}_2(\hat{I})$ does not change the discrete solution u_τ, we choose the last test function $\hat{\psi}_3$ such that

$$\hat{\psi}_3(-1) = 0, \qquad \hat{\psi}_3(0) = 0 \qquad \text{and} \qquad \hat{\psi}_3(1) = 1.$$

Then, it holds

$$\alpha_{3,j} = \widehat{w}_2 \hat{\phi}_j'(1), \qquad \beta_{3,j} = \widehat{w}_2 \hat{\phi}_j(1), \qquad \gamma_{3,\mu} = \widehat{w}_2 \delta_{2,\mu}$$

and we conclude from (8) with $i = 3$, the representation (3) and the properties of the affine transformation $T_n : \hat{I} \to [t_{n-1},t_n]$ that

$$\frac{\tau_n}{2}\widehat{w}_2 \big\{ (d_t u_\tau|_{I_n}(t_n), v)_\Omega + a(u_\tau|_{I_n}(t_n), v) \big\} = \frac{\tau_n}{2}\widehat{w}_2 \, (f(t_n), v)_\Omega \qquad \forall\, v \in V.$$

This implies

$$(d_t u_\tau|_{I_n}(t_n), v)_\Omega = (f(t_n), v)_\Omega - a(u_\tau|_{I_n}(t_n), v) \qquad \forall\, v \in V. \tag{16}$$

In an analogous way we can choose the first basis function $\hat{\psi}_1 \in \mathbb{P}_2(\hat{I})$ such that

$$\hat{\psi}_1(-1) = 1, \qquad \hat{\psi}_1(0) = 0 \qquad \text{and} \qquad \hat{\psi}_1(1) = 0$$

and obtain

$$(d_t u_\tau|_{I_n}(t_{n-1}), v)_\Omega = (f(t_{n-1}), v)_\Omega - a(u_\tau|_{I_n}(t_{n-1}), v) \qquad \forall\, v \in V.$$

Applying this equation on the next interval I_{n+1}, we get

$$\big(d_t u_\tau|_{I_{n+1}}(t_n), v\big)_\Omega = (f(t_n), v)_\Omega - a(u_\tau|_{I_{n+1}}(t_n), v) \qquad \forall\, v \in V. \tag{17}$$

Due to the construction of the method, the discrete solution u_τ is continuous such that the right hand sides of (16) and (17) coincide which proves (15). $\square$

If we would implement the cGP-C1(3)-method in the usual way, then on each time interval $I_n = [t_{n-1}, t_n]$ we would have to solve a coupled system for three "unknowns" $U_n^j = u_\tau(t_{n,j})$, $j = 1, 2, 3$, with $t_{n,j} := T_n(\hat{t}_j)$. These higher computational cost compared to the cGP(2)-method can be reduced if we exploit the connection between the cGP(2)-GL(3) and the cGP-C1(3)-method which was proved in [8] for finite dimensional case. In this way, the piecewise cubic time-discrete solution of the cGP-C1(3)-method on each time interval can be computed by solving the coupled 2×2 block system of the cGP(2)-method followed by a simple post-processing step. Thus, we obtain with nearly the computational cost of the cGP(2)-method a time discrete solution that has optimal 4th order accuracy in the L^2-norm whereas the solution of the cGP(2)-method is only super-convergent of 4th order at the end points of the time intervals.

In the following, we will describe how the post-processing works. To distinguish between the different discrete solutions, we will put the name of the method as an index to u_τ. Moreover, we will omit in the equations that each corresponding u_τ-value is meant as the value of its I_n-restriction. Then, the relationship between the two methods reads as follows [8]:

$$u_\tau^{\text{cGP-C1(3)}}(t) = u_\tau^{\text{cGP(2)}}(t) + a_n \zeta_n(t) \qquad \forall t \in I_n, \tag{18}$$

where the coefficient $a_n \in L^2(\Omega)$ is given by

$$(a_n, v)_\Omega := (f(t_n), v)_\Omega - a(u_\tau^{\text{cGP(2)}}(t_n), v) - \left(d_t u_\tau^{\text{cGP(2)}}(t_n), v\right)_\Omega \qquad \forall\, v \in V \tag{19}$$

and the polynomial $\zeta_n(t) := \frac{\tau_n}{2}\hat{\zeta}(T_n^{-1}(t))$ by the reference polynomial $\hat{\zeta} \in P_3([-1, 1])$ with

$$\hat{\zeta}'(1) = 1, \qquad \hat{\zeta}(-1) = \hat{\zeta}(0) = \hat{\zeta}(1) = 0. \tag{20}$$

Note that we have assumed the H^2-regularity for the solution of the Laplace equation such that $u_\tau^{\text{cGP(2)}}(t_n) \in H^2(\Omega)$ which ensures the existence of $a_n \in L^2(\Omega)$.

4 Space Discretization by FEM

After discretizing equation (1) in time, we now apply the finite element method to discretize each of the "I_n-problems" in space. To this end, let V_h denote a finite element space. In the numerical experiments, V_h will be defined by nonconforming $\widetilde{Q}_3$ finite elements (see Section 5) on a quadrilateral mesh T_h. Since $V_h \not\subset V$ for nonconforming elements, we have to extend

the bilinear form $a(\cdot,\cdot)$ to the discrete version

$$a_h(u_h, v_h) := \sum_{K\in\mathcal{T}_h} (\nabla u_h, \nabla v_h)_K \qquad \forall\, u_h, v_h \in V_h.$$

In order to simplify the coefficients $\beta_{i,j}$ and $\gamma_{i,\mu}$ in (8) for the cGP(k)-GL(k+1)-method where $m = k+1$ and $\hat{t}_\mu = \hat{q}_\mu$ for all $\mu = 0,\dots,k$, we choose the test function $\hat{\psi}_i \in \mathbb{P}_{k-1}(\hat{I})$ such that $\hat{\psi}_i(\hat{q}_\mu) = \delta_{i,\mu}$ for all $i,\mu \in \{1,\dots,k\}$. Then, we have $\beta_{i,j} = \gamma_{i,j} = \delta_{i,j}$ for all $j = 1,\dots,k$ and each "I_n-problem" for the cGP(k)-GL($k+1$)-method has the following structure:
For given $U_n^0 \in V$, find $U_n^1,\dots,U_n^k \in V$ such that

$$\sum_{j=1}^{k} \alpha_{i,j} \left(U_n^j, v\right)_\Omega + \frac{\tau_n}{2} a(U_n^i, v) = \ell_i(v) \qquad \forall\, v \in V, \tag{21}$$

for all $i = 1,\dots,k$, where

$$\ell_i(v) := \frac{\tau_n}{2}\big\{ \left(f(t_{n,i}), v\right)_\Omega + \gamma_{i,0} \left(f(t_{n-1}), v\right)_\Omega \big\} - \alpha_{i,0} \left(U_n^0, v\right)_\Omega - \frac{\tau_n}{2}\beta_{i,0}\, a(U_n^0, v).$$

In the space-time discretization, the system (21) of partial differential equations in weak form is solved approximately by means of the Galerkin method, i.e., each $U_n^j \in V$ is approximated by a finite element function $U_{n,h}^j \in V_h$ and the bilinear form $a(\cdot,\cdot)$ is replaced by its discrete version $a_h(\cdot,\cdot)$. Then, the fully discrete "I_n-problem" reads:
For given $U_{n,h}^0 \in V_h$, find $U_{n,h}^1,\dots,U_{n,h}^k \in V_h$ such that for all $i = 1,\dots,k$ it holds

$$\sum_{j=1}^{k} \alpha_{i,j} \left(U_{n,h}^j, v\right)_\Omega + \frac{\tau_n}{2} a_h(U_{n,h}^i, v) = \ell_{i,h}(v) \qquad \forall\, v \in V_h, \tag{22}$$

where $\ell_{i,h}$ is defined like ℓ_i with U_n^0 replaced by $U_{n,h}^0$ and $a(\cdot,\cdot)$ by $a_h(\cdot,\cdot)$.

Once we have solved this system, we can compute for each time $t \in I_n$ a finite element approximation $u_{\tau,h}(t) \in V_h$ of the time discrete solution $u_\tau(t) \in V$ if we replace in the representation (3) of $u_\tau|_{I_n}$ the coefficients $U_n^j \in V$ by the space discrete coefficients $U_{n,h}^j \in V_h$.

In the following, we will write problem (22) as a linear algebraic block system. Let $b_\mu \in V_h$, $\mu = 1,\dots,m_h$, denote the finite element basis functions of V_h and $\underline{U}_n^j \in \mathbb{R}^{m_h}$ the nodal vector of $U_{n,h}^j \in V_h$ such that

$$U_{n,h}^j(x) = \sum_{\mu=1}^{m_h} (\underline{U}_n^j)_\mu b_\mu(x) \qquad \forall\, x \in \Omega. \tag{23}$$

Furthermore, let us introduce the mass matrix $M \in \mathbb{R}^{m_h\times m_h}$, the discrete Laplacian matrix $L \in \mathbb{R}^{m_h\times m_h}$ and the right hand side vector $R_n^i \in \mathbb{R}^{m_h}$ as

$$M_{\nu,\mu} := (b_\mu, b_\nu)_\Omega\,, \qquad L_{\nu,\mu} := a(b_\mu, b_\nu), \qquad (R_n^i)_\nu := \ell_{i,h}(b_\nu). \tag{24}$$

Then the fully discrete "I_n-problem" is equivalent to the following linear $k \times k$ block system: *For given* $\underline{U}_n^0 \in \mathbb{R}^{m_h}$, *find* $\underline{U}_n^1, \ldots, \underline{U}_n^k \in \mathbb{R}^{m_h}$ *such that*

$$\sum_{j=1}^{k} \alpha_{i,j} M \underline{U}_n^j + \frac{\tau_n}{2} L \underline{U}_n^i = R_n^i, \qquad \forall\, i = 1, \ldots, k. \tag{25}$$

The vector $\underline{U}_n^0$ is defined as the finite element nodal vector of the fully discrete solution $u_{\tau,h}(t_{n-1})$ 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 u_0 if $n = 1$.

In the following, we will present the resulting block systems for the cGP(1), cGP(2) and cGP-C1(3) method which are used in our numerical experiments. To this end, we introduce the vector $F_n^i \in \mathbb{R}^{m_h}$ defined as

$$(F_n^i)_\nu := (f(t_{n,i}), b_\nu)_\Omega \qquad \forall\, \nu = 1, \ldots, m_h.$$

4.1 cGP(1)-method

The problem on time interval I_n reads: *For given* $\underline{U}_n^0 \in \mathbb{R}^{m_h}$, *find* $\underline{U}_n^1 \in \mathbb{R}^{m_h}$ *such that*

$$\left(M + \frac{\tau_n}{2} L\right) \underline{U}_n^1 = \frac{\tau_n}{2}(F_n^1 + F_n^0) + M \underline{U}_n^0 - \frac{\tau_n}{2} L \underline{U}_n^0. \tag{26}$$

4.2 cGP(2)-method

The 2×2 block system on time interval I_n reads: *For given* $\underline{U}_n^0 \in \mathbb{R}^{m_h}$, *find* $\underline{U}_n^1, \underline{U}_n^2 \in \mathbb{R}^{m_h}$ *such that*

$$\begin{pmatrix} M + \frac{\tau_n}{2} L & \frac{1}{4} M \\ -4M & 2M + \frac{\tau_n}{2} L \end{pmatrix} \begin{pmatrix} \underline{U}_n^1 \\ \underline{U}_n^2 \end{pmatrix} = \begin{pmatrix} R_n^1 \\ R_n^2 \end{pmatrix} \tag{27}$$

where

$$\begin{aligned} R_n^1 &= \tfrac{\tau_n}{2}(F_n^1 + \tfrac{1}{2} F_n^0) + \tfrac{5}{4} M \underline{U}_n^0 - \tfrac{\tau_n}{4} L \underline{U}_n^0 \\ R_n^2 &= \tfrac{\tau_n}{2}(F_n^2 - F_n^0) - 2M \underline{U}_n^0 + \tfrac{\tau_n}{2} L \underline{U}_n^0. \end{aligned} \tag{28}$$

4.3 cGP-C1(3)-method

As explained before, we start from the result of the cGP(2)-method and perform a post-processing step to get the discrete solution u_τ of the cGP-C1(3)-method. That means on each time interval I_n, we first have to compute for the given initial vector $\underline{U}_n^0 \in \mathbb{R}^{m_h}$ the solution vectors $\underline{U}_n^1, \underline{U}_n^2 \in \mathbb{R}^{m_h}$ of the coupled block-system (27). In a second step, we solve for the vector $\underline{a}_n \in \mathbb{R}^{m_h}$ the following system with the mass matrix:

$$M \underline{a}_n := F_n^2 - L \underline{U}_n^2 - M \sum_{j=0}^{2} \underline{U}_n^j \phi'_{n,j}(t_n). \tag{29}$$

Using the solution vector $\underline{a}_n$ and the coefficients $\underline{U}_n^j$ of the cGP(2)-method, we obtain the nodal vector of the discrete solution $u_{\tau,h}(t) \in V_h$ of the cGP-C1(3)-method at some time $t \in I_n$ by the formula

$$\underline{U}^{\text{cGP-C1(3)}}(t) = \sum_{j=0}^{2} \underline{U}_n^j \phi_{n,j}(t) + \underline{a}_n \zeta_n(t). \tag{30}$$

5 The nonconforming $\widetilde{Q}_3^n$-element

A family of higher order nonconforming quadrilateral finite elements ($\widetilde{Q}_r$*-elements*) based on a new compatiblity condition was presented in [7]. It was shown in [5] that on general unstructured grids, which are not multilevel grids, the order of approximation can be non-optimal. To overcome this drawback, it has been proposed in [5] to work either with a new $\widetilde{Q}_r^b$-element, which includes additional bubble function, or with the $\widetilde{Q}_r^n$-element based on the so-called *nonparametric* approach of [9] which does not use the usual bilinear reference mapping for the definition of the basis functions.

In this paper, we choose the latter approach and work with the $\widetilde{Q}_3^n$-element which is of fourth order accurate in the L^2-norm. We will shortly describe this element in the rest of this section. For a given mesh cell $K \in \mathcal{T}_h$, let $T_K : \widehat{K} \to K$ denote the usual bilinear mapping on the reference element $\widehat{K} := (-1,1)^2$. We linearize the mapping T_K in the barycenter of $\widehat{K}$ and obtain an affine mapping $\widetilde{T}_K$ defined by

$$\widetilde{T}_K(x,y) := DT_K|_{(0,0)} \cdot (x,y)^\top + T_K(0,0). \tag{31}$$

Furthermore, let $\widetilde{K} := \widetilde{T}_K^{-1}(K)$ denote the pre-image of the element K under $\widetilde{T}_K$, see Figure 1, such that the original mesh cell K is the affine image of the "modified reference element" $\widetilde{K}$.

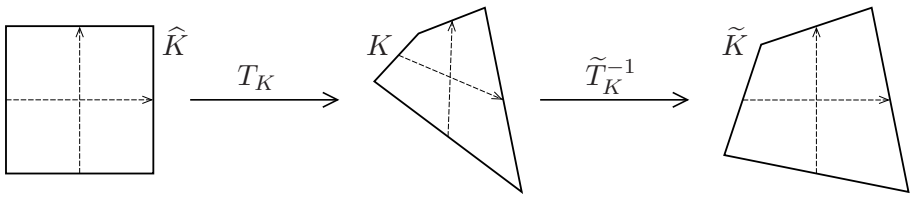


Figure 1: *Correlation between the reference element $\widehat{K}$, the real element K and $\widetilde{K}$.*

If the shape of K is close to that of a parallelogram then $\widetilde{K}$ is close to $\widehat{K}$. The basic idea of the nonparametric approach is to construct the generating local basis functions $\widetilde{\varphi}_k$ as polynomials on $\widetilde{K}$ (and not on $\widehat{K}$). Thus, the "real" local basis functions $\varphi_i : K \to \mathbb{R}$ defined as $\varphi_i(x,y) := \widetilde{\varphi}_i\big(\widetilde{T}_K^{-1}(x,y)\big)$ are polynomials in the real world coordinates (x,y). In the

following, let L_j and $L_{j,k}$ denote the 1D and 2D Legendre polynomials

$$L_0(s) := 1, \qquad L_1(s) := s, \qquad L_2(s) := \frac{1}{2}(3s^2 - 1), \qquad L_{j,k}(\tilde{x}, \tilde{y}) := L_j(\tilde{x}) \cdot L_k(\tilde{y}). \tag{32}$$

Then, the following fifteen nodal functionals are associated with the $\widetilde{Q}_3^n$-element:

- For $j, k \in \{0, 1\},\ j + k \leq 1$, the three *cell moments*

$$\widetilde{N}_{j,k}^{\widetilde{K}}(\widetilde{\varphi}) := |\widetilde{K}|^{-1} \int_{\widetilde{K}} \widetilde{\varphi}(\tilde{x}, \tilde{y}) \cdot L_{j,k}(\tilde{x}, \tilde{y})\, d\widetilde{K}. \tag{33}$$

- For $i \in \{1, \ldots, 4\}$ and $j \in \{0, 1, 2\}$ the twelve *edge moments*

$$\widetilde{N}_j^{\widetilde{E}_i}(\widetilde{\varphi}) := |\widetilde{E}_i|^{-1} \int_{\widetilde{E}_i} \widetilde{\varphi}(\tilde{x}, \tilde{y}) \cdot \big(L_j \circ \widetilde{T}_{E_i}^{-1}\big)(\tilde{x}, \tilde{y})\, d\widetilde{E}_i \tag{34}$$

  of the four associated edges $\widetilde{E}_i$ of $\widetilde{K}$, with $\widetilde{T}_{E_i} : (-1, 1) \to \widetilde{E}_i$ denoting the affine parameterization of $\widetilde{E}_i$.

For the case $\widetilde{K} = \widehat{K}$, it was shown in [3] that the space

$$\begin{aligned} \widetilde{V}_3 := \operatorname{span}\big\{\ & 1,\ \tilde{x},\ \tilde{y},\ \tilde{x}^2,\ \tilde{x}\tilde{y},\ \tilde{y}^2,\ \tilde{x}^3,\ \tilde{x}^2\tilde{y},\ \tilde{x}\tilde{y}^2,\ \tilde{y}^3,\ \tilde{x}^2\tilde{y}^2, \\ & \tilde{x}^3\tilde{y}^2,\ \tilde{x}^2\tilde{y}^3,\ \tilde{x}^3\tilde{y} - \tilde{x}\tilde{y}^3,\ \tilde{x}^4\tilde{y}^2 - \tilde{x}^2\tilde{y}^4\ \big\} \end{aligned} \tag{35}$$

is unisolvent with respect to the above set of nodal functionals. Let $\widetilde{p}_1, \ldots, \widetilde{p}_{15}$ denote the monomial basis functions of $\widetilde{V}_3$ given in (35), $\widetilde{N}_1, \ldots, \widetilde{N}_{15}$ the nodal functionals defined in (33) and (34) and $C \in \mathbb{R}^{15\times 15}$ the matrix defined by

$$C_{ij} := \widetilde{N}_i(\widetilde{p}_j). \tag{36}$$

Assuming that $\widetilde{V}_3$ is unisolvent with respect to the $\widetilde{N}_i$ also for the actual modified reference element $\widetilde{K} := \widetilde{T}_K^{-1}(K)$, we get that C is regular, that means that its inverse C^{-1} exsits. Then, the local basis functions $\widetilde{\varphi}_i : \widetilde{K} \to \mathbb{R}$, $i = 1, \ldots, 15$, that satisfy the usual duality property

$$\widetilde{N}_j(\widetilde{\varphi}_i) = \delta_{i,j} \qquad \forall\ i, j \in \{1, \ldots, 15\}, \tag{37}$$

can be computed by

$$\widetilde{\varphi}_i(\tilde{x}, \tilde{y}) := \sum_{j=1}^{15} (C^{-1})_{ij} \cdot \widetilde{p}_j(\tilde{x}, \tilde{y}). \tag{38}$$

The unisolvence for the general case $\widetilde{K} \neq \widehat{K}$ can be guaranteed if the unstructured mesh $\mathcal{T}_h$ does not have quadrilaterals that are too much distorted. The local basis functions $\varphi_i : K \to \mathbb{R}$ on the original mesh cell $K \in \mathcal{T}_h$ are defined by means of the affine mapping $\widetilde{T}_K : \widetilde{K} \to K$ as $\varphi_i := \widetilde{\varphi}_i \circ \widetilde{T}_K^{-1}$. Due to the choice of $\widetilde{V}_3$, we have that the polynomial space $\mathbb{P}_3(\widetilde{K})$ is contained in $\widetilde{V}_3$. Since the mapping $\widetilde{T}_K$ is affine we can show that

$$\mathbb{P}_3(K) \subset V_3 := \operatorname{span}\big\{\ \varphi_1, \ldots, \varphi_{15}\ \big\}, \tag{39}$$

which ensures the optimal approximation order of the $\widetilde{Q}_3^n$-element, see [1].

6 Solution of the linear systems

The resulting discretized linear systems in each time interval $[t_{n-1}, t_n]$, which are, for the cGP(1)-method, 1x1 block systems of the form

$$\left(M + \frac{\tau_n}{2} L\right) \underline{U}_n^1 = R_n^1 \tag{40}$$

and, for the cGP(2)- or cGP-C1(3)-method, 2x2 block systems of the form

$$\begin{pmatrix} M + \frac{\tau_n}{2} L & \frac{1}{4} M \\ -4M & 2M + \frac{\tau_n}{2} L \end{pmatrix} \begin{pmatrix} \underline{U}_n^1 \\ \underline{U}_n^2 \end{pmatrix} = \begin{pmatrix} R_n^1 \\ R_n^2 \end{pmatrix}, \tag{41}$$

are treated by using a geometrical multigrid solver with corresponding (block) smoothers and grid transfer operators. In this paper, we use the standard refinement scheme (see [11]) for the grid hierarchies, and for the smoothing operator, we choose the standard (pointwise) Gauß-Seidel method (with four pre- and post-smoothing steps); however, also corresponding standard (block) variants of Jacobi, SOR and ILU methods can be easily applied, too. Moreover, we use for the canonical grid transfer routines the standard FEM projection operator defined for the $\widetilde{Q}_3^n$-element (see [11] and [6] for a corresponding approach for first order nonconforming and biquadratic conforming finite elements). Finally, the coarse grid problem is solved by a direct solver. Once we have solved the block system (41) for the unknowns $\underline{U}_n^1$ and $\underline{U}_n^2$, we compute in the post-processing step the unknown $\underline{a}_n$ from equation (29) in order to get the cGP-C1(3) solution. To this end, the block system in (29) with a mass matrix can be easily solved by applying any iterative solver. In our numerical results, we apply the standard Gauß-Seidel method to solve this system.

7 Numerical results

In this section, we perform several numerical tests to analyze the presented spatial and temporal discretizations. For all examples, we use the domain $\Omega = (0,1)^2$ and the time interval $I = [0, T]$ with $T = 1$. In order to be able to measure the exact error of the numerical solution we prescribe an anlytical exact solution $u(x, y, t)$ and compute the associated data f and u_0 from the heat equation (1).

7.1 Error in space

For the study of the approximation properties of the $\widetilde{Q}_3^n$-element, we consider sequences of structured and semi-structured meshes which are generated by uniform refinement from a coarsest mesh with one and three quadrilateral mesh cells, respectively. Starting from the

coarsest grid defined as mesh level $\ell = 1$, we generate the grid of mesh level $\ell + 1$ by dividing each quadrilateral cell of grid level ℓ into four new quadrilaterals connecting the midpoints of opposite edges. Figure 2 shows the grids on level $\ell = 1, 2, 3$ for the structured and semi-structured meshes. In case of the structured mesh, the mesh size on grid level ℓ is $h = 2^{-\ell+1}$.

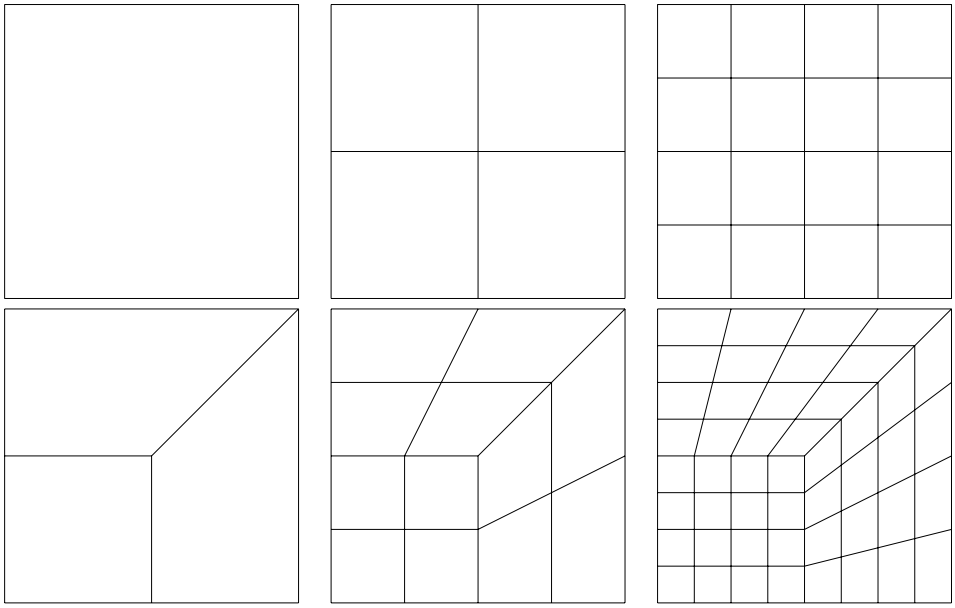


Figure 2: Structured (above) and semi-structured (below) grids on space mesh level=1,2,3 (from left to right).

Example 1 *We consider problem* (1) *with the prescribed exact solution*

$$u(x, y, t) := (t^3 - t^2 + t) \sin(\pi x) \sin(\pi y),$$

and the associated data f *and* u_0*.*

Since the exact solution is in the polynomial space $\mathbb{P}_3(I, V)$, we expect that the time discrete solution u_τ of the cGP-C1(3)-method is exact for any number of time intervals. Therefore, only an approximation error in space can occur in the fully discrete solution $u_{\tau,h}$. To analyze the quality of the spatial discretization, we present in Table 1 (structured meshes) and Table 2 (semi-structured meshes) for different refinement level ℓ and fixed equidistant time step size $\tau = T/N = 1/4$ the discretization error $u - u_{\tau,h}$ over the time interval $I = [0, 1]$ measured in the standard L^2-norm $\|\cdot\|_{2,L} := \|\cdot\|_{L^2(I,L^2(\Omega))}$ and the discrete L^∞-seminorm on $C^0(I, V)$ defined as

$$|v|_{\infty,L} := \max_{1 \le n \le N} \|v(t_n)\|_{L^2(\Omega)}, \qquad t_n := n\tau. \tag{42}$$

ℓ	$\|u-u_{\tau,h}\|_{2,L}$	EOC	$\|u-u_{\tau,h}\|_{\infty,L}$	EOC	#DOFs	Factor	CPU	Factor
3	1.77E-04		3.60E-04		336		0.16	
4	1.13E-05	3.97	2.30E-05	3.97	1248	3.71	0.53	3.35
5	7.13E-07	3.99	1.45E-06	3.99	4800	3.85	2.10	3.95
6	4.46E-08	4.00	9.05E-08	4.00	18816	3.92	9.87	4.69
7	2.79E-09	4.00	5.66E-09	4.00	74496	3.96	42.51	4.31
8	1.74E-10	4.00	3.54E-10	4.00	296448	3.98	181.62	4.27
9	1.49E-11	3.55	2.22E-11	4.00	1182720	3.99	769.93	4.24

Table 1: Discretization errors of the cGP-C1(3)-solution $u_{\tau,h}$ for Example 1 at different grid level ℓ of structured meshes for $\tau = 1/4$.

ℓ	$\|u-u_{\tau,h}\|_{2,L}$	EOC	$\|u-u_{\tau,h}\|_{\infty,L}$	EOC	#DOFs	Factor	CPU	Factor
3	4.91E-05		9.96E-05		936		0.40	
4	3.20E-06	3.94	6.50E-06	3.94	3600	3.85	1.90	4.80
5	2.02E-07	3.99	4.09E-07	3.99	14112	3.92	8.36	4.40
6	1.26E-08	4.00	2.56E-08	4.00	55872	3.96	35.12	4.20
7	7.89E-10	4.00	1.60E-09	4.00	222336	3.98	163.63	4.66
8	5.12E-11	3.95	1.00E-10	4.00	887040	3.99	751.51	4.59

Table 2: Discretization errors of the cGP-C1(3)-solution $u_{\tau,h}$ for Example 1 at different grid level ℓ of semi-structured meshes for $\tau = 1/4$.

By "EOC" we denote the *experimental order of convergence* and in the column "#DOFs" we show the total number of all unknowns of the 2x2 block-systems on each time interval which is $2m_h$ where m_h denotes the dimension of the space V_h. Furthermore, we present in column "CPU" the CPU-times in seconds for the linear solver implemented within the solver package FEAT2 (www.featflow.de) and performed on a Dual-Core AMD Opteron 8220 with eight CPUs at 2.8GHz.

Table 1 and 2 show (independent of the large $\tau = 1/4$) that the error of the fully discrete cGP-C1(3)-solution $u_{\tau,h}$ in both $L^2(\Omega)$-type norms behaves like $\mathcal{O}(h^4)$ with respect to the mesh size h which confirms the expected optimal order for the used $\widetilde{Q}_3^n$-element. Moreover, it can be seen that this element works well on the structured and the semi-structured meshes. Comparing the computational cost for the linear solver, we observe that the CPU-time on the next higher grid level $\ell+1$ increases by a factor of appr. 4 whereas the corresponding number of degrees of freedom grows by factor of 4. This shows that our coupled multigrid solver has nearly optimal computational complexity.

To investigate the approximation properties of the $\widetilde{Q}_3^n$-element on more general meshes we use on each grid level $\ell \geq 2$ a perturbed grid $\mathcal{T}^\ell$ which was created as follows. In a first

step, $\ell - 1$ recursive uniform refinements are applied to the original coarsest grid $\mathcal{T}^1$, see Figure 2, and in a second step the resulting grid is modified by a stochastic perturbation of the grid vertices with a magnitude of 5%, 10% and 20% of the mesh size h, see Figure 3. In

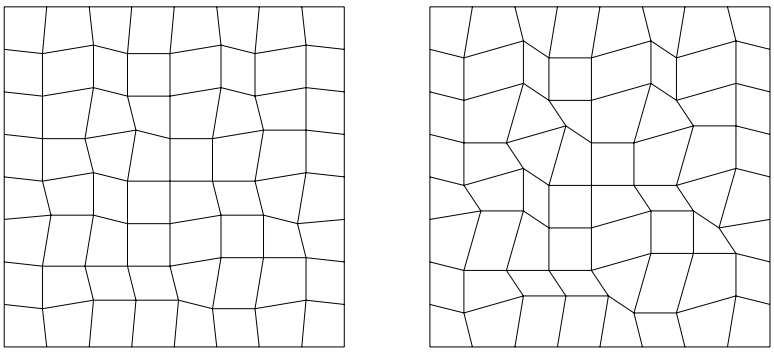


Figure 3: Perturbed grids on level $\ell = 4$ which are stochastically deformed with 10% (left) and 20% (right).

Table 3, the numerical results confirm that the nonparametric $\widetilde{Q}_3^n$-element also works well for the considered range of perturbed (but shape-regular) multilevel grids with an accuracy of 4th order in the L^2-norm and the discrete L^∞-norm. Thus, the nonparametric $\widetilde{Q}_3^n$-element converges on quite general meshes with an optimal rate of convergence.

	5% deformation		10% deformation		20% deformation	
ℓ	$\|u - u_{\tau,h}\|_{2,L}$	EOC	$\|u - u_{\tau,h}\|_{2,L}$	EOC	$\|u - u_{\tau,h}\|_{2,L}$	EOC
3	1.76E-04		1.80E-04		1.96E-04	
4	1.16E-05	3.93	1.25E-05	3.85	1.68E-05	3.54
5	7.28E-07	3.99	7.74E-07	4.01	9.87E-07	4.09
6	4.57E-08	3.99	4.96E-08	3.96	6.87E-08	3.84
7	2.88E-09	3.99	3.17E-09	3.97	4.54E-09	3.92
8	1.80E-10	4.00	1.98E-10	4.00	2.82E-10	4.01
ℓ	$\|u - u_{\tau,h}\|_{\infty,L}$	EOC	$\|u - u_{\tau,h}\|_{\infty,L}$	EOC	$\|u - u_{\tau,h}\|_{\infty,L}$	EOC
3	3.58E-04		3.65E-04		3.97E-04	
4	2.34E-05	3.93	2.53E-05	3.85	3.41E-05	3.54
5	1.48E-06	3.99	1.57E-06	4.01	2.00E-06	4.09
6	9.28E-08	3.99	1.01E-07	3.96	1.39E-07	3.84
7	5.84E-09	3.99	6.42E-09	3.97	9.19E-09	3.92
8	3.65E-10	4.00	4.00E-10	4.00	5.71E-10	4.01

Table 3: Discretization errors of the cGP-C1(3)-solution $u_{\tau,h}$ measured in $\|\cdot\|_{2,L}$ (above) and $|\cdot|_{\infty,L}$ (below) for Example 1 on different multilevel grids at level ℓ with stochastical deformation for $\tau = 1/4$.

7.2 Error in time

In this section, we perform numerical tests in order to analyze the accuracy of the proposed time discretization schemes. For all tests, we use an equidistant time step size $\tau = T/N$ where $T = 1$. As in the previous section, we measure the error in the norm $\|\cdot\|_{2,L}$ and seminorm $|\cdot|_{\infty,L}$.

Example 2 *We consider problem* (1) *with the prescribed exact solution*

$$u(x, y, t) = x(1 - x)y(1 - y)e^t,$$

and the associated data f *and* u_0.

This example has the property that, for each $t \in [0, T]$, the exact solution $u(t)$ is contained in the finite element space V_h and that the consistency error of the nonconforming discretization is zero on the used meshes. Therefore, we have no discretization error with respect to space and in the full discretization error $u - u_{\tau,h}$ we only see the error of the time approximation.

In Table 4, we compare the discretization errors $e_{\tau,h} := u - u_{\tau,h}$ between the cGP(2)- and the cGP-C1(3)-method for a fixed space mesh of level $\ell = 6$.

	cGP(2)				cGP-C1(3)			
$1/\tau$	$\|e_{\tau,h}\|_{2,L}$	EOC	$\|e_{\tau,h}\|_{\infty,L}$	EOC	$\|e_{\tau,h}\|_{2,L}$	EOC	$\|e_{\tau,h}\|_{\infty,L}$	EOC
10	4.05E-07		1.49E-08		5.83E-09		1.49E-08	
20	5.07E-08	3.00	9.41E-10	3.98	3.62E-10	4.01	9.41E-10	3.98
40	6.33E-09	3.00	5.90E-11	4.00	2.27E-11	4.00	5.90E-11	4.00
80	7.92E-10	3.00	3.69E-12	4.00	1.42E-12	4.00	3.69E-12	4.00
160	9.90E-11	3.00	2.29E-13	4.01	9.38E-14	3.92	2.29E-13	4.01
320	1.24E-11	3.00						
640	1.55E-12	3.00						
1280	1.93E-13	3.00						

Table 4: Discretization errors for Example 2 at space mesh level 6.

We see that, for fixed h and $\tau \to 0$, the cGP(2)-method converges with order 3 in the L^2-norm which is optimal with respect to its quadratic time polynomial ansatz. Moreover, it is superconvergent of order 4 at the discrete points t_n. The cGP-C1(3)-method converges of order 4 in the L^2-norm as well as in the discrete L^∞-norm which is optimal due to its cubic time polynomial ansatz. Moreover, the identical numbers in the seminorm $|e_{\tau,h}|_{\infty,L}$ confirm the fact that the cGP-C1(3)-method coincides with the cGP(2)-method at the discrete time points τ_n which follows from (18).

Next, we compare the cGP(2)- and cGP-C1(3)-method concerning the accuracy and the numerical cost to achieve this accuracy on a structured mesh of level $\ell = 6$. From Table 5, it

	cGP(2)				cGP-C1(3)		
$1/\tau$	$\|e_{\tau,h}\|_{2,L}$	CPU	Factor	$\|e_{\tau,h}\|_{\infty,L}$	$\|e_{\tau,h}\|_{2,L}$	CPU	Factor
10	4.05E-07	17.37		1.49E-08	6.00E-09	18.10	
20	5.07E-08	35.31	2.03	9.41E-10	3.68E-10	36.75	2.03
40	6.33E-09	69.87	1.98	5.90E-11	2.29E-11	72.75	1.98
80	7.92E-10	138.20	1.98				
160	9.90E-11	279.07	2.02				

Table 5: Discretization errors, total CPU-times (in seconds) and increasing time factor from the next coarser time step size for Example 2 at space mesh level 6.

can be seen that in order to get the accuracy of 10^{-10} in the L^2-norm we need the small time step size of $\tau = 1/160$ for the cGP(2)-method while this accuracy has been already achieved with $\tau = 1/40$ in the cGP-C1(3) method which is 4 times larger than for the cGP(2)-method. Furthermore, we observe that, for the same time step size τ, the CPU time of the cGP-C1(3)-method is only a little bit larger than that of the cGP(2)-method which means that the computational cost for the post-processing step are relatively small. Finally, the CPU time to obtain an accuracy of 10^{-10} is nearly 4 times smaller for the cGP-C1(3)-method.

7.3 Error in space and time

Now, we analyze the behavior of the full discretization error $u(t) - u_{h,\tau}(t)$ in an example where an approximation error in space as well as in time occurs.

Example 3 *We consider problem* (1) *with the prescribed exact solution*

$$u(x, y, t) := \sin(\pi x) \sin(\pi y) \sin(10\pi t),$$

and the associated data f *and* u_0.

In Table 6, we present in each of the three column blocks for grid level $\ell = 6, 7, 8$, the error norms and the experimental orders of convergence (EOC) for decreasing τ. 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 $\tau_0(h_\ell)$. Whereas Table 6 shows the results for the structured meshes we show in Table 7 the analogous results for the semi-structured meshes.

We see that in the standard L^2-norm $\|\cdot\|_{2,L}$ as well as in the discrete seminorm $|\cdot|_{\infty,L}$ the error of the cGP-C1(3)-method behaves, for fixed mesh size h_ℓ, like $\mathcal{O}(\tau^4)$ 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

	ℓ=6		ℓ=7		ℓ=8	
$1/\tau$	$\|u-u_{h,\tau}\|_{2,L}$	EOC	$\|u-u_{h,\tau}\|_{2,L}$	EOC	$\|u-u_{h,\tau}\|_{2,L}$	EOC
10	6.13E-02		6.13E-02		6.13E-02	
20	2.80E-03	4.45	2.80E-03	4.45	2.80E-03	4.45
40	1.73E-04	4.01	1.73E-04	4.01	1.73E-04	4.01
80	1.08E-05	4.00	1.08E-05	4.00	1.08E-05	4.00
160	6.78E-07	3.99	6.75E-07	4.00	6.75E-07	4.00
320	7.67E-08	3.14	4.24E-08	3.99	4.22E-08	4.00
640	6.41E-08	0.26	4.79E-09	3.14	2.65E-09	3.99
1280	6.40E-08	0.00	4.01E-09	0.26	3.02E-10	3.13
2560	6.40E-08	0.00	4.00E-09	0.00	2.53E-10	0.26
$1/\tau$	$\|u-u_{h,\tau}\|_{\infty,L}$	EOC	$\|u-u_{h,\tau}\|_{\infty,L}$	EOC	$\|u-u_{h,\tau}\|_{\infty,L}$	EOC
10	6.11E-02		6.11E-02		6.11E-02	
20	2.97E-03	4.36	2.97E-03	4.36	2.97E-03	4.36
40	2.16E-04	3.79	2.16E-04	3.79	2.16E-04	3.79
80	1.33E-05	4.02	1.33E-05	4.02	1.33E-05	4.02
160	8.40E-07	3.98	8.36E-07	3.99	8.36E-07	3.99
320	1.02E-07	3.05	5.24E-08	4.00	5.22E-08	4.00
640	9.06E-08	0.17	6.36E-09	3.04	3.28E-09	3.99
1280	9.05E-08	0.00	5.66E-09	0.17	3.97E-10	3.04
2560	9.05E-08	0.00	5.66E-09	0.00	3.54E-10	0.17

Table 6: Full discretization error $u - u_{h,\tau}$ measured in $\|\cdot\|_{2,L}$ and $|\cdot|_{\infty,L}$ for Example 3 at different levels of structured space meshes.

the sequence of space-time meshes with $(\tau, h) = (\tau_0(h_\ell), h_\ell)$, $\ell = 6, 7, 8$, we observe that the error decreases by a factor of about 16 if we increase the level ℓ by one. This indicates an asymptotic behaviour of the form

$$\|u - u_{\tau,h}\|_L \leq C(\tau^4 + h^4)$$

where $\|\cdot\|_L$ stands for $\|\cdot\|_{2,L}$ or $|\cdot|_{\infty,L}$. This asymptotic behaviour is optimal with respect to the cubic polynomial ansatz of the cGP-C1(3)-method in time and the cubic ansatz of the $\widetilde{Q}_3^n$-element in space. Table 7 shows analogous results for the case of semi-structured meshes. Moreover, it demonstrates that the temporal accuracy is not disturbed due to the semi-structured meshes.

Finally, we want to show that, for $\tau \geq \tau_0(h)$, we can use the quantity

$$\eta_n^h := \left\| u_{\tau,h}^{\text{cGP(2)}} - u_{\tau,h}^{\text{cGP-C1(3)}} \right\|_{L^2(I_n, L^2(\Omega))}$$

	ℓ=6		ℓ=7		ℓ=8	
$1/\tau$	$\|u-u_{h,\tau}\|_{2,L}$	EOC	$\|u-u_{h,\tau}\|_{2,L}$	EOC	$\|u-u_{h,\tau}\|_{2,L}$	EOC
10	6.13E-02		6.13E-02		6.13E-02	
20	2.80E-03	4.45	2.80E-03	4.45	2.80E-03	4.45
40	1.73E-04	4.01	1.73E-04	4.01	1.73E-04	4.01
80	1.08E-05	4.00	1.08E-05	4.00	1.08E-05	4.00
160	6.75E-07	4.00	6.75E-07	4.00	6.75E-07	4.00
320	4.59E-08	3.88	4.22E-08	4.00	4.22E-08	4.00
640	1.83E-08	1.33	2.87E-09	3.88	2.65E-09	3.99
1280	1.81E-08	0.02	1.14E-09	1.33	4.79E-10	2.47
2560	1.81E-08	0.00	1.14E-09	0.00	4.79E-10	0.00
$1/\tau$	$\|u-u_{h,\tau}\|_{\infty,L}$	EOC	$\|u-u_{h,\tau}\|_{\infty,L}$	EOC	$\|u-u_{h,\tau}\|_{\infty,L}$	EOC
10	6.11E-02		6.11E-02		6.11E-02	
20	2.97E-03	4.36	2.97E-03	4.36	2.97E-03	4.36
40	2.16E-04	3.79	2.16E-04	3.79	2.16E-04	3.79
80	1.33E-05	4.02	1.33E-05	4.02	1.33E-05	4.02
160	8.37E-07	3.99	8.36E-07	3.99	8.36E-07	3.99
320	5.67E-08	3.88	5.22E-08	4.00	5.22E-08	4.00
640	2.57E-08	1.14	3.54E-09	3.88	3.27E-09	4.00
1280	2.56E-08	0.01	1.61E-09	1.14	2.22E-10	3.88
2560	2.56E-08	0.00	1.61E-09	0.00	1.01E-10	1.14

Table 7: Full discretization error $u-u_{h,\tau}$ measured in $\|\cdot\|_{2,L}$ and $|\cdot|_{\infty,L}$ for Example 3 at different levels of semi-structured space meshes.

as an indicator for the error of the cGP(2)-solution $u_{\tau,h}^{\mathrm{cGP(2)}}$ on time interval I_n

$$e_n^{\tau,h} := \left\| u_{\tau,h}^{\mathrm{cGP(2)}} - u \right\|_{L^2(I_n,L^2(\Omega))}.$$

To this end, we consider Example 3 with $h=2^{-6}$ (grid level $\ell=7$), $\tau=1/40$ and present in the left part of Figure 4 the exact error $e_n^{\tau,h}$ vs. the indicator η_n^h for $n=1,\dots,40$. We observe that the indicator is a good approximation of the exact error. In order to explain the oscillating behaviour of the errros we show in the right part of Figure 4 the amplitude $\sin(10\pi t_n)$ of the exact solution $u(t_n)$.

7.4 Solver analysis and comparison with other schemes

Finally, we perform numerical tests to analyze the behavior of the multigrid solver in the cGP(2)-method for Example 3. Note that the numerical cost for cGP-C1(3) mainly depend on the cost to compute the cGP(2)-solution whereas the cost for performing the post-processing step are much less than for solving the block-system of the cGP(2)-method (see Table 5).

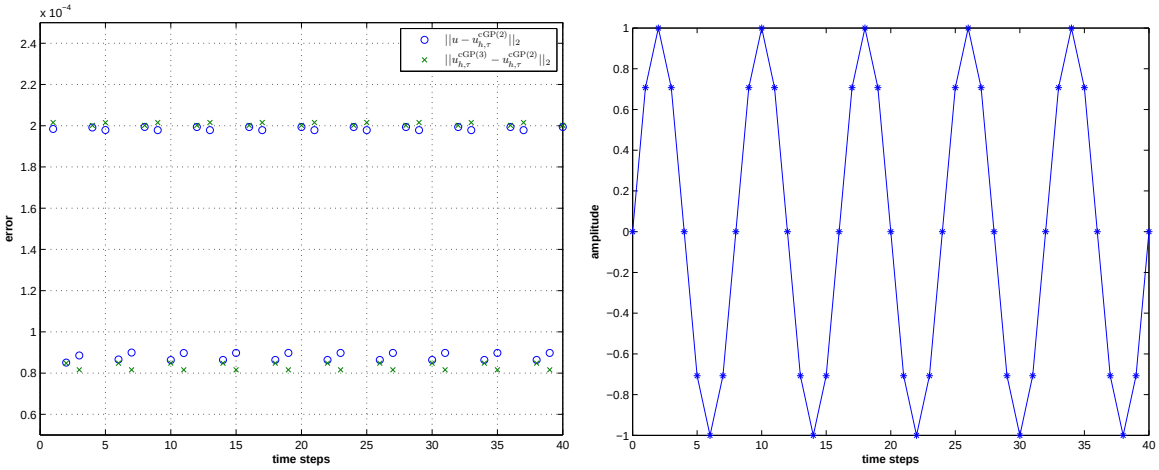


Figure 4: Left part: excact errors $e_n^{\tau,h}$ (symbol o) vs. error indicators η_n^h (symbol x). Right part: amplitudes of the exact solution at time points t_n.

In order to analyze the multigrid behavior, we present in Table 8 the averaged number of multigrid iterations per time step for solving the corresponding systems. The solver stops if the L^2-norm of the relative residual is smaller than 10^{-10} or the absolute residual drops down by 10^{-15}. From Table 8, we see that, for fixed time step τ, the multigrid solver requires almost the same number of iterations for increasing grid level ℓ. Moreover, if we decrease τ for fixed grid level ℓ, the number of multigrid iterations does not increase. This means that the behavior of the multigrid solver is almost independent of the space mesh size and the time step. Comparing the convergence behavior between structured and semi-structured meshes, we see that the averaged number of iterations is only slightly higher on semi-structured meshes but it remains also almost independent of the space mesh size and the time step.

Next, for Example 2 and on structured meshes, we compare the time discretization schemes cGP(2) and cGP-C1(3) with the cGP(1)- (or Crank-Nicolson-) method concerning the achieved accuracy and the required numerical cost. Table 9 shows, for different τ and grid level $\ell = 6$, the global L^2-norm error and the total CPU-time (in seconds) required for the computations on all time intervals. One can see that, in order to achieve the accuracy of 10^{-6}, we need the very small time step $\tau = 1/10240$ for the cGP(1)-scheme while the same accuracy can be already achieved with $\tau = 1/640$ and $\tau = 1/160$ in the case of the cGP(2)- and cGP-C1(3)-scheme, respectively, due to their 3rd and 4th order accuracy. Thus, for a desired accuracy of 10^{-6}, the scheme cGP-C1(3) is about 12 times faster than cGP(1) and 4 times faster than cGP(2).

Finally, we compare the accuracy and the numerical cost of our proposed higher order

ℓ	$\tau = 1/10$	$\tau = 1/40$	$\tau = 1/160$	$\tau = 1/640$
3	11.0	9.0	8.0	7.1
4	11.0	11.5	9.8	8.0
5	12.0	12.0	11.9	10.0
6	12.0	12.0	12.0	11.8
7	12.0	12.0	11.9	11.4
8	12.0	12.0	11.8	10.8
3	13.0	12.0	10.6	9.0
4	14.0	14.3	12.9	10.0
5	16.0	15.5	15.4	13.9
6	17.0	17.0	16.0	15.5
7	17.0	17.0	16.5	14.6
8	17.0	16.7	15.3	13.6

Table 8: Averaged multigrid iterations per time step for the cGP(2)-method on structured (above) and semi-structured (below) meshes.

space-time discretization schme cGP-C1(3)-$\widetilde{Q}_3^n$ with the lower order schemes cGP(2)-$\widetilde{Q}_2^n$ and cGP(1)-Q_1, respectively, where $\widetilde{Q}_3^n$, $\widetilde{Q}_2^n$ and Q_1 indicate the used finite element spaces V_h which are the nonparametric nonconforming spaces of order 3 and 2 and the usual conforming space of bilinear elements. To this end, we present in each of the Tables 10–12 the full discretization error $u - u_{h,\tau}$ of the corresponding scheme 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

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

For the scheme cGP-C1(3)-$\widetilde{Q}_3^n$, it can be seen from Table 10 that the full discretization error is reduced by a factor of 16 if we increase the level ℓ by one. That means we gain a factor of 16 in accuracy whereas the numerical cost in terms of the CPU-time increase only by a factor of 8 which is a big advantage for the higher order method. For the schemes cGP(2)-$\widetilde{Q}_2^n$ and cGP(1)-Q_1 we see in Table 11 and 12, respectively, that we gain only a factor of 8 and 4 whereas the CPU-time increases by a factor of 8.

If we ask, for instance, for a discrete solution with an accuracy of $2.5\,10^{-6}$ then we would need 66 s with the scheme cGP-C1(3)-$\widetilde{Q}_3^n$, 1165 s with the cGP(2)-$\widetilde{Q}_2^n$- and 14362 s with the cGP(1)-Q_1-method (see the underlined values in Table 10–12), i.e., the cGP-C1(3)-$\widetilde{Q}_3^n$-scheme is 217 times faster than the Crank-Nicolson scheme with bilinear elements and 17 times faster than the cGP(2)-$\widetilde{Q}_2^n$-method. The acceleration factors in the CPU-time of the higher order methods compared to the lower order ones are growing if we ask for a higher accuracy of the

	cGP(1)		cGP(2)		cGP-C1(3)	
$1/\tau$	$\|u-u_{h,\tau}\|_{2,L}$	CPU	$\|u-u_{h,\tau}\|_{2,L}$	CPU	$\|u-u_{h,\tau}\|_{2,L}$	CPU
10	3.58E-01	0.47	5.10E-02	18.82	6.13E-02	19.47
20	1.21E-01	12.76	8.87E-03	39.23	2.80E-03	40.62
40	3.22E-02	21.66	1.15E-03	77.96	1.73E-04	80.87
80	8.19E-03	47.47	1.45E-04	150.77	1.08E-05	156.03
160	2.05E-03	83.91	1.82E-05	308.22	6.78E-07	318.87
320	5.14E-04	194.63	2.28E-06	611.18		
640	1.29E-04	332.36	2.92E-07	1227.98		
1280	3.21E-05	718.33				
2560	8.03E-06	1213.70				
5120	2.01E-06	2468.65				
10240	5.06E-07	3859.79				

Table 9: Error norms $\|u-u_{h,\tau}\|_{2,L}$ and total CPU-times to achieve the accuracy of 10^{-6} on grid level $\ell = 6$.

ℓ	$1/\tau$	$\|u-u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
2	20	4.64E-03		96		0.06	
3	40	3.08E-04	15.10	336	3.50	0.56	9.79
4	80	1.95E-05	15.75	1248	3.71	5.65	10.13
5	160	1.23E-06	15.94	4800	3.85	65.54	11.59
6	320	7.67E-08	15.98	18816	3.92	642.22	9.80
7	640	4.79E-09	16.00	74496	3.96	5882.88	9.16

Table 10: Full discretization error, total number of unknowns and CPU-time for the cGP-C1(3)-$\widetilde{Q}_3^n$ scheme.

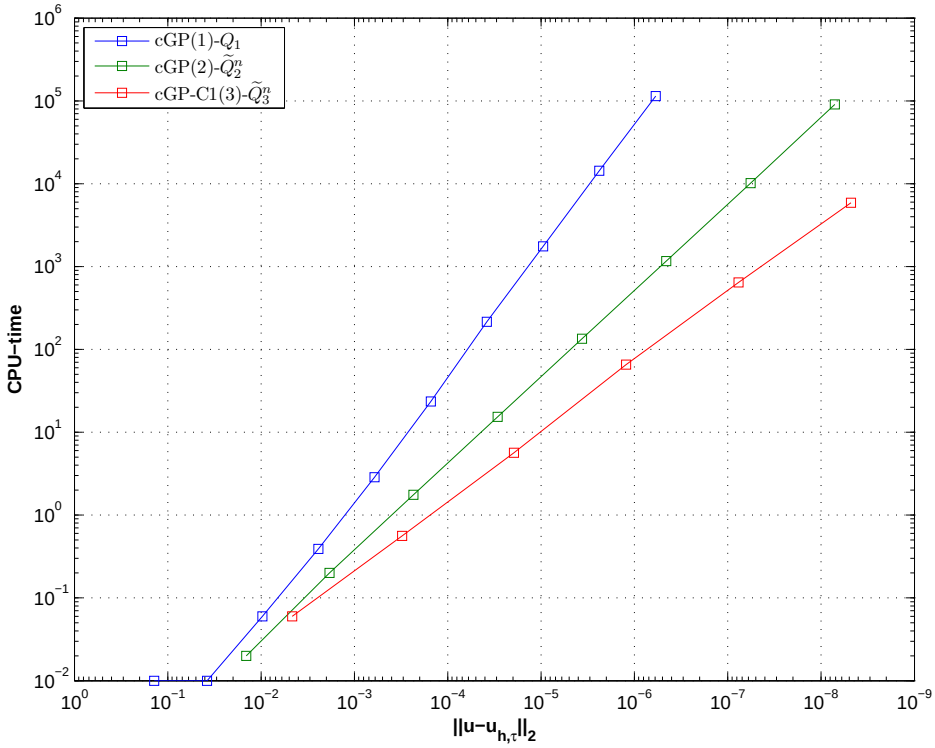
discrete solution. This can be seen in Figure 5 where we have plotted for all three methods the CPU-time against the accuracy.

ℓ	$1/\tau$	$\|u-u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
2	20	1.44E-02		56		0.02	
3	40	1.85E-03	7.75	192	3.43	0.20	8.50
4	80	2.33E-04	7.95	704	3.67	1.76	8.60
5	160	2.92E-05	7.99	2688	3.82	15.36	8.75
6	320	3.65E-06	8.00	10496	3.90	134.30	8.74
7	640	4.56E-07	8.00	41472	3.95	1165.47	8.68
8	1280	5.70E-08	8.00	164864	3.98	10163.34	8.72
9	2560	7.12E-09	8.00	657408	3.99	90764.43	8.93

Table 11: Full discretization error, total number of unknowns and CPU-time for the cGP(2)-$\widetilde{Q}_2^n$ scheme.

ℓ	$1/\tau$	$\|u-u_{h,\tau}\|_{2,L}$	Factor	#DOFs	Factor	CPU	Factor
2	20	1.40E-01		9		0.01	
3	40	3.79E-02	3.69	25		0.01	1.71
4	80	9.68E-03	3.92	81	3.24	0.06	5.00
5	160	2.43E-03	3.98	289	3.57	0.39	6.45
6	320	6.09E-04	3.99	1089	3.77	2.86	7.39
7	640	1.52E-04	4.00	4225	3.88	23.51	8.22
8	1280	3.81E-05	4.00	16641	3.94	215.11	9.15
9	2560	9.52E-06	4.00	66049	3.97	1750.89	8.14
10	5120	2.38E-06	4.00	263169	3.98	14362.14	8.20
11	10240	5.92E-07	4.01	1050625	3.99	114040.61	7.94

Table 12: Full discretization error, total number of unknowns and CPU-time for the cGP(1)-Q_1 scheme.



8 Conclusion

We have presented a new higher order continuous Galerkin-Petrov time discretization scheme, called cGP-C1(3)-method, in combination with a nonconforming $\widetilde{Q}_3^n$-finite element approximation in space for the numerical solution of the two dimensional heat equation. The numerical experiments show that the full discretization error in the L^2-norm is of order 4 with respect to the space mesh size and time step h and τ, respectively. The numerical cost for cGP-C1(3)-method are insignificantly higher than for the previously studied cGP(2)-method [4] since its solution can be obtained by a simple post-processing from the cGP(2)-solution. However, the solution of the new method is significantly more accurate than the cGP(2)-solution.

We have shown in numerical tests that our proposed multigrid solver is highly efficient and exhibits for general meshes a robust convergence behavior which is nearly independent of the space mesh size and the time step such that large time steps get also feasible with respect to the solver. As a future step, we will extend this combined cGP-C1(3)-$\widetilde{Q}_3^n$-scheme towards the nonstationary incompressible Navier-Stokes equations.

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; contract/grant number: SCHI 576/2-1, TU 102/35-1 and LC06052 by MSMT.

References

[1] D. Arnold, D. Boffi, and R.S. Falk. Approximation by quadrilateral finite elements. *Math. Comput.*, 17(239):909–922, 2002.

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

[3] J.P. Hennart, J. Jaffré, and J.E. Roberts. A constructive method for deriving finite elements of nodal type. *Numer. Math.*, 53(6):701–738, 1988.

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

[5] M. Köster, A. Ouazzi, F. Schieweck, S. Turek, and P. Zajac. New robust nonconforming finite elements of higher order. *Applied Numerical Mathematics*, 62(3):166–184, 2012.

[6] 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.

[7] G. Matthies and F. Schieweck. Nonconforming finite elements of higher order satisfying a new compatibility condition. *J. Numer. Math.*, 16(1):23–50, 2008.

[8] G. Matthies and F. Schieweck. Higher order variational time discretizations for nonlinear systems of ordinary differential equations. Preprint 23/2011, Otto-von-Guericke Universität Magdeburg, Fakultät für Mathematik, 2011.

[9] R. Rannacher and S. Turek. Simple nonconforming quadrilateral stokes element. *Numer. Methods Partial Differential Equations*, 8:97 – 111, 1992.

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

[11] 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.