Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 285 (2015) 83–101
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Adaptive time step control for higher order variational time discretizations applied to convection–diffusion–reaction equations

Naveed Ahmed[a], Volker John[a,b,*]

[a] *Weierstrass Institute for Applied Analysis and Stochastics, Leibniz Institute in Forschungsverbund Berlin e. V. (WIAS), Mohrenstr. 39, 10117 Berlin, Germany*

[b] *Free University of Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany*

Received 22 April 2014; received in revised form 28 October 2014; accepted 31 October 2014
Available online 13 November 2014

Abstract

Higher order variational time stepping schemes allow an efficient post-processing for computing a higher order solution. This paper presents an adaptive algorithm whose time step control utilizes the post-processed solution. The algorithm is applied to convection-dominated convection–diffusion–reaction equations. It is shown that the length of the time step properly reflects the dynamics of the solution. With respect to the performance (accuracy, efficiency), the variational time stepping schemes are compared with an adaptive Crank–Nicolson scheme, whose time step control relies on comparing two solutions computed with schemes of the same order.

Keywords: Continuous Galerkin–Petrov method; Discontinuous Galerkin method; Post-processed solution; Adaptive step length control; SUPG stabilization; Adaptive Crank–Nicolson scheme

1. Introduction

Adaptive time step control is a tool that might increase the efficiency of many simulations of problems from Computational Fluid Dynamics (CFD). There are several proposals in the literature for the way to control an adaptive time step. A simple approach monitors just the change of the solution in two subsequent discrete times, e.g., as applied in [1] to a semi-implicit Euler scheme for the Navier–Stokes equations. In more advanced methods, the time step control is based on comparing solutions computed with different time stepping schemes. A classical approach from the numerical solution of ordinary differential equations [2,3], the use of embedded schemes, was applied to the incompressible Navier–Stokes equations in [4]. Embedded schemes require only a post-processing step which can be performed very efficiently. One obtains the solution of a scheme with one order less than the originally used scheme and in this way an estimate of the error for the solution of the lower order method. However, the application

* Corresponding author at: Weierstrass Institute for Applied Analysis and Stochastics, Leibniz Institute in Forschungsverbund Berlin e. V. (WIAS), Mohrenstr. 39, 10117 Berlin, Germany.
E-mail addresses: naveed.ahmed@wias-berlin.de (N. Ahmed), volker.john@wias-berlin.de (V. John).

http://dx.doi.org/10.1016/j.cma.2014.10.054
0045-7825/

of embedded schemes is only possible for higher order time stepping schemes. Such schemes do not seem to be popular in the CFD community. Most often, one finds the use of first and second order schemes in the literature. For the Crank–Nicolson scheme, which is of second order, there are two proposals for controlling the time step on the basis of applying another second order scheme and then to estimate the truncation error. In [5], the other second order scheme is the fractional-step θ-scheme and in [6], the use of the explicit Adams–Bashforth scheme was studied. The approach from [5] has a high computational effort. Applying an explicit scheme reduces the computational cost drastically, but the issue of a CFL condition arises. The Adams–Bashforth approach was studied in [6] for one-dimensional convection–diffusion equations, its use for two-dimensional convection–diffusion equations can be found, e.g., in [7,8]. A slight modification of this scheme will be considered also in the present paper.

This paper studies higher order variational time discretizations, namely continuous Galerkin–Petrov (cGP(k), $k \in \{2, 3\}$) and discontinuous Galerkin (dG(k), $k \in \{1, 2\}$) methods. As already mentioned above, the use of higher order time stepping schemes does not seem to be popular for applications from CFD. There might be various reasons for this situation, among them are certainly the higher implementation effort and higher numerical costs. However, there are studies which show that the application of higher order time stepping schemes might give much more accurate results than the use of simple schemes, e.g., see [4,9]. In variational time stepping schemes, the temporal derivative is treated in a finite element way. To this end, one takes finite element functions which depend on space and time, makes the ansatz that the discrete solution can be represented with these functions, integrates the equation in space and time and applies the usual integration by parts in space. In the definition of the time stepping scheme, the test space is taken to be discontinuous in time, at the discrete times. This choice enables the performance of a standard time marching algorithm and it avoids the solution of a global system in space and time as in space–time finite element methods. In the discontinuous Galerkin method, ansatz and test space coincide. The application of this method in parabolic problems can be found, e.g., in [10,11]. It turns out that jump terms at the discrete times appear in this discretization. Using instead continuous-in-time ansatz functions, the jump terms are avoided. Besides calling this approach continuous Galerkin–Petrov method, one can find other names in the literature, like continuous Galerkin method [12] or discontinuous Galerkin–Petrov method [13]. Numerical studies with cGP(k) and dG(k) for convection-dominated convection–diffusion equations can be found in [14]. In these studies, an equidistant time step was used and two stabilization techniques for the spatial discretization were investigated. A super-convergence of the error in the l^∞ norm was observed. A space–time adaptive method for higher order variational time discretizations in the context of incompressible Navier–Stokes equations was presented in [15]. This method uses goal-oriented error estimation techniques for controlling the adaptivity. Algorithmic aspects for higher order variational time discretizations have been investigated recently in [16].

Low order variational time discretizations lead to well known methods, e.g., dG(0) is the implicit Euler scheme and cGP(1) the Crank–Nicolson scheme. An obvious drawback of higher order variational temporal discretizations is their high numerical effort: one has to solve not only a number of scalar problems in each time step but even a coupled system of problems. For cGP(k), a clever construction was proposed in [13] such that the coupling becomes much weaker than in the original method. But the coupling cannot be avoided completely.

The higher computational cost per time step of the variational time discretizations can be compensated if, for a given problem, only the necessary number of time steps, for achieving a prescribed accuracy, is applied. For obtaining a small number of time steps, usually the length of the time step has to vary such that an adaptive time step control is necessary. With an adaptive time step control, also the high order of the methods can be exploited best. Fortunately, a post-processing procedure was proposed in [17] that allows to compute a solution of one order higher than the original method in the $L^2(L^2)$ time–space norm, thus an estimate of the error for the original method can be obtained. For the cGP(k) methods, the post-processing from [17] requires the solution of a linear system of equations, which is inexpensive compared with one step of cGP(k). The availability of two solutions with different order enables also the application of well understood techniques from the numerical analysis of ordinary differential equations for controlling the adaptive time step, e.g., see [18]. Thus, in contrast to commonly used time stepping schemes in CFD, higher order variational time stepping schemes enable the possibility of applying the most standard way of adaptive time step control. However, to the best of our knowledge, the realization and numerical assessment of this approach cannot be found so far in the literature.

The goals of this paper are twofold. First, it will be shown that the adaptive time step control for higher order variational time stepping schemes, which is based on the post-processed solution, leads indeed to lengths of the time steps that properly reflect the dynamics of the solution. Besides establishing that the approach works principally, the

question arises in which situations its usage in practice can be recommended, in particular from the point of view of efficiency. Second, the performance of the studied methods will be compared with the performance of (a slight modification of) the adaptive Crank–Nicolson scheme proposed in [6], which is based on controlling the time step with the solution obtained with the Adams–Bashforth scheme.

The paper is organized as follows. Section 2 introduces the problem and describes its discretization in space. The higher order variational time stepping schemes are presented in Section 3. Then, the post-processing and the adaptive time step control are discussed in Section 4. Section 5 presents the numerical studies and their results are summarized in Section 6.

2. The model problem and its discretization in space

Consider the scalar convection–diffusion–reaction equation: Find $u : (0, T] \times \Omega \to \mathbb{R}$ such that

$$\begin{aligned} &\partial_t u - \varepsilon \Delta u + \boldsymbol{b} \cdot \nabla u + \sigma u = f \quad \text{in } (0, T) \times \Omega, \\ &u = 0 \qquad\qquad\qquad\qquad\quad \text{in } (0, T) \times \partial\Omega, \\ &u(\cdot, 0) = u_0 \qquad\qquad\qquad\quad \text{in } \Omega. \end{aligned} \tag{1}$$

Here, $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, is a polygonal or polyhedral domain with Lipschitz boundary $\partial\Omega$. Furthermore, $0 < \varepsilon \ll 1$ is a diffusivity constant, $\boldsymbol{b}(t, \boldsymbol{x})$ is the flow velocity, $\sigma(t, \boldsymbol{x})$ is the reaction coefficient, and $f(t, \boldsymbol{x})$ is a given outer source of the unknown scalar quantity u. It will be assumed that either $\nabla \cdot \boldsymbol{b}(t, \boldsymbol{x}) = 0$ and $\sigma(t, \boldsymbol{x}) \geq 0$, or that there exists a positive constant σ_0 such that

$$\sigma(t, \boldsymbol{x}) - \frac{1}{2} \operatorname{div} \boldsymbol{b}(t, \boldsymbol{x}) \geq \sigma_0 > 0 \quad \forall (t, \boldsymbol{x}) \in [0, T] \times \overline{\Omega},$$

which are standard assumptions for equations of type (1).

For the finite element discretization, (1) is transformed into a variational formulation. To this end, consider the space $V := H_0^1(\Omega)$, its dual space $H^{-1}(\Omega)$, and $\langle \cdot, \cdot \rangle$ as the duality pairing between these spaces. The inner product in $L^2(\Omega)$ is denoted by $(\cdot, \cdot)$.

A function u is a weak solution of problem (1) if $u \in L^2\left(0, T; H_0^1(\Omega)\right)$ and $\partial_t u \in L^2\left(0, T; H^{-1}(\Omega)\right)$, with

$$\langle \partial_t u(t), v \rangle + a(u(t), v) = \langle f(t), v \rangle \quad \forall v \in V \tag{2}$$

for almost all $t \in (0, T]$ and $u(0) = u_0$, where the bilinear form a is given by

$$a(u, v) := \varepsilon(\nabla u, \nabla v) + (\boldsymbol{b} \cdot \nabla u, v) + (\sigma u, v).$$

Let $\{\mathcal{T}_h\}$ denote a family of shape regular triangulations of Ω into compact d-simplices, quadrilaterals, or hexahedra such that $\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} K$. The diameter of $K \in \mathcal{T}_h$ will be denoted by h_K and the mesh size h is defined by $h := \max_{K \in \mathcal{T}_h} h_K$. Let $V_h \subset V$ be a finite element space defined on $\mathcal{T}_h$.

The standard Galerkin method in space applied to (2) consists in finding $u_h \in H^1(0, T; V_h)$ such that $u_h(0) = u_{h,0}$ and for almost all $t \in (0, T]$

$$(\partial_t u_h(t), v_h) + a\left(u_h(t), v_h\right) = (f(t), v_h) \quad \forall v_h \in V_h, \tag{3}$$

where $u_{h,0} \in V_h$ is a suitable approximation of u_0 and $f \in L^2(\Omega)$ was assumed for simplicity of notation. In the convection-dominated case, the standard Galerkin formulation (3) is inappropriate since the discrete solution is usually globally polluted by spurious oscillation, unless the mesh parameter is sufficiently small.

One of the most efficient stabilized methods is the Streamline-Upwind Petrov–Galerkin (SUPG) method [19,20] that is frequently used due to its stability properties, its higher-order accuracy in appropriate norms, and its easy implementation, e.g., see [21]. In the time-continuous case, the SUPG stabilized semi-discrete problem reads as follows: Find $u_h \in H^1(0, T; V_h)$ such that $u_h(0) = u_{h,0}$ and for almost every $t \in (0, T]$

$$\begin{aligned} &(\partial_t u_h(t), v_h) + a_h\left(u_h(t), v_h\right) + \sum_{K \in \mathcal{T}_h} \delta_K \left(\partial_t u_h(t), \boldsymbol{b} \cdot \nabla v_h\right)_K \\ &\quad = (f(t), v_h) + \sum_{K \in \mathcal{T}_h} \delta_K \left(f(t), \boldsymbol{b} \cdot \nabla v_h\right)_K \end{aligned} \tag{4}$$

for all $v_h \in V_h$. The bilinear form $a_h(\cdot,\cdot)$ is defined by

$$a_h(u_h, v_h) := a(u_h, v_h) + \sum_{K\in\mathcal{T}_h} \delta_K \left(-\varepsilon\Delta u_h + \boldsymbol{b}\cdot\nabla u_h + \sigma u_h, \boldsymbol{b}\cdot\nabla v_h\right)_K ,$$

where $(\cdot,\cdot)_K$ denotes the inner product in $L^2(K)$ and $\{\delta_K\}$ denotes the set of local stabilization parameters. A theoretically supported choice of the stabilization parameters is an open question. Even for simple time stepping schemes, like the backward Euler scheme, one has in the general situation only a convergence proof for $\delta_K = \mathcal{O}(\tau)$, were τ is the length of the time step, whereas in special cases optimal estimates for $\delta_K = \mathcal{O}(h_K)$ were proved, see [22] for details. Since the small scales, which require a stabilization, are the spatial layers, we think that the latter choice is more appropriate. Numerical studies in [22] came also to this conclusion. Concretely, for the numerical studies presented below, the stabilization parameters were set to be $\delta_K = 0.25h_K$.

Any other linear stabilization which is based on a modification of the bilinear form, like continuous interior penalty (CIP) or local projection stabilization (LPS) schemes, can be applied within higher order variational time discretizations in the same way as the SUPG method, see [14]. For nonlinear stabilizations, like spurious oscillation at layers diminishing (SOLD) methods, a stabilization term might be discretized explicitly and then they can be used also in the same way. However, for stabilization methods which are based on modifications of matrices and vectors, like finite element flux-corrected transport (FEM-FCT) methods [23], the application of higher order variational time discretizations seems to be an open problem.

3. Temporal discretization with variational time stepping schemes

The main topic of this paper is a study of the continuous Galerkin–Petrov and discontinuous Galerkin time stepping schemes. These schemes are basically the same as described in [14]. To keep this paper self-containing, a brief presentation of the schemes, which provides the basic ideas, will be given here.

Consider a partition $0 = t_0 < t_1 < \cdots < t_N = T$ of the time interval $I := [0, T]$ and set $I_n := (t_{n-1}, t_n]$, $\tau_n := t_n - t_{n-1}$, $n = 1, \ldots, N$, and $\tau := \max_{1\le n\le N} \tau_n$. For a given non-negative integer k, the fully discrete time-continuous and time-discontinuous spaces, respectively, are defined as follows:

$$X_k := \left\{u \in C(I, V_h) : u|_{I_n} \in \mathbb{P}_k(I_n, V),\ n = 1, \ldots, N\right\},$$
$$Y_k := \left\{u \in L^2(I, V_h) : u|_{I_n} \in \mathbb{P}_k(I_n, V),\ n = 1, \ldots, N\right\},$$

where

$$\mathbb{P}_k(I_n, V_h) := \left\{u : I_n \to V_h : u(t) = \sum_{j=0}^{k} U^j t^j,\ U^j \in V_h\ \forall j\right\}$$

denotes the space of V_h-valued polynomials of order k in time. The functions in the space Y_k are allowed to be discontinuous at the nodes t_n, $n = 1, \ldots, N-1$. For such functions, the left-sided value u_n^-, right-sided value u_n^+, and the jump $[u]_n$ are defined by

$$u_n^- := \lim_{t\to t_n - 0} u(t), \qquad u_n^+ := \lim_{t\to t_n + 0} u(t), \qquad [u]_n := u_n^+ - u_n^-.$$

The cGP(k) method applied to (4) leads to a time marching scheme with the following problems: Find $u_{h,\tau}|_{I_n} \in \mathbb{P}_k(I_n, V_h)$ such that for all $v_h \in V_h$

$$\int_{I_n} \left[(\partial_t u_{h,\tau}(t), v_h) + a_h(u_{h,\tau}(t), v_h) + \sum_{K\in\mathcal{T}_h} \delta_K \left(\partial_t u_{h,\tau}(t), \boldsymbol{b}\cdot\nabla v_h\right)_K\right] \psi(t)\, dt$$
$$= \int_{I_n} \left[(f(t), v_h) + \sum_{K\in\mathcal{T}_h} \delta_K \left(f(t), \boldsymbol{b}\cdot\nabla v_h\right)_K\right] \psi(t)\, dt \quad \forall \psi \in \mathbb{P}_{k-1}(I_n),$$

with $u_{h,\tau}|_{I_1}(t_0) = u_{h,0}$ and $u_{h,\tau}|_{I_n}(t_{n-1}) := u_{h,\tau}|_{I_{n-1}}(t_{n-1})$ for $n \ge 2$. The functions ψ denote scalar basis functions which are zero on $I \setminus I_n$ and which are a polynomial of degree less than or equal to $(k-1)$ on I_n.

The $(k-1)$-point Gauss–Lobatto quadrature rule for the numerical integration of time integrals is applied, which is exact for polynomials of degree less than or equal to $(2k-1)$. In order to determine the local solution $u_{h,\tau}|_{I_n}$, it is represented by

$$u_{h,\tau}|_{I_n}(t) = \sum_{j=0}^{k} U_{n,h}^{j} \phi_{n,j}(t) \quad \forall t \in I_n,$$

with coefficients $U_{n,h}^{j} \in V_h,\ j = 0, \ldots, k$.

Denote by $\hat{t}_j$ and $\hat{\omega}_j,\ j = 0, \ldots, k$, the Gauss–Lobatto points and the corresponding quadrature weights on $[-1, 1]$, respectively. Furthermore, let $\hat{\phi}_j \in \mathbb{P}_k,\ j = 0, \ldots, k$, and $\hat{\psi}_j \in \mathbb{P}_{k-1}$ denote the Lagrange basis functions with respect to $\hat{t}_j,\ j = 0, \ldots, k$, and $\hat{t}_j,\ j = 1, \ldots, k$, respectively. The basis functions $\phi_{n,j} \in \mathbb{P}_k(I_n),\ j = 0, \ldots, k$, and $\psi_{n,j} \in \mathbb{P}_{k-1}(I_n),\ j = 1, \ldots, k$, are defined via an affine reference transformation

$$T_n : [-1, 1] \to \overline{I}_n, \qquad \hat{t} \mapsto t_{n-1} + \frac{\tau_n}{2}(\hat{t} + 1), \tag{5}$$

see [14].

Using the same setting as in [14], the following fully discrete coupled system of equations is derived: For $U_{1,h}^{0} = u_{h,0}$ and $U_{n,h}^{0} = U_{n-1,h}^{k}$ if $n \geq 2$ find the coefficients $U_{n,h}^{j} \in V_h,\ j = 1, \ldots, k$, such that

$$\begin{aligned} \sum_{j=0}^{k} \alpha_{i,j}^{c} \left[\left(U_{n,h}^{j}, v_h \right) + \sum_{K \in \mathcal{T}_h} \delta_K \left(U_{n,h}^{j}, \boldsymbol{b} \cdot v_h \right) \right] + \frac{\tau_n}{2} a_h(U_{n,h}^{i}, v_h) = \frac{\tau_n}{2} \left[(f(t_{n,i}), v_h) + \beta_i^{c} \left(f(t_{n,0}), v_h \right) \right] \\ + \frac{\tau_n}{2} \sum_{K \in \mathcal{T}_h} \delta_K \left[\left(f(t_{n,i}), \boldsymbol{b} \cdot \nabla v_h \right)_K + \beta_i^{c} \left(f(t_{n,0}), \boldsymbol{b} \cdot \nabla v_h \right)_K \right] \end{aligned} \tag{6}$$

for $i = 1, \ldots, k$ and for all $v_h \in V_h$, where $\alpha_{i,j}^{c}$ and β_i^{c} are defined by

$$\alpha_{i,j}^{c} := \hat{\phi}_j'(\hat{t}_i) + \beta_i^{c} \hat{\phi}_j'(\hat{t}_0), \qquad \beta_i^{c} := \hat{\omega}_0 \hat{\psi}_i(\hat{t}_0), \quad i = 1, \ldots, k,\ j = 0, \ldots, k,$$

see [17].

In the following, (6) is written as a linear algebraic block system. To this end, let $\varphi_i \in V_h,\ i = 1, \ldots, m_h$, be finite element basis functions of V_h and $\boldsymbol{u}_n^{j} \in \mathbb{R}^{m_h}$ denote the nodal vector of $U_{n,h}^{j} \in V_h$, such that

$$U_{n,h}^{j}(\boldsymbol{x}) = \sum_{i=1}^{m_h} \left(\boldsymbol{u}_n^{j} \right)_i \varphi_i(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega.$$

Furthermore, the mass matrix $M \in \mathbb{R}^{m_h \times m_h}$, the matrices $C_n^{j} \in \mathbb{R}^{m_h \times m_h}$ associated with the additional time derivative term, the stiffness matrices $A_n^{j} \in \mathbb{R}^{m_h \times m_h}$, and the discrete right-hand side vector $\boldsymbol{F}_n^{j} \in \mathbb{R}^{m_h}$ are given by

$$\begin{aligned} (M)_{i,k} &:= (\varphi_k, \varphi_i), \\ (C_n^{j})_{i,k} &:= \sum_{K \in \mathcal{T}_h} \delta_K \left(\varphi_k, \boldsymbol{b}(t_{n,j}) \cdot \nabla \varphi_i \right)_K, \\ (A_n^{j})_{i,k} &:= a_h(\varphi_k, \varphi_i), \\ (\boldsymbol{F}_n^{j})_i &:= \left(f(t_{n,j}), \varphi_i \right) + \sum_{K \in \mathcal{T}_h} \delta_K \left(f(t_{n,j}), \boldsymbol{b}(t_{n,j}) \cdot \nabla \varphi_i \right)_K. \end{aligned} \tag{7}$$

Then, the fully discrete problem in I_n (6) is equivalent to the following $k \times k$ block system: For given $\boldsymbol{u}_n^{0}$, find $\boldsymbol{u}_n^{j} \in \mathbb{R}^{m_h},\ j = 1, \ldots, k$, such that

$$\sum_{j=0}^{k} \alpha_{i,j}^{c} \left(M + C_n^{j} \right) \boldsymbol{u}_n^{j} + \frac{\tau_n}{2} A_n^{i} \boldsymbol{u}_n^{i} = \frac{\tau_n}{2} \left[\boldsymbol{F}_n^{i} + \beta_i^{c} \left(\boldsymbol{F}_n^{0} - A_n^{0} \boldsymbol{u}_n^{0} \right) \right], \quad i = 1, \ldots, k. \tag{8}$$

The finite element nodal vector $\boldsymbol{u}_n^0$ of the solution $u_{h,\tau}|_{I_{n-1}}$ is given either via the discrete initial condition $u_{h,0}$ for $n = 1$ or by $\boldsymbol{u}_n^0 = \boldsymbol{u}_{n-1}^k$ for $n \geq 2$.

The dG(k) method applied to (4) leads to the following problem in I_n: Given u_n^- with $u_0^- = u_{h,0}$, find $u_{h,\tau}|_{I_n} \in \mathbb{P}_k(I_n, V_h)$ such that for all $\psi \in \mathbb{P}_k(I_n)$

$$\begin{aligned}
&\int_{I_n} \left[\left(\partial_t u_{h,\tau}(t), v_h\right) + a_h\left(u_{h,\tau}(t), v_h\right) + \sum_{K\in\mathcal{T}_h} \delta_K \left(\partial_t u_{h,\tau}(t), \boldsymbol{b}\cdot\nabla v_h\right) \right] \psi(t)\, dt \\
&\quad + \left[\left([u_{h,\tau}]_{n-1}, v_{n-1}^+\right) + \sum_{K\in\mathcal{T}_h} \delta_K \left([u_{h,\tau}]_{n-1}, \boldsymbol{b}\cdot\nabla v_{n-1}^+\right) \right] \psi(t_{n-1}) \\
&= \int_{I_n} \left[(f(t), v_h) + \sum_{K\in\mathcal{T}_h} \delta_K \left(f(t), \boldsymbol{b}\cdot\nabla v_h\right)_K \right] \psi(t)\, dt \quad \forall v_h \in V_h. \qquad (9)
\end{aligned}$$

Here, the $(k+1)$-point right-sided Gauss–Radau quadrature formula is applied for the numerical evaluation of the integrals, which is exact for polynomials up to degree $2k$. Let $\hat{t}_j$ and $\hat{\omega}_j$, $j = 1, \ldots, k+1$, denote the points and weights for this quadrature formula on $[-1, 1]$, respectively. Using the representation of $u_{h,\tau}$

$$u_{h,\tau}|_{I_n}(t) := \sum_{j=1}^{k+1} U_{n,h}^j \phi_{n,j}(t),$$

where $U_{n,h}^j \in V_h$, $j = 1, \ldots, k+1$, and following [14], one obtains the following system of equations: Find the coefficients $U_{n,h}^j \in V_h$, $j = 1, \ldots, k+1$, such that

$$\begin{aligned}
&\sum_{j=1}^{k+1} \alpha_{i,j}^d \left[\left(U_{n,h}^j, v_h\right) + \sum_{K\in\mathcal{T}_h} \delta_K \left(U_{n,h}^j, \boldsymbol{b}\cdot\nabla v_h\right) \right] + \frac{\tau_n}{2} a_h\left(U_{n,h}^i, v_h\right) \\
&\quad = \beta_i^d \left[\left(U_{n,h}^0, v_h\right) + \sum_{K\in\mathcal{T}_h} \delta_K \left(U_{n,h}^0, \boldsymbol{b}\cdot\nabla v_h\right) \right] \\
&\qquad + \frac{\tau_n}{2} \left[\left(f(t_{n,i}), v\right) + \sum_{K\in\mathcal{T}_h} \delta_K \left(f(t_{n,i}), \boldsymbol{b}\cdot\nabla v_h\right)_K \right]
\end{aligned}$$

for $i = 1, \ldots, k+1$, and for all $v_h \in V_h$, where

$$\alpha_{i,j}^d := \hat{\phi}_j' + \beta_i^d \hat{\phi}_j(-1), \qquad \beta_i^d := \frac{1}{\hat{\omega}_i}\hat{\phi}_i(-1), \qquad U_{n,h}^0 = U_{n-1,h}^-.$$

In matrix–vector notation, the following $(k+1)\times(k+1)$ block system of the problem in I_n is derived: Find $\boldsymbol{u}_n^j \in \mathbb{R}^{m_h}$ for $j = 1, \ldots, k+1$, such that

$$\sum_{j=1}^{k+1} \alpha_{i,j}^d \left(M + C_n^j\right) \boldsymbol{u}_n^j + \frac{\tau_n}{2} A_n^i \boldsymbol{u}_n^i = \beta_i^d \left(M + C_n^0\right) \boldsymbol{u}_n^0 + \frac{\tau_n}{2} \boldsymbol{F}_n^i, \quad i = 1, \ldots, k+1,$$

where $\boldsymbol{u}_n^j$ denotes the nodal vector of $U_{n,h}^j \in V_h$. After having solved this system, one enters the next time interval and sets the initial value of I_{n+1} to $\boldsymbol{u}_{n+1}^0 := \boldsymbol{u}_n^{k+1}$.

4. Post-processing and adaptive time step control

Following [17], a higher order in time approximation can be obtained by means of a post-processing of the time-discrete solution $u_{h,\tau}$ with low computational costs for the cGP(k) and dG(k) methods. Note that this approach is not restricted to linear problems.

Let $u_{h,\tau}$ denote the solution of the cGP(k) method (6). The post-processed solution $\Pi u^n_{h,\tau}$ on the time interval I_n is given by

$$(\Pi u^n_{h,\tau})(t) = u_{h,\tau}(t) + a_n \zeta_n(t), \quad t \in I_n,$$

where

$$\zeta_n(t) = \frac{\tau_n}{2}\hat{\zeta}(\hat{t}), \quad \hat{t} := T_n^{-1}(t),$$

with T_n from (5). The polynomial $\hat{\zeta} \in \mathbb{P}_{k+1}$ vanishes in all Gauss–Lobatto points $\hat{t}_j$, $j = 0, \ldots, k$, and it is scaled such that $\hat{\zeta}'(1) = 1$. The nodal vector $\boldsymbol{\gamma}_n$ of the finite element function $a_n \in V_h$ is the solution of

$$\left(M + C_n^k\right)\boldsymbol{\gamma}_n = \boldsymbol{F}_n^k - A_n^k \boldsymbol{u}_n^k - \left(M + C_n^k\right)\boldsymbol{\eta}_n^k, \tag{10}$$

where $\boldsymbol{\eta}_n^k$ denotes the nodal representation of $u'_{h,\tau}(t_n) \in V_h$ and the matrices and the vector are given in (7). It has been shown in [17] that the post-processed solution $\Pi u^n_{h,\tau}(t)$ can be interpreted as the solution obtained with a time stepping scheme of order $(k+2)$, $k \geq 2$. Thus, computing $\Pi u^n_{h,\tau}(t)$ requires the solution of the linear system of Eqs. (10), where the system matrix is dominated by the mass matrix. A similar post-processing can be performed for the dG(k) method (9).

However, for dG(k) the higher order in time post-processed solution can be computed even simpler. The post-processed solution $\Pi u^n_{h,\tau}$ of the solution $u_{h,\tau}$ of (9) on the interval I_n can be represented as

$$(\Pi u^n_{h,\tau})(t) = u_{h,\tau}(t) + b_n \vartheta_n(t), \quad t \in I_n,$$

where

$$\vartheta_n(t) = \frac{\tau_n}{2}\hat{\vartheta}(\hat{t}), \quad \hat{t} := T_n^{-1}(t),$$

with T_n given in (5). The polynomial $\hat{\vartheta} \in \mathbb{P}_{k+1}$ is uniquely defined by $\hat{\vartheta}(\hat{t}_j) = 0$ for all Gauss–Radau points $\hat{t}_j$, $j = 1, \ldots, k+1$, and $\hat{\vartheta}'(1) = 1$. The finite element function $b_n \in V_h$ is obtained by

$$b_n := \frac{1}{\vartheta_n(t_{n-1})}\left(u^-_{n-1} - u^+_{n-1}\right),$$

i.e., it is just a scaled difference between the initial condition u^-_{n-1} at $t = t_{n-1}$ and the calculated solution u^+_{n-1} at $t = t_{n-1}$. Also in this case, it was proved in [17] that there is an interpretation of $\Pi u^n_{h,\tau}(t)$ as the solution obtained with a scheme of order $(k+2)$, $k \geq 1$. In numerical studies, we could observe that both post-processing techniques for dG(k) gave the same results.

Adaptive time step control aims at computing a numerical solution with a prescribed accuracy using as few time steps as possible. In particular for problems where the dynamics changes in $(0, T)$, the application of equidistant time steps is governed by the subintervals with the fastest dynamics, such that in other subintervals much more time steps might be performed than necessary. This paper studies if the higher order in time post-processed solution $\Pi u^n_{h,\tau}$ is an appropriate tool for controlling the length of the time steps. Since two solutions of different order are available, one can use well understood techniques, from the numerical simulation of ordinary differential equations, for controlling the length of the time step.

In the numerical simulations presented in this paper, the adaptive time step control is based on the $L^2(\Omega)$ norm of the error of the numerical solution and its post-processed solution

$$r_n := \|u^n_{h,\tau} - \Pi u^n_{h,\tau}\|_{L^2(\Omega)},$$

which is a standard criterion. The use of other quantities of interest will be subject of future studies. There are several proposals in the literature for determining a new time step, so-called controllers, e.g., see [18]. In the numerical simulations presented in Section 5, generally the PC11 controller

$$\tau^*_{n+1} = \theta \left(\frac{\mathrm{TOL}\, r_n}{r^2_{n+1}}\right)^{1/(k+1)} \frac{\tau_n^2}{\tau_{n-1}} \tag{11}$$

was used, e.g., as in the simulations of [4]. For comparison, some results obtained with the PID controller

$$\tau_{n+1}^{*} = \theta \left(\frac{\mathrm{TOL}}{r_{n+1}}\right)^{0.525/(k+1)} \left(\frac{r_n}{r_{n+1}}\right)^{0.225/(k+1)} \left(\frac{r_n^2}{r_{n+1} r_{n-1}}\right)^{0.03/(k+1)} \tau_n \tag{12}$$

will be presented, too. For $k = 2$, one obtains the parameter set which is proposed in [24,25], where the Crank–Nicolson scheme was used. In (11) and (12), θ is a safety factor which is introduced to reduce the probability of rejecting τ_{n+1}^{*}. In our simulations $\theta = 0.8$ was used. The parameter TOL determines the required accuracy of the numerical solution. The impact of TOL on the number of time steps will be studied in Section 5. Finally, to avoid a strong increase or decrease of subsequent time steps, the proposal for the next time step is computed with

$$\tau_{n+1} = \min\left\{s_{\max}\tau_n, \max\left(s_{\min}\tau_n, \tau_{n+1}^{*}\right)\right\},$$

which is known as the integral step size controller in the deterministic framework [2]. In our simulations, $s_{\min} = 0.1$ and $s_{\max} = 2$ were used. A step size τ_{n+1} is accepted if $r_{n+1} \leq \mathrm{TOL}$, otherwise it is rejected.

5. Numerical results

This section presents numerical studies of the proposed algorithm for the adaptive time step control. On the one hand, it will be shown that the adaptive time step control for higher order variational schemes works properly. And on the other hand, the performance of these schemes will be compared with the performance of (a slight modification of) an adaptive Crank–Nicolson scheme proposed in [6].

The adaptive Crank–Nicolson scheme was implemented as presented in [6, p. 2020]. In order to prevent the ringing phenomenon of the Crank–Nicolson scheme, a stabilization of the integrator was proposed in [6] which avoids performing additional computations. Let v_n be the update computed with the Crank–Nicolson scheme at time t_n, then there is a backward-oriented averaging $u_n = (u_n + u_{n-1})/2$ and the new solution is computed for $t_{n+1} = t_n + \tau_{n+1}/2$ with $u_{n+1} = u_n + \tau_{n+1} v_n/4$. It will be explained in Example 1 that we obtained wrong results with this approach. For this reason, a central averaging as proposed in [26]

$$u_n = \frac{\tau_n u_{n+1} + (\tau_n + \tau_{n+1}) u_n + \tau_{n+1} u_{n-1}}{2(\tau_n + \tau_{n+1})} \tag{13}$$

was applied in our simulations. After having computed (13), the time step τ_{n+1} is rejected and the adaptive procedure is continued with the averaged solution u_n at t_n. Thus the central averaging is performed at the cost of one time step rejection. With respect to the frequency of averaging, it is proposed in [6] to prescribe a target time length t_* ($t_* = 10^{-4}$ in [6]) after which the next averaging should be performed. For examples where the appropriate length of the time step varies considerably, as in Example 1, we found it difficult to choose a good value for t_*. Thus, besides prescribing t_*, also a minimal number n_* of accepted time steps in between two averaging steps was prescribed. The time step of the adaptive Crank–Nicolson scheme is controlled with a tolerance TOL which bounds an estimate of the local truncation error in the $L^2(\Omega)$ norm.

In the first example considered below, an adaptive time step control is advantageous because the convection field is time-dependent. Time-dependent convection fields are a common feature of problems from applications as the convection field is often a velocity field computed from the Navier–Stokes equations. The temporal variation in the second example results from a time-dependent boundary condition at the inlet. This feature reflects, e.g., changes of the temperature or the concentration of a species at the inlet during the studied process, which is also a typical feature in many applications.

The use of higher order discretizations in time should be combined with the application of higher order discretizations in space. In the studied examples, quadrilateral (2d) or hexahedral (3d) meshes were used with the Q_2 finite element. As already mentioned in Section 2, the SUPG method was utilized as stabilization. It is well known from numerical experience, e.g., in [27], that the SUPG method is not a perfect stabilization, mainly because of the appearance of spurious oscillations. However, it is certainly the most popular finite element stabilization such that its study is worthwhile in our opinion.

Our experience in [28] for simple time stepping schemes was that Krylov subspace methods with standard preconditioners work most efficiently for the solution of the arising linear systems of equations. A major reason

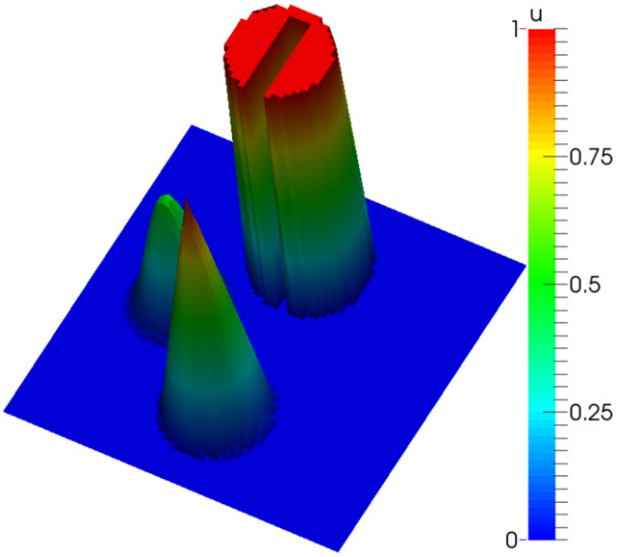

Fig. 1. Example 1. Initial condition.

is the availability of a good initial iterate. In the simulations for the variational time stepping schemes the restarted GMRES(50) [29] with a Jacobi preconditioner (without damping) was applied as solver for the block systems. The linear system for the post-processing (10) was solved with GMRES(50) and a SSOR preconditioner with relaxation parameter 1.5. The same solver was utilized in the adaptive Crank–Nicolson method. As initial iterate for the time stepping schemes, the solution of the previous discrete time was used. The initial iterate for solving the system for the post-processing was the solution of the variational time stepping schemes as given in Section 3. All iterations were stopped if the Euclidean norm of the residual vector was less than 10^{-10}. Usually the restart did not become effective.

The initial time step was set to be $\tau_1 = 10^{-8}$ in all simulations.

All simulations were performed with the code MooNMD [30] at a HP SL390s computer with Six-Core 3467 MHz Xeon processors.

Example 1 (*Time-Dependent Convection Field*)**.** This example is a generalization of the well known rotating body problem. It is defined in $\Omega = (0, 1)^2$, the diffusion is set to be $\varepsilon = 10^{-20}$. The other coefficients are $\sigma = 0$ and $f = 0$. Homogeneous Dirichlet boundary conditions are prescribed on $(0, T) \times \partial\Omega$. At the initial time, three disjoint bodies, are given, see Fig. 1. More precisely, for a given (x_0, y_0), set $r(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2}/r_0$. The center of the slotted cylinder is located at $(x_0, y_0) = (0.5, 0.75)$ and its shape is defined by

$$u(0; x, y) = \begin{cases} 1 & \text{if } r(x, y) \le 1,\ |x - x_0| \ge 0.0225 \text{ or } y \ge 0.85, \\ 0 & \text{otherwise.} \end{cases}$$

The hump at the left-hand side is given by $(x_0, y_0) = (0.25, 0.5)$ and

$$u(0; x, y) = \frac{1}{4}\Big(1 + \cos(\pi \min\{r(x, y), 1\})\Big),$$

and the conical body on the lower part is given by $(x_0, y_0) = (0.5, 0.25)$ and

$$u(0; x, y) = 1 - r(x, y).$$

The initial condition is zero outside the bodies. Finally, the convection field is defined by

$$\boldsymbol{b}(t, x, y) = \frac{1}{1 + 0.98 \cdot \sin(4t)} \begin{pmatrix} 0.5 - y \\ x - 0.5 \end{pmatrix}.$$

This field describes a rotation around $(0.5, 0.5)^T$ whose speed varies in time. In the standard rotating body problem, the rotation does not depend on time. Setting $T = 6.164546203$ leads to exactly five revolutions of the bodies.

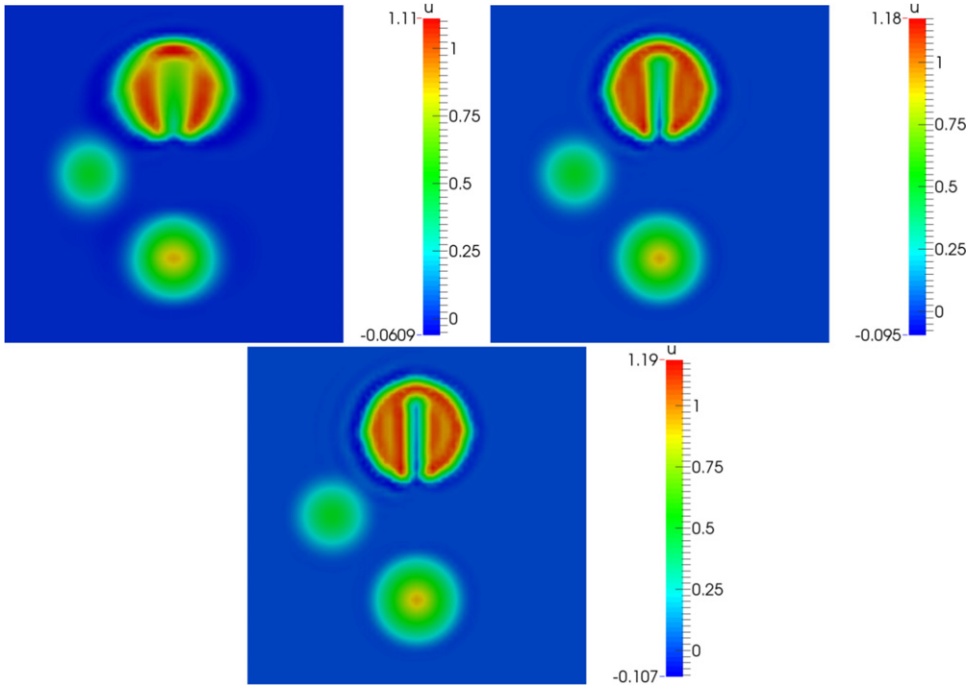

Fig. 2. Example 1. Solution at the final time with dG(1), left: TOL $= 10^{-2}$, right: TOL $= 5 \cdot 10^{-3}$, bottom: TOL $= 10^{-3}$.

Table 1
Example 1. Number of time steps with the variational time discretizations. Note that there were no rejections.

TOL	dG(1)	dG(2)	cGP(2)	cGP(3)
$5 \cdot 10^{-3}$	1 673	1 495	1 533	1 332
$1 \cdot 10^{-3}$	9 095	7 396	7 575	6 616
$5 \cdot 10^{-4}$	18 199	14 771	15 128	13 212

The simulations were performed on an equidistant quadrilateral grid with squares of edge length $h = 1/64$ such that there were 16 641 degrees of freedom (including Dirichlet nodes).

First, the behavior of the higher order variational time discretizations will be discussed. For the sake of brevity, only results with the PC11 controller (11) will be presented. With the PID controller (12), the computed solutions were similar for the same value of TOL (often a little bit more accurate), but the PID controller took notable more time steps. First of all, one had to find appropriate values for the parameter TOL in the PC11 controller (11). In our experience [4], appropriate values of TOL depend on the used temporal discretization. Choosing TOL too large introduced notable smearing, in particular in the region of the slotted cylinder. In Fig. 2 it can be seen that this situation occurs for dG(1) and TOL $= 10^{-2}$. For TOL $\leq 5 \cdot 10^{-3}$, there are only minor differences in the computed solutions. For all other methods and TOL $\leq 10^{-2}$ we obtained very similar solutions like for dG(1) and TOL $= 10^{-3}$, see the lower picture in Fig. 2. Note that the other methods are of higher order than dG(1). After having performed numerous simulations, we decided to present results with the same parameters TOL $\in \{5 \cdot 10^{-3}, 10^{-3}, 5 \cdot 10^{-4}\}$ for all studied methods. For the PID controller, it was found that appropriate parameters would be larger by a factor of 5–10.

For the chosen parameters, the evolution of the length of the time step is shown in Fig. 3. It can be observed that in all cases the time step reflects the speed of the rotation very well. A close look on the pictures and the numbers given in Table 1 shows that the higher the order of the method the less time steps were used. In our simulations, there were no rejections of proposed time steps for this example.

From Fig. 3 one can conclude that an accurate simulation of the time intervals with fast rotation requires a time step of around $\tau = 10^{-4}$. Thus, for using an equidistant time step, one has to choose this value. In Fig. 4 one can observe that there are no visible differences in the solutions obtained with this small equidistant time step and the solution

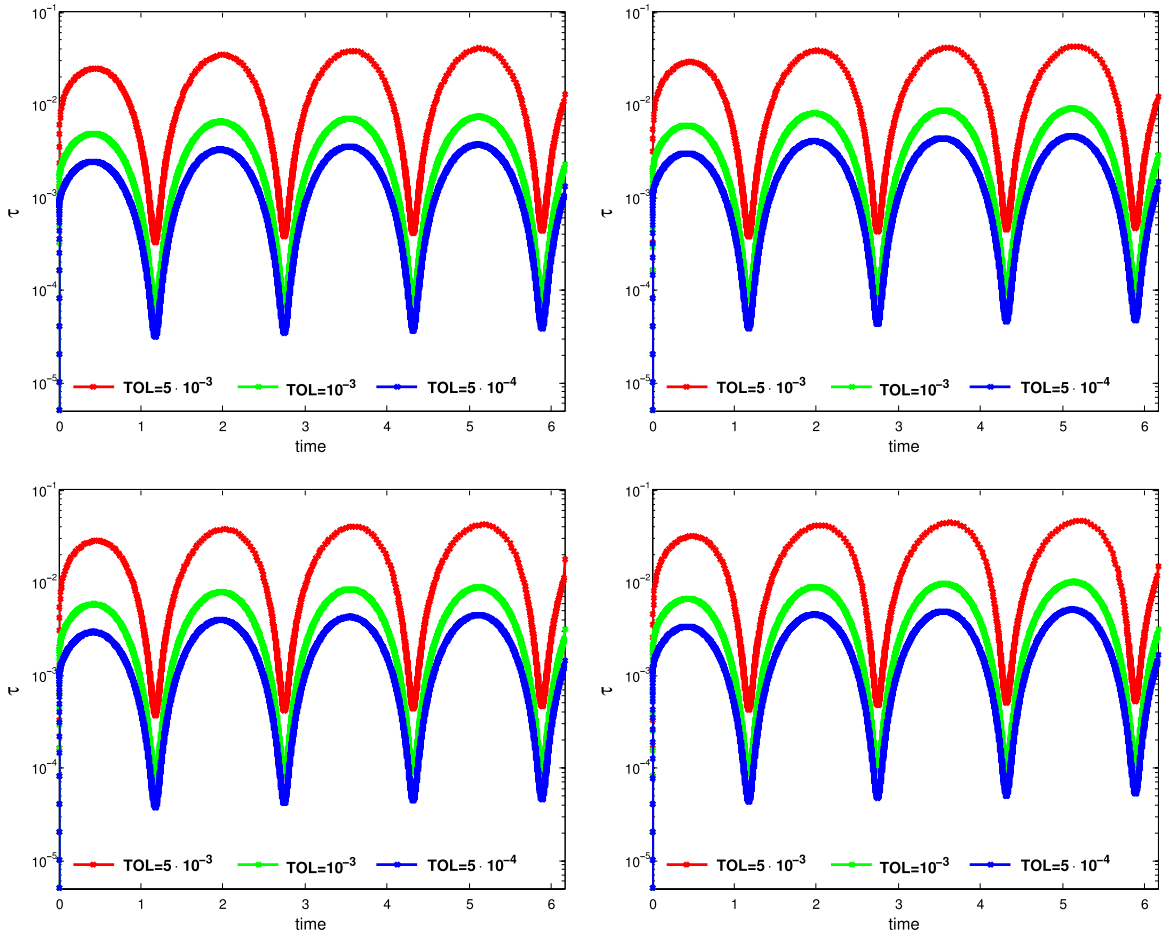

Fig. 3. Example 1. Temporal evolution of the length of the time step, dG(1), dG(2), cGP(2), cGP(3) (left to right, top to bottom).

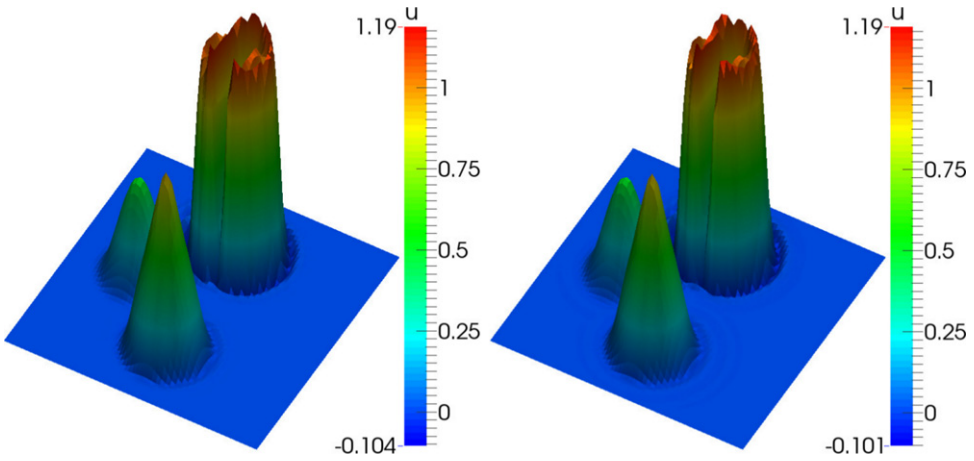

Fig. 4. Example 1. Solution at the final time, left: cGP(2) with adaptive time step and TOL $= 10^{-3}$ (1533 steps), right: cGP(2) with equidistant time step $\tau = 10^{-4}$ (61 646 steps).

Table 2
Example 1. Number of time steps (total, accepted, rejected, averaging among the rejected) for the adaptive Crank–Nicolson scheme, TOL $= 10^{-6}$.

n_*	$t_* = 10^{-3}$			$t_* = 10^{-4}$		
	Total	Acc.	Rej. (aver.)	Total	Acc.	Rej. (aver.)
10	16 020	11 121	4899 (1088)	15 981	10 928	4953 (1090)
20	16 060	12 236	3824 (600)	15 937	12 084	3853 (603)
50	16 672	13 400	3272 (263)	16 515	13 287	3238 (265)
100	15 690	13 356	2334 (131)	15 126	12 819	2307 (128)

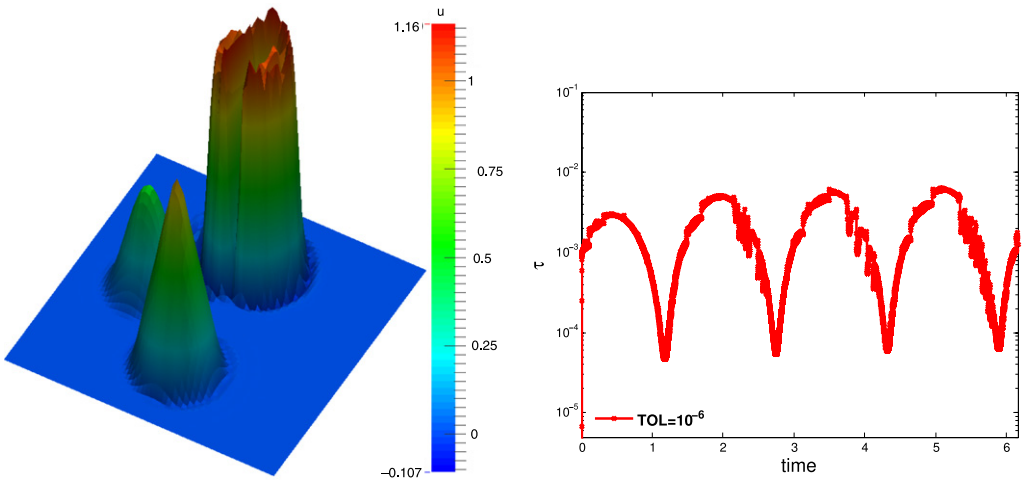

Fig. 5. Example 1. Solution at the final time and temporal evolution of the length of the time step for the adaptive Crank–Nicolson scheme with $t_* = 10^{-4}$ and $n_* = 100$.

computed with the adaptive time stepping algorithm (with only 2.5% of the number of time steps). Hence, in this respect, the adaptive algorithm works efficiently and accurately. As already mentioned above, the SUPG stabilization leads to spurious oscillations. In addition, some smearing, especially at the slotted cylinder, can be observed.

Applying the adaptive Crank–Nicolson scheme as propose in [6] to this example, i.e., with the backward-oriented averaging as explained at the beginning of this section, the bodies did not arrive at their correct positions at the final time, with large deviations from these positions. This behavior could be observed in all simulations. Thus, the backward-oriented averaging could not cope with the strongly changing dynamics of this example. Hence, we decided to replace this averaging with the central averaging (13). But even with the central averaging, the tolerance in the adaptive Crank–Nicolson scheme had to be chosen below 10^{-6} to obtain the exact positions of the bodies at the final time. For the tolerances 10^{-4}, 10^{-5}, $5 \cdot 10^{-6}$, there were still slight deviations, the rotation stopped less than two degrees too early, which could be also considered to be an acceptable accuracy. Altogether, our impression is that the averaging procedure for avoiding the ringing phenomenon still deteriorates the accuracy of the method to some extent. To have a comparison of solutions with the same accuracy, only results with TOL $= 10^{-6}$ will be presented and discussed.

The number of time steps is given in Table 2. It can be seen that for the total number of time steps one gets similar results for all choices of t_* and n_*. However, the choice of n_* influences the ratio of accepted and rejected steps considerably. With respect to this aspect, we prefer to choose larger values for n_*, e.g., $n_* \in \{50, 100\}$, since more accepted steps lead potentially to smaller time steps, which in turn might have a positive impact on the accuracy. For larger values of n_*, the additional cost for the central averaging is small, since the number of averaging steps is below 1.6% of the number of total steps. The evolution of the time step for one choice of the parameters is presented in Fig. 5. One can observe that the time step follows in the mean the dynamics of the problem, however there are a lot of small oscillations. In addition, there are some larger drops of the time step. The next averaging step increases the time step again. Since the occurrence of the next averaging step is determined mainly by n_*, large values of n_* lead potentially to more accepted small steps (after the drop of the time step).

Table 3
Example 1. Averaged computing times (in seconds) for solving the linear systems of equations and for performing one time step. The total times are for TOL $= 5 \cdot 10^{-3}$ for the higher order variational schemes and for $t_* = 10^{-4}$, $n_* = 100$ for the adaptive Crank–Nicolson scheme.

Method	dG(1)	dG(2)	cGP(2)	cGP(3)	CN/ (10)
Linear systems	0.22	1.36	0.26	1.08	0.025
Time/step	0.39	1.62	0.43	1.31	0.09
Total time	657	2433	659	1750	1355

Table 3 provides information on computing times for the studied methods: the averaged computing time for solving the linear systems of equations, the averaged time per time step, and total computing times for some cases where solutions with comparable accuracy were computed. The times for solving the linear systems and the times per time step were similar for all considered values of TOL and the values in Table 3 are also averages with respect to TOL. First of all, it can be seen that the overhead for computing the post-processed solution in the variational schemes is small. It is about 6% for dG(1) and cGP(2), with respect to the time per time step, whereas it was less than 2% for the higher order methods dG(2) and cGP(3). Using the alternative strategy for computing the post-processed solution for the dG(k) methods resulted practically in a negligible overhead. Thus, the savings in the number of time steps were much more important than the overhead of the adaptive time step control algorithm.

Among the variational time integrators, cGP(2) shows in our opinion the best combination of accuracy and efficiency. In this example, where the temporal dynamics strongly changes, the application of cGP(2) turned out to be more efficient than using the adaptive Crank–Nicolson scheme for computing a solution with comparable accuracy. The smoother evolution of the length of the time step, using the control with the post-processed solution for cGP(2), together with the higher order of cGP(2) resulted in considerably less needed time steps compared with the adaptive Crank–Nicolson scheme. These savings more than compensate the five times higher computational costs per time step.

Example 2 (*Time-Dependent Inlet Condition*)**.** This three-dimensional example was proposed in [31]. Given $\Omega = (0, 1)^3$, a species enters the domain at some inlet and it leaves the domain at the opposite side of the domain. While transported through the domain, the species is diffused somewhat and in the subregion where the species is transported, also a reaction occurs. The convection field points from the center of the inlet to the center of the outlet and it is not parallel to the coordinate axes.

Concretely, the inlet is located at $\{0\} \times (5/8, 6/8) \times (5/8, 6/8)$ and the position of the outlet is given by $\{1\} \times (3/8, 4/8) \times (4/8, 5/8)$. The convection field is prescribed by $\boldsymbol{b} = (1, -1/4, -1/8)^T$, the diffusion is given by $\varepsilon = 10^{-6}$, and the reaction by

$$\sigma(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \|\mathbf{x} - g\|_2 \leq 0.1, \\ 0 & \text{else,} \end{cases}$$

where g is the line through the center of the inlet and the center of the outlet and $\|\cdot\|_2$ denotes the Euclidean norm. The given ratio of diffusion and convection is typical in many applications. The boundary condition at the inlet is prescribed by

$$u_{\text{in}}(t) = \begin{cases} \sin(\pi t/2) & \text{if } t \in [0, 1], \\ 1 & \text{if } t \in (1, 2], \\ \sin(\pi(t-1)/2) & \text{if } t \in (2, 3]. \end{cases}$$

Homogeneous Neumann boundary conditions are set at the outlet and homogeneous Dirichlet conditions at the rest of the boundary. There are no sources, i.e., $f = 0$. The initial condition is set to be $u_0(\boldsymbol{x}) = 0$. In the time interval $(0, 1)$, the inflow is increasing and the injected species is transported towards the outlet. Then, in $(1, 2)$, there is a constant inflow and the species reaches the outlet. At the end of this time interval, there is almost a steady-state solution. Finally, in $(2, 3)$, the inflow decreases.

The simulations were performed on an equidistant hexahedral grid with the mesh width $h = 1/32$, leading to 274 625 degrees of freedom (including Dirichlet nodes).

Also for this example, we found that TOL $\in \{5 \cdot 10^{-3}, 10^{-3}, 5 \cdot 10^{-4}\}$ are appropriate parameters to be used in the PC11 controller (11). The evolution of the length of the time step is presented in Fig. 6. Starting with a small time

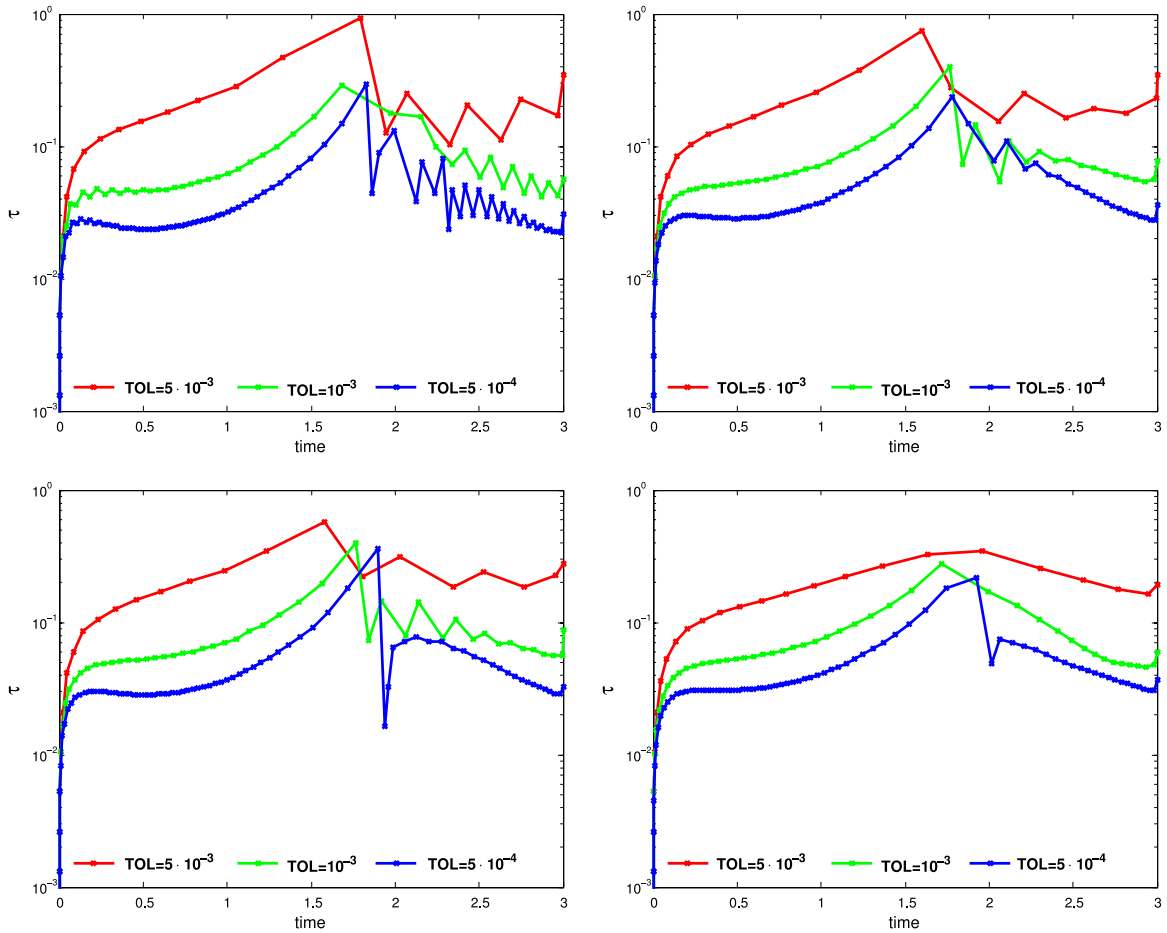

Fig. 6. Example 2. Temporal evolution of the length of the time step with the PC11 controller (11), dG(1), dG(2), cGP(2), cGP(3) (left to right, top to bottom).

Table 4
Example 2. Number of effective and rejected time steps with the PC11 controller (11).

TOL	dG(1)		dG(2)		cGP(2)		cGP(3)	
$5 \cdot 10^{-3}$	40	1	40	1	39	1	40	0
$1 \cdot 10^{-3}$	65	1	65	1	64	1	63	0
$5 \cdot 10^{-4}$	105	2	90	1	92	1	89	1

step, the time step increases in the time interval (1, 2). In particular, at the end of this interval, where the solution is nearly steady-state, it becomes comparably large. But it can be clearly seen that the length of the time step drops at $t = 2$, due to the change of the inlet condition. Thus, the evolution of the length of the time step reflects the dynamics of the problem well. It can be observed that the time step is oscillating in (2, 3) for some methods.

Detailed information on the needed number of time steps is provided in Table 4. Clearly, the number of time steps increases with decreasing parameter TOL. The second order method dG(1) needed few steps more than the higher order methods for small values of TOL.

As a measure of accuracy, the value of the solution (amount of species) at the center of the outlet was proposed in [28]. It can be observed in Fig. 7 that all simulations gave very similar results. For TOL $\leq 10^{-3}$, they are very close to the result obtained when applying the small equidistant time step $\tau = 0.01$.

For this examples, also results obtained with the PID controller are presented, see Table 5 and Figs. 8 and 9, since there are some qualitative differences to the results obtained with the PC11 controller. First, there were no rejections

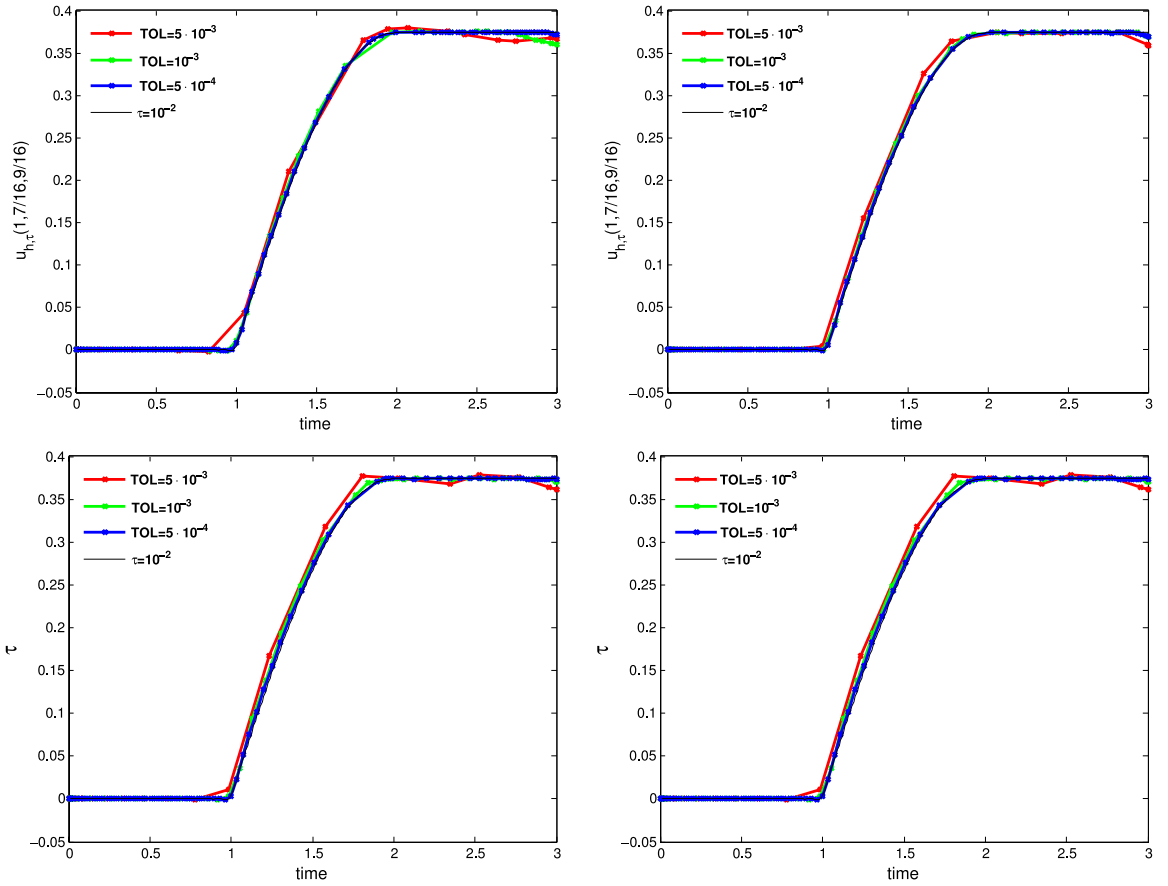

Fig. 7. Example 2. Temporal evolution of the amount of species at the center of the outlet, PC11 controller (11), dG(1), dG(2), cGP(2), cGP(3) (left to right, top to bottom).

Table 5
Example 2. Number of effective and rejected time steps with the PID controller (12).

TOL	dG(1)		dG(2)		cGP(2)		cGP(3)	
$5 \cdot 10^{-3}$	48	0	44	0	43	0	44	0
$1 \cdot 10^{-3}$	105	0	82	0	81	0	75	0
$5 \cdot 10^{-4}$	181	0	130	0	129	0	110	0

for the PID controller and second, the evolution of the time step for $t \geq 2$ was much smoother. For the same value of TOL, the time step control with the PID controller proposed more time steps than with the PC11 controller. In turn, the results for the PID controller and TOL $= 5 \cdot 10^{-3}$ were more accurate than the results obtained with the PC11 controller.

Results for the adaptive Crank–Nicolson scheme are presented in Table 6 and Fig. 10. One can see that the evolution of the solution at the center of the outlet becomes close to the reference curve for TOL $= 10^{-5}$ and the numerical results and the reference curve are almost indistinguishable for smaller values of TOL. Thus, an appropriate parameter TOL for the adaptive Crank–Nicolson scheme is also in this example smaller than for the higher order variational time integrators. Perhaps, this situation can be generally expected, since the tolerances are compared with the difference of two solutions and the difference of the solutions from two second order schemes can be anticipated to be smaller than the difference of two solutions obtained with methods of different order. In comparison with Example 1, the dynamics of Example 2 does not change so strongly. Table 6 shows that there is however a certain amount of rejected steps and

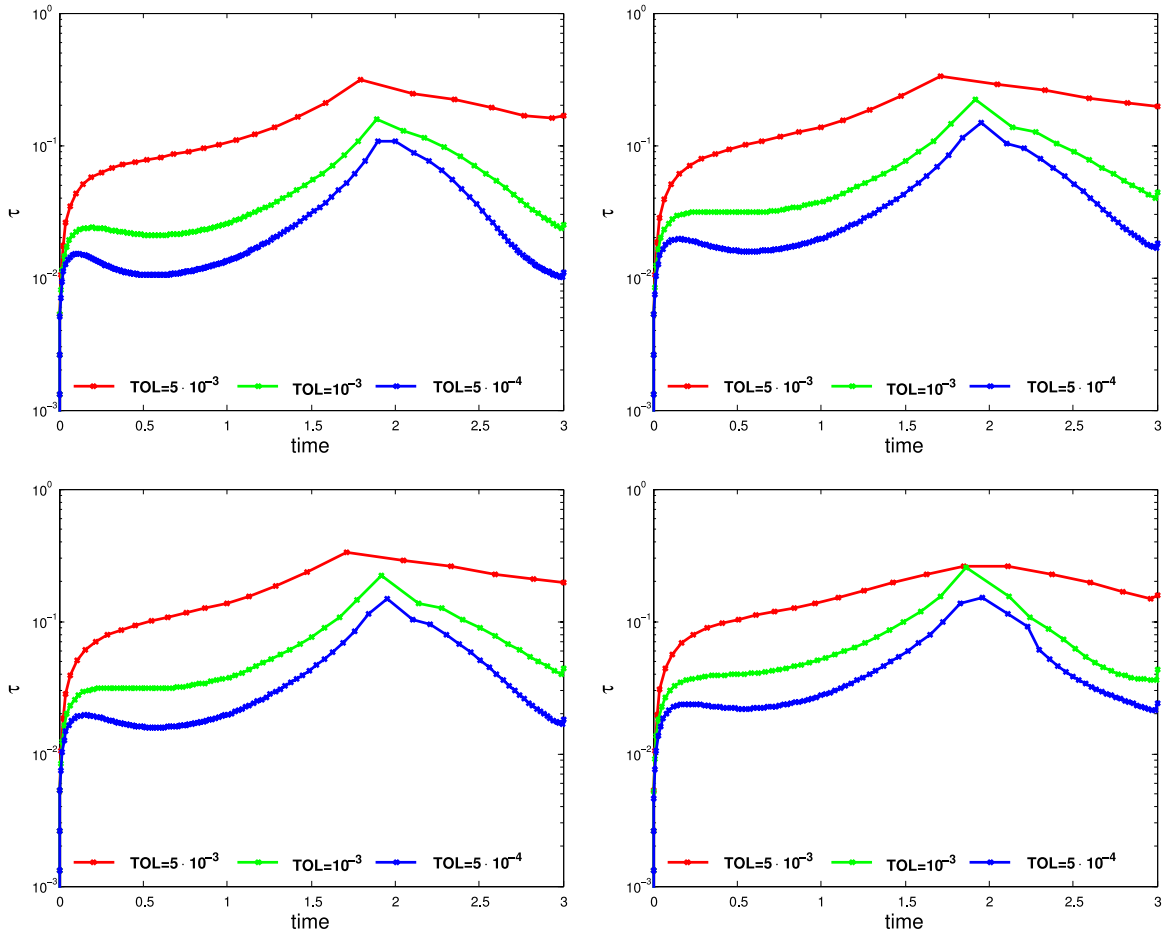

Fig. 8. Example 2. Temporal evolution of the length of the time step with the PID controller (12), dG(1), dG(2), cGP(2), cGP(3) (left to right, top to bottom).

Table 6
Example 2. Number of time steps (total, accepted, rejected, averaging among the rejected) for the adaptive Crank–Nicolson scheme, $t_* = 10^{-4}$.

n_*	TOL $= 10^{-5}$			TOL $= 5 \cdot 10^{-6}$		
	Total	Acc.	Rej. (aver.)	Total	Acc.	Rej. (aver.)
10	167	120	47(11)	205	157	48(15)
20	154	133	21(6)	195	167	28(8)
50	155	144	11(2)	222	203	19(4)
100	193	181	12(1)	263	239	24(2)

that the number of averaging steps is small. The evolution of the length of the time step follows in mean the dynamics of the problem, see Fig. 10, with some jumps after having applied the averaging procedure.

For assessing the efficiency, information concerning averaged computing times are given in Table 7. For the variational time stepping schemes, one can also in 3d observe that the computation of the post-processed solution imposed an almost negligible overhead. As already mentioned, cGP(2) offers in our opinion the best combination of accuracy and efficiency. Like in the 2d example, one step of cGP(2) took around five times as long as one step of the adaptive Crank–Nicolson scheme. Considering results with comparable accuracy, it turns out that in Example 2, where the temporal dynamics is comparably smooth, the adaptive Crank–Nicolson scheme was more efficient than the variational time stepping schemes.

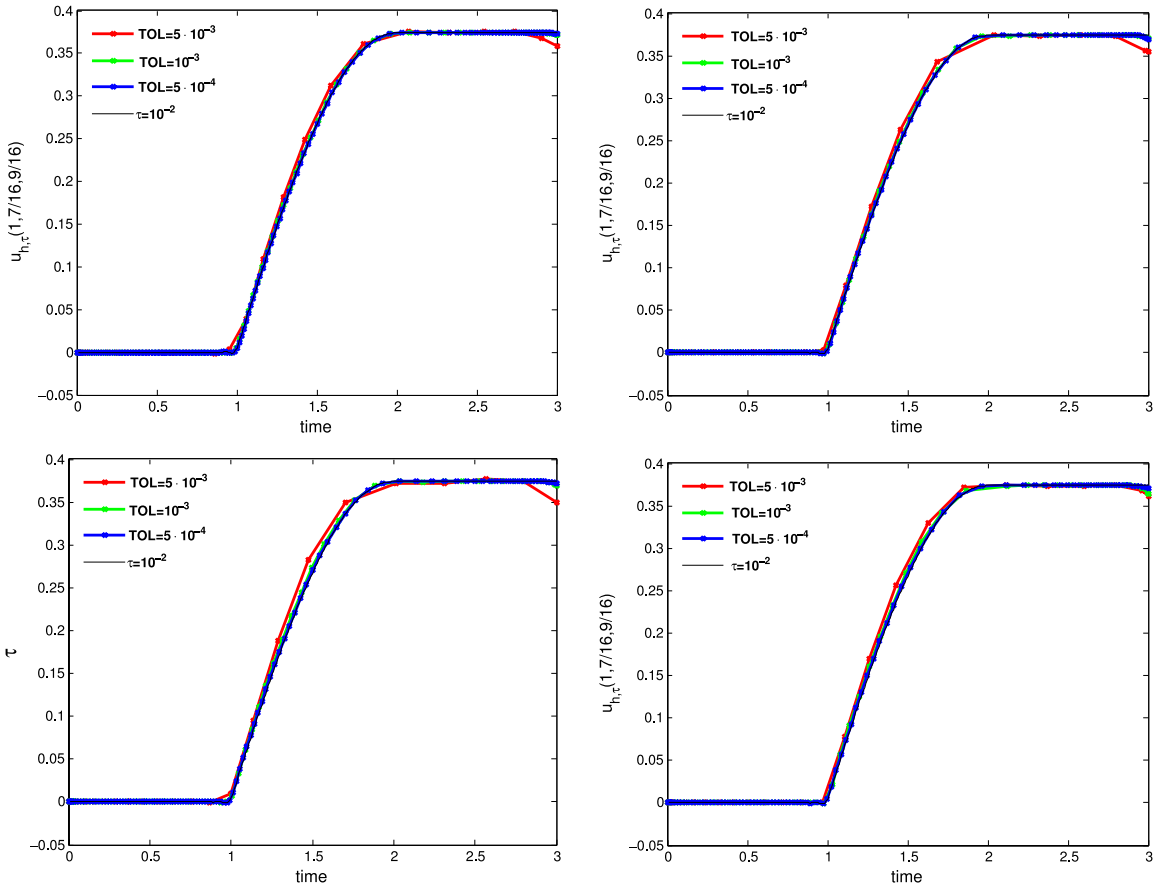

Fig. 9. Example 2. Temporal evolution of the amount of species at the center of the outlet, PID controller (12), dG(1), dG(2), cGP(2), cGP(3) (left to right, top to bottom).

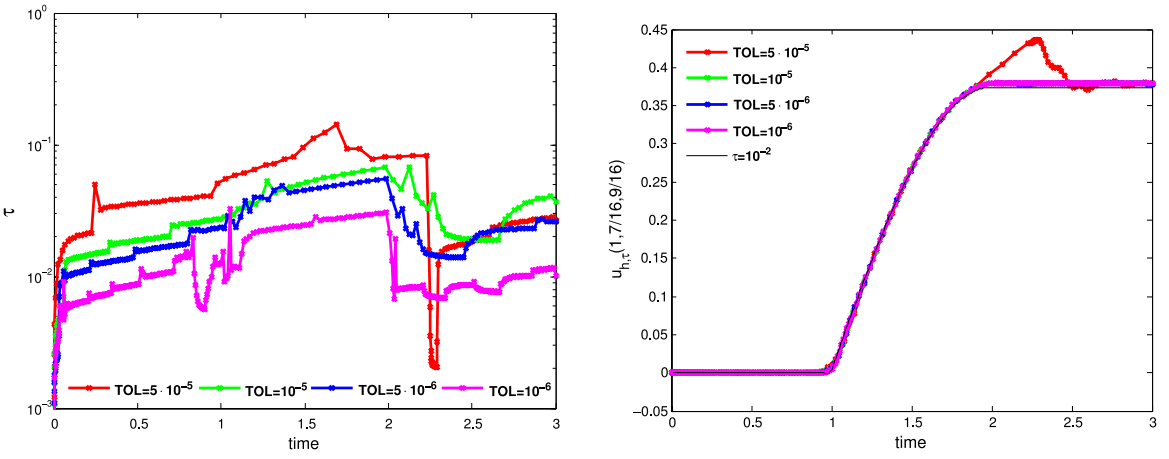

Fig. 10. Example 2. Temporal evolution of the length of the time step (left) and the amount of species at the center of the outlet (right) for the adaptive Crank–Nicolson scheme with TOL $= 10^{-5}$, $t_* = 10^{-4}$, and $n_* = 20$.

6. Summary

A method for the adaptive time step control in higher order variational time discretizations was proposed and studied numerically. This method was applied in the context of convection-dominated convection–diffusion–reaction

Table 7
Example 2. Total computing times and averaged computing times (in seconds) for solving the linear systems of equations and for performing one time step. The total time for the adaptive Crank–Nicolson scheme is for TOL $= 10^{-5}$, $t_* = 10^{-4}$, and $n_* = 20$.

Method	dG(1)	dG(2)	cGP(2)	cGP(3)	CN	(10)
TOL $= 5 \cdot 10^{-3}$, systems	19	39	17	32	3.5	1
Time/step	47	81	45	71	9	
Total	1895	3243	1756	2839	1386	
TOL $= 1 \cdot 10^{-3}$, systems	13	31	11	27		
Time/step	41	72	38	67		
Total	2655	4682	2472	4230		

equations. The adaptive time step control utilizes a post-processed solution which is of higher order than the solution of the time stepping scheme. The time step control was performed with the PC11 and the PID controller. Numerical examples were presented which has typical features appearing in applications, like a time-dependent convection field or a time-dependent boundary condition at the inlet. For comparison, a recently proposed adaptive Crank–Nicolson scheme was also employed in the numerical studies, which relies on comparing two solutions computed with schemes of the same order.

The numerical studies showed that the time step control for the variational time integrators works fine. The dynamics of the solutions were represented well by the length of the time step. Taking both, efficiency and accuracy into consideration, then cGP(2) is certainly the best of the studied methods.

The PC11 controller led to simulations with fewer time steps than the PID controller for the same value of TOL. With the PID controller, less rejections and a smoother evolution of the length of the time step (Example 2) were observed.

Compared with the adaptive Crank–Nicolson scheme, the number of time steps was considerably smaller for the variational time integrators for computing solutions with similar accuracy. In our opinion, the higher order of the variational time integrators as well as the different time step selection algorithms are responsible for this fact. In addition, the evolution of the length of the time steps was much smoother for the variational schemes. The averaging step for avoiding the ringing phenomenon has probably a certain impact on this behavior. We also have the impression that the time step control based on two solutions with different order is more reliable than the control with two solutions of the same order. The computing time for one step of the adaptive Crank–Nicolson scheme was around five times smaller than for one step of cGP(2). For a problem with strongly changing temporal dynamics, Example 1, the smaller number of time steps resulted in cGP(2) to be more efficient. In the case that the dynamics vary smoothly, Example 2, the adaptive Crank–Nicolson scheme was finally the most efficient method.

Acknowledgments

We would like to acknowledge two unknown referees whose suggestions greatly helped to improve this paper.

References

[1] S. Berrone, M. Marro, Space–time adaptive simulations for unsteady Navier–Stokes problems, Comput. & Fluids 38 (6) (2009) 1132–1144.
[2] E. Hairer, G. Wanner, Solving Ordinary Differential Equations. II: Stiff and Differential–Algebraic Problems, second revised ed., in: Springer Series in Computational Mathematics, vol. 14, Springer-Verlag, Berlin, 2010, paperback.
[3] Jens Lang, Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems: Theory, Algorithm, and Applications, in: Lecture Notes in Computational Science and Engineering, vol. 16, Springer-Verlag, Berlin, 2001.
[4] Volker John, Joachim Rang, Adaptive time step control for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 199 (9–12) (2010) 514–524.
[5] Stefan Turek, Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach, With 1 CD-ROM ("Virtual Album": UNIX/LINUX, Windows, POWERMAC; "FEATFLOW 1.1": UNIX/LINUX), in: Lecture Notes in Computational Science and Engineering, vol. 6, Springer-Verlag, Berlin, 1999.
[6] Philip M. Gresho, David F. Griffiths, David J. Silvester, Adaptive time-stepping for incompressible flow. I. Scalar advection–diffusion, SIAM J. Sci. Comput. 30 (4) (2008) 2018–2054.
[7] Javier de Frutos, Bosco García-Archilla, Volker John, Julia Novo, An adaptive SUPG method for evolutionary convection–diffusion equations, Comput. Methods Appl. Mech. Engrg. 273 (2014) 219–237.

[8] Javier de Frutos, Bosco García-Archilla, Julia Novo, An adaptive finite element method for evolutionary convection dominated problems, Comput. Methods Appl. Mech. Engrg. 200 (49–52) (2011) 3601–3612.
[9] Volker John, Gunar Matthies, Joachim Rang, A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 195 (44–47) (2006) 5995–6010.
[10] Kenneth Eriksson, Claes Johnson, Vidar Thomée, Time discretization of parabolic problems by the discontinuous Galerkin method, RAIRO Modél. Math. Anal. Numér. 19 (4) (1985) 611–643.
[11] D. Schötzau, C. Schwab, An hp a priori error analysis of the DG time-stepping method for initial value problems, Calcolo 37 (4) (2000) 207–232.
[12] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Computational Differential Equations, Cambridge University Press, Cambridge, 1996.
[13] F. Schieweck, A-stable discontinuous Galerkin–Petrov time discretization of higher order, J. Numer. Math. 18 (1) (2010) 25–57.
[14] N. Ahmed, G. Matthies, Numerical study of SUPG and LPS methods combined with higher order variational time discretization schemes applied to time-dependent convection–diffusion–reaction equations. Preprint 1948, WIAS, 2014.
[15] Michael Besier, Rolf Rannacher, Goal-oriented space–time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow, Internat. J. Numer. Methods Fluids 70 (9) (2012) 1139–1166.
[16] S. Hussain, F. Schieweck, S. Turek, An efficient and stable finite element solver of higher order in space and time for nonstationary incompressible flow, Internat. J. Numer. Methods Fluids 73 (11) (2013) 927–952.
[17] G. Matthies, F. Schieweck, Higher order variational time discretizations for nonlinear systems of ordinary differential equations. Preprint 23/2011, Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg, 2011.
[18] Gustaf Söderlind, Automatic control and adaptive time-stepping, Numer. Algorithms 31 (1–4) (2002) 281–310. Numerical methods for ordinary differential equations (Auckland, 2001).
[19] Alexander N. Brooks, Thomas J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1–3) (1982) 199–259. FENOMECH'81, Part I (Stuttgart, 1981).
[20] T.J.R. Hughes, A.N. Brooks, A multidimensional upwind scheme with no crosswind diffusion, in: Finite Element Methods for Convection Dominated Flows (Papers, Winter Ann. Meeting Amer. Soc. Mech. Engrs., New York, 1979), in: AMD, vol. 34, Amer. Soc. Mech. Engrs. (ASME), New York, 1979, pp. 19–35.
[21] Hans-Görg Roos, Martin Stynes, Lutz Tobiska, Robust Numerical Methods for Singularly Perturbed Differential Equations, second ed., in: Springer Series in Computational Mathematics, vol. 24, Springer-Verlag, Berlin, 2008.
[22] Volker John, Julia Novo, Error analysis of the SUPG finite element discretization of evolutionary convection–diffusion–reaction equations, SIAM J. Numer. Anal. 49 (3) (2011) 1149–1176.
[23] Dmitri Kuzmin, Explicit and implicit FEM-FCT algorithms with flux linearization, J. Comput. Phys. 228 (7) (2009) 2517–2534.
[24] A.M.P. Valli, G.F. Carey, A.L.G.A. Coutinho, Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reaction–convection–diffusion processes, Internat. J. Numer. Methods Fluids 47 (3) (2005) 201–231.
[25] A.M.P. Valli, G.F. Carey, A.L.G.A. Coutinho, Control strategies for timestep selection in simulation of coupled viscous flow and heat transfer, Comm. Numer. Methods Engrg. 18 (2) (2002) 131–139.
[26] Ole Østerby, Five ways of reducing the Crank–Nicolson oscillations, BIT 43 (4) (2003) 811–822.
[27] Volker John, Ellen Schmeyer, Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion, Comput. Methods Appl. Mech. Engrg. 198 (3–4) (2008) 475–494.
[28] Volker John, Julia Novo, On (essentially) non-oscillatory discretizations of evolutionary convection–diffusion equations, J. Comput. Phys. 231 (4) (2012) 1570–1586.
[29] Youcef Saad, Martin H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[30] Volker John, Gunar Matthies, MooNMD—a program package based on mapped finite element methods, Comput. Vis. Sci. 6 (2–3) (2004) 163–169.
[31] Volker John, Ellen Schmeyer, On finite element methods for 3D time-dependent convection–diffusion–reaction equations with small diffusion, in: BAIL 2008—Boundary and Interior Layers, in: Lect. Notes Comput. Sci. Eng., vol. 69, Springer, Berlin, 2009, pp. 173–181.