

Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint

ISSN 2198-5855

**Adaptive time step control for higher order variational time
discretizations applied to convection-diffusion equations**

Naveed Ahmed¹ , Volker John^{1,2}

submitted: April 23, 2014

¹ Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
email: Naveed.Ahmed@wias-berlin.de
email: Volker.John@wias-berlin.de

² Free University of Berlin
Department of Mathematics and Computer Science
Arnimallee 6, 14195 Berlin
Germany

No. 1945
Berlin 2014



2010 *Mathematics Subject Classification.* 65M60.

Key words and phrases. Continuous Galerkin–Petrov method, discontinuous Galerkin method, post-processed solution, adaptive step length control, SUPG stabilization.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: <http://www.wias-berlin.de/>

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 equations. It is shown that the length of the time step properly reflects the dynamics of the solution. The numerical costs of the adaptive algorithm are discussed.

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 [2] 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 [11,22], the use of embedded schemes, was applied to the incompressible Navier–Stokes equations in [18]. 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 lower order method. However, the application 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 [29], the other second order scheme is the fractional-step θ -scheme and in [10], the use of the explicit Adams–Bashforth scheme was studied. The approach from [29] 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 [10] for one-dimensional convection-diffusion equations, its use for two-dimensional convection-diffusion equations can be found, e.g., in [6,7].

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 [15,18]. 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 [9,27]. 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 [8] or discontinuous Galerkin–Petrov method [26]. Numerical studies with cGP(k) and dG(k)

for convection-dominated convection-diffusion equations can be found in [1]. 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 [3]. 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 [13].

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 [26] 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 [23] 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 [23] 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 [28].

The goal of this paper consists in exploring the potential of higher order variational time discretizations in connection with an adaptive time step control, based on the post-processed solution, for scalar convection-diffusion equations. The use of the post-processed solution for an adaptive time step control is a straightforward idea. But to the best of our knowledge, its realization and numerical assessment cannot be found so far in the literature.

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 $u : \Omega \times (0, T] \rightarrow \mathbb{R}$ such that

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

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, $\mathbf{b}(t, \mathbf{x})$ is the flow velocity, $\sigma(t, \mathbf{x})$ is the reaction coefficient, and $f(t, \mathbf{x})$ is a given outer source of the unknown scalar quantity u . It will be assumed that either $\nabla \cdot \mathbf{b}(t, \mathbf{x}) = 0$ and $c(t, \mathbf{x}) \geq 0$, or that there exists a positive constant σ_0 such that

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

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 $u \in L^2(0, T; H_0^1(\Omega))$ and $\partial_t u \in L^2(0, T; H^{-1}(\Omega))$, with

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

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

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

Let $\{\mathcal{T}_h\}$ denote a family of shape regular triangulations of Ω into compact d -simplices, quadrilaterals, or hexahedra such that $\bar{\Omega} = \bigcup_{K \in \mathcal{T}_h} K$. 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 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 $t \in (0, T]$

$$(\partial_t u_h(t), v_h) + a(u_h(t), v_h) = (f(t), v_h) \quad \forall v_h \in V_h, \quad (3)$$

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 [4,12] that is frequently used due to its stability properties, its higher-order accuracy in appropriate norms, and its easy implementation, e.g., see [24]. 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 $t \in (0, T]$

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

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

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

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 $\delta_K = \mathcal{O}(\tau)$, where τ is the length of the time step, whereas in special cases optimal estimates for $\delta_K = \mathcal{O}(h_K)$ were proved, see [16] 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 [16] 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 [1]. 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 [21], the application of higher order variational time discretizations seems to be an open problem.

3 Temporal Discretization

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 [1]. To keep this paper self-containing, a brief presentation of the schemes, which provides the basic ideas, will be given here.

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

$$\begin{aligned} X_k &:= \{u \in C(I, V_h) : u|_{I_n} \in \mathbb{P}_k(I_n, V), n = 1, \dots, N\}, \\ Y_k &:= \{v \in L^2(I, V_h) : v|_{I_n} \in \mathbb{P}_k(I_n, V), n = 1, \dots, N\}, \end{aligned}$$

where

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

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 , $n = 1, \dots, N-1$. For such functions, the left-sided value u_n^- , right-sided value u_n^+ , and the jump $[u]_n$ are defined by

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

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 $v_h \in V_h$

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

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 $n \geq 2$. 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 $U_{n,h}^j \in V_h$, $j = 0, \dots, k$.

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

$$T_n : [-1, 1] \rightarrow \bar{I}_n, \quad \hat{t} \mapsto t_{n-1} + \frac{\tau_n}{2}(\hat{t} + 1), \quad (5)$$

see [1].

Using the same setting as in [1], 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 $U_{n,h}^j \in V_h$, $j = 1, \dots, k$, such that

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

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

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

see [23].

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

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

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 $\mathbf{F}_n^j \in \mathbb{R}^{m_h}$ are given by

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

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

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

The finite element nodal vector \mathbf{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 $\mathbf{u}_n^0 = \mathbf{u}_{n-1}^k$ 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 $\psi \in \mathbb{P}_k(I_n)$

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

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

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

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

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

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

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

where \mathbf{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 $\mathbf{u}_{n+1}^0 := \mathbf{u}_n^{k+1}$.

4 Post-processing and Adaptive Time Step Control

Following [23], 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.

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

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

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 , $j = 0, \dots, k$, and it is scaled such that $\hat{\zeta}'(1) = 1$. The nodal vector $\boldsymbol{\gamma}_n$ of a finite element function $a_h \in V_h$ is the solution of

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

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 [23] that the post-processed solution $\Pi u_{h,\tau}^n(t)$ can be interpreted as the solution obtained with a time stepping scheme of order $(k+1)$, $k \geq 2$. Thus, computing $\Pi u_{h,\tau}^n(t)$ requires the solution of the linear system of equations (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 $\Pi u_{h,\tau}^n$ of the solution $u_{h,\tau}$ of (9) on the interval I_n can be represented as

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

where

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

with T_n from (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 , $j = 1, \dots, k+1$, and $\hat{\vartheta}'(1) = 1$. The finite element function $b_n \in V_h$ is obtained by

$$b_n := \frac{1}{\vartheta_n(t_{n-1})} (u_{n-1}^- - u_{n-1}^+),$$

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 [23] that there is an interpretation of $\Pi u_{h,\tau}^n(t)$ 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 $\Pi u_{h,\tau}^n$ 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 Euclidean error norm of the error of the numerical solution and its post-processed solution

$$r_n := \|u_{h,\tau}^n - \Pi u_{h,\tau}^n\|,$$

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 [28]. In the numerical simulations presented in Section 5, the PC11 controller

$$\tau_{n+1}^* = \theta \left(\frac{\text{TOL } r_n}{r_{n+1}^2} \right)^{1/(k+1)} \frac{\tau_n^2}{\tau_{n-1}} \quad (11)$$

was used, e.g., as in the simulations of [18]. In (11), θ is a safety factor which is introduced to reduce the chance of rejection of 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 [11]. In our simulations, $s_{\min} = 0.1$ and $s_{\max} = 2$ were used. A step size τ_{n+1} is accepted if $r_{n+1} \leq \text{TOL}$, otherwise it is rejected.

Besides the PC11 controller (11), we performed also simulations with the PID controller with the parameters proposed in [30]. Usually, the computed solutions were similar for the same value of TOL (often a little bit more accurate for the PID controller), but the PID controller took notable more time steps. For the sake of brevity, only the results obtained with the PC11 controller will be presented in Section 5.

5 Numerical Results

This section presents numerical studies of the proposed algorithm for the adaptive time step control. In the first example, 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 [19], 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 [17] 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 is the availability of a good initial iterate, namely the solution from the previous discrete time. In our numerical studies for the higher order variational time stepping schemes we found, however, that Krylov subspace methods with standard preconditioners did not perform as efficiently as for simple time stepping schemes. For two-dimensional problems, we found that even the application of the sparse direct solver UMFPACK [5] was faster. Thus, UMFPACK was used in the simulations of Example 1. In three dimensions, the use of iterative solvers was still more efficient than applying the direct solver. The simulations for Example 2 were performed with GMRES [25] and a damped Jacobi preconditioner with the damping parameter 0.7 as solver for the linear systems of equations.

All simulations were performed with the code MOONMD [14] at HP BL460c Gen8 computers with Eight-Core 2700 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 Figure 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

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

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

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

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

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

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.

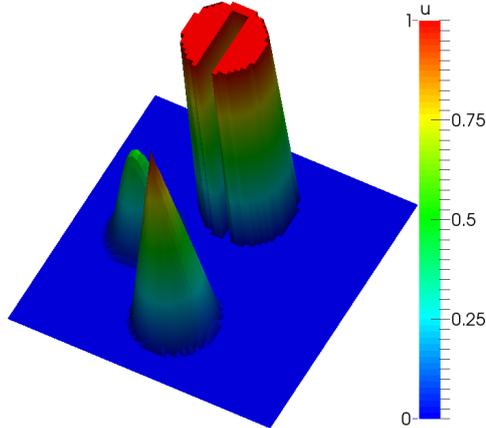


Fig. 1. Example 1. Initial condition.

The simulations were performed on an equidistant quadrilateral grid with squares of edge length $h = 1/64$ such that there were 16641 degrees of freedom (including Dirichlet nodes).

First of all, one had to find appropriate values for the parameter TOL in the PC11 controller (11). In our experience [18], 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 Figure 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 Figure 2. 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 chosen parameters, the evolution of the length of the time step is shown in Figure 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 Figure 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

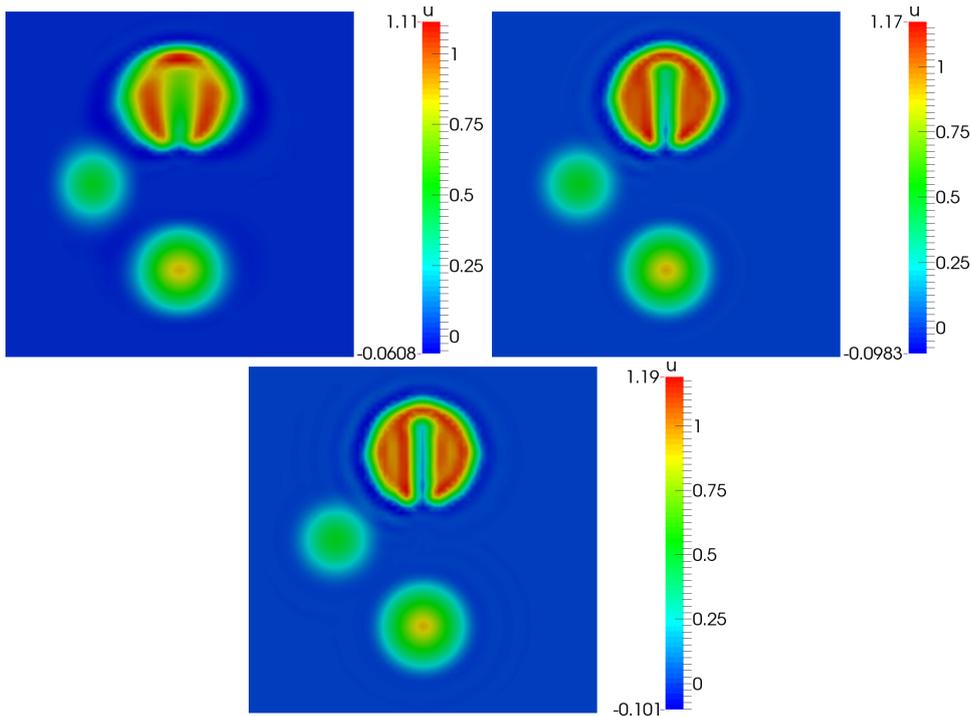


Fig. 2. Example 1. Solution at the final time with dG(1), left: $TOL = 10^{-2}$, right: $TOL = 5 \cdot 10^{-3}$, bottom: $TOL = 10^{-3}$.

Table 1

Example 1. Number of time steps. Note that there were no rejections.

TOL	dG(1)	dG(2)	cGP(2)	cGP(3)
$5 \cdot 10^{-3}$	1574	1405	1440	1250
$1 \cdot 10^{-3}$	8595	6999	7168	6260
$5 \cdot 10^{-4}$	17223	13989	14327	12500

using an equidistant time step, one has to choose this value. In Figure 4 one can observe that there are no visible differences in the solutions obtained with this small equidistant time step and the solution computed with the adaptive time stepping algorithm (with only 8.6 % of the number of time steps). Hence, in this respect, the adaptive algorithm worked 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.

Not only the number of time steps but also the overhead of the time step control algorithm is of importance for the efficiency of the method. Table 2 presents averaged computing times for the solution of the block systems arising in the time stepping schemes and for the solution of the linear system (10) for the post-processing. It turned out that the overhead for dG(1) and cGP(2) was around 20 %, whereas it was around 8 % for the higher order methods dG(2)

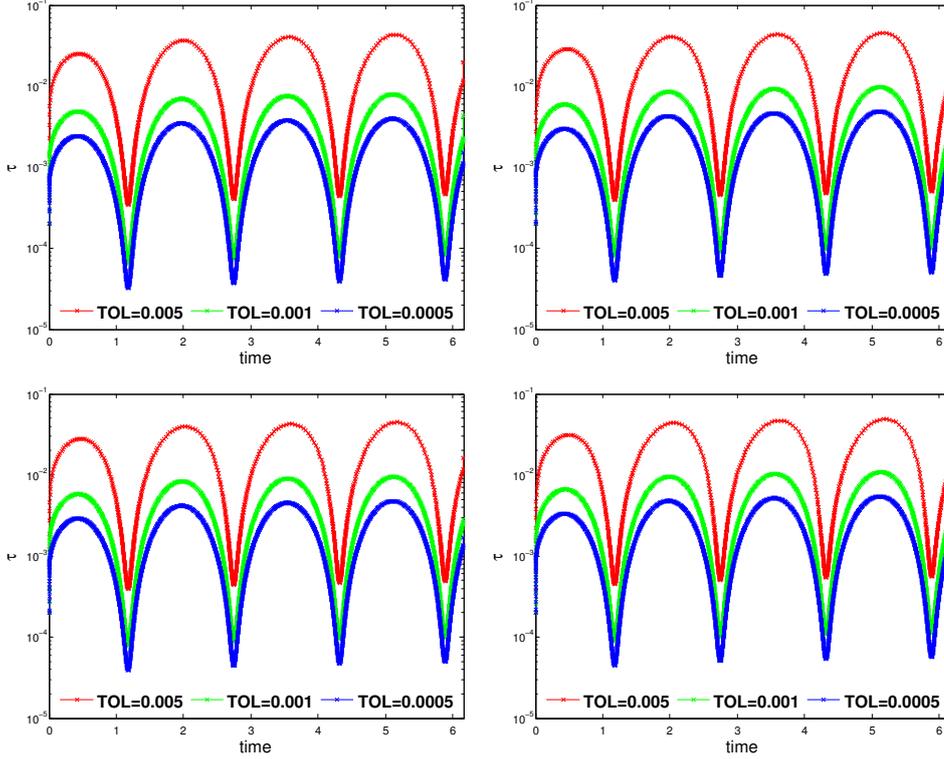


Fig. 3. Example 1. Temporal evolution of the length of the time step, dG(1), dG(2), cGP(2), cGP(3) (left to right, top to bottom).

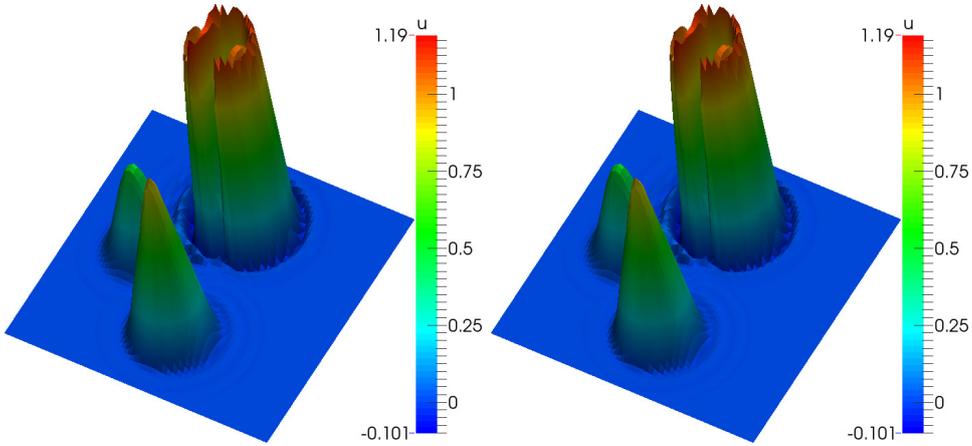


Fig. 4. Example 1. Solution at the final time, left: cGP(2) with adaptive time step and $TOL = 10^{-3}$ (7168 steps), right: cGP(2) with equidistant time step $\tau = 10^{-4}$ (61646 steps).

and cGP(3). Using the alternative strategy for computing the post-processed solution for the dG(k) methods resulted practically in a negligible overhead. In summary, the savings in the number of time steps were much more important than the overhead of the adaptive time step control algorithm.

The costs of one step with the popular (second order) Crank–Nicolson scheme

Table 2

Example 1. Averaged computing times (in seconds) for solving the block systems and for solving system (10) for computing the post-processed solution, both with the sparse direct solver UMFPACK.

method	dG(1)	dG(2)	cGP(2)	cGP(3)	(10)
time	0.50	1.28	0.50	1.28	0.10

are around the costs of the post-processing step, if also UMFPACK is used. Considering on the one hand the second order method dG(1) with adaptive time stepping and on the other hand the Crank–Nicolson scheme with equidistant time step $\tau = 10^{-4}$, then the adaptive time method becomes more efficient if it needs less than around one fifth of the number of equidistant time steps, i.e., less than around 12000 time steps. This situation is given for $\text{TOL} \in \{5 \cdot 10^{-3}, 10^{-3}\}$. Even the third order method cGP(2) would be faster than the Crank–Nicolson method for these values of the parameter TOL. If more efficient solvers can be applied for the Crank–Nicolson scheme, then the threshold for the number of time steps for the variational temporal discretizations to become more efficient decreases of course.

Example 2 *Time-dependent inlet condition.* This three-dimensional example was proposed in [20]. 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 $\mathbf{b} = (1, -1/4, -1/8)^T$, the diffusion is given by $\varepsilon = 10^{-6}$, and the reaction by

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

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 homo-

ogeneous Dirichlet conditions at the rest of the boundary. There are no sources, i.e., $f = 0$. The initial condition is set to be $u_0(\mathbf{x}) = 0$. 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 274625 degrees of freedom (including Dirichlet nodes).

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

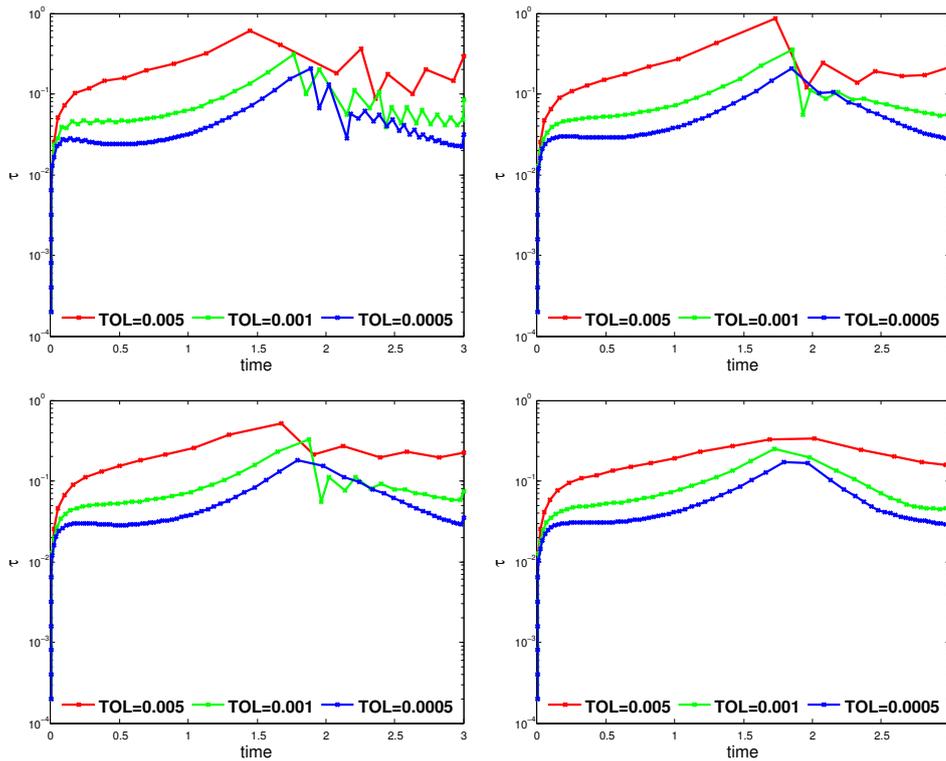


Fig. 5. Example 2. Temporal evolution of the length of the time step, $dG(1)$, $dG(2)$, $cGP(2)$, $cGP(3)$ (left to right, top to bottom).

Detailed information on the needed number of time steps is provided in Table 3. 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. It can be observed in Figure 5 that the length of the time step is oscillating in (2, 3) for this method.

Table 3

Example 2. Number of effective and rejected time steps.

TOL	dG(1)		dG(2)		cGP(2)		cGP(3)	
$5 \cdot 10^{-3}$	29	2	28	1	26	1	27	0
$1 \cdot 10^{-3}$	58	2	52	1	51	1	50	0
$5 \cdot 10^{-4}$	90	1	75	0	73	0	75	0

As a measure of accuracy, the value of the solution (amount of species) at the center of the outlet was proposed in [17]. It can be observed in Figure 6 that all simulations gave very similar results. Apart of dG(1) with $TOL = 5 \cdot 10^{-3}$, they are very close to the result obtained when applying the small equidistant time step $\tau = 0.01$.

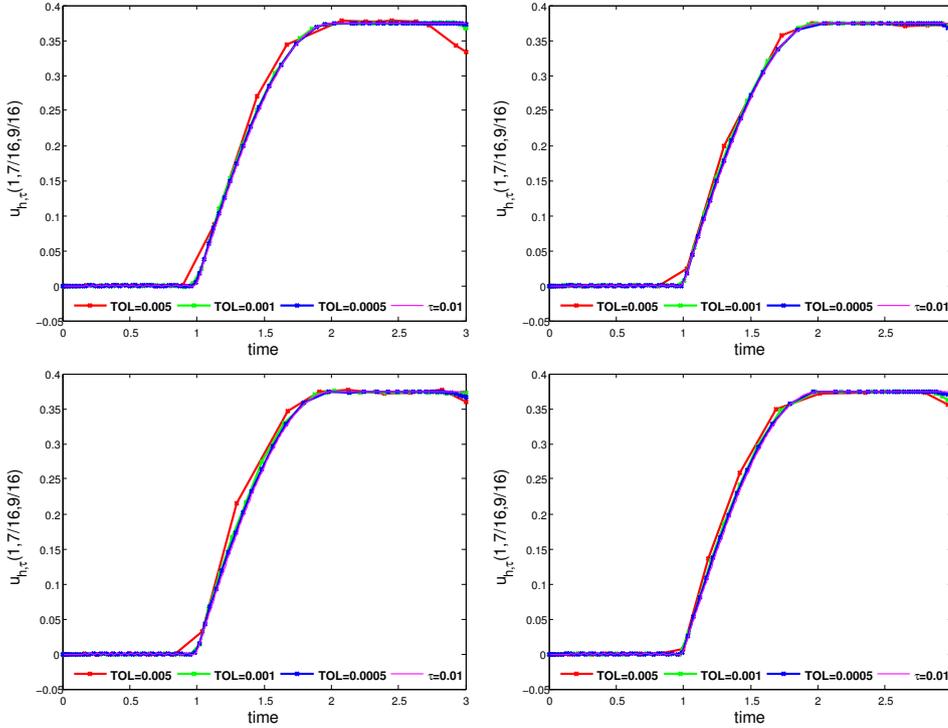


Fig. 6. Example 2. Temporal evolution of the amount of species at the center of the outlet, dG(1), dG(2), cGP(2), cGP(3) (left to right, top to bottom).

Averaged computing times per time step are provided in Table 4. Smaller values of the parameter TOL, which lead in the average to smaller time steps, improve obviously the initial iterates and, accordingly, result in faster solutions of the linear systems of equations. The numerical costs of dG(1) and cGP(2) on the one hand, and dG(2) and cGP(3) on the other hand, are comparable.

The overhead for computing the post-processed solution is small too negligible. Since the computing time for one time step with the Crank–Nicolson scheme is comparable with the time for the post-processing step, one finds that the variational time stepping schemes become rewarding, compared with an equidistant Crank–Nicolson scheme, only if the number of time steps can be reduced considerably by applying the adaptive time step control.

Table 4

Example 2. Averaged computing times (in seconds) for solving the block systems and for solving system (10) for computing the post-processed solution.

method	dG(1)	dG(2)	cGP(2)	cGP(3)	(10)
TOL = $5 \cdot 10^{-3}$	82	141	70	133	2
TOL = $1 \cdot 10^{-3}$	37	86	35	83	2
TOL = $5 \cdot 10^{-4}$	21	60	22	61	2

In our numerical studies, we considered also a two-dimensional example with the same features as Example 2. Since the spatial resolution could be chosen finer in two dimensions, the spatial error became smaller compared with the temporal error than in the three-dimensional example. As result, the drop in the length of the time step as the inlet condition decreases, at $t = 2$, is more pronounced than in the three-dimensional situation. We decided to present the three-dimensional results since they might be of more interest for potential applications.

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 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 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.

The numerical studies showed that the time step control 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. From the point of view of efficiency, it was estimated that the adaptive time step control with cGP(2) is rewarded, compared with the Crank–Nicolson scheme and equidistant time steps, if the

number of time steps can be reduced by a factor of 5 – 20, depending on the example. Since cGP(2) is of higher order than the Crank–Nicolson scheme, not only the efficiency is increased in this situation but also the temporal accuracy. In addition, the time step control based on solutions of schemes with different order might have some algorithmic advantage compared with a time step control based on two second order solutions as proposed for the Crank–Nicolson scheme.

The estimated factor for the reduction of the number of equidistant time steps is quite large. The key for increasing the competitiveness of the studied methods is certainly the development of efficient solvers for the arising block systems. A first step, for the incompressible Navier–Stokes equations, can be found in [13], but further research on this topic is needed.

References

- [1] N. Ahmed and 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. Technical report, 2013.
- [2] S. Berrone and M. Marro. Space-time adaptive simulations for unsteady Navier–Stokes problems. *Comput. & Fluids*, 38(6):1132–1144, 2009.
- [3] Michael Besier and 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):1139–1166, 2012.
- [4] Alexander N. Brooks and 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):199–259, 1982. FENOMECH '81, Part I (Stuttgart, 1981).
- [5] Timothy A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):196–199, 2004.
- [6] Javier de Frutos, Bosco García-Archilla, Volker John, and Julia Novo. An adaptive SUPG method for evolutionary convectiondiffusion equations. *Comput. Methods Appl. Mech. Engrg.*, 273:219–237, 2014.
- [7] Javier de Frutos, Bosco García-Archilla, and Julia Novo. An adaptive finite element method for evolutionary convection dominated problems. *Comput. Methods Appl. Mech. Engrg.*, 200(49-52):3601–3612, 2011.
- [8] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational differential equations*. Cambridge University Press, Cambridge, 1996.

- [9] Kenneth Eriksson, Claes Johnson, and Vidar Thomée. Time discretization of parabolic problems by the discontinuous Galerkin method. *RAIRO Modél. Math. Anal. Numér.*, 19(4):611–643, 1985.
- [10] Philip M. Gresho, David F. Griffiths, and David J. Silvester. Adaptive time-stepping for incompressible flow. I. Scalar advection-diffusion. *SIAM J. Sci. Comput.*, 30(4):2018–2054, 2008.
- [11] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2010. Stiff and differential-algebraic problems, Second revised edition, paperback.
- [12] T. J. R. Hughes and 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)*, volume 34 of *AMD*, pages 19–35. Amer. Soc. Mech. Engrs. (ASME), New York, 1979.
- [13] S. Hussain, F. Schieweck, and 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):927–952, 2013.
- [14] Volker John and Gunar Matthies. MoonNMD—a program package based on mapped finite element methods. *Comput. Vis. Sci.*, 6(2-3):163–169, 2004.
- [15] Volker John, Gunar Matthies, and Joachim Rang. A comparison of time-discretization/linearization approaches for the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 195(44-47):5995–6010, 2006.
- [16] Volker John and Julia Novo. Error analysis of the SUPG finite element discretization of evolutionary convection-diffusion-reaction equations. *SIAM J. Numer. Anal.*, 49(3):1149–1176, 2011.
- [17] Volker John and Julia Novo. On (essentially) non-oscillatory discretizations of evolutionary convection-diffusion equations. *J. Comput. Phys.*, 231(4):1570–1586, 2012.
- [18] Volker John and Joachim Rang. Adaptive time step control for the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 199(9-12):514–524, 2010.
- [19] Volker John and Ellen Schmeier. Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. *Comput. Methods Appl. Mech. Engrg.*, 198(3-4):475–494, 2008.
- [20] Volker John and Ellen Schmeier. On finite element methods for 3D time-dependent convection-diffusion-reaction equations with small diffusion. In *BAIL 2008—boundary and interior layers*, volume 69 of *Lect. Notes Comput. Sci. Eng.*, pages 173–181. Springer, Berlin, 2009.
- [21] Dmitri Kuzmin. Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.*, 228(7):2517–2534, 2009.

- [22] Jens Lang. *Adaptive multilevel solution of nonlinear parabolic PDE systems*, volume 16 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2001. Theory, algorithm, and applications.
- [23] G. Matthies and 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.
- [24] Hans-Görg Roos, Martin Stynes, and Lutz Tobiska. *Robust numerical methods for singularly perturbed differential equations*, volume 24 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2008.
- [25] Youcef Saad and Martin H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.
- [26] F. Schieweck. A-stable discontinuous Galerkin-Petrov time discretization of higher order. *J. Numer. Math.*, 18(1):25–57, 2010.
- [27] D. Schötzau and C. Schwab. An *hp* a priori error analysis of the DG time-stepping method for initial value problems. *Calcolo*, 37(4):207–232, 2000.
- [28] Gustaf Söderlind. Automatic control and adaptive time-stepping. *Numer. Algorithms*, 31(1-4):281–310, 2002. Numerical methods for ordinary differential equations (Auckland, 2001).
- [29] Stefan Turek. *Efficient solvers for incompressible flow problems*, volume 6 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 1999. An algorithmic and computational approach, With 1 CD-ROM (“Virtual Album”: UNIX/LINUX, Windows, POWERMAC; “FEATFLOW 1.1”: UNIX/LINUX).
- [30] A. M. P. Valli, Carey G. F., and 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):131–139, 2002.