Goal-oriented a posteriori error control for time discretizations involving incompressible flows

---

Dissertation

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

Der Fakultät für Mathematik der

Technischen Universität Dortmund

vorgelegt von

Xenia Anais Kerkhoff-Bernaciak

im März 2026

Dissertation

Goal-oriented a posteriori error control for time discretizations involving incompressible flows

Fakultät für Mathematik
Technische Universität Dortmund

Mitglieder der Prüfungskommission:

Erstgutachter:	Prof. Dr. Stefan Turek
Zweitgutachter:	Prof. Dr. Christian Kreuzer

Tag der mündlichen Prüfung: 27. 07. 2026

Abstract

We use the dual weighted residual (DWR) method to apply goal-oriented error control for time discretizations. We consider the heat equation and the Navier-Stokes equations and use finite difference and Galerkin methods to discretize in time. We propose a uniform approach to take quadrature errors into account.

For finite difference methods, we build upon the works of Meidner and Richter [48, 49] and give new proofs for their DWR error representations. Here, θ time-stepping schemes are considered. In contrast to classical DWR error estimates, an additional term is included here, which the authors refer to as Galerkin defect or quadrature error. For the Navier-Stokes equations, we modify their discretization so that the entire test space is θ dependent, which improves the error representation.

For continuous and discontinuous Galerkin methods, we modify the error representations known from literature by taking quadrature errors into account. This leads to an additional term in the error representations, which makes the Galerkin method error representations more similar to those derived with the θ method.

We demonstrate the importance of the additional terms in the modified error representations by numerical results. We focus on discontinuous Galerkin methods and the usage of the implicit Euler method. The error estimators are implemented in the software FEAT3.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Dr. Stefan Turek, for his guidance and support and for giving me the opportunity to pursue my Ph.D. His mentorship enabled me to grow both intellectually and personally throughout this time.
I would also like to thank everyone at TU Dortmund who contributed to my mathematical education and supported me during my studies in various ways.
Special thanks go to all the members of LSIII for creating such a wonderful working atmosphere. I am especially grateful to my office neighbor Arooj, as well as to Jonas, Peter and Max for always having time to talk about implementations. I would also like to thank Christoph for our spontaneous discussions.
Lydia, I am deeply grateful for your endless support in all matters throughout this journey.
Finally, I would like to thank my family and friends. Karolina, Kira, and Uli, thank you for always believing in me. Uli, without you, I might never have discovered the joy of mathematics. Patrick, thank you for always being there for me and for understanding me better than anyone else.

Xenia Anais Kerkhoff-Bernaciak

Contents

Abstract i

Acknowledgements iii

1 Introduction 1

2 Considered equations and discretizations 5
- 2.1 The equations . . . 5
- 2.2 Time discretization . . . 7
  - 2.2.1 Discontinuous Galerkin method . . . 8
  - 2.2.2 θ time-stepping scheme . . . 9
- 2.3 Space discretization . . . 13
  - 2.3.1 Heat equation . . . 13
  - 2.3.2 Navier-Stokes equations . . . 14

3 Derivation error representations 17
- 3.1 Basic theorem for DWR . . . 18
- 3.2 Heat equation . . . 20
- 3.3 Navier-Stokes equations . . . 27
- 3.4 Modifications . . . 34
  - 3.4.1 Incompressibility constraint . . . 34
  - 3.4.2 Quadrature rule . . . 36

4 Practical aspects for adaptive time-step refinement 41
- 4.1 Dual equations . . . 41
- 4.2 Evaluation error estimators . . . 42
  - 4.2.1 Interpolation operators . . . 43
  - 4.2.2 Error estimators . . . 46

4.3 Time-step adaption methods 51

5 Numerical results **53**
5.1 Heat equation 54
5.1.1 Exponential decrease in time (effectivity) 56
5.1.2 Oscillations in time (adaptivity) 60
5.2 Navier-Stokes equations 65
5.2.1 Drag and lift coefficients 68
5.2.2 Flow through a square (effectivity) 73
5.2.3 Flow around a cylinder (adaptivity) 78

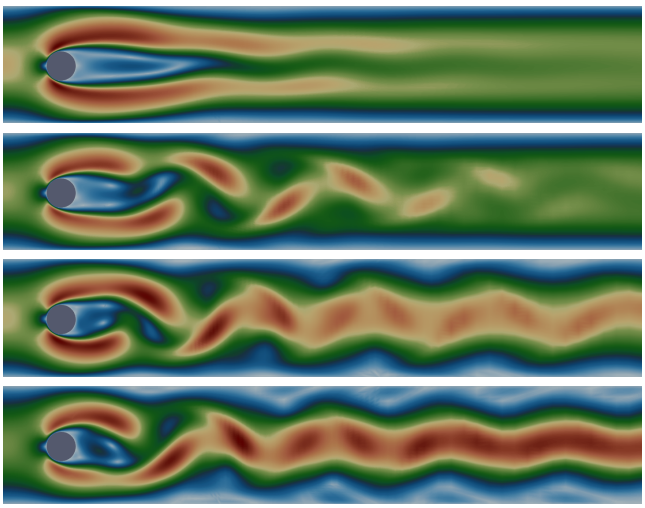
6 Conclusions and outlook **87**

A Appendix **91**
A.1 Drag and lift coefficients 91
A.2 Error representations for continuous Galerkin-Petrov discretizations . . . 94
A.2.1 Quadrature rule 95
A.2.2 Heat equation 95
A.2.2 Navier-Stokes equations 97

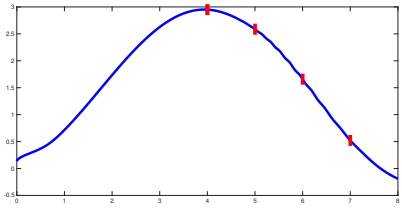
1 Introduction

Motivation

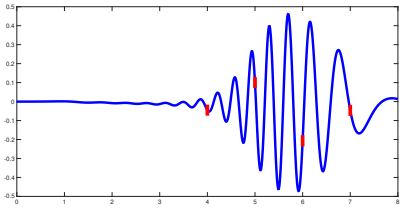
Partial differential equations (PDEs) appear in many technical and scientific applications. It is not always the accuracy of the complete solution to these PDEs that is of interest, but rather certain physical quantities that depend on the solution. For example, in a flow simulation around an object, we are not interested in the velocity and pressure in the calculation area, but rather in the mean drag and lift coefficient of the object.



(a) Velocity field at $t = 4, 5, 6, 7$.



(b) Drag coefficient
(mean value $\approx$ 1.606214).



(c) Lift coefficient
(mean value $\approx$ -1.0472e-02).

Figure 1.1: Flow around a cylinder benchmark.

A classical benchmark example in the field of incompressible Navier-Stokes equations (NSE) is a flow around a cylinder problem from [57]. Depending on a changing inflow, the Reynolds number changes and vortex flows form. The change of the velocity field is

shown in Figure 1.1a. Furthermore, the drag and lift coefficients over the time period are shown in Figures 1.1b and 1.1c. While the velocity field changes significantly over time, e.g., the drag coefficient behaves in a simpler manner. In contrast, oscillations in the lift coefficient can be observed. This raises the question of how small time steps must be chosen in order to represent the temporal dynamics of the problem with sufficient accuracy to obtain a good estimate of the mean values.
In this context, adaptive methods can be used to solve such complex time-dependent problems more efficiently. Especially, since the effort of solving instationary computational fluid dynamics (CFD) simulations with realistic flow processes is still very high. The solution should only be resolved more precisely in areas (space and/or time) where it is necessary in terms of accuracy. Therefore, a posteriori error estimates are needed, which give local error indicators where to higher resolve the equation or not. Since drag and lift coefficients behave very differently, it seems plausible that optimal grids cannot be easily determined based on prior knowledge of physics. This motivates the usage of goal-oriented adaptive methods for time-dependent problems.

A posteriori error estimates and dual weighted residual method

We would like to determine the quality of the numerical approximation using a posteriori errors. In general such estimates are based on residuals and estimate the error in suitable norms or pointwise errors, see for example [1, 14, 41, 42, 52]. One drawback is, that these errors are generally not the quantity of physical interest. Another drawback is, that the estimates may include constants, which cannot be accurately determined for more complex equations, such as the NSE. This makes it difficult to assess the quality of the error estimators, which is referred to as effectivity, meaning the ratio between the exact and estimated error.
The dual weighted residual method (DWR) method is an approach which tries to deal with both drawbacks at the same time. This method was introduced by Becker and Rannacher [8, 9]. It is based on deriving error representations for general aspect functionals

$$J(u) - J(u_k).$$

For us, the mean drag and lift coefficients are particularly important. Based on the error representations there are different ways of evaluation and localization of the error estimators. For example Richter and Wick [56] provide an overview of these methods.
The DWR method is already used in various fields. For example, on different types of equations [10, 17, 45, 55, 61, 71] or optimal control problems [6, 44]. There are also works that analyze bounds of the error estimators and their effectivity index [25, 46] or that try to improve the reliability of error estimation [51].

Goal and focus of this work

We implement goal-oriented error estimation in the software FEAT3[1], since this software provides fast solvers for running CFD simulations. We do not want to calculate adaptively in space in the first step, but rather in time, in order to reduce the computational effort by choosing suitable time-step sizes. This work should be seen as a first step in this direction. As time-stepping schemes, we examine finite difference and Galerkin methods. We focus on incorporating quadrature errors in DWR error representations. This is of interest because finite difference methods are often used numerically, whereas a Galerkin discretization is required for the application of the DWR approach.

For finite difference methods, we rely on the works of Meidner and Richter [48, 49] on θ methods. They derive error estimators for fractional θ time-stepping schemes. We justify their score function by establishing a connection to Hermite interpolation. Subsequently, we provide new proofs for their DWR error representations. Finally, for the NSE, we modify their θ scheme making the entire test space θ dependent, which leads to improved error estimation.

For Galerkin methods we use the works of Besier and Rannacher [15], and Schmich and Vexler [60] about the discontinuous Galerkin (DG) methods. The DWR error estimators derived here differ from those in the works of [48, 49]. An additional term appears in the error estimators for the θ methods, which is called the Galerkin defect. This Galerkin defect is essentially a quadrature error and missing for the DG schemes. The error representation formula for DG schemes does not include full time discretization on either side, making it less valuable from a time adaptivity perspective.

Schmich and Vexler [60] also use an additional term to improve error estimation when using the Crank-Nicholson time-stepping scheme for DWR error estimators derived by a continuous Galerkin-Petrov (cGP) discretization. However, DWR error representation formulas for cGP schemes are not named explicitly.

This leads to the observation that deriving error representations that incorporate quadrature errors is important for theoretical and numerical reasons. Consequently, we modify the DG error estimators to take quadrature into account. With this adjustment, the DG error estimators become closely analogous to those of the θ method. We also present error representations explicitly for the cGP schemes. These expressions are adapted to include quadrature effects in the same manner as for the DG schemes.

Numerically, we compare on the one hand the error estimators for the θ methods with the Galerkin methods and on the other hand our modified error estimators with those already used in the literature. The cases of interest are $\theta = 1$ and $\theta = 0.5$. These θ values are suitable for comparing the implicit Euler method with the DG(0) method and the Crank-Nicholson scheme with the cGP(1) scheme.

[1] http://www.featflow.de/en/software/feat3.html

In the latter case, the error estimators of our modified cGP(1) scheme and the θ method are identical. This means that the rather unsatisfactory behavior of the error estimators of the cGP(1) scheme mentioned in Schmich and Vexler [60] has been theoretically resolved and has already been analyzed numerically. Therefore, we limit our simulations to the implicit Euler method. Consequently, statements regarding the cGP methods are provided in the appendix while the main focus in this thesis is on DG schemes.

Structure of the thesis

The thesis is structured as follows. In Chapter 2 we give an overview of the PDEs and their discretizations. In particular, we focus on the discretization of time and consider two different methods: the DG and θ method along the standard lines. Using Hermite interpolation, we motivate the formulation of the θ method in function space.
The goal of Chapter 3 is to represent the error with respect to the aspect functional J using the DWR method. Representations of the errors are derived using weighted residuals with the help of Lagrange functions.
Section 3.1 deals with the theoretical fundamentals of the DWR method, an abstract error representation and a simpler variant. In the following sections we apply both of these variants and explain which of them we are using. Sections 3.2 and 3.3 contain error representations for the heat equation and NSE for the time discretizations from Chapter 2, which are well known. For the θ method we give new proofs. In Section 3.4, we modify the DG and θ schemes according to our proposed adaptations and provide new error representations.
In Chapter 4 we discuss practical aspects of the DWR method to run numerical experiments. First, we give some comments on solving the dual equations. Then, in Section 4.2 we derive computable error estimators with the help of interpolation operators, and provide an overview of all the estimators used in this work. In Section 4.3 we specify the methods for time-step adaption.
Finally, we present the numerical results in Chapter 5. Thereby, we compare the different error estimators numerically concerning effectivity and localization of the errors. To measure the effectivity of the error estimators, we construct examples in which the space discretization error is almost negligible. We also compare different marking strategies and evaluation options of drag and lift coefficients.
We conclude with summarizing and an outlook in Chapter 6.
In the appendix we discuss numerical evaluation methods for drag and lift coefficients in more detail. Furthermore, the appendix contains the definitions of the cGP discretizations and their DWR error representations. These are comparable to those presented by Besier and Rannacher [15] and by Schmich and Vexler [60]. We also modify the cGP method to account for quadrature errors and establish new DWR estimators.

2

Considered equations and discretizations

In this work we consider two equations. The heat equation as easy model problem for heat transfer and the Navier-Stokes equations as model problem for incompressible flow problems. We consider both equations in two space dimensions.
In Section 2.1, we define the function spaces, in which we will seek for a solution, then we state both equations in strong and weak form.
After that we discuss two time discretizations in Section 2.2

- the discontinuous Galerkin discretization of order *r* (DG(r) method) and
- the θ time-stepping scheme.

For the θ scheme, we give a formulation of the time-stepping scheme as Galerkin in time formulation following [48, 49]. We note that their choice of the score function is a Hermite interpolation problem, making it unique in a sense. This provides a different interpretation and proof of their lemma.
The space discretizations and full discrete form of the PDEs are given in Section 2.3.

2.1. The equations

Function spaces

We briefly define the used function spaces and notation. We consider an open domain $\Omega \subset \mathbb{R}^2$ with boundary $\Gamma = \partial\Omega$. We set

$$V = \{v \in H^1(\Omega) : v|_\Gamma = 0\}$$

with dual space V^*. Furthermore, we denote by $(\cdot,\cdot)$ the inner L^2 product. For time-dependent functions on the time interval $I = (0,T)$ we use the Hilbert space

$$W(I,V) = \{v : v \in L^2(I,V) \text{ and } \partial_t v \in L^2(I,V^*)\}.$$

For the Navier-Stokes equations, we use

$$X = \{u = (v,p)^T : v \in W(I,V^2) \text{ and } p \in L^2(I,L_0^2(\Omega))\}, \tag{2.1.1}$$

where $L_0^2(\Omega)$ is the space of all L^2 integrable functions with mean value zero.

Heat equation

As a first model problem, we consider the heat equation on a domain $\Omega \subset \mathbb{R}^2$. The equation describes the heat transfer from an initial temperature u_0 at time 0 up to a final time T. We seek a solution $u(x,t)$, which fulfills

$$\begin{aligned} \partial_t u - \Delta u &= g \quad \text{in } \Omega \times (0,T) \\ u &= 0 \quad \text{on } \partial\Omega \times (0,T) \\ u(\cdot,0) &= u_0 \quad \text{in } \Omega \end{aligned} \tag{2.1.2}$$

for a source term g and Dirichlet zero boundary condition at $\Gamma = \partial\Omega$. For given $g \in L^2(I,L^2(\Omega))$ and $u_0 \in L^2(\Omega)$ the weak formulation of (2.1.2) reads: Find $u \in W(I,V)$ such that

$$\int_0^T (\partial_t u, \varphi) + (\nabla u, \nabla\varphi)dt + (u(0) - u_0, \varphi(0)) = \int_0^T (g,\varphi)dt \quad \forall \varphi \in W(I,V). \tag{2.1.3}$$

This expression is well defined because the solution is continuous in time $u \in C(\bar{I},L^2(\Omega))$. For existence and uniqueness of a solution of (2.1.3), see e.g. [62].

Navier-Stokes equations

The incompressible Navier-Stokes equations in two dimensions on a domain $\Omega \subset \mathbb{R}^2$ are given by

$$\begin{aligned} \partial_t v - \vartheta\Delta v + (v\cdot\nabla)v + \nabla p &= g \quad \text{in } \Omega \times (0,T) \\ \nabla \cdot v &= 0 \quad \text{in } \Omega \times (0,T) \\ v(\cdot,0) &= v_0(\cdot) \quad \text{in } \Omega. \end{aligned} \tag{2.1.4}$$

v is the velocity field and p the pressure. Furthermore, $\vartheta > 0$ is the kinematic viscosity, and suitable boundary conditions must be set. We consider Dirichlet zero boundary conditions for our weak formulation of the PDE and the derivations in the following chapters. We get the weak form of (2.1.4) by testing with $\varphi = (\psi, \chi) \in X$ and integration by parts: Find $u = (v, p) \in X$ which fulfills

$$\begin{aligned}&\int_0^T (\partial_t v, \psi) + \vartheta(\nabla v, \nabla\psi) + (v \cdot \nabla v, \psi) - (p, \nabla \cdot \psi) + (\nabla \cdot v, \chi) dt + (v(0) - v_0, \psi(0)) \\ &= \int_0^T (g, \psi) dt \end{aligned} \tag{2.1.5}$$

The source function $g \in L^2(I, L^2(\Omega)^2)$ and initial data $v_0 \in V^2$ with $\nabla \cdot v_0 = 0$ are given. See [13, 65] for results of existence and uniqueness of solutions. To shorten the notation for later use we define

$$a(v)(\psi) = \vartheta(\nabla v, \nabla\psi) + (v \cdot \nabla v, \psi), \quad b(p, \psi) = (p, \nabla \cdot \psi) \tag{2.1.6}$$

and the continuous residual

$$\rho(u)(\varphi) = \int_0^T (g, \psi) - (\partial_t v, \psi) - a(v)(\psi) + b(p, \psi) - b(\chi, v) dt + (v_0 - v(0), \psi(0)). \tag{2.1.7}$$

Remark 1 (Boundary conditions and uniqueness of pressure)**.** *To ensure uniqueness of the pressure in the case of Dirichlet zero boundary conditions, we use the function space $L^2_0(\Omega)$ for the pressure. This is only done in the theoretical considerations. In our numerical studies, we consider three different types of boundary conditions. We distinguish between inflow boundary Γ_{in}, outflow boundary Γ_{out} and Dirichlet zero boundary Γ_D, with $\partial\Omega = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_D$. At the inflow boundary Γ_{in} the velocity field is given by a function v_{in}. In this case, the pressure is uniquely given by the natural boundary condition $\frac{\partial v}{\partial n} - pn$ at the outflow boundary, see [7, 33] for further discussions on boundary conditions.*

Now, we describe the discretizations in time and space of equations (2.1.3) and (2.1.5).

2.2. Time discretization

For semi discretization in time, we partition the time interval $\bar{I} = [0, T]$,

$$\bar{I} = \{0\} \cup I_1 \cup \cdots \cup I_M,$$

with subintervals $I_m = (t_{m-1}, t_m]$ using the discrete time points

$$0 = t_0 < t_1 < \cdots < t_M = T.$$

The size of the interval I_m is $k_m = t_m - t_{m-1}$ for $m = 1, \ldots, M$. We use the time discretization parameter k, defined as piecewise constant function by $k|_{I_m} = k_m$. On this time grid, we define two time-stepping schemes, a Galerkin discretization using discontinuous finite elements in time and a finite difference method, the θ time-stepping scheme. For the θ scheme, we give a formulation as Galerkin method because we need this to apply the DWR method to finite difference schemes.

2.2.1. Discontinuous Galerkin method

The DG discretization is based on the weak form of the PDE. The equation is set up for piecewise polynomial functions in time. In order to couple individual time intervals with each other, so-called fluxes are introduced. The upwind flux is used for the discretization of time.

Heat equation

For the heat equation, we define for $r \in \mathbb{N}_0$ the semi-discrete space

$$V_k^{DG(r)} = \{v_k : v_k|_{I_m} \in P^r(I_m, V), m = 1, \ldots, M, v_k(0) \in L^2(\Omega)\}.$$

Thereby, $P^r(I_m, V)$ denotes the space of polynomials up to degree r on I_m with values in V. Functions in $V_k^{DG(r)}$ have discontinuities at the time grid points t_m. At these points, we define the limits and jump by

$$v^{m,+} = \lim_{\varepsilon \to 0, \varepsilon > 0} v(t_m + \varepsilon), \quad v^{m,-} = \lim_{\varepsilon \to 0, \varepsilon > 0} v(t_m - \varepsilon), \quad [v]^m := v^{m,+} - v^{m,-}.$$

We use the notation that $v^{0,-}$ corresponds to the initial data or its projection. With these definitions we can formulate the semi-discrete weak form of the heat equation. We search for a solution $u_k \in V_k^{DG(r)}$ which fulfills

$$\sum_{m=1}^{M} \int_{I_m} (\partial_t u_k, \varphi_k) + (\nabla u_k, \nabla \varphi_k) dt + \sum_{m=0}^{M-1} ([u_k]^m, \varphi_k^{m,+}) + (u_k^{0,-} - u_0, \varphi_k^{0,-}) = \int_I (g, \varphi_k) dt \tag{2.2.1}$$

for all $\varphi_k \in V_k^{DG(r)}$.

Navier-Stokes equations

For the Navier-Stokes equations, we define for $r \in \mathbb{N}_0$

$$X_k^{DG(r)} = \{u_k = (v_k, p_k)^T : v_k|_{I_m} \in P^r(I_m, V^2), v_k^{0,-} \in V^2 \text{ with } \nabla \cdot v_k^{0,-} = 0 \\ \text{and } p_k|_{I_m} \in P^r(I_m, L_0^2(\Omega))\}.$$

With the same definitions of the jump term, we formulate the semi-discrete weak form as follows: Find $u_k \in X_k^{DG(r)}$ which fulfills

$$\begin{aligned} \sum_{m=1}^{M} \int_{I_m} (\partial_t v_k, \psi_k) + a(v_k)(\psi_k) - b(p_k, \psi_k) + b(\chi_k, v_k) dt \\ + \sum_{m=0}^{M-1} ([v_k]^m, \psi_k^{m,+}) + (v_k^{0,-} - v_0, \psi_k^{0,-}) = \int_I (g, \psi_k) dt \end{aligned} \tag{2.2.2}$$

for all $\varphi_k = (\psi_k, \chi_k)^T \in X_k^{DG(r)}$. See, e.g., [22] for results on convergence. Because ansatz and test space are discontinuous in time, these discretizations lead to a time-stepping scheme.

Remark 2 (Numerical quadrature)**.** *Since no quadrature rule was chosen to evaluate the time integrals, especially the integrals*

$$\sum_{m=0}^{M} \int_{I_m} (g, \varphi_k) dt \text{ or } \sum_{m=0}^{M} \int_{I_m} (g, \psi_k) dt,$$

equations (2.2.1) *and* (2.2.2) *are not fully time discretized. In Section 3.4.2, we propose a modification that takes quadrature errors into account, resulting in a complete time discretization.*

Remark 3 ($r = 0$)**.** *Choosing $r = 0$ leads to the implicit Euler method if the time integral is calculated using the right-sided box rule or the one point Gauss-Radau quadrature formula, see Section 3.4.2.*

2.2.2. θ time-stepping scheme

The θ scheme with $\theta \in [0, 1]$ for an initial value problem of the form

$$u'(t) = A(u(t)) + g(t), \quad u(0) = u_0$$

consists in finding the next approximation u^m by

$$u^m - u^{m-1} = k_m(\theta(A(u(t_m)) + g(t_m)) + (1-\theta)(A(u(t_{m-1}) + g(t_{m-1}))) \tag{2.2.3}$$

with the notation $u^m = u(t_m) \in V$ for $m = 1,\dots,M$ and $u^0 = u_0$. For the finite difference scheme, the explicit Euler method is obtained for $\theta = 0$, the implicit Euler scheme for $\theta = 1$, and the Crank-Nicolson scheme for $\theta = 0.5$. We now reformulate the finite difference method as a Galerkin method so that we can later derive the DWR error representations. One motivation for handling the θ scheme in the context of function spaces is to identify a function $\phi(t)$ with the following properties

$$\begin{aligned} \int_{I_m} f(t)\phi(t)dt &= \theta k_m f(t_m) + (1-\theta)k_m f(t_{m-1}) \quad \text{for } f \text{ linear,} \\ \int_{I_m} f(t)\phi(t)dt - \theta k_m f(t_m) - (1-\theta)k_m f(t_{m-1}) &= O(k_m^2). \end{aligned} \tag{2.2.4}$$

Remark 4 ($\theta = 0.5$)**.** *For $\theta = 0.5$ this is natural in the sense that*

$$\int_{I_m} f(t)dt \approx 0.5 k_m f(t_m) + 0.5 k_m f(t_{m-1})$$

is the trapezoid rule, or Gauß Lobatto rule of order 1, which integrates linear polynomials exactly. It is known, that the quadrature error is of order $O(k_m^2)$. The function $\phi(t) = 1$ fulfills the interpolation conditions (2.2.4) *in this case. This coincides with the formulation of the Crank-Nicholson time-stepping scheme as the continuous Galerkin-Petrov method of order 1. The first order continuous Galerkin-Petrov method has a linear continuous ansatz space and a DG(0) test space, see [3, 58] and the appendix.*

For a general θ, with partial integration and $\phi^{(1)}, \phi^{(2)}$ being the first and second anti-derivatives of ϕ, we get

$$\int_{I_m} f(t)\phi(t)dt = f(t)\phi^{(1)}(t)|_{I_m} - f'(t)\phi^{(2)}(t)|_{I_m} + \int_{I_m} f''(t)\phi^{(2)}(t)dt.$$

Therefore we set

$$\phi^{(1)}(t_m) = \theta k_m, \quad \phi^{(1)}(t_{m-1}) = -(1-\theta)k_m, \quad \phi^{(2)}(t_m) = 0, \quad \phi^{(2)}(t_{m-1}) = 0.$$

Using a polynomial ansatz, these conditions define a Hermite interpolation problem. For example, after [36] we can specify the unique polynomial of degree three by

$$\phi_m^{(2)}(t) = \frac{2\theta - 1}{k_m}(t - t_{m-1})^2(t - t_m) + (1-\theta)(t - t_{m-1})^2 - (1-\theta)k_m(t - t_{m-1}).$$

This proves the following Lemma from [48], with

$$\phi_m(t) = 1 + 6\big(\theta - \frac{1}{2}\big)\frac{2t - t_{m-1} - t_m}{k_m}. \tag{2.2.5}$$

Lemma 5 (Numerical quadrature)**.** *For $v \in W^{2,1}(I_m, \mathbb{R})$, it holds*

$$\int_{I_m} v(t)\phi_m(t)dt = \theta k_m v(t_m) + (1-\theta)k_m v(t_{m-1}) + \mathcal{R}_m(v)$$

with

$$|\mathcal{R}_m(v)| \leq k_m^2 \|\partial_t^2 v(t)\|_{L^1(I_m,\mathbb{R})}.$$

Heat equation

Applied to the heat equation (2.1.3), the time discretization leads to the scheme: Find the next approximation u^m by solving

$$\begin{aligned}(u^m - u^{m-1}, \varphi) + k_m(\theta(\nabla u^m, \nabla\varphi) + (1-\theta)(\nabla u^{m-1}, \nabla\varphi)) \\ = k_m(\theta(g(t_m), \varphi) + (1-\theta)(g(t_{m-1}), \varphi))\end{aligned} \tag{2.2.6}$$

for all $\varphi \in V$ in dependency of u^{m-1}, where $u^0 = u_0$ is defined by the initial data. In order to derive an error estimator with the DWR method, we need to work in function spaces in space and time. As ansatz space we use the space of linear continuous functions in time

$$V_k^1 = \{v_k \in C(\bar{I}, V) : v_k|_{I_m} \in P^1(I_m, V), m = 1, \ldots, M\}.$$

The θ dependency is introduced by the test space

$$V_k^\theta = \left\{v_k \in L^2(I, L^2(\Omega)) : v_k|_{I_m} = \phi_m\psi, \psi \in V, m = 1, \ldots, M, v_k(0) \in L^2(\Omega)\right\},$$

with the function $\phi_m(t) = 1 + 6(\theta - \frac{1}{2})\frac{2t - t_{m-1} - t_m}{k_m}$ from equation (2.2.5). The test space fulfills by construction the following properties.

Corollary 6 (Properties of the test space)**.** *The temporal mean value of ϕ_m is one*

$$\bar{\phi}_m = k_m^{-1} \int_{I_m} \phi_m(t)dt = 1. \tag{2.2.7}$$

For $u_k \in V_k^1$ it holds for the time derivative

$$\int_{I_m} \partial_t u_k(t)\phi_m(t)dt = u_k^m - u_k^{m-1} \tag{2.2.8}$$

and multiplication with ϕ_m gives the factors θ and $(1-\theta)$

$$\int_{I_m} u_k(t)\phi_m(t)dt = \theta k_m u_k^m + (1-\theta)k_m u_k^{m-1}. \tag{2.2.9}$$

Equation (2.2.8) *follows directly from* (2.2.7) *because* $u_k \in V_k^1$ *is linear. Equation* (2.2.9) *is a special case of Lemma 5.*

Note that, by construction, the θ time-stepping scheme (2.2.6) is equivalent to the following Galerkin in time formulation. In this context, equivalence means equivalence at the discrete time points. Find $u_k \in V_k^1$ such that

$$\begin{aligned}\sum_{m=1}^{M} \left\{ (u_k^m - u_k^{m-1}, \bar{\varphi}_{k,m}^{\theta}) + k_m \theta (\nabla u_k^m, \nabla \bar{\varphi}_{k,m}^{\theta}) + k_m (1-\theta)(\nabla u_k^{m-1}, \nabla \bar{\varphi}_{k,m}^{\theta}) \right\} \\ + (u_k^0 - u_0, \varphi_{k,0}^{\theta}) = \sum_{m=1}^{M} \left\{ k_m \theta (g(t_m), \bar{\varphi}_{k,m}^{\theta}) + k_m (1-\theta)(g(t_{m-1}), \bar{\varphi}_{k,m}^{\theta}) \right\}\end{aligned} \tag{2.2.10}$$

for all $\varphi_k^{\theta} \in V_k^{\theta}$. We use the $\bar{\cdot}$ notation for the temporal mean $\bar{\varphi}_{k,m}^{\theta} = k_m^{-1} \int_{I_m} \varphi_k(t) dt$.

Navier-Stokes equations

Here we follow [49]. Applied to the Navier-Stokes equations (2.1.5), the θ method leads to the scheme: Find the next approximation $u^m = (v^m, p^m)^T$ by solving

$$\begin{aligned}&(v^m - v^{m-1}, \psi) + k_m (\theta a(v^m)(\psi) + (1-\theta) a(v^{m-1}, \psi)) - k_m b(p^m, \psi) + k_m b(\chi, v^m) \\ &\quad = k_m (\theta g(t_m) + (1-\theta) g(t_{m-1}), \psi)\end{aligned} \tag{2.2.11}$$

for all $\varphi = (\psi, \chi) \in (V^2, L_0^2(\Omega))$ with initial data $v^0 = v_0$. Similarly to the heat equation, we formulate equation (2.2.11) as a Galerkin method. Therefore, we define the ansatz space as the space of linear and continuous functions in time for the velocity field, but keep the pressure discontinuous in time

$$\begin{aligned}X_k^1 = \big\{ &u_k = (v_k, p_k) \in C(\bar{I}, V^2) \times L^2(I, L_0^2(\Omega)) : \\ &v_k|_{I_m} \in P^1(I_m, V^2) \text{ and } p_k|_{I_m} \in P^0(I_m, L_0^2(\Omega)), m = 1, \ldots, M \big\}.\end{aligned}$$

With the function $\phi_m(t)$ from (2.2.5) we define the θ dependency for the test space like before

$$X_k^{\theta} = \{ u_k^{\theta} = (v_k^{\theta}, p_k) \in (V_k^{\theta})^2 \times L^2(I, L_0^2(\Omega)) : p_k|_{I_m} \in P^0(I_m, L_0^2(\Omega)), m = 1, \ldots, M \}.$$

Note that, by construction, (2.2.11) is equivalent to the following Galerkin in time formulation: Find $u_k \in X_k^1$ such that

$$\begin{aligned}\sum_{m=1}^{M} &\left\{(v_k^m - v_k^{m-1}, \bar{\psi}_{k,m}^{\theta}) + k_m\theta a(v_k^m)(\bar{\psi}_{k,m}^{\theta}) + k_m(1-\theta)a(v_k^{m-1})(\bar{\psi}_{k,m}^{\theta})\right. \\ &\left. - k_m b(p_k, \bar{\psi}_{k,m}^{\theta}) + k_m b(\chi_k, v_k^m)\right\} + (v_k^0 - v_0, \psi_{k,0}^{\theta}) \\ &= \sum_{m=1}^{M} \left\{k_m\theta(g(t_m), \bar{\psi}_{k,m}^{\theta}) + k_m(1-\theta)(g(t_{m-1}), \bar{\psi}_{k,m}^{\theta})\right\}\end{aligned} \tag{2.2.12}$$

for all $\varphi_k^{\theta} = (\psi_k^{\theta}, \chi_k) \in X_k^{\theta}$, because the mean value of ϕ_m is one.

Remark 7. *We would like to emphasize that this Galerkin in time formulation of* (2.2.11) *is exactly the same as that used [49]. In Chapter 3.4, we propose a modification.*

2.3. Space discretization

For a domain $\Omega \subset \mathbb{R}^2$, we denote the mesh by $\mathcal{T}_h$. The mesh consists of quasi-uniform quadrilateral cells, K.

2.3.1. Heat equation

We use the following conforming ansatz spaces for the space discretization of the weak form of the heat equation

$$\begin{aligned}V_h^1 &= \{v \in C(\bar{\Omega}) : v|_K \in Q^1(K) \text{ for } K \in \mathcal{T}_h\} \\ V_h^2 &= \{v \in C(\bar{\Omega}) : v|_K \in Q^2(K) \text{ for } K \in \mathcal{T}_h\}.\end{aligned}$$

with $Q^1 := span\{1, x, y, xy\}$ and $Q^2 := span\{1, x, y, xy, x^2, y^2, x^2y, y^2x, x^2y^2\}$. The semi-discrete weak formulation of (2.1.3) for $u_h \in V_h$, with $V_h = V_h^1$ or V_h^2, reads as follows

$$\int_0^T (\partial_t u_h, \varphi_h) + (\nabla u_h, \nabla \varphi_h) dt = \int_0^T (g, \varphi_h) dt \quad \forall \varphi_h \in V_h, \tag{2.3.1}$$

with $u_h(\cdot, 0) = \Pi_h u_0$, where Π_h is the L^2 projection in the discrete space V_h.
By defining the system matrices M and L and using time discretization, we can formulate the full discrete matrix system in terms of the nodal vectors u^m, by using $u^m = u^{m,-}$ or $u^m = u(t_m)$, in dependency of the time discretization. We denote the Lagrange basis functions of the ansatz space V_h, which has m_h degrees of freedom in space, by φ_i for $i = 1, \ldots, m_h$.
Using the mass matrix, $M = (\varphi_j, \varphi_i)_{i,j=1}^{m_h}$, and the stiffness matrix, $L = (\nabla\varphi_j, \nabla\varphi_i)_{i,j=1}^{m_h}$,

we solve the following system when using the DG(0) time-stepping scheme

$$k_m^{-1} M(u^m - u^{m-1}) + Lu^m = \bar{g}|_{I_m} \quad \text{for } m = 1, \dots, M,$$

with $(\bar{g}|_{I_m})_i = k_m^{-1} \int_{I_m} (g(\cdot,t), \varphi_i) dt$ for $i = 1, \dots, m_h$. The quadrature rule for evaluating the time integral is not specified. We will use either the right-sided box rule or the midpoint rule, compare Section 3.4.2 and Remark 28.
For the θ time-stepping, scheme we solve

$$k_m^{-1} M(u^m - u^{m-1}) + \theta Lu^m + (1-\theta) Lu^{m-1} = \theta g^m + (1-\theta) g^{m-1} \quad \text{for } m = 1, \dots, M,$$

with $g_i^m = (g(\cdot, t_m), \varphi_i)$ and $g_i^{m-1} = (g(\cdot, t_{m-1}), \varphi_i)$ for $i = 1, \dots, m_h$.

Remark 8 (Equivalence in discrete time points t_m)**.** *The implicit Euler method ($\theta = 1$) and DG(0) method coincide in the discrete time points t_m, when the source function g is piece-wise constant over each time interval I_m. For a general right-hand side g, they differ by a quadrature error. If $\bar{g}|_{I_m}$ is approximated with the right-sided box rule, they coincide for every g.*

We solve the linear systems using a standard geometric multigrid approach [18, 31].

2.3.2. Navier-Stokes equations

We discretize the semi-discrete formulations (2.2.2) and (2.2.11) in space with continuous finite elements for the velocity field and discontinuous ones for the pressure. It is known from literature [13] that a discretization should fulfill a discrete inf-sup condition, also called Ladyzhenskaya-Babuška-Brezzi condition, which reads

$$\inf_{p_h \in P_h, p_h \neq 0} \sup_{v_h \in V_h, v_h \neq 0} \frac{b(v_h, q_h)}{\|v_h\|_{H^1} \|q_h\|_{L^2}} \geq \beta > 0$$

for a mesh independent constant β. Furthermore, on quadrilateral cells, the so-called $Q_2 P_1^{disc}$ fulfills this condition and is stable, see [23]. This ensures that the saddle point problem, which we specify below, can be solved. We will use this element. The finite element ansatz and test space then read

$$V_h = \{v \in C(\bar{\Omega}) : v|_K \in (Q^2(K))^2 \text{ for } K \in \mathcal{T}_h\}, \quad P_h = \{p \in L^2(\Omega) : p|_K \in P^1(K)\}$$

with mesh $\mathcal{T}_h$ and cell K.
We denote by ψ_i the Lagrange basis functions of the ansatz space V_h, with $2m_h$ degrees of freedom in space, and by χ_i the Lagrange basis functions of the ansatz space P_h, with

n_h degrees of freedom. The mass matrix is given by $M = (\psi_j, \psi_i)_{i,j=1}^{2m_h}$ and the stiffness matrix is $L = (\nabla\psi_j, \nabla\psi_i)_{i,j=1}^{2m_h}$. Matrix $B \in \mathbb{R}^{n_h \times 2m_h}$ is given by $B_{ij} = (\psi_j, \nabla\chi_i)$ for $i = 1,\dots,2m_h, j = 1,\dots,n_h$. For the convection matrix $K = K(v)$ we have $K_{ij}(v) = (v, \nabla\psi_i\psi_j)_{i,j=1}^{2m_h}$. With the system matrices we can formulate the full discrete saddle point system in terms of the nodal vectors $u^m = (v^m, p^m)^T$. We use the notation $p^m = p^{m,-}$ and $v^m = v^{m,-}$ or $v(t_m)$ in dependency of the ansatz space.
We solve the following system, when we use the DG(0) time-stepping scheme

$$\begin{pmatrix} k_m^{-1}M + \vartheta L + K(v^m) & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} v^m \\ p^m \end{pmatrix} = \begin{pmatrix} \bar{g}|_{I_m} + k_m^{-1}Mv^{m-1} \\ 0 \end{pmatrix} \quad \text{for } m = 1,\dots,M$$

with $(\bar{g}|_{I_m})_i = k_m^{-1}\int_{I_m}(g(\cdot,t),\psi_i)dt$ for $i = 1,\dots,2m_h$, evaluated with the right-sided box rule or the midpoint rule, compare Section 3.4.2 and Remark 28.
For the θ time-stepping scheme, we solve

$$\begin{pmatrix} k_m^{-1}M + \theta(\vartheta L + K(v^m)) & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} v^m \\ p^m \end{pmatrix}$$
$$= \begin{pmatrix} k_m^{-1}Mv^{m-1} + \theta g^m + (1-\theta)(g^{m-1} - (\vartheta L + K(v^{m-1}))v^{m-1}) \\ 0 \end{pmatrix} \quad \text{for } m = 1,\dots,M$$

with $g_i^m = (g(\cdot,t_m),\psi_i)$ and $g_i^{m-1} = (g(\cdot,t_{m-1}),\psi_i)$ for $i = 1,\dots,2m_h$. We solve the nonlinear saddle point systems by an outer nonlinear iteration loop using Newton's method [66]. In each nonlinear iteration the linearized system has to be solved. The linearized systems are solved with a geometric multigrid using Vanka like smoother and BiCGSTAB with ILU preconditioner as coarse grid solver [68].

3

Derivation error representations

The goal of this chapter is to derive a posteriori error representations for time discretization errors with respect to an aspect functional J of physical interest. We consider functionals of the following form

$$J(u) = \int_0^T J_1(u(t))dt + J_2(u(T)). \tag{3.0.1}$$

The aim is to measure the difference

$$J(u) - J(u_k),$$

where u is the exact solution of the PDE and u_k is the solution of the time-discrete problem. Therefore, we use the so called dual weighted residual method. The DWR method was proposed by Becker and Rannacher [8, 9] and is based on Lagrange functions. This technique allows us to represent the error in the goal functional J by the Lagrange functionals L of the PDE and $\tilde{L}$ of the time-discrete problem

$$J(u) - J(u_k) = L(u,z) - \tilde{L}(u_k, z_k).$$

Here, the points (u,z) and (u_k, z_k) must be stationary points of the Lagrangians L and $\tilde{L}$, respectively. In this work all derivatives are Gâteaux derivatives. To achieve this, we must solve a dual equation in addition to the primal equation. The variables z and z_k are called dual variables. In general, the Lagrange functional for the time-discrete and continuous problem are not the same.

In order to calculate this difference, we apply an abstract error representation result,

Lemma 9, which we state in Section 3.1. This lemma deals with two different functionals. There is a simpler variant of Lemma 9 dealing with one Lagrange functional, Corollary 10. In the following sections, we apply Lemma 9 or Corollary 10.
We reproduce the known DWR error representations for the heat equation ([15, 48]) in Section 3.2 and for the NSE ([49, 60]) in Section 3.3, considering both time discretization schemes. For the θ schemes, we prove the error representations differently than in [48, 49]. We directly use Lemma 9, which allows us to apply the DWR methodology directly to finite difference methods. This enables us to prove the error representations for the θ method and DG discretization in the same way.
In the final Section 3.4, we propose two modifications of the time discretizations introduced in Section 2.2, which lead to new DWR error representations. The first modification is based on a detailed examination of the error representation proof for the NSE with the θ method, see proof of Theorem 16. The second one deals with modifications to the DG schemes inspired by [27] concerning quadrature.

3.1. Basic theorem for DWR

The main idea of the DWR method is to access the error in the aspect function through Lagrange functionals. Therefore, the following abstract result is the fundamental step in goal-oriented error estimation via the dual weighted residual approach. We slightly modify Lemma 5.1 from [15] to get a better idea of where which assumption is necessary.

Lemma 9 (Abstract error representation). *Let Y be a function space and L and $\tilde{L}$ three times Gâteaux differentiable functionals on Y. Let $y_1 \in Y_1 \subset Y$ be the stationary point of L on Y_1, that means y_1 fulfills*

$$L'(y_1)(\delta y_1) = 0 \quad \forall \delta y_1 \in Y_1.$$

Let $y_2 \in Y_2 \subset Y$ be the stationary point of $\tilde{L}$ on Y_2, that means y_2 fulfills

$$\tilde{L}'(y_2)(\delta y_2) = 0 \quad \forall \delta y_2 \in Y_2.$$

Then we have the error representation

$$\begin{aligned} L(y_1) - \tilde{L}(y_2) = &\frac{1}{2}\left(L'(y_2)(y_1 - \tilde{y}_2) + (L' - \tilde{L}')(y_2)(\tilde{y}_2 - y_2) - L'(y_1)(y_2)\right) \\ &+ (L - \tilde{L})(y_2) + \mathcal{R} \end{aligned} \tag{3.1.1}$$

for arbitrary $\tilde{y}_2 \in Y_2$, with the remainder term $\mathcal{R}$ in terms of the error $e := y_1 - y_2$ given

by

$$\mathcal{R} = \frac{1}{2}\int_0^1 L'''(y_2+se)(e)(e)(e)s(s-1)ds. \tag{3.1.2}$$

Proof. With the main theorem of integral and differential calculus we can write

$$\begin{aligned} L(y_1)-\tilde{L}(y_2) &= L(y_1)-L(y_2)+L(y_2)-\tilde{L}(y_2) \\ &= \int_0^1 L'(y_2+se)(e)ds + L(y_2)-\tilde{L}(y_2) \end{aligned}$$

since both functionals are well-defined and differentiable on Y and $y_1, y_2 \in Y$. Applying the trapezoid rule on the integral, we get

$$\begin{aligned} \int_0^1 L'(y_2+se)(e)ds &= \frac{1}{2}L'(y_2)(e)+\frac{1}{2}L'(y_1)(e)+\frac{1}{2}\int_0^1 L'''(y_2+se)(e)(e)(e)s(s-1)ds \\ &= \frac{1}{2}L'(y_2)(e)+\frac{1}{2}L'(y_1)(e)+\mathcal{R} \end{aligned}$$

with the remainder $\mathcal{R}$ given as in (3.1.2). Now, we specify the occurring derivatives

$$\begin{aligned} & L'(y_2)(y_1-y_2) = L'(y_2)(y_1-\tilde{y})+L'(y_2)(\tilde{y}-y_2) \quad \forall \tilde{y}\in Y \\ \Rightarrow\ & L'(y_2)(y_1-y_2) = L'(y_2)(y_1-\tilde{y}_2)+(L'-\tilde{L}')(y_2)(\tilde{y}_2-y_2) \quad \forall \tilde{y}_2 \in Y_2 \\ & L'(y_1)(y_1-y_2) = L'(y_1)(y_1)-L'(y_1)(y_2) = -L'(y_1)(y_2). \end{aligned}$$

We used that y_2 is stationary point of $\tilde{L}$ in Y_2 and $\tilde{y} \in Y$, as well as y_1 is stationary point of L in Y_1. □

Corollary 10 (**Special case**). *If, in addition to the conditions from Lemma 9 the functional L fulfills*

$$L'(y_1)(y_2) = 0 \quad \forall y_2 \in Y_2 \quad \textit{and} \quad L = \tilde{L}$$

formula (3.1.1) *reduces to*

$$L(y_1)-L(y_2) = \frac{1}{2}L'(y_2)(y_1-\tilde{y}_2)+\mathcal{R}$$

for arbitrary $\tilde{y}_2 \in Y_2$.

This special case is the commonly used variant of the error representation, see, e.g., the book from Bangerth and Rannacher [4].

Remark 11. *For the DG(r) discretizations* (2.2.1) *and* (2.2.2) *we use Corollary 10 since the additional conditions are fulfilled. This is not the case for the* θ *schemes, which is why we use Lemma 9.*

3.2. Heat equation

In this section, we derive error representations with respect to the aspect function J for the heat equation. We make the assumption that the aspect function $J : W(I,V) \to \mathbb{R}$ is three times continuously differentiable, see the assumptions of Lemma 9 , and defined by (3.0.1). We first consider the DG method from Section 2.2.1 and then the θ method from Section 2.2.2. In order to apply Lemma 9, we have to introduce Lagrange functions.

Lagrange function and dual equation

For $u, z \in W(I,V)$ we define the continuous Lagrangian $\mathcal{L}$ by

$$\begin{aligned} &\mathcal{L} : W(I,V) \times W(I,V) \to \mathbb{R} \\ &\mathcal{L}(u,z) := J(u) + \int_0^T (g - \partial_t u, z) - (\nabla u, \nabla z) dt - (u(\cdot,0) - u_0, z(\cdot,0)). \end{aligned} \tag{3.2.1}$$

If we differentiate the Lagrange function according to z, we obtain the residual of the primal PDE in weak form

$$\mathcal{L}_z(u,z)(\varphi) =: \rho(u)(\varphi). \tag{3.2.2}$$

The primal solution fulfills

$$\rho(u)(\varphi) = 0 \quad \forall \varphi \in W(I,V), \tag{3.2.3}$$

which is exactly the solution of the PDE (2.1.3). Differentiating in direction of u we get

$$\begin{aligned} \mathcal{L}_u(u,z)(\varphi) &= J'(u)(\varphi) - \int_0^T (\partial_t \varphi, z) + (\nabla \varphi, \nabla z) dt - (\varphi(0), z(0)) \\ &= \int_0^T J_1'(u)(\varphi) + (\partial_t z, \varphi) - (\nabla \varphi, \nabla z) dt - (z(T), \varphi(T)) + J_2'(u(T))(\varphi(T)) \\ &=: \rho^*(u,z)(\varphi), \end{aligned} \tag{3.2.4}$$

whereby we used partial integration in time. This results in the so-called dual solution $z \in W(I,V)$, which satisfies

$$\rho^*(u,z)(\varphi) = 0 \quad \forall \varphi \in W(I,V). \tag{3.2.5}$$

This corresponds to $z \in W(I,V)$ fulfilling the following dual equation in the weak sense

$$\begin{aligned} -\partial_t z - \Delta z &= J_1'(u) \quad \text{in } \Omega \times (0,T) \\ z &= 0 \quad \text{on } \partial\Omega \times (0,T) \\ z(\cdot,T) &= J_2'(u(T)) \quad \text{in } \Omega. \end{aligned}$$

This equation is also a heat equation, but with a different right-hand side, initial condition, and, especially, it is solved backwards in time. We see that (u,z) with u solution of (3.2.3) and z solution of (3.2.5) is stationary point of the Lagrangian.

DG method

We discretize the heat equation in time with the DG(r) method, see equation (2.2.1). The time discrete Lagrange function $\mathcal{L}^{DG(r)}$ is given by

$$\begin{aligned} &\mathcal{L}^{DG(r)} : V_k^{DG(r)} \times V_k^{DG(r)} \to \mathbb{R} \\ &\mathcal{L}^{DG(r)}(u_k, z_k) := J(u_k) + \sum_{m=1}^{M} \int_{I_m} (g - \partial_t u_k, z_k) - (\nabla u_k, \nabla z_k) dt \\ &\qquad - \sum_{m=0}^{M-1} [u_k]^m z_k^{m,+} - (u_k^{0,-} - u_0, z_k^{0,-}) \end{aligned} \tag{3.2.6}$$

with $u_k, z_k \in V_k^{DG(r)}$. We define the time-discrete residual of the primal equation by

$$\mathcal{L}_z^{DG(r)}(u_k, z_k)(\varphi_k) =: \rho_{DG(r)}(u_k)(\varphi_k). \tag{3.2.7}$$

Obviously, the semi-discrete DG(r) solution for the heat equation (2.2.1) is equivalent to

$$\rho_{DG(r)}(u_k)(\varphi_k) = 0 \quad \forall \varphi_k \in V_k^{DG(r)}. \tag{3.2.8}$$

Differentiating the Lagrange function in direction of u, we get

$$\begin{aligned} \mathcal{L}_u^{DG(r)}(u_k, z_k)(\varphi_k) =& J'(u_k)(\varphi_k) + \sum_{m=1}^{M} \int_{I_m} -(\partial_t \varphi_k, z_k) - (\nabla \varphi_k, \nabla z_k) dt \\ &- \sum_{m=0}^{M-1} [\varphi_k]^m z_k^{m,+} - (\varphi_k^{0,-}, z_k^{0,-}) \\ =& J'(u_k)(\varphi_k) + \sum_{m=1}^{M} \int_{I_m} (\partial_t z_k, \varphi_k) - (\nabla \varphi_k, \nabla z_k) dt \\ &+ \sum_{m=1}^{M-1} (z_k^{m,+} - z_k^{m,-}) \varphi_k^{m,-} + (\varphi_k^{0,-}, z_k^{0,+} - z_k^{0,-}) - (\varphi_k^{M,-}, z_k^{M,-}) \end{aligned}$$

$$
\begin{aligned}
&= J'(u_k)(\varphi_k) + \sum_{m=1}^{M} \int_{I_m} (\partial_t z_k, \varphi_k) - (\nabla \varphi_k, \nabla z_k) dt \\
&\quad + \sum_{m=1}^{M} [z_k]^m \varphi_k^{m,-} - (z_k^{M,+}, \varphi_k^{M,-}) + (\varphi_k^{0,-}, z_k^{0,+} - z_k^{0,-}) \\
&=: \rho^*_{DG(r)}(u_k, z_k)(\varphi_k) = \rho^*_{DG(r)}(z_k)(\varphi_k). \qquad (3.2.9)
\end{aligned}
$$

In the first step, we used partial integration and afterwards reordered the terms. The discrete dual equation

$$
\rho^*_{DG(r)}(z_k)(\varphi_k) = 0 \quad \forall \varphi_k \in V_k^{DG(r)} \qquad (3.2.10)
$$

corresponds to a DG(r) discretization applied to the continuous dual equation (3.2.5), where the upwind flux is used with different sign, because the PDE is solved backwards in time.
Based on the work of Schmich and Vexler [60], we formulate the following error representation.

Theorem 12 (Error representation: heat-DG(r)). *Let $u \in W(I,V)$ be the solution of* (3.2.3) *and $u_k \in V_k^{DG(r)}$ solution of the time semi-discrete problem* (3.2.8). *Let further $z \in W(I,V)$ be solution of* (3.2.5) *and $z_k \in V_k^{DG(r)}$ of* (3.2.10). *Then, we have the following error representation*

$$
J(u) - J(u_k) = \frac{1}{2}\rho_{DG(r)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(r)}(u_k, z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}^{DG(r)}} \qquad (3.2.11)
$$

with arbitrary $\tilde{u}_k, \tilde{z}_k \in V_k^{DG(r)}$. The remainder term $\mathcal{R}_{\mathcal{L}^{DG(r)}}$ is given by (3.1.2), *with $L = \mathcal{L}^{DG(r)}$.*

Proof. Because u, z are solutions of the continuous primal (3.2.3) and dual (3.2.5) equation, $\mathcal{L}'(u,z) = 0$ holds true. Also, $(\mathcal{L}^{DG(r)})'(u_k, z_k) = 0$ for the semi-discrete solutions u_k of (3.2.8) and z_k of (3.2.10). With the Lagrangians (3.2.1) and (3.2.6) we can represent the error in the output functional J

$$
J(u) - J(u_k) = \mathcal{L}(u,z) - \mathcal{L}^{DG(r)}(u_k, z_k) = \mathcal{L}^{DG(r)}(u,z) - \mathcal{L}^{DG(r)}(u_k, z_k)
$$

because the jump terms vanish for the exact solution u, which is global continuous in time. This means that $u \in W(I,V)$ is also a solution of the DG(r) discrete problem. Next, we apply Corollary 10, the special case of Lemma 9, with $L = \tilde{L} = \mathcal{L}^{DG(r)}$, the subspaces

$Y_1 = W(I,V)^2$, $Y_2 = (V_k^{DG(r)})^2$ of $Y = Y_1 + Y_2$ and $\mathcal{L}^{DG(r)} : Y \to \mathbb{R}$. That gives us

$$J(u) - J(u_k) = \frac{1}{2}(\mathcal{L}^{DG(r)})'(u_k, z_k)(u - \tilde{u}_k, z - \tilde{z}_k) + \mathcal{R}_{\mathcal{L}^{DG(r)}}$$

for arbitrary $(\tilde{u}_k, \tilde{z}_k) \in Y_2$. We only have to check that the derivative $(\mathcal{L}^{DG(r)})'(u,z)(u_k, z_k)$ is zero. It holds

$$\mathcal{L}_z^{DG(r)}(u,z)(z_k) = \rho_{DG(r)}(u)(z_k) = \rho(u)(z_k)$$

because the jump terms of u vanish since u is continuous in time. For the test function z_k we have $z_k \in V_k^{DG(r)} \not\subset W(I,V)$. Because $z_k|_{I_m}$ is in $W(I_m,V)$ for $m = 1,\dots,M$ we can conclude that the derivative is zero with a density argument. With the same arguments, $\mathcal{L}_u^{DG(r)}(u,z)(u_k) = 0$ holds true. □

θ time-stepping scheme

In this section, we follow Meidner and Richter [48]. In contrast to the work done there, we apply the abstract result Lemma 9 and not the simplified variant Corollary 10. Thus, we carry out the proof of the error representation differently.
First, we again define the time discrete Lagrangian. We use formulation (2.2.10) of the θ time-stepping scheme, to be able to work in function spaces. The time discrete Lagrangian $\mathcal{L}^\theta$ is given for $u_k \in V_k^1$ and $z_k^\theta \in V_k^\theta$ by

$$\begin{aligned}
&\mathcal{L}^\theta : V_k^1 \times V_k^\theta \to \mathbb{R} \\
&\mathcal{L}^\theta(u_k, z_k^\theta) := J(u_k) + \sum_{m=1}^{M} \big\{ k_m(\theta g^m + (1-\theta)g^{m-1}, \bar{z}_{k,m}^\theta) - (u_k^m - u_k^{m-1}, \bar{z}_{k,m}^\theta) \\
&\qquad - k_m\theta(\nabla u_k^m, \nabla \bar{z}_{k,m}^\theta - k_m(1-\theta)(\nabla u_k^{m-1}, \nabla \bar{z}_{k,m}^\theta) \big\} - (u_k^0 - u_0, z_{k,0}^\theta).
\end{aligned} \tag{3.2.12}$$

We use the superscript m when we evaluate the variable at time t_m. In the case of dual variables, we place the index m in the lower position, as it is not the value at time t^m but the mean value over the time interval I_m. If we define the primal residual as

$$\mathcal{L}_z^\theta(u_k, z_k^\theta)(\varphi_k^\theta) =: \rho_\theta(u_k)(\varphi_k^\theta), \tag{3.2.13}$$

we get the formulation of the θ method as derived in formula (2.2.10) by

$$\rho_\theta(u_k)(\varphi_k^\theta) = 0 \quad \forall \varphi_k^\theta \in V_k^\theta. \tag{3.2.14}$$

Furthermore, we define the dual residual with $\varphi_k \in V_k^1$ as

$$\begin{aligned}\mathcal{L}_u^\theta(u_k, z_k^\theta)(\varphi_k) =& J'(u_k)(\varphi_k) + \sum_{m=1}^{M} \big\{ -(\varphi_k^m - \varphi_k^{m-1}, \bar{z}_{k,m}^\theta) \\ & - k_m\theta(\nabla\varphi_k^m, \nabla\bar{z}_{k,m}^\theta) - k_m(1-\theta)(\nabla\varphi_k^{m-1}, \nabla\bar{z}_{k,m}^\theta)\big\} - (\varphi_k^0, z_{k,0}^\theta). \\ =& \rho_\theta^*(u_k, z_k^\theta)(\varphi_k) = \rho_\theta^*(z_k^\theta)(\varphi_k). \end{aligned} \tag{3.2.15}$$

Again, we define the semi-discrete dual equation by

$$\rho_\theta^*(u_k, z_k^\theta)(\varphi_k) = 0 \quad \forall \varphi_k \in V_k^1. \tag{3.2.16}$$

If we reorder the terms according to the test function φ_k^θ, formula (3.2.16) leads to the following time-stepping scheme for the dual equation

$$\begin{aligned} 0 =& J'(u_k)(\varphi_k) - (\varphi_k^M, \bar{z}_{k,m}^\theta) - k_M\theta(\nabla\varphi_k^M, \nabla\bar{z}_{k,M}^\theta) \\ & + \sum_{m=1}^{M-1} \big\{ (\bar{z}_{k,m+1}^\theta - \bar{z}_{k,m}^\theta, \varphi_k^m) - k_m\theta(\nabla\bar{z}_{k,m}^\theta, \nabla\varphi_k^m) - k_{m+1}(1-\theta)(\nabla\bar{z}_{k,m+1}^\theta, \nabla\varphi_k^m)\big\} \\ & + (\bar{z}_{k,1}^\theta - z_{k,0}^\theta, \varphi_k^0) - k_1(1-\theta)(\nabla\bar{z}_{k,1}^\theta, \nabla\varphi_k^0). \end{aligned} \tag{3.2.17}$$

We sum up the error representation result from [48] in a slightly different way. Instead of directly using the interpolation or projection $(i_k u, \pi_k^\theta z) \in V_k^1 \times V_k^\theta$ of the exact solutions, we use arbitrary functions $(\tilde{u}_k, \tilde{z}_k^\theta) \in V_k^1 \times V_k^\theta$.

Theorem 13 (Error representation: heat-θ). *Let $u \in W(I,V)$ be the solution of* (3.2.3) *and $u_k \in V_k^1$ solution of the semi-discrete problem* (3.2.14)*. Let further be $z \in W(I,V)$ solution of* (3.2.5) *and $z_k^\theta \in V_k^\theta$ solution of* (3.2.16)*. Then, we have the following error representation*

$$J(u) - J(u_k) = \rho(u_k)(z_k^\theta) + \frac{1}{2}\big(\rho(u_k)(z - \tilde{z}_k^\theta) + \rho^*(u_k, z_k^\theta)(u - \tilde{u}_k)\big) + \mathcal{R}_\mathcal{L} + \mathcal{R}_{p,g} \tag{3.2.18}$$

with arbitrary $\tilde{u}_k \in V_k^1$ and $\tilde{z}_k^\theta \in V_k^\theta$. The remainder term $\mathcal{R}_\mathcal{L}$ is given by (3.1.2) *with $L = \mathcal{L}$ and term $\mathcal{R}_{p,g}$ can be limited by*

$$|\mathcal{R}_{p,g}| \le \sum_{m=1}^{M} k_m^2 \|\partial_t^2(g, \tilde{\bar{z}}_k^\theta - \bar{z}_k^\theta)\|_{L^1(I_m,\mathbb{R})}.$$

Proof. Because u, z are solutions of the continuous primal (3.2.3) and dual (3.2.5) equation $\mathcal{L}'(u,z) = 0$ holds true. Also, $(\mathcal{L}^\theta)'(u_k, z_k^\theta) = 0$ for the semi-discrete solutions u_k of (3.2.14) and z_k^θ of (3.2.16). Using Lemma 9 with $L = \mathcal{L}$ and $\tilde{L} = \mathcal{L}^\theta$ as func-

tions from $Y \to \mathbb{R}$ with the space $Y = W(I,V) \times (W(I,V) + V_k^\theta)$ and the subspaces $Y_1 = W(I,V) \times W(I,V)$ and $Y_2 = V_k^1 \times V_k^\theta$ we get

$$\begin{aligned} J(u) - J(u_k) =& \mathcal{L}(u,z) - \mathcal{L}^\theta(u_k, z_k^\theta) \\ =& \frac{1}{2}\big[\underbrace{\mathcal{L}'(u_k, z_k^\theta)(u - \tilde{u}_k, z - \tilde{z}_k^\theta)}_{(I)} + \underbrace{(\mathcal{L}' - \mathcal{L}^{\theta\prime})(u_k, z_k^\theta)(\tilde{u}_k - u_k^\theta, \tilde{z}_k^\theta - z_k^\theta)}_{(II)} \\ & - \underbrace{\mathcal{L}'(u,z)(u_k, z_k^\theta)}_{(III)}\big] + \underbrace{(\mathcal{L} - \mathcal{L}^\theta)(u_k, z_k^\theta)}_{(IV)} + \mathcal{R}_\mathcal{L} \end{aligned}$$

for arbitrary $(\tilde{u}_k, \tilde{z}_k^\theta) \in V_k^1 \times V_k^\theta$. We now look at the individual terms one after the other.
Term (I): Using the definitions of ρ (3.2.2) and ρ^* (3.2.4), we get

$$(I) = \rho(u_k)(z - \tilde{z}_k^\theta) + \rho^*(u_k, z_k^\theta)(u - \tilde{u}_k).$$

Term (IV): Using the definitions of the Lagrange functions (3.2.1) and (3.2.12), we get with $\rho_\theta(u_k)(z_k^\theta) = 0$ that

$$(IV) = J(u_k) + \rho(u_k)(z_k^\theta) - J(u_k) - \rho_\theta(u_k)(z_k^\theta) = \rho(u_k)(z_k^\theta).$$

Term (III): Again, we use the definition of the residuals

$$(III) = \mathcal{L}'_u(u,z)(u_k) + \mathcal{L}'_z(u,z)(z_k^\theta) = \rho^*(u,z)(u_k) + \rho(u)(z_k^\theta).$$

The dual residual ρ^* is zero, because $V_k^1 \subset W(I,V)$. Since $V_k^\theta \subset V_k^{DG(1)}$, the primal residual is zero, see the proof of Theorem 12. Therefore, term (III) is zero.
Term (II): We use the definition of the residuals. Now we also need the definition of ρ_θ (3.2.13) and ρ_θ^* (3.2.15)

$$\begin{aligned} (II) &= \mathcal{L}'(u_k, z_k^\theta)(\tilde{u}_k - u_k, \tilde{z}_k^\theta - z_k^\theta) - \mathcal{L}^{\theta\prime}(u_k, z_k^\theta)(\tilde{u}_k - u_k, \tilde{z}_k^\theta - z_k^\theta) \\ &= \rho(u_k)(\tilde{z}_k^\theta - z_k^\theta) - \rho_\theta(u_k)(\tilde{z}_k^\theta - z_k^\theta) + \rho^*(z_k^\theta)(\tilde{u}_k - u_k) - \rho_\theta^*(z_k^\theta)(\tilde{u}_k - u_k) \\ &= \mathcal{R}_{p,g} + \mathcal{R}_{p,\partial_t} + \mathcal{R}_{p,\Delta} + \mathcal{R}_{d,J'} + \mathcal{R}_{d,\partial_t} + \mathcal{R}_{d,\Delta}, \end{aligned}$$

sorting differences of the terms according to the operators of the PDE. We introduce the subscript p for terms occurring in the primal equation and d as subscript for the dual terms. We now look at the terms one after the other by using the properties of the test space, see Corollary 6, and Lemma 5.

- Terms $\mathcal{R}_{p,\partial_t}$ and $\mathcal{R}_{d,\partial_t}$: Using (2.2.8) in the last step gives

$$\begin{aligned}\sum_{m=1}^{M}\int_{I_m}(\partial_t u_k, \tilde{z}_k^\theta - z_k^\theta)dt &= \sum_{m=1}^{M}\int_{I_m}\partial_t(u_k, \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta)\phi_m dt \\ &= \sum_{m=1}^{M}\left\{(u_k^m, \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta) - (u_k^{m-1}, \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta)\right\}.\end{aligned}$$

  Therefore, $\mathcal{R}_{p,\partial_t} = 0$. An analogous calculation shows $\mathcal{R}_{d,\partial_t} = 0$.

- Terms $\mathcal{R}_{p,\Delta}$ and $\mathcal{R}_{d,\Delta}$: Using (2.2.9) in the last step gives

$$\begin{aligned}\sum_{m=1}^{M}\int_{I_m}(\nabla u_k, \nabla(\tilde{z}_k^\theta - z_k^\theta))dt &= \sum_{m=1}^{M}\int_{I_m}(\nabla u_k, \nabla(\bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta))\phi_m dt \\ &= \sum_{m=1}^{M}\Big\{k_m\theta(\nabla u_k^m, \nabla(\bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta)) \\ &\qquad + k_m(1-\theta)(\nabla u_k^{m-1}, \nabla(\bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta))\Big\}.\end{aligned}$$

  Therefore, $\mathcal{R}_{p,\Delta} = 0$. An analogous calculation shows $\mathcal{R}_{d,\Delta} = 0$.

- Term $\mathcal{R}_{p,g}$: Using Lemma 5 in the last step gives

$$\begin{aligned}\sum_{m=1}^{M}\int_{I_m}(g, \tilde{z}_k^\theta - z_k^\theta)dt &= \sum_{m=1}^{M}\int_{I_m}(g, \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta)\phi_m dt \\ &= \sum_{m=1}^{M}\Big\{k_m\theta(g(t_m), \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta) \\ &\qquad + k_m(1-\theta)(g(t_{m-1}), \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta) \\ &\qquad + \mathcal{R}_m((g, \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta))\Big\}.\end{aligned}$$

  This leads to the additional remainder term in formula (3.2.18)

$$\mathcal{R}_{p,g} = \sum_{m=1}^{M}\mathcal{R}_m((g, \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta)) \leq \sum_{m=1}^{M} k_m^2 \|\partial_t^2(g, \bar{\tilde{z}}_{k,m}^\theta - \bar{z}_{k,m}^\theta)\|_{L^1(I_m,\mathbb{R})}$$

  by Lemma 5.

- Term $\mathcal{R}_{d,J'}$ is zero.

□

Remark 14 (Discussion of error representations heat-DG(r) and heat-θ)**.** *Two additional terms appear in the error representation of the* θ *scheme, as shown in formula* (3.2.18)*. Firstly, the residual* $\rho(u_k)(z_k^\theta)$*. The authors in [48] call this term Galerkin defect. The reason for this is that the exact solution does not solve the discrete equation. The Galerkin orthogonality is therefore lost, which gives the additional term its name. The additional remainder term* $\mathcal{R}_{p,g}$ *is created by taking quadrature errors into account. The DG method, as defined in* (3.2.8)*, does not incorporate quadrature errors. We do so in Section 3.4.*

3.3. Navier-Stokes equations

In this section, we derive the error representations for the Navier-Stokes equations using the same procedure as for the heat equation. Again, we consider the DG method from Section 2.2.1 and the θ method from Section 2.2.2. In order to apply Lemma 9 or Corollary 10, we first introduce the continuous Lagrange function. We also incorporate inflow boundary data and outflow boundary.

Lagrange function and dual equation

With $u = (v,p)$, $z = (w,q)$, $v,w \in W(0,T;H_0^1(\Omega)^2)$ and $q,p \in L^2(0,T;L_0^2(\Omega))$ and X defined in (2.1.1), we define the continuous Lagrangian $\mathcal{L}$ by

$$\begin{aligned}&\mathcal{L} : X \times X \to \mathbb{R}\\ &\mathcal{L}(u,z) = J(u) + \int_0^T (g - \partial_t v, w) - a(v,w) + b(p,w) - b(q,v)dt + (v^0 - v(0), w(0)).\end{aligned} \tag{3.3.1}$$

The aspect function $J : X \to \mathbb{R}$ is three times continuously differentiable (see the assumptions of Lemma 9) and is defined by (3.0.1). We define the primal residual by

$$\mathcal{L}_z(u,z)(\varphi) =: \rho(u)(\varphi). \tag{3.3.2}$$

We can rewrite the primal PDE (2.1.5) as

$$\rho(u)(\varphi) = 0 \quad \forall \varphi \in X. \tag{3.3.3}$$

We define the dual residual as

$$\rho^*(u,z)(\varphi) = \mathcal{L}_u(u,z)(\varphi). \tag{3.3.4}$$

We say, that $z \in X$ is a solution of the dual problem, if

$$\rho^*(u,z)(\varphi) = 0 \quad \forall \varphi \in X. \tag{3.3.5}$$

Now, let's take a closer look at the dual equation. The derivative reads for $\varphi = (\psi, \chi) \in X$

$$\begin{aligned}\rho^*(u,z)(\varphi) =& J'(u)(\varphi) + \int_0^T -(\partial_t \psi, w) - a_v(v,w)(\psi) + b(\chi,w) - b(q,\psi)dt \\ &+ (-\psi(0), w(0)).\end{aligned}$$

The reformulation of the linear terms is as in (3.2.4). For the nonlinearity $a_v(v,z)(\psi)$, we can rewrite

$$a_v(v,z)(\psi) = (\psi \cdot \nabla v, w) + (v \cdot \nabla \psi, w) = (\nabla v \cdot w, \psi) - (v \cdot \nabla w, \psi) \tag{3.3.6}$$

using the Dirichlet zero boundary conditions. Summing up, we can formulate the following strong form of the dual equation

$$\begin{aligned} -\partial_t w + \vartheta \Delta w + (v \cdot \nabla) w - \nabla v \cdot w + \nabla p &= J_1' \quad \text{in } \Omega \times (0,T) \\ \nabla \cdot v &= 0 \quad \text{in } \Omega \times (0,T) \\ v(\cdot, T) &= J_2' \quad \text{in } \Omega. \end{aligned} \tag{3.3.7}$$

Again, the time discrete Lagrange functions differ for the two considered discretizations. Therefore, the definition of the Lagrange functions, dual equations, and derivation of the error estimators corresponding to each discretization are given separately.

DG method

We discretize the Navier-Stokes equations in time with the DG method, see equation (2.2.2). The time discrete Lagrange function $\mathcal{L}^{DG(r)}$ is given by

$$\begin{aligned} &\mathcal{L}^{DG(r)} : X_k^{DG(r)} \times X_k^{DG(r)} \to \mathbb{R} \\ &\mathcal{L}^{DG(r)}(u_k, z_k) := J(u_k) + \sum_{m=1}^{M} \int_{I_m} (g - \partial_t v_k, w_k) - a(v_k)(w_k) + b(p_k, w_k) - b(q_k, v_k) dt \\ &\qquad - \sum_{m=0}^{M-1} [v_k]^m w_k^{m,+} - (v_k^{0,-} - v_0, w_k^{0,-}) \end{aligned} \tag{3.3.8}$$

with $u_k = (v_k, p_k), z_k = (w_k, q_k) \in X_k^{DG(r)}$. We define the primal residual by

$$\mathcal{L}_z^{DG(r)}(u_k, z_k)(\varphi_k) =: \rho_{DG(r)}(u_k)(\varphi_k). \tag{3.3.9}$$

This gives us the DG discretization for the Navier-Stokes equations (2.2.2)

$$\rho_{DG(r)}(u_k)(\varphi_k) = 0 \quad \forall \varphi_k \in X_k^{DG(r)}. \tag{3.3.10}$$

For the derivative in direction of u, we use partial integration in time (as in (3.2.9)) and define with $\varphi_k = (\psi_k, \chi_k)$

$$\begin{aligned}\mathcal{L}_u^{DG(r)}(u_k, z_k)(\varphi_k) =& J'(u_k)(\varphi_k) + \sum_{m=1}^{M} \int_{I_m} (\partial_t w_k, \psi_k) - a_v(v_k, w_k)(\psi_k) + b(\chi_k, w_k) \\ & - b(q_k, \psi_k) dt + \sum_{m=1}^{M} [w_k]^m \psi_k^{m,-} - (w_k^{M,+}, \psi_k^{M,-}) + (\psi_k^{0,-}, w_k^{0,+} - w_k^{0,-}) \\ =:& \rho^*_{DG(r)}(u_k, z_k)(\varphi_k). \end{aligned} \tag{3.3.11}$$

The discrete dual $z_k \in X_k^{DG(r)}$ is solution of

$$\rho^*_{DG(r)}(u_k, z_k)(\varphi_k) = 0 \quad \forall \varphi_k \in X_k^{DG(r)}. \tag{3.3.12}$$

Based on [15], we formulate the following error representation. The statement and the proof are analogous to Theorem 12.

Theorem 15 (Error representation: NSE-DG(r)). *Let $u \in X$ be the solution of* (3.3.3) *and $u_k \in X_k^{DG(r)}$ solution of the time semi-discrete problem* (3.3.10)*. Let further $z \in X$ be solution of* (3.3.5) *and $z_k \in X_k^{DG(r)}$ solution of* (3.3.12)*. Then, we have the following error representation*

$$J(u) - J(u_k) = \frac{1}{2}\rho_{DG(r)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(r)}(u_k, z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}^{DG(r)}}$$

with arbitrary $\tilde{u}_k, \tilde{z}_k \in X_k^{DG(r)}$. The remainder term $\mathcal{R}_{\mathcal{L}^{DG(r)}}$ is given by (3.1.2) *with $\mathcal{L}^{DG(r)}$.*

Proof. Because u, z are solutions of the continuous primal (3.3.3) and dual (3.3.5) equation, $\mathcal{L}'(u,z) = 0$ holds true. Also $(\mathcal{L}^{DG(r)})'(u_k, z_k) = 0$ for the semi-discrete solutions u_k of (3.3.10) and z_k of (3.3.12). With the Lagrangians (3.3.1) and (3.3.8), we can represent the error in the output functional J

$$J(u) - J(u_k) = \mathcal{L}(u,z) - \mathcal{L}^{DG(r)}(u_k, z_k) = \mathcal{L}^{DG(r)}(u,z) - \mathcal{L}^{DG(r)}(u_k, z_k)$$

because the jump terms vanish for the exact solution u, which is global continuous in time. Next, we apply Corollary 10, the special case of Lemma 9, with $L = \tilde{L} = \mathcal{L}^{DG(r)}$,

the subspaces $Y_1 = X^2$, $Y_2 = (X_k^{DG(r)})^2$ of $Y = Y_1 + Y_2$ and $\mathcal{L}^{DG(r)} : Y \to \mathbb{R}$

$$J(u) - J(u_k) = \frac{1}{2}(\mathcal{L}^{DG(r)})'(u_k, z_k)(u - \tilde{u}_k, z - \tilde{z}_k) + \mathcal{R}_{\mathcal{L}^{DG(r)}}$$

for arbitrary $(\tilde{u}_k, \tilde{z}_k) \in Y_2$. We only have to check that the derivative $(\mathcal{L}_{DG(r)})'(u,z)(u_k, z_k)$ is zero

$$\mathcal{L}_z^{DG(r)}(u)(z_k) = \rho_{DG(r)}(u)(z_k) = \rho(u)(z_k)$$

because the jump terms vanish since the velocity v is continuous in time. The test function is not in the continuous function space $z_k \in X_k^{DG(r)} \not\subset X$. However, since z_k is in X on each open time interval I_m, for $m = 1, \ldots, M$, we can use a density argument to conclude that the derivative is zero. With the same arguments $\mathcal{L}_u^{DG(r)}(u,z)(u_k) = 0$ holds true. □

θ time-stepping scheme

We proceed in the same way as in the case of the heat equation. We end with nearly the same result as from Meidner and Richter [49] but again use a different derivation.
First, we again define the time discrete Lagrangian, to apply Lemma 9. We use the formulation (2.2.12) of the θ time-stepping scheme. The time discrete Lagrangian $\mathcal{L}^\theta$ is, with $u_k = (v_k, p_k)^T, z_k^\theta = (w_k^\theta, q_k)^T$, given by

$$\begin{aligned}
&\mathcal{L}^\theta : X_k^1 \times X_k^\theta \to \mathbb{R} \\
&\mathcal{L}^\theta(u_k, z_k^\theta) = J(u_k) + \sum_{m=1}^{M} \big\{ k_m \theta (g(t_m), \bar{w}_{k,m}^\theta) + k_m(1-\theta)(g(t_{m-1}), \bar{w}_{k,m}^\theta) \\
&\qquad - (v_k^m - v_k^{m-1}, \bar{w}_{k,m}^\theta) - k_m \theta a(v_k^m)(\bar{w}_{k,m}^\theta) - k_m(1-\theta)a(v_k^{m-1})(\bar{w}_{k,m}^\theta) \\
&\qquad + k_m b(p_k, \bar{w}_{k,m}^\theta) - k_m b(q_k, v_k^m) \big\} + (v_k^0 - v_0, w_{k,0}^\theta). \qquad (3.3.13)
\end{aligned}$$

If we compute the derivative $\mathcal{L}_z^\theta(u_k, z_k^\theta)(\varphi_k^\theta) =: \rho_\theta(u_k)(\varphi_k^\theta)$ we get by

$$\rho_\theta(u_k)(\varphi_k^\theta) = 0 \quad \forall \varphi_k^\theta \in X_k^\theta \qquad (3.3.14)$$

the formulation of the θ method as variational formulation, as in formula (2.2.10). Again, we define the dual residual

$$\mathcal{L}_u^\theta(u_k, z_k^\theta)(\varphi_k) =: \rho_\theta^*(u_k, z_k^\theta)(\varphi_k).$$

The dual solution fulfills

$$\rho_\theta^*(u_k, z_k^\theta)(\varphi_k) = 0 \quad \forall \varphi_k \in X_k^1. \tag{3.3.15}$$

This leads to the time-stepping method (when we reorder the terms concerning the test function φ_k) with $\varphi_k = (\psi_k, \chi_k) \in X_k^1$

$$\begin{aligned} 0 =& J_2'(u_k^M)(\varphi_k^M) - (\psi_k^M, \bar{w}_{k,M}^\theta) - k_M \theta a(v_k^M, w_{k,M}^\theta)(\psi_k^M) + k_M b(\chi_k^M, \bar{w}_{k,M}^\theta) - k_M b(q_{k,M}^\theta, \psi_k^M) \\ &+ \sum_{m=1}^{M-1} \big\{ (\bar{w}_{k,m+1}^\theta - \bar{w}_{k,m}^\theta, \psi_k^m) - k_m \theta a_v(v_k^m, w_{k,m}^\theta)(\psi_k^m) \\ &\qquad - k_{m+1}(1-\theta) a_v(v_k^{m+1}, w_{k,m+1}^\theta)(\psi_k^m) + k_m b(\chi_k^m, \bar{w}_{k,m}^\theta) - k_m b(q_{k,m}^\theta, \psi_k^m) \big\} \\ &+ (\psi_k^0, \bar{w}_k^{\theta,1} - \bar{w}_k^{\theta,0}) - k_1(1-\theta)(\nabla \psi_k^0, \nabla \bar{w}_k^{\theta,1}) \\ &+ \sum_{m=1}^{M} \int_{I_m} J_1'(u_k)(\varphi_k) dt \end{aligned} \tag{3.3.16}$$

We sum up the error representation result. In contrast to [49] we also do not yet need the interpolation or projection of the exact data at this point.

Theorem 16 (**Error representation: NSE-θ**). *Let $u \in X$ be the solution of* (3.3.3) *and $u_k \in X_k^1$ solution of the time semi-discrete problem* (3.3.14). *Let further $z \in X$ be a weak solution of* (3.3.5) *and $z_k^\theta \in X_k^\theta$ be a solution of* (3.3.15). *Then, we have the following error representation*

$$J(u) - J(u_k) = \rho(u_k)(z_k^\theta) + \frac{1}{2}\big(\rho(u_k)(z - \tilde{z}_k^\theta) + \rho^*(u_k, z_k^\theta)(u - \tilde{u}_k)\big) + \mathcal{R}_\mathcal{L} + \mathcal{R}_p + \mathcal{R}_d \tag{3.3.17}$$

with arbitrary $\tilde{u}_k \in X_k^1$ and $\tilde{z}_k^\theta \in X_k^\theta$. The remainder term $\mathcal{R}_\mathcal{L}$ is given by (3.1.2) *with $L = \mathcal{L}$ and*

$$\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv} + \mathcal{R}_{p,div} \text{ and } \mathcal{R}_d = \mathcal{R}_{d,conv} + \mathcal{R}_{d,div}.$$

For the exact definitions of the individual terms, see the proof.

Proof. Because u, z are solutions of the continuous primal (3.3.3) and dual (3.3.5) equation, $\mathcal{L}'(u,z) = 0$ holds true. Also, $(\mathcal{L}^\theta)'(u_k, z_k^\theta) = 0$ for the semi-discrete solutions u_k of (3.3.14) and z_k^θ of (3.3.15). Using (3.1.1) with $L = \mathcal{L}$ and $\tilde{L} = \mathcal{L}^\theta$ as functions from $Y \to \mathbb{R}$

with the space $Y = X \times (X + X_k^\theta)$ and the subspaces $Y_1 = X \times X$ and $Y_2 = X_k^1 \times X_k^\theta$

$$\begin{aligned} J(u) - J(u_k) =& \mathcal{L}(u,z) - \mathcal{L}^\theta(u_k, z_k^\theta) \\ =& \frac{1}{2}\big[\underbrace{\mathcal{L}'(u_k, z_k^\theta)(u - \tilde{u}_k, z - \tilde{z}_k^\theta)}_{(I)} + \underbrace{(\mathcal{L}' - \mathcal{L}^{\theta\prime})(u_k, z_k^\theta)(\tilde{u}_k - u_k, \tilde{z}_k^\theta - z_k^\theta)}_{(II)} \\ & - \underbrace{\mathcal{L}'(u,z)(u_k, z_k^\theta)}_{(III)}\big] + \underbrace{(\mathcal{L} - \mathcal{L}^\theta)(u_k, z_k^\theta)}_{(IV)} + \mathcal{R}_\mathcal{L} \end{aligned}$$

for arbitrary $\tilde{u}_k \in X_k^1$, $\tilde{z}_k^\theta \in X_k^\theta$. Now, we look at the individual terms one after the other.
Term (I):

$$(I) = \rho(u_k)(z - \tilde{z}_k^\theta) + \rho^*(u_k, z_k^\theta)(u - \tilde{u}_k).$$

Term (IV): Using the definitions of the Lagrange functions and $\rho_\theta(u_k)(z_k^\theta) = 0$, we get

$$(IV) = J(u_k) + \rho(u_k)(z_k^\theta) - J(u_k) - \rho_\theta(u_k)(z_k^\theta) = \rho(u_k)(z_k^\theta).$$

Term (III):

$$(III) = \mathcal{L}'_u(u,z)(u_k) + \mathcal{L}'_z(u,z)(z_k^\theta) = \rho^*(u,z)(u_k) + \rho(u)(z_k^\theta).$$

The dual residual ρ^* is zero, because $X_k^1 \subset X$. Since $X_k^\theta \subset X_k^{DG(1)}$, the primal residual is zero, see the proof of Theorem 13.
Term (II): The second term is

$$\begin{aligned} (II) &= \mathcal{L}'(u_k, z_k^\theta)(\tilde{u}_k - u_k^\theta, \tilde{z}_k^\theta - z_k^\theta) - \mathcal{L}^{\theta\prime}(u_k, z_k^\theta)(\tilde{u}_k - u_k, \tilde{z}_k^\theta - z_k^\theta) \\ &= \rho(u_k)(\tilde{z}_k^\theta - z_k^\theta) - \rho_\theta(u_k)(\tilde{z}_k^\theta - z_k^\theta) + \rho^*(u_k, z_k^\theta)(\tilde{u}_k - u_k) - \rho_\theta^*(u_k, z_k^\theta)(\tilde{u}_k - u_k). \end{aligned}$$

For the difference of the primal and dual residuals, we define

$$\begin{aligned} &\rho(u_k)(\tilde{z}_k^\theta - z_k^\theta) - \rho_\theta(u_k)(\tilde{z}_k^\theta - z_k^\theta) \\ &\qquad = \mathcal{R}_{p,g} + \mathcal{R}_{p,\partial_t} + \mathcal{R}_{p,\Delta} + \mathcal{R}_{p,conv} + \mathcal{R}_{p,div} + \mathcal{R}_{p,pres} \\ &\rho^*(u_k, z_k^\theta)(\tilde{u}_k - u_k) - \rho_\theta^*(u_k, z_k^\theta)(\tilde{u}_k - u_k) \\ &\qquad = \mathcal{R}_{d,J'} + \mathcal{R}_{d,\partial_t} + \mathcal{R}_{d,\Delta} + \mathcal{R}_{d,conv} + \mathcal{R}_{d,div} + \mathcal{R}_{d,pres}. \end{aligned}$$

The terms $\mathcal{R}_{p,g}, \mathcal{R}_{p,\partial_t}, \mathcal{R}_{p,\Delta}, \mathcal{R}_{d,J'}, \mathcal{R}_{d,\partial_t}$ and $\mathcal{R}_{d,\Delta}$ are treated in the proof of Theorem 13. Therefore, we only have to consider the terms with convection, pressure and divergence.

- Terms $\mathcal{R}_{p,pres}$ and $\mathcal{R}_{d,pres}$

$$\sum_{m=1}^{M}\int_{I_m}(p_k,\nabla\cdot(\tilde{w}_k^\theta-w_k^\theta)dt=\sum_{m=1}^{M}\int_{I_m}(p_k,\nabla\cdot(\bar{\tilde{w}}_{k,m}^\theta-\bar{w}_{k,m}^\theta))\phi_m dt$$
$$=\sum_{m=1}^{M}k_m(p_k,\nabla\cdot(\bar{\tilde{w}}_{k,m}^\theta-\bar{w}_{k,m}^\theta))$$

  using (2.2.9) in the last step. Therefore, $\mathcal{R}_{p,pres}=0$. An analog calculation shows $\mathcal{R}_{d,pres}=0$.

- Terms $\mathcal{R}_{p,conv}$ and $\mathcal{R}_{d,conv}$:

$$\sum_{m=1}^{M}\int_{I_m}(\nabla\cdot v_k v_k,(\tilde{w}_k^\theta-w_k^\theta))dt=\sum_{m=1}^{M}\int_{I_m}(\nabla\cdot v_k v_k,\bar{\tilde{w}}_{k,m}^\theta-\bar{w}_{k,m}^\theta)\phi_m dt$$
$$=\sum_{m=1}^{M}\Big\{k_m\theta(\nabla\cdot v_k^m v_k^m,(\bar{\tilde{w}}_{k,m}^\theta-\bar{w}_{k,m}^\theta))$$
$$+k_m(1-\theta)(\nabla\cdot v_k^{m-1}v_k^{m-1},(\bar{\tilde{w}}_{k,m}^\theta-\bar{w}_{k,m}^\theta))+\mathcal{R}_{p,conv}^m\Big\}$$

  with

$$|\mathcal{R}_{p,conv}|\leq\sum_{m=1}^{M}k_m^2||\partial_t^2(\nabla\cdot v_k v_k,(\bar{\tilde{w}}_{k,m}^\theta-\bar{w}_{k,m}^\theta))||_{L^1(I_m,\mathbb{R})}$$

  using Lemma 5 in the last step. A similar consideration as for $\mathcal{R}_{p,conv}$ yields

$$|\mathcal{R}_{d,conv}|\leq\sum_{m=1}^{M}k_m^2||\partial_t^2[(\nabla\cdot v_k(\tilde{v}_k-v_k)+\nabla\cdot(\tilde{v}_k-v_k)v_k,\bar{w}_{k,m}^\theta)]||_{L^1(I_m,\mathbb{R})}.$$

- Terms $\mathcal{R}_{p,div}$ and $\mathcal{R}_{d,div}$:

$$\sum_{m=1}^{M}\int_{I_m}(\nabla\cdot v_k,\tilde{q}_k-q_k)dt$$
$$=\sum_{m=1}^{M}\frac{1}{2}k_m\big\{(\nabla\cdot v_k^m,\tilde{q}_k-q_k)+(\nabla\cdot v_k^{m-1},\tilde{q}_k-q_k)\big\}$$
$$=\sum_{m=1}^{M}\big\{k_m(\nabla\cdot v_k^m,\tilde{q}_k-q_k)+\mathcal{R}_{p,div}^m(\nabla\cdot v_k^m,\tilde{q}_k-q_k)\big\}$$

since $\tilde{q}_k, q_k \in V_k^{DG(0)}$, with

$$|\mathcal{R}_{p,div}| \leq \sum_{m=1}^{M} ck_m ||(\partial_t(\nabla \cdot v_k, \tilde{q}_k - q_k)||_{L^1(I_m,\mathbb{R})}.$$

Analog $|\mathcal{R}_{d,div}| \leq \sum_{m=1}^{M} ck_m ||(\partial_t(\nabla \cdot (\tilde{v}_k - v_k), q_k)||_{L^1(I_m,\mathbb{R})}$. Note that Lemma 5 cannot be applied here.

□

Remark 17 (Discussion error respresentations NSE-DG(r) and NSE-θ)**.** *First, we would like to reiterate the discussion from Remark 14. Additional remainder terms emerge, according to the additional terms of the NSE. All of these terms reflect quadrature errors. Unlike the incompressibility constraint term, no remainder term is produced by the pressure. This is because its dual variable is dependent on* θ*. In the next section, we will make the entire test space dependent on* θ *to eliminate this additional term.*

3.4. Modifications

Remarks 14 and 17 suggest adjustments to both time discretization schemes. In Section 3.4.1, we adapt the θ scheme for the NSE, making the entire test space θ dependent. This leads to a new θ error representation, as in Theorem 16, but without the additional remainder terms for the divergence $\mathcal{R}_{p,div}$ and $\mathcal{R}_{d,div}$.
In Section 3.4.2, we enrich the DG method with quadrature, turning it into a fully time discretized method. As a result, we obtain new DG error representations that incorporate the influence of quadrature errors in the DG schemes. They are comparable to those of the θ schemes.

3.4.1. Incompressibility constraint

We start with modifications of the θ scheme. Therefore, we define

$$P_k^{\theta} = \{p_k \in L^2(I, L_0^2(\Omega)) : p_k|_{I_m} = \phi_m \psi, \psi \in L_0^2(\Omega), m = 1, ..., M\}.$$

Instead of constant test functions $\chi_k|_{I_m} \in P^0$, we choose $\chi_k^{\theta_1} \in P_k^{\theta_1}$ and get with Lemma 5 for $m \in 1, \ldots, M$

$$\begin{aligned} &k_m\theta_1(\bar{\chi}_{k,m}^{\theta_1}, \nabla \cdot v_k^m) + k_m(1-\theta_1)(\bar{\chi}_{k,m}^{\theta_1}, \nabla \cdot v_k^{m-1}) \\ &= \int_{I_m} (\bar{\chi}_{k,m}^{\theta_1}, \nabla \cdot v_k^m)\Phi_m^{\theta_1}(t)dt = \int_{I_m} (\chi_{k,m}^{\theta_1}, \nabla \cdot v_k^m)dt. \end{aligned}$$

With the help of the bilinear form $b(\cdot,\cdot)$, this reads

$$k_m\theta_1 b(\bar{\chi}_{k,m}^{\theta_1}, v_k^m) + k_m(1-\theta_1) b(\bar{\chi}_{k,m-1}^{\theta_1}, v_k^{m-1}) = \int_{I_m} b(\chi_{k,m}^{\theta_1}, v_k^m) dt.$$

The parameter $\theta_1 \in [0,1]$ can be chosen independently of the parameter θ for the time-stepping scheme. To summarize, we use the following Lagrange function $\mathcal{L}^{\theta,mod}$

$$\begin{aligned}
&\mathcal{L}^{\theta,mod} : X_k^1 \times (V_k^\theta)^2 \times P_k^{\theta_1} \to \mathbb{R} \\
&\mathcal{L}^{\theta,mod}(u_k, z_k^{\theta,\theta_1}) = J(u_k) + \sum_{m=1}^{M} \big\{ k_m\theta(g(t_m), \bar{w}_{k,m}^\theta) + k_m(1-\theta)(g(t_{m-1}), \bar{w}_{k,m}^\theta) \\
&\qquad - (v_k^m - v_k^{m-1}, \bar{w}_{k,m}^\theta) - k_m\theta a(v_k^m)(\bar{w}_{k,m}^\theta) - k_m(1-\theta) a(v_k^{m-1})(\bar{w}_{k,m}^\theta) \\
&\qquad + k_m b(p_k, \bar{w}_{k,m}^\theta) - k_m\theta_1 b(\bar{q}_{k,m}^{\theta_1}, v_k^m) - k_m(1-\theta_1) b(\bar{q}_{k,m}^{\theta_1}, v_k^{m-1}) \big\} + (v_k^0 - v_0, w_{k,0}^\theta)
\end{aligned} \tag{3.4.1}$$

with $u_k = (v_k, p_k) \in X_k^1$ and $z_k^{\theta,\theta_1} = (w_k^\theta, q_k^{\theta_1}) \in (V_k^\theta)^2 \times P_k^{\theta_1}$.

Remark 18 (Incompressibility constraint and θ_1)**.** *The choice of* θ_1 *reflects the treatment of the incompressibility constraint. Since we use a fully implicit discretization, we always consider the case of* $\theta_1 = 1$.

Theorem 19 (**Error representation: NSE-θ-mod**)**.** *Let* $u \in X$ *be the solution of* (3.3.3) *and* $u_k \in X_k^1$ *be the solution of the time semi-discrete problem* $\mathcal{L}_z^{\theta,mod} = 0$*. Let further* $z \in X$ *be the solution of* (3.3.5) *and* $z_k^{\theta,\theta_1} \in (V_k^\theta)^2 \times P_k^{\theta_1}$ *be solution of* $\mathcal{L}_u^{\theta,mod} = 0$*. Then, we have the following error representation*

$$J(u) - J(u_k) = \rho(u_k)(z_k^{\theta,\theta_1}) + \frac{1}{2}\big(\rho(u_k)(z - \tilde{z}_k^{\theta,\theta_1}) + \rho^*(u_k, z_k^{\theta,\theta_1})(u - \tilde{u}_k)\big) + \mathcal{R}_\mathcal{L} + \mathcal{R}_p + \mathcal{R}_d \tag{3.4.2}$$

with arbitraty $\tilde{u}_k \in X_k^1$ *and* $\tilde{z}_k^{\theta,\theta_1} \in (V_k^\theta)^2 \times P_k^{\theta_1}$*. The remainder term* $\mathcal{R}_\mathcal{L}$ *is given by* (3.1.2) *with* $L = \mathcal{L}$*. It holds*

$$\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv} \text{ and } \mathcal{R}_d = \mathcal{R}_{d,conv},$$

wherby the terms are given as in the proof of Theorem 16.

Proof. The proof is nearly the same as for Theorem 16. We only discuss the differences, which are the remainder terms for the incompressible constraint $\mathcal{R}_{p,div}$ and $\mathcal{R}_{d,div}$: For

$\tilde{q}_k^{\theta_1}, q_k^{\theta_1} \in P_k^{\theta_1}$ it holds with $\Phi_m^{\theta_1}(t)$

$$\sum_{m=1}^{M} \int_{I_m} (\nabla \cdot v_k, \tilde{q}_k^{\theta_1} - q_k^{\theta_1}) dt = \sum_{m=1}^{M} \int_{I_m} (\nabla \cdot v_k, \tilde{\bar{q}}_k^{\theta_1} - \bar{q}_k^{\theta_1}) \Phi_m^{\theta_1}(t) dt$$
$$= \sum_{m=1}^{M} \left\{ \theta_1 k_m (\nabla \cdot v_k^m, \tilde{\bar{q}}_{k,m}^{\theta_1} - \bar{q}_{k,m}^{\theta_1}) + (1-\theta_1) k_m (\nabla \cdot v_k^{m-1}, \tilde{\bar{q}}_{k,m}^{\theta_1} - \bar{q}_{k,m}^{\theta_1}) \right\}$$

by Lemma 5. This shows $\mathcal{R}_{p,div} = 0$. Analog $\mathcal{R}_{d,div} = 0$. This leads to the following remainder terms

$$\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv}, \quad \mathcal{R}_d = \mathcal{R}_{d,conv}.$$

□

Remark 20. *If $\theta = \theta_1 = 1$, the θ scheme from* (2.2.12) *and the modified one look quite similar. The only difference is the θ dependency of the test space for the pressure. Even though this may not seem reasonable from an interpolation perspective, the remainder terms of this error representation are more consistent with those derived using the modified DG or cGP discretizations. See Theorem 24 and Theorem 39 in the appendix for a comparison.*

3.4.2. Quadrature rule

Usually, the DG(r) discretization is defined as in Chapter 2.2. But the equations (2.2.1) and (2.2.2) contain time integrals. However, this means that a full time discretization is not specified. The book of Ern and Guermond [27] introduces a DG discretization combined with quadrature formulas. This is exactly what we want to do here. We lean on their explanations.

We denote by ω_j the weights and by t^j the nodes of the Gauss-Radau quadrature formula on the interval $[-1,1]$. The weights $\{\omega_j\}_{j=1,\dots,k+1}$ and nodes $\{t^j\}_{j=1,\dots,k+1}$ are available in tabular form and of order $2k$, see [26]. For each m we denote by ω_j^m and t_m^j the on $[t^{m-1}, t^m]$ transformed weights and nodes. The quadrature formula introduces a discrete measure on each time interval $\mu_{k+1}^m(dt)$ with

$$\int_I g(t) dt \approx \sum_{m=1}^{M} \int_{I_m} g(t) \mu_{k+1}^m(dt) := \sum_{m=1}^{M} \sum_{j=1}^{k+1} \omega_j^m g(t_m^j).$$

We replace the time integrals in the DG(r) discretization by integration with respect to the discrete measures. In [27] it is mentioned that the use of the $(r+1)$-point Gauss-Raudau quadrature is the natural choice for the DG(r) discretization. Therefore, we choose $k = r$.

This quadrature formula is useful to us because it enables us to replicate the implicit Euler method when $r = 0$.

Heat equation

With $u_k, z_k \in V_k^{DG(r)}$, we define the modified Lagrange function $\mathcal{L}^{DG(r),mod}(u_k, z_k)$ by

$$\begin{aligned}
&\mathcal{L}^{DG(r),mod}(u_k, z_k) : V_k^{DG(r)} \times V_k^{DG(r)} \to \mathbb{R} \\
&\mathcal{L}^{DG(r),mod}(u_k, z_k) = J(u_k) + \sum_{m=1}^{M} \int_{I_m} g z_k \mu_{r+1}^m(dt) - \int_{I_m} (-\partial_t u_k, z_k) - (\nabla u_k, \nabla z_k)\mu_{r+1}^m(dt) \\
&\qquad\qquad - \sum_{m=0}^{M-1} [u_k]^m z_k^{m,+} - (u_k^{0,-} - u_0, z_k^{0,-}).
\end{aligned}$$

Remark 21 ($r = 0$)**.** *Choosing $r = 0$ the equation*

$$\mathcal{L}^{DG(r),mod}(u_k, z_k)_z(\varphi_k) = 0 \quad \forall \varphi_k \in V_k^{DG(r)}$$

rebuilds the implicit Euler method exact. So we have the same time-stepping scheme for the primal equation as for $\theta = 1$.

We are now able to formulate the following error estimation result.

Theorem 22 (**Error representation: heat-DG(r)-mod**)**.** *Let $u \in W(I,V)$ be the solution of* (3.2.3) *and $u_k \in V_k^{DG(r)}$ solution of the time semi-discrete problem $\mathcal{L}_z^{DG(r),mod} = 0$. Let further $z \in W(I,V)$ be the solution of* (3.2.5) *and $z_k \in V_k^{DG(r)}$ solution of $\mathcal{L}_u^{DG(r),mod} = 0$. Then, we have the following error representation*

$$\begin{aligned}
J(u) - J(u_k) =& \rho_{DG(r)}(u_k)(z_k) + \frac{1}{2}\rho_{DG(r)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(r)}(u_k, z_k)(u - \tilde{u}_k) \\
&+ \mathcal{R}_{\mathcal{L}^{DG(r)}} + \mathcal{R}_{p,g}
\end{aligned}$$

with arbitrary $\tilde{u}_k, \tilde{z}_k \in V_k^{DG(r)}$. The remainder term $\mathcal{R}_{\mathcal{L}^{DG(r)}}$ is given by (3.1.2) *with $L = \mathcal{L}^{DG(r)}$ and*

$$\mathcal{R}_{p,g} = \sum_{m=1}^{M} \Big\{ \int_{I_m} (g, \tilde{z}_k - z_k)dt - \int_{I_m} (g, \tilde{z}_k - z_k)\mu_{r+1}^m(dt) \Big\}.$$

Proof. The proof is analogous to the proof of Theorem 13 using $L = \mathcal{L}^{DG(r)}$ and $\tilde{L} = \mathcal{L}^{DG(r),mod}$. Only the remainder term $\mathcal{R}_{p,g}$ remains here, as we are using a quadrature formula of order $2r$. □

Navier-Stokes equations

With $u_k = (v_k, p_k), z_k = (w_k, q_k) \in X_k^{DG(r)}$, we define the modified Lagrange function $\mathcal{L}^{DG(r),mod}(u_k, z_k)$ by

$$\mathcal{L}^{DG(r)} : X_k^{DG(r)} \times X_k^{DG(r)} \to \mathbb{R}$$
$$\mathcal{L}^{DG(r)}(u_k, z_k) := J(u_k) + \sum_{m=1}^{M} \int_{I_m} (g - \partial_t v_k, w_k) - a(v_k)(w_k) + b(p_k, w_k) - b(q_k, v_k) \mu_{r+1}^m(dt)$$
$$- \sum_{m=0}^{M-1} [v_k]^m w_k^{m,+} - (v_k^{0,-} - v_0, w_k^{0,-}).$$

Remark 23 ($r = 0$)**.** *Choosing $r = 0$, the equation*

$$\mathcal{L}^{DG(r),mod}(u_k, z_k)_z(\varphi_k) = 0 \quad \forall \varphi_k \in X_k^{DG(r)}$$

rebuilds the implicit Euler method exact. So we have the same time-stepping scheme for the primal equations as for $\theta = 1$.

We are now able to formulate the following error estimation result.

Theorem 24 (**Error representation: NSE-DG(r)-mod**)**.** *Let $u \in X$ be the solution of* (3.3.3) *and $u_k \in X_k^{DG(r)}$ solution of the time semi-discrete problem $\mathcal{L}_z^{DG(r),mod} = 0$. Let further $z \in X$ be solution of* (3.3.5) *and $z_k \in X_k^{DG(r)}$ solution of $\mathcal{L}_u^{DG(r),mod} = 0$. Then, we have the following error representation*

$$J(u) - J(u_k) = \rho_{DG(r)}(u_k)(z_k) + \frac{1}{2}\rho_{DG(r)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(r)}(u_k, z_k)(u - \tilde{u}_k)$$
$$+ \mathcal{R}_{\mathcal{L}^{DG(r)}} + \mathcal{R}_p + \mathcal{R}_d$$

with arbitrary $\tilde{u}_k, \tilde{z}_k \in X_k^{DG(r)}$. The remainder term $\mathcal{R}_{\mathcal{L}^{DG(r)}}$ is given by (3.1.2) *with $L = \mathcal{L}^{DG(r)}$ and*

$$\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv} \text{ and } \mathcal{R}_d = \mathcal{R}_{d,conv}.$$

Proof. The proof is analogous to the proof of Theorem 16 using $L = \mathcal{L}^{DG(r)}$ and $\tilde{L} = \mathcal{L}^{DG(r),mod}$. Only the remainder terms $\mathcal{R}_{p,g}, \mathcal{R}_{p,conv}$ and $\mathcal{R}_{d,conv}$ remain here, as we are using a quadrature of order $2r$.

- Term $\mathcal{R}_{p,g}$ is given by

$$\mathcal{R}_{p,g} = \sum_{m=1}^{M} \Big\{ \int_{I_m} (g, \tilde{w}_k - w_k) dt - \int_{I_m} (g, \tilde{w}_k - w_k) \mu_{r+1}^m(dt) \Big\}.$$

- Terms $\mathcal{R}_{p,conv}$ and $\mathcal{R}_{d,conv}$: Term $\mathcal{R}_{p,conv}$ is

$$\mathcal{R}_{p,conv} = \sum_{m=1}^{M} \int_{I_m} (\nabla \cdot v_k v_k, (\tilde{w}_k - w_k))dt - \sum_{m=1}^{M} \int_{I_m} (\nabla \cdot v_k v_k, (\tilde{w}_k - w_k))\mu_{r+1}^{m}(dt)$$

  and $\mathcal{R}_{d,conv}$ is analog given.

□

Remark 25 (Remainder terms $\mathcal{R}_{p,conv}$ and $\mathcal{R}_{d,conv}$ for $r = 0$)**.** *For $r = 0$, the remainder terms caused by the convection term are zero by definition. This is not true for $r > 0$. This means that only the remainder $\mathcal{R}_{p,g}$ is left, which gives the same result as for the heat equation.*

The error representations for the modified DG(r) schemes stated in Theorem 22 and 24 have the same structure as the error representations for the θ method. The additional term, $\rho_{DG(r)}(u_k)(z_k)$, reflects quadrature errors.

4

Practical aspects for adaptive time-step refinement

We now use the error representations as indicators for adaptive time-step adjustment. The general algorithm for adaptive time-step adjustment is as follows.

Choose initial time discretization $\mathcal{I}^0$, set $l = 0$. Repeat until tolerance is reached.

1. Solve primal PDE for $\mathcal{I}^l$.
2. Solve dual PDE for $\mathcal{I}^l$ to compute error estimator on $\mathcal{I}^l$.
3. Construct new time discretization $\mathcal{I}^{l+1}$ with marking and refinement strategy.

We discuss how the steps are implemented numerically, whereby we start with a short comment on the dual equations. Then, we show how to approximate the error representations, which include the exact primal and dual solution, to get computable error estimators. With this we can define the error estimators used in the numerical simulations. This is done in Section 4.2. Step 3 is discussed in Section 4.3.

4.1. Dual equations

In Chapter 3, we did not specify a full discretization of the dual equations, which we need for solving them. We determined semi-discrete formulations of the dual equations by derivatives of the semi-discrete Lagrange functions. To apply the DWR method, we need that the discrete dual solution converges to the continuous one. This is the case for our time discretizations, compare [2, 50]. For the DG schemes, we showed in formula (3.2.9), that the time discretization is again the upwind flux.

It should be noted that, in general, problems can arise when using discretizations in connection with derivatives and dual equations. Issues related to this are often discussed in the context of optimal control problems. For example, the discrete dual derived from differentiating the discrete Lagrangian may lead to an inconsistent discretization of the continuous dual equation. For more information on some aspects, see, e.g., [12, 29, 43, 63].
This is not an issue in our considered PDEs and discretizations. However, it can become one when using stabilization terms in the Navier-Stokes equations. These terms are often used to discretize nonlinear terms when solving PDEs if diffusion is small relative to the grid width. However, these terms are not ideal for differentiation. For this reason, the more general DWR error representation result (Lemma 9) is used in [15] to include stabilization terms only in the space discretization of the primal NSE. We calculate our numerical results without using stabilization terms.
To calculate (the dual solutions and) error estimators, we need the numerical solution of the primal equation over the entire time period $\bar{I}$, because the dual equation is solved backwards in time. Instead of storing the complete primal solution, so-called checkpointing strategies can be used [30, 32]. In this case, the primal solution is only stored at the checkpoints. If the solution is needed between checkpoints, it is recalculated. However, this requires solving parts of the (nonlinear) equations again. This means that it is not always advantageous in terms of effort, see [47]. Alternatively, data compression techniques can be used to approximate the forward solution employed in the backward-in-time integration, see [21]. This is because the entire primal solution is not needed. We refrain from using these techniques and instead store the primal solution over the entire time period.

4.2. Evaluation error estimators

We now derive error estimators from the error representations. The general structure of the error representations from Chapter 3 is as follows

$$J(u)-J(u_k)=\rho(u_k)(z_k)+\frac{1}{2}\left(\rho(u_k)(z-\tilde{z}_k)+\rho^*(u_k,z_k)(u-\tilde{u}_k)\right)+\mathcal{R}_{\mathcal{L}}+\mathcal{R}_p+\mathcal{R}_d.$$

First, we neglect the remainder terms because they are of higher order. To get computable error estimators from the error representations, we have to approximate the primal and dual weights

- $u-\tilde{u}_k$ and $z-\tilde{z}_k$, in case of the DG or modified DG method,
- $u-\tilde{u}_k^\theta$ and $z-\tilde{z}_k^\theta$, for the θ or modified θ scheme,

occurring in the residuals, as they involve the unknown exact primal u and dual z solution. Different options are used in literature to approximate the weights:

1. One is to solve the differential equation on a finer grid and use the more accurate solution for approximation of u resp. z. However, this is expensive and, in terms of adaptivity combined with computational costs, the question arises why one should not proceed directly with discrete solutions on a finer mesh.

2. Alternatively, the equations are solved on the same mesh but with higher order methods. Again, the weights are then approximated by the difference between both solutions. This method also requires a high computational costs.

3. The third option is an approximation using locally defined higher-order interpolations. Because the quantities $\tilde{u}_k, \tilde{u}_k^\theta, \tilde{z}_k$ and $\tilde{z}_k^\theta$ can be chosen arbitrary in their underlying function spaces, the differences are essentially interpolation errors.

We use the latter one and approximate the errors by higher order interpolation in time of the numerical solution. In [4], variants 2 and 3 are compared with each other, and the more complex second variant does not offer any significant advantage. Since we use adaptivity in time, which is one dimensional, we do not have to think about the different techniques for error localization in space presented in [56]. The local errors on one time interval I_m are directly given by evaluation of the weighted residuals over this time interval.
As mentioned in the introduction, we are interested in low order time-stepping schemes. Therefore, we consider all DG(r) schemes with $r = 0$ and all θ time-stepping schemes with $\theta = 1$. Both, the DG(0)-mod discretization and the case $\theta = 1$, lead to the implicit Euler time-stepping scheme. The same is true for the DG(0) method with the right-sided box rule as quadrature formula for the source function.
In Section 4.2.1, we define the interpolation operators. Note that the interpolation operators for the θ scheme are independent of the choice of θ. Combining the interpolation operators and the error representation formulas, is the first step in deriving computable error estimators. This is done in Section 4.2.2 for the heat equation and the NSEs.
At this point, it should be mentioned that we continue to work semi-discrete. The derived error estimators are of course still not numerically evaluable. For the numerical results, however, we assume that the error due to space discretization is negligible. We replace u_k, u_k^θ and z_k, z_k^θ with u_{kh}, u_{kh}^θ and z_{kh}, z_{kh}^θ when evaluating the residuals and weights.

4.2.1. Interpolation operators

We define the used local interpolations in dependency of the time-stepping scheme. Interpolations of one order higher are usually used since higher-order interpolations usually do not offer any benefit.

DG(0) interpolation operator

For the discontinuous Galerkin method, ansatz and test space are the same. Because of this we apply the same interpolation operator for the primal and dual solution and for velocity and pressure. We use the notation $u^m = u_k(t_m^-), z^m = z_k(t_m^-)$ for $m = 0,\ldots,M$. In Figure 4.1 we visualize the used linear interpolation of a DG(0) function. This interpolation is the linear Lagrange interpolation over $\overline{I_m}$. One advantage of this interpolation

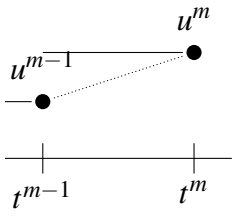


Figure 4.1: Interpolation *lin1* for $V_k^{DG(0)}$ and $X_k^{DG(0)}$.

is that it can be defined locally on a time interval I_m, which is advantageous in terms of adaptivity.

θ scheme interpolation operators

For the θ scheme, ansatz and test space are different. The ansatz space is the space of linear functions V_k^1 in case of the heat equation and $X_k^1 = (V_k^1)^2 \times V_k^{DG(0)}$ for Navier-Stokes equations. Therefore, we have to consider three different discrete function spaces V_k^1 and V_k^θ for the velocity field and $V_k^{DG(0)}$ for the pressure. In the case of the modified NSE-θ error representation, we additionally consider the space $P_k^{\theta_1}$, for which the same interpolation operators as for V_k^θ can be used.
First, we discuss the interpolation operator for the primal velocity $v_k \in V_k^1$. In Figure 4.2, a function $v_k \in V_k^1$ and its local quadratic interpolation is shown. We denote by v^m the value $v_k(t_m)$.

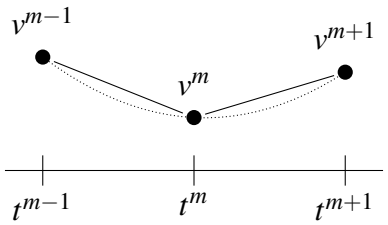


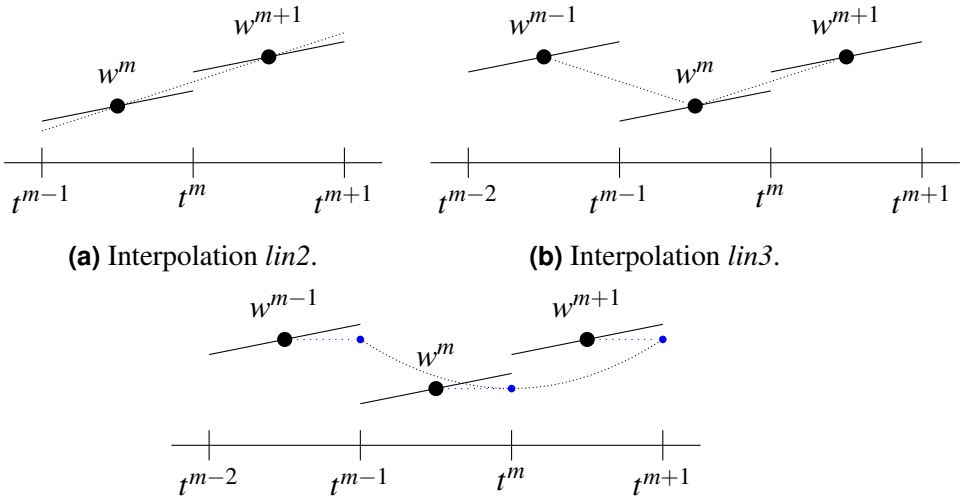
Figure 4.2: Quadratic interpolation *quad* for primal solution $v_k \in V_k^1$.

For the test space V_k^θ, we consider different options for defining an interpolation operator. Due to the construction of the test space, we have a piecewise linear function, but still only one degree of freedom. The degree of freedom is associated with the interval midpoint or

the mean value of w_k on I_m. We remember that $w_k \in V_k^\theta$ is defined by $w_k = \sum_{m=0}^{M} w^m \phi_m(t)$, with coefficient w^m and function $\phi_m(t)$, which defines the slope of w_k in dependency of θ. We visualize the different options of interpolation operators for $w_k \in V_k^\theta$ in Figure 4.3. The two interpolating functions *lin*2 and *quad* are used in the works of Meidner and Richter [48, 49]. *lin*2 for scalar equations and *quad* for the Navier-Stokes equations. Both interpolations are defined over two neighboring time intervals I_m and I_{m+1} and depend on the time-step sizes k_m and k_{m+1}. In the case of the quadratic interpolation *quad*, the degrees of freedom are shifted, which may introduce an error. We consider this interpolation variant only to make sure, that it has no benefit.

Remark 26. *As mentioned in [49], using the interpolations lin*2 *and quad requires, that the time-step sizes of the two neighboring time intervals I_m and I_{m+1} be similar. This limits the adaptive refinement. We analyze this numerically by considering a refinement whereby two neighboring time-step sizes are of the same length.*

To construct the interpolation operator *lin*3 over the time interval I_m, we additionally need the time-step sizes and values from the intervals I_{m-1} and I_{m+1}. Thus, information from three time intervals is required. We examine this interpolation to determine whether it leads to better results on equidistant time grids. In terms of adaptivity, it is not preferable.



(a) Interpolation *lin2*.

(b) Interpolation *lin3*.

(c) Interpolation *quad*.

Figure 4.3: Interpolation operators for dual solution $w_k \in V_k^\theta$.

For interpolating the pressure we have different options. For us, it seems reasonable to associate the degrees of freedom for the pressure of the primal equation p^m with the discrete time points t_m and the degrees of freedom for the pressure of the dual equation q^m with the time interval midpoints $0.5(t_m + t_{m-1})$. Under this assumption, we can use the interpolation variants *lin*1 or *quad* for the primal PDE and *lin*2, *lin*3 or *quad* for the dual PDE.

Remark 27 (Pressure at t_0)**.** *For the interpolation variants $lin1$ or quad we need the pressure at the initial time t_0. Since the pressure fulfills no initial condition, this is not given by definition of the equations. We use $p^0 = p(t_0^-) = 0$ in our calculations.*

4.2.2. Error estimators

We now derive the error estimators for the equations by inserting the interpolation errors into the error representations. As notation we use η for the error estimator and provide it with a superscript for the interpolation used ($lin1, lin2, lin3, quad$) and subscript for the used time discretization. An error estimator then consists of the terms

$$\eta = \eta_\rho + \eta_{\rho^*} + \eta_{g/gal}.$$

Here, η_ρ is the error estimator for the primal equation, η_{ρ^*} is the error estimator for the dual equation, and $\eta_{g/gal}$ is the additional term that appears in the modified DG methods and the θ methods. For the notation and definition of the individual terms, see Table 4.1 for the heat equation and Table 4.2 for the Navier-Stokes equations. We also list the remainder terms that we derived in the error representations and their orders, even though these are not considered numerically.

Remark 28 (Quadrature for DG(0))**.** *In contrast to the other methods, the quadrature formula for solving the PDE using the DG(0) method has not yet been specified. Here, we define two error estimators that use different quadrature formulas. One is the box rule, denoted by the subscript DG,B. Using this quadrature formula results in the same time discretization as the modified DG(0) scheme. For this case, all error estimators use the same time discretization for the primal PDE. The other is the midpoint rule, denoted by the subscript DG,M. We will see why this is relevant later in the numerical results.*

Error estimators heat equation

We give a description of all estimators in Table 4.1. In the numerical studies, we consider linear output functionals J. Since the heat equation is also a linear PDE, the remainder terms $\mathcal{R}_{\mathcal{L}^{DG(0)}}$ and $\mathcal{R}_{\mathcal{L}}$ are zero. Remember, that they are given by formula (3.1.2) and only depend on the third derivative of the Lagrangian. The column method indicates which error representation theorem is used. We apply

- Theorem 12 (heat-DG(r) error representation) for the error estimators $\eta^{lin1}_{DG,B}$ and $\eta^{lin1}_{DG,M}$. The error representation with $r = 0$ is

$$\frac{1}{2}\rho_{DG(0)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(0)}(z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}^{DG(0)}}$$

name	method	primal	weight	dual	weight	add. term	$\mathcal{R}$	order $\mathcal{R}$
$\eta_{DG,B}^{lin1}$	DG(0)[1]	η_{ρ}^{lin1}	lin1	$\eta_{\rho^*}^{lin1}$	lin1	-	-	-
$\eta_{DG,M}^{lin1}$								
η_{mod}^{lin1}	DG(0)-mod	$\eta_{\rho,mod}^{lin1}$	lin1	$\eta_{\rho^*,mod}^{lin1}$	lin1	η_g	$\mathcal{R}_{p,g}$	$\mathcal{O}(k^2)$
η_{θ}^{lin2}	θ ($\theta = 1$)	η_{ρ}^{lin2}	lin2	$\eta_{\rho^*}^{quad}$	quad	η_{gal}	$\mathcal{R}_{p,g}$	$\mathcal{O}(k^3)$
η_{θ}^{lin3}		η_{ρ}^{lin3}	lin3					
η_{θ}^{quad}		η_{ρ}^{quad}	quad					

Table 4.1: Heat equation: Notation error estimators.

and we approximate both weights with the $lin1$ interpolation.

- Theorem 22 (heat-DG(r)-mod error representation) for the error estimator η_{mod}^{lin1}. The error representation with $r = 0$ reads

$$\rho_{DG(r)}(u_k)(z_k) + \frac{1}{2}\rho_{DG(r)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(r)}(u_k, z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}^{DG(r)}} + \mathcal{R}_{p,g}.$$

We approximate both weights using the $lin1$ interpolation. The remainder term $\mathcal{R}_{p,g}$ is for $r = 0$ given by

$$\mathcal{R}_{p,g} = \sum_{m=1}^{M} \Big\{ \int_{I_m} (g, \tilde{z}_k - z_k)dt - \int_{I_m} (g, \tilde{z}_k - z_k)\mu_1^m(dt) \Big\}$$

for $\tilde{z}_k$, $z_k \in V_k^{DG(0)}$ and can be bounded by using $\tilde{z}_k = \Pi_k^{DG(0)} z$, where $\Pi_k^{DG(0)}$ denotes the L^2 projection of z in the space $V_k^{DG(0)}$

$$\begin{aligned} |\mathcal{R}_{p,g}| &\leq \sum_{m=1}^{M} c_B k_m \|\partial_t (g, \Pi_k^{DG(0)} z - z_k)\|_{L^1(I_m,\mathbb{R})} \\ &\leq c_B k \|\partial_t g\|_{L^\infty(I,L^2(\Omega))} \|\Pi_k^{DG(0)}(z - z_k)\|_{L^1(I,L^2(\Omega))} = \mathcal{O}(k^2). \end{aligned}$$

In the first step, we used the approximation result of the quadrature rule. With $k = \max_{m=1,\dots,M} k_m$ the second inequality holds. We can bound the norm of $z - z_k$ by the classical convergence results for the DG method, as in [22].

- Theorem 13 (heat-θ error representation) for the error estimators η_{θ}^{lin2}, η_{θ}^{lin3} and

[1] see Remark 28 for the used quadrature formulas

η_θ^{quad}. The error representation for $\theta = 1$ is

$$\rho(u_k^\theta)(z_k^\theta) + \frac{1}{2}\rho(u_k^\theta)(z - \tilde{z}_k^\theta) + \frac{1}{2}\rho^*(z_k^\theta)(u - \tilde{u}_k^\theta) + \mathcal{R}_\mathcal{L} + \mathcal{R}_{p,g}.$$

We approximate the weight $u - \tilde{u}_k^\theta$ using the quadratic interpolation *quad*. For the weight $z - \tilde{z}_k^\theta$, we choose one of the interpolations $lin2$, $lin3$ or *quad*. The estimator of the additional term $\rho(u_k^\theta)(z_k^\theta)$ is denoted by η_{gal}. In the proof of the heat-θ error representation theorem, we showed that the remainder term $\mathcal{R}_{p,g}$ can be bounded by

$$|\mathcal{R}_{p,g}| \leq \sum_{m=1}^{M} k_m^2 \|\partial_t^2 (g, \tilde{\tilde{z}}_k^\theta - \tilde{z}_k^\theta)\|_{L^1(I_m,\mathbb{R})}.$$

In [48], it is shown that $|\mathcal{R}_{p,g}| = \mathcal{O}(k^3)$ when $\tilde{z}^\theta = \Pi_k^\theta z$ is inserted using the same arguments as for $\mathcal{R}_{p,g}$ in the case of η_{mod}^{lin1}, where results on the convergence of the adjoint for continuous Petrov-Galerkin methods are discussed in [50].

Example 1 (Error estimators $\eta_{DG,B}^{lin1}$ and $\eta_{DG,M}^{lin1}$)**.**

$$\begin{aligned}
\eta_\rho^{lin1} &= \frac{1}{2}\rho_{DG(0)}(u_k)((i_k^{lin1} - id)z_k) \\
&= \frac{1}{2}\sum_{m=1}^{M}\Big\{(u_k^m - u_k^{m-1}, z_k^m - z_k^{m-1}) + \frac{k_m}{2}(\nabla u_k^m, \nabla(z_k^m - z_k^{m-1})) \\
&\qquad + \int_{I_m} \frac{t - t_m}{k_m}(g, z_k^m - z_k^{m-1})dt\Big\}, \\
\eta_{\rho^*}^{lin1} &= \frac{1}{2}\rho_{DG(0)}^*(z_k)((i_k^{lin1} - id)u_k) \\
&= \frac{1}{2}\big[J_2'(u_k)((i_k^{lin1} - id)u_k) \\
&\qquad + \sum_{m=1}^{M}\Big\{\frac{k_m}{2}(\nabla(u_k^m - u_k^{m-1}), \nabla z_k^m) - \frac{k_m}{2}J_1'(u_k^m)(z_k^m - z_k^{m-1})\Big\}\big].
\end{aligned}$$

The estimators, η_ρ^{lin1} and $\eta_{\rho^*}^{lin1}$ are exactly the ones used in [60] for estimating the time discretization error. We evaluate the integral $\int_{I_m}\frac{t-t_m}{k_m}(g, z_k^m - z_k^{m-1})dt$ with the trapezoid rule, when we use the error estimator $\eta_{DG,B}^{lin1}$ and with the simpson rule, when we use the error estimator $\eta_{DG,M}^{lin1}$, because the quadrature rule used here should be of higher order.

Example 2 (Error estimator $\theta = 1$)**.** *For dual residual we evaluate the terms with the simpson rule and get with $t_{m-1/2} = 0.5(t_m + t_{m-1})$*

$$\begin{aligned}
\rho_\theta^*(z_k^\theta)((i_k^{quad} - id)u_k^\theta) &= J'(u_k^\theta)((i_k^{quad} - id)u_k^\theta) \\
&+ \frac{4}{6}\sum_{m=1}^{M} k_m\Big\{(\bar{z}_{k,m}^\theta, (i_k^{quad} - id)u_k^\theta) - (\nabla\bar{z}_{k,m}^\theta, \nabla(i_k^{quad} - id)u_k^\theta)\Big\}|_{t=t_{m-1/2}}
\end{aligned}$$

because the interpolation error for $u_k^\theta \in V_k^1$ is zero in the time points t^m for $m = 0,\dots,M$. For the primal residual, the terms differ for the different interpolations. Again, we evaluate all terms with the simpson rule. As a special case it holds $\int_{I_m}(\partial_t u_k^\theta, (i_k^{lin2} - id)z_{k,m}^\theta)dt = 0$.

Error estimators Navier-Stokes equations

We give an overview of all used estimators in Table 4.2. Because the third derivative $\mathcal{L}$ is not zero, the remainder terms from the Lagrange functional occur. They are third order terms. Furthermore, we limit the interpolation options for the θ schemes that are used numerically. This is done on the basis of the numerical results of the heat equation.

Again, the column method indicates which error representation theorem is used. We

name	method	primal	weight	dual	weight	add. term	$\mathcal{R}$	order $\mathcal{R}$
$\eta_{DG,B}^{lin1}$ / $\eta_{DG,M}^{lin1}$	DG(0)[2]	η_ρ^{lin1}	lin1	$\eta_{\rho^*}^{lin1}$	lin1	-	$\mathcal{R}_{\mathcal{L}^{DG(0)}}$	-
η_{mod}^{lin1}	DG(0) mod	$\eta_{\rho,mod}^{lin1}$	lin1	$\eta_{\rho^*,mod}^{lin1}$	lin1	η_g	$\mathcal{R}_{\mathcal{L}^{DG(0)}} + \mathcal{R}_{p,g}$	$\mathcal{O}(k^2)$
η_θ^{lin1}	θ ($\theta = 1$)	η_ρ^{lin2}	lin2, lin2	$\eta_{\rho^*}^{lin1}$	quad, lin1	η_{gal}	$\mathcal{R}_\mathcal{L} + \mathcal{R}_{p,g} + \mathcal{R}_{p,conv} + \mathcal{R}_{d,conv} + \mathcal{R}_{p,div} + \mathcal{R}_{d,div}$	$\mathcal{O}(k^3 + k^2h^2)$
η_θ^{quad}				$\eta_{\rho^*}^{quad}$	quad, quad			
$\eta_{\theta,mod}^{lin1}$	θ mod ($\theta = 1$)	η_ρ^{lin2}	lin2, lin2	$\eta_{\rho^*}^{lin1}$	quad, lin1	η_{gal}	$\mathcal{R}_\mathcal{L} + \mathcal{R}_{p,g} + \mathcal{R}_{p,conv} + \mathcal{R}_{d,conv}$	$\mathcal{O}(k^3)$
$\eta_{\theta,mod}^{quad}$				$\eta_{\rho^*}^{quad}$	quad, quad			

Table 4.2: Navier-Stokes equations: Notation error estimators.

apply

- Theorem 15 (NSE-DG(r) error representation) for the error estimators $\eta_{DG,B}^{lin1}$ and $\eta_{DG,M}^{lin1}$. The error representation with $r = 0$ is

$$\frac{1}{2}\rho_{DG(0)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(0)}(u_k, z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}^{DG(0)}}$$

  and we approximate both weights with the $lin1$ interpolation. The interpolation is applied to each component of v or w, and to p or q.

[2] see Remark 28 for the used quadrature formulas

- Theorem 24 (NSE-DG(r)-mod error representation) for the error estimator η_{mod}^{lin1}. The error representation with $r = 0$ reads

$$\rho_{DG(0)}(u_k)(z_k) + \frac{1}{2}\rho_{DG(0)}(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*_{DG(r)}(u_k, z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}^{DG(r)}} + \mathcal{R}_p + \mathcal{R}_d.$$

  Again, we use the $lin1$ interpolation operator.
  As mentioned in Remark 25, the remainder terms derived for general r are given by $\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv}$ and $\mathcal{R}_d = \mathcal{R}_{d,conv}$ and simplify for $r = 0$ to $\mathcal{R}_p = \mathcal{R}_{p,g}$ and $\mathcal{R}_d = 0$. Analogous to the heat equation, we can bound the remainder by

$$|\mathcal{R}_{p,g}| \leq \Big| \sum_{m=1}^{M} \Big\{ \int_{I_m} (g, \Pi_k^{DG(0)} w - w_k) dt - \int_{I_m} (g, \Pi_k^{DG(0)} w - w_k) \mu_{r+1}^m(dt) \Big\} \Big| = \mathcal{O}(k^2).$$

- Theorem 16 (NSE-θ error representation) for the error estimators η_θ^{lin1} and η_θ^{quad}. The error representation for $\theta = 1$ is

$$\rho(u_k^\theta)(z_k^\theta) + \frac{1}{2}\rho(u_k^\theta)(z - \tilde{z}_k^\theta) + \frac{1}{2}\rho^*(u_k^\theta, z_k^\theta)(u - \tilde{u}_k^\theta) + \mathcal{R}_\mathcal{L} + \mathcal{R}_p + \mathcal{R}_d.$$

  To approximate the weight $u - \tilde{u}_k^\theta$, we use the interpolation *quad* for each component of v and the interpolations $lin1$ or *quad* for p. To approximate the weight $z - \tilde{z}_k^\theta$, we only use the interpolation $lin2$ for each component of w and for q. The estimator of the additional term $\rho(u_k^\theta)(z_k^\theta)$ is denoted by η_{gal}.
  The remainder terms are given by $\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv} + \mathcal{R}_{p,div}$ and $\mathcal{R}_d = \mathcal{R}_{d,conv} + \mathcal{R}_{d,div}$. In [49], it is stated that $\mathcal{R}_p = \mathcal{O}(k^3 + k^2 h^\alpha)$ and $\mathcal{R}_d = \mathcal{O}(k^3 + k^2 h^\alpha)$ with α the polynomial degree of the velocity space, which is in our case $\alpha = 2$.

- Theorem 19 (NSE-θ-mod error representation) for the error estimators $\eta_{\theta,mod}^{lin1}$ and $\eta_{\theta,mod}^{quad}$. The error representation for $\theta = 1$ is given by

$$\rho(u_k^\theta)(z_k^\theta) + \frac{1}{2}\rho(u_k^\theta)(z - \tilde{z}_k^\theta) + \frac{1}{2}\rho^*(u_k^\theta, z_k^\theta)(u - \tilde{u}_k^\theta) + \mathcal{R}_\mathcal{L} + \mathcal{R}_p + \mathcal{R}_d.$$

  We use the same interpolation operators and notation as used for the NSE-θ error representation.
  The remainder terms are given by $\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv}$ and $\mathcal{R}_d = \mathcal{R}_{d,conv}$. We will now specify which order is assigned to which remainder term as in [49]. Since changes are only necessary if the pressure is affected, we only need to make adjustments in this case. They state that $\mathcal{R}_{p,g} + \mathcal{R}_{p,conv} = \mathcal{O}(k^3)$ and $\mathcal{R}_{p,div} = \mathcal{O}(k^2 h^\alpha)$. As well as $\mathcal{R}_{d,conv} = \mathcal{O}(k^3)$ and $\mathcal{R}_{d,div} = \mathcal{O}(k^2 h^\alpha)$. Because $\mathcal{R}_{p,div}$ and $\mathcal{R}_{d,div}$ are zero for the modified discretization, the remainder terms can be bounded by $\mathcal{O}(k^3)$.

Remark 29 (Non zero Dirichlet boundary conditions)**.** *For the numerical results, we not always consider Dirichlet zero boundary conditions at the whole boundary. If that is the case, then the primal solution is not longer a test function of the dual equation. As a consequence, the derivative $L'(y_1)(e)$ is not zero. In [15], an additional term is considered because of that. We do not consider this term, since we only estimate the time discretization error.*

4.3. Time-step adaption methods

So far, we have derived error estimators that can be used to assign errors locally to a time interval. We denote the local error estimators by η_k, with

$$\eta = \sum_{k=1}^{M} \eta_k$$

and refrain from using further indices referring to the specific error estimators summed up in Tables 4.1 and 4.2. In order to adaptively adjust the time grid, we still need strategies for time-step adaptation. In general, the marking strategy decides which time step should be refined and the refinement rule determines how a single element is subdivided. We start with the marking strategies.

Marking

As marking strategies, we use two variants: a *tolerance based* and a *maximum value based*, which we now specify.

Definition 30 (Tolerance strategy)**.** *Let $\mathcal{I}^l$ be a partition of $\mathcal{I}$ and let η_m^l be error estimators for the time intervals I_m for $m = 1,\dots,M^l$. Let TOL be the specified tolerance. Select all time intervals for refinement, which fulfill*

$$|\eta_m^l| \geq TOL\frac{k_m}{T}.$$

In the book by Verfürth [69], the maximum strategy is defined as follows.

Definition 31 (Maximum strategy)**.** *Let $\mathcal{I}^l$ be a partition of $\mathcal{I}$ and let η_m^l be error estimators for the time intervals I_m for $m = 1,\dots,M^l$. Select a threshold $K_{max} \in (0,1)$. Compute the maximal value of the local error estimators*

$$\eta_m^{l,max} = \max_{m=1,\dots,M^l} |\eta_m^l|.$$

Select all time intervals for refinement, which fulfill

$$|\eta_m^l| \geq K_{max}\eta_m^{l,max}.$$

It is also mentioned in [69] that a well-established choice of K_{max} is $K_{max} \approx 0.5$. In addition, an adjustment to the strategy is suggested in order to reduce the number of iterations of time-step adaption if only a few error estimators have values that are much larger than the others.

Remark 32 (Application maximum strategy)**.** *If a few elements have very large errors, first mark the largest* 5 *or* 10 *percent of the elements with the largest error estimator, and then apply the marking strategy to all remaining error estimators. This prevents only a very small number of elements from being refined and rendering iteration of adaptive time-step control inefficient.*

In contrast to the tolerance based strategy, the refinement of the maximum based strategy is independent of the specified tolerance. However, for the maximum based strategy, all error estimators must be known, whereas with the tolerance based strategy, iterations can take place immediately after calculating a time step. This is not a disadvantage for us, since solving the complete primal equation across all time steps is necessary for calculating the error estimators anyway.

Refinement

As refinement rule we use the *bisection rule*, which means, that a marked time interval is divided into two new time steps of same length.
Having all error estimators at hand naturally results in the possibility of optimizing the adjustment of the time-step sizes. In particular, the old time grid points do not have to be grid points of the new time grid and a more optimal refinement strategy can be chosen to reduce the number of adaptive iterations. There are works [64, 67, 70], where the length of a time step is determined more optimally than with our halving strategy. However, many of these strategies are based on the order of the method. Since we are only considering numerical methods of order 1, this does not open up many new options. This aspect should be taken into account when investigating higher-order time-stepping schemes.

5

Numerical results

The error estimators serve two purposes:

- quantitative assessment of the overall discretization error and
- quantitative adjustment of underlying discretization errors.

We consider both questions separately for the heat equation in Section 5.1, as well as for NSE in Section 5.2. In particular, we compare the error estimators derived by the θ method ($\theta = 1$) with those derived by the DG(0) or DG(0)-mod discretization. For the NSE, we analyze in an additional Section 5.2.1 the influence of the evaluation methods on the mean drag and lift coefficients and compare the modified θ error estimators with the θ error estimators.
We discuss the *quantitative assessment* of the time discretization error for the heat equation in Section 5.1.1 and for the NSE in Section 5.2.2. We measure this with the help of the so-called effectivity index for an aspect functional J

$$I_{eff} = \frac{|\eta|}{|J(u) - J(u_{kh})|},$$

whereby u is the exact solution, u_{kh} the numerical solution and η the error estimator. I_{eff} measures the discrepancy between the estimated and computed error. We expect $I_{eff} \approx 1$ if the error is estimated in agreement to the error, which means that the estimator η gives a good quantitative assessment of the overall error. We should mention that we expect an efficiency index of 1 only if the space discretization error is negligible. To make sure that the space error is negligible, we construct for both equations a numerical example, where the exact solution can be represented by the ansatz space pointwise in time.

For analyzing the *adjustment in time*, we choose other test cases in which the quantities of interest change more significantly over time. Based on the results of the quantitative assessment, we do not consider all error estimators. We compare the marking strategies from Section 4.3. We state the results for adaptive refined time meshes in Section 5.1.2 for the heat equation and in Section 5.2.3 for the NSE.

5.1. Heat equation

For the heat equation we consider two different examples. In both examples we consider

$$J(u) = \int_0^T \int_\Omega u d(x,y) dt$$

as goal functional. Depending on the ansatz space for the solution, V_k^1 or $V_k^{DG(0)}$, we use different quadrature formulas to integrate the time discrete solution exactly. For $u_k \in V_k^1$, we use the trapezoid rule

$$J(u_k) = \sum_{m=1}^{M} \int_\Omega 0.5 k_m \{u_k(t_{m-1}) + u_k(t_m)\} d(x,y)$$

and for $u_k \in V_k^{DG(0)}$, we use the box rule

$$J(u_k) = \sum_{m=1}^{M} k_m \int_\Omega u_k(t_m^-) d(x,y)$$

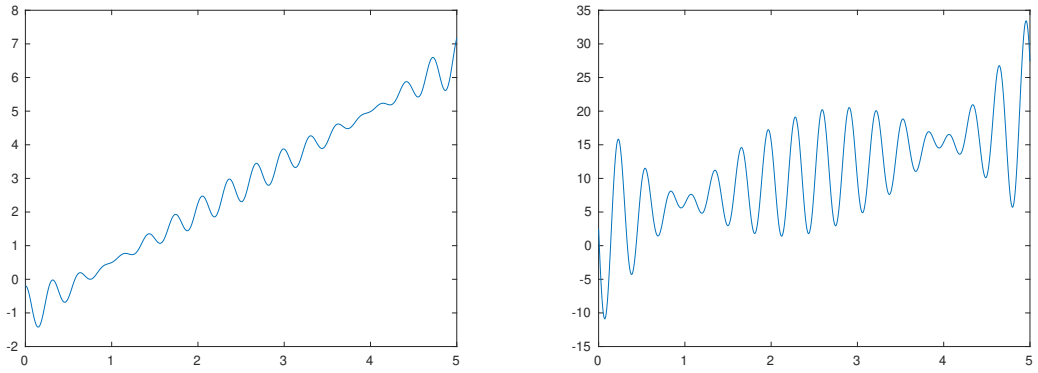
to compute the numerically calculated goal functional. The chosen quadrature rule also influences the effectivity index.
First, we present an example to analyze the effectivity index of the error estimators under equidistant time refinement. The numerical results are presented in Section 5.1.1.

Example 3 (Exponential decrease in time)**.** *We consider the heat equation on* $\Omega = (0,1)^2$ *with* $T = 1$*. With given initial data and source function*

$$\begin{aligned} u_0(x,y) &= x(1-x)y(1-y), \\ g(x,y,t) &= -x(1-x)y(1-y)e^{-t} + 2e^{-t}(y(1-y) + x(1-x)), \end{aligned}$$

the exact solution of the equation is

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



(a) $u(0.5,0.5,t)$. **(b)** $g(0.5,0.5,t)$.

Figure 5.1: Example 4: Behaviour of u and g in time.

The exact mean value of u is

$$J(u) = \int_0^T \int_\Omega u d(x,y) dt \approx 0.017558904411904.$$

Furthermore, when we use the space of quadratic finite elements (FE), it holds $u(t_m,\cdot) \in V_h^2$ for $m = 0,\dots,M$. For linear FE, this is not the case. We refer to linear or quadratic FE when we mean bi-linear or bi-quadratic FE, as we always work in a two-dimensional space. In this example, we consider the effectivity under optimal conditions (quadratic FE) and with the influence of the space discretization error (linear FE) for a very simple problem.

As a second example, we consider a problem with oscillatory behavior in time but with a mean value away from zero. This is used in Section 5.1.2 to analyze adaptive refined time-step sizes and localization of the error. We compare different strategies for marking mentioned in Section 4.3.

Example 4 (Oszillations in time)**.** *The heat equation on* $\Omega = (0,1)^2$ *has the exact solution*

$$u(x,y,t) = \frac{(t-1)(t-4)}{5}\cos(20t) + (1.5t+1)\sin(x+y+0.5)$$

with given initial data u_0 *and source function* g

$$\begin{aligned} u_0(x,y) &= u(x,y,0), \\ g(x,y,t) &= \frac{2t-5}{5}\cos(20t) - 4(t-1)(t-4)\sin(20t) + (3t+3.5)\sin(x+y+0.5). \end{aligned}$$

We consider this example on the time interval up to a final time of $T = 5$*. The time behavior of the solution and source function are visualized in Figure 5.1. The value of*

interest is again the mean value

$$J(u) = \frac{1}{5}\int_0^5 \int_\Omega u(x,y,t)d(x,y)dt \approx 4.353073677838436.$$

For space discretization, we use the domain $[0,1]^2$ as mesh level 0. In each level refinement, the number of elements in space is quadrupled.

5.1.1. Exponential decrease in time (effectivity)

We compare all the error estimators summarized in Table 4.1 for the manufactured solution from Example 3. We provide the values of all error estimator components, as well as the effectivity index I_{eff}.
In all cases, we consider global refinement in time for different space discretizations. Each table has the same structure with subdivisions according to the space discretization. In table part (a), we use quadratic finite elements in space on mesh level 4, in part (b) linear FE on mesh level 4, and in part (c) linear FE on level 6. We point out that we expect a small space discretization error for quadratic FE, resulting in an effectivity index of 1. For linear FE, we expect a smaller space discretization error on finer meshes (mesh level 6) to result in better effectivity, instead of the results on level 4. This means that we can also see the possible effects of the space error on I_{eff}.
We present the results for the DG method in Tables 5.1 and 5.2, and the results for the θ scheme error estimators in Table 5.3.

Comparison $\eta_{DG,B}^{lin1}$ with η_{mod}^{lin1}

Table 5.1 compares the error estimator $\eta_{DG,B}^{lin1}$ with our new estimator η_{mod}^{lin1}. An error estimator usually consists of the weighted primal and dual residuals $\eta_{DG,B}^{lin1} = \eta_{\rho}^{lin1} + \eta_{\rho^*}^{lin1}$. The only difference between the two estimators $\eta_{DG,B}^{lin1}$ and η_{mod}^{lin1} is our proposed additional term η_g, which takes the quadrature errors into account. It holds $\eta_{mod}^{lin1} = \eta_{DG,B}^{lin1} + \eta_g$. See Section 3.4.2 for the exact definitions.
We do not observe an effectivity index of 1 for $\eta_{DG,B}^{lin1}$, even when using quadratic finite elements in space. Incorporating the additional term η_g and computing η_{mod}^{lin1} leads to an effectivity index of around 1 for quadratic FE or linear FE on mesh level 6. This shows that the term η_g cannot simply be neglected. It makes the main contribution to the overall estimated error.

M	η_ρ^{lin1}	$\eta_{\rho^*}^{lin1}$	$\eta_{DG,B}^{lin1}$	I_{eff}	η_g	η_{mod}^{lin1}	I_{eff}
4	2.2054e-04	2.3125e-04	4.5179e-04	0.23	2.0142e-03	2.4659e-03	1.23
8	6.0499e-05	6.2872e-05	1.2337e-04	0.12	1.0094e-03	1.1328e-03	1.11
16	1.8927e-05	1.9482e-05	3.8409e-05	0.07	5.0525e-04	5.4366e-04	1.05
32	6.7679e-06	6.9022e-06	1.3670e-05	0.05	2.5276e-04	2.6643e-04	1.03
64	2.7263e-06	2.7594e-06	5.4857e-06	0.04	1.2641e-04	1.3190e-04	1.01

(a) Quadratic FE on mesh level 4.

M	η_ρ^{lin1}	$\eta_{\rho^*}^{lin1}$	$\eta_{DG,B}^{lin1}$	I_{eff}	η_g	η_{mod}^{lin1}	I_{eff}
4	2.1940e-04	2.2997e-04	4.4937e-04	0.22	2.0051e-03	2.4545e-03	1.18
8	6.0172e-05	6.2527e-05	1.2270e-04	0.11	1.0048e-03	1.1275e-03	1.02
16	1.8818e-05	1.9369e-05	3.8187e-05	0.06	5.0297e-04	5.4116e-04	0.90
32	6.7262e-06	6.8595e-06	1.3586e-05	0.04	2.5162e-04	2.6520e-04	0.77
64	2.7086e-06	2.7414e-06	5.4501e-06	0.03	1.2584e-04	1.3129e-04	0.61

(b) Linear FE on mesh level 4.

M	η_ρ^{lin1}	$\eta_{\rho^*}^{lin1}$	$\eta_{DG,B}^{lin1}$	I_{eff}	η_g	η_{mod}^{lin1}	I_{eff}
4	2.2047e-04	2.3117e-04	4.5164e-04	0.22	2.0136e-03	2.4652e-03	1.23
8	6.0479e-05	6.2851e-05	1.2333e-04	0.12	1.0091e-03	1.1324e-03	1.10
16	1.8920e-05	1.9475e-05	3.8396e-05	0.07	5.0510e-04	5.4350e-04	1.04
32	6.7652e-06	6.8995e-06	1.3665e-05	0.05	2.5269e-04	2.6635e-04	1.00
64	2.7252e-06	2.7582e-06	5.4835e-06	0.04	1.2638e-04	1.3186e-04	0.97

(c) Linear FE on mesh level 6.

Table 5.1: Example 3: Error estimators $\eta_{DG,B}^{lin1}$ and η_{mod}^{lin1} and the effectivity index.

DG with other quadrature $\eta_{DG,M}^{lin1}$

Table 5.2 shows the values for the DG discretization, where the right hand side is evaluated using the midpoint rule. Compared to the values of $\eta_{DG,B}^{lin1}$ in Table 5.1, we see a significant improvement concerning I_{eff}. The effectivity index varies more widely, between 1.15 and 2.9, in the case of quadratic FE. We point out that, unlike to the other methods, I_{eff} is not close to 1 on coarser grids. However, we also note that, when using the midpoint rule, the error and the error estimator converge with order two. This contrasts with the other results and explains why, when using the box rule, the additional term η_g is necessary to reflect the correct behavior of the error estimator. Convergence of the error with order two can be expected in this case, since the convergence order in this example depends on the order of the quadrature formula used.

Comparison of interpolation methods for η_θ

In Table 5.3 we present the error estimators derived with the θ method for $\theta = 1$. We focus on the different interpolation methods for the dual solution, resulting in the estimators denoted by η_ρ^{lin2}, η_ρ^{lin3} and η_ρ^{quad}, see Table 4.1.

M	$\eta_{\rho^*}^{lin1}$	η_{ρ}^{lin1}	$\eta_{DG,M}^{lin1}$	I_{eff}
4	2.5847e-04	7.5005e-05	3.3347e-04	2.90
8	6.6876e-05	2.6308e-05	9.3184e-05	2.06
16	2.0100e-05	1.0631e-05	3.0731e-05	1.56
32	7.0109e-06	4.7242e-06	1.1735e-05	1.29
64	2.7810e-06	2.2191e-06	5.0002e-06	1.15

(a) Quadratic FE on mesh level 4

M	$\eta_{\rho^*}^{lin1}$	η_{ρ}^{lin1}	$\eta_{DG,M}^{lin1}$	I_{eff}
4	2.5706e-04	7.4542e-05	3.3161e-04	1.65
8	6.6509e-05	1.3164e-04	9.2648e-05	0.70
16	1.9984e-05	1.0560e-05	3.0544e-05	0.29
32	6.9675e-06	4.6919e-06	1.1660e-05	0.12
64	2.7629e-06	2.2038e-06	4.9667e-06	0.06

(b) Linear FE on mesh level 4

M	$\eta_{\rho^*}^{lin1}$	η_{ρ}^{lin1}	$\eta_{DG,M}^{lin1}$	I_{eff}
4	2.5838e-04	7.4976e-05	3.3335e-04	2.78
8	6.6853e-05	2.6297e-05	9.3150e-05	1.84
16	2.0093e-05	1.0627e-05	3.0720e-05	1.23
32	7.0081e-06	4.7222e-06	1.1730e-05	0.81
64	2.7799e-06	2.2182e-06	4.9981e-06	0.51

(c) Linear FE on mesh level 6.

Table 5.2: Example 3: Error estimator $\eta_{DG,M}^{lin1}$ and the effectivity index.

There are small differences between the values of η_{ρ}^{lin2} and η_{ρ}^{lin3}. The values of η_{ρ}^{quad} differ slightly more. The choice of the interpolation does not play an important role here. In all cases, we obtain an effectivity index of approximately 1 for quadratic FE and linear on the finer mesh, compare Table 5.3a and Table 5.3c.

Summary

Using $\eta_{DG,B}^{lin1}$ leads to inaccurate results because we cannot estimate the true time discretization error. With a higher-order quadrature formula, such as the midpoint rule used here, estimated and exact error are closer to each other.

The error estimator η_{mod}^{lin1} performs best concerning effectivity, when compared to the other DG error estimators. The contribution of our proposed additional term η_g is significant.

The numerical results for η_{θ} and η_{mod}^{lin1} are comparable.

In all cases, the results in table parts (b) show that the space error directly has an influence on the effectivity. The values are around 0.6 for η_{mod}^{lin1}, η_{θ}^{lin2}, η_{θ}^{lin3} and η_{θ}^{quad} with $M = 64$ but significantly worse for the $\eta_{DG,M}^{lin1}$ estimator.

M	η_ρ^{lin2}	η_ρ^{lin3}	η_ρ^{quad}	$\eta_{\rho^*}^{quad}$	η_{gal}
4	1.1712e-04	1.2089e-04	6.9237e-06	1.3344e-05	1.6915e-03
8	7.0042e-05	7.0468e-05	6.8397e-06	4.8358e-06	9.3579e-04
16	3.7403e-05	3.7416e-05	3.9768e-06	2.2201e-06	4.8765e-04
32	1.9233e-05	1.9215e-05	2.0978e-06	1.0914e-06	2.4845e-04
64	9.7423e-06	9.7332e-06	1.0732e-06	5.4521e-07	1.2535e-04

M	η_θ^{lin2}	I_{eff}	η_θ^{lin3}	I_{eff}	η_θ^{quad}	I_{eff}
4	1.8220e-03	0.91	1.8258e-03	0.91	1.7118e-03	0.86
8	1.0107e-03	0.99	1.0111e-03	0.99	9.4747e-04	0.93
16	5.2727e-04	1.02	5.2729e-04	1.02	4.9384e-04	0.96
32	2.6878e-04	1.03	2.6876e-04	1.03	2.5164e-04	0.97
64	1.3564e-04	1.04	1.3563e-04	1.04	1.2697e-04	0.98

(a) Quadratic FE on mesh level 4.

M	η_ρ^{lin2}	η_ρ^{lin3}	η_ρ^{quad}	$\eta_{\rho^*}^{quad}$	η_{gal}
4	1.0069e-04	1.0458e-04	5.1752e-06	1.2998e-05	1.6838e-03
8	6.1502e-05	6.1929e-05	5.8974e-06	4.6110e-06	9.3157e-04
16	3.3111e-05	3.3125e-05	3.5012e-06	2.0712e-06	4.8545e-04
32	1.7084e-05	1.7065e-05	1.8592e-06	1.0007e-06	2.4733e-04
64	8.6665e-06	8.6575e-06	9.5370e-07	4.9396e-07	1.2478e-04

M	η_θ^{lin2}	I_{eff}	η_θ^{lin3}	I_{eff}	η_θ^{quad}	I_{eff}
4	1.7975e-03	0.86	1.8014e-03	0.87	1.7020e-03	0.82
8	9.9768e-04	0.90	9.9811e-04	0.90	9.4208e-04	0.85
16	5.2063e-04	0.87	5.2064e-04	0.87	4.9102e-04	0.82
32	2.6542e-04	0.77	2.6540e-04	0.77	2.5019e-04	0.72
64	1.3394e-04	0.62	1.3393e-04	0.62	1.2623e-04	0.58

(b) Linear FE on mesh level 4.

M	η_ρ^{lin2}	η_ρ^{lin3}	η_ρ^{quad}	$\eta_{\rho^*}^{quad}$	η_{gal}
4	1.1608e-04	1.1986e-04	6.8139e-06	1.3322e-05	1.6910e-03
8	6.9505e-05	6.9931e-05	6.7804e-06	4.8217e-06	9.3553e-04
16	3.7133e-05	3.7147e-05	3.9469e-06	2.2107e-06	4.8751e-04
32	1.9098e-05	1.9080e-05	2.0828e-06	1.0857e-06	2.4838e-04
64	9.6747e-06	9.6656e-06	1.0657e-06	5.4200e-07	1.2531e-04

M	η_θ^{lin2}	I_{eff}	η_θ^{lin3}	I_{eff}	η_θ^{quad}	I_{eff}
4	1.8204e-03	0.91	1.8242e-03	0.91	1.7112e-03	0.85
8	1.0099e-03	0.98	1.0103e-03	0.98	9.4713e-04	0.92
16	5.2685e-04	1.01	5.2687e-04	1.01	4.9367e-04	0.95
32	2.6857e-04	1.01	2.6855e-04	1.01	2.5155e-04	0.95
64	1.3553e-04	1.00	1.3552e-04	1.00	1.2692e-04	0.94

(c) Linear FE on mesh level 6.

Table 5.3: Example 3: Error estimators η_θ^{lin2}, η_θ^{lin3} and η_θ^{quad} and the effectivity index.

5.1.2. Oscillations in time (adaptivity)

For reasons of clarity, we compare the different adaptive strategies for three of the error estimators from before. We select

- $\eta^{lin1}_{DG,M}$, because using $\eta^{lin1}_{DG,B}$ leads to no good results, as the results from Table 5.1 show.
- η^{lin1}_{mod} as error estimator derived with the modified DG(0) method and comparable with the η_θ error estimators.
- η^{lin2}_θ as error estimator derived with the θ method because the variants η^{lin3}_θ and η^{quad}_θ showed similar results, see Table 5.3, and this estimator uses the most obvious interpolation variant.

Further, we restrict ourselves to space mesh level 6 and quadratic FE. We assign the global marking strategy the values for equidistant refinement in time. We compare the tolerance based marking strategy (tol) with the maximum value based ones (max). We choose the parameters $K_{max} = 0.5$ and 0.8 for the maximum value based method, and apply the method with prior marking of the largest 10 percent and without, see Chapter 4.3. We start with an initial time mesh consisting of $M = 50$ equidistant time steps. The marked time steps are halved.

Comparison of results for η^{lin1}_{mod} and $\eta^{lin1}_{DG,M}$

error est.	marking strategy	mark 10%	M	ratio steps	iter.	range I_{eff}
$\eta^{lin1}_{DG,M}$	global	-	3200	-	6	[0.53, 0.53]
	tol	-	2761	0.86	6	[0.53, 0.61]
	max	×	1981	0.62	21	[0.38, 0.60]
	$K_{max} = 0.5$	✓	1308	0.41	15	[0.42, 0.59]
	max	×	1451	0.45	48	[0.32, 0.59]
	$K_{max} = 0.8$	✓	1541	0.48	32	[0.33, 0.59]
η^{lin1}_{mod}	global	-	3200	-	6	[0.54, 0.59]
	tol	-	2926	0.91	6	[0.51, 0.58]
	max	×	1689	0.53	19	[0.45, 0.54]
	$K_{max} = 0.5$	✓	2018	0.63	15	[0.48, 0.55]
	max	×	1605	0.50	45	[0.45, 0.55]
	$K_{max} = 0.8$	✓	1768	0.55	29	[0.44, 0.55]

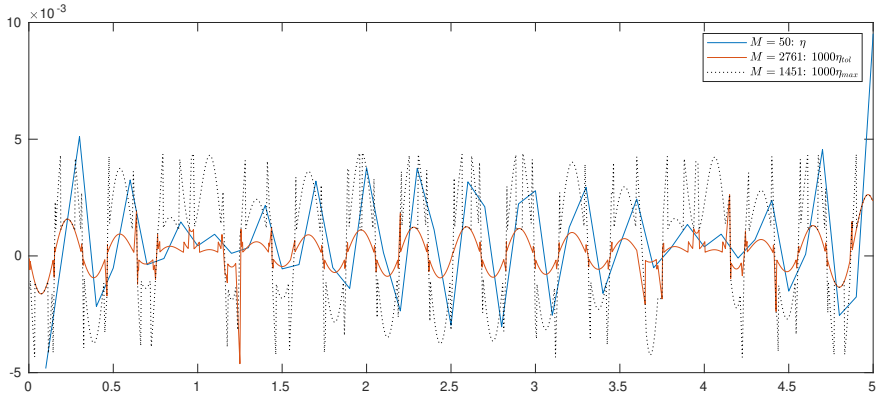
Table 5.4: Example 4: Comparison of marking strategies for $\eta^{lin1}_{DG,M}$ and η^{lin1}_{mod}.

In Table 5.4, we give an overview for the error estimators $\eta^{lin1}_{DG,M}$ and η^{lin1}_{mod} until a tolerance of TOL =1e-03 is reached. The tolerance based strategy requires the fewest number of

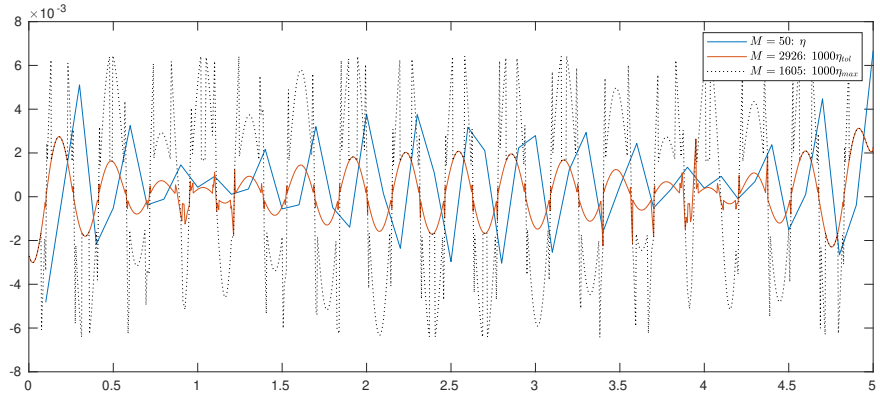
iterations. Here, as many iterations are performed as would be necessary for equidistant refinement, to reach the tolerance. For the maximum based strategies, with $K_{max} = 0.5$ and 0.8, more iterations are needed. The number can be reduced by marking the largest 10 percent of the values in advance. In the best case scenarios, we obtain an average value that is just as accurate with adaptive calculation as with global refinement with half the number of time steps, see the values in column ratio steps. However, this involves greater computational effort than global refinement.

We see, that there is a difference between the estimated errors η and the error of the mean value $J(u) - J(u_{kh})$ in all cases, but the ratio remains relatively constant, see the values in the column range I_{eff}. With range I_{eff} we denote the interval of the minimal and maximal value of I_{eff} computed over all iterations.

Overall, the results for the two error estimators are quite similar. We obtain slightly better values for the effectivity index in the case of estimator η_{mod}^{lin1}.



(a) Error estimator $\eta_{DG,M}^{lin1}$.

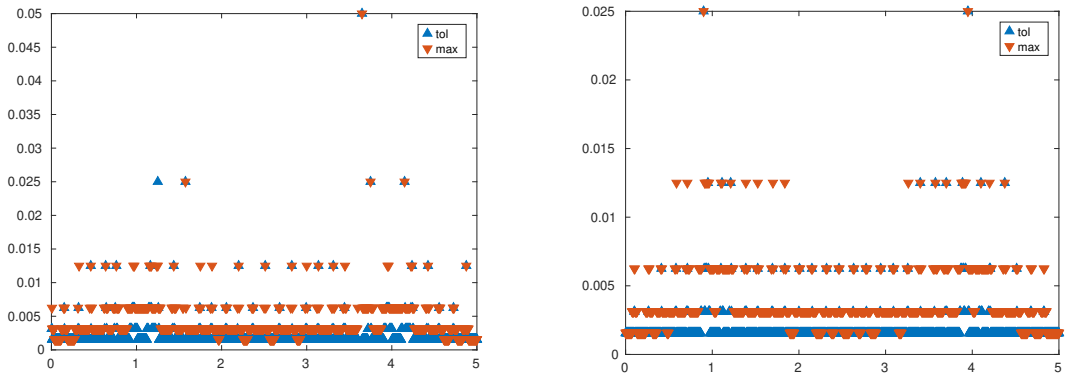


(b) Error estimator η_{mod}^{lin1}.

Figure 5.2: Example 4: Behaviour of local error indicators for $\eta_{DG,M}^{lin1}$ and η_{mod}^{lin1}.

We now take a closer look at the local error indicators and the adjusted time-step sizes. In Figure 5.2 we plot the local error indicators against time. We compare them at the

beginning of the adaptive refinement ($M = 50$) and when the tolerance is reached for the maximum method ($K_{max} = 0.5$, without marking 10 percent) and the tolerance based marking strategy. In the latter case, these are scaled accordingly to enable a comparison of local behavior. We do not observe any major differences between the plots for $\eta^{lin1}_{DG,M}$ and η^{lin1}_{mod}. The error estimators also exhibit oscillatory behavior. This behavior at the end of the adaptive iterations corresponds to the oscillation of the source function g, compare Figure 5.1, if the linear increase of g is ignored. At the start time grid with $M = 50$, the oscillations are shifted. The difference between the two local error indicators can be seen in the adaptive results, where a shift in the oscillations can be observed. In addition, the amplitude for the estimator η^{lin1}_{mod} varies more strongly.



(a) Error estimator $\eta^{lin1}_{DG,M}$.

(b) Error estimator η^{lin1}_{mod}.

Figure 5.3: Example 4: Refined time-step sizes for $\eta^{lin1}_{DG,M}$ and η^{lin1}_{mod}.

In Figure 5.3, we compare the refined time-step sizes for the maximum method with $K_{max} = 0.8$ without marking the biggest 10 percent and for the tolerance based method. Especially around the time points 1 and 4, where the value of the solution changes less significantly, the time-step sizes are less refined.

Results for error estimator η^{lin2}_{θ}

Now, we consider adaptive refinement and error localization for the error estimator η^{lin2}_{θ}. We use a tolerance of $TOL =$ 5e-04. In this case, we analyze whether the two time-step sizes on which the interpolation is defined must be of same length. We add this differentiation to Table 5.5 by adding the column $2k_m$. We mark a X if each time step is refined independently, and a ✓ if $k_m = k_{m+1}$ for $m = 2n-1$, $n = 1, \ldots, \frac{M}{2}$ and M even. In this case, both time steps are marked if the marking condition is satisfied for at least one of them.

From Table 5.5, we can conclude that the two neighboring time steps on which the interpolation is defined should have the same length. All results with $2k_m$ are better than those

error est.	marking strategy	$2k_m$	mark 10%	M	ratio steps	iter.	range I_{eff}
η_θ^{lin2}	global	-	-	1600	-	5	[0.20, 0.63]
	tol	×	-	3163	1.98	6	[0.20, 2.63]
	max	×	×	647	0.40	16	[0.02, 3980.51], [0.02, 1.15][1]
	$K_{max} = 0.5$	×	✓	403	0.25	9	[0.01, 2.65]
	max	×	×	57	0.04	2	[0.00, 0.63]
	$K_{max} = 0.8$	×	✓	196	0.12	13	[0.00, 53.53], [0.09, 3.16][2]
η_θ^{lin2}	tol	✓	-	1594	1.00	5	[0.20, 0.63]
	max	✓	×	426	0.27	6	[0.10, 1.28]
	$K_{max} = 0.5$	✓	✓	378	0.24	5	[0.07, 1.15]
	max	✓	×	180	0.11	9	[0.06, 5.70]
	$K_{max} = 0.8$	✓	✓	370	0.23	11	[0.06, 3.11]

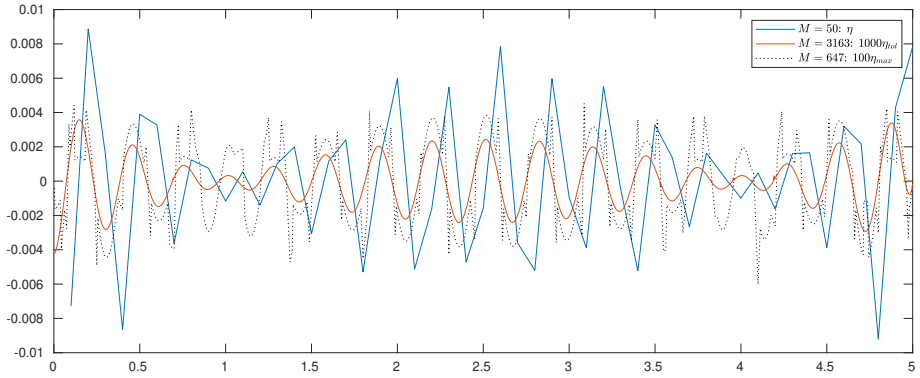
Table 5.5: Example 4: Comparison of marking strategies.

without. Compared to the results in Table 5.4, the effectivity index varies more. This could be explained by the choice of the weight function for the test space. Choosing 1 for the parameter θ yields values ranging from $z_k^\theta(t_{m-1}^+) = -2\bar{z}_k^\theta$ to $z_k^\theta(t_m^-) = 4\bar{z}_k^\theta$. Therefore, the interpolation error of the dual variable is not reduced by refining the time-step sizes. Since the source function oscillates, this scaling can negatively impact the error estimation and explain the difference, especially when non equidistant time-step sizes are considered. It should be mentioned, that [48] apply the θ method numerically to $\theta = 0.5$, which is Crank-Nicolson method or to a shifted Crank-Nicolson method with $\theta = 0.55$. In this cases, the weight function is 1 for Crank-Nicolson method, or next to 1 in the other cases, so that the error $\tilde{z}_k^\theta - z_k^\theta$ can be approximated by the interpolation errors considered in Figure 4.3.

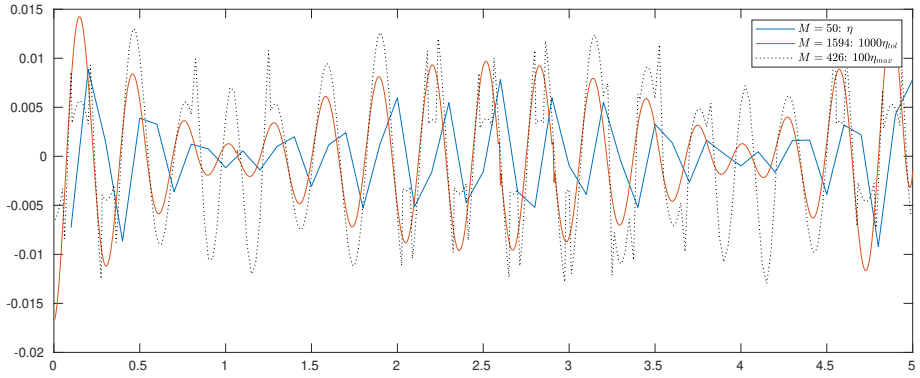
Figure 5.4 shows the localization of the error estimators. We compare the error estimators η_θ^{lin2} with refinement of two neighboring time steps at once and without. In both plots, we use the maximum method with $K_{max} = 0.5$ and without premarking 10% denoted by η_{max}, and the tolerance based strategy denoted by η_{tol}. Apart from the scaling due to the different number of time steps, we cannot detect any differences in the behavior of the error estimators in Figure 5.4a and 5.4b.

Figure 5.5 compares the refined time-step sizes. We denote the results using the maximum value based method with $K_{max} = 0.8$ and without marking 10% by max. In both cases, the tolerance based strategy demonstrates nearly global refinement. The max marking strategy uses six different time-step sizes in Figure 5.5a and three in Figure 5.5b. This difference in adjustment is explained by the different total number of iterations ($M = 16$ versus $M = 6$). In both cases, the least refinement occurs around $t = 1$ and 4, which is consistent with the plots in Figure 5.4.

[1] removed one value of error estimator, because value was greater 1
[2] removed two values of error estimator, because values changed by factor 100



(a) Error estimator η_θ^{lin2} without $2k_m$.



(b) Error estimator η_θ^{lin2} with $2k_m$.

Figure 5.4: Example 4: Behaviour of local error indicators for η_θ^{lin2}.

Summary

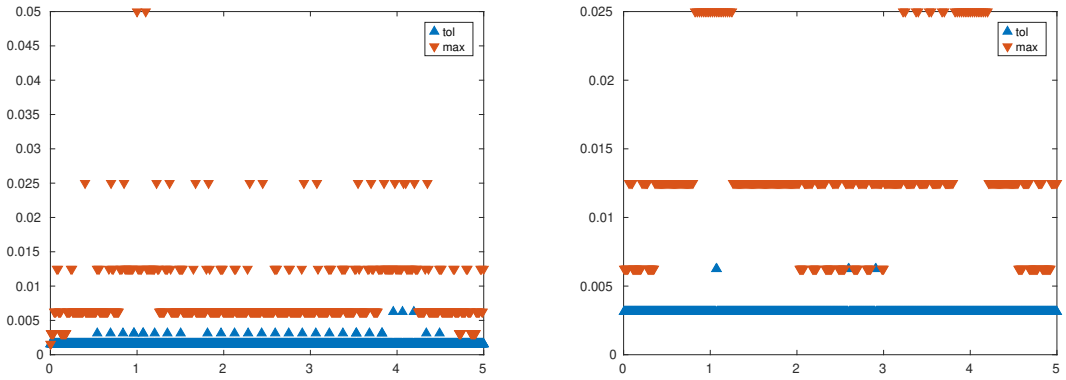
The results indicate that the tolerance based marking strategy and the maximum based with $K_{max} = 0.5$ (with 10% premarking) should be used, which corresponds to the recommendation from Verfürth's book [69]. Using $K_{max} = 0.8$ does not offer any real benefit in time-step adjustment and requires many more iterations.

Adding the additional term through the DG-mod error representation slightly improves effectivity at the expense of the number of time steps, but does not influence the localization of the error estimator.

We see that one must be careful when applying general θ error estimators to the implicit Euler time-stepping scheme. Applying the θ dependency for $\theta = 1$ can negatively affect the effectivity index.

The natural interpretation of the implicit Euler scheme as a DG(0) method appears to be better and it also seems preferable to use the modified estimator η_{mod}^{lin1}.

Remark 33 (Boundary conditions and effectivity)**.** *In this example, we do not use Dirichlet zero boundary data. This could explain why the effectivity index of 1 is not achieved,*



(a) Error estimator η_θ^{lin2} without $2k_m$.

(b) Error estimator η_θ^{lin2} with $2k_m$.

Figure 5.5: Example 4: Refined time-step sizes for η_θ^{lin2}.

compare [15]. Another reason we cannot observe an effectivity index of 1 is that this error may be introduced by the space discretization. We are comparing with the mean value of the exact solution. However, in this example, the exact solution cannot be represented by quadratic finite elements pointwise in time. Since our focus is on adaptivity in time, we will not examine this issue further. However, this should be taken into account for adaptivity in space.

5.2. Navier-Stokes equations

In the case of Navier-Stokes equations, we are interested in controlling the error of the mean drag and lift values. We consider two examples. We use both examples in Section 5.2.1 to analyze the evaluation of the mean drag and lift coefficients. With the help of the manufactured solution, Example 5, we compare the error estimators and their effectivity index under global time-step refinement. Finally, we use a well known benchmark problem, Example 6, for adaptive time-step control.
First, we define drag and lift coefficients using the example of flow through a channel section around a cylinder with surface S. The setup is shown in Figure 5.6.

Definition 34 (Drag and lift coefficients)**.** *The drag coefficient c_d and the lift coefficient c_l are time dependent quantities, defined as follows*

$$\begin{aligned} c_d(t) &= \frac{2}{\rho L U_{max}^2} \int_S \left(\rho \vartheta \frac{\partial v_{t_S}(t)}{\partial n} n_y - p(t) n_x \right) dS \\ c_l(t) &= \frac{-2}{\rho L U_{max}^2} \int_S \left(\rho \vartheta \frac{\partial v_{t_S}(t)}{\partial n} n_x + p(t) n_y \right) dS. \end{aligned} \tag{5.2.1}$$

Thus, $n = (n_x, n_y)^T$ is the outer normal vector on the circular boundary S of the cylinder,

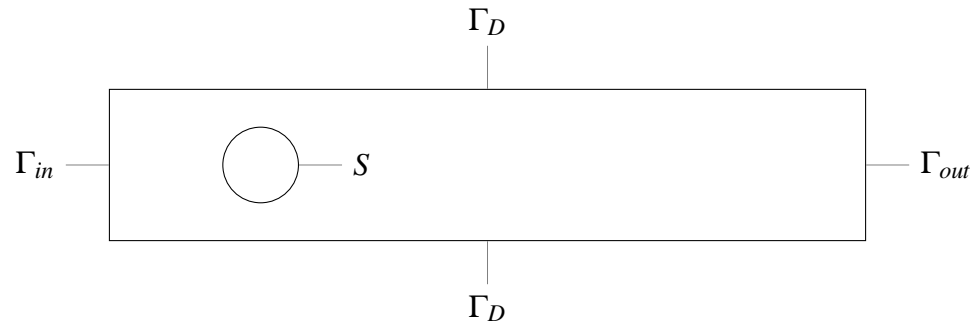


Figure 5.6: Channel section with circular boundary S.

$t_S = (n_y, -n_x)^T$ is the tangential vector on S, and v_{t_S} is the tangential velocity. Additionally, ϑ is the kinematic viscosity, L is the cylinder diameter, and U_{max} is the maximum inflow velocity. Furthermore, the fluid density, which is set to 1 in our case, is denoted by ρ and will be dropped in further considerations.

By this, we define the goal functionals, the mean drag and lift coefficient, as

$$\begin{aligned} J_d(u) &= \frac{1}{T}\int_0^T c_d(t)dt \approx \frac{1}{T}\sum_{m=1}^{M} k_m c_d(t_m) \\ J_l(u) &= \frac{1}{T}\int_0^T c_l(t)dt \approx \frac{1}{T}\sum_{m=1}^{M} k_m c_l(t_m) \end{aligned} \tag{5.2.2}$$

and approximate them with the right-sided box rule. This quadrature rule integrates functions $u_k \in X_k^{DG(0)}$ exactly, but not functions $u_k \in X_k^1$.
In [49], it is suggested to use the trapezoid rule for the θ methods to integrate $u_k \in X_k^1$ exactly. But, they consider θ near 0.5 in their numerical experiments. In the case of the NSE, our numerical experiments showed, that the error estimator applied to $\theta = 1$ converges against the reference value computed with the box rule. Therefore, we use the box rule for evaluation of the mean drag and lift values here, too. That makes sense in the context of the considered time-stepping scheme, the implicit Euler method, even if we do not integrate the associated solution in the function space exactly.

We now give the test examples. We start with Example 5.

Example 5 (Flow through a square)**.** *We consider flow through the square $S = [1/3, 2/3]^2$. For the area $\Omega = [0,1]^2 \setminus S$, we prescribe inflow boundary conditions on the left edge Γ_{in}. At the upper and lower boundaries Γ_D, we set Dirichlet zero data and Γ_{out} denotes the outflow boundary. The region with boundaries is shown in Figure 5.7(a). On Ω, we solve*

the NSE with $\vartheta = 0.001$. *We set the velocity field and pressure*

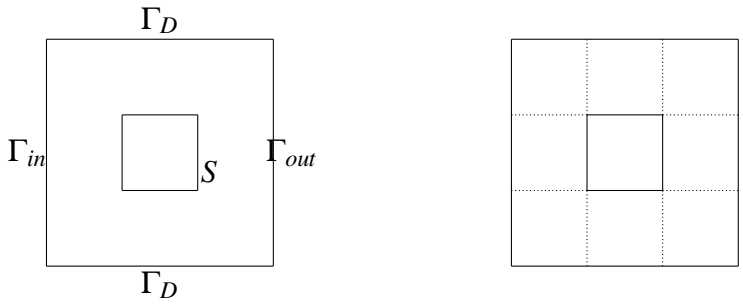
$$v(x,y,t) = \begin{pmatrix} 4y(1-y)1.5\sin(\frac{\pi}{2}t) \\ 0 \end{pmatrix}, \quad p(x,y,t) = (1-x)\sin(\frac{\pi}{2}t)$$

so that this is a solution of the NSE to the right-hand side

$$f(x,y,t) = \begin{pmatrix} -\frac{247}{250}\sin(\frac{\pi}{2}t) - 3\pi\cos(\frac{\pi}{2}t)y(y-1) \\ 0 \end{pmatrix},$$

with initial and boundary conditions at Γ_{in}, Γ_D *and* S *given by the exact solution. The pressure is chosen such that* $\vartheta\partial_{n_{\Gamma_{out}}} v - pn_{\Gamma_{out}} = 0$ *holds.*
The coarsest space mesh level 0 *is shown in Figure 5.7(b). A level index more means a global refinement of the initial level* 0. *If we discretize in space with the* $Q_2P_1^{disc}$ *element, our exact solution lies in the ansatz space at all time points. Therefore, the space discretization error should be negligible. Furthermore, it holds*

$$J_d(u) = \int_0^1 c_d(t)dt = \frac{988}{225\pi}, \quad J_l(u) = \int_0^1 c_l(t)dt = 0.$$



(a) Domain and boundary. **(b)** Mesh level 0.

Figure 5.7: Example 3: Domain with boundary and mesh level 0.

The next example is the benchmark problem from [57].

Example 6 (Flow around a cylinder)**.** *We consider a channel section* $[0,0.41] \times [0,2.2]$ *with a circular cross section, see Figure 5.6. The midpoint of the circle with radius* $\frac{L}{2} = 0.05$ *is located at* $(0.2,0.2)$. *The inflow condition at* Γ_{in} *is the time-dependent parabolic inflow profile*

$$v(0,y,t) = \begin{pmatrix} \frac{4*1.5}{0.41^2}y(0.41-y)\sin(\frac{\pi}{8}t) \\ 0 \end{pmatrix}$$

with $U_{max} = 1.5$ *and* $\vartheta = 10^{-3}$ *which leads to a Reynolds number* $Re = \frac{U_{max}L}{\vartheta}$ *between* 0

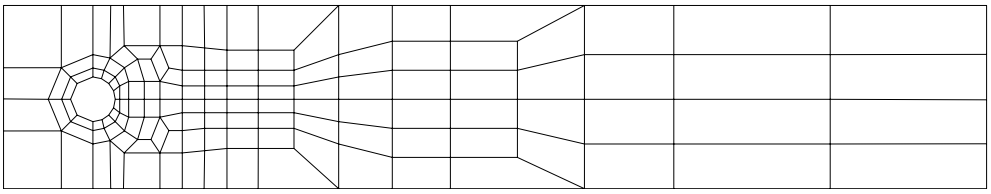


Figure 5.8: Example 6: Mesh level 0 for channel section from Figure 5.6.

and 100. *On* $\Gamma_D \cup S$, *we use Dirichlet zero boundary conditions. We solve the NSE on* $\Omega = [0,0.41] \times [0,2.2] \setminus S$ *up to the time* $T = 8$ *with initial condition* $v = (0,0)^T$. *The coarsest mesh level is shown in Figure 5.8.*

5.2.1. Drag and lift coefficients

Different methods are used in the literature for the numerical evaluation of drag and lift coefficients, which are divided in the volume and line approach, compare the appendix. In the line method we evaluate for $m = 1,\dots,M$

$$\begin{aligned} c_d^{line}(t_m) &= \frac{2}{LU_{max}^2} \int_S \vartheta (v_{1y}^m - v_{2x}^m) n_y - p^m n_x dS, \\ c_l^{line}(t_m) &= \frac{-2}{LU_{max}^2} \int_S \vartheta (v_{1y}^m - v_{2x}^m) n_x + p^m n_y dS. \end{aligned} \tag{5.2.3}$$

Hereby, we use the notation $v^m = v(t_m)$, $p^m = p(t_{m-1/2})$ for $u_k \in X_k^1$ and $v^m = v^{m,-}$, $p^m = p^{m,-}$ for $u_k \in X_k^{DG(0)}$. We evaluate drag and lift coefficients at t_m for $m = 1,\dots,M$ with the volume method by

$$\begin{aligned} c_d^{vol}(t_m) &= \frac{2}{LU_{max}^2} v_d^T \left(g - (k_m^{-1} M + \vartheta L + K(v^m)) v^m + k_m^{-1} M v^{m-1} - B p^m \right) \\ c_l^{vol}(t_m) &= \frac{-2}{LU_{max}^2} v_l^T \left(g - (k_m^{-1} M + \vartheta L + K(v^m)) v^m + k_m^{-1} M v^{m-1} + B p^m \right) \end{aligned} \tag{5.2.4}$$

with $v_d = (v_{S_h}, 0)^T$ and $v_l = (0, v_{S_h})^T$, where v_{S_h} is only 1 at the degrees of freedom on the approximated circular boundary S_h and otherwise zero. The evaluation of g depends on the used quadrature formula for solving the primal PDEs. We evaluate g at t_m when using the θ method with $\theta = 1$ or the DG(0)-mod discretization. We evaluate g at $t_{m-1/2}$ for the DG(0) method.

First, we briefly discuss the expected approximation quality regarding to the space discretization. Then, we use Example 5 to analyze the order of convergence of the mean drag coefficient in time numerically. Finally, we take a closer look at the volume method and examine whether certain terms have a significant impact on the quality of the approximation.

Expected order of convergence in space

For the stationary NSE, convergence with $\mathcal{O}(h^4)$ for the drag and lift coefficients can be seen in the numerical results, see [40]. Here, the equations are discretized with the $Q_2P_1^{disc}$ finite element pair and an isoparametric mapping is used for the circle. Furthermore, drag and lift coefficients are evaluated using the volume method.
In addition, convergence of drag and lift with $\mathcal{O}(h^2)$ is proven in [20], provided that the solution is sufficiently regular, a suitable space discretization and evaluation of drag and lift is used. They consider finite elements with linear ansatz space for the velocity, e.g. Q_1Q_0, and conclude, that they expect $\mathcal{O}(h^{2k})$ for k the polynomial degree of the velocity ansatz space.
We therefore expect convergence of the drag and lift coefficients with order 4 in space. This suggests that for a sufficiently fine space mesh level, the space error is negligible due to the low convergence order of our considered time discretizations.

Order of convergence for Example 5

First, we consider the evaluation of the mean drag coefficient for the manufactured solution from Example 5. In Table 5.6, we present errors of the mean drag value ($e[J_d]$) and the experimental order of convergence (EOC). We use M equidistant time-step sizes and global refinement in time. We compare the difference between the line and volume method for evaluating the source function with the right-sided box rule (implicit Euler and DG(0)-mod) and the midpoint rule. The convergence behavior of the mean drag is almost mesh level independent, because the exact solution can be represented by the ansatz space pointwise in time. Therefore, we only show results for level 4.
The results of Table 5.6 show that the numerically calculated values converge toward the exact value. Furthermore, there are no significant differences in the errors calculated using the line or volume method. Since $S = S_h$, we do not expect any differences between the drag coefficient evaluation methods.
The EOC depends on the used quadrature formula for the right hand side g. This makes sense, since $J_d(u_k)$ is evaluated exact for $u_k \in X_k^{DG(0)}$.

M	$e[J_d^{line}]$	EOC	$e[J_d^{vol}]$	EOC
10	9.0950e-02		8.8365e-02	
20	4.7108e-02	0.95	4.5841e-02	0.95
40	2.3982e-02	0.97	2.3354e-02	0.97
80	1.2101e-02	0.99	1.1789e-02	0.99
160	6.0779e-03	0.99	5.9224e-03	0.99
320	3.0460e-03	1.00	2.9684e-03	1.00

(a) Evaluation g at t_m.

M	$e[J_d^{line}]$	EOC	$e[J_d^{vol}]$	EOC
10	1.9510e-03		1.9856e-03	
20	4.8470e-04	2.01	4.9303e-04	2.01
40	1.2056e-04	2.01	1.2369e-04	1.99
80	2.9652e-05	2.02	3.0205e-05	2.03
160	7.7045e-06	1.94	7.7845e-06	1.96
320	1.9053e-06	2.02	1.9189e-06	2.02

(b) Evaluation g with midpoint rule.

Table 5.6: Example 5: Errors of mean drag coefficient and EOC on level 4.

Influence of system matrices in the volume method

We consider the evaluation of drag and lift with the volume method, where we calculate the residual using a specific test function. Instead of evaluating the full residual from equation (5.2.4), we analyze whether it can be approximated. We limit ourselves to the drag coefficient. For the lift, we use an analogous notation. Based on (5.2.4), we define the following simplified variants

$$
\begin{aligned}
c_d^{vol,stat}(t^m) &= \frac{2}{LU_{max}^2} v_d^T \left(g - (\vartheta L + K(v^m))v^m - Bp^m\right) \\
c_d^{vol,lin}(t^m) &= \frac{2}{LU_{max}^2} v_d^T \left(g - (k_m^{-1}M + \vartheta L)v^m + k_m^{-1}Mv^{m-1} - Bp^m\right) \\
c_d^{vol,stat,lin}(t^m) &= \frac{2}{LU_{max}^2} v_d^T \left(-\vartheta L v^m - Bp^m\right).
\end{aligned}
$$

The first variant $c_d^{vol,stat}$ corresponds to considering the stationary residual. The second approximation $c_d^{vol,lin}$, is a linearization that ignores the nonlinear term, resulting in the Stokes equation. In the third variant $c_d^{vol,stat,lin}$, we only use the Laplace matrix and B. We compare the different evaluation options.

First, we consider Example 5. In this example, $J_d^{vol,lin}(u) = J_d^{vol}(u)$ holds for the exact solution because $v \cdot \nabla v = 0$. Therefore, we expect the numerical values to be similar as well. Table 5.7 shows the error values of the mean drag computed on level 4. In addition to the values of the *vol* and *line* methods, we observe that the values of the linearized variant

M	$e[J_d^{line}]$	$e[J_d^{vol}]$	$e[J_d^{vol,lin}]$	$e[J_d^{vol,stat}]$	$e[J_d^{vol,lin,stat}]$
10	9.0950e-02	8.8365e-02	9.0648e-02	2.1818e-01	1.6438e-01
20	4.7108e-02	4.5841e-02	4.6956e-02	1.7570e-01	1.1321e-01
40	2.3982e-02	2.3354e-02	2.3906e-02	1.5323e-01	8.6444e-02
80	1.2101e-02	1.1789e-02	1.2063e-02	1.4168e-01	7.2753e-02
160	6.0779e-03	5.9224e-03	6.0591e-03	1.3582e-01	6.5827e-02
320	3.0460e-03	2.9684e-03	3.0366e-03	1.3286e-01	6.2343e-02

(a) Evaluation g at t_m.

M	$e[J_d^{line}]$	$e[J_d^{vol}]$	$e[J_d^{vol,lin}]$	$e[J_d^{vol,stat}]$	$e[J_d^{vol,lin,stat}]$
10	1.9510e-03	1.9856e-03	1.9535e-03	1.1684e-01	6.0726e-02
20	4.8470e-04	4.9303e-04	4.8568e-04	1.2293e-01	5.9314e-02
40	1.2056e-04	1.2369e-04	1.2052e-04	1.2630e-01	5.8962e-02
80	2.9652e-05	3.0205e-05	2.9542e-05	1.2807e-01	5.8875e-02
160	7.7045e-06	7.7845e-06	7.7595e-06	1.2898e-01	5.8853e-02
320	1.9053e-06	1.9189e-06	1.9095e-06	1.2944e-01	5.8847e-02

(b) Evaluation g with midpoint rule.

Table 5.7: Example 5 : Errors mean drag on level 4.

(*vol-lin*) also converge to the exact value. However, no convergence can be observed when using the *vol-stat* or *vol-lin-stat* methods. The error stagnates until $M = 40$. Therefore, the stationary residual evaluation methods significantly distort the results and can not be used for evaluation.

In Figure 5.9 we plot the drag value over time. We compare the exact value with the evaluation options *line*, *vol*, *vol-lin*, *vol-stat*, and *vol-lin-stat*. The volume method and the linearized volume method behave similarly to the line method. In particular, their plots correspond to the exact data. The stationary volume method and its linearized one do not behave accurately. The figure shows that we make errors at the beginning or at the end of the calculation when using the stationary methods. This error is not reduced with refinement.

In Tables 5.8, 5.9 and Figures 5.10, 5.11, we summarize the results for Example 6. Tables 5.8 and 5.9 show minimal variation in the calculated mean drag and lift values among the various methods. $J_d^{vol,lin}$ and $J_d^{vol,lin,stat}$ behave very similarly, as well as J_d^{vol} and $J_d^{vol,stat}$. As Figures 5.10 and 5.11 show, small differences become apparent when we zoom in. In this case, we see that *vol* and *vol-stat* method behave almost identically. The linearized evaluation options also appear to produce similar results. This coincides with the computed mean values from the Tables 5.8 and 5.9.

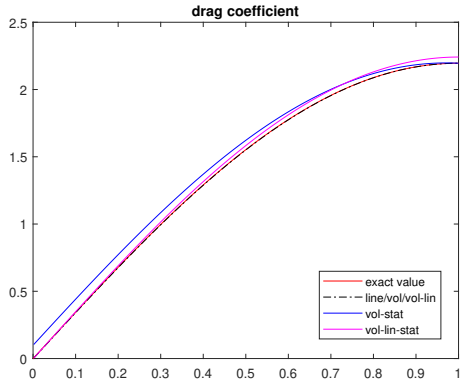


Figure 5.9: Example 5: Drag coefficient.

M	J_d^{line}	J_d^{vol}	$J_d^{vol,lin}$	$J_d^{vol,stat}$	$J_d^{vol,lin,stat}$
800	1.575987	1.575853	1.575782	1.575853	1.575782
1600	1.584954	1.584820	1.584749	1.584820	1.584749
3200	1.594535	1.594399	1.594328	1.594399	1.594328
6400	1.600539	1.600403	1.600332	1.600403	1.600332
12800	1.603895	1.603757	1.603686	1.603757	1.603686
25600	1.605670	1.605532	1.605461	1.605532	1.605461

Table 5.8: Example 6: Values mean drag on level 3.

M	J_l^{line}	J_l^{vol}	$J_l^{vol,lin}$	$J_l^{vol,stat}$	$J_l^{vol,lin,stat}$
800	-5.55984e-03	-5.55026e-03	-5.55008e-03	-5.55026e-03	-5.55008e-03
1600	-9.74333e-03	-9.73467e-03	-9.73449e-03	-9.73467e-03	-9.73449e-03
3200	-1.16705e-02	-1.16617e-02	-1.16616e-02	-1.16617e-02	-1.16616e-02
6400	-1.13902e-02	-1.13810e-02	-1.13809e-02	-1.13810e-02	-1.13809e-02
12800	-1.09207e-02	-1.09114e-02	-1.09112e-02	-1.09114e-02	-1.09112e-02
25600	-1.06103e-02	-1.06008e-02	-1.06006e-02	-1.06008e-02	-1.06006e-02

Table 5.9: Example 6: Values mean lift on level 3.

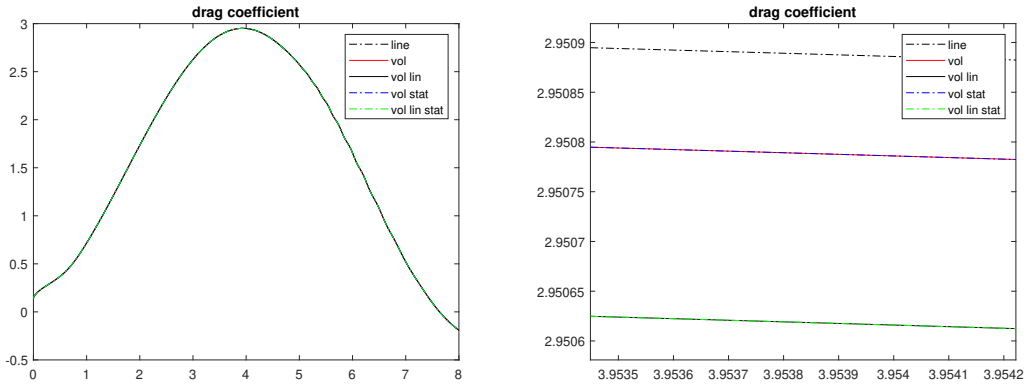


Figure 5.10: Example 6: Drag coefficient on mesh level 3 with $k_m = 6.25\text{e-}04$.

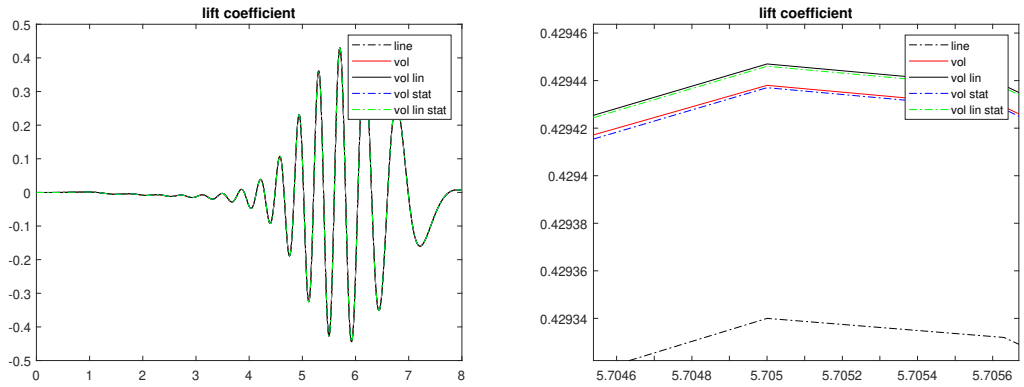


Figure 5.11: Example 6: Lift coefficient on mesh level 3 with $k_m = 6.25\text{e-}04$.

Summary

The stationary evaluation options (*vol-stat* and *vol-lin-stat*) should not be used, but linearization does not appear to distort the results. This means that, when using the quadrature formula for J, one must be careful with the time derivative.

If we use the variant *vol* instead of *vol-lin*, the third derivatives of J in the error representation are not zero. However, based on the numerical results for the considered examples, we conclude that we do not introduce an additional error. We evaluate the drag and lift coefficients by considering the full residual.

5.2.2. Flow through a square (effectivity)

We compare the error estimators summarized in Table 4.2 for the manufactured solution from Example 5, where we consider flow through a square. We provide the values of all error estimator components, as well as the effectivity index I_{eff}. We remember that the exact solution lies in the ansatz space for every time point and therefore the space discretization error should be negligible. The aim is now to compare the different error estimators from Table 4.2 for $J_d(u)$ under equidistant refinement in time.

Table 5.10 and Table 5.12 show the results for the DG error estimators. In Table 5.13 and Table 5.14 the results for the θ method are given.

Comparison $\eta_{DG,B}^{lin1}$ with η_{mod}^{lin1}

We consider the error estimators for the implicit Euler time-stepping scheme derived with the DG error representations in Table 5.10. We first discuss the values from Table 5.10a. For $\eta_{DG,B}^{lin1}$, the behavior of the exact and estimated error does not match. Adding the estimator of the quadrature error η_g, leads to better results concerning effectivity. We do not reach a value of 1, but the estimated error is approximately three times smaller than

M	η_ρ^{lin1}	$\eta_{\rho^*}^{lin1}$	$\eta_{DG,B}^{lin1}$	I_{eff}	η_g	η_{mod}^{lin1}	I_{eff}
10	-1.4177e-03	4.9778e-03	3.5601e-03	0.04	1.3729e-02	1.7290e-02	0.20
20	-4.2549e-04	1.2173e-03	7.9180e-04	0.02	1.1898e-02	1.2690e-02	0.28
40	-1.1810e-04	3.0012e-04	1.8202e-04	0.01	7.2923e-03	7.4743e-03	0.32
80	-3.1227e-05	7.4451e-05	4.3225e-05	0.00	3.9952e-03	4.0384e-03	0.34
160	-8.0362e-06	1.8537e-05	1.0501e-05	0.00	2.0867e-03	2.0972e-03	0.35
320	-2.0389e-06	4.6244e-06	2.5855e-06	0.00	1.0659e-03	1.0685e-03	0.36

(a) Mesh level 4.

M	η_ρ^{lin1}	$\eta_{\rho^*}^{lin1}$	$\eta_{DG,B}^{lin1}$	I_{eff}	η_g	η_{mod}^{lin1}	I_{eff}
10	-1.5214e-03	5.1425e-03	3.6211e-03	0.04	2.6870e-02	3.0491e-02	0.34
20	-4.4321e-04	1.2582e-03	8.1499e-04	0.02	1.8644e-02	1.9459e-02	0.42
40	-1.2160e-04	3.1034e-04	1.8874e-04	0.01	1.0699e-02	1.0888e-02	0.46
80	-3.1993e-05	7.7005e-05	4.5012e-05	0.00	5.7059e-03	5.7509e-03	0.49
160	-8.2149e-06	1.9175e-05	1.0960e-06	0.00	2.9438e-03	2.9547e-03	0.50
320	-2.0820e-06	4.7841e-06	2.7020e-06	0.00	1.4948e-03	1.4975e-03	0.50

(b) Mesh level 6.

Table 5.10: Example 5: Error estimators $\eta_{DG,B}^{lin1}$ and η_{mod}^{lin1} and effectivity index.

M	$\rho(i^k u_k)(z_k)$	$\tilde{\eta}_{mod}^{lin1}$	I_{eff}
10	2.8735e-02	3.2295e-02	0.37
20	1.5646e-02	1.6438e-02	0.36
40	8.2276e-03	8.4096e-03	0.36
80	4.2287e-03	4.2719e-03	0.36
160	2.1451e-03	2.1556e-03	0.36
320	1.0805e-03	1.0831e-03	0.36

(a) Mesh level 4.

M	$\rho(i^k u_k)(z_k)$	$\tilde{\eta}_{mod}^{lin1}$	I_{eff}
10	4.2824e-02	4.6445e-02	0.52
20	2.2639e-02	2.3454e-02	0.51
40	1.1697e-02	1.1886e-02	0.51
80	5.9553e-03	6.0003e-03	0.51
160	3.0061e-03	3.0171e-03	0.51
320	1.5104e-03	1.5131e-03	0.51

(b) Mesh level 6.

Table 5.11: Example 5: Error estimator $\tilde{\eta}_{mod}^{lin1} = \eta_{DG,B}^{lin1} + \rho(i_k^{lin1} u_k)(z_k)$ and effectivity index.

the exact error.

There are two possible reasons why we do not reach an effectivity of 1. One reason could be that the space error is not negligible. To analyze this, we present the error estimator values on level 6 in table part (b). We observe that the average I_{eff} improves from 0.31 to 0.45. This suggests that the discrepancy in the effectivity index is introduced by the space discretization error.

Another reason could be attributed to the quadrature error due to the nonlinearity. In the case of the DG(0) method, the nonlinearity is exactly approximated by the quadrature rule. We compute adapted error estimators $\tilde{\eta}_{mod}^{lin1} = \eta_{DG,B}^{lin1} + \rho(i^k u_k)(z_k)$, where we add the additional term $\rho(i^k u_k)(z_k)$ instead of η_g to the error estimator. This should increase the error due to the nonlinearity. The results are given in Table 5.11. The values of

$\tilde{\eta}_{mod}^{lin1}$ and η_{mod}^{lin1} are the same for finer time-step sizes. The difference between the two error estimators is that I_{eff} remains constant for $\tilde{\eta}_{mod}^{lin1}$ while η_{mod}^{lin1} has smaller values for coarser time-step sizes. Using this additional term improves the effectivity index only on coarser time meshes.

DG with other quadrature $\eta_{DG,M}^{lin1}$

Table 5.12 shows the results for the error estimator $\eta_{DG,M}^{lin1}$. The effectivity index is slightly above 1, but improves for finer time-step sizes. We can see that exact and estimated errors converge with order 2, which reflects the approximation quality of the quadrature formula. Again, in contrast to the values for $\eta_{DG,B}^{lin1}$ from Table 5.10, the DG method with midpoint rule leads to good results without considering an additional term. The error estimators $\eta_{DG,B}^{lin1}$ and $\eta_{DG,M}^{lin1}$ have nearly identical values, but the exact error differs. In this case, it is again not necessary to consider an additional term in order to generate relatively reliable values.

M	η_{ρ}^{lin1}	$\eta_{\rho^*}^{lin1}$	$\eta_{DG,M}^{lin1}$	I_{eff}
10	-1.6429e-03	4.8294e-03	3.1865e-03	1.60
20	-4.5856e-04	1.1979e-03	7.3938e-04	1.50
40	-1.2267e-04	2.9767e-04	1.7410e-04	1.41
80	-3.1830e-05	7.4144e-05	4.2314e-05	1.40
160	-1.8498e-05	8.1138e-06	-1.0384e-05	1.33
320	-2.0487e-06	4.6196e-06	2.5709e-06	1.34

Table 5.12: Example 5: Error estimators $\eta_{DG,M}^{lin1}$ and effectivity index on mesh level 4.

Comparison of interpolation methods for pressure for η_{θ} and $\eta_{\theta,mod}$

We consider the error estimators η_{θ} and η_{mod} for two different interpolation variants $lin1$ and *quad* for the pressure for the primal equation, denoted by η_{θ}^{lin1}, η_{θ}^{quad}, $\eta_{\theta,mod}^{lin1}$ and $\eta_{\theta,mod}^{quad}$, compare Table 4.2. Remember, that we associate the degrees of freedom of p with the discrete time points and the degrees of freedom for q with the interval midpoints. Table 5.13 shows the results for η_{θ} and Table 5.14 for $\eta_{\theta,mod}$. We see, that using the linear interpolation for the pressure leads to an effectivity index of approximately 1. For the error estimator $\eta_{\theta,mod}^{lin1}$, the effectivity index on mesh level 4 is slightly farther away from 1 compared to estimator η_{θ}^{lin1}. On mesh level 6 both error estimators lead to similar good approximations to the exact value with the difference that we either under or over estimate the error. For the *quad* variant, we do not reach an effectivity index of 1. All results are better when the linear interpolation variant is used.

M	η_ρ^{lin2}	$\eta_{\rho^*}^{lin1}$	η_{gal}	η_θ^{lin1}	I_{eff}
10	7.7610e-03	5.7310e-02	3.1790e-02	9.6861e-02	1.07
20	2.0146e-03	2.8758e-02	1.8027e-02	4.8799e-02	1.06
40	5.1400e-04	1.4421e-02	9.5601e-03	2.4495e-02	1.05
80	1.2987e-04	7.2229e-03	4.9182e-03	1.2271e-02	1.04
160	3.2643e-05	3.6149e-03	2.4938e-03	6.1413e-03	1.04
320	8.1831e-06	1.8083e-03	1.2556e-03	3.0721e-03	1.03
M	η_ρ^{lin2}	$\eta_{\rho^*}^{quad}$	η_{gal}	η_θ^{quad}	I_{eff}
10	7.9026e-04	1.1275e-03	3.1790e-02	3.3708e-02	0.38
20	1.3808e-04	3.1264e-04	1.8027e-02	1.8477e-02	0.40
40	2.6345e-05	8.1797e-05	9.5601e-03	9.6682e-03	0.42
80	5.5119e-06	2.0887e-05	4.9182e-03	4.9446e-03	0.42
160	1.2402e-06	5.2791e-06	2.4938e-03	2.5003e-03	0.42
320	2.9256e-07	1.3227e-06	1.2556e-03	1.2572e-03	0.42

(a) Mesh level 4.

M	η_ρ^{lin2}	$\eta_{\rho^*}^{lin1}$	η_{gal}	η_θ^{lin1}	I_{eff}
10	6.9593e-03	5.5202e-02	4.1150e-02	1.0331e-01	1.16
20	1.8136e-03	2.7791e-02	2.2712e-02	5.2316e-02	1.14
40	4.6367e-04	1.3959e-02	1.1905e-02	2.6327e-02	1.12
80	1.1728e-04	6.9976e-03	6.0910e-03	1.3206e-02	1.12
160	2.9494e-05	3.5038e-03	3.0803e-03	6.6134e-03	1.11
320	7.3958e-06	1.7530e-03	1.5489e-03	3.3093e-03	1.11
M	η_ρ^{lin2}	$\eta_{\rho^*}^{quad}$	η_{gal}	η_θ^{quad}	I_{eff}
10	1.2908e-03	1.1841e-03	4.1150e-02	4.3625e-02	0.49
20	2.6846e-04	3.2666e-04	2.2712e-02	2.3307e-02	0.51
40	5.9602e-05	8.5274e-05	1.1905e-02	1.2049e-02	0.51
80	1.3909e-05	2.1751e-05	6.0910e-03	6.1266e-03	0.52
160	3.3495e-06	5.4911e-06	3.0803e-03	3.0892e-03	0.52
320	8.2116e-07	1.3786e-06	1.5489e-03	1.5511e-03	0.52

(b) Mesh level 6.

Table 5.13: Example 5: Error estimators η_θ^{lin1}, η_θ^{quad} and effectivity index.

M	η_{ρ}^{lin2}	$\eta_{\rho^*}^{lin1}$	η_{gal}	$\eta_{\theta,mod}^{lin1}$	I_{eff}
10	7.7610e-03	3.8307e-02	3.1790e-02	7.7858e-02	0.88
20	2.0146e-03	1.9191e-02	1.8027e-02	3.9232e-02	0.86
40	5.1400e-04	9.6179e-03	9.5601e-03	1.9692e-02	0.84
80	1.2987e-04	4.8162e-03	4.9182e-03	9.8643e-03	0.84
160	3.2643e-05	2.4101e-03	2.4938e-03	4.9366e-03	0.83
320	8.1831e-06	1.2056e-03	1.2556e-03	2.4694e-03	0.83
M	η_{ρ}^{lin2}	$\eta_{\rho^*}^{quad}$	η_{gal}	$\eta_{\theta,mod}^{quad}$	I_{eff}
10	7.9026e-04	1.1275e-03	3.1790e-02	3.3708e-02	0.38
20	1.3808e-04	3.1264e-04	1.8026e-02	1.8477e-02	0.40
40	2.6345e-05	8.1797e-05	9.5601e-03	9.6682e-03	0.41
80	5.5119e-06	2.0887e-05	4.9182e-03	4.9446e-03	0.42
160	1.2401e-06	5.2791e-06	2.4938e-03	2.5003e-03	0.42
320	2.9256e-07	1.3227e-06	1.2556e-03	1.2572e-03	0.42

(a) Mesh level 4.

M	η_{ρ}^{lin2}	$\eta_{\rho^*}^{lin1}$	η_{gal}	$\eta_{\theta,mod}^{lin1}$	I_{eff}
10	6.9593e-03	3.6869e-02	4.1150e-02	8.4978e-02	0.96
20	1.8136e-03	1.8538e-02	2.2711e-02	4.3063e-02	0.93
40	4.6367e-04	9.3080e-03	1.1904e-02	2.1676e-02	0.92
80	1.1728e-04	4.6654e-03	6.0909e-03	1.0874e-02	0.92
160	2.9494e-05	2.3358e-03	3.0803e-03	5.4456e-03	0.91
320	7.3958e-06	1.1687e-03	1.5489e-03	2.7250e-03	0.91
M	η_{ρ}^{lin2}	$\eta_{\rho^*}^{quad}$	η_{gal}	$\eta_{\theta,mod}^{quad}$	I_{eff}
10	1.2908e-03	1.1841e-03	4.1150e-02	4.3625e-02	0.49
20	2.6846e-04	3.2666e-04	2.2712e-02	2.3307e-02	0.51
40	5.9602e-05	8.5274e-05	1.1905e-02	1.2049e-02	0.51
80	1.3909e-05	2.1751e-05	6.0910e-03	6.1266e-03	0.52
160	3.3495e-06	5.4911e-06	3.0803e-03	3.0892e-03	0.52
320	8.2116e-07	1.3786e-06	1.5489e-03	1.5511e-03	0.52

(b) Mesh level 6.

Table 5.14: Example 5: Error estimator $\eta_{\theta,mod}^{lin1}$, $\eta_{\theta,mod}^{quad}$ and effectivity index.

Summary

For the DG error estimators, we see the same behavior as for the heat equation. The error estimator $\eta_{DG,B}^{lin1}$ leads to inaccurate results, whereas $\eta_{DG,M}^{lin1}$ works well, but with more variability concerning the effectivity index. Using η_{mod}^{lin1} improves the results significantly. However, we do not reach an effectivity index of 1 for this example.
Here, the results for the θ error estimators are better than those for η_{mod}^{lin1} and $\eta_{DG,M}^{lin1}$. There is no major difference between the modified error estimator and the original one. For both estimators, we reach an effectivity index of 1 when interpolating the pressure linearly. The quadratic pressure interpolation should not be used.

5.2.3. Flow around a cylinder (adaptivity)

We consider adaptive time refinement for the error estimators from Table 4.2 for Example 6. We select the error estimators

- η_{DG}^{lin1} as representative of the DG error estimators. Because Example 6 has no source function, we do not have to distinguish between the estimators $\eta_{DG,B}^{lin1}, \eta_{DG,M}^{lin1}$ and η_{mod}^{lin1} in this case.
- η_{θ}^{lin1} and $\eta_{\theta,mod}^{lin1}$ because the linear interpolation of the pressure showed more reliable values than the quadratic interpolation, compare Table 5.13.

All results are computed on mesh level 3, resulting in 42016 degrees of freedom in space and with a relative tolerance of 0.01. We consider tolerance and maximum value based marking strategies with $K_{max} = 0.5$. Furthermore, we use prior marking of the largest 10% and do not mark the time steps independently from each other for the θ method error estimators, denoted by a checkmark in the column $2k_m$. We start with an initial time mesh consisting of $M = 80$ equidistant time steps and halve the marked time steps.
We examine two aspect functionals

- the mean drag coefficient over the entire time period

$$J_d(u) = \frac{1}{8}\int_0^8 c_d(t)dt$$

- and the mean lift coefficient over the time period $[5,6]$

$$J_l(u) = \int_5^6 c_l(t)dt.$$

First, we give the results for the mean drag coefficient, then those for the lift coefficient.

Mean drag coefficient

We summarize the results of adaptive refinement in Table 5.15 and Figures 5.12 and 5.13. Table 5.15 presents an overview of the error estimators and marking strategies. Figure 5.12 plots the drag coefficient behavior for the different methods, and Figure 5.13 shows a comparison of the refined time-step sizes. First, we discuss the table and then the figures.

error est.	marking strategy	$2k_m$	mark 10%	M	ratio steps	iter.	J_d
η_{DG}^{lin1}	global	-	-	5120	-	6	1.598817
	tol	×	-	4612	0.90	6	1.596622
	max	×	✓	10143	1.98	14	1.602087
η_{θ}^{lin1}	global	-	-	10240	-	7	1.602895
	tol	✓	-	8066	0.79	7	1.601424
	max	✓	✓	8432	0.82	10	1.601097
$\eta_{\theta,mod}^{lin1}$	global	-	-	10240	-	7	1.602895
	tol	✓	-	8690	0.85	7	1.601784
	max	✓	✓	13374	1.31	11	1.602757

Table 5.15: Example 6: Comparison of marking strategies for mean drag.

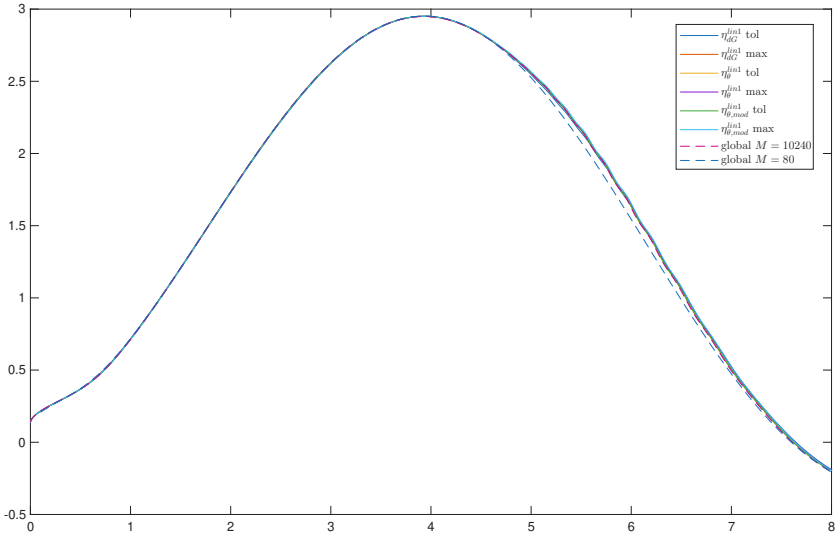
We note that, in all cases, the adaptive strategy does not provide a decisive advantage over global equidistant refinement. In the optimal case concerning the number of time steps, we reach the tolerance with 79% of the time steps compared to global refinement, see the column ratio steps. This is the case for the estimator η_{θ}^{lin1} with the tolerance strategy. There is no benefit to consider the error estimator $\eta_{\theta,mod}^{lin1}$ instead of η_{θ}^{lin1}. The results for η_{DG}^{lin1} in comparison to η_{θ}^{lin1} and $\eta_{\theta,mod}^{lin1}$ differ mainly because of the different number of time steps M required for global refinement to reach the tolerance.

The mean values calculated using adaptive strategies are similar to those calculated using global refinement for all error estimators. Compare the values of J_d.

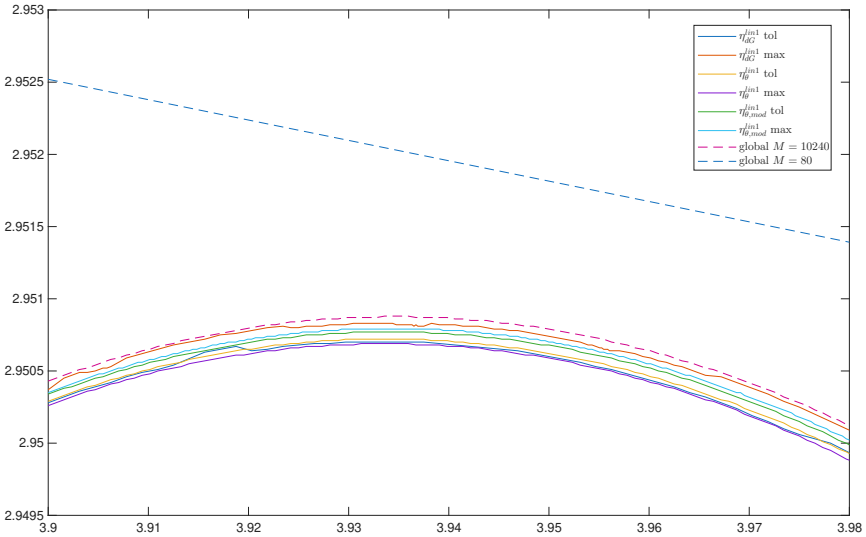
Furthermore, we observe that the tolerance strategy requires the same number of iterations as global refinement. In this example, the max marking strategy does not offer an advantage to global refinement or the tolerance strategy.

In Figure 5.12, the drag coefficients are plotted against time. We compare the values from adaptive refinement with the initial time grid (global $M = 80$) and uniform refinement to $M = 10240$ time steps. From the first plot, we can conclude that the pointwise values behave similarly, except for the initial time grid. The major differences probably arise after $t \approx 4.5$.

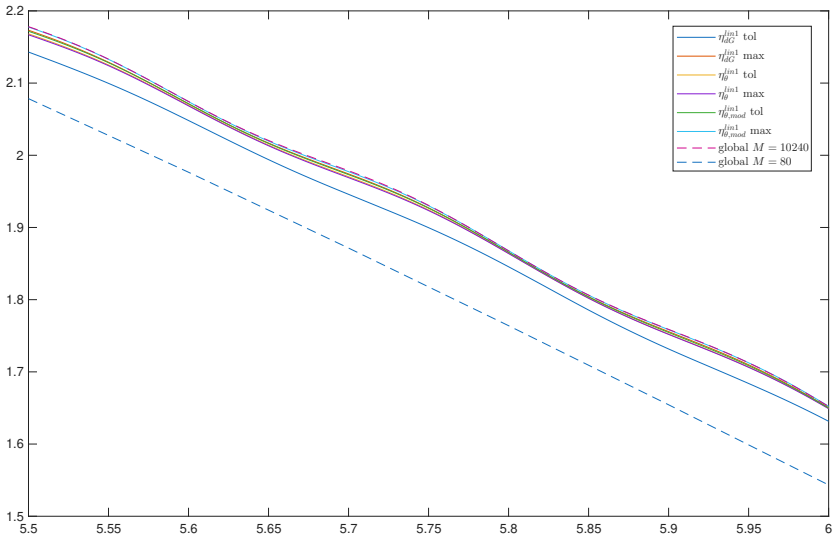
To see the differences between the estimators, we show two zoomed-in plots. First, we look at the maximum value of the drag coefficient in Figure 5.12b. We see that essentially a higher maximum value is reached for the adaptive strategies with the most time steps



(a) Over entire time period.

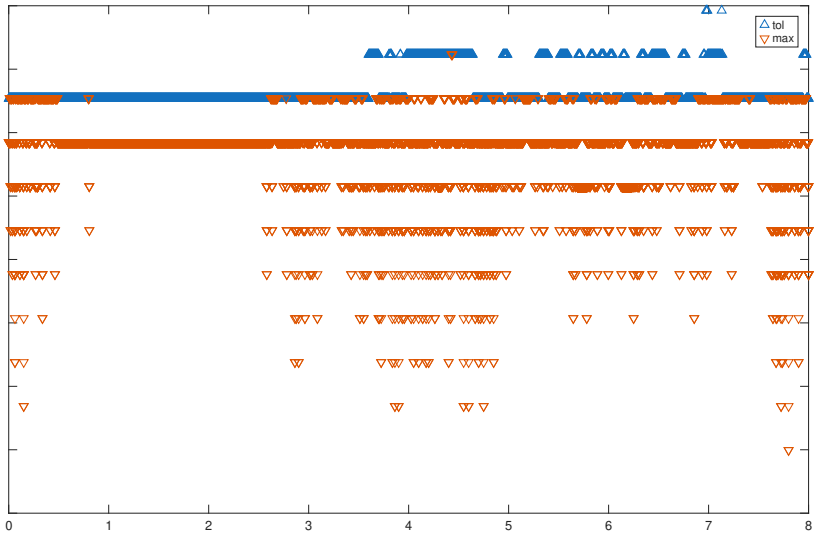


(b) Zoomed around the maximum value.

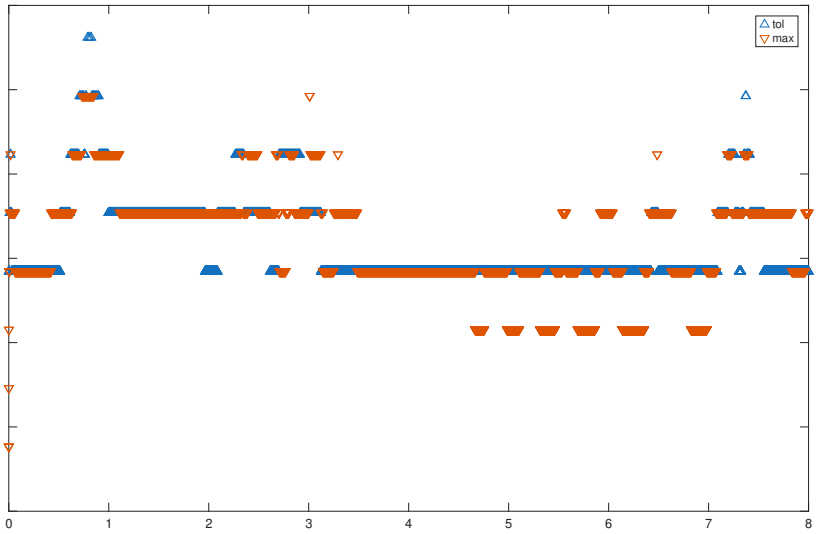


(c) Zoomed to $[5.5, 6]$.

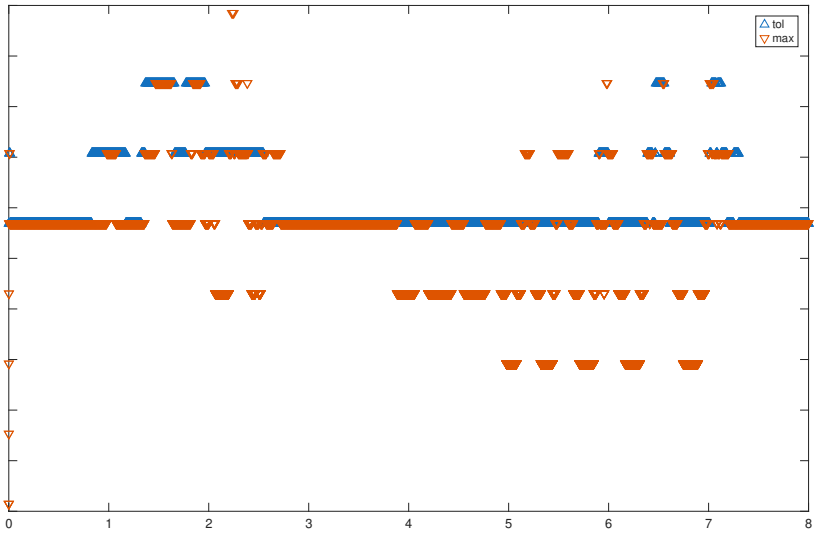
Figure 5.12: Example 6: Drag coefficient.



(a) Error estimator η_{DG}^{lin1}.



(b) Error estimator η_{θ}^{lin1}.



(c) Error estimator $\eta_{\theta,mod}^{lin1}$.

Figure 5.13: Example 6: Refined time-step sizes for mean drag.

(except for global $M = 80$).

From the last plot, we see that, in all cases, we get a similar wave-like behavior of the drag coefficient, except for the initial time grid and η_{DG}^{lin1} with the tolerance strategy. Thus, the adaptive calculated pointwise drag coefficient values reflect the behavior of the equidistant refined ones.

Figure 5.13 plots the refined time-step sizes in dependency of the corresponding error estimator. A logarithmic scale is used on the y-axis. This makes the different time-step sizes more visible. Overall, the max strategy results in a greater variety of time-step sizes, whereas the tolerance strategy produces only three to five different sizes.

Using the error estimator η_{DG}^{lin1} with the tol strategy, we observe more refinement after $t \approx 3.5$, but the time steps are almost equidistant. For the max strategy, we observe widely varying time steps in the time intervals $[0, 0.9]$ and $[2.5, 8]$, compare Figure 5.13a.

This is different for the estimators η_{θ}^{lin1} and $\eta_{\theta,mod}^{lin1}$, see Figures 5.13b and 5.13c. Using the max strategy results in more refinement for t between 2.5 and 6, compared to the estimator η_{DG}^{lin1}. The adjusted time-step sizes for both θ method error estimators exhibit similar behavior, except for the time interval $[0.5, 3]$. On this interval, larger differences become visible. Using η_{θ}^{lin1} results in less refinement around 1 and 2.8, whereas using $\eta_{\theta,mod}^{lin1}$ leads to less refinement around 1.8.

Mean lift coefficient over time period $[5, 6]$

Now, we consider the same results for the mean lift coefficient over the time interval $[5, 6]$ in Table 5.16 and Figures 5.15 and 5.14.

error est.	marking strategy	$2k_m$	mark 10%	M	ratio steps	iter.	J_l
η_{DG}^{lin1}	global	-	-	20480	-	8	-5.621743e-02
	tol	×	-	4608	0.23	7	-3.648889e-02
	max	×	✓	3246	0.16	17	-5.401760e-03
η_{θ}^{lin1}	global	-	-	81920	-	10	-6.029476e-02
	tol	✓	-	63578	0.78	11	-4.645055e-02
	max	✓	✓	2662	0.03	10	-8.541880e-03
$\eta_{\theta,mod}^{lin1}$	global	-	-	40960	-	9	-5.894187e-02
	tol	✓	-	63496	1.55	11	-4.556699e-02
	max	✓	✓	2496	0.06	11	-8.524184e-03

Table 5.16: Example 6: Comparison of marking strategies for mean lift.

To reach the relative tolerance of 0.01, we need 8 to 10 global refinements, compare Table 5.16. We observe differences between the marking strategies and the error estimators. For η_{DG}^{lin1}, the max strategy requires 2.4 times more iterations than the tol strategy, though

the number of time steps is of a more similar size.
For η_θ^{lin1} and $\eta_{\theta,mod}^{lin1}$, the tol and max strategy are very differnet concerning the total numer of time steps needed, while the number of iterations is similar. Up to approximately 15 times more time steps are needed using the tol strategy than the max strategy. Again, there are almost no differences in adaptive refinement between error estimators η_θ^{lin1} and $\eta_{\theta,mod}^{lin1}$. In this example, however, the number of global refinements differs.
The similarity between the values of J_l for the tolerance strategy and those for global refinement is noticeable. For the max strategy, the values do not improve significantly, since $J_l = -8.324769\text{e-}03$ for $M = 80$ and, for example, $J_l = -5.894187\text{e-}02$ for $M = 40960$.

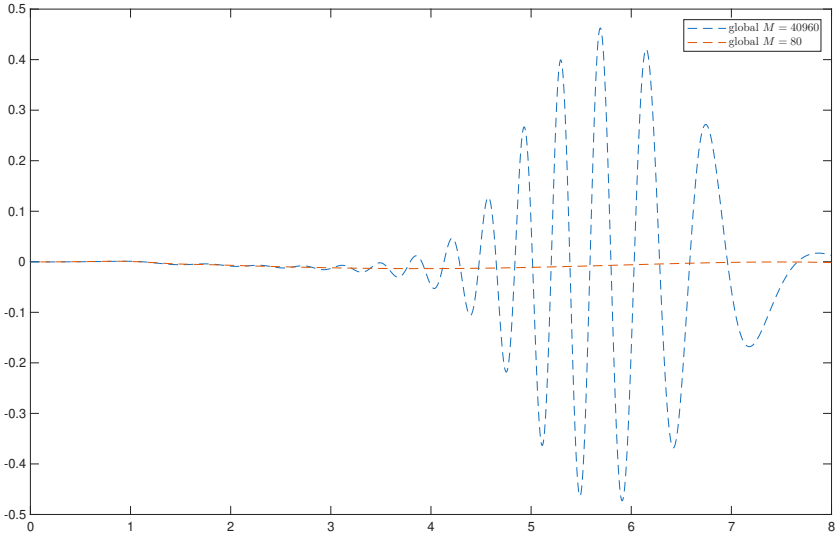
Figure 5.14 plots the lift values against time. First, we plot the lift coefficient for global refinement with $M = 80$ and $M = 40960$ time steps. Based on the plot, we can see how the value of interest behaves in dependence of the time-step size. Error estimators and marking strategies must recognize the average behavior of c_l over the interval $[5,6]$, which is not identifiable for the initial time grid.
The second plot, Figure 5.14b, shows the calculated lift coefficients after adaptive refinement. After $t = 6$, we see that the values no longer match those of global refinement in all cases. For $t < 6$, the lift coefficient oscillations are mapped in most cases. There is a slight shift and a smaller amplitude.
This behavior becomes more apparent when we examine the zoomed values over the interval $[5,6]$ in the third plot. The results computed with the tol marking strategy match the oscillatory behavior best. The shift and smaller amplitude can be observed, but this may not influence the mean value.
When using the max marking strategy, the computed lift coefficients differ completely from the correct values. Therefore, it can be assumed that the time steps selected for the max strategy are too large. This means that the behavior of the lift coefficient in the interval of interest cannot be mapped because the previous data already contains errors that are too large.

Figure 5.15 plots the refined time-step sizes for comparing the local time refinement. Overall, we see that the time steps for $t > 6$ are not refined. This corresponds with the fact, that the dual solution is zero and that the information from $t > 6$ onwards is irrelevant for controlling the error of J_l within the time period $[5,6]$. Additionally, we observe that most refinement is needed within the interval of interest.
Before $t = 5$, refinement mainly depends on the used marking strategy. With the tol strategy, the time steps are reduced more quickly at the beginning than with the max strategy. This too locally refinement of the max marking strategy will be the reason why the behavior of the lift coefficient cannot be mapped over the interval of interest, compare Figure 5.14. This, in turn, explains why the values of the goal functional J_l hardly improve

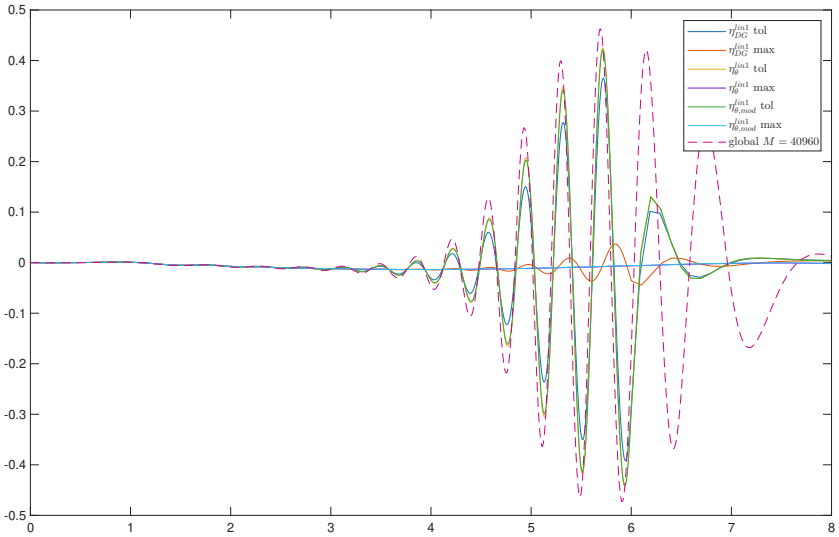
even after adaptive refinement.

Summary

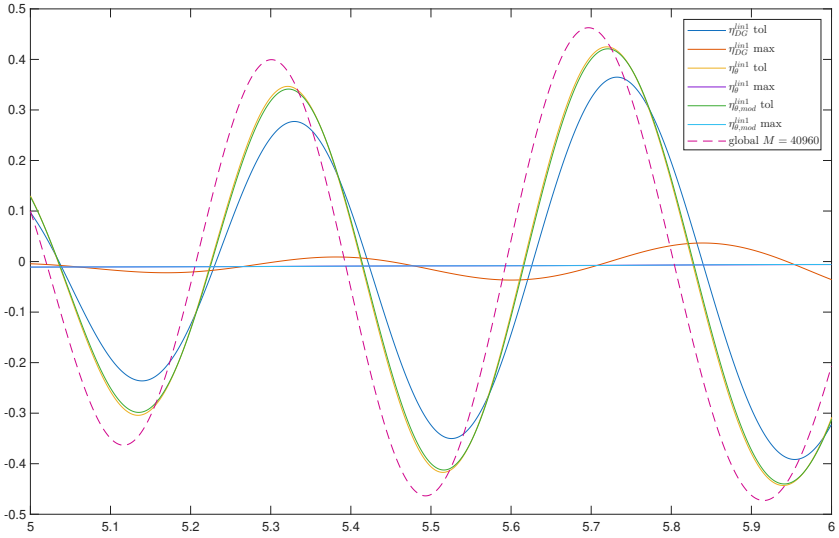
The estimators η_{θ}^{lin1} and $\eta_{\theta,mod}^{lin1}$ show similar results for adaptive refined time-step sizes. It seems that η_{θ}^{lin1} leads to better results for J_d and η_{DG}^{lin1} when considering J_l.
The tolerance strategy produces sensible results in both cases. From the results for J_l, we saw that the maximum strategy can result in time steps that are too locally refined, which can cause problems for goal-oriented error control.



(a) Global refinement over entire time period.

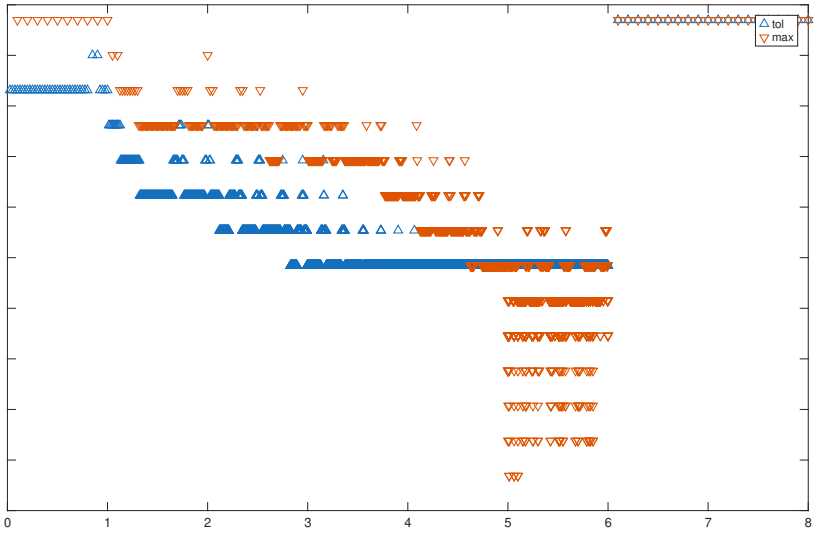


(b) All values over entire time period.

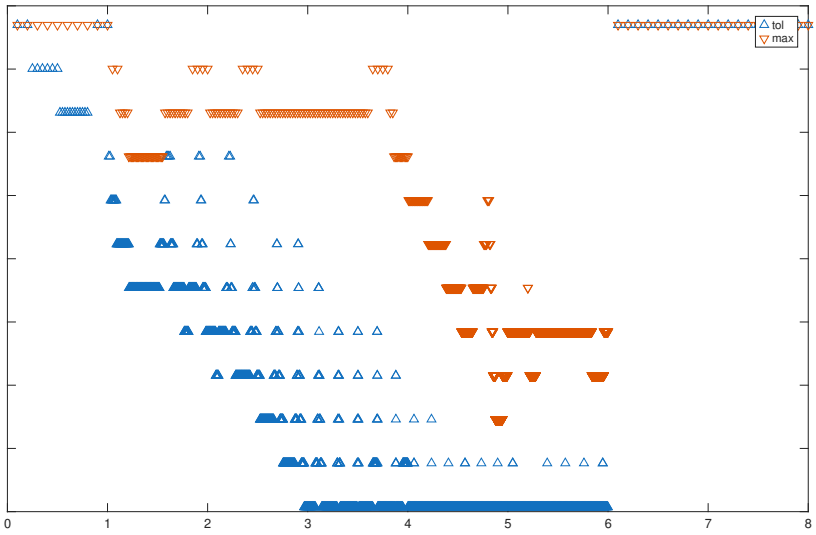


(c) Zoomed to $[5,6]$.

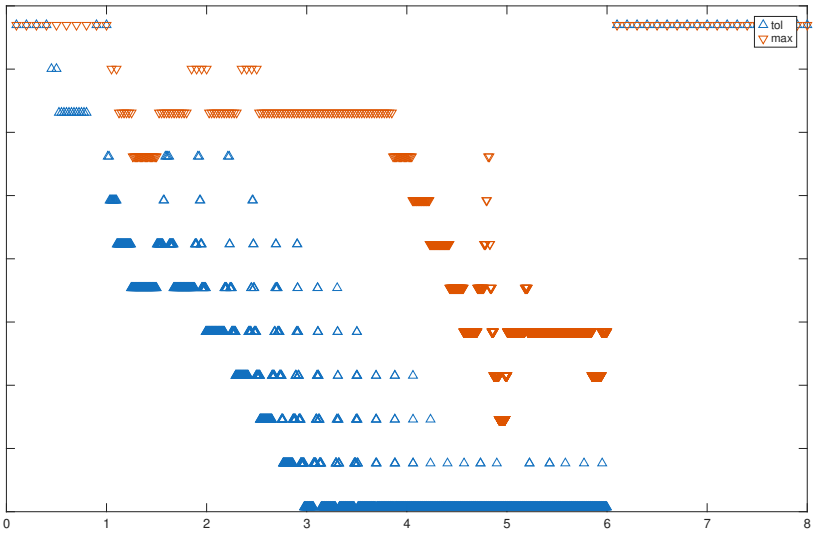
Figure 5.14: Example 6: Lift coefficient.



(a) Error estimator η_{DG}^{lin1}.



(b) Error estimator η_{θ}^{lin1}.



(c) Error estimator $\eta_{\theta,mod}^{lin1}$.

Figure 5.15: Example 6: Refined time-step sizes for mean lift over time period $[5,6]$.

6

Conclusions and outlook

Summary

We investigated DWR error estimators for adaptivity in time. We considered both Galerkin and θ time-stepping methods for time discretization of the primal equations.

In the *theoretical considerations*, we proved the error representations for the θ method from Meidner and Richter [48, 49] in the same manner as Besier and Rannacher [15] did for their error representations. We presented a uniform approach for Galerkin methods that incorporates quadrature errors. By interpreting these methods as fully time-discrete schemes, we can derive error estimators similar to those of the θ method. This results in an additional term in the error representations that is analogous to the Galerkin defect in the θ method. For the cGP(1) discretization, the additional term η_g coincides with η_{gal} from the θ method, see the appendix for details. As a result, all error representations are quite similar and comparable.

In the case of NSE, we gave an alternative error representation to the θ method, in which the remainder terms $\mathcal{R}_{p,div}$ and $\mathcal{R}_{d,div}$ do not longer appear. This is possible by adjusting the test space for the pressure. Due to this modification, the remainder terms of the θ-mod error representation are the same as those for the modified DG methods. In particular, for the case $\theta = 0.5$, which is the Crank-Nicholson time-stepping scheme, these also coincide with those for the cGP(1) method, compare the appendix.

Numerically, we compared the error estimators resulting from the error representations, approximating the weights using interpolation errors. We focused on first-order time-stepping methods. In particular, we found that using the DG error estimator from literature for the implicit Euler method can lead to errors, as its effectivity can be severely impaired. This problem can be fixed by taking the quadrature errors into account, which means

making an adjustment with our proposed additional term η_g. We found that, in contrast to the weighted residuals, this additional term is more space dependent.

For the *heat equation*, we observed an effectivity index of around 1, except when using the DG error estimator with the implicit Euler scheme. The most reliable values are provided by the θ method error estimators and the modified DG error estimator. The choice of how to interpolate the dual solution in the θ method does not affect error estimation. Regarding adaptivity, the DG error estimators, whether modified or unmodified, are preferable. Problems occurred, when applying the θ error estimators to adaptive refined time-step sizes. These problems may be related to the rotation in the test space.

For the *Navier-Stokes equations*, the modified and unmodified θ estimators with linear pressure interpolation and $\eta_{DG,M}^{lin1}$ appear capable of achieving an effectivity index of 1. Here, the modified DG estimator attains an effectivity of at most 0.5. However, the additional term η_g provides an error estimate within the correct order of magnitude. In the adaptive results for controlling the lift coefficient over a time interval, we found that the maximum based marking strategy could cause problems. With this strategy, the error in the goal functional could not be significantly reduced because the time steps were refined too locally.

Outlook

It makes sense to analyze different aspects of the *Crank-Nicolson scheme* numerically. Hereby, the numerical performance of the modified θ method, featuring a fully θ dependent test space, should be analyzed, even if we do not expect major differences to the original error estimator. In this context, one should take a closer look at the allocation of degrees of freedom, the evaluation of the goal functional using the trapezoidal rule, and the treatment of pressure resp. incompressibility constraint [39, 53].

Of course, *other time-stepping schemes* could be analyzed. For instance, one could examine backward differentiation formulas (BDF) methods. Hereby one can start with the BDF(2) method as another second-order method. The connection to Galerkin discretizations is not as clear here. See [11] for a discussion on goal-oriented error estimation for this scheme. However, higher-order BDF methods may be less suitable for adaptive calculations due to the coupling of time-step sizes [19, 24, 27].

Thus, the Galerkin methods with or without our modifications, as considered here, are suitable as *higher-order methods*. It may be of particular interest to investigate how the additional term behaves as a function of the method's order. In this context, it may also be interesting to connect this with works on balancing of discretization and iteration errors, see for example [25, 54]. Since a similar additional residual term is introduced in this works for measuring the iteration error of the solver. In a sense, the question arises, under

which assumptions this additional error is more connected to the time quadrature error or the iteration error.
On the other hand, it also makes sense to use *adaptivity in space*. This has already been studied previously [15, 60]. Here, the overall error is divided into space and time errors by

$$J(u) - J(u_{kh}) = J(u) - J(u_k) + J(u_k) - J(u_{kh}).$$

The DWR lemma is applied to both differences. For the first difference, the weights can be approximated by time interpolation errors, as was done in this study. For the second difference, the weights are approximated by space interpolation errors. Because space and time errors cannot be completely separated by space-time error estimators, the question arises whether the aspects examined in this work are important with regard to quadrature errors in space-time DWR error estimators. This is especially important since the additional term also depends on the space discretization. Particular care is required for the NSE when using time-stepping schemes with dynamically changing space meshes, see [16, 59].

A

Appendix

A.1. Drag and lift coefficients

Different methods are used in the literature for the numerical evaluation of drag and lift coefficients, see e.g. [28, 38]. These are usually divided into the line and volume approach. One method, called the line method, calculates values by evaluating the surface integral. The other approach is to calculate an integral over the entire area using specific test functions. This is known as the volumetric approach. From an analytical point of view, the two methods are identical. However, the volumetric approach appears to be less sensitive to the approximation of the circular boundary S. See [37, 38] for more information on this.

In the *line method*, we evaluate the integrals (5.2.1) directly. We can reformulate them into a simpler expression, following John [37]. For reasons of clarity, we drop the time dependency. Using the fact, that each component of the velocity field v is constant on S, we can write

$$\frac{\partial v_1}{\partial t_S} = v_{1x}n_y - v_{1y}n_x = 0, \quad \frac{\partial v_2}{\partial t_S} = v_{2x}n_y - v_{2y}n_x = 0 \quad \text{on } S. \tag{A.1}$$

Here, v_1 and v_2 are the first and second components of v and v_{ix} and v_{iy} denote the derivatives of the i-th component. With this, we can reformulate

$$\begin{aligned} \frac{\partial v_{t_S}(t)}{\partial n} &= (v_{1y}n_x - v_{2x}n_x)n_x + (v_{1y}n_y - v_{2x}n_y)n_y \\ &= v_{1y}(n_x^2 + n_y^2) - v_{2x}(n_x^2 + n_y^2) = v_{1y} - v_{2x}, \end{aligned} \tag{A.2}$$

where we first computed the derivative and then used (A.1). Inserting this into the definition of drag or lift yields

$$\begin{aligned} c_d^{line} &= \frac{2}{LU_{max}^2}\int_S \vartheta(v_{1y}-v_{2x})n_y - pn_x dS \\ c_l^{line} &= \frac{-2}{LU_{max}^2}\int_S \vartheta(v_{1y}-v_{2x})n_x + pn_y dS. \end{aligned} \tag{A.3}$$

Equations (A.3) are evaluated numerically when using the line method.
In the *volume method*, we do not evaluate boundary integrals over S. Instead, we transform the integrals in equation (5.2.1) into volume integrals. Starting with the NSE in strong form, we test the first equation with $v_S \in (H^1(\Omega))^2$, where $v_S = 0$ on $\partial\Omega \setminus S$. By partial integration, we obtain

$$(g, v_S) - (v_t, v_S) - \vartheta(\nabla v, \nabla v_S) - ((v\cdot\nabla)v, v_S) + (p, \nabla\cdot v_S) = \int_S \vartheta \nabla v \cdot v_S \cdot n + p v_S \cdot n dS.$$

The left side of the equation is the residual of the NSE in weak form, tested with v_S without the initial data and divergence condition. On the right-hand side, we collected the two boundary integrals resulting from partial integration.
To evaluate the drag coefficient, we choose the test function $u_d = (v_d, 0)^T$ with $v_d = (v_S, 0)^T$. So, we set the first component of the velocity to the indicator function over S. Evaluating the equation for u_d yields a residual that is identical to the drag coefficient without scaling

$$\begin{aligned} \rho(u)(u_d) &= \int_S \vartheta\nabla v\cdot v_d\cdot n + p v_d\cdot n dS = \int_S \vartheta(v_{1x}n_x + v_{1y}n_y) + pn_x dS \\ &= \int_S \vartheta(-v_{2y}n_x + v_{1y}n_y) + pn_x dS = \int_S \vartheta(-v_{2x}n_y + v_{1y}n_y) + pn_x dS \\ &= \int_S \vartheta \frac{\partial v_{t_S}}{\partial n} n_y + pn_x dS, \end{aligned}$$

using the definition of the derivative and the divergence-free velocity field v, as well as formula (A.2).
In the same way, we get the lift coefficient by testing with $u_l = (v_l, 0)^T$ with $v_l = (0, v_S)^T$

$$\rho(u)(u_l) = \int_S \vartheta\nabla v\cdot v_l\cdot n + p v_l\cdot n dS = \int_S \vartheta \frac{\partial v_{t_S}}{\partial n} n_x + pn_y dS.$$

To obtain the drag or lift coefficient, we only need to multiply the residuals by $2(LU_{max}^2)^{-1}$. So, we evaluate

$$c_d^{vol} = \frac{2}{LU_{max}^2}\rho(u)(u_d), \quad c_l^{vol} = \frac{2}{LU_{max}^2}\rho(u)(u_l)$$

when we talk about the volume method. Of course, we replace ρ by ρ_θ or $\rho_{DG(r)}$ for the time discrete solutions.
In contrast to the definition of the drag and lift coefficients, this evaluation variant is nonlinear in the velocity field. Note that the transformation is only exact for the exact solution of the differential equation. This requires that the solution lies in the corresponding function spaces and that the circle is approximated exactly, including the boundary conditions. This is not the case for numerically calculated solutions.

Right-hand side of dual PDE

To control the error of the mean drag and lift coefficients, we use the volume method. Depending on the used quadrature rule the right-hand side of the dual equation changes. With the system matrices from Section 2.3.2 the residual $\rho(u)(u_d)$ can be evaluated by, compare equation (5.2.4),

$$c_d^{vol}(t_m) = \frac{2}{LU_{max}^2} v_d^T \left(g - (k_m^{-1}M + \vartheta L + K(v^m))v^m + k_m^{-1}Mv^{m-1} - Bp^m \right) \tag{A.4}$$

for $m = 1,\dots,M$ with $v_d = (v_{S_h}, 0)^T$ and $v_l = (0, v_{S_h})^T$, where v_{S_h} is only 1 at the degrees of freedom on the approximated circular boundary S_h and otherwise zero. We use the notation $v^m = v(t_m)$, $p^m = p(t_{m-1/2})$ for $u_k \in X_k^1$ and $v^m = v^{m,-}$, $p^m = p^{m,-}$ for $u_k \in X_k^{DG(0)}$. The evaluation of g depends on the used quadrature formula for solving the primal PDEs. We evaluate g at t_m when using the θ method with $\theta = 1$ or the DG(0)-mod discretization. We evaluate g at $t_{m-1/2}$ for the DG(0) method.
We use the box rule for evaluating the time integral

$$J_d(u) = \frac{1}{T}\int_0^T c_d^{vol}(t)dt \approx \frac{1}{T}\sum_{m=1}^{M} k_m c_d^{vol}(t_m), \tag{A.5}$$

which integrates functions $u_k \in X_k^{DG(0)}$ exact but not $u_k \in X_k^1$. The derivative of (A.5) determines the right-hand side of our adjoint equation. Computing the derivative at t_m we get with formula (A.4)

$$\frac{LU_{max}^2}{2}\frac{\partial c_d^{vol}(u)}{\partial u}(\varphi)|_{t=t_m} = v_d^T \left(\left(-k_m^{-1}M - \vartheta L - K(v^m) - K'(v^m) \right)\psi^m + k_m^{-1}M\psi^{m-1} - B\chi^m \right)$$

with function $\varphi = (\psi, \chi)^T$. We get the following for the derivative of $J_d(u)$

$$\frac{LU_{max}^2}{2}\frac{\partial J_d(u)}{\partial u}(\varphi) = \frac{1}{T}\sum_{m=1}^{M} k_m \{ v_d^T \big(\big(-k_m^{-1}M - \vartheta L - K(v^m) - K'(v^m)\big)\psi^m + k_m^{-1}M\psi^{m-1} - B\chi^m \big) \}.$$

Now, if we reorder the terms according to the discrete time evaluation of the test function, we obtain for $m = M$

$$\frac{k_M}{T}\left(\psi^{M,T}\left(-k_M^{-1}M - \vartheta L - K^T(v^M) - K'^T(v^M)\right) - \chi^{M,T}B^T\right)v_d.$$

This corresponds to a time-dependent residual of a symmetric system without external force. In all inner time steps, for $m = 1, \ldots, M-1$, we get

$$\frac{k_m}{T}\left(\psi^{m,T}\left(-\vartheta L - K^T(v^m) - K'^T(v^m)\right) - \chi^{m,T}B^T\right)v_d,$$

which corresponds to a symmetric stationary residual without force function. This is exactly what goes into the right-hand side of the adjoint equation, when we use the box rule for integration of the cost functional. The term $\frac{1}{T}\psi^{0,T}Mv_d$ does not appear because v^0 is the projection of given data v_0 in the ansatz space.

A.2. Error representations for continuous Galerkin-Petrov discretizations

Continuous Galerkin-Petrov discretizations of order r (cGP(r)) are based on the weak form of the PDE and are defined for $r \geq 1$. In contrast to the DG(r) schemes, the ansatz space is continuous, but the test space is discontinuous. We will now apply the DWR method to the general cGP(r) schemes. We explicitly formulate the error representations because we could not find them directly in the literature.

As mentioned in [60], in the cGP(1) situation, quadrature errors must be considered for a more accurate error estimation. They add an additional term to the Lagrange function. Here, we include quadrature errors in DWR error representations analogous to modified DG error representations. This leads to error representations similar to those of the θ method. We select the quadrature formula such that, for the cGP(1) discretization, the Crank-Nicolson time-stepping scheme is rebuilt, compare Remark 4. This enables us to compare the error estimators for $\theta = 0.5$ from Chapter 3 with those derived using the cGP(1) discretization. Doing so, the unhandly Crank-Nicolson situation is theoretically

resolved.

We, first define the quadrature formula used for the modified error representations in Section A.2.1. Then, we specify the DWR error representations for the cGP(r) and modified cGP(r) discretizations for the heat equation in Section A.2.2 and the NSE in Section A.2.3. To shorten notation, we indirectly specify the discretizations of the PDEs via Lagrange functions and only define them.

A.2.1. Quadrature rule

We denote by ω_j the weights and by t^j the nodes of the Gauss-Lobatto quadrature formula on the interval $[-1,1]$. The weights $\{\omega_j\}_{j=1,\dots,k+1}$ and nodes $\{t^j\}_{j=1,\dots,k+1}$ are available in tabular form. This quadrature is defined for $k \geq 1$ and of order $2k-1$ [26]. For each m, we denote by ω_j^m and t_m^j the on $[t^{m-1},t^m]$ transformed weights and nodes. The quadrature formula introduces a discrete measure $\tilde{\mu}_{k+1}^m(dt)$ on each time interval with

$$\int_I g(t)dt \approx \sum_{m=1}^{M} \int_{I_m} g(t)\tilde{\mu}_{k+1}^m(dt) := \sum_{m=1}^{M} \sum_{j=1}^{k+1} \omega_j^m g(t_m^j).$$

We replace the time integrals in the cGP(r) discretization by integration with respect to the discrete measures. We choose $k = r$. Note, that we want to produce the Crank-Nicolson time-stepping scheme for $r = 1$, so we follow [58] and use Gauss-Lobatto quadrature points. For example, in [34], Gauss quadrature points are used.

A.2.2. Heat equation

The equation is set up for piecewise polynomial functions in time. We generalize the function space of continuous linear polynomials V_k^1 to continuous polynomials of arbitrary order and define, for $r \in \mathbb{N}$, the semi-discrete space V_k^r

$$V_k^r = \{v_k \in C(\bar{I}, L^2(\Omega)) : v_k|_{I_m} \in P^r(I_m, V)\}.$$

With this definition we can formulate the semi-discrete cGP(r) formulation of the heat equation by using V_k^r as ansatz space and $V_k^{DG(r-1)}$ as test space, as for example considered in [3, 34]. The Lagrange functions $\mathcal{L}^{cGP(r)}, \mathcal{L}^{cGP(r),mod} : V_k^r \times V_k^{DG(r-1)} \to \mathbb{R}$ are

given by

$$\mathcal{L}^{cGP(r)}(u_k,z_k) := J(u_k) + \sum_{m=1}^{M}\int_{I_m}(g-\partial_t u_k, z_k) - (\nabla u_k, \nabla z_k)dt - (u_k^{0,-}-u_0, z_k^{0,-})$$

$$\mathcal{L}^{cGP(r),mod}(u_k,z_k) := J(u_k) + \sum_{m=1}^{M}\int_{I_m}(g-\partial_t u_k, z_k) - (\nabla u_k, \nabla z_k)\tilde{\mu}_{r+1}^{m}(dt) - (u_k^{0,-}-u_0, z_k^{0,-}).$$

The continuous Lagrange function $\mathcal{L}$ is given as in (3.2.1). First, we formulate the DWR error representation using the cGP(r) discretization without quadrature errors, comparable to [60].

Theorem 35 (Error representation: heat-cGP(r)). *Let $u \in W(I,V)$ be the solution of* (3.2.3) *and $u_k \in V_k^r$ be the solution of the time semi-discrete problem $\mathcal{L}_z^{cGP(r)} = 0\ \forall z_k \in V_k^{DG(r-1)}$. Let further $z \in W(I,V)$ be solution of* (3.2.5) *and $z_k \in V_k^{DG(r-1)}$ solution of $\mathcal{L}_u^{cGP(r)} = 0\ \forall u_k \in V_k^r$. Then, we have the following error representation*

$$J(u) - J(u_k) = \frac{1}{2}\rho(u_k)(z-\tilde{z}_k) + \frac{1}{2}\rho^*(u_k,z_k)(u-\tilde{u}_k) + \mathcal{R}_{\mathcal{L}} \tag{A.6}$$

with arbitrary $\tilde{u}_k \in V_k^r$ and $\tilde{z}_k \in V_k^{DG(r-1)}$. The remainder term $\mathcal{R}_{\mathcal{L}}$ is given by (3.1.2) *with $L = \mathcal{L}$.*

Proof. The proof is analogous to the proof of the error representation heat-DG(r), Theorem 12 with $L = \mathcal{L}$, subspaces $Y_1 = W(I,V)^2$ and $Y_2 = V_k^r \times V_k^{DG(r-1)}$ of $Y = W(I,V) \times (W(I,V) + V_k^{DG(r-1)})$. □

This theorem is not so useful, mainly because quadrature errors are neglected. Therefore we now give the error representation result for the cGP(r) discretization incorporating quadrature errors.

Theorem 36 (Error representation: heat-cGP(r)-mod). *Let $u \in W(I,V)$ be the solution of* (3.2.3) *and $u_k \in V_k^r$ be the solution of the time semi-discrete problem $\mathcal{L}_z^{cGP(r),mod} = 0\ \forall z_k \in V_k^{DG(r-1)}$. Let further $z \in W(I,V)$ be solution of* (3.2.5) *and $z_k \in V_k^{DG(r-1)}$ solution of $\mathcal{L}_u^{cGP(r),mod} = 0\ \forall u_k \in V_k^r$. Then, we have the following error representation*

$$J(u) - J(u_k) = \rho(u_k)(z_k) + \frac{1}{2}\rho(u_k)(z-\tilde{z}_k) + \frac{1}{2}\rho^*(u_k,z_k)(u-\tilde{u}_k) + \mathcal{R}_{\mathcal{L}} + \mathcal{R}_{p,g}$$

with arbitrary $\tilde{u}_k \in V_k^r$ and $\tilde{z}_k \in V_k^{DG(r-1)}$. The remainder term $\mathcal{R}_{\mathcal{L}}$ is given by (3.1.2) *with*

$L = \mathcal{L}$ *and* $\mathcal{R}_{p,g}$ *is given by*

$$\mathcal{R}_{p,g} = \sum_{m=1}^{M} \Big\{ \int_{I_m} (g, \tilde{z}_k - z_k) dt - \int_{I_m} (g, \tilde{z}_k - z_k) \tilde{\mu}_{r+1}^m (dt) \Big\}.$$

Proof. The proof is analogous to the proof of Theorem 13 using $L = \mathcal{L}$ and $\tilde{L} = \mathcal{L}^{cGP(r),mod}$, subspaces $Y_1 = W(I,V)^2$ and $Y_2 = V_k^r \times V_k^{DG(r-1)}$ of $Y = W(I,V) \times (W(I,V) + V_k^{DG(r-1)})$. Only the remainder term $\mathcal{R}_{p,g}$ remains here, as we are using a method of order $2r-1$. □

Remark 37 (Crank-Nicolson)**.** *The error representation from Theorem 39 with* $r = 1$ *is exactly that one derived in Theorem 13 with* $\theta = 0.5$*. The remainder term can be limited by*

$$\begin{aligned} \mathcal{R}_{p,g} &= \sum_{m=1}^{M} \Big\{ \int_{I_m} (g, \tilde{w}_k - w_k) dt - \int_{I_m} (g, \tilde{w}_k - w_k) \tilde{\mu}_{r+1}^m (dt) \Big\} \\ &\leq \sum_{m=1}^{M} k_m^2 \| \partial_t^2 (g, \tilde{z}_k - z_k) \|_{L^1(I_m, \mathbb{R})} = O(k^3) \end{aligned}$$

by inserting $\tilde{z}_k = \Pi_k z$*. This coincides in the sense, that the* θ *discretization for* $\theta = 0.5$ *is exactly the cGP(1) discretization.*

A.2.3. Navier-Stokes equations

To define the time discretization, we generalize the ansatz space X_k^1 to continuous velocities of polynomial degree r. For $r \in \mathbb{N}$, we define the semi-discrete space X_k^r as

$$\begin{aligned} X_k^r = \{ u_k = (v_k, p_k)^T : & v_k \in C(\bar{I}, V), v_k|_{I_m} \in P^r(I_m, V^2), \nabla \cdot v_k(0) = 0 \\ & \text{and } p_k|_{I_m} \in P^r(I_m, L_0^2(\Omega)) \}. \end{aligned}$$

With this definition we can formulate the semi-discrete cGP(r) formulation of the NSE by using X_k^r as ansatz space and $X_k^{DG(r-1)}$ as test space, see, for example, [5, 35]. The

Lagrange functions $\mathcal{L}^{cGP(r)}, \mathcal{L}^{cGP(r),mod} : X_k^r \times X_k^{DG(r-1)} \to \mathbb{R}$ are given by

$$\begin{aligned}\mathcal{L}^{cGP(r)}(u_k, z_k) :=& J(u_k) + \sum_{m=1}^{M} \int_{I_m} (g - \partial_t u_k, z_k) - a(v_k)(w_k) + b(p_k, w_k) - b(q_k, v_k) dt \\ & - (v_k^0 - v_0, w_k^0) \\ \mathcal{L}^{cGP(r),mod}(u_k, z_k) :=& J(u_k) + \sum_{m=1}^{M} \int_{I_m} (g - \partial_t u_k, z_k) - a(v_k)(w_k) + b(p_k, w_k) \\ & - b(q_k, v_k) d\tilde{\mu}_{r+1}^m - (v_k^0 - v_0, w_k^0)\end{aligned}$$

with $u_k = (v_k, p_k)^T$ and $z_k = (w_k, q_k)^T$. The continuous Lagrange function $\mathcal{L}$ is given as in (3.3.1). We have the following DWR error representation theorem for the cGP(r) discretization.

Theorem 38 (Error representation: NSE-cGP(r)). *Let $u \in X$ be the solution of* (3.3.3) *and $u_k \in X_k^r$ solution of the time semi-discrete problem $\mathcal{L}_z^{cGP(r)} = 0 \ \forall z_k \in X_k^{DG(r-1)}$. Let further $z \in X$ be solution of* (3.3.5) *and and $z_k \in X_k^{DG(r-1)}$ solution of $\mathcal{L}_u^{cGP(r)} = 0 \ \forall u_k \in X_k^r$. Then we have the following error representation*

$$J(u) - J(u_k) = \frac{1}{2}\rho(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*(u_k, z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}}$$

with arbitrary $\tilde{u}_k \in X_k^r$ and $\tilde{z}_k \in X_k^{DG(r-1)}$. The remainder term $\mathcal{R}_{\mathcal{L}}$ is given by (3.1.2) *with $L = \mathcal{L}$.*

Proof. The proof is analogous to the proof of Theorem 15. Here we use $L = \mathcal{L}$ and the subspaces $Y_1 = X^2$ and $Y_2 = X_k^r \times X_k^{DG(r-1)}$ of $Y = X \times (X + X_k^{DG(r-1)})$. □

This theorem is not very useful, mainly because quadrature errors are neglected. Now, we reformulate the cGP error representation using modified cGP discretization.

Theorem 39 (Error representation: NSE-cGP(r)-mod). *Let $u \in X$ be the solution of* (3.3.3) *and $u_k \in X_k^r$ solution of the time semi-discrete problem $\mathcal{L}_z^{cGP(r),mod} = 0 \ \forall z_k \in X_k^{DG(r-1)}$. Let further $z \in X$ be solution of* (3.3.5) *and $z_k \in X_k^{DG(r-1)}$ solution of $\mathcal{L}_u^{cGP(r),mod} = 0 \ \forall u_k \in X_k^r$. Then we have the following error representation*

$$J(u) - J(u_k) = \rho(u_k)(z_k) + \frac{1}{2}\rho(u_k)(z - \tilde{z}_k) + \frac{1}{2}\rho^*(u_k, z_k)(u - \tilde{u}_k) + \mathcal{R}_{\mathcal{L}} + \mathcal{R}_p + \mathcal{R}_d$$

with arbitrary $\tilde{u}_k \in X_k^r$ and $\tilde{z}_k \in X_k^{DG(r-1)}$ and

$$\mathcal{R}_p = \mathcal{R}_{p,g} + \mathcal{R}_{p,conv} \text{ and } \mathcal{R}_d = \mathcal{R}_{d,conv}.$$

Proof. The proof is analogous to the proof of Theorem 24 with $L = \mathcal{L}$ and $\tilde{L} = \mathcal{L}^{cGP(r),mod}$, subspaces $Y_1 = X^2$ and $Y_2 = X_k^r \times X_k^{DG(r-1)}$ of $Y = X \times (X + X_k^{DG(r-1)})$. Again, only the remainder terms $\mathcal{R}_{p,g}, \mathcal{R}_{p,conv}$ and $\mathcal{R}_{d,conv}$ remain here, as we are using a method of order $2r-1$.
Term $\mathcal{R}_{p,g}$ is given by

$$\mathcal{R}_{p,g} = \sum_{m=1}^{M} \Big\{ \int_{I_m} (g, \tilde{w}_k - w_k) dt - \int_{I_m} (g, \tilde{w}_k - w_k) \tilde{\mu}_{r+1}^m(dt) \Big\}.$$

Terms $\mathcal{R}_{p,conv}$ and $\mathcal{R}_{d,conv}$: It holds

$$\mathcal{R}_{p,conv} = \sum_{m=1}^{M} \int_{I_m} (\nabla \cdot v_k v_k, (\tilde{w}_k - w_k)) dt - \sum_{m=1}^{M} \int_{I_m} (\nabla \cdot v_k v_k, (\tilde{w}_k - w_k)) \tilde{\mu}_{r+1}^m(dt)$$

$\mathcal{R}_{d,conv}$ analog. □

Remark 40 (Crank-Nicolson)**.** *For the error representations of the Crank-Nicolson time stepping scheme* $\theta = 0.5$ *and cGP(1)-mod, we obtain the same results, when we use the modified test space for the pressure, see Theorem 19.*

Bibliography

[1] N. Ahmed and V. John. Adaptive time step control for higher order variational time discretizations applied to convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 285:83–101, 2015.

[2] T. Akman and B. Karasözen. Variational time discretization methods for optimal control problems governed by diffusion-convection-reaction equations. *J. Comput. Appl. Math.*, 272:41–56, 2014.

[3] A. K. Aziz and P. Monk. Continuous finite elements in space and time for the heat equation. *Math. Comp.*, 52(186):255–274, 1989.

[4] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Lectures in Mathematics. ETH Zürich, Birkhäuser Basel, 2003.

[5] S. Becher and G. Matthies. Variational time discretizations of higher order and higher regularity. *BIT*, 61:721–755, 2020.

[6] R. Becker, M. Braack, D. Meidner, R. Rannacher, and B. Vexler. Adaptive finite element methods for pde-constrained optimal control problems. In W. Jäger, R. Rannacher, and J. Warnatz, editors, *Reactive Flows, Diffusion and Transport*, pages 177–205. Springer Berlin Heidelberg, 2007.

[7] R. Becker, D. Capatina, R. Luce, and D. Trujillo. Finite element formulation of general boundary conditions for incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 295:240–267, 2015.

[8] R. Becker and R. Rannacher. Weighted a posteriori error control in fe methods. In H. G. Bock et al., editor, *Proc. of ENUMATH'97*, pages 609–637. World Scientific, Singapore, 1995.

[9] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.

[10] S. Beckers. *An extended dual weighted residual error estimator for discontinuous solutions of transport problems*. PhD thesis, Universität Hamburg, 2017.

[11] D. Beigel. *Efficient goal-oriented global error estimation for BDF-type methods using discrete adjoints*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2012.

[12] J. Berg and J. Nordström. Duality based boundary conditions and dual consistent finite difference discretizations of the navier-stokes and euler equations. *J. Comput. Phys.*, 259:135–153, 02 2014.

[13] C. Bernardi, V. Girault, F. Hecht, P.-A. Raviart, and B. Rivière. *Mathematics and Finite Element Discretizations of Incompressible Navier-Stokes Flows*. Classics Appl. Math., 2024.

[14] S. Berrone and M. Marro. Space-time adaptive simulations for unsteady navier-stokes problems. *Comput. & Fluids*, 38:1132–1144, 2009.

[15] M. Besier and R. 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.

[16] M. Besier and W. Wollner. On the pressure approximation in nonstationary incompressible flow simulations on dynamically varying spatial meshes. *Internat. J. Numer. Methods Fluids*, 69(6):1045–1064, 2012.

[17] M. Braack, E. Burman, and N. Taschenberger. Duality based a posteriori error estimation for quasi-periodic solutions using time averages. *SIAM J. Sci. Comput.*, 33:2199–2216, 09 2011.

[18] D. Braess. *Finite Elemente*. Springer Spektrum Berlin, Heidelberg, 2013.

[19] M. Calvo and R.D. Grigorieff. Time discretisation of parabolic problems with the variable 3-step bdf. *BIT*, 42:689–701, 2002.

[20] D. Capatina, R. Luce, and D. Trujillo. Drag computation for incompressible flows with a nitsche's type stabilization method. *Comput. Methods Appl. Mech. Engrg.*, 360:112775, 2020.

[21] E.C. Cyr, J.N. Shadid, and T. Wildey. Towards efficient backward-in-time adjoint computations using data compression techniques. *Comput. Methods Appl. Mech. Engrg.*, 288:24–44, 2015.

[22] V. Dolejší and M. Feistauer. *Discontinuous Galerkin Method*. Springer Ser. Comput. Math., 2015.

[23] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2014.

[24] E. Emmrich. Convergence of the variable two-step bdf time discretisation of nonlinear evolution problems governed by a monotone potential operator. *BIT*, 49:297–323, 2009.

[25] B. Endtmayer, U. Langer, and T. Wick. Two-side a posteriori error estimates for the dual-weighted residual method. *SIAM J. Sci. Comput.*, 42:A371–A394, 02 2020.

[26] A. Ern and J.-L. Guermond. *Finite Elements I*. Texts Appl. Math., 2021.

[27] A. Ern and J.-L. Guermond. *Finite Elements III*. Texts Appl. Math., 2022.

[28] M. Giles, M.G. Larson, J.M. Levenstam, and E. Süli. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow. *SIAM*, 1997.

[29] M. B. Giles and E. Süli. Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 11:145–236, 2002.

[30] A. Griewank. Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optim. Methods Softw.*, 1(1):35–54, 1992.

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

[32] V. Heuveline and A. Walther. Online checkpointing for parallel adjoint computation in pdes: Application to goal-oriented adaptivity and flow control. In W.E. Nagel, W.V. Walter, and W. Lehner, editors, *Euro-Par 2006 Parallel Processing*, pages 689–699. Springer Berlin Heidelberg, 2006.

[33] J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible navier-stokes equations. *Internat. J. Numer. Methods Fluids*, 22(5):325–352, 1996.

[34] S. Hussain, F. Schieweck, and S. Turek. Higher order galerkin time discretizations and fast multigrid solvers for the heat equation. *J. Numer. Math.*, 19(1):41–61, 2011.

[35] 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.

[36] A. Iske. *Approximation*. Springer-Lehrb. Mastercl., 2018.

[37] V. John. *Parallele Lösung der inkompressiblen Navier-Stokes Gleichungen auf adaptiv verfeinerten Gittern*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 1997.

[38] V. John. Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder. *Internat. J. Numer. Methods Fluids*, 44(7):777–788, 2004.

[39] V. John. *Finite Element Methods for Incompressible Flow Problems*. Springer Cham, 2016.

[40] V. John and G. Matthies. Higher-order finite element discretizations in a benchmark problem for incompressible flows. *Internat. J. Numer. Methods Fluids*, 37(8):885–903, 2001.

[41] V. John and J. Rang. Adaptive time step control for the incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 199(9):514–524, 2010.

[42] D. Kay, P. Gresho, D. Griffiths, and D. Silvester. Adaptive time-stepping for incompressible flow part ii: Navier-stokes equations. *SIAM J. Sci. Comput.*, 32:111–128, 02 2010.

[43] X. Kerkhoff and S. May. Commutative properties for conservative space-time dg discretizations of optimal control problems involving the viscous burgers equation. *Math. Control Relat. Fields*, 13(1):193–214, 2023.

[44] K. Kraft and S. Larsson. The dual weighted residuals approach to optimal control of ordinary differential equations. *BIT*, 50:587–607, 09 2010.

[45] D. Kuzmin and S. Korotov. Goal-oriented a posteriori error estimates for transport problems. *Math. Comput. Simulation*, 80(8):1674–1683, 2010.

[46] P. Ladevèze, F. Pled, and L. Chamoin. New bounding techniques for goal-oriented error estimation applied to linear problems. *Internat. J. Numer. Methods Engrg.*, 93(13):1345–1380, 2013.

[47] D. Meidner. *Adaptive Space-Time Finite Element Methods for Optimization Problems Governed by Nonlinear Parabolic Systems*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2008.

[48] D. Meidner and T. Richter. Goal-oriented error estimation for the fractional step theta scheme. *Comput. Methods Appl. Math.*, 14(2):203–230, 2014.

[49] D. Meidner and T. Richter. A posteriori error estimation for the fractional step theta discretization of the incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 288:45–59, 2015.

[50] D. Meidner and B. Vexler. A priori error analysis of the petrov-galerkin crank-nicolson scheme for parabolic optimal control problems. *SIAM J. Control Optim.*, 49(5):2183–2211, 2011.

[51] R. Nochetto, A. Veeser, and M. Verani. A safeguarded dual weighted residual method. *IMA J. Numer. Anal.*, 29(1):126–140, 2009.

[52] R. H. Nochetto, G. Savaré, and C. Verdi. A posteriori error estimates for variable time-step discretizations of nonlinear evolution equations. *Comm. Pure Appl. Math.*, 53(5):525–589, 2000.

[53] J. Rang. Pressure corrected implicit θ-schemes for the incompressible navier–stokes equations. *Appl. Math. Comput.*, 201(1):747–761, 2008.

[54] R. Rannacher and J. Vihharev. Adaptive finite element analysis of nonlinear problems: balancing of discretization and iteration errors. *J. Numer. Math.*, 21(1):23–62, 2013.

[55] T. Richter. Goal-oriented error estimation for fluid-structure interaction problems. *Comput. Methods Appl. Mech. Engrg.*, 223-224:28–42, 2012.

[56] T. Richter and T. Wick. Variational localizations of the dual weighted residual estimator. *J. Comput. Appl. Math.*, 279:192–208, 2015.

[57] M. Schäfer, S. Turek (with support by F. Durst, E. Krause, and R. Rannacher). *Benchmark Computations of Laminar Flow Around a Cylinder*, pages 547–566. Vieweg+Teubner Verlag, Wiesbaden, 1996.

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

[59] M. Schmich. *Adaptive Finite Element Methods for Computing Nonstationary Incompressible Flows*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, 2009.

[60] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM J. Sci. Comput.*, 30(1):369–393, 2008.

[61] K. Schwegler, M. P. Bruchhauser, and M. Bause. Goal-oriented a posteriori error control for nonstationary convection-dominated transport problems. *arXiv: Numerical Analysis*, 2016.

[62] B. Schweizer. *Partielle Differentialgleichungen*. Springer-Lehrb. Mastercl., 2013.

[63] Z. K. Seymen, H. Yücel, and B. Karasözen. Distributed optimal control of time-dependent diffusion-convection-reaction equations using space-time discretization. *J. Comput. Appl. Math.*, 261:146–157, 2014.

[64] G. Söderlind. Automatic control and adaptive time-stepping. *Numer. Algorithms*, 31:281–310, 2002.

[65] R. Temam. *Navier-Stokes Equations: Theory and Numerical Analysis*. AMS Chelsea Publishing, 1984.

[66] S. Turek. *Efficient Solvers for Incompressible Flow Problems*. Springer Berlin, Heidelberg, 1999.

[67] A. Valli, G. Carey, and A. Coutinho. Control strategies for timestep selection in simulation of incompressible flows and coupled reaction-convection-diffusion processes. *Internat. J. Numer. Methods Fluids*, 47:201–231, 01 2005.

[68] S.P. Vanka. Block-implicit multigrid calculation of two-dimensional recirculating flows. *Comput. Methods Appl. Mech. Engrg.*, 59(1):29–48, 1986.

[69] R. Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Numer. Math. Sci. Comput., 2013.

[70] L. C. Wambach. *Adaptive time step control for global-in-time Galerkin-Petrov discretizations of evolution equations in the context of incompressible flows*. PhD thesis, Technische Universität Dortmund, 2025.

[71] A. I. Ávila, A. Meister, and M. Steigemann. An adaptive galerkin method for the time-dependent complex schrödinger equation. *Appl. Numer. Math.*, 121:149–169, 2017.