

Goal oriented time adaptivity using local error estimates

Peter Meisrimel¹, Philipp Birken¹

March 4, 2022

¹*Centre for the mathematical sciences, Numerical Analysis, Lund University, Lund, Sweden
email: peter.meisrimel@na.lu.se*

Abstract

We consider initial value problems where we are interested in a quantity of interest (QoI) that is the integral in time of a functional of the solution of the IVP. For these, we look into local error based time adaptivity. We derive a goal oriented error estimate and timestep controller, based on error contribution to the error in the QoI, for which we prove convergence of the error in the QoI for tolerance to zero under weak assumptions. We analyze global error propagation of this method and derive guidelines to predict performance of the method. In numerical tests we verify convergence results and guidelines on method performance. Additionally, we compare with the dual-weighted residual method (DWR) and classical local error based time-adaptivity. The local error based methods show better performance than DWR and the goal oriented method shows good results in most examples, with significant speedups in some cases.

Keywords: *Time Adaptivity, IVPs, Goal oriented problems, Error estimation*

Mathematics Subject Classification (2000): *65L05, 65L06, 65L20*

The authors gratefully acknowledge support from the Craaford Foundation under grant number 20150681.

1 Introduction

A typical situation in numerical simulations based on differential equations is that one is not interested in the solution of the differential equation per se, but a *Quantity of Interest* (QoI) that is given as a functional of the solution. For example, when designing an airplane, the QoI would be the lift coefficient divided by the drag coefficient. In simulations of the Greenland ice sheet, one would like to know the net amount of ice loss over a year. When simulating wind turbines, the amount of energy produced during a certain time period is more important than the actual flow solution.

Further examples are found in optimization problems with ODEs or PDEs as constraints. In the turbine example, one may want to optimize blade shape or determine

optimal placement of e.g. tidal turbines [8] for maximal energy output. Inverse problems in e.g. oceanography [5] can also be considered. Here the aim is to determine model parameters or initial conditions to fit measurement data to goal functions of simulation results. An example for such an inverse problem is to determine vertical mixing parameters with the QoI being the total inflow of salt water from the North Sea into the Baltic Sea.

In this article, we restrict ourselves to problems where the QoI is given as an integral over time of a functional of the solution. From the examples above, only the steady state problem in air plane design does not qualify. The basic problem we consider is thus: Given the initial value problem

$$\dot{\mathbf{u}}(t) = \mathbf{f}(t, \mathbf{u}(t)), \quad t \in [t_0, t_e], \quad \mathbf{u}(t_0) = \mathbf{u}_0, \quad (1)$$

for a sufficiently smooth function $\mathbf{f} : [t_0, t_e] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ with solution $\mathbf{u}(t)$, we are interested in the QoI

$$J(\mathbf{u}) := \int_{t_0}^{t_e} j(t, \mathbf{u}(t)) dt, \quad (2)$$

with $j : [t_0, t_e] \times \mathbb{R}^d \rightarrow \mathbb{R}$, which we will refer to as *density function*, following the notation in [1].

When solving PDEs, the system of ODEs originates from a semi-discretization, thus \mathbf{u} consists of unknowns of the space discretization. Consequently j can be used to provide spatial weighting and to select only specific points or regions of the spatial discretization.

The goal here is to determine an adaptive discrete approximation $\mathbf{u}_h \approx \mathbf{u}(t)$. Our degrees of freedom are the timesteps and we want to use as few as possible. This strategy will not yield an optimal solution, but works well in practice. We adapt the timesteps Δt using a timestep controller, which is based on local error estimates. The solution process as a whole involves a variety of schemes.

An *adaptive method* consists of a *time-integration* scheme for (1), an *error estimator*, a *timestep controller* and an *initial timestep* Δt_0 . If we consider problem (1) - (2), the adaptive method also includes a discrete approximation $J_h \approx J$ given by a *quadrature scheme*.

The input for an adaptive method is a tolerance τ , which is used in the timestep controller and possibly to determine Δt_0 . The output is an approximation to the solution, this can be a discrete solution \mathbf{u}_h , or $J_h(\mathbf{u}_h)$, depending on which problem is considered. Since we have an adaptive method, we cannot use the usual notion of convergence for $\Delta t \rightarrow 0$ for a time-integration scheme. Instead, we consider the limit of the tolerance going to zero.

Definition 1. *An adaptive method for an IVP (1) is called convergent (in the solution), if*

$$\|\mathbf{u}(t_e) - \mathbf{u}_N\| = 0, \quad \text{for } \tau \rightarrow 0,$$

where \mathbf{u}_N is an approximation to $\mathbf{u}(t_e)$ and $\|\cdot\|$ is an appropriate norm.

An adaptive method for an IVP (1) with QoI (2) is convergent in the QoI, if

$$|J(\mathbf{u}) - J_h(\mathbf{u}_h)| = 0, \quad \text{for } \tau \rightarrow 0.$$

For a convergent adaptive method we are naturally interested in the convergence rate and will express it in terms of $\mathcal{O}(\tau^q)$. This definition of adaptive methods and convergence is targeted to local error based methods, but can also be considered for methods based on global error estimates.

For goal oriented adaptivity, the standard approach is the dual weighted residual (DWR) method [2, 18]. Originally, it was developed for spatial problems, but has been extended to time dependent problems. The basic idea is to use the adjoint (dual) problem to get an estimate of the error in the QoI. In the time dependent case, the adjoint problem is a terminal value problem (IVP backwards in time). For linear problems, this gives rise to global error bounds, in the nonlinear case, global error estimates are obtained.

The DWR method is based on global a-posteriori error estimates. To obtain these error estimates one needs to subsequently integrate forward and backward in time. Here the primal and adjoint solution need to be stored. The error estimate is obtained from the primal and dual solution and is used to refine the meshes. This iterative process is repeated until a discretization is found, where the error estimate $\eta(u_h)$ fulfills

$$|J(\mathbf{u}) - J_h(\mathbf{u}_h)| \approx \eta(u_h) \leq \tau.$$

The major drawback of this method is its cost, both in implementation and computation. To reduce computational effort, Carey et. al. suggested to apply the approach in a blockwise manner, thus making it more local [4]. The storage of the primal and dual solution can be problematic for high resolutions. This, for example, can be solved by check-pointing [14, 15], but will further increase computational costs. The method requires a full variational formulation, restricting it to Galerkin type schemes in space and time.

An alternative is to use a classical time adaptive method for IVPs based on estimating the local error. Results on convergence are well established and described in standard textbooks [21, 11]. This adaptive method is not goal oriented, but can be used to solve problems with QoIs. We do not have global error bounds, since the accumulation of local errors is hard to analyse. This approach works particularly well for stiff problems, since there, local errors typically dissipate with time.

We choose a different approach, aiming to get the best of both methods. To this end we derive a new error estimator for the classic adaptive method to make it goal oriented. We estimate the time-stepwise error contribution to the error in the QoI, which consists of both quadrature and time-integration errors. Neglecting the quadrature contribution, we derive a local error estimate and use it in the deadbeat controller.

We show that convergence in the QoI follows from convergence in the solution, with additional requirements on the timesteps. The derived goal oriented adaptive method fulfills these requirements and is convergent in the QoI under weak assumptions. To obtain high convergence rates in the QoI when using higher order (> 2) time-integration schemes, one needs solutions of sufficiently high order in all quadrature evaluation points. We explain how to obtain these from the stage value of a given RK scheme.

We do our analysis for one-step methods for time-integration, embedded Runge-Kutta schemes [11] for error estimation, the deadbeat controller (13) and simple choices for Δt_0 . These restrictions are done for easier analysis, but it is straightforward to extend the results to other error estimation techniques, such as Richardson-extrapolation [11]. For different

controllers, such as PID controllers [22], our results allow for simple convergence proofs based on similarity to the deadbeat controller. The results hold for a wide range of initial timesteps and thus for any reasonable scheme used to compute Δt_0 .

Implementation of this method only requires a standard deadbeat controller, an embedded Runge-Kutta scheme and the density function $j(t, \mathbf{u}(t))$. Due to being based on local error estimates, the method is computationally very cheap. For problems where the density function only regards a small part of the state vector \mathbf{u} , the error estimate will be even cheaper than the classical one.

A similar method has been proposed by [12, 24, 25], using various other techniques for error estimation. John, Rang propose it for drag and lift coefficients in incompressible flows, but do not show numerical results [12]. Turek describes a case where using the method for an alternating lift coefficient leads to "catastrophical results" [24]. Wick uses a point-wise evaluation of the displacement field in fluid-structure interaction [25, 6]. The author describes inconsistent convergence patterns but concludes satisfying results.

To be able to make statements on the performance of the goal oriented adaptive method, we analyse the impact of global error dynamics on the error in the QoI. This analysis revolves around the nullspace of the density function $j(t, \mathbf{u})$ and thus our error estimator. A method performs well, if all relevant processes are sufficiently resolved in time. To be able to sufficiently resolve a process, its local error or the local error of a faster process, must appear in the error estimate. The question if a process is relevant for the QoI is a matter of global error dynamics. Thus, with sufficient knowledge on the global error dynamics, we are able to make predictions on the performance of the goal oriented adaptive method.

We use numerical tests with widely different global error behaviors with respect to the QoI. For these we confirm the convergence results and are able to explain the performance results. It turns out to be relatively easy to predict bad performance, but hard to predict good performance. Our results show that the local error based methods are more efficient than the DWR method. The goal oriented adaptive method shows good performance in most cases and significant speedups in some.

The structure of the article is as follows: We first review current adaptive methods in section 2, then we explain and analyse our approach in section 3. Numerical results are presented in section 4.

2 Current adaptive methods

2.1 A posteriori error estimation via the dual weighted residual method

The starting point of the DWR method is an initial value problem in variational formulation: Find $u \in U$, such that

$$A(u; v) = F(v), \quad u(t_0) = u_0, \quad \forall v \in V.$$

Here, U and V are appropriate spaces, A is linear in v and possibly nonlinear in u . Here we have $A(u; v) = (u_t, v) - (f(t, u), v)$ and $F(v) = 0$, see (1). Furthermore, there is a discrete

approximation to this problem, also in weak form: Find $u_h \in U_h$, such that

$$A(u_h; v_h) = F(v_h), \quad u_h(t_0) = u_0, \quad \forall v_h \in V_h. \quad (3)$$

Here, $U_h \subset U$ and $V_h \subset V$ are finite element spaces in time.

2.1.1 The error estimate

To obtain an estimate of the error $e^J = J(u) - J(u_h)$ in the QoI (2), one uses the linearised adjoint problem for $J(u)$: Find $z \in V$, such that

$$A'(u; v, z) = J'(u; v), \quad z(t_e) = 0, \quad \forall v \in U$$

and its discrete version

$$A'(u_h; v_h, z_h) = J'(u_h; v_h), \quad z_h(t_e) = 0, \quad \forall v_h \in U_h, \quad (4)$$

where A' and J' are the Gateaux derivatives of A and J with respect to u in direction v . Note that the adjoint problem is an initial value problem backwards in time.

An approximation of the error in the QoI is given by

$$e^J \lesssim A(u_h, z - z_h) - F(z - z_h),$$

with equality for linear functionals and approximate upper bounds for the general nonlinear case. Using an approximation $z_h^+ \approx z$, which is of higher accuracy than z_h , using e.g. higher order interpolation or a discrete solution on a finer grid [1], one gets an estimate

$$e^J \lesssim \eta(u_h) := A(u_h, z_h^+ - z_h) - F(z_h^+ - z_h). \quad (5)$$

This can be further bounded by decomposing it into timestep wise contributions and thus giving a guide on where and how to adapt. For this to work, it is imperative that the solutions of the primal and adjoint problems are obtained at all points. This can cause storage problems for long time simulations and can be dealt with using check-pointing [10].

2.1.2 Adaptation scheme

A large number of different adaptation strategies exist. Here we use a fixed-rate strategy [1], where the $r \in [0, 1]$ elements with largest error are refined. Summarizing, the following scheme is obtained.

1. Start with initial grid.
2. Solve forward problem (3) to obtain u_h .
3. Construct and solve adjoint problem (4) to obtain z_h .
4. Calculate $z_h^+ \approx z$.

5. Calculate error estimate $\eta(u_h)$.
6. Check $\eta(u_h) \leq \tau$, if not met, refine grids and restart.

The scheme is very expensive due to the need of solving adjoint problems to obtain an error estimate. While one can use generic schemes for grid adaptation, the adjoint problem and the error estimate are specific to a given equation and goal functional. Construction and solution of the adjoint problem can be automated using software such as `dolphin-adjoint` [7]. An advantage of the method is that the error estimate is global and one can expect the resulting discretizations to be of high quality.

We use a finer grid to approximate z by z_h^+ , making this the most expensive step in the computation of $\eta(u_h)$.

2.2 Time Adaptivity based on local error estimates

The second adaptive method we discuss is the standard in ODE solvers. It uses local error estimates of the solution and does not take into account QoIs. The results from this section for One-step methods and the deadbeat controller (13) are in principal classic [20].

Here, we present a new convergence proof that separates requirements on the error estimate, timesteps and Δt_0 , for generic One-step methods. This makes it easier to show convergence for general controllers and estimates, and we use it to show convergence in the QoI for the goal oriented adaptive method in section 3. We first introduce the relevant terminology used in this paper. For readers familiar with time adaptivity for ODEs, we use *local extrapolation* and *Error Per Step (EPS)* based control, see [20].

Definition 2. *The flow [23] of an IVP (1) is the map*

$$\mathcal{M}^{t,\Delta t} : \mathbf{u}(t) \rightarrow \mathbf{u}(t + \Delta t),$$

where $t \in [t_0, t_e]$ and $t + \Delta t \leq t_e$ for $\Delta t > 0$.

The flow acts as the solution operator for $\mathbf{u}(t)$. To numerically solve an IVP means to approximate the flow by a *numerical flow map* $\mathcal{N}^{t,\Delta t} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ defined by some numerical scheme. A timestep can be written in the form

$$\mathbf{u}_{n+1} = \mathcal{N}^{t_n, \Delta t_n} \mathbf{u}_n.$$

We generally assume problem (1) to have the unique solution $\mathbf{u}(t)$ guaranteeing existence of the flow map $\mathcal{M}^{t,\Delta t}$.

We define the global error by

$$\mathbf{e}_{n+1} := \mathbf{u}_{n+1} - \mathbf{u}(t_{n+1}) = \mathcal{N}^{t_n, \Delta t_n} \mathbf{u}_n - \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}(t_n). \quad (6)$$

By adding zero we obtain the global error propagation form

$$\mathbf{e}_{n+1} = \underbrace{(\mathcal{N}^{t_n, \Delta t_n} - \mathcal{M}^{t_n, \Delta t_n}) \mathbf{u}_n}_{\text{global error increment}} + \underbrace{\mathcal{M}^{t_n, \Delta t_n} \mathbf{u}_n - \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}(t_n)}_{\text{global error propagation}}. \quad (7)$$

The dynamics of global error propagation are usually not known. The global error increments, however, have a known structure.

Definition 3. Assume a sufficiently smooth right-hand side \mathbf{f} for (1). The principal error function [11] ϕ of a scheme $\mathcal{N}^{t,\Delta t}$ of order p is

$$\phi(t, \mathbf{u}) := \lim_{\Delta t \rightarrow 0} \frac{(\mathcal{N}^{t,\Delta t} - \mathcal{M}^{t,\Delta t})\mathbf{u}}{\Delta t^{p+1}}. \quad (8)$$

The local error of a scheme $\mathcal{N}^{t,\Delta t}$ of order p is

$$(\mathcal{N}^{t_n, \Delta t_n} - \mathcal{M}^{t_n, \Delta t_n}) \mathbf{u}_n = \Delta t_n^{p+1} \phi(t_n, \mathbf{u}_n) + \mathcal{O}(\Delta t_n^{p+2}).$$

Here, the local error is equivalent to the global error increment (7). We will estimate the local error and derive a timestep controller to keep the norm of the local error in check. Then we show that the resulting adaptive method is convergent, that is, the global error can be controlled by the global error increments and goes to zero for $\tau \rightarrow 0$.

2.2.1 Error estimation and timestep controller

We now derive an estimate for the local error using the two solutions of order p and \hat{p} . We approximate the local error behaviour by a simplified model, focusing on the leading terms. Aiming to keep the norm of the local error equal to a desired tolerance, this determines the new timestep. This timestep controller gives us Δt_{n+1} based on the previous timestep Δt_n , the local error estimate and a tolerance τ .

Assume two time-integration schemes $(\mathcal{N}^{t,\Delta t}, \mathcal{N}_-^{t,\Delta t})$ with orders (p, \hat{p}) and principal error functions (ϕ, ϕ_-) . Embedded Runge-Kutta schemes [11] are a possible choice, as they have the advantage that the embedded solution uses the same stage derivatives, requiring essentially no extra computation.

We use a *local extrapolation* approach to estimate the local error

$$\ell_n := (\mathcal{N}_-^{t_n, \Delta t_n} - \mathcal{M}^{t_n, \Delta t_n}) \mathbf{u}_n \quad (9)$$

by

$$\hat{\ell}_n := (\mathcal{N}_-^{t_n, \Delta t_n} - \mathcal{N}^{t_n, \Delta t_n}) \mathbf{u}_n. \quad (10)$$

The leading term of this error estimate, characterized by the principal error function ϕ_- , matches the leading term of the local error (9). Higher order terms with regards to Δt_n will differ. Note that this local error is not the global error increment from (7), but the one corresponding to $\mathcal{N}_-^{t,\Delta t}$. We model the local error using

$$\mathbf{m}_n := \Delta t_n^{\hat{p}+1} \phi_-(t_n, \mathbf{u}_n), \quad (11)$$

assuming $\phi_-(t_n, \mathbf{u}_n)$ to be slowly changing. The next step of this model yields

$$\mathbf{m}_{n+1} \approx \Delta t_{n+1}^{\hat{p}+1} \phi_-(t_n, \mathbf{u}_n) = \left(\frac{\Delta t_{n+1}}{\Delta t_n} \right)^{\hat{p}+1} \mathbf{m}_n \approx \left(\frac{\Delta t_{n+1}}{\Delta t_n} \right)^{\hat{p}+1} \hat{\ell}_n. \quad (12)$$

Aiming for $\|\mathbf{m}_{n+1}\| = \tau$ gives the well-known deadbeat controller

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\tau}{\|\hat{\ell}_n\|} \right)^{1/(\hat{p}+1)}. \quad (13)$$

2.2.2 Convergence in the solution

We now show convergence with $\mathbf{e}_n = \mathcal{O}(\tau^{p/(\hat{p}+1)})$ for the adaptive method consisting of a time-integration scheme of order p , the error estimate (10), controller (13) and a suitable initial step-size.

First we build a relation between global error and maximal timestep with Lemma 1. Corollary 2 relaxes this relation to general timesteps in dependence on the tolerance τ . We cannot use the timesteps from controller (13) directly, since their dependence on τ is more involved. Instead, we construct a reference timestep series which fulfills the requirements in both Lemma and Corollary and gives the targeted convergence rate. With Theorem 3 we show the timesteps from the controller (13) converge to the reference timesteps for $\tau \rightarrow 0$, which gives convergence with the rate $\mathcal{O}(\tau^{p/(\hat{p}+1)})$.

Lemma 1. *Let problem (1) have a sufficiently smooth right-hand side \mathbf{f} , such that a scheme $\mathcal{N}^{t, \Delta t}$ of order p has the global error increment*

$$(\mathcal{N}^{t_n, \Delta t_n} - \mathcal{M}^{t_n, \Delta t_n})\mathbf{u}_n = \Delta t_n^{p+1} \phi(t_n, \mathbf{u}_n) + \mathcal{O}(\Delta t_n^{p+2}).$$

Assume a mesh $t_0 < \dots < t_N = t_e$ with timesteps $\Delta t_n = t_{n+1} - t_n$ and the step-size function

$$\theta : [t_0, t_e] \rightarrow (\theta_{\min}, 1], \quad \theta_{\min} > 0,$$

fulfilling

$$\Delta t_n = \theta(t_n) \Delta T + \mathcal{O}(\Delta T^{1+\epsilon}), \quad \epsilon > 0, \quad (14)$$

for some $\Delta T > 0$. Then the global error (6) fulfills

$$\mathbf{e}_n = \mathbf{u}_n - \mathbf{u}(t_n) = \mathcal{O}(\Delta T^p), \quad \forall t_n, \quad n = 0, \dots, N,$$

for $\Delta T \rightarrow 0$.

Proof. We first neglect the $\mathcal{O}(\Delta T^{1+\epsilon})$ term in (14). Under these assumptions a proof of $\|\mathbf{e}_n\| = \mathcal{O}(\Delta T^p)$ can be found in [9, pp. 68].

To extend this result to $\Delta t_n = \theta(t_n) \Delta T + \mathcal{O}(\Delta T^{1+\epsilon})$, we define

$$\theta^*(t_n) := \frac{\theta(t_n)}{c} + \mathcal{O}(\Delta T^\epsilon) \quad \text{and} \quad \Delta T^* := c \Delta T,$$

for some $c > 1$. This gives $\Delta t_n = \theta^*(t_n) \Delta T^*$, where

$$\theta^*(t_n) \leq \frac{1}{c} + \mathcal{O}(\Delta T^\epsilon),$$

which fulfills $0 < \theta^*(t_n) \leq 1$ for ΔT sufficiently small. This means the general case (14) is also covered by the proof in [9, pp. 68] and for $\Delta T \rightarrow 0$ we get $\mathbf{e}_n = \mathcal{O}(\Delta T^{*p}) = \mathcal{O}(\Delta T^p)$. \square

Using this Lemma, we can now link the global error to the tolerance.

Corollary 2. *Assume the smoothness requirements of Lemma 1 to be met and assume a scheme of order p to get \mathbf{u}_n . Assume a mesh $t_0 < \dots < t_N = t_e$ with timesteps $\Delta t_n = t_{n+1} - t_n$, $n = 0, \dots, N - 1$ that fulfill*

$$\Delta t_n = \mathcal{O}(\tau^{1/q}), \quad \Delta t_n > 0.$$

Then, the global error fulfills

$$\mathbf{e}_n = \mathbf{u}_n - \mathbf{u}(t_n) = \mathcal{O}(\tau^{p/q}), \quad \forall t_n, \quad n = 0, \dots, N,$$

for $\tau \rightarrow 0$.

Proof. The maximal step-size is $\Delta T = \max\{\Delta t_n \mid 0 \leq n \leq N - 1\}$. The corresponding step-size function is

$$\theta(t_n) = \frac{\Delta t_n}{\Delta T},$$

which fulfills $0 < \theta(t_n) \leq 1$. We thus meet all assumptions of Lemma 1 and get $\mathbf{e}_n = \mathbf{u}_n - \mathbf{u}(t_n) = \mathcal{O}(\Delta T^p) = \mathcal{O}(\tau^{p/q})$. \square \square

We cannot apply Corollary 2 to the timesteps (13) directly, since they have a more complex dependence on τ . Therefore we use reference timesteps $\Delta t_n^{\text{ref}} = \mathcal{O}(\tau^{1/q})$. We show $\Delta t_n \rightarrow \Delta t_n^{\text{ref}}$ for $\tau \rightarrow 0$ with a difference of at most $\mathcal{O}(\Delta T_{\text{ref}}^2)$ and can apply Lemma 1.

We define the reference timesteps

$$\Delta t_n^{\text{ref}} := \left(\frac{\tau}{c_n \|\phi_-(t_n, \mathbf{u}(t_n))\|} \right)^{1/(\hat{p}+1)}, \quad (15)$$

where ϕ_- is the principal error function (8) corresponding to $\mathcal{N}_-^{t, \Delta t}$ and c_n is given by

$$c_n = \begin{cases} \mathcal{O}(1), & n = 0, \\ 1, & n > 0, \end{cases}$$

where $\mathcal{O}(1)$ is with respect to $\tau \rightarrow 0$ and $c_0 > 0$. This adds a degree of freedom to choose the initial timestep. For (15) to be well-defined we require \mathbf{f} in problem (1) to be sufficiently smooth and define

$$\phi_{-, \min} := \min_{t \in [t_0, t_e]} \|\phi_-(t, \mathbf{u}(t))\|,$$

where we assume $\phi_{-, \min} > 0$. This gives the maximal timestep

$$\Delta T_{\text{ref}} = \left(\frac{\tau}{\max\{1, c_0\} \phi_{-, \min}} \right)^{1/(\hat{p}+1)}. \quad (16)$$

We have $\Delta t_n^{\text{ref}} \leq \Delta T_{\text{ref}} = \mathcal{O}(\tau^{1/(\hat{p}+1)})$. Applying Lemma 1 gives us $\mathbf{e}_n = \mathcal{O}(\tau^{p/(\hat{p}+1)})$, for a time-integration scheme of order p . We now show convergence of the adaptive method with timesteps from (13).

Theorem 3. Let problem (1) have a sufficiently smooth \mathbf{f} . Assume an adaptive method consisting of:

1. A pair of schemes $(\mathcal{N}^{t,\Delta t}, \mathcal{N}_-^{t,\Delta t})$ with orders (p, \hat{p}) with $p > \hat{p}$,
2. the error estimator (10),
3. the deadbeat controller (13),
4. an initial timestep $\Delta t_0 = \mathcal{O}(\tau^{1/(\hat{p}+1)})$.

If the principal error function ϕ_- to $\mathcal{N}_-^{t,\Delta t}$ fulfills

$$\min_{t_0 \leq t \leq t_e} \|\phi_-(t, \mathbf{u}(t))\| > 0, \quad (17)$$

then the adaptive method is convergent with

$$\mathbf{e}_n = \mathbf{u}_n - \mathbf{u}(t_n) = \mathcal{O}(\tau^{p/(\hat{p}+1)}), \quad \forall t_n, \quad n = 0, \dots, N, \quad \text{and} \quad \tau \rightarrow 0.$$

Proof. By induction we show the timesteps fulfill

$$\Delta t_n = \Delta t_n^{\text{ref}} + \mathcal{O}(\Delta T_{\text{ref}}^2) = \mathcal{O}(\tau^{1/(\hat{p}+1)}).$$

We choose c_0 in Δt_0^{ref} such that $\Delta t_0 = \Delta t_0^{\text{ref}}$, meaning the *induction base* is met. The timestep given by the controller is

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\tau}{\|\hat{\mathbf{e}}_n\|} \right)^{1/(\hat{p}+1)} = \left(\frac{\tau}{\|\phi_-(t_n, \mathbf{u}_n) + \mathcal{O}(\Delta t_n)\|} \right)^{1/(\hat{p}+1)}.$$

We expand the denominator in $\phi_-(t_{n+1}, \mathbf{u}(t_{n+1}))$ and get

$$\Delta t_{n+1} = \left(\frac{\tau}{\|\phi_-(t_{n+1}, \mathbf{u}(t_{n+1})) + \mathcal{O}(\Delta t_n)\|} \right)^{1/(\hat{p}+1)}.$$

We perform another expansion to separate the $\mathcal{O}(\Delta t_n)$ term and get

$$\Delta t_{n+1} = \underbrace{\left(\frac{\tau}{\|\phi_-(t_{n+1}, \mathbf{u}(t_{n+1}))\|} \right)^{1/(\hat{p}+1)}}_{=\Delta t_n^{\text{ref}}} + \mathcal{O}(\tau^{1/(\hat{p}+1)} \Delta t_n).$$

We now consider the \mathcal{O} term. From the definition of the maximal timestep (16) we know $\Delta T_{\text{ref}} = \mathcal{O}(\tau^{1/(\hat{p}+1)})$. The induction hypothesis gives us $\Delta t_n = \Delta t_n^{\text{ref}} + \mathcal{O}(\Delta T_{\text{ref}}^2) \leq \Delta T_{\text{ref}} + \mathcal{O}(\Delta T_{\text{ref}}^2)$. Thus we showed the induction step. By Corollary 2 we then get the result $\mathbf{e}_n = \mathcal{O}(\tau^{p/(\hat{p}+1)})$. □ □

Thus we established convergence for the derived adaptive method for a suitable initial timestep Δt_0 . The assumption (17) is a requirement of controllability in the asymptotic regime. The global error would not be controllable by means of local errors, if the local error vanishes at some point. Further we built a structure with which one can prove similar results for different controllers, e.g. PID controllers [22]. To prove convergence one can either show (14) using suitable reference timesteps or show a maximal deviation of $\mathcal{O}(\Delta T_{\text{ref}}^{1+\epsilon})$, $\epsilon > 0$ of a given controller from the deadbeat controller (13).

3 Goal oriented adaptivity using local error estimates

We now consider the goal oriented setting (2) for problem (1) and are only interested in the QoI $J(\mathbf{u})$. We approximate the integral in J using quadrature and $\mathbf{u}(t)$ by the numerical solution \mathbf{u}_h to get

$$J_h(\mathbf{u}_h) := \sum_{n=0}^{N-1} \Delta t_n \sum_{k=0}^s \sigma_k j(t_n^{(k)}, \mathbf{u}_n^{(k)}) \approx \int_{t_0}^{t_e} j(t, \mathbf{u}(t)) dt = J(\mathbf{u}). \quad (18)$$

Here $\mathbf{u}_n^{(k)} \approx \mathbf{u}(t_n^{(k)})$ and σ_k are the evaluation points resp. weights for the quadrature scheme. We assume an embedded Runge-Kutta scheme for time-integration.

As we are now only interested in the QoI, we derive an adaptive method that is convergent in the QoI and goal oriented. The method aims to be more efficient by taking into account the QoI for the error estimate. Convergence in the QoI will be shown based on convergence in the solution. With the following Theorem we establish the connection between convergence rates.

Theorem 4. *Assume \mathbf{f} in problem (1) and j in the QoI (2) to be sufficiently smooth, a mesh $t_0 < \dots < t_N = t_e$ with timesteps $\Delta t_n = t_{n+1} - t_n = \mathcal{O}(\tau^{1/q})$ and an approximation $J_h \approx J$ (18) by a quadrature scheme of order r . Further assume an approximation $\mathbf{u}_h \approx \mathbf{u}(t)$ with order*

$$\mathbf{u}_n^{(k)} - \mathbf{u}(t_n^{(k)}) = \mathcal{O}(\tau^{p/q}) \quad (19)$$

for all n, k . Then the error in the QoI fulfills

$$e^J := |J(\mathbf{u}) - J_h(\mathbf{u}_h)| = \mathcal{O}(\tau^{\min(r,p)/q}).$$

Proof. By splitting the error, we obtain

$$e^J \leq \underbrace{|J(\mathbf{u}) - J_h(\mathbf{u})|}_{\text{quadrature error}} + \underbrace{|J_h(\mathbf{u}) - J_h(\mathbf{u}_h)|}_{\text{time-integration error}} \quad (20)$$

and can deal with the two errors separately.

An estimate for general numerical quadrature schemes of order r gives

$$|J(\mathbf{u}) - J_h(\mathbf{u})| \leq \left| \sum_{n=0}^{N-1} c_q \Delta t_n^{r+1} \max_{t_n \leq t \leq t_{n+1}} |(j(t, \mathbf{u}(t)))^{(r)}| \right|,$$

with a constant c_q . Using the bound $j_{\max} := \max_{t_0 \leq t \leq t_e} |(j(t, \mathbf{u}(t)))^{(r)}|$, we get

$$|J(\mathbf{u}) - J_h(\mathbf{u})| \leq c_q j_{\max} \sum_{n=0}^{N-1} \Delta t_n^{r+1} \leq c_q j_{\max} \sum_{n=0}^{N-1} \Delta t_n \mathcal{O}(\tau^{r/q}) = \mathcal{O}(\tau^{r/q}).$$

For the time-integration error we have

$$|J_h(\mathbf{u}) - J_h(\mathbf{u}_h)| \leq \sum_{n=0}^{N-1} \Delta t_n \sum_{k=0}^s \left| \sigma_k \left(j(t_n^{(k)}, \mathbf{u}_n^{(k)}) - j(t_n^{(k)}, \mathbf{u}(t_n^{(k)})) \right) \right|,$$

where we linearise $j(t_n^{(k)}, \mathbf{u}(t_n^{(k)}))$ and use assumption (19) to get

$$j(t_n^{(k)}, \mathbf{u}(t_n^{(k)})) = j(t_n^{(k)}, \mathbf{u}_n^{(k)}) + \frac{\partial j(t_n^{(k)}, \mathbf{u}_n^{(k)})}{\partial \mathbf{u}} \underbrace{(\mathbf{u}(t_n^{(k)}) - \mathbf{u}_n^{(k)})}_{=\mathcal{O}(\tau^{p/q})} + \mathcal{O}((\tau^{p/q})^2).$$

This yields

$$|J_h(\mathbf{u}) - J_h(\mathbf{u}_h)| \leq \sum_{n=0}^{N-1} \Delta t_n \sum_{k=0}^s \mathcal{O}(\tau^{p/q}) = \mathcal{O}(\tau^{p/q}).$$

Summing up quadrature and time-integration error yields

$$e^J \leq |J(\mathbf{u}) - J_h(\mathbf{u})| + |J_h(\mathbf{u}) - J_h(\mathbf{u}_h)| = \mathcal{O}(\tau^{r/q}) + \mathcal{O}(\tau^{p/q}) = \mathcal{O}(\tau^{\min(r,p)/q}). \quad (21)$$

□

□

Here, we combined statements on convergence in the QoI and the respective rates. The assumption $\Delta t_n = \mathcal{O}(\tau^{1/q})$ gives $\mathbf{u}_n - \mathbf{u}(t_n) = \mathcal{O}(\tau^{p/q})$, for all $n = 0, \dots, N-1$ by Corollary 2. Using linear interpolation for an intermediate point $t_n^{(k)} \in (t_n, t_{n+1})$, one gets at most $\mathbf{u}_n^{(k)} - \mathbf{u}(t_n^{(k)}) = \mathcal{O}(\tau^{2/q})$. The requirement (19) becomes relevant for schemes of order $p > 2$ and is discussed in the end of section 3.2.

Our idea is now to use a goal oriented error estimate to obtain step-sizes more suitable to address the error in the QoI. In practical computations this should lead to a gain in efficiency.

We first derive our error estimate and controller, for the resulting goal oriented adaptive method we show convergence in the QoI with Theorem 5. We make an analysis to predict the performance in section 3.3.

3.1 Error estimate and timestep controller

In the proof of Theorem 4 we see two different error sources - time-integration and quadrature, see (20). While one can estimate the quadrature error, doing so is not necessary.

Using an error estimate based on the time-integration error only, we will get an adaptive method that is convergent in the QoI.

Neglecting the quadrature error we have

$$e^J \approx \sum_{n=0}^{N-1} \left| \Delta t_n \sum_{k=0}^s \sigma_k \left(j(t_n^{(k)}, \mathbf{u}_n^{(k)}) - j(t_n^{(k)}, \mathbf{u}(t_n^{(k)})) \right) \right|.$$

As we generally do not have error estimates for the intermediate points of the quadrature scheme, we approximate the above term-wise by the rectangular rule

$$e_n^J := \Delta t_n \left| j(t_{n+1}, \mathbf{u}_{n+1}) - j(t_{n+1}, \mathbf{u}(t_{n+1})) \right|. \quad (22)$$

Note that this is an approximation of the time-integration error $J_h(\mathbf{u}) - J_h(\mathbf{u}_h)$ and does not place general restrictions on choices for quadrature schemes. The global error propagation form of (22) is

$$e_{n+1}^J = \Delta t_n \left(j(t_{n+1}, \mathcal{N}^{t_n, \Delta t_n} \mathbf{u}_n) - j(t_{n+1}, \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}(t_n)) \right) \quad (23)$$

$$= \underbrace{\Delta t_n \left(j(t_{n+1}, \mathcal{N}^{t_n, \Delta t_n} \mathbf{u}_n) - j(t_{n+1}, \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}_n) \right)}_{\text{global error increment}} + \quad (24)$$

$$\underbrace{\Delta t_n \left(j(t_{n+1}, \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}_n) - j(t_{n+1}, \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}(t_n)) \right)}_{\text{global error propagation}}. \quad (25)$$

Again, we do not know the global error propagation dynamics, but we can estimate the global error increment and control it using timesteps. We use local extrapolation with a scheme $\mathcal{N}_-^{t, \Delta t}$ of order $\hat{p} < p$ and control

$$\Delta t_n \ell_n^j := \Delta t_n \left(j(t_{n+1}, \mathcal{N}_-^{t_n, \Delta t_n} \mathbf{u}_n) - j(t_{n+1}, \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}_n) \right). \quad (26)$$

This is the global error increment (24), but corresponding to $\mathcal{N}_-^{t_n, \Delta t_n}$. We estimate (26) by

$$\Delta t_n \hat{\ell}_n^j := \Delta t_n \left(j(t_{n+1}, \mathcal{N}_-^{t_n, \Delta t_n} \mathbf{u}_n) - j(t_{n+1}, \mathcal{N}^{t_n, \Delta t_n} \mathbf{u}_n) \right). \quad (27)$$

To construct a controller we need a model for (26). As j may be non-linear, we linearise in \mathbf{u} and get

$$\begin{aligned} \ell_n^j &= j(t_{n+1}, \mathcal{N}_-^{t_n, \Delta t_n} \mathbf{u}_n) - j(t_{n+1}, \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}_n) \\ &= \frac{\partial j(t_{n+1}, \mathcal{M}^{t_n, \Delta t_n} \mathbf{u}_n)}{\partial \mathbf{u}} \underbrace{\left(\mathcal{N}_-^{t_n, \Delta t_n} - \mathcal{M}^{t_n, \Delta t_n} \right)}_{\stackrel{(9)}{=} \ell_n} \mathbf{u}_n + \mathcal{O}(\Delta t_n^{2\hat{p}+2}). \end{aligned} \quad (28)$$

As model we choose the leading term of (28)

$$m_n^j := \Delta t_n \frac{\partial j(t_{n+1}, \mathbf{u}_{n+1})}{\partial \mathbf{u}} \mathbf{m}_n$$

and assume the derivative term to be slowly changing. Here, \mathbf{m}_n is the classical error estimate (11). For this model the next step yields

$$\begin{aligned} m_{n+1}^j &\approx \Delta t_{n+1} \frac{\partial j(t_{n+1}, \mathbf{u}_{n+1})}{\partial \mathbf{u}} \mathbf{m}_{n+1} = \frac{\Delta t_{n+1}^{\hat{p}+2}}{\Delta t_n^{\hat{p}+1}} \frac{\partial j(t_{n+1}, \mathbf{u}_{n+1})}{\partial \mathbf{u}} \mathbf{m}_n \\ &\stackrel{(12)}{\approx} \frac{\Delta t_{n+1}^{\hat{p}+2}}{\Delta t_n^{\hat{p}+1}} \frac{\partial j(t_{n+1}, \mathbf{u}_{n+1})}{\partial \mathbf{u}} \hat{\ell}_n \stackrel{(27)-(28)}{\approx} \Delta t_{n+1} \left(\frac{\Delta t_{n+1}}{\Delta t_n} \right)^{\hat{p}+1} \hat{\ell}_n^j. \end{aligned}$$

We aim to control the error *per unit interval*, per step, which means aiming for $|m_{n+1}^j| = \Delta t_{n+1} \tau$. This is not to be confused with the common *Error Per Unit Step* (EPUS) approach in classical timestep control. We get the deadbeat controller

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\tau}{|\hat{\ell}_n^j|} \right)^{1/(\hat{p}+1)}. \quad (29)$$

We thus constructed a timestep controller to control the error in the QoI (2) using only local error estimates in $j(t, \mathbf{u})$. In the next section we show that the resulting adaptive method is convergent in the solution and QoI.

Comparing the implementation of this adaptive method to the classical one from section 2.2, we only require the density function j . This we need regardless of the used method, as it is necessary for the evaluation of $J(\mathbf{u})$.

3.2 Convergence in the quantity of interest

We now show convergence of the derived goal oriented adaptive scheme in the QoI, using Theorem 4. While we use the same controller, we have a different error estimator and cannot use Corollary 2 directly. This is due to the timesteps (29) converging to a different series of reference timesteps. We define these and repeat the steps of Theorem 3, showing convergence of the steps from the controller (29) to our reference. We use

$$\Delta t_n^{\text{ref}} := \left(\frac{\tau}{c_n \left| \frac{\partial j(t_n, \mathbf{u}(t_n))}{\partial \mathbf{u}} \phi_-(t_n, \mathbf{u}(t_n)) \right|} \right)^{1/(\hat{p}+1)}, \quad (30)$$

with

$$c_n = \begin{cases} \mathcal{O}(1), & n = 0, \\ 1, & n > 0, \end{cases}$$

where $\mathcal{O}(1)$ is for $\tau \rightarrow 0$ and $c_0 > 0$. This gives a degree of freedom in choosing Δt_0 . For the timesteps to be well-defined we require

$$\phi_{-, \min}^j := \min_{t \in [t_0, t_e]} \left| \frac{\partial j(t, \mathbf{u}(t))}{\partial \mathbf{u}} \phi_{-}(t, \mathbf{u}(t)) \right| > 0,$$

yielding the maximal timestep

$$\Delta T_{\text{ref}} := \left(\frac{\tau}{\max\{1, c_0\} \phi_{-, \min}^j} \right)^{1/(\hat{p}+1)}. \quad (31)$$

We have $\Delta t_n^{\text{ref}} \leq \Delta T_{\text{ref}} = \mathcal{O}(\tau^{1/(\hat{p}+1)})$. With the following Theorem we show convergence of the timesteps from controller (29) with error estimate (27) to the reference timesteps (30).

Theorem 5. *Let \mathbf{f} in (1) and j in (2) be sufficiently smooth. Assume an adaptive method consisting of:*

1. A pair of schemes $\mathcal{N}^{t, \Delta t}, \mathcal{N}_{-}^{t, \Delta t}$ with orders p, \hat{p} and $p > \hat{p}$,
2. a quadrature scheme of order r to approximate $J(\mathbf{u})$ as in (18),
3. schemes $\mathcal{N}_{(k)}^{t, \Delta t}$ to obtain solutions of order $p - 1$ for all quadrature evaluation points, that are not part of the resulting grid,
4. the error estimator (27),
5. the deadbeat controller (29),
6. an initial timestep $\Delta t_0 = \mathcal{O}(\tau^{1/(\hat{p}+1)})$.

If the principal error function ϕ_{-} to $\mathcal{N}_{-}^{t, \Delta t}$ fulfills

$$\min_{t \in [t_0, t_e]} \left| \frac{\partial j(t, \mathbf{u}(t))}{\partial \mathbf{u}} \phi_{-}(t, \mathbf{u}(t)) \right| > 0, \quad (32)$$

then

$$e^J := |J(\mathbf{u}) - J_h(\mathbf{u}_h)| = \mathcal{O}(\tau^{p/(\hat{p}+1)}), \quad \text{for } \tau \rightarrow 0.$$

Proof. We first show convergence in the solution by inductively showing that every step given by the controller (29) fulfills

$$\Delta t_n = \Delta t_n^{\text{ref}} + \mathcal{O}(\Delta T_{\text{ref}}^2). \quad (33)$$

We choose c_0 for Δt_0^{ref} such that $\Delta t_0 = \Delta t_0^{\text{ref}}$. Thus the *induction base* $\Delta t_0 = \Delta t_0^{\text{ref}} + \mathcal{O}(\Delta T_{\text{ref}}^2)$ is met. In the controller (29) we have the denominator

$$\begin{aligned}\hat{\ell}_n^j &= j\left(t_{n+1}, \mathcal{N}_-^{t_n, \Delta t_n} \mathbf{u}_n\right) - j\left(t_{n+1}, \mathcal{N}^{t_n, \Delta t_n} \mathbf{u}_n\right) \\ &= \frac{\partial j(t_{n+1}, \mathbf{u}_{n+1})}{\partial \mathbf{u}} \hat{\ell}_n + \mathcal{O}(\Delta t_n^{2\hat{p}+2}).\end{aligned}\quad (34)$$

Repeating the same expansions as in the proof for Theorem 3 we have

$$\hat{\ell}_n = \Delta t_n^{\hat{p}+1} (\phi_-(t_{n+1}, \mathbf{u}(t_{n+1})) + \mathcal{O}(\Delta t_n))$$

and similarly

$$\frac{\partial j(t_{n+1}, \mathbf{u}_{n+1})}{\partial \mathbf{u}} = \frac{\partial j(t_{n+1}, \mathbf{u}(t_{n+1}))}{\partial \mathbf{u}} + \mathcal{O}(e_{n+1}).$$

Here we have $\mathcal{O}(e_{n+1}) = \mathcal{O}(\Delta T_{\text{ref}}^p)$ by Corollary 2, as we assume all timesteps leading up to e_{n+1} to fulfill (33). Using these approximations in (34) we get

$$\hat{\ell}_n^j = \Delta t_n^{\hat{p}+1} \left(\frac{\partial j(t_{n+1}, \mathbf{u}(t_{n+1}))}{\partial \mathbf{u}} \phi_-(t_{n+1}, \mathbf{u}(t_{n+1})) + \mathcal{O}(\Delta t_n) + \mathcal{O}(\Delta T_{\text{ref}}^p) \right),$$

which we insert into the controller (29) to get

$$\Delta t_{n+1} = \left(\frac{\tau}{\frac{\partial j(t_{n+1}, \mathbf{u}(t_{n+1}))}{\partial \mathbf{u}} \phi_-(t_{n+1}, \mathbf{u}(t_{n+1})) + \mathcal{O}(\Delta t_n) + \mathcal{O}(\Delta T_{\text{ref}}^p)} \right)^{1/(\hat{p}+1)}.$$

Here we can pull out $\mathcal{O}(\Delta t)$ terms and use the induction hypothesis to get

$$\begin{aligned}\Delta t_{n+1} &\stackrel{(30)}{=} \Delta t_{n+1}^{\text{ref}} + \tau^{1/(\hat{p}+1)} (\mathcal{O}(\Delta t_n) + \mathcal{O}(\Delta T_{\text{ref}}^p)) \\ &\stackrel{(31), (33)}{=} \Delta t_{n+1}^{\text{ref}} + \mathcal{O}(\Delta T_{\text{ref}}^2),\end{aligned}$$

which shows the induction step holds, yielding $\Delta t_{n+1} = \mathcal{O}(\tau^{1/(\hat{p}+1)})$. We thus proved the induction and get $\Delta t_n = \mathcal{O}(\tau^{1/(\hat{p}+1)})$ for all n . This gives us the assumption on step-sizes as needed by Theorem 3.1 and convergence in the solution in the grid-points by Corollary 2, with a rate of $\mathcal{O}(\tau^{p/(\hat{p}+1)})$.

Now, we want to show that we fulfill assumption (19) in Theorem 4. Taking a single step of size $\Delta t_n^{(k)}$ from t_n , to the quadrature evaluation point $t_n^{(k)} \in (t_n, t_{n+1})$, with the scheme $\mathcal{N}_{(k)}^{t_n, \Delta t_n^{(k)}}$, gives the error

$$\begin{aligned}e_n^{(k)} &= \mathbf{u}(t_n^{(k)}) - \mathbf{u}_n^{(k)} = \mathcal{N}_{(k)}^{t_n, \Delta t_n^{(k)}} \mathbf{u}_n - M^{t_n, \Delta t_n^{(k)}} \mathbf{u}(t_n) \\ &= \underbrace{M^{t_n, \Delta t_n^{(k)}}}_{=\mathcal{O}(1)} \underbrace{e_n}_{=\mathcal{O}(\tau^{p/(\hat{p}+1)})} + \underbrace{\left(\mathcal{N}_{(k)}^{t_n, \Delta t_n^{(k)}} - M^{t_n, \Delta t_n^{(k)}} \right)}_{=\mathcal{O}\left(\left(\Delta t_n^{(k)}\right)^p\right)} \mathbf{u}_n.\end{aligned}$$

Here we have $\Delta t_n^{(k)} \leq \Delta t_n = \Delta t_n^{\text{ref}} + \mathcal{O}(\Delta T_{\text{ref}}^2) = \mathcal{O}(\tau^{p/(\hat{p}+1)})$. Hence we have $e_n^{(k)} = \mathcal{O}(\tau^{p/(\hat{p}+1)})$ and fulfill assumption (19) of Theorem 4 for $q = \hat{p}+1$, which gives us convergence in the QoI with $e^J = \mathcal{O}(\tau^{\min(r,p)/(\hat{p}+1)})$. \square \square

Thus our adaptive method is convergent in the QoI. The requirement of $\phi_{-, \min}^j > 0$ is a requirement on controllability of the global error by means of the local error (26). For a j which is linear in \mathbf{u} this is equivalent to the local error not being in the nullspace of $j(t, \cdot)$. Possible consequences of it being in the nullspace of $j(t, u)$ are shown in the following example. A more general analysis of this is subject of the next section.

Example 6. In [24] the author describes using the lift-coefficient of the flow around a cylinder as density function $j(t, \mathbf{u})$, which changes sign over time, implying it has zeros. It is observed that large timesteps are chosen when the lift-coefficient is close to zero, leading to "catastrophical results". By the description of this example we can see criterion (32) not being fulfilled and thus not guaranteeing convergence in the QoI.

We now discuss the time-integration schemes for the quadrature evaluation points, as needed in the assumptions of Theorem 5. Here, we only need a solution for the points, that are not part of the grid. This is relevant only for quadrature schemes of order $r > 2$, for $r = 2$ there is the trapezoidal rule. One can also use linear interpolation of the solution at grid points, which can be formally expressed using a combination of the identity operator and $\mathcal{N}^{t_n, \Delta t_n}$, using suitable weights. Linear interpolation will, however, yield at most $e_n^{(k)} = \mathcal{O}(\tau^{2/(\hat{p}+1)})$.

We instead want to use RK schemes and use the already calculated stage derivatives. To determine weights of RK schemes for intermediate points $t_n^{(k)} = t_n + \gamma_k \Delta t_n$ for $\gamma_k \in (0, 1]$ one has to modify the RK order conditions the following way:

Taking the order conditions for order p , e.g. $\sum_s b_s c_s = \frac{1}{2}$ for $p = 2$, one has to multiply the right-hand side by γ_k^p . This becomes clear when looking into the details of a proof on order conditions [11, pp. 142].

Example 7. Assume the classic 4th order Runge-Kutta scheme for time-integration. Since the convergence rate in the QoI (21) is determined by the minimum order of quadrature and time-integration scheme, we pick the Simpson rule ($r = 4$) for quadrature. To get a 4th order (3rd order local) solution for the point $t_n + \Delta t_n/2$ one can use the RK weights $b^* = \frac{1}{24}(5, 4, 4, -1)$ for the same stage derivatives.

3.3 The nullspace of $j(t, \mathbf{u})$ and global error propagation

Convergence of a method is a statement for the limit $\tau \rightarrow 0$ and does not regard global error dynamics. Here, we analyse them to make a qualitative statement about the grid obtained from the goal oriented adaptive method for $\tau > 0$ not in the limit. We establish guidelines to predict grid quality and thus performance of the goal-oriented adaptive method.

We assume a QoI with a density function that is linear in \mathbf{u} ,

$$j(t, \mathbf{u}(t)) = \mathbf{w}^T \mathbf{u}(t), \quad w_i \geq 0.$$

Considering the split (20) of e^J , the quadrature error does not involve the numerical solution. Global error propagation only appears in the time-integration part of the error, which in this case is given by

$$|J_h(\mathbf{u}) - J_h(\mathbf{u}_h)| \leq \sum_{n=0}^{N-1} \left| \Delta t_n \sum_{k=0}^s \sigma_k \mathbf{w}^T \mathbf{e}_n^{(k)} \right|.$$

We define a weighted seminorm $\|\cdot\|_w : \mathbb{R}^d \mapsto \mathbb{R}$ by

$$\|\mathbf{x}\|_w := \sum_{i=1}^d w_i |x_i|, \quad w_i \geq 0. \quad (35)$$

This is a seminorm, since it may have a non-trivial nullspace if $w_i = 0$ for some indices i . Throughout this section, we assume $\|\cdot\|_w$ to have a non-trivial nullspace.

Using (35) we get the bound

$$|J_h(\mathbf{u}) - J_h(\mathbf{u}_h)| \leq \sum_{n=0}^{N-1} \Delta t_n \sum_{k=0}^s \sigma_k \|\mathbf{e}_n^{(k)}\|_w$$

and we need to further investigate how $\|\mathbf{e}_n^{(k)}\|_w$ is affected by global error propagation. Starting from (23) - (25), we extract the error corresponding to a single quadrature evaluation point and get

$$\begin{aligned} j(t_{n+1}, \mathbf{e}_{n+1}^{(k)}) &= j\left(t_{n+1}, \mathcal{M}_{(k)}^{t_n, \Delta t_n} \mathbf{u}_n\right) - j\left(t_{n+1}, \mathcal{M}_{(k)}^{t_n, \Delta t_n} \mathbf{u}(t_n)\right) \\ &\quad + j\left(t_{n+1}, \mathcal{N}_{(k)}^{t_n, \Delta t_n} \mathbf{u}_n\right) - j\left(t_{n+1}, \mathcal{M}_{(k)}^{t_n, \Delta t_n} \mathbf{u}_n\right). \end{aligned}$$

Replacing j by $\|\cdot\|_w$ yields

$$\|\mathbf{e}_{n+1}^{(k)}\|_w \leq \underbrace{\left\| \mathcal{M}_{(k)}^{t_n, \Delta t_n} \mathbf{u}_n - \mathcal{M}_{(k)}^{t_n, \Delta t_n} \mathbf{u}(t_n) \right\|_w}_{\text{global error propagation}} \quad (36)$$

$$+ \underbrace{\left\| \left(\mathcal{N}_{(k)}^{t_n, \Delta t_n} - \mathcal{M}_{(k)}^{t_n, \Delta t_n} \right) \mathbf{u}_n \right\|_w}_{\text{global error increment}}. \quad (37)$$

We now want to find a bound for the global error propagation term (36) depending on \mathbf{e}_n . For this we use Lipschitz-conditions.

Assume a map $\mathbf{f} : U \rightarrow W$ fulfills the Lipschitz condition

$$\|\mathbf{f}(\mathbf{u}) - \mathbf{f}(\mathbf{v})\|_W \leq \mathcal{L} \|\mathbf{u} - \mathbf{v}\|_U$$

with some constant \mathcal{L} and suitable norms for the spaces U and W . The *Lipschitz-norm* of \mathbf{f} is the minimal \mathcal{L} fulfilling the Lipschitz condition, cf. [23]. We can define an according *Lipschitz-seminorm* in the following way.

Definition 4. Assume a map $\mathbf{f} : U \rightarrow W$ with $\|\cdot\|_w$ being a seminorm on W and $\|\cdot\|_U$ being a norm on U . We define the Lipschitz-seminorm (with respect to a given norm on U) by

$$\mathcal{L}_w[\mathbf{f}] := \sup_{\mathbf{u} \neq \mathbf{v}} \frac{\|\mathbf{f}(\mathbf{u}) - \mathbf{f}(\mathbf{v})\|_w}{\|\mathbf{u} - \mathbf{v}\|_U},$$

where $\mathbf{u}, \mathbf{v} \in U$.

Due to the possible non-trivial nullspace, we yet require a norm in the denominator. We can use this definition to bound the global error propagation in (36) - (37) by

$$\|\mathbf{e}_{n+1}^{(k)}\|_w \leq \mathcal{L}_w[\mathcal{M}_{(k)}^{t_n, \Delta t_n}] \|\mathbf{e}_n\|_U + \left\| \left(\mathcal{N}_{(k)}^{t_n, \Delta t_n} - \mathcal{M}_{(k)}^{t_n, \Delta t_n} \right) \mathbf{u}_n \right\|_w,$$

where $\|\cdot\|_U$ is a norm. It will be non-zero in the nullspace of $\|\cdot\|_w$. For a non-trivial nullspace this can be problematic, which we first illustrate by a simple test case and later using numerical test problems in section 4.

To get a better idea of the dynamics, we look at the linear case $\mathbf{f}(\mathbf{u}) = \mathbf{A}\mathbf{u}$ and take $\|\cdot\|_U$ to be the 1-norm.

Lemma 8. The Lipschitz-seminorm of a linear operator $\mathbf{A} \in \mathbb{R}^{n \times m}$, with respect to the 1-norm, is given by

$$\|\mathbf{A}\|_w = \sup_{\|\mathbf{x}\|_1=1} \|\mathbf{A}\mathbf{x}\|_w = \max_{j=1, \dots, n} \left(\sum_{i=1}^m w_i |a_{ij}| \right). \quad (38)$$

Proof. The first form using the supremum is obtained by defining $\mathbf{x} := \mathbf{u} - \mathbf{v}$ and scaling $\|\mathbf{x}\|_1 \neq 0$ to $\|\mathbf{x}\|_1 = 1$ using the homogeneity of the seminorm. For the second part we have

$$\begin{aligned} \|\mathbf{A}\mathbf{x}\|_w &= \sum_{i=1}^n w_i \sum_{j=1}^m |a_{ij} x_j| = \sum_{i=1}^n w_i \sum_{j=1}^m |a_{ij}| |x_j| \\ &= \sum_{j=1}^m |x_j| \sum_{i=1}^n w_i |a_{ij}| \leq \sum_{j=1}^m |x_j| \left(\max_{j=1, \dots, m} \sum_{i=1}^n w_i |a_{ij}| \right). \end{aligned}$$

Using the supremum over all $\|\mathbf{x}\|_1 = 1$ yields the result. \square \square

Thus the Lipschitz-seminorm of a linear operator is a (weighted) column max-seminorm of the given matrix.

Example 9. Consider

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 0 & 4 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \mathbf{A}\mathbf{x} = \begin{pmatrix} 4 \\ 8 \end{pmatrix}.$$

We have $\|\mathbf{A}\|_w = 2$, $\|\mathbf{x}\|_w = 1$, $\|\mathbf{x}\|_1 = 3$ and $\|\mathbf{Ax}\|_w = 4$. From (38) we get the inequality

$$\|\mathbf{Ax}\|_w \leq \|\mathbf{A}\|_w \|\mathbf{x}\|_1.$$

Here we fulfill $\|\mathbf{Ax}\|_w \leq \|\mathbf{A}\|_w \|\mathbf{x}\|_1$, but have $\|\mathbf{Ax}\|_w \not\leq \|\mathbf{A}\|_w \|\mathbf{x}\|_w$. This shows the inequality $\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \|\mathbf{x}\|$ for norms does not hold for seminorms.

Consider the flow map and weights

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}, \quad w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The diagonal entries of \mathbf{M} describe dampening or amplification in the nullspace or image and the off-diagonal entries describe (scaled) transport from the nullspace into the image or vice versa. This can also be formulated in an analogous blockwise formulation, then the diagonal blocks m_{11} and m_{22} may include transport inside the nullspace resp. image.

Dampening is generally favorable and amplification may be unavoidable if it is part of the ODE/PDE. Transport from the image into the nullspace is unproblematic, but transport from the nullspace to the image can be highly problematic. In this example the corresponding component is m_{21} .

We choose timesteps to control the global error increments (37) in the image. Controlling the timesteps we try to keep the seminorm of these increments below a given tolerance. We do not control the error increments in the nullspace, which can be problematic if errors from the nullspace are transported into the image.

As simulating a process results in errors regardless of the step-size, this is a question of sufficiently resolving relevant processes. Assume a process in the nullspace is faster than a process in the image at a given time. The timesteps are chosen to sufficiently resolve the slow process in the image. The process in the nullspace remains under-resolved, its error exceeding the tolerance. **If** this error is then transported into the image, the performance of the goal oriented adaptive method suffers.

Likewise the goal oriented controller performs well, if all processes whose errors end up in the image, are sufficiently resolved. This can be due to the image containing the processes which require smaller timesteps. The other case is that processes in the nullspace remain under-resolved but have neglectable impact on $J(\mathbf{u})$. This may be due to strong damping in m_{22} or lack of transport with m_{12} being small.

Due to potentially complicated dynamics of the system, it is hard to clearly identify which processes are neglectable.

Example 10. In [25] the author simulates flow-driven fracturing of an obstacle. The QoI $j(t, \mathbf{u})$ is the displacement of the obstacle in flow direction, measured at the tip of the outflow edge. Using this density function to control timesteps, there may be a small delay from the flow building up around the obstacle and the displacement occurring. This delay would result in choosing too large timesteps when the displacement is just starting to grow, but the flow pattern around the obstacle is already beginning to form. The author observes significant error reductions after a certain tolerance, which may be the point at which the inflow is sufficiently resolved.

4 Numerical Results

We now test the results of Theorem 4 on convergence rates numerically. Further we compare performance of the DWR method and the local error based adaptive methods. Verification of the results of section 2.2 are not presented as they are well established.

The experiments were run on a Intel i7-3930K 3.20 GHz CPU and implemented in Python 2.7 using FEniCS [13].

The code is available at http://www.maths.lu.se/fileadmin/maths/personal_staff/PeterMeisrimel/goal_oriented_time.zip.

The following specifications are shared for all test-cases. For local error based methods using timestep-controllers, we bound the rate by which timesteps change [11] by

$$\Delta t_{n+1} = \Delta t \min(f_{\max}, \max(f_{\min}, \text{ind})), \quad \text{ind} = \left(\frac{\tau}{\tilde{\ell}_n} \right)^{1/(\hat{p}+1)}. \quad (39)$$

Here $\tilde{\ell} = \|\hat{\ell}_n\|$ for (13) resp. $\tilde{\ell}_n = |\hat{\ell}_n^j|$ for (29). The purpose of this is to provide more computational stability by preventing too large or too small timestep changes. In practice this will not take effect for $\tau \rightarrow 0$. We use $f_{\max} = 3$ and $f_{\min} = 0.01$ and do not reject timesteps. For the initial timestep we use $\Delta t_0 = \tau^{1/(\hat{p}+1)}$.

For the DWR method we use an initial grid with 10 equidistant cells. As refinement strategy we use fixed-rate refinement [1] with $X = 0.8$ and $Y = 0$. This means we refine 80% of cells corresponding to the largest errors, where refinement means to split the cell into two equally sized cells. To approximate $z \approx z_h^+$ we use a finer grid, dividing all time-intervals in half.

We refer to the adaptive method from section 2.2 as the "Classic" method and to the one from section 3 as the "Goal oriented" method.

4.1 Test problem

As a simple test problem with known global error dynamics we consider

$$\dot{\mathbf{u}}(t) = \begin{pmatrix} -1 & 1 \\ 0 & k \end{pmatrix} \mathbf{u}(t), \quad \mathbf{u}(t_0) = \mathbf{u}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad t \in [t_0, t_e]. \quad (40)$$

We use $[t_0, t_e] = [0, 2]$ and vary the stiffness by $k < 0$.

DWR estimate

The unique solution to (40) is in $\mathcal{C}^\infty([t_0, t_e]) \times \mathcal{C}^\infty([t_0, t_e])$. We define the finite element space $V_h := \{u \in \mathcal{C}([t_0, t_e]) : u|_{I_n} \in \mathcal{P}^q(I_n)\}$ denoting the space of continuous piece-wise q -th order polynomials, where $I_n = [t_n, t_{n+1}]$. Using test-functions $\phi_h := (\phi_1, \phi_2) \in V_h \times V_h$ and $\mathbf{u}_h := (u_1, u_2) \in V_h \times V_h$ we have a weak formulation

$$\int_{t_0}^{t_e} (\dot{u}_1 + u_1 - u_2) \phi_1 + (\dot{u}_2 - k u_2) \phi_2 dt = 0.$$

Using $(\cdot, \cdot)_{I_n}$ as the standard L^2 scalar product over I_n , we have

$$\sum_{n=0}^{N-1} (\dot{u}_1 + u_1 - u_2, \phi_1)_{I_n} + (\dot{u}_2 - k u_2, \phi_2)_{I_n} = 0,$$

where the entire left-hand side defines the bilinear form $A(\mathbf{u}_h, \phi_h)$. Let \mathbf{z} be the exact adjoint solution and $\mathbf{z}_h = (z_1, z_2)$ its finite element approximation, we get $e^J = A(\mathbf{u}_h, \mathbf{z} - \mathbf{z}_h)$. We approximate \mathbf{z} by $\mathbf{z}_h^+ = (z_1^+, z_2^+)$ to get

$$\begin{aligned} e^J &\approx A(\mathbf{u}_h, \mathbf{z}_h^+ - \mathbf{z}_h) \\ &\lesssim \sum_{n=0}^{N-1} \left| \int_{t_n}^{t_{n+1}} (\dot{u}_1 + u_1 - u_2) (z_1^+ - z_1) + (\dot{u}_2 + k u_2) (z_2^+ - z_2) dt \right| \end{aligned}$$

Defining

$$\begin{aligned} R_1(t) &:= (\dot{u}_1 + u_1 - u_2) (z_1^+ - z_1), \\ R_2(t) &:= (\dot{u}_2 + k u_2) (z_2^+ - z_2), \end{aligned}$$

we get the final error estimate $\eta_h(\mathbf{u}_h)$ using the composite trapezoidal rule

$$\eta(\mathbf{u}_h) := \sum_{n=0}^{N-1} \frac{\Delta t_n}{4} \left| \sum_{i=1}^2 R_i(t_n) + 2R_i(t_n + \Delta t_n/2) + R_i(t_{n+1}) \right|.$$

4.1.1 Numerical verification of Theorem 4

We first verify Theorem 4 for the goal oriented method. Figure 1 shows results for the Crank-Nicolson scheme with Implicit Euler for error estimation, trapezoidal rule for quadrature and a range of different density functions. With $p, \hat{p} = (2, 1)$ and $r = 2$, we expect at least $e^J = \mathcal{O}(\tau)$, which the plots clearly show.

Further we consider fourth order schemes $p = r = 4$ for the goal oriented adaptive method with time-dependent density functions j . We use Simpson's rule for quadrature and the classical Runge-Kutta scheme for time-integration. As embedded scheme we use the weights $\hat{b} = \frac{1}{3}(1, 1, 0, 1)^T$, which give a second order (third order for autonomous systems) solution. To get a fourth order solution in $t_n + \Delta t_n/2$ needed by the Simpson rule we use the weights $\mathbf{b}^* = \frac{1}{24}(5, 4, 4, -1)^T$. As the test problem (40) is autonomous we have $p, \hat{p} = (4, 3)$. With $r = 4$ and 4th order solutions for all evaluation points of the quadrature scheme, we expect to get $e^J = \mathcal{O}(\tau) = \mathcal{O}(N^{-4})$ from Theorem 4. This can be observed in Figure 2.

4.1.2 Method comparison and performance tests

We now compare the DWR method with the local error based classic and goal oriented method. For the DWR method we additionally have the estimate $\eta(u_h)$ of the error e^J ,

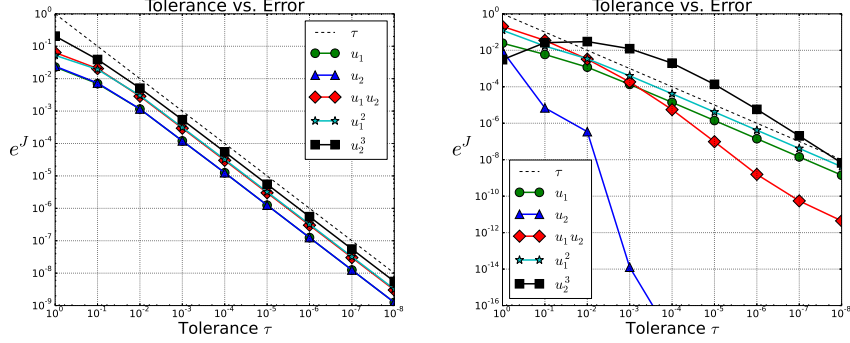


Figure 1: Verification of Theorem 4 using the goal oriented adaptive method on problem (40) for $k = -1$ (left) and $k = -100$ (right), the legend shows $j(t, \mathbf{u})$.

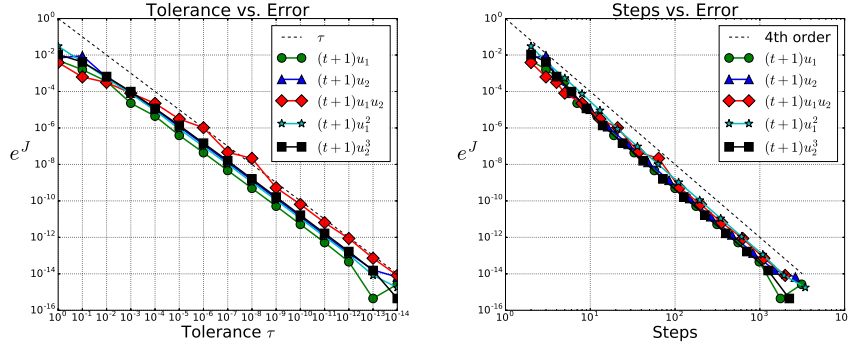


Figure 2: Verification of Theorem 4 using goal oriented adaptive method on problem (40) for $k = -1$, using 4-th order schemes, the legend shows $j(t, \mathbf{u})$.

which we denote by "DWR Est" in Figures, the actual error is denoted by "DWR Err". We only consider the final grid with $\text{DWR Est} = \eta(u_h) \leq \tau$. We use second order time-integration for both DWR and the local error methods. As DWR requires a variational formulation, we use the Crank-Nicolson scheme for time-integration and for the local error based methods Implicit Euler for error estimation. For quadrature we use the trapezoidal rule.

We compare methods in terms of computational efficiency (error vs. computational time spent) and grid quality (error vs. number of timesteps). We consider the density functions $j(t, \mathbf{u}) = u_1$ for $k \in \{-1, -100\}$ (Figures 3, 5) and $j(t, \mathbf{u}) = u_2$ for $k = -1$ (Figure 4). Results for DWR are considered first and the local error based adaptive methods are then discussed based on the results of section 3.3.

Looking at Figures 3 - 5 and considering the actual error (DWR Err), we see the method

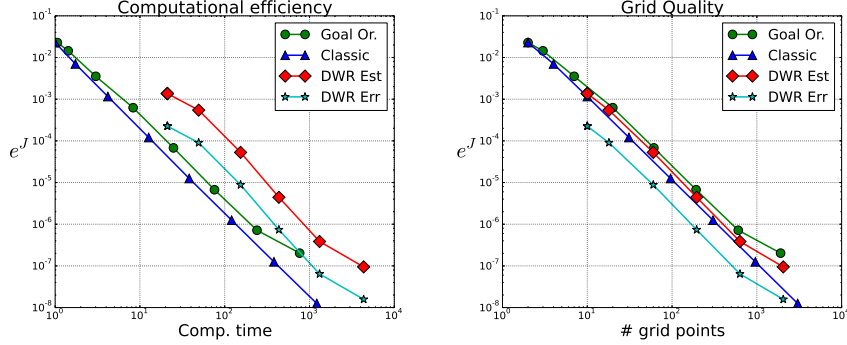


Figure 3: Performance comparison of the various methods for problem (40) for $k = -1$ and $j(t, \mathbf{u}) = u_1$.

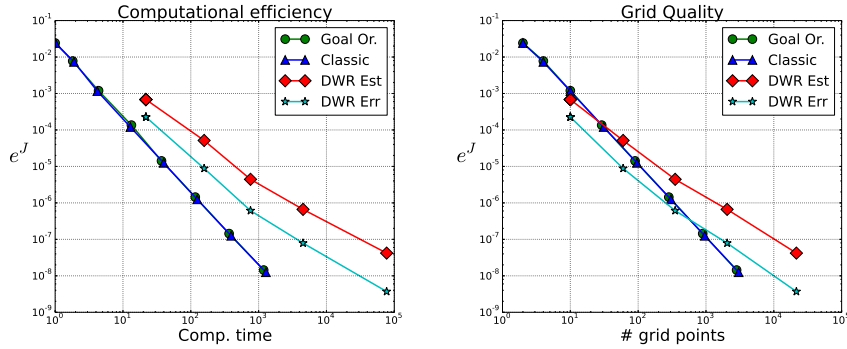


Figure 4: Performance comparison of the various methods for problem (40) for $k = -1$ and $j(t, \mathbf{u}) = u_2$.

produces the best grids. This is expected, since the method uses global grid adaptation. But the DWR method is significantly slower in performance due to the need of solving adjoint equations in computing the error estimate.

The differences in the local error methods have to be discussed for each case individually. For $k = -1$ and $j(t, \mathbf{u}) = u_1$, see Figure 3, the derivative of u_1 is slightly smaller, due to the additional off-diagonal term. Thus only controlling the error in the first component under-resolves the second component, which is relevant to $J(\mathbf{u})$ due to coupling. While not immediately evident, we do not fulfill the criterion (32) needed for convergence in the QoI, we have

$$j(t, \phi(t, \mathbf{u}(t))) = \frac{1}{2}e^{-t}(t - 1), \quad (41)$$

meaning the error estimate vanishes at $t = 1$ for $\tau \rightarrow 0$. As a result we do not have

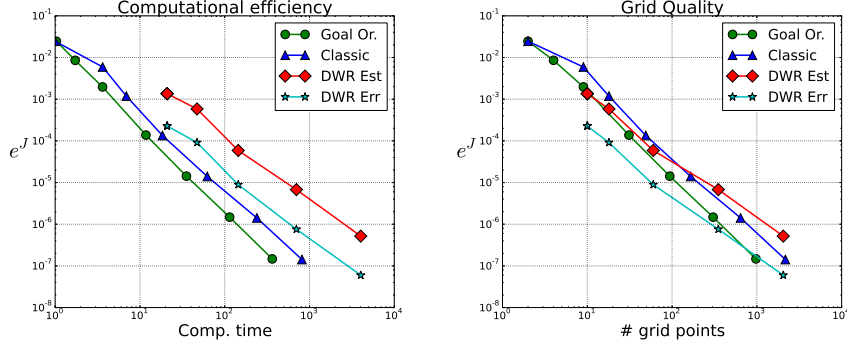


Figure 5: Performance comparison of the various methods for problem (40) for $k = -100$ and $j(t, \mathbf{u}) = u_1$.

convergence in the QoI for $\tau \rightarrow 0$, since the timestep taken at $t = 1$ will tend to infinity. This trend can be observed when looking at the timesteps over time in Figure 6, which form an upward cusp. We are, however, using an extremely small tolerance of $\tau = 10^{-14}$ and have the error $e^J \approx 4 \cdot 10^{-13}$, which is already close to machine zero. This shows that the requirement (32) may not be a strict requirement on convergence in the QoI in practice for some problems.

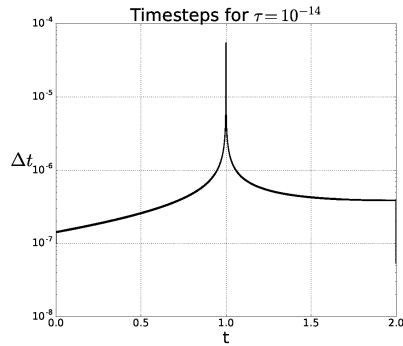


Figure 6: Timesteps for numerical solution of (40) for $j(t, \mathbf{u}) = u_1$, $k = -1$ and $\tau = 10^{-14}$. A cusp at $t \approx 1$ can be observed, where the principal error function (41) has a zero. Here we did not use the step-size limiter (39).

In the case of $k = -1$ and $j(t, \mathbf{u}) = u_2$, see Figure 4, we do control the error in the fastest process with the goal oriented method. Thus the chosen timesteps sufficiently resolve all processes. The results show that the two local error based methods have grids of identical quality and require the same computational effort.

For $k = -100$ and $j(t, \mathbf{u}) = u_1$, see Figure 5, we similarly to the case of $k = -1$ do not control the fastest process, but the impact of u_2 on $J(\mathbf{u})$ is small. It turns out the efficiency gain in not properly resolving the second component is worth the additional error, resulting in a more efficient method by a factor of around two.

4.2 Convection-diffusion equation

Moving to a problem involving a spatial component, we look at a linear convection-diffusion equation

$$\begin{aligned} \partial_t u(t, \mathbf{x}) + a \mathbf{v} \cdot \nabla u(t, \mathbf{x}) - \gamma \Delta u(t, \mathbf{x}) &= f(t, \mathbf{x}), & (t, \mathbf{x}) \in [t_0, t_e] \times \Omega, \\ u(t_0, \mathbf{x}) &= u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \nabla u \cdot \mathbf{n} &= -c \mathbf{u}(t, \mathbf{x}), & (t, \mathbf{x}) \in [t_0, t_e] \times \partial\Omega. \end{aligned} \quad (42)$$

We want to model the case of having error build-up in the nullspace of $j(t, u)$, which is transported into its image. We consider the domain $\Omega = [0, 3] \times [0, 1]$ and restrict the source term f to $\Omega_f = [0.25, 0.75] \times [0.25, 0.75]$. As QoI we consider

$$J(u) = \int_{t_0}^{t_e} j(t, u(t)) dt, \quad j(t, u(t)) = \int_{\Omega_*} \frac{u(t, \mathbf{x})}{t_e - t_0} dx. \quad (43)$$

with $\Omega_* = [2.25, 2.75] \times [0.25, 0.75]$. For a visualization of the spatial domain, see Figure 7, for the time-domain we use $[t_0, t_e] = [0, 6]$.

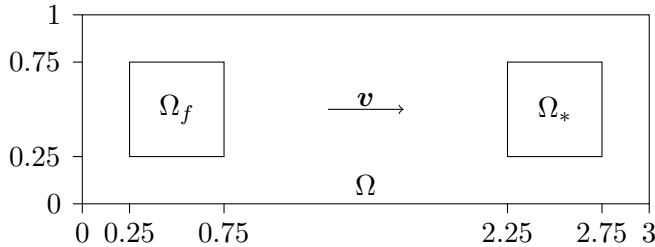


Figure 7: Geometry of Ω for the convection-diffusion equation problem (42).

We use the source term

$$f(t, \mathbf{x}) = \begin{cases} 5t^3, & \mathbf{x} \in \Omega_f \text{ and } t < 1, \\ 5(2-t)^3, & \mathbf{x} \in \Omega_f \text{ and } 1 \leq t < 2, \\ 0, & \mathbf{x} \notin \Omega_f \text{ or } 2 \leq t, \end{cases}$$

providing a spike-shaped build-up in the first 2 time units. The remaining parameters are $a = 0.5$, $\gamma = 0.01$, $c = 0.15$ and $\mathbf{v} = (1, 0)^T$. We use the initial condition $u_0(\mathbf{x}) = 1$. Since

we do not have an analytical solution, we use as reference solution from using the classic adaptive method with $\tau_{\text{ref}} = \tau_{\text{min}}/10$, where τ_{min} is the minimal tolerance for which tests are done.

Discretization

For our convection-diffusion problem we have a weak solution in the space

$$V := H^1(t_0, t_e; L^2(\Omega)) \cap L^2(t_0, t_e; H^1(\Omega)),$$

see [19]. We discretize time along the points t_n with $I_n := [t_n, t_{n+1}]$ and space by $2(3 \cdot 32) \cdot 32 = 6144$ regular triangular cells K defining the finite element mesh. We define the global finite element space by

$$V_{h,k} = \{v \in L^\infty(t_0, t_e; H^1(\Omega)) : v(\cdot, t)|_{Q_K^n} \in Q^1(K), \\ v(x, \cdot)|_{Q_K^n} \in P^q(I_n), \forall Q_K^n\},$$

where $P^q(I_n)$ is the space of polynomials on I_n of degree up to q and $Q^1(K)$ being the space of polynomials on K with partial degrees up to 1. In this space the variational formulation becomes

$$A_h(u_h, \phi_h) = F(\phi_h)$$

for all $\phi \in V_{h,k}$ with the bilinear form

$$A_h(u_h, \phi_h) := \int_{t_0}^{t_e} (\partial_t u_h, \phi_h) + a(\mathbf{v} \cdot \nabla u_h, \phi_h) - \gamma(\Delta u_h, \phi_h) dt$$

and right-hand side

$$F(\phi_h) = \int_{t_0}^{t_e} (f(t), \phi_h) dt.$$

The weak formulation is

$$\int_{t_0}^{t_e} (\partial_t u_h + a\mathbf{v} \cdot \nabla u_h - f, \phi_h) + \gamma(\nabla u_h, \nabla \phi_h) + \gamma c(u_h, \phi_h) \partial\Omega dt = 0,$$

from which one can directly write down the θ -method yielding both Crank-Nicolson and Implicit Euler.

We have the adjoint equation

$$-z_t(t, \mathbf{x}) - a\mathbf{v} \cdot \nabla z(t, \mathbf{x}) - \gamma \Delta z(t, \mathbf{x}) = \frac{1}{t_e - t_0} \Big|_{\Omega_*}, \quad (t, \mathbf{x}) \in [t_0, t_e] \times \Omega, \\ z(t_e, \mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \\ \nabla z(t, \mathbf{x}) \cdot \mathbf{n} = -\frac{a}{\gamma} z(t, \mathbf{x}) \mathbf{v} \cdot \mathbf{n} - c z(t, \mathbf{x}), \quad (t, \mathbf{x}) \in [t_0, t_e] \times \partial\Omega.$$

The weak formulation is

$$\int_{t_0}^{t_e} (z_h + a \mathbf{v} \cdot \nabla z_h, \phi_h)_\Omega + \gamma (\nabla z_h, \nabla \phi_h)_\Omega + \left(\frac{1}{t_e - t_0}, \phi_h\right)_{\Omega_*} - (c \gamma z_h + a z_h \mathbf{v} \cdot \mathbf{n}, \phi_h)_{\partial\Omega} dt = 0.$$

DWR Estimate

We have

$$\begin{aligned} e^J &= \left| \int_{t_0}^{t_e} \int_{\Omega} A_h(u_h, z - z_h) - F(z_h) dx dt \right| \\ &= \left| \int_{t_0}^{t_e} \int_{\Omega} ((u_h)_t + a \mathbf{v} \cdot \nabla u_h - \gamma \Delta u_h - f)(z - z_h) dx dt \right|, \end{aligned}$$

where we approximate $z \approx z_h^+$ using a finer grid in time. Splitting this by timesteps gives

$$e^J \lesssim \sum_{n=0}^{N-1} \left| \int_{t_n}^{t_{n+1}} \underbrace{\int_{\Omega} ((u_h)_t + a \mathbf{v} \cdot \nabla u_h - \gamma \Delta u_h - f)(z_h^+ - z_h) dx}_{=: R_z(t)} dt \right|.$$

We use the composite trapezoidal rule to get the error estimate

$$e^J \lesssim \eta_h(u_h) := \sum_{n=0}^{N-1} \frac{\Delta t_n}{4} |R_z(t_n) + 2R_z(t_n + \Delta t_n/2) + R_z(t_{n+1})|,$$

using linear interpolation for u_h and z_h in computing $R_z(t_n + \Delta t_n/2)$.

4.2.1 Method comparison and performance tests

We use the same schemes for time-integration as in section 4.1.2. We again compare DWR with the two local error based adaptive methods. The way we set up the problem, we expect the goal oriented method to perform poorly. Due to the source term being in the nullspace of $j(t, u)$, the resulting timesteps will not sufficiently resolve it. The convection transports the build-up from the source term and its error into the image of $j(t, u)$. This leads to an increase in error, which can no longer be controlled by the step-size.

The results can be seen in Figure 8. One can observe the classic adaptive method performs fine and the goal oriented adaptive method shows the expected poor performance. In Figure 9 one can see the timesteps chosen by the goal oriented method are too large to resolve the source term. Nevertheless we have convergence in the QoI with $e^J = \mathcal{O}(\tau)$, as predicted by Theorem 4, see Figure 9.

The DWR method is computationally expensive, but gives high quality grids. Here, we used it to only adapt the grid in time to get a fair comparison with the other methods.

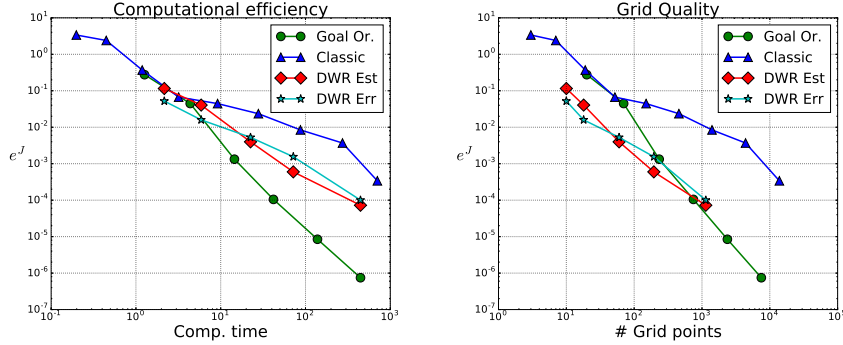


Figure 8: Performance comparison of the various methods for problem (42) with QoI (43).

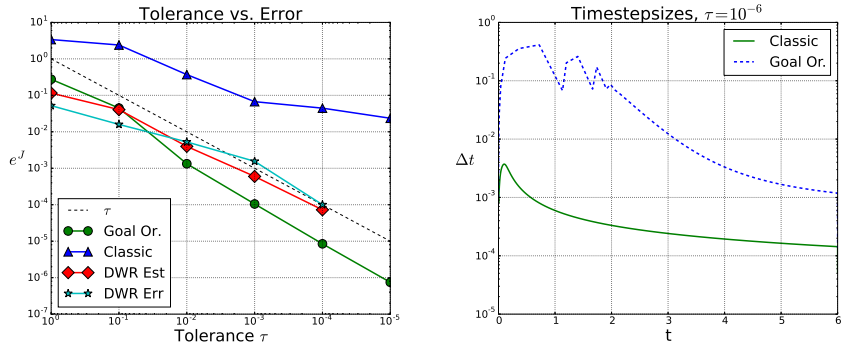


Figure 9: Solving problem (42) with QoI (43). Left: Tolerance over error for the goal oriented adaptive method; Right: Timesteps chosen by the different methods for $\tau = 10^{-6}$.

Changing the sign of the convection term we expect good results for the goal oriented method, since it is no longer required to properly resolve the source term. Considering only $[t_0, t_e] = [0, 3]$ we get the results seen in Figure 10.

The goal oriented method performs well in this example, but not better than the classic one. While not properly resolving the source term does allow larger timesteps, it does not seem to yield an advantage in terms of computational efficiency or grid quality. The DWR method performs better than in the previous examples, but is still slower than the local error based methods.

4.3 Coupled Heat equations

As a third test problem we consider the coupling of two heat equations with different thermal conductivities and diffusivities. As QoI we choose the average heat transfer over

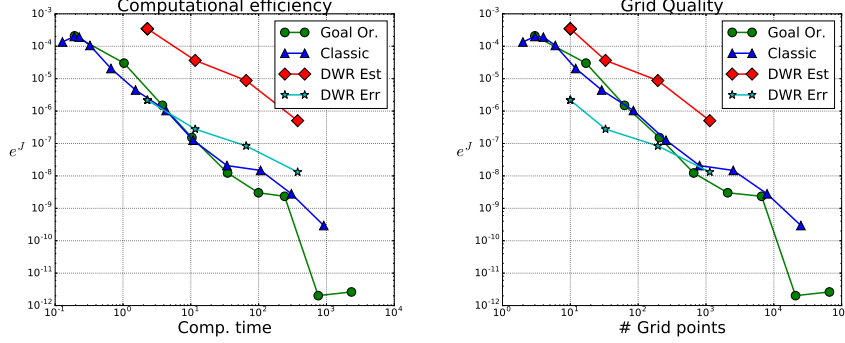


Figure 10: Performance comparison of the various methods for problem (42) and QoI (43). We changed the sign in \mathbf{v} and use $t_e = 3$.

their interface Γ . The model equations for this problem are

$$\begin{aligned}
 \alpha_m \frac{\partial u_m(t, \mathbf{x})}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m(t, \mathbf{x})) &= 0, & (t, \mathbf{x}) \in [t_0, t_e] \times \Omega_m, & \quad m = 1, 2, \\
 u(t, \mathbf{x}) &= 0, & (t, \mathbf{x}) \in [t_0, t_e] \times \Omega_m \setminus \Gamma, & \\
 u_1(t, \mathbf{x}) = u_2(t, \mathbf{x}), \quad \lambda_2 \frac{\partial u_2(t, \mathbf{x})}{\partial \mathbf{n}_2} &= -\lambda_1 \frac{\partial u_1(t, \mathbf{x})}{\partial \mathbf{n}_1}, & (t, \mathbf{x}) \in [t_0, t_e] \times \Gamma, & \quad (44) \\
 u_m(t_0, \mathbf{x}) &= u_m^0(\mathbf{x}), & \mathbf{x} \in \Omega_m, & \quad m = 1, 2.
 \end{aligned}$$

The QoI

$$J(u) = \int_{t_0}^{t_e} \int_{\Gamma} \frac{1}{t_e - t_0} \lambda_1 \frac{\partial u_1(t, \mathbf{x})}{\partial \mathbf{n}_1} dx dt = - \int_{t_0}^{t_e} \int_{\Gamma} \frac{1}{t_e - t_0} \lambda_2 \frac{\partial u_2(t, \mathbf{x})}{\partial \mathbf{n}_2} dx dt,$$

describes the time-averaged heat transfer over the interface. We consider the spatial domains $\Omega_1 = [0, 1] \times [0, 1]$, $\Omega_2 = [1, 2] \times [0, 1]$, $\Gamma = \Omega_1 \cap \Omega_2$ and $[t_0, t_e] = [0, 1]$. For discretization in space we use standard linear finite elements for both domains with identical triangular meshes for $\Delta x = 1/21$. In the discrete case the QoI becomes a summed finite difference, which we calculate based on the solution in Ω_1 .

For time-integration we use the SDIRK2 scheme, which is implicit with $(p, \hat{p}) = (2, 1)$. To solve the problem arising from the so called transmission conditions (44) on the interface Γ , we use the Dirichlet-Neumann iteration for each stage derivate of SDIRK2 [3]. In the heat equations we choose the parameters $\alpha_1 = 0.6$, $\lambda_1 = 0.3$ and $\alpha_2 = \lambda_2 = 1$. Based on the results of [17], this gives us a convergence rate of approximately $\alpha_1/\alpha_2 = 0.6$ for the Dirichlet-Neumann iteration for $\Delta t \rightarrow 0$. The cancellation criterion for the Dirichlet-Neumann iteration is based on the update between two iterates for which we use a tolerance of 10^{-10} , such that the arising error does not exceed the local errors.

Implementation of the discretization and methods are thanks to Azahar Monge, more details on the discretization in space are found in [16]. We use the initial timestep $\Delta t_0 = \tau$ for all computations. As our reference we use the solution from the classical adaptive method with $\tau = 10^{-7}$.

4.3.1 Method comparison and performance tests

Based on the results from the previous problems, we no longer consider the DWR method. While we specifically considered both grid quality and computational efficiency because of the DWR method, we now look only at grid quality, as these two performance measures are essentially identical here.

As our problem has zero Dirichlet boundary conditions and no source term, the solution will vanish. The question is how much heat transfer over the interface will occur during this process.

We consider the problem for two different sets of initial conditions given by

$$u_1^0(\mathbf{x}) = |200 \sin(\pi/2 x_1^2) \sin(\pi x_2^2)|, \quad (45)$$

$$u_2^0(\mathbf{x}) = \begin{cases} 200, & \mathbf{x} \in \Omega_2/\partial\Omega, \\ 0, & \text{otherwise.} \end{cases} \quad (46)$$

With (45), the initial conditions at the interface $x_1 = 1$ are symmetric, but the steepest

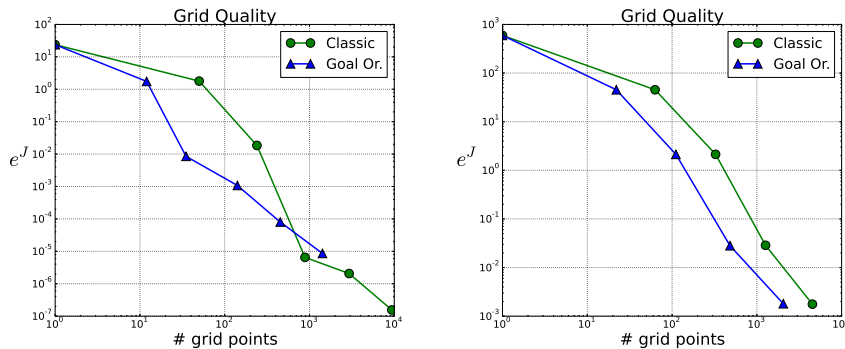


Figure 11: Performance comparison of the local error based adaptive methods for the coupled heat problem with initial conditions (45) on the left and (46) on the right.

heat gradient is inside Ω_2 . The choice of timesteps of the classical method is governed by the internal dynamics of Ω_2 , whereas the goal-oriented method will choose larger timesteps, especially in the beginning. Not correctly resolving the internal dynamics of Ω_2 does, however, not have a big impact on the values at the interface, due to diffusion. The results in Figure 11 (left) show that the grid quality of both methods are on the same parameterized

curve for sufficiently small tolerances. For the same tolerance, the classical method gives a smaller error, since it resolves the internal dynamics of Ω_2 .

For the initial condition (46), the heat transfer over the interface is a good measure of the speed of the diffusion process. While the heat gradient is likely to be steeper at the non-interface boundaries of Ω_2 , these areas have little to no impact on our QoI. Hence the classical method will choose smaller timesteps than necessary for the QoI. This is confirmed by the results in Figure 11 (right), which show that the goal oriented method performs better.

The timesteps over time in Figure 12 show that for the initial conditions (45), the chosen timesteps have a similar shape, but are shifted. This explains that the performance for both methods lie on the same curve for $\tau \rightarrow 0$. However, for the initial condition (46) the timesteps have a different shape, one which gives better performance.

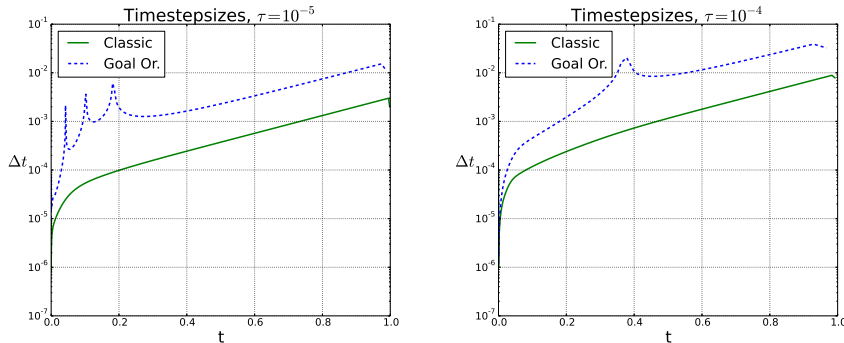


Figure 12: Timestep series from solving the coupled heat problem with initial conditions (45) for $\tau = 10^{-5}$ on the left and (46) for $\tau = 10^{-4}$ on the right.

5 Conclusions

We derived a simple and easy to implement goal oriented local error estimator. For the resulting goal oriented adaptive method we prove convergence in the QoI. The constructive nature of our proof gives us necessary requirements for convergence and on Δt_0 . Specifically, we require the error estimate to be non-zero at all times. While this is a natural assumption on controllability, one has to keep in mind that the error estimate is not based on a norm and can have a non-trivial nullspace.

A broad range of initial timesteps are allowed, as long as they are of the right order with respect to the tolerance. This means our results hold for any reasonable scheme used to compute initial timesteps.

Furthermore we show convergence rates and sufficient requirements on the involved schemes to get high convergence rates in the QoI. This involves the need for high order solutions in the quadrature evaluation points, for which we describe how to get the right

coefficients for RK schemes. The structure of our proof allows to immediately conclude the same result for closely related controllers.

We further derived guidelines to predict performance of the goal oriented method in relation to classical adaptive methods. These are based on analyzing global error propagation with respect to the nullspace of the error estimator. The goal oriented adaptive method does not regard errors in the nullspace of the error estimator when choosing timesteps. If processes in the nullspace are not sufficiently resolved by the chosen timesteps and the resulting error affects the QoI, due to global error propagation, performance of the goal oriented method will suffer. The goal oriented method will perform well, if all relevant processes are sufficiently resolved. To use these guidelines one requires sufficient knowledge of the global error dynamics of a problem.

In numerical experiments designed to test these guidelines, we confirm the results on convergence rates and that the guidelines hold true for our test-cases. We test a linear system with two variables with varying stiffness for various QoIs. As more complex test cases we have a 2D convection diffusion equation with source term that is outside the QoI. Further we test two coupled heat equations with varying coefficients and have heat transfer over the interface as the QoI.

The tests show that it is easy to correctly predict bad performance of the goal oriented method. It is, however, hard to predict if the goal oriented method will perform better than a classical norm-based adaptive method.

The results further show that the local error based adaptive methods perform better than the DWR method. The goal oriented method is shown to perform well in many cases, it is, however, not recommended to use it as a black-box solver for general goal oriented problems.

Acknowledgements

The authors want to thank Patrick Farrell for helping with FEniCS and dolfin-adjoint, Claus Führer and Gustaf Söderlind for many interesting discussions and feedback, and Azahar Monge for the implementation of the final test problem.

References

- [1] W. BANGERTH AND R. RANNACHER, *Adaptive finite element methods for differential equations*, Birkhäuser, 2013.
- [2] R. BECKER AND R. RANNACHER, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica 2001, 10 (2001), pp. 1–102.
- [3] P. BIRKEN, K. J. QUINT, S. HARTMANN, AND A. MEISTER, *A time-adaptive fluid-structure interaction method for thermal coupling*, Computing and Visualization in Science, 13 (2010), pp. 331–340.

- [4] V. CAREY, D. ESTEP, A. JOHANSSON, M. LARSON, AND S. TAVENER, *Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2121–2145.
- [5] W. CARLISLE THACKER, *Oceanographic inverse problems*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 16–37.
- [6] L. FAILER AND T. WICK, *Adaptive Time-Step Control for Nonlinear Fluid-Structure Interaction*, Journal of Computational Physics, to appear (2018).
- [7] P. E. FARRELL, D. A. HAM, S. W. FUNKE, AND M. E. ROGNES, *Automated derivation of the adjoint of high-level transient finite element programs*, SIAM Journal on Scientific Computing, 35 (2013), pp. C369–C393.
- [8] S. W. FUNKE, P. E. FARRELL, AND M. D. PIGGOTT, *Tidal turbine array optimisation using the adjoint approach*, Renewable Energy, 63 (2014), pp. 658–673.
- [9] C. W. GEAR, *Numerical initial value problems in ordinary differential equations*, Prentice Hall PTR, 1971.
- [10] A. GRIEWANK AND A. WALTHER, *Algorithm 799: Revolve: An Implementation of Checkpointing for the Reverse or Adjoint Mode of Computational Differentiation*, ACM Transactions on Mathematical Software (TOMS), 26 (2000), pp. 19–45.
- [11] E. HAIRER, S. P. NØRSETT, AND G. WANNER, *Solving ordinary differential equations I*, Springer, Berlin, 1993.
- [12] V. JOHN AND J. RANG, *Adaptive time step control for the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 514–524.
- [13] A. LOGG, K.-A. MARDAL, AND G. WELLS, *Automated solution of differential equations by the finite element method: The FEniCS book*, vol. 84, Springer, 2012.
- [14] D. MEIDNER AND T. RICHTER, *Goal-oriented error estimation for the fractional step theta scheme*, Computational Methods in Applied Mathematics, 14 (2014), pp. 203–230.
- [15] ———, *A posteriori error estimation for the fractional step theta discretization of the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 288 (2015), pp. 45–59.
- [16] A. MONGE, *The Dirichlet-Neumann iteration for unsteady thermal fluid structure interaction*, licentiate thesis, Lund University, 2016.
- [17] A. MONGE AND P. BIRKEN, *On the convergence rate of the Dirichlet-Neumann iteration for unsteady thermal fluid-structure interaction*, Computational Mechanics, (2017), pp. 1–17.

- [18] S. PRUDHOMME, *A Posteriori Error Estimates of Quantities of Interest*, 2015.
- [19] M. RENARDY AND R. C. ROGERS, *An introduction to partial differential equations*, vol. 13, Springer, 2006.
- [20] L. F. SHAMPINE, *The step sizes used by one-step codes for ODEs*, *Applied Numerical Mathematics*, 1 (1985), pp. 95–106.
- [21] ———, *Numerical solution of ordinary differential equations*, vol. 4, CRC Press, 1994.
- [22] G. SÖDERLIND, *Digital filters in adaptive time-stepping*, *ACM Transactions on Mathematical Software*, 29 (2003), pp. 1–26.
- [23] ———, *The logarithmic norm. History and modern theory*, *BIT Numerical Mathematics*, 46 (2006), pp. 631–652.
- [24] S. TUREK, *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, Springer, 6 ed., 1999.
- [25] T. WICK, *Coupling fluid-structure interaction with phase-field fracture: algorithmic details*, in *Fluid-structure interaction: Modeling, Adaptive Discretizations and Solvers*, S. Frei, B. Holm, T. Richter, T. Wick, and H. Yang, eds., *Radon Series 20*, de Gruyter, 2017, pp. 1–37.