

# A comparative study of sensitivity computations in ESDIRK-based optimal control problems

**Christensen, Anders Hilmar Damm; Jørgensen, John Bagterp**

*Published in:*
European Journal of Control

*Link to article, DOI:*
10.1016/j.ejcon.2024.101064

*Publication date:*
2024

*Document Version*
Publisher's PDF, also known as Version of record

European Journal of Control 80 (2024) 101064

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.sciencedirect.com/journal/european-journal-of-control

# A comparative study of sensitivity computations in ESDIRK-based optimal control problems

Anders Hilmar Damm Christensen, John Bagterp Jørgensen *

*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Matematiktorvet, Building 303B, Kgs. Lyngby, DK-2800, Denmark*

ARTICLE INFO

Recommended by T. Parisini

*Keywords:*
Integrator sensitivity computations
Internal numerical differentiation
ESDIRK methods
Optimal control problems
Numerical comparison

ABSTRACT

This paper compares the impact of iterated and direct approaches to sensitivity computation in fixed step-size explicit singly diagonally implicit Runge–Kutta (ESDIRK) methods when applied to optimal control problems (OCPs). We strictly use the principle of internal numerical differentiation (IND) for the iterated approach, i.e., reusing iteration matrix factorizations, the number of Newton-type iterations, and Newton iterates, to compute the sensitivities. The direct method computes the sensitivities without using the Newton schemes. We compare the impact of these sensitivity computations in OCPs for the quadruple tank system (QTS). We discretize the OCPs using multiple shooting and solve these with a sequential quadratic programming (SQP) solver. We benchmark the iterated and direct approaches against a base case. This base case applies the ESDIRK methods with exact Newton schemes and a direct approach for sensitivity computations. In these OCPs, we vary the number of integration steps between control intervals and evaluate the performance based on the number of SQP and QPs iterations, KKT violations, function evaluations, Jacobian updates, and iteration matrix factorizations. We also provide examples using the continuous-stirred tank reactor (CSTR), and the IPOPT algorithm instead of the SQP. For OCPs solved using SQP, the QTS results show the direct method converges only once, while the iterated approach and base case converges in all situations. Similar results are seen with the CSTR. Using IPOPT, both the iterated approach and base case converge in all cases. In contrast, the direct method only converges in all cases regarding the CSTR.

## 1. Introduction

Efficient computation of integrator sensitivities is essential for gradient-based optimization algorithms applied to systems described by ordinary differential equations (ODEs). Optimal control problems (OCPs) discretized by direct multiple shooting belong to this category. They require the numerical solutions of the systems to construct nonlinear programming (NLP) problems and the corresponding integrator sensitivities to form the gradients of the objective and the Jacobians of the constraints. One-step methods such as Runge–Kutta (RK) integration schemes have proven to be particularly effective when dealing with frequent discontinuities inherent in OCPs that implement zero-order hold parameterizations for the inputs (Kristensen et al., 2004).

For systems described by stiff ODEs or differential–algebraic equations, an OCP discretized using direct multiple shooting may require implicit Runge–Kutta (IRK) methods. Explicit singly diagonally implicit Runge–Kutta (ESDIRK) integrators are such types of methods and we characterize them by a diagonal structure with identical coefficients on the diagonal and with the first stage being explicit. We can implement these methods such that they reuse the matrix factorizations for all Newton-type iterations within a single integration step. We refer to this matrix as the *iteration matrix*, and its reuse leads to a computationally efficient method (Jørgensen et al., 2018; Kværnø, 2004). ESDIRK integration methods have successfully been used in OCPs. Examples include (Capolei & Jørgensen, 2012), who applies ESDIRK methods to OCPs discretized using multiple shooting for the continuous-stirred tank reactor (CSTR) and the quadruple tank system (QTS), and Capolei et al. (2012) who use ESDIRK-based single shooting methods in oil reservoir production optimization. In addition to their application in OCPs, ESDIRK integration methods have also been used in the implementation of computationally efficient extended Kalman filters for state estimation in nonlinear continuous-discrete stochastic systems (Jørgensen et al., 2007a, 2007b).

There exist numerous approaches for sensitivity computations such as the *Staggered direct method*, the *Simultaneous corrector method*, and the *Staggered corrector method* (Kristensen et al., 2004). The Staggered direct method computes the sensitivity solutions directly from the iteration matrix after the system of nonlinear equations has converged. However, it requires frequent factorization of the iteration matrix to

* Corresponding author.
*E-mail address:* jbjo@dtu.dk (J.B. Jørgensen).

https://doi.org/10.1016/j.ejcon.2024.101064
Received 15 May 2024; Accepted 10 June 2024
Available online 17 June 2024

produce accurate integrator sensitivity information. The Simultaneous corrector method solves a larger combined state and sensitivity equations system. This approach utilizes a special Jacobian structure and allows the factored iteration matrix to be reused for multiple steps while keeping a high accuracy of the sensitivities. Finally, the Staggered corrector method applies a separate Newton-type scheme to obtain sensitivities, after converging the system of nonlinear equations (Feehery et al., 1997).

The principle of *internal numerical differentiation* (IND) provides a methodology for developing sensitivities that are closely related to the integrator code (Albersmeyer, 2010; Bock, 1981). The principle of IND involves computing the sensitivities by directly differentiating the discretization scheme generated adaptively from the integrator. A strict implementation of IND for IRK methods results in reusing not only the step-sizes of the integrator but also the iteration matrix factorizations, number of Newton-type iterations, and the sequence of Newton iterates to compute sensitivities. Albersmeyer (2010) apply such sensitivity computations to backward differentiation formula (BDF) methods and refer to it as *iterated* IND. The iterated IND method is similar to the Staggered corrector method.

In an alternative IND approach to sensitivity computations, we may assume that we solve the system of nonlinear equations exactly at each integration step. This allows for direct computation of the sensitivities based on the iteration matrix factorizations without using the Newton-type scheme. Albersmeyer (2010) refer to this as the *direct* IND approach and it is identical to the Staggered direct method.

Different sensitivity computations for ESDIRK methods have already been shown, e.g., Kristensen et al. (2004, 2005) applies the direct approach for sensitivity computation for ESDIRK34 and suggests reusing the Jacobians for all the internal stages for one integrator iteration. The Staggered corrector approach is applied to ESDIRK12, ESDIRK23, and ESDIRK34 methods in Capolei and Jørgensen (2012). The direct method, unlike the iterated method, avoids using the Newton-type scheme for sensitivity computation, making it a more computationally efficient option. However, this efficiency comes at the cost of obtaining only approximate sensitivities. A computational overview of the iterated and direct approaches is shown in Albersmeyer (2010). Quirynen et al. (2012) compares various computational aspects such as LU factorizations and Jacobian updates for iterated and direct IND methods. However, there has been no clear demonstration of the impact of these sensitivities on gradient-based optimization. The sensitivity computations in OCPs are especially important when used in nonlinear model predictive control (NMPC) solutions. Even though approximate sensitivities can be computed efficiently, they can result in gradient information of poor quality. Consequently, an NMPC using a direct IND approach to integrator sensitivity may experience slow or non-convergence in the OCPs compared to an iterated IND approach.

In this paper, we compare the computational performance of using either an iterated or a direct approach for sensitivity computations in ESDIRK-based OCPs. We discretize the OCPs using direct multiple shooting and solve them with a basic implementation of a sequential quadratic program (SQP) algorithm. We compare the performances in terms of the number of SQP and QPs iterations, function evaluations, Jacobian updates, and the number of iteration matrix factorizations. Our comparison is based on repeatedly solving OCPs for a model of the QTS with varying numbers of integration steps between control intervals. We evaluate both approaches against a base case, where a direct approach to sensitivity computation is employed for an ESDIRK method using an exact Newton scheme, i.e., the iteration matrix being refactorized in all Newton-type iterations. In addition, we provide a second OCP-example using a CSTR model and we test the different sensitivity methods when using IPOPT (Wächter & Biegler, 2006).

In terms of computational performance, the iterated method is similar to the base case using the SQP algorithm for both the QTS and CSTR examples. However, the iterated method avoids some Jacobian updates and refactorizations in the integrator. In these results, the direct approach only convergences once. With IPOPT, both the iterated approach and base case converge. However, while the direct method achieves convergence in all CSTR experiments, it only achieves this in 2 out of 3 experiments for the QTS.

The rest of the paper is organized as follows. Section 2 introduces the ESDIRK integration method for ODEs. Section 3 shows how we compute sensitivities for the ESDIRK methods using the iterated and direct IND approaches. In Section 4 we present the optimal control problem. Section 5 shows numerical examples using the ESDIRK-based multiple shooting OCP with either a direct or iterated IND approach and the base case applied to the QTS. This section also shows examples using the CSTR model and the IPOPT solver. Finally, Section 6 presents conclusions.

## 2. The ESDIRK methods for ODEs

We consider the initial value problem (IVP)

$$\dot{x}(t) = f(t, x(t), u(t), d(t)), \quad t \in [t_0, t_f], \tag{1a}$$

$$x(t_0) = x_0, \tag{1b}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state vector, $u(t) \in \mathbb{R}^{n_u}$ is the vector of inputs, and $d(t) \in \mathbb{R}^{n_d}$ is the vector of disturbances. The ESDIRK integration scheme for solving (1) is

$$T_i = t_k + c_i h, \tag{2a}$$

$$X_i = x_k + h \sum_{j=1}^{i} a_{ij} f(T_j, X_j, u, d), \tag{2b}$$

$$x_{k+1} = x_k + h \sum_{i=1}^{s} b_i f(T_i, X_i, u, d), \tag{2c}$$

$$\hat{x}_{k+1} = x_k + h \sum_{i=1}^{s} \hat{b}_i f(T_j, X_j, u, d), \tag{2d}$$

where $T_i$ and $X_i$ are the internal nodes and stages at iteration $k$ for $i = 1, \dots, s$, with $s$ being the number of stages, and $x_k$ and $x_{k+1}$ are the steps computed at $t_k$ and $t_{k+1} = t_k + h$, respectively. We may compute the integration step-size $h$ based on the local error estimate $e_{k+1} = x_{k+1} - \hat{x}_{k+1}$, using the embedded method in (2d). However, for simplicity, we only consider a fixed integration step-size selection in this paper. Here, $h$ is defined as $h = \frac{t_f - t_0}{N}$, where $N$ is the number of integration steps.

Table 1 presents the Butcher tableaus for ESDIRK12, ESDIRK23, and ESDIRK34. We apply the numerical values for these tableaus from (Jørgensen et al., 2018). For the remaining part of the paper, we apply the simplified notation $f(X_i, u) = f(T_i, X_i, u(t), d(t))$.

For each integration step, we solve the systems of nonlinear equations

$$R_i(X_i) = X_i - h\gamma f(X_i, u) - \psi_i = 0, \tag{3}$$

for $i = 2, \dots, s$, with

$$\psi_i = \psi_i(\{X_j\}_{j=1}^{i-1}, x_k, u) = x_k + h \sum_{j=1}^{i-1} a_{ij} f(X_j, u). \tag{4}$$

We do this by applying the inexact Newton method

$$M_k \Delta X_i^{[l]} = -R_i(X_i^{[l]}), \tag{5a}$$

$$X_i^{[l+1]} = X_i^{[l]} + \Delta X_i^{[l]}, \tag{5b}$$

for $l = 0, \dots, w_{i,k} - 1$, with $w_{i,k}$ being the number of iterations required for stage $i$ at iteration $k$ to satisfy a chosen convergence criteria. $M_k$ is the iteration matrix at iteration $k$ defined as

$$M_k = I - h\gamma \frac{\partial f(x_k, u)}{\partial x_k} \approx \frac{\partial R_i(X_i^{[l]})}{\partial X_i^{[l]}}, \tag{6}$$

**Table 1**
Butcher tableau's for some ESDIRK methods.

| ESDIRK12 | | |
|---|---|---|
| 0 | 0 | |
| 1 | $b_1$ | $\gamma$ |
| $x_{k+1}$ | $b_1$ | $\gamma$ |
| $\hat{x}_{k+1}$ | $\hat{b}_1$ | $\hat{b}_2$ |

| ESDIRK23 | | | |
|---|---|---|---|
| 0 | 0 | | |
| $c_2$ | $a_{21}$ | $\gamma$ | |
| 1 | $b_1$ | $b_1$ | $\gamma$ |
| $x_{k+1}$ | $b_1$ | $b_1$ | $\gamma$ |
| $\hat{x}_{k+1}$ | $\hat{b}_1$ | $\hat{b}_2$ | $\hat{b}_3$ |

| ESDIRK34 | | | | |
|---|---|---|---|---|
| 0 | 0 | | | |
| $c_2$ | $a_{21}$ | $\gamma$ | | |
| $c_3$ | $a_{31}$ | $a_{32}$ | $\gamma$ | |
| 1 | $b_1$ | $b_2$ | $b_3$ | $\gamma$ |
| $x_{k+1}$ | $b_1$ | $b_2$ | $b_3$ | $\gamma$ |
| $\hat{x}_{k+1}$ | $\hat{b}_1$ | $\hat{b}_2$ | $\hat{b}_3$ | $\hat{b}_4$ |

where $I$ is the identity matrix. We solve (5) using the LU factorizations of $M_k$ and update these in every integration step. We choose the convergence criteria for the inexact Newton method as

$$\left\| R_i(X_i^{[l]}) \right\| = \max_{j\in 1,\dots,n_x} \frac{|R_i(X_i^{[l]})_j|}{\max(\text{abs}, \text{rel}(X_i^{[l]})_j)} < \tau, \tag{7}$$

where index $j$ represents the $j$th component of $R_i(X_i^{[l]})$ and $X_i^{[l]}$, and abs and rel are absolute and relative tolerances, respectively. We choose $\tau = 0.1$ for $i = 2, \dots, s$ (Capolei & Jørgensen, 2012). As the ESDIRK methods are stiffly accurate, i.e., $a_{sj} = b_j$, we avoid the computations in (2c) and obtain the next step directly as

$$x_{k+1} = X_s^{[w_{s,k}-1]}. \tag{8}$$

### 2.1. Computation of initial guesses for Newton iterations

We apply stage value predictors (SVPs) to generate initial guesses of the Newton iteration schemes (Higueras & Roldán, 2000, 2005). The SVPs use information from the previously converged step, $x_{k-1}$, and previous converged stages, $\hat{X}_j = \hat{X}_j^{[w_{j,k-1}-1]}$ for $j = 2, \dots s$, to construct guesses in the form

$$X_i^{[0]} = \alpha_i(r)x_{k-1} + \sum_{j=2}^{s} \beta(r)_{ij}\hat{X}_j, \text{ for } i = 2, \dots, s. \tag{9}$$

$\alpha(r) \in \mathbb{R}^{s-1}$ and $\beta(r) \in \mathbb{R}^{s-1\times s-1}$ are the predictor coefficients computed using step-size ratio $r = h_k/h_{k-1}$. For fixed integration step-size $r = 1$. We construct these predictors using the order conditions in Higueras and Roldán (2005). The SVPs for ESDIRK12, ($\alpha^{12}$, $\beta^{12}$), and for ESDIRK23, ($\alpha^{23}$, $\beta^{23}$) are

$$\alpha^{12}(r) = -r, \quad \beta^{12}(r) = 1 + r, \tag{10a}$$

$$\alpha^{23}(r) = \begin{bmatrix} r - 2\gamma r + 2\gamma r^2 \\ \frac{r-2\gamma r+r^2}{2\gamma} \end{bmatrix}, \tag{10b}$$

$$\beta^{23}(r) = \begin{bmatrix} \frac{2\gamma r^2+r}{2\gamma-1} & \frac{-(4\gamma^2 r^2-4\gamma^2 r+4\gamma r-2\gamma+1)}{(2\gamma-1)} \\ \frac{r^2+r}{(2\gamma(2\gamma-1))} & \frac{-(2r-2\gamma-2\gamma r+r^2+1)}{(2\gamma-1)} \end{bmatrix}. \tag{10c}$$

The SVPs for ESDIRK34 are of the form

$$\alpha^{34}(r) = \begin{bmatrix} \alpha_1(r) \\ \alpha_2(r) \\ \alpha_3(r) \end{bmatrix}, \quad \beta^{34}(r) = \begin{bmatrix} \beta_{11}(r) & \beta_{12}(r) & \beta_{13}(r) \\ \beta_{21}(r) & \beta_{22}(r) & \beta_{23}(r) \\ \beta_{31}(r) & \beta_{32}(r) & \beta_{33}(r) \end{bmatrix}, \tag{11}$$

with the coefficients shown in Appendix. For the first integration step ($k = 0$), we use the trivial predictor, i.e., $X_i^{[0]} = x_k$. We note, that due to (8), (9) applies information of $x_k$ as $\hat{X}_s^{[w_{s,k-1}-1]} = x_k$.

## 3. Sensitivity analysis

### 3.1. Iterated IND

In the iterated IND, we differentiate the adaptively generated discretization scheme, including the operations in the Newton-type scheme. The adaptive components for the fixed step-size ESDIRK methods are the LU factorizations of the iteration matrices, $M_k$, the number of Newton-type iterations for all stages, $w_{i,k}$, and the sequence of Newton iterates, $X_i^{[l]}$ for $l = 0, \dots, w_{i,k} - 1$ and for $i = 2, \dots, s$. We express the implementation of the ESDIRK methods as a combination of the elementary operations

$$X_i^{[0]} = \phi_i^{\text{SVP}}(x_{k-1}, \hat{X}), \tag{12a}$$

$$X_i^{[l+1]} = \phi_i^{\text{it}}(X_i^{[l]}, \psi_i, u) = X_i^{[l]} - M_k^{-1} R_i(X_i^{[l]}, \psi_i, u), \tag{12b}$$

$$x_{k+1} = \phi^{\text{s}}(X_s^{[w_{s,k}-1]}) = X_s^{[w_{s,k}-1]}, \tag{12c}$$

where (12a) represent the SVPs for stage $i$ in (9) with $\hat{X} = \left[\hat{X}_2^{[w_{2,k-1}-1]}; \quad \dots; \quad x_k\right]$, (12b) represent the Newton-type iterations in (5), and (12c) represent the mapping of the final stage to the next step in (8).

#### 3.1.1. State sensitivity

We construct the state sensitivities by computing the partial derivatives of (12) with respect to the initial state, $x_0$, i.e.,

$$\frac{\partial X_i^{[0]}}{\partial x_0} = \frac{\partial \phi_i^{\text{SVP}}(x_{k-1}, \hat{X})}{\partial x_0}, \tag{13a}$$

$$\frac{\partial X_i^{[l+1]}}{\partial x_0} = \frac{\partial \phi_i^{\text{it}}(X_i^{[l]}, \psi_i, u)}{\partial x_0}, \tag{13b}$$

$$\frac{\partial x_{k+1}}{\partial x_0} = \frac{\partial \phi^{\text{s}}(X_s^{[w_{s,k}-1]})}{\partial x_0}, \tag{13c}$$

for $i = 2, \dots, s$, with (13b) being

$$\frac{\partial X_i^{[l+1]}}{\partial x_0} = \frac{\partial X_i^{[l]}}{\partial x_0} - M_k^{-1} \frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial x_0}. \tag{14}$$

We compute the partial derivative of the residual function as

$$\frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial x_0} = \frac{\partial R_i}{\partial X_i^{[l]}} \frac{\partial X_i^{[l]}}{\partial x_0} + \frac{\partial R_i}{\partial \psi_i} \frac{\partial \psi_i}{\partial x_0}, \tag{15}$$

with

$$\frac{\partial R_i}{\partial X_i^{[l]}} = \left( I - h\gamma \frac{\partial f(X_i^{[l]}, u)}{\partial X_i^{[l]}} \right), \quad \frac{\partial R_i}{\partial \psi_i} = -I, \tag{16a}$$

$$\frac{\partial \psi_i}{\partial x_0} = \frac{\partial x_k}{\partial x_0} + h \sum_{j=1}^{i-1} a_{ij} \frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial x_0}, \tag{16b}$$

where $X_j = X_j^{[w_{j,k}-1]}$ for $j = 1, \dots, i-1$ in (16b) are the stages converged after $w_{j,k}$-number of Newton-type iterations at integrator iteration $k$. For (13a), we compute the state sensitivities of the SVPs as

$$\frac{\partial X_i^{[0]}}{\partial x_0} = \begin{cases} \alpha_i(r)\frac{\partial x_{k-1}}{\partial x_0} + \sum_{j=2}^{s} \beta_{ij}(r)\frac{\partial \hat{X}_j}{\partial x_0}, & \text{if } k \geq 1 \\ \frac{\partial x_k}{\partial x_0}, & \text{if } k = 0, \end{cases} \tag{17}$$

for $i = 2, \dots, s$ and for (13c) we write

$$\frac{\partial x_{k+1}}{\partial x_0} = \frac{\partial X_s^{[w_{s,k}-1]}}{\partial x_0}. \tag{18}$$

#### 3.1.2. Input sensitivity

We construct sensitivities with respect to the input as

$$\frac{\partial X_i^{[0]}}{\partial u} = \frac{\partial \phi_i^{\text{SVP}}(x_{k-1}, \hat{X})}{\partial u}, \tag{19a}$$

$$\frac{\partial X_i^{[l+1]}}{\partial u} = \frac{\partial \phi_i^{\mathrm{it}}(X_i^{[l]}, \psi_i, u)}{\partial u}, \tag{19b}$$

$$\frac{\partial x_{k+1}}{\partial u} = \frac{\partial \phi^{\mathrm{s}}(X_s^{[w_{s,k}-1]})}{\partial u}, \tag{19c}$$

for $i = 2, \dots, s$. We compute (19b) as

$$\frac{\partial X_i^{[l+1]}}{\partial u} = \frac{\partial X_i^{[l]}}{\partial u} - M_k^{-1} \frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial u}, \tag{20}$$

where the partial derivative of the residual function is

$$\frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial u} = \frac{\partial R_i}{\partial X_i^{[l]}} \frac{\partial X_i^{[l]}}{\partial u} + \frac{\partial R_i}{\partial \psi_i} \frac{\partial \psi_i}{\partial u} + \frac{\partial R_i}{\partial u}. \tag{21}$$

$\frac{\partial R_i}{\partial X_i^{[l]}}$ and $\frac{\partial R_i}{\partial \psi_i}$ are the same as in (16a) and we calculate $\frac{\partial \psi_i}{\partial u}$ and $\frac{\partial R_i}{\partial u}$ as

$$\frac{\partial \psi_i}{\partial u} = \frac{\partial x_k}{\partial u} + h \sum_{j=1}^{i-1} a_{ij} \left( \frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial u} + \frac{\partial f(X_j, u)}{\partial u} \right), \tag{22a}$$

$$\frac{\partial R_i}{\partial u} = -h\gamma \frac{\partial f(X_i^{[l]}, u)}{\partial u}. \tag{22b}$$

The contribution to the sensitivities for the SVPs in (19a) is

$$\frac{\partial X_i^{[0]}}{\partial u} = \begin{cases} \alpha_i(r) \frac{\partial x_{k-1}}{\partial u} + \sum_{j=2}^{s} \beta_{ij}(r) \frac{\partial \hat{X}_j}{\partial u}, & \text{if } k \geq 1 \\ \frac{\partial x_k}{\partial u}, & \text{if } k = 0, \end{cases} \tag{23}$$

for $i = 2, \dots, s$. For (19c) the input sensitivity at the next iteration is directly obtained as

$$\frac{\partial x_{k+1}}{\partial u} = \frac{\partial X_s^{[w_{s,k}-1]}}{\partial u}. \tag{24}$$

We initialize the state and input sensitivities at integration iteration $k = 0$ as

$$\frac{\partial x_k}{\partial x_0} = I, \quad \frac{\partial x_k}{\partial u} = 0. \tag{25}$$

#### 3.1.3. Implementations details

The ESDIRK integration schemes are implemented such that at least one Newton-type iteration is performed for each stage. This ensures that all elementary operations in (12) are included when computing the sensitivities.

### 3.2. Direct IND

In the direct IND approach, we assume that we solve (3) for $i = 2, \dots, s$, exactly. Using the implicit function theorem, this enables the direct computation of state and input sensitivities for (2) as

$$\begin{aligned} \frac{\partial X_i}{\partial x_0} &= \frac{\partial x_k}{\partial x_0} + h \sum_{j=1}^{i} a_{ij} \frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial x_0} \\ &= J_i^{-1} \frac{\partial \psi_i}{\partial x_0}, \end{aligned} \tag{26a}$$

$$\begin{aligned} \frac{\partial X_i}{\partial u} &= \frac{\partial x_k}{\partial u} + h \sum_{j=1}^{i} a_{ij} \left( \frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial u} + \frac{\partial f(X_j, u)}{\partial u} \right) \\ &= J_i^{-1} \left( \frac{\partial \psi_i}{\partial u} + h\gamma \frac{\partial f(X_i, u)}{\partial u} \right), \end{aligned} \tag{26b}$$

where $J_i = \frac{\partial R_i(X_i)}{\partial X_i}$, and $\frac{\partial \psi_i}{\partial x_0}$ and $\frac{\partial \psi_i}{\partial u}$ are described in (16b) and (22a), respectively. We apply the approximation in (6) to reuse the factorizations of $\frac{\partial R_i(X_i)}{\partial X_i}$ in (26)

$$\frac{\partial X_i}{\partial x_0} \approx M_k^{-1} \frac{\partial \psi_i}{\partial x_0}, \tag{27a}$$

$$\frac{\partial X_i}{\partial u} \approx M_k^{-1} \left( \frac{\partial \psi_i}{\partial u} + h\gamma \frac{\partial f(X_i, u)}{\partial u} \right). \tag{27b}$$

Identically to the iterated approach, we obtained $\frac{\partial x_{k+1}}{\partial x_0}$ and $\frac{\partial x_{k+1}}{\partial u}$ as (18) and (24), respectively, and we apply the same initial values as in (25). We refer to (27) as the *direct* IND approach for the ESDIRK methods (2).

### 3.3. Comments on iteration matrix refactorization strategies

Some ODE/DAE solvers improve computational efficiency by reusing iteration matrix factorizations through multiple consecutive integration steps when solving (1). Refactorization strategies similar to the strategies in Gustafsson and Söderlind (1997) have successfully been implemented for ESDIRK methods, as demonstrated in Völcker et al. (2010). However, integrators that apply such strategies in gradient-based optimization may experience convergence issues, especially those using the direct IND method. This is because factorizations are not only used for solving nonlinear equations but also for directly obtaining the sensitivities. In contrast, as suggested in Albersmeyer (2010), we can equip integrators using the iterated approach with such refactorization strategies. This makes the iterated IND approach an attractive choice for IRK methods, where these strategies significantly enhance efficiency. However, it is important to note that the iterated method still requires frequent Jacobian updates, which can potentially eliminate advantages gained from such a strategy.

## 4. Optimal control problem

We formulate an OCP as a weighted least-squares problem that minimizes the deviation between outputs and setpoints and penalizes the input rate of movement. We formulate this OCP from $t_0$ to $t_f = t_0 + T$, where $T$ is the prediction horizon. We apply a zero-order hold parameterization of the inputs, i.e., $u(t) = u_n$ for $t_n \leq t \leq t_{n+1} = t_n + T_s$ with $n = 0, \dots, N_c - 1$, and $T_s = T/N_c$ being the interval between inputs (sampling time). We define the input rate of movement as $\Delta u_n = u_n - u_{n-1}$ and we also impose input bound constraints. This constitutes the OCP

$$\min_{x,u} \quad \phi = \phi_z + \phi_{\Delta u} \tag{28a}$$

$$s.t. \quad x(t_0) = x_0, \tag{28b}$$

$$\dot{x}(t) = f(t, x(t), u(t), d(t)), \quad t \in [t_0, t_f], \tag{28c}$$

$$z(t) = h(t, x(t), u(t), d(t)), \quad t \in [t_0, t_f], \tag{28d}$$

$$u(t) = u_n, \quad n \in \mathcal{N}, \quad t_n \leq t \leq t_{n+1}, \tag{28e}$$

$$u_{\min} \leq u(t) \leq u_{\max}, \quad t \in [t_0, t_f], \tag{28f}$$

with $\mathcal{N} = 0, \dots, N_c - 1$, and with the objectives

$$\phi_z = \frac{1}{2} \int_{t_0}^{t_f} \|z(t) - \bar{z}(t)\|_{Q_z}^2 \, dt, \tag{29a}$$

$$\phi_{\Delta u} = \frac{1}{2} \sum_{j=0}^{N_c-1} \left\| \Delta u_j \right\|_{\bar{Q}_{\Delta u}}^2. \tag{29b}$$

$\phi_z$ penalizes the difference between the output $z(t)$ and the setpoint $\bar{z}(t)$ and $\phi_{\Delta u}$ penalizes the input rate of movement. $Q_z$ and $\bar{Q}_{\Delta u} = Q_{\Delta u}/T_s$ represent the weight matrices for $\phi_z$ and $\phi_{\Delta u}$, respectively. We transcribe the OCP (28)–(29) into an NLP using direct multiple shooting. We solve an IVP between each input interval using the ESDIRK methods with $N$ steps. To solve the NLP, we apply an SQP solver with a backtracking line-search algorithm and BFGS to update the Lagrangian Hessian (Kaysfeld, 2023).

## 5. Numerical experiments

We compare the computational performance of the iterated and direct approaches in ESDIRK-based OCPs. We do this by repeatedly solving OCPs with a varying number of integration steps between

**Table 2**
Parameters for QTS for $i = 1, \ldots, 4$.

| $a_i$ [cm$^2$] | $A_i$ [cm$^2$] | $\gamma_{\text{valves}}$ [-] | $\rho$ [g/cm$^3$] | $g$ [cm/s$^2$] |
|---|---|---|---|---|
| 1.2272 | 380.1327 | [0.6; 0.7] | 1.0 | 981 |

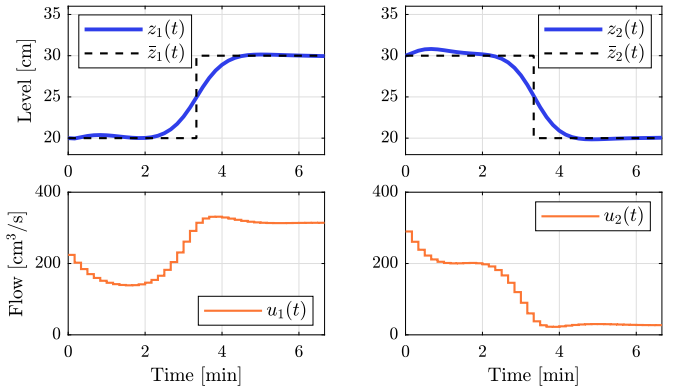


**Fig. 1.** A solution of the OCP applying ESDIRK12 with $N = 10$ and iterated IND sensitivities.

control intervals for the model of the QTS presented in Andersen et al. (2023). We compare these two approaches to a base case. This base case applies an ESDIRK method with an exact Newton method, i.e., refactorizing the iteration matrix at every Newton-type iteration. This base case applies a direct approach which due to the frequent refactorizations, achieves accurate sensitivity information. Table 2 presents the model parameters for the QTS. We use the initial conditions $x(t_0) = [7602.7\,\text{g}, 11404.0\,\text{g}, 1000.0\,\text{g}, 1000.0\,\text{g}]'$ and apply the constant vector of disturbances $d(t) = [0\,\text{cm}^3/\text{s}, 0\,\text{cm}^3/\text{s}, 100\,\text{cm}^3/\text{s}, 100\,\text{cm}^3/\text{s}]'$ for all experiments. The outputs of the QTS are the water levels in the two bottom tanks, i.e., $z(t) = \left[\frac{1}{\rho A_1} x_1(t), \quad \frac{1}{\rho A_2} x_2(t)\right]'$.

For the OCPs, we choose the sampling time $T_s = 10$ s. We use the weight matrices $Q_z = \text{diag}([10, 10])$ and $Q_{\Delta u} = \text{diag}([0.1, 0.1])$, and the number of steps in the horizon $N_c = 40$. The bounds are between 0 cm$^3$/s and 500 cm$^3$/s for both inputs. For the experiments, we construct the time-varying setpoints $\bar{z}(t) = [20\,\text{cm}, \quad 30\,\text{cm}]'$ for $0 \le t < T/2$ and $\bar{z}(t) = [30\,\text{cm}, \quad 20\,\text{cm}]'$ for $T/2 \le t \le T$. We initialize all states and inputs for the SQP as a vector with all entries set to 300. For the SQP algorithm, we choose the SQP tolerances as $10^{-3}$, the QP tolerance $10^{-8}$, the SQP step length tolerance $10^{-8}$, and abs and rel in (7) both as $10^{-8}$. The maximum number of iterations in the Newton schemes is set to 10. Fig. 1 presents one solution of the OCP using the ESDIRK12 with 10 integrator steps between control intervals and iterated IND sensitivity computations.

### 5.1. Varying integration step-size between control intervals

We solve the OCPs repeatedly while increasing the number of integration steps between control intervals. We apply the sequence $N = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$ of integration steps between controls. We also do this with the base case. To measure the performance of the three cases, we store the number of SQP iterations, the KKT violations, and the number of QP iterations in the SQP solver. We also store the total number of function and Jacobian evaluations in the ESDIRK integration scheme and LU factorizations of the iteration matrix. Figs. 2, 3, and 4 present the results using ESDIRK12, ESDIRK23, and ESDIRK34, respectively. The dotted black lines are the SQP tolerance representing the allowed KKT violations. The crosses in these figures represent experiments that do not converge. The non-convergence is due to the step length, computed using the backtracking line-search algorithm, exceeding the step tolerance. The Newton schemes in the ESDIRK methods converged in all experiments. Fig. 5 shows the KKT violations against the number of SQP iterations for all ESDIRK methods and sensitivity computations using $N = 10$.

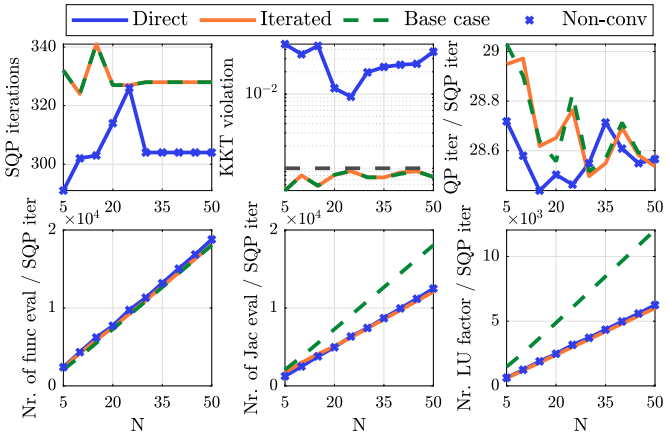


**Fig. 2.** OCP statistics using ESDIRK12.

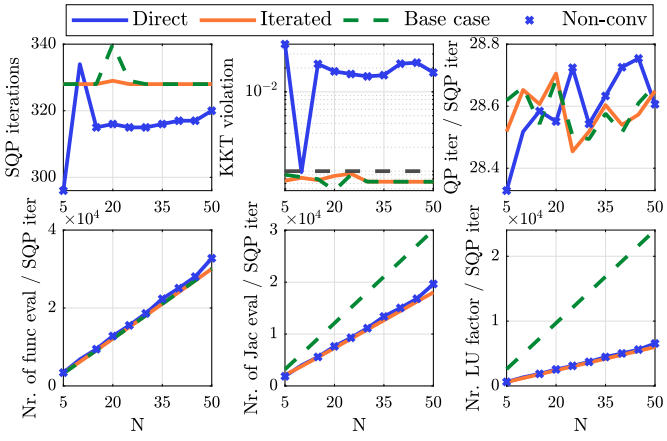


**Fig. 3.** OCP statistics using ESDIRK23.

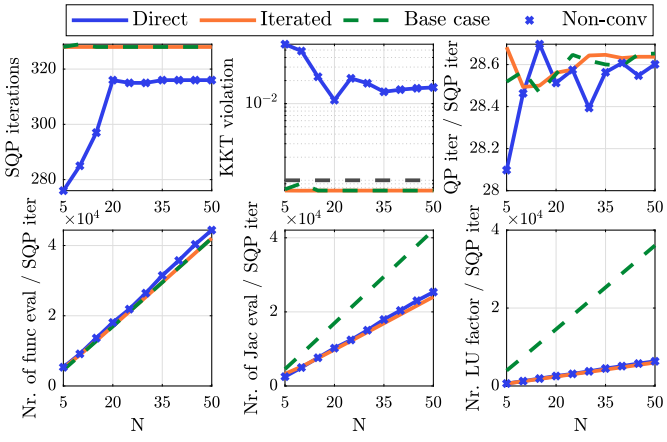


**Fig. 4.** OCP statistics using ESDIRK34.

### 5.2. Continuous-stirred tank reactor example

In addition to the QTS OCP examples, we apply the setup to the CSTR example from (Jørgensen et al., 2020; Wahlgreen et al., 2020). Fig. 6 shows experiments for the ESDIRK methods using either the iterated or direct approach. We also provide experiments using the base

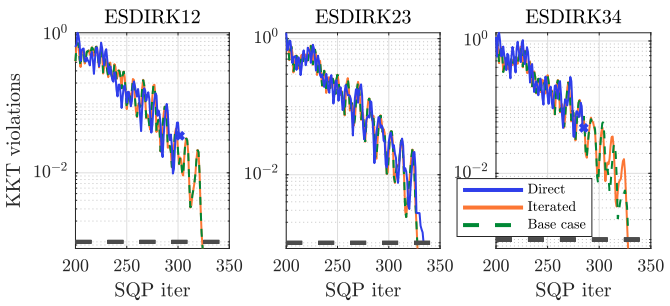


**Fig. 5.** Comparison of ESDIRK-based OCP algorithms with direct and iterated sensitivity computations (QTS). $N = 10$ for all methods.

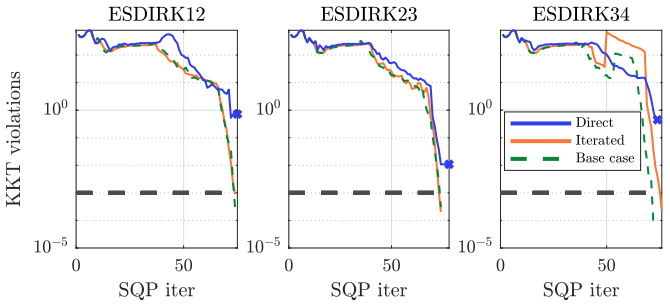


**Fig. 6.** Comparison of ESDIRK-based OCP algorithms with direct and iterated sensitivity computations (CSTR). $N = 10$ for all methods.

**Table 3**
Number of iterations in IPOPT for solving OCPs for QTS.

| ESDIRK | Iterated | Direct | Base case |
|---|---|---|---|
| 12 | 169 | 191 | 165 |
| 23 | 156 | 163R | 163 |
| 34 | 171 | 189 | 156 |

**Table 4**
Number of iterations in IPOPT for solving OCPs for CSTR.

| ESDIRK | Iterated | Direct | Base case |
|---|---|---|---|
| 12 | 28 | 24 | 24 |
| 23 | 28 | 24 | 26 |
| 34 | 26 | 23 | 27 |

case. We reuse the SQP configurations and weight matrices from the previous example for these experiments. We choose the sampling time $T_s = 1$ s, the number of steps in the horizon $N_c = 40$, and the number of integration steps $N = 10$ in each shooting interval. The OCPs track the temperature setpoint $\bar{z}(t) = 10°\text{C}$ for $0 \leq t \leq T/4$, $\bar{z}(t) = 20°\text{C}$ for $T/4 < t \leq T$. The input bounds are between 0.0 mL/min and 2000 mL/min. We initialize the SQP with 0.1 for all the decision variables.

### 5.3. Using IPOPT as NLP solver

We test the influence of the different sensitivity computations when solving the OCPs with IPOPT (Wächter & Biegler, 2006). IPOPT implements an interior point line-search filter method. We repeat the OCPs for the QTS and CSTR in Figs. 5 and 6. Tables 3 and 4 present the number of iterations for IPOPT to converge. The result marked with "R" means that the restoration phase failed and did not converge to the desired tolerance of $10^{-3}$. During the initial phase of the IPOPT iterations, both the iterated and direct methods surpass the maximum number of Newton iterations in the ESDIRK a few times, whereas the base case does not.

### 5.4. Comments on the numerical experiments

Figs. 2–4 demonstrate that ESDIRK methods using the iterated sensitivity approach converge to the desired SQP tolerance of $10^{-3}$ in all experiments. The direct approach only achieves convergence for ESDIRK23 when using 10 integration steps between control intervals. Additionally, the iterated approach performs similarly to the base case while requiring fewer total Jacobian updates and LU factorizations. The base case requires more LU factorizations compared to the iterated approach as it refactorizes the iteration matrix at every Newton iteration. Both the base case and the iterated approach update the Jacobians once per Newton iteration, but the base case requires one additional Jacobian update before every Newton scheme. It requires this, as it computes the Jacobian for the explicit state and also for constructing the iteration matrix based on the SVPs before every Newton scheme. In contrast, the iterated approach uses the same Jacobians for both the explicit step and the iteration matrix.

Fig. 5 illustrates that the methods perform similarly in the initial phases of the SQP iterations, but ESDIRK12 and ESDIRK34 using the direct method do not converge to the desired tolerance. We assume that the direct method fails to converge due to insufficient gradient information at SQP iterations near convergence. We assume this because all sensitivity methods in SQP result in full step lengths in the initial phases, while the line-search algorithm drastically reduces this length near the solution when using the direct method. This reduction in step length suggests a lack of accurate gradient information, which hinders the SQP method's ability to navigate to the optimal solution. Fig. 6 shows a similar behavior of the different sensitivity methods when using the CSTR. The results in Tables 3 and 4 show that when using IPOPT instead of the SQP solver, we see that 5 out of 6 experiments using the direct method converge, while the iterated method and the base case succeed in all experiments. The results for the QTS show that the number of iterations required for IPOPT to converge using the direct method is, on average, higher compared to the iterated method and the base case. However, for the CSTR, we observe the opposite trend, suggesting that a more sophisticated optimization algorithm better handles poor gradient information than a standard SQP algorithm.

## 6. Conclusions

In this paper, we compare the iterated and direct approaches to sensitivity computation for ESDIRK-based OCPs discretized using direct multiple shooting. The iterated approach strictly applies the principle of internal numerical differentiation, i.e., reusing the iteration matrix factorizations, the number of Newton-type iterations, and Newton iterates, to compute the sensitivities. The direct method approximates sensitivities without employing the Newton-type schemes from the ESDIRK methods. We evaluate both methods in terms of some computational performance metrics by repeatedly solving ESDIRK-based OCPs for the QTS with a varying number of integration steps between control intervals. The computational performance metrics are the number of SQP and QPs iterations, KKT violations, the total number of function evaluations, Jacobian updates, and iteration matrix factorizations. We compare these approaches to an ESDIRK method that applies an exact Newton scheme and employs a direct sensitivity computation approach. We refer to this as the base case and the study shows that the iterated method performs similarly to the base case but requires fewer Jacobian updates and matrix factorizations. On the other hand, the direct method converges to the desired tolerance only once. We see a similar behavior when using the CSTR example. Here, only the iterated method and the base case converge to the desired tolerance. When applying IPOPT instead of the SQP algorithm for the QTS, the direct method converges in 2 out of 3 experiments, while both the iterated method and the base case converge in all scenarios. In addition, the direct method requires more iterations to converge compared to the iterated method and the base case. However, for the CSTR, the direct method converges in all situations, requiring fewer iterations than the iterated method and the base case.

## CRediT authorship contribution statement

**Anders Hilmar Damm Christensen:** Conceptualization, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **John Bagterp Jørgensen:** Conceptualization, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix. Svp coefficients for esdirk34

This section presents the SVP coefficients of ESDIRK34 in (11). The coefficients for $\alpha^{34}(r)$ are

$$\alpha_1 = \frac{-r(1-c_2-c_3+p_1)}{c_3}, \tag{A.1a}$$

$$\alpha_2 = \frac{-r(1-c_2-c_3+p_2)}{c_2}, \tag{A.1b}$$

$$\alpha_3 = \frac{-r(1-c_2-c_3-p_3)}{c_2c_3}, \tag{A.1c}$$

where

$$p_1 = c_2c_3 + r(2c_2 - c_2^2 - c_2c_3 + c_2^2 r), \tag{A.2a}$$

$$p_2 = c_2c_3 + r(2c_3 - c_3^2 - c_2c_3 + c_3^2 r), \tag{A.2b}$$

$$p_3 = c_2c_3 + r(2 + c_2 - c_3 + r). \tag{A.2c}$$

The coefficients for $\beta^{34}(r)$ are

$$\beta_{11} = \frac{r(1-c_3+2c_2r+c_2^2r^2-c_2c_3r)}{(c_2-c_3)(c_2-1)}, \tag{A.3a}$$

$$\beta_{12} = \frac{-r(c_2^3r^2-c_2^3r+2c_2^2r-c_2^2+c_2)}{c_3(c_2-c_3)(c_3-1)}, \tag{A.3b}$$

$$\beta_{13} = \frac{c_2^3r^2-c_2^3r^3+c_2^2c_3r^2+2c_2c_3r-c_2^2c_3r+q_1}{(c_2+c_3-c_2c_3-1)}, \tag{A.3c}$$

$$\beta_{21} = \frac{r(c_3^3r^2-c_3^3r+2c_3^2r-c_3^2+c_3)}{c_2(c_2-c_3)(c_2-1)}, \tag{A.3d}$$

$$\beta_{22} = \frac{-r(1-c_2+2c_3r+c_3^2r^2-c_2c_3r)}{(c_2-c_3)(c_3-1)}, \tag{A.3e}$$

$$\beta_{23} = \frac{c_3^3r^2-c_3^3r^3+c_2c_3^2r^2+2c_2c_3r-c_2c_3^2r+q_2}{(c_2+c_3-c_2c_3-1)}, \tag{A.3f}$$

$$\beta_{31} = \frac{r(1-c_3-c_3r+2r+r^2)}{c_2(c_2-c_3)(c_2-1)}, \tag{A.3g}$$

$$\beta_{32} = \frac{-r(1-c_2-c_2r+2r+r^2)}{c_3(c_2-c_3)(c_3-1)}, \tag{A.3h}$$

$$\beta_{33} = \frac{c_3r^2-3r^2-r^3-c_2c_3r+q_3}{(c_2+c_3-c_2c_3-1)}, \tag{A.3i}$$

where

$$q_1 = c_2+c_3-c_2c_3-r(3c_2+2c_2^2-3c_2^2r)-1, \tag{A.4a}$$

$$q_2 = c_2+c_3-c_2c_3-r(3c_3+2c_3^2-3c_3^2r)-1, \tag{A.4b}$$

$$q_3 = c_2+c_3-c_2c_3-r(3+2c_2+2c_3+c_2r)-1. \tag{A.4c}$$

## References


Albersmeyer, J. (2010). *Adjoint-based algorithms and numerical methods for sensitivity generation and optimization of large scale dynamic systems* (Ph.D. thesis), Ruprecht-Karls-Universität Heidelberg.

Andersen, A. H. D., Ritschel, T. K. S., Hørsholt, S., Huusom, J. K., & Jørgensen, J. B. (2023). Model-based control algorithms for the quadruple tank system: An experimental comparison. In *Foundations of computer aided process operations / chemical process control*. FOCAPO / CPC, San Antonio Hill Country, San Antonio, Texas.

Bock, H. G. (1981). Numerical treatment of inverse problems in chemical reaction kinetics. In K. H. Ebert, P. Deuflhard, & W. Jäger (Eds.), *Springer series in chemical physics*: *vol. 18*, *Modelling of chemical reaction systems* (pp. 102–125). Springer.

Capolei, A., & Jørgensen, J. B. (2012). Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. In *2012 American control conference* (pp. 295–300). Fairmont Queen Elizabeth, Montréal, Canada: IEEE.

Capolei, A., Völcker, C., Frydendall, J., & Jørgensen, J. B. (2012). Oil reservoir production optimization using single shooting and ESDIRK methods. *IFAC Proceedings Volumes*, *45*(8), 286–291.

Feehery, W. F., Tolsma, J. E., & Barton, P. I. (1997). Efficient sensitivity analysis of large-scale differential-algebraic systems. *Applied Numerical Mathematics*, *25*, 41–54.

Gustafsson, K., & Söderlind, G. (1997). Control strategies for the iterative solution of nonlinear equations in ODE solvers. *SIAM Journal on Scientific Computing*, *18*(1), 23–40.

Higueras, I., & Roldán, T. (2000). Starting algorithms for some DIRK methods. *Numerical Algorithms*, *23*, 357–369.

Higueras, I., & Roldán, T. (2005). Starting algorithms for a class of RK methods for index-2 DAEs. *Computers & Mathematics with Applications*, *49*(7–8), 1081–1099.

Jørgensen, J. B., Kristensen, M. R., & Thomsen, P. G. (2018). A family of ESDIRK integration methods. arXiv preprint. arXiv:1803.01613.

Jørgensen, J. B., Kristensen, M. R., Thomsen, P. G., & Madsen, H. (2007). New extended Kalman filter algorithms for stochastic differential algebraic equations. In R. Findeisen, F. Allgöwer, & L. T. Biegler (Eds.), *Lecture notes in control and information sciences*: *vol. 358*, *Assessment and future directions of nonlinear model predictive control* (pp. 359–366). Springer Berlin Heidelberg.

Jørgensen, J. B., Ritschel, T. K. S., Boiroux, D., Schroll-Fleischer, E., Wahlgreen, M. R., Nielsen, M. K., Wu, H., & Huusom, J. K. (2020). Simulation of NMPC for a laboratory adiabatic CSTR with an exothermic reaction. In *European control conference* (pp. 202–207). Saint Petersburg, Russia.

Jørgensen, J. B., Thomsen, P. G., Madsen, H., & Kristensen, M. R. (2007). A computationally efficient and robust implementation of the continuous-discrete extended Kalman filter. In *American control conference* (pp. 3706–3712).

Kaysfeld, M. W. (2023). *Economic nonlinear model predictive control for integrated and optimized non-stationary operation of biotechnological processes* (Ph.D. thesis), 2800 Kongens Lyngby, Denmark: Technical University of Denmark.

Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., & Jørgensen, S. B. (2004). An ESDIRK method with sensitivity analysis capabilities. *Computers & Chemical Engineering*, *28*(12), 2695–2707.

Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., Michelsen, M. L., & rgensen, S. B. J. (2005). Sensitivity analysis in index-1 differential algebraic equations by ESDIRK methods. In *IFAC proceedings volumes IFAC Proceedings Volumes*, *38*(1), 212–217.

Kværnø, A. (2004). Singly diagonally implicit Runge–Kutta methods with an explicit first stage. *BIT Numerical Mathematics*, *44*, 489–502.

Quirynen, R., Vukov, M., & Diehl, M. (2012). Auto generation of implicit integrators for embedded NMPC with microsecond sampling times. In *4th IFAC nonlinear model predictive control conference IFAC Proceedings Volumes*, *45*(17), 175–180.

Völcker, C., Jørgensen, J. B., Thomsen, P. G., & Stenby, E. H. (2010). Adaptive stepsize control in implicit Runge-Kutta methods for reservoir simulation. In *9th international symposium on dynamics and control of process systems IFAC Proceedings Volumes*, *43*(5), 523–528.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*, 25–57.

Wahlgreen, M. R., Schroll-Fleischer, E., Boiroux, D., Ritschel, T. K. S., Wu, H., Huusom, J. K., & Jørgensen, J. B. (2020). Nonlinear model predictive control for an exothermic reaction in an adiabatic CSTR. *IFAC-PapersOnLine*, *53*(1), 500–505.