

A Comparative Study of Sensitivity Computations in ESDIRK-Based Optimal Control Problems

Anders Hilmar Damm Christensen and John Bagterp Jørgensen

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 use the principle of internal numerical differentiation (IND) strictly for the iterated approach, i.e., reusing the 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 the iterated and direct sensitivity computations in OCPs for the quadruple tank system. We benchmark the iterated and direct approaches with a base case. This base case is an OCP that applies an ESDIRK method with an exact Newton scheme and uses a direct approach for sensitivity computations. In these OCPs, we vary the number of integration steps between control intervals and we evaluate the performance based on the number of SQP and QPs iterations, KKT violations, and the total number of function evaluations, Jacobian updates, and iteration matrix factorizations. The results indicate that the iterated approach outperforms the direct approach but yields similar performance to the base case.

I. 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) that are discretized using a multiple shooting approach belong to this category. They require the numerical solution of the system dynamics and the corresponding integrator sensitivities. 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 [1].

For systems described by stiff ODEs or differential-algebraic equations, a multiple shooting-based OCP algorithm 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. ESDIRK-based multiple shooting OCPs have been

applied to the continuous stirred tank reactor (CSTR) and the quadruple tank system (QTS) in [2].

There exist numerous approaches for sensitivity computations such as the *Staggered direct method*, the *Simultaneous corrector method*, and the *Staggered corrector method* [1]. 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 iteration matrix factorization to 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 [3].

The principle of *internal numerical differentiation* (IND) provides a methodology for developing sensitivities that are closely related to the integrator code [4], [5]. 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. Such sensitivity computations are applied to backward differentiation formula (BDF) methods and [5] refers to it as the *iterated* IND. This 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. This is referred to as the *direct* IND approach in [5] and is identical to the Staggered direct method.

Different sensitivity computations for ESDIRK methods have already been shown, e.g., [1] 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 [2]. 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

A. H. D. Christensen and J. B. Jørgensen are with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark. Corresponding author: J. B. Jørgensen (E-mail: jbj@dtu.dk). jbj@dtu.dk

overview of the iterated and direct approaches is shown in [5] and [6] 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.

In this paper, we compare the computational performance of using either an iterated or a direct approach for sensitivity computations in ESDIRK-based multiple shooting OCPs. We do this using a basic implementation of a sequential quadratic program (SQP) solver and compare the performances in terms of the number of SQP and QP 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. Additionally, 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. The study shows that the iterated method is similar to the base case in terms of computational performance while avoiding some Jacobian updates and refactorizations in the integrator. We also show that the direct approach may only achieve convergence in the optimizer if we perform many integration steps for a small sampling time of the controller.

The rest of the paper is organized as follows. Section II introduces the ESDIRK integration method for ODEs. Section III shows how we compute sensitivities for the ESDIRK methods using the iterated and direct IND approaches. In Section IV we present the optimal control problem. Section V 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. Finally, Section VI presents conclusions.

II. THE ESDIRK METHODS FOR ODES

We consider the initial value problem (IVP)

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

$$x(t_0) = x_0, \quad (1b)$$

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

$$T_i = t_k + c_i h, \quad (2a)$$

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

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

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

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

TABLE I
BUTCHER TABLEAU'S FOR SOME ESDIRK METHODS.

ESDIRK12		ESDIRK23				ESDIRK34			
0	0	0	0	0	0	0	0		
1	$b_1 \gamma$	c_2	$a_{21} \gamma$	c_3	$a_{31} \gamma$	a_{31}	a_{32}	γ	
x_{k+1}	$b_1 \gamma$	1	$b_1 \gamma$	1	$b_1 \gamma$	1	b_1	b_2	$b_3 \gamma$
\hat{x}_{k+1}	$\hat{b}_1 \hat{b}_2$	x_{k+1}	$b_1 \gamma$	\hat{x}_{k+1}	$b_1 \gamma$	x_{k+1}	b_1	b_2	$b_3 \gamma$
			$b_1 \gamma$		$b_2 \hat{b}_3$		\hat{b}_1	\hat{b}_2	$\hat{b}_3 \hat{b}_4$

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 I presents the Butcher tableaus for ESDIRK12, ESDIRK23, and ESDIRK34. We apply the numerical values for these tableaus from [7]. 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

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

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

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

We do this by applying the inexact Newton method

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

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

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

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

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

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

with abs and rel being absolute and relative tolerances, respectively, and $\tau = 0.1$, for $i = 2, \dots, s$ [2]. As the ESDIRK methods are stiffly accurate, i.e., $a_{s,j} = b_j$, we avoid the computations in (2c) and obtain the next step directly as

$$x_{k+1} = X_s^{[w_s, k-1]}. \quad (8)$$

A. Computation of initial guesses for Newton iterations

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

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

$\alpha(r) \in \mathbb{R}^{s-1}$ 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 [8]. The SVPs for ESDIRK12, (α^{12} , β^{12}), and for ESDIRK23, (α^{23} , β^{23}) are

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

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

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

We don't show the SVPs for ESDIRK34 due to the lack of space. 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 $\hat{X}_s^{[w_s, k-1]} = x_k$.

III. SENSITIVITY ANALYSIS

A. 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

$$X_i^{[0]} = \phi_i^{\text{SVP}}(x_{k-1}, \hat{X}), \quad (11a)$$

$$\begin{aligned} X_i^{[l+1]} &= \phi_i^{\text{it}}(X_i^{[l]}, \psi_i, u) \\ &= X_i^{[l]} - M_k^{-1} R_i(X_i^{[l]}, \psi_i, u), \end{aligned} \quad (11b)$$

$$x_{k+1} = \phi^s(X_s^{[w_s, k-1]}) = X_s^{[w_s, k-1]}, \quad (11c)$$

where (11a) represent the SVPs for stage i in (9) with $\hat{X} = [\hat{X}_2^{[w_2, k-1]}, \dots, x_k]$, (11b) represent the Newton-type iterations in (5), and (11c) represent the mapping of the final stage to the next step in (8).

1) *State sensitivity*: We construct the state sensitivities by computing the partial derivatives of (11) with respect to the

initial state, x_0 , i.e.,

$$\frac{\partial X_i^{[0]}}{\partial x_0} = \frac{\partial \phi_i^{\text{SVP}}(x_{k-1}, \hat{X})}{\partial x_0}, \quad (12a)$$

$$\frac{\partial X_i^{[l+1]}}{\partial x_0} = \frac{\partial \phi_i^{\text{it}}(X_i^{[l]}, \psi_i, u)}{\partial x_0}, \quad (12b)$$

$$\frac{\partial x_{k+1}}{\partial x_0} = \frac{\partial \phi^s(X_s^{[w_s, k-1]})}{\partial x_0}, \quad (12c)$$

for $i = 2, \dots, s$, with (12b) being

$$\frac{\partial X_i^{[l+1]}}{\partial x_0} = \frac{\partial X_i^{[l]}}{\partial x_0} - M_k^{-1} \frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial x_0}. \quad (13)$$

We compute the partial derivative of the residual function as

$$\frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial x_0} = \frac{\partial R_i}{\partial X_i^{[l]}} \frac{\partial X_i^{[l]}}{\partial x_0} + \frac{\partial R_i}{\partial \psi_i} \frac{\partial \psi_i}{\partial x_0}, \quad (14)$$

with

$$\frac{\partial R_i}{\partial X_i^{[l]}} = \left(I - h\gamma \frac{\partial f(X_i^{[l]}, u)}{\partial X_i^{[l]}} \right), \quad \frac{\partial R_i}{\partial \psi_i} = -I, \quad (15a)$$

$$\frac{\partial \psi_i}{\partial x_0} = \frac{\partial x_k}{\partial x_0} + h \sum_{j=1}^{i-1} a_{ij} \frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial x_0}, \quad (15b)$$

where $X_j = X_j^{[w_j, k-1]}$ for $j = 1, \dots, i-1$ in (15b) are the stages converged after $w_{j,k}$ -number of Newton-type iterations at integrator iteration k . For (12a), we compute the state sensitivities of the SVPs as

$$\frac{\partial X_i^{[0]}}{\partial x_0} = \begin{cases} \alpha_i \frac{\partial x_{k-1}}{\partial x_0} + \sum_{j=2}^s \beta_{ij} \frac{\partial \hat{X}_j}{\partial x_0}, & \text{if } k \geq 1 \\ \frac{\partial x_k}{\partial x_0}, & \text{if } k = 0 \end{cases}, \quad (16)$$

for $i = 2, \dots, s$ and for (12c) we write

$$\frac{\partial x_{k+1}}{\partial x_0} = \frac{\partial X_s^{[w_s, k-1]}}{\partial x_0}. \quad (17)$$

2) *Input sensitivity*: We construct sensitivities with respect to the input as

$$\frac{\partial X_i^{[0]}}{\partial u} = \frac{\partial \phi_i^{\text{SVP}}(x_{k-1}, \hat{X})}{\partial u}, \quad (18a)$$

$$\frac{\partial X_i^{[l+1]}}{\partial u} = \frac{\partial \phi_i^{\text{it}}(X_i^{[l]}, \psi_i, u)}{\partial u}, \quad (18b)$$

$$\frac{\partial x_{k+1}}{\partial u} = \frac{\partial \phi^s(X_s^{[w_s, k-1]})}{\partial u}, \quad (18c)$$

for $i = 2, \dots, s$. We compute (18b) as

$$\frac{\partial X_i^{[l+1]}}{\partial u} = \frac{\partial X_i^{[l]}}{\partial u} - M_k^{-1} \frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial u}, \quad (19)$$

where the partial derivative of the residual function is

$$\frac{\partial R_i(X_i^{[l]}, \psi_i, u)}{\partial u} = \frac{\partial R_i}{\partial X_i^{[l]}} \frac{\partial X_i^{[l]}}{\partial u} + \frac{\partial R_i}{\partial \psi_i} \frac{\partial \psi_i}{\partial u} + \frac{\partial R_i}{\partial u}. \quad (20)$$

$\frac{\partial R_i}{\partial X_i^{[l]}}$ and $\frac{\partial R_i}{\partial \psi_i}$ are the same as in (15a) and we calculate $\frac{\partial \psi_i}{\partial u}$ and $\frac{\partial R_i}{\partial u}$ as

$$\frac{\partial \psi_i}{\partial u} = \frac{\partial x_k}{\partial u} + h \sum_{j=1}^{i-1} \left(a_{ij} \frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial u} + \frac{\partial f(X_j, u)}{\partial u} \right), \quad (21a)$$

$$\frac{\partial R_i}{\partial u} = -h\gamma \frac{\partial f(X_i^{[l]}, u)}{\partial u}. \quad (21b)$$

The contribution to the sensitivities for the SVPs in (18a) is

$$\frac{\partial X_i^{[0]}}{\partial u} = \begin{cases} \alpha_i \frac{\partial x_{k-1}}{\partial u} + \sum_{j=2}^s \beta_{ij} \frac{\partial \hat{X}_j}{\partial u}, & \text{if } k \geq 1 \\ \frac{\partial x_k}{\partial u}, & \text{if } k = 0 \end{cases}, \quad (22)$$

for $i = 2, \dots, s$. For (18c) the input sensitivity at the next iteration is directly obtained as

$$\frac{\partial x_{k+1}}{\partial u} = \frac{\partial X_s^{[w_s, k-1]}}{\partial u}. \quad (23)$$

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

$$\frac{\partial x_k}{\partial x_0} = I, \quad \frac{\partial x_k}{\partial u} = 0. \quad (24)$$

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 (11) are included when computing the sensitivities.

B. Direct IND

In the direct IND approach, we assume 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

$$\begin{aligned} \frac{\partial X_i}{\partial x_0} &= \frac{\partial x_k}{\partial x_0} + h \sum_{j=1}^i a_{ij} \frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial x_0} \\ &= J_i^{-1} \frac{\partial \psi_i}{\partial x_0}, \end{aligned} \quad (25a)$$

$$\begin{aligned} \frac{\partial X_i}{\partial u} &= \frac{\partial x_k}{\partial u} + h \sum_{j=1}^i a_{ij} \left(\frac{\partial f(X_j, u)}{\partial X_j} \frac{\partial X_j}{\partial u} + \frac{\partial f(X_j, u)}{\partial u} \right) \\ &= J_i^{-1} \left(\frac{\partial \psi_i}{\partial u} + h\gamma \frac{\partial f(X_i, u)}{\partial u} \right), \end{aligned} \quad (25b)$$

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 (15b) and (21a), respectively. We apply the approximation in (6) to reuse the factorizations of $\frac{\partial R_i(X_i)}{\partial X_i}$ in (25)

$$\frac{\partial X_i}{\partial x_0} \approx M_k^{-1} \frac{\partial \psi_i}{\partial x_0}, \quad (26a)$$

$$\frac{\partial X_i}{\partial u} \approx M_k^{-1} \left(\frac{\partial \psi_i}{\partial u} + h\gamma \frac{\partial f(X_i, u)}{\partial u} \right). \quad (26b)$$

Identically to the iterated approach, we obtained $\frac{\partial x_{k+1}}{\partial x_0}$ and $\frac{\partial x_{k+1}}{\partial u}$ as (17) and (23), respectively, and we apply the same initial values as in (24). We refer to (26) as the *direct* IND approach for the ESDIRK methods (2).

C. 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 [9] have successfully been implemented for ESDIRK methods, as demonstrated in [10]. 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 [5], 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.

IV. 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

$$\min_{x, u} \quad \phi = \phi_z + \phi_{\Delta u}, \quad (27a)$$

$$\text{s.t.} \quad x(t_0) = x_0, \quad (27b)$$

$$\dot{x}(t) = f(t, x(t), u(t), d(t)), \quad t \in [t_0, t_f], \quad (27c)$$

$$z(t) = h(t, x(t), u(t), d(t)), \quad t \in [t_0, t_f], \quad (27d)$$

$$u(t) = u_n, \quad t_n \leq t \leq t_{n+1}, \quad n \in \mathcal{N}, \quad (27e)$$

$$u_{\min} \leq u(t) \leq u_{\max}, \quad t \in [t_0, t_f], \quad (27f)$$

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

$$\phi_z = \frac{1}{2} \int_{t_0}^{t_f} \|z(t) - \bar{z}(t)\|_{Q_z}^2 dt, \quad (28a)$$

$$\phi_{\Delta u} = \frac{1}{2} \sum_{j=0}^{N_c-1} \|\Delta u_j\|_{Q_{\Delta u}}^2. \quad (28b)$$

ϕ_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 $Q_{\Delta u} = Q_{\Delta u}/T_s$ represent the weight matrices for ϕ_z and $\phi_{\Delta u}$, respectively. We transcribe the OCP (27)-(28) into a nonlinear programming (NLP) problem using a multiple shooting approach. 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, developed in [11].

TABLE II
PARAMETERS FOR QTS FOR $i = 1, \dots, 4$.

a_i [cm ²]	A_i [cm ²]	γ_{valves} [-]	ρ [g/cm ³]	g [cm/s ²]
1.2272	380.1327	[0.6; 0.7]	1.0	981

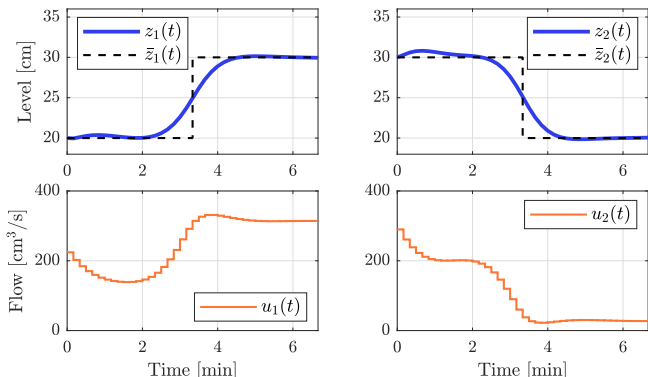


Fig. 1. A solution of the OCP applying ESDIRK12 with $N = 10$ and iterated IND sensitivities.

V. NUMERICAL EXPERIMENTS

We compare the computational performance of the iterated and direct approaches in ESDIRK-based OCP. We do this by repeatedly solving OCPs with a varying number of integration steps between control intervals for the model of the QTS presented in [12]. 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 II presents the model parameters for the QTS. We use the initial conditions $x(t_0) = [7602.7, 11404.0, 1000.0, 1000.0]'$ g and apply the constant vector of disturbances $d(t) = [0, 0, 100, 100]'$ cm³/s for all experiments. The outputs of the QTS are the water levels in the two bottom tanks, i.e., $z(t) = [\frac{1}{\rho A_1} x_1(t), \frac{1}{\rho A_2} x_2(t)]'$.

For the OCP, 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 cm³/s and 500 cm³/s for both inputs. For the experiments, we construct the time-varying setpoints $\bar{z}(t) = [20, 30]'$ cm for $0 \leq t < T/2$ and $\bar{z}(t) = [30, 20]'$ cm for $T/2 \leq t \leq T$. 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} . Fig. 1 presents one solution of the OCP using the ESDIRK12 with 10 integrator steps between control intervals and iterated IND sensitivity computations.

A. 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

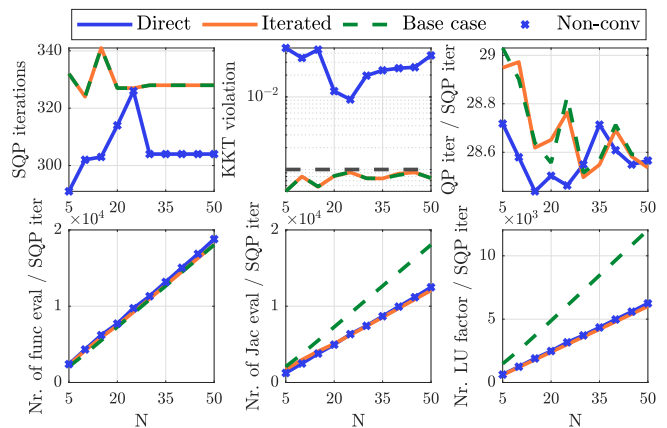


Fig. 2. OCP statistics using ESDIRK12.

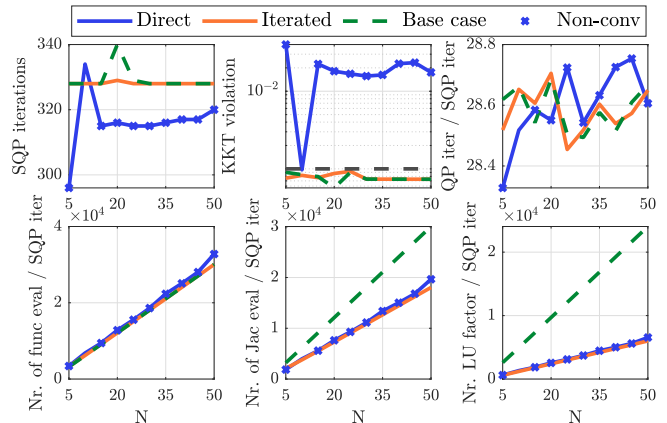


Fig. 3. OCP statistics using ESDIRK23.

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. Fig. 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.

B. Low SQP tolerance and short control intervals

We also demonstrate the performance when we lower the tolerances for the SQP to 10^{-6} , the QP to 10^{-10} , and abs and rel to 10^{-10} . We also choose a smaller control interval at $T_s = 2.0$ s. We keep the same step tolerance. Fig. 5 shows the results. Crosses represent experiments that do not converge to the SQP tolerance of 10^{-6} .

C. Comments on the numerical experiments

Fig. 2-4 demonstrate that ESDIRK methods using the iterated sensitivity approach converge to the desired SQP

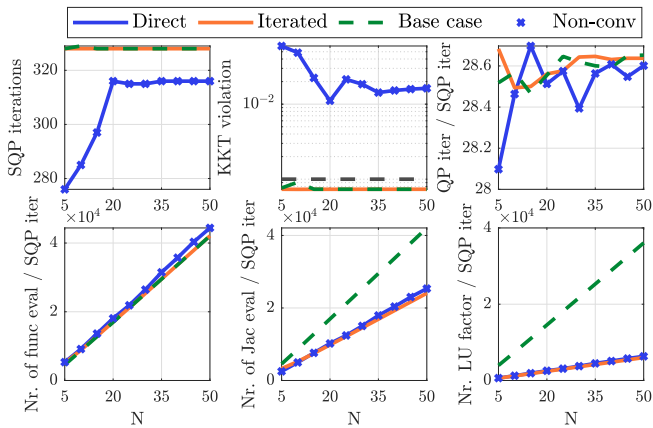


Fig. 4. OCP statistics using ESDIRK34.

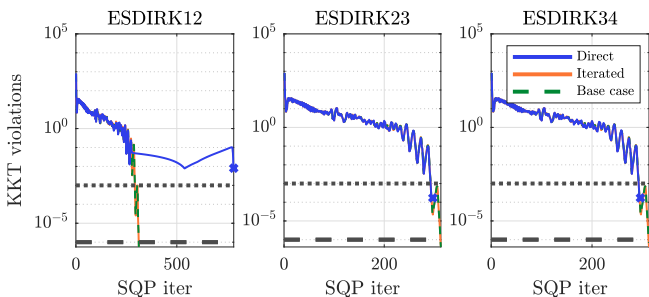


Fig. 5. Comparison of ESDIRK-based OCP algorithms with direct and iterated sensitivity computations. $N = 10$ for all methods.

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 demonstrates that the direct method can achieve convergence for ESDIRK23 and ESDIRK34 as long as we maintain an SQP tolerance of 10^{-3} and a short control interval. However, it still cannot reach the low tolerance requirement at 10^{-6} . We assume that the direct method fails to converge due to insufficient gradient information at SQP iterations close to 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 of the step length suggests a lack of accurate gradient information, which hinders the SQP method's ability to navigate to the optimal solution.

VI. CONCLUSIONS

In this paper, we compare the iterated and direct approaches to sensitivity computation for ESDIRK-based OCPs using multiple shooting discretization. The iterated approach strictly applies the principle of internal numerical differentiation, and 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 QP 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. Furthermore, the iterated approach outperforms the direct method for longer sampling times.

REFERENCES

- [1] M. R. Kristensen, J. B. Jørgensen, P. G. Thomsen, and S. B. Jørgensen, "An ESDIRK method with sensitivity analysis capabilities," *Computers & Chemical Engineering*, vol. 28, no. 12, pp. 2695–2707, 2004.
- [2] A. Capolei and J. B. Jørgensen, "Solution of Constrained Optimal Control Problems using Multiple Shooting and ESDIRK Methods," in *2012 American Control Conference*. Fairmont Queen Elizabeth, Montréal, Canada: IEEE, Jun. 2012, pp. 295–300.
- [3] W. F. Feehery, J. E. Tolsma, and P. I. Barton, "Efficient sensitivity analysis of large-scale differential-algebraic systems," *Applied Numerical Mathematics*, vol. 25, pp. 41–54, 1997.
- [4] H. G. Bock, "Numerical Treatment of Inverse Problems in Chemical Reaction Kinetics," in *Modelling of Chemical Reaction Systems*, K. H. Ebert, P. Deuffhard, and W. Jäger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 102–125.
- [5] J. Albersmeyer, "Adjoint based algorithms and numerical methods for sensitivity generation and optimal control of large scale dynamic systems," Ph.D. dissertation, Ruprecht-Karls-Universität Heidelberg, Dec. 2010.
- [6] R. Quirynen, M. Vukov, and M. Diehl, "Auto Generation of Implicit Integrators for Embedded NMPC with Microsecond Sampling," in *4th IFAC Nonlinear Model Predictive Control Conference*, Noordwijkerhout, The Netherlands, Aug. 2012, pp. 175–180.
- [7] J. B. Jørgensen, M. R. Kristensen, and P. G. Thomsen, "A Family of ESDIRK Integration Methods," *arXiv preprint arXiv:1803.01613*, 2018.
- [8] I. Higuera and T. Roldán, "Starting Algorithms for a Class of RK Methods for Index-2 DAEs," *Computers & Mathematics with Applications*, vol. 49, pp. 1081–1099, 2005.
- [9] K. Gustafsson and G. Söderlind, "Control Strategies for the Iterative Solution of Nonlinear Equations in ODE Solvers," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 23–40, 1997.
- [10] C. Völcker, J. B. Jørgensen, P. G. Thomsen, and E. H. Stenby, "Adaptive Stepsize Control in Implicit Runge-Kutta Methods for Reservoir Simulation," in *9th International Symposium on Dynamics and Control*, Leuven, Belgium, 2010, pp. 523–528.
- [11] M. W. Kaysfeld, "Economic Nonlinear Model Predictive Control for Integrated and Optimized Non-Stationary Operation of Biotechnological Processes," Ph.D. dissertation, Technical University of Denmark, 2800 Kongens Lyngby, Denmark, Jul. 2023.
- [12] A. H. D. Andersen, T. K. S. Ritschel, S. Hørsholt, J. K. Huusom, and J. B. Jørgensen, "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, Jan. 2023.