Frontiers In Global Optimization, pp. #-#
C. A. Floudas and P. M. Pardalos, Editors

A New Approach in Deterministic Global Optimisation of Problems with Ordinary Differential Equations

B. Chachuat
Laboratoire des Sciences du Génie Chimique, CNRS-ENSIC,
1 rue Grandville, B.P. 451, 54001 Nancy Cedex, France
chachuat@ensic.inpl-nancy.fr

M.A. Latifi
Laboratoire des Sciences du Génie Chimique, CNRS-ENSIC,
1 rue Grandville, B.P. 451, 54001 Nancy Cedex, France
latifi@ensic.inpl-nancy.fr

Abstract

This paper presents an alternative approach to the deterministic global optimisation of problems with ordinary differential equations in the constraints. The algorithm uses a spatial branch-and-bound approach and a novel procedure to build convex underestimation of nonconvex problems is developed. Each nonconvex functional in the original problem is underestimated by adding a separate convex quadratic term. Two approaches are presented to compute rigorous values for the weight coefficients of the quadratic terms used to relax implicitly known state-dependent functionals. The advantages of the proposed underestimation procedure are that no new decision variables nor constraints are introduced in the relaxed problem, and that functionals with state-dependent integral terms can be directly handled. The resulting global optimisation algorithm is illustrated on several case studies which consist in parameter estimation and simple optimal control problems.

Keywords: Deterministic global optimisation, convex relaxations, dynamic optimisation, nonlinear differential system.

1 Introduction

Optimisation problems with ordinary differential equations are frequently used to describe systems dynamic behaviour in many engineering fields. Typical examples which are encountered in the chemical engineering field in particular are the determination of optimal operating profiles for batch processes, fed-batch processes or periodic processes that operate at a cyclic steady-state (*e.g.*, PSA processes), or the parameter estimation of differential systems from experimental data. Since these problems generally exhibit nonconvexities, the application of classical dynamic optimisation methods may fail to determine the global optimum. This is true even for simple dynamic optimisation problems as shown in [23] for the temperature control of a batch reactor. Another illustration of these aspects can be found in [23] and [13] for a bifunctional catalyst example where hundreds of local optima have been identified from a set of random initialisations using control vector parameterisation.

The classical methods used to solve dynamic optimisation problems are based either on variational methods [30, 9] or on discretisation techniques to yield finite dimensional optimisation problems. This work is focussed on the latter class of methods. Two approaches can be further distinguished: (*i*) the complete discretisation approach, also known as the simultaneous approach, consists in discretising both state and control profiles based on either spline or orthogonal collocations (*e.g.*, [27, 38, 5]), and then solving the resulting finite dimensional nonlinear programming (NLP) problem using standard methods; (*ii*) in the control parameterisation approach, also termed the sequential approach, only the control profiles are discretised [36, 40, 41]. Since all the aforementioned methods are based on necessary optimality conditions, there is no theoretical guarantee that the global optimum is determined. This deficiency can have a marked impact on the operation of chemical processes from the economical, environmental and/or safety points of view, and motivates the development of global optimisation algorithms.

In order to overcome the convergence to local minima, stochastic search [7] as well as deterministic [20, 16] methods can be applied. The former class of methods [22, 4] consists in sampling the feasible domain to locate the global optimum. But although they improve the likelihood of finding the global optimum, no theoretical guarantee can be given in a finite number of iterations. From this point of view, deterministic methods are more interesting since they guarantee ε-convergence to the global solution in finite time. Global optimisation of dynamic problem based on the simultaneous discretisation approach has been addressed by Smith and Pantelides [35] and, more recently, by Esposito and Floudas [14] who applied the αBB approach [24, 3, 1] with applications to parameter estimation. However, such algorithms are inherently limited to moderately sized optimisation problems, and were shown to perform poorly for nonlinear systems.

On the other hand, Singer and Barton [34] have presented a rigorous global optimisation technique for problems with embedded linear dynamic systems. They exploit special structural properties of linear systems in a composition approach to build convex relaxation of the original nonconvex problem. In the case of nonlinear dynamic systems, the solution of the NLP problems resulting from the application of control parameterisation has been addressed by Esposito and Floudas [13, 14] based on the αBB algorithm. In order to formulate a convex relaxation of the dynamic problem, real valued decision variables are substituted for each state at a fixed time, and additional equality constraints are defined between the new decision variables and the states. These equality constraints are then relaxed by deriving both convex under-bounding and concave upper-bounding inequalities. This latter point is one of the main issues of the algorithm since a theoretical guarantee of attaining the global solution requires that rigorous convex/concave bounding inequalities are constructed. Papamichail and Adjiman [29] have recently proposed a rigorous method for the construction of such bounding inequalities; their approach is based on the use of differential inequalities to bound the solutions of ordinary differential equations (ODEs).

In this work, we propose an alternative method to derive convex relaxation for optimisation problems with ODEs in the constraints. Applying the basic concepts of the αBB relaxation technique, each nonconvex twice continuously differentiable functional of the original problem is relaxed by adding a suitable quadratic term. In order to ensure that the underestimator is convex, the weight coefficients α of the added quadratic terms are chosen such that the nonconvexity of the original functionals is overpowered, *i.e.*, such that the Hessians for the combined quadratic terms and nonconvex functionals are positive semi-definite. In this feature, two methods are proposed to compute suitable values of the α coefficients for the implicitly known state-dependent functionals, based on differential inequalities, namely the sensitivity and the adjoint based approaches. The convex-underestimation procedure is then used in a branch-and-bound framework. The main advantages of this procedure are (i) the ability to consider functionals involving state-dependent integral terms and, (ii) the absence of new decision variables or constraints in the relaxed problem formulation.

The remaining of this paper is organised as follows. The formulation of the optimisation problem will be discussed in section 2. The global optimisation approach is presented in section 3 with emphasis placed on the convex-underestimation procedure. An example illustrating the key points will be considered throughout this section. Finally, both theoretical and computational aspects of the proposed approach will be discussed in section 4 based on a series of test problems taken from the literature.

2 Dynamic optimisation problem formulation

The following notations are used throughout the paper. $\boldsymbol{x} \in \mathbb{R}^{n_x}$ $(n_x \geq 1)$ denotes the state vector, $\boldsymbol{p} \in \mathbb{R}^{n_p}$ $(n_p \geq 1)$ denotes the time-invariant parameter vector, and $t \in \mathbb{R}$ is the independent variable (*e.g.*, time). The mathematical formulation of the dynamic optimisation problems that are under consideration in this work is stated in Definition 1.

Definition 1 *Let $P = \left[\boldsymbol{p}^L, \boldsymbol{p}^U\right]$ be a non-empty compact convex subset of $\mathbb{R}^{n_p}$, and $X \subset \mathbb{R}^{n_x}$ such that $\boldsymbol{x}(t,\boldsymbol{p}) \in X, \forall (t,\boldsymbol{p}) \in [t_0,t_f] \times P$. The dynamic optimisation problem considered here is given by:*

$$\begin{aligned}
\min_{\boldsymbol{p}\in P} \mathcal{J}(\boldsymbol{p}) &= \mathcal{G}_0\left(\boldsymbol{x}\left(t_1^-,\boldsymbol{p}\right),\ldots,\boldsymbol{x}\left(t_{n_s}^-,\boldsymbol{p}\right),\boldsymbol{p}\right) + \sum_{k=1}^{n_s}\int_{t_{k-1}^+}^{t_k^-} \mathcal{L}^{(k)}(\boldsymbol{x},\boldsymbol{p})\,dt \\
s.t. \qquad \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{p}) \ , \ \forall t \in [t_0, t_{n_s}] \\
\boldsymbol{x}(\boldsymbol{p},t_0) &= \boldsymbol{h}(\boldsymbol{p}) \qquad\qquad (P) \\
0 &\geq \mathcal{G}_j\left(\boldsymbol{x}\left(t_1^-,\boldsymbol{p}\right),\ldots,\boldsymbol{x}\left(t_{n_s}^-,\boldsymbol{p}\right),\boldsymbol{p}\right) + \sum_{k=1}^{n_s}\int_{t_{k-1}^+}^{t_k^-} \mathcal{L}_j^{(k)}(\boldsymbol{x},\boldsymbol{p})\,dt\ , \\
& \qquad j = 1,\ldots,n_c
\end{aligned}$$

where n_c is the number of constraints; n_s is the number of time intervals; $t_0 \leq t_1 \leq \ldots \leq t_{n_s}$ are <u>fixed</u> time instants; $\mathcal{G}_j$, $\mathcal{L}_j^{(k)}$, f_i and h_i are mappings defined as $\mathcal{G}_j : X^{n_s} \times P \longmapsto \mathbb{R}$

$(j=1,\ldots,n_c)$, $\mathcal{L}_j^{(k)} : X \times P \longmapsto \mathbb{R}$ $(j=1,\ldots,n_c;\ k=1,\ldots,n_s)$, $f_i : X \times P \longmapsto \mathbb{R}$ $(i=1,\ldots,n_x)$ and $h_i : P \longmapsto \mathbb{R}$ $(i=1,\ldots,n_x)$.

The mathematical formulation given in Definition 1 allows to deal with a broad class of problems encountered in dynamic optimisation:

- Problems with time dependent ODE systems, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{p},t)$, can be handled by considering the extended state vector $\tilde{\boldsymbol{x}}^t = \left(t, \boldsymbol{x}^t\right)$, and then reformulating the dynamic system as $\dot{\tilde{\boldsymbol{x}}} = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\boldsymbol{p})$, where $\tilde{\boldsymbol{f}} : \mathbb{R}^{n_x+1} \times P \longmapsto \mathbb{R}^{n_x+1}$ is defined as $\tilde{\boldsymbol{f}}^t = \left(1, \boldsymbol{f}^t\right)$. This transformation can also be applied to deal with problems containing time dependent integrands $\mathcal{L}$ and/or non-integral costs $\mathcal{G}$,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x},\boldsymbol{p},t) &= \mathcal{L}(\tilde{\boldsymbol{x}},\boldsymbol{p}) \\
\mathcal{G}\left(\boldsymbol{x}\left(t_1^-,\boldsymbol{p}\right),\ldots,\boldsymbol{x}\left(t_{n_s}^-,\boldsymbol{p}\right),\boldsymbol{p},t_1,\ldots,t_{n_s}\right) &= \mathcal{G}\left(\tilde{\boldsymbol{x}}\left(t_1^-,\boldsymbol{p}\right),\ldots,\tilde{\boldsymbol{x}}\left(t_{n_s}^-,\boldsymbol{p}\right),\boldsymbol{p}\right)
\end{aligned}$$

- Problems with time-varying instants $t_0, t_1, \ldots, t_{n_s}$ may also be reformulated in order to fall into the scope of problem (P). This can be done, *e.g.*, by normalising time t as $\tau^{(k)} = k - 1 + \frac{t - t_{k-1}}{\Delta t_k}$, and considering the augmented parameter vector $\tilde{\boldsymbol{p}}^t = \left(\boldsymbol{p}^t, \Delta t_1, \ldots, \Delta t_{n_s}\right)$, where $\Delta t_k = t_k - t_{k-1}$. Any function $\mathcal{F}$ with varying time events is then transformed into a function with fixed (normalised) time events. For example, the function $\mathcal{F} = \sum_{k=1}^{n_s} \int_{t_{k-1}^+}^{t_k^-} \mathcal{L}^{(k)}(\boldsymbol{x},\boldsymbol{p})\, dt$ is transformed into $\mathcal{F} = \sum_{k=1}^{n_s} \int_{k-1^+}^{k^-} \Delta t_k \mathcal{L}^{(k)}(\boldsymbol{x},\boldsymbol{p})\, d\tau$.
- Last but not least, problems containing time dependent control variables $\boldsymbol{u}(t)$ can be transformed to the form given in Definition 1 by applying standard control parameterisation techniques [36], such as piecewise constant or polynomial discretisations. The control variables can then be written as functions of the parameters and time, $\boldsymbol{u}(t) = \mathcal{U}(\boldsymbol{p},t)$.

3 Global optimisation approach

For NLPs, many modern deterministic optimisation methods in Euclidean spaces are based on the concepts of branch-and-bound and convex-underestimators to solve non-convex optimisation problems to global optimality. A non-decreasing sequence of lower bounds and a non-increasing sequence of upper bounds are generated on the global solution, and finite convergence is obtained by successively subdividing the parameter space at each branch-and-bound node. While upper bounds can be easily obtained by solving the original nonconvex problem to local optimality, the computation of valid lower bounds needs to build, and then solve, a convex relaxation which underestimates the original problem. This latter point is clearly one of the key issues of such algorithms. Maranas and Floudas [24] have demonstrated that convex underestimators can be built for any twice continuously differentiable nonconvex function by applying a shift to the diagonal elements of its Hessian matrix. Additionally, several interval-based methods giving rigorous values for the shift coefficients α can be applied provided that the functions of the optimisation problem are algebraic, known explicitly as elementary functions, and can be manipulated symbolically [1].

For optimisation problems with ODEs in the constraints, these methods cannot be directly used due to the presence of implicitly known state-dependent functionals. A typical example of state-dependent functional arising in dynamic optimisation problems is:

$$
\begin{array}{rcl}
\mathcal{F} & = & \mathcal{G}\left(\boldsymbol{x}\left(t_1^-,\boldsymbol{p}\right),\ldots,\boldsymbol{x}\left(t_{n_s}^-,\boldsymbol{p}\right),\boldsymbol{p}\right)+\displaystyle\sum_{k=1}^{n_s}\int_{t_{k-1}^+}^{t_k^-}\mathcal{L}^{(k)}\left(\boldsymbol{x},\boldsymbol{p}\right)\,dt \\
\text{where}\quad \dot{\boldsymbol{x}}\left(t,\boldsymbol{p}\right) & = & \boldsymbol{f}\left(\boldsymbol{x}\left(t,\boldsymbol{p}\right),\boldsymbol{p}\right)\ ,\ \forall t\in\left[t_0,t_{n_s}\right] \\
\boldsymbol{x}\left(t_0,\boldsymbol{p}\right) & = & \boldsymbol{h}\left(\boldsymbol{p}\right)
\end{array}
\tag{1}
$$

We propose to build rigorous convex-underestimators for such functionals based on similar considerations to those used for twice-differentiable explicit algebraic functions. Clearly, a functional $\breve{\mathcal{F}}$ underestimating $\mathcal{F}$ for any $\boldsymbol{p}\in\left[\boldsymbol{p}^L,\boldsymbol{p}^U\right]$ can be obtained by adding a quadratic term as:

$$
\breve{\mathcal{F}}\left(\boldsymbol{p}\right) \ = \ \mathcal{F}\left(\boldsymbol{p}\right)+\sum_{i=1}^{n_p}\alpha_i\left(p_i^U-p_i\right)\left(p_i^L-p_i\right) \tag{2}
$$

where $\alpha_i\geq 0, i=1,\ldots,n_p$. The challenging question that arises is to compute values for the α_i coefficients such that $\breve{\mathcal{F}}$ is convex over the set $\left[\boldsymbol{p}^L,\boldsymbol{p}^U\right]$, *i.e.*, such that the quadratic term overpowers the nonconvexity of the original function. This issue is addressed subsequently. Two methods for computing the second-order derivatives of $\mathcal{F}$ with respect to the parameters $\boldsymbol{p}$ are first presented. Then, the concept of differential inequalities is used to generate rigorous bounds on the coefficients of the Hessian matrix of $\mathcal{F}$. At that point, rigorous values for the α coefficients can be deduced from the application of any method proposed by Adjiman *et al.* [1], *e.g.*, the scaled Gershgorin method. Finally, the convex relaxations of the original dynamic problem are used in the framework of a spatial branch-and-bound algorithm.

3.1 Second-order derivatives of state-dependent functionals

The following conditions are imposed on problem (1).

Assumption 1 *$\mathcal{G}$, $\mathcal{L}^{(k)}$ $(k=1,\ldots,n_s)$, $\boldsymbol{f}_i$ $(i=1,\ldots,n_x)$ and $\boldsymbol{h}_i$ $(i=1,\ldots,n_x)$ are twice continuously differentiable with respect to any participating variable over their sets of interest.*

A useful result on continuity and differentiability of the solution of parameter dependent ODE systems established in [30] is recalled in Theorem 1.

Theorem 1 *Let $(\mathcal{S})$ be the following parametric ODE system:*

$$
\begin{array}{rcl}
\dot{\boldsymbol{x}}\left(t\right) & = & \boldsymbol{f}\left(\boldsymbol{x}\left(t,\boldsymbol{p}\right),\boldsymbol{p},t\right)\ ,\ \forall t\in\left[t_0,t_{n_s}\right] \\
\boldsymbol{x}\left(t_0,\boldsymbol{p}\right) & = & \boldsymbol{h}\left(\boldsymbol{p}\right)
\end{array}
\tag{$\mathcal{S}$}
$$

Suppose the partial derivatives of $\boldsymbol{f}$ with respect to the variables $\boldsymbol{x}$ and parameters $\boldsymbol{p}$ exist and are continuous up to the mth-order inclusive. Also suppose that the partial derivatives of $\boldsymbol{h}$ with respect to $\boldsymbol{p}$ exist and are continuous up to the mth-order inclusive. Then the solution $x\left(t,\boldsymbol{p}\right)=\varphi\left(t,\boldsymbol{p},\boldsymbol{h}\left(\boldsymbol{p}\right)\right)$ of $(\mathcal{S})$ has continuous partial derivatives with respect to $\boldsymbol{p}$ up to the mth-order inclusive.

From Theorem 1, Assumption 1 and well-known results on the composition of differentiable mappings, it is immediate that $\mathcal{F}$ is twice continuously differentiable with respect to the parameters $\boldsymbol{p}$.

Two approaches can be used to compute the second-order derivatives, namely the sensitivity [18] and the adjoint [9, 32] methods. It should be noted that both are routinely applied to provide the first-order derivatives in optimisation problems with ODE systems in the constraints. In addition, the use of the sensitivity method to compute second-order derivatives has been investigated in [39] and is applied in [29] to derive convex underestimators for the state variables $\boldsymbol{x}(t, \boldsymbol{p})$ at fixed time instants. In this work, we also investigate the application of the adjoint approach in order to compute the second-order derivatives of state-dependent functionals.

3.1.1 Sensitivity approach

The first-order derivatives of $\mathcal{F}$ can be obtained by applying a chain differentiation rule with respect to the parameters p_i, $i = 1, \ldots, n_p$,

$$\frac{\partial \mathcal{F}}{\partial p_i} = \frac{\partial \mathcal{G}}{\partial p_i} + \sum_{k=1}^{n_s} \left(\frac{\partial \mathcal{G}}{\partial \boldsymbol{x}^t(t_k, \boldsymbol{p})} \boldsymbol{x}_{p_i}(t_k, \boldsymbol{p}) + \int_{t_{k-1}^+}^{t_k^-} \frac{\partial \mathcal{L}}{\partial p_i} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}^t} \boldsymbol{x}_{p_i}(t, \boldsymbol{p})\, dt \right)$$

where the first-order state sensitivities $\boldsymbol{x}_{p_i} = \frac{\partial \boldsymbol{x}}{\partial p_i}$ are defined by the ODE system:

$$\begin{aligned} \dot{\boldsymbol{x}}_{p_i}(t, \boldsymbol{p}) &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^t} \boldsymbol{x}_{p_i}(t, \boldsymbol{p}) + \frac{\partial \boldsymbol{f}}{\partial p_i}\ , \ \forall t \in [t_0, t_{n_s}] \\ \boldsymbol{x}_{p_i}(t_0, \boldsymbol{p}) &= \frac{\partial \boldsymbol{h}}{\partial p_i} \end{aligned}$$

The second-order derivatives of $\mathcal{F}$ are then produced by applying the chain differentiation rule once again with respect to the parameters:

$$\begin{aligned} \frac{\partial^2 \mathcal{F}}{\partial p_j \partial p_i} = & \frac{\partial^2 \mathcal{G}}{\partial p_j \partial p_i} + \sum_{k=1}^{n_s} \left[\frac{\partial^2 \mathcal{G}}{\partial p_j \partial \boldsymbol{x}^t(t_k, \boldsymbol{p})} \boldsymbol{x}_{p_i}(t_k, \boldsymbol{p}) + \boldsymbol{x}_{p_j}^t(t_k, \boldsymbol{p}) \frac{\partial^2 \mathcal{G}}{\partial \boldsymbol{x}(t_k, \boldsymbol{p}) \partial p_i} \right. \\ & + \sum_{\ell=1}^{n_s} \boldsymbol{x}_{p_j}^t(t_\ell, \boldsymbol{p}) \frac{\partial^2 \mathcal{G}}{\partial \boldsymbol{x}(t_\ell, \boldsymbol{p}) \partial \boldsymbol{x}^t(t_k, \boldsymbol{p})} \boldsymbol{x}_{p_i}(t_k, \boldsymbol{p}) + \frac{\partial \mathcal{G}}{\partial \boldsymbol{x}^t(t_k, \boldsymbol{p})} \boldsymbol{x}_{p_j p_i}(t_k, \boldsymbol{p}) \\ & + \int_{t_{k-1}^+}^{t_k^-} \left(\frac{\partial^2 \mathcal{L}}{\partial p_j \partial p_i} + \frac{\partial^2 \mathcal{L}}{\partial p_j \partial \boldsymbol{x}^t} \boldsymbol{x}_{p_i}(t, \boldsymbol{p}) + \boldsymbol{x}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{x} \partial p_i} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}^t} \boldsymbol{x}_{p_j p_i} \right. \\ & \left. \left. + \boldsymbol{x}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{x} \partial \boldsymbol{x}^t} \boldsymbol{x}_{p_i}(t, \boldsymbol{p}) \right) dt \right] \end{aligned}$$

with the second-order state sensitivities $\boldsymbol{x}_{p_j p_i} = \frac{\partial^2 \boldsymbol{x}}{\partial p_j \partial p_i}$ being the solution of the system:

$$\begin{aligned} \dot{\boldsymbol{x}}_{p_j p_i}(t, \boldsymbol{p}) = & \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^t} \boldsymbol{x}_{p_j p_i}(t, \boldsymbol{p}) + \boldsymbol{x}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}^2} \boldsymbol{x}_{p_i}(t, \boldsymbol{p}) + \frac{\partial^2 \boldsymbol{f}}{\partial p_j \partial \boldsymbol{x}^t} \boldsymbol{x}_{p_i}(t, \boldsymbol{p}) \\ & + \boldsymbol{x}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x} \partial p_i} + \frac{\partial^2 \boldsymbol{f}}{\partial p_j \partial p_i}\ , \ \forall t \in [t_0, t_{n_s}] \\ \boldsymbol{x}_{p_j p_i}(t_0, \boldsymbol{p}) = & \frac{\partial^2 \boldsymbol{h}}{\partial p_i \partial p_j} \end{aligned}$$

Illustrative example – Consider the following state-dependent functional:

$$\begin{array}{rcl} \mathcal{F}(p) & = & -\left[x(1,p)\right]^2 \\ \text{where} \quad \dot{x}(t,p) & = & -\left[x(t,p)\right]^2 + p \ , \ \forall t \in [0,1] \\ x(0,p) & = & 9 \end{array} \qquad (P_1)$$

In the sensitivity approach, the second-order derivative of $\mathcal{F}$ with respect to p are given by:

$$\frac{\partial^2 \mathcal{F}}{\partial p^2} = -2x(1,p)\, x_{p^2}(1,p) - 2\left(x_p(1,p)\right)^2$$

$$\text{with} \quad \left.\begin{array}{rcl} \dot{x}(t,p) & = & -\left[x(t,p)\right]^2 + p \\ \dot{x}_p(t,p) & = & -2x(t,p)\, x_p(t,p) + 1 \\ \dot{x}_{p^2}(t,p) & = & -2x(t,p)\, x_{p^2}(t,p) - 2\left(x_p(t,p)\right)^2 \end{array}\right\} \forall t \in [0,1]$$

$$\begin{array}{rcl} x(0,p) & = & 9 \\ x_p(0,p) & = & 0 \\ x_{p^2}(0,p) & = & 0 \end{array}$$

3.1.2 Adjoint approach

The adjoint approach may also be applied to compute the second-order derivatives of $\mathcal{F}$ with respect to the parameters. The following proposition first recalls results concerning the use of adjoint variables to obtain the first-order derivatives of a state-dependent functional.

Proposition 1 *Let $\mathcal{F}$ be a Bolza-type functional of the form (1), with $\mathcal{G} : X^{n_s} \times P \longmapsto \mathbb{R}$, $\mathcal{L}^{(k)} : X \times P \longmapsto \mathbb{R}$ ($k = 1, \ldots, n_s$), $f_i : X \times P \longmapsto \mathbb{R}$ ($i = 1, \ldots, n_x$) and $h_i : P \longmapsto \mathbb{R}$ ($i = 1, \ldots, n_x$) being continuously differentiable. Then, the first-order derivatives of $\mathcal{F}$ with respect to the parameters p_i, $i = 1, \ldots, n_p$, are given by:*

$$\frac{\partial \mathcal{F}}{\partial p_i} = \frac{\partial \mathcal{G}}{\partial p_i} + \boldsymbol{\lambda}^t(t_0, \boldsymbol{p})\frac{\partial \boldsymbol{h}}{\partial p_i} + \sum_{k=1}^{n_s}\int_{t_{k-1}^+}^{t_k^-} \frac{\partial \mathcal{H}^{(k)}}{\partial p_i}\, dt \qquad (3)$$

where the Hamiltonians $\mathcal{H}^{(k)}$, $k = 1, \ldots, n_s$, are defined as:

$$\mathcal{H}^{(k)}\left(\boldsymbol{x}(t,\boldsymbol{p}), \boldsymbol{\lambda}(t,\boldsymbol{p}), \boldsymbol{p}\right) = \mathcal{L}^{(k)}\left(\boldsymbol{x}(t,\boldsymbol{p}), \boldsymbol{p}\right) + \boldsymbol{\lambda}^t(t,\boldsymbol{p})\, \boldsymbol{f}\left(\boldsymbol{x}(t,\boldsymbol{p}), \boldsymbol{p}\right)$$

and where the adjoint (or costate) variables $\boldsymbol{\lambda} \in \mathbb{R}^{n_x}$ are solutions of ODE system (4), from terminal condition (5) and with transition conditions (6) at time instants t_k, $k = 1, \ldots, n_s - 1$.

$$\dot{\boldsymbol{\lambda}}^t(t,\boldsymbol{p}) = -\boldsymbol{\lambda}^t(t,\boldsymbol{p})\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^t} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}^t} \ , \ \forall t \in \left[t_{k-1}^+, t_k^-\right], k = 1, \ldots, n_s \qquad (4)$$

$$\boldsymbol{\lambda}^t\left(t_{n_s}^-, \boldsymbol{p}\right) = \frac{\partial \mathcal{G}}{\partial \boldsymbol{x}^t\left(t_{n_s}^-, \boldsymbol{p}\right)} \qquad (5)$$

$$\boldsymbol{\lambda}^t\left(t_k^-, \boldsymbol{p}\right) = \boldsymbol{\lambda}^t\left(t_k^+, \boldsymbol{p}\right) + \frac{\partial \mathcal{G}}{\partial \boldsymbol{x}^t\left(t_k^-, \boldsymbol{p}\right)} \qquad (6)$$

Proof. See, *e.g.*, [9, 32]. $\square$

The same approach can be used to derive the second-order derivatives of $\mathcal{F}$. However, the results given in Proposition 1 (obtained for initial value problems, IVP) are no longer applicable due to the form of the differential system – which is a two-point boundary value problem (TPBVP) in this case –. Therefore, we demonstrate the following result.

Proposition 2 *Let $\mathcal{F}$ be a Bolza-type functional of the form (1), where $\mathcal{G} : X^{n_s} \times P \longmapsto \mathbb{R}$, $\mathcal{L}^{(k)} : X \times P \longmapsto \mathbb{R}$ $(k = 1, \ldots, n_s)$, $f_i : X \times P \longmapsto \mathbb{R}$ $(i = 1, \ldots, n_x)$ and $h_i : P \longmapsto \mathbb{R}$ $(i = 1, \ldots, n_x)$ are twice continuously differentiable. Then, the second-order derivatives of $\mathcal{F}$ with respect to the parameters p_i, p_j $(i, j = 1, \ldots, n_p)$ are given by:*

$$
\begin{aligned}
\frac{\partial^2 \mathcal{F}}{\partial p_j \partial p_i} \;=\;& \frac{\partial^2 \mathcal{G}}{\partial p_j \partial p_i} + \sum_{k=1}^{n_s} \boldsymbol{x}_{p_j}^t\left(t_k^-, \boldsymbol{p}\right) \frac{\partial^2 \mathcal{G}}{\partial \boldsymbol{x}\left(t_k^-, \boldsymbol{p}\right) \partial p_i} + \boldsymbol{\lambda}^t\left(t_0, \boldsymbol{p}\right) \frac{\partial^2 \boldsymbol{h}}{\partial p_j \partial p_i} + \boldsymbol{\lambda}_{p_j}^t\left(t_0, \boldsymbol{p}\right) \frac{\partial \boldsymbol{h}}{\partial p_i} \\
&+ \sum_{k=1}^{n_s} \int_{t_{k-1}^+}^{t_k^-} \left[\frac{\partial^2 \mathcal{H}^{(k)}}{\partial p_j \partial p_i} + \boldsymbol{x}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \mathcal{H}^{(k)}}{\partial \boldsymbol{x} \partial p_i} + \boldsymbol{\lambda}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \mathcal{H}^{(k)}}{\partial \boldsymbol{\lambda} \partial p_i} \right] dt
\end{aligned}
$$

where $\boldsymbol{x}_{p_j}$ and $\boldsymbol{\lambda}_{p_j}$ denote the first-order sensitivities of the state and costate variables respectively, and are defined by the following TPBVP:

$$
\left.
\begin{aligned}
\dot{\boldsymbol{x}}_{p_j}(t, \boldsymbol{p}) &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^t} \boldsymbol{x}_{p_j}(t, \boldsymbol{p}) + \frac{\partial \boldsymbol{f}}{\partial p_j} \\
\dot{\boldsymbol{\lambda}}_{p_j}^t(t, \boldsymbol{p}) &= -\boldsymbol{\lambda}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^t} - \boldsymbol{\lambda}^t(t, \boldsymbol{p}) \left[\frac{\partial^2 \boldsymbol{f}}{\partial p_j \partial \boldsymbol{x}^t} + \boldsymbol{x}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{x}^2} \right] \\
&\quad - \left[\frac{\partial^2 \mathcal{L}}{\partial p_j \partial \boldsymbol{x}^t} + \boldsymbol{x}_{p_j}^t(t, \boldsymbol{p}) \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{x}^2} \right]
\end{aligned}
\right\} \begin{array}{l} \forall t \in \left[t_{k-1}^+, t_k^-\right], \\ k = 1, \ldots, n_s \end{array}
$$

$$
\begin{aligned}
\boldsymbol{x}_{p_j}(t_0, \boldsymbol{p}) &= \frac{\partial \boldsymbol{h}}{\partial p_j} \\
\boldsymbol{\lambda}_{p_j}^t\left(t_{n_s}^-, \boldsymbol{p}\right) &= \frac{\partial^2 \mathcal{G}}{\partial p_j \partial \boldsymbol{x}^t\left(t_{n_s}^-, \boldsymbol{p}\right)} + \sum_{\ell=1}^{n_s} \boldsymbol{x}_{p_j}^t\left(t_\ell^-, \boldsymbol{p}\right) \frac{\partial^2 \mathcal{G}}{\partial \boldsymbol{x}\left(t_\ell^-, \boldsymbol{p}\right) \partial \boldsymbol{x}^t\left(t_{n_s}^-, \boldsymbol{p}\right)}
\end{aligned}
$$

$$
\left.
\begin{aligned}
\boldsymbol{x}_{p_j}\left(t_k^+, \boldsymbol{p}\right) &= \boldsymbol{x}_{p_j}\left(t_k^-, \boldsymbol{p}\right) \\
\boldsymbol{\lambda}_{p_j}^t\left(t_k^-, \boldsymbol{p}\right) &= \boldsymbol{\lambda}_{p_j}^t\left(t_k^+, \boldsymbol{p}\right) + \frac{\partial^2 \mathcal{G}}{\partial p_j \partial \boldsymbol{x}^t\left(t_{n_s}^-, \boldsymbol{p}\right)} \\
&\quad + \sum_{\ell=1}^{n_s} \boldsymbol{x}_{p_j}^t\left(t_\ell^-, \boldsymbol{p}\right) \frac{\partial^2 \mathcal{G}}{\partial \boldsymbol{x}\left(t_\ell^-, \boldsymbol{p}\right) \partial \boldsymbol{x}^t\left(t_k^-, \boldsymbol{p}\right)}
\end{aligned}
\right\} k = 1, \ldots, n_s
$$

Proof. The proof is obtained by applying a chain differentiation rule to Eq. (3) and the associated TPBVP (4)–(6). $\square$

Remark 1 It should be noted that the application of the adjoint system approach twice to derive the second-order derivatives of $\mathcal{F}$ yields exactly the same mathematical expressions as those given in Proposition 2. In particular, it was shown that the new costate variables

$\boldsymbol{\mu}_x$, $\boldsymbol{\mu}_\lambda$ associated to $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ respectively correspond to the first-order sensitivities variables $\boldsymbol{\lambda}_{p_j}$ and $\boldsymbol{x}_{p_j}$ as:

$$\boldsymbol{\mu}_x = \boldsymbol{\lambda}_{p_j} \quad \text{and} \quad \boldsymbol{\mu}_\lambda = -\boldsymbol{x}_{p_j}$$

Further details can be found in [12].

Remark 2 The adjoint approach may also be used as an alternative to the sensitivity approach in the former work of Papamichail and Adjiman [29] to relax the equality state constraints. Indeed, bounds on the second-order state sensitivities at a fixed time instant t_k can be produced by considering $\mathcal{F}(\boldsymbol{p}) = x_i(t_k, \boldsymbol{p})$, and then integrating the state/costate TPBVP and its associated sensitivities.

Illustrative example (continued) – The example discussed in the previous paragraph is reconsidered here. From the application of the adjoint approach, the second-order derivative of functional (P_1) with respect to p, is given by:

$$\frac{\partial^2 \mathcal{F}}{\partial p^2} = \int_0^1 \lambda_p(t,p)\, dt$$

$$\text{with} \quad \left.\begin{array}{rcl} \dot{x}(t,p) &=& -\left[x(t,\boldsymbol{p})\right]^2 + p \\ \dot{\lambda}(t,p) &=& 2x(t,\boldsymbol{p})\,\lambda(t,\boldsymbol{p}) \\ \dot{x}_p(t,p) &=& -2x(t,\boldsymbol{p})\,x_p(t,\boldsymbol{p}) + 1 \\ \dot{\lambda}_p(t,p) &=& 2x(t,\boldsymbol{p})\,\lambda_p(t,\boldsymbol{p}) + 2x_p(t,\boldsymbol{p})\,\lambda(t,\boldsymbol{p}) \end{array}\right\} \forall t \in [0,1]$$

$$\begin{array}{rcl} x(0,p) &=& 9 \\ \lambda(1,p) &=& -2x(1,p) \\ x_p(0,p) &=& 0 \\ \lambda_p(1,p) &=& -2x_p(1,p) \end{array}$$

3.1.3 Comparison between the sensitivity and adjoint approaches

The sensitivity and adjoint approaches both provide accurate estimations of the second-order derivatives of $\mathcal{F}$. However, the use of one method rather than the other may be guided by several considerations. First note that the sensitivity method is easier to implement since it only consists in integrating the state/state-sensitivity system forward in time, whereas the costate system must be integrated backward in time in the adjoint approach. In addition, the computational efficiency of both methods strongly depends on the size of the optimisation problem: (i) in the sensitivity approach, the state/state-sensitivity system is a $n_x \times \left(1 + n_p + n_p^2\right)$ dimensional system and only needs to be integrated once, regardless of the number of cost/constraints in the optimisation problem; (ii) in the adjoint approach, one needs to solve a $2 \times n_x \times (1 + n_p)(n_c + 1)$ dimensional differential system, with n_c denoting the number of implicitly known state-dependent cost/constraints. Note in particular that the computation of the second-order state sensitivities grows quadratically with the number of optimised parameters in the sensitivity approach, whereas it increases linearly in the adjoint approach but also depends linearly on the number of constraints in the optimisation problem. Therefore, it is clearly advantageous to apply the former for problems with few parameters, whereas the latter might be more computationally efficient for problems with many parameters and relatively few constraints.

At last, one should keep in mind that the mathematical expressions of the second-order derivatives are to be used for the computation of rigorous enclosures of the interval Hessian matrix of $\mathcal{F}$, for $\boldsymbol{p} \in \left[\boldsymbol{p}^L, \boldsymbol{p}^U\right]$. In this context, it is obvious that the enclosures provided by both methods will be different, and the choice of one approach rather than the other will thus be dictated by the tightness of the resulting bounds. These aspects are discussed in the next subsection and illustrated in an example.

3.2 Convex underestimators for state-dependent functionals

In the case of explicitly known algebraic functions, α-based convex-underestimator can be derived for any general nonconvex twice continuously differentiable term $f(\boldsymbol{p})$, on a given domain $\left[\boldsymbol{p}^L, \boldsymbol{p}^U\right] \subset \mathbb{R}^{n_p}$ of the parameter space [24, 3]. This is done by adding a negative convex quadratic term which overpowers the nonconvexity of the original function as:

$$\breve{f}(\boldsymbol{p}) = f(\boldsymbol{p}) + \sum_{i=1}^{n_p} \alpha_i \left(p_i^U - p_i\right)\left(p_i^L - p_i\right)$$

Several methods have been developed in [1] for the computation of the positive coefficients α_i. They are based on the use of an interval matrix $[H_f]$ which encloses all the Hessian matrices of f in a given domain, *i.e.*,

$$[H_f] \supset \left\{H_f(\boldsymbol{p}) = \nabla^2 f(\boldsymbol{p}), \forall \boldsymbol{p} \in \left[\boldsymbol{p}^L, \boldsymbol{p}^U\right]\right\}$$

In particular, such interval matrices can be calculated by applying natural interval extensions to the analytical expression of each second-order derivative of f. Then, the α coefficients are deduced from $[H_f]$ by applying either a uniform or a non-uniform shift diagonal matrix. For example, the application of the scaled Gerschgorin method (non-uniform shift diagonal matrix) yields,

$$\alpha_i = \max\left\{0, -\frac{1}{2}\left(h_{ii}^L - \sum_{j \neq i} |h|_{ij}\right)\right\}$$

where $[H_f] = \left[h_{ij}^L, h_{ij}^U\right]$ and $|h|_{ij} = \max\left\{\left|h_{ij}^L\right|, \left|h_{ij}^U\right|\right\}$.

We propose to extend this approach to build convex underestimators for the state-dependent functionals encountered in dynamic optimisation problems whenever they are twice continuously differentiable. In this case however, the difficulty arises from the fact that the Hessian matrix of these functionals cannot be written as an analytical function of $\boldsymbol{p}$. Esposito and Floudas [13, 14] have proposed several methods based on sampling to build an underestimation of the space of the Hessian matrices. But although these methods may give satisfactory results, it cannot be guaranteed that the corresponding α values lead to valid convex underestimators since the parameter space is only sampled at a finite number of points. More recently, Papamichail and Adjiman [29] have used differential inequalities to derive bounds on the second-order sensitivities of any individual state variable at fixed time instants, and then formulate a convex relaxation of the original optimisation problem by introducing new variables and equality constraints.

In the scope of this work, enclosures for the Hessian matrix of any twice continuously differentiable state-dependent functional are derived by applying the same concept of differential inequalities. More precisely, the analytical expressions presented in subsection 3.1 combined with differential inequalities are used to compute bounds on the second-order derivatives of state-dependent functionals. Two approaches can thus be used to produce interval matrices enclosing the Hessian of state-dependent functionals, namely the sensitivity and the adjoint based approaches. The following results, adapted from [29], provide a practical procedure to construct bounding trajectories for ODE systems subject to mild assumptions.

Theorem 2 *Let $\mathcal{I} = [t_0, t_f]$, $\mathcal{I}_0 = (t_0, t_f]$ and $\mathcal{X}$ be defined as the set of all functions $\boldsymbol{x} : \mathcal{I} \longmapsto \mathbb{R}^{n_x}$, continuous on $\mathcal{I}$ and differentiable on $\mathcal{I}_0$. Consider an ODE system of the form ($\mathcal{S}$), where $\boldsymbol{f}$ is a continuous mapping and satisfy a uniqueness condition on $X \times \left[\boldsymbol{p}^L, \boldsymbol{p}^U\right]$. If $\boldsymbol{x}^L(t)$, $\boldsymbol{x}^U(t) \in \mathcal{X}$ satisfy the following relationships:*

$$\left.\begin{array}{l}\left.\begin{array}{rcl}\dot{x}_i^L(t) & = & \inf\limits_{\boldsymbol{p},\boldsymbol{x}} \left\{ f_i(\boldsymbol{x},\boldsymbol{p}) \ : \ \boldsymbol{p}^L \le \boldsymbol{p} \le \boldsymbol{p}^U, \boldsymbol{x}^L \le \boldsymbol{x} \le \boldsymbol{x}^U, x_i = x_i^L \right\} \\ \dot{x}_i^U(t) & = & \sup\limits_{\boldsymbol{p},\boldsymbol{x}} \left\{ f_i(\boldsymbol{x},\boldsymbol{p}) \ : \ \boldsymbol{p}^L \le \boldsymbol{p} \le \boldsymbol{p}^U, \boldsymbol{x}^L \le \boldsymbol{x} \le \boldsymbol{x}^U, x_i = x_i^U \right\}\end{array}\right\} \forall t \in \mathcal{I}_0 \\ \begin{array}{rcl}x_i^L(t_0) & = & \inf\limits_{\boldsymbol{p}} \left\{ x_i(t_0,\boldsymbol{p}) \ : \ \boldsymbol{p}^L \le \boldsymbol{p} \le \boldsymbol{p}^U \right\} \\ x_i^U(t_0) & = & \sup\limits_{\boldsymbol{p}} \left\{ x_i(t_0,\boldsymbol{p}) \ : \ \boldsymbol{p}^L \le \boldsymbol{p} \le \boldsymbol{p}^U \right\}\end{array}\end{array}\right\} i = 1, \ldots, n_x$$

then $\boldsymbol{x}^L(t)$ is a subfunction and $\boldsymbol{x}^U(t)$ is a superfunction on the solutions of ODE system ($\mathcal{S}$), i.e.,

$$\boldsymbol{x}_i^L(t) \ \le \ \boldsymbol{x}_i(t,\boldsymbol{p}) \ \le \ \boldsymbol{x}_i^U(t) \ , \ \forall \boldsymbol{p} \in \left[\boldsymbol{p}^L, \boldsymbol{p}^U\right] \ , \ \forall t \in \mathcal{I} \ , \ i = 1, \ldots, n_x$$

Remark 3 Except for quasi-monotone increasing functions $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{p})$ on $X \times \left[\boldsymbol{p}^L, \boldsymbol{p}^U\right]$, the bounds on the state variables resulting from Theorem 2 may be very loose due to the well-known wrapping effect (more details on differential inequalities can be found, *e.g.*, in [42, 26]). Many works have been undertaken in order to reduce (or even eliminate) these effects. But although several methods can provide significant improvements on the bounds of the state variable (see, *e.g.*, [19]), the additional computational efforts are hardly compatible with the primary objectives of this work.

Although the theory provides valid convex-underestimators from both sensitivity and adjoint based approaches, two issues must be examined more closely from a practical point of vue. First, it is of paramount importance that the integration itself ensures valid enclosures for the solution of the differential equations, otherwise no garantee can be given that the relaxed state-dependent functional is convex. In this work, we apply a simple, conservative, method by taking part of the local error control performed by most numerical integration softwares (*e.g.*, DASSL [8]). It was found that the resulting overestimation remains limited for small size problems as those addressed in section 4, although it is known for years that this method may lead to severe overestimation due to the aforementionned wrapping effect [28] (see Remark 3).

The second point concerns the numerical procedure used to compute rigorous enclosures for the costate variables. In contrast to the sensitivity based approach into which the original set of differential equations is augmented by the first and second order sensitivity equations, and bounds on $\boldsymbol{x}$, $\boldsymbol{x}_{p_i}$ and $\boldsymbol{x}_{p_ip_j}$ can be directly derived from Theorem 2 (*e.g.*, by applying natural interval extensions), a two-step integration procedure is required in the adjoint based approach. The integration of the state/costate system as a coupled system is indeed unfeasible since the state system may be unstable when solved backward in time [10]. The procedure is thus as follow:

1. compute bound on $\boldsymbol{x}$ and $\boldsymbol{x}_{p_i}$ by solving the state system (coupled to the state sensitivity system) forward to the final time t_{n_s};

2. compute bounds on $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_{p_i}$ by solving the costate system (coupled to the costate sensitivity system) backward to the start point t_0, from their natural boundary conditions at t_{n_s} and satisfying the transition conditions at intermediate times t_k ($k = 1,\ldots,n_s-1$).

In particular, all of the necessary information about the state variables can be stored at some points during the forward pass, and then used to obtain the values of the state variables by interpolating during the backward integration of the adjoint system. Note that in the current implementation of the adjoint based approach, the state and state sensitivity variables are stored at pre-specified mesh points in each time subinterval $\left[t_{k-1}^-, t_k^+\right]$, $k = 1,\ldots,n_s$ (along with their time derivatives), and the solution is then reconstructed at any time by cubic Hermite interpolation. It is obvious that the validity of the α coefficients thus strongly depends upon the number of mesh points used. However, it was found that accurate values for these coefficients can be obtained for a limited number of points in any of the test problems presented in section 4 (*i.e.*, several hundred points in each time subinterval). These considerations are illustrated hereafter in an example.

In order to guarantee rigorous α coefficients regardless of the number of used mesh points, an interpolation procedure based on the interpolating polynomials underlying the BDF formula (*e.g.*, in the case solver DASSL is used) would be preferable. These aspects are currently being tackled. In addition, algorithms for global error control in ordinary differential equations, *i.e.* algorithms being capable to control the error in the final solution within a given tolerance, are becoming applicable within reasonable computational times (see, *e.g.*, [11]). It is anticipated that the use of such algorithms could be advantageous to derive valid interval enclosure more efficiently for initial value problems, as well as boundary value problems.

Illustrative example (continued) – We apply both sensitivity and adjoint based approaches in order to derive a convex-underestimator for the functional $\mathcal{F}$ defined in problem (P_1). Bounds on the variables appearing in the differential systems are first derived based on Theorem 2 *via* natural interval extensions. Then, values for the α coefficient are computed in each case so that,

$$\begin{array}{rcl} \breve{\mathcal{F}}(p) & = & -\left[x(1,p)\right]^2 + \alpha\left(p^U - p\right)\left(p^L - p\right) \\ \text{where} \quad \dot{x}(t,p) & = & -\left[x(t,\boldsymbol{p})\right]^2 + p\ ,\ \forall t \in [0,1] \\ x(0,p) & = & 9 \end{array}$$

yields a rigorous convex underestimator of $\mathcal{F}$ for any $p \in \left[p^L, p^U\right]$

Sensitivity based approach – A value for the α coefficient is obtained from:

$$\begin{array}{rcl} \alpha & = & \max\left\{x^L(1)\,x^L_{p^2}(1), x^L(1)\,x^U_{p^2}(1), x^U(1)\,x^L_{p^2}(1), x^U(1)\,x^U_{p^2}(1)\right\} \\ & & +\max\left\{x^L_p(1)^2, x^U_p(1)^2\right\} \end{array}$$

$$\text{with} \quad \left.\begin{array}{rcl} \dot{x}^L & = & -{x^L}^2 + p^L \\ \dot{x}^U & = & -{x^U}^2 + p^U \\ \dot{x}^L_p & = & -2\max\left\{x^L x^L_p, x^U x^L_p\right\} + 1 \\ \dot{x}^U_p & = & -2\min\left\{x^L x^U_p, x^U x^U_p\right\} + 1 \\ \dot{x}^L_{p^2} & = & -2\max\left\{x^L x^L_{p^2}, x^U x^L_{p^2}\right\} - 2\max\left\{{x^L_p}^2, {x^U_p}^2\right\} \\ \dot{x}^U_{p^2} & = & -2\min\left\{x^L x^U_{p^2}, x^U x^U_{p^2}\right\} - 2\min\left\{{x^L_p}^2, {x^U_p}^2\right\} \end{array}\right\} \forall t \in [0,1]$$

$$\begin{array}{rcl} x^L(0) & = & 9 \\ x^U(0) & = & 9 \\ x^L_p(0) & = & 0 \\ x^U_p(0) & = & 0 \\ x^L_{p^2}(0) & = & 0 \\ x^U_{p^2}(0) & = & 0 \end{array}$$

Adjoint based approach – The application of the second approach yields the following results:

$$\alpha = \int_0^1 \lambda_p^L \, dt$$

$$\text{with} \quad \left.\begin{array}{rcl}
\dot{x}^L & = & -{x^L}^2 + p^L \\
\dot{x}^U & = & -{x^U}^2 + p^U \\
\dot{x}_p^L & = & -2\max\left\{x^L x_p^L, x^U x_p^L\right\} + 1 \\
\dot{x}_p^U & = & -2\min\left\{x^L x_p^U, x^U x_p^U\right\} + 1 \\
\dot{\lambda}^L & = & 2\max\left\{x^L \lambda^L, x^U \lambda^L\right\} \\
\dot{\lambda}^U & = & 2\min\left\{x^L \lambda^U, x^U \lambda^U\right\} \\
\dot{\lambda}_p^L & = & 2\max\left\{x^L \lambda_p^L, x^U \lambda_p^L\right\} \\
 & & +2\max\left\{x_p^L \lambda^L, x_p^L \lambda^U, x_p^U \lambda^L, x_p^U \lambda^U\right\} \\
\dot{\lambda}_p^U & = & 2\min\left\{x^L \lambda_p^U, x^U \lambda_p^U\right\} \\
 & & +2\min\left\{x_p^L \lambda^L, x_p^L \lambda^U, x_p^U \lambda^L, x_p^U \lambda^U\right\}
\end{array}\right\} \forall t \in [0,1]$$

$$\begin{array}{rcl}
x^L(0) & = & 9 \\
x^U(0) & = & 9 \\
x_p^L(0) & = & 0 \\
x_p^U(0) & = & 0 \\
\lambda^L(1) & = & -2x^U(1) \\
\lambda^U(1) & = & -2x^L(1) \\
\lambda_p^L(1) & = & -2x_p^U(1) \\
\lambda_p^U(1) & = & -2x_p^L(1)
\end{array}$$

It should be noted that the "max" function is used in the expression of $\dot{\lambda}^L$ (instead of the "min" one) since the costate equations and their sensitivities are integrated backward in time. A proof to this assertion can be easily obtained by reversing time (*e.g.*, by using the variable change $t \to -t$), and then applying Theorem 2. This remark also holds for the expressions of $\dot{\lambda}^U$, $\dot{\lambda}_p^L$ and $\dot{\lambda}_p^U$

The results obtained from the numerical implementation of both approaches are given in Tab. 1 for different parameter ranges $\left[p^L, p^U\right]$. The values corresponding to various number of mesh points in the adjoint based approach are also shown in this table, along with the value corresponding the tightest α-based convex-underestimators.

In this example, the application of the adjoint method yields tighter underestimators than the sensitivity method, but the results are close however. Note in particular that accurate values are obtained for the α coefficients in the adjoint based approach even for rather limited numbers of mesh points, hence validating the numerical approach detailled in subsection 3.2. On the other hand, comparisons with the optimal α coefficients indicate that both underestimating strategies provide satisfactory results. In particular, the best α-based underestimator is obtained for both strategies on the domains $[-5, 5]$ and $[-5, 0]$. As an illustration, the functional $\mathcal{F}$ and its convex relaxation are pictured in Fig. 1 on different parameter domains for the adjoint underestimating procedure.

Table 1: Computation of the α coefficient for different parameter ranges $\left[p^L, p^U\right]$.

	Parameter range $\left[p^L, p^U\right]$				
Method	$[-5, 5]$	$[-5, 0]$	$[0, 5]$	$[0, 2.5]$	$[2.5, 5]$
Sensitivity	$\alpha = 6.049$	$\alpha = 6.049$	$\alpha = 0.1222$	$\alpha = 0.1111$	$\alpha = 0.04296$
Adjoint (500 pts)	$\alpha = 6.049$	$\alpha = 6.049$	$\alpha = 0.1138$	$\alpha = 0.1036$	$\alpha = 0.03727$
Adjoint (100 pts)	$\alpha = 6.050$	$\alpha = 6.050$	$\alpha = 0.1138$	$\alpha = 0.1036$	$\alpha = 0.03727$
Adjoint (20 pts)	$\alpha = 6.082$	$\alpha = 6.082$	$\alpha = 0.1138$	$\alpha = 0.1036$	$\alpha = 0.03728$
Tightest value	$\alpha = 6.049$	$\alpha = 6.049$	$\alpha = 0.0823$	$\alpha = 0.0823$	$\alpha = 0.0223$

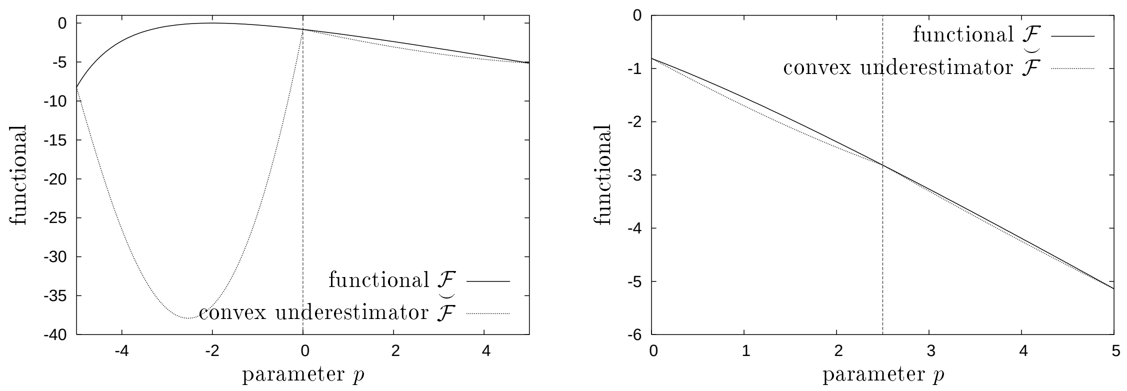

Figure 1: Functional $\mathcal{F}$ and convex relaxation on different parameter domains for the adjoint underestimating procedure. **left plot**: $p \in [-5, 0] \cup [0, 5]$; **right plot**: $p \in [0, 2.5] \cup [2.5, 5]$.

3.3 Spatial branch-and-bound algorithm

The optimisation problem (P) is solved to global optimality by applying the proposed underestimation procedure in the framework of a spatial branch-and-bound algorithm [20]. The structure of the algorithm is similar to those implemented in [13, 14]. The sequence of upper bounds on the global solution is generated by solving problem (P) to local optimality (see Definition 1). On the other hand, valid lower bounds on the global optimum are generated by constructing convex-underestimators for any individual function from the approach described in subsection 3.2; the lower bounding problem is thus defined as:

$$
\begin{array}{rcl}
\min\limits_{\boldsymbol{p}\in[\boldsymbol{p}^L,\boldsymbol{p}^U]} \breve{\mathcal{J}}(\boldsymbol{p}) & = & \mathcal{G}_0\left(\boldsymbol{x}\left(t_1^-,\boldsymbol{p}\right),\ldots,\boldsymbol{x}\left(t_{n_s}^-,\boldsymbol{p}\right),\boldsymbol{p}\right) + \sum\limits_{k=1}^{n_s}\int_{t_{k-1}^+}^{t_k^-} \mathcal{L}^{(k)}(\boldsymbol{x},\boldsymbol{p})\,dt \\
& & + \sum\limits_{i=1}^{n_p} \alpha_{0i}\left(p_i^U - p_i\right)\left(p_i^L - p_i\right) \\
\text{s.t.} \qquad \dot{\boldsymbol{x}} & = & \boldsymbol{f}(\boldsymbol{x},\boldsymbol{p}) \ , \ \forall t \in [t_0, t_{n_s}] \\
\boldsymbol{x}(\boldsymbol{p},t_0) & = & \boldsymbol{h}(\boldsymbol{p}) \\
0 & \geq & \mathcal{G}_j\left(\boldsymbol{x}\left(t_1^-,\boldsymbol{p}\right),\ldots,\boldsymbol{x}\left(t_{n_s}^-,\boldsymbol{p}\right),\boldsymbol{p}\right) + \sum\limits_{k=1}^{n_s}\int_{t_{k-1}^+}^{t_k^-} \mathcal{L}_j^{(k)}(\boldsymbol{x},\boldsymbol{p})\,dt \\
& & + \sum\limits_{i=1}^{n_p} \alpha_{j_i}\left(p_i^U - p_i\right)\left(p_i^L - p_i\right) \ , \ j = 1,\ldots,n_c
\end{array}
\qquad (\breve{P})
$$

In comparison to the convex underestimation procedure described in [13, 14, 29], it should be noted that our procedure (i) is applicable to functionals with state-dependent integral terms and, (ii) does not add any new decision variable or constraint to formulate the relaxed problem $(\breve{P})$. To illustrate this latter point, we consider a parameter estimation problem. For such problems, the objective is typically to minimise the weighted squared error between the observed values and those predicted by the model at given time instants. If hundreds of experimental points were considered, the application of the underestimation procedure developped in [13, 14, 29] would result in a large number of additional variables and equality constraints in the relaxed problem formulation, hence yielding a large-scale optimisation problem (and increasing the computational time). In this work however, the size of the relaxed problem remains unchanged with respect to the original problem. In addition, since parameter estimation problems typically involve a large number of decision variables but relatively few constraints, the use of the adjoint based approach might be far more efficient from the computational point of vue. The same remarks also hold for problems with continuous control profiles being discretised over a large number of time intervals.

Since natural interval extensions are used to compute the α values at each branch-and-bound node, the bounding operation is improved by successive partitioning the parameter space (*e.g.*, by applying a simple bisection rule). Guaranteed ε-convergence of the algorithm is thus obtained in finite time. Note however that, apart from the quality of the lower bounding problem, the convergence of the branch-and-bound algorithm may be affected by the methods used to select the branching variable and determine appropriate variable bounds at each node of the tree.

- Two branching strategies have been investigated in this work [1]. The first one corresponds to the classical *least reduced axis rule*. In the second strategy, a measure of the overall influence of each variable on the tightness of the lower bounding problem is calculated, and the variable having the worst measure is selected for further bisection.

- The update of the bounds on some or any of the optimisation parameter can also be addressed at each branch-and-bound node by solving a series of additional dynamic optimisation problems (see, *e.g.*, [13, 14]).

4 Implementation and case studies

The algorithmic procedure is implemented in an extensive `Fortran90/95` program. A link to the dynamic optimisation program `DYNO` [15] is used to perform the local optimisations; `DYNO` has itself links to various local SQP solvers and integration routines (in this case, `NLPQL` [33] and `DDASSL` [8] are used). For the computation of rigorous convex underestimators for the state-dependent functionals, natural interval extensions [25] are used as inclusion functions due to their ease of implementation. This is done by using the `INTERVAL_ARITHMETIC` package by Kearfott in `Fortran90` [21] which performs directed outward roundings during the calculations (based on the `INTLIB` package). Again, the generation of lower and upper bounds on the state/costate variables and their sensitivities uses the integration routine `DDASSL`. Links between this program and automatic differentiation packages such as ADIFOR [6] are currently being implemented in order to generate all the necessary differentiations. Finally, it should also be noted that, to date, the program is still a prototype implementation and, consequently, it is not optimised. Therefore, the computational times are reported only for the sake of comparison between the relaxation methods and the branching strategies.

In order to illustrate the theoretical and computational aspects of the proposed approach, four example problems dealing with either parameter estimation problems from time series data or simple optimal control problems are presented. Except the first optimisation problem which is the continuation of the illustrative example considered throughout the paper, all other problems are taken from the book of Floudas *et al.* [17].

4.1 Illustrative example (continued)

In this example, the same dynamic problem used to illustrate the convex-underestimation procedure in section 3 is considered. This problem has also been addressed as an illustrative example in [29] . It is formulated as:

$$\begin{aligned} \min_p \ \mathcal{J} \ &= \ -x(1)^2 \\ \text{s.t. } \dot{x}(t) \ &= \ -x^2 + p \ , \ \forall t \in [0,1] \\ x(0) \ &= \ 9 \\ -5 \ &\le \ p \ \le \ 5 \end{aligned} \tag{P_1}$$

It can be shown that problem P_1 has two local optima $(p_1^\star, \mathcal{J}_1^\star) = (-5, -8.23262)$ and $(p_2^\star, \mathcal{J}_2^\star) = (5, -5.13944)$, *i.e.*, one at each bound of the parameter domain. Details on the procedures applied to build α-based convex-underestimators for the objective function have been given in section 3. Both adjoint and sensitivity based approaches are implemented here to compute the global solution. The results are given in Tab. 2.

Table 2: Global optimisation results for the illustrative example using different underestimation procedures and branching strategies.

Underestimation approach	Branching strategy	Iterations	CPU time (s)
Sensitivity	–	3	2
Adjoint	–	3	7

The branching procedure is not specified in this example since the problem is monodimensional. For both approaches, the global optimum $(p_1^\star, \mathcal{J}_1^\star)$ is reached in 3 iterations. Large differences can however be observed in terms of computational efficiency since the use of the adjoint approach leads CPU times which are three times longer; this is due to the time spent to integrate the TBPVPs. Note in particular that these results are coherent with the considerations discussed in subsection 3.1.3).

4.2 First-order irreversible series reaction

The second example concerns the estimation of the yield coefficients k_1 or k_2 for a two-step irreversible isothermal liquid-phase chain reaction from times series data:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

It has been addressed in [37], as well as in [17], [14] and [29]. The parameter estimation problem involves two parameters k_1, k_2 corresponding to the rate constants of each reaction, and two state variables x_1, x_2 representing the mole fractions of components A and B respectively; it can be formulated as follow:

$$\begin{aligned}
\min_{k_1,k_2} \mathcal{J} &= \sum_{k=1}^{10}\sum_{i=1}^{2}\left(x_i(t_k) - x_i^{\text{mes}}(t_k)\right)^2 \\
\text{s.t. } \dot{x}_1(t) &= -k_1 x_1(t) \\
\dot{x}_2(t) &= k_1 x_1(t) - k_2 x_2(t) \quad \Big\} \ \forall t \in [0,1] \\
x_1(0) &= 1 \\
x_2(0) &= 0 \\
0 \leq \ & k_1, k_2 \leq 10
\end{aligned} \qquad (P_2)$$

where $x_i^{\text{mes}}(t_k)$ denotes the experimental point for the ith state variable at time t_k; these values are taken from [17].

Table 3: Global optimisation results for the first-order irreversible series reaction problem (P_2) using different underestimation procedures and branching strategies.

Underestimation approach	Branching strategy	Iterations	CPU time (s)
Sensitivity	1	32	328
Sensitivity	2	27	280
Adjoint	1	32	611
Adjoint	2	25	482

The global optimum in problem P_2 is obtained for $(k_1, k_2) = (5.0035, 1.0000)$; the corresponding value of the objective function at that point is $\mathcal{J} = 1.18562\,10^{-6}$. The result of the global optimisation algorithm are shown in Tab. 3 for a relative tolerance of $\varepsilon = 0.001$.

These results indicate that the adjoint approach is advantageous in terms of iterations – for the same branching strategy –, *i.e.*, the adjoint approach yields tighter underestimators than those obtained in the sensitivity approach. On the contrary, the adjoint approach leads to much larger CPU times due to the integration of the TPBVPs (as in problem P_1). Recall however that the program which is used is not optimised; we believe in particular that the efficiency of the adjoint approach could be significantly improved. On the other hand, it can be seen that the use of the branching strategy based on the underestimation quality (strategy 2) yields better results than the simple least reduced axis rule (strategy 1) – for the same type of underestimation –; the benefits can be up to 25%. Finally, it should be mentioned that the proposed approach provides the global optimum of problem P_2 in fewer iterations than the approach of Papamichail and Adjiman [29] for the same relative tolerance; these results thus validate the proposed approach as a valuable tool to perform global optimisation of problems with ODEs in the constraints.

4.3 Catalytic cracking of gas oil

The third case study deals with the catalytic cracking of gas oil (A) to gasoline (Q) and other products (S). The overall reaction is given by:

$$\begin{tikzcd} A \arrow[rr, "k_1"] \arrow[dr, "k_3"'] & & Q \arrow[dl, "k_2"] \\ & S & \end{tikzcd}$$

The parameter estimation problem has been previously addressed in [37], and is also studied in [14, 29]. It can be formulated as:

$$
\begin{array}{rcll}
\displaystyle\min_{k_1,k_2,k_3} \mathcal{J} & = & \displaystyle\sum_{k=1}^{20}\sum_{i=1}^{2}\left(x_i\left(t_k\right)-x_i^{\text{mes}}\left(t_k\right)\right)^2 & \\
\text{s.t. } \dot{x}_1\left(t\right) & = & -\left(k_1+k_3\right)x_1\left(t\right)^2 & \left.\right\} \forall t\in[0,0.95] \\
\dot{x}_2\left(t\right) & = & k_1x_1\left(t\right)^2-k_2x_2\left(t\right) & \\
x_1\left(0\right) & = & 1 & \\
x_2\left(0\right) & = & 0 & \\
0 & \leq & k_1,k_2,k_3 \ \leq \ 20 &
\end{array}
\qquad (P_3)
$$

where $x_i^{\text{mes}}\left(t_k\right)$ denotes the experimental point for the ith state variable at time t_k; again, these values are taken from [17]. In addition, x_1 an x_2 are the mole fractions of components A and Q respectively, whereas the molar fraction of component S is unmeasured.

The results obtained by applying the global optimisation algorithm for different underestimation and branching strategies are shown in Tab. 4. In any cases, the algorithm converges to the global solution $\mathcal{J}^\star = 2.65567\,10^{-3}$, with parameter values $(k_1^\star, k_2^\star, k_3^\star) = (12.2140, 7.9798, 2.2217)$.

Table 4: Global optimisation results for the catalytic cracking problem (P_3) using different underestimation procedures and branching strategies.

Underestimation approach	Branching strategy	Iterations	CPU time (s)
Sensitivity	1	234	11,500
Sensitivity	2	210	10,400
Adjoint	1	268	19,300
Adjoint	2	215	15,400

On the contrary to the results obtained in the previous examples, the sensitivity approach is here preferable either in terms of iterations or with respect to the overall computational time. Additionally, it is also found that the branching strategy which is based on the underestimation quality (strategy 2) yields better results. One should note however, that the results reported in [29], on the same example problem, are better since convergence is obtained in less than 100 hundred iterations.

4.4 Singular optimal control problem

The last example considers a nonlinear singular optimal control problem. This problem has been studied in [22] first, and is also addressed in [31] and [13]. The problem formulation is as follow:

$$
\begin{aligned}
\min_{u(t)} \mathcal{J} &= \int_0^1 x_1(t)^2 + x_2(t)^2 + 0.0005\left(x_2(t) + 16t - 8 - 0.1\, x_3(t)\, u(t)^2\right)^2 dt \\
\text{s.t. } \dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -x_3(t)\, u(t) + 16t - 8 \quad \forall t \in [0,1] \\
\dot{x}_3(t) &= u(t) \\
x_1(0) &= 0 \\
x_2(0) &= -1 \\
x_3(0) &= -\sqrt{5} \\
-4 \leq\ & u(t) \leq 10 \ , \ \forall t \in [0,1]
\end{aligned}
\qquad (P_4)
$$

Note that the original problem contains an integral term which explicitly depends on time t. Problem (P_4) was therefore reformulated by using an additional state variable x_4 representing time, *i.e.*, $\dot{x}_4 = 1$, and then replacing t in the integral term by $x_4(t)$ (see section 2). In addition, the control variable $u(t)$ was parameterised as a piecewise constant profile (with fixed interval lengths). These modifications reformulate problem (P_4) into the form specified in Definition 1.

The results computed from the global optimisation algorithm with 2 control intervals are given in Tab. 5 for different underestimation procedures and branching strategies. By applying local optimisation techniques from random starting points, it can be shown that problem (P_4) has at least 2 local optima: $\mathcal{J}_1^\star = 0.27711$ and $\mathcal{J}_2^\star = 0.35175$, with the second one being found more frequently.

Table 5: Global optimisation results for the singular control problem (P_4) using different underestimation procedures and branching strategies.

Underestimation approach	Branching strategy	Iterations	CPU time (s)
Sensitivity	1	79	640
Sensitivity	2	72	502
Adjoint	1	92	1,011
Adjoint	2	75	781

The applications of the global optimisation approach yields the global optimum $\mathcal{J}_1^\star$ in any case, with parameters $(u_1^\star, u_2^\star) = (5.5748, -4.0000)$. As in the previous example, the use of the sensitivity based approach is preferable here since it improves the convergence to the global solution; the second branching option also achieves faster convergence results than the standard least reduced axis rule.

5 Conclusions and perspectives

In this paper, a deterministic global optimisation algorithm has been presented to solve problems with ordinary differential equations in the constraints. The algorithm is built on a spatial branch-and-bound framework. The novelty is in the procedure applied to derive convex relaxations of the original nonconvex dynamic optimisation problems. Convex underestimators are derived for each individual nonconvex twice continuously differentiable state-dependent functionals by adding a convex quadratic term of which weight coefficient α can be computed rigorously based on either a sensitivity or an adjoint based approach. Both uses the concept of differential inequalities to compute bounds on the state trajectories of ODE systems.

A prototype implementation of the algorithm has been realised in `Fortran90` and the approach is tested on several simple parameter estimation and optimal control problems taken from the literature. The results suggest that both sensitivity and adjoint based approaches provide accurate convex underestimators for nonconvex functionals. It was found that the use of either one of the approaches may be advantageous depending on the optimisation problem. It is worth noticing that the results are close in any case. In addition, the branching strategy has a large influence on the convergence of the algorithm. The use of branching strategy quantifying the influence of each variable on the tightness of the underestimator, rather than the standard least reduced axis rule was found to improve the convergence in all case studies. Finally, these theoretical and practical considerations validate the proposed method as an alternative approach to the optimisation of problems with ODEs in the constraints.

Current investigations are focussed on the application of this deterministic global optimisation algorithm to larger dynamic optimisation problems, and in particular problems with many decision variables such as those resulting from the application of control parameterisation techniques. For these problems, we believe that the application of the adjoint based approach might be preferable since its complexity increases linearly with the number of decision variables. In addition, the need to derive tight convex underestimators will be even more crucial for large-scale optimisation problems in order to compute the global solution in reasonable computational times.

References

[1] Adjiman C. S., Dallwig S., Floudas C. A. and Neumaier A. (1998a), "A global optimization method, αBB, for general twice differentiable constrained NLPs – I. Theoretical advances", *Computers and Chemical Engineering*, **22**(9):1137-1158.

[2] Adjiman C. S., Dallwig S. and Floudas C. A. (1998b), "A global optimization method, αBB, for general twice differentiable constrained NLPs – II. Implementation and computational results", *Computers and Chemical Engineering*, **22**(9):1159-1179.

[3] Androulakis I., Maranas C. D. and Floudas C. A. (1995), "αBB: A global optimization method for general constrained nonconvex problems", *Journal of Global Optimization*, **7**:337-363.

[4] Banga J. R. and Seider W. D. (1996), "*Global optimization of chemical processes using stochastic algorithms*", In: "State of the Art in Global Optimization", C. A. Floudas and P. M. Pardalos Editors, Series in nonconvex optimization and its applications, pp. 563-583, Dordrecht, Kluwert Academic Publishers.

[5] Biegler L. T. (1984), "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation", *Computers and Chemical Engineering*, **8**(3/4):243-248.

[6] Bischof C., Carle A., Corliss G., Griewank A., Hovland P. (1992), "`ADIFOR` - Generating Derivative Codes from Fortran Programs" *Scientific Programming*, **1**:1-29.

[7] Boender C. G. and Romeijn H. E. (1995), "*Stochastic methods*", In: "Handbook of Global Optimization", R. Horst and P. M. Pardalos Editors, Series in nonconvex optimization and its applications, pp. 829-869, Dordrecht, Kluwert Academic Publishers.

[8] Brenan K. E., Campbell S. E. and Petzold L. R. (1989), "*Numerical solution of initial value problem in differential-algebraic equations*", North-Holland, New York.

[9] Bryson A. E. and Ho Y. C. (1975), "*Applied Optimal Control*", Hemisphere Publishing Corporation.

[10] Cao Y., Li S. and Petzold L. (2002), "Adjoint sensitivity analysis for Differential-Algebraic Equations: Algorithms and software", *J. Comp. Appl. Math.*, **149**:171-192.

[11] Cao Y. and Petzold L. (2003), "A posteriori error estimation and global error control for Ordinary Differential Equations by the adjoint method", *SIAM J. Sci. Comput.*, **submitted for publication**.

[12] Chachuat B. and Latifi M. A. (2003), "`GDO`: a `fortran90` program for global optimisation of problems with ordinary differential equations", *Technical report, LSGC-CNRS (Nancy, France)*, **in preparation**.

[13] Esposito W. R. and Floudas C. A. (2000a), "Deterministic global optimization in nonlinear optimal control problems", *Journal of Global Optimization*, **17**:96-126.

[14] Esposito W. R. and Floudas C. A. (2000b), "Global optimization for the parameter estimation of differential-algebraic systems", *Industrial & Engineering Chemistry Research*, **39**(5):1291-1310.

[15] Fikar M. and Latifi M. A. (2002), "User's guide for `fortran` dynamic optimisation code `DYNO`", *Technical report, LSGC-CNRS (Nancy, France) & STU Bratislava (Slovak Republic)*".

[16] Floudas C. A (2000), "*Deterministic global optimization: Theory, methods and applications*", Series in nonconvex optimization and its applications, Dordrecht, Kluwert Academic Publishers.

[17] Floudas C. A., Pardalos P. M., Adjiman C. S., Esposito W. R.,Gümüs Z. H., Harding S. T., Klepeis J. L., Meyer C. A. and Schweiger C. A. (1999), "*Handbook of test problems in local and global optimization*", Series in nonconvex optimization and its applications, Dordrecht, Kluwert Academic Publishers.

[18] Frank P. M. (1978), "*Introduction to System Sensitivity Theory*", Academic Press, New York.

[19] Harrison G. W. (1979), "Compartmental models with uncertain flow rates", *Mathematical Biosciences*, **43**:131-139.

[20] Horst R. and Tuy H. (1996), "*Global Optimization. Deterministic Approaches*" (3rd edition), Springer-Verlag, Berlin.

[21] Kearfott R. B. (1996), "`INTERVAL_ARITHMETIC`: A `Fortran90` module for an interval data type", *ACM Transaction on Mathematical Software*, **22**(4):385-392.

[22] Luus R. (1990), "Optimal control by dynamic programming using systematic reduction in grid size", *International Journal of Control*, **51**(5):995-1013.

[23] Luus R. (1992), "Mutiplicity of solutions in the optimization of a bifunctional catalyst blend in a tubular reactor", *Canadian Journal of Chemical Engineering*, **70**:780-785.

[24] Maranas C. D. and Floudas C. A. (1994), "Global minimum potential energy conformations of small molecules", *Journal of Global Optimization*, **4**:135-170.

[25] Moore R. E. (1966), "*Interval Analysis*", Prentice-Hall, Engelwood Cliffs, NJ.

[26] Moore R. E. (1979), "*Methods and Applications of Interval Analysis*", SIAM, Philadelphia, PA.

[27] Neuman C. and Sen A. (1973), "A suboptimal control algorithm for constraint problems using cubic splines", *Automatica*, **9**:601-613.

[28] Neumaier A. (1993), "Global, rigorous and realistic bounds for the solution of dissipative differential equations. Part I: Theory", *Computing*, **52**:315-336.

[29] Papamichail I. and Adjiman C. S. (2002), "A rigorous global optimization algorithm for problems with ordinary differential equations", *Journal of Global Optimization*, **24**:1-33.

[30] Pontryagin L. (1962), "*The Mathematical Theory of Optimal Processes*", Interscience Publishers, New York.

[31] Rosen O. and Luus R.(1997), "Global optimization approach to nonlinear optimal control", *Journal of Optimization Theory & Applications*, **73**:547-562.

[32] Ruban A. I. (1997), "Sensitivity coefficients for discontinuous dynamic systems", *Jounal of Computer and System Sciences International*, **36**(4):536-542.

[33] Schittkowski K. (1985), "NLPQL : a Fortran Subroutine Solving Constrained Nonlinear Programming Problems", *Annals of Operations Research*", **5**(6):485-500.

[34] Singer A. B. and Barton P. I. (2003), "Global solution of linear dynamic embedded optimization problems – Part I: Theory", *Journal of Optimization Theory & Applications*, **Submitted for publication**.

[35] Smith E. M. and Pantelides C. C. (1995), "*Global optimisation of general process models*", In: "Global optimization in engineering design", I.E. Grossmann Editor, Series in nonconvex optimization and its applications, Chapter 12, pp. 355-386, Dordrecht, Kluwert Academic Publishers.

[36] Teo K., Goh G. and Wong K. (1991), "*A Unified Computational Approach to Optimal Control Problems*", Pitman Monographs and Surveys in Pure and Applied Mathematics, John Wiley & Sons, Inc., New York, 1991.

[37] Tjoa I. B. and Biegler L. T. (1991), "Simulatneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems", *Industrial & Engineering Chemistry Research*, **30**:376-385.

[38] Tsang T. H., Himmelblau D. M. and Edgar T. F. (1975), "Optimal control via collocation and nonlinear programming", *International Jounal of Control*, **21**(5):763-768.

[39] Vassiliadis V. S., Canto E. B. and Banga J. R. (1999), "Second-order sensitivity of general dynamic systems with application to optimal control problems", *Chemical Engineering Science*, **54**(17):3851-3860.

[40] Vassiliadis V. S., Sargent R. W. and Pantelides C. C. (1994a), "Solution of a class of multistage dynamic optimisation problems – 1. Problems without path constraints", *Industrial & Engineering Chemistry Research*, **33**(9):2111-2122.

[41] Vassiliadis V. S., Sargent R. W. and Pantelides C. C. (1994b), "Solution of a class of multistage dynamic optimisation problems – 2. Problems with path constraints", *Industrial & Engineering Chemistry Research*, **33**(9):2123-2133.

[42] Walter W. (1970), "*Differential and Integral Inequalities*", Springer-Verlag, Berlin.