

## Research Article

# Partitioned Quasi-Newton Approximation for Direct Collocation Methods and Its Application to the Fuel-Optimal Control of a Diesel Engine

Jonas Asprión,<sup>1</sup> Oscar Chinellato,<sup>2</sup> and Lino Guzzella<sup>1</sup>

<sup>1</sup> Institute for Dynamic Systems and Control, ETH Zurich, Sonneggstrasse 3, 8092 Zurich, Switzerland

<sup>2</sup> FPT Motorenforschung AG, Schlossgasse 2, 9320 Arbon, Switzerland

Correspondence should be addressed to Jonas Asprión; [jonas.asprien@alumni.ethz.ch](mailto:jonas.asprien@alumni.ethz.ch)

Received 26 November 2013; Revised 17 February 2014; Accepted 18 February 2014; Published 25 March 2014

Academic Editor: M. Montaz Ali

Copyright © 2014 Jonas Asprión et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The numerical solution of optimal control problems by direct collocation is a widely used approach. Quasi-Newton approximations of the Hessian of the Lagrangian of the resulting nonlinear program are also common practice. We illustrate that the transcribed problem is separable with respect to the primal variables and propose the application of dense quasi-Newton updates to the small diagonal blocks of the Hessian. This approach resolves memory limitations, preserves the correct sparsity pattern, and generates more accurate curvature information. The effectiveness of this improvement when applied to engineering problems is demonstrated. As an example, the fuel-optimal and emission-constrained control of a turbocharged diesel engine is considered. First results indicate a significantly faster convergence of the nonlinear program solver when the method proposed is used instead of the standard quasi-Newton approximation.

## 1. Introduction

Quasi-Newton (QN) methods have become very popular in the context of nonlinear optimisation. Above all, in nonlinear programs (NLPs) arising from direct transcription of optimal control problems (OCPs), the Hessian of the Lagrangian often cannot be derived analytically in a convenient way. Algorithmic differentiation may fail due to unsupported operations or black-box parts in the model functions. Furthermore, both approaches are computationally expensive if the model functions are complex and yield long expressions for the second derivatives. On the other hand, numerical approximation by finite differences is inaccurate and hardly improves the computational performance.

A common approach in these cases is to approximate the Hessian by QN updates using gradient information collected during the NLP iterations. However, if applied to real-world OCPs, several limitations arise. These problems often exhibit large dimensions; thus limited-memory versions of the approximations are applicable only. Since many updates

may be necessary until a good approximation of the full Hessian is obtained, the approximation remains poor when using the most recent steps only. Furthermore, the favourable sparsity structure of the underlying discretisation scheme is generally not preserved. This fill-in drastically reduces the performance of solvers for the linear system of equations defining the step direction during each NLP iteration.

Partial separability, a concept introduced by [1], describes a structural property of a nonlinear optimisation problem. When present, this property allows for partitioned QN updates of the Hessian of the Lagrangian (or of the objective, in the unconstrained case). For unconstrained optimisation, this approach was proposed and its convergence properties were analysed in [2]. Although only superlinear local convergence can be proven, a performance close to that obtained with exact Newton methods, which exhibit quadratic local convergence, was observed in practical experiments [3, 4].

This brief paper presents how partitioned QN updates can be applied to the NLPs resulting from direct collocation of OCPs. The concept of direct collocation to solve OCPs

has been widely used and analysed [5–9]. We will show that the Lagrangian of the resulting NLPs is separable in the primal variables at each discretisation point. Due to this separability, its Hessian can be approximated by full-memory QN updates of the small diagonal blocks. This procedure increases the accuracy of the approximation, preserves the sparsity structure, and resolves memory limitations. The results are first derived for Radau collocation schemes, which include the right interval boundary as a collocation point. The adaptations to Gauss schemes, which have internal collocation points only, and to Lobatto schemes, which include both boundaries, are provided thereafter in condensed form. A consistent description of all three families of collocation is provided in [10].

The partitioned update is applied to a real-world engineering problem. The fuel consumption of a turbocharged diesel engine is minimised while the limits on the cumulative pollutant emissions need to be satisfied. This problem is cast in the form of an OCP and is transcribed by Radau collocation. The resulting NLP is solved by an exact and a quasi-Newton method. For the latter, the partitioned update achieves an increased convergence rate and a higher robustness with respect to a poor initialisation of the approximation as compared to the full QN update. Therefore, the findings for the unconstrained case seem to transfer to the NLPs resulting from direct collocation of OCPs.

## 2. Material and Methods

Consider the system of nonautonomous ordinary differential equations (ODEs)  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  on the interval  $[t_0, t_f]$ , with  $\mathbf{x}, \mathbf{f} \in \mathbb{R}^{1 \times n_x}$ ,  $\mathbf{u} \in \mathbb{R}^{1 \times n_u}$ . Radau collocation represents each element of the state vector  $\mathbf{x}(t)$  as a polynomial, say of degree  $N$ . The time derivative of this polynomial is then equated to the values of  $\mathbf{f}$  at  $N$  collocation points  $t_0 < \tau_1 < \tau_2 < \dots < \tau_N = t_f$ ,

$$\begin{pmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \\ \vdots \\ \dot{\mathbf{x}}_N \end{pmatrix} \approx \underbrace{[\mathbf{d}_0, \tilde{\mathbf{D}}]}_{=: \mathbf{D}} \cdot \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} \mathbf{f}(\mathbf{x}_1, \mathbf{u}_1) \\ \mathbf{f}(\mathbf{x}_2, \mathbf{u}_2) \\ \vdots \\ \mathbf{f}(\mathbf{x}_N, \mathbf{u}_N) \end{pmatrix}. \quad (1)$$

The notation  $\mathbf{x}_j := \mathbf{x}(\tau_j)$  is adopted. The left boundary  $\tau_0 = t_0$  is a noncollocated point in Radau collocation schemes. By introducing the appropriate matrices, this matrix equation in  $\mathbb{R}^{N \times n_x}$  can be written in short as

$$\mathbf{D} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{X} \end{bmatrix} = \mathbf{F}(\mathbf{X}, \mathbf{U}). \quad (2)$$

The rows of  $\mathbf{X}$  and  $\mathbf{F}$  correspond to one collocation point each. In turn, the columns of  $\mathbf{X}$  and  $\mathbf{F}$  represent one state variable and its corresponding ODE right-hand side at all collocation points. In the following, consider the notation in (2) as shorthand for stacking the transpose of the rows in one large column vector.

The step length  $h = t_f - t_0$  of the interval is assumed to be accounted for in the differentiation matrix  $\mathbf{D}$ . Lagrange

interpolation by barycentric weights is used to calculate  $\mathbf{D}$  along with the vector of the quadrature weights  $\mathbf{w}$  [11]. The latter may be used to approximate the definite integral of a function  $g(t)$  as  $\int_{t_0}^{t_f} g(t) dt \approx \sum_{j=1}^N w_j g(\tau_j)$ .

*2.1. Direct Collocation of Optimal Control Problems.* We consider an OCP of the form

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_0^T L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (3a)$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = 0, \quad \forall t \in [0, T], \quad (3b)$$

$$\int_0^T \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) dt \leq 0, \quad (3c)$$

$$\mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0, \quad \forall t \in [0, T], \quad (3d)$$

where  $L \in \mathbb{R}$ ,  $\mathbf{g} \in \mathbb{R}^{n_g}$ , and  $\mathbf{c} \in \mathbb{R}^{n_c}$ . Simple bounds on  $\mathbf{x}$  and  $\mathbf{u}$ , equality constraints, and a fixed initial or end state can be included in the path constraints (3d). An objective term in Lagrange form is used, which is preferable over an equivalent Mayer term [12, Section 4.9].

Direct transcription discretises all functions and integrals by consistently applying an integration scheme. Here,  $k = 1, \dots, m$  integration intervals  $[t_{k-1}, t_k]$  are used with  $0 = t_0 < t_1 < \dots < t_m = T$ . The number of collocation points  $N_k$  can be different for each interval. Summing up the collocation points throughout all integration intervals results in a total of  $M = l(m, N_m)$  discretisation points. The “linear index”  $l$  thereby corresponds to collocation node  $i$  in interval  $k$ ,

$$l := l(k, i) = i + \sum_{\alpha=1}^{k-1} N_\alpha. \quad (4)$$

The transcribed OCP reads

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{l=1}^M W_l \cdot L(\mathbf{x}_l, \mathbf{u}_l) \quad (5a)$$

$$\text{s.t.} \quad \mathbf{D}^{(k)} \begin{bmatrix} \mathbf{X}_{N_{k-1}, \bullet}^{(k-1)} \\ \mathbf{X}_{N_k, \bullet}^{(k)} \end{bmatrix} - \mathbf{F}(\mathbf{X}^{(k)}, \mathbf{U}^{(k)}) = 0, \quad k = 1, \dots, m, \quad (5b)$$

$$\sum_{l=1}^M W_l \cdot \mathbf{g}(\mathbf{x}_l, \mathbf{u}_l) \leq 0, \quad (5c)$$

$$\mathbf{c}_l(\mathbf{x}_l, \mathbf{u}_l) \leq 0, \quad l = 1, \dots, M. \quad (5d)$$

The notation  $x_l$  denotes all instances of variable  $x_i$  at any applicable index  $l$ . The vector of the “global” quadrature weights  $\mathbf{W}$  results from stacking the vectors of the quadrature weights  $\mathbf{w}^{(k)}$  of each interval  $k$  after removing the first element, which is zero. For the first interval,  $\mathbf{X}_{N_{k-1}, \bullet}^{(k-1)}$  is the initial state  $\mathbf{x}(0)$ .

The Lagrangian of the NLP (5a)–(5d) is the sum of the objective (5a) and all constraints (5b), (5c), and (5d), which are weighted by the Lagrange multipliers  $\lambda$ . To clarify the

notation, the Lagrange multipliers are grouped according to the problem structure. The  $n_x \cdot N_k$  multipliers for the discretised dynamic constraints on each integration interval  $k$  are denoted by  $\lambda_d^{(k)}$ , the  $n_g$  multipliers for the integral inequalities are stacked in  $\lambda_g$ , and the  $n_c$  multipliers for the path constraints at each discretisation point  $l$  are gathered in the vector  $\lambda_{c,l}$ .

**2.2. Separability in the Primal Variables.** The objective (5a), the integral inequalities (5c), and the path constraints (5d) are inherently separated with respect to time; that is, the individual terms are pairwise disjoint in  $\mathbf{x}_i$  and  $\mathbf{u}_i$ . We thus focus on the separability of the dynamic constraints (5b). For the derivation, we assume that  $f$ ,  $x$ , and  $u$  are scalar. The extension to the vector-valued case is straightforward and will be provided subsequently.

Consider the term of the Lagrangian representing the discretised dynamic constraints (5b) for interval  $k$ ,

$$\widetilde{\mathcal{L}}_d^{(k)} := \sum_{i=1}^{N_k} \lambda_{d,i}^{(k)} \left( d_{0,i}^{(k)} x_{N_{k-1}}^{(k-1)} + \sum_{j=1}^{N_k} \widetilde{D}_{ij}^{(k)} x_j^{(k)} - f(x_i^{(k)}, u_i^{(k)}) \right). \quad (6)$$

This formulation constitutes a separation in the dual variables (the Lagrange multipliers). By collecting terms at each collocation point and accounting for the  $d_0$  terms in the previous interval, we obtain a separation in the primal variables,

$$\mathcal{L}_d^{(k)} = \sum_{i=1}^{N_k} \left( \left[ \sum_{j=1}^{N_k} \widetilde{D}_{ji}^{(k)} \lambda_{d,j}^{(k)} + \delta_i^{(k)} \sum_{j=1}^{N_{k+1}} d_{0,j}^{(k+1)} \lambda_{d,j}^{(k+1)} \right] x_i^{(k)} - \lambda_{d,i}^{(k)} f(x_i^{(k)}, u_i^{(k)}) \right), \quad (7)$$

with

$$\delta_i^{(k)} = \begin{cases} 1, & \text{if } i = N_k, k \neq m, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Each term inside the round brackets in (7) is a collocation-point separated part of the Lagrangian which stems from the dynamic constraints. We denote these terms by  $\mathcal{L}_{d,i}^{(k)}$  and introduce the notation

$$\omega := (x, u), \quad \omega_i^{(k)} := (x_i^{(k)}, u_i^{(k)}). \quad (9)$$

The gradient with respect to the primal variables is

$$\nabla_{\omega} \mathcal{L}_{d,i}^{(k)} = \left( \lambda_d^{(k)T} \widetilde{D}_{\cdot i}^{(k)} + \delta_i^{(k)} \lambda_d^{(k+1)T} \mathbf{d}_0^{(k+1)}, 0 \right) - \lambda_{d,i}^{(k)} \frac{\partial f(\omega_i^{(k)})}{\partial \omega}. \quad (10)$$

The Hessian is simply

$$\nabla_{\omega}^2 \mathcal{L}_{d,i}^{(k)} = -\lambda_{d,i}^{(k)} \frac{\partial^2 f(\omega_i^{(k)})}{\partial \omega^2}. \quad (11)$$

### 2.2.1. Vector-Valued Case and Complete Element Lagrangian.

For multiple control inputs and state variables, the primal variables  $\omega_l$  at each collocation point become a vector in  $\mathbb{R}^{1 \times (n_u + n_x)}$ . Consistently, we define the gradient of a scalar function with respect to  $\omega$  as a row vector. Thus, the model Jacobian  $\partial \mathbf{f} / \partial \omega$  is a matrix in  $\mathbb{R}^{n_x \times (n_u + n_x)}$ , and the Hessian of each model-function element  $s$ ,  $\partial^2 f_s / \partial \omega^2$ , is a square matrix of size  $(n_u + n_x)$ . The multiplier  $\lambda_{d,i}^{(k)}$  itself also becomes a vector in  $\mathbb{R}^{n_x}$ . All terms involving  $\mathbf{f}$ , its Jacobian, or its Hessian therefore turn into sums.

The full element Lagrangian  $\mathcal{L}_i^{(k)}$  consists of the terms of the dynamic constraints  $\mathcal{L}_{d,i}^{(k)}$  as derived above, plus the contributions of the objective, the integral inequalities, and the path constraints,

$$\begin{aligned} \mathcal{L}_i^{(k)} &= W_l \cdot L(\omega_l) \\ &+ \mathbf{x}_i^{(k)} \cdot \left( \sum_{j=1}^{N_k} \widetilde{D}_{ji}^{(k)} \lambda_{d,j}^{(k)} + \delta_i^{(k)} \sum_{j=1}^{N_{k+1}} d_{0,j}^{(k+1)} \lambda_{d,j}^{(k+1)} \right) \\ &- \mathbf{f}(\omega_i^{(k)}) \lambda_{d,i}^{(k)} + W_l \cdot \lambda_g^T \mathbf{g}(\omega_l) + \lambda_{c,l}^T \mathbf{c}_l(\omega_l). \end{aligned} \quad (12)$$

The Lagrangian of the full NLP is obtained by summing these element Lagrangians, which are separated in the primal variables. Its Hessian thus is a perfect block-diagonal matrix with uniformly sized square blocks of size  $(n_u + n_x)$ .

**2.2.2. Extension to Gauss and Lobatto Collocation.** Gauss collocation does not include the right interval boundary. Thus, the terms involving  $\mathbf{d}_0$  can be included locally in each interval, which simplifies the separation in the primal variables. However, the continuity constraint

$$\mathbf{x}_0^{(k+1)} = \mathbf{x}_0^{(k)} + \mathbf{w}^{(k)T} \mathbf{F}(\mathbf{X}^{(k)}, \mathbf{U}^{(k)}) \quad (13)$$

has to be introduced for each interval. Similarly to the procedure above, this constraint can be separated. The quadrature weights  $\mathbf{w}^{(k)}$  are stacked in  $\mathbf{W}$  without any modification.

Lobatto collocation includes both boundaries as collocation points. Thus, the matrix  $\mathbf{D}$  in (1) and (2) has an additional ‘‘zeroth’’ row, and the argument of  $\mathbf{F}$  becomes  $[\mathbf{x}_0^T, \mathbf{X}^T]^T$  in (2). The additional term

$$-\delta_i^{(k)} \lambda_{d,0}^{(k+1)} f(\omega_i^{(k)}) \quad (14)$$

arises in  $\mathcal{L}_{d,i}^{(k)}$ . Each element of  $\mathbf{W}$  corresponding to the interval boundary between any two neighbouring intervals  $k$  and  $k+1$  is a sum of the two weights  $w_{N_k}^{(k)}$  and  $w_0^{(k+1)}$ .

**2.3. Block-Diagonal Approximation of the Hessian.** The separability of the problem with respect to the primal variables allows a perfect exploitation of the problem sparsity. In fact, the Jacobian of the objective and the constraints, as well as the Hessian of the Lagrangian, can be constructed from the first and second partial derivatives of the nonlinear model functions  $L$ ,  $\mathbf{f}$ ,  $\mathbf{g}$ , and  $\mathbf{c}$  at each discretisation point [13].

We propose to also exploit the separability when calculating QN approximations of the Hessian. These iterative updates collect information about the curvature of the Lagrangian by observing the change of its gradient along the NLP iterations. Although they perform well in practice, they exhibit several drawbacks for large problems.

- (I) *Loss of sparsity.* QN approximations generally do not preserve the sparsity pattern of the exact Hessian, which leads to low computational performance [12, Section 4.13]. Enforcing the correct sparsity pattern results in QN schemes with poor performance [14, Section 7.3].
- (II) *Storage versus accuracy.* Due to the loss of sparsity, the approximated Hessian needs to be stored in dense format. To resolve possible memory limitations, “limited-memory” updates can be applied, which rely on a few recent gradient samples only. However, these methods provide less accuracy than their full-memory equivalents [14, Section 7.2].
- (III) *Dimensionality versus sampling.* When sampling the gradient of a function that lives in a high-dimensional space, many samples are required to construct an accurate approximation. In fact, to obtain an approximation that is valid along any direction, an entire spanning set needs to be sampled. Although QN methods require accurate second-order information only along the direction of the steps [15], the step direction may change fast in highly nonlinear problems such as the one considered here. In these cases, an exhaustive set of gradient samples would ensure a fast convergence, which conflicts with (II).

Using approximations of the small diagonal blocks, that is, exploiting the separability illustrated in Section 2.2, resolves these problems.

- (I) The exact sparsity pattern of the Hessian is preserved.
- (II) Only  $M(n_x + n_u)^2$  numbers have to be stored, compared to  $M^2(n_x + n_u)^2$  for the full Hessian.
- (III) Since the dimension of each diagonal block is small, a good approximation is already obtained after few iterations of the Hessian update [3, 4].

The partitioned QN update can be combined with the exploitation of the problem sparsity to reduce the number of the model evaluations required. In fact, when these two concepts are combined, the gradients of the model functions at each collocation point are sufficient to construct an accurate and sparsity-preserving approximation of the Hessian of the Lagrangian.

**2.4. Implementation.** Any QN approximation operates with the differences between two consecutive iterates and the corresponding gradients of the Lagrangian. For constrained problems,

$$\mathbf{s}^T := \widehat{\boldsymbol{\omega}} - \boldsymbol{\omega}, \quad (15)$$

$$\mathbf{y}^T := \nabla_{\boldsymbol{\omega}} \mathcal{L}(\widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\lambda}}) - \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}). \quad (16)$$

The hat indicates the values at the current iteration, that is, the new data. In the following formulas,  $\mathbf{B}$  denotes the QN approximation. Here, the damped BFGS update is used [14, Section 18.3], which reads

$$\theta = \begin{cases} 1, & \text{if } \mathbf{s}^T \mathbf{y} \geq 0.2 \mathbf{s}^T \mathbf{B} \mathbf{s} \\ \frac{0.8 \mathbf{s}^T \mathbf{B} \mathbf{s}}{\mathbf{s}^T \mathbf{B} \mathbf{s} - \mathbf{s}^T \mathbf{y}}, & \text{otherwise,} \end{cases} \quad (17a)$$

$$\mathbf{r} = \theta \mathbf{y} + (1 - \theta) \mathbf{B} \mathbf{s}, \quad (17b)$$

$$\widehat{\mathbf{B}} = \mathbf{B} - \frac{\mathbf{B} \mathbf{s} \mathbf{s}^T \mathbf{B}}{\mathbf{s}^T \mathbf{B} \mathbf{s}} + \frac{\mathbf{r} \mathbf{r}^T}{\mathbf{s}^T \mathbf{r}}. \quad (17c)$$

This update scheme preserves positive definiteness, which is mandatory if a line-search globalisation is used. In a trust-region framework, indefinite approaches such as the safeguarded SR1 update [14, Section 6.2] could be advantageous since they can approximate the generally indefinite Hessian of the full or element Lagrangian more accurately.

The Hessian block  $\mathbf{B}_l$  corresponding to the element Lagrangian (12) at collocation point  $l$  is approximated as follows. In the difference of the gradients, all linear terms cancel. Thus, (16) becomes

$$\begin{aligned} \mathbf{y}_l^T &= \nabla_{\boldsymbol{\omega}} \mathcal{L}_l(\widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\lambda}}) - \nabla_{\boldsymbol{\omega}} \mathcal{L}_l(\boldsymbol{\omega}, \boldsymbol{\lambda}) \\ &= W_l \cdot \left( \frac{\partial L(\widehat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial L(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} \right) \\ &\quad + \boldsymbol{\lambda}_{d,l}^T \cdot \left( \frac{\partial \mathbf{f}(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial \mathbf{f}(\widehat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} \right) \\ &\quad + W_l \cdot \boldsymbol{\lambda}_g^T \cdot \left( \frac{\partial \mathbf{g}(\widehat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial \mathbf{g}(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} \right) \\ &\quad + \boldsymbol{\lambda}_{c,l}^T \cdot \left( \frac{\partial \mathbf{c}_l(\widehat{\boldsymbol{\omega}}_l)}{\partial \boldsymbol{\omega}} - \frac{\partial \mathbf{c}_l(\boldsymbol{\omega}_l)}{\partial \boldsymbol{\omega}} \right). \end{aligned} \quad (18)$$

Recall that the linear index  $l$  is defined such that  $\boldsymbol{\lambda}_{d,l} = \boldsymbol{\lambda}_{d,i}^{(k)}$ . The QN update (17a)–(17c) is applied to each diagonal block  $\mathbf{B}_l$  individually, with  $\mathbf{s}_l^T = \widehat{\boldsymbol{\omega}}_l - \boldsymbol{\omega}_l$  and  $\mathbf{y}_l$  given by (18). As initialisation, one of the approaches described in [14, Chapter 6] can be used.

**2.5. Engineering Test Problem.** As a real-world engineering problem, we consider the minimisation of the fuel consumption of a turbocharged diesel engine. The statutory limits for the cumulative  $\text{NO}_x$  and soot emissions are imposed as  $n_g = 2$  integral inequality constraints. The  $n_u = 4$  control inputs are the position of the variable-geometry turbine, the start of the main injection, the common-rail pressure, and the fuel mass injected per cylinder and combustion cycle. The model is described and its experimental validation is provided in [16]. It features a dynamic mean-value model for the air path with  $n_x = 5$  state variables, and physics-based models for the combustion and for the pollutant emissions. The resulting OCP is stated in [17, 18]. The desired load torque, the bounds

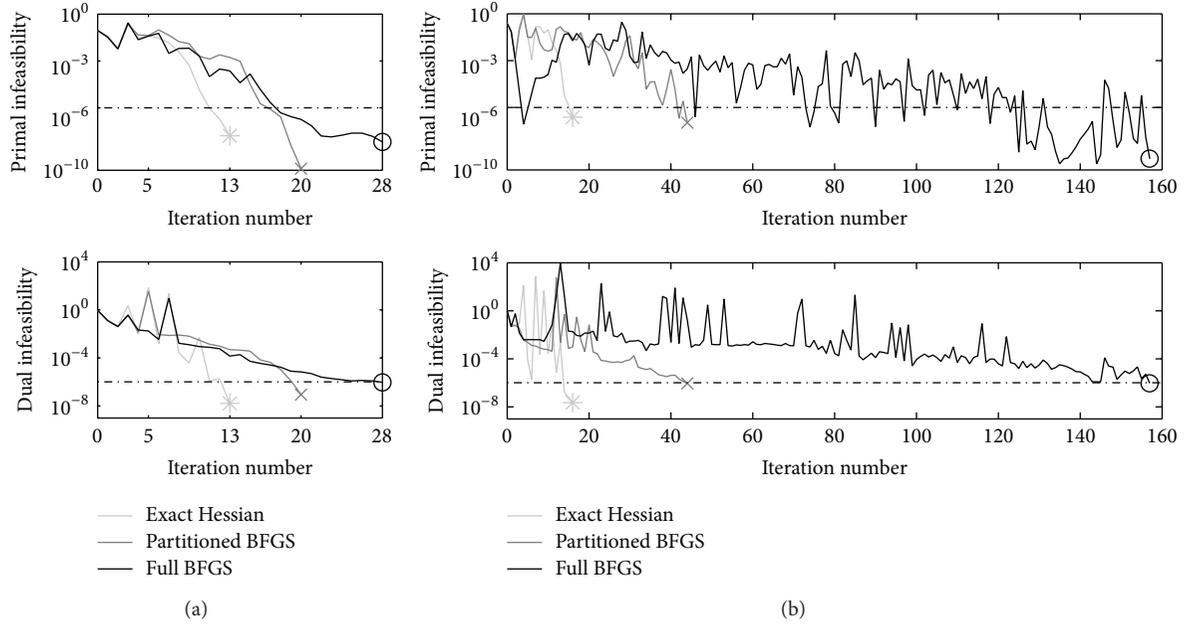


FIGURE 1: Convergence behaviour of IPOPT on the two test cases.

on the actuator ranges, and mechanical and thermal limits are imposed as nonlinear and linear path constraints (3d).

The model evaluations are expensive. Therefore, QN updates are preferable over exact Newton methods to achieve a fast solution process. The main drawback is the slow local convergence rate of QN methods when applied to the large NLPs resulting from the consideration of long time horizons in the OCP [18].

### 3. Results and Discussion

The results presented here are generated using the NLP solver IPOPT [19, 20] with the linear solver MUMPS [21] and the fill-reducing preordering implemented in METIS [22]. Either the exact Hessian, calculated using central finite differences on the model functions, or a full or the partitioned QN update just described is supplied to the solver as user-defined Hessian. In all cases, the first derivatives are calculated by forward finite differences.

Radau collocation at flipped Legendre nodes is applied. These collocation points are the roots of the orthogonal Legendre polynomials and have to be computed numerically [23, Section 2.3]. The resulting scheme, sometimes termed Radau IIA, exhibits a combination of advantageous properties [24, [25, Section 3.5].

Two test cases for the engineering problem outlined in Section 2.5 are considered. Case (a) is a mild driving pattern of 6 s duration which is discretised by first-order collocation with a uniform step length of 0.5 s. This discretisation results in  $M = 13$  total collocation points and 117 NLP variables. Test case (b) considers a more demanding driving pattern of 58 s duration and uses third-order collocation with a step size of 0.8 s, resulting in  $M = 217$  collocation points and 1,953 NLP variables.

The performance is assessed in terms of the number of iterations required to achieve the requested tolerance. Figure 1 shows the convergence behaviour of the exact-Newton method and of the two QN methods. Starting with iteration 5, the full Newton step is always accepted. Thus, the difference between the local quadratic convergence of the exact Newton method and the local superlinear convergence of the full BFGS update becomes obvious. The partitioned update performs substantially better than the full update. Moreover, the advantage becomes more pronounced when the size (longer time horizon) and the complexity (more transient driving profile, higher-order collocation) of the problem are increased from the simple test case (a) to the more meaningful case (b).

The Hessian approximation is initialised by a multiple of the identity matrix;  $\mathbf{B}_0 = \beta \mathbf{I}$ . A factor of  $\beta = 0.05$  is found to be a good choice for the problem at hand. Table 1 shows the number of iterations required as  $\beta$  is changed. The partitioned update is robust against a poor initialisation, whereas the full update requires a significant number of iterations to recover. This finding confirms that an accurate approximation is obtained in fewer iterations when the partitioned QN update is applied.

### 4. Conclusion

We illustrated the separability of the nonlinear program resulting from the application of direct collocation to an optimal control problem. Subsequently, we presented how this structure can be exploited to apply a partitioned quasi-Newton update to the Hessian of the Lagrangian. This sparsity-preserving update yields a more accurate approximation of the Hessian in fewer iterations and thus increases the convergence rate of the NLP solver.

TABLE 1: Effect of the initialisation of the Hessian approximation on the number of NLP iterations until convergence, test case (a).

Method	$\beta = 0.01$	$\beta = 0.05$	$\beta = 0.1$
Full BFGS	36	28	38
Partitioned BFGS	22	20	21

A more accurate approximation of the second derivatives from first order information is especially beneficial for highly nonlinear problems for which the exact second derivatives are expensive to evaluate. In fact, for the real-world engineering problem used as a test case here, symbolic or algorithmic differentiation is not an expedient option due to the complexity and the structure of the model. In this situation, using a quasi-Newton approximation based on first derivatives calculated by finite differences is a valuable alternative. The numerical tests presented in this paper indicate that a convergence rate close to that of an exact Newton method can be reclaimed by the application of a partitioned BFGS update.

A self-contained implementation of the partitioned update in the framework of an NLP solver itself could fully exploit the advantages of the method proposed. Furthermore, it should be assessed whether a trust-region globalisation is able to take advantage of an indefinite but possibly more accurate quasi-Newton approximation of the diagonal blocks of the Hessian of the Lagrangian.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This work was partially funded by the Swiss Innovation Promotion Agency CTI under Grant no. 10808.1 PFIW-IW.

## References

- [1] A. Griewank and A. Toint, "On the unconstrained optimization of partially separable functions," in *Nonlinear Optimization*, M. Powell, Ed., Academic Press, New York, NY, USA, 1981.
- [2] A. Griewank and P. L. Toint, "Local convergence analysis for partitioned quasi-Newton updates," *Numerische Mathematik*, vol. 39, no. 3, pp. 429–448, 1982.
- [3] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.
- [4] J. Nocedal, "Large scale unconstrained optimization," in *The State of the Art in Numerical Analysis*, pp. 311–338, Oxford University Press, New York, NY, USA, 1996.
- [5] L. T. Biegler, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation," *Computers and Chemical Engineering*, vol. 8, no. 3–4, pp. 243–247, 1984.
- [6] D. Tieu, W. R. Cluett, and A. Penlidis, "A comparison of collocation methods for solving dynamic optimization problems," *Computers and Chemical Engineering*, vol. 19, no. 4, pp. 375–381, 1995.
- [7] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 592–599, 1996.
- [8] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006.
- [9] S. Kameswaran and L. T. Biegler, "Convergence rates for direct transcription of optimal control problems using collocation at Radau points," *Computational Optimization and Applications*, vol. 41, no. 1, pp. 81–126, 2008.
- [10] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.
- [11] J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation," *SIAM Review*, vol. 46, no. 3, pp. 501–517, 2004.
- [12] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Advances in Design and Control, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2nd edition, 2010.
- [13] M. A. Patterson and A. V. Rao, "Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems," *Journal of Spacecraft and Rockets*, vol. 49, no. 2, pp. 364–337, 2012.
- [14] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, NY, USA, 2nd edition, 2006.
- [15] P. E. Gill and E. Wong, "Sequential quadratic programming methods," in *Mixed Integer Nonlinear Programming*, J. Lee and S. Leyffer, Eds., vol. 154 of *The IMA Volumes in Mathematics and its Applications*, pp. 301–312, Springer, New York, NY, USA, 2012.
- [16] J. Asprion, O. Chinellato, and L. Guzzella, "Optimisation-oriented modelling of the NO<sub>x</sub> emissions of a diesel engine," *Energy Conversion and Management*, vol. 75, pp. 61–73, 2013.
- [17] J. Asprion, C. H. Onder, and L. Guzzella, "Including drag phases in numerical optimal control of diesel engines," in *Proceedings of the 7th IFAC Symposium on Advances in Automotive Control*, pp. 489–494, Tokyo, Japan, 2013.
- [18] J. Asprion, O. Chinellato, and L. Guzzella, "Optimal control of diesel engines: numerical methods, applications, and experimental validation," *Mathematical Problems in Engineering*, vol. 2014, Article ID 286538, 21 pages, 2014.
- [19] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [20] "IPOPT homepage," 2013, <http://www.coin-or.org/Ipopt>.
- [21] "MUMPS homepage," 2013, <http://mumps.enseiht.fr>.
- [22] "METIS—serial graph partitioning and fill-reducing matrix ordering," 2013, <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>.
- [23] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods: Fundamentals in Single Domains*, Scientific Computation, Springer, Berlin, Germany, 2006.
- [24] E. Hairer and G. Wanner, "Stiff differential equations solved by Radau methods," *Journal of Computational and Applied Mathematics*, vol. 111, no. 1–2, pp. 93–111, 1999.
- [25] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, John Wiley & Sons, Winchester, UK, 2003.