A control Hamiltonian preserving discretisation for optimal control

Ashutosh Bijalwan[1], José J. Muñoz[2]

[1] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE)
Escola d'Enginyeria Barcelona Est (EEBE)
Universitat Poltièchica de Catalunya (UPC)
Av. Eduard Maristany 16, 08019
abijalwan@cimne.upc.edu

[2] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE)
Escola d'Enginyeria Barcelona Est (EEBE)
Insitut de Matemàtiques de la UPC - BarcelonaTech
Universitat Poltièchica de Catalunya (UPC)
Av. Eduard Maristany 16, 08019
j.munoz@upc.edu

Abstract

Optimal control theory allows finding the optimal input of a mechanical system modelled as a initial value problem. The resulting minimisation problem may be solved with known direct and indirect methods. We here propose time discretisations for both methods, direct midpoint (DMP) and indirect midpoint (IMP) algorithms, which despite their similarities result in different convergence orders for the adjoint (or co-state) variables. We additionally propose a third time-integration scheme, Indirect Hamiltonian Preserving (IHP) algorithm, which preserves the control Hamiltonian, an integral of the analytical Euler-Lagrange equations of the optimal control problem.

We test the resulting algorithms to linear and non-linear problems with and without dissipative forces: a propelled falling mass subjected to gravity and a drag force, an elastic inverted pendulum, and the locomotion of a worm-like organism on a frictional substrate. In order to improve the convergence of the solution process of the discretised equations in non-linear problems, we also propose a computational simple suboptimal initial guess, and apply a forward-backward sweep method, which computes each set of variables (state, adjoint and control) in a staggered manner. We demonstrate in our examples their practical advantage for computing optimal solutions.

Keywords: Optimal control, Time-discretisation, Hamiltonian preserving, Mechanical system, Adjoint method, Worm-like locomotion.

1 Introduction

Optimal control theory is a powerful decision-making tool for the controlled evolution of dynamical systems subject to constraints. This theory has a broad range of applications in engineering and natural sciences such as pandemic modelling [1, 15], aeronautics [7], or robotics and multibody systems [25], to name a few. Since system variables are optimised over a finite time interval, optimal control theory falls in the class of dynamics optimisation problem [7].

Optimal control problems (OCP) may be posed as the minimisation or maximisation of a cost functional subjected to an Ordinary Differential Equations

(ODE), with given initial conditions, i.e. as an initial value problem (IVP). The direct time discretisation of this IVP results in a Non-Linear Programming problem which can be solved with well-known numerical techniques [4]. Such discretisation-optimisation approach is the so-called direct method [31]. Alternatively, the first-order necessary conditions for stationarity of the original time dependent input give rise to Euler-Lagrange equations, which also require imposing two-point end conditions, and thus yields a two-point boundary value problem (TPBVP). The advantage of the indirect approach resides in the fact that it shows that the evolution of a dynamical system with optimal control always extremes the control Hamiltonian, a result known as Pontryagin's maximum principle [40]. This optimisation-discretisation process is known as indirect method [31]. Other solution approaches based on Hamilton-Jacobi-Bellman equation and dynamic programming have been also popular for solving OCPs (see [28] for a historical review).

In the literature, direct and indirect approaches have been successfully applied to model various classes of OCPs [4, 6]. These references and other authors [3, 8] have also shown that OCP equations have an underlying structure, where the *control* Hamiltonian is preserved in autonomous systems, and with a symplectic structure (i.e. the Hamiltonian flow in the phase space is divergence-free). Similar symmetries are well-known in Hamiltonian mechanical systems, which preserve linear momentum, angular momentum, and Hamiltonian/energy, and time discretisation schemes that preserve some of these quantities have been proposed. For instance, conservative systems discretised with midpoint rule preserve angular momenta, while the use of the discrete gradient allows also preserving the energy-momentum map [11, 35, 37]. Inspired by these results, second order structure-preserving algorithms have been proposed for solving OCPs [3, 17, 27]. Symplectic discretisations (symplectic Euler) for modelling OCPs subject to first order and second order state ODEs have been studied in [9], where it has been found that the accuracy of the symplectic method depends on the weight of regularisation parameter for control and symplectic Euler is unable to preserve the control Hamiltonian exactly.

Motivated by these works, and in order to exploit the underlying structure of the OCP, we propose in the present paper an Indirect Hamiltonian Preserving algorithm (IHP). The control Hamiltonian is an integral of the analytical solution, for linear and non-linear problems, even for dissipative, frictional or forced systems. We choose a set of simple problems that contain these features, and show that indeed the numerical solution of our algorithm preserves the control Hamiltonian in all these problem types. We compare its numerical properties with midpoint discretisations [3] that we implement in the direct and indirect approaches, and that we respectively name DMP and IMP algorithms. We pinpoint their main differences with respect to the IHP algorithm, and we also compare the order of convergence for simple linear problems with known analytical solutions.

When solving non-linear OCPs, the performance of all algorithms strongly depends on the numerical strategy. Gradient-based line search methods (e.g. Newton-Raphson, Quasi-Newton, Krylov subspace methods), shooting techniques (single or multiple [6]) and collocation methods (Galerkin, pseudospectral, Gauss-Lobatto methods) are classical methods to solve non-linear problems [31]. A major drawback of these methods is that convergence is only achieved with a good initial guess and no global minima is guaranteed. For these reason, we

also propose a strategy for computing a suboptimal initial guess, which has a minimal cost and eases the convergence of iterative processes for solving the system of non-linear equations.

Moreover, when dealing with large systems, iterative solvers for the sparse system, such as gradient descent (GD) and conjugate gradient (CG) methods, are an unavoidable choice [10, 16, 32]. We consequently also implement a forward-backward sweep method (FBSM), which exploits the structure of the Euler-Lagrange equations, as frequently used for large scale OCPs [19, 29, 36]. Instead of solving the OCP monolithically for all state, adjoint and control variables, the FBSM strategy solves at each iteration a forward and a backward (linear) ODE for respectively computing the state and adjoint variables, and updates the control variables from these solutions. We show in a non-linear problem of optimal worm locomotion that indeed, the suboptimal initial guess and the FSBM strategy allow us to compute optimal solutions in an iterative manner.

The article is organised as follows. Section 2 presents the mechanical optimisation problem and the two approaches being compared: direct and indirect approaches, jointly with the proposed IHP algorithm. Section 3 introduces numerical strategies for solving the set of non-linear equations to be solved, the initial guess and the FBSM strategy. In Section 4 we apply the methodology to some illustrative examples. Section 5 concludes with some final remarks.

2 Direct and indirect approaches of Optimal Control Problem

2.1 Optimal Control Problem

We will focus on the optimal control of a mechanical system with an input $\boldsymbol{u}(t)$ and a system state variable $\boldsymbol{x}(t)$ that aims at achieving a final target or desired value $\boldsymbol{x}_d$. For this, we consider the minimisation of cost functional $J(\boldsymbol{x}(t), \boldsymbol{u}(t))$ subject to an Initial Value Problem (IVP) in semi-implicit form:

$$\min_{\boldsymbol{u}(t),\ \boldsymbol{x}(t)} J(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{1}$$

subject to,

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ \boldsymbol{c}(\boldsymbol{x}(0)) &:= \boldsymbol{x}(0) - \bar{\boldsymbol{x}}_0 = \boldsymbol{0} \end{aligned} \tag{2}$$

where a superimposed dot denotes time derivative, $\bar{\boldsymbol{x}}_0$ is the given initial value of $\boldsymbol{x}(t)$, and function $\boldsymbol{c}(\boldsymbol{x}(0))$ is used to define the initial condition of the IVP. The cost function is defined by,

$$J(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \int_0^T \left(r(\boldsymbol{x}(t)) + q(\boldsymbol{u}(t))\right) dt + \phi(\boldsymbol{x}(T)) \tag{3}$$

with T a fixed time of interest, and the following quadratic forms for $r(\boldsymbol{x}(t))$ and $q(\boldsymbol{u}(t))$

$$r(\boldsymbol{x}(t)) = \frac{1}{2}(\boldsymbol{x}(t) - \boldsymbol{x}_d)^\mathsf{T}\mathbf{R}(\boldsymbol{x}(t) - \boldsymbol{x}_d), \tag{4a}$$

$$q(\boldsymbol{u}(t)) = \frac{\alpha}{2}\boldsymbol{u}(t)^\mathsf{T}\mathbf{Q}\boldsymbol{u}(t), \tag{4b}$$

$$\phi(\boldsymbol{x}(T)) = \frac{\beta}{2}(\boldsymbol{x}(T) - \boldsymbol{x}_d)^\mathsf{T}\mathbf{S}(\boldsymbol{x}(T) - \boldsymbol{x}_d) \tag{4c}$$

Matrices $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{S}$ are symmetric, with given constant components, and generally diagonal for simplicity. Their particular expressions depend on the problem at hand, and they are introduced to provide measures of the deviation with respect to the target $\boldsymbol{x}_d$ and input control $\boldsymbol{u}(t)$. Parameters $\alpha, \beta \geq 0$ are defined as regularisation factors that control the magnitude of the input $\boldsymbol{u}(t)$ and terminal cost, respectively. Figure 1 shows an example where the optimisation problem in (1)-(2) could be applied to, and the meaning of the variables. However, in order to highlight our proposed methodology, we will resort to simpler applications with fewer degrees of freedom (dofs).

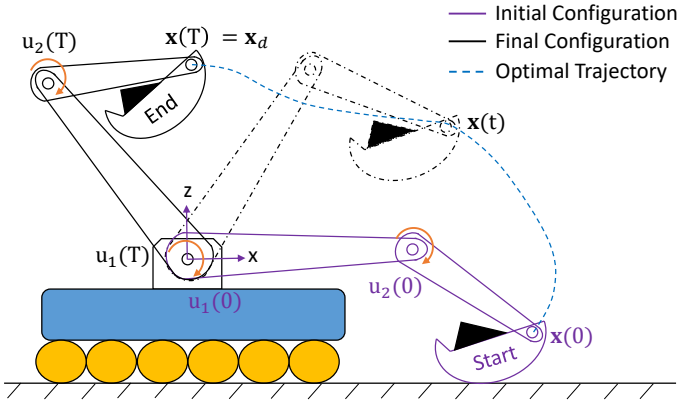


Figure 1: Example of optimal control of a mechanical system.

Remark 1. We have used for simplicity a 1st order IVP, but 2nd order IVPs may be equally represented by the form given in (2) and employed in the methodology to be presented in this work, with a proper transformation and extension of the number of states, as it is customary in mechanics.

In order to solve the Optimal Control Problem with non-linear IVPs, we will next analyse the numerical solution of discretised forms of the minimisation problem above, resorting to two common strategies: direct and indirect approaches [27].

2.2 Direct method (Discretise-differentiate)

2.2.1 Discretisation

In the direct approach, we first discretise the cost functional in Eqn. (1) and the ordinary differential equation (ODE) in the IVP with $N + 1$ time points

$t_0, \ldots, t_N$, and approximated state and control values $\boldsymbol{x}_n \approx \boldsymbol{x}(t_n)$, $\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$, $n = 0, \ldots, N$. We will resort to a midpoint integration scheme of the ODE, although other schemes are also possible [21, 20, 4]. Our choice is motivated by the subsequent comparison with indirect methods in Section 2.3. After applying the integration scheme, the time continuous optimisation problem turns into the following Non-linear Programming problem:

$$\min_{\boldsymbol{u},\boldsymbol{x}} J_\Delta(\boldsymbol{x}, \boldsymbol{u}) \tag{5}$$

subject to,

$$\frac{\boldsymbol{x}_n - \boldsymbol{x}_{n-1}}{\Delta t} = \boldsymbol{f}_{(n-\frac{1}{2})}, n = 1, \ldots, N$$
$$\boldsymbol{c}(\boldsymbol{x}_0) := \boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0 = \boldsymbol{0}$$

Here and in subsequent expression we use the symbol $\Delta(\bullet) = (\bullet)_n - (\bullet)_{n-1}$. Also, we have introduced the control vector $\boldsymbol{u} = \{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_N\}^T$ and state vector $\boldsymbol{x} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N\}^T$, which define the solution of the non-linear problem. The right hand side of the ODE has been replaced by $\boldsymbol{f}_{(n-\frac{1}{2})} = \boldsymbol{f}(\boldsymbol{x}_{n-\frac{1}{2}}, \boldsymbol{u}_{n-\frac{1}{2}})$, with $(\bullet)_{n-\frac{1}{2}} = \frac{1}{2}\left((\bullet)_n + (\bullet)_{n-1}\right)$, and the cost functional has been replaced by a discretised counterpart with a rectangular rule,

$$J_\Delta(\boldsymbol{x}, \boldsymbol{u}) = \sum_{n=1}^{N} \Delta t \left(r(\boldsymbol{x}_n) + q(\boldsymbol{u}_n)\right) + \phi(\boldsymbol{x}_N) \tag{6}$$

2.2.2 Differentiation

The optimal values of $\boldsymbol{x}$ and $\boldsymbol{u}$ are obtained by building the extended Lagrangian with a vector of Lagrange multipliers (or co-state/adjoint variables) $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_1 \ldots \boldsymbol{\lambda}_N\}$ [26]

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = J_\Delta + \sum_{n=1}^{N} \boldsymbol{\lambda}_n^T \left(\boldsymbol{f}_{(n-\frac{1}{2})} - \frac{\Delta \boldsymbol{x}}{\Delta t}\right) \tag{7}$$

and deriving the optimality conditions from the stationarity of the Lagrangian, i.e. $\partial_{\boldsymbol{x}_n}\mathcal{L} = \boldsymbol{0}$, $\partial_{\boldsymbol{u}_n}\mathcal{L} = \boldsymbol{0}$, $\partial_{\boldsymbol{\lambda}_n}\mathcal{L} = \boldsymbol{0}$, which yields the equations in Box 1. We will call this algorithm the direct midpoint (DMP). Note that since no Lagrange multiplier at time t_{N+1} exist, the DMP algorithm at end time ($n = N$) is given by

$$\Delta t \mathbf{R}(\boldsymbol{x}_N - \boldsymbol{x}_d) + \left(\frac{\partial \boldsymbol{f}_{(N-\frac{1}{2})}}{\partial_{\boldsymbol{x}_N}}\right)^{\mathsf{T}} \boldsymbol{\lambda}_N - \frac{\boldsymbol{\lambda}_N}{\Delta t} + \frac{\partial \phi}{\partial \boldsymbol{x}_N} = \boldsymbol{0},$$
$$\frac{\boldsymbol{x}_N - \boldsymbol{x}_{N-1}}{\Delta t} = \boldsymbol{f}_{(N-\frac{1}{2})},$$
$$\Delta t \alpha \mathbf{Q} \boldsymbol{u}_N + \left(\frac{\partial \boldsymbol{f}_{(N-\frac{1}{2})}}{\partial \boldsymbol{u}_N}\right)^{\mathsf{T}} \boldsymbol{\lambda}_N = \boldsymbol{0}.$$

The solution of the equations in the DMP algorithm requires in general resorting to iterative numerical strategies, overall for non-linear forms of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$. Section 3 discusses some procedures based on conjugate gradient and gradient descent, and strategies for computing the initial guess.

Direct midpoint algorithm (DMP)

$$\Delta t\mathbf{R}(\boldsymbol{x}_n - \boldsymbol{x}_d) + \left(\frac{\partial \boldsymbol{f}_{(n-\frac{1}{2})}}{\partial \boldsymbol{x}_n}\right)^{\mathsf{T}} \boldsymbol{\lambda}_n + \left(\frac{\partial \boldsymbol{f}_{(n+\frac{1}{2})}}{\partial \boldsymbol{x}_n}\right)^{\mathsf{T}} \boldsymbol{\lambda}_{n+1} = \frac{\boldsymbol{\lambda}_n - \boldsymbol{\lambda}_{n+1}}{\Delta t},$$

$$\frac{\boldsymbol{x}_n - \boldsymbol{x}_{n-1}}{\Delta t} = \boldsymbol{f}_{(n-\frac{1}{2})}$$

$$\Delta t\alpha\mathbf{Q}\boldsymbol{u}_n + \left(\frac{\partial \boldsymbol{f}_{(n-\frac{1}{2})}}{\partial \boldsymbol{u}_n}\right)^{\mathsf{T}} \boldsymbol{\lambda}_n + \left(\frac{\partial \boldsymbol{f}_{(n+\frac{1}{2})}}{\partial \boldsymbol{u}_n}\right)^{\mathsf{T}} \boldsymbol{\lambda}_{n+1} = \mathbf{0},$$

for $n = 1, \ldots, N-1$, and with

$\boldsymbol{c}(\boldsymbol{x}(0)) = \mathbf{0}$

Box 1: Direct approach when discretising the ODE with a midpoint scheme.

2.3 Indirect method (differentiate-Discretised)

2.3.1 Time continuous optimality conditions

In the indirect approach, we first deduce the optimality conditions, which in fact form a system of ODEs that will be eventually discretised [9, 7]. For this, we introduce time-varying Lagrangian multipliers ($\boldsymbol{\lambda}(t)$ and $\boldsymbol{\xi}(t)$) and define the augmented Lagrangian functional associated to the optimisation problem in (1)-(2),

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{x}(t), \boldsymbol{u}(t); \boldsymbol{\lambda}(t), \boldsymbol{\xi}\right) := &\int_0^T \Big(r(\boldsymbol{x}(t)) + q(\boldsymbol{u}(t)) + \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))\Big)\, dt \\ &- \int_0^T \boldsymbol{\lambda}(t)^{\mathsf{T}} \dot{\boldsymbol{x}} dt + \boldsymbol{\xi}^{\mathsf{T}} \boldsymbol{c}(\boldsymbol{x}(0)) + \phi(\boldsymbol{x}(T)).\end{aligned} \tag{8}$$

In control theory, the integrand of left most integral in Eqn. (8) is defined as *control Hamiltonian* [38, 23]

$$\mathcal{H}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\lambda}(t)\right) := r(\boldsymbol{x}(t)) + q(\boldsymbol{u}(t)) + \boldsymbol{\lambda}(t)^{\mathsf{T}} \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)). \tag{9}$$

Using integration by parts and the definition of control Hamiltonian, the functional $\mathcal{L}$ can be rewritten as

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\lambda}(t), \boldsymbol{\xi}) = &\int_0^T \Big(\mathcal{H}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\lambda}(t)) + \dot{\boldsymbol{\lambda}}^{\mathsf{T}} \boldsymbol{x}\Big)\, dt \\ &- \boldsymbol{\lambda}(T)^{\mathsf{T}} \boldsymbol{x}(T) + \boldsymbol{\lambda}(0)^{\mathsf{T}} \boldsymbol{x}(0) + \boldsymbol{\xi}^{\mathsf{T}} \boldsymbol{c}(\boldsymbol{x}(0)) + \phi(\boldsymbol{x}(T))\end{aligned} \tag{10}$$

The stationarity of the optimal solution requires the solution of the following set of ODEs [7],

$$\dot{\boldsymbol{\lambda}} = -\nabla_{\boldsymbol{x}} \mathcal{H} \tag{11a}$$

$$\dot{\boldsymbol{x}} = \nabla_{\boldsymbol{\lambda}} \mathcal{H} \tag{11b}$$

$$\mathbf{0} = \nabla_{\boldsymbol{u}} \mathcal{H} \tag{11c}$$

where we have used the notation $\nabla_{(\bullet)} = \frac{\partial}{\partial (\bullet)}$. The differential-algebraic equations (DAE) in (11) are respectively called *adjoint*, *governing* and *control* equations,

or *Euler-Lagrange* equations [38]. These must be solved jointly with the two-point boundary conditions:

$$\boldsymbol{c}(\boldsymbol{x}(0)) = \mathbf{0},\ \boldsymbol{\lambda}(T) = \frac{\partial \phi}{\partial \boldsymbol{x}(T)}. \tag{12}$$

In case that the final time T is unknown, this parameter can be treated as an additional input variable, which can be determined from the addition equations $\frac{\partial \mathcal{L}}{\partial T} = 0$ [7]. The system of coupled ODE (11) along with boundary conditions in (12) forms a two-point boundary value problem (TPBVP). We note that after defining the variable $\boldsymbol{z}(t) = \{\boldsymbol{x}(t),\ \boldsymbol{\lambda}(t)\}$, the Euler-Lagrange equations can be alternatively written as

$$\begin{aligned} \dot{\boldsymbol{z}} &= \mathbf{J}\nabla_{\boldsymbol{z}}\mathcal{H}, \\ \mathbf{0} &= \nabla_{\boldsymbol{u}}\mathcal{H}, \end{aligned}$$

where $\mathbf{J}$ is the canonical symplectic matrix:

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix}.$$

The form above reveals the symplectic structure of the solution, even in the presence of a dissipative or forced systems. This has motivated the design of structure-preserving numerical schemes [9, 3]. In this work we aim to exploit an integral of the optimal control problem [7, 14]:

Proposition 1. *For an autonomous dynamical system (independent of time t), control Hamiltonian $\mathcal{H}$ is a first integral of the motion.*

Proof. For an autonomous dynamical system we have that $\frac{\partial \mathcal{H}}{\partial t} = 0$, and therefore the preservation of the total time derivative of the control Hamiltonian $\mathcal{H}\left(\boldsymbol{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{u}(t)\right)$ follows directly from the Euler-Lagrange equations in (11),

$$\dot{\mathcal{H}} = \nabla_{\boldsymbol{x}}\mathcal{H}^{\mathsf{T}}\dot{\boldsymbol{x}} + \nabla_{\boldsymbol{\lambda}}\mathcal{H}^{\mathsf{T}}\dot{\boldsymbol{\lambda}} + \nabla_{\boldsymbol{u}}\mathcal{H}^{\mathsf{T}}\dot{\boldsymbol{u}} = -\dot{\boldsymbol{\lambda}}^{T}\dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^{T}\dot{\boldsymbol{\lambda}} + \mathbf{0}^{\mathsf{T}}\dot{\boldsymbol{u}} = 0.$$

□

This result motivates the discretisation proposed in the next Section.

2.3.2 Time discretisation of continuous system

Let us first consider the application of a midpoint rule on the governing and adjoint equations in (11). This results in the indirect midpoint (IMP) algorithm indicated in Box 2.

Motivated by the analytical result in Proposition 1, that is, in order to obtain a discretisation that conserves the discrete counterpart of the Hamiltonain,

$$\mathcal{H}_n = r(\boldsymbol{x}_n) + q(\boldsymbol{u}_n) + \boldsymbol{\lambda}_n^{T}\boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{u}_n) \tag{13}$$

the algorithm in Box 2 is modified and transformed in the Indirect Hamiltonian Preserving algorithm in Box 3. It is based on the definition of a discrete derivative in [11] for function $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ in the governing ODE, similarly to other energy conserving schemes [3, 17].

Indirect midpoint algorithm (IMP)

$$\mathbf{R}\left(\boldsymbol{x}_{n-\frac{1}{2}}-\boldsymbol{x}_d\right)+\left(\frac{\partial \boldsymbol{f}_{(n-\frac{1}{2})}}{\partial \boldsymbol{x}_n}\right)^{\mathsf{T}}\boldsymbol{\lambda}_{n-\frac{1}{2}}+\frac{\boldsymbol{\lambda}_n-\boldsymbol{\lambda}_{n-1}}{\Delta t}=\mathbf{0}$$

$$\frac{\boldsymbol{x}_n-\boldsymbol{x}_{n-1}}{\Delta t}=\boldsymbol{f}_{(n-\frac{1}{2})}$$

$$\alpha\mathbf{Q}\boldsymbol{u}_{n-\frac{1}{2}}+\left(\frac{\partial \boldsymbol{f}_{(n-\frac{1}{2})}}{\partial \boldsymbol{u}_n}\right)^{\mathsf{T}}\boldsymbol{\lambda}_{n-\frac{1}{2}}=\mathbf{0}$$

for $n=1,\ldots,N-1$, and with

$$\boldsymbol{c}(\boldsymbol{x}(0))=\mathbf{0},\ \boldsymbol{\lambda}_N=\nabla_{\boldsymbol{x}_N}\phi(\boldsymbol{x}_N)$$

Box 2: Discretisation proposed for the Euler-Lagrange equations in the indirect approach and using a midpoint rule.

Indirect Hamiltonian Preserving (IHP)

$$\mathbf{R}\left(\boldsymbol{x}_{n-\frac{1}{2}}-\boldsymbol{x}_d\right)+\overline{\nabla}_{\boldsymbol{x}}\boldsymbol{f}^{\mathsf{T}}\boldsymbol{\lambda}_{n-\frac{1}{2}}+\frac{\boldsymbol{\lambda}_n-\boldsymbol{\lambda}_{n-1}}{\Delta t}=\mathbf{0}$$

$$\frac{\boldsymbol{x}_n-\boldsymbol{x}_{n-1}}{\Delta t}=\boldsymbol{f}_{n-\frac{1}{2}}$$

$$\alpha\mathbf{Q}\boldsymbol{u}_{n-\frac{1}{2}}+\overline{\nabla}_{\boldsymbol{u}}\boldsymbol{f}^{\mathsf{T}}\boldsymbol{\lambda}_{n-\frac{1}{2}}=\mathbf{0}$$

$$\alpha\mathbf{Q}\boldsymbol{u}_N+\nabla_{\boldsymbol{u}}\boldsymbol{f}^{\mathsf{T}}\boldsymbol{\lambda}_N=\mathbf{0}$$

for $n=1,\ldots,N-1$, and with

$$\boldsymbol{c}(\boldsymbol{x}(0))=\mathbf{0},\ \boldsymbol{\lambda}_N=\nabla_{\boldsymbol{x}_N}\phi(\boldsymbol{x}_N)$$

using definitions

$$\overline{\nabla}_{\boldsymbol{x}}\boldsymbol{f}=\nabla_{\boldsymbol{x}}\boldsymbol{f}_{(n-\frac{1}{2})}+\Delta\boldsymbol{f}\otimes\frac{\Delta\boldsymbol{x}}{\Delta\boldsymbol{x}^{\mathsf{T}}\Delta\boldsymbol{x}+\Delta\boldsymbol{u}^{\mathsf{T}}\Delta\boldsymbol{u}}-\left(\nabla_{\boldsymbol{x}}\boldsymbol{f}_{(n-\frac{1}{2})}\Delta\boldsymbol{x}\right)\otimes\frac{\Delta\boldsymbol{x}}{\Delta\boldsymbol{x}^{\mathsf{T}}\Delta\boldsymbol{x}}$$

$$\overline{\nabla}_{\boldsymbol{u}}\boldsymbol{f}=\nabla_{\boldsymbol{u}}\boldsymbol{f}_{(n-\frac{1}{2})}+\Delta\boldsymbol{f}\otimes\frac{\Delta\boldsymbol{u}}{\Delta\boldsymbol{x}^{\mathsf{T}}\Delta\boldsymbol{x}+\Delta\boldsymbol{u}^{\mathsf{T}}\Delta\boldsymbol{u}}-\left(\nabla_{\boldsymbol{u}}\boldsymbol{f}_{(n-\frac{1}{2})}\Delta\boldsymbol{u}\right)\otimes\frac{\Delta\boldsymbol{u}}{\Delta\boldsymbol{u}^{\mathsf{T}}\Delta\boldsymbol{u}}$$

$$\Delta\boldsymbol{x}=\boldsymbol{x}_n-\boldsymbol{x}_{n-1},\ \Delta\boldsymbol{u}=\boldsymbol{u}_n-\boldsymbol{u}_{n-1},\ \Delta\boldsymbol{f}=\boldsymbol{f}_n-\boldsymbol{f}_{n-1}$$

Box 3: Discretisation proposed for the Euler-Lagrange equations in the indirect approach and using discrete gradient $\overline{\nabla}$.

We note that the discrete $\overline{\nabla}$ gradient has the following important property,

$$\overline{\nabla}_{\boldsymbol{x}}\boldsymbol{f}\Delta\boldsymbol{x}+\overline{\nabla}_{\boldsymbol{u}}\boldsymbol{f}\Delta\boldsymbol{u}=\Delta\boldsymbol{f}, \tag{14}$$

which will allow us to prove the following result:

Proposition 2. *If the input and output cost have the quadratic forms in* (4)*, and the Euler-Lagrange equations are discretised with scheme in Box 3, then the discrete control Hamiltonian is algorithmically preserved, i.e.* $\Delta\mathcal{H}=0$.

Proof. From the expression of the control Hamiltonian at time t_n in (13), its increment between consecutive times-steps is given by,

$$\Delta\mathcal{H}=\Delta\boldsymbol{x}^{\mathsf{T}}\mathbf{R}\left(\boldsymbol{x}_{n-\frac{1}{2}}-\boldsymbol{x}_d\right)+\Delta\boldsymbol{u}^{\mathsf{T}}\alpha\mathbf{Q}\boldsymbol{u}_{n-\frac{1}{2}}+\Delta\boldsymbol{f}^{\mathsf{T}}\boldsymbol{\lambda}_{n-\frac{1}{2}}+\Delta\boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{f}_{n-\frac{1}{2}}.$$

Substituting the discretised adjoint, governing and control equations in Box 3 into the previous expression yields

$$\Delta\mathcal{H} = \boldsymbol{\lambda}_{n-\frac{1}{2}}^{\mathsf{T}} \left(\Delta \boldsymbol{f} - \overline{\nabla}_{\boldsymbol{x}} \boldsymbol{f} \Delta \boldsymbol{x} - \overline{\nabla}_{\boldsymbol{u}} \boldsymbol{f} \Delta \boldsymbol{u} \right). \tag{15}$$

From the property of the discrete gradient in (14), the expression in the parenthesis vanishes and the preservation of the Hamiltonian follows. □

Remark 2. The DMP, IMP and IHP algorithms in Boxes 1, 2 and 3, respectively, differ on the time-stepping considered, but are both consistent, in the sense that converge to the Euler-Lagrange equations in (11) as $\Delta t \to 0$. Moreover, algorithms IMP and IHP coincide for linear governing ODEs, i.e. for a bilinear function $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, and DMP differs in this case on the first term of the first (adjoint) equation and end condition for $\boldsymbol{\lambda}_N$, which is missing.

3 Numerical Solution

In linear problems, the discretised Euler-Lagrange equations have the structure of a sparse saddle-point problem, which can be solved with known iterative methods [2]. In non-linear problems instead, the equations form a set of non-linear equations, which require specific iterative techniques and special choice of an initial guess. We will discuss these issues in the next paragraphs.

3.1 Initial guess

In order to obtain an initial guess that is computationally not too expensive, but that is also not too far from the exact numerical solution, we propose a partial optimisation of the direct approach, where at each time t_n, only the current variables $\boldsymbol{x}_n$ and $\boldsymbol{u}_n$ are computed, maintaining all the other variables fixed, and that minimises the objective functional $J_n(\boldsymbol{x}_n, \boldsymbol{u}_n) = \Delta t(r(\boldsymbol{x}_n) + q(\boldsymbol{u}_n))$. This is tantamount to solving the following N problems ($n = 1, \ldots, N$):

$$\begin{aligned} &\min_{\boldsymbol{u}_n, \boldsymbol{x}_n} J_n \\ &\text{subject to,} \\ &\qquad \frac{\boldsymbol{x}_n - \boldsymbol{x}_{n-1}}{\Delta t} = \boldsymbol{f}_{(n-\frac{1}{2})}, \\ &\qquad \boldsymbol{c}(\boldsymbol{x}(0)) = \boldsymbol{0} \end{aligned} \tag{16}$$

By building the Lagrangian function,

$$\mathcal{L}_n(\boldsymbol{x}_n, \boldsymbol{\lambda}_n, \boldsymbol{u}_n) = J_n(\boldsymbol{x}_n, \boldsymbol{u}_n) + \boldsymbol{f}_{(n-\frac{1}{2})}^{\mathsf{T}} \boldsymbol{\lambda}_n \tag{17}$$

the KKT conditions with respect to unknowns $\boldsymbol{x}_n$, $\boldsymbol{\lambda}_n$ and $\boldsymbol{u}_n$ ($n = 1, \ldots, N$) read:

$$\Delta t \mathbf{R}(\boldsymbol{x}_n - \boldsymbol{x}_d) + \left(\frac{\partial \boldsymbol{f}_{(n-\frac{1}{2})}}{\partial \boldsymbol{x}_n} \right)^{\mathsf{T}} \boldsymbol{\lambda}_n - \frac{1}{\Delta t} \boldsymbol{\lambda}_n = \boldsymbol{0} \tag{18a}$$

$$\boldsymbol{f}_{(n-\frac{1}{2})} = \frac{\Delta \boldsymbol{x}}{\Delta t} \tag{18b}$$

$$\Delta t \mathbf{Q} \boldsymbol{u}_n + \left(\frac{\partial \boldsymbol{f}_{(n-\frac{1}{2})}}{\partial \boldsymbol{u}_n} \right)^{\mathsf{T}} \boldsymbol{\lambda}_n = \boldsymbol{0} \tag{18c}$$

Note that they are very similar to those in Box 1, but removing variable $\boldsymbol{\lambda}_{n+1}$. Once we solve these set of equations, we will get $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ at all time points. Of course, such solution is suboptimal, but is has a very minor cost, and in some of our numerical examples has been a better strategy than using for instance the initial guess $\boldsymbol{u} = \mathbf{0}$.

3.2 Iterative solution strategy

For moderate size problems, Newton-Raphson or quasi-Newton methods are applicable for achieving convergence of the set of non-linear discrete Euler-Lagrange equations. However, when the time step size Δt becomes small, the problem dimensions may increase substantially, resulting in too large systems that are difficult to converge. In these cases, iterative methods become a useful choice. We have resorted to the following forward-backward sweep method (FBSM) [22, 36] to solve the optimal control problem iteratively. The solution procedure is as follows:

1. **Initial guess**: Give or compute initial guess $\boldsymbol{u}^0$.

2. **Forward time marching:** Compute state variables $\boldsymbol{x}^{k+1}$ from $\boldsymbol{u}^k$ and $\boldsymbol{\lambda}^k$. Solve the discrete governing ODE,

$$\frac{\boldsymbol{x}_n - \boldsymbol{x}_{n-1}}{\Delta t} = \boldsymbol{f}(\boldsymbol{x}_{n-\frac{1}{2}}, \boldsymbol{u}_{n-\frac{1}{2}}), \quad n = 1, 2, \ldots, N,$$

   with initial condition $\boldsymbol{c}(\boldsymbol{x}(0)) = \mathbf{0}$. In non-linear problems, function $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ is not bilinear, and therefore a Newton-Raphson procedure may be employed. In the IHP algorithm, the right hand side should be replaced by $\boldsymbol{f}_{n-\frac{1}{2}}$.

3. **Back substitution:** Compute adjoint variables $\boldsymbol{\lambda}^{k+1}$ from $\boldsymbol{u}^k$ and $\boldsymbol{x}^{k+1}$. Solve discretised adjoint equation (first equation in Box 1 and 3 for the DMP and IHP, respectively), with the final condition $\boldsymbol{\lambda}_N = \frac{\partial \phi(\boldsymbol{x}_N)}{\partial \boldsymbol{x}_N}$. Since $\boldsymbol{x}$ has been obtained in the previous step, these equations allow computing $\boldsymbol{\lambda}_{n-1}$ from $\boldsymbol{\lambda}_n$. Note that these equations are always linear with respect to $\boldsymbol{\lambda}$. For instance, in the IHP algorithm, the following recursive equation should be used:

$$\boldsymbol{\lambda}_{n-1} = -\left(\overline{\nabla}_{\boldsymbol{x}} \boldsymbol{f}^{\mathsf{T}} - \frac{1}{\Delta t}\mathbf{I}\right)^{-1} \left(\mathbf{R}\left(\boldsymbol{x}_{n-\frac{1}{2}} - \boldsymbol{x}_d\right) + \left(\overline{\nabla}_{\boldsymbol{x}} \boldsymbol{f}^{\mathsf{T}} + \frac{1}{\Delta t}\mathbf{I}\right)\boldsymbol{\lambda}_n\right).$$

4. **Update control variable**: Compute $\boldsymbol{u}^{k+1}$ from state and adjoint variables $\boldsymbol{x}^{k+1}$ and $\boldsymbol{\lambda}^{k+1}$ computed in Steps 2 and 3. Apply one iteration of conjugate gradient (CG) or gradient descent (GD) for solving the discrete control equation $\nabla_u \mathcal{H} = \mathbf{0}$.

Steps 2 and 3 allow expressing the objective functional solely as function of (admissible) control values ($\hat{J}(\boldsymbol{u}) = J(\boldsymbol{x}(\boldsymbol{u}), \boldsymbol{u})$). Therefore, for solving the control equation in Step 4, we can compute the directional derivative of the objective functional along $\boldsymbol{u}$ as ($\delta \boldsymbol{u} = \epsilon \boldsymbol{p}$, with $\boldsymbol{p}$ a search direction)

$$\delta \hat{J}(\boldsymbol{u}) = \left.\frac{\hat{J}(\boldsymbol{u} + \epsilon \boldsymbol{p}) - \hat{J}(\boldsymbol{u})}{\epsilon}\right|_{\epsilon=0} = \int_0^T \nabla_{\boldsymbol{u}} \mathcal{H}^{\mathsf{T}} \delta \boldsymbol{u} dt.$$

Maximum reduction in the functional value can be assured if $\delta \boldsymbol{u}$ is chosen opposite to the direction of $\nabla_{\boldsymbol{u}} \mathcal{H}$ [19, 25, 29]. The control equations (third equation in Box 1 and 3 for the DMP and IHP algorithms respectively) measure $\boldsymbol{r}_n = \nabla_{\boldsymbol{u}_n} \mathcal{H}$. For instance, in the IHP approach this residual is given by

$$\begin{aligned} \boldsymbol{r}_{n-1} &= \alpha \mathbf{Q} \boldsymbol{u}_{n-\frac{1}{2}} + \overline{\nabla}_{\boldsymbol{u}} \boldsymbol{f}^{\mathsf{T}} \boldsymbol{\lambda}_{n-\frac{1}{2}}, \ n = 1, 2, \ldots, N-1, \\ \boldsymbol{r}_N &= \alpha \mathbf{Q} \boldsymbol{u}_N + \nabla_{\boldsymbol{u}} \boldsymbol{f}^{\mathsf{T}} \boldsymbol{\lambda}_N, \end{aligned}$$

and we set the residual vector $\boldsymbol{r} = \{\boldsymbol{r}_0, \ldots, \boldsymbol{r}_N\}^T$. Consequently, in Step 4, and at each iteration k ,we update the control variable as,

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \theta^k \boldsymbol{p}_k$$

with $\boldsymbol{p}_k$ a search direction and θ^k the step length at the k-th iteration. In inexact line search algorithm, one can use Armijo rule with reducing step length ($\theta^k \rightarrow \theta^k/2$) until acceptance criteria achieved, i.e. $J(\boldsymbol{u}^k + \theta^k \boldsymbol{p}_k) < J(\boldsymbol{u}_k) - \gamma \theta^k \boldsymbol{p}_k^{\mathsf{T}} \boldsymbol{r}^k$, $\gamma \in (10^{-4}, 0.5)$. To avoid too many backtracking/bisection during line search, Armijo rule with curvature conditions ($|\boldsymbol{p}_k^{\mathsf{T}} \boldsymbol{r}^{k+1}| \leq \sigma |\boldsymbol{p}_k^{\mathsf{T}} \boldsymbol{r}^k|$, $\sigma \in (10^{-4}, 0.1)$) results in optimum step length (Wolfe conditions) [26]. Although the selection of γ and σ combination depends on the optimisation problem, one must ensure that there is a desired decrease in the cost functional with iterations and that the step length is not too small or big to avoid convergence problems in time integration of ODE (step 2) [16]. When using $\theta_{max} = \frac{1}{||\boldsymbol{p}_k||}$ as the initial estimate of θ^k along $\boldsymbol{p}_k$, the rate of convergence is faster than using unit step size. In case that line search algorithm results in no suitable step length, we move on by taking a small step length ($\theta^k = \theta_{min}$) along $\boldsymbol{p}_k$. Search direction has been used to update according to (with the initial search direction $\boldsymbol{p}_0 = -\boldsymbol{r}^0$)

$$\boldsymbol{p}_{k+1} = -\boldsymbol{r}^{k+1} + \tilde{\beta} \boldsymbol{p}_k,$$

using the Fletcher-Ribier parameter $\tilde{\beta} = \frac{||\boldsymbol{r}^{k+1}||}{||\boldsymbol{r}^k||}$ for CG [10] and $\tilde{\beta} = 0$ for GD.

4 Numerical Examples

In this section we compare the behaviour of the DMP and IHP algorithms in mechanical systems governed by linear and non-linear ODEs. The first example under consideration analyses the optimal control of a particle in linear and non-linear viscous medium. In the second example, we will study the motion of a two-particle system with a non-linear elastic spring. Finally, we model the optimal locomotion of planar worm (flexible body system) subjected to elastic forces and viscous friction. In all algorithms, we use $\beta = 0$ in the definition of $\phi(\boldsymbol{x}(T))$ in (4), unless stated otherwise. Convergence is assumed when the L^2 norm of the error in the global residual and the primary field iterative changes fall below the tolerance of 10^{-10}.

4.1 Falling particle in viscous medium

4.1.1 Linear problem

We consider the rectilinear motion of a particle with mass m and subjected to a drag force proportional and antiparallel to the velocity v along z direction.

During the motion, particle is under influence of the gravitational acceleration with intensity a and a controlling vertical force u. Velocity v is our state variable, and thus has the role of variable $\boldsymbol{x}$ in our previous derivations. Consequently, a cost functional

$$J(v,u) = \int_0^T (r(v) + q(u))dt, \tag{19a}$$

will be minimised with

$$r(v) = \frac{1}{2}(v - v_d)^2, \quad q(u) = \frac{\alpha}{2}u^2 \tag{19b}$$

and $v_d \in \mathbb{R}$ a target velocity. Gravitational potential energy of the particle is given by $U(z) = maz$ and Rayleigh dissipation induces an external force equal to $-bv$. Equation of motion is then given by the following balance equation,

$$\dot{v} = f(v,u), \tag{20}$$

with

$$f(v,u) = -\frac{b}{m}v - a + \frac{u}{m},$$

and $(\alpha, m, b, a) \in \mathbb{R}^+$, and initial state $v(0) = v_o$.

The analytical solution can be deduced from the Euler-Lagrange equations in (11) with the control Hamiltonian $\mathcal{H}(v,\lambda,u) = r(v) + q(u) + f(v,u)\lambda$,

$$\dot{\lambda} + \frac{\partial \mathcal{H}}{\partial v} = 0 \Rightarrow \dot{\lambda} + v - \frac{b}{m}\lambda - v_d = 0, \tag{21a}$$

$$\dot{v} - \frac{\partial \mathcal{H}}{\partial \lambda} = 0 \Rightarrow \dot{v} + \frac{b}{m}v + a - \frac{u}{m} = 0, \tag{21b}$$

$$\frac{\partial \mathcal{H}}{\partial u} = 0 \Rightarrow u = -\frac{\lambda}{\alpha m}, \tag{21c}$$

with boundary conditions $v(0) = v_o$ and $\lambda(T) = 0$. Substituting Eqn. (21c) into Eqn. (21b), the following system of coupled linear first-order non-homogeneous differential equations is derived,

$$\dot{\lambda} - \frac{b}{m}\lambda + v - v_d = 0, \tag{22a}$$

$$\dot{v} + \frac{b}{m}v + \frac{\lambda}{\alpha m^2} + a = 0. \tag{22b}$$

Solution of Eqn. (22) can be written as a linear combination of eigenvectors corresponding to eigenvalues $\pm\gamma$. Finally, state, co-state and optimal control trajectories take the form

$$v(t) = C_1\left(\frac{b}{m} - \gamma\right)e^{\gamma t} + C_2\left(\frac{b}{m} + \gamma\right)e^{-\gamma t} + v_p \tag{23a}$$

$$\lambda(t) = C_1 e^{\gamma t} + C_2 e^{-\gamma t} + \lambda_p \tag{23b}$$

$$u(t) = -\frac{\lambda(t)}{\alpha m} \tag{23c}$$

where

$$\gamma = \frac{1}{m}\sqrt{\frac{1}{\alpha} + b^2};\ v_p = \frac{v_d - \alpha bma}{\alpha m^2\gamma^2};\ \lambda_p = -\frac{bv_d + ma}{m\gamma^2},$$
$$C_1 = \frac{m\,(v_o - v_p)\,e^{-\gamma T} + (b + m\gamma)\,\lambda_p}{(b - m\gamma)\,e^{-\gamma T} - (b + m\gamma)\,e^{\gamma T}};\ C_2 = \frac{m\,(v_o - v_p) - (b - m\gamma)\,C_1}{b + m\gamma}$$

In this study, system dynamics is analysed with parameters $\Delta t = 0.1$ s, $T = 10$ s, $m = 1$ kg, $b = 1$ Ns/m, $a = 1$ m/s^2, $v_o = 0$ m/s, $v_d = 20$ m/s and α values are varied from 10^{-2} to 10^2. It can be seen that the particular solution (v_p, λ_p) is the equilibrium/stationary point of the dynamical system in Eqn. (22). Since stationary point is a saddle point (and is unstable), particle spends significant amount of total time near this point. From analytical solution, we can also deduce that as α increases, (v_p, λ_p) approaches (v_o, λ_o), while for lower α we have that (v_p, λ_p) approaches (v_d, λ_T). The results in Fig 2a-2b confirm also this behaviour.

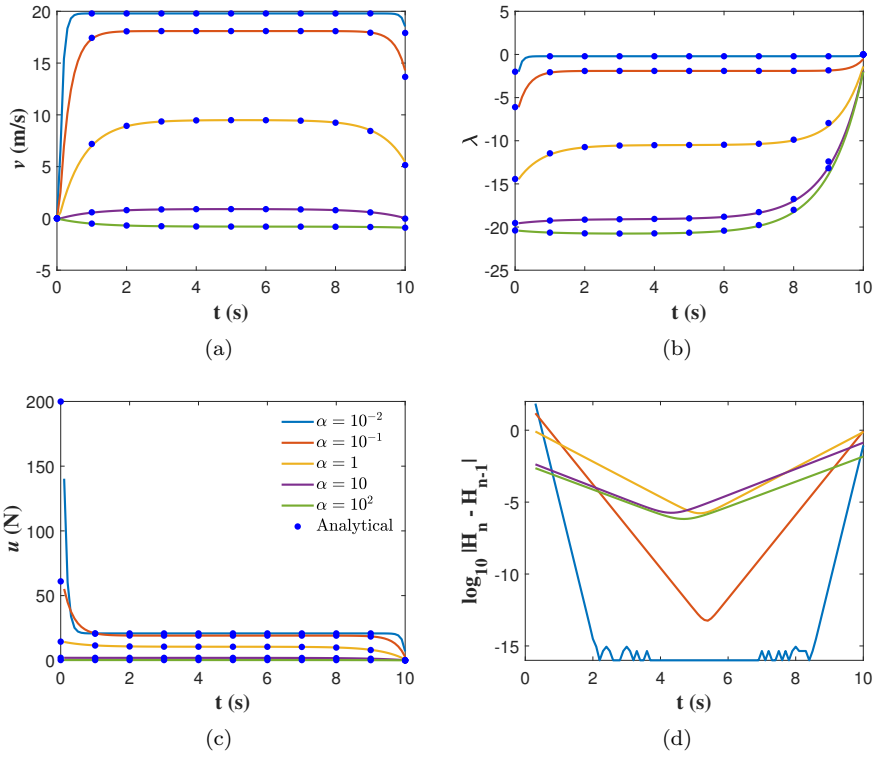


Figure 2: Particle with linear viscosity: DMP algorithm. Time evolution of (a) velocity, (b) co-state, (c) control force (d), control Hamiltonian.

Results of DMP algorithm are in agreement with the analytical solution (overlapped with dots in Fig. 2). Since this approach results in an initial value problem with no restriction on the final condition of the Lagrange multiplier, the Hamiltonian fluctuates near the boundaries. Although these effects vanish quickly as we move away from both boundaries (see Fig. 2), and results are

close to analytical solution, it can be observed that DMP discretisation does not preserve Hamiltonian exactly in general, irrespective of α values. As the α value decreases, error in Hamiltonian values decreases and with α=10^{-2}, DMP approach nearly conserves Hamiltonian in the domain sufficiently far from the boundaries.

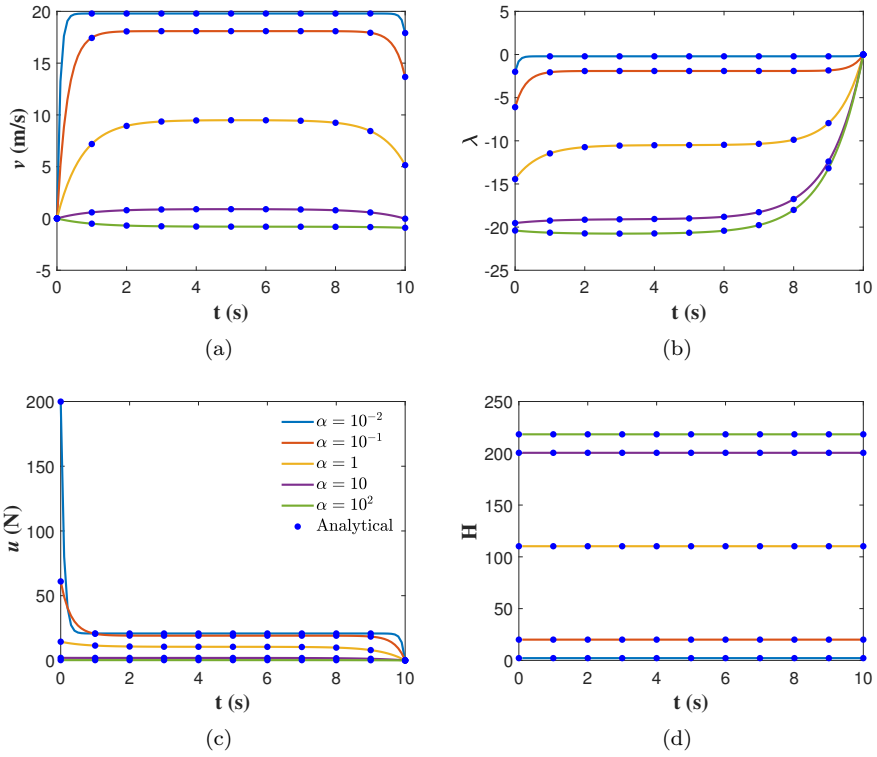
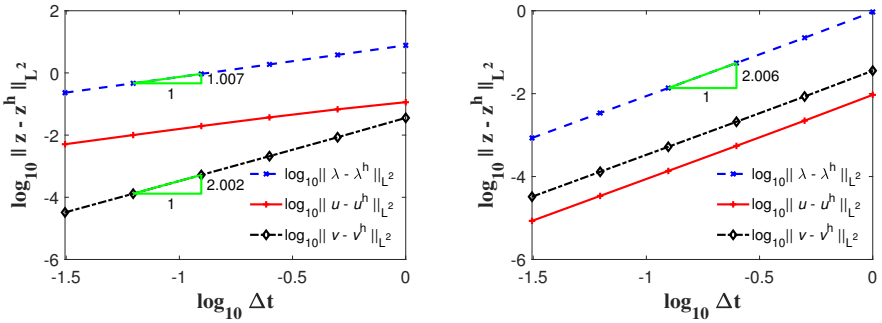


Figure 3: Particle with linear viscosity: IHP algorithm. Time evolution of (a) velocity, (b) co-state, (c) control force, (d) control Hamiltonian.

The proposed Hamiltonian preserving scheme (IHP) has been employed also for the same five values of α. The results for $\Delta t = 0.1$s are also very close to the analytical solution, and unless the DMP algorithm, they preserve the boundary end condition $\lambda_N = 0$ (overlapped with dots in Fig. 3). As expected, the control Hamiltonian is preserved up to machine accuracy, regardless of the value of α.

The two algorithms, DMP and IHP have been compared with the analytical solution, and discrete L^2 norm has been used for quantifying the error. Note that due to the different discretisation of the adjoint equations in both algorithms, $\boldsymbol{\lambda}$ in DMP is divided by Δt for comparing the co-state variables. It can be seen in Fig. 4 that for the IHP approach, the error in primal or dual variables varies quadratically with the time step size Δt, i.e., $\|\cdot\|_{L^2} \sim O(\Delta t^2)$. Instead, DMP approach exhibits linear rate of convergence for the co-state and control trajectories, but quadratic for the primal variable v. Indeed, the governing equation is discretised with a second order midpoint rule (see Fig. 4), but the adjoint equation (first equation in Box 1) does not use a second order scheme, even if the cost function J_Δ in (6) would be approximated with a midpoint



(a) DMP algorithm

(b) IHP algorithm

Figure 4: Particle with linear viscosity: convergence analysis of L^2 error with $\alpha = 10^2$.

rectangular rule. Therefore, the resulting time discretisation of the adjoint equation and the missing boundary condition for λ are responsible of the linear rate of convergence.

4.1.2 Non-linear problem

Linear drag law is applicable in low Reynold´s number flow (Stokes flow) and appropriate to study the optimum swimming strategies of micro-swimmers (micro-robots, bacteria, etc.) [30]. Instead, for moderately high Reynold's number flow, quadratic variation of drag force with particle velocity is more adequate and covers a broad range of engineering problems, such as skydiver motion or car motion [41].

In this example we replace the linear drag of the previous example with the quadratic drag law $-bv^2$, and study the evolution of system optimal state and co-state trajectories. The same cost functional $J(u, v)$ and form of the governing equation in (19)-(20) is employed, but with $f(u, v)$ taking the expression,

$$f(v, u) = -\frac{b}{m}v^2 - a + \frac{u}{m} \tag{24}$$

with $(\alpha, m, b, a) \in \mathbb{R}^+$.

Optimality principles along with Euler-Lagrange equations result in system of non-linear first-order ODE and a close form solution may not exist. The resulting system of equations is solved using IMP and IHP algorithms. In both cases, a high value of α results in insufficient control and particle starts descending, while a decrement in α enhances the system controllability. With a sufficiently small value, $\alpha = 10^{-5}$, particle attains the target velocity in an oscillatory manner. Although the state, co-state and control trajectories of both schemes are quite similar, the IMP algorithm does not preserve the Hamiltonian exactly (see Fig. 5), whereas IHP approach does (see Fig. 6).

For the validation of iterative solver, the non-linear particle problem is solved with Armijo line search algorithm ($\gamma = 0.1$, $\theta_{max} = \frac{1}{||\boldsymbol{p}^k||}$, $\theta_{min} = 0.01$). A constant initial guess $\boldsymbol{u}(t) = 15N$ (10N and 5N furnish the same results); $\alpha =$

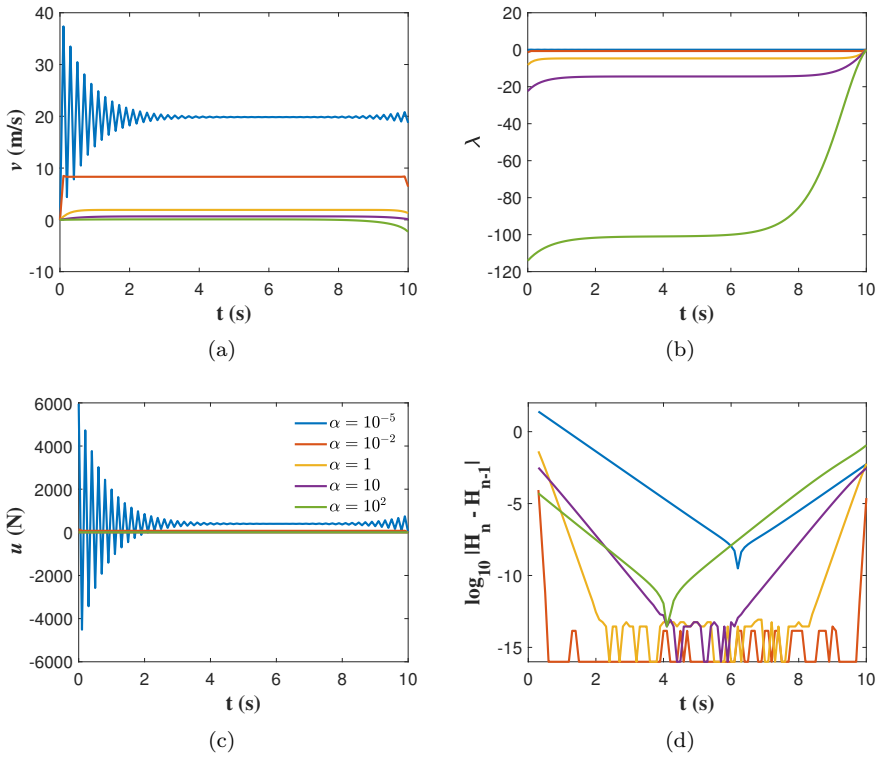


Figure 5: Particle with non-linear viscosity: IMP algorithm. Time evolution of (a) velocity, (b) co-state, (c) control force, and (d) control Hamiltonian increment.

10^{-5} and convergence criteria,

$$max(|\nabla_{\boldsymbol{u}} J|, |\delta \boldsymbol{u}|) \leq 10^{-3}.$$

After 200 iterations, GD did not converge, while conjugate gradient (CG) and Newton-Raphson (NR) did (see Fig. 7). The latter exhibits higher oscillatory response than CG, but a slightly lower value of the cost functional, and achieved convergence in seven iterations (value of J is plotted for reference in Fig. 7d).

Figures 5-7 also show that for small values of α, the response may exhibit oscillations. As shown by the authors in another publication [5], the source of these numerical artifacts is due to the numerical discretisation, and may appear for high values of Δt and small optimisaation parameter α, even in implicit numerical discretisations of the ODEs.

4.2 Inverted elastic pendulum

Inverted elastic pendulum system is often used to model the self-stabilising characteristics of locomotion in humans, animals and biped robots [33, 34]. In this section, we study the non-linear self-stabilising behaviour of inverted elastic pendulum system in the framework of optimal control theory. For the sake of

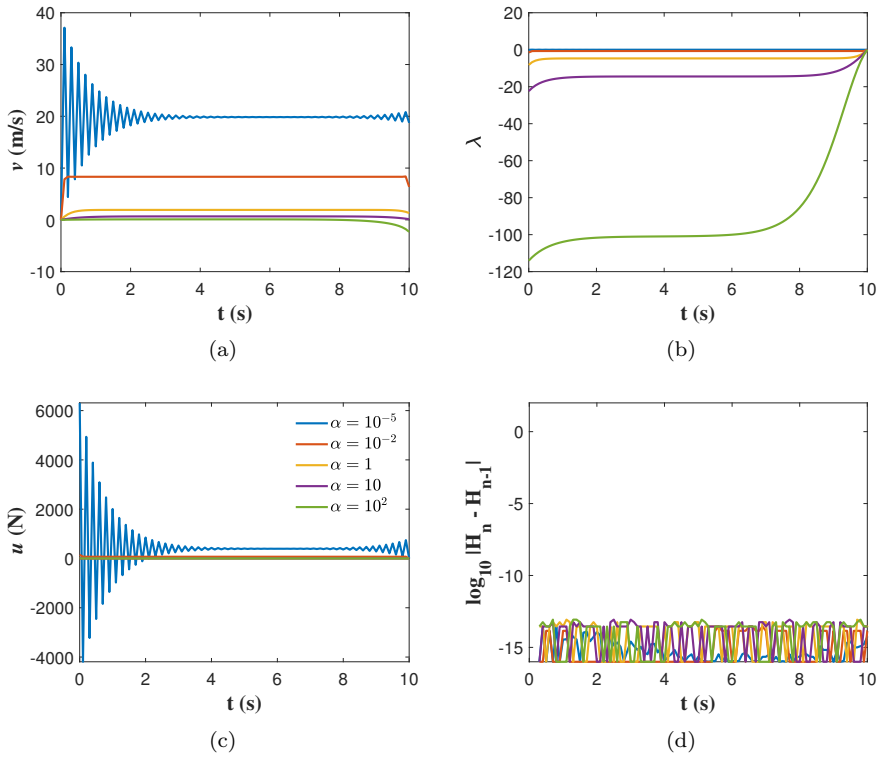


Figure 6: Particle with non-linear viscosity: IHP algorithm. Time evolution of (a) velocity, (b) co-state, (c) control force, and (d) control Hamiltonian increment.

brevity, two particle system has been analysed, although the extension to multi-particle system is formally straightforward.

We simulate two masses m_1 and m_2 linked by an elastic spring and subjected to gravitational field with acceleration a along the z direction (see Fig. 8). Generalized coordinates and velocities of the two particles are $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2\} = \{x_1 \dots x_6\}$ and $\boldsymbol{v} = \{\boldsymbol{v}_1, \boldsymbol{v}_2\} = \{v_1 \dots v_6\}$, respectively. The initial and strain free separation between the two particles is equal to l_o, and the elastic potential energy function takes the form $U_e(\boldsymbol{x}) = \frac{k_s}{2}\left(l(\boldsymbol{x}) - l_o\right)^2$, with $k_s \in \mathbb{R}^+$ the spring stiffness. Gravitational potential energy is given by $U_c(\boldsymbol{x}) = \sum_{i=1}^{2} m_i \boldsymbol{a}_i^{\mathsf{T}} \boldsymbol{x}_i$, and potential energy of the system is the sum of elastic and force field potential energies, i.e. $U(\boldsymbol{x}) = U_e(\boldsymbol{x}) + U_c(\boldsymbol{x})$. Equation of motion of two particle system is given by Euler-Lagrange equation,

$$\mathbf{M}\ddot{\boldsymbol{x}} + \nabla_{\boldsymbol{x}} U = \mathbf{0} \tag{25}$$

where $\mathbf{M} = \mathsf{diag}(m_1, m_1, m_1, m_2, m_2, m_2)$. From the definition of $U(\boldsymbol{x})$, we have that $\nabla_{\boldsymbol{x}} U = k_s\left(l - l_o\right)\nabla_{\boldsymbol{x}} l + \mathbf{M}\boldsymbol{a}$, with $\hat{\boldsymbol{e}} = (\boldsymbol{x}_2 - \boldsymbol{x}_1)/l$, $\nabla_{\boldsymbol{x}} l = \{-\hat{\boldsymbol{e}}, \hat{\boldsymbol{e}}\}^{\mathsf{T}}$, and $\boldsymbol{a} = \{0, 0, a, 0, 0, a\}$. Since equation (25) is a second order ODE, we transformed the governing equations in a first order ODE doubling the number of state and co-state variables. We then chose to control the horizontal velocity of the first mass m_1, so that $u(t) \equiv v_1(t)$.

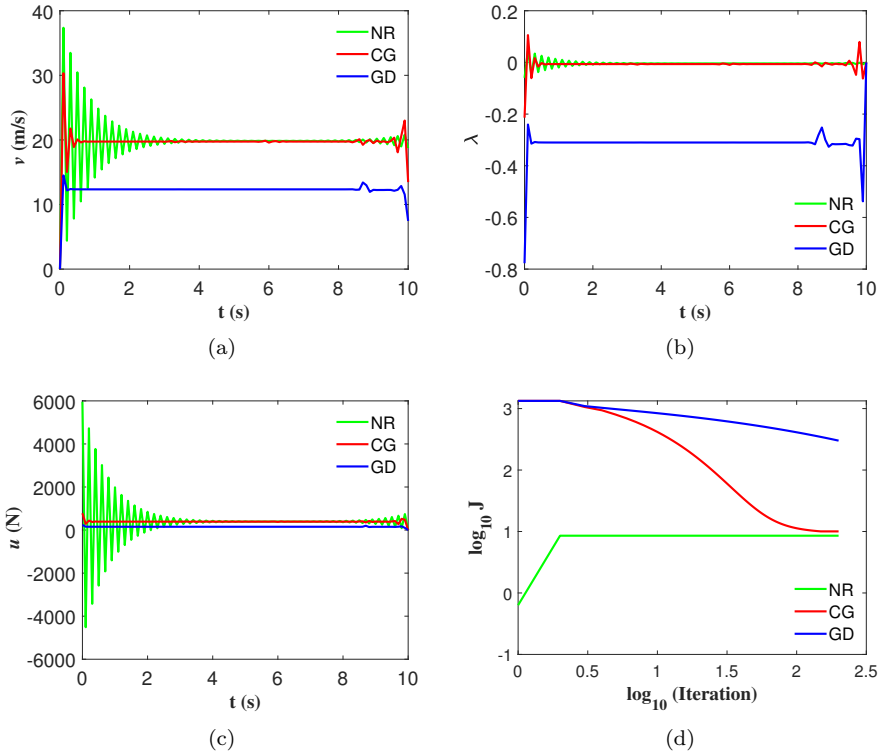


Figure 7: Non-linear falling particle problem with IMP ($\alpha = 10^{-5}$). Time evolution of (a) velocity, (b) co-state, and (c) control variable. (d) Evolution of cost-functional as a function of number of iterations. NR=Newton-Raphson, CG=Conjugate gradient, GD=Gradient descent.

Particles 1 and 2 are initially placed at positions $(0, 0, 0)$ and $(0.3, 0, 1)$m, respectively. Numerical values of spring parameter $k_s = 10$ N/m and force field intensity $a = 0.1$ m/s^2 are assumed constant throughout the motion. In addition, at all time instant motion is confined on z-x plane and particle 1 is constrained to move only horizontally according to our control variable $u(t)$. System of DAEs resulting from DMP, IMP and IHP approaches are solved with $\Delta t = 0.05$ s, $\boldsymbol{x}_d = \{0, 0, 0, 0, 0, 2\}^\mathsf{T}$, $\mathbf{R} = \mathsf{diag}(0, 0, 0, 0, 0, 1)$, $\mathbf{Q} = \mathsf{diag}(1, 0, 0, 0, 0, 0)$, and $\alpha = 10^{-5}$. We point out that in this case the initial guess described in Section 3.1 allowed us to attain convergence with DMP, IMP and IHP algorithm in 7, 6 and 7 iterations respectively using a fully Newton-Raphson procedure in all variables. Without the initial guess, convergence was not achieved.

It can be observed in Fig. 8 that the optimal trajectories and control predicted from the three algorithms, DMP, IMP and IHP are similar. Initially, particle 1 horizontal position rapidly changes from 0 m to 0.3, and then varies coherently with the particle 2 horizontal translation. In order to attain a vertical alignment of the segment joining particles 1 and 2, without a oscillatory response, this coherent horizontal translation of both particles is essential. Moreover, with the horizontal movement of both particles, particle 2 keeps on its free

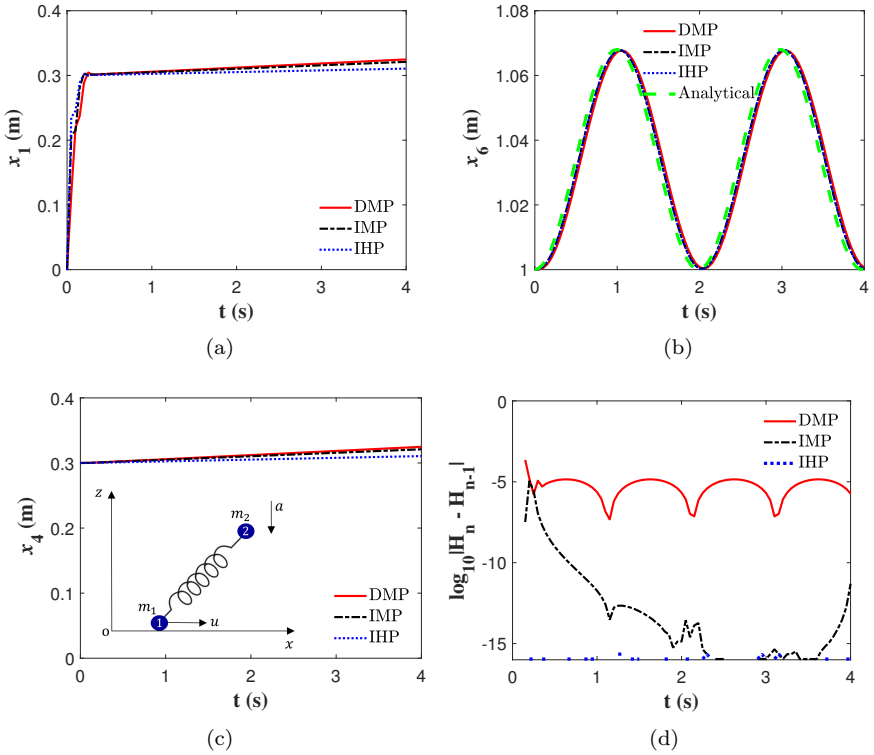


Figure 8: Inverted elastic pendulum. Time evolution of (a) control, (b) particle 2 vertical position, (c) particle 2 horizontal position, and (d) Hamiltonian increment.

oscillation about equilibrium vertical position of 1.034. All algorithms show small variation of control Hamiltonian over time between 0 and 4 seconds, although the only one that exactly preserves $\mathcal{H}$, up to machine accuracy, is the IHP algorithm (see Fig. 8).

4.3 Optimal locomotion of worm-like organism

Let us consider an undeformed elongated body with length L resting along the x-axis. We aim at simulating the motion of a worm-like organism due to bending active internal movements, such as *C. elegans* nematode or similar organism on an assumed rigid substrate. The locomotion of *C. elegans* has been extensively used as model organism in biomedical research [39, 12], where the displacement is thought to be a result of coordinated function of the dorsal and ventral muscle system [13, 18] and non-isotropic frictional forces [24].

The worm is modelled as planar elastic rod subjected to finite stretching and bending deformations only. Worm is discretised into $\overline{n}+1$ discrete nodes with position vector/state (at time t) $\boldsymbol{x} = \{\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{\overline{n}}\}$ and $\overline{n}$ straight segments oriented with respect to each other $\boldsymbol{e}^0, \boldsymbol{e}^1, \ldots, \boldsymbol{e}^{\overline{n}}$ such that $\boldsymbol{e}^i = \boldsymbol{x}^i - \boldsymbol{x}^{i-1}$, $\hat{\boldsymbol{e}}^i = \frac{\boldsymbol{e}^i}{||\boldsymbol{e}^i||}$, $\hat{\boldsymbol{p}}_i = \frac{\boldsymbol{e}^{i-1}+\boldsymbol{e}^i}{||\boldsymbol{e}^{i-1}+\boldsymbol{e}^i||}$ and $\hat{\boldsymbol{p}}_i^T \hat{\boldsymbol{n}}^i = 0$, as depicted in Fig. 9a. The worm mechanics is defined by the following discrete form of stretching (U_s) and bending strain

energy (U_b),

$$U_s(\boldsymbol{x}) = \frac{k_s}{2}\int_0^L \left(\frac{ds}{dS}-1\right)^2 dS \approx \frac{k_s}{2}\sum_{i=1}^{\overline{n}} \frac{1}{\tilde{l}_i}\left(l_i-\tilde{l}_i\right)^2$$

$$U_b(\boldsymbol{x}) = \frac{k_b}{2}\int_0^L \kappa^2 dS \approx \frac{k_b}{2}\sum_{i=1}^{\overline{n}-1} \frac{1}{q_i}(1-\cos^2\theta_i)$$

where dS and ds are differential arc lengths in the underformed and deformed configuration, respectively. Parameter κ is elastic curve curvature, k_s is stretching modulus, k_b is bending modulus, $\tilde{l}_i$ and $l_i = ||\boldsymbol{e}^{i-1}||$ are respectively the undeformed and deformed length of the i^{th} segment, $q_i = (\tilde{l}_i + \tilde{l}_{i+1})/2$ is a Voronoi length region associated with each node, and $\theta_i = \cos^{-1}(\hat{\boldsymbol{e}}^{i-1}\cdot\hat{\boldsymbol{e}}^{i})$ is the bending angle at node i.

Following resistive force theory, anisotropic friction model has been chosen with fixed value of tangential (μ_t) and normal (μ_n) coefficient of friction [12]. At time t, drag force generated at the i^{th} node due to crawling over substrate is assumed directly proportional to the nodal velocity, so that the following frictional forces are applied at each node i:

$$\boldsymbol{f}_i^v = -\left(\mu_t\mathbf{I} + (\mu_n-\mu_t)\left(\hat{\boldsymbol{n}}^i\otimes\hat{\boldsymbol{n}}^i\right)\right)\dot{\boldsymbol{x}}^i$$

where $\mathbf{I}$ is the identity second rank tensor. The assembling of all the nodal contributions gives rise to the following global frictional force vector,

$$\boldsymbol{f}^v = \bigcup_{i=0}^{\overline{n}+1} \boldsymbol{f}_i^v = -\mathbf{B}\dot{\boldsymbol{x}} \tag{26}$$

with $\mathbf{B}$ is a deformation dependent anisotropic friction tensor.

Worm muscle system has been modelled as self equilibrated active force system producing pure bending effects as described in [24]. Internal bending moment at node i, denoted by u_i, induces three nodal forces,

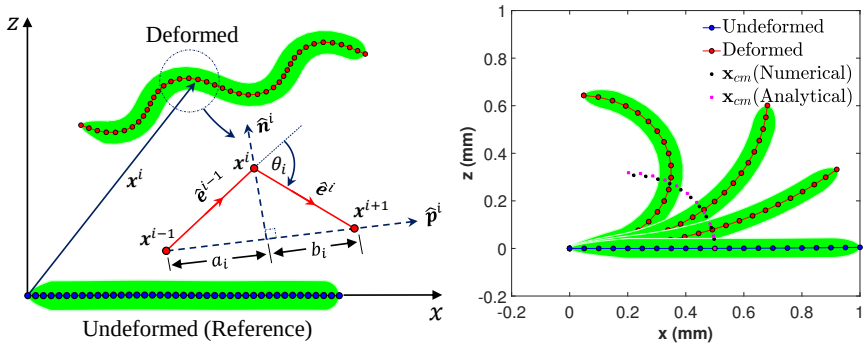
$$\begin{aligned}
\overline{\boldsymbol{f}}^{i-1} &= \frac{u^i}{a_i}\hat{\boldsymbol{n}}^i,\\
\overline{\boldsymbol{f}}^{i} &= -\overline{\boldsymbol{f}}^{i-1} - \overline{\boldsymbol{f}}^{i+1},\\
\overline{\boldsymbol{f}}^{i+1} &= \frac{u^i}{b_i}\hat{\boldsymbol{n}}^i,
\end{aligned}$$

at $i-1^{th}, i^{th}$, and $i+1^{th}$ nodes, respectively. Resultant external force due to internal moment is given by $\boldsymbol{f}^u(\boldsymbol{u}) = \bigcup_{i=1}^{\overline{n}-1}\{\overline{\boldsymbol{f}}^{i-1}, \overline{\boldsymbol{f}}^{i}, \overline{\boldsymbol{f}}^{i+1}\}^T$. At time t, input moment vector/control is defined as $\boldsymbol{u} = \{u^1, u^2, \ldots, u^{\overline{n}-1}]$. Neglecting inertial effects, linear momentum balance gives

$$\mathbf{B}\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \tag{27}$$

where $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{f}^u(\boldsymbol{u}) - \nabla_{\boldsymbol{x}}U_s(\boldsymbol{x}) - \nabla_{\boldsymbol{x}}U_b(\boldsymbol{x})$.

First, we validate the discrete form of elastic strain energy and active force model with the elastica theory. Consider the worm motion on a frictionless



(a) (b)

Figure 9: Wom-like organism model: (a) Geometry of deformation (b) tethered worm coiling on a frictionless substrate

substrate with the anchored tail. We are interested in the evolution of the worm centre of mass ($\boldsymbol{x}_{cm}$) with arbitrary time-varying bending moment ($u(t)$). During motion, worm exhibits insignificant stretching strains compared to bending strain, hence worm is modelled as nearly inextensible rod (length L) with $k_s \approx 500k_b$. If the bending moment is a function of time only ($\boldsymbol{u} = u(t)\mathbf{1}$), then during the motion each point of worm resides on the circle of radius $r(t)$. Using moment-curvature relation ($r(t) = \frac{k_b}{u(t)}$), close-form solution for $\boldsymbol{x}_{cm}$ trajectories can be expressed as

$$\boldsymbol{x}_{cm}(t) = r(t)\hat{\boldsymbol{k}} + \overline{r}(t)\hat{\boldsymbol{d}}(t)$$

where $\overline{r}(t) = \frac{2r(t)^2}{L}\sin\left(\frac{L}{2r(t)}\right)$, $\hat{\boldsymbol{d}}(t) = \sin\left(\frac{L}{2r(t)}\right)\hat{\boldsymbol{i}} - \cos\left(\frac{L}{2r(t)}\right)\hat{\boldsymbol{k}}$, $\hat{\boldsymbol{i}}$ and $\hat{\boldsymbol{k}}$ are unit vector along x and z direction, respectively.

Following, material and physiological parameters has been used in the simulation: $k_b = 1$, $L = 1$ mm, $\mu_n = \mu_t = 0$, $m(t) = \pi\frac{t}{T}$, $T = 4$ s, $\Delta t = 0.01$ s and body is uniformly discretised into 20 segments ($\overline{n} = 20$). It can be observed that the trajectories of $\boldsymbol{x}_{cm}$ are in good agreement with the theoretical results. Furthermore, with the application of constant bending moment ($= \pi$), the unit length worm should deform into a semi-circular arc, and present discrete strain energy form, as we obtained in the simulation shown in see Fig. 9b.

We next used the developed model to investigate the optimal locomotion. We are interested in optimal internal moment distribution $u(t)$ such that the worm centre of mass attains the prescribed target ($\boldsymbol{x}_d$) in a given amount of time (T). We consider the motion which minimises the cost functional in Eqn. (3) with

$$\begin{aligned} r(\boldsymbol{x}) &= \frac{1}{2}(\boldsymbol{x}_{cm} - \boldsymbol{x}_d)^{\mathsf{T}}(\boldsymbol{x}_{cm} - \boldsymbol{x}_d) \\ q(\boldsymbol{u}) &= \frac{\alpha}{2}\boldsymbol{u}^{\mathsf{T}}\mathbf{Q}\boldsymbol{u} \\ \phi(\boldsymbol{x}_{cm}(T)) &= \frac{\beta}{2}(\boldsymbol{x}_{cm}(T) - \boldsymbol{x}_d)^{\mathsf{T}}(\boldsymbol{x}_{cm}(T) - \boldsymbol{x}_d) \end{aligned}$$

Since worm has limited internal energy, we restrict nodal bending moment (u_i) to an admissible control set $\mathcal{U} \in [-2, 2]$. To the best of our knowledge, no optimal locomotion study in the control Hamiltonian framework has been attempted in the literature.

The problem has been solved using IMP algorithm, and FBSM strategy with CG solver using these parameters: $L = 1$ mm; $k_b = 1; \mu_n = 1; \mu_t = 0.1, \Delta t = 0.01$s; $T = 4$s; $\overline{n} = 20$; $\boldsymbol{x}_d = [1.05, 0]$ mm; $\beta = 1, \alpha = \{10^{-5}, 10^{-2}, 10^{-1}\}$, $\mathbf{Q}_{ij} = \delta_{ij}$ (kronecker delta). Wolfe line search algorithm has been used with $\gamma = \sigma = 0.1$, $\theta_{min} = 0.005$ (minimum value) (see Section 3). Half bending moment pulse with amplitude 1 has been used as an initial guess to start the FBSM algorithm (see Fig. 10(a)).

It can be observed that in order to attain the target position optimally, the worm initially generates a series of full sinusoidal moment pulse for short time duration, followed by long term half moment pulses. With the reduction of α, point $\boldsymbol{x}_{cm}$ starts advancing towards target $\boldsymbol{x}_d$, and as α decreases to 10^{-5}, worm reached the prescribed target in the given amount of time of $T = 4s$ (see Fig. 10).

The IHP algorithm has been discarded in this example due to the additional non-linearities and the moderate relative gain in the previous example, despite the advantageous theoretical preserving properties. More importantly, the balance equation in (27) has the form $\mathbf{B}\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, with matrix $\mathbf{B}$ depending on $\boldsymbol{x}$. This fact poses additionally complexities and non-linearities in the implementation of the IHP algorithm. We are currently investigating on how to extend the IHP to this type of governing ODEs. For completeness we plot in Figure 10d the evolution of the control Hamiltonian H, which is oscillatory but remains bounded.

This example also numerically demonstrates the performance of the IMP FBSM strategy for a system with 41303 dof (including state, adjoint and control variables for all time-steps), which is much larger than the 303 dofs employed in the previous example. When attempting to solve the IMP algorithm with a full monolithic Newton-Rapshon (NR) method, no converge was achieved. Instead, the FBSM converged after 200 iterations, with a computational time that was similar to the time of one iteration in the NR process.

5 Conclusions

We have presented three different algorithms that use different time discretisations of the governing and adjoint ODEs found in optimal control theory. The direct midpoint (DMP) resorts to the direct approach, while the indirect midpoint (IMP) and Indirect Hamiltonian Preserving (IHP) are derived from the indirect approach. We have highlighted the differences in the resulting algorithms with the two approaches. Mainly, DMP omits the final boundary condition, and consequently, as shown in Figure 4, has a lower order of convergence in the adjoint ODEs, unless specific modifications are employed.

We have proposed an algorithm, IHP, that preserves an integral of the Euler-Lagrange equations, the control Hamiltonian. In the examples that we have used so far though, no substantial difference has been detected in the robustness of the resulting algorithm, i.e. IMP and IHP algorithms converged or failed to converge for the same parameters and problems. Indeed, the solution of the

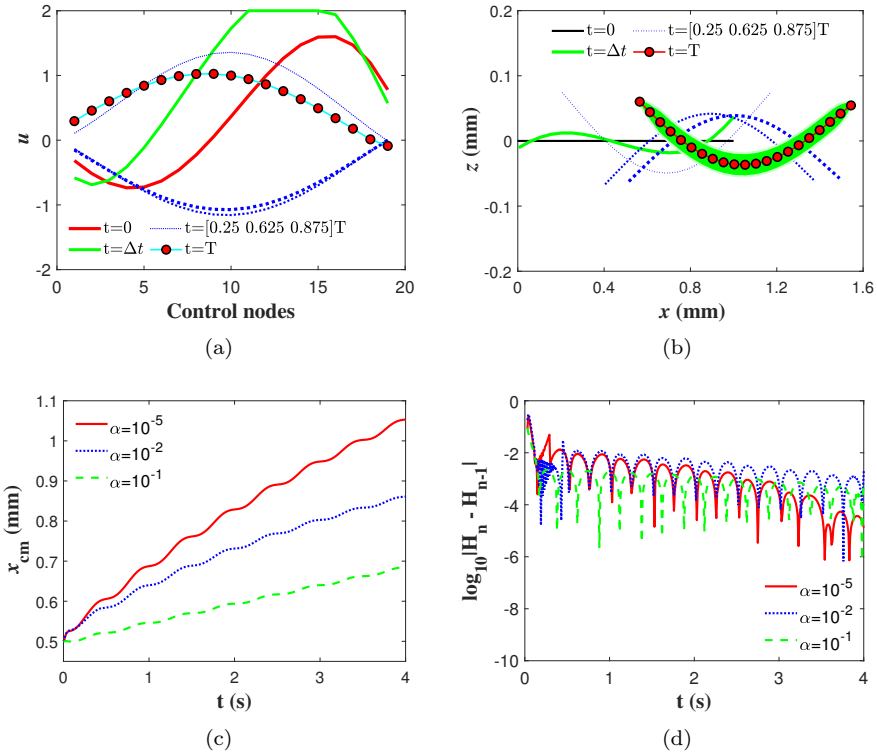


Figure 10: Worm-like organism optimal locomotion: (a) Optimal moment distribution ($\alpha = 10^{-5}$) (b) Optimal locomotion ($\alpha = 10^{-5}$) (c) Centre of mass trajectories with α (d) Evolution of increments of control Hamiltonian in the IMP algorithm.

two point BVP in non-linear systems poses numerical challenges, which must be surmounted before the preservation of the Hamiltonian can make a clear difference.

In this paper we have also suggested and employed some iterative strategies for the solution of the non-linear equations. The use of conjugate gradient and gradient descent techniques are attractive, due to the fact that the size of the global system increases with the number of time-steps, and thus a reduction of the time-step may become a drawback for the numerical solutions with monolithic Newton-Raphson or quasi-Newton strategies. The proposed forward-backward sweep method allows reducing the impact of this computational cost when smaller time-steps are employed. It is easy to implement, for it only involves the solution of a standard forward problem, and a backward (linear) problem, in addition to an update stage of the control variables. However, its convergence is problem dependent. Further investigation of convergence properties for non-linear problems visited here is under progress. We also have left for future investigations the combination of structure preserving discretisations with strategies that yield a constant control Hamiltonian.

The proposed algorithms have been applied to linear and non-linear problems, and also used in practical biological applications that aim at deciphering optimal

mechanical strategies for propulsion of active bodies. The optimal locomotion of worm-like organisms or soft robots poses modelling and numerical challenges. We have detected that the optimal input strongly depends on the initial guess, and that consequently, the algorithms capture local minima. The search of global minima with different initial frequencies and wavelengths is beyond the scope of this article, but we believe it has important implications.

6 Acknowledgements

The authors acknowledge Dr. Michael Krieg and Dr. Ravi Das from the Institute of Photonic Sciences (ICFO, Spain) for their helpful discussions. This work is financially supported by the Spanish Ministry of Science and Innovation, under grants CEX2018-000797-S and PID2020-116141GB-I00.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Bayón, L., Otero, J., Suárez, P., Tasis, C.: New developments in the application of optimal control theory to therapeutic protocols. Math. Biosc. **272**, 34–43 (2016)

[2] Benzi, M., Golub, G., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)

[3] Betsch, P., Becker, C.: Conservation of generalized momentum maps in mechanical optimal control problems with symmetry. IJNME **111**, 144–155 (2017)

[4] Betts, J.: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2nd edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, USA (2010)

[5] Bijalwan, A., Muñoz, J.J.: On the numerical stability of discretised Optimal Control Problems. IUTAM Bookseries, Springer. Under review. Available at `http://arxiv.org/abs/2302.02464`.

[6] Bottasso, C., Croce, A.: Optimal control of multibody systems using an energy preserving direct transcription method. Mult. Syst. Dyn. **12**(4), 17–45 (2004)

[7] Bryson, A., Ho, Y.: Applied Optimal Control. Optimization, Estimation and Control. Taylor & Francis, New York, U.S.A. (1975)

[8] Djukić, D.S.: Noether's theorem for optimum control systems. Int. J. Control **18**(3), 667–672 (1973)

[9] Flaßkamp, K., Murphey, T.: Structure-preserving local optimal control of mechanical systems. Opt. Cont. Appl. Meth. **40**(2), 310–329 (2019)

[10] Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. The Comp. J. **7**(2), 149–154 (1964)

[11] Gonzalez, O.: Time integration and discrete Hamiltonian systems. J. Nonlin. Sc. **8**, 449–467 (1996)

[12] Gray, J., Lissmann, H.: The locomotion of nematodes. J. Exp. Biol. **41**(1), 135–154 (1964)

[13] Huang, Kuang-Man and Cosman, Pamela and Schafer, William R: Machine vision based detection of omega bends and reversals in *C. elegans*. J. Neurosc. Meth. **158**(2), 323–336 (2006)

[14] Hull, D.G.: Optimal control theory for applications. Mechanical Engineering Series. Springer, New York (2003)

[15] Karrakchou, J., Rachik, M., Mostafa, S.: Optimal control and infectiology: application to an HIV/AIDS model. Appl. Math. Comp. **177**(2), 807–818 (2006)

[16] Kelley, C.T.: Iterative methods for optimization. SIAM (1999)

[17] Koch, M., Leyendecker, S.: Energy momentum consistent force formulation for the optimal control of multibody systems. Mult. Syst. Dyn. **29**, 381–401 (2013)

[18] Krieg, M., Stühmer, J., Cueva, J., Fetter, R., Spilker, K., Cremers, D., Shen, K., Dunn, A., Goodman, M.: Genetic defects in ß-spectrin and tau sensitize c. elegans axons to movement-induced damage via torque-tension coupling. eLIFE **6**, e20172 (2017)

[19] Lasdon, L., Mitter, S., Waren, A.: The conjugate gradient method for optimal control problems. IEEE Trans. Aut. Cont. **12**(2), 132–138 (1967)

[20] Lauß, T., Oberpeilsteiner, S., Steiner, W., Nachbagauer, K.: The discrete adjoint gradient computation for optimization problems in multibody dynamics. J. Comput. Nonlinear Dynam. **12**(3), 031016 (2017)

[21] Leyendecker, S., Ober-Blöbaum, S., Marsden, J.E., Ortiz, M.: Discrete mechanics and optimal control for constrained systems. Opt. Cont. Appl. Meth. **39**(6), 505–528 (2009)

[22] McAsey, M., Mou, L., Han, W.: Convergence of the forward-backward sweep method in optimal control Comput. Optim. Appl. **53**, 207–226 (2012)

[23] Miller, M., Trouvé, A., Younes, L.: Hamiltonian systems and optimal control in computational anatomy: 100 years since D'Arcy Thompson. Annals Biomed. Engin. **17**, 447–509 (2015)

[24] Muñoz, J.J., Condamin, L., Doste, D.: On the net displacement of contact surface centroid in contractile bodies. Mech. Res. Comm. **119**, 103809 (2022)

[25] Nachbagauer, K., Oberpeilsteiner, S., Sherif, K., Steiner, W.: The use of the adjoint method for solving typical optimization problems in multibody dynamics. J. Comput. Nonlinear Dynam. **10**(6) (2015)

[26] Nocedal, J., Wright, S.: Numerical Optimization, 2nd edn. Springer, New York (2006)

[27] Ober-Blöbaum, S., Junge, O., Marsden, J.: Discrete mechanics and optimal control: an analysis. ESAIM: Cont. Opt. Cal. Var. **17**(2), 322–352 (2011)

[28] Pesch, H., Plail, M.: The cold war and the maximum principle of optimal control. Optimization Stories. Documenta Mathematica (2012)

[29] Pikuliński, M., Malczyk, P.: Adjoint method for optimal control of multibody systems in the Hamiltonian setting. Mech. Mach. Th. **166**, 104473 (2021)

[30] Purcell, E.: Live at low Reynolds number. Amer. J. Phys. **45**(1), 3–11 (1977)

[31] Rao, Anil V: A survey of numerical methods for optimal control. Advances in the Astronautical Sciences **135**(1), 497–528 (2009)

[32] Saad, Y.: Iterative methods for sparse linear systems, 2nd edn. Society of Industrial and Applied Mathematics, SIAM (2003)

[33] Schmitthenner, D., Martin, A.: Comparing system identification techniques for identifying human-like walking controllers. R. Soc. Open Sci. **8**(12), 211031 (2021)

[34] Schwind, W., Koditschek, D.: Approximating the stance map of a 2-DOF monoped runner. J. Nonlin. Sc. **10**(5), 533–568 (2000)

[35] Sharma, H., Patil, M., Woolsey, C.: A review of structure-preserving numerical methods for engineering applications. CMAME **366**, 113067 (2020)

[36] Sharp, J., Burrage, K., Simpson, M.: Implementation and acceleration of optimal control for systems biology. J. R. Soc. Interface **18**, 20210241 (2021)

[37] Simo, J., Tarnow, N.: The discrete energy-momentum method. conserving algorithms for nonlinear elastodynamics. J. Appl. Math. Phys. (ZAMP) **43**, 757–792 (1992)

[38] Stengel, R.: Optimal control and estimation. Dover, New York, USA (1994)

[39] Stephens, G., Johnson-Kerner, B., Bialek, W., Ryu, W.: Dimensionality and dynamics in the behavior of *c. elegans*. PLOS Comp. Biol. **4**, e1000028 (2008)

[40] Sussmann, H., Willems, J.: 300 years of optimal control: from the brachystochrone to the maximum principle. IEEE Cont. Sys. Mag. **17**(3), 32–44 (1997)

[41] Timmerman, P., van der Weele, J.P.: On the rise and fall of a ball with linear or quadratic drag. Amer. J. Phys. **67**(6), 538–546 (1999)