Arbitrarily high-order energy-conserving methods for Poisson problems

Pierluigi Amodio[1] · Luigi Brugnano[2] ⬤ **· Felice Iavernaro[1]**

Abstract

In this paper, we are concerned with energy-conserving methods for Poisson problems, which are effectively solved by defining a suitable generalization of HBVMs, a class of energy-conserving methods for Hamiltonian problems. The actual implementation of the methods is fully discussed, with a particular emphasis on the conservation of Casimirs. Some numerical tests are reported, in order to assess the theoretical findings.

Keywords Poisson problems · Casimir function · Line integral methods · Hamiltonian boundary value methods · HBVMs

Mathematics Subject Classification (2010) 65L05 · 65P10

1 Introduction

A Poisson problem is in the form

$$\dot{y} = B(y)\nabla H(y) =: f(y), \quad t > 0, \qquad y(0) = y_0 \in \mathbb{R}^m, \qquad B(y)^\top = -B(y), \tag{1}$$

where $H(y)$ is a scalar function, usually called the *Hamiltonian*. For sake of simplicity, hereafter both $H(y)$ and $B(y)$ will be assumed suitably regular. From the

✉ Luigi Brugnano
luigi.brugnano@unifi.it

Pierluigi Amodio
pierluigi.amodio@uniba.it

Felice Iavernaro
felice.iavernaro@uniba.it

[1] Dipartimento di Matematica, Università di Bari, Bari, Italy

[2] Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Firenze, Italy

🖄 Springer

skew-symmetry of $B(y)$ one easily deduces that $H(y)$ is a constant of motion, since, along the solution of (1),

$$\frac{\mathrm{d}}{\mathrm{d}t}H(y) = \nabla H(y)^\top \dot{y} = \nabla H(y)^\top B(y)\nabla H(y) = 0,$$

due to the skew-symmetry of $B(y)$. Possible additional invariants of (1) are its *Casimirs*, namely scalar functions $C(y)$ for which

$$\nabla C(y)^\top B(y) = (0, \ldots, 0) \in \mathbb{R}^{1 \times m} \tag{2}$$

holds true for all y. When matrix $B(y)$ is constant, as in the case of Hamiltonian problems,

$$\dot{y} = J\nabla H(y), \qquad t > 0, \qquad y(0) = y_0, \qquad J^\top = -J, \tag{3}$$

energy conservation can be obtained by solving problem (3) via HBVMs, a class of energy-conserving Runge-Kutta methods for Hamiltonian problems (see, e.g., [7, 8, 11, 12] and the monograph [5], see also the recent review paper [6]). Nevertheless, in the case where the problem is not Hamiltonian, HBVMs are no more energy-conserving. This motivates the present paper, where an energy-conserving variant of HBVMs for Poisson problems is derived and analyzed.

The numerical solution of Poisson problems has been tackled by following many different approaches (see, e.g., [20, Chapter VII] and references therein). More recently, it has been considered in [19], where an extension of the AVF method [22] is proposed, and in [1, 3], where a line integral approach has been used instead. Functionally fitted methods have been proposed in [21, 24–26]. In this paper we further pursue the line integral approach to the problem, which will provide an energy-conserving variant of HBVMs for solving (1).

With this premise, the structure of the paper is the following: in Section 2 we describe the new framework, in which the methods will be derived; in Section 3 we provide the final shape of the method, while in Section 4 its actual implementation is studied; in Section 5 we present a few numerical tests confirming the theoretical findings; at last, in Section 6 we give some concluding remarks.

2 The new framework

As anticipated above, the framework that we shall use to derive and analyze the methods is that of the so-called *line integral methods*, namely methods where the conservation properties are derived by the vanishing of a corresponding *line integral* [5, 6]. Such methods have been largely investigated in the case of Hamiltonian problems, providing their major instance given by Hamiltonian Boundary Value Methods (HBVMs). The analysis will strictly follow that in [11] and [17]. To begin with, let us consider problem (1) on the interval $[0, h]$,

$$\dot{y}(ch) = B(y(ch))\nabla H(y(ch)), \qquad c \in [0, 1], \qquad y(0) = y_0. \tag{4}$$

In fact, since we shall speak about a one-step method, it suffices to analyze its first step of application, with h the time-step. Next, let us consider the orthonormal

Legendre polynomial basis $\{P_j\}_{j \geq 0}$ on the interval [0, 1],

$$\deg P_j = j, \qquad \int_0^1 P_i(x)P_j(x)\mathrm{d}x = \delta_{ij}, \qquad \forall i, j = 0, 1, \ldots, \tag{5}$$

with δ_{ij} the Kronecker symbol, and the following expansions for the functions at the right-hand side in (4):

$$\begin{aligned}
\nabla H(y(ch)) = \sum_{j \geq 0} P_j(c)\gamma_j(y), \quad & P_j(c)B(y(ch)) = \sum_{i \geq 0} P_i(c)\rho_{ij}(y), \qquad c \in [0, 1], \\
\gamma_j(y) = \int_0^1 P_j(\tau)\nabla H(y(\tau h))\mathrm{d}\tau, \quad & \rho_{ij}(y) = \int_0^1 P_i(\tau)P_j(\tau)B(y(\tau h))\mathrm{d}\tau, \quad i, j = 0, 1, \ldots.
\end{aligned} \tag{6}$$

The following properties hold true.

Lemma 1 *Assume $\psi : [0, h] \to V$, with V a vector space, admit a Taylor expansion at 0. Then, for all $j = 0, 1, \ldots$:*

$$\int_0^1 P_j(c)c^i \psi(ch)\mathrm{d}c = O(h^{j-i}), \qquad i = 0, \ldots, j.$$

Proof By the hypotheses on ψ, one has:

$$c^i \psi(ch) = \sum_{r \geq 0} \frac{\psi^{(r)}(0)}{r!}h^r c^{r+i}.$$

Consequently, for all $i = 0, \ldots, j$, by virtue of (5) it follows that:

$$\begin{aligned}
\int_0^1 P_j(c)c^i \psi(ch)\mathrm{d}c &= \sum_{r \geq 0} \frac{\psi^{(r)}(0)}{r!}h^r \int_0^1 P_j(c)c^{r+i}\mathrm{d}c \\
&= \sum_{r \geq j-i} \frac{\psi^{(r)}(0)}{r!}h^r \int_0^1 P_j(c)c^{r+i}\mathrm{d}c = O(h^{j-i}).
\end{aligned}$$

$\square$

Corollary 1 *With reference to (6), for any suitably regular path $\sigma : [0, h] \to \mathbb{R}^m$ one has:*

$$\gamma_j(\sigma) = O(h^j), \qquad \rho_{ij}(\sigma) = O(h^{|i-j|}). \qquad \forall i, j = 0, 1, \ldots. \tag{7}$$

Proof Immediate from Lemma 1, by taking into account (6). $\square$

We also state, without proof, the following straightforward property, deriving from the skew-symmetry of B.

Lemma 2 *With reference to (6), for any path $\sigma : [0, h] \to \mathbb{R}^m$ one has:*

$$\rho_{ij}(\sigma) = \rho_{ji}(\sigma) = -\rho_{ij}(\sigma)^\top, \qquad \forall i, j = 0, 1, \ldots. \tag{8}$$

Taking into account (6), the right-hand side in (4) can be rewritten as:

$$\dot{y}(ch) = B(y(ch))\nabla H(y(ch)) = \sum_{j\geq0} P_j(c)B(y(ch))\gamma_j(y) = \sum_{i,j\geq0} P_i(c)\rho_{ij}(y)\gamma_j(y), \qquad c \in [0, 1],$$

(9)

from which one obtains that the solution of (4) can be formally written as:

$$y(ch) = y_0 + h \sum_{i,j\geq0} \int_0^c P_i(x)\mathrm{d}x\rho_{ij}(y)\gamma_j(y), \qquad c \in [0, 1].$$

(10)

In particular, by considering (5) and that $P_0(c) \equiv 1$, from which $\int_0^1 P_i(x)\mathrm{d}x = \delta_{i0}$, one has:

$$y(h) = y_0 + h \sum_{j\geq0} \rho_{0j}(y)\gamma_j(y) \equiv y_0 + h \sum_{j\geq0} \int_0^1 P_j(c)B(y(ch))\mathrm{d}c \int_0^1 P_j(c)\nabla H(y(ch))\mathrm{d}c. \quad (11)$$

In order to obtain a polynomial approximation of degree s to y, it suffices to truncate the two infinite series in (9) after s terms:

$$\dot{\sigma}(ch) = \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma)\gamma_j(\sigma), \qquad c \in [0, 1],$$

(12)

with $\rho_{ij}(\sigma)$ and $\gamma_j(\sigma)$ defined according to (6) by formally replacing y by σ. Consequently, (10) becomes

$$\sigma(ch) = y_0 + h \sum_{i,j=0}^{s-1} \int_0^c P_i(x)\mathrm{d}x\rho_{ij}(\sigma)\gamma_j(\sigma), \qquad c \in [0, 1],$$

(13)

providing the approximation

$$y_1 := \sigma(h) = y_0 + h \sum_{j=0}^{s-1} \rho_{0j}(\sigma)\gamma_j(\sigma) \equiv y_0 + h \sum_{j=0}^{s-1} \int_0^1 P_j(c)B(\sigma(ch))\mathrm{d}c \int_0^1 P_j(c)\nabla H(\sigma(ch))\mathrm{d}c,$$

(14)

in place of (11).

2.1 Interpretation of σ

We now provide an interesting interpretation of the polynomial approximation σ. For this purpose, let us rewrite (9), by taking into account (6), as follows:

$$
\begin{aligned}
\dot{y}(ch) &= \sum_{i,j\geq 0} P_i(c)\rho_{ij}(y)\gamma_j(y) \\
&= \sum_{i\geq 0} P_i(c) \sum_{j\geq 0} \int_0^1 P_i(\tau)B(y(\tau h))P_j(\tau)\mathrm{d}\tau \int_0^1 P_j(\tau_1)\nabla H(y(\tau_1 h))\mathrm{d}\tau_1 \\
&= \sum_{i\geq 0} P_i(c) \int_0^1 P_i(\tau)B(y(\tau h)) \underbrace{\left(\sum_{j\geq 0} P_j(\tau)\int_0^1 P_j(\tau_1)\nabla H(y(\tau_1 h))\mathrm{d}\tau_1 \right)}_{=\nabla H(y(\tau))} \mathrm{d}\tau \\
&\equiv B(y(ch))\nabla H(y(ch)),
\end{aligned}
$$

as is expected. In a similar way, we can rewrite (12) as:

$$
\begin{aligned}
\dot{\sigma}(ch) &= \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma)\gamma_j(\sigma) \\
&= \sum_{i=0}^{s-1} P_i(c) \sum_{j=0}^{s-1} \int_0^1 P_i(\tau)B(\sigma(\tau h))P_j(\tau)\mathrm{d}\tau \int_0^1 P_j(\tau_1)\nabla H(\sigma(\tau_1 h))\mathrm{d}\tau_1 \\
&= \sum_{i=0}^{s-1} P_i(c) \int_0^1 P_i(\tau)B(\sigma(\tau h)) \left(\sum_{j=0}^{s-1} P_j(\tau)\int_0^1 P_j(\tau_1)\nabla H(\sigma(\tau_1 h))\mathrm{d}\tau_1 \right) \mathrm{d}\tau \\
&=: \sum_{i=0}^{s-1} P_i(c) \int_0^1 P_i(\tau)B(\sigma(\tau h))\left[\nabla H(\sigma(\tau h))\right]_s \mathrm{d}\tau \\
&\equiv \left[B(\sigma(\tau h))\left[\nabla H(\sigma(\tau h))\right]_s \right]_s,
\end{aligned}
$$

having denoted by $[\cdot]_s$ the best approximation in Π_{s-1} (i.e., $[\cdot]_s$ is the best polynomial approximation of degree $s-1$) of the function in argument. This fact provides a noticeable interpretation of the polynomial approximation σ, which is the solution of the initial value problem

$$
\dot{\sigma}(ch) = \left[B(\sigma(ch))\left[\nabla H(\sigma(ch))\right]_s \right]_s, \qquad c \in [0,1], \qquad \sigma(0) = y_0, \tag{15}
$$

equivalent to (12). Thus, the vector field of (15) is defined by a double projection procedure onto the finite dimensional vector space Π_{s-1} which involves, in turn, the vector fields $\nabla H(\sigma(ch))$ and $B(\sigma(ch))\left[\nabla H(\sigma(ch))\right]_s$, respectively.

2.2 Analysis

We now analyze the method (12)–(14). The following result then holds true, stating that the method is energy-conserving.

Theorem 1 $H(y_1) = H(y_0)$.

Proof In fact, by virtue of (1), (6), and (12)–(14) one has, by using the standard line integral argument:

$$
H(y_1) - H(y_0) = H(\sigma(h)) - H(\sigma(0)) = \int_0^h \nabla H(\sigma(t))^\top \dot{\sigma}(t) dt
$$

$$
= h \int_0^1 \nabla H(\sigma(ch))^\top \dot{\sigma}(ch) dc = h \int_0^1 \nabla H(\sigma(ch))^\top \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma)\gamma_j(\sigma) dc
$$

$$
= h \sum_{i,j=0}^{s-1} \left[\int_0^1 P_i(c)\nabla H(\sigma(ch)) dc \right]^\top \rho_{ij}(\sigma)\gamma_j(\sigma) = h \sum_{i,j=0}^{s-1} \gamma_i(\sigma)^\top \rho_{ij}(\sigma)\gamma_j(\sigma) = 0,
$$

where the last equality follows from (8). $\qquad\square$

Concerning the accuracy of the new approximation, the following result holds true.

Theorem 2 *Let y_1 be defined according to* (12)–(14). *Then, $y_1 - y(h) = O(h^{2s+1})$.*[1]

Proof Let $y(t) \equiv y(t, \xi, \eta)$ denote the solution of the initial value problem (see (1))

$$
\dot{y} = B(y)\nabla H(y) =: F(y), \qquad t \geq \xi, \qquad y(\xi) = \eta, \tag{16}
$$

Moreover, let us denote

$$
\Phi(t, \xi, \eta) = \frac{\partial}{\partial \eta} y(t, \xi, \eta), \qquad t \geq \xi,
$$

also recalling that

$$
\frac{\partial}{\partial \xi} y(t, \xi, \eta) = -\Phi(t, \xi, \eta) F(\eta).
$$

Then, by taking into account Lemma 1 and Corollary 1, and setting

$$
\Psi_i(\sigma) = \int_0^1 P_i(c)\Phi(h, ch, \sigma(ch)) dc = O(h^i), \qquad i = 0, 1, \ldots,
$$

---

[1] I.e., the approximation procedure has order of convergence $2s$.

one has:

$$y_1 - y(h) = \sigma(h) - y(h) = y(h, h, \sigma(h)) - y(h, 0, \sigma(0)) = \int_0^h \frac{d}{dt} y(h, t, \sigma(t)) dt$$

$$= \int_0^h \left[\frac{\partial}{\partial \xi} y(h, \xi, \sigma(t)) \Big|_{\xi=t} + \frac{\partial}{\partial \eta} y(h, t, \eta) \Big|_{\eta=\sigma(t)} \dot{\sigma}(t) \right] dt$$

$$= \int_0^h [-\Phi(h, t, \sigma(t)) F(\sigma(t)) + \Phi(h, t, \sigma(t)) \dot{\sigma}(t)] dt$$

$$= -h \int_0^1 \Phi(h, ch, \sigma(ch)) [F(\sigma(ch)) - \dot{\sigma}(ch)] dc$$

$$= -h \int_0^1 \Phi(h, ch, \sigma(ch)) \left[B(\sigma(ch)) \nabla H(\sigma(ch)) - \sum_{i,j=0}^{s-1} P_i(c) \rho_{ij}(\sigma) \gamma_j(\sigma) \right] dc$$

$$= -h \int_0^1 \Phi(h, ch, \sigma(ch)) \left[\sum_{i,j \geq 0} P_i(c) \rho_{ij}(\sigma) \gamma_j(\sigma) - \sum_{i,j=0}^{s-1} P_i(c) \rho_{ij}(\sigma) \gamma_j(\sigma) \right] dc$$

$$= -h \left[\sum_{i,j \geq 0} \Psi_i(\sigma) \rho_{ij}(\sigma) \gamma_j(\sigma) - \sum_{i,j=0}^{s-1} \Psi_i(\sigma) \rho_{ij}(\sigma) \gamma_j(\sigma) \right]$$

$$= -h \left[\sum_{i=0}^{s-1} \sum_{j \geq s} \underbrace{\Psi_i(\sigma) \rho_{ij}(\sigma)}_{=O(h^j)} \gamma_j(\sigma) + \sum_{i \geq s} \sum_{j=0}^{s-1} \Psi_i(\sigma) \underbrace{\rho_{ij}(\sigma) \gamma_j(\sigma)}_{=O(h^i)} + \sum_{i,j \geq s} \Psi_i(\sigma) \rho_{ij}(\sigma) \gamma_j(\sigma) \right]$$

$$= O(h^{2s+1}).$$

$$\qquad \square$$

At last, we observe that, since the procedure (12)–(14) is equivalent to defining the path σ that joins $\sigma(0) = y_0$ to $\sigma(h) = y_1$, then the same procedure, when started at y_0 and going forward provides y_1 and, when started from y_1 and going backward, brings back to y_0. In other words, the following result holds true.

Theorem 3 *The procedure* (12)–(14) *is symmetric.*

Proof This result comes as an easy consequence of Theorem 11, where the analogous property for the full discretized method is shown. $\qquad \square$

Remark 1 We conclude this section emphasizing that, when problem (1) is Hamiltonian, i.e., in the form (3), then matrix $B(y) \equiv J$ is constant, and therefore (see (6)) $\rho_{ij}(\sigma) = \delta_{ij} J$. Consequently, (13) becomes

$$\sigma(ch) = y_0 + h \sum_{j=0}^{s-1} \int_0^c P_j(x) dx \int_0^1 P_j(\tau) J \nabla H(\sigma(\tau h)) d\tau, \qquad c \in [0, 1],$$

which (see [6, 8, 10]) is the so-called *master functional equation* defining the class of energy-conserving methods named *Hamiltonian Boundary Value Methods (HBVMs)*.

Consequently, when the problem is Hamiltonian, then the procedure (12)–(14) reduces to the *HBVM*(∞, s) method in [8].

2.3 Conservation of Casimirs

In this section, we study the required modifications, in order to conserve Casimirs, i.e., functions satisfying (2). For sake of simplicity, we shall consider the simpler case of one Casimir, but multiple ones can be handled by slightly adapting the arguments, as is sketched at the end of the section. To begin with, for the original problem (4), and its equivalent formulation (9), one has:

$$
\begin{aligned}
0 &= C(y(h)) - C(y_0) = \int_0^h \nabla C(y(t))^\top \dot{y}(t) \mathrm{d}t = h \int_0^1 \nabla C(y(ch))^\top \dot{y}(ch) \mathrm{d}c \\
&= h \int_0^1 \nabla C(y(ch))^\top B(y(ch)) \nabla H(y(ch)) \mathrm{d}c = h \int_0^1 \nabla C(y(ch))^\top \sum_{i,j \ge 0} P_i(c) \rho_{ij}(y) \gamma_j(y) \\
&= h \sum_{i,j \ge 0} \left[\int_0^1 P_i(c) \nabla C(y(ch)) \mathrm{d}c \right]^\top \rho_{ij}(y) \gamma_j(y) =: h \sum_{i,j \ge 0} \pi_i(y)^\top \rho_{ij}(y) \gamma_j(y), \quad (17)
\end{aligned}
$$

having set

$$
\pi_i(y) = \int_0^1 P_i(c) \nabla C(y(ch)) \mathrm{d}c = O(h^i), \quad (18)
$$

the i-th Fourier coefficient of the gradient of the Casimir, with the last equality following from Lemma 1. Clearly, again from (2), one derives, by taking into account (12):

$$
\begin{aligned}
C(y_1) - C(y_0) &= C(\sigma(h)) - C(\sigma(0)) = \int_0^h \nabla C(\sigma(t))^\top \dot{\sigma}(t) \mathrm{d}t = h \int_0^1 \nabla C(\sigma(ch))^\top \dot{\sigma}(ch) \mathrm{d}c \\
&= h \int_0^1 \nabla C(\sigma(ch))^\top [\dot{\sigma}(ch) - B(\sigma(ch)) \nabla H(\sigma(ch))] \mathrm{d}c \\
&\quad + h \overbrace{\int_0^1 \nabla C(\sigma(ch))^\top B(\sigma(ch)) \nabla H(\sigma(ch)) \mathrm{d}c}^{=0} \\
&= -h \int_0^1 \nabla C(\sigma(ch))^\top \left[\sum_{i,j \ge 0} P_i(c) \rho_{ij}(\sigma) \gamma_j(\sigma) - \sum_{i,j=0}^{s-1} P_i(c) \rho_{ij}(\sigma) \gamma_j(\sigma) \right] \mathrm{d}c \\
&= -h \left[\sum_{i,j \ge s} \pi_i(\sigma)^\top \rho_{ij}(\sigma) \gamma_j(\sigma) + \sum_{i=0}^{s-1} \sum_{j \ge s} \underbrace{\pi_i(\sigma)^\top \rho_{ij}(\sigma)}_{=O(h^j)} \gamma_j(\sigma) + \sum_{j=0}^{s-1} \sum_{i \ge s} \pi_i(\sigma)^\top \underbrace{\rho_{ij}(\sigma) \gamma_j(\sigma)}_{=O(h^i)} \right] \\
&= O(h^{2s+1}). \quad (19)
\end{aligned}
$$

In order to recover the conservation of Casimirs, we shall use a strategy akin to that used in [18] for HBVMs (see also [4]), i.e., suitably perturbing some of its coefficients. In more details, let us consider the following modified polynomial in place

of (12):[2]

$$\dot{\sigma}_\alpha(ch) = \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha\tilde{B}\gamma_0(\sigma_\alpha), \qquad c \in [0,1], \qquad \sigma_\alpha(0) = y_0,$$
(20)

with $\tilde{B}^\top = -\tilde{B} \neq O$ an arbitrary skew-symmetric matrix. As is usual, the new approximation will be $y_1 := \sigma_\alpha(h)$. In other words, we have considered the following perturbed coefficient:

$$\rho_{00}(\sigma_\alpha) - \alpha\tilde{B},$$

in place of $\rho_{00}(\sigma)$ in (12). The following result holds true.

Theorem 4 *Assume that $\pi_0(\sigma_\alpha)^\top \tilde{B}\gamma_0(\sigma_\alpha) \neq 0$. Then the Casimir $C(y)$ is conserved, provided that*

$$\alpha = \frac{\sum_{i,j=0}^{s-1} \pi_i(\sigma_\alpha)^\top \rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha)}{\pi_0(\sigma_\alpha)^\top \tilde{B}\gamma_0(\sigma_\alpha)}.$$
(21)

Moreover, $\alpha = O(h^{2s})$.

Proof In fact, by repeating similar steps as in (19), and replacing σ by σ_α, as defined in (20), one obtains:

$$C(y_1) - C(y_0) = C(\sigma_\alpha(h)) - C(\sigma_\alpha(0)) = h\int_0^1 \nabla C(\sigma_\alpha(ch))^\top \dot{\sigma}_\alpha(ch)\mathrm{d}c$$

$$= h\int_0^1 \nabla C(\sigma_\alpha(ch))^\top \left[\sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha\tilde{B}\gamma_0(\sigma_\alpha)\right]\mathrm{d}c$$

$$= h\left[\sum_{i,j=0}^{s-1} \pi_i(\sigma_\alpha)^\top \rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha\pi_0(\sigma_\alpha)^\top \tilde{B}\gamma_0(\sigma_\alpha)\right] = 0,$$

provided that (21) holds true. The statement is completed by observing that the numerator is $O(h^{2s})$, whereas, the denominator is $O(1)$. $\square$

We now prove that the results of Theorems 1 and 2 continue to hold for the polynomial (20).

Theorem 5 *For any α: $H(\sigma_\alpha(h)) = H(\sigma_\alpha(0))$.*

---

[2]Here, we take into account that $P_0(c) \equiv 1$.

Proof Following similar steps as in the proof of Theorem 1, one has:

$$H(\sigma_\alpha(h)) - H(\sigma_\alpha(0)) = \int_0^h \nabla H(\sigma_\alpha(t))^\top \dot{\sigma}_\alpha(t) dt$$

$$= h \int_0^1 \nabla H(\sigma_\alpha(ch))^\top \dot{\sigma}_\alpha(ch) dc = h \int_0^1 \nabla H(\sigma_\alpha(ch))^\top \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) dc$$

$$- h\alpha \left[\int_0^1 \nabla H(\sigma_\alpha(ch)) dc \right]^\top \tilde{B}\gamma_0(\gamma_\alpha)$$

$$= h \underbrace{\sum_{i,j=0}^{s-1} \gamma_i(\sigma_\alpha)^\top \rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha)}_{=0} - h\alpha\, \gamma_0(\sigma_\alpha)^\top \tilde{B}\gamma_0(\sigma_\alpha) = 0,$$

due to the fact that $\tilde{B}$ is skew-symmetric, independently of the considered value of the parameter α. $\qquad\square$

Theorem 6 *Assume that the parameter α in* (20) *is chosen according to* (21)*. Then,*

$$\sigma_\alpha(h) - y(h) = O(h^{2s+1}).$$

Proof Repeating similar steps as those in the proof of Theorem 2 (and using the same notation), and taking into account (20), one arrives at:[3]

$$\sigma_\alpha(h) - y(h)$$

$$= -h \left[\sum_{i,j\geq 0} \Psi_i(\sigma_\alpha)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \sum_{i,j=0}^{s-1} \Psi_i(\sigma_\alpha)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) + \alpha\Psi_0(\sigma_\alpha)\tilde{B}\gamma_0(\sigma_\alpha) \right]$$

$$= -h \left[\sum_{i=0}^{s-1}\sum_{j\geq s} \underbrace{\Psi_i(\sigma_\alpha)\rho_{ij}(\sigma_\alpha)}_{=O(h^j)}\gamma_j(\sigma_\alpha) + \sum_{i\geq s}\sum_{j=0}^{s-1} \Psi_i(\sigma_\alpha)\underbrace{\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha)}_{=O(h^i)} \right.$$

$$\left. + \sum_{i,j\geq s} \Psi_i(\sigma_\alpha)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) + \underbrace{\alpha\Psi_0(\sigma_\alpha)\tilde{B}\gamma_0(\sigma_\alpha)}_{=O(h^{2s})} \right] = O(h^{2s+1}).$$

$\qquad\square$

Remark 2 We observe that the modified polynomial σ_α in (20) is the solution of the approximate perturbed ODE-IVPs:

$$\dot{\sigma}_\alpha(ch) = \left[\left(B(\sigma_\alpha(ch)) + \alpha\tilde{B} \right) [\nabla H(\sigma_\alpha(ch))]_s \right]_s, \qquad c \in [0,1], \qquad \sigma_\alpha(0) = y_0, \tag{22}$$

---

[3]For sake of brevity, we skip here some of the intermediate passages, which are identical to those used in the proof of Theorem 2.

where the parameter α is such that $C(\sigma_\alpha(h)) = C(\sigma_\alpha(0))$. Clearly, when $\alpha = 0$, one recovers the problem (15) defining σ.

We end this section by sketching the case when we have r independent Casimirs, so that $C : \mathbb{R}^m \to \mathbb{R}^r$. In such a case, the notation introduced above formally still holds true, with the following differences:

- The Fourier coefficients (see (18)) $\pi_i(u_\alpha) \in \mathbb{R}^{m \times r}$, $i = 0, \ldots, s - 1$;
- The polynomial (20) now becomes

$$\dot{\sigma}_\alpha(ch) = \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \sum_{\ell=1}^{r} \alpha_\ell \tilde{B}_\ell \gamma_0(\sigma_\alpha), \qquad c \in [0,1], \qquad \sigma_\alpha(0) = y_0,$$
(23)

having set $\alpha = (\alpha_1, \ldots, \alpha_r)^\top$ and with $\tilde{B}_i^\top = -\tilde{B}_i$, $i = 1, \ldots, r$, arbitrary skew-symmetric matrices such that

$$M := \left[\pi_0(\sigma_\alpha)^\top \tilde{B}_1 \gamma_0(\sigma_\alpha), \ldots, \pi_0(\sigma_\alpha)^\top \tilde{B}_r \gamma_0(\sigma_\alpha) \right] \in \mathbb{R}^{r \times r}$$
(24)

is nonsingular;
- The vector α, providing the conservation of all Casimirs, is given by (compare with (21))

$$\alpha = M^{-1} \sum_{i,j=0}^{s-1} \pi_i(\sigma_\alpha)^\top \rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha).$$
(25)

3 Discretization

The procedure (12)–(14) described in the previous section is not yet a *ready to use* numerical method. In fact, in order for this to happen, the integrals $\gamma_j(\sigma)$, $\rho_{ij}(\sigma)$, $i, j = 0, \ldots, s-1$, defined in (6) need to be conveniently computed or approximated. For this purpose, as it has been done in the case of HBVMs [8], we shall use a Gauss-Legendre quadrature formula of order $2k$, i.e., the interpolatory quadrature rule based at the zeros of $P_k(c)$, with abscissae and weights (c_i, b_i), for a convenient value $k \geq s$. In so doing, we shall in general obtain a new polynomial approximation $u \in \Pi_s$, in place of σ as defined in (12)–(13):

$$\dot{u}(ch) = \sum_{i,j=0}^{s-1} P_i(c)\hat{\rho}_{ij}(u)\hat{\gamma}_j(u), \quad u(ch) = y_0 + h \sum_{i,j=0}^{s-1} \int_0^c P_i(x)\mathrm{d}x\,\hat{\rho}_{ij}(u)\hat{\gamma}_j(u), \qquad c \in [0,1],$$
(26)
$$\hat{\gamma}_j(u) = \sum_{\ell=1}^{k} b_\ell P_j(c_\ell)\nabla H(u(c_\ell h)), \quad \hat{\rho}_{ij}(u) = \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) P_j(c_\ell) B(u(c_\ell h)), \quad i, j = 0, \ldots, s-1.$$

Consequently, the new approximation to $y(h)$ will be given by

$$y_1 := u(h) = y_0 + h \sum_{j=0}^{s-1} \hat{\rho}_{0j}(u)\hat{\gamma}_j(u) \equiv y_0 + h \sum_{j=0}^{s-1} \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) B(u(c_\ell h)) \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) \nabla H(u(c_\ell h)),$$

(27)

which is the discrete counterpart of (14).

It is worth mentioning that, in a similar way as done in Section 2.1 for the polynomial σ, for u one obtains, by virtue of (26):

$$\begin{aligned}
\dot{u}(ch) &= \sum_{i,j=0}^{s-1} P_i(c)\hat{\rho}_{ij}(u)\hat{\gamma}_j(u) \\
&= \sum_{i=0}^{s-1} P_i(c) \sum_{j=0}^{s-1} \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) B(u(c_\ell h)) P_j(c_\ell) \sum_{\ell_1=1}^{k} b_{\ell_1} P_j(c_{\ell_1}) \nabla H(u(c_{\ell_1}h)) \\
&= \sum_{i=0}^{s-1} P_i(c) \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) B(u(c_\ell h)) \sum_{j=0}^{s-1} P_j(c_\ell) \sum_{\ell_1=1}^{k} b_{\ell_1} P_j(c_{\ell_1}) \nabla H(u(c_{\ell_1}h)) \\
&=: \sum_{i=0}^{s-1} P_i(c) \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) B(u(c_\ell h)) \left[\nabla H(u(c_\ell h)) \right]_s^{(2k)} \\
&\equiv \left[B(u(ch)) \left[\nabla H(u(ch)) \right]_s^{(2k)} \right]_s^{(2k)},
\end{aligned}$$

having set $[\cdot]_s^{(2k)}$ the approximate best approximation in Π_{s-1} obtained by using a quadrature of order $2k$ for approximating the involved integrals.[4] Consequently (compare with (15)), the polynomial u is the solution of the initial value problem:

$$\dot{u}(ch) = \left[B(u(ch)) \left[\nabla H(u(ch)) \right]_s^{(2k)} \right]_s^{(2k)}, \qquad c \in [0, 1], \qquad u(0) = y_0. \quad (28)$$

Remark 3 We observe that the polynomial approximation defined by the problem

$$\dot{u}(ch) = \left[B(u(ch)) \left[\nabla H(u(ch)) \right]_s \right]_s^{(2s)}, \qquad c \in [0, 1], \qquad u(0) = y_0,$$

corresponds to that provided by the methods in [19], when the Gauss-Legendre abscissae are used, and to the methods in [26, Definition 3.2], upon selecting the derivative space as Π_{s-1}. Similarly, some of the methods in [1] provide the approximation

$$\dot{u}(ch) = \left[B(u(ch)) \left[\nabla H(u(ch)) \right]_s^{(2k)} \right]_s^{(2s)}, \qquad c \in [0, 1], \qquad u(0) = y_0.$$

---

[4] I.e., $\lim_{p \to \infty} [\cdot]_s^{(p)} = [\cdot]_s$.

Remark 4 When in (28) $B(\sigma(ch)) \equiv J$, a constant skew-symmetric matrix, we derive the polynomial approximation provided by a HBVM(k, s) method:

$$u(ch) = y_0 + h \int_0^c [J\nabla H(u(\tau h))]_s^{(2k)} \, d\tau, \qquad c \in [0, 1]. \tag{29}$$

3.1 Analysis

As was done in Section 2.2 for the continuous procedure, let us now analyze the fully discrete method (26)–(27). To begin with, the following straightforward result holds true.

Theorem 7 *If*

$$B \in \Pi_\mu, \quad H \in \Pi_\nu, \qquad with \qquad \mu \le \frac{2k+1}{s} - 2, \quad \nu \le \frac{2k}{s}. \tag{30}$$

Then (see (26) and (6)),

$$\hat{\rho}_{ij}(u) = \rho_{ij}(u), \qquad \hat{\gamma}_j(u) = \gamma_j(u), \qquad \forall i, j = 0, \dots, s-1, \tag{31}$$

and, consequently, with reference to (26) and (13), one has $u \equiv \sigma$.

Proof In fact, if B is a polynomial of degree μ and H a polynomial of degree ν, the integrand defining $\rho_{ij}(u)$ has at most degree $\mu s + 2s - 2$, whereas that defining $\gamma_j(u)$ has at most degree $\nu s - 1$. Consequently, these degrees do not exceed $2k - 1$, when (30) holds true. As a result, the quadrature is exact, so that (31) is valid and, therefore, $u \equiv \sigma$. $\qquad \square$

Consequently, when (30) hold true, the method is energy-conserving and has order $2s$, as stated by Theorems 1 and 2, respectively.

Concerning energy conservation, the following additional result holds true, in the case where only H is a polynomial.

Theorem 8 *If*

$$H \in \Pi_\nu, \qquad with \qquad \nu \le \frac{2k}{s}, \tag{32}$$

then $H(y_1) = H(y_0)$.

Proof In fact, in such a case $\gamma_j(u) = \hat{\gamma}_j(u)$, $j = 0, \dots, s-1$, and the proof of Theorem 1 continues formally to hold, upon replacing σ with u, and ρ_{ij} with $\hat{\rho}_{ij}$, due to the fact that (compare with (8))

$$\hat{\rho}_{ij}(u) = \hat{\rho}_{ji}(u) = -\hat{\rho}_{ij}(u)^\top, \qquad \forall i, j = 0, 1, \dots, s-1. \tag{33}$$

$\qquad \square$

When (31) does not hold true, there is a quadrature error that, upon regularity assumptions, can be easily seen to be given by (see (6)):

$$\hat{\rho}_{ij}(u) - \rho_{ij}(u) = \chi_{ij}(h) \equiv O(h^{2k-i-j}), \tag{34}$$

$$\hat{\gamma}_j(u) - \gamma_j(u) = \Delta_j(h) \equiv O(h^{2k-j}), \qquad \forall i, j = 0, \ldots, s-1.$$

Nonetheless, also in this case it is straightforward to verify that (compare with (7)),

$$\forall k \geq s : \qquad \hat{\gamma}_j(u) = O(h^j), \qquad \hat{\rho}_{ij}(u) = O(h^{|i-j|}). \qquad \forall i, j = 0, 1, \ldots, s-1. \tag{35}$$

Consequently, with reference to the approximation y_1 defined in (27), the following result is easily obtained, when (32) is not valid.

Theorem 9 $\forall k \geq s : H(y_1) = H(y_0) + O(h^{2k+1}).$

Proof In fact, using arguments similar to those used in the proof of Theorem 1, one has, by taking into account (33)–(35):

$$H(y_1) - H(y_0) = H(u(h)) - H(u(0)) = \int_0^h \nabla H(u(t))^\top \dot{u}(t) \mathrm{d}t$$

$$= h \int_0^1 \nabla H(u(ch))^\top \dot{u}(ch) \mathrm{d}c = h \int_0^1 \nabla H(u(ch))^\top \sum_{i,j=0}^{s-1} P_i(c) \hat{\rho}_{ij}(u) \hat{\gamma}_j(u) \mathrm{d}c$$

$$= h \sum_{i,j=0}^{s-1} \underbrace{\left[\int_0^1 P_i(c) \nabla H(u(ch)) \mathrm{d}c \right]^\top}_{=\gamma_i(u)^\top} \hat{\rho}_{ij}(u) \left[\gamma_j(u) + \Delta_j(h) \right]$$

$$= h \sum_{i,j=0}^{s-1} \underbrace{\gamma_i(u)^\top \hat{\rho}_{ij}(u) \gamma_j(u)}_{=0} + h \sum_{i,j=0}^{s-1} \gamma_i(u)^\top \hat{\rho}_{ij}(u) \Delta_j(h)$$

$$= h \left[\sum_{j=0}^{s-1} \sum_{i=j}^{s-1} \underbrace{\gamma_i(u)^\top \hat{\rho}_{ij}(u)}_{=O(h^{2i-j})} \Delta_j(h) + \sum_{j=0}^{s-1} \sum_{i=0}^{j-1} \gamma_i(u)^\top \underbrace{\hat{\rho}_{ij}(u) \Delta_j(h)}_{=O(h^{2k-i})} \right] = O(h^{2k+1}).$$

$\square$

Concerning the accuracy of the approximation (27), the following result, stating that the convergence order of Theorem 2 is retained, holds true.

Theorem 10 $\forall k \geq s : y_1 - y(h) = O(h^{2s+1}).$

Proof By using arguments and notations similar to those used in the proof of Theorem 2, one has, by taking into account (34) and that $k \geq s$:

$$y_1 - y(h) = u(h) - y(h) = y(h, h, u(h)) - y(h, 0, u(0)) = \int_0^h \frac{d}{dt} y(h, t, u(t)) dt$$

$$= \int_0^h \left[\frac{\partial}{\partial \xi} y(h, \xi, u(t)) \Big|_{\xi=t} + \frac{\partial}{\partial \eta} y(h, t, \eta) \Big|_{\eta=u(t)} \dot{u}(t) \right] dt$$

$$= \int_0^h \left[-\Phi(h, t, u(t)) F(u(t)) + \Phi(h, t, u(t)) \dot{u}(t) \right] dt$$

$$= -h \int_0^1 \Phi(h, ch, u(ch)) \left[F(u(ch)) - \dot{u}(ch) \right] dc$$

$$= -h \int_0^1 \Phi(h, ch, u(ch)) \left[B(u(ch)) \nabla H(u(ch)) - \sum_{i,j=0}^{s-1} P_i(c) \hat{\rho}_{ij}(u) \hat{\gamma}_j(u) \right] dc$$

$$= -h \underbrace{\int_0^1 \Phi(h, ch, u(ch)) \left[\sum_{i,j \geq 0} P_i(c) \rho_{ij}(u) \gamma_j(u) - \sum_{i,j=0}^{s-1} P_i(c) \rho_{ij}(u) \gamma_j(u) \right] dc}_{=O(h^{2s+1}), \text{ from the proof of Theorem 2}}$$

$$-h \int_0^1 \Phi(h, ch, u(ch)) \left[\sum_{i,j=0}^{s-1} P_i(c) \rho_{ij}(u) \gamma_j(u) - \sum_{i,j=0}^{s-1} P_i(c) \hat{\rho}_{ij}(u) \hat{\gamma}_j(u) \right] dc$$

$$= O(h^{2s+1}) - h \sum_{i,j=0}^{s-1} \underbrace{\int_0^1 P_i(c) \Phi(h, ch, u(ch)) dc}_{=\Psi_i(u)} \left[\rho_{ij}(u) \gamma_j(u) - \hat{\rho}_{ij}(u) \hat{\gamma}_j(u) \right]$$

$$= O(h^{2s+1}) - h \sum_{i,j=0}^{s-1} \Psi_i(u) \left[\rho_{ij}(u) \gamma_j(u) - \left(\rho_{ij}(u) + \chi_{ij}(h) \right) \left(\gamma_j(u) + \Delta_j(h) \right) \right]$$

$$= O(h^{2s+1}) + h \sum_{i,j=0}^{s-1} \Psi_i(u) \left[\rho_{ij}(u) \Delta_j(h) + \underbrace{\chi_{ij}(h) \gamma_j(u)}_{=O(h^{2k-i})} + \underbrace{\chi_{ij}(h) \Delta_j(h)}_{=O(h^{4k-2j-i})} \right]$$

$$= O(h^{2s+1}) + O(h^{2k+1}) + h \sum_{i,j=0}^{s-1} \Psi_i(u) \rho_{ij}(u) \Delta_j(h) = O(h^{2s+1}) + h \sum_{i,j=0}^{s-1} \Psi_i(u) \rho_{ij}(u) \Delta_j(h).$$

Concerning the latter sum, one has:

$$h \sum_{i,j=0}^{s-1} \Psi_i(u) \rho_{ij}(u) \Delta_j(h) = h \sum_{i=0}^{s-1} \sum_{j=i}^{s-1} \underbrace{\Psi_i(u) \rho_{ij}(u)}_{=O(h^j)} \Delta_j(h) + h \sum_{i=0}^{s-1} \sum_{j=0}^{i-1} \Psi_i(u) \underbrace{\rho_{ij}(u) \Delta_j(h)}_{=O(h^{2k-2j+i})}$$

$$= O(h^{2k+1}) + O(h^{2k+2}) = O(h^{2k+1}),$$

and, consequently, the statement follows. □

Remark 5 By taking into account the results of Theorems 7 and 9, it follows that, by choosing k large enough, one obtains either an exact conservation of the Hamiltonian function, in the polynomial case, or a *practical* conservation, in the non polynomial case. In fact, as it has been also observed for HBVMs [11], in the latter case it is enough to choose k large enough so that the Hamiltonian error falls within the round-off error level of the finite precision arithmetic used in the simulation.

It is worth mentioning that a result similar to that of Theorem 3 holds true for the fully discrete method.

Theorem 11 *The method* (26)–(27) *is symmetric, provided that the abscissae of the quadrature satisfy* [5]

$$c_{k-i+1} = 1 - c_i, \qquad i = 1, \ldots, k. \tag{36}$$

Proof Symmetry of a given one-step method $y_1 = \Phi_h(y_0)$ applied to an initial value problem $y' = f(y)$ with $y(0) = y_0$, means that $\Phi_h^{-1} = \Phi_{-h}$ that is, applying the method to the state vector y_1, but with the direction of time reversed, yields the initial state vector y_0, independently of the choice of the initial value y_0. In our context, with reference to (26)–(27), Φ_h is defined by

$$y_1 = y_0 + h \sum_{j=0}^{s-1} \hat{\rho}_{0j} \hat{\gamma}_j, \tag{37}$$

where $\hat{\rho}_{ij}$ and $\hat{\gamma}_j$ are the solutions of the nonlinear system

$$\hat{\gamma}_j = \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) \nabla H \left(y_0 + h \sum_{\mu,\nu=0}^{s-1} \int_0^{c_\ell} P_\mu(x) \mathrm{d}x \, \hat{\rho}_{\mu\nu} \hat{\gamma}_\nu \right),$$

$$\hat{\rho}_{ij} = \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) P_j(c_\ell) B \left(y_0 + h \sum_{\mu,\nu=0}^{s-1} \int_0^{c_\ell} P_\mu(x) \mathrm{d}x \, \hat{\rho}_{\mu\nu} \hat{\gamma}_\nu \right),$$

$$i, j = 0, \ldots, s - 1. \tag{38}$$

We can obtain the explicit formulation of $\bar{y}_0 = \Phi_{-h}(y_1)$ by introducing in (37)–(38) the following substitutions: y_1 in place of y_0 and $-h$ in place of h. In so doing, we arrive at the method defined as

$$\bar{y}_0 = y_1 - h \sum_{j=0}^{s-1} \bar{\rho}_{0j} \bar{\gamma}_j, \tag{39}$$

---

[5]This is, in fact, the case, for the Gauss-Legendre quadrature abscissae.

where the unknown quantities $\bar{\rho}_{ij}$ and $\bar{\gamma}_j$ satisfy the following nonlinear system

$$\bar{\gamma}_j = \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) \nabla H \left(y_1 - h \sum_{\mu,\nu=0}^{s-1} \int_0^{c_\ell} P_\mu(x) dx \, \bar{\rho}_{\mu\nu} \bar{\gamma}_\nu \right),$$

$$\bar{\rho}_{ij} = \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) P_j(c_\ell) B \left(y_1 - h \sum_{\mu,\nu=0}^{s-1} \int_0^{c_\ell} P_\mu(x) dx \, \bar{\rho}_{\mu\nu} \bar{\gamma}_\nu \right),$$

$$i, j = 0, \ldots, s-1, \tag{40}$$

and we want to show that $\bar{y}_0 = y_0$. To this end, we introduce the following variables:

$$\gamma_j^* := (-1)^j \hat{\gamma}_j, \quad \rho_{ij}^* := (-1)^{i+j} \hat{\rho}_{ij}, \quad i, j = 0, \ldots, s-1.$$

Exploiting the symmetry property of Legendre polynomials, $(-1)^j P_j(c) = P_j(1 - c)$, from the first equation in (38) we get

$$\gamma_j^* = \sum_{\ell=1}^{k} b_\ell (-1)^j P_j(c_\ell) \nabla H \left(y_0 + h \sum_{\mu,\nu=0}^{s-1} \left(\int_0^1 P_\mu(x) dx - \int_{c_\ell}^1 P_\mu(x) dx \right) \hat{\rho}_{\mu\nu} \hat{\gamma}_\nu \right)$$

$$= \sum_{\ell=1}^{k} b_\ell P_j(1 - c_\ell) \nabla H \left(y_1 - h \sum_{\mu,\nu=0}^{s-1} \int_{c_\ell}^1 P_\mu(x) dx \, \hat{\rho}_{\mu\nu} \hat{\gamma}_\nu \right).$$

Introducing the change of variables $\tau = 1 - x$, transforms the latter integral as

$$\int_{c_\ell}^1 P_\mu(x) dx = -\int_{1-c_\ell}^0 P_\mu(1 - \tau) d\tau = (-1)^\mu \int_0^{1-c_\ell} P_\mu(\tau) d\tau.$$

Exploiting the symmetry assumption (36), which in turn implies symmetric weights, $b_{k-i+1} = b_i, i = 1, \ldots, k$, we finally get

$$\gamma_j^* = \sum_{\ell=1}^{k} b_{k-\ell+1} P_j(c_{k-\ell+1}) \nabla H \left(y_1 - h \sum_{\mu,\nu=0}^{s-1} \int_0^{c_{k-\ell+1}} P_\mu(x) dx (-1)^\mu (-1)^\nu \hat{\rho}_{\mu\nu} (-1)^\nu \hat{\gamma}_\nu \right)$$

$$= \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) \nabla H \left(y_1 - h \sum_{\mu,\nu=0}^{s-1} \int_0^{c_\ell} P_\mu(x) dx \rho_{\mu\nu}^* \gamma_\nu^* \right).$$

The same flow of computation may be employed on the second equation in (38) to see that

$$\rho_{ij}^* = \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) P_j(c_\ell) B \left(y_1 - h \sum_{\mu,\nu=0}^{s-1} \int_0^{c_\ell} P_\mu(x) dx \rho_{\mu\nu}^* \gamma_\nu^* \right).$$

We then realize that γ_j^* and ρ_{ij}^* satisfy the very same nonlinear system (40) governing the quantities $\bar{\gamma}_j$ and $\bar{\rho}_{ij}$. Thus we may conclude that

$$\bar{\gamma}_j = (-1)^j \hat{\gamma}_j, \quad \bar{\rho}_{ij} = (-1)^{i+j} \hat{\rho}_{ij}, \quad i, j = 0, \ldots, s-1,$$

and hence from (39),

$$\bar{y}_0 = y_1 - h \sum_{j=0}^{s-1} (-1)^j \hat{\rho}_{0j} (-1)^j \hat{\gamma}_j = y_1 - h \sum_{j=0}^{s-1} \hat{\rho}_{0j} \hat{\gamma}_j = y_0.$$

□

In the limit case when $k \to \infty$, since the Gauss-Legendre quadrature formule are convergent, we are led back to the symmetry property of the original non-discretized procedure (12)–(14) that we anticipated in Theorem 3.

We conclude this section by emphasizing that, when problem (1) is in the form (3), then (see (26)) $\hat{\rho}_{ij}(\sigma) = \delta_{ij} J$ and, consequently, the polynomial approximation u becomes

$$u(ch) = y_0 + h \sum_{j=0}^{s-1} \int_0^c P_j(x) \mathrm{d}x \sum_{i=1}^k b_i P_j(c_i) J \nabla H(u(c_i h)), \qquad c \in [0, 1],$$

which is equivalent to (29). As anticipated in Remark 4, this equation (see, e.g., the monograph [5] or the review paper [6, Section 2.2]) defines a *Hamiltonian Boundary Value Method with parameters k and s, in short HBVM(k, s)*. Moreover, also Theorems 7–10 exactly describe, in the case of problem (3), their conservation and accuracy properties. For this reason, the methods here presented can be regarded as a generalization of HBVMs for solving Poisson problems and, therefore, this fact, motivates the following definition.

Definition 1 We shall refer to the method (26)–(27) as to a *PHBVM(k, s) method*.

3.2 Conservation of Casimirs

This section is devoted to the conservation of Casimirs for the discrete PHBVM(k, s) method (26)–(27), following similar steps as those in Section 2.3. For this purpose, it is convenient to define the approximate Fourier coefficients, for $k \geq s$, which we assume hereafter:

$$\hat{\pi}_i(\sigma) = \sum_{\ell=1}^k b_\ell P_i(c_\ell) \nabla C(\sigma(c_\ell h)) = O(h^j), \qquad i = 0, \ldots, s - 1. \tag{41}$$

The following straightforward result is reported without proof.

Lemma 3 *Assume that $\sigma \in \Pi_s$. Then, with reference to (18), for all $i = 0, \ldots, s-1$, one has:*

$$\hat{\pi}_i(\sigma) = \pi_i(\sigma), \qquad if \qquad C \in \Pi_\nu, \quad \nu \leq 2k/s,$$

$$\hat{\pi}_i(\sigma) = \pi_i(\sigma) - \theta_i(h), \qquad \theta_i(h) = O(h^{2k-i}), \qquad otherwise.$$

Next, let us consider the following perturbed polynomial, in place of the polynomial u in (26),

$$\dot{u}_\alpha(ch) = \sum_{i,j=0}^{s-1} P_i(c)\hat{\rho}_{ij}(u_\alpha)\hat{\gamma}_j(u_\alpha) - \alpha \tilde{B}\hat{\gamma}_0(u_\alpha), \qquad c \in [0,1], \qquad u_\alpha(0) = y_0,$$

(42)

with $\tilde{B}^\top = -\tilde{B} \neq O$ an arbitrary matrix, and (compare with (21))

$$\alpha = \frac{\sum_{i,j=0}^{s-1} \hat{\pi}_i(u_\alpha)^\top \hat{\rho}_{ij}(u_\alpha)\hat{\gamma}_j(u_\alpha)}{\hat{\pi}_0(u_\alpha)^\top \tilde{B}\hat{\gamma}_0(u_\alpha)} = O(h^{2s}).$$

(43)

Theorem 12 *If*

$$B \in \Pi_\mu, \quad C, H \in \Pi_\nu, \qquad with \qquad \mu \leq \frac{2k+1}{s} - 2, \quad \nu \leq \frac{2k}{s}.$$

(44)

Then (see (26), (6)), (18), and (41)

$$\hat{\rho}_{ij}(u_\alpha) = \rho_{ij}(u_\alpha), \qquad \hat{\gamma}_j(u_\alpha) = \gamma_j(u_\alpha), \qquad \hat{\pi}_i(u_\alpha) = \pi_i(u_\alpha), \qquad \forall i,j = 0,\ldots,s-1,$$

(45)

and, consequently, with reference to (42) and (20), one has $u_\alpha \equiv \sigma_\alpha$.

Differently, by setting $y_1 := u_\alpha(h)$ the new approximation, Theorems 8, 9, 10, and 11 continue formally to hold. Moreover, the following result holds true.

Theorem 13 *With reference to (42)–(43), one has:*

$$C(y_1) - C(y_0) = \begin{cases} 0, & if \quad C \in \Pi_\nu, \quad with \quad \nu \leq 2k/s, \\ O(h^{2k+1}), & otherwise. \end{cases}$$

Proof By taking into account Lemma 3, one obtains:

$$C(y_1) - C(y_0) = h \int_0^1 \nabla C(u_\alpha(ch))^\top \dot{u}_\alpha(ch)dc$$

$$= h \int_0^1 \nabla C(u_\alpha(ch))^\top \left[\sum_{i,j=0}^{s-1} P_i(c)\hat{\rho}_{ij}(u_\alpha)\hat{\gamma}_j(u_\alpha) - \alpha \tilde{B}\hat{\gamma}_0(u_\alpha)\right]dc$$

$$= h \left[\sum_{i,j=0}^{s-1} \left(\int_0^1 \nabla C(u_\alpha(ch))P_i(c)dc\right)^\top \hat{\rho}_{ij}(u_\alpha)\hat{\gamma}_j(u_\alpha) - \alpha \left(\int_0^1 \nabla C(u_\alpha(ch))P_i(c)dc\right)^\top \tilde{B}\hat{\gamma}_0(u_\alpha)\right]dc$$

$$= h \left[\sum_{i,j=0}^{s-1} \pi_i(u_\alpha)^\top \hat{\rho}_{ij}(u_\alpha)\hat{\gamma}_j(u_\alpha) - \alpha \pi_0(u_\alpha)^\top \tilde{B}\hat{\gamma}_0(u_\alpha)\right] =: (*).$$

In case $C \in \Pi_\nu$ with $\nu \leq 2k/s$, then, by virtue of Lemma 3, $\pi_i(u_\alpha) = \hat{\pi}_i(u_\alpha)$ and, consequently, $(*) = 0$ because of (43). Conversely, again from Lemma 3, one has:

$$(*) = h \left[\sum_{i,j=0}^{s-1} \left[\hat{\pi}_i(u_\alpha) + \theta_i(h) \right]^\top \hat{\rho}_{ij}(u_\alpha) \hat{\gamma}_j(u_\alpha) - \alpha \left[\hat{\pi}_0(u_\alpha) + \theta_0(h) \right]^\top \tilde{B} \hat{\gamma}_0(u_\alpha) \right]$$

$$= h \left[\underbrace{\sum_{i,j=0}^{s-1} \theta_i(h)^\top \hat{\rho}_{ij}(u_\alpha) \hat{\gamma}_j(u_\alpha)}_{=O(h^{2k})} - \underbrace{\alpha \, \theta_0(h)^\top \tilde{B} \hat{\gamma}_0(u_\alpha)}_{=O(h^{2k+2s})} \right] = O(h^{2k+1}).$$

$\square$

Remark 6 From the arguments here exposed, one deduces that when using a finite precision arithmetic, a (at least) practical conservation of both the Hamiltonian and the Casimirs can be obtained, by choosing k large enough.

Definition 2 Following [18], we name *enhanced PHBVM(k, s)*, in short *EPHBVM(k, s)*, the method defined by (42)–(43).

Following Remark 2, the polynomial u_α defined by an EPHBVM(k, s) method is the solution of the ODE-IVP (compare with (22)):

$$\dot{u}_\alpha(ch) = \left[\left(B(u_\alpha(ch)) + \alpha \tilde{B} \right) \left[\nabla H(u_\alpha(ch)) \right]_s^{(2k)} \right]_s^{(2k)}, \quad c \in [0, 1], \quad u_\alpha(0) = y_0.$$

Clearly, when $\alpha = 0$ one retrieves the problem (28) defining the polynomial approximation of the PHBVM(k, s) method.

Remark 7 From the results of Theorems 8, 9, and 13, one deduces the clear advantage of choosing values of k suitably larger than s, in order to obtain a suitable conservation of non-quadratic Hamiltonians and/or Casimirs. This fact will be duly confirmed in the numerical tests. Moreover, this is not a serious drawback, from the point of view of the computational cost, since, as we shall see in the next section, the discrete problem to be solved will always have (block) dimension s, independently of k.[6]

4 The discrete problem

In this section we deal with the efficient solution of the discrete problem generated by the PHBVM(k, s) method (26). For this purpose, we observe that only the values

---

[6]This latter feature is inherited, in turn, from the original methods, HBVM(k, s) and EHBVM(k, s) for Hamiltonian problems.

$Y_\ell := u(c_\ell h)$, $\ell = 1, \ldots, k$, are actually needed. Consequently, (26) can be rewritten as:

$$Y_\ell = y_0 + h \sum_{i,j=0}^{s-1} \int_0^{c_\ell} P_i(x)\mathrm{d}x\, \hat{\rho}_{ij}\hat{\gamma}_j, \qquad \ell = 1, \ldots, k, \tag{46}$$

$$\hat{\gamma}_j = \sum_{\ell=1}^k b_\ell P_j(c_\ell)\nabla H(Y_\ell), \qquad \hat{\rho}_{ij} = \sum_{\ell=1}^k b_\ell P_i(c_\ell) P_j(c_\ell) B(Y_\ell), \qquad i,j = 0, \ldots, s-1,$$

where, for sake of brevity, we have omitted the argument u for $\hat{\gamma}_j$ and $\hat{\rho}_{ij}$, as was already done in (37)–(38). The (46) can be cast in matrix form by defining the block vectors and matrices

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_k \end{pmatrix} \in \mathbb{R}^{k \cdot m}, \quad \boldsymbol{\gamma} = \begin{pmatrix} \hat{\gamma}_0 \\ \vdots \\ \hat{\gamma}_{s-1} \end{pmatrix} \in \mathbb{R}^{s \cdot m}, \quad \Gamma = \begin{pmatrix} \hat{\rho}_{00} & \cdots & \hat{\rho}_{0,s-1} \\ \vdots & & \vdots \\ \hat{\rho}_{s-1,0} & \cdots & \hat{\rho}_{s-1,s-1} \end{pmatrix} \in \mathbb{R}^{s \cdot m \times s \cdot m}, \tag{47}$$

and

$$\boldsymbol{e} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^k, \quad \Omega = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_k \end{pmatrix}, \quad \mathcal{I}_s = \left(\int_0^{c_i} P_{j-1}(x)\mathrm{d}x \right), \quad \mathcal{P}_s = \left(P_{j-1}(c_i) \right) \in \mathbb{R}^{k \times s}. \tag{48}$$

In fact, by also setting hereafter I_r the identity of dimension r, we can rewrite (46) as:

$$Y = \boldsymbol{e} \otimes y_0 + h(\mathcal{I}_s \otimes I_m)\Gamma\boldsymbol{\gamma}, \quad \boldsymbol{\gamma} = (\mathcal{P}_s^\top \Omega \otimes I_m)\nabla H(Y), \quad \Gamma = (\mathcal{P}_s^\top \Omega \otimes I_m)\mathcal{B}(Y)(\mathcal{P}_s \otimes I_m), \tag{49}$$

where

$$\nabla H(Y) = \begin{pmatrix} \nabla H(Y_1) \\ \vdots \\ \nabla H(Y_k) \end{pmatrix} \qquad \text{and} \qquad \mathcal{B}(Y) = \begin{pmatrix} B(Y_1) & & \\ & \ddots & \\ & & B(Y_k) \end{pmatrix}.$$

As is clear, the discrete problem (49) can be further reformulated in terms of the product of the Fourier coefficients $\hat{\rho}_{ij}$ and $\hat{\gamma}_j$. In fact, by setting

$$\boldsymbol{\phi} \equiv \begin{pmatrix} \phi_0 \\ \vdots \\ \phi_{s-1} \end{pmatrix} := \Gamma\boldsymbol{\gamma} \qquad \Rightarrow \qquad \phi_i = \sum_{j=0}^{s-1} \hat{\rho}_{ij}\hat{\gamma}_j, \quad i = 0, \ldots, s-1, \tag{50}$$

the first equation in (49) becomes

$$Y = \boldsymbol{e} \otimes y_0 + h\mathcal{I}_s \otimes I_m\boldsymbol{\phi}, \tag{51}$$

whereas, multiplying side by side the third by the second gives

$$\boldsymbol{\phi} = (\mathcal{P}_s^\top \Omega \otimes I_m)\, \mathcal{B}(Y)\, (\mathcal{P}_s \mathcal{P}_s^\top \Omega \otimes I_m) \nabla H(Y). \tag{52}$$

Consequently, by substituting the right-hand side of (51) in the right-hand side of (52), provides the new discrete problem:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\phi}) &:= \boldsymbol{\phi} - \mathcal{P}_s^\top \Omega \otimes I_m\, \mathcal{B}\,(\boldsymbol{e} \otimes y_0 + h\mathcal{I}_s \otimes I_m\boldsymbol{\phi})\,(\mathcal{P}_s \mathcal{P}_s^\top \Omega \otimes I_m)\nabla H\,(\boldsymbol{e} \otimes y_0 + h\mathcal{I}_s \otimes I_m\boldsymbol{\phi}) \\
&= \boldsymbol{0}.
\end{aligned} \tag{53}$$

Moreover, computing the vector $\boldsymbol{\phi}$ in (50) allows us to obtain the new approximation (27) as:

$$y_1 = y_0 + h\phi_0.$$

Remark 8 In case where the problem (1) is in the form (3), one has that $\mathcal{B}(Y) = I_s \otimes J$. Consequently, considering that

$$\mathcal{P}_s^\top \Omega \mathcal{P}_s = I_s, \tag{54}$$

the discrete problem (53) reduces to:

$$\boldsymbol{\phi} - \mathcal{P}_s^\top \Omega \otimes J \nabla H\,(\boldsymbol{e} \otimes y_0 + h\mathcal{I}_s \otimes J\boldsymbol{\phi}) = \boldsymbol{0}.$$

This latter problem is exactly that generated by a HBVM(k, s) method applied for solving (3) [9]. We observe, however, that while the original HBVM(k, s) method is actually a k-stage Runge-Kutta method with Butcher tableau (see (48))

$$\begin{array}{c|c} \boldsymbol{c} & \mathcal{I}_s \mathcal{P}_s^\top \Omega \\ \hline & \boldsymbol{b}^\top \end{array}, \qquad \boldsymbol{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix}, \qquad \boldsymbol{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix},$$

this is no more the case for the generalization defined by (49).

Remark 9 In the case $k = s$, one has that $\mathcal{P}_s \mathcal{P}_s^\top \Omega = I_s$. Consequently, (53) becomes, by using the notation (16),

$$\boldsymbol{\phi} = (\mathcal{P}_s^\top \Omega \otimes I_m)\, F(\boldsymbol{e} \otimes y_0 + h\mathcal{I}_s \otimes I_m\boldsymbol{\phi}).$$

This, in turn, is equivalent to the application of the s-stage Gauss method to the problem (1) (see, e.g., [9]).

Instead, in the case $k > s$, the discrete problem (53) is equivalent to the application of the HBVM(k, s) method to the problem (see (28)),

$$\dot{y} = B(y)[\nabla H(y)]_s^{(2k)}, \qquad t > 0, \qquad y(0) = y_0,$$

in place of (1). This application, in turn, provides the polynomial approximation (28).

We observe that the formulation (53) naturally induces a straightforward iterative procedure for solving the discrete problem,

$$
\begin{aligned}
\boldsymbol{\phi}^{r+1} = {} & \\
& \mathcal{P}_s^\top \Omega \otimes I_m \, \mathcal{B} \left(\boldsymbol{e} \otimes y_0 + h \mathcal{I}_s \otimes I_m \boldsymbol{\phi}^r \right) \left(\mathcal{P}_s \mathcal{P}_s^\top \Omega \otimes I_m \right) \nabla H \left(\boldsymbol{e} \otimes y_0 + h \mathcal{I}_s \otimes I_m \boldsymbol{\phi}^r \right), \\
& r = 0, 1, \ldots,
\end{aligned}
\tag{55}
$$

for which the initial approximation $\boldsymbol{\phi}^0 = \mathbf{0}$ can be conveniently used. It is also possible to use the simplified Newton iteration for solving (53) which, taking into account (48), (54), and that (see, e.g., [5])

$$
\mathcal{P}_s^\top \Omega \mathcal{I}_s = X_s := \begin{pmatrix} \xi_0 & -\xi_1 & & & \\ \xi_1 & 0 & \ddots & & \\ & \ddots & \ddots & -\xi_{s-1} & \\ & & \xi_{s-1} & 0 \end{pmatrix}, \qquad \xi_i = \left(2\sqrt{|4i^2 - 1|} \right)^{-1}, \quad i = 0, \ldots, s-1,
\tag{56}
$$

takes the form:

$$
\text{solve: } \left[I_s \otimes I_m - h X_s \otimes F'(y_0) \right] \boldsymbol{\delta}^r = -\mathcal{F}(\boldsymbol{\phi}^r), \qquad \boldsymbol{\phi}^{r+1} := \boldsymbol{\phi}^r + \boldsymbol{\delta}^r, \qquad r = 0, 1, \ldots, \tag{57}
$$

with F' the Jacobian of F (see (16)). Nevertheless, the iteration (57) requires the factorization of a matrix having size s times larger than that of problem (1), which can be costly, when s and/or m are large. Consequently, it is much more effective to resort to a *blended* iteration for solving (53) (see, e.g., [9], we also refer to [13] for a more detailed analysis of *blended methods*). In the present case, this latter iteration, considering the matrix X_s defined in (56), denoting $\sigma(X_s)$ its spectrum, and setting

$$
\Lambda := I_m - h \lambda_s F'(y_0) \in \mathbb{R}^{m \times m}, \qquad \text{with} \qquad \lambda_s = \min_{\lambda \in \sigma(X_s)} |\lambda|, \tag{58}
$$

assumes the form:

$$
\boldsymbol{\eta}^r := -\mathcal{F}(\boldsymbol{\phi}^r), \qquad \boldsymbol{\eta}_1^r := \lambda_s X_s^{-1} \otimes I_m \, \boldsymbol{\eta}^r, \tag{59}
$$

$$
\boldsymbol{\phi}^{r+1} := \boldsymbol{\phi}^r + I_s \otimes \Lambda^{-1} \left(\boldsymbol{\eta}_1^r + I_s \otimes \Lambda^{-1} \left(\boldsymbol{\eta}^r - \boldsymbol{\eta}_1^r \right) \right), \qquad r = 0, 1, \ldots.
$$

Consequently, only the matrix Λ in (58), having the same size as that of the continuous problem (1), needs to be factored. This fact is a common feature, in the many instances where the blended iteration can be used. For this reason, in such cases, it turns out to be extremely efficient (see, e.g., [2, 14–16, 23]).

Remark 10 In the practical use of the methods, it is customary to choose the parameter k, related to the order of the quadrature, so that the discretization error falls within

the round-off error level. Nevertheless, round-off errors are unavoidable, as are iteration errors in (55) or (59). This may cause a small numerical drift in the invariants,[7] even in the case where the quadrature is exact. This phenomenon has been duly studied in [5, Chapter 4.3], where a simple correction procedure is given to avoid this problem. The same procedure can be conveniently used in this setting, too. The reader is referred to the above reference for full details.

4.1 Conservation of Casimirs

In this section we sketch the implementation of EPHBVM(k, s) methods described in Section 3.2. For this purpose, besides the vector $\boldsymbol{\phi}$ defined in (50), we need to define the block vector

$$\hat{\boldsymbol{\pi}} = \begin{pmatrix} \hat{\pi}_0 \\ \vdots \\ \hat{\pi}_{s-1} \end{pmatrix} \tag{60}$$

with the approximate Fourier coefficients (41) of the gradient of the Casimir.[8] In so doing, the discrete problem generated by an EHBVM(k, s) method becomes:

$$\mathcal{F}(\boldsymbol{\phi}, \alpha) := \begin{pmatrix} \boldsymbol{\phi} - \mathcal{P}_s^\top \Omega \otimes I_m \, \mathcal{B}\,(Y)\,(\mathcal{P}_s \mathcal{P}_s^\top \Omega \otimes I_m) \nabla H\,(Y) \\ \alpha - \frac{\hat{\boldsymbol{\pi}}^\top \boldsymbol{\phi}}{\hat{\pi}_0^\top \tilde{B} \hat{\gamma}_0} \end{pmatrix} = \mathbf{0}, \quad (61)$$

with

$$\begin{aligned} Y &= \boldsymbol{e} \otimes y_0 + h \mathcal{I}_s \otimes I_m \boldsymbol{\phi} - \alpha h \boldsymbol{c} \otimes (\tilde{B}\hat{\gamma}_0), \\ \hat{\gamma}_0 &= \boldsymbol{b}^\top \otimes I_m \, \nabla H(Y), \\ \hat{\boldsymbol{\pi}} &= \mathcal{P}_s^\top \Omega \otimes I_m \, \nabla C(Y), \end{aligned}$$

and the new approximation given by

$$y_1 = y_0 + h\left(\phi_0 - \alpha\,\tilde{B}\hat{\gamma}_0\right).$$

We conclude this section by mentioning that, in case of multiple Casimirs, the discrete problem (61) can be readily generalized, by considering discrete counterparts of (24)–(25).

5 Numerical tests

In this section we present a couple of numerical tests concerning the solution of Lotka-Volterra problems, with the last one possessing a Casimir. The numerical tests have been carried out on a 3GHz Intel Xeon W10 core computer with 64GB of memory running Matlab 2020a.

---

[7] Usually, this is a very small drift, of the order of the machine epsilon per step.

[8] As before, for sake of brevity we now omit the argument u_α of the approximate Fourier coefficients.

Example 1 We consider the following Lotka-Volterra problem:

$$\dot{y} = \begin{pmatrix} 0 & y_1 y_2 \\ -y_1 y_2 & 0 \end{pmatrix} \nabla H(y),$$

$$H(y) = a \left(\ln y_1 - \frac{y_1}{y_1^*} \right) + b \left(\ln y_2 - \frac{y_2}{y_2^*} \right), \tag{62}$$

with

$$a = 1, \qquad b = 3, \qquad y_1^* = y_2^* = 1, \qquad y(0) = (5, \ 1)^\top,$$

whose solution, which is periodic of period $T \approx 4.633434168477889$, is depicted in Fig. 1. At first, we solve the problem on one period with time-step $h = T/n$, by using the following methods:

- The s-stage Gauss method, $s = 1, 2, 3$;
- The PHBVM(4, s), $s = 1, 2$, and PHBVM(6, 3) methods, which become soon energy-conserving, as the value of n is increased.

The obtained results are summarized in Table 1, where we have denoted by e_y and e_H the error in the solution and in the Hamiltonian after one period, respectively. Their numerical rate of convergence is also reported, along with the mean blended iterations (58)–(59) per time-step (it), in order to obtain convergence within full machine accuracy, and the execution time in *sec*. From the listed results, one infers that:

- As is striking clear, the higher order methods are much more efficient than the lower order ones, especially when a high accuracy is required;
- The theoretical rate of convergence for both the solution and the Hamiltonian errors is that we expected (for PHBVMs, until the Hamiltonian error falls within the round-off error level);
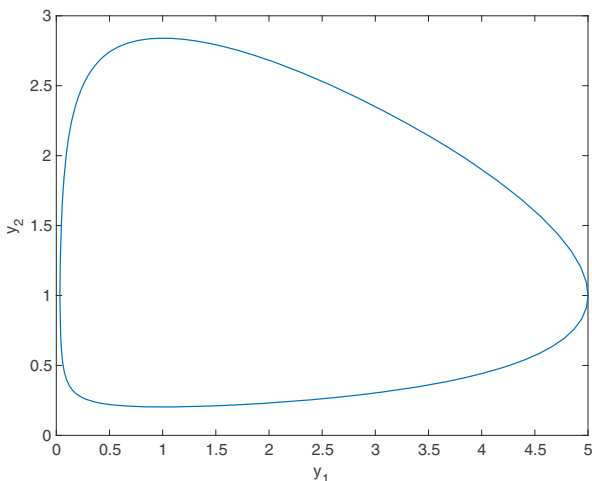

Fig. 1 Solution of problem (62)

Table 1 Results for problem (62)

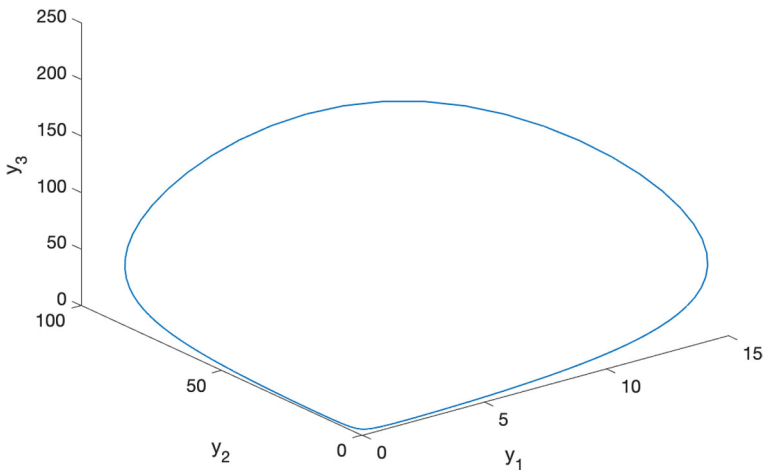| n	e_y	Rate	e_H	Rate	it	Time	e_y	Rate	e_H	Rate	it	Time
	Gauss-1						PHBVM(4, 1)					
50	3.54e-02	—	4.47e-02	—	7.4	0.01	7.64e-02	—	1.72e-07	—	8.5	0.02
100	8.56e-03	2.1	1.09e-02	2.0	5.8	0.01	1.85e-02	2.0	6.48e-10	8.1	6.7	0.03
200	2.12e-03	2.0	2.71e-03	2.0	5.1	0.02	4.58e-03	2.0	2.37e-12	8.1	5.5	0.05
400	5.29e-04	2.0	6.77e-04	2.0	4.4	0.04	1.14e-03	2.0	8.88e-16	**	4.6	0.09
800	1.32e-04	2.0	1.69e-04	2.0	4.1	0.07	2.86e-04	2.0	8.88e-16	**	4.3	0.16
1600	3.30e-05	2.0	4.23e-05	2.0	3.5	0.13	7.14e-05	2.0	1.78e-15	**	4.0	0.30
3200	8.25e-06	2.0	1.06e-05	2.0	3.2	0.25	1.79e-05	2.0	8.88e-16	**	3.4	0.56
6400	2.06e-06	2.0	2.64e-06	2.0	3.1	0.48	4.46e-06	2.0	1.78e-15	**	3.1	1.08
12800	5.16e-07	2.0	6.61e-07	2.0	3.0	0.95	1.12e-06	2.0	1.78e-15	**	3.0	2.12
25600	1.29e-07	2.0	1.65e-07	2.0	3.0	1.95	2.79e-07	2.0	1.78e-15	**	3.0	4.18
51200	3.22e-08	2.0	4.13e-08	2.0	3.0	3.87	6.97e-08	2.0	1.78e-15	**	3.0	8.41
102400	8.06e-09	2.0	1.03e-08	2.0	2.6	7.45	1.74e-08	2.0	1.78e-15	**	2.9	16.54
204800	2.02e-09	2.0	2.58e-09	2.0	2.3	14.32	4.36e-09	2.0	2.66e-15	**	2.4	30.83
409600	5.04e-10	2.0	6.45e-10	2.0	2.1	27.64	1.09e-09	2.0	2.66e-15	**	2.2	59.74
819200	1.26e-10	2.0	1.61e-10	2.0	2.0	55.22	2.72e-10	2.0	2.66e-15	**	2.0	114.18
	Gauss-2						PHBVM(4, 2)					
50	3.43e-04	—	1.83e-04	—	8.9	0.01	4.89e-05	—	7.97e-09	—	9.1	0.02
100	2.16e-05	4.0	1.15e-05	4.0	7.8	0.02	3.05e-06	4.0	3.19e-11	8.0	7.9	0.03
200	1.35e-06	4.0	7.21e-07	4.0	6.8	0.03	1.90e-07	4.0	8.88e-16	**	6.9	0.05
400	8.44e-08	4.0	4.51e-08	4.0	6.0	0.06	1.19e-08	4.0	8.88e-16	**	6.1	0.10
800	5.28e-09	4.0	2.82e-09	4.0	5.5	0.10	7.44e-10	4.0	1.78e-15	**	5.6	0.18
1600	3.30e-10	4.0	1.76e-10	4.0	5.1	0.19	4.65e-11	4.0	8.88e-16	**	5.2	0.35
3200	2.06e-11	4.0	1.10e-11	4.0	4.7	0.36	2.97e-12	4.0	1.78e-15	**	4.7	0.65
6400	1.28e-12	4.0	6.84e-13	4.0	4.3	0.68	2.24e-13	3.7	1.78e-15	**	4.3	1.23
	Gauss-3						PHBVM(6, 3)					
50	5.49e-07	—	2.88e-07	—	9.7	0.01	1.23e-07	—	8.88e-16	—	9.8	0.02
100	8.58e-09	6.0	4.49e-09	6.0	8.1	0.02	1.92e-09	6.0	8.88e-16	**	8.2	0.03
200	1.34e-10	6.0	7.00e-11	6.0	7.0	0.03	3.00e-11	6.0	8.88e-16	**	7.1	0.06
400	2.12e-12	6.0	1.10e-12	6.0	6.3	0.06	5.08e-13	5.9	1.78e-15	**	6.4	0.12
800	5.30e-14	5.3	2.04e-14	5.7	5.7	0.11	4.80e-14	3.4	1.78e-15	**	5.7	0.22

- For a fixed time-step h, the numerical solutions provided by the Gauss methods and by the corresponding PHBVM method have a comparable accuracy, despite the negligible Hamiltonian error of these latter methods;
- The execution times of the PHBVM methods are about double than those of the corresponding Gauss methods, even though the mean number of blended
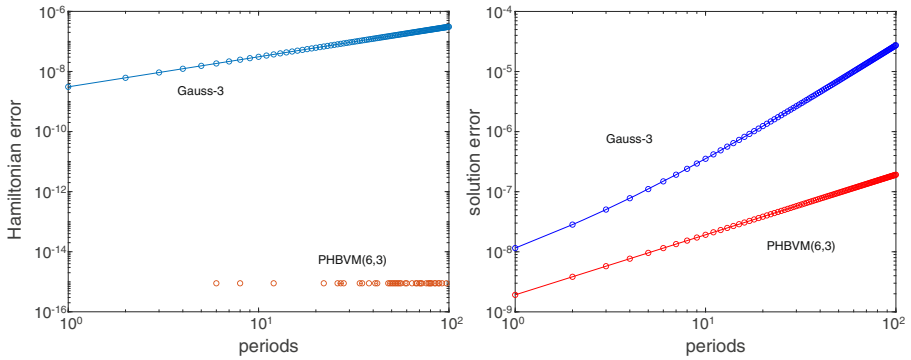
Fig. 2 Hamiltonian error (left plot) and solution error (right plot) when solving problem (62) with time-step $h = T/100$ over 100 periods

iterations per time-step is practically the same (this latter, decreasing with the time-step h, and slightly increasing with s).

As a result, one would conclude that the conservation of the Hamiltonian apparently gains no practical advantage. However, this conclusion is readily confuted if we look at the error growth in the Hamiltonian and in the solution. In fact, in Fig. 2 there is the plot of the Hamiltonian error (left plot) and the solution error (right plot) by using the 3-stage Gauss method and the PHBVM(6,3) method with time-step $h = T/100$ over 100 periods. As one may see, now it is clear that the 3-stage Gauss method exhibits a numerical drift in the energy, unlike PHBVM(6,3). As a result, this latter method exhibits a linear error growth, whereas the former one has a quadratic error growth.


Fig. 3 Solution of problem (63)

Table 2 Results for problem (63)

Gauss-1

n	e_y	Rate	e_H	Rate	e_C	Rate	it	Time
50	6.25e-02	—	4.89e-01	—	3.86e-02	—	8.4	0.03
100	1.62e-02	1.9	1.26e-01	2.0	9.80e-03	2.0	6.5	0.01
200	4.10e-03	2.0	3.18e-02	2.0	2.44e-03	2.0	5.6	0.03
400	1.03e-03	2.0	7.97e-03	2.0	6.09e-04	2.0	4.7	0.04
800	2.57e-04	2.0	1.99e-03	2.0	1.52e-04	2.0	4.3	0.08
1600	6.42e-05	2.0	4.99e-04	2.0	3.81e-05	2.0	3.8	0.14
3200	1.60e-05	2.0	1.25e-04	2.0	9.51e-06	2.0	3.5	0.25
6400	4.01e-06	2.0	3.12e-05	2.0	2.38e-06	2.0	3.1	0.49
12800	1.00e-06	2.0	7.79e-06	2.0	5.95e-07	2.0	3.0	0.97
25600	2.51e-07	2.0	1.95e-06	2.0	1.49e-07	2.0	3.0	1.93
51200	6.27e-08	2.0	4.87e-07	2.0	3.72e-08	2.0	3.0	3.87
102400	1.57e-08	2.0	1.22e-07	2.0	9.29e-09	2.0	2.9	7.67
204800	3.92e-09	2.0	3.04e-08	2.0	2.32e-09	2.0	2.5	14.70
409600	9.79e-10	2.0	7.61e-09	2.0	5.81e-10	2.0	2.3	28.79
819200	2.45e-10	2.0	1.90e-09	2.0	1.45e-10	2.0	2.1	55.80

Gauss-2

PHBVM(4, 1)

n	e_y	Rate	e_H	Rate	e_C	Rate	it	Time
50	1.23e-01	—	1.01e-05	—	5.45e-02	—	9.8	0.05
100	3.00e-02	2.0	3.80e-08	8.0	1.32e-02	2.0	7.2	0.03
200	7.46e-03	2.0	5.55e-15	**	3.26e-03	2.0	5.9	0.06
400	1.86e-03	2.0	5.11e-15	**	8.13e-04	2.0	5.0	0.10
800	4.65e-04	2.0	2.00e-15	**	2.03e-04	2.0	4.4	0.18
1600	1.16e-04	2.0	2.44e-15	**	5.07e-05	2.0	4.1	0.33
3200	2.91e-05	2.0	3.77e-15	**	1.27e-05	2.0	3.6	0.59
6400	7.27e-06	2.0	3.33e-15	**	3.17e-06	2.0	3.4	1.14
12800	1.82e-06	2.0	3.77e-15	**	7.93e-07	2.0	3.1	2.19
25600	4.54e-07	2.0	3.77e-15	**	1.98e-07	2.0	3.0	4.32
51200	1.14e-07	2.0	3.77e-15	**	4.95e-08	2.0	3.0	8.61
102400	2.84e-08	2.0	4.22e-15	**	1.24e-08	2.0	3.0	17.18
204800	7.10e-09	2.0	5.11e-15	**	3.10e-09	2.0	2.7	33.34
409600	1.77e-09	2.0	5.55e-15	**	7.74e-10	2.0	2.5	63.85
819200	4.44e-10	2.0	5.55e-15	**	1.93e-10	2.0	2.2	122.89

PHBVM(4, 2)

Table 2 (continued)

n	e_y	Rate	e_H	Rate	e_C	Rate	it	Time	e_y	Rate	e_H	Rate	e_C	Rate	it	Time
50	2.56e-04	—	1.79e-03	—	8.08e-04	—	10.4	0.02	2.18e-04	—	3.49e-07	—	9.72e-04	—	10.4	0.03
100	1.58e-05	4.0	1.11e-04	4.0	5.37e-05	3.9	8.5	0.02	1.30e-05	4.1	1.52e-09	7.8	6.22e-05	4.0	8.7	0.04
200	9.88e-07	4.0	6.97e-06	4.0	3.32e-06	4.0	7.4	0.03	8.05e-07	4.0	3.77e-15	**	3.86e-06	4.0	7.5	0.06
400	6.17e-08	4.0	4.35e-07	4.0	2.09e-07	4.0	6.5	0.06	5.02e-08	4.0	2.00e-15	**	2.42e-07	4.0	6.5	0.11
800	3.86e-09	4.0	2.72e-08	4.0	1.31e-08	4.0	6.0	0.11	3.13e-09	4.0	2.00e-15	**	1.51e-08	4.0	6.2	0.21
1600	2.41e-10	4.0	1.70e-09	4.0	8.16e-10	4.0	5.4	0.20	1.96e-10	4.0	3.77e-15	**	9.46e-10	4.0	5.4	0.37
3200	1.51e-11	4.0	1.06e-10	4.0	5.10e-11	4.0	5.0	0.38	1.23e-11	4.0	3.77e-15	**	5.91e-11	4.0	5.1	0.71
6400	9.98e-13	3.9	6.65e-12	4.0	3.19e-12	4.0	4.5	0.71	8.54e-13	3.8	3.77e-15	**	3.70e-12	4.0	4.5	1.35

Gauss-3   PHBVM(6, 3)

n	e_y	rate	e_H	rate	e_C	rate	it	time	e_y	rate	e_H	rate	e_C	rate	it	time
50	1.13e-06	—	6.47e-06	—	3.57e-06	—	10.9	0.01	5.51e-07	—	5.11e-15	—	1.97e-06	—	11.1	0.03
100	1.76e-08	6.0	1.00e-07	6.0	5.43e-08	6.0	8.9	0.02	9.34e-09	5.9	3.33e-15	**	2.79e-08	6.1	9.1	0.04
200	2.76e-10	6.0	1.60e-09	6.0	8.86e-10	5.9	7.7	0.03	1.49e-10	6.0	2.89e-15	**	4.26e-10	6.0	7.8	0.07
400	4.23e-12	6.0	2.50e-11	6.0	1.38e-11	6.0	6.8	0.06	2.25e-12	6.0	2.89e-15	**	6.62e-12	6.0	6.9	0.12
800	3.08e-14	7.1	4.05e-13	5.9	2.17e-13	6.0	6.3	0.11	6.30e-14	5.2	3.33e-15	**	1.03e-13	6.0	6.3	0.23
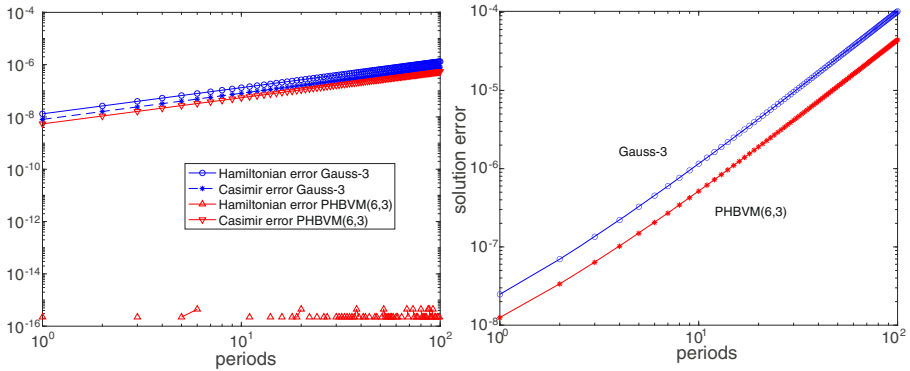
Fig. 4 Hamiltonian and Casimir errors (left plot) and solution error (right plot) when solving problem (63) with time-step $h = T/100$ over 100 periods with the 3-stage Gauss and PHBVM(6,3) methods

Example 2 The second example, taken from [20], is given by:

$$\dot{y} = \begin{pmatrix} 0 & y_1 y_2 & y_1 y_3 \\ -y_1 y_2 & 0 & -y_2 y_3 \\ -y_1 y_3 & y_2 y_3 & 0 \end{pmatrix} \nabla H(y),$$

$$H(y) = a\left(\ln y_1 - \frac{y_1}{y_1^*}\right) + b\left(\ln y_2 - \frac{y_2}{y_2^*}\right) + c\left(\ln y_3 - \frac{y_3}{y_3^*}\right), \qquad (63)$$

$$C(y) = -\ln y_1 - \ln y_2 + \ln y_3,$$

with

$$a = 1, \qquad b = 2, \qquad c = 3, \qquad y_1^* = 1, \qquad y_2^* = 10, \qquad y_3^* = 50, \qquad y(0) = (1, \, 1, \, 1)^\top,$$

whose solution, which is periodic of period $T \approx 2.143610709155912$, is depicted in Fig. 3.

At first, we compare the same methods used for the previous example, again with time-step $h = T/n$. The obtained results for the *s*-stage Gauss and PHBVM methods are listed in Table 2: the conclusions that one can derive from them are similar to those driven from Table 1 for the previous example, with the additional remark that now the Casimir $C(y)$ is not conserved.[9] This fact, in turn, produces the results depicted in the two plots in Fig. 4, concerning the application of the Gauss-3 and PHBVM(6,3) methods for solving the problem with time-step $h = T/100$ over 100 periods. From the two plots, one infers that both methods exhibit a drift in the Casimir, whereas only Gauss-3 exhibits a drift in the Hamiltonian, too (left plot); however, *both* methods exhibit a quadratic error growth in the solution, despite the fact that PHBVM(6,3) conserves the Hamiltonian. For this reason, in Table 3 we list the obtained results by using the EHBVM(4,1), EHBVM(4,2), and EHBVM(6,3)

---

[9]In Table 2 e_C denotes the error in the Casimir.

Table 3 Further results for problem (63)

n	e_y	Rate	e_H	Rate	e_C	Rate	it	Time
	EPHBVM(4, 1)							
50	1.26e-01	—	9.36e-06	—	2.21e-06	—	11.1	0.06
100	3.07e-02	2.0	3.73e-08	8.0	9.48e-09	7.9	8.2	0.04
200	7.45e-03	2.0	5.55e-15	**	1.78e-15	**	6.8	0.07
400	1.90e-03	2.0	3.77e-15	**	1.78e-15	**	5.6	0.12
800	4.70e-04	2.0	3.77e-15	**	8.88e-16	**	4.8	0.22
1600	1.18e-04	2.0	3.77e-15	**	1.78e-15	**	4.4	0.39
3200	2.96e-05	2.0	3.77e-15	**	1.78e-15	**	4.1	0.71
6400	7.44e-06	2.0	5.11e-15	**	1.78e-15	**	3.9	1.39
12800	1.83e-06	2.0	5.11e-15	**	1.78e-15	**	3.5	2.64
25600	4.64e-07	2.0	6.88e-15	**	1.78e-15	**	3.3	5.16
51200	1.18e-07	2.0	6.88e-15	**	1.78e-15	**	3.1	9.98
102400	2.83e-08	2.1	6.88e-15	**	1.78e-15	**	3.0	19.71
204800	7.20e-09	2.0	5.55e-15	**	1.78e-15	**	2.8	38.16
409600	1.79e-09	2.0	6.88e-15	**	1.78e-15	**	2.5	72.98
819200	4.53e-10	2.0	7.33e-15	**	1.78e-15	**	2.2	140.73
	EPHBVM(4, 2)							
50	1.29e-04	—	3.49e-07	—	1.72e-07	—	10.4	0.03
100	6.20e-06	4.4	1.52e-09	7.8	6.01e-10	8.2	8.7	0.04
200	1.34e-07	5.5	5.11e-15	**	8.88e-16	**	7.5	0.07
400	2.54e-08	2.4	3.33e-15	**	1.78e-15	**	6.5	0.12
800	6.73e-10	5.2	3.33e-15	**	8.88e-16	**	6.2	0.24
1600	9.47e-11	2.8	3.33e-15	**	1.78e-15	**	5.4	0.42
3200	6.25e-12	3.9	4.22e-15	**	1.78e-15	**	5.1	0.83
6400	5.46e-13	3.5	3.77e-15	**	1.78e-15	**	4.5	1.54
	EPHBVM(6, 3)							
50	5.32e-07	–	3.33e-15	—	8.88e-16	—	11.1	0.03
100	7.44e-09	6.2	3.77e-15	**	1.78e-15	**	9.1	0.05
200	5.78e-11	7.0	3.33e-15	**	1.78e-15	**	7.8	0.08
400	1.81e-12	5.0	2.00e-15	**	8.88e-16	**	6.9	0.14
800	8.24e-14	4.5	4.22e-15	**	1.33e-15	**	6.3	0.27

methods for solving problem (63), by using the same time-steps considered for obtaining the results of Table 2. As one may see, now the conservation of the Casimir is soon obtained, as the time-step is decreased, besides that of the Hamiltonian, with a computational cost perfectly comparable to that of the corresponding PHBVM method. The conservation of both invariants, in turn, allows to recover a linear error growth in the numerical solution, as is shown in the plot of Fig. 5.
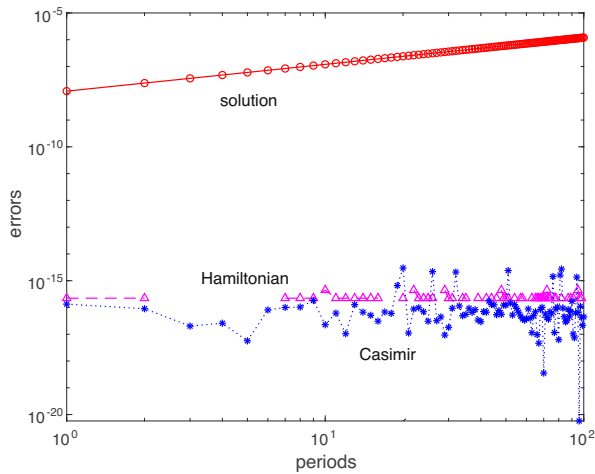
Fig. 5 Hamiltonian, Casimir, and solution errors when solving problem (63) with time-step $h = T/100$ over 100 periods with the EPHBVM(6,3) method

6 Conclusions

In this paper we have presented a class of energy-conserving line integral methods for Poisson problems. In the case where the problem is Hamiltonian, these methods reduce to the class of Hamiltonian Boundary Value Methods (HBVMs), which are energy-conserving methods for such problems. Consequently, the new methods can be regarded as an extension of HBVMs for Poisson (not Hamiltonian) problems, which we called PHBVMs. Moreover, a further enhancement of such methods (EPHBVMs) allows to obtain the conservation of Casimirs, too. A thorough analysis of the methods has been carried out, confirmed by a couple of numerical tests. As a further direction of investigation, we mention the study of the application of the methods for solving highly oscillatory Poisson problems, similarly as done with HBVMs in the Hamiltonian case [27–29].

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Brugnano, L., Calvo, M., Montijano, J.I., Rández, L.: Energy preserving methods for Poisson systems. J. Comput. Appl. Math. **236**, 3890–3904 (2012)
2. Brugnano, L., Frasca Caccia, G., Iavernaro, F.: Efficient implementation of Gauss collocation and Hamiltonian Boundary Value Methods. Numer. Algorithms **65**, 633–650 (2014)
3. Brugnano, L., Gurioli, G., Iavernaro, F.: Analysis of energy and QUadratic invariant preserving (EQUIP) methods. J. Comput. Appl. Math. **335**, 51–73 (2018)
4. Brugnano, L., Iavernaro, F.: Line integral methods which preserve all invariants of conservative problems. J. Comput. Appl. Math. **236**, 3905–3919 (2012)
5. Brugnano, L., Iavernaro, F.: Line Integral Methods for Conservative Problems. Chapman and Hall/CRC, Boca Raton (2016)
6. Brugnano, L., Iavernaro, F.: Line integral solution of differential problems. Axioms **7**(2), article n. 36 (2018). https://doi.org/10.3390/axioms7020036
7. Brugnano, L., Iavernaro, F., Trigiante, D.: Hamiltonian BVMs (HBVMs): A family of "drift-free" methods for integrating polynomial Hamiltonian systems. AIP Conf. Proc. **1168**, 715–718 (2009)
8. Brugnano, L., Iavernaro, F., Trigiante, D.: Hamiltonian boundary value methods (energy preserving discrete line integral methods). JNAIAM J. Numer. Anal. Ind. Appl. Math. **5**(1–2), 17–37 (2010)
9. Brugnano, L., Iavernaro, F., Trigiante, D.: A note on the efficient implementation of Hamiltonian BVMs. J. Comput. Appl. Math. **236**, 375–383 (2011)
10. Brugnano, L., Iavernaro, F., Trigiante, D.: The lack of continuity and the role of infinite and infinitesimal in numerical methods for ODEs: the case of symplecticity. Appl. Math. Comput. **218**, 8056–8063 (2012)
11. Brugnano, L., Iavernaro, F., Trigiante, D.: A simple framework for the derivation and analysis of effective one-step methods for ODEs. Appl. Math. Comput. **218**, 8475–8485 (2012)
12. Brugnano, L., Iavernaro, F., Trigiante, D.: Analisys of Hamiltonian Boundary Value Methods (HBVMs): a class of energy-preserving Runge-Kutta methods for the numerical solution of polynomial Hamiltonian systems. Commun. Nonlinear Sci. Numer. Simul. **20**, 650–667 (2015)
13. Brugnano, L., Magherini, C.: Blended implementation of block implicit methods for ODEs. Appl. Numer. Math. **42**, 29–45 (2002)
14. Brugnano, L., Magherini, C.: The BiM code for the numerical solution of ODEs. J. Comput. Appl. Math. **164-165**, 145–158 (2002)
15. Brugnano, L., Magherini, C.: Blended implicit methods for the numerical solution of DAE problems. J. Comput. Appl. Math. **189**, 34–50 (2006)
16. Brugnano, L., Magherini, C.: Recent advances in linear analysis of convergence for splittings for solving ODE problems. Appl. Numer. Math. **59**, 542–557 (2009)
17. Brugnano, L., Montijano, J.I., Rández, L.: High-order energy-conserving line integral methods for charged particle dynamics. J. Comput. Phys. **396**, 209–227 (2019)
18. Brugnano, L., Sun, Y.: Multiple invariants conserving Runge-Kutta type methods for Hamiltonian problems. Numer. Algorithms **65**, 611–632 (2014)
19. Cohen, D., Hairer, E.: Linear energy-preserving integrators for Poisson systems. BIT Numer. Math. **51**, 91–101 (2011)
20. Hairer, E., Lubich, C., Wanner, G. Geometric Numerical Integration, 2nd edn. Springer, Berlin (2006)
21. Miyatake, Y.: A derivation of energy-preserving exponentially-fitted integrators for Poisson systems. Comput. Phys. Commun. **187**, 156–161 (2015)
22. Quispel, G.R.W., McLaren, D.I.: A new class of energy-preserving numerical integration methods. J. Phys. A **41**(4), 045206 (2008)
23. Wang, B., Meng, F., Fang, Y.: Efficient implementation of RKN-type Fourier collocation methods for second-order differential equations. Appl. Numer. Math. **119**, 164–178 (2017)
24. Wang, B., Wu, X.: Functionally-fitted energy-preserving integrators for Poisson systems. J. Comput. Phys. **364**, 137–152 (2018)

25. Wang, B., Wu, X.: Geometric Integrators for Differential Equations with Highly Oscillatory Solutions. Springer Nature Singapore Pte Ltd (2021)
26. Mei, L., Huang, L., Wu, X.: A unified framework for the study of high-order energy-preserving integrators for solving Poisson systems. J. Comput. Phys. https://doi.org/10.1016/j.jcp.2021.110822
27. Brugnano, L., Montijano, J.I., Rández, L.: On the effectiveness of spectral methods for the numerical solution of multi-frequency highly-oscillatory Hamiltonian problems. Numer. Algorithms **81**, 345–376 (2019)
28. Brugnano, L., Iavernaro, F., Montijano, J.I., Rández, L.: Spectrally accurate space-time solution of Hamiltonian PDEs. Numer. Algorithms **81**, 1183–1202 (2019)
29. Amodio, P., Brugnano, L., Iavernaro, F.: Analysis of spectral Hamiltonian boundary value methods (SHBVMs) for the numerical solution of ODE problems. Numer. Algorithms **83**, 1489–1508 (2020)