Nonlinear Studies - www. nonlinearstudies.com
MESA - www.journalmesa.com

Preprint submitted to Nonlinear Studies / MESA

HAMILTONIAN BOUNDARY VALUE METHODS (HBVMs) and their efficient implementation

Luigi Brugnano[1,⋆], Gianluca Frasca Caccia[2], Felice Iavernaro[3]

[1,2] Dipartimento di Matematica e Informatica "U. Dini", University of Firenze, Italy.
[3] Dipartimento di Matematica, University of Bari, Italy.

⋆ *Luigi Brugnano.* luigi.brugnano@unifi.it.

Abstract. One of the main features when dealing with Hamiltonian problems is the conservation of the energy. In this paper we review, at an elemental level, the main facts concerning the family of low-rank Runge-Kutta methods named Hamiltonian Boundary Value Methods (HBVMs) for the efficient numerical integration of these problems. Using these methods one can obtain, an at least "practical", conservation of the Hamiltonian. We also discuss the efficient implementation of HBVMs by means of two different procedures: the *blended* implementation of the methods and an iterative procedure based on a particular triangular splitting of the corresponding Butcher's matrix. We analyze the computational cost of these two procedures that result to be an excellent alternative to a classical fixed-point iteration when the problem at hand is a stiff one. A few numerical tests confirm all the theoretical findings.

1 Introduction

The numerical solution of conservative problems is an active field of investigation dealing with the geometrical properties of the discrete vector field induced by numerical methods. The final goal is to reproduce, in the discrete setting, a number of geometrical properties shared by the original continuous problem. Because of this reason, it has become customary to refer to this field of investigation as *geometric integration*. Actually, this concept can be led back to the early work of G. Dahlquist on differential equations, aimed at reproducing the asymptotic stability of equilibria for the numerical trajectories, according to the well-known linear stability analysis of the methods (see, e.g., [32]).

[2010] **Mathematics Subject Classification:** *65P10, 65L05.*
Keywords: energy-conserving methods; Hamiltonian Boundary Value Methods; Implicit Runge-Kutta methods; Blended methods; Splitting Methods; Hamiltonian problems

In particular, we shall deal with the numerical solution of *Hamiltonian problems*, which are encountered in many real-life applications, ranging from the nano-scale of molecular dynamics to the macro-scale of celestial mechanics. Such problems have the following general form,

$$y' = J\nabla H(y), \qquad y(0) = y_0 \in \mathbb{R}^{2m}, \tag{1.1}$$

where $J^T = -J = J^{-1}$ is a constant, orthogonal and skew-symmetric matrix, usually given by

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \tag{1.2}$$

(here I is the identity matrix of dimension m). In such a case, we speak about a problem in *canonical form*. The scalar function $H(y)$ is the *Hamiltonian* of the problem and its value is constant during the motion, namely

$$H(y(t)) \equiv H(y_0), \qquad \forall t \geq 0,$$

for the solution of (1.1). Indeed, one has:

$$\frac{\mathrm{d}}{\mathrm{d}t} H(y(t)) = \nabla H(y(t))^T y'(t) = \nabla H(y(t))^T J \nabla H(y(t)) = 0, \qquad \forall t \geq 0, \tag{1.3}$$

due to the fact that $J^T = -J$. Often, the Hamiltonian H is also called the *energy*, since for isolated mechanical systems it has the physical meaning of total energy. Consequently, *energy conservation* is an important feature in the simulation of such problems. The state vector of a Hamiltonian system splits in two m-length components

$$y = \begin{pmatrix} q \\ p \end{pmatrix},$$

where q and p are the vectors of generalized positions and momenta, respectively. Consequently, (1.1)-(1.2) becomes

$$q' = \nabla_p H(q,p), \qquad p' = -\nabla_q H(q,p).$$

Depending on the case, we shall use either one or the other notation.

Another important feature of Hamiltonian dynamical systems is that they possess a *symplectic* structure. To introduce this property we need a couple of ingredients:

- The *flow of the system*: it is the map acting on the phase space $\mathbb{R}^{2m}$ as

  $$\phi_t : y_0 \in \mathbb{R}^{2m} \to y(t) \in \mathbb{R}^{2m},$$

  where $y(t)$ is the solution at time t of (1.1) originating from the initial condition y_0. Differentiating both sides of (1.1) by y_0 and observing that

  $$\frac{\partial y(t)}{\partial y_0} = \frac{\partial \phi_t(y_0)}{\partial y_0} \equiv \phi_t'(y_0),$$

  we see that the Jacobian matrix of the flow ϕ_t is the solution of the variational equation associated with (1.1), namely

  $$\frac{d}{dt} A(t) = J\nabla^2 H(y(t)) A(t), \qquad A(0) = I, \tag{1.4}$$

  where $\nabla^2 H(y)$ is the Hessian matrix of $H(y)$.

- The definition of a *symplectic transformation*: a map $u = (q,p) \in \mathbb{R}^{2m} \mapsto u(q,p)\mathbb{R}^{2m}$ is said *symplectic* if its Jacobian matrix $u'(q,p) \in \mathbb{R}^{2m\times 2m}$ is a symplectic matrix, that is

$$u'(q,p)^T J u'(q,p) = J, \quad \text{for all } \ q,p \in \mathbb{R}^m.$$

That said, it is not difficult to prove that, under regularity assumptions on $H(q,p)$, the flow associated to a Hamiltonian system is symplectic. Indeed, setting

$$A(t) = \frac{\partial \phi_t}{\partial y_0},$$

and considering (1.4), on has that

$$\begin{aligned}\frac{d}{dt}\left(A(t)^T J A(t)\right) &= \left(\frac{d}{dt}A(t)\right)^T J A(t) + A(t)^T J \left(\frac{d}{dt}A(t)\right)\\ &= A(t)^T \nabla^2 H(y(t)) \underbrace{J^T J}_{=I} A(t) \ + \ A(t)^T \underbrace{JJ}_{=-I} \nabla^2 H(y(t)) A(t) = 0.\end{aligned}$$

Therefore

$$A(t)^T J A(t) \equiv A(0)^T J A(0) = J.$$

The converse of the above property is also true: if the flow associated with a dynamical system $\dot y = f(y)$ defined on $\mathbb{R}^{2m}$ is symplectic, then necessarily $f(y) = J\nabla H(y)$ for a suitable scalar function $H(y)$. Consequently, conservation of $H(y)$ follows, by virtue of (1.3).

Symplecticity has relevant implications on the dynamics of Hamiltonian systems. Among the most important are:

(i) *Canonical transformations.* A change of variables $z = \psi(y)$ is *canonical*, namely it preserves the structure of (1.1), if and only if it is symplectic. Canonical transformations were known from Jacobi and used to recast (1.1) in simpler form.

(ii) *Volume preservation.* The flow ϕ_t of a Hamiltonian system is volume preserving in the phase space. Recall that if V is a (suitable) domain of $\mathbb{R}^{2m}$, we have:

$$\mathrm{vol}(V) = \int_V \mathrm{d}y \qquad \Rightarrow \qquad \mathrm{vol}(\phi_t(V)) = \int_{\phi_t(V)} \mathrm{d}y = \int_V \left|\det \frac{\partial \phi_t(y)}{\partial y}\right| \mathrm{d}y.$$

However, since $\frac{\partial \phi_t(y)}{\partial y} \equiv A(t)$ is a symplectic matrix, from $A(t)^T J A(t) = J$ it follows that $\det(A(t))^2 = 1$ for any t and, hence, $\mathrm{vol}(\phi_t(V)) = \mathrm{vol}(V)$. More in general, volume preservation is a characteristic feature of divergence-free vector fields. Recall that the divergence of a vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is the trace of its Jacobian matrix:

$$\mathrm{div} f(y) = \frac{\partial f_1}{\partial y_1} + \frac{\partial f_2}{\partial y_2} + \cdots + \frac{\partial f_n}{\partial y_n},$$

so that f is divergence-free if

$$\mathrm{div} f(y) = 0, \qquad \forall y.$$

The vector field $J\nabla H$ associated with a Hamiltonian system has zero divergence. In fact, considering that $J\nabla H = [\frac{\partial H}{\partial p_1}, \ldots, \frac{\partial H}{\partial p_m}, -\frac{\partial H}{\partial q_1}, \ldots, -\frac{\partial H}{\partial q_m}]^T$ we obtain

$$\operatorname{div} J\nabla H = \frac{\partial^2 H}{\partial q_1 \partial p_1} + \cdots + \frac{\partial^2 H}{\partial q_m \partial p_m} - \frac{\partial^2 H}{\partial p_1 \partial q_1} - \cdots - \frac{\partial^2 H}{\partial p_m \partial q_m} = 0,$$

since the partial derivatives commute. An important consequence of the previous property is Liouville's theorem, which states that the flow ϕ_t associated with a divergence-free vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is volume preserving.

The above properties, and the fact that symplecticity is a characterizing property of Hamiltonian systems, somehow reinforces the search of symplectic methods for their numerical integration. A one-step method

$$y_1 = \Phi_h(y_0)$$

is per se a transformation of the phase space. Therefore the method is symplectic if Φ_h is a symplectic map, i.e., if

$$\frac{\partial \Phi_h(y_0)}{\partial y_0}^T J \frac{\partial \Phi_h(y_0)}{\partial y_0} = J.$$

An important consequence of symplecticity in Runge-Kutta methods is the conservation of all *quadratic first integrals* of a Hamiltonian system.

A *first integral* for system (1.1) is a scalar function $I(y)$ which remains constant if evaluated along any solution $y(t)$ of (1.1): $I(y(t)) = I(y_0)$ or, equivalently,

$$\nabla I(y)^T J \nabla H(y) = 0, \qquad \forall y.$$

A quadratic first integral takes the form $I(y) = y^T C y$, with C a symmetric matrix.

As previously seen, the most noticeable first integral of a Hamiltonian system is the Hamiltonian function itself. It is worth noticing that, while in the continuous setting energy conservation derives from the property of *symplecticity* of the flow (see, e.g., [44]) as sketched above, the same is no longer true in the discrete setting: a symplectic integrator is not able to yield energy conservation in general. Consequently, devising *energy conserving* methods is an important branch of geometric integration.

Symplectic methods can be found in early work of Gröbner (see, e.g., [46]). Symplectic Runge-Kutta methods have been then studied by Feng Kang [42], Sanz Serna [62], and Suris [66]. Such methods are obtained by imposing that the discrete map, associated with a given numerical method, is symplectic, as is the continuous one. In particular, in [62] an easy criterion for symplecticity is provided, for an s-stage Runge-Kutta method with tableau given by

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array} \tag{1.5}$$

where, as usual, $\mathbf{c} = (c_i) \in \mathbb{R}^s$ is the vector of the abscissae, $\mathbf{b} = (b_i) \in \mathbb{R}^s$ is the vector of the weights, and $A = (a_{ij}) \in \mathbb{R}^{s\times s}$ is the corresponding coefficient matrix.

Theorem 1 ([62])**.** The Runge-Kutta method (1.5) is symplectic if and only if

$$BA + A^T B = \mathbf{b}\mathbf{b}^T, \qquad \text{where} \qquad B = \operatorname{diag}(\mathbf{b}). \tag{1.6}$$

Moreover, in [62] the existence of infinitely many symplectic Runge-Kutta methods is proved, by observing that all Gauss-Legendre Runge-Kutta collocation methods satisfy (1.6).

Since for the continuous map symplecticity implies energy-conservation, though this is no more true for the discrete map, one could still expect that at least *an approximate conservation* holds for the discrete map. As a matter of fact, under suitable assumptions, it can be

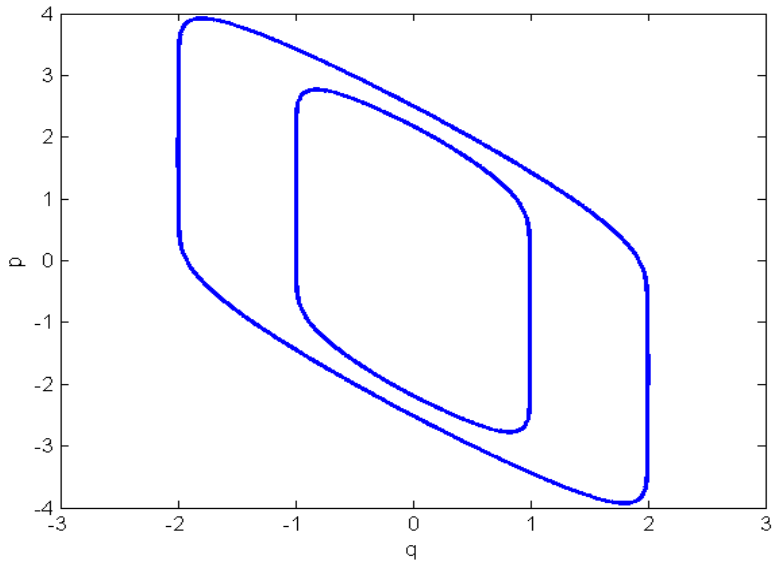

Fig. 1 Level curves for problem (1.7)–(1.9).

proved that, when a symplectic method is used with a constant stepsize, the numerical solution satisfies a perturbed Hamiltonian problem, thus providing a quasi-conservation property over "exponentially long times" [2] (see also [48]). Even though this is an interesting feature, nonetheless, it constitutes a somewhat weak stability result since, in general, it does not extend to infinite intervals.

Moreover, the perturbed dynamical system could be not "so close" to the original one, meaning that, if the stepsize h is not small enough, then the perturbed Hamiltonian could not correctly approximate the exact one. As an example, consider the problem defined by the Hamiltonian

$$H(q,p) = (p/\beta)^2 + (\beta q)^2 + \alpha(q+p)^{2n}. \tag{1.7}$$

The corresponding dynamical system has exactly one (marginally stable) equilibrium at the origin. Let us select the following parameters

$$\beta = 50, \qquad \alpha = 1, \qquad n = 5, \tag{1.8}$$

and suppose we are interested in approximating the level curves of the Hamiltonian (shown in Figure 1) passing from the points

$$(q_0, p_0) = (i, -i), \qquad i = 1, 2. \tag{1.9}$$

This can be done by integrating the trajectories starting at such initial points, for the corresponding Hamiltonian system but, if we use the *symplectic* 2-stage Gauss method, with stepsize $h = 10^{-4}$, we obtain the phase portrait depicted in Figure 2 which is clearly wrong.[1]

A way to get rid of this problem is to directly look for *energy-conserving* methods, able to provide an exact conservation of the Hamiltonian function along the numerical trajectory.

[1] Additional examples may be found in reference [14].

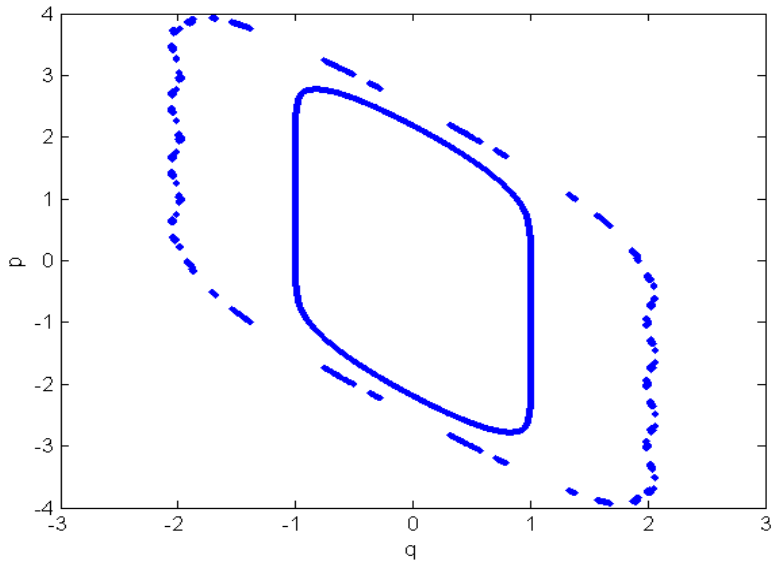

Fig. 2 2-stage Gauss method, $h = 10^{-4}$, for approximating problem (1.7)–(1.9).

The very first attempts to face this problem were based on projection techniques coupled with standard non-conservative numerical methods. However, it is well-known that this approach suffers from many drawbacks, in that this is usually not enough to correctly reproduce the dynamics (see, e.g., [48, p. 111]).

A different approach is represented by *discrete gradient methods*, which are based upon the definition of a discrete counterpart of the gradient operator, so that energy conservation of the numerical solution is guaranteed at each step and for any choice of the integration stepsize [45, 59].

A further approach is based on the concept of *time finite element methods* [53], where one finds local Galerkin approximations on each subinterval of a given mesh of size h for the equation (1.1). This, in turn, has led to the definition of energy-conserving Runge-Kutta methods [3, 4, 67, 68].

A partially related approach is given by *discrete line integral methods* [54, 55, 56], where the key idea is to exploit the relation between the method itself and the *discrete line integral*, i.e., the discrete counterpart of the line integral in conservative vector fields. This tool yields exact conservation for polynomial Hamiltonians of arbitrarily high-degree, and results in the class of methods later named *Hamiltonian Boundary Value Methods (HBVMs)*, which have been developed in a series of papers [15, 16, 14, 18, 19, 21, 22, 23].

Another approach, strictly related to the latter one, is given by the *Averaged Vector Field* method [61, 36] and its generalizations [47], which have been also analysed in the framework of B-series [37] (i.e., methods admitting a Taylor expansion with respect to the stepsize), see e.g., [50].

Further generalizations of HBVMs can be also found in [24, 25, 6, 9, 33].

2 Geometric Integration

In this section, we will discuss the basic issues about Geometric Integration, giving a concrete motivation to look for *energy-conserving* methods, for the efficient numerical solution of Hamiltonian problems. In particular, we will focus on the fundamental tool which has permitted the definition of Hamiltonian Boundary Value Methods (HBVMs), i.e., the discrete line integral. The material of this section is based on references [55, 56, 15, 16, 10].

2.1 Discrete line integral methods

The basic idea which HBVMs rely on is straightforward. We shall at first sketch it in the simplest case, as was done in [54], and then the argument will be generalized. Assume that, in problem (1.1), the Hamiltonian is a polynomial of degree ν. Starting from the initial condition y_0 we want to produce a new approximation at $t = h$, say y_1, such that the Hamiltonian is conserved. Considering the simplest possible path joining y_0 and y_1, i.e., the segment

$$\sigma(ch) = cy_1 + (1-c)y_0, \qquad c \in [0,1], \tag{2.1}$$

one obtains:

$$\begin{aligned} H(y_1) - H(y_0) \;&=\; H(\sigma(h)) - H(\sigma(0)) \;=\; \int_0^h \nabla H(\sigma(t))^T \sigma'(t) \mathrm{d}t \\ &= h \int_0^1 \nabla H(\sigma(ch))^T \sigma'(ch) \mathrm{d}c \;=\; h \int_0^1 \nabla H(cy_1 + (1-c)y_0)^T (y_1 - y_0) \mathrm{d}c \\ &= h \left[\int_0^1 \nabla H(cy_1 + (1-c)y_0) \mathrm{d}c \right]^T (y_1 - y_0). \end{aligned}$$

To obtain energy conservation, $H(y_1) = H(y_0)$, we set

$$y_1 = y_0 + hJ \int_0^1 \nabla H(cy_1 + (1-c)y_0) \mathrm{d}c. \tag{2.2}$$

In fact, since J is skew symmetric, one obtains:

$$\begin{aligned} & h^{-1} \left[\int_0^1 \nabla H(cy_1 + (1-c)y_0) \mathrm{d}c \right]^T (y_1 - y_0) \\ &= \left[\int_0^1 \nabla H(cy_1 + (1-c)y_0) \mathrm{d}c \right]^T J \left[\int_0^1 \nabla H(cy_1 + (1-c)y_0) \mathrm{d}c \right] \;=\; 0. \end{aligned}$$

If $H \in \Pi_\nu$, then the integrand at the right-hand side in (2.2) has degree $\nu - 1$ and, therefore, can be exactly computed by using, say, a Newton-Cotes formula based at ν equidistant abscissae in $[0,1]$. By setting, hereafter,

$$f(\cdot) = J \nabla H(\cdot), \tag{2.3}$$

one then obtains

$$y_1 = y_0 + h \sum_{i=1}^{\nu} b_i f(c_i y_1 + (1-c_i)y_0) \equiv y_0 + h \sum_{i=1}^{\nu} b_i f(Y_i) \tag{2.4}$$

where

$$c_i = \frac{i-1}{\nu-1}, \qquad Y_i = \sigma(c_i) \equiv c_i y_1 + (1-c_i)y_0, \qquad i = 1, \ldots, \nu, \tag{2.5}$$

and the $\{b_i\}$ are the quadrature weights:

$$b_i = \int_0^1 \prod_{j=1,\, j\neq i}^{\nu} \frac{t-c_j}{c_i-c_j} \mathrm{d}t, \qquad i = 1, \ldots, \nu.$$

Some examples:

- when $\nu = 2$, one obtains the usual *trapezoidal method*,

$$y_1 = y_0 + \frac{h}{2}\left(f(y_0) + f(y_1)\right)$$

- when $\nu = 3$, one obtains the following formula:

$$y_1 = y_0 + \frac{h}{6}\left(f(y_0) + 4f\left(\frac{y_0+y_1}{2}\right) + f(y_1)\right)$$

- when $\nu = 5$, one obtains the formula:

$$y_1 = y_0 + \frac{h}{90}\left(7f(y_0) + 32f\left(\frac{3y_0+y_1}{4}\right) + 12f\left(\frac{y_0+y_1}{2}\right) + 32f\left(\frac{y_0+3y_1}{4}\right) + 7f(y_1)\right).$$

The above formulae were named *s-stage trapezoidal methods* in [54]. They provide exact conservation for polynomial Hamiltonian functions of degree no larger than $2\lceil\frac{\nu}{2}\rceil$, for all $\nu \geq 1$. Their order of accuracy can be easily determined by recasting (2.4)–(2.5) as a ν-stage Runge-Kutta method

$$\begin{array}{c|c} \mathbf{c} & \mathbf{c}\mathbf{b}^T \\ \hline & \mathbf{b}^T \end{array} \qquad \text{with} \quad \mathbf{c} = (c_1, \ldots, c_\nu)^T \quad \text{and} \quad \mathbf{b} = (b_1, \ldots, b_\nu)^T, \tag{2.6}$$

which satisfies some of the usual *simplifying assumptions* (see, e.g., [49, p. 71]) for an s-stage Runge-Kutta method (see (1.5)) with coefficients b_i, c_i, a_{ij}, $i, j = 1, \ldots, s$:

$$\begin{array}{ll} B(p): & \sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}, \qquad q = 1, \ldots, p, \\ C(\eta): & \sum_{j=1}^s a_{ij} c_i^{q-1} = \frac{c_i^q}{q}, \qquad q = 1, \ldots, \eta, \quad i = 1, \ldots, s, \\ D(\zeta): & \sum_{i=1}^s b_i c_i^{q-1} a_{ij} = \frac{b_j}{q}(1-c_j^q), \qquad q = 1, \ldots, \zeta, \quad j = 1, \ldots, s. \end{array}$$

In such a case, in fact, the following result holds true.

Theorem 2 (Butcher, 1964)**.** If a Runge-Kutta method satisfies conditions $B(p)$, $C(\eta)$, and $D(\zeta)$, with

$$p \leq \min\{\eta + \zeta + 1,\ 2(\eta+1)\},$$

then it has order p.

As a matter of fact, $B(2)$ and $C(1)$ turn out to be satisfied, for (2.6), thus resulting in a second order method. In more details:

- the quadrature is exact for polynomials of degree 1, so that $B(2)$ holds true;

- moreover, by setting $\mathbf{e} = (1, \dots, 1)^T$, one has

$$\mathbf{c}\mathbf{b}^T\mathbf{e} = \mathbf{c} \qquad \Leftrightarrow \qquad C(1).$$

Remark 1. It is worth mentioning that, even though (2.6) is formally a ν-stage implicit Runge-Kutta method, nevertheless the actual size of the generated discrete problem consists of only *one* nonlinear equation, in the unknown y_1, as the above examples clearly show. The *mono-implicit* character of these methods comes from the fact that their Butcher matrix (i.e., $\mathbf{c}\mathbf{b}^T$), has rank one.

2.2 Generalizing the approach

The next step is to generalize the above approach, where we assumed that the path $\sigma(ch)$, defined in (2.1), is a linear function. For this purpose, we now consider a polynomial path σ of degree $s \geq 1$. Having fixed a suitable basis $\{P_0, \dots, P_{s-1}\}$ for Π_{s-1}, one can expand the derivative of σ as

$$\sigma'(ch) = \sum_{j=0}^{s-1} P_j(c)\gamma_j, \qquad c \in [0,1], \tag{2.7}$$

for certain set of coefficients $\{\gamma_j\}$ to be determined. By imposing the initial condition

$$\sigma(0) = y_0,$$

one then formally obtains

$$\sigma(ch) = y_0 + h\sum_{j=0}^{s-1} \int_0^c P_j(x)\mathrm{d}x\,\gamma_j, \qquad c \in [0,1], \tag{2.8}$$

with the new approximation given by $y_1 \equiv \sigma(h)$. Energy conservation may be obtained by following a similar computation as before, namely

$$\begin{aligned} H(y_1) - H(y_0) \;&=\; H(\sigma(h)) - H(\sigma(0)) \;=\; \int_0^h \nabla H(\sigma(t))^T\sigma'(t)\mathrm{d}t \\ &= h\int_0^1 \nabla H(\sigma(ch))^T\sigma'(ch)\mathrm{d}c \;=\; h\int_0^1 \nabla H(\sigma(ch))^T \sum_{j=0}^{s-1} P_j(c)\gamma_j\mathrm{d}c \\ &= h\sum_{j=0}^{s-1}\left[\int_0^1 \nabla H(\sigma(ch))P_j(c)\mathrm{d}c\right]^T \gamma_j \;=\; 0, \end{aligned} \tag{2.9}$$

provided that the unknown coefficients γ_j satisfy

$$\gamma_j = \eta_j J\int_0^1 \nabla H(\sigma(ch))P_j(c)\mathrm{d}c, \qquad j = 0, \dots, s-1, \tag{2.10}$$

for a suitable set of nonzero scalars $\eta_0, \dots, \eta_{s-1}$. The new approximation is then given by plugging (2.10) into (2.8):

$$y_1 \equiv \sigma(h) = y_0 + h\sum_{j=0}^{s-1}\eta_j \int_0^1 P_j(x)\mathrm{d}x\int_0^1 P_j(\tau)f(\sigma(\tau h))\mathrm{d}\tau. \tag{2.11}$$

As before, assume $H \in \Pi_\nu$. Then, the integrands in (2.10) and (2.11) have at most degree $(\nu-1)s + s - 1 \equiv \nu s - 1$. Therefore, by fixing a suitable set of k abscissae $0 \le c_1 < \cdots < c_k \le 1$, and corresponding quadrature weights $\{b_1, \ldots, b_k\}$, such that the resulting quadrature formula is exact for polynomials of degree $\nu s - 1$, the integrals in (2.10) and (2.11) may be replaced by the corresponding quadrature formulae, which yields

$$\gamma_j = \eta_j \sum_{i=1}^{k} b_i f(\sigma(c_i h)) P_j(c_i), \qquad j = 0, \ldots, s,$$

and

$$y_1 \equiv \sigma(h) = y_0 + h \sum_{j=0}^{s-1} \eta_j \int_0^1 P_j(x)\mathrm{d}x \sum_{i=1}^{k} b_i P_j(c_i) f(\sigma(c_i h)),$$

respectively. By setting, as before,

$$Y_i = \sigma(c_i h), \qquad i = 1, \ldots, k,$$

one then obtains:

$$Y_i = y_0 + h \sum_{j=1}^{k} \overbrace{\left[b_j \sum_{\ell=0}^{s-1} \eta_\ell P_\ell(c_j) \int_0^{c_i} P_\ell(x)\mathrm{d}x \right]}^{=a_{ij}} f(Y_j) \;\equiv\; y_0 + h \sum_{j=1}^{k} a_{ij} f(Y_i), \qquad (2.12)$$

$$i = 1, \ldots, k,$$

$$y_1 = y_0 + h \sum_{i=1}^{k} \underbrace{\left[b_i \sum_{\ell=0}^{s-1} \eta_\ell P_\ell(c_i) \int_0^{1} P_\ell(x)\mathrm{d}x \right]}_{=\hat{b}_i} f(Y_i) \;\equiv\; y_0 + h \sum_{i=1}^{k} \hat{b}_i f(Y_i). \qquad (2.13)$$

We are then speaking about the following k-stage Runge-Kutta method:

$$\begin{array}{c|c} \mathbf{c} & A \equiv (a_{ij}) \in \mathbb{R}^{k\times k} \\ \hline & \hat{\mathbf{b}}^T \end{array} \qquad \text{with} \quad \mathbf{c} = (c_1, \ldots, c_k)^T, \qquad \hat{\mathbf{b}} = (\hat{b}_1, \ldots, \hat{b}_k)^T, \qquad (2.14)$$

with a_{ij}, $\hat{b}_i$ defined according to (2.12) and (2.13), respectively.

In so doing, energy conservation can always be achieved, provided that the quadrature has a suitable high order. For example, we can place the k abscissae $\{c_i\}$ at the k Gauss-Legendre nodes on $[0, 1]$ thus obtaining the maximum order $2k$ (see Section 3 below). In such a case, energy conservation is guaranteed for polynomial Hamiltonians of degree no larger than

$$\nu \le \frac{2k}{s}.$$

However, it is quite difficult to discuss the order of accuracy and the properties of the k-stage Runge-Kutta method (2.14), when a generic polynomial basis is considered. As matter of fact, different choices of the basis could provide different methods, having different orders. As an example, fourth-order energy-conserving Runge-Kutta methods were derived in [55, 56], by using the Newton polynomial basis, defined at the abscissae $\{c_i\}$. We shall see that things will greatly simplify, by choosing an orthonormal polynomial basis.

Remark 2. It is worth noticing that we can cast the Butcher tableau of the k-stage Runge-Kutta method (2.14) in matrix form by introducing the matrices

$$\mathcal{P}_s = \begin{pmatrix} P_0(c_1) & \dots & P_{s-1}(c_1) \\ \vdots & & \vdots \\ P_0(c_k) & \dots & P_{s-1}(c_k) \end{pmatrix} \in \mathbb{R}^{k\times s},$$

$$\mathcal{I}_s = \begin{pmatrix} \int_0^{c_1} P_0(x)\mathrm{d}x & \dots & \int_0^{c_1} P_{s-1}(x)\mathrm{d}x \\ \vdots & & \vdots \\ \int_0^{c_k} P_0(x)\mathrm{d}x & \dots & \int_0^{c_k} P_{s-1}(x)\mathrm{d}x \end{pmatrix} \in \mathbb{R}^{k\times s},$$

$$\Lambda_s = \begin{pmatrix} \eta_0 & & \\ & \ddots & \\ & & \eta_{s-1} \end{pmatrix} \in \mathbb{R}^{s\times s}, \qquad \Omega = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_k \end{pmatrix} \in \mathbb{R}^{k\times k},$$

and the row vector

$$\mathcal{I}_s^1 = \begin{pmatrix} \int_0^1 P_0(x)\mathrm{d}x & \dots & \int_0^1 P_{s-1}(x)\mathrm{d}x \end{pmatrix}. \tag{2.15}$$

One easily checks that (2.14) becomes

$$\begin{array}{c|c} \mathbf{c} & \mathcal{I}_s\Lambda_s\mathcal{P}_s^T\Omega \\ \hline & \mathcal{I}_s^1\Lambda_s\mathcal{P}_s^T\Omega \end{array} \tag{2.16}$$

which will be further studied later.

3 Background results

In this section, we state a few preliminary results concerning Legendre polynomials and perturbation results for differential equations, for later reference. This section is based on references [15, 19, 23].

3.1 Legendre polynomials

The following polynomials, denoted by P_i, are the *Legendre* polynomials shifted on the interval $[0, 1]$, and scaled in order to be orthonormal:

$$\deg P_i = i, \qquad \int_0^1 P_i(x)P_j(x)\mathrm{d}x = \delta_{ij}, \qquad \forall i, j \geq 0, \tag{3.1}$$

where δ_{ij} is the Kronecker symbol (its value is 1, when $i = j$, and 0, otherwise). As any family of orthogonal polynomials, they satisfy a 3-terms recurrence, which is given, in this specific case, by:

$$\begin{aligned} &P_0(x) \equiv 1, \qquad P_1(x) = \sqrt{3}(2x-1), \\ &P_{i+1}(x) = (2x-1)\frac{2i+1}{i+1}\sqrt{\frac{2i+3}{2i+1}}P_i(x) - \frac{i}{i+1}\sqrt{\frac{2i+3}{2i-1}}P_{i-1}(x), \qquad i \geq 1. \end{aligned}$$

We recall that the roots $\{c_1, \ldots, c_k\}$ of $P_k(x)$ are all distinct and belong to the interval $(0,1)$. Thus they may be identified via the following conditions:

$$P_k(c_i) = 0, \quad i = 1, \ldots, k, \qquad \text{with} \qquad 0 < c_1 < \cdots < c_k < 1. \tag{3.2}$$

It is known that they are symmetrically distributed in the interval [0,1]:

$$c_i = 1 - c_{k-i+1}, \qquad i = 1, \ldots, k. \tag{3.3}$$

They are referred to as the Gauss-Legendre abscissae on $[0,1]$, and generate the Gauss-Legendre quadrature formula of order $2k$, namely an interpolating quadrature formula which is exact for polynomials of degree no larger than $2k-1$. In fact, if $p(x) \in \Pi_{2k-1}$, then it can be written as

$$p(x) = q(x)P_k(x) + r(x), \qquad q(x), r(x) \in \Pi_{k-1}.$$

Consequently, since $P_k(x)$ is orthogonal to polynomials of degree less than k (see (3.1)),

$$\begin{aligned} \int_0^1 p(x)\mathrm{d}x &= \int_0^1 \left[q(x)P_k(x) + r(x)\right]\mathrm{d}x \\ &= \underbrace{\int_0^1 q(x)P_k(x)\mathrm{d}x}_{=0} + \int_0^1 r(x)\mathrm{d}x \;=\; \int_0^1 r(x)\mathrm{d}x. \end{aligned}$$

On the other hand, for the quadrature formula (c_i, b_i), with the quadrature weights given by

$$b_i = \int_0^1 \prod_{\substack{j=1 \\ j\neq i}}^{k} \frac{x - c_j}{c_i - c_j}\mathrm{d}x, \qquad i = 1, \ldots, k,$$

one obtains:

$$\sum_{i=1}^{k} b_i p(c_i) \;=\; \sum_{i=1}^{k} b_i \left[q(c_i)\overbrace{P_k(c_i)}^{=0} + r(c_i)\right] \;=\; \sum_{i=1}^{k} b_i r(c_i) \;=\; \int_0^1 r(x)\mathrm{d}x,$$

due to the fact that any quadrature based at k distinct abscissae is exact for polynomials of degree no larger than $k-1$. As a matter of fact, for such a quadrature formula, and for any function $f \in C^{2k}([0,1])$, one has

$$\int_0^1 f(x)\mathrm{d}x = \sum_{i=1}^{k} b_i f(c_i) + \Delta_k, \qquad \Delta_k = \rho_k f^{(2k)}(\zeta),$$

for a suitable $\zeta \in (0,1)$, and with ρ_k independent of f. More in general, if the quadrature had order $q \leq 2k$, one would obtain

$$\int_0^1 f(x)\mathrm{d}x = \sum_{i=1}^{k} b_i f(c_i) + \Delta_k, \qquad \Delta_k = \rho_k f^{(q)}(\zeta), \tag{3.4}$$

with ζ and ρ_k defined similarly as above, thus showing that the formula is exact for polynomials of degree no larger than $q-1$.

In particular, in the sequel, we shall need to discuss the case where the integrand in (3.4) has the following form,

$$f(\tau) = P_j(\tau)G(\tau h), \qquad \tau \in [0,1], \tag{3.5}$$

with P_j the jth Legendre polynomial. The following result then holds true.

Lemma 1. Let $G \in C^{(q)}$, q being the order of the given quadrature formula (c_i, b_i) over the interval $[0,1]$. Then

$$\int_0^1 P_j(\tau)G(\tau h)\mathrm{d}\tau - \sum_{i=1}^{k} b_i P_j(c_i)G(c_i h) = O(h^{q-j}), \qquad j = 0,\dots,q.$$

Proof. The thesis follows from (3.4), by considering that

$$\begin{aligned}\frac{\mathrm{d}^q}{\mathrm{d}\tau^q}P_j(\tau)G(\tau h) \equiv [P_j(\tau)G(\tau h)]^{(q)} &= \sum_{i=0}^{q}\binom{q}{i}P_j^{(i)}(\tau)G^{(q-i)}(\tau h)h^{q-i}\\ &= \sum_{i=0}^{j}\binom{q}{i}P_j^{(i)}(\tau)G^{(q-i)}(\tau h)h^{q-i} \;=\; O(h^{q-j}),\end{aligned}$$

since $P_j^{(i)}(\tau) \equiv 0$, for $i > j$. □

We also need a further result concerning integrals with integrands in the form (3.5), which is stated below.

Lemma 2. Let $G : [0,h] \to V$, with V a suitable vector space, a function which admits a Taylor expansion at 0. Then

$$\int_0^1 P_j(\tau)G(\tau h)\mathrm{d}\tau = O(h^j), \qquad j \geq 0.$$

Proof. One obtains, by expanding $G(\tau h)$ at $\tau = 0$:

$$\begin{aligned}\int_0^1 P_j(t)G(\tau h)\mathrm{d}\tau = \int_0^1 P_j(t)\sum_{k\geq 0}\frac{G^{(k)}(0)}{k!}(\tau h)^k\mathrm{d}\tau &= \sum_{k\geq 0}\frac{G^{(k)}(0)}{k!}h^k\int_0^1 P_j(\tau)\tau^k\mathrm{d}\tau\\ &= \sum_{k\geq j}\frac{G^{(k)}(0)}{k!}h^k\int_0^1 P_j(\tau)\tau^k\mathrm{d}\tau \;=\; O(h^j),\end{aligned}$$

where the last but one equality follows from the fact that

$$\int_0^1 P_j(\tau)\tau^k\mathrm{d}\tau = 0, \qquad \text{for} \qquad k < j. \quad \square$$

3.2 Matrices defined by the Legendre polynomials

The integrals of the Legendre polynomials are related to the polynomial themselves as follows. For all $c \in [0,1]$:

$$\int_0^c P_0(x)\mathrm{d}x = \xi_1 P_1(c) + \frac{1}{2}P_0(c), \qquad \int_0^c P_i(x)\mathrm{d}x = \xi_{i+1}P_{i+1}(c) - \xi_i P_{i-1}(c), \quad i \geq 1, \tag{3.6}$$

$$\text{with} \qquad \xi_i \;=\; \frac{1}{2\sqrt{4i^2-1}}. \tag{3.7}$$

Remark 3. From the orthogonality conditions (3.1), and taking into account that $P_0(x) \equiv 1$, one obtains:

$$\int_0^1 P_0(x)\mathrm{d}x = 1, \qquad \int_0^1 P_j(x)\mathrm{d}x = 0, \qquad \forall j \geq 1. \tag{3.8}$$

Legendre polynomials possess the following symmetry property:

$$P_j(c) = (-1)^j P_j(1-c), \qquad c \in [0,1], \qquad j \geq 0. \tag{3.9}$$

Consequently, their integrals share a similar symmetry:

$$\int_{\tau_1}^{\tau_2} P_j(x)\mathrm{d}x = (-1)^j \int_{1-\tau_2}^{1-\tau_1} P_j(x)\mathrm{d}x, \qquad \forall\ \tau_1, \tau_2 \in [0,1] \quad \text{and} \quad j \geq 0. \tag{3.10}$$

In the sequel, we shall use the following matrices, defined by means of the Legendre polynomials evaluated at the $k \geq s$ abscissae (3.2):[2]

$$\mathcal{P}_s = \begin{pmatrix} P_0(c_1) & \dots & P_{s-1}(c_1) \\ \vdots & & \vdots \\ P_0(c_k) & \dots & P_{s-1}(c_k) \end{pmatrix} \in \mathbb{R}^{k\times s}, \tag{3.11}$$

and

$$\mathcal{I}_s = \begin{pmatrix} \int_0^{c_1} P_0(x)\mathrm{d}x & \dots & \int_0^{c_1} P_{s-1}(x)\mathrm{d}x \\ \vdots & & \vdots \\ \int_0^{c_k} P_0(x)\mathrm{d}x & \dots & \int_0^{c_k} P_{s-1}(x)\mathrm{d}x \end{pmatrix} \in \mathbb{R}^{k\times s}. \tag{3.12}$$

Because of (3.6), one obtains the following relation:

$$\mathcal{I}_s = \mathcal{P}_{s+1}\hat{X}_s, \qquad \hat{X}_s = \left(\begin{array}{ccccc} \frac{1}{2} & -\xi_1 & & & \\ \xi_1 & 0 & \ddots & & \\ & \ddots & \ddots & -\xi_{s-1} & \\ & & \xi_{s-1} & 0 & \\ \hline & & & & \xi_s \end{array}\right) \equiv \left(\begin{array}{c} X_s \\ \hline 0\ \dots\ 0\ \xi_s \end{array}\right). \tag{3.13}$$

We also set

$$\Omega = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_k \end{pmatrix} \in \mathbb{R}^{k\times k} \tag{3.14}$$

the diagonal matrix with the corresponding Gauss-Legendre weights. The following simple properties then hold true [7].

Lemma 3.

$$\det(X_s) = \begin{cases} \prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i-1}^2\,, & \text{if } s \text{ is even,} \\ \frac{1}{2}\prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i}^2\,, & \text{if } s \text{ is odd.} \end{cases}$$

Proof. The thesis easily follows from the Laplace expansion, by considering that, from (3.13), $\det(X_1) = \frac{1}{2}$ and $\det(X_2) = \xi_1^2$. □

[2] They have been formally introduced, for a generic polynomial basis, at the end of the previous section.

Theorem 3. Matrices (3.11) and (3.12) have full column rank, for all $s = 1, \ldots, k$. Moreover,

$$\mathcal{P}_s^T \Omega \mathcal{P}_{s+1} = \begin{pmatrix} I_s & \mathbf{0} \end{pmatrix}. \tag{3.15}$$

Proof. By considering any set of $s \le k$ rows of $\mathcal{P}_s$, the resulting sub-matrix is the Gram matrix of the s linearly independent polynomials $P_0, \ldots, P_{s-1}$ defined at the corresponding s (distinct) abscissae. It is, therefore, nonsingular and, then, $\mathcal{P}_s$ has full column rank. Moreover, when $s = k$, one has

$$\mathcal{P}_{k+1} = \begin{pmatrix} \mathcal{P}_k & \mathbf{0} \end{pmatrix}, \tag{3.16}$$

since the entries in last column are $P_k(c_i) = 0$, $i = 1, \ldots, k$. As a consequence, because of (3.13), for matrix $\mathcal{I}_s$ one obtains:

- when $s < k$, then both $\mathcal{P}_{s+1}$ and $\hat{X}_s$ have full column rank and so has $\mathcal{I}_s$;
- when $s = k$, then from (3.16) it follows that

$$\mathcal{I}_k = \mathcal{P}_{k+1}\hat{X}_k = \mathcal{P}_k X_k,$$

  and both $\mathcal{P}_k$ and X_k are nonsingular (see Lemma 3).

Concerning (3.15), one has, by considering that the quadrature formula (c_i, b_i) is exact for polynomials of degree no larger than $2k - 1 \ge 2s - 1$, and setting $\mathbf{e}_i \in \mathbb{R}^s$ and $\hat{\mathbf{e}}_j \in \mathbb{R}^{s+1}$ the ith and jth unit vectors:

$$\mathbf{e}_i^T \mathcal{P}_s^T \Omega \mathcal{P}_{s+1} \hat{\mathbf{e}}_j = \sum_{\ell=1}^{k} b_\ell P_{i-1}(c_\ell) P_{j-1}(c_\ell) \;=\; \int_0^1 P_{i-1}(x) P_{j-1}(x) \mathrm{d}x \;=\; \delta_{ij},$$
$$\forall i = 1, \ldots, s, \text{ and } j = 1, \ldots, s+1. \quad \square$$

From the previous theorem, the following result easily follows.

Corollary 1. When $k = s$, then $\mathcal{P}_s^{-1} = \mathcal{P}_s^T \Omega$.

3.3 Additional preliminary results

In order to carry out a complete analysis of the methods, we need a perturbation result concerning the initial value problem for ordinary differential equations

$$y'(t) = f(y(t)), \qquad t \ge t_0, \qquad y(t_0) = y_0, \tag{3.17}$$

whose solution will be denoted by $y(t; t_0, y_0)$, in order to emphasize the dependence on the initial condition, set at (t_0, y_0).

Associated with this problem is the corresponding *fundamental matrix*, $\Phi(t, t_0)$, satisfying the *variational problem* (see also (1.4))

$$\Phi'(t, t_0) = J_f(y(t; t_0, y_0))\Phi(t, t_0), \qquad t \ge t_0, \qquad \Phi(t_0, t_0) = I,$$

where the derivative (i.e., $'$) is with respect to t, and $J_f(y)$ is the Jacobian matrix of $f(y)$. The following result then holds true.

Lemma 4. With reference to the solution $y(t; t_0, y_0)$ of problem (3.17), one has:

$$(i) \quad \frac{\partial}{\partial y_0} y(t; t_0, y_0) = \Phi(t, t_0); \qquad\qquad (ii) \quad \frac{\partial}{\partial t_0} y(t; t_0, y_0) = -\Phi(t, t_0) f(y_0).$$

Proof. Let us consider a perturbation δy_0 of the initial condition, and let $y(t; t_0, y_0 + \delta y_0)$ be the corresponding solution. Consequently,

$$\begin{aligned} y'(t; t_0, y_0 + \delta y_0) &= f(y(t; t_0, y_0 + \delta y_0)) \\ &= \underbrace{f(y(t; t_0, y_0))}_{= y'(t; t_0, y_0)} + J_f(y(t; t_0, y_0)) \left[y(t; t_0, y_0 + \delta y_0) - y(t; t_0, y_0) \right] \\ &\qquad + O\left(\| y(t; t_0, y_0 + \delta y_0) - y(t; t_0, y_0) \|^2 \right). \end{aligned}$$

Therefore, by setting

$$z(t) = y(t; t_0, y_0 + \delta y_0) - y(t; t_0, y_0),$$

one obtains that, at first order (as is the case, when we let $\delta y_0 \to 0$),

$$z'(t) = J_f(y(t; t_0, y_0)) z(t), \qquad z(t_0) = \delta y_0.$$

The solution of this linear problem is easily seen to be

$$z(t) = \Phi(t, t_0) \delta y_0$$

and, consequently,

$$\frac{\partial}{\partial y_0} y(t; t_0, y_0) = \frac{\partial}{\partial (\delta y_0)} z(t) = \Phi(t, t_0),$$

i.e., the part (i) of the thesis.

Concerning the part (ii), let us consider a scalar $\varepsilon \approx 0$ and observe that, by setting, $y(t) = y(t; t_0, y_0)$, then

$$y(t; t_0 + \varepsilon, y_0) \equiv y(t - \varepsilon).$$

Consequently, at first order, the solution of the perturbed problem

$$y'(t) = f(y(t)), \qquad t \geq t_0 + \varepsilon, \qquad y(t_0 + \varepsilon) = y_0,$$

coincides with that of the problem

$$y'(t) = f(y(t)), \qquad t \geq t_0, \qquad y(t_0) = y_0(\varepsilon) \equiv y_0 - \varepsilon f(y_0).$$

Letting $\varepsilon \to 0$, one then obtains:

$$\frac{\partial}{\partial t_0} y(t; t_0, y_0) \;=\; \underbrace{\frac{\partial}{\partial y_0} y(t; t_0, y_0)}_{= \Phi(t, t_0)} \overbrace{\frac{\partial}{\partial \varepsilon} y_0(\varepsilon)}^{= -f(y_0)} \;=\; -\Phi(t, t_0) f(y_0).$$

This concludes the proof. □

4 A framework for HBVMs

In this section, we provide a novel framework for discussing the order and the linear stability properties of HBVMs. This framework is based on a local Fourier expansion of the vector field defining the dynamical system and also provides an alternative tool, independent of the material presented in Section 2, for introducing HBVMs and stating their conservation properties, in the specific case when an orthonormal polynomial basis is considered.

The material in this section is based on [19, 23, 21]. It is worth noticing that an interesting extension of this approach has been recently proposed in [68].

4.1 Local Fourier expansion

The previously introduced Legendre polynomials constitute an *orthonormal basis* for the functions defined on the interval $[0,1]$. Therefore, we can formally expand the second member of (1.1) over the interval $[0,h]$ as follows (we use the notation (2.3)):

$$f(y(ch)) = \sum_{j\geq 0} P_j(c)\gamma_j(y), \qquad c\in[0,1], \tag{4.1}$$

where

$$\gamma_j(y) = \int_0^1 P_j(\tau) f(y(\tau h))\mathrm{d}\tau, \qquad j\geq 0. \tag{4.2}$$

The expansion (4.1)-(4.2) is known as the *Neumann expansion* of an analytic function,[3] and converges uniformly, provided that the function $g(c) = f(y(ch))$ has continuous second derivative:[4] for sake of simplicity, hereafter we shall assume $g(t)$ to be analytic.

In so doing, we are transforming the initial value problem

$$y'(t) = f(y(t)), \qquad t\in[0,h], \qquad y(0) = y_0, \tag{4.3}$$

into the equivalent *integro-differential* problem

$$y'(ch) = \sum_{j\geq 0} P_j(c)\int_0^1 P_j(\tau) f(y(\tau h))\mathrm{d}\tau, \qquad c\in[0,1], \qquad y(0) = y_0. \tag{4.4}$$

In order to obtain a polynomial approximation of degree s to (4.3)-(4.4), we just truncate the infinite series to a finite sum. The resulting initial value problem is (see (4.2))

$$\begin{aligned} \sigma'(ch) &= \sum_{j=0}^{s-1} P_j(c)\int_0^1 P_j(\tau) f(\sigma(\tau h))\mathrm{d}\tau \;\equiv\; \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma), \qquad c\in[0,1], \\ \sigma(0) &= y_0, \end{aligned} \tag{4.5}$$

whose solution evidently defines a polynomial $\sigma\in\Pi_s$. Integrating both sides of (4.5) yields the equivalent formulation

$$\sigma(ch) = y_0 + h\sum_{j=0}^{s-1}\int_0^c P_j(x)\mathrm{d}x\,\gamma_j(\sigma), \qquad c\in[0,1]. \tag{4.6}$$

One easily recognizes that (4.5) defines the very same expansion (2.7)–(2.10) seen before (with all $\eta_j = 1$). Consequently, such a method is energy-conserving, if we are able to exactly compute the integrals providing the coefficients $\gamma_j(\sigma)$ at the right-hand side in (4.5). From (3.8) one obtains that

$$\sigma(h) = y_0 + \int_0^h f(\sigma(\tau))\mathrm{d}\tau.$$

Let us now discuss the order of the approximation $\sigma(h)\approx y(h)$.

[3] E.T. Whittaker, G.N. Watson, *A Course in Modern Analysis, Fourth edition*, Cambridge University Press, 1950, page 322.

[4] E. Isaacson, H.B. Keller, *Analysis of Numerical Methods*, Wiley & Sons, 1966, page 206.

Lemma 5. Let $\gamma_j(\sigma)$ be defined according to (4.2). Then $\gamma_j(\sigma) = O(h^j)$.

Proof. The proof follows immediately from (4.2), by virtue of Lemma 2. □

We are now able to prove the following result.

Theorem 4. $\sigma(h) - y(h) = O(h^{2s+1})$.

Proof. Denoting by $y(t; t_0, y_0)$ the solution of problem (3.17) and considering that $\sigma(0) = y_0$, by virtue of (4.1), (4.5), and Lemmas 2, 4, and 5, one has:

$$
\begin{aligned}
\sigma(h) - y(h) \;&=\; y(h; h, \sigma(h)) - y(h; 0, y_0) \;\equiv\; y(h; h, \sigma(h)) - y(h; 0, \sigma(0)) \\
&= \int_0^h \frac{\mathrm{d}}{\mathrm{d}t} y(h; t, \sigma(t)) \mathrm{d}t \;=\; \int_0^h \left(\frac{\partial}{\partial \theta} y(h; \theta, \sigma(t))\Big|_{\theta=t} + \frac{\partial}{\partial \omega} y(h; t, \omega)\Big|_{\omega=\sigma(t)} \sigma'(t) \right) \mathrm{d}t \\
&= \int_0^h \left[-\Phi(h,t) f(\sigma(t)) + \Phi(h,t)\sigma'(t) \right] \mathrm{d}t \;=\; \int_0^h \Phi(h,t)[-f(\sigma(t)) + \sigma'(t)] \mathrm{d}t \\
&= h \int_0^1 \Phi(h, \tau h)[-f(\sigma(\tau h)) + \sigma'(\tau h)] \mathrm{d}\tau \\
&= -h \int_0^1 \Phi(h, \tau h) \left[\sum_{j \ge 0} P_j(\tau)\gamma_j(\sigma) - \sum_{j=0}^{s-1} P_j(\tau)\gamma_j(\sigma) \right] \mathrm{d}\tau \\
&= -h \int_0^1 \Phi(h, \tau h) \sum_{j \ge s} P_j(\tau)\gamma_j(\sigma) \mathrm{d}\tau \;=\; -h \sum_{j \ge s} \underbrace{\left[\int_0^1 \overbrace{\Phi(h, \tau h)}^{\equiv G(\tau h)} P_j(\tau) \mathrm{d}\tau \right]}_{= O(h^j)} \overbrace{\gamma_j(\sigma)}^{= O(h^j)} \\
&= h \sum_{j \ge s} O(h^{2j}) \;=\; O(h^{2s+1}). \quad \square
\end{aligned}
$$

We observe, however, that (4.5) is *not yet* an operative method, but rather a *formula* since, quoting the famous book by Dahlquist and Bijörk [41], "as is well known, even many relatively simple integrals cannot be expressed in finite terms of elementary functions, and thus must be evaluated by numerical methods". In other words, in order to obtain an actual *numerical method*, we need to approximate the integrals appearing in (4.5) by means of a suitable quadrature formula. By recalling Lemma 1, we approximate the integrals in (4.5) by means of a quadrature (c_i, b_i) over k distinct abscissae. Consequently, in place of σ defined by (4.5) or (4.6), we shall compute the new polynomial $u \in \Pi_s$ such that:

$$
\begin{aligned}
u'(ch) &= \sum_{j=0}^{s-1} P_j(c) \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)), \qquad c \in [0,1], \\
u(0) &= y_0,
\end{aligned}
\tag{4.7}
$$

that is,

$$
u(ch) = y_0 + h \sum_{j=0}^{s-1} \int_0^c P_j(x) \mathrm{d}x \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)), \qquad c \in [0,1], \tag{4.8}
$$

with the new approximation given by:

$$y_1 \equiv u(h) = y_0 + h\sum_{i=1}^{k} b_i f(u(c_i h)). \tag{4.9}$$

If the quadrature formula (c_i, b_i) has order q, then, by virtue of Lemma 1 and taking into account (4.2), one obtains

$$\begin{aligned} \gamma_j(u) \equiv \int_0^1 P_j(\tau) f(u(\tau h)) \mathrm{d}\tau \;&=\; \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)) \;+\; \Delta_j(h), \\ \Delta_j(h) = O(h^{q-j}), &\quad j = 0, \dots, q. \end{aligned} \tag{4.10}$$

Consequently, we can rewrite the first equation in (4.7) in the following equivalent form:

$$u'(ch) = \sum_{j=0}^{s-1} P_j(c) \left[\gamma_j(u) - \Delta_j(h)\right], \qquad c \in [0,1].$$

This allows us to derive the following result.

Theorem 5. $y_1 - y(h) = O(h^{p+1})$, where $p = \min\{q, 2s\}$.

Proof. The proof proceeds on the same line as that of Theorem 4:

$$\begin{aligned} y_1 - y(h) = u(h) - y(h) \;&=\; y(h; h, u(h)) - y(h; 0, u(0)) \\ &= \int_0^h \frac{\mathrm{d}}{\mathrm{d}t} y(h; t, u(t)) \mathrm{d}t \;=\; \int_0^h \left(\frac{\partial}{\partial \theta} y(h; \theta, u(t))\Big|_{\theta = t} + \frac{\partial}{\partial \omega} y(h; t, \omega)\Big|_{\omega = u(t)} u'(t) \right) \mathrm{d}t \\ &= \int_0^h \Phi(h, t)[-f(u(t)) + u'(t)] \mathrm{d}t \;=\; h \int_0^1 \Phi(h, \tau h)[-f(u(\tau h)) + u'(\tau h)] \mathrm{d}\tau \\ &= -h \int_0^1 \Phi(h, \tau h) \left[\sum_{j \ge 0} P_j(\tau) \gamma_j(u) - \sum_{j=0}^{s-1} P_j(\tau) \left(\gamma_j(u) - \Delta_j(h)\right) \right] \mathrm{d}\tau \\ &= -h \int_0^1 \Phi(h, \tau h) \sum_{j=0}^{s-1} P_j(\tau) \Delta_j(u) \mathrm{d}\tau - h \int_0^1 \Phi(h, \tau h) \sum_{j \ge s} P_j(\tau) \gamma_j(u) \mathrm{d}\tau \\ &= -h \sum_{j=0}^{s-1} \underbrace{\left[\int_0^1 \overbrace{\Phi(h, \tau h)}^{\equiv G(\tau h)} P_j(\tau) \mathrm{d}\tau \right]}_{= O(h^j)} \overbrace{\Delta_j(u)}^{= O(h^{q-j})} - h \sum_{j \ge s} \underbrace{\left[\int_0^1 \overbrace{\Phi(h, \tau h)}^{\equiv G(\tau h)} P_j(\tau) \mathrm{d}\tau \right]}_{= O(h^j)} \overbrace{\gamma_j(u)}^{= O(h^j)} \\ &= O(h^{q+1}) + h \sum_{j \ge s} O(h^{2j}) \;=\; O(h^{p+1}), \qquad p = \min\{q, 2s\}. \quad \square \end{aligned}$$

Definition 1. The method (4.7)–(4.9) is named *Hamiltonian Boundary Value Method (HBVM) with k stages and degree s*, in short HBVM(k, s).

On the basis of the result of Theorem 5, it is natural to chose the k abscissae so that the order of the quadrature is maximized. Consequently, we shall fix them at the $k \geq s$ Gauss abscissae on $[0,1]$ defined in (3.2). As was seen in Section 3.1, the corresponding quadrature formula (c_i, b_i) has order $q = 2k$. That is, it is exact for polynomials of degree no larger than $2k-1$. As a consequence, the following result holds true.

Corollary 2. By choosing the k abscissae $\{c_i\}$ as in (3.2), a HBVM(k,s) method has order $2s$, for all $k \geq s$.

4.2 Runge-Kutta form of HBVM(k,s)

Before studying the conservation properties of the methods, let us derive the Runge-Kutta formulation of HBVM(k,s). The basic fact is that, at the right-hand sides of equations (4.8)–(4.9), one only needs to know the value of the polynomial u at the abscissae $\{c_i h\}$. Consequently, by setting

$$Y_i = u(c_i h), \qquad i = 1, \dots, k,$$

one obtains:

$$Y_i = y_0 + h \sum_{j=1}^{k} \overbrace{\left[b_j \sum_{\ell=0}^{s-1} P_\ell(c_j) \int_0^{c_i} P_\ell(x) \mathrm{d}x \right]}^{\equiv a_{ij}} f(Y_j) \;\equiv\; y_0 + h \sum_{j=1}^{k} a_{ij} f(Y_j), \qquad (4.11)$$
$$i = 1, \dots, k,$$

$$y_1 = y_0 + h \sum_{i=1}^{k} b_i f(Y_i). \qquad (4.12)$$

In other words, we have defined the following k-stage Runge-Kutta method:

$$\begin{array}{c|c} \mathbf{c} & A \equiv (a_{ij}) \\ \hline & \mathbf{b}^T \end{array} \qquad (4.13)$$

with (see (4.11)),

$$\mathbf{c} = (c_1, \dots, c_k)^T, \quad \mathbf{b} = (b_1, \dots, b_k)^T, \quad \text{and} \quad A = (a_{ij}) \in \mathbb{R}^{k \times k}.$$

The Butcher tableau (4.13) defines the *Runge-Kutta shape of a HBVM(k,s) method*. We can easily derive a more compact form for the Butcher array A in (4.13).

Theorem 6. $A = \mathcal{I}_s \mathcal{P}_s^T \Omega$, with the matrices $\mathcal{I}_s, \mathcal{P}_s, \Omega$ defined according to (3.11)–(3.14).

Proof. By setting $\mathbf{e}_i, \mathbf{e}_j \in \mathbb{R}^k$ the i-th and j-th unit vectors, one obtains:

$$\mathbf{e}_i^T \mathcal{I}_s \mathcal{P}_s^T \Omega \mathbf{e}_j = \left(\int_0^{c_i} P_0(x)\mathrm{d}x \dots \int_0^{c_i} P_{s-1}(x)\mathrm{d}x \right) \begin{pmatrix} P_0(c_j) \\ \vdots \\ P_{s-1}(c_j) \end{pmatrix} b_j$$
$$= b_j \sum_{\ell=0}^{s-1} P_\ell(c_j) \int_0^{c_i} P_\ell(x)\mathrm{d}x \;\equiv\; a_{ij} \;=\; \mathbf{e}_i^T A \mathbf{e}_j,$$

according to (4.11). □

Consequently, the Butcher tableau (4.13) can be cast as:

$$\begin{array}{c|c} \mathbf{c} & \mathcal{I}_s \mathcal{P}_s^T \Omega \\ \hline & \mathbf{b}^T \end{array} \tag{4.14}$$

or, equivalently, by taking into account (3.13),

$$\begin{array}{c|c} \mathbf{c} & \mathcal{P}_{s+1} \hat{X}_s \mathcal{P}_s^T \Omega \\ \hline & \mathbf{b}^T \end{array}. \tag{4.15}$$

Remark 4. We observe that the Runge-Kutta form (4.14) of a HBVM(k,s) method is simplified, with respect to that sketched in Remark 2 for a discrete line-integral method defined by using a general polynomial basis. In particular, the diagonal matrix Λ_s is now automatically fixed, in order to maximize the order of accuracy of the method. Moreover, the vector of the quadrature weights coincide with that used for approximating the integrals involved in the coefficients of the polynomial u.

4.3 HBVM(s,s)

In the case $k = s$, the matrices $\mathcal{I}_s, \mathcal{P}_s, \Omega \in \mathbb{R}^{s\times s}$. Moreover, the following results follow immediately from Theorem 3 and Corollary 1:

$$\mathcal{I}_s = \mathcal{P}_s X_s, \qquad \mathcal{P}_s^T \Omega = \mathcal{P}_s^{-1}.$$

Consequently, in such a case, we can write the Butcher tableau (4.15) as that of the following s-stage method,

$$\begin{array}{c|c} \mathbf{c} & \mathcal{P}_s X_s \mathcal{P}_s^{-1} \\ \hline & \mathbf{b}^T \end{array} \tag{4.16}$$

resulting in the *W-transformation* defining the s-stage Gauss-Legendre Runge-Kutta collocation method [49, p. 79], which has order $2s$. In this sense, in the case $k \geq s$, HBVM(k,s) can be regarded as *low-rank generalizations* of the s-stage Gauss method. Indeed, the following result holds true.

Theorem 7. For all $k \geq s$ the rank of the matrix $A = \mathcal{P}_{s+1} \hat{X}_s \mathcal{P}_s^T \Omega$ is s. Moreover, the nonzero eigenvalues of A coincide with those of the basic s-stage Gauss method.

Proof. The rank of the matrix $\mathcal{P}_{s+1}$ is s or $s+1$ (when $k > s$), whereas that of matrices $\hat{X}_s, \mathcal{P}_s$ is s, and Ω is nonsingular. Therefore, the rank of A cannot exceed s. Moreover, from Theorem 3, one has

$$\mathcal{P}_s^T \Omega A \mathcal{P}_s \;=\; \mathcal{P}_s^T \Omega \mathcal{P}_{s+1} \hat{X}_s \mathcal{P}_s^T \Omega \mathcal{P}_s \;=\; (I_s \; \mathbf{0}) \hat{X}_s I_s \;=\; X_s \in \mathbb{R}^{s\times s},$$

which is known to be nonsingular. Consequently, $\mathrm{rank}(A) = s$. Moreover,

$$\mathcal{P}_s^T \Omega A = \mathcal{P}_s^T \Omega \mathcal{P}_{s+1} \hat{X}_s \mathcal{P}_s^T \Omega = (I_s \; \mathbf{0}) \hat{X}_s \mathcal{P}_s^T \Omega = X_s \mathcal{P}_s^T \Omega.$$

This means that the columns of $\Omega \mathcal{P}_s$ span an s-dimensional left invariant subspace of A. Therefore, the eigenvalues of X_s will coincide with the nonzero eigenvalues of A. On the other hand, from (4.16) one obtains immediately that the eigenvalues of X_s are the eigenvalues of the Butcher matrix of the s-stage Gauss method. □

This property has been named *isospectrality of HBVMs*, in [22]. It will be used later for the efficient implementation of HBVM(k,s) methods.

4.4 Energy conservation

We now consider the issue of energy conservation for HBVM(k,s) methods. From (4.7)–(4.9) with $f = J\nabla H$, one obtains:

$$\begin{aligned}
H(y_1) - H(y_0) = H(u(h)) - H(u(0)) \; &= \; \int_0^h \nabla H(u(t))^T u'(t)\mathrm{d}t \\
&= h\int_0^1 \nabla H(u(\tau h))^T u'(\tau h)\mathrm{d}\tau \\
&= h\int_0^1 \nabla H(u(\tau h))^T \sum_{j=0}^{s-1} P_j(\tau) \sum_{i=1}^{k} b_i P_j(c_i) J\nabla H(c_i h)\mathrm{d}\tau \\
&= h\sum_{j=0}^{s-1} \left[\underbrace{\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau}_{=O(h^j)} \right]^T J \left[\sum_{i=1}^{k} b_i P_j(c_i) J\nabla H(c_i h) \right] \\
&\equiv E_H
\end{aligned}$$

Now, two possibilities may occur:

- $\displaystyle\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau = \sum_{i=1}^{k} b_i P_j(c_i) J\nabla H(c_i h)$: in such case, $E_H = 0$, so that energy is *exactly conserved*. This is the case of a polynomial Hamiltonian of degree ν no larger than $2k/s$;
- $\displaystyle\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau = \sum_{i=1}^{k} b_i P_j(c_i) J\nabla H(c_i h) + \Delta_j(h)$: in such a case, by taking into account (4.10) with $q = 2k$, one obtains that $E_H = O(h^{2k+1})$, provided that the Hamiltonian is suitably regular, as we have assumed.

We have then proved the following result.

Theorem 8. HBVM(k,s) is energy-conserving for all polynomial Hamiltonian of degree

$$\nu \le \frac{2k}{s}. \tag{4.17}$$

In any other case, $H(y_1) - H(y_0) = O(h^{2k+1})$, even though the method has order s.

Remark 5. We observe that:

- for polynomial Hamiltonians, energy conservation can be *always* obtained, by choosing k large enough, by virtue of (4.17);
- even in the case of non polynomial Hamiltonians, energy conservation can still be *practically* gained by choosing k large enough, so that $|E_H|$, which is $O(h^{2k+1})$, is within roundoff errors.

As a first example, in Figure 1 we plotted the level curves passing through the points defined at (1.9) for the Hamiltonian problem with Hamiltonian (1.7) with parameters (1.8). By using the 2-stage Gauss method (fourth-order), with stepsize $h = 10^{-4}$, the obtained

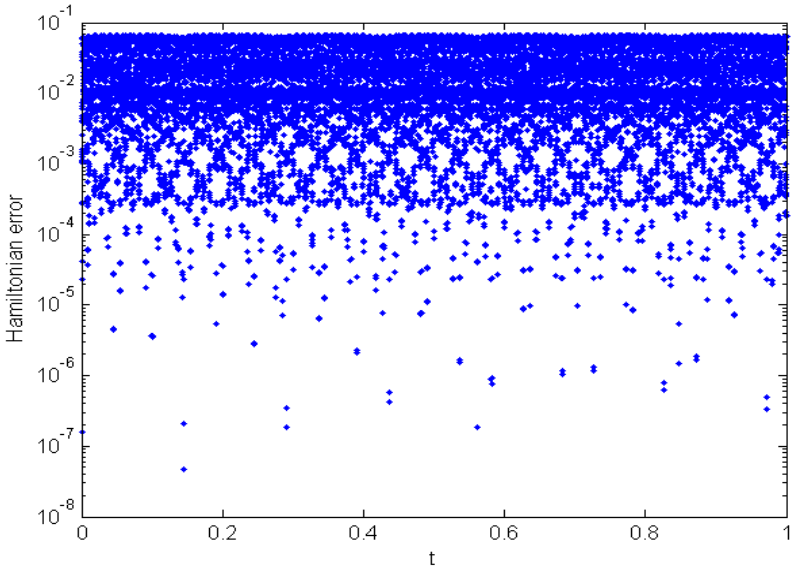

Fig. 3 Hamiltonian error for problem (1.7)–(1.9), 2-stage Gauss method, $h = 10^{-4}$.

phase portrait is wrong, as was shown in Figure 2, due to the error in the numerical Hamiltonian, which is shown in Figure 3. Indeed, even though no drift in the Hamiltonian occurs, nevertheless it is not negligible, for the problem at hand.

However, if we use HBVM$(3, 2)$ with the same stepsize, the error in the Hamiltonian is of sixth-order: this is enough to have a smaller error in the numerical Hamiltonian, as is shown in Figure 4, resulting in a correct phase portrait, as is shown in Figure 5.

At last, by using HBVM$(10, 2)$ with the same stepsize, the Hamiltonian error is of the order of roundoff errors, as is shown in Figure 6, thus allowing a perfect reconstruction of the phase portrait, depicted in Figure 7. Indeed, since the Hamiltonian (1.7) has degree ten, the quadrature is exact, in this case, according to (4.17).

For sake of completeness, in Figure 8 we also plot the mean error in the numerical Hamiltonian, for HBVM$(k, 2)$ methods, used with the stepsize $h = 10^{-4}$, for $k = 2, \ldots, 10$. As one can see, for the largest values of k the error is essentially due to roundoff.

As a second example we consider the problem defined by the following Hamiltonian:

$$H(q, p) = (q^2 + p^2)^2 - 10(q^2 - p^2). \tag{4.18}$$

The level curves for this problem are the Cassini ovals and in Figure 9 we plot the one passing at

$$(q_0, p_0) = \left(0, 10^{-5}\right). \tag{4.19}$$

By using the 2-stage Gauss method with stepsize $h = 10^{-2}$, the obtained phase portrait is "almost" correct at first sight as one can see in Figure 10. This portrait is, actually, qualitatively wrong as one can see in the zoom in Figure 12 and in the comparison between the correct portrait of component q and the approximated one in Figure 14 (similar results are obtained for p), due to the error in the Hamiltonian shown in Figure 16.

By using a HBVM(4,2) method with the same stepsize, since the Hamiltonian (4.18) has degree 4, according to (4.17), the error is of the order of roundoff errors, as is shown in

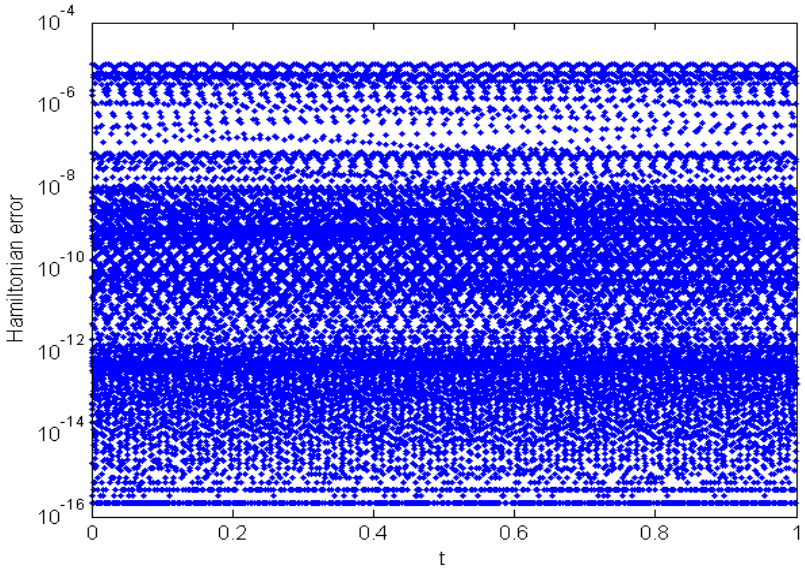

Fig. 4 Hamiltonian error for problem (1.7)–(1.9), HBVM(3,2) method, $h = 10^{-4}$.

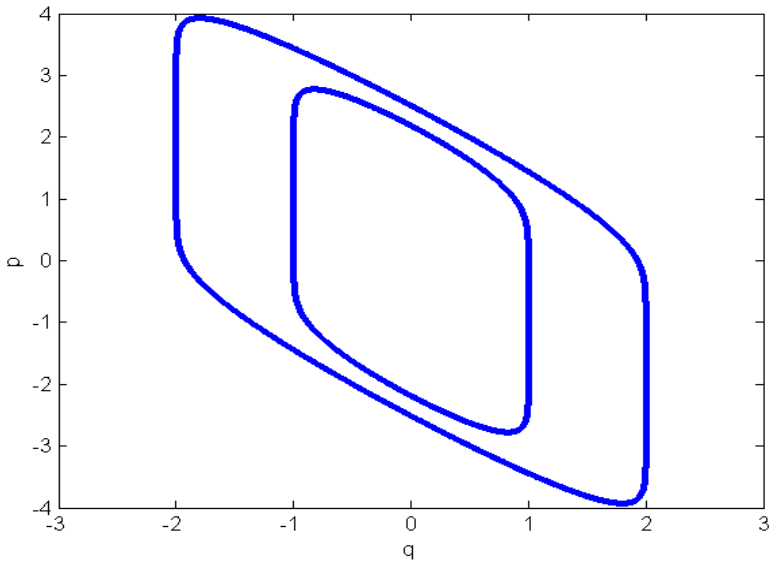

Fig. 5 Numerical level curves for problem (1.7)–(1.9), HBVM(3,2) method, $h = 10^{-4}$.

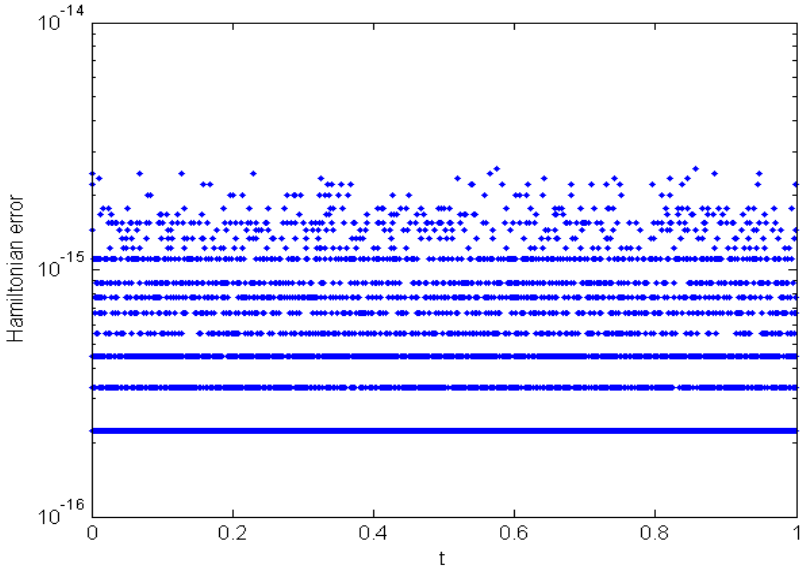

Fig. 6 Hamiltonian error for problem (1.7)–(1.9), HBVM(10,2) method, $h = 10^{-4}$.

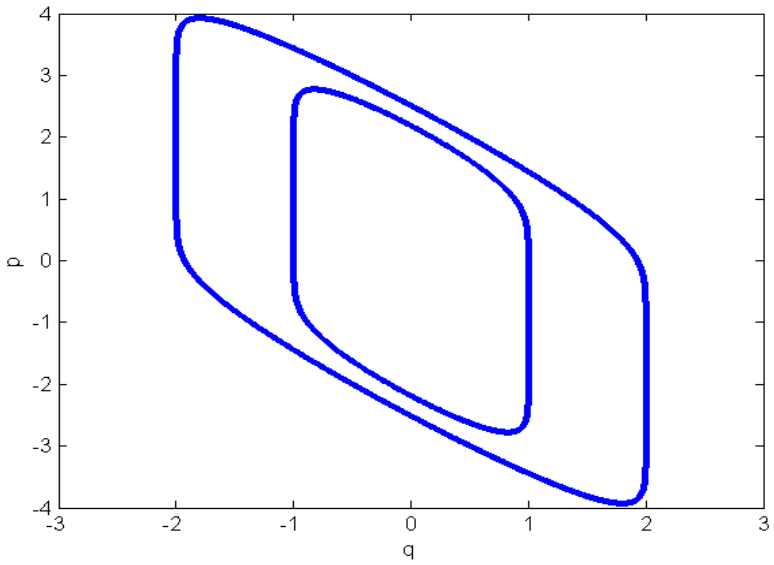

Fig. 7 Numerical level curves for problem (1.7)–(1.9), HBVM(10,2) method, $h = 10^{-4}$.

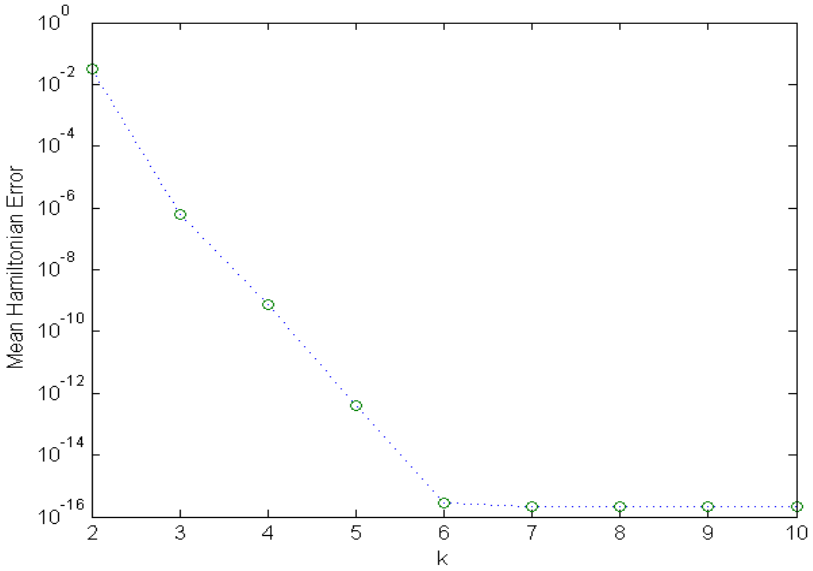

Fig. 8 Mean Hamiltonian error for problem (1.7)–(1.9), HBVM(k,2) method, $k = 2, \ldots, 8$, by using a stepsize $h = 10^{-3}$.

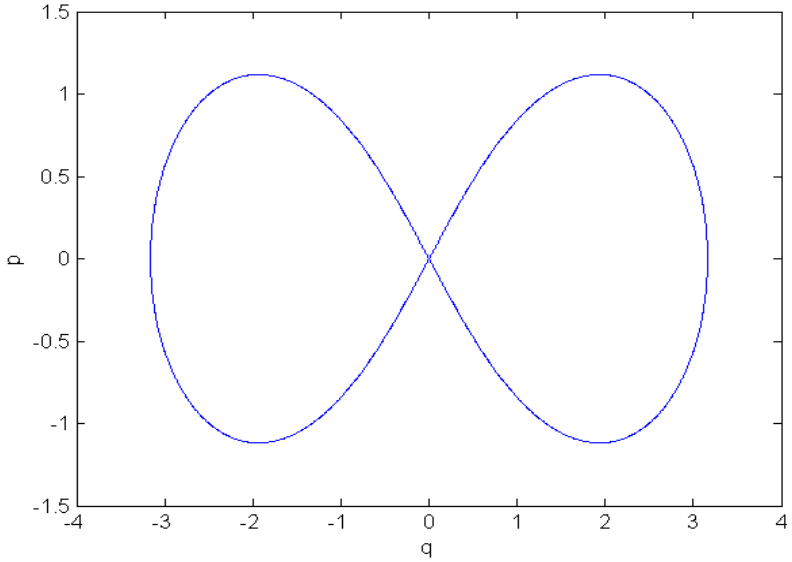

Fig. 9 Level curve for problem (4.18)–(4.19).

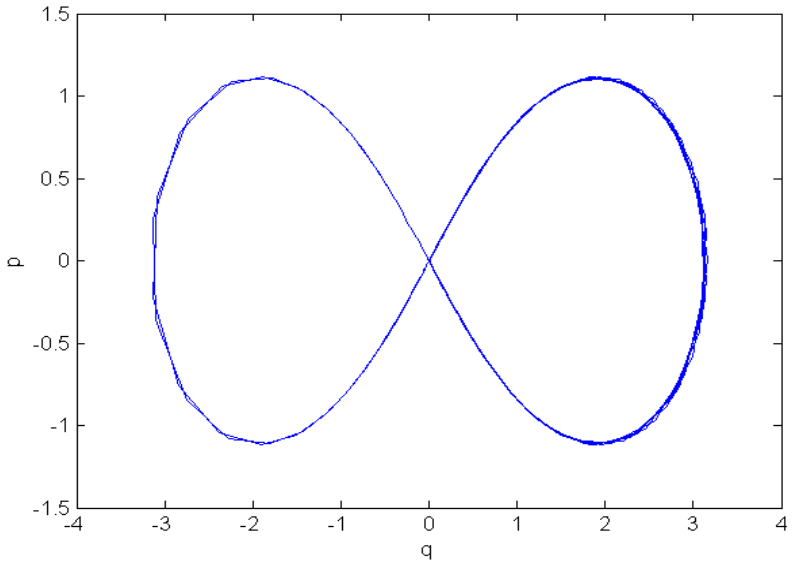

Fig. 10 Numerical level curve for problem (4.18)–(4.19), 2-stage Gauss method, $h = 10^{-2}$.

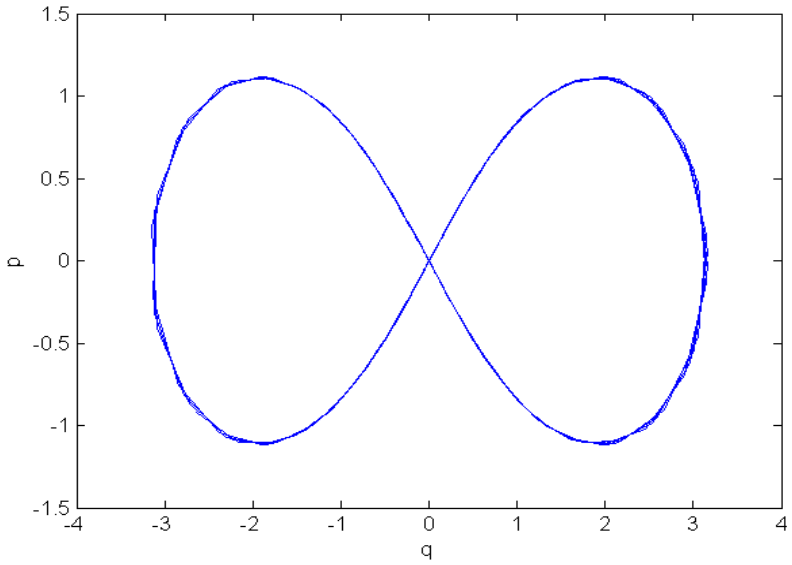

Fig. 11 Numerical level curve for problem (4.18)–(4.19), HBVM(4,2) method, $h = 10^{-2}$.

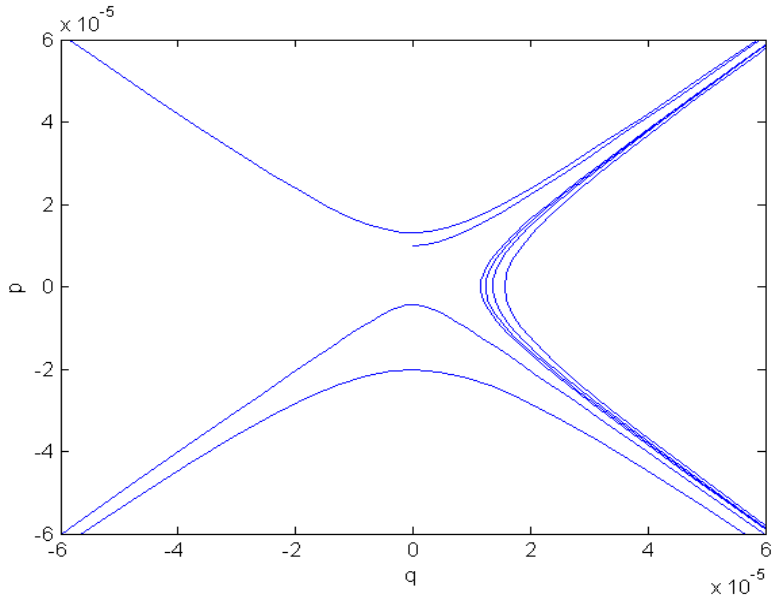

Fig. 12 Zoom of the numerical level curve for problem (4.18)–(4.19) around $(0, 0)$, 2-stage Gauss method, $h = 10^{-2}$.

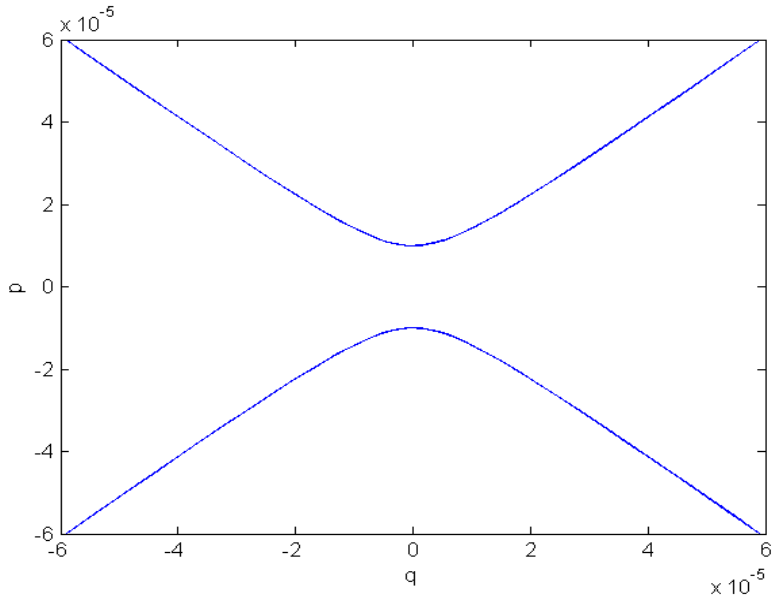

Fig. 13 Zoom of the numerical level curve for problem (4.18)–(4.19) around $(0, 0)$, HBVM(4,2) method, $h = 10^{-2}$.

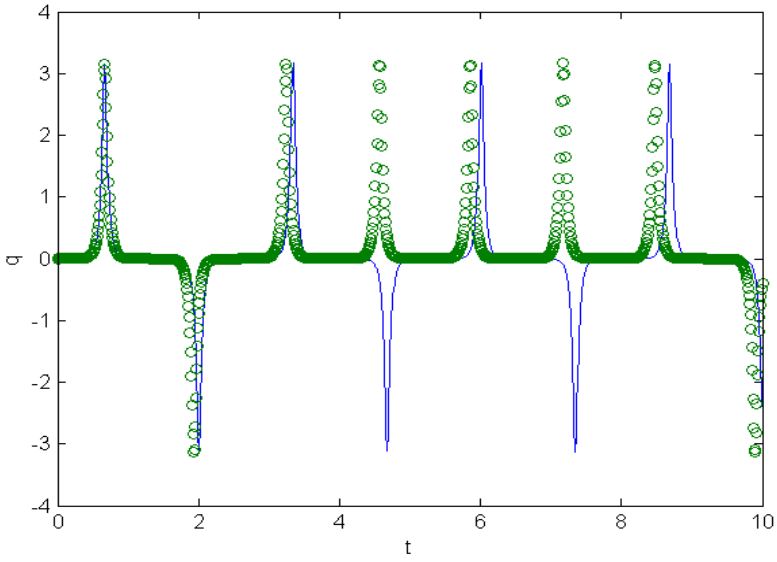

Fig. 14 Component q (solid line) and its numerical approximation (circles) by using the 2-stage Gauss method, $h = 10^{-2}$, for problem (4.18)–(4.19).

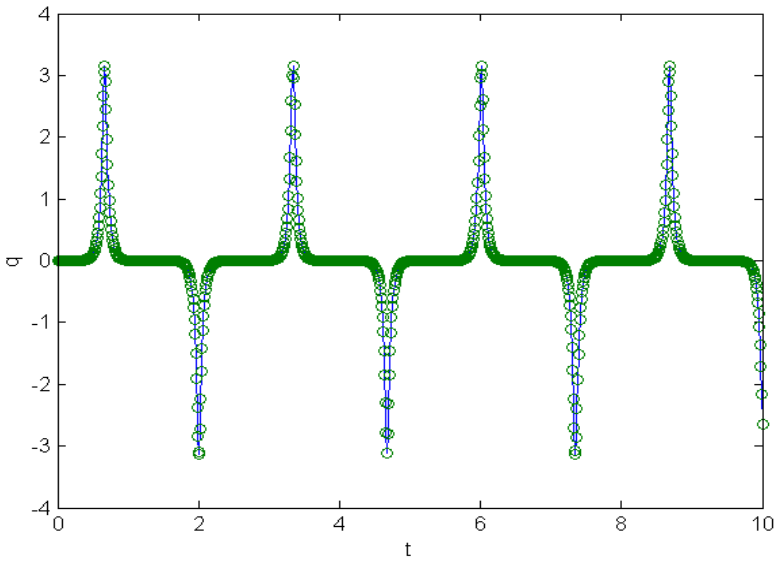

Fig. 15 Component q (solid line) and its numerical approximation (circles) by using the HBVM(4,2) method, $h = 10^{-2}$, for problem (4.18)–(4.19).

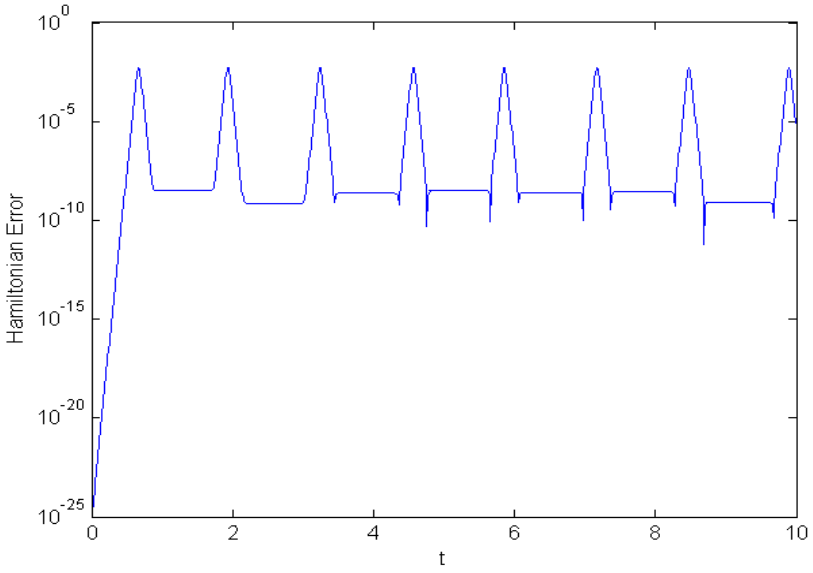

Fig. 16 Hamiltonian error for problem (4.18)–(4.19) by using the 2-stage Gauss method, $h = 10^{-2}$.

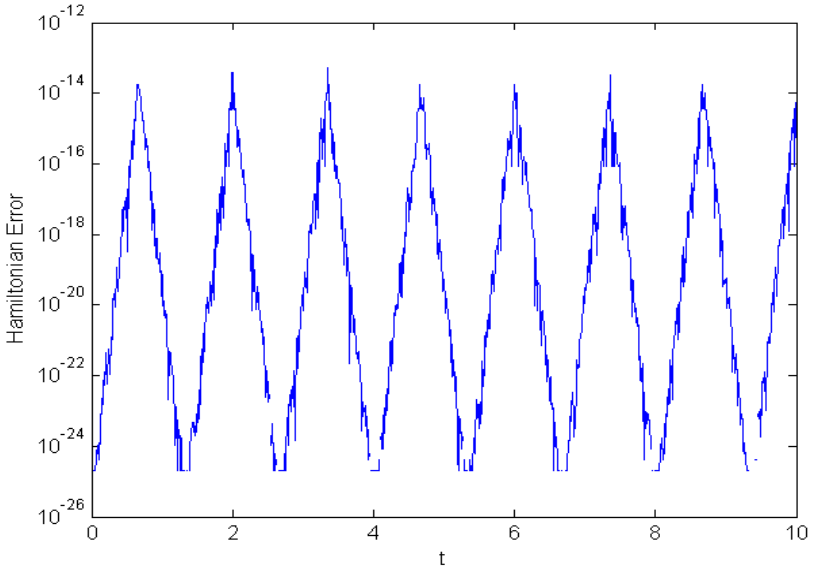

Fig. 17 Hamiltonian error for problem (4.18)–(4.19) by using the HBVM(4,2) method, $h = 10^{-2}$.

Figure 17. This allows to obtain a phase portrait (Figure 11) qualitatively correct, which is confirmed by the zoom in Figure 13. As a consequence, one also obtains a correct numerical approximation of the component q (similarly for p) as is shown in Figure 15.

4.5 Symmetry

We here prove that, provided that the abscissae $\{c_i\}$ are symmetrically distributed in the interval $[0,1]$, as is the case of the Gauss-Legendre nodes (see (3.3)), a HBVM(k,s) method is symmetric. In more detail [32], if applied to the initial value problem

$$y' = f(y), \qquad y(0) = y_0,$$

yielding the approximation $y_1 \approx y(h)$, then it will provide the same discrete solution, as well as the same internal stages, though in reversed order, when applied to the initial value problem

$$z' = -f(z), \qquad z(0) = y_1. \tag{4.20}$$

For proving this property, let us define the following matrices:

$$J_r = \begin{pmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{pmatrix} \in \mathbb{R}^{r\times r}, \qquad r = k, k+1, k+2,$$

$$L = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{k+1\times k+1}, \qquad D = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & \ddots & \\ & & & (-1)^{s-1} \end{pmatrix} \in \mathbb{R}^{s\times s},$$

and, by recalling the vector $\mathcal{I}_s^1$ defined at (2.15),

$$\hat{\mathcal{I}}_s = \begin{pmatrix} \mathcal{I}_s \\ \mathcal{I}_s^1 \end{pmatrix} \in \mathbb{R}^{k+1\times s}.$$

Moreover, by setting

$$0 \equiv c_0 < c_1 < \cdots < c_k < c_{k+1} \equiv 1, \tag{4.21}$$

we need to define the matrix

$$L\,\hat{\mathcal{I}}_s \equiv \Delta\mathcal{I}_s = \left(\int_{c_{i-1}}^{c_i} P_{j-1}(x)\mathrm{d}x \right)_{\substack{i=1,\ldots,k+1 \\ j=1,\ldots,s}}.$$

The following properties then hold true, provided that the abscissae are symmetrically distributed in the interval [0,1], i.e., by taking into account (4.21), $c_i = 1 - c_{k-i+1}$, $i = 0,\ldots,k+1$:

- $J_r^T = J_r^{-1} = J_r$;
- $J_k \Omega J_k = \Omega \quad \Rightarrow \quad \Omega J_k = J_k \Omega$;
- $J_{k+1}\Delta\mathcal{I}_s = \Delta\mathcal{I}_s D$;

- $J_k\mathcal{P}_s = \mathcal{P}_s D$;

where the last two points follow from the properties (3.10) and (3.9) of Legendre polynomials, respectively. The discrete solution generated by a HBVM(k,s) method can then be cast in vector form as

$$\left(-\hat{\mathbf{e}}\ \ I_{k+1}\right) \otimes I\, \hat{Y} = h\hat{\mathcal{I}}_s\mathcal{P}_s^T\Omega \otimes I\, f(\hat{Y}),$$

where $\hat{\mathbf{e}} \in \mathbb{R}^{k+1}$ is the unit vector, and

$$\hat{Y} = \begin{pmatrix} y_0 \\ Y \\ y_1 \end{pmatrix}, \qquad Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_k \end{pmatrix}.$$

Left-multiplication by $L \otimes I$ then gives

$$\hat{A} \otimes I\, \hat{Y} = h\hat{B} \otimes I\, f(\hat{Y}), \tag{4.22}$$

with

$$\hat{A} = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}, \quad \hat{B} = \left(\mathbf{0}\ \Delta\mathcal{I}_s\mathcal{P}_s^T\Omega\ \mathbf{0}\right) \quad \in \mathbb{R}^{k+1\times k+2}.$$

Since one easily realizes that

$$J_{k+1}\hat{A}J_{k+2} = -\hat{A},$$

the method would be symmetric provided that

$$J_{k+1}\hat{B}J_{k+2} = \hat{B} \tag{4.23}$$

holds true. In fact, by observing that

$$\hat{Z} = J_{k+2} \otimes I\, \hat{Y} = \begin{pmatrix} y_1 \\ J_k \otimes I\, Y \\ y_0 \end{pmatrix}$$

is the reversed-time discrete solution, left multiplication of (4.22) by $J_{k+1} \otimes I$ then gives:

$$\begin{aligned} \mathbf{0} = J_{k+1}\hat{A} \otimes I\, \hat{Y} - hJ_{k+1}\hat{B} \otimes I\, f(\hat{Y}) \ &= \ J_{k+1}\hat{A}J_{k+2}^2 \otimes I\, \hat{Y} - hJ_{k+1}\hat{B}J_{k+2}^2 \otimes I\, f(\hat{Y}) \\ = -\hat{A} \otimes I\hat{Z} - h\hat{B} \otimes I\, f(\hat{Z}). \end{aligned}$$

That is, the reversed-time vector satisfies the equation

$$\hat{A} \otimes I\, \hat{Z} = -h\hat{B} \otimes I\, f(\hat{Z}),$$

which consists in applying the HBVM(k,s) method to problem (4.20), thus providing the approximation $z_1 = y_0$, by using stages $Z = J_k \otimes I\, Y$. In order to prove (4.23), one has:

$$\begin{aligned} J_{k+1}\hat{B}J_{k+2} = \left(\mathbf{0}\ J_{k+1}\Delta\mathcal{I}_s\mathcal{P}_s^T\Omega J_k\ \mathbf{0}\right) \ &= \ \left(\mathbf{0}\ \Delta\mathcal{I}_s D\mathcal{P}_s^T J_k\Omega\ \mathbf{0}\right) \\ = \left(\mathbf{0}\ \Delta\mathcal{I}_s D(J_k\mathcal{P}_s)^T\Omega\ \mathbf{0}\right) \ &= \ \left(\mathbf{0}\ \Delta\mathcal{I}_s D^2\mathcal{P}_s^T\Omega\ \mathbf{0}\right) \ = \ \hat{B}, \end{aligned}$$

and the symmetry of the method follows.

4.6 Linear stability analysis

We now consider the linear stability analysis of HBVM(k,s): indeed, such methods can be defined independently from the problem of energy conservation, by considering a general function f in (4.7). As matter of fact, in Section 4.3 we have seen that HBVM(k,s) methods, with $k>s$, can be regarded as a low-rank generalization of the basic s-stage Gauss-Legendre method.

Then, let us apply one such a method to the celebrated test equation

$$y' = \lambda y, \qquad y(0) = y_0 \neq 0, \qquad \Re(\lambda) < 0.$$

Setting

$$\lambda = \alpha + i\beta, \qquad y = x_1 + ix_2,$$

with $\alpha, \beta, x_1, x_2 \in \mathbb{R}$ and i the imaginary unit, the test equation becomes:

$$\mathbf{x}' \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}' = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \equiv A\mathbf{x}, \qquad \mathbf{x}(0) = \mathbf{x}_0 \neq \mathbf{0}. \tag{4.24}$$

Defining the scalar function

$$V(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{x} \equiv \frac{1}{2}\|\mathbf{x}\|_2^2, \tag{4.25}$$

the application of a HBVM(k,s) method for solving (4.24) defines the polynomial u such that $u(0)=\mathbf{x}_0$ and, moreover,

$$\begin{aligned} u'(ch) &= \sum_{j=0}^{s-1} P_j(c) \sum_{i=1}^{k} b_i P_j(c_i) A u(c_i h) \;=\; A \sum_{j=0}^{s-1} P_j(c) \sum_{i=1}^{k} b_i P_j(c_i) u(c_i h) \\ &= A \sum_{j=0}^{s-1} P_j(c) \int_0^1 P_j(\tau) u(\tau h) \mathrm{d}\tau \;\equiv\; A \sum_{j=0}^{s-1} P_j(c) \int_0^1 P_j(\tau) \nabla V(u(\tau h)) \mathrm{d}\tau, \end{aligned}$$

where the third equality follows from the fact that the quadrature is exact for polynomials of degree $2s-1$. By considering that $u(0)=\mathbf{x}_0$, and that the new approximation is defined by

$$\mathbf{x}_1 \equiv u(h),$$

one then obtains:

$$\begin{aligned} \Delta V(\mathbf{x}_0) \;&=\; V(\mathbf{x}_1) - V(\mathbf{x}_0) \;=\; V(u(h)) - V(u(0)) \;=\; \int_0^h \frac{\mathrm{d}}{\mathrm{d}t} V(u(t)) \mathrm{d}t \\ &= \int_0^h \nabla V(u(t))^T u'(t) \mathrm{d}t \;=\; h \int_0^1 \nabla V(u(\tau h))^T A \sum_{j=0}^{s-1} P_j(\tau) \left[\int_0^1 P_j(c) \nabla V(u(ch)) \mathrm{d}c \right] \mathrm{d}\tau \\ &= \alpha h \sum_{j=0}^{s-1} \left\| \int_0^1 P_j(\tau) \nabla V(u(\tau h)) \mathrm{d}\tau \right\|_2^2 \;=\; \alpha h \sum_{j=0}^{s-1} \left\| \int_0^1 P_j(\tau) u(\tau h) \mathrm{d}\tau \right\|_2^2 \;\equiv\; \alpha h \Gamma^2. \end{aligned} \tag{4.26}$$

Moreover, the following result holds true.

Lemma 6. $\quad \Gamma^2 = 0 \quad \Rightarrow \quad \mathbf{x}_0 = \mathbf{0}.$

Proof. Indeed, one has:

$$\Gamma^2 = 0 \qquad \Rightarrow \qquad u'(ch) \equiv \mathbf{0} \quad \text{and} \quad \int_0^1 \overbrace{P_0(ch)}^{\equiv 1} u(ch)\mathrm{d}c = \int_0^1 u(ch)\mathrm{d}c = \mathbf{0}.$$

From the first equality one obtains $u(ch) \equiv \mathbf{x}_0$ and, therefore, from the second equality one derives $\mathbf{x}_0 = \mathbf{0}$. □

From (4.26) and Lemma 6, the following result then easily follows.

Theorem 9. For all $k \geq s$, and for any choice of the nodes, HBVM(k, s) is *perfectly A-stable*, i.e., its stability region coincides with the negative-real complex plane, $\mathbb{C}^-$.

Proof. From (4.26) and Lemma 6, one has, by considering (4.25) and that $\alpha = \Re(\lambda)$:

$$\|\mathbf{x}_1\|_2^2 = \|\mathbf{x}_0\|_2^2 + 2\alpha h\Gamma^2 < \|\mathbf{x}_0\|_2^2 \qquad \Leftrightarrow \qquad \Re(\lambda) < 0.$$

Consequently, a HBVM(k, s) method turns out to be perfectly A-stable, since its absolute stability region coincides with $\mathbb{C}^-$, for all $k \geq s \geq 1$. □

5 Implementation of the methods

In this section, we discuss the efficient implementation of HBVM(k, s) methods. In particular, we shall make clear that their computational cost depends essentially on s, in the sense that, for all $k \geq s$, the discrete problem turns out always to have block-dimension s. Two different nonlinear iteration procedures, the first based on the *blended implementation* of the methods the latter based on a particular splitting of the Butcher matrix defining the methods, are also sketched. The material in this section is based on [56, 15, 17, 21, 5, 26, 27, 30, 29, 12, 1, 51, 52, 28, 7, 8, 63].

5.1 Fundamental and silent stages

From (4.14)-(4.15), we see that a HBVM(k, s) method, with $k > s$, is defined by a Butcher matrix of rank s. Consequently, $k - s$ of the stages of the method can be expressed as a linear combination of the remaining s stages: we shall, therefore, name *fundamental stages* the latter ones, and *silent stages* the former ones. For this purpose, let us partition the stage vector Y as

$$Y = \begin{pmatrix} Y^{(1)} \\ Y^{(2)} \end{pmatrix}$$

where, by supposing for sake of brevity that the fundamental stages are the first s-ones,[5]

$$Y^{(1)} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_s \end{pmatrix}, \qquad Y^{(2)} = \begin{pmatrix} Y_{s+1} \\ \vdots \\ Y_k \end{pmatrix}.$$

Similarly, we partition matrices $\mathcal{I}_s$ and $\mathcal{P}_s$, respectively, as

[5] Indeed, this can be always achieved, by using a suitable permutation of the abscissae.

$$\mathcal{I}_s = \begin{pmatrix} \mathcal{I}_s^{(1)} \\ \mathcal{I}_s^{(2)} \end{pmatrix}, \qquad \mathcal{P}_s = \begin{pmatrix} \mathcal{P}_s^{(1)} \\ \mathcal{P}_s^{(2)} \end{pmatrix}, \qquad \mathcal{I}_s^{(1)}, \mathcal{P}_s^{(1)} \in \mathbb{R}^{s\times s}, \qquad \mathcal{I}_s^{(2)}, \mathcal{P}_s^{(2)} \in \mathbb{R}^{k-s\times s},$$

containing the corresponding rows as those of $Y^{(1)}$ and $Y^{(2)}$, respectively. Moreover, we also consider the partition

$$\Omega = \begin{pmatrix} \Omega_1 & \\ & \Omega_2 \end{pmatrix}, \qquad \Omega_1 \in \mathbb{R}^{s\times s}, \qquad \Omega_2 \in \mathbb{R}^{k-s\times k-s}.$$

Consequently, by setting $\mathbf{e}^{(1)}$ and $\mathbf{e}^{(2)}$ the unit vectors of length s and $k-s$, respectively, one obtains:

$$Y^{(1)} = \mathbf{e}^{(1)} \otimes y_0 + h\mathcal{I}_s^{(1)}\mathcal{P}_s^T\Omega \otimes I \begin{pmatrix} f(Y^{(1)}) \\ f(Y^{(2)}) \end{pmatrix}, \tag{5.1}$$

$$Y^{(2)} = \mathbf{e}^{(2)} \otimes y_0 + h\mathcal{I}_s^{(2)}\mathcal{P}_s^T\Omega \otimes I \begin{pmatrix} f(Y^{(1)}) \\ f(Y^{(2)}) \end{pmatrix}. \tag{5.2}$$

From (5.1), one then obtains that

$$\mathcal{P}_s^T\Omega \otimes I \begin{pmatrix} f(Y^{(1)}) \\ f(Y^{(2)}) \end{pmatrix} = \left(h\mathcal{I}_s^{(1)}\right)^{-1} \otimes I \left[Y^{(1)} - \mathbf{e}^{(1)} \otimes y_0\right],$$

which substituted into (5.2) gives:

$$\begin{aligned} Y^{(2)} &= \mathbf{e}^{(2)} \otimes y_0 + \mathcal{I}_s^{(2)} \left(\mathcal{I}_s^{(1)}\right)^{-1} \otimes I \left[Y^{(1)} - \mathbf{e}^{(1)} \otimes y_0\right] \\ &= \underbrace{\left[\mathbf{e}^{(2)} - \mathcal{I}_s^{(2)} \left(\mathcal{I}_s^{(1)}\right)^{-1} \mathbf{e}^{(1)}\right]}_{=\mathbf{a}} \otimes y_0 + \mathcal{I}_s^{(2)} \left(\mathcal{I}_s^{(1)}\right)^{-1} \otimes I\, Y^{(1)} \\ &\equiv \mathbf{a} \otimes y_0 + \mathcal{I}_s^{(2)} \left(\mathcal{I}_s^{(1)}\right)^{-1} \otimes I\, Y^{(1)}. \end{aligned}$$

Consequently, we can rewrite (5.1)-(5.2) as:

$$\begin{aligned} Y^{(1)} &= \mathbf{e}^{(1)} \otimes y_0 + h\mathcal{I}_s^{(1)}\mathcal{P}_s^T\Omega \otimes I \begin{pmatrix} f(Y^{(1)}) \\ f\left(\mathbf{a} \otimes y_0 + \mathcal{I}_s^{(2)}(\mathcal{I}_s^{(1)})^{-1} \otimes I\, Y^{(1)}\right) \end{pmatrix} \\ &\equiv \mathbf{e}^{(1)} \otimes y_0 + h\mathcal{I}_s^{(1)} \Big[(\mathcal{P}_s^{(1)})^T\Omega_1 \otimes I\, f(Y^{(1)}) + \\ &\quad (\mathcal{P}_s^{(2)})^T\Omega_2 \otimes I\, f\left(\mathbf{a} \otimes y_0 + \mathcal{I}_s^{(2)}(\mathcal{I}_s^{(1)})^{-1} \otimes I\, Y^{(1)}\right)\Big], \end{aligned}$$

involving only the fundamental stages, thus confirming that the actual discrete problem, to be solved at each time step, amounts to a set of s (generally) nonlinear equations, each having the same size as that of the continuous problem. For solving such a problem, one could use, e.g., a *fixed-point iteration*,

$$Y_{\ell+1}^{(1)} = \mathbf{e}^{(1)} \otimes y_0 + h\mathcal{I}_s^{(1)}\mathcal{P}_s^T\Omega \otimes I \begin{pmatrix} f(Y_\ell^{(1)}) \\ f\left(\mathbf{a} \otimes y_0 + \mathcal{I}_s^{(2)}(\mathcal{I}_s^{(1)})^{-1} \otimes I\, Y_\ell^{(1)}\right) \end{pmatrix}, \qquad \ell = 0, 1, \ldots, \tag{5.3}$$

or, if the case, a *simplified-Newton iteration*. In more details, setting

$$F(Y^{(1)}) = Y^{(1)} - \mathbf{e}^{(1)} \otimes y_0 - h\mathcal{I}_s^{(1)} \left[(\mathcal{P}_s^{(1)})^T \Omega_1 \otimes I\, f(Y^{(1)}) + (\mathcal{P}_s^{(2)})^T \Omega_2 \otimes I\, f\left(\mathbf{a} \otimes y_0 + \mathcal{I}_s^{(2)} (\mathcal{I}_s^{(1)})^{-1} \otimes I\, Y^{(1)} \right) \right],$$

one then solves,

$$[I - hC \otimes J_0]\, \Delta_\ell = -F(Y_\ell^{(1)}), \qquad Y_{\ell+1}^{(1)} = Y_\ell^{(1)} + \Delta_\ell, \qquad \ell = 0, 1, \ldots, \tag{5.4}$$

where $J_0 = J_f(y_0)$ and matrix C is defined as follows:

$$C = \mathcal{I}_s^{(1)} \left[(\mathcal{P}_s^{(1)})^T \Omega_1 + (\mathcal{P}_s^{(2)})^T \Omega_2 \mathcal{I}_s^{(2)} (\mathcal{I}_s^{(1)})^{-1} \right] \tag{5.5}$$

The following result holds true.

Theorem 10. The eigenvalues of matrix C, as defined in (5.5), coincide with those of matrix X_s defined in (3.13), that is the eigenvalues of the Butcher matrix of the s-stage Gauss method.

Proof. One has:

$$\begin{aligned} C &= \mathcal{I}_s^{(1)} \left[(\mathcal{P}_s^{(1)})^T \Omega_1 + (\mathcal{P}_s^{(2)})^T \Omega_2 \mathcal{I}_s^{(2)} (\mathcal{I}_s^{(1)})^{-1} \right] \\ &= \mathcal{I}_s^{(1)} \left[(\mathcal{P}_s^{(1)})^T \Omega_1 \mathcal{I}_s^{(1)} + (\mathcal{P}_s^{(2)})^T \Omega_2 \mathcal{I}_s^{(2)} \right] (\mathcal{I}_s^{(1)})^{-1} \;=\; \mathcal{I}_s^{(1)} \left[\mathcal{P}_s^T \Omega \mathcal{I}_s \right] (\mathcal{I}_s^{(1)})^{-1} \\ &\sim \mathcal{P}_s^T \Omega \mathcal{I}_s \;=\; \mathcal{P}_s^T \Omega \mathcal{P}_{s+1} \hat{X}_s \;=\; [I_s\ \mathbf{0}] \hat{X}_s \;=\; X_s. \quad \square \end{aligned}$$

Consequently, matrix C has *always* the same spectrum, independently of the choice of the *fundamental* and *silent abscissae*.[6] This, in turn, coincides with the set of the nonzero eigenvalues of the corresponding Butcher array (see Theorem 7). Nevertheless, its condition number is greatly affected from this choice. Clearly, a badly conditioned matrix C would affect the convergence of both the iterations (5.3) and (5.4). As an example, in Figures 18 and 19 we plot the condition number of matrix C corresponding to the following choices of the fundamental abscissae, in the case $k \geq s = 3$:

- the first s abscissae of the k ones (Figure 18);
- s approximately evenly spaced abscissae among the k ones (Figure 19).

As one may see, in the first case $\kappa(C)$ grows exponentially with k, whereas it is uniformly bounded in the second case. Because of this reason, we shall consider a more favorable formulation of the discrete problem, which will be independent of the choice of the fundamental abscissae.

5.2 Alternative formulation of the discrete problem

In order to overcome the previous drawback, the basic idea is to reformulate the discrete problem by considering as unknowns the coefficients, say

[6] I.e., the abscissae corresponding to the fundamental and silent stages, respectively.

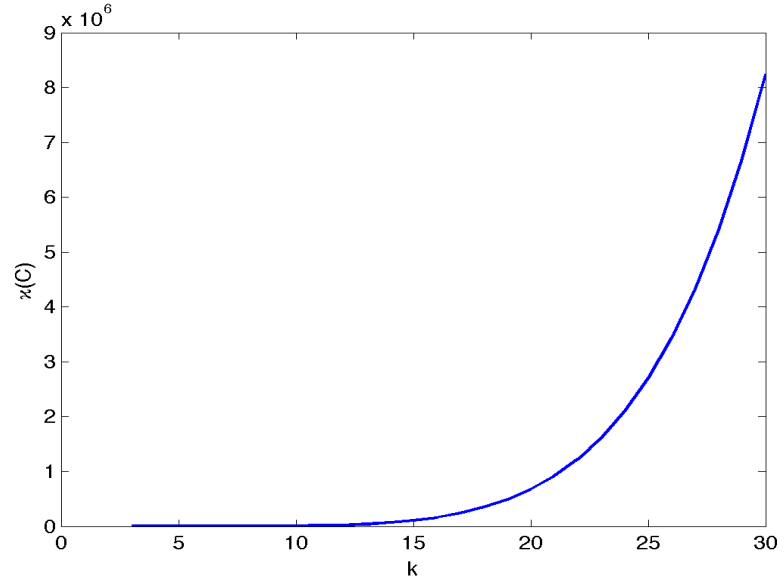

Fig. 18 Condition number of matrix (5.5), fundamental abscissae fixed at the first s $(= 3)$ ones.

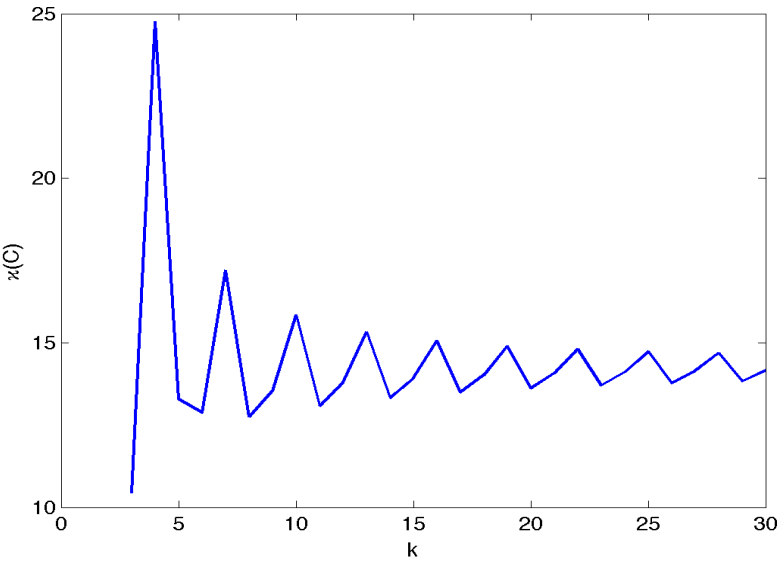

Fig. 19 Condition number of matrix (5.5), with the s $(= 3)$ fundamental abscissae approximately evenly spaced.

$$\hat{\gamma}_j = \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)), \qquad j = 0, \dots, s-1, \tag{5.6}$$

of the polynomial approximation defining the given HBVM(k, s) method (see (4.8)). In more details, recalling that

$$Y_i \equiv u(c_i h) = y_0 + h \sum_{j=0}^{s-1} \hat{\gamma}_j \int_0^{c_i} P_j(x) \mathrm{d}x, \qquad i = 1, \dots, k,$$

one may cast the discrete problem as follows,

$$\hat{\boldsymbol{\gamma}} \equiv \begin{pmatrix} \hat{\gamma}_0 \\ \vdots \\ \hat{\gamma}_{s-1} \end{pmatrix} = \mathcal{P}_s^T \Omega \otimes I \, f\left(\mathbf{e} \otimes y_0 + h\mathcal{I}_s \otimes I \, \hat{\boldsymbol{\gamma}}\right), \tag{5.7}$$

with the new approximation given by

$$y_1 = y_0 + h\hat{\gamma}_0.$$

We observe that (5.7) has always (block) dimension s, whatever is the value of k considered. For solving such a problem, one can still use a *fixed-point iteration*,

$$\hat{\boldsymbol{\gamma}}^{\ell+1} = \mathcal{P}_s^T \Omega \otimes I \, f\left(\mathbf{e} \otimes y_0 + h\mathcal{I}_s \otimes I \, \hat{\boldsymbol{\gamma}}^{\ell}\right), \qquad \ell = 0, 1, \dots, \tag{5.8}$$

whose implementation is straightforward. One can also consider a *simplified-Newton iteration*. Setting

$$F(\hat{\boldsymbol{\gamma}}) = \hat{\boldsymbol{\gamma}} - \mathcal{P}_s^T \Omega \otimes I \, f\left(\mathbf{e} \otimes y_0 + h\mathcal{I}_s \otimes I \, \hat{\boldsymbol{\gamma}}\right), \tag{5.9}$$

and, as before, $J_0 = J_f(y_0)$, it takes the form

$$[I - hC \otimes J_0] \, \Delta^\ell = -F(\hat{\boldsymbol{\gamma}}^\ell), \qquad \hat{\boldsymbol{\gamma}}^{\ell+1} = \hat{\boldsymbol{\gamma}}^\ell + \Delta^\ell, \qquad \ell = 0, 1, \dots, \tag{5.10}$$

where matrix C is now defined as follows:

$$C = \mathcal{P}_s^T \Omega \mathcal{I}_s = \mathcal{P}_s^T \Omega \mathcal{P}_{s+1} \hat{X}_s = (I_s \ \mathbf{0}) \hat{X}_s = X_s. \tag{5.11}$$

Consequently, the iteration (5.10) becomes:

$$[I - hX_s \otimes J_0] \, \Delta^\ell = -F(\hat{\boldsymbol{\gamma}}^\ell), \qquad \hat{\boldsymbol{\gamma}}^{\ell+1} = \hat{\boldsymbol{\gamma}}^\ell + \Delta^\ell, \qquad \ell = 0, 1, \dots. \tag{5.12}$$

Remark 6. It is worth noticing that (5.12) holds independently of the choice of the k abscissae $\{c_i\}$, the only requirement being the order $2s$ of the quadrature, so that the property $\mathcal{P}_s^T \Omega \mathcal{P}_{s+1} \hat{X}_s = (I_s \ \mathbf{0})$ holds true.

Remark 7. We observe that both matrices (5.5) and (5.11) share the same eigenvalues which, in turn, are the nonzero eigenvalues of the Butcher array of the given HBVM(k, s) method (see Theorem 7).

Remark 8. Even though, in general, by using the fixed point iteration (5.8) one is able to solve (5.7) quite inexpensively, when we have at hand a stiff oscillatory problem, this procedure could require a stepsize h so small as to be not practical. In such a case, the simplified-Newton iteration (5.10) would be more appropriate and this is the procedure that we shall consider in the sequel.

We observe that, remarkably enough, at each step of the simplified-Newton iteration we have to solve a linear system of dimension $sm \times sm$ of the form

$$[I - hX_s \otimes J_0]\,\mathbf{x} = \boldsymbol{\eta}, \tag{5.13}$$

whose coefficient matrix is thus *independent* of k and of the choice of the abscissae. Its cost is then approximately given by

$$\frac{2}{3}(sm)^3 \qquad \textit{flops},$$

due to the cost of the LU factorization of the coefficient matrix. We shall now consider an alternative iterative procedure, able to reduce the cost for the factorization to approximately

$$\frac{2}{3}m^3 \qquad \textit{flops}.$$

5.3 *Blended* HBVMs

We now introduce an iterative procedure for solving (5.13), which has been already successfully implemented in the computational codes `BiM` [27] and `BiMD` [30] for the numerical solution of stiff ODE-IVPs and linearly implicit DAEs up to order 3.

For this iterative procedure a linear analysis of convergence is provided. To this purpose, let us consider the classical test equation,

$$y' = \lambda y, \qquad \Re(\lambda) < 0. \tag{5.14}$$

In such a case, by setting as usual $q = h\lambda$, problem (5.13) becomes the linear system, of dimension s,

$$(I - qX_s)\mathbf{x} = \boldsymbol{\eta}. \tag{5.15}$$

The solution of this linear system is not affected by left-multiplication by ζX_s^{-1}, where $\zeta > 0$ is a free parameter to be chosen later. Thus, we obtain the following equivalent formulation of (5.15):

$$\zeta(X_s^{-1} - qI)\mathbf{x} = \zeta X_s^{-1}\boldsymbol{\eta} \equiv \boldsymbol{\eta}_1. \tag{5.16}$$

Let us define the *weighting function*

$$\theta(q) = I(1 - \zeta q)^{-1}, \tag{5.17}$$

satisfying the following properties:

- $\theta(q)$ is well defined for all $q \in \mathbb{C}^-$, since $\zeta > 0$;
- $\theta(0) = I$;
- $\theta(q) \to O$, as $q \to \infty$.

We can derive a further equivalent formulation of problem (5.15), as the *blending*, with weights $\theta(q)$ and $I - \theta(q)$ of the two equivalent formulations (5.15) and (5.16), thus obtaining

$$M(q)\mathbf{x} = \boldsymbol{\eta}(q), \tag{5.18}$$

with:

$$\begin{aligned} M(q) &= \theta(q)(I - qX_s) + \zeta(I - \theta(q))(X_s^{-1} - qI), \\ \boldsymbol{\eta}(q) &= \theta(q)\boldsymbol{\eta} + \zeta(I - \theta(q))X_s^{-1}\boldsymbol{\eta}. \end{aligned} \tag{5.19}$$

Equations (5.18)-(5.19) define the *blended formulation* of the original problem (5.15). The next step is now to devise an iterative procedure, defined by a suitable splitting, for solving (5.18)-(5.19). To this end we observe that, due to the properties of the weighting function $\theta(q)$ defined in (5.17), one has:

$$\begin{aligned} M(q) &\approx I, \qquad q \approx 0, \\ M(q) &\approx -\zeta q I, \qquad |q| \gg 1. \end{aligned}$$

Consequently, $N(q) = I(1 - \zeta q) \approx M(q)$, both for $q \approx 0$, and $|q| \gg 1$. It is then natural to define the following iterative procedure, for solving (5.18):

$$N(q)\mathbf{x}_{r+1} = (N(q) - M(q))\mathbf{x}_r + \boldsymbol{\eta}(q), \qquad r = 0, 1, \ldots.$$

That is, observing that $N(q)^{-1} = \theta(q)$:

$$\mathbf{x}_{r+1} = (I - \theta(q)M(q))\mathbf{x}_r + \theta(q)\boldsymbol{\eta}(q), \qquad r = 0, 1, \ldots. \tag{5.20}$$

Equation (5.20) defines the *blended iteration* associated with the blended formulation (5.18) of the problem. By considering that the solution, say $\mathbf{x}^*$, of (5.18) satisfies also (5.20), by setting

$$\mathbf{e}_r = \mathbf{x}_r - \mathbf{x}^* \tag{5.21}$$

the error at the r-th iteration, one then obtains the *error equation*

$$\mathbf{e}_{r+1} = (I - \theta(q)M(q))\mathbf{e}_r \equiv Z(q)\mathbf{e}_r, \qquad r = 0, 1, \ldots, \tag{5.22}$$

with $Z(q)$ the corresponding *iteration matrix*. Consequently, the iteration (5.20) will converge to the solution $\mathbf{x}^*$ of the problem iff the spectral radius of $Z(q)$,

$$\rho(q) = \max_{\xi \in \sigma(Z(q))} |\xi|,$$

is less than 1, where $\sigma(\cdot)$ denotes the spectrum of the matrix in argument. The set

$$\mathbb{D} = \{q \in \mathbb{C} : \rho(q) < 1\}$$

is the *region of convergence* of the iteration (5.20). The iteration will be said to be:

- A-convergent if $\mathbb{C}^- \subseteq \mathbb{D}$;
- L-convergent if, in addition, $\rho(q) \to 0$, as $q \to \infty$

Remark 9. A-convergent iterations are then appropriate when the underlying method is A-stable. Similarly, L-convergent iterations are appropriate in the case of L-stable methods. We observe that:

- $Z(0) = O \quad \Rightarrow \quad \rho(0) = 0$;
- $Z(q) \to O \quad \Rightarrow \quad \rho(q) \to 0, \quad \text{as} \quad q \to \infty$;
- $Z(q)$ is well-defined for all $q \in \mathbb{C}^-$, since $\zeta > 0$.

Consequently, for the blended iteration (5.20) A-convergence and L-convergence are equivalent to each other. From the maximum-modulus theorem, in turn, it follows that this is equivalent to requiring that the *maximum amplification factor*,

$$\rho^* = \sup_{\Re(q)=0} \rho(q) = \sup_{x\in\mathbb{R}} \rho(ix),$$

satisfies

$$\rho^* \leq 1.$$

For the blended iteration, due to the fact that $\rho(q) \to 0$, as $q \to \infty$, and since the matrix X_s is real, so that $\rho(\bar{q}) = \rho(q)$, one has actually to prove that

$$\rho^* = \max_{x>0} \rho(ix) \leq 1. \tag{5.23}$$

We shall choose the free positive parameter ζ, in order to minimize ρ^*, so that (5.23) turns out to be fulfilled for all $s \geq 1$. The following result holds true.

Theorem 11. $\mu \in \sigma(X_s) \quad \Leftrightarrow \quad \dfrac{q(\mu-\zeta)^2}{\mu(1-q\zeta)^2} \in \sigma(Z(q))$.

Proof. From (5.22), (5.19), (5.17), and (3.13), one obtains:

$$\begin{aligned}
Z(q) = I - \theta(q)M(q) \;&=\; I - \theta(q)\left[\theta(q)(I-qX_s) + \zeta(I-\theta(q))(X_s^{-1}-qI)\right] \\
&= I - \theta(q)^2\left[(I-qX_s) - \zeta^2 q(X_s^{-1}-qI)\right] \\
&= \theta(q)^2\left[(1+\zeta^2q^2-2\zeta q)I - I + qX_s + \zeta^2 qX_s^{-1} - \zeta^2 q^2 I\right] \\
&= q\theta(q)^2 X_s^{-1}\left[X_s^2 - 2\zeta X_s + \zeta^2 I\right] \;=\; q\theta(q)^2 X_s^{-1}(X_s-\zeta I)^2 \\
&\equiv q(X_s-\zeta I)^2\left[X_s(1-\zeta q)^2 I\right]^{-1},
\end{aligned}$$

from which the thesis easily follows. □

As a consequence, one obtains the following result.

Corollary 3. The maximum amplification factor (5.23) of the blended iteration (5.20) is given by:

$$\rho^* = \max_{\mu\in\sigma(X_s)} \frac{|\mu-\zeta|^2}{2\zeta|\mu|}.$$

Proof. One has:

$$\rho^* \;=\; \max_{x>0} \max_{\mu\in\sigma(X_s)} \frac{x|\mu-\zeta|^2}{|\mu|\,|1-ix\zeta|^2} \;=\; \max_{x>0} \frac{x}{1+\zeta^2x^2} \max_{\mu\in\sigma(X_s)} \frac{|\mu-\zeta|^2}{|\mu|}.$$

The thesis then follows immediately, by considering that

$$\max_{x>0} \frac{x}{1+\zeta^2x^2} \;=\; \frac{1}{2\zeta},$$

which is obtained at $x = \zeta^{-1}$. □

We are now in the position to choose the positive parameter ζ in order for ρ^* to be minimized. This clearly will depend on the eigenvalues of matrix X_s. Since this matrix is real, the complex ones occur as complex-conjugate pairs. Consequently, if we set

$$\mu_j = |\mu_j| \mathrm{e}^{i\phi_j}, \qquad j = 1, \ldots, s,$$

we can sort them by decreasing arguments:

$$\frac{\pi}{2} > \phi_1 > \phi_2 > \cdots > \phi_s > -\frac{\pi}{2},$$

due to the fact that

$$\Re(\mu_j) > 0, \qquad j = 1, \ldots, s.$$

Moreover, we can neglect the complex conjugate ones, thus obtaining:

$$\frac{\pi}{2} > \phi_1 > \ldots \phi_\ell \geq 0, \qquad \ell = \left\lceil \frac{s}{2} \right\rceil.$$

In addition to this, it turns out that the eigenvalues of matrix X_s also satisfy:

$$0 < |\mu_1| < \cdots < |\mu_\ell|,$$

as is shown in Figures 20 and 21, in the cases $s = 6$ and $s = 7$, respectively. In such a case, the following result holds true.

Theorem 12. ρ^* is minimized by choosing

$$\zeta = |\mu_1| \equiv \min_{\mu \in \sigma(X_s)} |\mu|, \tag{5.24}$$

resulting in

$$\rho^* = \frac{1}{2\zeta} \frac{|\mu_1 - \zeta|^2}{|\mu_1|}\bigg|_{\zeta = |\mu_1|}. \tag{5.25}$$

In such a case, one obtains:

$$\rho^* = 1 - \cos\phi_1 < 1. \tag{5.26}$$

Proof. For (5.24)-(5.25), see [26]. Concerning (5.26), one has:

$$\begin{aligned} \rho^* = \frac{1}{2|\mu_1|} \frac{|\mu_1 - |\mu_1||^2}{|\mu_1|} &= \frac{|\mu_1|^2 \left[(1-\cos\phi_1)^2 + (\sin\phi_1)^2\right]}{2|\mu_1|^2} \\ = \frac{1 + (\cos\phi_1)^2 + (\sin\phi_1)^2 - 2\cos\phi_1}{2} &= \frac{2 - 2\cos\phi_1}{2} \\ = 1 - \cos\phi_1. \quad \square \end{aligned}$$

Consequently, the blended implementation of HBVM(k, s) methods is *always* A-convergent and, therefore, L-convergent.

We can also characterize the speed of convergence when $q \approx 0$, by considering that, from Theorem 11, Corollary 3, and Theorem 12, it follows that

$$\rho(q) = \frac{|q|\,|\mu_1 - |\mu_1||^2}{|\mu_1|\,|1 - q|\mu_1||^2} = \frac{|\mu_1 - |\mu_1||^2}{|\mu_1|}|q| + O(|q|^2) \approx \tilde{\rho}|q|,$$

where the parameter

$$\tilde{\rho} = \frac{|\mu_1 - |\mu_1||^2}{|\mu_1|}$$

is called the *non-stiff* amplification factor. In Table 1 we list the relevant information for the iteration of HBVM(k, s) methods.

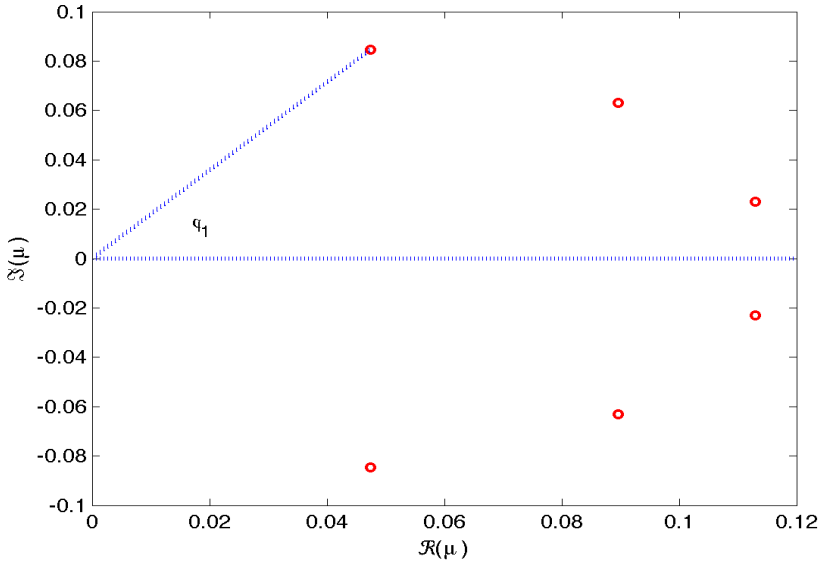

Fig. 20 Eigenvalues of matrix X_6.

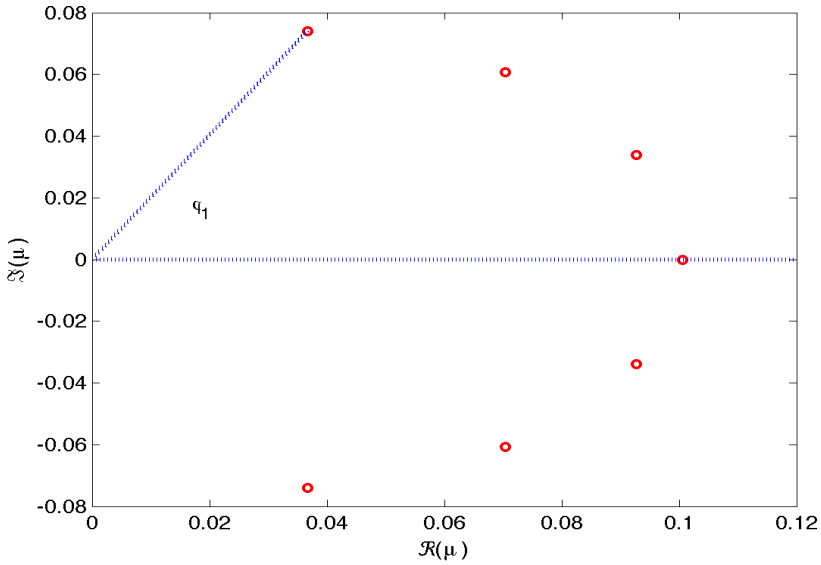

Fig. 21 Eigenvalues of matrix X_7.

Table 1 Blended iteration of HBVM(k,s) methods.

s	ζ	ρ^*	$\tilde{\rho}$
2	0.2887	0.1340	0.0774
3	0.1967	0.2765	0.1088
4	0.1475	0.3793	0.1119
5	0.1173	0.4544	0.1066
6	0.0971	0.5114	0.0993
7	0.0827	0.5561	0.0919

5.4 Actual blended implementation

Let us now sketch the blended implementation of HBVMs, when applied to a general, nonlinear system, also analyzing its complexity. In the case of the initial value problem

$$y' = f(y), \qquad y(0) = y_0 \in \mathbb{R}^m, \tag{5.27}$$

the previous arguments can be generalized in a straightforward way, by considering that now the weighting function becomes

$$\theta = I_s \otimes \Gamma^{-1}, \qquad \text{with} \qquad \Gamma = I - h\zeta J_0 \in \mathbb{R}^{m\times m}, \tag{5.28}$$

where h is the stepsize, ζ is the optimal parameter specified in the second column in Table 1, and J_0 is the Jacobian of f evaluated at y_0 (clearly, we are speaking about the very first step in the numerical integration).

From (5.9) and (5.12), we have to solve the *outer-inner* iteration described in Table 2 (where $e \in \mathbb{R}^k$ denotes the unit vector). Let us analyze its computational complexity, by denoting, as 1 *flop*, an elementary (binary) algebraic *fl*oating-point *op*eration. One obtains:

- μ: 1 *flop*, H_s: $2s-1$ *flops*, $\mathcal{T}_s$: ks *flops*;
- Γ: 1 Jacobian evaluation plus m^2+m *flops*;
- θ: $\frac{2}{3}m^3 - \frac{1}{2}m^2 - \frac{1}{6}m$ *flops* for computing the LU factorization of Γ;
- y^ℓ: $km + 2ksm$ *flops*;
- f^ℓ: k function evaluations;
- η^ℓ: $sm + 2ksm$ *flops*;
- $z^{\ell,r}$: $2sm^2$ *flops*;
- $t^{\ell,r}$: sm *flops*;
- $u^{\ell,r}$: $2s^2m + 2sm$ *flops*;
- $w^{\ell,r}$: $2s^2m + sm$ *flops*;
- $\Delta^{\ell,r+1}$: $4sm^2 + 3sm$ *flops*;
- $\hat{\gamma}^{\ell+1}$: sm *flops*.

Consequently, this algorithm has a fixed computational cost of 1 Jacobian evaluation and $\frac{2}{3}m^3 + \frac{1}{2}m^2 + ks + \frac{5}{6}m + 2s$ *flops*, plus, assuming that ν *inner* iterations are performed, a cost of k function evaluations and $4ksm + km + 2sm + \nu(6sm^2 + 4s^2m + 7sm)$ *flops* per *outer* iteration.

A simplified (and sometimes more efficient) procedure is that of performing a *nonlinear* iteration, obtained by performing exactly 1 inner iteration (i.e., that with $r = 0$ in the inner cycle in Table 2) in the above procedure, thus obtaining the algorithm depicted in Table 3. In such a case, the resulting computational cost is obtained as follows:

Table 2 Outer-inner iteration for the blended implementation of HBVMs.

$\mu = h\zeta, \quad Z_s = (X_s/\zeta)^{-1}, \quad H_s = hX_s, \quad \mathcal{T}_s = h\mathcal{I}_s, \quad W_s = P_s^T\Omega$
$\Gamma = I - \mu J_0$
$\theta = I_s \otimes \Gamma^{-1}$ % actually, Γ is factored LU
$\hat{\gamma}^0$ *given* % e.g., $\hat{\gamma}^0 = 0$
for $\ell = 0, 1, \ldots$
$y^\ell = e \otimes y_0 + \mathcal{T}_s \otimes I\, \hat{\gamma}^\ell$
$f^\ell = f(y^\ell)$
$\eta^\ell = \hat{\gamma}^\ell - W_s \otimes I\, f^\ell$ % $F(\hat{\gamma}^\ell)$
$\Delta^{\ell,0} = 0$
for $r = 0, 1, \ldots$
if $r > 0$
$z^{\ell,r} = [I_s \otimes J_0]\Delta^{\ell,r}$
$t^{\ell,r} = \Delta^{\ell,r} + \eta^\ell$
$u^{\ell,r} = [Z_s \otimes I]t^{\ell,r} - \mu z^{\ell,r}$
$w^{\ell,r} = t^{\ell,r} - [H_s \otimes I]z^{\ell,r}$
else
$u^{\ell,0} = [Z_s \otimes I]\eta^\ell$
$w^{\ell,0} = \eta^\ell$
end
$\Delta^{\ell,r+1} = \Delta^{\ell,r} - \theta\left[u^{\ell,r} + \theta(w^{\ell,r} - u^{\ell,r})\right]$
end $\Rightarrow$ returns Δ^ℓ
$\hat{\gamma}^{\ell+1} = \hat{\gamma}^\ell + \Delta^\ell$
end

- $\mathcal{T}_s$: ks *flops*;
- Γ: 1 Jacobian evaluation plus $m+1$ *flops* (we include the factor $h\zeta$ in the function before computing the Jacobian);
- θ: $\frac{2}{3}m^3 - \frac{1}{2}m^2 - \frac{1}{6}m$ *flops* for computing the LU factorization of Γ;
- y^ℓ: $km + 2ksm$ *flops*;
- f^ℓ: k function evaluations;
- η^ℓ: $sm + 2ksm$ *flops*;
- u^ℓ: $2s^2m$ *flops*;
- Δ^ℓ: $4sm^2 + 2sm$ *flops*;
- $\hat{\gamma}^{\ell+1}$: sm *flops*.

Consequently, this latter algorithm has a fixed computational cost of 1 Jacobian evaluation and $\frac{2}{3}m^3 + ks - \frac{1}{2}m^2 + \frac{5}{6}m + 1$ *flops*, plus a cost of k function evaluations and $4sm^2 + 4ksm + 2s^2m + km + 4sm$ *flops* per iteration.

Table 3 Nonlinear iteration for the blended implementation of HBVMs.

$$
\begin{array}{ll}
Z_s = (X_s/\zeta)^{-1}, \quad \mathcal{T}_s = h\mathcal{I}_s, \quad W_s = P_s^T\Omega & \\
\Gamma = I - (h\zeta)J_0 & \\
\theta = I_s \otimes \Gamma^{-1} & \% \text{ actually, } \Gamma \text{ is factored } LU \\
\hat\gamma^0 \quad given & \% \text{ e.g., } \hat\gamma^0 = 0 \\
for \quad \ell \;=\; 0, 1, \dots & \\
\quad y^\ell = e \otimes y_0 + \mathcal{T}_s \otimes I\, \hat\gamma^\ell & \\
\quad f^\ell = f(y^\ell) & \\
\quad \eta^\ell = \hat\gamma^\ell - W_s \otimes I\, f^\ell & \% \; F(\hat\gamma^\ell) \\
\quad u^\ell = [Z_s \otimes I]\eta^\ell & \\
\quad \Delta^\ell = \theta\left[\theta(u^\ell - \eta^\ell) - u^\ell\right] & \\
\quad \hat\gamma^{\ell+1} = \hat\gamma^\ell + \Delta^\ell & \\
end &
\end{array}
$$

5.5 The triangular splitting procedure

For the efficient implementation of the simplified-Newton iteration, additional iterative procedures have been also devised: some of them are based on suitable triangular splittings [51, 52, 1, 12]. However, the iteration defined in [51, 52], as well as its modified version defined in [1], result to be not effective for (5.12), due to the particular structure of the matrix X_s (see (3.13)).

The *blended implementation*, as already shown in Section 5.4, turns out more appropriate to solve the iteration (5.12), but we now shall describe another procedure introduced in [7], based on a particular triangular splitting, which appears to be even more favourable. The basic idea is similar to that introduced in [12] for the efficient implementation of RadauIIA collocation methods, but the framework, the general details, and results are completely different.

We start by introducing a set of *auxiliary abscissae* (whose actual choice will be explained in the sequel),

$$\tilde c_1 < \cdots < \tilde c_s, \tag{5.29}$$

the polynomial (see (5.6))

$$\tilde\gamma(c) = \sum_{j=0}^{s-1} P_j(c)\hat\gamma_j, \qquad c \in \mathbb{R}, \tag{5.30}$$

and a new set of (block) unknowns,

$$\tilde\gamma_i \equiv \sum_{j=0}^{s-1} P_j(\tilde c_i)\hat\gamma_j, \qquad i = 1, \dots, s, \tag{5.31}$$

that are the evaluations of (5.30) at the auxiliary abscissae (5.29). Introducing the (block) vector

$$\tilde{\boldsymbol{\gamma}} = \begin{pmatrix} \tilde{\gamma}_1 \\ \vdots \\ \tilde{\gamma}_s \end{pmatrix},$$

and the matrix

$$\tilde{\mathcal{P}} = (\, P_{j-1}(\tilde{c}_i)\,) \in \mathbb{R}^{s\times s}, \tag{5.32}$$

we can recast (5.31) in vector form as

$$\tilde{\boldsymbol{\gamma}} = \tilde{\mathcal{P}} \otimes I\hat{\boldsymbol{\gamma}}. \tag{5.33}$$

Left-multiplication of (5.12) by $\tilde{\mathcal{P}} \otimes I$, allows to recast the problem in terms of $\tilde{\boldsymbol{\gamma}}$ as:

$$\tilde{M}_0\tilde{\Delta}^\ell \equiv [I - h\tilde{A} \otimes J_0]\tilde{\Delta}^\ell = \boldsymbol{\eta}^\ell, \qquad \tilde{\boldsymbol{\gamma}}^{\ell+1} = \tilde{\boldsymbol{\gamma}}^\ell + \tilde{\Delta}^\ell, \qquad \ell = 0, 1, \dots \tag{5.34}$$

where

$$\tilde{A} = \tilde{\mathcal{P}} X_s \tilde{\mathcal{P}}^{-1}, \qquad \tilde{\Delta}^\ell = \tilde{\mathcal{P}} \otimes I\Delta^\ell, \qquad \boldsymbol{\eta}^\ell = -\tilde{\mathcal{P}} \otimes IF(\tilde{\mathcal{P}}^{-1} \otimes I\tilde{\boldsymbol{\gamma}}^\ell).$$

Remark 10. We stress that the matrix $\tilde{A}$ is independent of k but it only depends on s whatever is $k \geq s$. Consequently the following approach applies also in the particular case of $k = s$, that is, to the s-stage Gauss method.

With these premises, the choice of the auxiliary abscissae (5.29) will be done in such a way that the matrix $\tilde{A}$ can be factored as

$$\tilde{A} = \tilde{L}\tilde{U}, \tag{5.35}$$

with $\tilde{U}$ upper triangular with unit diagonal entries, and $\tilde{L}$ lower triangular with *constant* diagonal entries. In such a case, by following the approach of van der Houwen et al. [51, 52], we replace the iteration (5.34) with the *inner-outer iteration*

$$\begin{aligned} \left[I - h\tilde{L} \otimes J_0\right] \tilde{\Delta}^{\ell,r+1} &= h\tilde{L}(\tilde{U} - I) \otimes J_0\tilde{\Delta}^{\ell,r} + \boldsymbol{\eta}^\ell, \qquad r = 0, 1, \dots, \mu - 1, \\ \tilde{\boldsymbol{\gamma}}^{\ell+1} &= \tilde{\boldsymbol{\gamma}}^\ell + \tilde{\Delta}^{\ell,\mu}, \qquad \ell = 0, 1, \dots. \end{aligned} \tag{5.36}$$

In particular, since $\tilde{\Delta}^{\ell,0} = 0$, the choice $\mu = 1$ corresponds to the approach used by van der Houwen et al. to devise PTIRK methods [51], whereas, by choosing μ large enough to obtain full convergence of the inner-iteration (the one on r), one has that the outer iteration is equivalent to (5.34). Clearly, all the intermediate possibilities can be suitably considered. After the convergence of (5.36), the new approximation is computed (see (4.9)) as $y_1 = y_0 + h\hat{\gamma}_0$, where $\hat{\gamma}_0$ is retrieved from (5.33).

We observe that, since the diagonal entries of the factor $\tilde{L}$ are all equal to a given value, say d_s, then one needs to factor only the matrix

$$I - hd_sJ_0 \in \mathbb{R}^{m\times m}, \tag{5.37}$$

having the same size as that of the continuous problem (5.27), for performing the inner-outer iteration (5.36).

Remark 11. Actually, in a computational code this matrix can be kept constant until one needs to compute again the Jacobian matrix J_0 and/or to choose a different stepsize h. In this paper we deliberately do not take into account this issue, which requires a further analysis (see, e.g., [28] for the code described in [27]). Consequently in the numerical tests we shall use a constant stepsize and evaluate the Jacobian matrix at each step.

Concerning d_s the following result holds true.

Theorem 13. Assume that the factorization (5.35) is defined and that the diagonal entries of the factor $\tilde{L}$ are all equal to d_s. Then, with reference to (3.7), one has:

$$d_s = \begin{cases} \sqrt[s]{\Pi_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i-1}^2}, & \text{if } s \text{ is even,} \\ \sqrt[s]{\frac{1}{2}\Pi_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i}^2}, & \text{if } s \text{ is odd.} \end{cases} \tag{5.38}$$

Proof. Since, by hypothesis (5.35) holds true, we have

$$\det(X_s) = \det(\tilde{\mathcal{P}} X_s \tilde{\mathcal{P}}^{-1}) = \det(\tilde{A}) = \det(\tilde{L}\tilde{U}) = \det(\tilde{L}) = d_s^s,$$

since $\tilde{U}$ has unit diagonal entries and all the entries of $\tilde{L}$ are equal to d_s. Consequently

$$d_s = \sqrt[s]{\det(X_s)},$$

and (5.38) follows from Lemma 3. □

By virtue of the previous result, we have computed the auxiliary abscissae (5.29) by symbolically solving the following set of equations, which is equivalent to requiring that $\tilde{L}$ has constant diagonal entries:

$$\det(\tilde{A}_{\ell+1}) = d_s \det(\tilde{A}_\ell), \qquad \ell = 1, \ldots, s-1, \tag{5.39}$$

where $\tilde{A}_\ell$ denotes the principal leading submatrix of order ℓ of $\tilde{A}$ and d_s is given by (5.38).

We observe that the auxiliary abscissae (5.29) are s whereas the algebraic conditions (5.39) are $s-1$. This means that we can express $s-1$ abscissae as a function of the remaining *free abscissa*. We shall choose such free abscissa in order to optimize the convergence properties of the iteration. To this end, according to the linear analysis of convergence in [51] (see also [29]), we apply the splitting procedure (5.36) to the celebrated test equation (5.14). Since the problem is linear, the iteration (5.36) consists in solving only the inner iteration, so that we can skip the index ℓ of the outer iteration. By setting, as is usual, $q = h\lambda$ one obtains that the error equation associated with (5.36) is given by

$$\mathbf{e}_{r+1} = q(I - q\tilde{L})^{-1}\tilde{L}(\tilde{U} - I)\mathbf{e}_r \equiv Z(q)\mathbf{e}_r, \qquad r = 0, 1, \ldots, \mu - 1, \tag{5.40}$$

where $\mathbf{e}_r$ is the error vector at step r (see (5.21)), and $Z(q)$ is the iteration matrix induced by the splitting procedure. This latter will converge if and only if its spectral radius, $\rho(q)$, is less than 1. According to the definitions given in Section 5.3, in our case, since

$$Z(q) \to (I - \tilde{U}), \qquad q \to \infty,$$

which is a nilpotent matrix of index s, the iteration is L-convergent if and only if it is A-convergent. Since the iteration is well defined for all $q \in \mathbb{C}^-$ (due to the fact that the diagonal

entry of $\tilde{L}$, d_s, is positive as was shown in (5.38)) and $\rho(0) = 0$, from the maximum-modulus theorem it follows immediately that A-convergence is, in turn, equivalent to require that the *maximum amplification factor*,

$$\rho^* = \max_{x\in\mathbb{R}} \rho(ix)$$

is not larger than 1. Similarly to what seen for the blended implementation, the *non-stiff amplification factor*, which is now given by

$$\tilde{\rho} = \rho(\tilde{L}(\tilde{U} - I)), \tag{5.41}$$

governs the convergence of the iteration for small values of q, since

$$\rho(q) \approx \tilde{\rho}|q|, \qquad \text{for} \quad q \approx 0.$$

As seen before, the smaller ρ^* and $\tilde{\rho}$, the better the convergence properties of the iteration. For this reason, we choose the free auxiliary abscissa in order to (approximately) minimize the maximum amplification factor ρ^* of the iteration, while fulfilling the conditions (5.39).

In Table 4 we list the obtained values of the auxiliary abscissae, and the diagonal entry d_s of the corresponding factor $\tilde{L}$, for a generic HBVM(k, s) with $k \geq s$ and $s = 2, \ldots, 6$. One can see that in all cases the abscissae are distinct and inside the interval $[0, 1]$.

We emphasize that, for a given s, the distribution of the auxiliary abscissae $\{\tilde{c}_i\}$ and the factorization (5.35) of the matrix $\tilde{A}$, whose computation is responsible of the bulk of the computational effort during the integration process, are both independent of k. Consequently, when one is going to implement this class of methods, it is possible to conjecture a procedure to advance the time that dynamically selects the most appropriate value of k and, depending on the specific problem at hand and the configuration of the system at the given time. In so doing, one could easily switch, having fixed s, from a symplectic method (choosing $k = s$ (Gauss method)) to an energy preserving one (choosing $k > s$).

For sake of comparisons in Table 5 we list the maximum amplification factors and the nonstiff amplification factors for the following L-convergent iterations applied to the s-stage Gauss-Legendre methods:

(i) the iteration obtained by the original triangular splitting in [51];
(ii) the iteration obtained by the modified triangular splitting in [1];
(iii) the *blended* iteration obtained by the *blended implementation* of the methods, as defined in Table 3;
(iv) the iteration defined by (5.36).

We recall that the scheme (i) (first column) requires s real factorizations per iteration, whereas (ii)–(iv) only need one factorization per iteration, of a matrix having the same size as that of the continuous problem. From the parameters listed in the table, one concludes that the proposed splitting procedure is the most effective among all the considered ones.

Remark 12. For sake of accuracy, we stress that, when dealing with the actual implementation of HBVM(k, s) methods, only the blended iteration and the one described in (5.36) can be considered, whereas the triangular splitting defined in [51] and its modified version [1] turn out to be not effective, as was pointed out at the beginning of this section (Section 5.5). Consequently, in such a case, one has to consider only the last two groups of columns in Table 5.

Table 4 Auxiliary abscissae (5.29) for the HBVM(k,s) and s-stage Gauss method, $s = 2,\dots,6$, and the diagonal entry d_s (see (5.38)) of the corresponding factor $\tilde{L}$.

$s=2$	
$\tilde{c}_1$	0.26036297108184508789101036587842555
$\tilde{c}_2$	1
d_2	0.28867513459481288225457439025097873
$s=3$	
$\tilde{c}_1$	0.15636399930006671060146617869938122
$\tilde{c}_2$	0.45431868644630821020177903150137523
$\tilde{c}_3$	0.948
d_3	0.20274006651911333949661483325792675
$s=4$	
$\tilde{c}_1$	0.11004843257056123468614502691988075
$\tilde{c}_2$	0.31588689139705398683980065724981436
$\tilde{c}_3$	0.53114668286639796587351917750274705
$\tilde{c}_4$	0.884
d_4	0.15619699684601279005430416526875577
$s=5$	
$\tilde{c}_1$	0.08422178443461232088418554160093421 8
$\tilde{c}_2$	0.24861852058856201805181177902229394 4
$\tilde{c}_3$	0.41372526881522095641549864330214528 4
$\tilde{c}_4$	0.58709874897187711603088243675196238 4
$\tilde{c}_5$	0.9338
d_5	0.12702337351164258963093490787943281
$s=6$	
$\tilde{c}_1$	0.20985774196263657630356114041757724
$\tilde{c}_2$	0.36816786358152563671526302698797908
$\tilde{c}_3$	0.39607328223635472401921951140390213
$\tilde{c}_4$	0.62783521091780460858476326939502046
$\tilde{c}_5$	0.04580307227138364391540767310611717
$\tilde{c}_6$	0.94225
d_6	0.10702845478806509529222890981996019

5.6 Averaged amplification factors

The previous amplification factors measure the asymptotic speed of convergence when an infinite number of iterations is performed. In the computational practice, however, only a small number of iterations is performed. For this reason, it is useful also to check the *averaged amplification factors* over μ iterations, measuring the "average" convergence when μ inner iterations are performed. They are defined as follows:

$$\rho^*_\mu = \sup_{x\in\mathbb{R}} \sqrt[\mu]{\|Z(ix)^\mu\|}, \quad \tilde{\rho}_\mu = \sqrt[\mu]{\left\|\left[\tilde{L}(\tilde{U}-I)\right]^\mu\right\|}, \quad \rho^\infty_\mu = \sqrt[\mu]{\|(\tilde{U}-I)^\mu\|}, \tag{5.42}$$

where $\|\cdot\|$ is a suitable matrix norm. Clearly,

$$\lim_{\mu\to\infty}\rho^*_\mu = \rho^*, \qquad \lim_{\mu\to\infty}\tilde{\rho}_\mu = \tilde{\rho},$$

and

Table 5 Amplification factors for the triangular splitting in [51], the modified triangular splitting in [1], the *blended* iteration in Table 3, and the splitting (5.36), for the s-stage Gauss-Legendre formulae. The last two cases coincide with those for the HBVM(k,s) methods, $k \geq s$.

	(i): triangular splitting in [51]		(ii): triangular splitting in [1]		(iii): *blended* iteration in Table 3		(iv): triangular splitting (5.36)	
s	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$
2	0.1429	0.0833	0.1340	0.0774	0.1340	0.0774	0.1340	0.0774
3	0.3032	0.1098	0.2537	0.0856	0.2765	0.1088	0.2536	0.0870
4	0.4351	0.1126	0.3492	0.0803	0.3793	0.1119	0.3291	0.0859
5	0.5457	0.1058	0.4223	0.0730	0.4544	0.1066	0.3709	0.0654
6	0.6432	0.0973	0.4861	0.0702	0.5114	0.0993	0.4353	0.0650

Table 6 Averaged amplification factors (5.42) for the splitting (5.36), used for the HBVM(k,s) methods, $k \geq s$, when performing $\mu = 1, 2, 3$ iterations.

s	ρ_1^*	$\tilde{\rho}_1$	ρ_1^∞	ρ_2^*	$\tilde{\rho}_2$	ρ_2^∞	ρ_3^*	$\tilde{\rho}_3$	ρ_3^∞
2	0.1340	0.0774	0.0981	0.1340	0.0774	0	0.1340	0.0774	0
3	0.4492	0.0874	0.2606	0.3423	0.0873	0.1091	0.3087	0.0872	0
4	0.4751	0.1459	0.4751	0.4098	0.1200	0.1757	0.3848	0.1091	0.1294
5	0.8625	0.2045	0.7471	0.6775	0.1385	0.2872	0.5874	0.1154	0.1747
6	3.0797	0.2747	1.4988	1.2780	0.1356	0.4929	0.9451	0.1121	0.2697

$$\rho_\mu^\infty = 0, \qquad \forall \mu \geq s.$$

In Table 6 we list the averaged amplification factors when performing $\mu = 1, 2, 3$ iterations, and considering the infinity norm. As one may see, the resulting iteration turns out to be A-convergent also when using just one inner iteration, unless the case $s = 6$, which requires at least 3 inner iterations.

Remark 13. When performing only μ inner-iterations for solving the discrete problem generated by (5.14), we have to consider also the *outer* iteration (i.e., the one on ℓ in (5.36)), even though the problem is linear. In such a case, by setting E_ℓ the error at the ℓ-th outer iteration, it is quite straightforward to see that the error equation is now given by:

$$E_{\ell+1} = Z(q)^\mu E_\ell, \qquad \ell = 0, 1, \ldots.$$

Consequently, the convergence analysis made for (5.40) also applies to the present case.

5.7 Computational cost of the triangular splitting implementation

We now analyze the computational complexity of the triangular splitting procedure described in Section 5.5 when the method is applied for approximating the initial value problem (5.27), having dimension m, by using the stepsize h. We denote $e \in \mathbb{R}^k$ the unit vector and J_0 the Jacobian of f evaluated at y_0 (clearly, we refer to the very first step in the numerical integration). In order to reduce the computational cost of the procedure, we first multiply both sides of (5.36) by

$$h^{-1}\tilde{L}^{-1} \otimes I,$$

as done in [12]. Considering that

$$\tilde{L}^{-1} = d_s^{-1} I - S,$$

with S strictly lower triangular, system (5.36) then takes the form

$$\left[\frac{1}{hd_s} I - I \otimes J_0\right] \tilde{\Delta}^{\ell,r+1} = \frac{1}{h}(S \otimes I)\tilde{\Delta}^{\ell,r+1} + (C \otimes J_0)\tilde{\Delta}^{\ell,r} + R^\ell, \quad r = 0, 1, \ldots, \mu - 1, \tag{5.43}$$

where

$$C = \tilde{U} - I \quad \text{and} \quad R^\ell = \frac{1}{h}(\tilde{L}^{-1} \otimes I)\boldsymbol{\eta}^\ell.$$

As a consequence we have now to factor only the matrix

$$\frac{1}{hd_s} I - J_0 \in \mathbb{R}^{m \times m} \tag{5.44}$$

in place of (5.37). We now show that in the computation of

$$(C \otimes J_0)\tilde{\Delta}^{\ell,r}$$

at the right-hand side of (5.43), one can completely eliminate any $O(m^2)$ complexity term (that would be the leading one since, usually, $m \gg s$). This is true at the very first step, since by definition,

$$\tilde{\Delta}^{\ell,0} = 0.$$

By setting

$$\boldsymbol{w}_r = (C \otimes J_0)\tilde{\Delta}^{\ell,r} + R^\ell, \quad \text{and} \quad \boldsymbol{v}_{r+1} = h^{-1}(S \otimes I)\tilde{\Delta}^{\ell,r+1} + \boldsymbol{w}_r.$$

we have that $\boldsymbol{w}_0 = R^\ell$ and, after solving the first step of (5.43), which reads

$$\left[\frac{1}{hd_s} I - I \otimes J_0\right] \tilde{\Delta}^{\ell,1} = \frac{1}{h}(S \otimes I)\tilde{\Delta}^{\ell,1} + \boldsymbol{w}_0 \equiv \boldsymbol{v}_1,$$

for the unknown $\tilde{\Delta}^{\ell,1}$, we are able to compute the term

$$(I \otimes J_0)\tilde{\Delta}^{\ell,1} = (hd_s)^{-1}\tilde{\Delta}^{\ell,1} - \boldsymbol{v}_1,$$

at a cost of $O(ms)$ operations. It follows that

$$(C \otimes J_0)\tilde{\Delta}^{\ell,1} = (C \otimes I)\left[(I \otimes J_0)\tilde{\Delta}^{\ell,1}\right] = (C \otimes I)\left[(hd_s)^{-1}\tilde{\Delta}^{\ell,1} - \boldsymbol{v}_1\right].$$

and thus $\boldsymbol{w}_1 = (C \otimes J_0)\tilde{\Delta}^{\ell,1} + R^\ell$ can be computed with $O(s^2 m)$ *flops*. The same procedure can then be repeated in the subsequent steps, as shown in Table 7, thus avoiding the $O(s^2 m^2)$ complexity term. Let us now analyze the computational cost of each step of the procedure in Table 7, in terms of *flops*:

- $\mathcal{T}_s$: ks *flops*, H: $\frac{s(s+1)}{2}$ *flops*, μ_s: 1 *flop*;
- θ: 1 Jacobian evaluation plus $\frac{2}{3}m^3 - \frac{1}{2}m^2 + \frac{11}{6}m$ *flops* ($2m$ operations plus those required to compute a LU factorization);
- y^ℓ: $km + 2ksm$ *flops*;

Table 7 Outer-inner iteration for the triangular splitting implementation of HBVMs.

$\mathcal{T}_s = h\mathcal{I}_s, \quad W_s = P_s^T\Omega, \quad \tilde{A} = \tilde{P}X_s\tilde{P}^{-1} \equiv \tilde{L}\tilde{U}, \quad C = \tilde{U} - I, \quad H = \frac{1}{h}\tilde{L}^{-1} \equiv (H_{i,j}), \quad \mu_s = hd_s$

$\hat{\gamma}^0$ *given* % e.g., $\hat{\gamma}^0 = 0$

$\theta = (\mu_s^{-1}I - J_0)^{-1}$

for $\ell = 0, 1, \ldots$

$y^\ell = e \otimes y_0 + \mathcal{T}_s \otimes I\hat{\gamma}^\ell$

$f^\ell = f(y^\ell)$

$\eta^\ell = [\tilde{P} \otimes I]\left[(W_s \otimes I)f^\ell - \hat{\gamma}^\ell\right]$

$R^\ell = (H \otimes I)\eta^\ell$

$\tilde{\Delta}^{\ell,0} = 0$

$w^{\ell,0} = R^\ell$

for $r = 0, 1, \ldots$

for $i = 1, \ldots, s$ % resolution of the block-triangular system by solving s systems of dimension m

$W_i^{\ell,r} = (w_{(i-1)m+1}^{\ell,r}, \ldots, w_{im}^{\ell,r})$

for $j = 1, \ldots, i-1$

$W_i^{\ell,r} = W_i^{\ell,r} + H_{i,j}W_j^{\ell,r}$

end

$v_i^{\ell,r+1} = W_i^{\ell,r}$

$W_i^{\ell,r} = W_i^{\ell,r}\theta^T$

end

$\tilde{\Delta}^{\ell,r+1} = (W_1^{\ell,r}, \ldots, W_s^{\ell,r})^T$

$v^{\ell,r+1} = (v_1^{\ell,r+1}, \ldots, v_s^{\ell,r+1})^T$

$w^{\ell,r+1} = (C \otimes I)\left[\mu_s^{-1}\tilde{\Delta}^{\ell,r+1} - v^{\ell,r+1}\right] + R^\ell$

end $\Rightarrow$ returns $\tilde{\Delta}^\ell$

$\hat{\gamma}^{\ell+1} = \hat{\gamma}^\ell + [\tilde{P}^{-1} \otimes I]\tilde{\Delta}^\ell$

end

- f^ℓ: k function evaluations;
- η^ℓ: $2s^2m + 2ksm + sm$ *flops*;
- R^ℓ: s^2m *flops* (taking into account that H is lower triangular);
- $W_i^{\ell,r}$: $2sm^2 + s^2m - sm$ *flops* to solve the block-triangular system;
- $w^{\ell,r+1}$: $2s^2m + 3sm$;
- $\gamma^{\ell+1}$: $2s^2m + sm$ *flops*;

Consequently, this algorithm has a fixed computational cost of 1 Jacobian evaluation and $\frac{2}{3}m^3 - \frac{1}{2}m^2 + ks + \frac{1}{2}s^2 + \frac{11}{6}m + \frac{1}{2}s + 1$ *flops*, plus, assuming that ν *inner* iterations are performed, a cost of k function evaluations and $4ksm + 5s^2m + km + 2sm + \nu(3s^2m + 2sm^2 + 2sm)$ *flops* per *outer* iteration.

5.8 The triangular splitting procedure for separable Hamiltonian problems.

We now describe, according to [8], how the triangular splitting procedure can be further improved, when the method is applied to a separable Hamiltonian problem with the following simpler Hamiltonian:

$$H(q,p) = \frac{1}{2}p^T p + U(q).$$

Consequently, the problem assumes the simplified form

$$q' = p, \qquad p' = -\nabla U(q), \qquad q(0) = q_0, \quad p(0) = p_0 \in \mathbb{R}^m, \tag{5.45}$$

which we plan to solve on the interval $[0, h]$. The HBVM method (4.14) provides the approximations (see (4.12))

$$q_1 = q_0 + h\mathbf{b}^T \otimes IP \approx q(h), \qquad p_1 = p_0 - h\mathbf{b}^T \otimes I\nabla U(Q) \approx p(h),$$

with stage vectors

$$Q = (Q_1^T, \dots, Q_k^T)^T, \qquad P = (P_1^T, \dots, P_k^T)^T$$

given by (see (4.11)):

$$Q = \mathbf{e} \otimes q_0 + h\mathcal{I}_s\mathcal{P}_s^T\Omega \otimes I\, P, \qquad P = \mathbf{e} \otimes p_0 - h\mathcal{I}_s\mathcal{P}_s^T\Omega \otimes I\, \nabla U(Q), \tag{5.46}$$

where $\mathbf{e} \in \mathbb{R}^k$ is the unit vector and $\nabla U(Q) = (\nabla U(Q_1)^T, \dots, \nabla U(Q_s)^T)^T$. Substituting the second equation of (5.46) into the first one, also considering that $\mathcal{I}_s\mathcal{P}_s^T\Omega\mathbf{e} = \mathbf{c}$, and taking into account (3.13) and (5.11), one has

$$Q = \mathbf{e} \otimes q_0 + h\mathbf{c} \otimes p_0 - h^2\mathcal{P}_{s+1}\hat{X}_s X_s \mathcal{P}_s^T\Omega \otimes I\, \nabla U(Q). \tag{5.47}$$

This problem has (block) dimension k. In order to recover a problem of (block) dimension s, independently of k, similarly to what has been done in Section 5.2, we consider as unknown the (block) vector with the coefficients of the underlying polynomial of degree s:

$$\hat{\boldsymbol{\gamma}} = \mathcal{P}_s^T\Omega \otimes I\, \nabla U(Q). \tag{5.48}$$

Substituting (5.47) into (5.48), we then obtain the following discrete problem:

$$F(\hat{\boldsymbol{\gamma}}) \equiv \hat{\boldsymbol{\gamma}} - \mathcal{P}_s^T\Omega \otimes I\, \nabla U\left(\mathbf{e} \otimes q_0 + h\mathbf{c} \otimes p_0 - h^2\mathcal{P}_{s+1}\hat{X}_s X_s \otimes I\, \hat{\boldsymbol{\gamma}}\right) = \mathbf{0}. \tag{5.49}$$

By taking into account that

$$\mathcal{P}_s^T\Omega\mathcal{P}_{s+1}\hat{X}_s X_s = [I_s\ \mathbf{0}]\hat{X}_s X_s = X_s^2,$$

the application of the simplified-Newton method for solving (5.49) results in the following iteration:

$$\left[I + h^2 X_s^2 \otimes \nabla^2 U(q_0)\right]\Delta^\ell = -F(\hat{\boldsymbol{\gamma}}^\ell), \qquad \hat{\boldsymbol{\gamma}}^{\ell+1} = \hat{\boldsymbol{\gamma}}^\ell + \Delta^\ell, \qquad \ell = 0, 1, \dots. \tag{5.50}$$

This is the problem which we now attack by means of a triangular splitting procedure. As before, we introduce a set of auxiliary abscissae (5.29), the matrix $\tilde{\mathcal{P}}$ defined as in (5.32) and the unknown vector $\tilde{\boldsymbol{\gamma}}$ in form (5.33). Similarly as previously done, left-multiplication

of (5.50) by $\tilde{\mathcal{P}} \otimes I$ allows to recast the problem in terms of $\tilde{\boldsymbol{\gamma}}$, thus obtaining the following equivalent linear system,

$$\left[I + h^2 \tilde{A} \otimes \nabla^2 U(q_0)\right] \tilde{\Delta}^\ell = \boldsymbol{\eta}^\ell, \tag{5.51}$$

where (see (5.49))

$$\tilde{A} = \tilde{\mathcal{P}} X_s^2 \tilde{\mathcal{P}}^{-1}, \qquad \tilde{\Delta}^\ell = \tilde{\mathcal{P}} \otimes I \Delta^\ell, \qquad \boldsymbol{\eta}^\ell = -\tilde{\mathcal{P}} \otimes I \, F(\tilde{\mathcal{P}}^{-1} \otimes I \, \tilde{\boldsymbol{\gamma}}^\ell).$$

As before the abscissae (5.29) are chosen in such a way that $\tilde{A}$ admits the factorization (5.35) with $\tilde{U}$ upper triangular with unit diagonal entries, and $\tilde{L}$ lower triangular with diagonal entries all equals to

$$d_s = \sqrt[s]{\det X_s^2} \tag{5.52}$$

(this can be proved in a similar way as done in Theorem 13). As we have already seen, this allows one to express $s-1$ abscissae as a function of a remaining *free abscissa*. The free abscissa will be chosen in order to (approximately) optimize the convergence properties of the following inner iteration, coupled with the outer iteration (5.51):

$$\left[I + h^2 \tilde{L} \otimes \nabla^2 U(q_0)\right] \tilde{\Delta}^{\ell,r+1} = h^2 \left[\tilde{L} - \tilde{A}\right] \otimes \nabla^2 U(q_0) \tilde{\Delta}^{\ell,r} + \boldsymbol{\eta}^\ell, \qquad r = 0, 1, \ldots. \tag{5.53}$$

Similarly as before, we have now that the coefficient matrix is lower block triangular, with diagonal block entries all equals to

$$I + h^2 d_s \nabla^2 U(q_0) \in \mathbb{R}^{m \times m},$$

which is a symmetric matrix having the same size as that of the continuous problem (5.45), independently of s. According to the analysis in [29], a linear convergence analysis of the iteration (5.53) is obtained by considering the scalar problem

$$y'' = -\nu^2 y, \qquad \nu \in \mathbb{R}.$$

By setting $x = h\nu$ one obtains that the corresponding iteration matrix is given by

$$Z(x^2) = x^2 (I + x^2 \tilde{L})^{-1} \tilde{L} (I - \tilde{U}). \tag{5.54}$$

Let $\rho(x^2)$ denote the spectral radius of the iteration matrix (5.54). Consequently, the iteration will be convergent if and only if $\rho(x^2) < 1$. As before we observe that

$$\rho(x^2) \to 0 \qquad \text{as} \qquad x \to \infty.$$

The maximum amplification factor of the iteration is then defined as

$$\rho^* = \max_{x \geq 0} \rho(x^2).$$

Moreover, according to the analysis in [29], one has

$$\rho(x^2) \approx \tilde{\rho} x^2, \qquad \text{for} \qquad x \approx 0,$$

with the non-stiff amplification factor $\tilde{\rho}$ formally still given by (5.41). Clearly, the smaller the parameters ρ^* and $\tilde{\rho}$, the better the iteration properties.

Table 8 Auxiliary abscissae (5.29) for the HBVM(k,s) and s-stage Gauss method, $s = 2,\dots,6$ for separable Hamiltonian problems, and the diagonal entry d_s (see (5.52)) of the corresponding factor $\tilde{L}$.

$s=2$	
$\tilde{c}_1$	0.3
$\tilde{c}_2$	1
d_2	1/12

$s=3$	
$\tilde{c}_1$	0.18838718112360613351895144351002434 2
$\tilde{c}_2$	0.42541922141818347835430054689468788 8
$\tilde{c}_3$	0.87
d_3	0.04110353457217450169152685538590981 74

$s=4$	
$\tilde{c}_1$	0.13839179546033992293368756080079890 5
$\tilde{c}_2$	0.29921388106651576439415717217989267 3
$\tilde{c}_3$	0.53860119088715235705995710475964603 6
$\tilde{c}_4$	0.895
d_4	0.02439750182371332948385961590600250 47

$s=5$	
$\tilde{c}_1$	0.26469193829071739344114929036861174 0
$\tilde{c}_2$	0.34712660870759669498183464008420098 8
$\tilde{c}_3$	0.05364559835125359823531505991964866 1
$\tilde{c}_4$	0.49913966664119541624914013850859470 2
$\tilde{c}_5$	0.771
d_5	0.01613493741827826427253049380882892 56

$s=6$	
$\tilde{c}_1$	0.22598589148959878075904037670795849 6
$\tilde{c}_2$	0.36643189170258729608056886185439036 4
$\tilde{c}_3$	0.43980743420584080268412154191319197 1
$\tilde{c}_4$	0.04059509783777282807206774082004015 12
$\tilde{c}_5$	0.61582504525880070596908268045894827
$\tilde{c}_6$	0.8865
d_6	0.01145509013432089422202647128222134 70

In addition to this, by repeating similar arguments as those reported in Section 5.6, we also introduce the averaged amplification factors for the iteration (5.53), measuring the "average" convergence when exactly μ iterations are performed. They are defined as (see (5.54))

$$\rho^*_\mu = \sup_{x\in\mathbb{R}} \sqrt[\mu]{\|Z(x^2)^\mu\|}, \quad \tilde{\rho}_\mu = \sqrt[\mu]{\left\|\left[\tilde{L}(\tilde{U}-I)\right]^\mu\right\|}, \quad \rho^\infty_\mu = \sqrt[\mu]{\|(\tilde{U}-I)^\mu\|}, \tag{5.55}$$

where $\|\cdot\|$ is a suitable matrix norm (compare with (5.42)). These parameters are very useful in the actual implementation of the methods, where generally only a finite number of iteration is performed. For this reason we choose the free abscissa in order to (approximately) minimize the values of $\rho^*_\mu, \mu = 1,2,3,4$: in particular, it is optimized the first parameter which turns out to be less than 1. This is different from what done in [8], where the free abscissa has been chosen in order to (approximately) minimize the maximum amplification factor ρ^*.

Table 9 Amplification factors for the splitting (5.53), for the HBVM(k,s) methods, $k \geq s$, applied to a separable Hamiltonian problem.

s	ρ_1^*	ρ_2^*	ρ_3^*	ρ_4^*	$\tilde{\rho}_1$	$\tilde{\rho}_2$	$\tilde{\rho}_3$	$\tilde{\rho}_4$	ρ_1^∞	ρ_2^∞	ρ_3^∞	ρ_4^∞	ρ^*	$\tilde{\rho}$
2	0.25	0.25	0.25	0.25	0.08333	0.08333	0.08333	0.8333	0.2	0	0	0	0.25	0.0833
3	0.6298	0.4825	0.4471	0.4335	0.1734	0.1133	0.0954	0.0873	0.6297	0.1699	0	0	0.4329	0.0668
4	1.0653	0.6185	0.6020	0.5880	0.2585	0.1299	0.0903	0.0718	1.0653	0.4522	0.2201	0	0.5562	0.0328
5	2.3097	0.9928	0.7772	0.7143	0.4230	0.1297	0.0789	0.0588	2.3097	0.6291	0.3717	0.0998	0.5820	0.0219
6	5.1065	1.5794	1.0022	0.8298	0.3040	0.0797	0.0604	0.0478	3.1391	1.3276	0.5155	0.2458	0.5434	0.0178

In Table 8 we list the computed optimal auxiliary nodes for $s = 2, \ldots, 6$, along with the corresponding diagonal entry d_s, with 36 significant digits: one may see that the auxiliary nodes are all distinct and in the interval $[0, 1]$.

Table 9 shows the convergence factors for the iteration (5.53), where the infinite norm has been used for the computation of (5.55). As one can see, in order to obtain an A-convergent iteration, one inner iteration is sufficient for the case $s = 2$ and $s = 3$, while two inner iterations are needed in the cases $s = 4$ and $s = 5$. Finally, at least four inner iterations are needed to obtain an A-convergent iteration, in the case $s = 6$.

For sake of completeness we also mention that actually, in order to reduce the computational cost of the procedure, one can solve the following system obtained by multiplying both sides of (5.53) by $h^{-2}\tilde{L}^{-1} \otimes I$:

$$\left[\frac{1}{h^2 d_s} I + I \otimes \nabla^2 U(q_0)\right] \tilde{\Delta}^{\ell,r+1} = \frac{1}{h^2}(S \otimes I)\tilde{\Delta}^{\ell,r+1} - C \otimes \nabla^2 U(q_0)\tilde{\Delta}^{\ell,r} + R^\ell, \quad r = 0, 1, \ldots,$$

with

$$S = d_s^{-1} I - \tilde{L}^{-1}$$

strictly lower triangular and

$$C = \tilde{U} - I, \qquad R^\ell = \frac{1}{h^2}(\tilde{L}^{-1} \otimes I)\boldsymbol{\eta}^\ell.$$

The remaining details are then similar to those explained in Section 5.7.

5.9 Numerical Tests

We end this section by showing a couple of numerical examples aimed to put into evidence the features and effectiveness of the methods. For both problems, we list the computational cost for HBVM(k,s) methods, in terms of required iterations for solving the generated discrete problems with a constant stepsize, when using:

(i) the fixed-point iteration;
(ii) the blended iteration described in Table 3;
(iii) the triangular splitting iteration described in Table 7, by using 2 inner iterations.

We choose 2 inner iterations for the triangular splitting iteration in (iii), so that the cost of one outer iteration is comparable to that of one blended iteration in (ii). We stress that, for all the three above iterations, the total number of functional evaluations equals the number

Table 10 Results when solving Problem (5.56)-(5.57) by using the HBVM$(k,2)$ method with stepsize $h = 0.1$ over the interval $[0, 10^3]$.

k	Hamiltonian error	solution error	fixed-point iterations	blended iterations	splitting iterations
2	$1.6 \cdot 10^{-3}$	$9.97 \cdot 10^{-2}$	79511	66854	48030
4	$8.3 \cdot 10^{-6}$	$1.82 \cdot 10^{-2}$	79846	66884	48252
6	$5.9 \cdot 10^{-9}$	$1.81 \cdot 10^{-2}$	79911	66941	48349
8	$1.7 \cdot 10^{-12}$	$1.81 \cdot 10^{-2}$	79939	66963	48377
10	$4.4 \cdot 10^{-16}$	$1.81 \cdot 10^{-2}$	79962	66976	48402

of iterations times k. Moreover, for the last two iterations, at each step one also needs to evaluate the Jacobian J_0, as well as to factor a matrix having the same size as that of the continuous problem (i.e., (5.28) for (ii) and (5.44) for (iii)).

The first problem is a nonlinear Hamiltonian problem describing the motion of a charged particle, with charge e and mass m, in a magnetic field with Biot-Savart potential [15]. It is defined by the Hamiltonian:

$$H(x,y,z,x',y',z') = \frac{1}{2m}\left[\left(x' - \alpha\frac{x}{\rho^2}\right)^2 + \left(y' - \alpha\frac{y}{\rho^2}\right)^2 + (z' + \alpha\log\rho)^2\right], \tag{5.56}$$

with $\rho = \sqrt{x^2+y^2}$ and $\alpha = eB_0$, B_0 being the intensity of the magnetic field. We have used the values

$$m = 1, \qquad e = -1, \qquad B_0 = 1,$$

and the initial values

$$x = 0.5, \qquad y = 10, \qquad x' = -0.1, \qquad y' = -0.3, \qquad z = z' = 0. \tag{5.57}$$

In Table 10 we list the results obtained by applying the HBVM$(k,2)$ methods, $k = 2,4,6,8,10$, for solving this problem over the interval $[0, 10^3]$ with stepsize $h = 0.1$. From the results in the table, one infers that:

- the Hamiltonian error monotonically decreases as k is increased and, for $k = 10$, one obtains a practical conservation, for the given stepsize (consequently larger values of k would be useless);
- the solution error when using the symplectic 2-stages Gauss method (i.e., HBVM(2,2)) is larger than that obtained when the energy error decreases;
- the triangular splitting procedure (iii) is more effective than the blended iteration (ii). In such a case, however, both iterations turn out to be not very competitive, with respect to the use of a fixed-point iteration, since this problem is not *stiff*;
- all iterations provide a total cost which is essentially independent of k.

As a second test problem we consider, on the contrary, a *stiff oscillatory* problem. It is defined as a slight modification of the Fermi-Pasta-Ulam problem described in [48].[7] The Hamiltonian defining this problem is given by:

[7] The original problem reported in [48] is obtained by setting $m = 3$ and $\omega_i = 50, i = 1,\dots,m$ in (5.58).

Table 11 Fixed-point iterations for solving problem (5.58)-(5.60) on the interval $[0, 10]$, by using HBVM$(6, 3)$ with stepsize h (*** means that the iteration doesn't converge)

h	fixed-point iterations
10^{-4}	2278912
$2 \cdot 10^{-4}$	1904534
$4 \cdot 10^{-4}$	4540389
$5 \cdot 10^{-4}$	***

Table 12 Hamiltonian error, obtained by using a sixth-order explicit composition method based on the Störmet-Verlet method, for solving problem (5.58)-(5.60) on the interval $[0, 10]$, by using stepsize h (*** means that the iteration doesn't converge)

h	Hamiltonian error
10^{-5}	$9.2 \cdot 10^{-8}$
$5 \cdot 10^{-5}$	$1.5 \cdot 10^{-3}$
10^{-4}	$8.5 \cdot 10^{-2}$
$2 \cdot 10^{-4}$	***
$4 \cdot 10^{-4}$	***
$5 \cdot 10^{-4}$	***

Table 13 Newton-type iterations for solving problem (5.58)-(5.60), on the interval $[0, 10]$ by using HBVM$(6, 3)$ with stepsize h.

h	blended iterations	splitting iterations
10^{-4}	1634792	856691
$5 \cdot 10^{-4}$	599927	299586
10^{-3}	241468	141506
$5 \cdot 10^{-3}$	29051	19148
10^{-2}	12721	8955
$5 \cdot 10^{-2}$	2369	1556
10^{-1}	1400	864
$5 \cdot 10^{-1}$	440	258

$$H(p,q) = \frac{1}{2}\sum_{i=1}^{m}(p_{2i-1}^2 + p_{2i}^2) + \frac{1}{4}\sum_{i=1}^{m}\omega_i^2(q_{2i} - q_{2i-1})^2 + \sum_{i=0}^{m}(q_{2i+1} - q_{2i})^4, \tag{5.58}$$

with $q, p \in \mathbb{R}^{2m}$ and $q_0 = q_{2m+1} = 0$. We choose $m = 7$, so that the problem has dimension 28, and

$$\omega_i = \omega_{m-i+1} = 10, \qquad i = 1, 2, 3, \qquad \text{and} \qquad \omega_4 = 10^4. \tag{5.59}$$

The starting vector is

$$p_i = 0, \qquad q_i = \frac{i-1}{2m-1}, \qquad i = 1, \ldots, 2m. \tag{5.60}$$

In such a case, the Hamiltonian function is a polynomial of degree 4, so that the HBVM$(2s, s)$ method (having order $2s$), is able to exactly preserve the Hamiltonian. As

an example, we fix $s = 3$ and integrate the problem on the interval $[0, 10]$. In this case, the fixed-point iteration cannot be expected to work, when using stepsizes much larger than $\|\omega\|_\infty^{-1} = 10^{-4}$, as is confirmed by the results in Table 11. Similarly, explicit methods, which exist in this specific case since the problem is separable (see [63, Chapter 8]), suffer from a similar restriction on the stepsize because of stability reasons. In particular we consider a composition method, having order 6, based on the Störmer-Verlet method (see [48, Chapter II.4] for details), requiring 18 function evaluations per step:[8] the results li sted in Table 12 clearly confirm this fact.

Conversely, the use of Newton-type iterations for solving the discrete problems generated by the HBVM(6,3) method, permits to use much larger stepsizes, thus allowing to approximate the low frequencies without being hindered by the high ones. By using the blended iteration (ii) and the triangular splitting iteration (iii), one obtains the figures in Table 13. Even when using very coarse stepsizes, the approximation of the slowly-oscillating components of the solution (24 out of 28) is satisfactory: as an example in Figure 22 and Figure 23 there is the plot of the slowly-oscillating components q_{11} and p_{11}, respectively, by using a finer step, $h = 10^{-4}$, and a much coarser one, $h = 0.5$.[9] Last but not least, from the figures in Table 13, one sees that the tr iangular splitting procedure (iii) is the most effective one, though using only 2 inners iterations.

6 Further developments

The line-integral approach which HBVMs rely on, has been generalized to devise methods able to preserve multiple invariants of motion. This has resulted in the *Line Integral Methods* (LIMs) [9] and the *Enhanced Line Integral Methods* (ELIMs) [31]. Both classes of methods are able to preserve any number of independent invariants of a general conservative problem, even though the latter methods appears to be more favorable, when the problem at hand is Hamiltonian.

6.1 Line Integral Methods

Let consider the problem

$$y'(ch) = f(y(ch)) \equiv \sum_{j\ge 0} \gamma_j(y) P_j(c), \qquad c \in [0,1], \qquad y(0) = y_0 \in \mathbb{R}^m, \tag{6.1}$$

$$\gamma_j(y) = \int_0^1 P_j(\tau) f(y(\tau h)) \mathrm{d}\tau, \qquad j \ge 0, \tag{6.2}$$

assuming that (6.1) is a general *conservative* (not necessarily Hamiltonian) problem. For sake of simplicity, we discuss the case of two (functionally independent) invariants that we denote by $H(y)$ and $L(y)$, though the arguments can be straightforwardly extended to any number of invariants [9]. The basic idea is that of introducing the polynomial approximation (4.5) with a perturbed coefficient, in order to obtain *both*

[8] Consequently, each step of this composition method has a cost which is comparable to 3 fixed-point iterations for HBVM$(6, 3)$.

[9] By the way, we mention that also the *amplitude* of the remaining 4 highly-oscillatory components turns out to be well approximated, when using a stepsize $h = 0.1$.

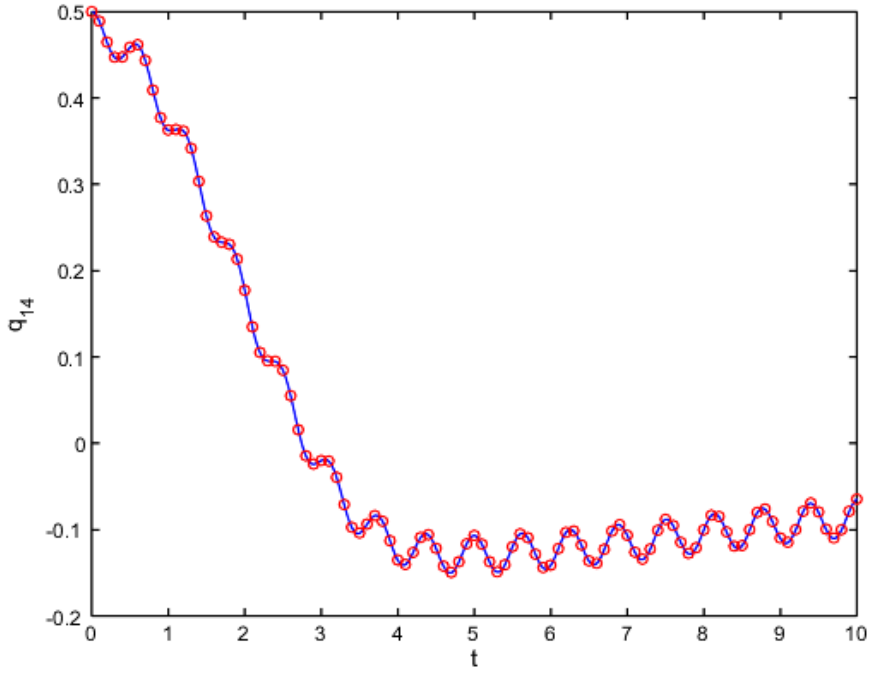

Fig. 22 Numerical approximation obtained by using HBVM(6,3) with stepsizes $h = 10^{-4}$ (continuous line) and $h = 10^{-1}$ (circles).

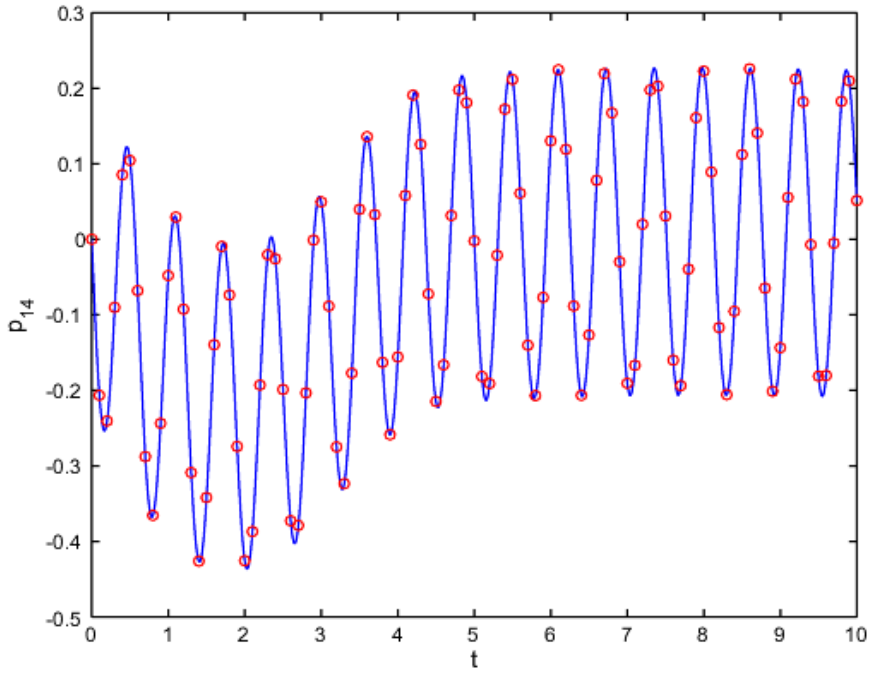

Fig. 23 Numerical approximation obtained by using HBVM(6,3) with stepsizes $h = 10^{-4}$ (continuous line) and $h = 10^{-1}$ (circles).

$$L(\sigma(h)) = L(y_0) \quad \text{and} \quad H(\sigma(h)) = H(y_0). \tag{6.3}$$

We define, besides to (6.2), the coefficients of the expansions of ∇H and ∇L along the basis $\{P_j\}$:

$$\rho_j(y) = \int_0^1 P_j(\tau)\nabla L(y(\tau h))\mathrm{d}\tau, \qquad j \geq 0, \tag{6.4}$$

$$\varphi_j(y) = \int_0^1 P_j(\tau)\nabla H(y(\tau h))\mathrm{d}\tau, \qquad j \geq 0. \tag{6.5}$$

The perturbation of the polynomial approximation to $y(ch)$ will be done by perturbing the first coefficient of the expansion in (4.5),[10] so that we obtain the new expansion (we continue to denote by σ the resulting polynomial)

$$\sigma'(ch) = \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma) + P_0(c)\left[x_1\rho_0(\sigma) + x_2\varphi_0(\sigma)\right], \quad c \in [0,1], \tag{6.6}$$

in place of (4.5). Actually, also the scalar coefficients x_1 and x_2 in (6.6) do depend on σ: they will be determined in order to fulfill the conservation conditions (6.3), which reads

$$\begin{aligned} L(\sigma(h)) - L(y_0) &= h\int_0^1 \nabla L(\sigma(\tau h))^T\sigma'(\tau h)\mathrm{d}\tau \\ &= h\int_0^1 \nabla L(\sigma(\tau h))^T\left[\sum_{j=0}^{s-1} P_j(\tau)\gamma_j + x_1P_0(\tau)\rho_0 + x_2P_0(\tau)\varphi_0\right]\mathrm{d}\tau \\ &= \sum_{j=0}^{s-1}\rho_j^T\gamma_j + x_1\rho_0^T\rho_0 + x_2\rho_0^T\varphi_0 = 0, \end{aligned} \tag{6.7}$$

and, similarly,

$$H(\sigma(h)) - H(y_0) = \sum_{j=0}^{s-1}\varphi_j^T\gamma_j + x_1\varphi_0^T\rho_0 + x_2\varphi_0^T\varphi_0 = 0. \tag{6.8}$$

Let define the matrix $A \equiv (\rho_0, \varphi_0) \in \mathbb{R}^{m\times 2}$.[11] Consequently, (6.7) and (6.8) can be recast as the system

$$A^TA\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -\begin{pmatrix} \sum_{j=0}^{s-1}\rho_j^T\gamma_j \\ \sum_{j=0}^{s-1}\varphi_j^T\gamma_j \end{pmatrix} \equiv \begin{pmatrix} O(h^{2s}) \\ O(h^{2s}) \end{pmatrix} \tag{6.9}$$

where the last equality[12] comes from Lemma 2.

We assume that matrix A has full rank, so that A^TA is symmetric and positive definite, and from (6.9) we then obtain

$$x_1 = O(h^{2s}), \qquad x_2 = O(h^{2s}),$$

so that in (6.6) the perturbation of the $O(h^0)$ coefficient $\gamma_0(\sigma)$ is $O(h^{2s})$ and, therefore, very small, as $h \to 0$.[13]

[10] The choice, indeed, is the most recommendable, as is shown in the sequel.

[11] In the case of ℓ invariants A would have ℓ columns given by the Fourier coefficients of the corresponding gradients.

[12] For Hamiltonian problems, the first entry of the vector at the right-hand side of (6.9) would be 0.

[13] In general, by perturbing the j-th coefficient would introduce a $O(h^{2(s-j)})$ perturbation. Consequently the choice of perturbing γ_0, provides the smallest possible perturbation, both absolute and relative, as $h \to 0$.

Theorem 14. Under the assumption that matrix A has full rank, $y(h) - \sigma(h) = O(h^{2s+1})$, for all $k \geq s$.

Proof. The proof strictly follows that of Theorem 4. □

In order to obtain an actual numerical methods, we need to approximate the integrals defining the coefficients appearing in (6.6) (see (6.2), (6.4), and (6.5)). Even though, in principle, they could be approximated by means of different quadrature formulae, for sake of simplicity we assume to use:

- a Gauss-Legendre formula based at the k abscissae $0 < c_1 < \ldots < c_k < 1$ and corresponding weights $\{b_i\}$ for approximating $\gamma_j(\sigma)$:

$$\gamma_j(\sigma) = \sum_{i=1}^{k} b_i P_j(c_i) f(\sigma(c_i h)) + \Delta_j(h) \equiv \hat{\gamma}_j(\sigma) + \Delta_j(h), \quad j = 0, \ldots, s-1, \tag{6.10}$$

  with

$$\Delta_j(h) = O(h^{2k-j}), \quad j = 0, \ldots, s-1;$$

- a Gauss-Legendre formula based at the r abscissae $0 < \tau_1 < \ldots < \tau_r < 1$ and corresponding weights $\{\beta_\ell\}$ for approximating $\rho_j(\sigma)$ and $\varphi_j(\sigma)$:

$$\begin{aligned} \rho_j(\sigma) &= \sum_{\ell=1}^{r} \beta_\ell P_j(\tau_\ell) \nabla L(\sigma(\tau_\ell h)) + \Phi_j(h) \equiv \hat{\rho}_j(\sigma) + \Phi_j(h), \quad j = 0, \ldots, s-1, \\ \varphi_j(\sigma) &= \sum_{\ell=1}^{r} \beta_\ell P_j(\tau_\ell) \nabla H(\sigma(\tau_\ell h)) + \Psi_j(h) \equiv \hat{\varphi}_j(\sigma) + \Psi_j(h), \ j = 0, \ldots, s-1, \end{aligned} \tag{6.11}$$

  with

$$\Phi_j(h) = O(h^{2r-j}) \quad \text{and} \quad \Psi_j(h) = O(h^{2r-j}), \qquad j = 0, \ldots, s-1.$$

Definition 2. We shall refer to such a method as *LIM*(r, k, s), where LIM is the acronym for *Line Integral Method*.

Remark 14. We observe that:

- LIM$(0, s, s)$ is the s-stage Gauss method,
- LIM$(0, k, s)$ is the HBVM(k, s) method,

where $r = 0$ means that no invariant conservation is sought.

The polynomial σ is obviously formally replaced by the polynomial $u \in \Pi_s$ such that

$$u'(ch) = \sum_{j=0}^{s-1} \hat{\gamma}_j(u) P_j(c) + x_1 \hat{\rho}_0(u) + x_2 \hat{\varphi}_0(u), \qquad c \in [0, 1], \qquad u(0) = y_0,$$

where we have taken into account that $P_0 \equiv 1$ and where $\hat{\gamma}_j(u), \hat{\rho}_0(u), \hat{\varphi}_0(u)$ are obtained by formally replacing σ with u in (6.10) and in (6.11), respectively.

Defining the matrix $\hat{A} \equiv (\hat{\rho}_0, \hat{\varphi}_0)$ one has

$$\hat{A}^T\hat{A}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = -\begin{pmatrix} \sum_{j=0}^{s-1} \hat{\gamma}_j^T \hat{\rho}_j \\ \sum_{j=0}^{s-1} \hat{\gamma}_j^T \hat{\varphi}_j \end{pmatrix} = -\begin{pmatrix} \sum_{j=0}^{s-1} (\gamma_j - \Delta_j)^T (\rho_j - \Phi_j) \\ \sum_{j=0}^{s-1} (\gamma_j - \Delta_j)^T (\varphi_j - \Psi_j) \end{pmatrix},$$

so that, by considering that the assumption that A has full rank holds true also for $\hat{A}$,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \equiv \begin{pmatrix} O(h^{2s}) + O(h^{2r}) + O(h^{2k}) + O(h^{2(r+k-s+1)}) \\ O(h^{2s}) + O(h^{2r}) + O(h^{2k}) + O(h^{2(r+k-s+1)}) \end{pmatrix} \equiv \begin{pmatrix} O(h^{2s}) \\ O(h^{2s}) \end{pmatrix},$$

where the last equality holds true for all $r \geq s$ and $k \geq s$.

To conclude, under the assumption that A has full rank, we just mention the following theorems whose proofs are similar to the corresponding ones described in the previous sections for HBVMs (see [9] for details).

Theorem 15. For all $r, k \geq s$, LIM(r, k, s) provides an approximation $u(h)$ such that:

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

Corollary 4. For all $r, k \geq s$, LIM(r, k, s) has order $2s$.

Concerning the conservation of the invariants, the following result holds true.

Theorem 16. For given $r, k \geq s$, LIM(r, k, s) exactly conserves the invariant L and H if they are polynomials of degree

$$\nu \leq \frac{2r}{s}.$$

In the non polynomial case, one obtains, provided that L and H are suitably regular,

$$L(u(h)) - L(y_0) = O(h^{2r+1}), \qquad H(u(h)) - H(y_0) = O(h^{2r+1}). \tag{6.12}$$

Remark 15. From (6.12), one obtains that, for any suitably regular set of invariants, conservation can always be practically gained, provided that r and k are large enough. Indeed, it is enough to obtain conservation up to roundoff errors.

The following result can be also proved, by using arguments similar to those used in Section 4.5.

Theorem 17. Provided that the nodes of the quadrature formulae (6.10) and (6.11) are symmetrically distributed in the interval $[0, 1]$, LIM(r, k, s) is a symmetric method.

6.2 Enhanced Line Integral Methods

We now introduce a different multiple-invariants conserving class of methods. This approach is more tailored for Hamiltonian problems. Let then consider the problem

$$y' = J\nabla H(y), \qquad y(0) = y_0 \in \mathbb{R}^{2m} \tag{6.13}$$

and assume that it admits a set of ν functionally independent smooth invariants,

$$L : \mathbb{R}^{2m} \to \mathbb{R}^{\nu}, \tag{6.14}$$

besides the Hamiltonian H. Consequently, one has

$$\nabla L(y)^T J \nabla H(y) = 0 \in \mathbb{R}^\nu, \qquad \forall y, \tag{6.15}$$

where $\nabla L(y)^T$ is the Jacobian matrix of L.

Following the same approach seen for HBVMs in Section 4, we define our polynomial approximation $\sigma \in \Pi_s$ (where $s > \nu$) as in (2.7)–(2.10),

$$\sigma'(ch) = \sum_{j=0}^{s-1} P_j(c)\eta_j \bar\gamma_j(\sigma), \qquad c \in [0,1], \tag{6.16}$$

with the vector coefficients $\{\bar\gamma_j(\sigma)\}$ given by

$$\bar\gamma_j(\sigma) = \int_0^1 P_j(\tau) J \nabla H(\sigma(\tau h)) \mathrm{d}\tau \in \mathbb{R}^{2m}, \qquad j = 0, \dots, s-1. \tag{6.17}$$

Consequently, energy conservation is assured from (2.9). The main difference, with respect to HBVMs, for which $\eta_j = 1$, $j = 0, \dots, s-1$, consists in looking for scalar coefficients $\{\eta_j\}$ in the form

$$\begin{aligned} &\eta_j = 1, \qquad j = 0, \dots, s-\nu-1, \\ &\eta_j = \left[1 - h^{2(s-1-j)}\alpha_j\right], \qquad j = s-\nu, \dots, s-1, \end{aligned} \tag{6.18}$$

with the coefficients $\{\alpha_j\}$ determined in order to obtain the conservation of the ν additional invariants (6.14)–(6.15). By setting the new approximation

$$y_1 \equiv \sigma(h) \approx y(h),$$

from (6.16)–(6.18) one obtains, by requiring conservation of all invariants through a line-integral:

$$\begin{aligned} 0 = L(y_1) - L(y_0) = L(\sigma(h)) - L(\sigma(0)) &= \int_0^h \nabla L(\sigma(t))^T \sigma'(t) \mathrm{d}t \\ = h\int_0^1 \nabla L(\sigma(\tau h))^T \sigma'(\tau h)\mathrm{d}\tau &= h\sum_{j=0}^{s-1} \eta_j \left[\int_0^1 P_j(\tau)\nabla L(\sigma(\tau h))\mathrm{d}\tau\right]^T \bar\gamma_j(\sigma)\mathrm{d}\tau \\ \equiv h\left[\sum_{j=0}^{s-1} \phi_j(\sigma)^T\bar\gamma_j(\sigma) - \sum_{j=s-\nu}^{s-1} h^{2(s-1-j)}\alpha_j\phi_j(\sigma)^T\bar\gamma_j(\sigma)\right], \end{aligned}$$

where, for all $j \ge 0$:

$$\phi_j(\sigma) = \int_0^1 P_j(\tau)\nabla L(\sigma(\tau h))\mathrm{d}\tau \in \mathbb{R}^{2m\times\nu}. \tag{6.19}$$

Consequently, energy conservation is "for free", unless the case of LIMs, and the conservation of the invariants is gained provided that

$$\sum_{j=s-\nu}^{s-1} h^{2(s-1-j)}\alpha_j\phi_j(\sigma)^T\bar\gamma_j(\sigma) = \sum_{j=0}^{s-1}\phi_j(\sigma)^T\bar\gamma_j(\sigma). \tag{6.20}$$

By defining the matrix

$$\Gamma(\sigma) = \left[h^{2(\nu-1)}\phi_{s-\nu}(\sigma)^T\bar\gamma_{s-\nu}(\sigma), \ldots, h^0\phi_{s-1}(\sigma)^T\bar\gamma_{s-1}(\sigma)\right] \in \mathbb{R}^{\nu\times\nu} \tag{6.21}$$

and the vectors

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_{s-\nu} \\ \vdots \\ \alpha_{s-1} \end{pmatrix}, \quad \boldsymbol{b}(\sigma) = \sum_{j=0}^{s-1} \phi_j(\sigma)^T\bar\gamma_j(\sigma) \quad \in \mathbb{R}^\nu,$$

equation (6.20) can be recast in vector form as

$$\Gamma(\sigma)\boldsymbol{\alpha} = \boldsymbol{b}(\sigma). \tag{6.22}$$

It easily follows from (6.17), (6.19), and Lemma 2, that matrix $\Gamma(\sigma)$ in (6.21) has $O(h^{2s-2})$ entries and, taking into account (6.15), that $\boldsymbol{b}(\sigma) = O(h^{2s})$. Therefore, assuming that matrix $\Gamma(\sigma)$ is nonsingular and that $s > \nu$, it follows from (6.22) that the vector $\boldsymbol{\alpha}$ has $O(h^2)$ entries. We stress that it is necessary that $s > \nu$, since when $s = \nu$, from (6.20) one obtains that $h^{2(s-1-j)}\alpha_j = 1$, $j = 0, \ldots, s-1$ and then, from (6.18), it follows that $\eta_j = 0$, $j = 0, \ldots, s-1$. Consequently, in the sequel we shall always assume that $s > \nu$, so that:

$$\eta_0 = 1. \tag{6.23}$$

From the definition of the method one has that all the invariants are conserved (including the Hamiltonian) and, with similar steps as in Theorem 4, one obtains:

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

That is, the method has order $2s$.

However, in order to obtain an effective numerical method, we now proceed to approximate the integrals in (6.17) and (6.19) by means of suitable quadrature formulae. For sake of brevity, as in the case of LIM(r, k, s) methods illustrated in Section 6.1, we assume to use:

- a Gauss-Legendre formula based at the k abscissae $0 < c_1 < \ldots < c_k < 1$, and corresponding weights $\{b_i\}$, for approximating $\bar\gamma_j(\sigma)$:

$$\bar\gamma_j(\sigma) = \sum_{i=1}^{k} b_i P_j(c_i) J\nabla H(\sigma(c_i h)) + \Delta_j(h) \equiv \hat\gamma_j(\sigma) + \Delta_j(h), \quad j = 0, \ldots, s-1, \tag{6.24}$$

  with

$$\Delta_j(h) = \begin{cases} 0, & \text{if } H \in \Pi_{\mu(k)}, \\ O(h^{2k-j}), & \text{otherwise;} \end{cases}$$

- a Gauss-Legendre formula based at the r abscissae $0 < \tau_1 < \ldots < \tau_r < 1$, and corresponding weights $\{\beta_\ell\}$, for approximating $\phi_j(\sigma)$:[14]

$$\phi_j(\sigma) = \sum_{\ell=1}^{r} \beta_\ell P_j(\tau_\ell)\nabla L(\sigma(\tau_\ell h)) + \Psi_j(h) \equiv \hat\phi_j(\sigma) + \Psi_j(h), \quad j = 0, \ldots, s-1, \tag{6.25}$$

  with

$$\Psi_j(h) = \begin{cases} 0, & \text{if } L \in \Pi_{\mu(r)}, \\ O(h^{2r-j}), & \text{otherwise;} \end{cases}$$

[14] In principle, for any invariant in (6.19) one could use a different quadrature formula, depending on the required accuracy.

where we have assumed L and H to be suitably regular, and denoted

$$\mu(j) = \left\lfloor \frac{2j}{s} \right\rfloor, \qquad j \in \{r, k\}. \tag{6.26}$$

Consequently, the polynomial σ is formally replaced by the polynomial $u \in \Pi_s$ given by

$$\begin{aligned} u(ch) &= y_0 + h \sum_{j=0}^{s-1} \int_0^c P_j(x) \mathrm{d}x \, \hat{\eta}_j \hat{\gamma}_j(u) \qquad (6.27) \\ &\equiv y_0 + h \left[\sum_{j=0}^{s-1} \int_0^c P_j(x) \mathrm{d}x \hat{\gamma}_j(u) - \sum_{j=s-\nu}^{s-1} \int_0^c P_j(x) \mathrm{d}x h^{2(s-1-j)} \hat{\alpha}_j \hat{\gamma}_j(u) \right], \end{aligned}$$

where $\hat{\eta}_j$ denote the discrete approximation to (6.18), the scalars $\hat{\alpha}_j$ satisfying the equation (compare with (6.20)):

$$\sum_{j=s-\nu}^{s-1} h^{2(s-1-j)} \hat{\alpha}_j \hat{\phi}_j(u)^T \hat{\gamma}_j(u) = \sum_{j=0}^{s-1} \hat{\phi}_j(u)^T \hat{\gamma}_j(u), \tag{6.28}$$

and $\hat{\gamma}_j(u)$ and $\hat{\phi}_j(u)$ obtained by formally replacing σ with u in (6.24) and in (6.25), respectively. Similarly as previously done in (6.21)-(6.22), by defining the matrix

$$\hat{\Gamma} = \left[h^{2(\nu-1)} \hat{\phi}_{s-\nu}^T(u) \hat{\gamma}_{s-\nu}(u), \, \dots \, , h^0 \hat{\phi}_{s-1}^T(u) \hat{\gamma}_{s-1}(u) \right] \in \mathbb{R}^{\nu \times \nu}$$

and the vectors

$$\hat{\boldsymbol{\alpha}} = \begin{pmatrix} \hat{\alpha}_{s-\nu} \\ \vdots \\ \hat{\alpha}_{s-1} \end{pmatrix}, \quad \hat{\boldsymbol{b}}(u) = \sum_{j=0}^{s-1} \hat{\phi}_j(u)^T \hat{\gamma}_j(u) \quad \in \mathbb{R}^{\nu},$$

equation (6.28) can be recast in vector form as

$$\hat{\Gamma} \hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{b}}.$$

Theorem 18. Under the assumption that matrix Γ is nonsingular, for all $r, k \geq s$ and for all sufficiently small stepsizes h, matrix $\hat{\Gamma}$ is nonsingular and the vector $\hat{\boldsymbol{\alpha}}$ has $O(h^2)$ entries.

Since (6.23) holds true, similarly as in the case of HBVM(k, s) methods, the new approximation is given by

$$y_1 \equiv u(h) = y_0 + h\hat{\gamma}_0 = y_0 + h \sum_{i=1}^{k} b_i J \nabla H(u(c_i h)). \tag{6.29}$$

Definition 3. We shall refer to the methods defined by (6.27)–(6.29) as *ELIMs*(r, k, s) (*Enhanced LIMs*(r, k, s)). In particular when $r = k$ we speak about an *EHBVM*(k, s) (*Enhanced HBVM*(k, s)) method.

With similar steps as in Section 4.4, the following results can be proved [31].

Theorem 19. Let $k, r \geq s$. If the Hamiltonian function defining problem (6.13) is a polynomial of degree less than or equal to $\mu(k)$ as defined in (6.26), the ELIM(r, k, s) method is energy conserving. Differently, for all general and suitably regular Hamiltonians, one obtains:

$$H(y_1) - H(y_0) = O(h^{2k+1}), \qquad \forall k \geq s.$$

Theorem 20. Let $k, r \geq s$. If the invariants (6.14)-(6.15) of problem (6.13) are polynomials of degree less than or equal to $\mu(r)$ as defined in (6.26), the ELIM(r, k, s) method is invariants-conserving. Differently, for all general and suitably regular L, one obtains:

$$L(y_1) - L(y_0) = O(h^{2r+1}), \qquad \forall r \geq s.$$

Moreover, with similar steps as in Theorem 5 one can prove the following result (see [31]).

Theorem 21. Let $k, r \geq s$. Assuming that both H and L are suitably regular, the numerical solution generated by a ELIM(r, k, s) method satisfies

$$y_1 - y(h) = O(h^{2s+1}).$$

That is, the method has order $2s$.

The following result can be also proved, by using arguments similar to those used in Section 4.5 (see, e.g., [15, 13]).

Theorem 22. Let the nodes of the quadrature formulae (6.24) and (6.25) be symmetrically distributed in the interval $[0, 1]$. Then, the ELIM(r, k, s) method is symmetric.

We finally observe that, since the only formal difference between a classical HBVM(k, s) method and an ELIM(r, k, s) method consists in the coefficients $\hat{\eta}_1, \ldots, \hat{\eta}_{s-1}$ which may assume values different from 1, an ELIM(r, k, s) method admits the Runge-Kutta type formulation (2.16) with

$$\Lambda_s = \mathrm{diag}(1, \hat{\eta}_1, \ldots, \hat{\eta}_{s-1}) \qquad \text{and} \qquad \mathcal{I}_s^1 \Lambda_s \mathcal{P}_s^T \Omega = \mathbf{b}^T,$$

where we have taken into account account of (6.23).

6.3 Further developments and references

It is worth mentioning that further additional developments, such as the possibility of getting, in a weakened sense, methods which are *both* symplectic and energy-conserving, have been considered in [25, 20]. We also mention that a noticeable extension of this approach, for PRK methods, has been recently devised in [69]. A further line of investigation deals with multistep energy-preserving method, as is sketched in [24].

References

[1] P. Amodio, L. Brugnano. A Note on the Efficient Implementation of Implicit Methods for ODEs. *J. Comput. Appl. Math.* **87** (1997) 1–9.

[2] G. Benettin, A. Giorgilli. On the Hamiltonian interpolation of near to the identity symplectic mappings with application to symplectic integration algorithms. *J. Statist. Phys.* **74** (1994) 1117–1143.

[3] P. Betsch, P. Steinmann. Inherently Energy Conserving Time Finite Elements for Classical Mechanics. *Journal of Computational Physics* **160** (2000) 88–116.

[4] C.L. Bottasso. A new look at finite elements in time: a variational interpretation of Runge–Kutta methods. *Applied Numerical Mathematics* **25** (1997) 355–368.

[5] L. Brugnano. Blended Block BVMs (B3VMs): A Family of Economical Implicit Methods for ODEs. *Journal of Computational and Applied Mathematics* **116** (2000) 41–62.

[6] L. Brugnano, M. Calvo, J.I. Montijano, L. Ràndez. Energy preserving methods for Poisson systems. *Journal of Computational and Applied Mathematics* **236** (2012) 3890–3904.

[7] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Efficient implementation of Gauss collocation and Hamiltonian Boundary Value Methods. *Numer. Algor.* DOI:10.1007/s11075-014-9825-0

[8] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Efficient implementation of geometric integrators for separable Hamiltonian problems. *AIP Conference Proceedings* **1558**, 734 (2013).

[9] L. Brugnano, F. Iavernaro. Line Integral Methods which preserve all invariants of conservative problems. *Journal of Computational and Applied Mathematics* **236** (2012) 3905–3919.

[10] L. Brugnano, F. Iavernaro. Recent Advances in the Numerical Solution of Conservative Problems. *AIP Conference Proc.* **1493** (2012) 175–182.

[11] L. Brugnano, F. Iavernaro. Geometric Integration by Playing with Matrices. *AIP Conference Proceedings* **1479** (2012) 16–19.

[12] L. Brugnano, F. Iavernaro, C. Magherini. Efficient implementation of Radau collocation methods. (submitted for publication) 2012, arXiv:1302.1037

[13] L. Brugnano, F. Iavernaro. Line integral methods and their application to the numerical solution of conservative problems arXiv:1301.2367

[14] L. Brugnano, F. Iavernaro, T. Susca. Numerical comparisons between Gauss-Legendre methods and Hamiltonian BVMs defined over Gauss points. *Monografias de la Real Acedemia de Ciencias de Zaragoza* **33** (2010) 95–112.

[15] L. Brugnano, F. Iavernaro, D. Trigiante. Analysis of Hamiltonian Boundary Value Methods (HBVMs) for the numerical solution of polynomial Hamiltonian dynamical systems. (2009) arXiv:0909.5659v1

[16] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian BVMs (HBVMs): a family of "drift-free" methods for integrating polynomial Hamiltonian systems. *AIP Conf. Proc.* **1168** (2009) 715–718.

[17] L. Brugnano, F. Iavernaro, D. Trigiante. *The Hamiltonian BVMs (HBVMs) Homepage*, 2010. arXiv:1002.2757

[18] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian Boundary Value Methods (Energy Preserving Discrete Line Methods). *Journal of Numerical Analysis, Industrial and Applied Mathematics* **5**,1-2 (2010) 17–37.

[19] L. Brugnano, F. Iavernaro, D. Trigiante. Numerical Solution of ODEs and the Columbus' Egg: Three Simple Ideas for Three Difficult Problems. *Mathematics in Engineering, Science and Aerospace* **1**,4 (2010) 407–426.

[20] L. Brugnano, F. Iavernaro, D. Trigiante. Energy and quadratic invariants preserving integrators of Gaussian type. *AIP Conference Proceedings* **1281** (2010) 227–230.

[21] L. Brugnano, F. Iavernaro, D. Trigiante. A note on the efficient implementation of Hamiltonian BVMs. *Journal of Computational and Applied Mathematics* **236** (2011) 375–383.

[22] L. Brugnano, F. Iavernaro, D. Trigiante. The Lack of Continuity and the Role of Infinite and Infinitesimal in Numerical Methods for ODEs: the Case of Symplecticity. *Applied Mathematics and Computation* **218** (2012) 8053–8063.

[23] L. Brugnano, F. Iavernaro, D. Trigiante. A simple framework for the derivation and analysis of effective one-step methods for ODEs. *Applied Mathematics and Computation* **218** (2012) 8475–8485.

[24] L. Brugnano, F. Iavernaro, D. Trigiante. A two-step, fourth-order method with energy preserving properties. *Computer Physics Communications* **183** (2012) 1860–1868.

[25] L. Brugnano, F. Iavernaro, D. Trigiante. Energy and QUadratic Invariants Preserving integrators based upon Gauss collocation formulae. *SIAM Journal on Numerical Analysis* **50**, No. 6 (2012) 2897–2916.

[26] L. Brugnano, C. Magherini. Blended Implementation of Block Implicit Methods for ODEs. *Appl. Numer. Math.* **42** (2002) 29–45.

[27] L. Brugnano, C. Magherini. The BiM Code for the Numerical Solution of ODEs. *Jour. Comput. Appl. Mathematics* **164-165** (2004) 145–158.

[28] L. Brugnano, C. Magherini. Some Linear Algebra Issues Concerning the Implementation of Blended Implicit Methods. *Numer. Lin. Alg. Appl.* **12** (2005) 305–314.

[29] L. Brugnano, C. Magherini. Recent Advances in Linear Analysis of Convergence for Splittings for Solving ODE problems. *Applied Numerical Mathematics* **59** (2009) 542–557.

[30] L. Brugnano, C. Magherini, F. Mugnai. Blended Implicit Methods for the Numerical Solution of DAE Problems. *Jour. Comput. Appl. Mathematics* **189** (2006) 34–50.

[31] L. Brugnano, Y. Sun. Multiple invariants conserving Runge-Kutta type methods for Hamiltonian problems. *Numer. Algor.* DOI:10.1007/s11075-013-9769-9

[32] L. Brugnano, D. Trigiante. *Solving ODEs by Linear Multistep Initial and Boundary Value Methods*, Gordon and Breach, Amsterdam, 1998.

[33] K. Burrage, P.M. Burrage. Low rank Runge-Kutta methods, symplecticity and stochastic Hamiltonian problems with additive noise. *Journal of Computational and Applied Mathematics* **236** (2012) 3920–3930.

[34] K. Burrage, J.C. Butcher. Stability criteria for implicit Runge–Kutta methods. *SIAM Journal on Numerical Analysis* **16** (1979) 46–57.

[35] M. Calvo, M.P. Laburta, J.I. Montijano, L. Rández. Error growth in the numerical integration of periodic orbits, *Math. Comput. Simulation* **81** (2011) 2646–2661.

[36] E. Celledoni, R.I. McLachlan, D. Mc Laren, B. Owren, G.R.W. Quispel, W.M. Wright. Energy preserving Runge–Kutta methods. *M2AN* **43** (2009) 645–649.

[37] E. Celledoni, R.I. McLachlan, B. Owren, G.R.W. Quispel. Energy-Preserving Integrators and the Structure of B-series. *Found. Comput. Math.* **10** (2010) 673–693.

[38] P. Chartier, E. Faou, A. Murua. An algebraic approach to invariant preserving integrators: the case of quadratic and Hamiltonian invariants. *Numer. Math.* **103**, 4 (2006) 575–590.

[39] D. Cohen, E. Hairer. Linear energy-preserving integrators for Poisson systems. *BIT Numer. Math.* **51** (1) (2011) 91–101.

[40] M. Crouzeix. Sur la B-stabilité des méthodes de Runge–Kutta. *Numerische Mathematik* **32** (1979) 75–82.

[41] G. Dahlquist, A. Björk. *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

[42] Feng Kang. On Difference Schemes and Symplectic Geometry. In *Proceedings of the 1984 Beijing symposium on differential geometry and differential equations*. Science Press, Beijing, 1985, pp. 42–58.

[43] Z. Ge, J.E. Marsden. Lie-Poisson Hamilton-Jacobi theory and Lie-Poisson integrators. *Phys. Lett. A* **133** (1988) 134–139.

[44] H. Goldstein, C.P. Poole, J.L. Safko. *Classical Mechanics*. Addison Wesley, 2001.

[45] O. Gonzales. Time integration and discrete Hamiltonian systems. *J. Nonlinear Sci.* **6** (1996) 449–467.

[46] W. Gröbner. *Gruppi, Anelli e Algebre di Lie*. Collana di Informazione Scientifica "Poliedro", Edizioni Cremonese, Rome, 1975.

[47] E. Hairer. Energy preserving variant of collocation methods. *Journal of Numerical Analysis, Industrial and Applied Mathematics* **5**,1-2 (2010) 73–84.

[48] E. Hairer, C. Lubich, G. Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, Second ed., Springer, Berlin, 2006.

[49] E. Hairer, G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems, 2nd edition*. Springer-Verlag, Berlin, 1996.

[50] E. Hairer, C.J. Zbinden. On conjugate symplecticity of B-series integrators. *IMA J. Numer. Anal.* (2012) 1–23.

[51] P.J. van der Houwen, J.J.B. de Swart. Triangularly implicit iteration methods for ODE-IVP solvers. *SIAM J. Sci. Comput.* **18** (1997) 41–55.

[52] P.J. van der Houwen, J.J.B. de Swart. Parallel linear system solvers for Runge-Kutta methods. *Adv. Comput. Math.* **7**, 1-2 (1997) 157–181.

[53] B.L. Hulme. One-Step Piecewise Polynomial Galerkin Methods for Initial Value Problems. *Mathematics of Computation*, **26**, 118 (1972) 415–426.

[54] F. Iavernaro, B. Pace. *s*-Stage Trapezoidal Methods for the Conservation of Hamiltonian Functions of Polynomial Type. *AIP Conf. Proc.* **936** (2007) 603–606.

[55] F. Iavernaro, B. Pace. Conservative Block-Boundary Value Methods for the Solution of Polynomial Hamiltonian Systems. *AIP Conf. Proc.* **1048** (2008) 888–891.

[56] F. Iavernaro, D. Trigiante. High-order symmetric schemes for the energy conservation of polynomial Hamiltonian problems. *Journal of Numerical Analysis, Industrial and Applied Mathematics* **4**,1-2 (2009) 87–101.

[57] C. Kane, J.E. Marsden, M. Ortiz. Symplectic-energy-momentum preserving variational integrators. *Jour. Math. Phys.* **40**, 7 (1999) 3353–3371.

[58] V. Lakshmikantham, D. Trigiante. *Theory of Difference Equations. Numerical Methods and Applications.* Academic Press, 1988.

[59] R.I. Mc Lachlan, G.R.W. Quispel, N. Robidoux. Geometric integration using discrete gradient. *Phil. Trans. R. Soc. Lond. A* **357** (1999) 1021–1045.

[60] J.E. Marsden, J.M. Wendlandt. Mechanical Systems with Symmetry, Variational Principles, and Integration Algorithms. in *"Current and Future Directions in Applied Mathematics" M. Alber, B. Hu, and J. Rosenthal, Eds.*, Birkhäuser, 1997, pp. 219–261.

[61] G.R.W. Quispel, D.I. Mc Laren. A new class of energy-preserving numerical integration methods. *J. Phys. A: Math. Theor.* **41** (2008) 045206 (7pp).

[62] J.M. Sanz Serna. Runge-Kutta schemes for Hamiltonian systems. *BIT* **28** (1988) 877–883.

[63] J.M. Sanz Serna, M.P. Calvo. *Numerical Hamiltonian Problems.* Chapman & Hall, London, 1994.

[64] J.C. Simo, N. Tarnow. A new energy and momentum conserving algorithm for the non-linear dynamics of shells. *Internat. Jour. for Numerical Meth. in Engineering* **37** (1994) 2527–2549.

[65] J.C. Simo, N. Tarnow and K.K. Wong. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Computer Methods in Applied Mechanics and Engineering* **100** (1992) 63–116.

[66] Y.B. Suris. On the canonicity of mappings that can be generated by methods of Runge–Kutta type for integrating systems $x'' = \partial U/\partial x$. *U.S.S.R. Comput. Math. and Math. Phys.* **29**, 1 (1989) 138–144.

[67] Q. Tang, C.-m. Chen. Continuous finite element methods for Hamiltonian systems. *Applied Mathematics and Mechanics* **28**,8 (2007) 1071–1080.

[68] W. Tang, Y. Sun. Time finite element methods: a unified framework for numerical discretizations of ODEs. *Applied Mathematics and Computation* **219**, 4 (2012) 2158–2179.

[69] D. Wang, A. Xiao, X. Li. Parametric symplectic partitioned Runge-Kutta methods with energy-preserving properties for Hamiltonian systems. *Computer Physics Communications* **184** (2013) 303–310.