Numer Algor (2014) 65:633–650
DOI 10.1007/s11075-014-9825-0

ORIGINAL PAPER

Efficient implementation of Gauss collocation and Hamiltonian boundary value methods

Luigi Brugnano · Gianluca Frasca Caccia · Felice Iavernaro

Received: 3 April 2013 / Accepted: 3 January 2014 / Published online: 17 January 2014

Abstract In this paper we define an efficient implementation for the family of low-rank energy-conserving Runge-Kutta methods named Hamiltonian Boundary Value Methods (HBVMs), recently defined in the last years. The proposed implementation relies on the particular structure of the Butcher matrix defining such methods, for which we can derive an efficient splitting procedure. The very same procedure turns out to be automatically suited for the efficient implementation of Gauss-Legendre collocation methods, since these methods are a special instance of HBVMs. The linear convergence analysis of the splitting procedure exhibits excellent properties, which are confirmed by a few numerical tests.

Keywords Energy-conserving methods · Hamiltonian Boundary Value Methods · W-transformation · Gauss-Legendre collocation methods · Implicit Runge-Kutta methods · Splitting

Mathematics Subject Classifications (2010) 65P10 · 65L05 · 65L06 · 65L99

Warmly dedicated to celebrate the 80th birthday of John Butcher

L. Brugnano (✉) · G. Frasca Caccia
Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Viale Morgagni, 67/A, 50134 Firenze, Italy
e-mail: luigi.brugnano@unifi.it

G. Frasca Caccia
e-mail: frasca@math.unifi.it

F. Iavernaro
Dipartimento di Matematica, Università di Bari, Via Orabona, 4, 70125 Bari, Italy
e-mail: felice.iavernaro@uniba.it

1 Introduction

The efficient numerical solution of implicit Runge-Kutta methods has been the subject of many investigations in the last decades, starting from the seminal papers of Butcher [15, 16] (see also [17]). This aspect is even more relevant when dealing with *geometric* Runge-Kutta methods, that is, methods used in the framework of *Geometric Integration* where, usually, the discrete problems generated by the methods need to be solved to within full machine accuracy, in order not to waste the specific properties of the methods.

In more details, in this paper we shall deal with the numerical solution of Hamiltonian problems, namely problems in the form,

$$y' = J\nabla H(y), \qquad y(t_0) = y_0 \in \mathbb{R}^{2m}, \tag{1}$$

where

$$y = \begin{pmatrix} q \\ p \end{pmatrix}, \quad q, p \in \mathbb{R}^m, \qquad J = \begin{pmatrix} O & I_m \\ -I_m & O \end{pmatrix}, \tag{2}$$

$H(y)$ is the (scalar) *Hamiltonian* function defining the problem, and I_m the identity matrix of dimension m.[1] Due to the skew-symmetry of matrix J one has

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

so that

$$H(y(t)) = H(y_0), \qquad \forall\, t \geq t_0.$$

For isolated mechanical systems, the Hamiltonian has the physical meaning of the total energy of the system, so that often the Hamiltonian is referred to as the *energy*. Its conservation is, therefore, a significant feature for the discrete dynamical system induced by a numerical method for solving (1): methods having this property are usually called *energy-conserving methods*. Among such methods, we are interested in the class of energy-conserving methods named *Hamiltonian Boundary Value Methods (HBVMs)* [5, 8] (see also [6, 7], and [3, 9] for generalizations), which have been recently devised starting from the concept of *discrete line integrals*, defined in [26–28]. For such methods, the discrete problem can be conveniently posed in a suitable form which can be exploited to derive efficient implementation strategies, as was done in [6]. Here we further improve on such results, by proposing and analysing an iterative procedure based on the particular structure of the discrete problem. As a by-product, an efficient implementation of Gauss-Legendre collocation methods is also obtained. Indeed, these latter methods may be interpreted as a particular instance of HBVMs. The proposed procedure is strictly related to that recently devised in [4] for Radau IIA collocation formulae, though the two approaches are substantially different.

With this premise, the paper is organized as follows: in Section 2 we describe the structure of the discrete problem generated by HBVMs, along with the way of solving it, as done so far; in Section 3 we introduce the new iterative procedure, which is based on a suitable splitting; in Section 4 we study the convergence properties of

[1] In the following, when the size of the identity is not specified, it can be deduced from the context.

the new iteration, also comparing it with known existing ones; in Section 5 a few numerical tests are reported; at last, a few conclusions are contained in Section 6.

2 Discrete problem induced by HBVMs

We now recall the basic facts about HBVMs, and derive the most efficient formulation of the generated discrete problems. Let us assume, for sake of brevity, $t_0 = 0$ in (1), and consider the approximation of the problem over the interval $[0, h]$, which will clearly concern the very first application of a given numerical method. Let us then consider the orthonormal polynomial basis, on the interval $[0, 1]$, provided by the shifted and scaled Legendre polynomials $\{P_j\}$:

$$\deg P_i = i, \qquad \int_0^1 P_i(x)P_j(x)\mathrm{d}x = \delta_{ij}, \qquad \forall\, i, j \geq 0, \tag{3}$$

where δ_{ij} is the usual Kronecker symbol. Under suitable mild assumptions on the Hamiltonian function H, the right-hand side of the differential equation (1) can be expanded along the considered basis, thus giving

$$y'(ch) = \sum_{j\geq 0} \gamma_j(y) P_j(c), \qquad c \in [0, 1], \tag{4}$$

where

$$\gamma_j(y) = \int_0^1 J\nabla H(y(\tau h))P_j(\tau)\mathrm{d}\tau, \qquad j \geq 0. \tag{5}$$

By imposing the initial condition, the solution of this problem is formally obtained by

$$y(ch) = y_0 + h\sum_{j\geq 0} \gamma_j(y) \int_0^c P_j(x)\mathrm{d}x, \qquad c \in [0, 1]. \tag{6}$$

In order to derive a polynomial approximation σ of degree s to (6), we consider the following approximated ODE-IVPs:

$$\sigma'(ch) = \sum_{j=0}^{s-1} \gamma_j(\sigma) P_j(c), \qquad c \in [0, 1], \qquad \sigma(0) = y_0, \tag{7}$$

where $\gamma_j(\sigma)$ is defined according to (5), by formally replacing y by σ. Consequently, the approximation to (6) will be given by

$$\sigma(ch) = y_0 + h\sum_{j=0}^{s-1} \gamma_j(\sigma) \int_0^c P_j(x)\mathrm{d}x, \qquad c \in [0, 1]. \tag{8}$$

For sake of simplicity, assume now that the Hamiltonian function is a polynomial of degree ν (for the general case, see Theorem 1 below). Consequently, the (unknown) vector coefficients $\{\gamma_j(\sigma)\}$ can be exactly obtained by using a quadrature formula defined at the Gaussian abscissae $\{c_1, \ldots, c_k\}$, i.e.,

$$P_k(c_i) = 0, \qquad i = 1, \ldots, k, \tag{9}$$

and corresponding weights $\{b_1, \ldots, b_k\}$,[2]

$$\gamma_j(\sigma) = \sum_{i=1}^{k} b_i J \nabla H(\sigma(c_i h)) P_j(c_i), \qquad j = 0, \ldots, s-1, \tag{10}$$

provided that

$$\nu \leq \frac{2k}{s}. \tag{11}$$

By setting

$$Y_i = \sigma(c_i h), \qquad i = 1, \ldots, k, \tag{12}$$

and considering that the new approximation is given by

$$y_1 \equiv \sigma(h) = y_0 + h \int_0^1 J \nabla H(\sigma(\tau h)) \mathrm{d}\tau = y_0 + h \sum_{i=1}^{k} b_i J \nabla H(Y_i),$$

one then obtains the following k-stage Runge-Kutta method,

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array} \tag{13}$$

where, as usual, $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^k$ are the vectors containing the weights and the abscissae, respectively, and (see, e.g. [6–8])

$$A = \mathcal{P}_{s+1} \hat{X}_s \mathcal{P}_s^T \Omega \in \mathbb{R}^{k \times k}, \tag{14}$$

with

$$\mathcal{P}_r = \begin{pmatrix} P_0(c_1) & \ldots & P_{r-1}(c_1) \\ \vdots & & \vdots \\ P_0(c_k) & \ldots & P_{r-1}(c_k) \end{pmatrix} \in \mathbb{R}^{k \times r}, \qquad r = s, s+1, \tag{15}$$

$$\hat{X}_s = \left(\begin{array}{ccccc} \frac{1}{2} & -\xi_1 & & & \\ \xi_1 & 0 & \ddots & & \\ & \ddots & \ddots & -\xi_{s-1} & \\ & & \xi_{s-1} & 0 & \\ \hline & & & & \xi_s \end{array}\right) \equiv \left(\begin{array}{c} X_s \\ \hline 0 \ldots 0 \, \xi_s \end{array}\right) \in \mathbb{R}^{s+1 \times s}, \tag{16}$$

$$\xi_i = \left(2\sqrt{4i^2 - 1}\right)^{-1}, \qquad i = 1, \ldots, s, \tag{17}$$

$$\Omega = \mathrm{diag}(\boldsymbol{b}) \in \mathbb{R}^{k \times k}. \tag{18}$$

We observe that, when $k = s$, (14) becomes the W-transformation [21, pag. 79] of the s-stage Gauss-Legendre Runge-Kutta method. Consequently, (14) can be also regarded as a generalization of the W-transformation.

[2]Hereafter, we shall always assume this choice.

Clearly, the Runge-Kutta method (13)–(18) makes sense also for general non-polynomial Hamiltonians. Consequently, according to [5], we give the following definition.

Definition 1 The Runge-Kutta method (13)–(18) is called HBVM(k, s).

The following properties [5, 8] elucidate the role of two indices k (number of ascissae) and s (degree of the underlying polynomial σ) in the previous definition.

Theorem 1 *For all $k \geq s$, a HBVM(k, s) method:*

- *has order $2s$, that is:*
$$y_1 - y(h) = O(h^{2s+1});$$
- *is energy conserving for all polynomial Hamiltonians of degree ν satisfying (11);*
- *for general non-polynomial (but suitably regular) Hamiltonians, one has:*
$$H(y_1) - H(y_0) = O(h^{2k+1}). \tag{19}$$

Remark 1 From (19) one deduces that a HBVM(k, s) method is *practically* energy-conserving also in the case of non-polynomial Hamiltonians, provided that k is large enough. Indeed, on a computer, it is enough to approximate the involved integrals to within round-off errors.

Remark 2 Though the method (13)–(18) has been derived in the context of Hamiltonian systems, we stress that it makes sense also when replacing problem (1) by a generic (i.e., non Hamiltonian) initial value problem in the form $y' = f(t, y)$ [8].

For sake of completeness, and for later reference, we also report the following result, which actually shows that HBVM(k, s) methods, with the choice (9) of the abscissae, can be regarded as a generalization of the s-stage Gauss-Legendre collocation formulae [5].

Theorem 2 *HBVM(s, s) coincides with the s-stage Gauss-Legendre collocation method.*

If we set $\boldsymbol{y}$ the (block) vector with the internal stages (12) and $e = (1, \ldots, 1)^T \in \mathbb{R}^k$, the discrete problem generated by a HBVM(k, s) method is given by

$$\boldsymbol{y} = e \otimes y_0 + hA \otimes J \nabla H(\boldsymbol{y}), \tag{20}$$

which is a nonlinear system of (block) dimension k.[3] However, in view of (19), k is likely to be much larger than s and, consequently, such a formulation is in general not recommendable.

[3] Here $\nabla H(\boldsymbol{y})$ is the block vector whose entries are given by the gradient of H evaluated at the k stages.

To derive a more efficient formulation, let us set γ the (block) vector containing the coefficients defining the polynomial σ in (8), thus obtaining:

$$\gamma = \mathcal{P}_s^T \Omega \otimes J \nabla H(\boldsymbol{y}), \qquad \boldsymbol{y} = e \otimes y_0 + h\mathcal{P}_{s+1}\hat{X}_s \otimes I\, \gamma.$$

Combined together, such equations provide us with the following discrete problem,

$$F(\gamma) \equiv \gamma - \mathcal{P}_s^T \Omega \otimes J \nabla H\left(e \otimes y_0 + h\mathcal{P}_{s+1}\hat{X}_s \otimes I\, \gamma\right) = \mathbf{0}, \tag{21}$$

whose (block) size is always s, independently of k. In general, quite inexpensive iterations (e.g., the fixed-point iteration) could be used for solving (21). Nevertheless, in case, e.g., of *stiff oscillatory problems*, this could not be practical, since a very small stepsize h would be required: in such a case, a Newton-type iteration is more appropriate (see the second test problem in Section 5). As a popular example, one easily checks that the simplified Newton iteration, applied for solving (21), consists in the following iteration [6]:

$$\begin{aligned} \text{solve}: \quad \left[I - hX_s \otimes J\nabla^2 H_0\right] \Delta^\ell &= -F(\gamma^\ell) \\ \gamma^{\ell+1} &= \gamma^\ell + \Delta^\ell, \qquad \ell = 0, 1, \ldots, \end{aligned} \tag{22}$$

where $\nabla^2 H_0$ is the Hessian of $H(y)$ evaluated at y_0. Consequently, the bulk of the computational cost is due to the factorization of the matrix

$$M_0 = I - hX_s \otimes J\nabla^2 H_0,$$

having dimension $2sm \times 2sm$. In the next section, we shall see how to efficiently solve the iteration (22).

Remark 3 As is clear from the previous arguments, HBVM(k, s) methods, with a suitable choice of k, is (at least practically) energy-conserving, whereas HBVM(s, s) (i.e., the symplectic s-stage Gauss method) in general is not. Consequently, by taking into account that the (block) dimension of the discrete problem generated by a HBVM(k, s) method is always s independently of k, this method is preferable to the s-stage Gauss method when an accurate conservation of the energy is required.

3 The new splitting procedure

The iteration (22) is similar in structure to the simplified-Newton iteration applied to the original system (20), for which a number of splitting procedures have been devised: as an example, triangular splittings are defined in [1, 4, 24, 25]; a diagonal splitting, derived from the so called *blended implementation* of the methods, is studied in [10, 11]; additional approaches are described, e.g., in [2, 18, 19, 22, 23, 30]; moreover, we mention that a comprehensive linear analysis of convergence for such iterations (generalizing that at first proposed in [24]) is reported in [13]. However, the triangular splitting iteration defined in [24, 25], along with the modified triangular splitting iteration defined in [1], turn out to be not effective for (22), due to the particular structure of the matrix X_s (see (16)). Conversely, the *blended iteration* defined in [10, 11] (see also [13]), turns out more appropriate, as is shown in [6].

We here shall devise a different iterative procedure, which appears to be even more favourable. This is the subject of the remaining part of this section. The main idea is similar to that explained in [4] for Radau IIA collocation methods, even though the framework and the overall details (and results) are definitely different: i.e., to replace the set of s (block) unknowns, given by entries of the (block) vector γ defined in (21), with a more convenient one. To begin with, let us consider the polynomial (7) and introduce the new set of (block) unknowns,

$$\hat{\gamma}_i \equiv \sum_{j=0}^{s-1} P_j(\hat{c}_i)\gamma_j(\sigma), \qquad i = 1, \ldots, s, \tag{23}$$

defined as the evaluation of (7) at the set of distinct *auxiliary abscissae*

$$\hat{c}_1\,,\ \ldots\ ,\ \hat{c}_s. \tag{24}$$

Introducing the (block) vector

$$\hat{\gamma} = \begin{pmatrix} \hat{\gamma}_1 \\ \vdots \\ \hat{\gamma}_s \end{pmatrix}, \tag{25}$$

and the matrix

$$\hat{\mathcal{P}} = \left(P_{j-1}(\hat{c}_i) \right) \in \mathbb{R}^{s\times s}, \tag{26}$$

we can recast (23) in vector form as

$$\hat{\gamma} = \hat{\mathcal{P}} \otimes I\, \gamma. \tag{27}$$

In terms of the new unknown vector $\hat{\gamma}$, the simplified Newton iteration (22) reads:

$$\begin{aligned} \text{solve}: \ \hat{M}_0 \hat{\Delta}^{\ell} &= -\hat{\mathcal{P}} \otimes I\, F(\hat{\mathcal{P}}^{-1} \otimes I\, \hat{\gamma}^{\ell}) \equiv \boldsymbol{\eta}^{\ell}, \\ \hat{\gamma}^{\ell+1} &= \hat{\gamma}^{\ell} + \hat{\Delta}^{\ell}, \qquad \ell = 0, 1, \ldots, \end{aligned} \tag{28}$$

where

$$\hat{M}_0 = I - h\left(\hat{\mathcal{P}} X_s \hat{\mathcal{P}}^{-1}\right) \otimes J\nabla^2 H_0 \equiv I - h\hat{A} \otimes J\nabla^2 H_0. \tag{29}$$

Remark 4 We stress that matrix $\hat{A} = \hat{\mathcal{P}} X_s \hat{\mathcal{P}}^{-1}$ is independent of k: it only depends on s, whichever is the considered value of $k \geq s$. Consequently, the approach presented below also applies to the case $k = s$, that is, to the s-stages Gauss method.

The key idea is that of choosing the abscissae (24) such that $\hat{A}$ can be factored as

$$\hat{A} = \hat{L}\hat{U}, \tag{30}$$

with $\hat{U}$ upper triangular with unit diagonal entries, and $\hat{L}$ lower triangular with constant diagonal entries. In such a case, by following the approach of van der Houwen et al. [24, 25], the iteration (28) is replaced by the *inner-outer* iteration

$$\begin{aligned} \text{solve}: \ \left[I - h\hat{L} \otimes J\nabla^2 H_0\right] \hat{\Delta}^{\ell,r+1} &= h\hat{L}(\hat{U} - I) \otimes J\nabla^2 H_0\, \hat{\Delta}^{\ell,r} + \boldsymbol{\eta}^{\ell}, \\ & \qquad r = 0, 1, \ldots, \mu - 1, \\ \hat{\gamma}^{\ell+1} &= \hat{\gamma}^{\ell} + \hat{\Delta}^{\ell,\mu}, \qquad \ell = 0, 1, \ldots. \end{aligned} \tag{31}$$

In particular, since $\hat{\Delta}^{\ell,0} = 0$, the choice $\mu = 1$ corresponds to the approach used by van der Houwen et al. to devise PTIRK methods [24], whereas, if μ is large enough to have full convergence of the inner-iteration (the one on r), then the outer iteration is equivalent to (28). Clearly, all the intermediate possibilities can be suitably considered.

After the convergence of (31), the new approximation is computed (see (8)) as

$$y_1 = y_0 + h\gamma_0,$$

where γ_0 (i.e., the first block entry of the vector γ), is retrieved from (27). We observe that the diagonal entries of the factor $\hat{L}$ are all equal to a given value, say d_s, which has the obvious advantage that one only needs to factor the matrix

$$I - hd_s J \nabla^2 H_0 \in \mathbb{R}^{2m\times 2m}. \tag{32}$$

Remark 5 In an actual computational code, such a matrix can be kept constant over a number of steps, being factored only when the Hessian needs to be revaluated and/or the stepsize is modified. In this paper, we deliberately ignore this issue, which requires a further analysis (see, e.g., [12] for the code described in [11]). Consequently, in the numerical tests we shall use a constant stepsize and compute the Hessian at each step.

Concerning d_s, the following result holds true.

Theorem 3 *Assume that the factorization (30) is defined and that the factor $\hat{L}$ has all its diagonal entries equal to d_s. Then, with reference to (17), one has:*

$$d_s = \begin{cases} \sqrt[s]{\prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i-1}^2}, & \text{if } s \text{ is even,} \\ \sqrt[s]{\frac{1}{2}\prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i}^2}, & \text{if } s \text{ is odd.} \end{cases} \tag{33}$$

Proof Assume that (29)–(30) hold true. Then

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

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

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

The thesis then follows by considering that, from (16),

$$\det(X_1) = \frac{1}{2}, \qquad \det(X_2) = \xi_1^2,$$

and, by applying the Laplace expansion, one obtains:

$$\det(X_s) = \begin{cases} \prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i-1}^2, & \text{if } s \text{ is even,} \\ \frac{1}{2}\prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i}^2, & \text{if } s \text{ is odd.} \end{cases} \tag{34}$$

□

By virtue of the previous result, in order to compute the auxiliary abscissae (24), we have symbolically solved the following set of equations, which is obviously equivalent to requiring that the factor $\hat{L}$ has the diagonal entries equal to each other:

$$\det(\hat{A}_{\ell+1}) = d_s \det(\hat{A}_\ell), \qquad \ell = 1, \ldots, s-1, \tag{35}$$

where $\hat{A}_\ell$ denotes the principal leading submatrix of order ℓ of $\hat{A}$, and d_s is given by (33).

Remark 6 We observe that the auxiliary abscissae (24) are s, whereas the algebraic conditions (35) are $s-1$. This means that a further condition can be imposed on the abscissae: we have chosen it in order to improve the convergence properties of the iteration (31), according to the linear analysis of convergence reported in Section 4; in particular, we shall (approximately) minimize the *maximum amplification factor* of the iteration, as it will be later explained.

The obtained results are listed in Table 1, for $s = 2, \ldots, 6$, from which one sees that in all cases the abscissae are distinct and inside the interval $[0, 1]$.

We emphasize that, for any given s, the distribution of the auxiliary abscissae (24) is independent of k and so is the factorization (30) of the matrix $\hat{A}$ whose computation is responsible of the bulk of the computational effort during the integration process. This property has a relevant consequence during the implementation phase of this class of methods. In fact, one can conjecture a procedure to advance the time that dynamically selects the most appropriate value of k. Depending on the specific problem at hand and the configuration of the system at the given time, one can easily switch from a symplectic to an energy preserving method by choosing $k = s$ (Gauss method) or $k > s$, respectively.

4 Convergence analysis and comparisons

In this section we briefly analyze the splitting procedure (31). In general, its convergence properties could be discussed in the framework of quasi-Newton methods, leading to the (quite) obvious result that linear convergence is obtained for sufficiently small h, provided that H is suitably regular. Nevertheless, a more suited approach, which has proved to be very effective in the actual design of efficient variable-order/variable-stepsize codes for ODE-IVPs (see, e.g., [11]), is based on the linear analysis of convergence in [24] (further developed in [13]). Such an analysis is well motivated from the fact that the inner iteration in (31) amounts to solving a linear system. This latter system can be thought of being obtained by applying the original numerical method to the local (frozen) linearized problem. As a consequence, one can decompose it in the subspaces spanned by the eigenvalues of the Jacobian. Equivalently, one can directly consider the scalar problem defined by each eigenvalue. Consequently, one is led to study the behavior of the method when applied to the celebrated test equation:

$$y' = \lambda y, \qquad y(t_0) = y_0. \tag{36}$$

Table 1 Auxiliary abscissae (24) for the HBVM(k, s) and s-stage Gauss method, $s = 2, \ldots, 6$, and the diagonal entry d_s (see (33)) of the corresponding factor $\hat{L}$

$s = 2$	
$\hat{c}_1$	0.26036297108184508789101036587842555
$\hat{c}_2$	1
d_2	0.28867513459481288225457439025097873
$s = 3$	
$\hat{c}_1$	0.15636399930006671060146617869938122
$\hat{c}_2$	0.45431868644630821020177903150137523
$\hat{c}_3$	0.948
d_3	0.20274006651911333949661483325792675
$s = 4$	
$\hat{c}_1$	0.11004843257056123468614502691988075
$\hat{c}_2$	0.31588689139705398683980065724981436
$\hat{c}_3$	0.53114668286639796587351917750274705
$\hat{c}_4$	0.884
d_4	0.15619699684601279005430416526875577
$s = 5$	
$\hat{c}_1$	0.08422178443461232088418554160093421 8
$\hat{c}_2$	0.24861852058856201805181177902229 3944
$\hat{c}_3$	0.41372526881522095641549864330214 5284
$\hat{c}_4$	0.58709874897187711603088243675196 2384
$\hat{c}_5$	0.9338
d_5	0.12702337351164258963093490787943281
$s = 6$	
$\hat{c}_1$	0.20985774196263657630356114041757724
$\hat{c}_2$	0.36816786358152563671526302698797908
$\hat{c}_3$	0.39607328223635472401921951140390213
$\hat{c}_4$	0.62783521091780460858476326939502046
$\hat{c}_5$	0.04580307227138364391540767310611717
$\hat{c}_6$	0.94225
d_6	0.10702845478806509529222890981996019

Clearly, one directly arrives to the same conclusion in case problem (1) is separable, with a quadratic Hamiltonian, with the eigenvalues lying on the imaginary axis. Since problem (36) is linear, the iteration (31) consists in solving the inner iteration alone, so that we can skip the index ℓ of the outer iteration. By setting, as is usual, $q = h\lambda$, one then obtains that the error equation associated with the iteration (31) is given by

$$e_{r+1} = Z(q)e_r, \qquad Z(q) := q(I - q\hat{L})^{-1}\hat{L}(\hat{U} - I), \qquad r = 0, 1, \ldots, \tag{37}$$

where e_r is the error vector at step r and $Z(q)$ is the iteration matrix induced by the splitting procedure. This latter will converge if and only if its spectral radius,

$$\rho(q) := \rho(Z(q)),$$

is less than 1. The *region of convergence* of the iteration is then defined as

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

The iteration is said to be *A-convergent* if $\mathbb{C}^- \subseteq \mathbb{D}$. If, in addition, the *stiff amplification factor*,

$$\rho^\infty := \lim_{q\to\infty} \rho(q),$$

is null, then the iteration is said to be *L-convergent*. [4] In our case, since

$$Z(q) \to (\hat{U} - I), \qquad q \to \infty, \tag{38}$$

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 $\hat{L}$, d_s, is positive, as was shown in (33)) 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. Another useful parameter is the *nonstiff amplification factor*,

$$\tilde{\rho} := \rho(\hat{L}(\hat{U} - I)), \tag{39}$$

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

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

Clearly, the smaller ρ^* and $\tilde{\rho}$, the better the convergence properties of the iteration.

With these premises, we can now better specify what anticipated in Remark 6, concerning the additional condition imposed to derive the auxiliary abscissae (24), while fulfilling the conditions (35). In more details, the abscissae listed in Table 1 have been computed by (approximately) solving the following constrained minimization problem:

$$\begin{aligned} &\min_{\hat{c}_1,\ldots,\hat{c}_s} \quad \rho^* \\ &\quad s.t. \quad \det(\hat{A}_{\ell+1}) = d_s \det(\hat{A}_\ell), \qquad \ell = 1,\ldots,s-1. \end{aligned}$$

Clearly, this has been made possible by the introduction of the transformation (27).

In Table 2 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 [24];
(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 [10];
(iv) the iteration defined by (31).

[4] In general, A-convergent iterations are appropriate for A-stable methods, and L-convergent iterations are appropriate for L-stable methods.

Table 2 Amplification factors for the triangular splitting in [24], the modified triangular splitting in [1], the *blended* iteration in [6], and the splitting (31), for the s-stage Gauss-Legendre formulae

	(i): triangular splitting in [24]		(ii): triangular splitting in [1]		(iii): *blended* iteration in [6]		(iv): triangular splitting (31)	
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

The last two cases coincide with those for the HBVM(k, s) methods, $k \geq s$

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

Remark 7 For sake of accuracy, we stress that, when dealing with the actual implementation of HBVM(k, s) methods, only the blended iteration [6] and the newly proposed one (31) can be considered, whereas the triangular splitting defined in [24] and its modified version [1] turn out to be not effective, as was pointed out at the beginning of Section 3. Consequently, in such a case, one has to consider only the last two group of columns in Table 2.

4.1 Averaged amplification factors

The previous amplification factors measure the asymptotic speed of convergence when an infinite number of iterations are performed, which is not the case, in the actual implementation of the methods. For this purpose (see, e.g., [13]) it is also customary to define corresponding *averaged* amplification factors, which measure the "average" convergence when a prescribed number of iterations is performed. In particular, by considering a suitable matrix norm $\|\cdot\|$, and with reference to what previously has been set out, we define the following *averaged amplification factors* when μ iterations of (37) are carried out:

$$\rho^*_\mu := \sup_{x\in\mathbb{R}} \sqrt[\mu]{\|Z(ix)^\mu\|},\ \tilde{\rho}_\mu := \sqrt[\mu]{\left\|\left[\hat{L}(\hat{U}-I)\right]^\mu\right\|},\ \rho^\infty_\mu := \sqrt[\mu]{\|(\hat{U}-I)^\mu\|}. \tag{40}$$

Clearly,

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

Table 3 Averaged amplification factors (40) for the splitting (31), 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

and

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

In Table 3 we list the obtained averaged amplification factors (40) when performing $\mu = 1, 2, 3$ iterations, by 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 8 When performing only μ inner-iterations for solving the discrete problem generated by (36), we have to consider also the *outer* iteration, 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 for the outer iteration is given by (compare with (31)):

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

Consequently, the previous convergence analysis also applies to the present case.

5 Numerical Tests

In this section, we report a couple of numerical examples, aimed to put into evidence the features of the methods, and/or the effectiveness of the proposed iterative

Table 4 Results when solving Problem (41)–(42) 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

Table 5 Fixed-point iterations for solving problem (43)–(45), 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
$5 \cdot 10^{-4}$	***

procedure. 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:

- the fixed-point iteration;
- the blended iteration in [6];
- the splitting iteration (31) with 2 inner iterations.

The choice of 2 inner iterations in (31) makes the cost of one outer iteration comparable to that of one blended iteration, provided that (31) is implemented as suggested in [4]. The total number of functional evaluations equals the number of iterations times k. Moreover, for the latter two iterations, at each step one also needs to evaluate the Hessian $\nabla^2 H$, as well as to factor a matrix having the same size as that of the continuous problem (i.e., (32), in the case of the iteration (31)).

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. It is defined by the Hamiltonian:

$$H(x, y, z, x', y', z') = \frac{1}{2m}\left[\left(x' - \alpha\frac{x}{\rho^2}\right)^2 + \left(y' - \alpha\frac{y}{\rho^2}\right)^2 + \left(z' + \alpha \log \rho\right)^2\right], \tag{41}$$

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

$$x = 0.5, \qquad y = 10, \qquad x' = -0.1, \qquad y' = -0.3, \qquad z = z' = 0. \tag{42}$$

Table 6 Hamiltonian error, obtained by using a sixth-order explicit composition method based on the Störmer-Verlet method, for solving problem (43)–(45) on the interval [0,10] by using stepsize h

(*** means that the numerical solution diverges)

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 7 Newton-type iterations for solving problem (43)–(45), 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

In Table 4 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 proposed iterative procedure (31) is more effective than the blended iteration proposed in [6]. 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 independent of k.

The second test problem that we consider is, on the contrary, a *stiff oscillatory* problem. It is defined as a slight modification of the Fermi-Pasta-Ulam problem described in [20].[5] The Hamiltonian is now given by:

$$H(p,q) = \frac{1}{2}\sum_{i=1}^{m}\left(p_{2i-1}^2 + p_{2i}^2\right) + \frac{1}{4}\sum_{i=1}^{m}\omega_i^2\,(q_{2i} - q_{2i-1})^2 + \sum_{i=0}^{m}(q_{2i+1} - q_{2i})^4\,, \tag{43}$$

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

$$\omega_i = \omega_{m-i+1} = 10, \qquad i = 1, 2, 3, \qquad \text{and} \qquad \omega_4 = 10^4. \tag{44}$$

The starting vector is

$$p_i = 0, \qquad q_i = \frac{i-1}{2m-1}, \qquad i = 1, \ldots, 2m. \tag{45}$$

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.

[5] The original problem reported in [20] is obtained by setting $m = 3$ and $\omega_i = 50$, $i = 1, \ldots, m$, in (43).

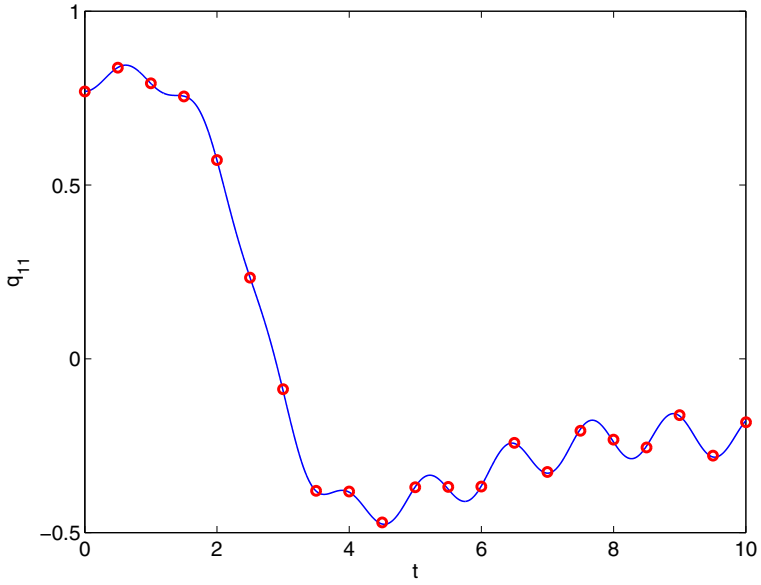

Fig. 1 Numerical approximation obtained by using HBVM(6,3) with stepsizes $h = 10^{-4}$ (*continuous line*) and $h = 0.5$ (*circles*) for solving problem (43)–(45)

As an example, 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 listed in Table 5. Similarly, explicit methods, which exist in this specific case since the problem is separable (see [29, Chapter 8]), suffer from similar restrictions on the stepsize because of stability reasons. In particular, we consider a composition method, having order 6, based on the second order Störmer-Verlet method (see [20, Chapter II.4] for details), requiring 18 function evaluations per step:[6] the results listed in Table 6 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 defined in [6] and the iteration (31) previously defined, one obtains the results listed in Table 7. 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 Figs. 1 and 2 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, $h = 0.5$.[7] Last but not least, from the figures in Table 7, one sees that

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

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

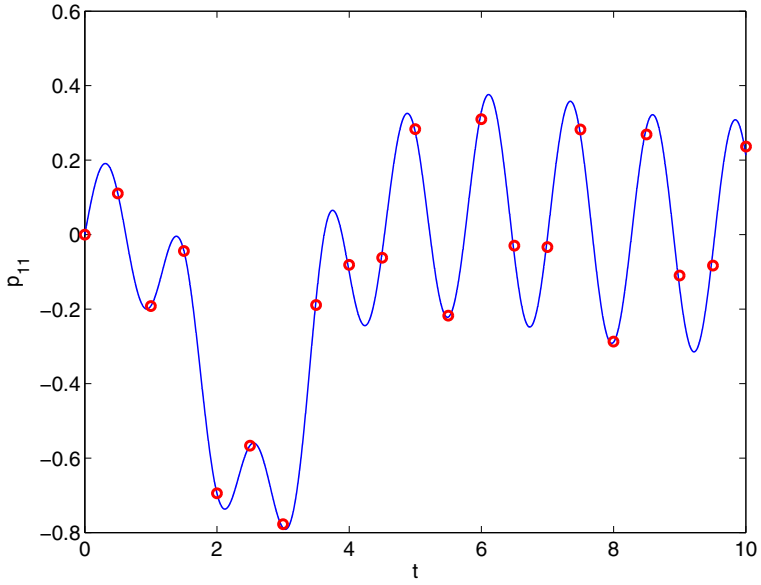

Fig. 2 Numerical approximation obtained by using HBVM(6,3) with stepsizes $h = 10^{-4}$ (*continuous line*) and $h = 0.5$ (*circles*) for solving problem (43)–(45)

the new iterative procedure (31) is the most effective one, though using only 2 inner iterations.

6 Conclusions

In this paper we have defined an efficient iterative procedure for solving the discrete problems generated by the application of HBVM(k, s) methods, a class of energy-conserving methods for polynomial Hamiltonian dynamical systems. The proposed implementation turns out to improve over that proposed in [6]. Moreover, it also applies to Gauss-Legendre formulae and the resulting linear convergence analysis shows that the proposed iterative procedure is the most effective, among those based on suitable splittings of the corresponding Butcher array of the methods, known from the literature. A few numerical tests confirm the effectiveness of the proposed iteration when numerically solving stiff oscillatory problems.

Acknowledgments The authors wish to thank the anonymous referees, for the useful comments and remarks.

References

1. Amodio, P., Brugnano, L.: A note on the efficient implementation of implicit methods for ODEs. J. Comput. Appl. Math. **87**, 1–9 (1997)

2. Bickart, T.A.: An efficient solution process for implicit Runge–Kutta methods. SIAM J. Numer. Anal. **14**(6), 1022–1027 (1977)
3. Brugnano, L., Iavernaro, F.: Line integral methods which preserve all invariants of conservative problems. J. Comput. Appl. Math. **236**, 3905–3919 (2012)
4. Brugnano, L., Iavernaro, F., Magherini, C.: Efficient implementation of Radau collocation methods (submitted for publication) (2012). arXiv:1302.1037
5. Brugnano, L., Iavernaro, F., Trigiante D.: Hamiltonian boundary value methods (Energy Preserving Discrete Line Methods). JNAIAM J. Numer. Anal. Ind. Appl. Math. **5**(1–2), 17–37 (2010)
6. Brugnano, L., Iavernaro, F., Trigiante, D.: A note on the efficient implementation of Hamiltonian BVMs. J. Comput. Appl. Math. **236**, 375–383 (2011)
7. Brugnano, L., Iavernaro, F., Trigiante, D.: The lack of continuity and the role of infinite and infinitesimal in numerical methods for ODEs: the case of symplecticity. Appl. Math. Comput. **218**, 8053–8063 (2012)
8. Brugnano, L., Iavernaro, F., Trigiante, D.: A simple framework for the derivation and analysis of effective one-step methods for ODEs. Appl. Math. Comput. **218**, 8475–8485 (2012)
9. Brugnano, L., Iavernaro, F., Trigiante, D.: Energy and quadratic invariants–preserving integrators based upon Gauss collocation formulae. SIAM J. Numer. Anal. **50**(6), 2897–2916 (2012)
10. Brugnano, L., Magherini, C.: Blended implementation of block implicit methods for ODEs. Appl. Numer. Math. **42**, 29–45 (2002)
11. Brugnano, L., Magherini, C.: The BiM code for the numerical solution of ODEs. J. Comput. Appl. Math. **164–165**, 145–158 (2004)
12. Brugnano, L., Magherini, C.: Some linear algebra issues concerning the implementation of blended implicit methods. Numer. Lin. Alg. Appl. **12**, 305–314 (2005)
13. Brugnano, L., Magherini, C.: Recent advances in linear analysis of convergence for splittings for solving ODE problems. Appl. Numer. Math. **59**, 542–557 (2009)
14. Burrage, K., Burrage, P.M.: Low-rank Runge-Kutta methods, symplecticity and stochastic Hamiltonian problems with additive noise. J. Comput. Appl. Math. **236**, 3920–3930 (2012)
15. Butcher, J.C.: On the implementation of implicit Runge-Kutta methods. BIT **16**, 237–240 (1976)
16. Butcher, J.C.: A transformed implicit Runge-Kutta method. J. Assoc. Comput. Mach. **26**, 237–240 (1979)
17. Cooper, G.J., Butcher, J.C.: An iteration scheme for implicit Runge-Kutta methods. IMA J. Numer. Anal. **3**, 127–140 (1983)
18. Cooper, G.J., Vignesvaran, R.: Some schemes for the implementation of implicit Runge-Kutta methods. J. Comput. Appl. Math. **45**, 213–225 (1993)
19. González-Pinto, S., González-Conceptión, S., Montijano, J.I.: Iterative schemes for Gauss methods. Comput. Math. Appl. **27**, 67–81 (1994)
20. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration, 2nd edn. Springer, Berlin (2006)
21. Hairer, E., Wanner, G.: Solving ordinary differential equations II. Stiff and differential-algebraic problems, 2nd edn. Springer-Verlag, Berlin (1996)
22. van der Houwen, P.J., Messina, E.: Splitting methods for second-order initial value problems. Numer. Algoritm. **18**, 233–257 (1998)
23. van der Houwen, P.J., Sommeijer, B.P.: The use of approximate factorization in stiff ODE solvers. J. Comput. Appl. Math. **100**, 11–21 (1998)
24. van der Houwen, P.J., de Swart, J.J.B.: Triangularly implicit iteration methods for ODE-IVP solvers. SIAM J. Sci. Comput. **18**, 41–55 (1997)
25. van der Houwen, P.J., de Swart, J.J.B.: Parallel linear system solvers for Runge-Kutta methods. Adv. Comput. Math. **7**(1–2), 157–181 (1997)
26. Iavernaro, F., Pace, B.: s-stage trapezoidal methods for the conservation of Hamiltonian functions of polynomial type. AIP Conf. Proc. **936**, 603–606 (2007)
27. Iavernaro, F., Pace, B.: Conservative block-boundary value methods for the solution of polynomial Hamiltonian systems. AIP Conf. Proc. **1048**, 888–891 (2008)
28. Iavernaro, F., Trigiante, D.: High-order symmetric schemes for the energy conservation of polynomial Hamiltonian problems. JNAIAM J. Numer. Anal. Ind. Appl. Math. **4**(1–2), 87–101 (2009)
29. Sanz-Serna, J.M., Calvo, M.P.: Numerical Hamiltonian problems. Chapman & Hall, London (1994)
30. Schlenkrich, S., Walther, A., Griewank, A.: Application of AD-based quasi-Newton methods to stiff ODEs. Lect. Notes Comput. Sci. Eng. **50**, 89–98 (2006)