A note on the efficient implementation of Hamiltonian BVMs

Luigi Brugnano\textsuperscript{a,∗}, Felice Iavernaro\textsuperscript{b}, Donato Trigiante\textsuperscript{c}

\textsuperscript{a} Dipartimento di Matematica, Università di Firenze, Viale Morgagni 67/A, 50134 Firenze, Italy
\textsuperscript{b} Dipartimento di Matematica, Università di Bari, Via Orabona 4, 70125 Bari, Italy
\textsuperscript{c} Dipartimento di Energetica, Università di Firenze, Via Lombroso 6/17, 50134 Firenze, Italy

\textbf{A R T I C L E I N F O}

\textbf{Keywords:}
Runge–Kutta methods
Hamiltonian problems
Hamiltonian Boundary Value Methods
Energy preserving methods
Blended methods
Energy drift

\textbf{A B S T R A C T}

We discuss the efficient implementation of Hamiltonian BVMs (HBVMs), a recently introduced class of energy preserving methods for canonical Hamiltonian systems (see Brugnano et al. \cite{8} and references therein), also sketching their \textit{blended} formulation. We also discuss the case of separable problems, for which the structure of the problem can be exploited to gain efficiency.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The conservation of energy allows to avoid the numerical drift sometimes observed when using standard numerical methods for solving canonical Hamiltonian problems, i.e., problems in the form

\[ y' = J \nabla H(y), \quad J = \begin{pmatrix} 0 & I_m \\ -I_m & 0 \end{pmatrix}, \quad y(t_0) = y_0 \in \mathbb{R}^{2m}, \tag{1} \]

where \( H(y) \) is a smooth scalar function and, in general, \( I_r \) will hereafter denote the identity matrix of dimension \( r \) (when the lower index will be omitted, the size of the matrix can be deduced from the context). In this respect, \textit{Hamiltonian Boundary Value Methods (HBVMs)} is a recently introduced class of methods able to conserve energy when \( H(y) \) is a polynomial of arbitrarily high degree. Clearly, this implies a \textit{practical} conservation of energy for any suitably regular Hamiltonian function, which will be assumed hereafter.

Energy preserving methods have been investigated by many authors even though the process has been initially so slow that the first unsuccessful attempts to derive energy-preserving Runge–Kutta methods culminated in the wrong general feeling that such methods could not even exist \cite{1}. The first successful attempts to solve the problem have been derived outside the class of Runge–Kutta methods, resulting in \textit{discrete gradient methods} (see \cite{2} and references therein) which are second order accurate. The first instance of energy-conserving Runge–Kutta method is given by the \textit{s-stage trapezoidal methods} \cite{3}. Such methods have been subsequently rederived in \cite{4}, through the discretization of the \textit{averaged vector field method} defined in \cite{5}. Additional examples of fourth and sixth-order conservative Runge–Kutta methods (for polynomial Hamiltonians of suitable degree) were presented in \cite{6,7}. All such energy-conserving Runge–Kutta methods have been derived by means of the new concept of \textit{discrete line integral}. The evolution of this idea eventually led to the definition of \textit{Hamiltonian Boundary Value Methods (HBVMs)} mentioned above \cite{8–13}, which admit a straightforward Runge–Kutta formulation.

The discrete line integral is nothing but a discrete counterpart of the line integral associated with a conservative vector field which is obtained by simply approximating the line integral by a suitable quadrature formula. In particular, in \cite{9}...
HBVMs based on Lobatto quadratures have been analyzed, whereas in [10] HBVMs based on Gauss–Legendre formulae have been considered. When the Hamiltonian function is a polynomial, such an approximation turns out to be exact, provided that a suitable number of nodes is introduced. In [10], it has been shown that all HBVMs having the same order are equivalent (i.e. they provide the very same numerical solution), under the assumptions that $H(y)$ is a polynomial and that the discrete line integral matches the line integral.

When $H(y)$ is not a polynomial, one can still get conservative formulae by letting the number of nodes tend to infinity, thus forcing the discrete line integral to approach the line integral. This is tantamount to skipping the discretization of the line integral during the computation of the formulae. Limit formulæ based on the Lagrange basis have been proposed in [14] and generalized to Poisson systems in [15]. Evidently, the polynomial case or the discretization of the integrals appearing in a limit method by means of a quadrature formula, leads back to a HBVM (see [10] for a discussion of this issue and for the definition of limit formulae over the Legendre and other bases).

In the case where the Hamiltonian is separable, i.e., when

$$H(y) \equiv H(q, p) = \frac{1}{2} p^T M p - U(q), \quad q, p \in \mathbb{R}^m,$$

with $M$ a symmetric and positive definite matrix, then (1) reduces to a special second order equation,

$$q' = M \nabla U(q),$$

and the associated HBVM may be properly formulated in order to take advantage, in terms of efficiency, from the above simplification. As a matter of fact, in this paper we investigate the efficient implementation of HBVMs also in the case of separable problems. In more detail, in Section 2 we briefly derive HBVMs, also providing the most convenient formulation of the discrete problem. Then, in Section 3 we investigate its efficient solution via the blended implementation of the methods, which has already proved to be very effective in other settings (see, e.g., [16–23]). The case of separable problems is then discussed in Section 4. A few numerical tests, along with some concluding remarks are then given in Section 5.

2. Hamiltonian BVMs (HBVMs)

The derivation of HBVMs will be done according to the approach followed in [12,13], which further simplifies the already simple idea initially used in [8–11] (see also [6,7]). Let us then consider the restriction of problem (1) to the interval $[t_0, t_0 + h]$, with the right-hand side expanded along an orthonormal basis $\{\hat{P}_j\}_{j \geq 0}$:

$$y'(t_0 + \tau h) = \int_0^1 \sum_{j \geq 0} \hat{P}_j(\tau) \int_0^1 \hat{P}_j(c) \nabla H(y(t_0 + ch)) \, dc, \quad \tau \in [0, 1].$$

(3)

In particular, we here consider an orthonormal polynomial basis, provided by the shifted and scaled Legendre polynomials on the interval $[0, 1]$, even though the arguments can be easily extended to more general bases. The basic idea, is now that of looking for an approximate solution belonging to the set of polynomials of degree not larger than $s$. This is achieved by truncating the series at the right-hand side in (3), thus obtaining the approximate problem

$$\sigma'(t_0 + \tau h) = \int_0^1 \sum_{j=0}^{s-1} \hat{P}_j(\tau) \int_0^1 \hat{P}_j(c) \nabla H(\sigma(t_0 + ch)) \, dc, \quad \tau \in [0, 1], \quad \sigma(t_0) = y_0.$$

(4)

The approximation to $y(t_0 + h)$ is then given by

$$y_1 \equiv \sigma(t_0 + h).$$

(5)

It can be easily seen that the energy is conserved at the new approximation since, considering that $J$ is skew-symmetric,

$$H(y_1) - H(y_0) = h \int_0^1 \nabla H(\sigma(t_0 + \tau h))^T \sigma'(t_0 + \tau h) \, d\tau$$

$$= h \int_0^1 \left[ \int_0^1 \hat{P}_j(\tau) \nabla H(\sigma(t_0 + ch)) \, d\tau \right]^T J \left[ \int_0^1 \hat{P}_j(c) \nabla H(\sigma(t_0 + ch)) \, dc \right] = 0.$$

Integrating both sides of the first equation in (4) yields

$$\sigma(t_0 + \tau h) = y_0 + h \sum_{j=0}^{s-1} \int_0^1 \hat{P}_j(\chi) \, d\chi \int_0^1 \hat{P}_j(c) \nabla H(\sigma(t_0 + ch)) \, dc,$$

(6)

which may be exploited to determine the shape of the unknown polynomial $\sigma$, provided that a technique to handle the rightmost integrals is taken into account: the obvious choice is the use of quadrature formulæ. If we assume that $H(y)$ is a
polynomial of degree \( v \), then the integrals appearing in (4) can be exactly computed by a Gaussian formula with \( k \) abscissas \( \{c_i\} \), in the event that
\[
k \geq \frac{3v}{2},
\]
thus obtaining a discrete problem in the form
\[
\sigma(t_0 + c_i h) \equiv \sigma_i = y_0 + h \sum_{j=0}^{s-1} \int_0^{c_j} \hat{p}_j(x) \, dx \sum_{\ell=1}^k b_i \hat{p}_j(c_i) \nabla H(\sigma_\ell), \quad i = 1, \ldots, k,
\]
where the \( b_i \) are the quadrature weights of the formula defined over the abscissae \( c_i \). For general, suitably regular (e.g., analytical) Hamiltonian functions, we can still use formula (8) in place of (6), provided that the integrals in (6) are approximated to machine precision\(^1\): in the following, we will always assume such an accuracy level when a non-polynomial function is considered, and consequently we will make no distinction between the integrals and the corresponding approximations as well as between the two polynomials \( \sigma \) obtained by solving either (8) or (6) (see [13] for more details).

Method (8)–(5) is called HBVM \((k, s)\): it was shown [10] that its order is 2 \( s \), for all \( k \geq s \). In particular, for \( k = s \) it reduces to the well-known \( s \)-stage Gauss method.

By introducing the matrices \( \Omega = \text{diag}(b_1, \ldots, b_k) \) and
\[
I_{s-1} = \left( \int_0^{c_i} \hat{p}_j(x) \, dx \right)_{j=1}^{s-1}_{j=1} \in \mathbb{R}^{k \times s}, \quad P_{s-1} = \left( \hat{p}_j(c_i) \right)_{j=1}^{s-1}_{j=1} \in \mathbb{R}^{k \times r},
\]
the HBVM \((k, s)\) can be recast as a Runge–Kutta method with Butcher tableau
\[
\begin{array}{c|c}
\hline
\{c_i\} & A \equiv I_{s-1} P_{s-1}^T \Omega \\
\hline
\end{array}
\]

The next result follows from well-known properties of Legendre polynomials (hereafter \( e_i \) denotes the \( i \)-th unit vector in \( \mathbb{R}^k \)).

**Lemma 1.** Let \( I_{s-1} \) and \( P_s \) be defined according to (9). Then,
\[
I_{s-1} = P_s X_s \equiv P_s \left( X_s^T \xi_s e_i \right)
\]
where
\[
X_s = \begin{pmatrix}
\frac{1}{2} & -\xi_1 & & \\
\xi_1 & 0 & \ddots & \\
& \ddots & \ddots & -\xi_{s-1} & \\
& & \xi_{s-1} & 0
\end{pmatrix}, \quad \xi_i = \frac{1}{2\sqrt{4j^2 - 1}}, \quad i \geq 1.
\]

Consequently, the matrix in the Butcher tableau (10) can be written as
\[
A = P_s X_s P_{s-1}^T \Omega.
\]

Notice that, since \( P_s X_s \) has \( s \) linearly independent columns, the \( k \times k \) coefficient matrix \( A \) has rank \( s \): it is then possible to recast the discrete problem in a more convenient form, whose (block) size is \( s \), rather than \( k \). This fact has been discussed in detail in [9,24] as sketched in the sequel. At first, let us partition the abscissae \( c_i \) into two sets: one with \( s \) abscissae, the other with the remaining \( k - s \) ones. For the sake of simplicity, we choose them as the first \( s \) ones and the remaining \( k - s \), respectively. According to [9,7], the corresponding stages are called fundamental stages and silent stages, respectively. The key idea is now that the silent stages can be obtained as a linear combination of the fundamental ones. Let us then partition the matrices \( I_{s-1}, P_{s-1}, \Omega \) as follows:
\[
I_{s-1} = \begin{pmatrix}
I_{s-1,1} & I_{s-1,2} \end{pmatrix}, \quad P_{s-1} = \begin{pmatrix}
P_{s-1,1} & P_{s-1,2} \end{pmatrix}, \quad \Omega = \begin{pmatrix}
\Omega_1 & \Omega_2 \end{pmatrix}, \quad I_{s-1,1}, P_{s-1,1}, \Omega_1 \in \mathbb{R}^{s \times s}.
\]

\(^1\) As we will see, increasing the order of the quadrature formula, namely the integer \( k \), will not result in an increase of the dimension of the discrete problem generated by the implementation of the method.
Similarly, let us denote by $\mathbf{y}_1$ the (block) vector, of dimension $s$, containing the fundamental stages, and by $\mathbf{y}_2$ the (block) vector, of dimension $k-s$, with the silent stages. One then obtains the equations:

$$\begin{align*}
\mathbf{y}_1 &= u_1 \otimes y_0 + hI_{s-1,1}\left[\mathcal{P}^T_{s-1,1} \Omega_1 \otimes J\nabla H(\mathbf{y}_1) + \mathcal{P}^T_{s-1,2} \Omega_2 \otimes J\nabla H(\mathbf{y}_2)\right], \\
\mathbf{y}_2 &= u_2 \otimes y_0 + hI_{s-1,2}\left[\mathcal{P}^T_{s-1,1} \Omega_1 \otimes J\nabla H(\mathbf{y}_1) + \mathcal{P}^T_{s-1,2} \Omega_2 \otimes J\nabla H(\mathbf{y}_2)\right],
\end{align*}$$

where $u_1 = (1, \ldots, 1)^T \in \mathbb{R}^s$, $u_2 = (1, \ldots, 1)^T \in \mathbb{R}^{k-s}$, and an obvious meaning of the (block) vector argument of $\nabla H(\cdot)$.

By considering that from Lemma 1 one readily deduces that matrix $I_{s-1,1}$ is nonsingular, from (14)–(15) one obtains that

$$\mathbf{y}_2 = \hat{u} \otimes y_0 + A_1 \otimes I_{2m} \mathbf{y}_1, \quad \hat{u} = u_2 - A_1u_1 \in \mathbb{R}^{k-s}, \quad A_1 = I_{s-1,2}I_{s-1,1}^{-1} \in \mathbb{R}^{k-s \times s}.$$  \hfill (16)

Then, by setting the matrices

$$B_1 = I_{s-1,1}\mathcal{P}^T_{s-1,1} \Omega_1 \in \mathbb{R}^{s \times s}, \quad B_2 = I_{s-1,1}\mathcal{P}^T_{s-1,2} \Omega_2 \in \mathbb{R}^{s \times k-s},$$

substituting (16) into (14) results in a discrete problem involving only the $s$ fundamental stages,

$$\Psi(\mathbf{y}_1) \equiv \mathbf{y}_1 - u_1 \otimes y_0 - h \left[B_1 \otimes J\nabla H(\mathbf{y}_1) + B_2 \otimes J\nabla H(\hat{u} \otimes y_0 + A_1 \otimes I_{2m} \mathbf{y}_1)\right] = 0.$$  \hfill (17)

The application of the simplified Newton method for solving (17) then gives

$$\left[\mathbf{I} - hC \otimes G_0\right] \delta^\ell = -\Psi(\mathbf{y}_1^\ell), \quad \mathbf{y}_1^{\ell+1} = \mathbf{y}_1^\ell + \delta^\ell, \quad \ell = 0, 1, \ldots,$$  \hfill (18)

where

$$C = B_1 + B_2A_1 \in \mathbb{R}^{s \times s} \quad \text{and} \quad G_0 = (J\nabla^2 H(y_0)) = 0.$$  \hfill (19)

It can be proved [24] that the spectrum of matrix $C$ is independent of the choice of the fundamental stages and coincides with that of matrix $X_1$ in (12). On the other hand, its condition number crucially depends on such a choice which, therefore, strongly affects the convergence of the iteration (18).

It is then advisable to derive a more convenient formulation of the discrete problem, still having (block) dimension $s$, no more requiring to distinguish between fundamental and silent stages. For this purpose, let us define the (block) vectors (see (4) and (8))

$$\begin{align*}
\mathbf{y} &= \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_k \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \gamma_0 \\ \vdots \\ \gamma_{s-1} \end{pmatrix}, \quad \gamma_j = \sum_{t=1}^k b_j \hat{p}_j(c_t) \nabla H(\sigma(t_0 + c_th)), \quad j = 0, \ldots, s - 1.
\end{align*}$$  \hfill (20)

In view of (4), we see that the vectors $\gamma_j$ may be interpreted as the coefficients in the expansion of the degree $s-1$ polynomial $\sigma'(t_0 + \tau h)$ along the orthonormal basis $\{\hat{p}_j\}_{j=0,...,s-1}$. From (8) one then obtains

$$\mathbf{y} = e \otimes y_0 + hI_{s-1} \otimes I_{2m} \mathbf{y},$$  \hfill (21)

with $e = (1, \ldots, 1)^T \in \mathbb{R}^k$, and then, by virtue of (20), one can solve the equation in the unknown $\mathbf{y}$,

$$F(\mathbf{y}) \equiv \mathbf{y} - (\mathcal{P}^T_{s-1} \Omega I_{s-1} \otimes J) \nabla H(e \otimes y_0 + hI_{s-1} \otimes I_{2m} \mathbf{y}) = 0.$$  \hfill (22)

The application of the simplified Newton iteration for solving (22) yields (see (19))

$$\left[\mathbf{I} - h\mathcal{P}^T_{s-1} \Omega I_{s-1} \otimes G_0\right] \Delta^\ell = -F(\mathbf{y}^\ell), \quad \mathbf{y}^{\ell+1} = \mathbf{y}^\ell + \Delta^\ell.$$  \hfill (23)

By virtue of (11), and considering that

$$\mathcal{P}^T_{s-1} \Omega \mathcal{P}_s = (I_0, 0) \in \mathbb{R}^{s \times s+1},$$  \hfill (24)

(23) reduces to

$$\left[\mathbf{I} - hX_1 \otimes G_0\right] \Delta^\ell = -F(\mathbf{y}^\ell), \quad \mathbf{y}^{\ell+1} = \mathbf{y}^\ell + \Delta^\ell, \quad \ell = 0, 1, \ldots.$$  \hfill (25)

As one may see, the iteration (25) now involves the matrix $X_1$ defined in (12), only depending on the parameter $s$, in place of $C$ (see (18)–(19)).
3. Blended implementation

From the arguments in the previous section, one then concludes that the discrete problem, to be solved at each integration step when approximating the Hamiltonian problem (1), is given by (22), thus requiring the solution of (25). We are going to solve such an equation by means of a blended implementation of the method, according to [16–18,23]. Indeed, such an implementation of block implicit methods has proved to be very effective, leading to the development of the codes BİN [18] and BİMD [22] for stiff ODE-IVPs and linearly implicit DAEs (the codes are available at the url [25], see also [26]). Let us, for the sake of simplicity, discard the iteration index. Consequently, we have to solve the linear system

\[ (I - hX_s \otimes G_0) \Delta = -F(y) \equiv \eta. \]  

(26)

Considering that matrix \( X_s \) (see (12)) is nonsingular, such an equation can be equivalently written as

\[ \rho \left( X_s^{-1} \otimes I_{2m} - hI_s \otimes G_0 \right) \Delta = \rho X_s^{-1} \otimes I_{2m} \eta \equiv \eta_1, \]

(27)

where \( \rho \) is a positive constant. By introducing the (matrix) weight function

\[ \theta = I_s \otimes S_0^{-1}, \quad S_0 = (I_{2m} - \rho h G_0)^{-1}, \]

(28)

we then obtain the following problem, which still has the same solution as (26):

\[ T(\Delta) \equiv \theta \left[ (I - hX_s \otimes G_0) \Delta - \eta \right] + (I - \theta) \left[ \rho \left( X_s^{-1} \otimes I_{2m} - hI_s \otimes G_0 \right) \Delta - \eta_1 \right] = 0. \]

(29)

One easily realizes that it is obtained as the blending, with weights \( \theta \) and \( (1 - \theta) \), of the two equivalent problems (26) and (27), respectively. Problem (29) defines the blended method associated with the original one, which we call blended HBVM, in the present case. The free parameter \( \rho \) is chosen in order to optimize the convergence properties of the corresponding blended iteration,

\[ \Delta_{n+1} = \Delta_n - \theta T(\Delta_n), \quad n \geq 0, \]

(30)

with an obvious meaning of the lower index. Such iteration only requires (see (28)) the factorization of the matrix \( S_0 \) having the same size as that of the continuous problem. According to the linear analysis of convergence in [20], the free parameter \( \rho \) is chosen as

\[ \rho = \rho_s \equiv \min\{ |\lambda| : \lambda \in \sigma(X_s) \}, \]

(31)

which provides optimal convergence properties (in particular, an \( L \)-convergent iteration [20]). This simple choice is due to the fact that the eigenvalues of matrix \( X_s \) coincide with those of the Butcher matrix of the \( s \)-stage Gauss–Legendre method [17]. A few values of (31) are listed in the table below, for the sake of completeness.

<table>
<thead>
<tr>
<th>s</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho_s )</td>
<td>0.2887</td>
<td>0.1967</td>
<td>0.1475</td>
<td>0.1173</td>
</tr>
</tbody>
</table>

**Remark 1.** A nonlinear version of (30) can be readily derived, by taking \( \Delta_n = 0 \) and updating the vectors \( \eta \) and \( \eta_1 \) in (29) at each iteration.

4. The case of separable problems

Let us now apply the method to the separable problem (2). By setting the (block) vectors

\[ q = (q_1^T, \ldots, q_s^T)^T, \quad p = (p_1^T, \ldots, p_s^T)^T, \]

one then obtains (see (13)),

\[ q = e \otimes q_0 + hA \otimes Mp, \quad p = e \otimes p_0 + hA \otimes I_n \nabla U(q). \]

i.e., since \( Ae = c \equiv (c_1, \ldots, c_s)^T \),

\[ q = e \otimes q_0 + hc \otimes Mp_0 + h^2 A^2 \otimes M \nabla U(q). \]

(32)

Moreover, taking into account (10)–(13) and (24), one obtains

\[ A^2 = I_{s^{-1}X_s} \otimes \left( s^{-1} \Omega \right). \]

(33)

The new approximations to \( q(t_0 + h) \) and \( p(t_0 + h) \) are then given by

\[ q_0 + hMp_0 + h^2 e^T \Omega A \otimes M \nabla U(q), \quad p_0 + he^T \Omega \otimes I_n \nabla U(q), \]

respectively. By using similar arguments as those given in Section 2 (see (21)), we set

\[ q = e \otimes q_0 + hc \otimes Mp_0 + h^2 I_{s^{-1}X_s} \otimes I_n \eta. \]
where the (block) entries of \( y \) are defined by (compare with (20))

\[
y_j = \sum_{\ell=1}^{k} b_{j,\ell} P(c_{\ell}) M_\nabla U(q_\ell), \quad j = 0, \ldots, s - 1,
\]

i.e., \( y = P^T \gamma \otimes M_\nabla U(q) \). Thus, we obtain the following equation in the unknown \( y \) (which is analogous to (22)):

\[
F(y) \equiv y - (P^T \gamma \otimes M) \nabla U(e \otimes q_0 + hc \otimes M_0 + h^2 I_{s-1} X_0 \otimes I_{m}) = 0.
\]  

(34)

Similarly to what was seen in Section 3, the application of the simplified Newton iteration for solving (34) then gives, by virtue of (11) and (24), and setting \( G_0 = M_\nabla U(q_0) \),

\[
[I - h^2 X_0^2 \otimes G_0] \Delta = -F(y^{(\ell)}), \quad y^{(\ell+1)} = y^{(\ell)} + \Delta^{(\ell)}, \quad \ell = 0, 1, \ldots.
\]

(35)

which, as in the previous case, has (block) size \( s \), rather than \( k \). The problem is then exactly that seen in (25), via the formal substitutions

\[
h \rightarrow h^2, \quad X_0 \rightarrow X_0^2.
\]

(36)

This means that we can repeat similar steps for the blended solution of (35), by following the same arguments seen in Section 3. In more detail, (26)–(30) can be repeated, by considering the formal substitutions (36) and, moreover,

\[
\rho \rightarrow \rho^2, \quad I_{2m} \rightarrow I_{m}.
\]

Also in this case [19,20], the optimal choice of the parameter \( \rho \) turns out to be given by (31). Moreover, a nonlinear version of (35) is readily obtained, according to what is said in Remark 1.

5. Numerical tests

We here consider a few model problems to test the proposed algorithms and methods, in order to confirm the usefulness of the proposed approach. Newton-type iterations, and then, the blended implementation of the methods, proved to be very effective, when speaking about dissipative problems, in case of stiff problems (see [27] for a recent survey about stiffness). In the case of special second order problems, their counterpart is given by highly oscillatory problems, namely Hamiltonian problems whose solution \( y(t) \) combines components with dominant short frequencies and components with large frequencies (with respect to the interval of integration).

The first test problem is defined by the non-polynomial Hamiltonian

\[
H(q, p) = \frac{1}{2} p^2 + \sin^2 (10^2 q),
\]

which is solved over the interval \([0, 10]\) starting from \((q_0, p_0) = (0, 10^{-1})\). We use the following fourth-order methods:

- the symplectic 2-stage Gauss (GAUSS2 hereafter) method (i.e., HBVM(2, 2), which is not conservative, for the problem at hand);
- the (practically) conservative HBVM(8, 2) method (for the used step-sizes).

For all methods, we use their blended and fixed-point implementations. Moreover, the methods are used both on the first and second order formulation of the problem. Table 1 summarizes the obtained results, in terms of iterations needed for convergence, as well as of error in the numerical Hamiltonian. It is evident that the conservative method is more effective than the symplectic one, with respect to the conservation of the Hamiltonian; moreover, the blended implementations of the methods appears to be more robust than the fixed-point one.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of fixed-point or blended iterations (n.c. = no convergence) required for solving problem (37), along with the maximum relative error (errH) in the numerical Hamiltonian, for the HBVM(8, 2) and GAUSS2 methods, by using step size ( h_i = 2^{-i}10^{-1} ), ( i = 0, \ldots, 6 ).</td>
</tr>
<tr>
<td>( i )</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>First order</td>
</tr>
<tr>
<td>Fixed-pt</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>


Fig. 1. Level curves for problem (38)–(40) numerically computed by the energy conserving HBVM(8, 2) method (left plot) and by the symplectic GAUSS2 method (right plot), by performing $10^3$ steps with step size $h = 10^{-3}$. The conservative method evidently outperform the symplectic one, for the given step size $h$.

The second test problem is defined by the polynomial Hamiltonian

$$H(q, p) = p^2 + \beta^2 q^2 + \alpha^2 (q + p)^2$$

which has a marginally stable equilibrium at the origin. We consider the following set of parameters

$$\beta = 10, \quad \alpha = 1, \quad n = 4,$$

so that the Hamiltonian has degree 8 and the problem in not separable. We consider the same fourth-order methods as in the previous example, noticing that, for the problem at hand, the HBVM(8, 2) method is conservative. Both methods are used, with step size $h = 10^{-3}$, for integrating the problem on the interval $[0, 1]$, with initial points

$$(q_0, p_0) = (i, -i), \quad i = 1, \ldots, 10,$$

(40)

to obtain the corresponding level curves of the Hamiltonian. We notice that, since the problem has one degree of freedom, the numerical solution provided by the HBVM(8, 2) lies on the very same continuous trajectory, or level curve, of the original problem. Consequently the level curves are recovered without error, as illustrated in the left plot of Fig. 1. This is not the case with the GAUSS2 method which evidently introduces huge deformations of the level curves especially when the numerical solution is not close to the equilibrium point. Table 2 summarizes the obtained results, in terms of iterations to obtain convergence (blended or fixed-point). The maximum error in the Hamiltonian is also listed in the table. As one can see, the blended implementation of the method is more efficient than the fixed-point implementation.

Finally, we consider the well-known Fermi–Pasta–Ulam problem [see [28, Section 1.5.1]], defined by the Hamiltonian

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

(41)
Table 3

<table>
<thead>
<tr>
<th>Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of fixed-point or blended iterations (n.c. = no convergence) required for solving problem (41)–(42), along with the maximum relative error (errH) in the numerical Hamiltonian, for the HBVM(8, 2) and GAUSS2 methods, by using step size $h_i = 2^{i-1}$, $i = 0, \ldots, 9$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i$</th>
<th>HBVM(4, 2)</th>
<th>GAUSS2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First order</td>
<td>Second order</td>
</tr>
<tr>
<td>0</td>
<td>n.c.</td>
<td>n.c.</td>
</tr>
<tr>
<td>1</td>
<td>4.176</td>
<td>10.647</td>
</tr>
<tr>
<td>2</td>
<td>8.300</td>
<td>17.324</td>
</tr>
<tr>
<td>3</td>
<td>13.517</td>
<td>25.144</td>
</tr>
<tr>
<td>4</td>
<td>21.242</td>
<td>38.233</td>
</tr>
<tr>
<td>5</td>
<td>45.984</td>
<td>46.478</td>
</tr>
<tr>
<td>6</td>
<td>60.943</td>
<td>63.291</td>
</tr>
<tr>
<td>7</td>
<td>102.400</td>
<td>107.968</td>
</tr>
<tr>
<td>8</td>
<td>155.546</td>
<td>155.988</td>
</tr>
<tr>
<td>9</td>
<td>342.512</td>
<td>327.912</td>
</tr>
</tbody>
</table>

with $q_0 = q_{2n+1} = 0$. We use the following parameters and starting point, with $[0, 10]$ the integration interval:

$m = 6, \quad \omega = 100, \quad p_i = 0, \quad q_i = (i − 1)/10, \quad i = 1, \ldots, 6. \quad (42)$

Here, the Hamiltonian function is a polynomial of degree 4, so that the fourth-order HBVM(4, 2) method is energy conserving.

Table 3 summarizes the obtained results by using the HBVM(4, 2) and GAUSS2 methods, both for the first and second order formulation of the problem, as well as by using the blended or fixed-point iterations for solving the discrete problems. The maximum relative error in the numerical Hamiltonian is also listed, thus confirming that the method HBVM(4, 2) is conservative, contrary to GAUSS2.

From the numerical tests, one can then conclude that the proposed blended implementation of HBVMs turns out to be robust and efficient for larger step sizes (as it usually happens for Newton-like iterations). Moreover, the energy-conserving property of such methods turns out to be very remarkable. Finally, the methods can be conveniently recast into a second order formulation, in the case of separable problems.

References

The codes BiM and BiMD Home Page: http://www.math.unifi.it/~brugnano/BiM/.

TestSet for IVP Solvers (rel. 2.4): http://pitagora.dm.uniba.it/~testset/.
