On the effectiveness of spectral methods for the numerical solution of multi-frequency highly-oscillatory Hamiltonian problems

Luigi Brugnano [*]   Juan I. Montijano[†]   Luis Rández[‡]

May 15, 2018

Abstract

Multi-frequency, highly-oscillatory Hamiltonian problems derive from the mathematical modelling of many real life applications. We here propose a variant of Hamiltonian Boundary Value Methods (HBVMs), which is able to efficiently deal with the numerical solution of such problems.

Keywords: Multi-frequency highly-oscillatory problems, Hamiltonian problems, energy-conserving methods, spectral methods, Legendre polynomials, Hamiltonian Boundary Value Methods, HBVMs.

MSC: 65P10, 65L05.

1   Introduction

Multi-frequency highly-oscillatory Hamiltonian problems appear often in mathematical models of real life applications such as molecular dynamics [48] or multibody mechanical systems [45, 46]. They also occur when solving Hamiltonian PDEs by means of a proper space-semidiscretization. One common feature of this class of problems is that the spectrum of the Jacobian matrix of the vector field has one or more eigenvalues located on the imaginary axis and with a very large modulus. We here consider the efficient numerical solution of such problems. For purposes of analysis and to sketch the main facts about the methods, we consider the model problem

$$\ddot{q} + A^2 q + \nabla f(q) = 0, \quad t \geq 0, \qquad q(0) = q_0, \ \dot{q}(0) = \dot{q}_0 \in \mathbb{R}^m, \tag{1}$$

where, without loss of generality, we can assume A to be a symmetric and positive definite (spd) matrix: in fact, possible zero eigenvalues could be, e.g., set to 1, then moving the residual to the

[*]Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Viale Morgagni 67/A, 50134 Firenze, Italy. E-mail: `luigi.brugnano@unifi.it`

[†]Departamento de Matemática Aplicada, Universidad de Zaragoza, Pza. San Francisco s/n, 50009 Zaragoza, Spain. E-mail: `monti@unizar.es`

[‡]Departamento de Matemática Aplicada, Universidad de Zaragoza, Pza. San Francisco s/n, 50009 Zaragoza, Spain. E-mail: `randez@unizar.es`

$\nabla f(q)$ term. The equation (1) is defined by the separable Hamiltonian

$$H(q, \dot{q}) = \frac{1}{2}(\|\dot{q}\|_2^2 + \|Aq\|_2^2) + f(q), \tag{2}$$

with f a regular enough function. Moreover, in (1) we assume the nonlinear term to be "small" when compared with the linear part. Consequently, we shall assume

$$\|A\| =: \omega \gg \|\nabla f\|, \tag{3}$$

where the last inequality holds in a suitable domain containing the trajectory solution. Clearly, since A is sdp, then there exists an orthogonal matrix Q such that

$$A = Q\Lambda Q^\top, \qquad \Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_m), \tag{4}$$

where $0 < \lambda_1 \le \cdots \le \lambda_m \le \omega$ are the eigenvalues of A. Problem (1) can be cast in first order form, by setting

$$y = \begin{pmatrix} q \\ p \end{pmatrix}, \qquad p = A^{-1}\dot{q}, \qquad J_2 = \begin{pmatrix} & 1 \\ -1 & \end{pmatrix}, \qquad \tilde{f}(y) = \begin{pmatrix} A^{-1}\nabla f(q) \\ 0 \end{pmatrix}, \tag{5}$$

and denoting $p_0 = A^{-1}\dot{q}_0$, as

$$\dot{y} = J_2 \otimes A\, y + J_2 \otimes I_m \tilde{f}(y), \qquad y(0) = y_0 \equiv \begin{pmatrix} q_0 \\ p_0 \end{pmatrix}. \tag{6}$$

Consequently, the used arguments naturally extends to first order Hamiltonian problems in the form

$$\dot{y} = J\left[Ay + \nabla f(y)\right], \qquad y(0) = y_0 \in \mathbb{R}^{2m}, \qquad J = J_2 \otimes I_m, \tag{7}$$

with Hamiltonian

$$H(y) = \frac{1}{2}y^\top Ay + f(y), \tag{8}$$

and $A \in \mathbb{R}^{2m \times 2m}$ a spd matrix formally still satisfying (3).

The numerical solution of this class of problems presents two important difficulties. On one side, the fact that the matrix JA of the linear term in (7) has some large pure imaginary eigenvalues makes the system stiff oscillatory and the stepsize must be small enough to guarantee that ωh, with ω the largest eigenvalue modulus, belongs to the stability domain of the method. This can be very restrictive unless the numerical method has adequate stability properties.

On the other hand, the solution $y(t)$ of (7) can also be highly oscillatory which means that their derivatives can behave as the powers of ω, that is, $y^{(j)}(t) = \mathcal{O}(\omega^j)$. Since the numerical methods are usually based on Taylor expansions and for a method of order p the leading term of its local truncation error is of the order of $h^{p+1}y^{(p+1)}(t_n)$, then the error will behave as $(\omega h)^{p+1}$. Consequently, to have a small error in the numerical solution, the stepsize must again satisfy $\omega h < 1$. Otherwise, even though the error can be bounded, the error will not decrease with the stepsize h until $\omega h < 1$. That is, its observed numerical order can be zero for larger values of the stepsize (see, e.g., [8]).

The numerical solution of highly oscillatory problems has been the subject of many researches in the last years. Trigonometric methods [28, 30, 31, 34, 33, 38] are a class of explicit exponential

2

methods intended for second order problems. They provide a bounded numerical solution and can be symplectic. However, even though they integrate exactly linear problems, they can present instabilities for $\omega h \geq 2\pi$ and their numerical order is zero unless $\omega h \leq 1$. Functionally fitted methods provide a generalization of the previous approach which, however, still suffers from stepsize restrictions (see, e.g., [42, Thm. 3.2]).

Exponential methods [38] have been proved to be efficient for the solution of highly oscillatory problems coming from the semidiscretization of semilinear Hamiltonian PDEs [37, 27]. These methods require the computation of matrix exponentials, that can be expensive if the order of the method is high or if the method advances with a variable stepsize strategy. Again, to ensure the right numerical order, the stepsize must be small.

When the problem has a single high-frequency, the numerical methods can exploit such a feature for efficiently solving it. This is the case for example of the so called multi-revolution methods [26, 43] and the averaging or stroboscopic methods [25, 29, 30]. These classes of methods combine outer integrators, that adapt to the scale of the slow components, with inner integrators, that adapt to the fast components. They have proved to be efficient with problems that have one high frequency, but we are not aware of any result with problems with several high frequencies. Related to this methods are multiscale techniques [2] and parareal methods.

A different approach is used in [8] to solve second order problems with one high frequency. These methods use a combination of Taylor and Fourier expansions to follow the high oscillations and integrate exactly linear problems, which make them stable and they can integrate with large stepsizes. Nevertheless, when multiple frequencies are present and/or they are not a priori known, the problem is more difficult.

From the point of view of the stability, Gauss-Legendre Runge-Kutta methods are a very good option because they are A-stable and P-stable, that is, the stability function at pure imaginary points has unit modulus. Therefore, they are stable for any stepsize h and, moreover, they are zero dissipative. They have high order of accuracy and they are symplectic, an important property when solving Hamiltonian problems [44, 36]. A more general class of methods are Hamiltonian Boundary Value Methods (HBVMs) [13, 12]. They are also A-stable, P-stable and, moreover, they are energy–conserving. These two classes of methods have the inconvenient that they are fully implicit and can require a high computational cost. Also, since they are based on Taylor expansions, the local truncation error will depend on $(\omega h)^{p+1}$ and the stepsize could be restricted by accuracy reasons. Nevertheless, HBVMs can be also regarded as spectral methods along the orthonormal Legendre polynomial basis [16], and this opens a new perspective in their application. Early references on the usage of spectral methods in time are [4, 5, 39, 40], and a further related reference is [47]. In this paper, we shall use HBVMs as spectral methods also considering a very efficient implementation of such methods, when solving problem (7) (or (1)).

With these premises, the paper is organized as follows: in Section 2 we consider the use of the Legendre basis to define a spectral method in time; in Section 3 we see that, by approximating the involved integrals via a Gaussian quadrature, one retrieves HBVMs; in Section 4 we consider a very efficient nonlinear iteration for solving the generated discrete problems; in Section 5 we present some numerical tests; at last, in Section 6 we report a few conclusions and remarks.

2 Spectral methods

We shall here consider, as a suitable orthonormal basis for representing the solution of (6) (or, more in general, of (7)) on the interval $[0, h]$, the orthonormal basis for $L^2[0, 1]$ functions given by

Legendre polynomials:

$$P_i \in \Pi_i, \qquad \int_0^1 P_i(x) P_j(x) \mathrm{d}x = \delta_{ij}, \qquad \forall i, j = 0, 1, \ldots. \tag{9}$$

We start considering the approximation of the linear part in (6). For this purpose, we need the following preliminary results.

Lemma 1 *Let $\omega h > 0$. For $s = 1, 2, \ldots,$ set*

$$g(s, \omega h) := \sqrt{\frac{(2s+1)\pi}{\omega h}} \left| J_{s+\frac{1}{2}} \left(\frac{\omega h}{2} \right) \right|, \tag{10}$$

where $J_{s+\frac{1}{2}}(\cdot)$ is the Bessel function of the first kind. Then, for all $s = 1, 2, \ldots$:

$$\left| \int_0^1 P_s(c) \cos(\omega h c) \mathrm{d}c \right|, \left| \int_0^1 P_s(c) \sin(\omega h c) \mathrm{d}c \right| \le g(s, \omega h). \tag{11}$$

<u>Proof</u>   The proof easily derives from [32, Eq. (9)], which states that

$$\left| \int_0^1 P_s(c) \cos(\omega h c) \mathrm{d}c \right|^2 + \left| \int_0^1 P_s(c) \sin(\omega h c) \mathrm{d}c \right|^2 = g(s, \omega h)^2. \ \Box \tag{12}$$

In Figure 1 we plot the values of the integrals in (11), numerically computed via a high-order Gauss-Legendre quadrature formula (solid lines and dashed lines, respectively, for the two integrals at the left-hand side in (11)), together with the bound provided by the function defined in (10) (dotted lines), for $\omega h = 1, 5, 10$. As one may see, they are in very good agreement, until round-off error level is reached, so that the numerical quadrature becomes ineffective.

Lemma 2 *For $\omega h > 0$, and $s \gg 1$, the function $g(s, \omega h)$ defined in (10) is an increasing function of ωh and a decreasing function of s.*

<u>Proof</u>   From [32, Eq. (20)], one derives that, for $\omega h > 0$ and $s \gg 1$,

$$g(s, \omega h) = \sqrt{\frac{(2s+1)\pi}{\omega h}} \left| J_{s+\frac{1}{2}} \left(\frac{\omega h}{2} \right) \right| \approx \sqrt{\frac{e}{2(2s+1)}} \left(\frac{e \omega h}{2(2s+1)} \right)^s. \tag{13}$$

The latter function, in turn, is an increasing function of ωh, for fixed s, and a decreasing function of s, for any fixed $\omega h > 0$ and all s such that $2(2s+1) > e\omega h$. $\Box$

We observe that also the result of Lemma 2 is clearly confirmed by the plots in Figure 1.

Let now consider the approximation of the linear part of the problem (6) on the interval $[0, h]$, i.e.

$$\dot{y} = J_2 \otimes A \, y, \qquad y(0) = y_0 := \begin{pmatrix} q_0 \\ p_0 \end{pmatrix}. \tag{14}$$

The following result holds true.

4
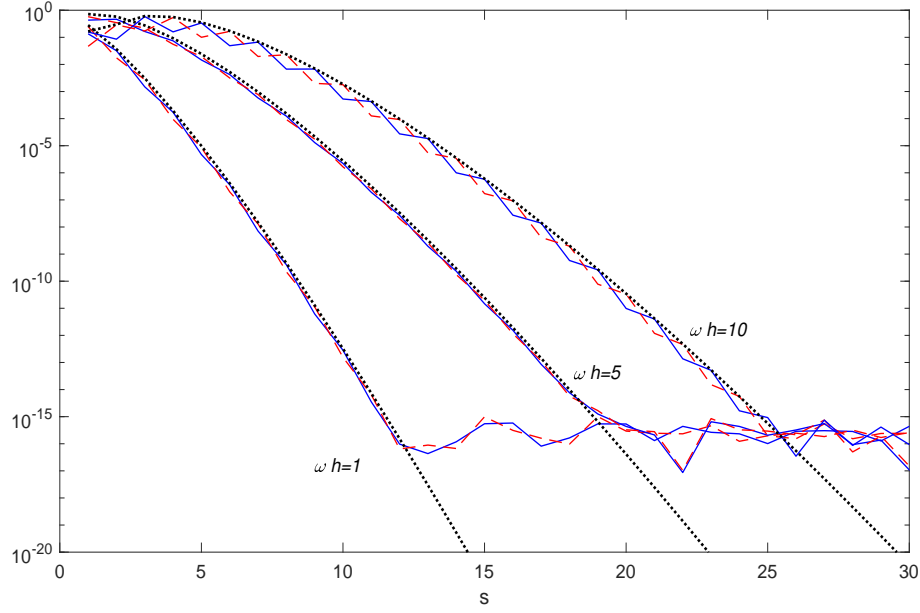
Figure 1: Values of the integrals at the left-hand side in (11), numerically evaluated by a high order Gauss-Legendre formula (solid and dashed lines, respectively), along with their bound $g(s, \omega h)$ defined in (10) (dotted lines), for $\omega h = 1, 5, 10$.

Theorem 1 *The solution of (14) satisfies*

$$y(ch) \equiv e^{J_2 \otimes Ahc} y_0 = y_0 + h \sum_{j \geq 0} \int_0^c P_j(x) \mathrm{d}x \, \gamma_j(y), \qquad c \in [0, 1], \tag{15}$$

with

$$\gamma_j(y) = J_2 \otimes A \int_0^1 P_j(\tau) y(\tau h) \mathrm{d}\tau, \qquad j = 0, 1, 2, \dots. \tag{16}$$

Moreover,

$$y(h) = y_0 + h\gamma_0(y). \tag{17}$$

<u>Proof</u> According to [16], the right-hand side of (14) can be expanded, on the interval $[0, h]$, along the Legendre basis:

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

with the coefficients $\gamma_j(y)$ clearly given by (16), because of the orthonormality conditions (9). Integration side by side of such equation, and imposing the initial condition, then gives (15). At last, (17) follows from (15), by considering that

$$\int_0^1 P_j(x) \mathrm{d}x = \delta_{j0}. \ \square$$

5

By further considering that

$$e^{J_2 \otimes A} = \begin{pmatrix} \cos(A) & \sin(A) \\ -\sin(A) & \cos(A) \end{pmatrix},$$

one has then that y in (15) is also given by

$$y(ch) = \begin{pmatrix} \cos(Ahc) & \sin(Ahc) \\ -\sin(Ahc) & \cos(Ahc) \end{pmatrix} y_0, \qquad c \in [0,1], \tag{18}$$

namely, y is obtained as the combination of sines and cosines, with frequencies not larger than

$$\omega = \|A\|. \tag{19}$$

As a consequence of Theorem 1 and (19), one has the following.

Criterion 1 *When using a finite precision arithmetic with machine epsilon u, and with reference to the function g defined in (10), the series at the rigt-hand side in (15) can be truncated at a convenient value $s_0 - 1$ such that:*

$$g(s_0, \omega h) < u \cdot \max_{j < s_0} g(j, \omega h). \tag{20}$$

The derivation of such criterion is as follows. From the results of Lemmas 1 and 2, (4), (16), (18), and with reference to the function g defined in (10),

$$\begin{aligned} \gamma_j(y) &= J_2 \otimes A \int_0^1 P_j(c) \begin{pmatrix} \cos(Ahc) & \sin(Ahc) \\ -\sin(Ahc) & \cos(Ahc) \end{pmatrix} y_0 \, dc \\ &= (J_2 \otimes Q) \underbrace{\int_0^1 P_j(c) \begin{pmatrix} \Lambda \cos(\Lambda hc) & \Lambda \sin(\Lambda hc) \\ -\Lambda \sin(\Lambda hc) & \Lambda \cos(\Lambda hc) \end{pmatrix} dc}_{=: M_j} (I_2 \otimes Q^\top) y_0 \\ &=: (J_2 \otimes Q) M_j (I_2 \otimes Q^\top) y_0, \end{aligned}$$

where I_2 is the 2×2 identity matrix. By considering the 2-norm, we shall then consider the approximation

$$\|\gamma_j(y)\|_2 \sim \|M_j\|_2 \|y_0\|_2. \tag{21}$$

Moreover, from (12), one has:

$$\begin{aligned} \|M_j\|_2^2 &= \max_{\lambda \in \sigma(A)} \lambda^2 \left[\left(\int_0^1 P_j(\tau) \cos(\lambda h \tau) d\tau \right)^2 + \left(\int_0^1 P_j(\tau) \sin(\lambda h \tau) d\tau \right)^2 \right] \\ &= \max_{\lambda \in \sigma(A)} [\lambda g(j, \lambda h)]^2 \leq [\omega g(j, \omega h)]^2. \end{aligned} \tag{22}$$

Clearly, the last inequality in (22) becomes an equality, if in (19) one considers the $\|\cdot\|_2$. By taking into account that, according to Lemma 2, for $j \gg 1$ the function $g(j, \omega h)$ is a decreasing function of j, and considering that

$$\left| \int_0^c P_j(x) dx \right| \leq \frac{1}{\sqrt{2j+1}}, \qquad c \in [0,1], \qquad j \geq 0,$$

6

Table 1: Values of the parameter s_0 in (20), as a function of ωh, for the double precision IEEE.

ωh	0.1	0.5	1	5	10	25	50	75	100
s_0	9	11	13	20	26	40	59	76	93

one can neglect the terms in the series (15), starting from the index s_0 such that

$$\|\gamma_{s_0}(y)\|_2 < u \cdot \max_{j<s_0} \|\gamma_j(y)\|_2.$$

The criterion (20) then follows from (21) and (22), by considering the estimate

$$\|\gamma_j(y)\|_2 \sim \omega \, g(j,\omega h)\|y_0\|_2. \quad \square$$

By means of Criterion 1, from (15) one obtains

$$y(ch) \ \dot{=} \ y_{s_0}(ch) \ := \ y_0 + h \sum_{j=0}^{s_0-1} \int_0^c P_j(x)\mathrm{d}x \, \gamma_j(y_{s_0}), \qquad c \in [0,1], \tag{23}$$

where $\gamma_j(y_{s_0})$ is formally still defined by (16), by replacing y with y_{s_0}. In the above expression and hereafter, $\dot{=}$ means "equal within round-off error level". In fact, neither the terms starting from s_0 would be taken into account by the used finite precision arithmetic, nor they could be numerically reliably computed, as is confirmed by the plots in Figure 1.

In addition to this, from Criterion 1, one clearly obtains that s_0 is provided by a function, say φ_u, of ωh and also depending on the used machine epsilon u:

$$s_0 = \varphi_u(\omega h). \tag{24}$$

In Table 1 we list a few values of s_0, depending on the product ωh, when considering the double precision IEEE. Moreover, in Figure 2 we plot $\varphi_u(\omega h)$ versus ωh, for the double precision IEEE. From the figure, one has that

$$\varphi_u(\omega h) \approx 24 + 0.7 \cdot \omega h, \qquad \omega h \gg 1.$$

Next, let us consider the complete problem (5)–(6). By assuming the ansatz

$$\nabla f(q(t)) \ \sim \ \cos(\nu A t)\tilde{q}_0, \qquad t \in [0,h], \tag{25}$$

for suitables $\tilde{q}_0 \in \mathbb{R}^m$ and $\nu > 1$.[1] Consequently, by the nonlinear variation of constants formula, one obtains that the solution of (6), on the interval $[0,h]$, can be approximately expressed as

$$q(ch) \ \sim \ \cos(Ach)q_0 + \sin(Ach)p_0 - hA^{-1}\int_0^c \sin(Ah(c-\tau))\cos(\nu Ah\tau)\mathrm{d}\tau \, \tilde{q}_0, \tag{26}$$

$$p(ch) \ \sim \ -\sin(Ach)q_0 + \cos(Ach)p_0 - hA^{-1}\int_0^c \cos(Ah(c-\tau))\cos(\nu Ah\tau)\mathrm{d}\tau \, \tilde{q}_0, \quad c \in [0,1].$$

---

[1]I.e., locally ∇f approximately behaves as a polynomial of degree ν.
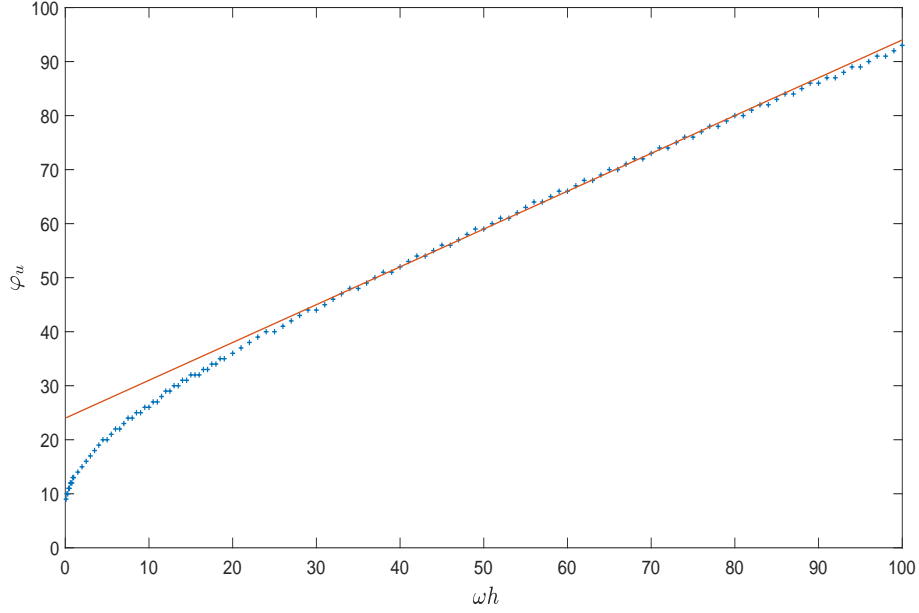
Figure 2: function φ_u versus ωh, for the double precision IEEE. For $\omega h \gg 1$ one has that $\varphi_u(\omega h) \approx 24 + 0.7 \cdot \omega h$.

Further, by considering that, by the Werner formulae,

$$\sin(Ah(c-\tau))\cos(\nu Ah\tau) = \frac{1}{2}\left[\sin\left(Ah(c+(\nu-1)\tau)\right) + \sin\left(Ah(c-(\nu+1)\tau)\right)\right],$$

$$\cos(Ah(c-\tau))\cos(\nu Ah\tau) = \frac{1}{2}\left[\cos\left(Ah(c+(\nu-1)\tau)\right) + \cos\left(Ah(c-(\nu+1)\tau)\right)\right],$$

so that

$$\int_0^c \sin(Ah(c-\tau))\cos(\nu Ah\tau)\mathrm{d}\tau$$

$$= \frac{1}{2}\int_0^c \left[\sin\left(Ah(c+(\nu-1)\tau)\right) + \sin\left(Ah(c-(\nu+1)\tau)\right)\right]\mathrm{d}\tau$$

$$= A^{-1}h^{-1}\frac{1}{\nu^2-1}\left[\cos\left(hAc\right) - \cos\left(hA\nu c\right)\right],$$

and, similarly,

$$\int_0^c \cos(Ah(c-\tau))\cos(\nu Ah\tau)\mathrm{d}\tau$$

$$= \frac{1}{2}\int_0^c \left[\cos\left(Ah(c+(\nu-1)\tau)\right) + \cos\left(Ah(c-(\nu+1)\tau)\right)\right]\mathrm{d}\tau$$

$$= A^{-1}h^{-1}\frac{1}{\nu^2-1}\left[\nu\sin\left(hA\nu c\right) - \sin\left(hAc\right)\right],$$

the following result is proved.

8

Theorem 2 *Under the ansatz (25) the solution (26) of (5)–(6) has an oscillatory behavior, with component in the maximum frequency proportional to:*

$$\frac{A^{-2}}{\nu^2 - 1}\left[\cos\left(hA\nu c\right) - \nu \sin\left(hA\nu c\right)\right]\tilde{q}_0, \qquad c \in [0, 1]. \tag{27}$$

As a consequence of the previous result, by expanding again $y(t) = (q(t), p(t))^\top$ in (6) along the Legendre basis,

$$y(ch) \;=\; y_0 + h\sum_{j\geq 0}\int_0^c P_j(x)\mathrm{d}x\,\psi_j(y), \qquad c \in [0, 1], \tag{28}$$

with

$$\psi_j(y) = \int_0^1 P_j(\tau)\left[J_2 \otimes A\,y(\tau h)) + J_2 \otimes I_m\tilde{f}(y(\tau h))\right]\mathrm{d}\tau, \qquad j = 0, 1, 2, \ldots, \tag{29}$$

one derives the following criterion, which generalises the one given by Criterion 1, for the linear problem (14), to the whole nonlinear problem (5)–(6).

Criterion 2 *When using a finite precision arithmetic with machine epsilon u, and with reference to the function g defined in (10), the series at the rigt-hand side in (28) can be truncated at a convenient value $s-1$ such that:*

$$g(s, \nu\omega h) < u \cdot \max_{j < s} g(j, \nu\omega h). \tag{30}$$

Consequently, one obtains full machine accuracy by considering, in place of (28):

$$y(ch) \;\doteq\; y_s(ch) \;:=\; y_0 + h\sum_{j=0}^{s-1}\int_0^c P_j(x)\mathrm{d}x\,\psi_j(y_s), \qquad c \in [0, 1], \tag{31}$$

where the coefficients $\psi_j(y_s)$ are formally defined as in (29), by replacing y by y_s. We observe that, similarly to what seen for the parameter s_0, now

$$s = \varphi_u(\nu\omega h),$$

where φ_u is the same function defined in (24).[2]

Finally, we observe that, when solving problem (7), instead of (6), one would formally obtain the same relation (31), with the coefficients $\psi_j(y)$ defined as:

$$\psi_j(y) = \int_0^1 P_j(\tau)J\left[Ay(\tau h) + \nabla f(y(\tau h))\right]\mathrm{d}\tau, \qquad j = 0, 1, 2, \ldots, \tag{32}$$

in place of (29). Also in such a case, the Criteria 1 and 2 continue formally to hold.

In the next section, we study the properties of the numerical method obtained by approximating to full machine accuracy the integrals appearing in (29) or (32) by means of a suitable quadrature rule. In particular, we shall consider a Gaussian quadrature based at the k Legendre abscissae (thus, of order $2k$) where, in order to guarantee full machine accuracy, when using the IEEE double precision, we choose

$$k = \max\{s + 2, 20\}, \tag{33}$$

with s defined according to (30). However, more refined choices could in principle be used.

---

[2]Clearly, when $\nu = 1$, then s coincides with s_0, as defined in (20).

3 Hamiltonian Boundary Value Methods (HBVMs)

In this section we illustrate the application of Hamiltonian Boundary Value Methods (HBVMs) for solving problem (7), which is more general than (6). HBVMs form a class of energy-conserving Runge-Kutta methods which has been studied in a series of papers for the numerical solution of Hamiltonian problems [13, 14, 15, 16, 9, 19]. They have been also extended along a number of directions [17, 18, 11, 7, 10, 1, 3] (we also refer to the recent monograph [12]).

In more details, a HBVM(k, s) method is the k-stage Runge-Kutta method defined by the Butcher tableau

$$\frac{\boldsymbol{c} \mid \mathcal{I}_s \mathcal{P}_s^\top \Omega}{\boldsymbol{b}^\top}, \qquad \boldsymbol{b} = \left(\begin{array}{ccc} b_1 & \ldots & b_k \end{array} \right)^\top, \quad \boldsymbol{c} = \left(\begin{array}{ccc} c_1 & \ldots & c_k \end{array} \right)^\top, \qquad (34)$$

where, with reference to the Legendre polynomial basis defined in (9), (c_i, b_i) are the abscissae and weights of the Gauss-Legendre quadrature formula of order $2k$ (i.e., $P_k(c_i) = 0$, $i = 1, \ldots, k$), and

$$
\begin{aligned}
\mathcal{P}_s &= \left(\begin{array}{ccc} P_0(c_1) & \ldots & P_{s-1}(c_1) \\ \vdots & & \vdots \\ P_0(c_k) & \ldots & P_{s-1}(c_k) \end{array} \right) \in \mathbb{R}^{k \times s}, \\[1em]
\mathcal{I}_s &= \left(\begin{array}{ccc} \int_0^{c_1} P_0(x)\mathrm{d}x & \ldots & \int_0^{c_1} P_{s-1}(x)\mathrm{d}x \\ \vdots & & \vdots \\ \int_0^{c_k} P_0(x)\mathrm{d}x & \ldots & \int_0^{c_k} P_{s-1}(x)\mathrm{d}x \end{array} \right) \in \mathbb{R}^{k \times s}, \qquad (35) \\[1em]
\Omega &= \left(\begin{array}{ccc} b_1 & & \\ & \ddots & \\ & & b_k \end{array} \right) \in \mathbb{R}^{k \times k}.
\end{aligned}
$$

The following result is known to hold for such methods (see, e.g., [16, 12]).

Theorem 3 *For all $k \geq s$, the HBVM(k, s) method (34):*

- *is symmetric and has order $2s$;*

- *when $k = s$ it reduces to the (symplectic) s-stage Gauss collocation method;*

- *it is energy-conserving, when applied for solving Hamiltonian problems with a polynomial Hamiltonian of degree not larger than $2k/s$;*

- *for general and suitably regular Hamiltonians, the Hamiltonian error per step is $O(h^{2k+1})$.*

Remark 1 *Because of the result of Theorem 3, one has that an exact energy-conservation is always obtained, by choosing k large enough, in the polynomial case. Moreover, even in the non-polynomial case, a practical energy-conservation can always be gained, by choosing k large enough so that the Hamiltonian error falls within the round-off error level. This, in turn, doesn't affect too much the computational cost of the method, as is shown in Section 4 (see also [14, 9, 12]). In particular, the choice (33) of k will always provide us with a practical energy conservation.*

For sake of completeness, we also mention that one may consider the limit as $k \to \infty$ of HBVMs [13], thus obtaining a continuous-stage Runge-Kutta method (see also [13, 35] and [12, Chapter 3.5]).

A few properties of the matrices defined in (35) are here recalled, for later use.

Lemma 3 *For all $k \geq s$, one has:*

$$\mathcal{P}_s^\top \Omega \mathcal{I}_s = X_s \equiv \begin{pmatrix} \xi_0 & -\xi_1 & & \\ \xi_1 & 0 & \ddots & \\ & \ddots & \ddots & -\xi_{s-1} \\ & & \xi_{s-1} & 0 \end{pmatrix}, \qquad \xi_i = \frac{1}{2\sqrt{|4i^2-1|}}, \quad i = 0, 1, \ldots. \tag{36}$$

Moreover,

$$\det(X_s) = \begin{cases} \prod_{i=1}^{s/2} \xi_{2i-1}^2, & s \text{ even}, \\ \\ \prod_{i=0}^{(s-1)/2} \xi_{2i}^2, & s \text{ odd}, \end{cases}$$

so that matrix X_s is nonsingular, for all $s = 1, 2, \ldots$.

<u>Proof</u>   See, e.g., [12, Lemmas 3.6 and 3.7]. $\square$

Let now study the application of a HBVM(k, s) method, $k > s$, for solving (7). In so doing, by setting

$$Y := \begin{pmatrix} Y_1 \\ \vdots \\ Y_k \end{pmatrix}, \qquad \nabla f(Y) := \begin{pmatrix} \nabla f(Y_1) \\ \vdots \\ \nabla f(Y_k) \end{pmatrix}, \qquad e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^k,$$

with Y the stage vector of the method, one obtains the discrete problem of block dimension k,

$$Y = e \otimes y_0 + h \mathcal{I}_s \mathcal{P}_s^\top \Omega \otimes J \left[I_k \otimes A \, Y + \nabla f(Y) \right], \tag{37}$$

and the following approximation to $y(h)$:

$$y_1 = y_0 + h \sum_{i=1}^{k} b_i J [A Y_i + \nabla f(Y_i)]. \tag{38}$$

Nevertheless, by considering that

$$Y_i = y_0 + h \sum_{j=0}^{s-1} \int_0^{c_i} P_j(x) \mathrm{d}x \, \psi_j \ =: \ \sigma_s(c_i h), \qquad i = 1, \ldots, k, \tag{39}$$

with $\sigma_s \in \Pi_s$, and

$$\begin{aligned} \psi_j &= \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) J [A Y_\ell + \nabla f(Y_\ell)] \\ &\equiv \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) J [A \sigma_s(c_\ell h) + \nabla f(\sigma_s(c_\ell h))], \qquad j = 0, \ldots, s-1, \end{aligned} \tag{40}$$

comparison of (39)–(40) with (31), provides us with the following result.

11

Theorem 4 *Provided that the quadrature is exact within full machine accuracy, with reference to (31) and (32), one has:*

$$\sigma_s(ch) \doteq y_s(ch) \quad \Rightarrow \quad \psi_j \doteq \psi_j(y_s), \quad j = 0, \dots, s-1.$$

Remark 2 *The result of Theorem 4, with s chosen according to Criterion 2, clearly shows that for such a method the concept of order doesn't apply, since it will always provide the maximum possible accuracy, for the used finite precision arithmetic.*

In order to improve the computational efficiency, instead of directly solving the discrete problem (37), having block dimension k, we shall consider a more convenient formulation of it [14]. In more details, by setting

$$\psi := \begin{pmatrix} \psi_0 \\ \vdots \\ \psi_{s-1} \end{pmatrix} \equiv \mathcal{P}_s^\top \Omega \otimes J \left[(I_k \otimes A) Y + \nabla f(Y) \right], \tag{41}$$

from (37) one obtains,

$$Y = e \otimes y_0 + h\mathcal{I}_s \otimes I_{2m}\psi,$$

which, plugged into (41), provides the following new discrete problem, clearly equivalent to (37):

$$G(\psi) := \psi - \mathcal{P}_s^\top \Omega \otimes J \left[(I_k \otimes A) (e \otimes y_0 + h\mathcal{I}_s \otimes I_{2m}\psi) + \nabla f (e \otimes y_0 + h\mathcal{I}_s \otimes I_{2m}\psi) \right] = 0. \tag{42}$$

Once it has been solved, the new approximation (38) turns out to be given by:

$$y_1 = y_0 + h\psi_0 \equiv \sigma_s(h) \doteq y_s(h). \tag{43}$$

We observe that the advantage of solving the discrete problem (42) instead of the stage problem (37) is twofold:

1. the problem (42) has block dimension s, independently of k;

2. its numerical solution via a simplified Newton-type iteration is very efficient.

The last point is elucidated in the next section.

4 Efficient implementation of the methods

The use of the simplified Newton method for solving (42) is described, by virtue of (36), by the following iteration:

$$
\begin{aligned}
&\text{initialize} &&\psi^0 \\
&\text{for } \ell &&= \quad 0, 1, \dots : \\
& &&\text{solve} \quad [I - hX_s \otimes J(A + \nabla^2 f(y_0))]\Delta^\ell = -G(\psi^\ell) \\
& &&\text{set} \quad \psi^{\ell+1} = \psi^\ell + \Delta^\ell \\
&\text{end}
\end{aligned}
\tag{44}
$$

with I the identity matrix of dimension $s \cdot 2m$.

12

Firstly, we notice that, by virtue of (3), we can consider the approximation

$$A + \nabla^2 f(y_0) \approx A, \tag{45}$$

thus obtaining a coefficient matrix which is *constant* and is the same for all the integration steps. Consequently, the iteration (44) simplifies to:

$$
\begin{aligned}
&\text{initialize} \quad\quad \boldsymbol{\psi}^0 \\
&\quad \text{for } \ell \;=\; 0, 1, \ldots : \\
&\quad\quad\quad \text{solve} \quad [I - hX_s \otimes JA]\,\Delta^\ell = -G(\boldsymbol{\psi}^\ell) \\
&\quad\quad\quad\; \text{set} \quad \boldsymbol{\psi}^{\ell+1} = \boldsymbol{\psi}^\ell + \Delta^\ell \\
&\quad\; \text{end}
\end{aligned}
\tag{46}
$$

This iteration can be further simplified by using a Newton-splitting *blended iteration*. This technique, at first devised in [6, 20], has then be generalized [22, 23] and implemented in the computational codes BiM [21] and BiMD [24] for stiff ODE/DAE IVPs. It has been also considered for HBVMs [14] and is implemented in the Matlab code HBVM [12]. The novelty, in the present case, is due to the approximation (45), which makes it extremely efficient. As a result, the iteration (46) modifies as follows:

$$
\begin{aligned}
&\text{initialize} \quad\quad \boldsymbol{\psi}^0 \\
&\quad \text{for } \ell \;=\; 0, 1, \ldots : \\
&\quad\quad\quad \text{set} \quad \boldsymbol{\eta}^\ell = -G(\boldsymbol{\psi}^\ell) \\
&\quad\quad\quad \text{set} \quad \boldsymbol{\eta}_1^\ell = \left[(\rho_s X_s^{-1}) \otimes I_{2m} \right] \boldsymbol{\eta}^\ell \\
&\quad\quad\quad \text{set} \quad \boldsymbol{u}^\ell = [I_s \otimes \Sigma]\,(\boldsymbol{\eta}^\ell - \boldsymbol{\eta}_1^\ell) \\
&\quad\quad\quad \text{set} \quad \Delta^\ell = [I_s \otimes \Sigma]\,(\boldsymbol{\eta}_1^\ell + \boldsymbol{u}^\ell) \\
&\quad\quad\quad \text{set} \quad \boldsymbol{\psi}^{\ell+1} = \boldsymbol{\psi}^\ell + \Delta^\ell \\
&\quad\; \text{end}
\end{aligned}
\tag{47}
$$

Here, according to [20, 23, 14],

$$\rho_s = \min_{\lambda \in \sigma(X_s)} |\lambda|, \tag{48}$$

and

$$\Sigma = (I_{2m} - h\rho_s JA)^{-1}. \tag{49}$$

Consequently, we notice that one needs only to compute *once and for all* matrix Σ (or factorize Σ^{-1}), having the same size as that of the continuous problem.

Secondly, in order to gain convergence for relatively large values of ωh, it is important to choose an appropriate starting value $\boldsymbol{\psi}^0$ in (47). For this purpose, we use the solution of the associated homogeneous problem (14). This latter can be conveniently computed, by virtue of Criterion 1 (see (20)) by the HBVM(s_0, s_0) method (i.e., the s_0-stage Gauss collocation method). Consequently, by repeating similar steps as above, by defining the vectors

$$
\boldsymbol{\gamma} = \begin{pmatrix} \gamma_0 \\ \vdots \\ \gamma_{s_0-1} \end{pmatrix} \in \mathbb{R}^{s_0 2m}, \qquad \tilde{\boldsymbol{e}} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{s_0},
$$

13

and the matrices, with a structure similar to (35),

$$\tilde{\mathcal{P}}_{s_0} = (P_{j-1}(\tilde{c}_i)), \ \ \tilde{\mathcal{I}}_{s_0} = \left(\int_0^{\tilde{c}_i} P_{j-1}(x)\mathrm{d}x \right), \ \ \tilde{\Omega} = \begin{pmatrix} \tilde{b}_1 & & \\ & \ddots & \\ & & \tilde{b}_{s_0} \end{pmatrix} \in \mathbb{R}^{s_0 \times s_0},$$

with $(\tilde{c}_i, \tilde{b}_i)$ the abscissae and weights of the Gauss-Legendre quadrature of order $2s_0$, one solves the discrete problem

$$\tilde{G}(\boldsymbol{\gamma}) := \boldsymbol{\gamma} - \tilde{\mathcal{P}}_{s_0}^\top \tilde{\Omega} \otimes JA \left[\tilde{\boldsymbol{e}} \otimes y_0 + h\tilde{\mathcal{I}}_{s_0} \otimes I_{2m}\boldsymbol{\gamma} \right] = \boldsymbol{0}.$$

This, in turn, can be done by means of the following approximate *blended* iteration, similar to (47):

$$
\begin{aligned}
\text{initialize} \qquad & \boldsymbol{\gamma}^0 = \boldsymbol{0} \\
\text{for } \ell \ = \ & 0, 1, \dots : \\
& \text{set} \quad \tilde{\boldsymbol{\eta}}^\ell = -\tilde{G}(\boldsymbol{\gamma}^\ell) \\
& \text{set} \quad \tilde{\boldsymbol{\eta}}_1^\ell = \left[(\rho_{s_0} X_{s_0}^{-1}) \otimes I_{2m} \right] \tilde{\boldsymbol{\eta}}^\ell \\
& \text{set} \quad \tilde{\boldsymbol{u}}^\ell = [I_{s_0} \otimes \Sigma] \, (\tilde{\boldsymbol{\eta}}^\ell - \tilde{\boldsymbol{\eta}}_1^\ell) \\
& \text{set} \quad \tilde{\Delta}^\ell = [I_{s_0} \otimes \Sigma] \, (\tilde{\boldsymbol{\eta}}_1^\ell + \tilde{\boldsymbol{u}}^\ell) \\
& \text{set} \quad \boldsymbol{\gamma}^{\ell+1} = \boldsymbol{\gamma}^\ell + \tilde{\Delta}^\ell \\
\text{end} &
\end{aligned}
\qquad (50)
$$

where matrix $X_{s_0} \in \mathbb{R}^{s_0 \times s_0}$ is defined according to (36), and the same matxix Σ defined in (49) can be used.[3] Once this has been done, the following initialization can be conveniently used for the iteration (47):

$$\boldsymbol{\psi}^0 = \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{0} \end{pmatrix}, \qquad \text{with} \qquad \boldsymbol{0} \in \mathbb{R}^{(s-s_0)2m}. \qquad (51)$$

Definition 1 *We shall refer to the method defined by the iterations (47) and (50)–(51) to as* spectral HBVM *with parameters* k, s, s_0, *in short SHBVM(k, s, s_0).*

5 Numerical tests

In this section we consider a few numerical tests, aimed at assessing the effectiveness of the SHBVM(k, s, s_0) method, implementing the spectral-Legendre methods described in Section 2. This will be done by comparing such methods with known existing ones. In particular, we consider the methods below specified, where the first three methods can be used only for problems in the form (1).

---

[3]The approximation stems from the fact that in (49) ρ_s is used in place of ρ_{s_0}.

Methods:

Störmer-Verlet : the Störmer-Verlet method [36], of order 2;

Gautschi : the trigonometric fitted method of Gautschi [33], of order 2;

Deuflhard : the improved trigonometric fitted method of Deuflhard [31], of order 2;

Expode : the exponential integrator code EXPODE [41], which is based on exponential integrators. We have used the routine `exprk` based on an exponential RK methods of order 4 described in [37];

Gauss : the s-stage Gauss method, $s = 1, \ldots, 4$, of order $2s$;

SHBVM : the spectral method SHBVM(k, s, s_0).

All numerical tests have been performed on a laptop with a 2.2 GHz dual-core $i7$ and 8GB of memory, running Matlab (R2017b).

Duffing equation

The first test problem is the Duffing equation:

$$\ddot{q} = -(\kappa^2 + \beta^2)q + 2\kappa^2 q^3, \quad t > 0, \qquad q(0) = 0, \quad \dot{q}(0) = \beta, \tag{52}$$

with Hamiltonian

$$H(q, p) = \frac{1}{2}\left[p^2 + (\kappa^2 + \beta^2)q - \kappa^2 q^4\right], \qquad p = \dot{q}. \tag{53}$$

In such a case, the solution is known to be

$$q(t) = \text{sn}(\beta t, M), \qquad p(t) = \beta \text{cn}(\beta t, M)\text{dn}(\beta t, M), \tag{54}$$

with sn, cn, dn the elliptic Jacobi functions with elliptic modulus $M = \kappa^2/\beta^2$. We consider here the parameters:

$$\kappa = 7, \qquad \beta = 500. \tag{55}$$

At the best of our knowledge, so far such large values of the two parameters have never been considered for benchmarking. As a matter of fact, the chosen values are 100 times larger than those used, e.g., in [42] (i.e., $\kappa = 0.07$, $\beta = 5$). For solving such problem, we compare the methods listed above by performing N integration steps with a constant stepsize $h = 20/N$. In Table 2 we list the obtained result for the first four methods (i.e., SV, Gautschi, Deuflhard, and Expode), for increasing values of N, in terms of:

- execution time;

- maximum error on q (e_q);

- maximum error on p (e_p);

- maximum error on the Hamiltonian (e_H).

From the listed results, one verifies that all methods have an execution time proportional to the number of steps, N, and very similar for the first three methods. Moreover, as expected, the accuracy of the Störmer-Verlet method is less than that of the Gautschi method which, in turn, is less than that of the Deuflhard method, even though the latter method evidently suffers from cancellation errors, as the stepsize is decreased. One also verifies that, for such methods the condition $\omega h < 1$ is satisfied. This latter condition is not required by Expode which, however, though 4-th order accurate, appears to be not competitive w.r.t. the Deuflhard method.

Next, in Table 3, we list the obtained results for the s-stage Gauss method, $s = 1, \ldots, 4$. They have been implemented through the SHBVM(s, s, s) method. From the listed results, one verifies that, for all methods, the higher the order of the method, the smaller the execution time. Moreover, the higher-order methods are also competitive w.r.t. the previous methods. In particular, Expode appears to be not competitive w.r.t. the higher-order Gauss methods. For this reason, by also considering that its implementation is relatively difficult for higher dimensional problems, we shall not consider such method further.

At last, in Table 4 we list the obtained results by using the spectral method SHBVM(k, s, s_0), where (s_0, s, k) have been computed according to (20), (30), and (33), respectively, by considering $\omega = \sqrt{\kappa^2 + \beta^2}$ and $\nu = 3$. As one may see, all errors are very small and almost constant, according to the analysis made in Sections 2–3 and to what observed in Remark 2 (in particular, the Hamiltonian error is within the round-off error level). Moreover, also the execution times, which are very small, are almost equal, though there is a small positive trend, when reducing the stepsize $h = 20/N$ (i.e., when increasing N). In order to make this latter statement more precise, we computed the numerical solution by using the stepsizes

$$h = \frac{20}{N}, \qquad N = 800, 900, 1000, \ldots, 5000.$$

In so doing, though the error remains approximately constant when increasing N, nevertheless, the execution time increases, as is shown in Figure 3.[4] From that figure, one has that a value $N \approx 1000$ (to which corresponds a value $\omega h \approx 10$) seems to be the most efficient, when using SHBVM(k, s, s_0). Clearly, such a method is the most efficient, among those here considered.

Fermi-Pasta-Ulam problem

The second test problem is the well-known Fermi-Pasta-Ulam problem (see, e.g., [12, 36]), which models a physical system composed by $2m$ unit point masses disposed in series along a line, chained together by alternating weak nonlinear springs and stiff linear springs. In particular, we assume that the force exerted by the nonlinear springs is proportional to the cube of the displacement of the associated masses (cubic springs). The endpoints of the external springs are taken fixed. We denote by $q_1, q_2, \ldots, q_{2m}$ the displacements of the masses from their rest points and define the conjugate momenta as $p_i = \dot{q}_i$, $i = 1, \ldots, 2m$. The resulting problem is Hamiltonian and is defined by the energy function

$$H(q, p) = \frac{1}{2} \sum_{i=1}^{m} (p_{2i-1}^2 + p_{2i}^2) + \frac{1}{2} \sum_{i=1}^{m} \omega_i^2 (q_{2i} - q_{2i-1})^2 + \sum_{i=0}^{m} (q_{2i+1} - q_{2i})^4, \tag{56}$$

---

[4]Note in Figure 3 that doubling the number N of steps does not imply doubling the CPU time. This is due to the fact that when the number of steps is doubled, the number of stages decreases, since ωh is halved.
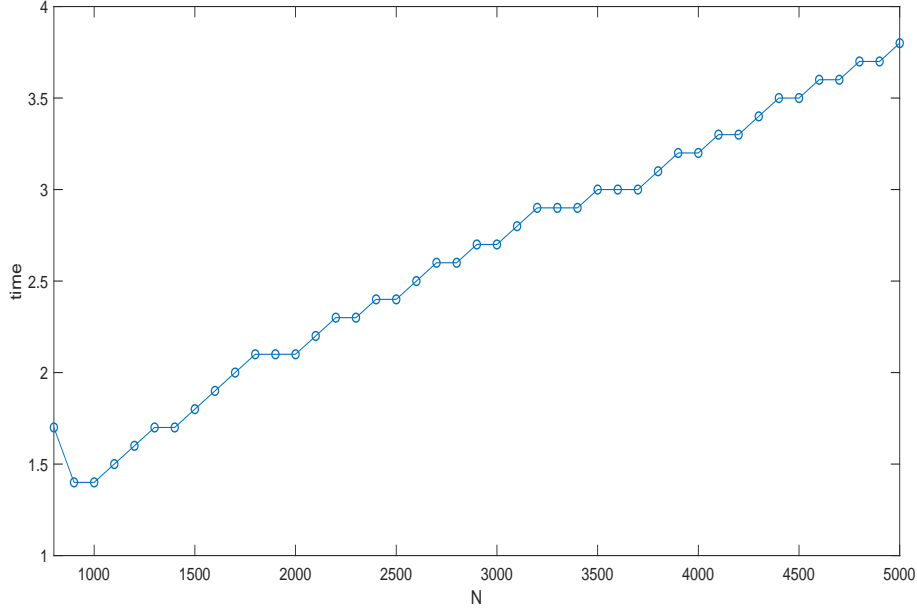
Figure 3: Problem (52) solved by the SHBVM method with stepsize $h = 20/N$. Execution time versus N.

with $q_0 = q_{2m+1} = 0$, and with the coefficients ω_i, ruling the stiffness of the linear strings, which may be large, thus yielding a stiff oscillatory problem. We consider the parameters

$$m = 8, \qquad \omega_i = 10^{i-1}, \qquad \omega_{4+i} = (\pi - 4 + i)\cdot 10^{4-i}, \qquad i = 1, 2, 3, 4, \tag{57}$$

with the initial conditions

$$q_i = \frac{i-1}{2(2m-1)}, \qquad p_i = 0, \qquad i = 1, \dots, 2m, \tag{58}$$

which evidently provide a (severe) multi-frequency highly-oscillatory problem. A reference solution at $T = 10$ has been computed by using a high-order Gauss method with a suitably small time-step. All methods have been used to perform N integration steps, with a constant stepsize $h = 10/N$.

In Table 5 we list the obtained results for the Störmer-Verlet, Gautschi, and Deuflhard methods, which, as one may see, require very small stepsizes. Moreover, as for the previous problem, the Deuflhard method suffers from cancelation errors, for the smallest stepsizes used.

In Table 6, we list the obtained results for the s-stage Gauss methods, $s = 1, \dots, 4$. As in the previous example, only the higher order methods are relatively efficient.

At last, in Table 7, we list the obtained results by using the SHBVM(k, s, s_0) method, where (s_0, s, k) have been computed according to (20), (30), and (33), respectively, by considering $\omega = 10^3$ and $\nu = 3$. From the listed results, one deduces that the Hamiltonian error is within the round-off error level, and the solution error is always very small, independently of the value of N used, even though it seems that the optimal value of N is 900 (to which corresponds a value $\omega h \approx 11$), with the minimum solution error (among those displayed) and an almost minimum execution time

17

Table 2: Numerical results when solving problem (52) by using a stepsize $h = 20/N$.

Störmer-Verlet method							
N	time	e_q	rate	e_p	rate	e_H	rate
1.25e6	18.4	2.65e-02	—	1.30e 01	—	8.00e-05	—
2.5e6	35.1	6.63e-03	2.0	3.24e 00	2.0	2.00e-05	2.0
5e6	72.3	1.66e-03	2.0	8.11e-01	2.0	5.00e-06	2.0
1e7	143.5	4.15e-04	2.0	2.04e-01	2.0	1.25e-06	2.0
2e7	274.5	1.04e-04	2.0	5.10e-02	2.0	3.13e-07	2.0
Gautschi method							
N	time	e_q	rate	e_p	rate	e_H	rate
1.25e6	17.2	7.83e-06	—	3.83e-03	—	7.17e-09	—
2.5e6	34.3	1.96e-06	2.0	9.57e-04	2.0	1.83e-09	2.0
5e6	68.2	4.89e-07	2.0	2.39e-04	2.0	5.43e-10	1.8
1e7	137.2	1.22e-07	2.0	6.02e-05	2.0	3.91e-10	**
2e7	275.8	3.05e-08	2.0	1.51e-05	2.0	4.65e-10	**
Deuflhard method							
N	time	e_q	rate	e_p	rate	e_H	rate
6.25e5	8.2	3.26e-08	—	1.60e-05	—	1.67e-08	—
1.25e6	16.5	1.09e-09	4.9	4.39e-07	5.2	4.20e-09	2.0
2.5e6	32.4	5.12e-09	**	2.50e-06	**	1.05e-09	2.0
5e6	65.2	1.21e-07	**	5.91e-05	**	3.07e-10	1.8
1e7	129.8	5.00e-07	**	2.46e-04	**	8.03e-10	**
Expode method							
N	time	e_q	rate	e_p	rate	e_H	rate
6.25e3	5.4	8.58e-03	—	2.81e 00	—	7.43e-04	—
1.25e4	10.5	6.07e-04	3.8	1.98e-01	3.8	1.68e-06	8.8
2.5e4	20.8	3.85e-05	4.0	1.25e-02	4.0	8.15e-08	4.4
5e4	45.0	2.43e-06	4.0	8.16e-04	3.9	4.24e-09	4.3
1e5	82.8	1.63e-07	3.9	7.67e-05	3.4	9.60e-10	**
2e5	165.0	2.61e-08	**	1.22e-05	**	8.30e-10	**

(approximately 8 sec). Consequently, this SHBVM method turns out to be the most efficient, among those here considered. This fact is confirmed by the plots in Figure 4, showing the execution time and the solution error versus N (upper and lower plot, respectively), when using a stepsize

$$h = \frac{10}{N}, \qquad N = 500, 600, 700, \ldots, 5000.$$

In fact, even though a smaller error (of the order of 10^{-11}) is obtained for $N > 2500$, a larger execution time is required.

Nonlinear Schrödinger equation

At last, we consider a highly-oscillatory problem, in the form (7), deriving from the space semi-discretization of the nonlinear Schödinger equation [3], which we sketch below in a much simplified

Table 3: Numerical results when solving problem (52) by using a stepsize $h = 20/N$.

1-stage Gauss method							
N	time	e_q	rate	e_p	rate	e_H	rate
1.25e6	192.6	5.32e-02	—	2.60e 01	—	3.14e-09	—
2.5e6	365.6	1.33e-02	2.0	6.51e 00	2.0	7.84e-10	2.0
5e6	700.2	3.33e-03	2.0	1.63e 00	2.0	1.96e-10	2.0
1e7	1378.9	8.32e-04	2.0	4.09e-01	2.0	4.93e-11	2.0
2e7	2739.2	2.08e-04	2.0	1.02e-01	2.0	1.24e-11	2.0
2-stage Gauss method							
N	time	e_q	rate	e_p	rate	e_H	rate
2e5	62.9	8.63e-05	—	4.08e-02	—	2.72e-11	—
4e5	114.0	5.40e-06	4.0	2.59e-03	4.0	9.35e-12	**
8e5	210.1	3.39e-07	4.0	1.65e-04	4.0	2.36e-12	**
16e5	403.8	2.12e-08	4.0	1.03e-05	4.0	5.44e-13	**
3-stage Gauss method							
N	time	e_q	rate	e_p	rate	e_H	rate
2.5e4	11.8	3.98e-04	—	1.29e-01	—	4.09e-10	—
5e4	20.1	6.27e-06	6.0	2.11e-03	5.9	6.08e-12	6.1
1e5	35.1	9.85e-08	6.0	4.66e-05	5.5	1.51e-10	**
2e5	65.4	1.47e-09	6.1	6.97e-07	6.1	1.09e-10	**
4-stage Gauss method							
N	time	e_q	rate	e_p	rate	e_H	rate
1.25e4	7.3	6.35e-05	—	2.07e-02	—	4.68e-10	—
2.5e4	11.6	2.53e-07	8.0	8.20e-05	8.0	9.05e-13	9.0
5e4	20.3	9.93e-10	8.0	3.34e-07	7.9	2.36e-13	**
1e5	36.1	1.60e-10	**	7.38e-08	**	2.16e-10	**

form.[5] The problem we want to solve is

$$i\psi_t(x,t) + \psi_{xx}(x,t) + \kappa|\psi(x,t)|^2\psi(x,t) = 0, \qquad (x,t) \in [0, 2\pi] \times [0,5], \qquad (59)$$
$$\psi(x,0) = e^{irx},$$

coupled with periodic boundary conditions. The solution of this "toy" problem is readily seen to be given by

$$\psi(x,t) = e^{i(rx - \mu t)}, \qquad \mu = r^2 - \kappa.$$

In particular, in order to simplify the arguments, we consider the values:

$$r = 20, \qquad \kappa = \frac{\pi}{10} \qquad \Rightarrow \qquad \mu \approx 399.7.$$

To begin with, we separate the real and imaginary parts of the solution,

$$\psi(x,t) = u(x,t) + iv(x,t),$$

---

[5]This is due to the particular initial condition considered.

Table 4: Numerical results when solving problem (52) by using a stepsize $h = 20/N$.

		SHBVM(k, s, s_0) method			
N	time	e_q	e_p	e_H	(s_0, s, k)
800	1.7	3.96e-10	7.70e-08	4.44e-16	(29,50,52)
900	1.4	5.47e-11	1.20e-08	2.22e-16	(28,47,49)
1000	1.4	2.70e-11	1.28e-09	4.44e-16	(26,44,46)
1100	1.5	5.90e-11	2.35e-08	2.22e-16	(25,42,44)
1200	1.6	1.08e-11	1.63e-09	3.33e-16	(25,40,42)
1300	1.7	2.63e-11	5.07e-09	4.44e-16	(24,39,41)
1400	1.7	2.41e-11	2.50e-09	4.44e-16	(23,37,39)
1500	1.8	1.77e-11	6.40e-09	4.44e-16	(22,36,38)

and consider their expansions (in space) along an orthonormal Fourier basis,

$$u(x,t) = \sum_{j=0}^{r} [c_j(x)\xi_j(t) + s_j(x)\eta_j(t)], \quad v(x,t) = \sum_{j=0}^{r} [c_j(x)\alpha_j(t) + s_j(x)\beta_j(t)], \quad x \in [0, 2\pi],$$

with

$$c_j(x) = \sqrt{\frac{2 - \delta_{j0}}{2\pi}} \cos(jx), \qquad s_j(x) = \sqrt{\frac{2 - \delta_{j0}}{2\pi}} \sin(jx), \qquad j = 0, \dots, r,$$

δ_{j0} the Kronecker delta, and unknown time-dependent coefficients $\xi_j(t), \eta_j(t), \alpha_j(t), \beta_j(t)$.[6] Subsequently, by setting the vectors

$$\boldsymbol{q}(t) = \begin{pmatrix} \xi_0(t) \\ \vdots \\ \xi_r(t) \\ \eta_1(t) \\ \vdots \\ \eta_r(t) \end{pmatrix}, \quad \boldsymbol{p}(t) = \begin{pmatrix} \alpha_0(t) \\ \vdots \\ \alpha_r(t) \\ \beta_1(t) \\ \vdots \\ \beta_r(t) \end{pmatrix}, \quad \boldsymbol{w}(t) = \begin{pmatrix} c_0(t) \\ \vdots \\ c_r(t) \\ s_1(t) \\ \vdots \\ s_r(t) \end{pmatrix} \in \mathbb{R}^{2r+1}, \quad \boldsymbol{y} = \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{pmatrix},$$

and the matrix

$$D = \begin{pmatrix} 0 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & r & \\ & & & & & r \end{pmatrix} \in \mathbb{R}^{2r+1 \times 2r+1},$$

we obtain, by setting $e_i \in \mathbb{R}^2$ the i-th unit vector, $i = 1, 2$,

$$|\psi(x,t)|^2 = \left[\left((e_1 \otimes \boldsymbol{w}(x))^\top \boldsymbol{y}(t) \right)^2 + \left((e_2 \otimes \boldsymbol{w}(x))^\top \boldsymbol{y}(t) \right)^2 \right],$$

---

[6] For this simple problem, actually only the coefficients $\xi_r(t), \eta_r(t), \alpha_r(t), \beta_r(t)$ are nonzero.

Table 5: Numerical results when solving problem (56)–(58) by using a stepsize $h = 10/N$.

Störmer-Verlet method					
$10^{-4}N$	time	e_y	rate	e_H	rate
16	7.8	1.71e00	—	9.25e-04	—
32	15.5	9.63e-01	0.8	2.33e-04	2.0
64	30.9	2.60e-01	1.9	5.85e-05	2.0
128	62.7	6.60e-02	2.0	1.46e-05	2.0
256	125.0	1.66e-02	2.0	3.66e-06	2.0
Gautschi method					
$10^{-4}N$	time	e_y	rate	e_H	rate
1	0.5	7.75e-05	—	1.79e-07	—
2	1.0	1.79e-05	2.1	4.08e-08	2.1
4	2.0	4.39e-06	2.0	1.03e-08	2.0
8	4.3	1.09e-06	2.0	2.54e-09	2.0
16	7.9	2.73e-07	2.0	6.26e-10	2.0
32	15.8	6.81e-08	2.0	1.48e-10	2.1
64	31.5	1.71e-08	2.0	6.58e-11	**
128	62.9	6.32e-09	**	2.55e-10	**
Deuflhard method					
$10^{-4}N$	time	e_y	rate	e_H	rate
1	0.8	4.93e-05	—	4.17e-07	—
2	1.5	1.05e-05	2.2	9.73e-08	2.1
4	3.1	2.48e-06	2.1	2.38e-08	2.0
8	6.2	6.11e-07	2.0	5.92e-09	2.0
16	11.9	1.52e-07	2.0	1.44e-09	2.0
32	24.0	2.80e-07	**	5.62e-10	**
64	47.9	1.55e-06	**	8.75e-10	**
128	96.1	4.07e-06	**	2.63e-09	**

thus arriving at the Hamiltonian system of ODEs

$$\dot{\boldsymbol{y}} = J_2 \otimes D^2\,\boldsymbol{y} - \kappa \int_0^{2\pi} J_2 \otimes (\boldsymbol{w}(x)\boldsymbol{w}(x)^\top)\,\boldsymbol{y}\left[\left((e_1 \otimes \boldsymbol{w}(x))^\top \boldsymbol{y}(t)\right)^2 + \left((e_2 \otimes \boldsymbol{w}(x))^\top \boldsymbol{y}(t)\right)^2\right]\mathrm{d}x, \quad (60)$$

(here J_2 is the same matrix as that defined in (5)), with Hamiltonian

$$H(\boldsymbol{y}) = \frac{1}{2}\left(\boldsymbol{y}^\top I_2 \otimes D^2\,\boldsymbol{y} - \frac{\kappa}{2}\int_0^{2\pi}\left[\left((e_1 \otimes \boldsymbol{w}(x))^\top \boldsymbol{y}(t)\right)^2 + \left((e_2 \otimes \boldsymbol{w}(x))^\top \boldsymbol{y}(t)\right)^2\right]^2\mathrm{d}x\right). \quad (61)$$

This latter function, in turn, is equivalent to the Hamiltonian functional defining (59), i.e.,[7]

$$\mathcal{H}[\psi] = \frac{1}{2}\int_0^{2\pi} |\psi_x|^2 - \frac{\kappa}{2}|\psi|^4 \mathrm{d}x.$$

---

[7]The reader is referred to [3] for full details.

Table 6: Numerical results when solving problem (56)–(58) by using a stepsize $h = 10/N$.

1-stage Gauss method					
$10^{-4}N$	time	e_y	rate	e_H	rate
32	107.2	2.16e 00	—	4.90e-11	—
64	213.5	5.44e-01	2.0	1.27e-11	1.9
128	428.2	1.34e-01	2.0	3.33e-12	1.9
256	855.2	3.32e-02	2.0	2.04e-12	**
2-stage Gauss method					
$10^{-4}N$	time	e_y	rate	e_H	rate
4	27.7	1.42e-01	—	5.43e-12	—
8	50.0	8.83e-03	4.0	4.51e-13	3.6
16	95.4	5.52e-04	4.0	3.04e-13	**
32	187.6	3.45e-05	4.0	3.89e-13	**
64	354.9	2.16e-06	4.0	6.08e-13	**
128	661.5	1.35e-07	4.0	2.74e-13	**
3-stage Gauss method					
$10^{-4}N$	time	e_y	rate	e_H	rate
1	11.1	2.52e-01	—	1.35e-11	—
2	18.9	4.00e-03	6.0	2.46e-13	5.8
4	33.0	6.30e-05	6.0	1.33e-13	**
8	56.6	9.86e-07	6.0	1.21e-13	**
16	106.4	1.54e-08	6.0	1.27e-13	**
32	211.9	2.76e-10	5.8	4.17e-13	**
4-stage Gauss method					
$10^{-4}N$	time	e_y	rate	e_H	rate
1	11.8	9.96e-04	—	4.71e-13	—
2	19.9	3.98e-06	8.0	1.27e-13	**
4	34.8	1.56e-08	8.0	8.79e-14	**
8	59.6	5.28e-11	8.2	1.36e-13	**
16	109.8	8.00e-12	**	1.12e-13	**

At last, in order to derive a fully discrete problem, the integrals in (60)–(61) are (exactly) computed via a composite trapezoidal rule, at the abscissae

$$x_\ell = \ell \frac{2\pi}{m}, \qquad \ell = 0, \ldots, m, \qquad m = 4r + 1. \tag{62}$$

Because of the structure of the problem, to implement the SHBVM method we shall consider the parameters

$$\omega = 400 \equiv r^2 = \|D^2\|, \qquad \nu = 1,$$

so that, in this case, the two parameters s_0 and s (see (20) and (30), respectively) coincide. The parameter k, in turn, is computed, as usual, from (33).

In Table 8, we list the obtained results for the s-stage Gauss methods, $s = 1, \ldots, 4$, by using a time-step $h = 5/N$, for increasing values of N. As in the previous examples, only the higher order

22

Table 7: Numerical results when solving problem (56)–(58) by using a stepsize $h = 10/N$.

SHBVM(k, s, s_0) method				
N	time	e_y	e_H	(s_0, s, k)
500	11.4	2.13e-07	1.78e-15	(36,66,68)
600	9.0	2.95e-09	1.78e-15	(33,59,61)
700	8.7	2.77e-09	1.78e-15	(31,54,56)
800	9.1	2.05e-10	2.00e-15	(29,50,52)
900	8.3	2.95e-11	1.78e-15	(28,47,49)
1000	7.9	8.28e-08	1.78e-15	(26,44,46)
1100	8.0	2.33e-08	1.78e-15	(25,42,44)
1200	8.9	1.46e-09	2.00e-15	(25,40,42)
1300	9.0	1.20e-09	1.78e-15	(24,39,41)
1400	8.1	2.22e-10	1.78e-15	(23,37,39)
1500	8.8	1.56e-09	2.00e-15	(22,36,38)

methods are relatively efficient, even though the Hamiltonian error is always very small.

At last, in Table 9, we list the obtained results by using the SHBVM(k, s, s) method,[8] by using the time-steps:

$$h = 5/N, \qquad N = 200, 250, 300, 350, 400, 450, 500.$$

From the listed results, one deduces that the Hamiltonian error is always within the round-off error level. Moreover, also the solution error is always very small and uniform, independently of the considered value of N, even though it seems that the optimal value of N is 250 (to which corresponds a value $\omega h \approx 8$), with an almost minimum solution error (among those displayed) and a minimum execution time (2.7 sec). Consequently, with such a value of N, SHBVM(26,24,24) is clearly the most efficient method, among those here considered.

6 Concuding remarks

In this paper, we have seen that spectral methods in time can be very efficient for solving, possibly multi-frequency, highly-oscillatory Hamiltonian problems. In particular, their implementation via a slight modification of Hamiltonian Boundary Value Methods, provides a very efficient energy-conserving procedure, able to solve such problems up to the maximum accuracy allowed by the used finite-precision arithmetic. Some numerical tests on some severe tests problems duly confirm the theoretical achievements.

It is worth noticing that the proposed approach could be in principle used also for solving general ODE-IVPs, thus providing a spectrally accurate method of solution in time. This will be the subject of future investigations.

---

[8]In fact, in this case $s_0 = s$.

Table 8: Numerical results when solving problem (60)–(61) by using a stepsize $h = 5/N$.

1-stage Gauss method				
$10^{-3}N$	time	e_y	rate	e_H
16	6.3	2.01e 02	—	8.55e-15
32	10.9	6.65e 01	1.6	1.22e-14
64	21.7	1.69e 01	2.0	2.10e-14
128	44.6	4.23e 00	2.0	3.85e-14
256	88.7	1.06e 00	2.0	5.02e-14
512	174.4	2.64e-01	2.0	2.35e-14
1024	290.4	6.61e-02	2.0	1.38e-13

2-stage Gauss method				
$10^{-3}N$	time	e_y	rate	e_H
4	4.0	1.77e 01	—	1.56e-13
8	6.8	1.12e 00	4.0	7.75e-14
16	11.1	7.02e-02	4.0	6.95e-14
32	19.5	4.39e-03	4.0	8.44e-15
64	34.7	2.75e-04	4.0	1.78e-14
128	68.9	1.72e-05	4.0	2.82e-14
256	120.9	1.07e-06	4.0	2.49e-14

3-stage Gauss method				
$10^{-4}N$	time	e_y	rate	e_H
2	3.1	1.97e 00	—	2.00e-14
4	5.1	3.17e-02	6.0	2.26e-14
8	8.1	4.99e-04	6.0	1.93e-14
16	14.7	7.82e-06	6.0	7.33e-15
32	26.4	1.22e-07	6.0	7.11e-15
64	48.6	2.20e-09	5.8	1.42e-14
128	84.4	3.28e-10	**	2.24e-14

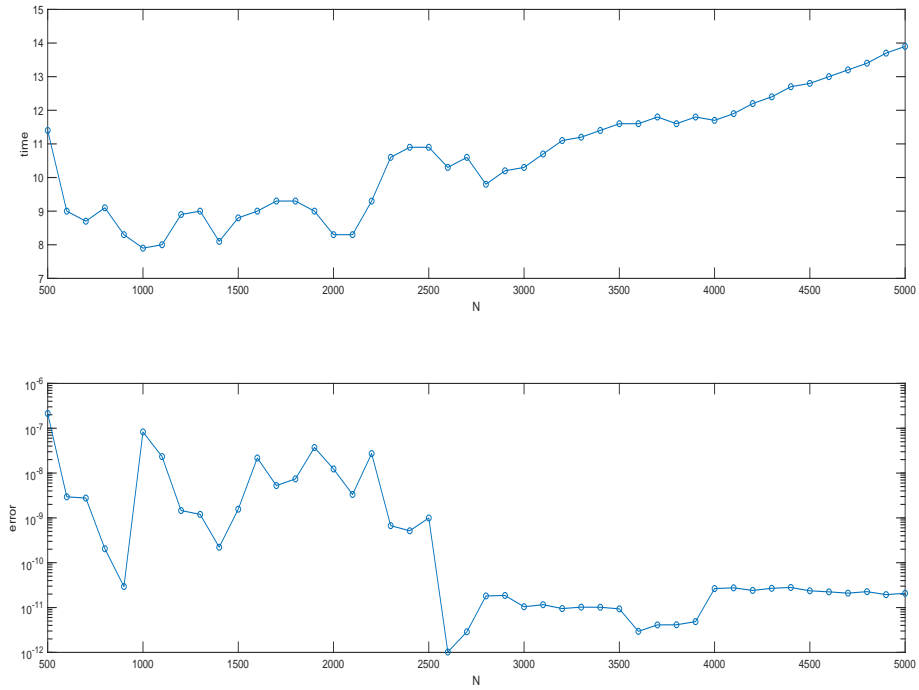| 4-stage Gauss method				
$10^{-3}N$	time	e_y	rate	e_H
1	2.1	1.84e 00	—	3.97e-13
2	3.0	7.88e-03	7.9	5.48e-14
4	4.7	3.14e-05	8.0	3.87e-14
8	8.2	1.23e-07	8.0	9.99e-15
16	13.7	4.33e-10	8.2	5.22e-15
32	25.4	1.01e-10	**	1.40e-14

Figure 4: Problem (56)–(58) solved by the spectral method with stepsize $h = 10/N$. Upper plot: execution time versus N. Lower-plot: solution error versus N.

References

[1] P. Amodio, L. Brugnano, F. Iavernaro. Energy-conserving methods for Hamiltonian Boundary Value Problems and applications in astrodynamics. *Adv. Comput. Math.* **41** (2015) 881–905.

[2] G. Ariel, B. Engquist, S. Kim, Y. Lee, R. Tsai. A Multiscale Method for Highly Oscillatory Dynamical Systems Using a Poincaré Map Type Technique, *J. Sci. Comput.*, **54** (2013) 247–268.

[3] L. Barletti, L. Brugnano, G. Frasca Caccia, F. Iavernaro. Energy-conserving methods for the nonlinear Schrödinger equation. *Appl. Math.Comput.* **318** (2018) 3–18.

[4] P. Betsch, P. Steinmann. Inherently Energy Conserving Time Finite Elements for Classical Mechanics. *J. Comp. Phys.* **160** (2000) 88–116.

[5] C.L. Bottasso. A new look at finite elements in time: a variational interpretation of Runge–Kutta methods. *Appl. Numer. Math.* **25** (1997) 355–368.

[6] L. Brugnano. Blended Block BVMs (B$_3$VMs): A Family of Economical Implicit Methods for ODEs. *J. Comput. Appl. Math.* **116** (2000) 41–62.

[7] L. Brugnano, M. Calvo, J.I. Montijano, L. Ràndez. Energy preserving methods for Poisson systems. *J. Comput. Appl. Math.* **236** (2012) 3890–3904.

Table 9: Numerical results when solving problem (60)–(61) by using a stepsize $h = 5/N$.

SHBVM(k,s,s) method				
N	time	e_y	e_H	(s,k)
200	2.7	1.50e-10	4.44e-16	(26,28)
250	2.7	4.94e-11	4.44e-16	(24,26)
300	2.9	2.43e-10	4.44e-16	(22,24)
350	3.0	1.43e-10	4.44e-16	(21,23)
400	3.1	4.83e-11	3.33e-16	(20,22)
450	3.5	4.33e-11	4.44e-16	(19,21)
500	3.6	5.53e-11	4.44e-16	(19,21)

[8] L. Brugnano, M. Calvo, J. I. Montijano, L. Rández. Fourier Methods for Oscillatory Differential Problems with a Constant High Frequency. *AIP Conference Proc.* **1863** (2017) 020003.

[9] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Efficient implementation of Gauss collocation and Hamiltonian Boundary Value Methods. *Numer. Algorithms* **65** (2014) 633–650.

[10] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Energy conservation issues in the numerical solution of the semilinear wave equation. *Appl. Math.Comput.* **270** (2015) 842–870

[11] L. Brugnano, F. Iavernaro. Line Integral Methods which preserve all invariants of conservative problems. *J. Comput. Appl. Math.* **236** (2012) 3905–3919.

[12] L. Brugnano, F. Iavernaro. *Line Integral Methods for Conservative Problems*. Chapman et Hall/CRC, Boca Raton, FL, 2016.

[13] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian Boundary Value Methods (Energy Preserving Discrete Line Integral Methods). *JNAIAM. J. Numer. Anal. Ind. Appl. Math.* **5**, No. 1-2 (2010) 17–37.

[14] L. Brugnano, F. Iavernaro, D. Trigiante. A note on the efficient implementation of Hamiltonian BVMs. *J. Comput. Appl. Math.* **236** (2011) 375–383.

[15] 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. *Appl. Math.Comput.* **218** (2012) 8053–8063.

[16] L. Brugnano, F. Iavernaro, D. Trigiante. A simple framework for the derivation and analysis of effective one-step methods for ODEs. *Appl. Math.Comput.* **218** (2012) 8475–8485.

[17] L. Brugnano, F. Iavernaro, D. Trigiante. A two-step, fourth-order method with energy preserving properties. *Comput. Phys. Commun.* **183** (2012) 1860–1868.

[18] L. Brugnano, F. Iavernaro, D. Trigiante. Energy and QUadratic Invariants Preserving integrators based upon Gauss collocation formulae. *SIAM J. Numer. Anal.* **50**, No. 6 (2012) 2897–2916.

[19] L. Brugnano, F. Iavernaro, D. Trigiante. Analisys of Hamiltonian Boundary Value Methods (HBVMs): a class of energy-preserving Runge-Kutta methods for the numerical solution of polynomial Hamiltonian systems. *Commun. Nonlinear Sci. Numer. Simul.* **20** (2015) 650–667.

[20] L. Brugnano, C. Magherini. Blended Implementation of Block Implicit Methods for ODEs. *Appl. Numer. Math.* **42** (2002) 29–45.

[21] L. Brugnano, C. Magherini. The BiM code for the numerical solution of ODEs. *J. Comput. Appl. Math.* **164-165** (2004) 145–158.

[22] L. Brugnano, C. Magherini. Blended Implicit Methods for solving ODE and DAE problems, and their extension for second order problems. *J. Comput. Appl. Math.* **205** (2007) 777–790.

[23] L. Brugnano, C. Magherini. Recent advances in linear analysis of convergence for splittings for solving ODE problems. *Appl. Numer. Math.* **59** (2009) 542–557.

[24] L. Brugnano, C. Magherini, F. Mugnai. Blended Implicit Methods for the Numerical Solution of DAE Problems. *J. Comput. Appl. Math.* **189** (2006) 34–50.

[25] M.P. Calvo, P. Chartier, A. Murua, J.M. Sanz-Serna. Numerical stroboscopic averaging for ODEs and DAEs. *Appl. Numer. Math.* , **61**, 10 (2011) 1077–1095.

[26] M. Calvo, L.O. Jay, J.I. Montijano, L. Rández. Approximate compositions of a near identity map by multi-revolution Runge-Kutta methods. *Numer. Math.* **4** (2004) 635–666.

[27] B. Cano, A. González-Pachón. Exponential time integration of solitary waves of cubic Schrödinger equation. *Appl. Numer. Math.*, **91** (2015) 26–45.

[28] B. Cano, M.J. Moreta. High-order symmetric multistep cosine methods. *Appl. Numer. Math.* **66** (2013) 30–44.

[29] F. Castella, P. Chartier, E. Fao. An averaging technique for highly oscillatory Hamiltonian problems. *SIAM J. Numer. Math.* **47**, 4 (2009) 2808–2837.

[30] D. Cohen, T. Jahnke, K. Lorenz, C. Lubich. Numerical Integrators for Highly Oscillatory Hamiltonian Systems: A Review. In: *Analysis, Modeling and Simulation of Multiscale Problems*, A. Mielke (eds). Springer, Berlin, Heidelberg (2006) pp. 553–576.

[31] P. Deuflhard. A study of extrapolation methods based on multistep schemes without parasitic solutions. *Z. angew. Math. Phys.* **30** (1979) 177–189.

[32] G.A. Evans, J.R. Webster. A comparison of some methods for the evaluation of highly oscillatory integrals. *J. Comput. Appl. Math.* **112** (1999) 55–69.

[33] W. Gautschi. Numerical integration of ordinary differential equations based on trigonometric polynomials. *Numer. Math.* **3** (1961) 381–397.

[34] B. García-Archilla, J.M. Sanz-Serna, R.D. Skeel. Long-time steps methods for oscillatory differential equations. *SIAM J. Sci. Comput.*,**20**, 3 (1998) 930–963.

[35] E. Hairer. Energy-Preserving Variant of Collocation Methods. *JNAIAM. J. Numer. Anal. Ind. Appl. Math.* **5**, No. 1-2 (2010) 73–84.

[36] E. Hairer, C. Lubich, G. Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations, Second ed.*, Springer, Berlin, 2006.

[37] M. Hochbruck, A. Ostermann. Explicit exponential Runge-Kutta methods for semilinear parabolic problems. *SIAM J. Numer. Anal.* **43** (2005) 1069–1090.

[38] M. Hochbruck, A. Ostermann. Exponential integrators. *Acta Numer.* **19** (2010) 209–286.

[39] B.L. Hulme. One-Step Piecewise Polynomial Galerkin Methods for Initial Value Problems. *Math. Comp.* **26** (1972) 415–426.

[40] B.L. Hulme. Discrete Galerkin and related one-step methods for ordinary differential equations. *Math. Comp.* **26** (1972) 881–891.

[41] G. Jansing. EXPODE – Advanced Exponential Time Integration Toolbox for MATLAB. `arXiv:1404.4580v1 [math.NA]`, 2014

[42] Yu-wen Li, Xinyuan Wu. Functionally fitted energy–preserving methods for solving oscillatory nonlinear Hamiltonian systems. *SIAM J. Numer. Anal.* **54**, No. 4 (2016) 2036–2059.

[43] L.R. Petzold, L.O. Jay, J. Yen. Numerical solution of highly oscillatory ordinary differential equations. *Acta Numer.* **6** (1997) 437–483.

[44] J.M. Sanz-Serna, M.P. Calvo. *Numerical Hamiltonian problems.* Chapman & Hall, London, 1994.

[45] S.S. Shome, E.J. Haug, L.O. Jay. Dual-rate integration using partitioned runge-kutta methods for mechanical systems with interacting subsystems. *Mechanics Based Design of Structures and Machines*, **32**, 3 (2004) 253–282.

[46] T. Solcia, P. Masarati. Multirate simulation of complex multibody systems. In ECCOMAS Thematic Conference, 2011.

[47] W. Tang, Y. Sun. Time finite element methods: a unified framework for numerical discretizations of ODEs. *Appl. Math. Comp.* **219** (2012) 2158–2179.

[48] M.E. Tuckerman, B.J. Berne. Molecular dynamics in systems with multiple time scales: Systems with stiff and soft degrees of freedom and with short and long range forces. *J. Chem. Phys.* **95**, 11 (1991) 8362–8364.