



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

Energy-conserving Hamiltonian Boundary Value Methods for the numerical solution of the Korteweg-de Vries equation

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

Energy-conserving Hamiltonian Boundary Value Methods for the numerical solution of the Korteweg-de Vries equation / Luigi Brugnano, Gianmarco Gurioli, Yajuan Sun. - In: JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS. - ISSN 1879-1778. - STAMPA. - 351:(2019), pp. 117-135.
[10.1016/j.cam.2018.10.014]

Availability:

The webpage <https://hdl.handle.net/2158/1137213> of the repository was last updated on 2019-07-17T15:00:12Z

Published version:

DOI: 10.1016/j.cam.2018.10.014

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)

Energy-conserving Hamiltonian Boundary Value Methods for the numerical solution of the Korteweg–de Vries equation

Luigi Brugnano

*Dipartimento di Matematica e Informatica “U. Dini”, Università di Firenze, Viale
Morgagni 67/A, I-50134 Firenze, Italy.*

Gianmarco Gurioli

*Dipartimento di Matematica e Informatica “U. Dini”, Università di Firenze, Viale
Morgagni 67/A, I-50134 Firenze, Italy.*

Yajuan Sun

*LSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
Beijing 1000190, China.
University of Chinese Academy of Sciences, Beijing 100049, China.*

Abstract

In this paper we study the efficient solution of the well-known Korteweg–de Vries equation, equipped with periodic boundary conditions. A Fourier-Galerkin space semi-discretization at first provides a large-size Hamiltonian ODE problem, whose solution in time is then carried out by means of energy-conserving methods in the HBVM class (Hamiltonian Boundary Value Methods). The efficient implementation of the methods for the resulting problem is also considered and several numerical examples are reported.

Keywords: Korteweg–de Vries equation, Hamiltonian partial differential equations, Hamiltonian problems, energy-conserving methods, Hamiltonian Boundary Value Methods, HBVMs.

Email addresses: `luigi.brugnano@unifi.it` (Luigi Brugnano),
`gianmarco.gurioli@unifi.it` (Gianmarco Gurioli), `sunyj@lsec.cc.ac.cn` (Yajuan Sun)

1. Introduction

In this paper, we consider the numerical solution of the well-known *Korteweg-de Vries (KdV) equation*,

$$\begin{aligned} u_t(x, t) &= \alpha u_{xxx}(x, t) + \beta u(x, t)u_x(x, t), \quad (x, t) \in \Omega := [a, b] \times [0, \infty), \quad (1) \\ \alpha, \beta &\in \mathbb{R}, \quad \alpha\beta \neq 0, \end{aligned}$$

where, as is usual, the subscript denotes the partial derivative w.r.t. the given variable. Typical values of the parameters considered for this equation are, e.g., $\alpha = -1$, $\beta = -6$. The equation (1) is completed with the initial conditions

$$u(x, 0) = u_0(x), \quad x \in [a, b], \quad \text{and} \quad \text{periodic boundary conditions.} \quad (2)$$

Consequently, u_0 will be assumed to be a periodic function, smooth enough (as a periodic function) so that the solution u turns out to be smooth as well.¹ For sake of brevity, when not necessary, we shall hereafter skip the arguments (x, t) for u and its derivatives.

This equation, originally proposed to describe wave propagation on the surface of shallow water, has then been rediscovered as the continuum limit of the Fermi-Pasta-Ulam experiment [54] (see also [3]), and one of its main features is that of possessing *soliton solutions*. It has been the subject, for about half a century, of many investigations both from a theoretical point of view (see, e.g., [26, 5, 44, 38, 45, 35, 29, 36]) and from its numerical approximation. In this regard, besides the first numerical approaches in [49, 1, 50, 7], conservative methods have been developed by using various approaches [33, 25], including Galerkin methods [53, 41, 52, 6, 34, 22], finite difference schemes [2, 55, 42], operator splitting and exponential-type integrators [32, 31], structure and energy-preserving methods [51, 23, 39, 43, 37, 48].

¹ Ideally, in the most favourable case where u is analytic, its n -th Fourier coefficient decays exponentially with n , whereas it decays as n^{-r} if $u \in C^r$. A fast decay of the Fourier coefficients, in turn, is useful in view of what we are going to study in Sections 2 and 3. We refer, e.g., to [38] for more refined regularity results.

From a mathematical point of view, the equation (1) has a *bi-Hamiltonian structure*, since it can be written in Hamiltonian form in two different ways [44]. In particular, we shall consider here the following Hamiltonian formulation,

$$u_t = \mathcal{J} \frac{\delta}{\delta u} \mathcal{H}[u],$$

where $\mathcal{J} = \frac{\partial}{\partial x}$ and $\frac{\delta}{\delta u} \mathcal{H}[u]$ is the functional derivative² of the *Hamiltonian functional*

$$\mathcal{H}[u] = \frac{1}{2} \int_a^b -\alpha(u_x)^2 + \frac{\beta}{3} u^3 \, dx. \quad (3)$$

Consequently, because of the periodic boundary conditions, the Hamiltonian functional turns out to be conserved,

$$\mathcal{H}[u](t) = \mathcal{H}[u](0), \quad \forall t \geq 0. \quad (4)$$

Due to the bi-Hamiltonian structure, there are, however, infinitely many invariants. Among them, the simplest one, whose conservation can be easily derived from (1), is

$$\mathcal{U}[u] = \int_a^b u \, dx, \quad \Rightarrow \quad \mathcal{U}[u](t) = \mathcal{U}[u](0), \quad \forall t \geq 0. \quad (5)$$

In more details, (3) represents the *energy* of the system, whereas (5) is the *mass*. Consequently, the conservation properties (4) and (5) are important for the correct numerical simulation of such problem. In particular, the conservation of the energy will follow from a suitable space semi-discretization, able to preserve the Hamiltonian structure of the problem. For this reason, in this paper we are concerned with the numerical solution of problem (1)–(2), while exactly conserving (3)–(4) and (5).

With this premise, the structure of the paper is as follows: in Section 2 we cast the problem into Hamiltonian form, by considering a Fourier-type expansion in space; next, in Section 3 we consider a semi-discrete problem, which amounts to a large-size Hamiltonian system of ODEs; in Section 4 we sketch the basic

²See any book of calculus of variations, for the definition of functional derivative, e.g., [27]. See also [40, 30].

facts about Hamiltonian Boundary Value Methods (HBVMs), which we shall use to solve the problem in time while conserving the Hamiltonian, and also explaining the details about their efficient implementation for the considered problem; in Section 5 we collect a number of test problems; at last, in Section 6 we report a few concluding remarks.

We conclude this section by stressing the fact that the efficient implementation of the methods is an important feature, when solving the high-dimensional ODE problems derived from the semi-discretization of the PDE.

2. Fourier expansion in space

Since the solution $u(x, t)$ of (1) we look for is periodic in space, we shall consider its space expansion along the following orthonormal basis for periodic functions in $L^2[a, b]$,

$$\begin{aligned} c_j(x) &= \sqrt{\frac{2 - \delta_{j0}}{b - a}} \cos\left(2\pi j \frac{x - a}{b - a}\right), & j = 0, 1, \dots, \\ s_j(x) &= \sqrt{\frac{2}{b - a}} \sin\left(2\pi j \frac{x - a}{b - a}\right), & j = 1, 2, \dots, \end{aligned} \quad (6)$$

with δ_{j0} the Kronecker delta, such that for all allowed values of i and j :

$$\int_a^b c_i(x) c_j(x) dx = \delta_{ij} = \int_a^b s_i(x) s_j(x) dx, \quad \int_a^b c_i(x) s_j(x) dx = 0. \quad (7)$$

Consequently, for suitable time dependent coefficients $q_j(t), p_j(t)$, one has the expansion:

$$u(x, t) = c_0 q_0(t) + \sum_{j \geq 1} [c_j(x) q_j(t) + s_j(x) p_j(t)], \quad (8)$$

where we take into account that (see (6)) $c_0(x) \equiv (b - a)^{-1/2}$. Clearly, from (8) it follows that the periodic boundary conditions are fulfilled for all t . The expansion (8) can be cast in a more compact form, by defining the infinite

vectors

$$\mathbf{c}(x) = \begin{pmatrix} c_1(x) \\ c_2(x) \\ \vdots \end{pmatrix}, \mathbf{s}(x) = \begin{pmatrix} s_1(x) \\ s_2(x) \\ \vdots \end{pmatrix}, \mathbf{q}(t) = \begin{pmatrix} q_1(t) \\ q_2(t) \\ \vdots \end{pmatrix}, \mathbf{p}(t) = \begin{pmatrix} p_1(t) \\ p_2(t) \\ \vdots \end{pmatrix}, \quad (9)$$

as follows:

$$u(x, t) = c_0 q_0(t) + \mathbf{c}(x)^\top \mathbf{q}(t) + \mathbf{s}(x)^\top \mathbf{p}(t). \quad (10)$$

Moreover, we set the vectors

$$\mathbf{c}'(x) = \begin{pmatrix} c'_1(x) \\ c'_2(x) \\ \vdots \end{pmatrix}, \quad \mathbf{s}'(x) = \begin{pmatrix} s'_1(x) \\ s'_2(x) \\ \vdots \end{pmatrix},$$

containing the first derivatives of the basis functions $c_j(x)$ and $s_j(x)$, and similarly the vectors $\mathbf{c}''(x), \mathbf{s}''(x), \mathbf{c}'''(x), \mathbf{s}'''(x)$ with the second and third derivatives, respectively. We also define the vectors

$$\dot{\mathbf{q}}(t) = \begin{pmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \\ \vdots \end{pmatrix}, \quad \dot{\mathbf{p}}(t) = \begin{pmatrix} \dot{p}_1(t) \\ \dot{p}_2(t) \\ \vdots \end{pmatrix},$$

containing the time derivatives of the coefficients $q_j(t)$ and $p_j(t)$, respectively.

In so doing, we can easily compute the partial derivatives of $u(x, t)$:

$$\begin{aligned} u_t(x, t) &= c_0 \dot{q}_0(t) + \mathbf{c}(x)^\top \dot{\mathbf{q}}(t) + \mathbf{s}(x)^\top \dot{\mathbf{p}}(t), \\ u_x(x, t) &= \mathbf{c}'(x)^\top \mathbf{q}(t) + \mathbf{s}'(x)^\top \mathbf{p}(t), \\ u_{xx}(x, t) &= \mathbf{c}''(x)^\top \mathbf{q}(t) + \mathbf{s}''(x)^\top \mathbf{p}(t), \\ u_{xxx}(x, t) &= \mathbf{c}'''(x)^\top \mathbf{q}(t) + \mathbf{s}'''(x)^\top \mathbf{p}(t). \end{aligned} \quad (11)$$

The following results hold true.

Lemma 1. *Let us define the infinite matrix³*

³Hereafter, for all matrices, all the entries not explicitly defined are assumed to be 0.

$$D = \frac{2\pi}{b-a} \begin{pmatrix} 1 & & & \\ & 2 & & \\ & & 3 & \\ & & & \ddots \end{pmatrix}. \quad (12)$$

Then:

$$\begin{aligned} \mathbf{c}'(x) &= -D\mathbf{s}(x), & \mathbf{s}'(x) &= D\mathbf{c}(x), \\ \mathbf{c}''(x) &= -D^2\mathbf{c}(x), & \mathbf{s}''(x) &= -D^2\mathbf{s}(x), \\ \mathbf{c}'''(x) &= D^3\mathbf{s}(x), & \mathbf{s}'''(x) &= -D^3\mathbf{c}(x). \end{aligned} \quad (13)$$

Proof For the first derivatives, one has:

$$c'_j(x) = -\frac{2\pi j}{b-a} s_j(x), \quad s'_j(x) = \frac{2\pi j}{b-a} c_j(x), \quad j = 1, 2, \dots,$$

which, in vector form, can be written as the first line in (13). The proof for the other derivatives is similar. \square

Lemma 2. *With reference to (11) and (5) one has:*

$$q_0(t) \equiv c_0 \mathcal{U}[u](0). \quad (14)$$

Consequently, $q_0(t)$ is constant.

Proof In fact, from (10) one has, by taking into account that $c_0 = (b-a)^{-1/2}$:

$$\mathcal{U}[u](t) := \int_a^b u(x, t) dx = (b-a) c_0 q_0(t) = c_0^{-1} q_0(t), \quad t \geq 0.$$

Consequently, since $\mathcal{U}[u]$ is conserved (see (5)), one has then

$$q_0(t) \equiv q_0 := c_0 \mathcal{U}[u](0), \quad \forall t \geq 0,$$

as required. \square

By virtue of Lemmas 1 and 2, the equations (10)–(11) can be written as (see (2) and (12)–(14)):

$$u(x, t) = \widehat{u}_0 + \mathbf{c}(x)^\top \mathbf{q}(t) + \mathbf{s}(x)^\top \mathbf{p}(t), \quad \widehat{u}_0 := \frac{1}{b-a} \int_a^b u_0(x) dx, \quad (15)$$

and

$$\begin{aligned}
u_t(x, t) &= \mathbf{c}(x)^\top \dot{\mathbf{q}}(t) + \mathbf{s}(x)^\top \dot{\mathbf{p}}(t), \\
u_x(x, t) &= -[D\mathbf{s}(x)]^\top \mathbf{q}(t) + [D\mathbf{c}(x)]^\top \mathbf{p}(t), \\
u_{xx}(x, t) &= -[D^2\mathbf{c}(x)]^\top \mathbf{q}(t) - [D^2\mathbf{s}(x)]^\top \mathbf{p}(t), \\
u_{xxx}(x, t) &= [D^3\mathbf{s}(x)]^\top \mathbf{q}(t) - [D^3\mathbf{c}(x)]^\top \mathbf{p}(t),
\end{aligned} \tag{16}$$

respectively.

Remark 1. *As is clear from (6)–(7), the conservation property (5) is fulfilled by the function $u(x, t)$ defined in (15).*

Lemma 3. *With reference to the notations (9)–(16), one obtains that the problem (1)–(2) can be rewritten as the following formal set of ODEs,⁴*

$$\begin{aligned}
\dot{\mathbf{q}} &= D \left[-\alpha D^2 \mathbf{p} + \frac{\beta}{2} \int_a^b \mathbf{s} \left(\hat{u}_0 + (\mathbf{c}^\top \mathbf{q}) + (\mathbf{s}^\top \mathbf{p}) \right)^2 dx \right], \\
\dot{\mathbf{p}} &= -D \left[-\alpha D^2 \mathbf{q} + \frac{\beta}{2} \int_a^b \mathbf{c} \left(\hat{u}_0 + (\mathbf{c}^\top \mathbf{q}) + (\mathbf{s}^\top \mathbf{p}) \right)^2 dx \right],
\end{aligned} \tag{17}$$

with the initial conditions

$$\mathbf{q}(0) = \int_a^b \mathbf{c}(x) u_0(x) dx =: \mathbf{q}_0, \quad \mathbf{p}(0) = \int_a^b \mathbf{s}(x) u_0(x) dx =: \mathbf{p}_0. \tag{18}$$

Proof Let us substitute u_t and u_{xxx} from (16) into (1). Multiplying by $\mathbf{c}(x)$, then integrating in space, and considering that

$$\int_a^b \mathbf{c}(x) \mathbf{c}(x)^\top dx = \int_a^b \mathbf{s}(x) \mathbf{s}(x)^\top dx = I, \tag{19}$$

the identity operator,

$$\int_a^b \mathbf{c}(x) \mathbf{s}(x)^\top dx = \int_a^b \mathbf{s}(x) \mathbf{c}(x)^\top dx = O, \tag{20}$$

⁴Hereafter, for sake of brevity, we shall sometimes omit the arguments of the functions.

and $uu_x = (u^2)_x/2$, provide us with the equation

$$\dot{\mathbf{q}} = -\alpha D^3 \mathbf{p} + \frac{\beta}{2} \int_a^b \mathbf{c}(u^2)_x dx. \quad (21)$$

Integrating by parts and taking into account the periodic boundary conditions, one then obtains

$$\int_a^b \mathbf{c}(u^2)_x dx = - \int_a^b \mathbf{c}' u^2 dx \equiv D \int_a^b \mathbf{s} u^2 dx.$$

Substitution into (21) then provides us with the first equation in (17). The second equation is similarly proved by multiplying (1) by $\mathbf{s}(x)$, integrating in space, and considering that

$$\int_a^b \mathbf{s}(u^2)_x dx = - \int_a^b \mathbf{s}' u^2 dx \equiv -D \int_a^b \mathbf{c} u^2 dx.$$

Finally, (18) follows by multiplying (2) by $\mathbf{c}(x)$ and $\mathbf{s}(x)$, respectively, then integrating in space. \square

The following result then holds true.

Theorem 1. *With reference to matrix D defined in (12), system (17) can be formally written as⁵*

$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} & 1 \\ -1 & \end{pmatrix} \otimes D \begin{pmatrix} \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{q}} \\ \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{p}} \end{pmatrix}, \quad (22)$$

which is Hamiltonian, with Hamiltonian

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \left[-\alpha (\mathbf{q}^\top D^2 \mathbf{q} + \mathbf{p}^\top D^2 \mathbf{p}) + \frac{\beta}{3} \int_a^b (\widehat{u}_0 + \mathbf{c}^\top \mathbf{q} + \mathbf{s}^\top \mathbf{p})^3 dx \right]. \quad (23)$$

This latter, in turn, is equivalent to the functional $\mathcal{H}[u]$ defined in (3), via the expansion (15).

⁵As is usual, \otimes denotes the Kronecker product.

Proof With reference to (23), it is straightforward to prove that

$$\begin{aligned}\frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{q}} &= -\alpha D^2 \mathbf{q} + \frac{\beta}{2} \int_a^b \mathbf{c} (\widehat{u}_0 + \mathbf{c}^\top \mathbf{q} + \mathbf{s}^\top \mathbf{p})^2 dx, \\ \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{p}} &= -\alpha D^2 \mathbf{p} + \frac{\beta}{2} \int_a^b \mathbf{s} (\widehat{u}_0 + \mathbf{c}^\top \mathbf{q} + \mathbf{s}^\top \mathbf{p})^2 dx.\end{aligned}$$

Consequently, (17) is equivalent to (22)–(23). In order to prove that (23) is equivalent to $\mathcal{H}[u]$ as defined in (3), it suffices to consider that

$$\int_a^b (\widehat{u}_0 + \mathbf{c}^\top \mathbf{q} + \mathbf{s}^\top \mathbf{p})^3 dx = \int_a^b u^3 dx,$$

because of (15), and

$$\begin{aligned}\mathbf{q}^\top D^2 \mathbf{q} + \mathbf{p}^\top D^2 \mathbf{p} &= \int_a^b [\mathbf{q}^\top D \mathbf{s}(x) \mathbf{s}(x)^\top D \mathbf{q} + \mathbf{p}^\top D \mathbf{c}(x) \mathbf{c}(x)^\top D \mathbf{p}] dx \\ &= \int_a^b u_x^2 dx,\end{aligned}$$

by virtue of (16) and (19). \square

3. Fourier-Galerkin space semi-discretization

In order for problem (17)–(18) to be solvable on a computer, one needs to truncate the infinite expansion (15) to a finite sum. Therefore, having fixed a conveniently large value $N \gg 1$, one approximates (15) as

$$u(x, t) \approx \hat{u}(x, t) := \widehat{u}_0 + \sum_{j=1}^N [c_j(x) q_j(t) + s_j(x) p_j(t)]. \quad (24)$$

We can still pose the expansion (24) in vector form as (15), by formally replacing the infinite vectors (9) by

$$\begin{aligned}\mathbf{c}(x) &= \begin{pmatrix} c_1(x) \\ \vdots \\ c_N(x) \end{pmatrix}, & \mathbf{s}(x) &= \begin{pmatrix} s_1(x) \\ \vdots \\ s_N(x) \end{pmatrix}, \\ \mathbf{q}(t) &= \begin{pmatrix} q_1(t) \\ \vdots \\ q_N(t) \end{pmatrix}, & \mathbf{p}(t) &= \begin{pmatrix} p_1(t) \\ \vdots \\ p_N(t) \end{pmatrix},\end{aligned} \quad (25)$$

having length N . Similarly, the matrix (12) is formally replaced by the $N \times N$ matrix

$$D = \frac{2\pi}{b-a} \begin{pmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & N \end{pmatrix}. \quad (26)$$

For the sake of simplicity, we continue to use the same notation for the truncated version of the infinite vectors and matrices: clearly, hereafter, they will denote the finite ones. Consequently, expressions similar to (16) hold true for the partial derivatives of \hat{u} , and (19)–(20) continue formally to hold. Nevertheless, the function (24) does not satisfy the equation (1) anymore. However, in the spirit of Galerkin methods, by requiring the residual be orthogonal to the functional space

$$\mathcal{V}_N = \text{span} \{c_0(x), c_1(x), s_1(x), \dots, c_N(x), s_N(x)\},$$

to which the approximation (24) belongs for all t , one formally obtains again the equations (17), with the initial conditions formally still given by (18). Consequently, Theorem 1 continues formally to hold, even though the Hamiltonian (23) is now only an approximation to the functional \mathcal{H} defined in (3). Nevertheless, it is known from the theory of Fourier methods [21] that, under regularity assumptions on u (and, thus, on the initial condition u_0), one has that the truncated approximations to u and \mathcal{H} converge more than exponentially to them, as $N \rightarrow \infty$, as we sketched in footnote 1 (this fact is usually referred to as *spectral accuracy*).

Remark 2. *A criterion for getting an estimate for N is to check that both the residual corresponding to the initial condition (see (15) and (18)),*

$$E_0 := \|u_0 - \hat{u}_0 - \mathbf{c}^\top \mathbf{q}_0 - \mathbf{s}^\top \mathbf{p}_0\|_{L_2} \equiv \|u_0(x) - \hat{u}(x, 0)\|_{L_2}, \quad (27)$$

and the difference of the values of $H(\mathbf{q}_0, \mathbf{p}_0)$ is within the round-off error level, for nearby values of N .

Finally, in order to obtain a full space semi-discretization, one needs to compute the integrals appearing in (17), whose arguments are trigonometric polynomials of degree at most $3N$ in the space variable. For this purpose, as observed in [9], one can use a composite trapezoidal rule, evaluated at the abscissae,

$$x_i = a + i \frac{b-a}{m}, \quad i = 0, \dots, m, \quad (28)$$

with m a suitably large natural number. In particular, $\forall m > 3N$ the integrals are exactly computed (see, e.g., [24, Th. 5.1.4]). For this reason, we shall hereafter consider the value

$$m = 3N + 1. \quad (29)$$

Consequently, the truncated problem (17), having dimension $2N$, with the integrals computed via the composite trapezoidal rule at the abscissae (28)–(29), define the semi-discrete problem in space to be integrated in time. The corresponding semi-discrete Hamiltonian is then formally still given by (23), with the integral appearing in it computed via the composite trapezoidal rule based at the abscissae (28)–(29).

4. Hamiltonian Boundary Value Methods

In order to obtain a fully discrete method, we now need to integrate the Hamiltonian problem (17)–(18), having dimension $2N$, by taking into account that the vectors $\mathbf{c}, \mathbf{s}, \mathbf{q}, \mathbf{p}$, and matrix D , are defined by (25)–(26). As observed in [47], it is important to obtain a Hamiltonian semi-discrete ODE problem, from the space semi-discretization of a PDE with Hamiltonian structure. In fact, in such a case one may use a suitable *geometric integrator* (see, e.g., [47, 40, 30, 10]), for efficiently solving the resulting Hamiltonian ODE problem. Hereafter, we shall consider *Hamiltonian Boundary Value Methods (HBVMs)* for numerically solving (17)–(18). They are a class of energy-conserving Runge-Kutta methods which has been studied in a series of papers (see, e.g., [11, 12, 13, 14, 8, 15]). Moreover, HBVMs have been also generalized along several directions, including

the application to Hamiltonian PDEs [4, 9] (the reader is also referred to the recent monograph [10]).

A HBVM(k, s) method is the k -stage Runge-Kutta method defined by the Butcher tableau (see, e.g., [14, 10])

$$\begin{array}{c|c} \mathbf{c} & \mathcal{I}_s \mathcal{P}_s^\top \Omega \\ \hline & \mathbf{b}^\top \end{array}, \quad \mathbf{b} = \begin{pmatrix} b_1 & \dots & b_k \end{pmatrix}^\top, \quad \mathbf{c} = \begin{pmatrix} c_1 & \dots & c_k \end{pmatrix}^\top, \quad (30)$$

where, by setting $\{P_j\}_{j \geq 0}$ the Legendre polynomial basis orthonormal on $[0, 1]$, i.e.,

$$P_i \in \Pi_i, \quad \int_0^1 P_i(x) P_j(x) dx = \delta_{ij}, \quad \forall i, j = 0, 1, \dots,$$

(b_i, c_i) are the weights and abscissae of the Gauss-Legendre quadrature formula of order $2k$ (i.e., $P_k(c_i) = 0$, $i = 1, \dots, k$), and

$$\begin{aligned} \mathcal{P}_s &= \begin{pmatrix} P_0(c_1) & \dots & P_{s-1}(c_1) \\ \vdots & & \vdots \\ P_0(c_k) & \dots & P_{s-1}(c_k) \end{pmatrix} \in \mathbb{R}^{k \times s}, \\ \mathcal{I}_s &= \begin{pmatrix} \int_0^{c_1} P_0(x) dx & \dots & \int_0^{c_1} P_{s-1}(x) dx \\ \vdots & & \vdots \\ \int_0^{c_k} P_0(x) dx & \dots & \int_0^{c_k} P_{s-1}(x) dx \end{pmatrix} \in \mathbb{R}^{k \times s}, \\ \Omega &= \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_k \end{pmatrix} \in \mathbb{R}^{k \times k}. \end{aligned} \quad (31)$$

By using standard arguments in the analysis of such methods (see, e.g., [14, 10]), it is possible to prove the following result.

Theorem 2. *For all $s = 1, 2, \dots$, and $k \geq s$, the HBVM(k, s) method (30):*

- *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 (17)–(23), for all $k \geq 3s/2$.

Remark 3. Because of the result of Theorem 2, hereafter, we shall consider the choice

$$k = \left\lceil \frac{3s}{2} \right\rceil, \quad s = 1, 2, \dots, \quad (32)$$

for all HBVM(k, s) methods. Consequently, they are energy-conserving and of order $2s$, when applied for numerically solving (17)–(18). Moreover, because of the expansion (15), the semi-discrete solution also satisfies the conservation property (5).

Let us now study the efficient implementation of a generic HBVM(k, s) method when applied for solving (17)–(18) by using a timestep $\Delta t = h$. By setting, with reference to (23) and (25),

$$\mathbf{y} := \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix}, \quad H(\mathbf{y}) := H(\mathbf{q}, \mathbf{p}), \quad \mathbf{y}_0 := \begin{pmatrix} \mathbf{q}_0 \\ \mathbf{p}_0 \end{pmatrix}, \quad (33)$$

and considering matrix D defined at (26), one has that (17)–(18) can be formally rewritten as

$$\dot{\mathbf{y}} = J \nabla H(\mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad J = \begin{pmatrix} & D \\ -D & \end{pmatrix}. \quad (34)$$

By also setting

$$Y \equiv \begin{pmatrix} Y_1 \\ \vdots \\ Y_k \end{pmatrix} \in \mathbb{R}^{2Nk}, \quad \nabla H(Y) := \begin{pmatrix} \nabla H(Y_1) \\ \vdots \\ \nabla H(Y_k) \end{pmatrix}, \quad (35)$$

i.e., the stage vector of the method (30) applied for solving (33)–(34), and ∇H evaluated at the stages, respectively, one obtains the nonlinear set of k vector equations,

$$Y = \mathbf{e} \otimes \mathbf{y}_0 + h \mathcal{I}_s \mathcal{P}_s^\top \Omega \otimes J \nabla H(Y), \quad \mathbf{e} = (1, \dots, 1)^\top \in \mathbb{R}^k. \quad (36)$$

Once this system has been solved, the approximation $\mathbf{y}_1 \approx \mathbf{y}(h)$ is computed as:

$$\mathbf{y}_1 = \mathbf{y}_0 + h \sum_{i=1}^k b_i J \nabla H(Y_i). \quad (37)$$

Remark 4. *It is worth mentioning that, when $s = 1$ and $k = 2$, according to (32), the quadrature in (37) is exact and one retrieves the averaged vector field method [46] for solving (17)–(18).*

According to [13], we now derive a more convenient formulation of the discrete problem (36). For this purpose, by setting hereafter I the identity matrix of dimension $2N$, and defining the vector

$$\boldsymbol{\gamma} \equiv \begin{pmatrix} \gamma_0 \\ \vdots \\ \gamma_{s-1} \end{pmatrix} := \mathcal{P}_s^\top \Omega \otimes J \nabla H(Y), \quad (38)$$

one has that (36) can be written as

$$Y = \mathbf{e} \otimes \mathbf{y}_0 + h \mathcal{I}_s \otimes I \boldsymbol{\gamma}. \quad (39)$$

In fact, by plugging (38) into (39), one recovers (36). However, an equivalent formulation of the discrete problem (36) can be derived by substituting (39) at the right-hand side of (38), thus obtaining the equation

$$F(\boldsymbol{\gamma}) := \boldsymbol{\gamma} - \mathcal{P}_s^\top \Omega \otimes J \nabla H(\mathbf{e} \otimes \mathbf{y}_0 + h \mathcal{I}_s \otimes I \boldsymbol{\gamma}) = \mathbf{0}, \quad (40)$$

whose (block) dimension is s , *independently* of k . Once the discrete problem (40) has been solved, the approximation (37) is given by

$$\mathbf{y}_1 = \mathbf{y}_0 + h \gamma_0.$$

In fact, taking into account that $P_0(x) \equiv 1$, from (31), (35), and (38) one obtains that:

$$\gamma_0 = \sum_{i=1}^k b_i J \nabla H(Y_i).$$

Consequently, when implementing the HBVM(k, s) method (30), the complexity for solving the equivalent discrete problem (40), having (block) dimension s , is simplified w.r.t. solving the stage equation (36), which has (block) dimension k , due to the fact that, because of (32), $k > s$.⁶ In addition to this, by taking into account that, because of the properties of Legendre polynomials,

$$\mathcal{P}_s^\top \Omega \mathcal{I}_s = X_s := \begin{pmatrix} \xi_0 & -\xi_1 & & \\ \xi_1 & 0 & \ddots & \\ & \ddots & \ddots & -\xi_{s-1} \\ & & \xi_{s-1} & 0 \end{pmatrix} \in \mathbb{R}^{s \times s}, \quad (41)$$

$$\xi_i = \frac{1}{2\sqrt{|4i^2 - 1|}}, \quad i = 0, \dots, s-1,$$

one has that the simplified Newton iteration for solving (40), representing the reference method of solution, reads:

$$\begin{aligned} \text{set } \boldsymbol{\gamma}^0 &= \mathbf{0} \\ \text{for } r &= 0, 1, \dots : \\ &\text{solve } [I_s \otimes I - hX_s \otimes J\nabla^2 H(\mathbf{y}_0)] \Delta \boldsymbol{\gamma}^r = -F(\boldsymbol{\gamma}^r) \\ &\text{set } \boldsymbol{\gamma}^{r+1} = \boldsymbol{\gamma}^r + \Delta \boldsymbol{\gamma}^r \\ \text{end} \end{aligned} \quad (42)$$

We observe that the coefficient matrix of the linear system in (42) has dimension $s \cdot 2N$, i.e., s times larger than that of the continuous problem (17). Moreover, we need to factor such matrix at each integration step. However, we can gain a twofold simplification of the iteration (42), as explained below.

Firstly, by considering matrix D defined at (26) and the expansion (15), one has

$$\nabla^2 H(\mathbf{y}_0) = \begin{pmatrix} -\alpha D^2 + \beta \int_a^b u \mathbf{c} \mathbf{c}^\top dx & \beta \int_a^b u \mathbf{c} \mathbf{s}^\top dx \\ \beta \int_a^b u \mathbf{s} \mathbf{c}^\top dx & -\alpha D^2 + \beta \int_a^b u \mathbf{s} \mathbf{s}^\top dx \end{pmatrix}.$$

⁶We refer to [13] for full details.

Table 1: Parameter defined at (45).

s	1	2	3	4	5	6
ρ_s	0.5000	0.2887	0.1967	0.1475	0.1173	0.0971

By approximating u with its mean in space, given by \hat{u}_0 (see (15)), then, by virtue of (19)–(20), we can consider the approximate Hessian matrix

$$\nabla^2 H(\mathbf{y}_0) \approx \begin{pmatrix} \hat{D} & \\ & \hat{D} \end{pmatrix} =: G, \quad \hat{D} := -\alpha D^2 + \beta \hat{u}_0 I_N, \quad (43)$$

which is *diagonal* and *constant*.

Secondly, in place of the simplified Newton iteration (42) with the simplified Hessian (43), we consider a “splitting-Newton” *blended iteration*. This iteration, previously devised (see, e.g., [16]) for block Boundary Value Methods,⁷ has then been generalized in [17] and implemented in the computational Fortran codes **BiM** [18] and **BiMD** [19] for stiff ODE-IVPs and linearly implicit differential algebraic equations (the latter code is also available at the *Test Set for IVP Solvers* [56], and is one of the best codes currently available for numerically solving such problems). The blended iteration has also been considered for HBVMs [13, 8], proving to be very efficient when applied to Hamiltonian PDEs, as is shown in [9] for the semi-linear wave equation, and in [4] for the nonlinear Schrödinger equation. We here sketch the main facts for the solution of problem (17)–(18). In fact, each PDE has its own structural properties, which need to be exploited in order to optimize the nonlinear iteration. As a result, the iteration (42) is replaced by the following one:

⁷We refer, e.g., to [20] for details on block Boundary Value Methods.

```

set  $\gamma^0 = \mathbf{0}$ 
for  $r = 0, 1, \dots$  :
    set  $\eta^r = -F(\gamma^r)$ 
    set  $\eta_1^r = \rho_s X_s^{-1} \otimes I \eta^r$ 
    set  $\Delta\gamma^r = I_s \otimes \Sigma [\eta_1^r + I_s \otimes \Sigma (\eta^r - \eta_1^r)]$ 
    set  $\gamma^{r+1} = \gamma^r + \Delta\gamma^r$ 
end

```

(44)

where X_s is the matrix defined at (41),

$$\rho_s = \min_{\lambda \in \sigma(X_s)} |\lambda|, \quad (45)$$

with $\sigma(X_s)$ denoting the spectrum of X_s (a few values of the parameter ρ_s are listed in Table 1), and (see (34) and (43))

$$\Sigma := (I - h\rho_s JG)^{-1} \equiv \begin{pmatrix} I_N & -B \\ B & I_N \end{pmatrix}^{-1}, \quad B := h\rho_s D\hat{D}. \quad (46)$$

Remark 5. We observe that matrix Σ is the only matrix which needs to be factored to perform the iteration (44). Moreover, its dimension equals that of the continuous problem (17), i.e., $2N$. Conversely, even using the approximation (43), the simplified Newton iteration (42) would require to factor the matrix

$$[I_s \otimes I - hX_s \otimes JG] \in \mathbb{R}^{2Ns \times 2Ns},$$

that is, s times larger. Consequently, the use of the blended iteration (44) reduces the computational cost for the implementation of the methods, both in terms of memory requirement and floating-point operations per iteration. Also, the extensive numerical experimentation performed in [18, 19] (see also [56]), testifies the effectiveness of the blended iteration itself, so that we shall not go further into details concerning this aspect.

Next result states that Σ , alike Σ^{-1} , has a block diagonal structure.

Theorem 3. *With reference to matrix (46), one has*

$$\Sigma = \begin{pmatrix} (I_N + B^2)^{-1} & B(I_N + B^2)^{-1} \\ -B(I_N + B^2)^{-1} & (I_N + B^2)^{-1} \end{pmatrix}.$$

Consequently, matrix Σ :

- is *constant* and, therefore, needs to be computed *only once*;
- has a 2×2 *block* diagonal structure. *Consequently, only two vectors of length N are needed for storing it, respectively containing the diagonal entries of $(I_N + B^2)^{-1}$ and $B(I_N + B^2)^{-1}$.*

In conclusion, one obtains that, besides the evaluation of $F(\gamma)$ in (40), the linear algebra cost for performing the iteration (44) is *linear* in the dimension of the problem (17) to be solved, both in terms of required operations and memory requirements. Concerning the evaluation of $F(\gamma)$ one has a complexity which is $O(N \log N)$ operations and $O(N)$ memory requirements [28], since the evaluation of the integrals via the composite trapezoidal rule at the abscissae (28)–(29), can be done via the FFT and its inverse. This, in turn, allows the use of relatively large values of N .

Remark 6. *For completeness, we mention that the use of a fixed-point iteration for solving the discrete problem (40), i.e.,*

$$\gamma^{r+1} = \mathcal{P}_s^\top \Omega \otimes J \nabla H(e \otimes \mathbf{y}_0 + h \mathcal{I}_s \otimes I \gamma^r) \quad r = 0, 1, \dots,$$

would require, to converge, the use of a timestep Δt (see (34) and (43)) of the order of $\|JG\|^{-1} \equiv \|D\hat{D}\|^{-1}$, i.e., such that

$$\Delta t \approx |\alpha|^{-1} \left(\frac{b-a}{2\pi N} \right)^3, \quad (47)$$

which is, therefore, very small, when N is large. The blended iteration (44), on the other hand, allows the use of much larger timesteps (actually, the iteration is guaranteed to converge for any timestep, in the case where $\beta = 0$ in (1) [16, 17]).

5. Numerical examples

In this section we provide a few numerical examples, aimed at confirming what exposed in Sections 3 and 4. In all cases, we use periodic boundary conditions, according to (1)–(2). All numerical tests have been performed by using Matlab (R2016a) on a 2.2 GHz dual core i7 laptop with 8GB of memory.

Example 1. This example is adapted from [23, Example. 5.3]:

$$\begin{aligned} u_t(x, t) + \epsilon u_{xxx}(x, t) + u(x, t)u_x(x, t) &= 0, & (x, t) \in [-3, 5] \times [0, 24], \\ \epsilon &= 0.0013020833. \end{aligned} \quad (48)$$

The initial condition at $t = 0$ is derived from the known solution of the problem, i.e.,

$$u(x, t) = 3c \left[\operatorname{sech} \left(\sqrt{\frac{c}{4\epsilon}} (x - ct)_{[-3, 5]} \right) \right]^2, \quad c = \frac{1}{3}, \quad (49)$$

where, in general,

$$(\xi)_{[a, b]} := \begin{cases} \xi, & \text{if } \xi \in [a, b], \\ a + \operatorname{rem}(\xi - a, b - a), & \text{if } \xi > b, \\ b - \operatorname{rem}(b - \xi, b - a), & \text{if } \xi < a, \end{cases} \quad (50)$$

with rem the remainder in the integer division between the two arguments. As a result, one verifies that the solution (49) is periodic in time with period $T = 24$. In Figure 1, we plot the solution of problem (48)–(50). Moreover, in Figure 2 we plot the value of the residual (27) for the initial condition, E_0 , and the difference between the corresponding values of the numerical Hamiltonian, ΔH_0 , for increasing values of the parameter N in (24). As one may observe from the figure, both E_0 and ΔH_0 decrease more than exponentially with N and, for $N \approx 250$, both of them become almost constant. Consequently, according to Remark 2, in the sequel we consider the value $N = 250$ for the numerical tests concerning this example (we recall that the value of m in (28) is chosen according to (29), in order to exactly compute the required integrals in the space variable).

In Table 2 we list the maximum errors in the computed solution, e_u , for decreasing timesteps also estimating the numerical rate of convergence, along with the error in the numerical Hamiltonian, e_H , for the HBVM(k, s) methods, $s = 1, 2, 3$, with k chosen according to (32). All the errors are computed at $T = 24$: we see that e_u decreases with order $2s$, according to Theorem 2,⁸ whereas e_H is negligible (it is within the round-off error level), as predicted. In the table we also list the mean number of required blended iterations (44) per step, from which we see that they quickly decrease with the timestep and, for the finest timestep considered ($\Delta t = 0.0125$), they are almost independent of s . It is worth observing that even though the mean number of blended iterations per step appears to be very high for the coarsest timestep used ($\Delta t = 0.4$), it must be stressed that, for the considered value $N = 250$, according to (47), a fixed-point iteration would require $\Delta t \approx 10^{-4}$, in order to converge.⁹ The plots of Figure 3 contain the *work-precision diagrams*, namely accuracy (of the solution, in the upper plot, and of the Hamiltonian, in the lower plot) vs. execution time [56], for the methods listed in Table 2. For comparison, we have also included the plots concerning the methods HBVM(12,8) and HBVM(15,10) (the former used with timesteps $\Delta t = 24/M$, $M = 60, 120, 240, 480$, the latter used with timesteps $\Delta t = 24/M$, $M = 60, 120, 240$), and the Matlab code CHEBFUN [57]. The script for this latter code has been adapted from [58] by using 500 grid-points in space (equivalent to the spatial accuracy of the Fourier-Galerkin discretization considered for HBVMs) and timesteps $\Delta t = (25M)^{-1}$, $M = 1, 2, 4, 8, 16, 32, 128, 256, 512$.

From the diagrams in Figure 3, one deduces that the higher the order of the HBVM method, the better its efficiency. Moreover, the highest-order HBVMs

⁸For larger values of s , the solution error becomes soon negligible, as the timestep is decreased, but, due to round-off errors, the numerical assessment of the order is more difficult.

⁹As an example, we found experimentally that HBVM(5,3) can be implemented by using a fixed-point iteration with a timestep $\Delta t = 4 \cdot 10^{-4}$ and an execution time of about 500 sec. On the other hand, the use of the blended iteration with the timesteps listed in Table 2, results into execution times ranging from 4 to 42 sec.

are competitive with CHEBFUN, when a high solution accuracy is required. On the other hand, when energy conservation is an issue then HBVMs turn out to be more efficient than CHEBFUN. Energy conservation, in turn, is an important property of HBVMs. In order to assess this point, let us look at the circle in the upper plot in Figure 3, from which we see that CHEBFUN, using a timestep $\Delta t = 1/3200$ and HBVM(15,10), using a timestep $\Delta t = 0.2$, provide a comparably accurate numerical solution after one period (the solution error is $\approx 1.5 \cdot 10^{-10}$ for both methods), in approximately the same time (i.e., ≈ 23 sec). Nevertheless, if we continue the integration for 20 periods, measuring the errors at the end of each period, we see that CHEBFUN exhibits a drift in the Hamiltonian error, as is shown in the upper plot in Figure 4. This, in turn, is responsible for an almost quadratic error growth in the numerical solution, as confirmed by the lower plot in the same figure. On the other hand, the Hamiltonian error for HBVM(15,10) remains within the round-off error level, resulting into a much smaller growth of the solution error. As matter of fact, in the lower plot in Figure 4, an almost constant error is reported for HBVM(15,10). In general, for energy-conserving HBVMs a linear error growth of the solution error is at most observed. This fact is confirmed by the plots in Figures 5 and 6, where we plot the Hamiltonian and solution errors w.r.t. time, respectively, over the time-interval $[0, 500]$, when using the HBVM(k, s) methods, $s = 1, 2, 3$ and k according to (32), with timestep $\Delta t = 0.05$. In all cases, the Hamiltonian errors depicted in Figure 5 are within the round-off error level (as is expected), whereas the solution errors in Figure 6 grow at most linearly (in the case $s = 1$, due to the low order of the method, the linear growth is until the error saturates).

Example 2. This example is taken from [43, Ex. 4.1]:

$$\begin{aligned} u_t(x, t) + \epsilon u_{xxx}(x, t) + u(x, t)u_x(x, t) &= 0, & (x, t) \in [0, 1] \times [0, \infty), \\ \epsilon &= (24)^{-2}. \end{aligned} \tag{51}$$

The initial condition at $t = 0$ is derived from the known solution of the problem, known as the *cnoidal-wave solution*,

$$u(x, t) = a \operatorname{cn}^2(4K(m)(x - \nu t - x_0)), \quad (52)$$

where $\operatorname{cn}(z) := \operatorname{cn}(z|m)$ is the Jacobi elliptic function with modulus m , $K(m)$ is the complete elliptic integral of the first kind, and

$$m = 0.9, \quad a = 192m\epsilon K^2(m), \quad \nu = 64\epsilon(2m - 1)K^2(m), \quad x_0 = 1/2.$$

According to Remark 2, for this problem, one has that the value $N = 50$ for the truncation parameter in (24) is sufficient to guarantee a solution accurate enough (as matter of fact one has that the parameter defined at (27) is $E_0 \leq 10^{-15}$, and the value of the numerical Hamiltonian remains constant, when considering larger values of N).

In [43, Fig. 1], there is the plot of the numerical and true solutions at $t = 0, 200, 500, 1000$ (for completeness, the reference solutions are shown in Figure 7), when a timestep $\Delta t = 10^{-3}$ is used: as is clear from the plots in that Figure, the error can be appreciated even with the naked eye. In Table 3, we list the maximum errors at the same times $t = 0, 200, 500, 1000$, when using HBVM(k, s) methods (with k given by (32)), for increasing values of s , by using a timestep as large as $\Delta t = 0.1$. From this table, it is clear that, despite the large stepsize used, the error becomes very small as s increases, because of the increasing order of the method used. Moreover, we also list the maximum error in the numerical Hamiltonian, e_H , thus confirming that it is conserved up to round-off.

Example 3. This example is slightly adapted from [43, Ex. 4.2],¹⁰

$$u_t(x, t) + u_{xxx}(x, t) + u(x, t)u_x(x, t) = 0, \quad (x, t) \in [-115, 103] \times [0, \infty). \quad (53)$$

¹⁰We have considered a larger space interval, w.r.t. that considered in [43, Ex. 4.2], in order to have a better approximation when using periodic boundary conditions. In fact, by using the original interval $[-40, 40]$, the solution turns out to be discontinuous, as a periodic function. This fact is much less notable, when considering the new interval.

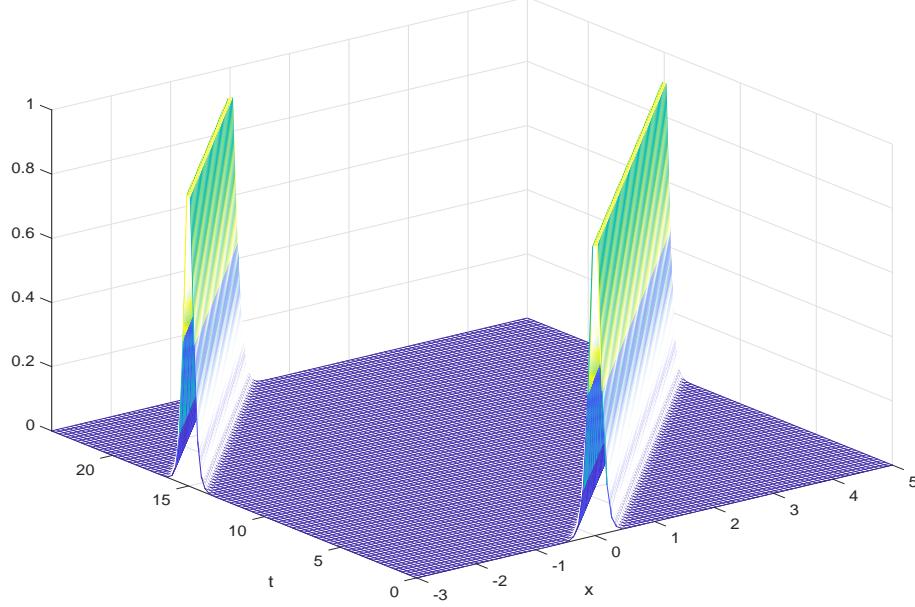


Figure 1: Solution (49) of problem (48) for $(x, t) \in [-3, 5] \times [0, 24]$.

The initial condition at $t = 0$ is derived from the two-soliton waves solution,

$$u(x, t) = 12 \frac{k_1^2 e^{\theta_1} + k_2^2 e^{\theta_2} + 2(k_2 - k_1)^2 e^{\theta_1 + \theta_2} + a^2 (k_2^2 e^{\theta_1} + k_1^2 e^{\theta_2}) e^{\theta_1 + \theta_2}}{(1 + e^{\theta_1} + e^{\theta_2} + a^2 e^{\theta_1 + \theta_2})^2}, \quad (54)$$

where

$$k_1 = 0.4, \quad k_2 = 0.6, \quad a^2 := \left(\frac{k_2 - k_1}{k_2 + k_1} \right)^2 = \frac{1}{25}, \quad x_1 = 4, \quad x_2 = 15,$$

and (see (50))

$$\begin{aligned} \theta_1 &:= \theta_1(x, t) = (k_1 x - k_1^3 t + x_1)_{[-115, 103]}, \\ \theta_2 &:= \theta_2(x, t) = (k_2 x - k_2^3 t + x_2)_{[-115, 103]}. \end{aligned}$$

In this case, according to Remark 2, the parameter N in (24) is conveniently chosen as $N = 300$ (in fact, with reference to (27), one has $E_0 < 10^{-15}$ and, moreover, the numerical Hamiltonian remains constant within round-off, when using larger values of N). Inspired from the numerical results reported in [43,

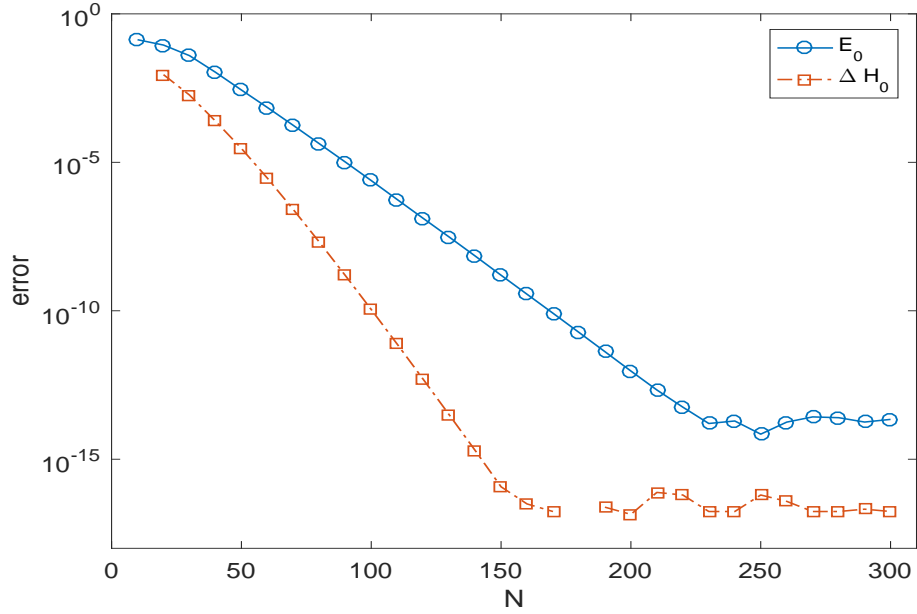


Figure 2: Error E_0 in the initial condition (see (27)) and differences in the initial numerical Hamiltonian, ΔH_0 , for increasing values of N .

Ex. 4.2], in Table 4 we list the maximum errors in the numerical solution at $t = 0, 40, 80, 120$, obtained by using HBVM(k, s) methods, $s = 1, \dots, 4$ and k according to (32), with timestep $\Delta t = 0.1$, along with the corresponding Hamiltonian error. It is worth mentioning that the numerical experiments show that, for this problem, larger values of s cannot improve further the obtained accuracy (which is of the order of the round-off error level for $s = 4$). The choice of the above mentioned reference times is due to the fact that, as is shown in Figure 8, the two waves, a taller one and a lower one (see the plot for $t = 0$), gradually approach one another (see the plot for $t = 40$), when moving towards right, until they collide (see the plot for $t = 80$), then continuing moving away from each other (see the final plot at $t = 120$). From the results listed in Table 4, one has that, as expected, the numerical Hamiltonian turns out to be conserved and, moreover, the numerical solution soon reaches machine accuracy, as s is increased from 1 to 4. This, in turn, means that the collision of the two waves is approximated to full machine accuracy.

Example 4. The last example is the famous Zabusky-Kruskal example [54] (see also [23, Ex. 5.5] or [43, Ex. 4.3]):

$$\begin{aligned} u_t(x, t) + \epsilon u_{xxx}(x, t) + u(x, t)u_x(x, t) &= 0, & (x, t) \in [0, 1] \times [0, \infty), \\ \epsilon &= (0.022)^2, & u(x, 0) = \cos(\pi x), & x \in [0, 2]. \end{aligned} \quad (55)$$

A good description of the main features of the solution of such problem can be found in [23], and here we sketch the main facts reported in that reference:

- a) the solution starts with a cosine wave and later on develops a train of 8 solitons which travel at different speeds and interact with each other. In more detail,
- b) at $t_1 := t_B \equiv \pi^{-1}$, the solution is about to breakdown;
- c) at $t_2 := 3.6t_B$, the train of 8 solitons has been developed;
- d) at $t_3 := 0.5t_R \equiv 0.5 \cdot 30.4t_B$, all the odd-numbered solitons overlap at $x = 0.385$ and all the even-numbered overlap at $x = 1.385$;
- e) at $t_4 := t_R \equiv 30.4t_B$, the recurrence time, all the solitons arrive in almost the same phase to reconstruct the initial state.

Also in this case, according to Remark 2, the parameter N in (24) is conveniently chosen as $N = 300$ (the parameter defined at (27) is $E_0 \approx 7 \cdot 10^{-16}$ and the numerical Hamiltonian remains constant within round-off for nearby values of N). In Figure 9 is the plot of the computed numerical solution at the times t_1, \dots, t_4 defined above, with a maximum estimated error (infinity norm) of $\approx 6 \cdot 10^{-13}$. The error estimate has been obtained by computing, at first, the solution with the HBVM(3,2) method, with timesteps¹¹

$$\Delta t = h_i := t_B(2^{i-1}5)^{-1}, \quad i = 1, \dots, 8. \quad (56)$$

Then, on the finest time grid, we have computed the solution by using higher order methods, with the same value of N , until the difference in the computed solutions becomes negligible. In so doing, we computed the solutions

¹¹ $h_1 \approx 6 \cdot 10^{-2}$, $h_8 \approx 5 \cdot 10^{-4}$.

with the HBVM(5,3), HBVM(6,4), and HBVM(8,5) methods. The solution of HBVM(8,5) has then been used as reference solution, and the difference with the solution computed by the other methods, at the times t_1, \dots, t_4 , is listed in Table 5. As one may see, the (actually, very small) difference between the solutions computed by HBVM(8,5) and HBVM(6,4) is approximately the same as the difference between the solutions of HBVM(8,5) and HBVM(5,3). This fact clearly indicates that we have reached the maximum possible accuracy. The fact that the computed reference solution by HBVM(8,5) is correct, is enforced by observing that the corresponding errors of the HBVM(3,2) method decrease with the prescribed order 4. Moreover, in order to exclude a possible underestimation of the parameter N in (24), we have also computed the solution by means of the HBVM(8,5) method on the finest time grid using the parameter $N = 600$, instead of 300. In the last row of Table 5 we list the differences in the computed solutions at t_1, \dots, t_4 , as well as the difference between the corresponding numerical Hamiltonians, w.r.t. the reference ones. As one may see, all the differences are compatible with the round-off error level of the double precision IEEE. This, in turn, further confirms the accuracy and reliability of the reference solutions plotted in Figure 9.¹² Moreover, such plots are in agreement with the more accurate plots reported in Figures 6 and 7 in [23] (i.e., those with 800 cells). In particular, the first three plots in Figure 9 confirm the features described at the points a)–d) above, whereas the plot at $t = t_4$ confirms what observed in [23, Ex. 5.5], where it was noticed that the solution at the recurrence time t_R does not coincide with the initial condition, thus contradicting the feature described at e).

At last, in Figure 10 is the plot of the error in the numerical Hamiltonian for the computed reference solution of Figure 9 for $t \in [0, t_R]$, by using HBVM(8,5) with the finest timestep specified in (56). As is expected, this error is within the round-off error level.

¹²We remind that the reference solution has been obtained by using HBVM(8,5) on the finest mesh in (56).

6. Concluding remarks

In this paper we studied the numerical solution of the Korteweg–de Vries equation with periodic boundary conditions. The problem has been cast into Hamiltonian form, by means of a Fourier-Galerkin space semi-discretization. Energy-conserving Runge-Kutta methods, of arbitrarily high-order, in the HB-VMs class have then been used for the time integration, while conserving the energy of the system. The efficient implementation of such methods has been also studied, showing that their computational complexity per step is *linear* in the dimension $2N$ of the semi-discrete problem, for memory requirements, and $O(N \log N)$ for operations count. The effectiveness of the methods has been evaluated on some test problems.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (11271357) and by Università di Firenze (project “Risoluzione numerica di problemi Hamiltoniani ed applicazioni”). The paper emerged during a visit of the first author at the State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences, in June 2017.

The authors sincerely thank the reviewers for the very careful reading of the manuscript and the many valuable comments which have greatly improved the original version.

References

- [1] M.E. Alexander, J.L. Morris. Galerkin methods applied to some model equations for non-linear dispersive waves. *J. Comput. Phys.* **30**, no. 3 (1979) 428–451.
- [2] U. Ascher, R. McLachlan. Multisymplectic box schemes and the Korteweg–de Vries equation. *Appl. Numer. Math.* **48** (2004) 255–269.
- [3] D. Bambusi, A. Carati, A. Maiocchi, A. Maspero. Some Analytic Results on the FPU Paradox. In *Hamiltonian Partial Differential Equations and Applications*, P. Guyenne, D. Nicholls, C. Sulem Eds., *Fields Institute Communications* **75** (2015) 235–254.

- [4] L. Barletti, L. Brugnano, G. Frasca Caccia, F. Iavernaro. Energy-conserving methods for the nonlinear Schrödinger equation. *Appl. Math. Comput.* **318** (2018) 3–18.
- [5] D. Bättig, T. Kappeler, B. Mityagin. On the Korteweg-De Vries equation: frequencies and initial value problem. *Pacific J. Math.* **181**, No. 1 (1997) 1–5.
- [6] J.L. Bona, H. Chen, O. Karakashian, Y. Xing. Conservative, discontinuous Galerkin-methods for the generalized Korteweg-de Vries equation. *Math. Comp.* **82**, no. 283 (2013) 1401–1432.
- [7] J.L. Bona, V.A. Dougalis, O. Karakashian A. Fully discrete Galerkin methods for the Korteweg-de Vries equation. *Comput. Math. Appl. Ser. A* **12**, no. 7 (1986) 859–884.
- [8] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Efficient implementation of Gauss collocation and Hamiltonian Boundary Value Methods. *Numer. Algorithms* **65** (2014) 633–650.
- [9] 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
- [10] L. Brugnano, F. Iavernaro. *Line Integral Methods for Conservative Problems*. Chapman et Hall/CRC, Boca Raton, FL, 2016.
- [11] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian BVMs (HBVMs): a family of “drift-free” methods for integrating polynomial Hamiltonian systems. *AIP Conf. Proc.* **1168** (2009) 715–718.
- [12] 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.
- [13] L. Brugnano, F. Iavernaro, D. Trigiante. A note on the efficient implementation of Hamiltonian BVMs. *J. Comput. Appl. Math.* **236** (2011) 375–383.
- [14] 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.
- [15] L. Brugnano, F. Iavernaro, D. Trigiante. Analysis 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.
 - [16] L. Brugnano, C. Magherini. Blended Implementation of Block Implicit Methods for ODEs. *Appl. Numer. Math.* **42** (2002) 29–45.
 - [17] L. Brugnano, C. Magherini. Recent advances in linear analysis of convergence for splittings for solving ODE problems. *Appl. Numer. Math.* **59** (2009) 542–557.
 - [18] L. Brugnano, C. Magherini. The BiM code for the numerical solution of ODEs. *J. Comput. Appl. Math.* **164–165** (2004) 145–158.
 - [19] L. Brugnano, C. Magherini, F. Mugnai. Blended Implicit Methods for the Numerical Solution of DAE Problems. *J. Comput. Appl. Math.* **189** (2006) 34–50.
 - [20] L. Brugnano, D. Trigiante. Boundary Value Methods: the Third Way Between Linear Multistep and Runge-Kutta Methods. *Comput. Math. Appl.* **36**, no. 10–12 (1998) 269–284.
 - [21] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang. *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.
 - [22] Y. Chen, B. Cockburn, B. Dong. Superconvergent HDG methods for linear, stationary, third-order equations in one-space dimension. *Math. Comp.* **85**, no. 302 (2016) 2715–2742.
 - [23] Y. Cui, D.-k. Mao. Numerical method satisfying the first two conservation laws for the Korteweg–de Vries equation. *J. Comput. Phys.* **227** (2007) 376–399.
 - [24] G. Dahlquist, Å. Björk. *Numerical Methods in Scientific Computing, vol. 1*. SIAM, Philadelphia, 2008.
 - [25] J. de Frutos, J.M. Sanz-Serna. Accuracy and conservation properties in numerical integration: the case of the Korteweg–de Vries equation. *Numer.*

- Math.* **75** (1997) 421–445.
- [26] C.S. Gardner. Korteweg–de Vries equation and generalization. IV. The Korteweg–de Vries equation as a Hamiltonian system. *J. Math. Phys.* **12**, no. 8 (1971) 1548–1651.
 - [27] E. Giusti. *Direct methods in the calculus of variations*. World Scientific Publishing Co., Inc., River Edge, NJ, 2003.
 - [28] G.H. Golub, C.F. Van Loan. *Matrix Computations, 3rd Ed.* The Johns Hopkins University Press, Baltimore, 1996.
 - [29] H. Guan, S. Kuksin. The KdV equation under periodic boundary conditions and its perturbations. *Nonlinearity* **27**, no. 9 (2014) R61–R88.
 - [30] E. Hairer, C. Lubich, G. Wanner. *Geometric Numerical Integration: Structure-preserving algorithms for ordinary differential equations, 2nd Edition*. Springer-Verlag, Berlin, 2006.
 - [31] M. Hofmanová, K. Schratz. An exponential-type integrator for the KdV equation. *Numer. Math.* **136**, no. 4 (2017) 1117–1137.
 - [32] H. Holden, K.H. Karlsen, N.H. Risebro, T. Tao. Operator splitting methods for the Korteweg–de Vries equation. *Math. Comp.* **80** (2011) 821–846.
 - [33] M. Huang. A Hamiltonian approximation to simulate solitary waves of the Korteweg–de Vries equation. *Math. Comp.* **56**, no. 194 (1991) 607–620.
 - [34] C. Hufford, Y. Xing. Superconvergence of the local discontinuous Galerkin method for the linearized Korteweg–de Vries equation. *J. Comput. Appl. Math.* **255** (2014) 441–455.
 - [35] P. Isaza, F. Linares, G. Ponce. On Decay Properties of Solutions of the k -Generalized KdV Equation. *Commun. Math. Phys.* **324** (2013) 129–146.
 - [36] P. Isaza, F. Linares, G. Ponce. On the Propagation of Regularity and Decay of Solutions to the k -Generalized Korteweg–de Vries Equation. *Commun. Partial Diff. Equat.* **40** (2015) 1336–1364.
 - [37] J. Jackaman, G. Papamikos, T. Pryer. The design of conservative finite element discretisations for the vectorial modified KdV equation. (2017) [arXiv:1710.03527](https://arxiv.org/abs/1710.03527) [math.NA]

- [38] T. Kappeler, J. Pöschel. On the Well-Posedness of the Periodic KdV Equation in High Regularity Classes. In: *Hamiltonian Systems and Applications*, W.Craig (ed). Springer, 2008, pp. 431–441.
- [39] B. Karasözen, G. Şimşek. Energy preserving integration of bi-Hamiltonian partial differential equations. *Appl. Math. Let.* **26** (2013) 1125–1133
- [40] B. Leimkuhler, S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, Cambridge, 2004.
- [41] H. Liu, J. Yan. A local discontinuous Galerkin method for the Korteweg-de Vries equation with boundary effect. *J. Comput. Phys.* **215** (2006) 197–218.
- [42] X. Li, L. Zhang, S. Wang. A compact finite difference scheme for the nonlinear Schrödinger equation with wave operator. *Appl. Math. Comput.* **219** (2012) 3187–3197.
- [43] H. Liu, N. Yi. A Hamiltonian preserving discontinuous Galerkin method for the generalized Korteweg–de Vries equation. *J. Comp. Phys.* **321** (2016) 776–796.
- [44] P.J. Olver. Hamiltonian and non-Hamiltonian models for water waves. In *Trends and Applications of Pure Mathematics to Mechanics*, pp. 273–290. *Lecture Notes in Phys.* **195** (1984).
- [45] J. Nahas, G. Ponce. On the persistence properties of solutions of nonlinear dispersive equations. In *Weighted Sobolev spaces. Harmonic analysis and nonlinear partial differential equations*. Res. Inst. Math. Sci. (RIMS), Kyoto, 2011, pp. 23–36.
- [46] G.R.W. Quispel, D.I. McLaren. A new class of energy-preserving numerical integration methods. *J. Phys. A: Math. Theor.* **41** (2008) 045206 (7pp).
- [47] J.M. Sanz-Serna, M.P. Calvo. *Numerical Hamiltonian problems*. Chapman & Hall, London, 1994.
- [48] M.Z. Song, X. Qian, H. Zhang, S.H. Song. Hamiltonian Boundary Value Method for the Nonlinear Schrödinger Equation and the Korteweg–de Vries Equation. *Adv. Appl. Math. Mech.* **9** (2017) 868–886.
- [49] F. Tappert. Numerical solutions of the Korteweg-de Vries equation and

- its generalizations by the split-step Fourier method. *In: Newell, A.C. (ed.) Nonlinear Wave Motion*, American Mathematical Society, Providence (1974), pp. 215–216.
- [50] R. Winther. A Conservative Finite Element Method for the Korteweg-de Vries Equation. *Mathematics of Computation* **34**, No. 149 (1980) 23–43.
 - [51] J. Wang, Y. Wang. Local structure-preserving algorithms for the KdV equation. *J. Comput. Math.* **35**, no. 3 (2017) 289–318.
 - [52] Y. Xu, C.W. Shu. Error estimates of the semi-discrete local discontinuous Galerkin method for nonlinear convection-diffusion and KdV equations. *Comput. Methods Appl. Mech. Engrg.* **196**, (37-40) (2007) 3805–3822.
 - [53] J. Yan, C.-W. Shu. A local discontinuous Galerkin method for KdV type equations. *SIAM J. Numer. Anal.* **40**, no. 2 (2002) 769–791.
 - [54] N.L. Zabusky, M.D. Kruskal. Interaction of Solitons in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Let.* **15** (1965) 240–243.
 - [55] P. Zhao, M. Qin. Multisymplectic geometry and multisymplectic Preissmann scheme for the KdV equation. *J. Phys. A: Math. Gen.* **33** (2006) 3613–3626.
 - [56] <https://archimede.dm.uniba.it/~testset/testsetivpsolvers/>
 - [57] T.A. Driscoll, N. Hale, L.N. Trefethen. *Chebfun Guide*. Pafnuty Publications, Oxford, UK, 2014. URL: <http://www.chebfun.org>
 - [58] <http://www.chebfun.org/examples/pde/KdV.html>

Table 2: Problem (48) solved by HBVM(k, s) methods, $s = 1, 2, 3$, and k according to (32), by using $N = 250$.

HBVM(k, s)	HBVM(2,1)				HBVM(3,2)				HBVM(5,3)			
Δt	e_u	rate	e_H	it	e_u	rate	e_H	it	e_u	rate	e_H	it
0.4	1.03e-00	–	1.39e-17	516	4.29e-01	–	1.39e-17	105	5.40e-02	–	1.39e-17	71
0.2	9.91e-01	0.0	2.08e-17	80	2.88e-02	3.9	1.39e-17	49	9.17e-04	5.9	2.81e-17	42
0.1	5.96e-01	0.7	1.73e-17	42	3.16e-03	3.2	2.81e-17	31	2.98e-05	4.9	1.39e-17	30
0.05	1.74e-01	1.8	1.73e-17	27	2.56e-04	3.6	1.73e-17	23	1.01e-06	4.9	1.73e-17	25
0.025	4.42e-02	2.0	2.08e-17	20	1.61e-05	4.0	1.73e-17	19	3.00e-08	5.1	1.39e-17	22
0.0125	1.11e-02	2.0	2.08e-17	16	9.90e-07	4.0	1.73e-17	16	3.51e-10	6.4	1.73e-17	20

Table 3: Solution errors for problem (51) solved by HBVM(k, s) methods, $s = 3, \dots, 6$, and k according to (32), by using $N = 50$ and a timestep $\Delta t = 0.1$, along with the maximum Hamiltonian error, e_H .

$s \mid t$	0	200	500	1000	e_H
3	8.88e-16	9.13e-02	2.34e-01	4.54e-01	3.05e-16
4	8.88e-16	1.62e-03	3.36e-03	6.81e-03	3.33e-16
5	8.88e-16	4.38e-05	5.46e-05	7.98e-05	2.78e-16
6	8.88e-16	5.71e-06	3.76e-06	4.44e-06	2.78e-16

Table 4: Solution errors for problem (53) solved by HBVM(k, s) methods, $s = 1, \dots, 4$, and k according to (32), by using $N = 300$ and a timestep $\Delta t = 0.1$, along with the maximum Hamiltonian error, e_H .

$s \mid t$	0	40	80	120	e_H
1	9.99e-16	8.66e-05	7.41e-05	1.92e-04	6.66e-16
2	9.99e-16	1.02e-09	6.49e-10	1.73e-09	7.77e-16
3	9.99e-16	1.41e-13	4.69e-14	1.39e-13	7.77e-16
4	9.99e-16	6.12e-15	4.66e-15	1.42e-14	7.77e-16

Table 5: Estimated errors in the numerical solution, and in the numerical Hamiltonian, of problem (55) at t_i , $i = 1, \dots, 4$, by using a timestep $\Delta t = t_B/n \equiv (\pi n)^{-1}$ and the listed parameter N . The reference solution has been computed on the finest time grid (i.e., that with $n = 640$) by using the HBVM(8,5) method with $N = 300$.

		$t_1 \equiv t_B$		$t_2 \equiv 3.6t_B$		$t_3 \equiv 15.2t_B$		$t_4 \equiv 30.4t_B$		
	n	e_1	rate	e_2	rate	e_3	rate	e_4	rate	e_H
HBVM(3,2) $N = 300$	5	1.35e-03	—	2.56e-02	—	7.81e-02	—	3.60e-01	—	2.05e-16
	10	1.01e-04	3.7	3.59e-03	2.8	1.06e-02	2.9	1.96e-02	4.2	1.77e-16
	20	6.59e-06	3.9	2.95e-04	3.6	5.43e-04	4.3	9.97e-04	4.3	2.39e-16
	40	4.13e-07	4.0	1.66e-05	4.2	3.13e-05	4.1	5.01e-05	4.3	1.77e-16
	80	2.57e-08	4.0	7.76e-07	4.4	1.46e-06	4.4	2.91e-06	4.1	2.32e-16
	60	1.60e-09	4.0	4.72e-08	4.0	9.07e-08	4.0	1.78e-07	4.0	2.19e-16
	320	1.00e-10	4.0	2.94e-09	4.0	5.67e-09	4.0	1.12e-08	4.0	2.32e-16
	640	6.36e-12	4.0	1.83e-10	4.0	3.55e-10	4.0	6.97e-10	4.0	2.95e-16
HBVM(5,3) $N = 300$	640	6.12e-13		6.33e-13		5.55e-13		5.56e-13		2.88e-16
HBVM(6,4) $N = 300$	640	6.12e-13		6.28e-13		5.42e-13		6.72e-13		6.66e-16
HBVM(8,5) $N = 600$	640	8.33e-13		8.61e-13		6.81e-13		6.99e-13		2.47e-17

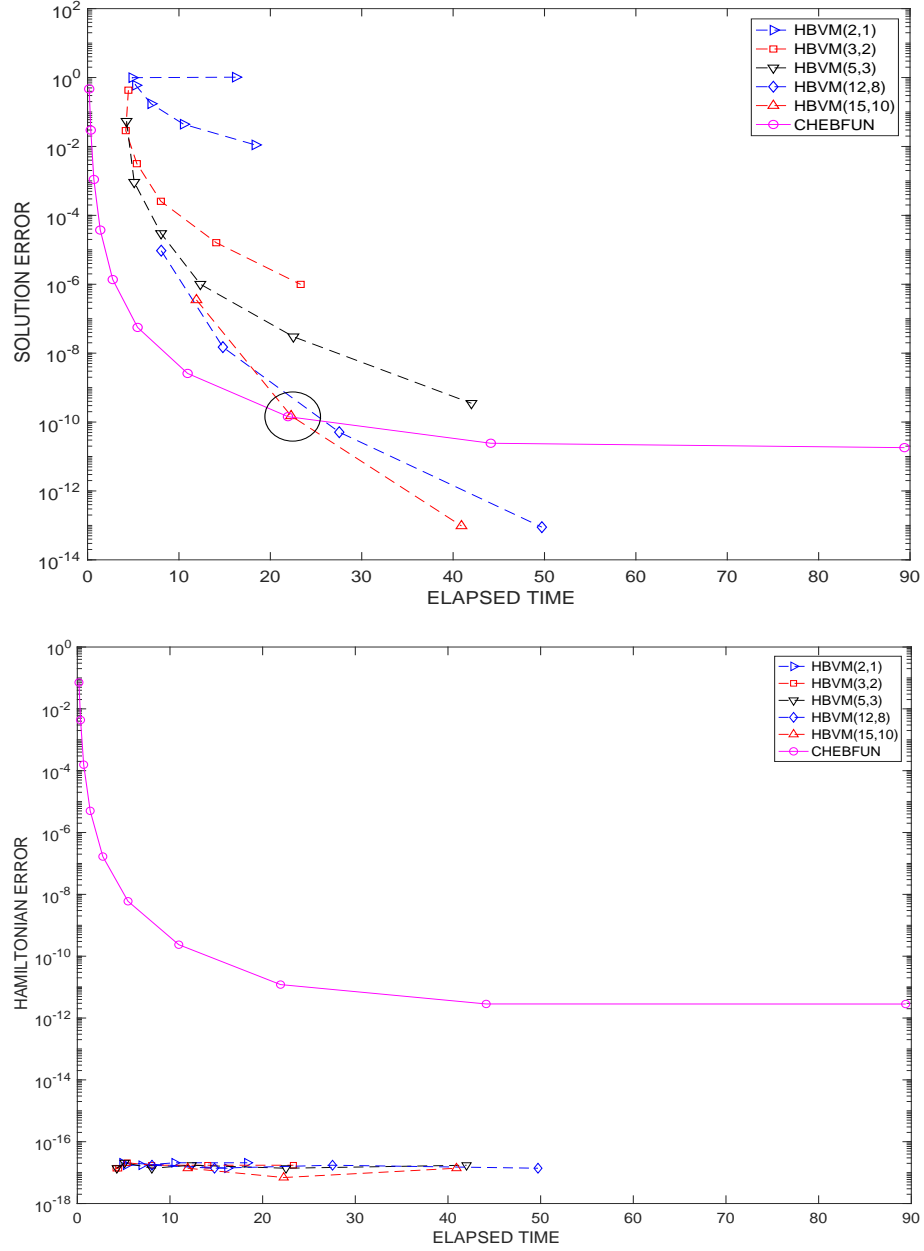


Figure 3: Work-precision diagrams for problem (48) (times are in sec): upper plot solution; lower plot Hamiltonian. The circle in the upper plot is for later use.

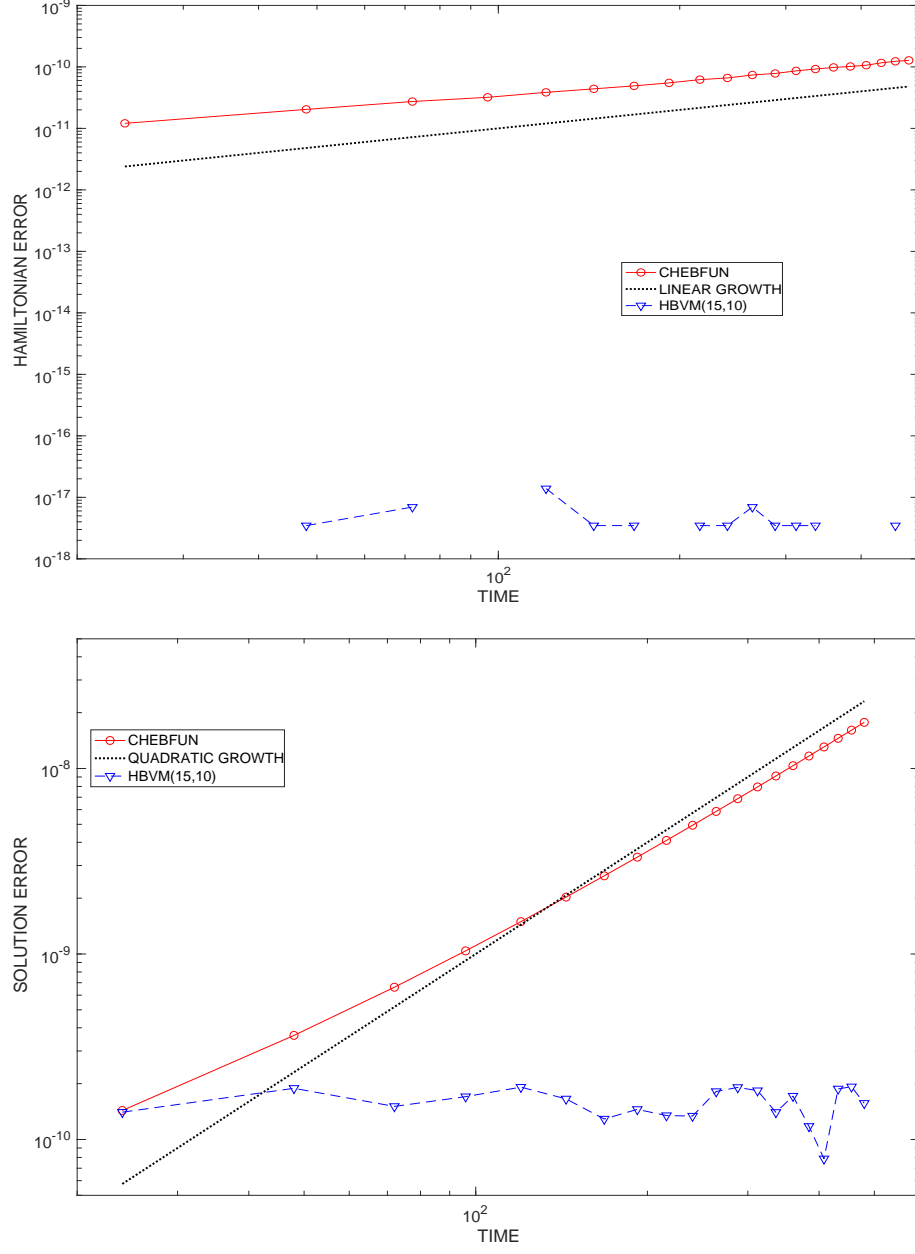


Figure 4: Upper plot: Hamiltonian error over 20 periods when solving problem (48) with CHEBFUN, using a timestep $\Delta t = 1/3200$, and HBVM(15,10), using a timestep $\Delta t = 0.2$; for the former method a linear drift is observed. Lower plot: corresponding solution errors for the above methods; for CHEBFUN, an almost quadratic error growth is observed.

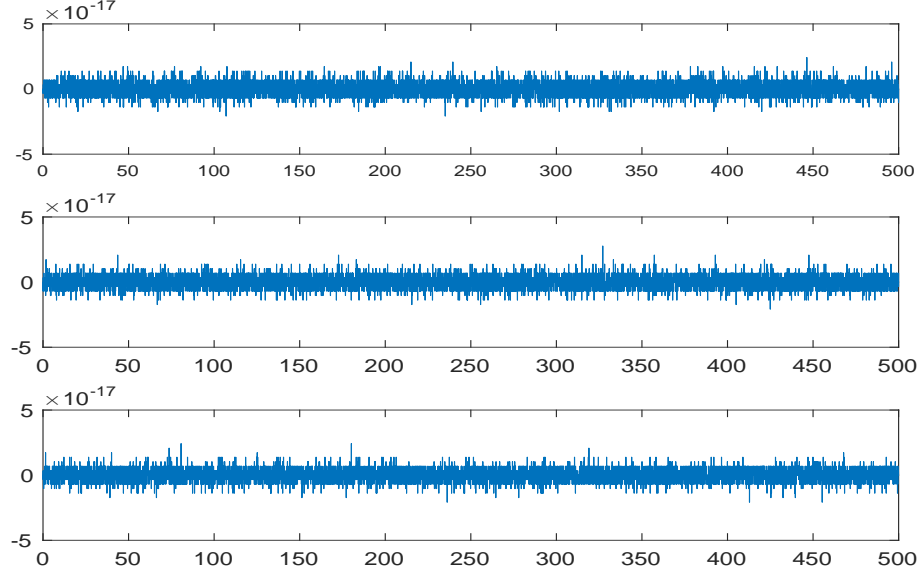


Figure 5: Hamiltonian error versus time when solving problem (48) with timestep $\Delta t = 0.05$ and HBVM(k, s), k given by (32). Upper plot, $s = 1$; middle plot, $s = 2$; lower plot, $s = 3$.

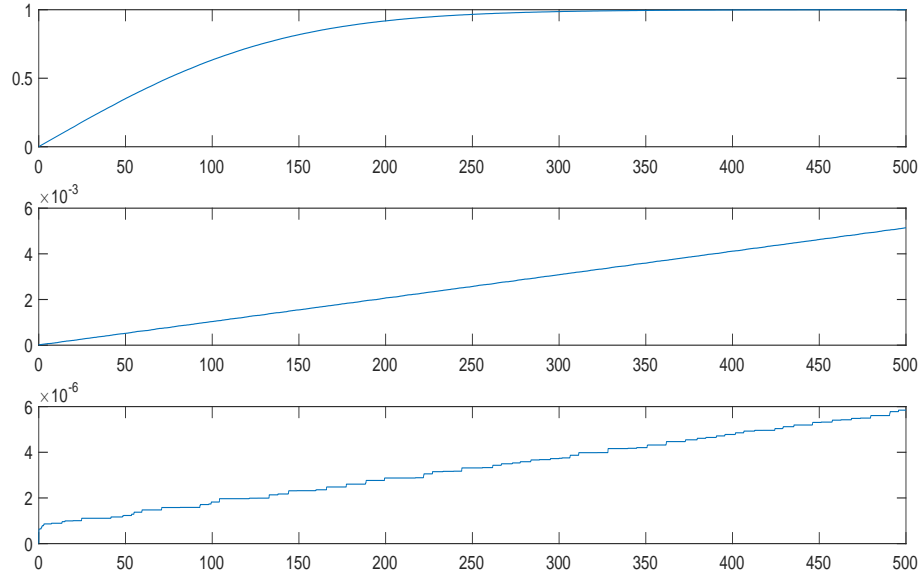


Figure 6: Linear growth of the solution error versus time when solving problem (48) for $t \in [0, 500]$ with timestep $\Delta t = 0.05$ and HBVM(k, s), k given by (32). Upper plot, $s = 1$; middle plot, $s = 2$, lower plot, $s = 3$. In the case $s = 1$, the growth is linear until the error saturates.

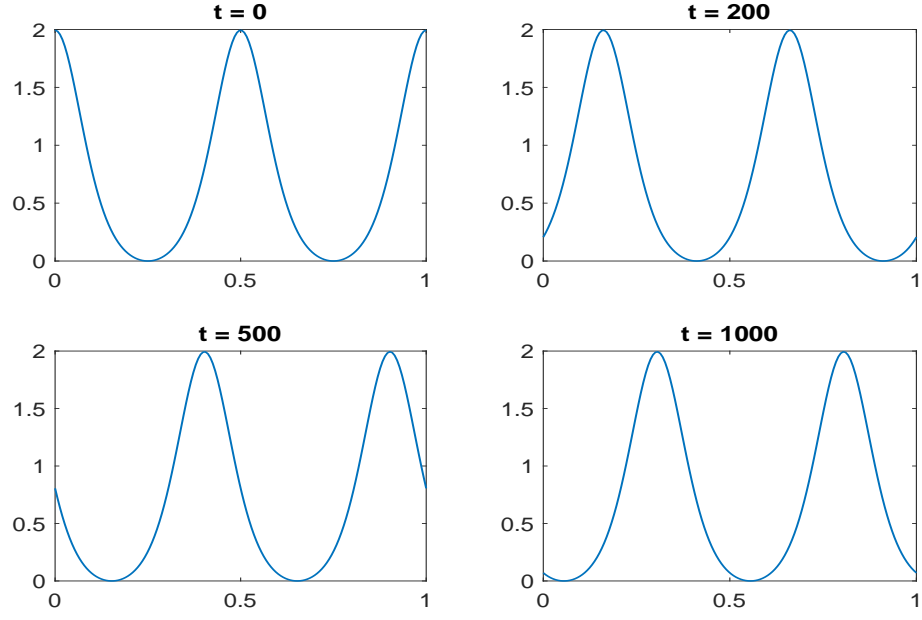


Figure 7: Plot of the exact solution (52) of problem (51) versus x at $t = 0, 200, 500, 1000$.

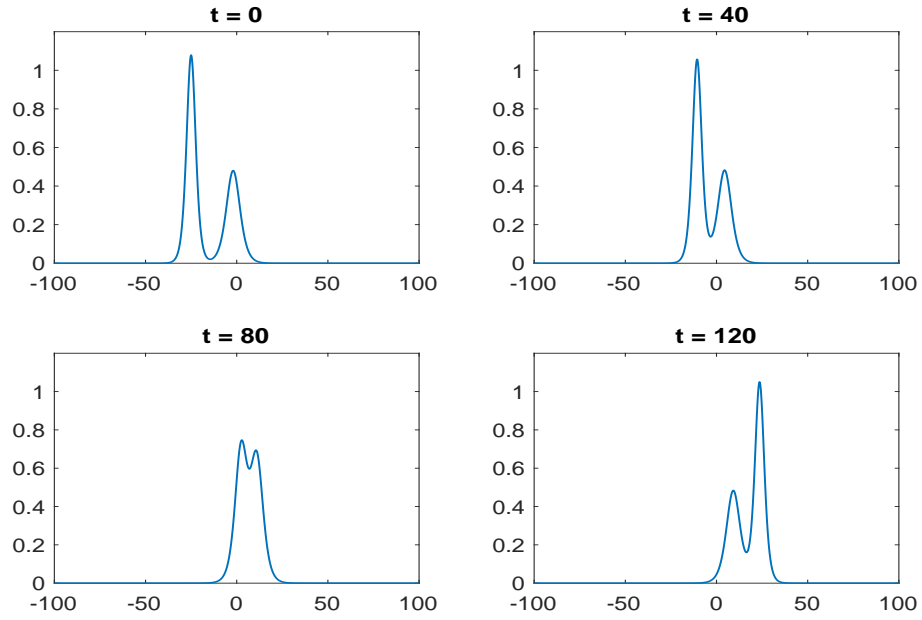


Figure 8: Solution (54) of problem (53) versus x at $t = 0, 40, 80, 120$.

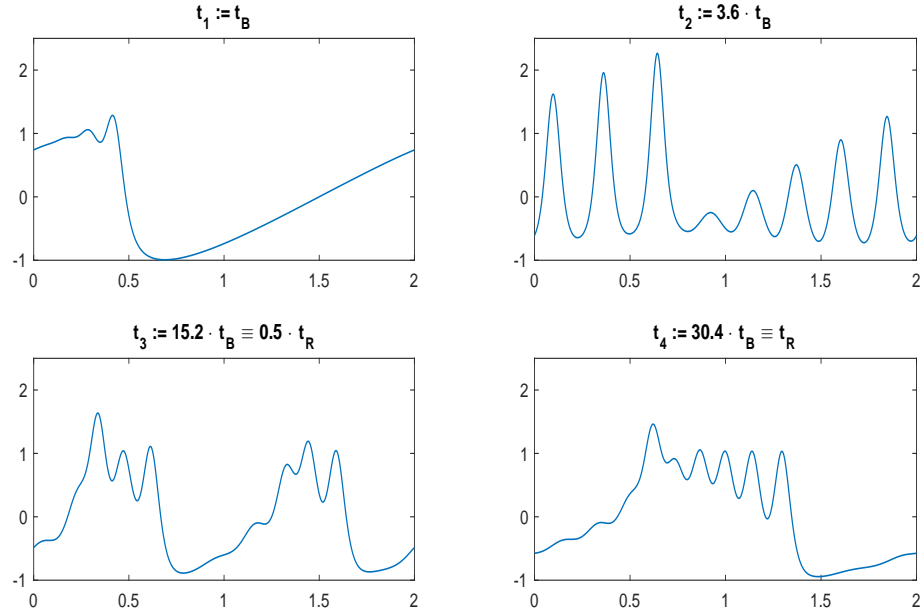


Figure 9: Reference solution of the Zabusky–Kruskal problem (55) versus x at $t = t_i$, $i = 1, 2, 3, 4$ (see text), with an estimated maximum error smaller than 10^{-12} , computed by using HBVM(8,5) with the parameters specified in the caption of Table 5.

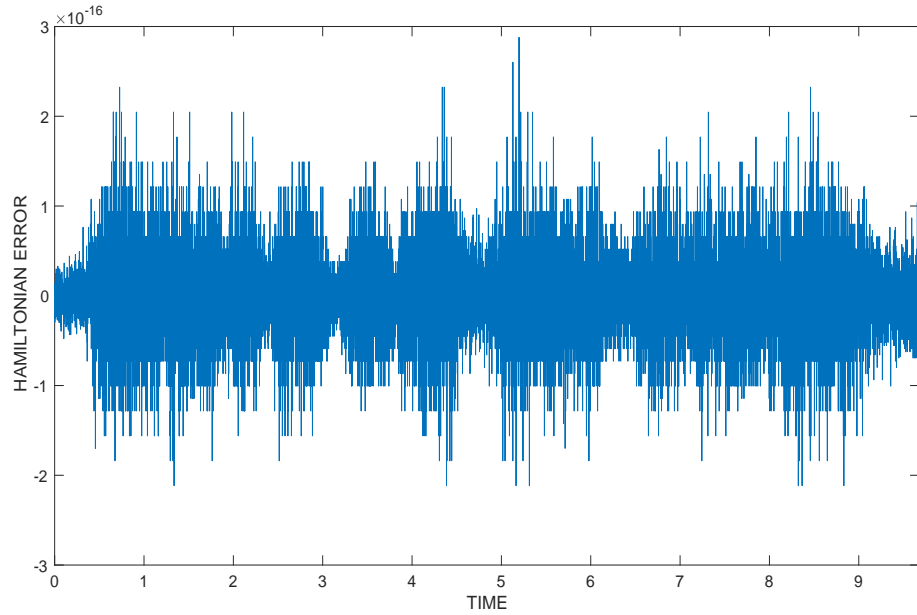


Figure 10: Error in the numerical Hamiltonian for the reference solution of the Zabusky–Kruskal problem (55), from $t = 0$ to $t = t_R \approx 9.6766$, computed by using HBVM(8,5).