UNIVERSITÀ DEGLI STUDI DI FIRENZE

Dipartimento di Matematica e di Informatica "Ulisse Dini"

Dottorato di Ricerca in Matematica

A new efficient implementation for HBVMs and their application to the semilinear wave equation.

Gianluca Frasca Caccia

Tutor

Prof. Luigi Brugnano

Coordinatore del Dottorato

Prof. Alberto Gandolfi

Relatore

Prof. Luigi Brugnano

CICLO XXVII, SETTORE SCIENTIFICO DISCIPLINARE MAT/08

Acknowledgements

"Tell me and I forget,
teach me and I may remember,
involve me and I learn."

- Benjamin Franklin -

First and foremost I would like to deeply thank my advisor, Professor Luigi Brugnano, who introduced me to a very challenging scientific research field. I am grateful not only for the endless support and the large amount of time devoted to the care of my work of thesis, but especially for the motivational and contagious enthusiasm for his research and scientific spirit that characterize him. During my Ph.D. experience he never stopped encouraging me and has been an example to follow, not only from a professional perspective but also from a human point of view.

I am also grateful to Professor Felice Iavernaro for all his precious suggestions and significant contributions to the work developed in this thesis.

I would like also to express my sincere gratitude to the referees, Professor John C. Butcher and Professor Juan I. Montijano. It was an honour to receive their comments and suggestions on my thesis.

Last but not least, I would like to thank my whole family for the support in all these years.

Thank you.

Contents

Introduction

The numerical solution of conservative problems is a relevant issue of research since many years. In fact, a numerical method introduces a small perturbation in the original system which, in general, destroys some of its fundamental properties. It is then of interest to be able to reproduce such properties in the discrete vector field induced by a numerical method. The field of investigation having the final goal to reproduce, in the discrete setting, a number of geometric properties shared by the original continuous problem, is known as *geometric integration*. The very first attempt to perform geometrical integration can be led back to the early work of G. Dalquist, where it is required that the numerical methods are able to reproduce the asymptotic stability of equilibria.

The most famous class of problems dealt within geometrical integration is given by *Hamiltonian problems* of ordinary differential equations (ODEs) which are encountered in many real-life applications, ranging from the nano-scale of molecular dynamics to the macro-scale of celestial mechanics. Hamiltonian problems satisfy two fundamental features: the symplecticity of the flow in the phase space and the conservation of a number of first integrals, among which the most important is the Hamiltonian function itself which is also referred to as the *energy*, since for isolated mechanical systems it has the physical meaning of total energy.

Although a wide literature exists about the analytic treatment of Hamiltonian systems, their study from a numerical point of view is relatively recent due to the lack of appropriate means of investigation and to the difficulties in determining methods able to reproduce on a computer the correct behaviour of their solution.

This difficulty is due to the fact that Hamiltonian systems are not structurally stable against non-Hamiltonian perturbations. Therefore, the use of an ordinary numerical method introduces a perturbation which, in general, destroys the qualitative properties of the original solution, such as, in particular, the symplecticity of the flow and the conservation of the first integrals. Moreover, it has been proved that it is impossible to define a numerical method satisfying, in general, both of these two relevant properties.

As a consequence, concerning the numerical integration of Hamiltonian problems, two main lines of investigation have led to the definition of *symplectic* methods and *energy-conserving* methods, respectively.

At the time when the research on this topic was started, there were no available numerical methods possessing such conservation features, whereas the study of their symplecticity properties seemed certainly more manageable to handle. This partly explains the development of a number of symplectic methods.

Symplectic methods are obtained by imposing that the discrete map, associated with a given numerical method, is symplectic as is the continuous one. Although for the continuous map symplecticity implies energy-conservation, this is no more true for the discrete map. In particular, even though the numerical solution generated by a symplectic method shows interesting long-time behaviour [3, 49], it was observed that symplecticity can assure, at most, the conservation of only quadratic Hamiltonian functions. In the general case conservation cannot be assured and, conse-

quently, it makes sense to look for energy-conserving methods able to exactly satisfy the conservation property of the Hamiltonian along the numerical trajectory.

The very first attempts to face this problem were based on projection techniques coupled with standard non conservative numerical methods. However, it is well-known that this approach suffers from many drawbacks, in that this is usually not enough to correctly reproduce the dynamics (see, e.g. [49, p. 111]).

A different approach is represented by *discrete gradient methods* introduced and studied in the pioneering work [46] and later in [64]. These methods are based upon the definition of a discrete counterpart of the gradient operator, and are able to assure energy conservation of the numerical solution at each step and for any choice of the integration stepsize.

A further approach is based on the concept of *time finite element methods* [54], where one finds local Galerkin approximations on each subinterval of a fixed mesh for the given equation. This, in turn, has led to the definition of energy-conserving Runge-Kutta methods [4, 5, 73, 74].

A partially related approach is given by *discrete line integral methods* [55, 56, 57], where the key idea is to exploit the relation between the method itself and the *discrete line integral*, i.e. the discrete counterpart of the line integral in conservative vector fields. This tool yields exact conservation for polynomial Hamiltonians of arbitrarily high-degree, and results in the class of methods later named *Hamiltonian Boundary Value Methods (HBVMs)*, which have been developed in a series of papers [11, 18, 19, 20, 21, 22, 24, 25, 26].

Another approach, strictly related to the latter one, is given by the *Averaged Vector Field* method [36, 67] and its generalizations [48], which have been also analysed in the framework of B-series [37, 51] (i.e., methods admitting a Taylor expansion with respect to the stepsize).

In the last decades there has been also a growing interest in the numerical treatment of Hamiltonian partial derivative equations (PDEs) arising in many application fields, such as meteorology and weather prediction, quantum mechanics and nonlinear optics [8]. As a direct extension, the ideas and tools related to geometric integration of ODEs has led to the definition and analysis of various structure preserving algorithms for PDEs. Two main lines of investigation are based on a multisymplectic reformulation of the equations and their semi-discretization by means of the method of lines (MOL), respectively.

Multisymplectic structures generalize the classical Hamiltonian structure of a Hamiltonian ODE by assigning a distinct symplectic operator for each unbounded space direction and time [7]. A clear advantage of this approach is that it allows for an easy generalization from symplectic to multisymplectic integration. Multisymplectic integrators are numerical methods which precisely conserve a discrete space-time symplectic structure of Hamiltonian PDEs [8, 43, 44, 59, 63] (a backward error analysis of such schemes may be found in [60, 61, 66]).

When the method of lines approach is used, the spatial derivatives are usually approximated by finite differences [12, 13, 34] or by discrete Fourier transforms [6, 12, 14, 35, 42, 65, 70, 71, 75] and the resulting system is then integrated in time by some standard integrator, usually symplectic or energy-conserving.

With these premises, the thesis is organized as follows. In Chapter 1 we discuss the basic issues about geometric integration, in particular of symplectic methods, and we give a concrete motivation to look for energy-conserving methods for the numerical solution of Hamiltonian problems. In particular, we focus on the so-called *discrete line integral methods* that are the basis which led to the definition and the development of HBVMs.

Chapter 2 is devoted to a few preliminary results concerning Legendre polynomials and perturbation results for differential equations.

In Chapter 3 HBVMs methods are introduced and all their main features are investigated. In particular we show that HBVMs are A-stable, symmetric and energy conserving in the case of poly-

nomial Hamiltonians. This latter property results also in a *practical* numerical energy conservation for any problem defined by a suitably regular Hamiltonian.

In Chapter 4 we introduce two different procedures for the efficient implementation of HBVMs: the first one is based on a *blended implementation* of the methods, the latter, which is one of the two main results of the research developed in this thesis, is based on a particular splitting of the Butcher matrix defining the methods. We also provide a linear analysis of convergence for these two procedures and a few numerical tests in order to make a comparison between different types of implementation.

The developed research addresses another main topic which is discussed in Chapter 5, where we move to the field of PDEs. In particular, we consider a problem for the semilinear wave equation and we use a method of lines approach, consisting of a spatial discretization obtained by means of a finite difference approximation and a full discretization in time accomplished by means of a method in the family of the HBVMs. We consider the different cases when the differential problem presents periodic, Dirichlet or Neumann boundary conditions and we focus our attention on the fact that the proposed methods are able to numerically reproduce the variation of the energy density which, integrated over an interval, depends only on the net flux through its endpoints. In the particular case of periodic boundary conditions, we study also a different strategy for the spatial discretization, obtained by means of a spectral method. Finally, we give evidence of the effectiveness of the proposed methods by showing significant numerical tests where a not negligible error in the reproduction of the energy variation (which is null in the particular case of periodic boundary conditions), due to the use of non energy-conserving methods in time, results in a wrong dynamics for the obtained approximation, whereas the use of energy-conserving methods in time makes it possible to obtain the correct portrait of the solution.

Chapter 1

Geometric Integration

Geometric integration is a recent branch of numerical analysis and computational mathematics.

The philosophy of geometric integration is that numerical methods should preserve relevant qualitative attributes of the original problem (in particular its geometric properties) to the extent it is possible.

The motivation for developing structure-preserving algorithms arises in different areas of research such as celestial mechanics, molecular dynamics, control theory and particle accelerators physics.

The qualitative nature of the phenomena studied in all these areas strongly depends on the conservation of some geometric structure of the underlying system. By introducing these properties into the numerical method, geometric integration allows for an improved qualitative behaviour of the method.

In the first part of this thesis, we shall deal, in particular, with Hamiltonian problems of ODEs having the following general form,

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

where $J^\top = -J = J^{-1}$ is a constant, orthogonal and skew-symmetric matrix. Problem (1.1) is in *canonical form* if

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

where I is the identity matrix of dimension m. The scalar function $H(y)$ is the *Hamiltonian* or the *energy* of the problem and its value is constant during the motion, namely

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

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

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

due to the fact that $J^\top = -J$. For isolated mechanical systems the Hamiltonian H has the physical meaning of total energy and, for this reason, its conservation is deeply important in the simulation of these problems.

A different way to write problem (1.1) is obtained by splitting the state vector of the Hamiltonian system in two m-length components

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

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

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

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

1.1 Symplectic vs energy-conserving methods

In order to introduce another important feature of Hamiltonian dynamical system we need a couple of ingredients:

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

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

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

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

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

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

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

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

  $$u'(q,p)^\top J u'(q,p) = J, \qquad \forall q,p \in \mathbb{R}^m.$$

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

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

and considering (1.4), one has that

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

Therefore

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

The converse of this property is also true, namely, if the flow associated with a dynamical system $\dot{y} = f(y)$ defined on $\mathbb{R}^{2m}$ is symplectic, then necessarily there exists a scalar function $H(y)$ such that $f(y) = J\nabla H(y)$. Because of (1.3) one also has that $H(y)$ is conserved during the motion.

Among the most important implications of symplecticity on the dynamics of Hamiltonian systems there are:

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

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

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

Since $\partial\phi_t(y)/\partial y \equiv A(t)$ is a symplectic matrix, from $A(t)^\top J A(t) = J$ it follows that $\det(A(t))^2 = 1$ for any t and hence $\mathrm{vol}(\phi_t(V)) = \mathrm{vol}(V)$.

More in general, the Liouville theorem states that the flow ϕ_t associated with a divergence-free vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is volume preserving. We recall that the divergence of a vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is the trace of its Jacobian matrix:

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

and f is divergence-free if

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

This is the case of the vector field $J\nabla H$ associated with a Hamiltonian system, in fact, considering that $J\nabla H = [\,\partial H/\partial p_1, \ldots, \partial H/\partial p_m, -\partial H/\partial q_1, \ldots, -\partial H/\partial q_m\,]^\top$, we obtain

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

since the partial derivatives commute.

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

$$y_1 = \Phi_h(y_0),$$

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

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

Symplectic methods can be found in the early work of Gröbner (see, e.g, [47]). Symplectic Runge-Kutta methods have been then studied by Feng [41], Sanz Serna [68], and Suris [72]. Such methods are obtained by imposing that the numerical flow of the given numerical method is symplectic as is the continuous one. In particular, in [68] an easy criterion for symplecticity is provided for an s-stage Runge-Kutta method with tableau given by

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

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

Theorem 1.1.1 ([68]). *The Runge-Kutta method (1.5) is symplectic if and only if*

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

Moreover, in [68] is also proved the existence of infinitely many symplectic Runge-Kutta methods, due to the fact that all Gauss-Legendre Runge-Kutta collocation methods satisfy (1.6).

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

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

$$\frac{\mathrm{d}}{\mathrm{d}t} I(y(t)) = \nabla I(y(t))^\top y'(t) = \nabla I(y(t))^\top J \nabla H(y(t)) = 0, \qquad \forall y.$$

A quadratic first integral is in the form $I(y) = y^\top C y$, with C a symmetric matrix.

As previously seen, the most noticeable first integral of a Hamiltonian system is the Hamiltonian function itself. But, though in the continuous setting the property of symplecticity of the flow implies energy conservation (see, e.g., [45]), the same is no longer true in the discrete setting: a symplectic integrator is not able to yield energy conservation in general.

Nevertheless one could still expect that at least an *approximate conservation* holds for the discrete map. As a matter of fact, under suitable assumptions, it can be proved that the numerical solution obtained by using a symplectic method with constant stepsize satisfies a perturbed Hamiltonian problem, thus providing a quasi-conservation property over an "exponentially" long time [3, 49]. Even though this is an interesting feature, nonetheless, it constitutes a somewhat weak stability result since, in general, it does not extend to infinite intervals.

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

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

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

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

and suppose we are interested in approximating the level curves of the Hamiltonian (shown in Figure 1.1) passing from the points

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

This can be done by integrating the trajectories starting at such initial points, for the corresponding Hamiltonian system.

By using a *symplectic* 2-stage Gauss method with stepsize $h = 10^{-4}$, we obtain the phase portrait depicted in Figure 1.2 which is clearly wrong.[1]

A way to get rid of this problem is to directly look for *energy-conserving* methods, able to provide an exact conservation of the Hamiltonian function along the numerical trajectory. In this thesis we shall consider, in particular, a class of energy-conserving Runge-Kutta methods named *Hamiltonian Boundary Value Methods (HBVMs)* and in the present chapter we focus on the basic idea these methods rely on, i.e. the definition of discrete line integrals.

[1] Additional examples may be found in [18].

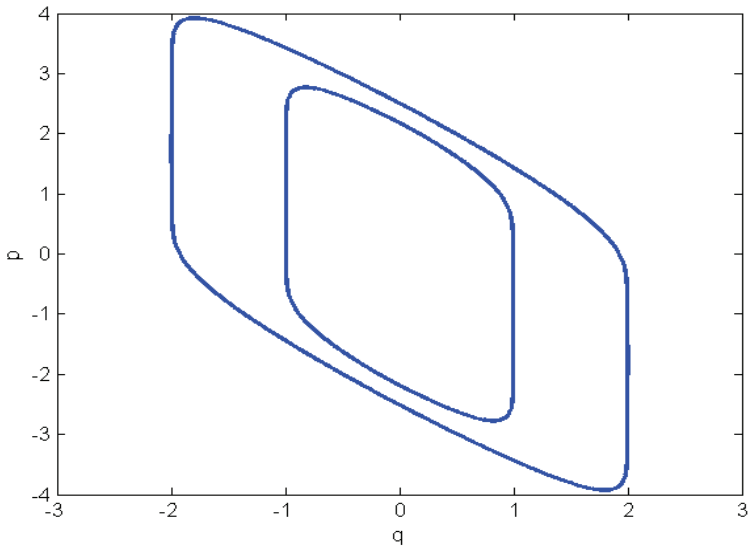

Figure 1.1: Level curves for problem (1.7)–(1.9).

1.2 Discrete line integral methods

In order to present the quite straightforward idea at the basis of such methods, we shall first sketch the simplest case, as was done in [55] and then we will generalize the arguments. Assume that in problem (1.1) the Hamiltonian is a polynomial of degree ν. Starting from the initial condition y_0, our goal is to produce a new approximation at time $t = h$, say y_1, such that the Hamiltonian is conserved. Let us consider the simplest possible path joining y_0 and y_1, i.e. the segment

$$\sigma(ch) = cy_1 + (1-c)y_0, \qquad c \in [0,1], \tag{1.10}$$

one obtains

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

provided that

$$y_1 = y_0 + hJ\int_0^1 \nabla H(cy_1 + (1-c)y_0)\mathrm{d}c. \tag{1.11}$$

In fact, due to the fact that J is skew-symmetric, one obtains:

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

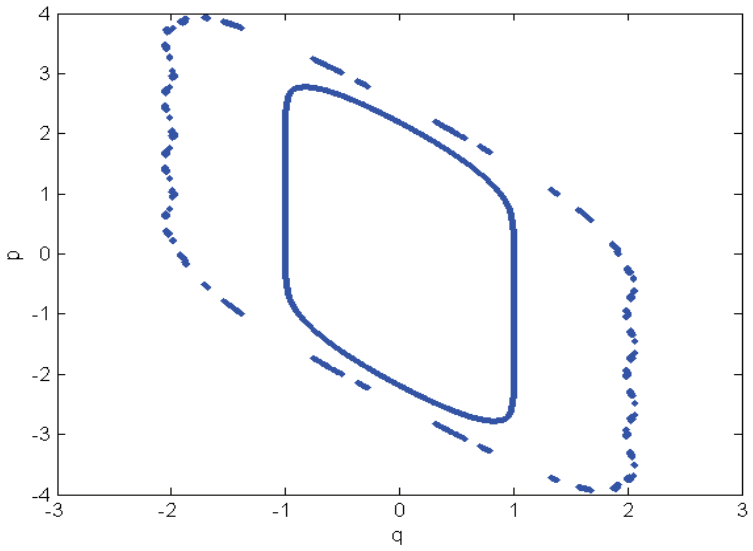

Figure 1.2: 2-stage Gauss method, $h = 10^{-4}$, for approximating problem (1.7)–(1.9).

Therefore, in the discrete setting, we can obtain energy conservation by imposing only that the first and the last point of the path (1.10) are on the manifold where the Hamiltonian is constant although, unlike the continuous setting, in general, our path does not lies entirely on this manifold.

Moreover, being $H \in \Pi_\nu$, the integrand at the right hand side in (1.11) is a polynomial of degree $\nu - 1$ and therefore can be exactly computed by using, say, a Newton-Cotes formula based at ν equidistant abscissae in $[0, 1]$. By setting, hereafter,

$$f(\cdot) = J\nabla H(\cdot), \tag{1.12}$$

one obtains

$$y_1 = y_0 + h\sum_{i=1}^{\nu} b_i f(c_i y_1 + (1 - c_i)y_0) \equiv y_0 + h\sum_{i=1}^{\nu} b_i f(Y_i), \tag{1.13}$$

where

$$c_i = \frac{i-1}{\nu - 1}, \qquad Y_i = \sigma(c_i h) \equiv c_i y_1 + (1 - c_i)y_0, \qquad i = 1, \ldots, \nu, \tag{1.14}$$

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

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

Some examples:

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

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

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

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

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

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

In [55], the above formulae are named *s-stage trapezoidal methods*. They provide exact conservation for polynomial Hamiltonian functions of degree no larger than $2\lceil\frac{\nu}{2}\rceil$, for all $\nu \geq 1$. Their order of accuracy can be easily determined by recasting (1.13)–(1.14) as a ν-stage Runge-Kutta method,

$$\begin{array}{c|c} \mathbf{c} & \mathbf{c}\mathbf{b}^\top \\ \hline & \mathbf{b}^\top \end{array} \qquad \text{with} \qquad \mathbf{c} = (c_1, \ldots, c_\nu)^\top \quad \text{and} \quad \mathbf{b} = (b_1, \ldots, b_\nu)^\top. \tag{1.15}$$

The order conditions for an ν-stage Runge-Kutta method are established in the following theorem.

Theorem 1.2.1 (Butcher, [33]). *If a Runge-Kutta method with coefficients b_i, c_i, a_{ij}, $i, j = 1, \ldots, \nu$, satisfies the following conditions:*

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

with

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

then it has order p.

As a matter of facts, (1.15) satisfies conditions $B(2)$ and $C(1)$, thus resulting in a second order method. In fact:

- the quadrature is exact for polynomials of degree 1, so that $B(2)$ holds true;
- by setting $\mathbf{e} = (1, \ldots, 1)^\top \in \mathbb{R}^\nu$, one has

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

Remark 1.2.1. *It is worth noticing that, even though (1.15) is formally a ν-stage implicit Runge-Kutta method, nevertheless the actual size of the generated discrete problem, consists of only* one *nonlinear equation, in the unknown y_1, as the above examples clearly show. This peculiarity is due to the fact that the Butcher matrix of these methods (i.e. $\mathbf{c}\mathbf{b}^\top$), has rank one.*

1.3 Generalizing the approach

In this section we are going to generalize the above approach by considering a polynomial path σ of degree $s \geq 1$. Let us expand the derivative of σ along a suitable basis for Π_{s-1}, call it $\{P_0, \ldots, P_{s-1}\}$, as

$$\sigma'(ch) = \sum_{j=0}^{s-1} P_j(c)\gamma_j, \qquad c \in [0, 1], \tag{1.16}$$

for a certain set of coefficient $\{\gamma_j\}$ to be determined. By integrating both sides and imposing the initial condition

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

one formally obtains

$$\sigma(ch) = y_0 + h\sum_{j=0}^{s-1}\int_0^c P_j(x)\mathrm{d}x\gamma_j, \qquad c \in [0,1], \tag{1.17}$$

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

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

provided that the unknown coefficients γ_j satisfy

$$\gamma_j = \eta_j J\int_0^1 \nabla H(\sigma(ch))P_j(c)\mathrm{d}c = \eta_j\int_0^1 f(\sigma(ch))P_j(c)\mathrm{d}c, \quad j = 0,\ldots,s-1, \tag{1.18}$$

for a suitable set of nonzero scalars $\eta_0,\ldots,\eta_{s-1}$. The new approximation is obtained by plugging (1.18) into (1.17):

$$y_1 \equiv \sigma(h) = y_0 + h\sum_{j=0}^{s-1}\eta_j\int_0^1 P_j(x)\mathrm{d}x\int_0^1 P_j(\tau)f(\sigma(\tau h))\mathrm{d}\tau. \tag{1.19}$$

Assuming, as before, $H \in \Pi_\nu$, the integrands in (1.18) and (1.19) are polynomials of degree at most $(\nu-1)s + s - 1 \equiv \nu s - 1$. Therefore, by means of a quadrature formula such that it is exact for polynomials of degree $\nu s - 1$, the integrals in (1.18) and (1.19) may be exactly evaluated. Let $0 \le c_1 < \ldots < c_k \le 1$ and $\{b_1,\ldots,b_k\}$ denote, respectively, the abscissae and the weights of the chosen quadrature formula. We obtain

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

and

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

respectively. By setting, as before,

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

one has:

$$Y_i = y_0 + h\sum_{j=1}^{k}\overbrace{\left[b_j\sum_{\ell=0}^{s-1}\eta_\ell P_\ell(c_j)\int_0^{c_i}P_\ell(x)\mathrm{d}x\right]}^{=a_{ij}} f(Y_j) \equiv y_0 + h\sum_{j=1}^{k}a_{ij}f(Y_j), \tag{1.20}$$

$$i = 1,\dots,k,$$

$$y_1 = y_0 + h\sum_{i=1}^{k}\underbrace{\left[b_i\sum_{\ell=0}^{s-1}\eta_\ell P_\ell(c_i)\int_0^{1}P_\ell(x)\mathrm{d}x\right]}_{=\hat{b}_i} f(Y_i) \equiv y_0 + h\sum_{i=1}^{k}\hat{b}_i f(Y_i). \tag{1.21}$$

What we have obtained is the k-stage Runge-Kutta method

$$\begin{array}{c|c}\mathbf{c} & A\equiv(a_{ij})\in\mathbb{R}^{k\times k}\\ \hline & \hat{\mathbf{b}}^\top\end{array} \qquad \text{with} \qquad \mathbf{c}=(c_1,\dots,c_k)^\top, \qquad \hat{\mathbf{b}}=(\hat{b}_1,\dots,\hat{b}_k)^\top, \tag{1.22}$$

with $a_{ij},\hat{b}_i$ defined according to (1.20) and (1.21), respectively.

In so doing, under the assumption that the Hamiltonian is a polynomial, we can always achieve energy conservation, provided that the quadrature has a suitable high order. As we will see, the best choice is placing the k abscissae $\{c_i\}$ at the k Gauss-Legendre nodes on $[0,1]$ thus obtaining the maximum order $2k$. In such a case, energy conservation is guaranteed for polynomial Hamiltonians of degree ν such that

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

However, it is quite difficult to discuss the order of accuracy and the properties of the k-stage Runge-Kutta method (1.22), when a generic polynomial basis is considered. In fact, different choices of the basis provide different methods and having different orders. As an example, fourth-order energy-conserving Runge-Kutta methods were derived in [56, 57], by using the Newton polynomial basis. We shall see that things will greatly simplify, by choosing an orthonormal polynomial basis.

Remark 1.3.1. *It is worth noticing that the Butcher tableau of the k-stage Runge-Kutta method (1.22) can be cast in matrix form by introducing the matrices:*

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

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

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

and the row vector

$$\mathcal{I}_s^1 = \begin{pmatrix}\int_0^1 P_0(x)\mathrm{d}x & \dots & \int_0^1 P_{s-1}(x)\mathrm{d}x\end{pmatrix}. \tag{1.23}$$

In fact, one easily checks that (1.22) becomes

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

which will be further studied later.

Chapter 2

Background results

In the present chapter we sketch some results about Legendre polynomials and perturbation of ODEs that will be useful in the sequel.

2.1 Legendre polynomials

In the following we will denote by P_i the Legendre polynomials shifted on the interval $[0,1]$ and scaled in order to be orthonormal:

$$\deg P_i = i, \qquad \int_0^1 P_i(x)P_j(x)\mathrm{d}x = \delta_{ij}, \qquad \forall i,j \geq 0, \tag{2.1}$$

where δ_{ij} is the Kronecker symbol. As any family of orthogonal polynomials, these polynomials satisfy a 3-terms recursive formula, given by:

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

The roots $\{c_1, \ldots, c_k\}$ of $P_k(x)$ are all distinct and belong to the interval $(0,1)$, thus we can identify them via the following conditions:

$$P_k(c_i) = 0, \qquad \text{with} \qquad 0 < c_1 < \ldots < c_k < 1. \tag{2.2}$$

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

$$c_i = 1 - c_{k-i+1}, \qquad i = 1, \ldots, k. \tag{2.3}$$

They are referred to as the Gauss-Legendre abscissae on $[0,1]$ and generate the Gauss-Legendre quadrature formula with quadrature weights

$$b_i = \int_0^1 \prod_{j=1, j\neq i}^{k} \frac{x - c_j}{c_i - c_j}\mathrm{d}x = \frac{4(2k-1)c_i(1-c_i)}{[kP_{k-1}(c_i)]^2}, \qquad i = 1, \ldots, k. \tag{2.4}$$

Concerning the Gauss-Legendre quadrature formula, the following theorem holds true.

Theorem 2.1.1. *The Gauss-Legendre formula (c_i, b_i) (see (2.2) and (2.4)) has order $2k$, namely it is exact for polynomials of degree no larger than $2k-1$.*

Proof. Given a generic polynomial $p(x) \in \Pi_{2k-1}$, it can be written as

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

Integrating both sides, since $P_k(x)$ is orthonormal to polynomials of degree less than k (see (2.1)), we have

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

On the other hand, for our quadrature formula (c_i, b_i) one obtains:

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

where the last equality is due to the fact that any quadrature based at k distinct abscissae is exact for polynomial of degree no larger than $k-1$. □

As a matter of facts, for the Gauss-Legendre formula and for any function $f \in C^{2k}([0,1])$, one has

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

for a suitable $\zeta \in (0,1)$ and with ρ_k independent of f. Actually, this result holds in general: if the quadrature had order $q \leq 2k$, for any $f \in C^q([0,1])$, one would obtain,

$$\int_0^1 f(x)\mathrm{d}x = \sum_{i=1}^k b_i f(c_i) + \Delta_k \qquad \Delta_k = \rho_k f^{(q)}(\zeta), \tag{2.5}$$

with ζ and ρ_k defined similarly as above.

In the sequel, we shall need to discuss, in particular, the case where the integrand in (2.5) has the following form,

$$f(\tau) = P_j(\tau)G(\tau h), \qquad \tau \in [0,1], \tag{2.6}$$

with P_j the Legendre polynomial of degree j. The following result then holds true.

Lemma 2.1.1. *Let $G \in C^{(q)}([0,1])$, being q the order of the given quadrature formula (c_i, b_i) over the interval $[0,1]$. Then*

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

Proof. Our claim easily follows from (2.5), by considering that

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

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

We also need a further result concerning integrals with integrands in the form (2.6), which is stated below.

Lemma 2.1.2. *Let $G : [0, h] \to V$, with V a suitable vector space, a function which admits a Taylor expansion at 0. Then*

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

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

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

where the last but one equality follows from the fact that

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

because of (2.1). □

2.2 Matrices defined by the Legendre polynomials

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

$$\int_0^c P_0(x)\mathrm{d}x = \xi_1 P_1(c) + \frac{1}{2}P_0(c), \qquad \int_0^c P_i(x)\mathrm{d}x = \xi_{i+1}P_{i+1}(c) - \xi_i P_{i-1}(c), \quad i \geq 1, \tag{2.7}$$

$$\text{with} \quad \xi_i = \frac{1}{2\sqrt{4i^2-1}}. \tag{2.8}$$

Remark 2.2.1. *From the orthogonality conditions (2.1), and taking into account that $P_0(x) \equiv 1$, one obtains:*

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

Moreover, since the Legendre polynomials satisfy the following symmetry property:

$$P_j(c) = (-1)^j P_j(1-c), \qquad c \in [0, 1], \qquad j \geq 0, \tag{2.9}$$

then their integrals share a similar symmetry:

$$\int_{\tau_1}^{\tau_2} P_j(x)\mathrm{d}x = (-1)^j \int_{1-\tau_2}^{1-\tau_1} P_j(x)\mathrm{d}x, \qquad \forall \tau_1, \tau_2 \in [0, 1], \quad j \geq 0. \tag{2.10}$$

In the sequel, we shall use the following matrices, defined by means of the Legendre polynomials evaluated at the $k \geq s$ abscissae (2.2):[1]

$$\mathcal{P}_s = \begin{pmatrix} P_0(c_1) & \dots & P_{s-1}(c_1) \\ \vdots & & \vdots \\ P_0(c_k) & \dots & P_{s-1}(c_k) \end{pmatrix} \in \mathbb{R}^{k\times s}, \tag{2.11}$$

[1] Such matrices have been formally introduced, for a generic polynomial basis, at the end of Chapter 1.

and

$$\mathcal{I}_s = \begin{pmatrix} \int_0^{c_1} P_0(x)\mathrm{d}x & \dots & \int_0^{c_1} P_{s-1}(x)\mathrm{d}x \\ \vdots & & \vdots \\ \int_0^{c_k} P_0(x)\mathrm{d}x & \dots & \int_0^{c_k} P_{s-1}(x)\mathrm{d}x \end{pmatrix} \in \mathbb{R}^{k\times s}. \tag{2.12}$$

Because of (2.7), one obtains the following relation:

$$\mathcal{I}_s = \mathcal{P}_{s+1}\hat{X}_s, \qquad \hat{X}_s = \left(\begin{array}{cccc} \frac{1}{2} & -\xi_1 & & \\ \xi_1 & 0 & \ddots & \\ & \ddots & \ddots & -\xi_{s-1} \\ & & \xi_{s-1} & 0 \\ \hline & & & \xi_s \end{array}\right) \equiv \left(\begin{array}{c} X_s \\ \hline 0 \dots 0\ \xi_s \end{array}\right). \tag{2.13}$$

We also set

$$\Omega = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_k \end{pmatrix} \in \mathbb{R}^{k\times k}, \tag{2.14}$$

the diagonal matrix with the corresponding Gauss-Legendre weights. The following simple properties then hold true [9].

Lemma 2.2.1.

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

Proof. Our claim easily follows from the Laplace expansion, by considering that, from (2.13), $\det(X_1) = \frac{1}{2}$ and $\det(X_2) = \xi_1^2$. □

Theorem 2.2.1. *Matrices (2.11) and (2.12) have full column rank, for all $s = 1, \dots, k$. Moreover,*

$$\mathcal{P}_s^\top \Omega \mathcal{P}_{s+1} = (I_s\ \mathbf{0}). \tag{2.15}$$

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

$$\mathcal{P}_{k+1} = (\mathcal{P}_k\ \mathbf{0}), \tag{2.16}$$

since the entries in last column are $P_k(c_i) = 0$, $i = 1, \dots, k$. As a consequence, because of (2.13), for matrix $\mathcal{I}_s$ one obtains:

- when $s < k$, then both $\mathcal{P}_{s+1}$ and $\hat{X}_s$ have full column rank and so has $\mathcal{I}_s$;
- when $s = k$, then from (2.16) it follows that

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

  and both $\mathcal{P}_k$ and X_k are nonsingular (see Lemma 2.2.1).

Concerning (2.15), by considering that the quadrature formula (c_i, b_i) is exact for polynomials of degree no larger than $2k-1 \geq 2s-1$, and setting $\mathbf{e}_i \in \mathbb{R}^s$ and $\hat{\mathbf{e}}_j \in \mathbb{R}^{s+1}$ the i-th and j-th unit vectors, one has:

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

so that (2.15) is verified. □

From the previous theorem, the following result easily follows.

Corollary 2.2.1. *When $k = s$, then $\mathcal{P}_s^{-1} = \mathcal{P}_s^\top \Omega$.*

2.3 Additional preliminary results

To conclude this chapter, we present some perturbation results, that will be useful in order to carry out a complete analysis of the methods, concerning the initial value problem for ordinary differential equations

$$y'(t) = f(y(t)), \qquad t \geq t_0, \qquad y(t_0) = y_0. \tag{2.17}$$

We denote the solution of (2.17) by $y(t; t_0, y_0)$, in order to emphasize the dependence on the initial condition.

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

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

where the derivative (i.e.,$'$) is with respect to t, and J_f is the Jacobian matrix of $f(y)$.

The following result then holds true.

Lemma 2.3.1. *With reference to the solution $y(t; t_0, y_0)$ of problem (2.17), one has:*

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

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

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

Therefore, by setting

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

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

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

The solution of this linear problem is easily seen to be

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

and, consequently,

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

i.e., (i) is proved.

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

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

Consequently, the solution of the perturbed problem

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

coincides, at first order, with that of the problem

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

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

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

This concludes the proof. □

Chapter 3

A framework for HBVMs

In this chapter we introduce a new class of methods for the numerical solution of canonical Hamiltonian systems, named *Hamiltonian Boundary Value Methods (HBVMs)*. The main feature of these methods is that they are able to exactly preserve, in the numerical solution, the value of the Hamiltonian function when it is a polynomial. We will see that this property also implies a *practical* conservation of any analytical Hamiltonian function. The basic idea which HBVMs rely on is the discrete line integral, already sketched in Chapter 1, i.e., the discrete counterpart of the line integral associated with a conservative vector field.

In this chapter we provide a novel framework for discussing the order, the linear stability and conservation properties of HBVMs, based on a local Fourier expansion of the vector field defining the dynamical systems.

3.1 Local Fourier expansion

In Chapter 2 we introduced the Legendre polynomials which constitute an *orthonormal basis* for the functions defined on the interval $[0,1]$. Therefore, by using the notation (1.12), we can formally expand, over the interval $[0,h]$, the right hand side of the differential equation in (1.1), as follows:

$$f(y(ch)) = \sum_{j\geq 0} P_j(c)\gamma_j(y), \qquad c\in[0,1], \tag{3.1}$$

where

$$\gamma_j(y) = \int_0^1 P_j(\tau) f(y(\tau h))\mathrm{d}\tau, \qquad j\geq 0. \tag{3.2}$$

The expansion (3.1)–(3.2) is known as the *Neumann expansion* of an analytic function [76, p. 322], and converges uniformly provided that the function $g(c) = f(y(ch))$ has continuous derivative [58, p. 206]. However, for sake of simplicity, hereafter we shall assume $g(c)$ to be analytic.

In so doing, we are transforming the initial value problem

$$y'(t) = f(y(t)), \qquad t\in[0,h], \qquad y(0) = y_0, \tag{3.3}$$

into the equivalent *integro-differential* problem

$$y'(ch) = \sum_{j\geq 0} P_j(c) \int_0^1 P_j(\tau) f(y(\tau h))\mathrm{d}\tau, \qquad c\in[0,1], \qquad y(0) = y_0. \tag{3.4}$$

In order to obtain a polynomial approximation of degree s to (3.3)–(3.4), we just truncate the infinite series to a finite sum. The resulting initial value problem is (see (3.2))

$$\begin{aligned}\sigma'(ch) &= \sum_{j=0}^{s-1} P_j(c)\int_0^1 P_j(\tau)f(\sigma(\tau h))\mathrm{d}\tau \equiv \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma), \qquad c\in[0,1], \\ \sigma(0) &= y_0,\end{aligned} \tag{3.5}$$

whose solution is a polynomial $\sigma\in\Pi_s$. In fact, by integrating both sides of (3.5) and taking into account the initial condition, we have

$$\sigma(ch) = y_0 + h\sum_{j=0}^{s-1}\int_0^c P_j(x)\mathrm{d}x\gamma_j(\sigma), \qquad c\in[0,1]. \tag{3.6}$$

One easily recognizes that (3.5)–(3.6) define the very same expansion (1.16)–(1.18) with all $\eta_j = 1$. Consequently, such a method is energy-conserving, if we are able to exactly compute the integrals providing the coefficients $\gamma_j(\sigma)$ at the right-hand side in (3.2). From (3.6) one obtains that

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

In order to discuss the order of the approximation $\sigma(h)\approx y(h)$, we state the following result whose proof is a direct consequence of (3.2) and Lemma 2.1.2.

Lemma 3.1.1. *Let $\gamma_j(\sigma)$ be defined according to (3.2). Then $\gamma_j(\sigma) = O(h^j)$.*

We are now able to prove the following result.

Theorem 3.1.1. $\sigma(h) - y(h) = O(h^{2s+1})$.

Proof. Denoting by $y(t;t_0,y_0)$ the solution of problem (2.17) and considering that $\sigma(0) = y_0$, by virtue of (3.1), (3.5), and Lemmas 2.1.2, 2.3.1, and 3.1.1, one has:

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

□

We observe, however, that (3.5) is *not yet* an operative method, but rather a *formula* since, quoting Dahlquist and Björk [38], "as is well known, even many relatively simple integrals cannot be expressed in finite terms of elementary functions, and thus must be evaluated by numerical methods". In other words, in order to obtain an actual *numerical method*, we need to approximate the integrals appearing in (3.5) by means of a suitable quadrature formula. We assume to use a quadrature (c_i, b_i) over k distinct abscissae, and, as a consequence, in place of σ defined by (3.5) or (3.6), we shall compute the new polynomial $u \in \Pi_s$ such that

$$\begin{aligned} u'(ch) &= \sum_{j=0}^{s-1} P_j(c) \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)), \qquad c \in [0,1], \\ u(0) &= y_0, \end{aligned} \tag{3.7}$$

or, equivalently,

$$u(ch) = y_0 + h \sum_{j=0}^{s-1} \int_0^c P_j(x)\mathrm{d}x \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)), \qquad c \in [0,1], \tag{3.8}$$

with the new approximation given by

$$y_1 \equiv u(h) = y_0 + h \sum_{i=1}^{k} b_i f(u(c_i h)). \tag{3.9}$$

If the quadrature formula (c_i, b_i) has order $q \geq s$, then, by virtue of Lemma 2.1.1 and taking into account (3.2), one obtains

$$\begin{aligned} \gamma_j(u) &\equiv \int_0^1 P_j(\tau) f(u(\tau h))\mathrm{d}\tau = \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)) + \Delta_j(h), \\ \Delta_j(h) &= O(h^{q-j}), \quad j = 0, \ldots, s-1. \end{aligned} \tag{3.10}$$

Consequently, we can rewrite the first equation in (3.7) in the following equivalent form:

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

This allows us to derive the following result.

Theorem 3.1.2. *$y_1 - y(h) = O(h^{p+1})$, where $p = \min\{q, 2s\}$.*

Proof. The proof proceeds on the same line as that of Theorem 3.1.1:

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

□

Definition 3.1.1. *The method (3.7)–(3.9) is named* Hamiltonian Boundary Value Method (HBVM) with k stages and degree s, *in short HBVM*(k,s).

On the basis of the result of Theorem 3.1.2, it appears natural to choose the k abscissae as the $k \ge s$ Gauss-Legendre abscissae on $[0,1]$ defined in (2.2), so that the order of the quadrature is maximized. As we have seen in Chapter 2, the Gauss-Legendre quadrature formula with k points has order $q = 2k$. As a consequence, the following result holds true.

Corollary 3.1.1. *By choosing the k abscissae $\{c_i\}$ as in (2.2), a HBVM(k,s) method has order $2s$, for all $k \ge s$.*

For this reason, we shall always consider in the sequel a k-points Gauss-Legendre formula for the quadrature of the integrals defining the coefficients $\gamma_j(\sigma)$ in (3.2).

3.2 Runge-Kutta form of HBVMs

In this section we show that a HBVM(k,s) method admits a Runge-Kutta formulation. The basic fact is that, at right hand side of equations (3.8)–(3.9), one only needs to know the value of the polynomial u at the abscissae $\{c_i h\}$. By setting

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

one obtains:

$$Y_i = y_0 + h\sum_{j=1}^{k}\overbrace{\left[b_j\sum_{\ell=0}^{s-1}P_\ell(c_j)\int_0^{c_i}P_\ell(x)\mathrm{d}x\right]}^{a_{ij}} f(Y_j) \equiv y_0 + h\sum_{j=1}^{k}a_{ij}f(Y_j), \tag{3.11}$$
$$i = 1,\dots,k,$$

$$y_1 = y_0 + h\sum_{i=1}^{k}b_i f(Y_i). \tag{3.12}$$

In such a way, we have defined the following k-stage Runge-Kutta method:

$$\begin{array}{c|c} \mathbf{c} & A \equiv (a_{ij}) \\ \hline & \mathbf{b}^\top \end{array} \tag{3.13}$$

with (see (3.11)),

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

The Butcher tableau (3.13) defines the *Runge-Kutta shape of a HBVM(k,s) method*. The Butcher matrix A in (3.13) can be easily written in a more compact form.

Theorem 3.2.1. *$A = \mathcal{I}_s\mathcal{P}_s^\top\Omega$, with the matrices $\mathcal{I}_s$, $\mathcal{P}_s$ and Ω defined according to (2.11)–(2.14).*

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

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

according to (3.11). □

Consequently, the Butcher tableau (3.13) can be rewritten as:

$$\begin{array}{c|c} \mathbf{c} & \mathcal{I}_s\mathcal{P}_s^\top\Omega \\ \hline & \mathbf{b}^\top \end{array} \tag{3.14}$$

or, equivalently, by taking into account (2.13),

$$\begin{array}{c|c} \mathbf{c} & \mathcal{P}_{s+1}\hat{X}_s\mathcal{P}_s^\top\Omega \\ \hline & \mathbf{b}^\top \end{array}. \tag{3.15}$$

Remark 3.2.1. *We observe that the Runge-Kutta form (3.14) of a HBVM(k, s) method is the same obtained in Remark 1.3.1 for a discrete line-integral method defined by using a general polynomial basis, but with the diagonal matrix Λ_s now automatically fixed in order to maximize the order of accuracy of the method. Moreover, the vector of the quadrature weights coincides with that used for approximating the integrals involved in the coefficients of the polynomial u.*

3.3 HBVM(s,s)

In the particular case $k=s$ the matrices $\mathcal{I}_s, \mathcal{P}_s, \Omega \in \mathbb{R}^{s\times s}$. From Theorem 2.2.1 and Corollary 2.2.1 one has:

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

Consequently, we can write the Butcher tableau (3.14) as that of the following s-stage method,

$$\begin{array}{c|c} \mathbf{c} & \mathcal{P}_s X_s \mathcal{P}_s^{-1} \\ \hline & \mathbf{b}^\top \end{array} \tag{3.16}$$

resulting in the *W-transformation* defining the s-stage Gauss-Legendre Runge-Kutta collocation method [50, p. 79] which has order $2s$. In this sense, in the case $k \geq s$, HBVM(k,s) can be regarded as *low-rank generalizations* of the s-stage Gauss method. Indeed, the following result, known as *isospectrality of HBVMs* [23], holds true.

Theorem 3.3.1. *For all $k \geq s$ the rank of the matrix $A = \mathcal{P}_{s+1}\hat{X}_s\mathcal{P}_s^\top\Omega$ is s. Moreover, the non-zero eigenvalues of A coincide with those of the underlying s-stage Gauss method.*

Proof. The rank of the matrix $\mathcal{P}_{s+1}$ is s or $s+1$ (when $k>s$), whereas the rank of the matrices $\hat{X}_s, \mathcal{P}_s$ is s and Ω is nonsingular. Therefore, the rank of A cannot exceed s. Moreover, from Theorem 2.2.1, one has

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

which is known to be nonsingular (see Lemma 2.2.1). Consequently, $\mathrm{rank}(A) = s$. Concerning the second part of the proof, taking into account the result of Theorem 2.2.1, one has

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

This means that the columns of $\Omega\mathcal{P}_s$ span an s-dimensional left invariant subspace of A. Therefore, the eigenvalues of X_s will coincide with the non-zero eigenvalues of A. On the other hand, from (3.16) one obtains immediately that the eigenvalues of X_s are the eigenvalues of the Butcher matrix of the s-stage Gauss method. □

3.4 Energy conservation

We now consider the issue of energy conservation for HBVM(k,s) methods. From (3.7)–(3.9) with $f = J\nabla H$, one obtains:

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

where in the last but one equality we have exploited the orthogonality of matrix J. At this point, two possibilities may occur:

- $\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau = \sum_{i=1}^{k} b_i P_j(c_i) J\nabla H(u(c_i h))$, that is, the Gauss-Legendre quadrature formula is exact for the integral appearing in $\gamma_j(u)$. This is the case of a polynomial Hamiltonian of degree ν no larger than $2k/s$. In this case, $E_H = 0$, so that energy is *exactly conserved*;

- $\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau = \sum_{i=1}^{k} b_i P_j(c_i) J\nabla H(u(c_i h)) + \Delta_j(h)$, that is, the Gauss-Legendre quadrature formula of order $q = 2k$ is not exact for the integral in the expression of $\gamma_j(u)$, but, provided that the Hamiltonian H is suitably regular, as we have assumed, and according to (3.10), it gives an error $\Delta_j(h) = O(h^{2k-j})$. In such a case, by taking into account the result of Lemma 2.1.2 and the skew-symmetry of J, one has:

$$\begin{aligned} E_H &= h\sum_{j=0}^{s-1}\left[\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau\right]^\top J\left[\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau - \Delta_j(h)\right] \\ &= -h\sum_{j=0}^{s-1}\underbrace{\left[\int_0^1 P_j(\tau) J\nabla H(u(\tau h))\mathrm{d}\tau\right]^\top}_{O(h^j)} J\underbrace{\Delta_j(h)}_{O(h^{2k-j})} = O(h^{2k+1}). \end{aligned}$$

We have then proved the following result.

Theorem 3.4.1. *HBVM(k, s) is energy-conserving for all polynomial Hamiltonians of degree*

$$\nu \le \frac{2k}{s}. \tag{3.17}$$

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

Remark 3.4.1. *As a consequence of Theorem 3.4.1, one can observe that*

- *for polynomials Hamiltonian of* any *degree, energy conservation can* always *be obtained, by choosing k large enough so that* (3.17) *is satisfied;*

- *even in the case of non polynomial, but suitably regular, Hamiltonians, energy conservation can still be* practically *gained by choosing k large enough, so that $|E_H|$, which is $O(h^{2k+1})$, falls within round-off errors.*

As we will see in the next chapter, choosing k large enough is not a drawback from a computational point of view since, as a consequence of the isospectrality property of HBVMs, the computational cost for the implementation of these methods essentially depends on s rather than on k.

To show the validity of our results and the potential advantage of using energy-conserving methods, we consider, as a first example, the Hamiltonian problem with Hamiltonian (1.7) and parameters (1.8). For this problem we have plotted in Figure 1.1 the level curves passing through the points defined in (1.9) and in Figure 1.2 the wrong phase portrait that one obtains by using the symplectic 2-stage Gauss method (fourth-order), with stepsize $h = 10^{-4}$, corresponding to the error in the numerical Hamiltonian shown in Figure 3.1. Indeed, even though no drift in the Hamiltonian occurs, nevertheless such error is not negligible for the problem at hand.

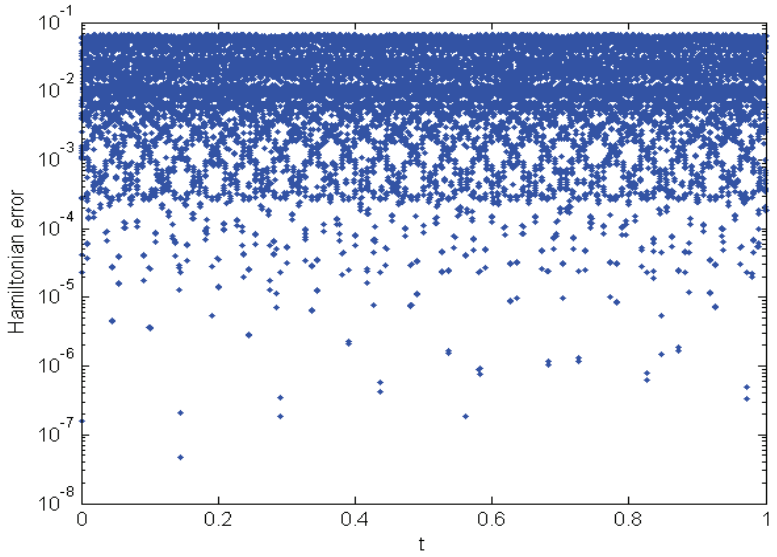

Figure 3.1: Hamiltonian error for problem (1.7)–(1.9), 2-stage Gauss method, $h = 10^{-4}$.

However, if we use the HBVM(3,2) method (fourth-order) with the same stepsize, the error in the Hamiltonian is of sixth-order: this is enough to have a smaller error in the numerical Hamiltonian, as is shown in Figure 3.2, resulting in a correct phase portrait, as is shown in Figure 3.3.

At last, by using the HBVM(10,2) method (fourth-order) with the same stepsize, the Hamiltonian error is within roundoff errors, as is shown in Figure 3.4, thus allowing a perfect reconstruction of the phase portrait, depicted in Figure 3.5.

For sake of completeness, in Figure 3.6 we also plot the mean error in the numerical Hamiltonian for HBVM$(k, 2)$ methods (all of order 4), used with the stepsize $h = 10^{-4}$, for $k = 2, \dots, 10$. As one can see for $k \geq 6$ the error is essentially due to roundoff.

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

$$H(q, p) = (q^2 + p^2)^2 - 10(q^2 - p^2). \tag{3.18}$$

The level curves for this problem are the Cassini ovals and in Figure 3.7 we plot the one passing at

$$(q_0, p_0) = (0, 10^{-5}). \tag{3.19}$$

By using the 2-stage Gauss method with stepsize $h = 10^{-2}$, the obtained phase portrait is "almost" correct, at a first sight, as one can see in Figure 3.8. This portrait is, actually, qualitatively wrong as one can realize by zooming around the origin (see Figure 3.10), due to the error in the Hamiltonian displayed in Figure 3.14. Moreover, comparing the correct profile of component q and the approximated one, as shown in Figure 3.12 (similar results are obtained for p), one can observe that the error in the Hamiltonian results in the loss of periodicity for the numerical solution.

By using the HBVM$(4, 2)$ method with the same stepsize, since the Hamiltonian (3.18) is a polynomial of degree 4, according to (3.17), the error on the Hamiltonian is of the order of roundoff errors, as is confirmed by the plot in Figure 3.15. This makes it possible to obtain a phase portrait (see Figure 3.9) qualitatively correct, as is confirmed by the zoom in Figure 3.11. Moreover, one also obtains that the periodicity of the solution is maintained, resulting in a profile of the numerical approximation of the component q (similarly for p) which well matches the correct one, as is shown in Figure 3.13.

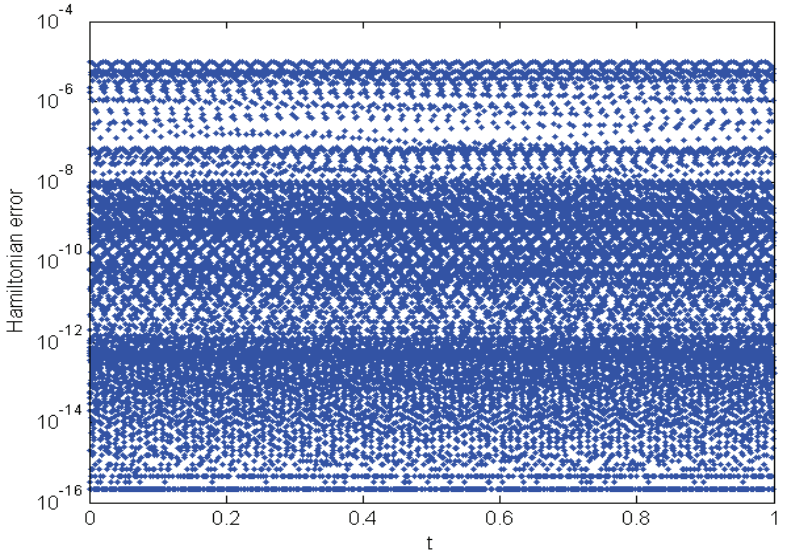

Figure 3.2: Hamiltonian error for problem (1.7)–(1.9), HBVM(3,2) method, $h = 10^{-4}$.

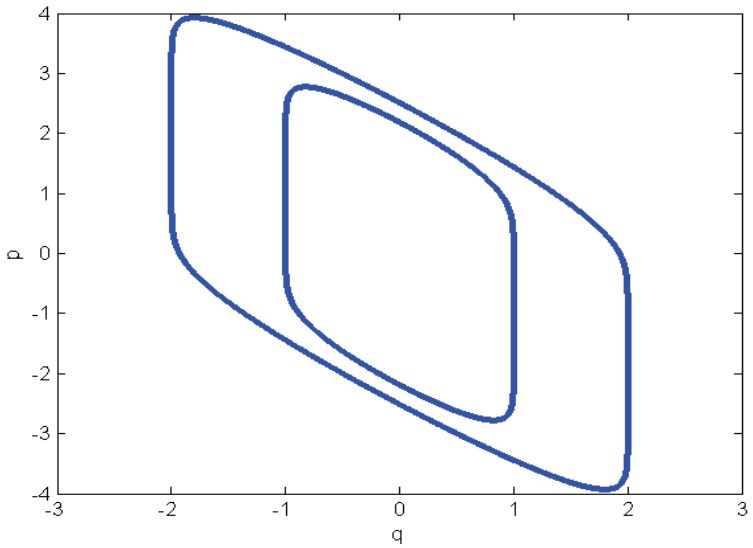

Figure 3.3: Numerical level curves for problem (1.7)–(1.9), HBVM(3,2) method, $h = 10^{-4}$.

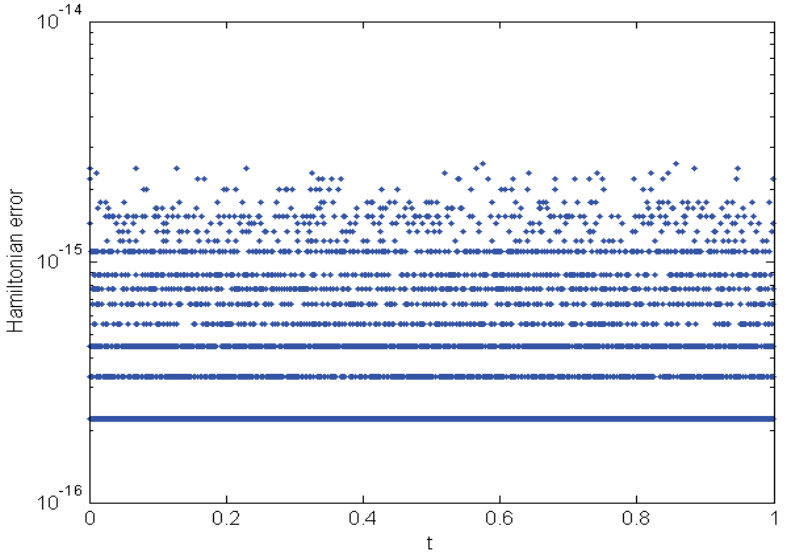

Figure 3.4: Hamiltonian error for problem (1.7)–(1.9), HBVM(10,2) method, $h = 10^{-4}$.

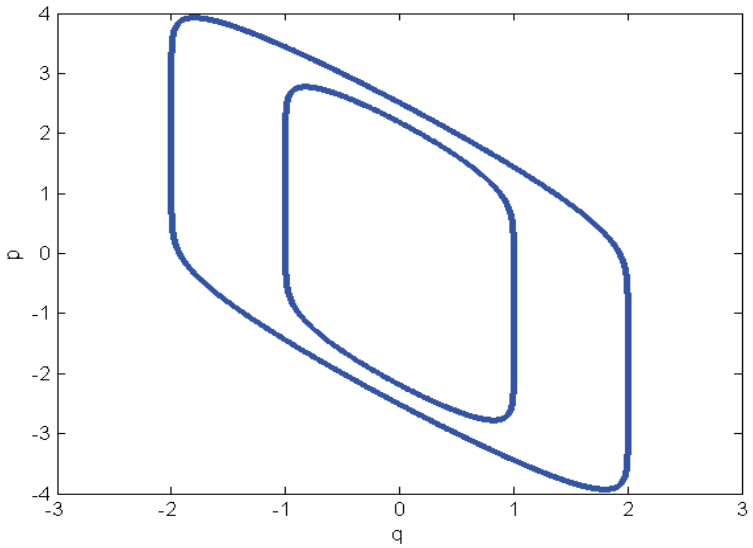

Figure 3.5: Numerical level curves for problem (1.7)–(1.9), HBVM(10,2) method, $h = 10^{-4}$.

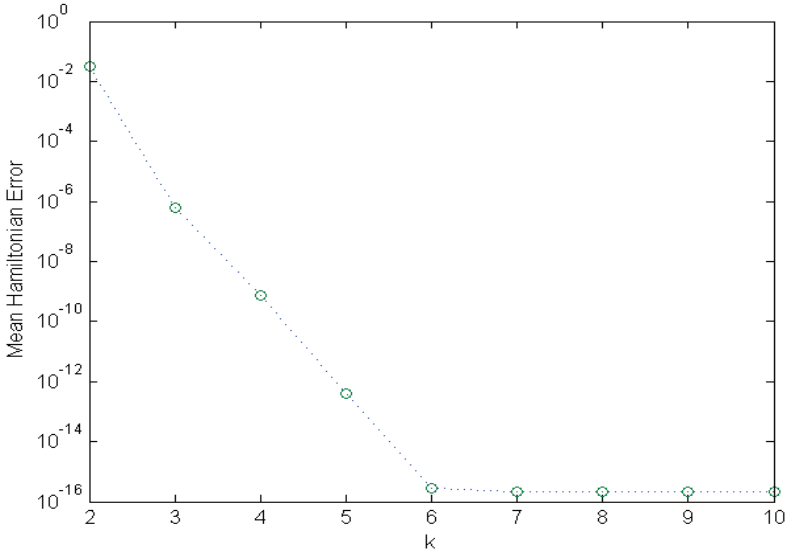

Figure 3.6: Mean Hamiltonian error for problem (1.7)–(1.9), HBVM$(k,2)$ method, $k = 2,\dots,10$, by using a stepsize $h = 10^{-4}$.

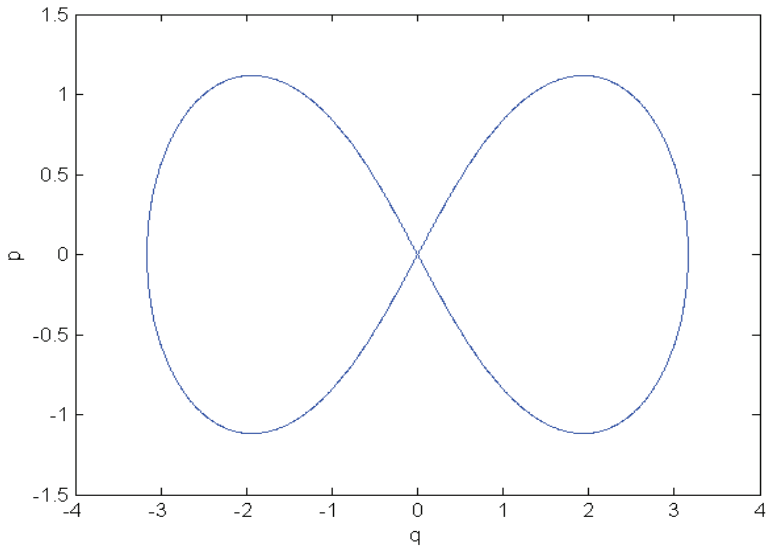

Figure 3.7: Level curve for problem (3.18)–(3.19).

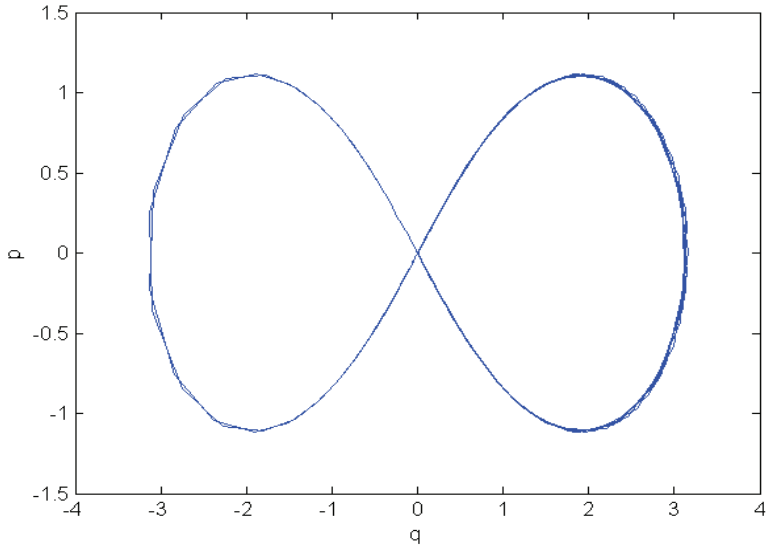

Figure 3.8: Numerical level curve for problem (3.18)–(3.19), 2-stage Gauss method, $h = 10^{-2}$.

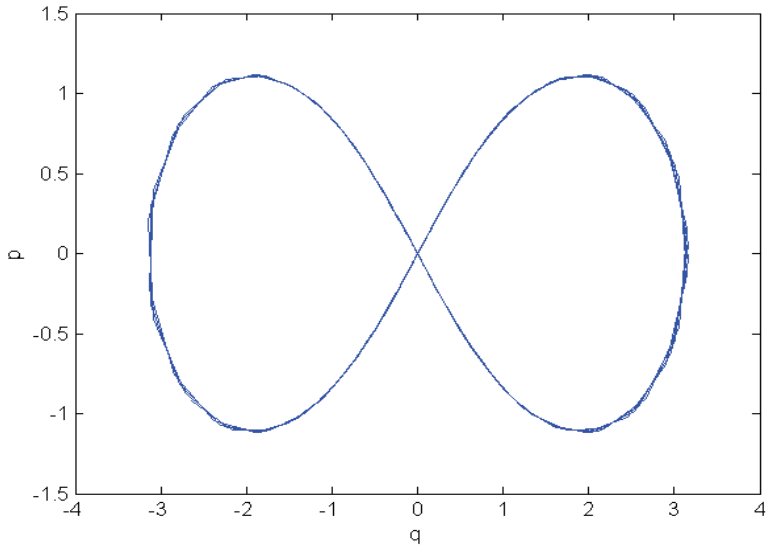

Figure 3.9: Numerical level curve for problem (3.18)–(3.19), HBVM(4,2) method, $h = 10^{-2}$.

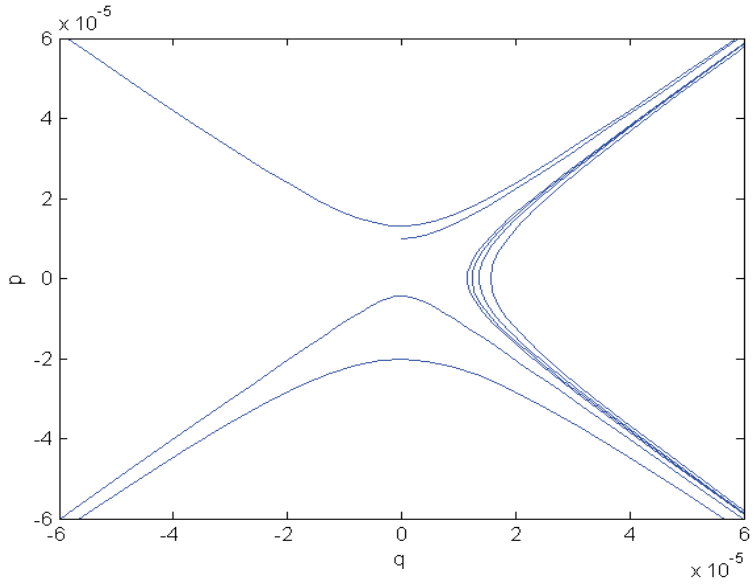

Figure 3.10: Zoom of the numerical level curve for problem (3.18)–(3.19) around $(0,0)$, 2-stage Gauss method, $h = 10^{-2}$.

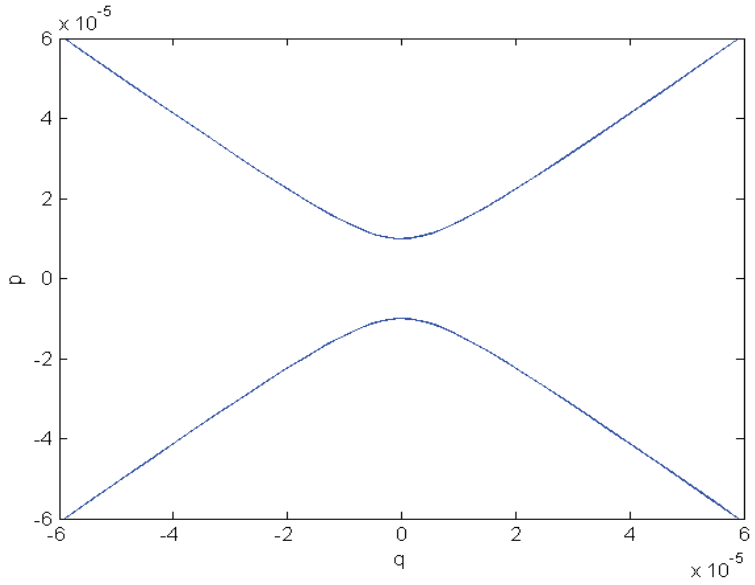

Figure 3.11: Zoom of the numerical level curve for problem (3.18)–(3.19) around $(0,0)$, HBVM(4,2) method, $h = 10^{-2}$.

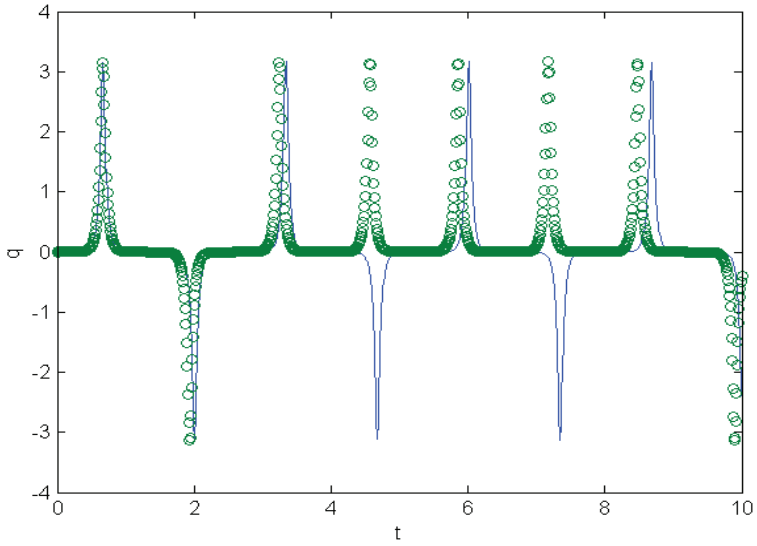

Figure 3.12: Component q (solid line) and its numerical approximation (circles) by using the 2-stage Gauss method, $h = 10^{-2}$, for problem (3.18)–(3.19).

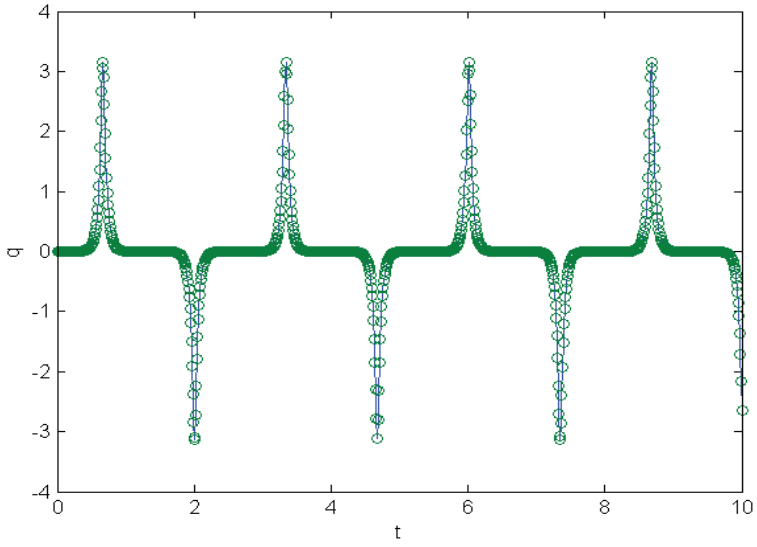

Figure 3.13: Component q (solid line) and its numerical approximation (circles) by using the HBVM(4,2) method, $h = 10^{-2}$, for problem (3.18)–(3.19).

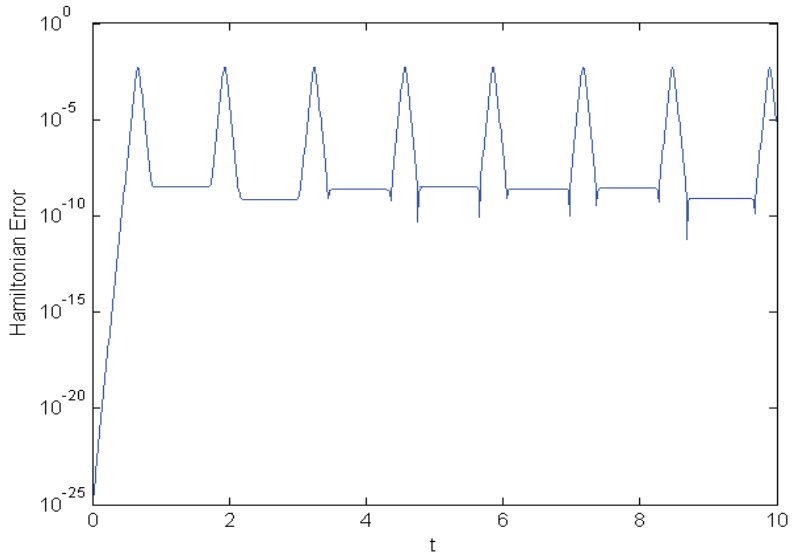

Figure 3.14: Hamiltonian error for problem (3.18)–(3.19) by using the 2-stage Gauss method, $h = 10^{-2}$.

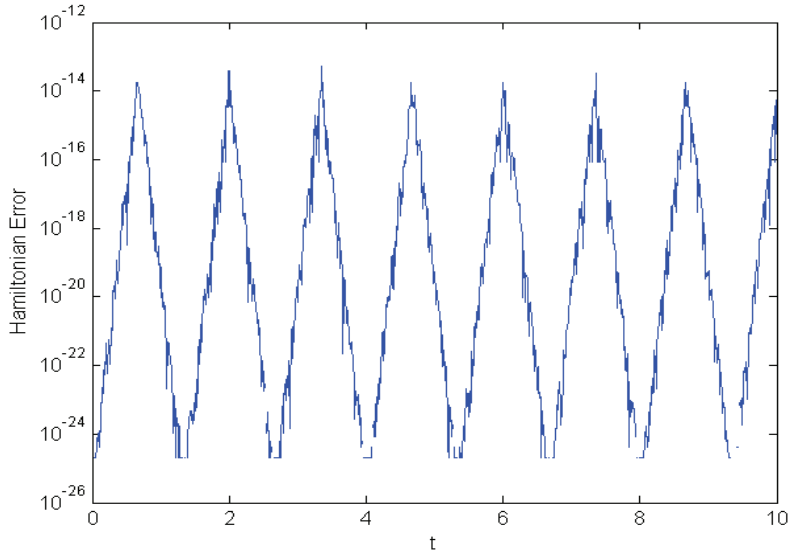

Figure 3.15: Hamiltonian error for problem (3.18)–(3.19) by using the HBVM(4,2) method, $h = 10^{-2}$.

3.5 Symmetry

We here prove that, provided that the abscissae $\{c_i\}$ are symmetrically distributed in the interval $[0,1]$, as is the case for the Gauss-Legendre nodes (see (2.3)), a HBVM(k,s) method is symmetric. In more details [32], if applied to the initial value problem

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

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

$$z' = -f(z), \qquad z(0) = y_1. \tag{3.20}$$

In order to prove this property, let us introduce the following matrices:

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

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

and, by recalling the vector $\mathcal{I}_s^1 \equiv \mathbf{e}_1^\top$ defined in (1.23) and the matrix $\mathcal{I}_s$ defined at (2.12),

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

Moreover, by setting

$$0 \equiv c_0 < c_1 < \cdots < c_k < c_{k+1} \equiv 1, \tag{3.21}$$

we need to define the matrix

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

If the abscissae are symmetrically distributed in the interval $[0,1]$ than, by taking into account (3.21), we have $c_i = 1 - c_{k-i+1}$, $i = 0,\dots,k+1$ and the following properties hold true:

(i) $J_r^\top = J_r^{-1} = J_r$;

(ii) $J_k\Omega J_k = \Omega \quad \Rightarrow \quad \Omega J_k = J_k\Omega$;

(iii) $J_{k+1}\Delta\mathcal{I}_s = \Delta\mathcal{I}_s D$;

(iv) $J_k\mathcal{P}_s = \mathcal{P}_s D$;

where the last two points follow from the properties (2.10) and (2.9) of Legendre polynomials, respectively. The discrete solution generated by a HBVM(k,s) method can then be cast in vector form as

$$\begin{pmatrix} -\hat{\mathbf{e}} & I_{k+1} \end{pmatrix} \otimes I\,\hat{Y} = h[\mathbf{0}\ \hat{\mathcal{I}}_s\mathcal{P}_s^\top\Omega\ \mathbf{0}] \otimes I\,f(\hat{Y}), \tag{3.22}$$

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

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

Left-multiplication of (3.22) by $L \otimes I$ then gives

$$\hat{A} \otimes I\,\hat{Y} = h\hat{B} \otimes I\,f(\hat{Y}), \tag{3.23}$$

with

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

and one easily realizes that

$$J_{k+1}\hat{A}J_{k+2} = -\hat{A}. \tag{3.24}$$

Moreover, exploiting the properties (i)–(iv) listed above, one also has:

$$\begin{aligned} J_{k+1}\hat{B}J_{k+2} &= \begin{pmatrix} \mathbf{0} & J_{k+1}\Delta\mathcal{I}_s\mathcal{P}_s^\top\Omega J_k & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \Delta\mathcal{I}_s D\mathcal{P}_s^\top J_k\Omega & \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{0} & \Delta\mathcal{I}_s D(J_k\mathcal{P}_s)^\top\Omega & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \Delta\mathcal{I}_s D^2\mathcal{P}_s^\top\Omega & \mathbf{0} \end{pmatrix} = \hat{B}. \end{aligned} \tag{3.25}$$

Now, by observing that

$$\hat{Z} \equiv J_{k+2} \otimes I\,\hat{Y} = \begin{pmatrix} y_1 \\ J_k \otimes I\,Y \\ y_0 \end{pmatrix}, \tag{3.26}$$

is the reversed-time discrete solution, left multiplication of (3.23) by $J_{k+1} \otimes I$ then gives:

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

where in the last equality (3.24), (3.25) and (3.26) have been used. What we have found is that the reversed-time vector satisfies the equation

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

which consists in applying the HBVM(k,s) method to problem (3.20), thus providing the approximation $z_1 = y_0$, by using the stages $Z = J_k \otimes IY$. In other words, we have proved that the HBVMs are symmetric methods.

3.6 Linear stability analysis

We end this chapter providing a linear stability analysis of HBVM(k,s): indeed, such methods can be defined independently from the problem of energy conservation, by considering a general function f in (3.7). Let us, then, apply our method to the celebrated test equation

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

Setting

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

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

$$\mathbf{x}' \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}' = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \equiv A\mathbf{x}, \qquad \mathbf{x}(0) = \mathbf{x}_0 \neq \mathbf{0}. \tag{3.27}$$

The application of a HBVM(k,s) method for solving (3.27) defines the polynomial u such that $u(0) = \mathbf{x}_0$ and, moreover,

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

where the last equality follows from the fact that the quadrature is exact for polynomials of degree $2s-1$.

Since independently of the value of $k \geq s$ we obtain the same polynomial u, and for $k = s$ this is the one provided by the s-stage Gauss-Legendre method, one has that all HBVM(k,s) methods, with $k \geq s$, have the same linear stability properties of the s-stage Gauss-Legendre method. I.e, their absolute stability region coincides with $\mathbb{C}^-$.

Chapter 4

Implementation of the methods

In this chapter we deal with the implementation of HBVM(k,s) methods. First, we will make clear that their computational cost depends essentially on s, as we have already mentioned in Remark 3.4.1, in the sense that for all $k \geq s$, the discrete problem turns out always to have block-dimension s. On the basis of this interesting property, we sketch two different efficient implementations of the methods: one based on a *blended implementation*, the remaining one based on a *triangular splitting* procedure. At the end of the chapter, we provide some numerical tests for comparing the different procedures.

4.1 Fundamental and silent stages

From (3.14)–(3.15), by considering the isospectrality property of a HBVM(k,s), with $k \geq s$, proved in Theorem 3.3.1, we have that such a method is defined by a Butcher matrix of rank s. Therefore, we can express $k-s$ of the stages of the method as a linear combination of the remaining s stages. We shall name *fundamental stages* the latter ones and *silent stages* the former ones and suppose, for simplicity, that the fundamental stages are the first s-ones.[1] We partition the stage vector Y as

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

with

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

the vectors with the fundamental and the silent stages, respectively. Similarly, we partition matrices $\mathcal{I}_s$ and $\mathcal{P}_s$ as

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

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

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

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

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

$$Y^{(1)} = \mathbf{e}^{(1)} \otimes y_0 + h\mathcal{I}_s^{(1)}\mathcal{P}_s^\top \Omega \otimes I \begin{pmatrix} f(Y^{(1)}) \\ f(Y^{(2)}) \end{pmatrix}, \tag{4.1}$$

$$Y^{(2)} = \mathbf{e}^{(2)} \otimes y_0 + h\mathcal{I}_s^{(2)}\mathcal{P}_s^\top \Omega \otimes I \begin{pmatrix} f(Y^{(1)}) \\ f(Y^{(2)}) \end{pmatrix}. \tag{4.2}$$

From (4.1), one then obtains that

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

which substituted into (4.2) gives:

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

Consequently, we can rewrite (4.1)-(4.2) as:

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

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

$$Y_{\ell+1}^{(1)} = \mathbf{e}^{(1)} \otimes y_0 + h\mathcal{I}_s^{(1)}\mathcal{P}_s^\top \Omega \otimes I \begin{pmatrix} f(Y_\ell^{(1)}) \\ f\left(\mathbf{a} \otimes y_0 + \mathcal{I}_s^{(2)}(\mathcal{I}_s^{(1)})^{-1} \otimes I\, Y_\ell^{(1)}\right) \end{pmatrix}, \ \ell = 0, 1, \ldots, \tag{4.3}$$

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

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

one then solves,

$$[I - hC \otimes J_0]\, \Delta_\ell = -F(Y_\ell^{(1)}), \qquad Y_{\ell+1}^{(1)} = Y_\ell^{(1)} + \Delta_\ell, \qquad \ell = 0, 1, \ldots, \tag{4.4}$$

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

$$C = \mathcal{I}_s^{(1)} \left[(\mathcal{P}_s^{(1)})^\top \Omega_1 + (\mathcal{P}_s^{(2)})^\top \Omega_2 \mathcal{I}_s^{(2)} (\mathcal{I}_s^{(1)})^{-1}\right]. \tag{4.5}$$

The following result holds true.

Theorem 4.1.1. *The eigenvalues of matrix C, as defined in (4.5), coincide with those of matrix X_s defined in (2.13), that is the eigenvalues of the Butcher matrix of the s-stage Gauss method.*

Proof. One has:

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

□

Consequently, matrix C has *always* the same spectrum, independently of the choice of the *fundamental* and *silent abscissae*.[2] This, in turn, coincides with the set of the nonzero eigenvalues of the corresponding Butcher array (see Theorem 3.3.1). Nevertheless, its condition number is greatly affected from this choice. Clearly, a badly conditioned matrix C would affect the convergence of both the iterations (4.3) and (4.4). As an example, in Figures 4.1 and 4.2 we plot the condition number of matrix C corresponding to the following choices of the fundamental abscissae, in the case $k \geq s = 3$:

- the first s abscissae of the k ones (Figure 4.1);
- s approximately evenly spaced abscissae among the k ones (Figure 4.2).

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

4.2 Alternative formulation of the discrete problem

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

$$\hat{\gamma}_j = \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) f(u(c_\ell h)), \qquad j = 0, \ldots, s-1, \tag{4.6}$$

of the polynomial approximation defining the given HBVM(k,s) method (see (3.8)). In more details, recalling (3.11), we have

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

which, substituted into (4.6), gives us the following formulation of the discrete problem:

$$\hat{\boldsymbol{\gamma}} \equiv \begin{pmatrix} \hat{\gamma}_0 \\ \vdots \\ \hat{\gamma}_{s-1} \end{pmatrix} = \mathcal{P}_s^\top \Omega \otimes I f(\mathbf{e} \otimes y_0 + h \mathcal{I}_s \otimes I \hat{\boldsymbol{\gamma}}), \tag{4.7}$$

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

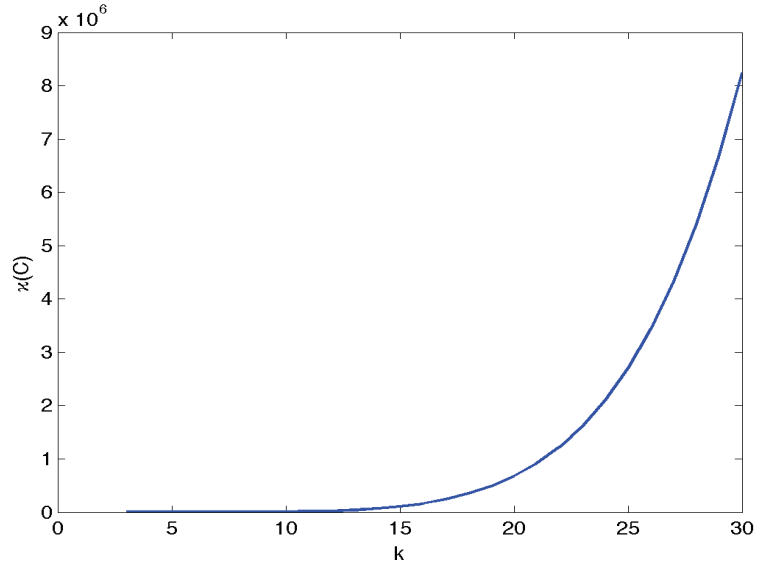

Figure 4.1: Condition number of matrix (4.5), fundamental abscissae fixed as the first s $(= 3)$ ones.

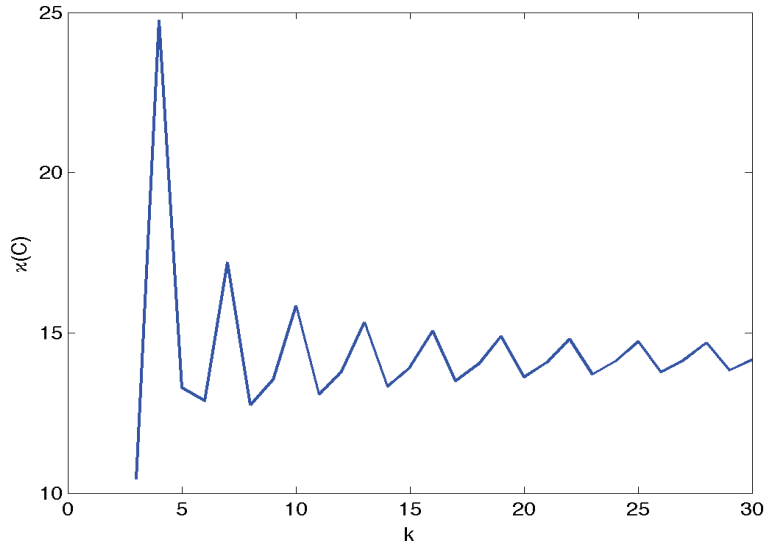

Figure 4.2: Condition number of matrix (4.5), with s $(= 3)$ fundamental abscissae approximately evenly spaced.

being $\mathbf{e} \in \mathbb{R}^k$ the unit vector and with the new approximation given by (see (3.9))

$$y_1 = y_0 + h\hat{\gamma}_0. \tag{4.8}$$

We observe that (4.7) has always (block) dimension s, whatever is the value of k considered. For solving such a problem, one can still use a *fixed-point iteration*,

$$\hat{\boldsymbol{\gamma}}^{\ell+1} = \mathcal{P}_s^\top \Omega \otimes I\, f\left(\mathbf{e} \otimes y_0 + h\mathcal{I}_s \otimes I\, \hat{\boldsymbol{\gamma}}^{\ell}\right), \qquad \ell = 0, 1, \ldots, \tag{4.9}$$

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

$$F(\hat{\boldsymbol{\gamma}}) = \hat{\boldsymbol{\gamma}} - \mathcal{P}_s^\top \Omega \otimes I\, f\left(\mathbf{e} \otimes y_0 + h\mathcal{I}_s \otimes I\, \hat{\boldsymbol{\gamma}}\right), \tag{4.10}$$

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

$$\left[I - hC \otimes J_0\right] \Delta^\ell = -F(\hat{\boldsymbol{\gamma}}^\ell), \qquad \hat{\boldsymbol{\gamma}}^{\ell+1} = \hat{\boldsymbol{\gamma}}^\ell + \Delta^\ell, \qquad \ell = 0, 1, \ldots, \tag{4.11}$$

where matrix C is now defined as follows:

$$C = \mathcal{P}_s^\top \Omega \mathcal{I}_s = \mathcal{P}_s^\top \Omega \mathcal{P}_{s+1} \hat{X}_s = (I_s\ \mathbf{0})\hat{X}_s = X_s. \tag{4.12}$$

Consequently, the iteration (4.11) becomes:

$$\left[I - hX_s \otimes J_0\right] \Delta^\ell = -F(\hat{\boldsymbol{\gamma}}^\ell), \qquad \hat{\boldsymbol{\gamma}}^{\ell+1} = \hat{\boldsymbol{\gamma}}^\ell + \Delta^\ell, \qquad \ell = 0, 1, \ldots. \tag{4.13}$$

Remark 4.2.1. *It is worth noticing that (4.13) holds independently of the choice of the k abscissae $\{c_i\}$, the only requirement being the order $2s$ of the quadrature, so that the property $\mathcal{P}_s^\top \Omega \mathcal{P}_{s+1} = (I_s\ \mathbf{0})$ holds true.*

Remark 4.2.2. *We observe that both matrices (4.5) and (4.12) share the same eigenvalues which, in turn, are the nonzero eigenvalues of the Butcher array of the given HBVM(k, s) method (see Theorem 3.3.1).*

Remark 4.2.3. *Even though, usually, by using the fixed point iteration (4.9) one is able to solve (4.7) quite inexpensively, when dealing with the solution of a stiff oscillatory problem, this procedure could require a stepsize h so small as to be not practical. In such a case, the simplified-Newton iteration (4.11) would be more appropriate and this is the procedure that we shall consider in the sequel.*

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

$$\left[I - hX_s \otimes J_0\right] \mathbf{x} = \boldsymbol{\eta}, \tag{4.14}$$

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

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

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

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

4.3 *Blended* HBVMs

The iterative procedure that we shall introduce in this section in order to solve (4.14), has been already successfully implemented in the computational codes `BiM` [28] and `BiMD` [31] for the numerical solution of stiff ODE-IVPs and linearly implicit DAEs up to order 3.

In order to provide a linear analysis of convergence [30, 52], we consider the classical test equation,

$$y' = \lambda y, \qquad \mathrm{Re}(\lambda) < 0. \tag{4.15}$$

In such a case, by setting as usual $q = h\lambda$, problem (4.14) becomes the linear system, of dimension s,

$$(I - qX_s)\mathbf{x} = \boldsymbol{\eta}. \tag{4.16}$$

By means of a left-multiplication by ζX_s^{-1}, where $\zeta > 0$ is a free parameter to be chosen later, we obtain the following equivalent formulation of (4.16):

$$\zeta(X_s^{-1} - qI)\mathbf{x} = \zeta X_s^{-1}\boldsymbol{\eta} \equiv \boldsymbol{\eta}_1. \tag{4.17}$$

Let us define the *weighting function*

$$\theta(q) = I(1 - \zeta q)^{-1}, \tag{4.18}$$

satisfying the following properties:

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

Then, we can derive a further equivalent formulation of problem (4.16), as the *blending*, with weights $\theta(q)$ and $I - \theta(q)$ of the two equivalent formulations (4.16) and (4.17), thus obtaining

$$M(q)\mathbf{x} = \boldsymbol{\eta}(q), \tag{4.19}$$

with:

$$\begin{aligned} M(q) &= \theta(q)(I - qX_s) + \zeta(I - \theta(q))(X_s^{-1} - qI), \\ \boldsymbol{\eta}(q) &= \theta(q)\boldsymbol{\eta} + \zeta(I - \theta(q))X_s^{-1}\boldsymbol{\eta}. \end{aligned} \tag{4.20}$$

Equations (4.19)-(4.20) define the *blended formulation* of the original problem (4.16). The next step is now to devise an iterative procedure, defined by a suitable splitting for solving (4.19)-(4.20). To this end we observe that, due to the properties of the weighting function $\theta(q)$ defined in (4.18), one has:

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

Consequently, $N(q) \equiv I(1 - \zeta q) \approx M(q)$, both for $q \approx 0$, and $|q| \gg 1$. It is then natural to define the following iterative procedure for solving (4.19):

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

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

$$\mathbf{x}_{r+1} = (I - \theta(q)M(q))\mathbf{x}_r + \theta(q)\boldsymbol{\eta}(q), \qquad r = 0, 1, \ldots. \tag{4.21}$$

Equation (4.21) defines the *blended iteration* associated with the blended formulation (4.19) of the problem.
By considering that the solution, say $\mathbf{x}^*$, of (4.19) satisfies also (4.21), by setting

$$\mathbf{e}_r = \mathbf{x}_r - \mathbf{x}^*, \tag{4.22}$$

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

$$\mathbf{e}_{r+1} = (I - \theta(q)M(q))\mathbf{e}_r \equiv Z(q)\mathbf{e}_r, \qquad r = 0, 1, \ldots, \tag{4.23}$$

with $Z(q)$ the corresponding *iteration matrix*. Consequently, the iteration (4.21) will converge to the solution $\mathbf{x}^*$ of the problem iff the spectral radius of $Z(q)$,

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

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

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

is the *region of convergence* of the iteration (4.21). The iteration will be said to be:

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

Remark 4.3.1. *A-convergent iterations are then appropriate when the underlying method is A-stable. Similarly, L-convergent iterations are appropriate in the case of L-stable methods.*

We observe that, for the matrix $Z(q)$ defined at (4.23),

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

Consequently, for the blended iteration (4.21) A-convergence and L-convergence are equivalent to each other. From the maximum-modulus theorem, in turn, it follows that this is equivalent to requiring that the *maximum amplification factor* of the iteration,

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

satisfies

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

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

$$\rho^* = \max_{x>0} \rho(ix) \leq 1. \tag{4.24}$$

We shall choose the free positive parameter ζ, in order to minimize ρ^*, so that (4.24) turns out to be fulfilled for all $s \geq 1$. The following result holds true.

Theorem 4.3.1. $\mu \in \sigma(X_s) \quad \Leftrightarrow \quad \dfrac{q(\mu-\zeta)^2}{\mu(1-q\zeta)^2} \in \sigma(Z(q))$.

Proof. From (4.23), (4.20), (4.18), and (2.13), one obtains:

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

from which our assertion easily follows. □

As a consequence, one obtains the following result.

Corollary 4.3.1. *The maximum amplification factor (4.24) of the blended iteration (4.21) is given by:*

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

Proof. One has:

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

The proof is completed by considering that

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

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

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

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

we can sort them by decreasing arguments:

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

due to the fact that

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

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

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

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

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

as is shown in Figures 4.3 and 4.4, in the cases $s=6$ and $s=7$, respectively. In such a case, the following result holds true.

Table 4.1: Blended iteration of HBVM(k,s) methods.

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

Theorem 4.3.2. *ρ^* is minimized by choosing*

$$\zeta = |\mu_1| \equiv \min_{\mu\in\sigma(X_s)} |\mu|, \tag{4.25}$$

resulting in

$$\rho^* = \frac{1}{2\zeta}\frac{|\mu_1-\zeta|^2}{|\mu_1|}\bigg|_{\zeta=|\mu_1|}. \tag{4.26}$$

In such a case, one obtains:

$$\rho^* = 1-\cos\phi_1 < 1. \tag{4.27}$$

Proof. For (4.25)-(4.26), see [27]. Concerning (4.27), one has:

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

□

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

We can also characterize the speed of convergence when $q\approx 0$, by considering that, from Theorem 4.3.1 and Theorem 4.3.2, it follows that

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

where the parameter

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

is called the *non-stiff* amplification factor. In Table 4.1 we list the relevant information for the blended iteration (4.21) of HBVM(k,s) methods.

4.4 Actual blended implementation

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

$$y' = f(y), \qquad y(0) = y_0 \in \mathbb{R}^m, \tag{4.28}$$

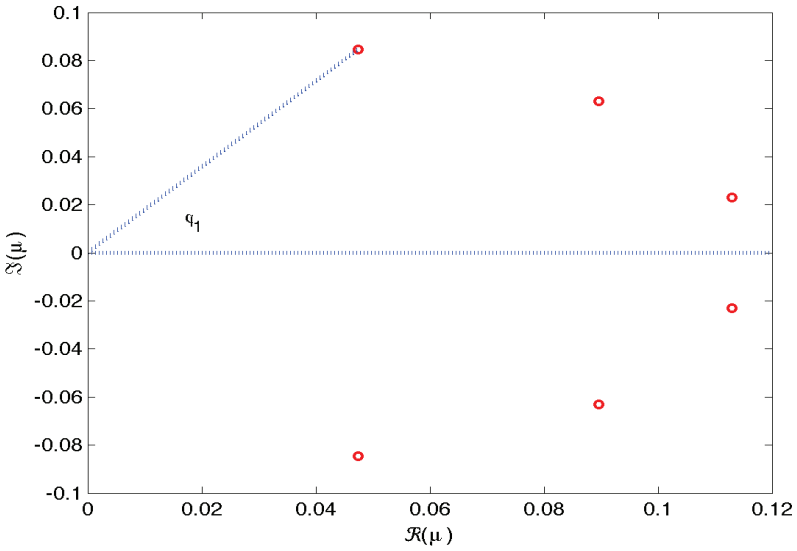

Figure 4.3: Eigenvalues of matrix X_6.

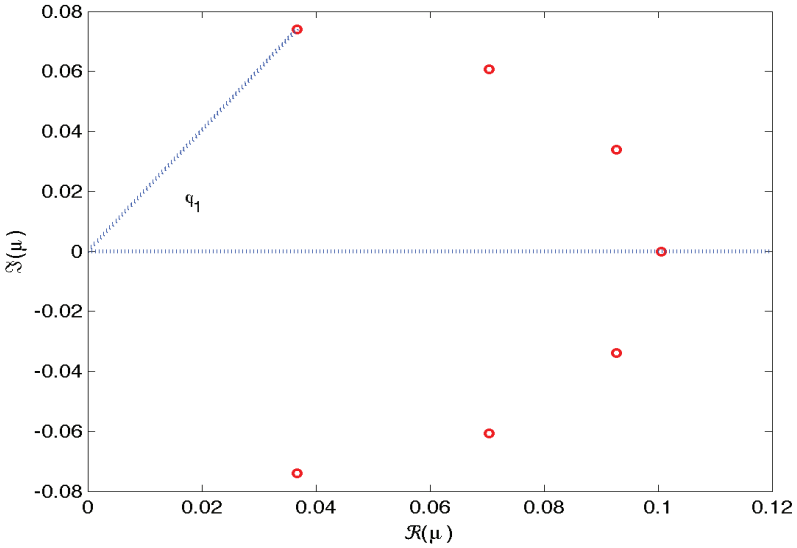

Figure 4.4: Eigenvalues of matrix X_7.

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

$$\theta = I_s \otimes \Gamma^{-1}, \qquad \text{with} \qquad \Gamma = I - h\zeta J_0 \in \mathbb{R}^{m\times m}, \tag{4.29}$$

where h is the stepsize, ζ is the optimal parameter specified in the second column in Table 4.1, and J_0 is the Jacobian of f evaluated at y_0 (clearly, we are speaking about the very first step in the numerical integration).

From (4.10) and (4.13), we have to solve the *outer-inner* iteration described in Table 4.2 (where $e \in \mathbb{R}^k$ denotes the unit vector).

Table 4.2: Outer-inner iteration for the blended implementation of HBVMs.

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

Let us analyze its computational complexity, by denoting, as 1 *flop*, an elementary (binary) algebraic *fl*oating-point *op*eration. One obtains:

- Γ: 1 Jacobian evaluation plus $2m$ *flops* (we multiply the function $f(y_0)$ by μ before computing its Jacobian);
- θ: $\frac{2}{3}m^3 - \frac{1}{2}m^2 - \frac{1}{6}m$ *flops* for computing the LU factorization of Γ;
- Y^ℓ: $km + 2ksm$ *flops*;
- f^ℓ: k function evaluations;
- η^ℓ: $sm + 2ksm$ *flops*;
- $z^{\ell,r}$: $2sm^2$ *flops*;
- $t^{\ell,r}$: sm *flops*;
- $u^{\ell,r}$: $2s^2m + 2sm$ *flops*;
- $w^{\ell,r}$: $2s^2m + sm$ *flops*;
- $\Delta^{\ell,r+1}$: $4sm^2 + 3sm$ *flops*;
- $\hat{\gamma}^{\ell+1}$: sm *flops*.

Consequently, this algorithm has a fixed computational cost of 1 Jacobian evaluation and $\frac{2}{3}m^3 - \frac{1}{2}m^2 + \frac{11}{6}m$ *flops*, plus, assuming that ν *inner* iterations are performed, a cost of k function evaluations and $4ksm + km + 2sm + \nu(6sm^2 + 4s^2m + 7sm)$ *flops* per *outer* iteration.

Table 4.3: Nonlinear iteration for the blended implementation of HBVMs.

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

A simplified (and sometimes more efficient) procedure is that of performing a *nonlinear* iteration obtained by executing exactly 1 inner iteration (i.e., setting $r = 0$ in the inner cycle in Table 4.2) in the above procedure, thus obtaining the algorithm depicted in Table 4.3. In such a case, the resulting computational cost is obtained as follows:

- Γ: 1 Jacobian evaluation plus $2m$ *flops* (we multiply the function $f(y_0)$ by $h\zeta$ before computing its Jacobian);
- θ: $\frac{2}{3}m^3 - \frac{1}{2}m^2 - \frac{1}{6}m$ *flops* for computing the LU factorization of Γ;
- Y^ℓ: $km + 2ksm$ *flops*;
- f^ℓ: k function evaluations;
- η^ℓ: $sm + 2ksm$ *flops*;
- u^ℓ: $2s^2m$ *flops*;
- Δ^ℓ: $4sm^2 + 2sm$ *flops*;
- $\hat{\gamma}^{\ell+1}$: sm *flops*.

Consequently, this latter algorithm has a fixed computational cost of 1 Jacobian evaluation and $\frac{2}{3}m^3 - \frac{1}{2}m^2 + \frac{11}{6}m$ *flops*, plus a cost of k function evaluations and $4sm^2 + 4ksm + 2s^2m + km + 4sm$ *flops* per iteration.

4.5 The triangular splitting procedure

For the efficient implementation of the simplified-Newton method, additional iterative procedures have been also devised: some of them are based on suitable triangular splittings [1, 17, 52, 53]. However, the iteration defined in [52, 53], as well as its modified version defined in [1], result to be not effective for (4.13), due to the particular structure of the matrix X_s (see (2.13)).

The *blended implementation* of HBVM(k, s) methods, as already shown in Section 4.4, turns out more appropriate for solving (4.13), but we now shall describe a new procedure, introduced in [9], based on a particular triangular splitting, which appears to be even more favourable. The basic idea is similar to that introduced in [17] for the efficient implementation of RadauIIA collocation methods, but the framework, the general details and results are completely different.

With reference to (4.13), we then start by introducing a set of s *auxiliary abscissae* (whose actual choice will be explained in the sequel),

$$\tilde{c}_1 < \cdots < \tilde{c}_s, \tag{4.30}$$

the polynomial (see (4.6))

$$\tilde{\gamma}(c) = \sum_{j=0}^{s-1} P_j(c)\hat{\gamma}_j, \qquad c \in \mathbb{R}, \tag{4.31}$$

and a new set of (block) unknowns,

$$\tilde{\gamma}_i \equiv \sum_{j=0}^{s-1} P_j(\tilde{c}_i)\hat{\gamma}_j, \qquad i = 1, \ldots, s, \tag{4.32}$$

that are the evaluations of (4.31) at the auxiliary abscissae (4.30). Introducing the (block) vector

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

and the matrix

$$\tilde{\mathcal{P}} = (\, P_{j-1}(\tilde{c}_i)\,) \in \mathbb{R}^{s\times s}, \tag{4.33}$$

we can recast (4.32) in vector form as

$$\tilde{\boldsymbol{\gamma}} = \tilde{\mathcal{P}} \otimes I\hat{\boldsymbol{\gamma}}. \tag{4.34}$$

Left-multiplication of (4.13) by $\tilde{\mathcal{P}} \otimes I$, allows to recast the problem in terms of $\tilde{\boldsymbol{\gamma}}$ as:

$$\tilde{M}_0\tilde{\Delta}^\ell \equiv [I - h\tilde{A} \otimes J_0]\tilde{\Delta}^\ell = \boldsymbol{\eta}^\ell, \qquad \tilde{\boldsymbol{\gamma}}^{\ell+1} = \tilde{\boldsymbol{\gamma}}^\ell + \tilde{\Delta}^\ell, \qquad \ell = 0, 1, \ldots, \tag{4.35}$$

where

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

Remark 4.5.1. *We stress that the matrix $\tilde{A}$ is independent of k, but it only depends on s whatever is $k \geq s$. Consequently the following approach applies also in the particular case of $k = s$, that is, to the s-stage Gauss method.*

With these premises, the choice of the auxiliary abscissae (4.30) will be done in such a way that the matrix $\tilde{A}$ in (4.35) can be factored as

$$\tilde{A} = \tilde{L}\tilde{U}, \tag{4.36}$$

with $\tilde{U}$ upper triangular with unit diagonal entries, and $\tilde{L}$ lower triangular with *constant* diagonal entries. In such a case, by following the approach of van der Houwen et al. [52, 53], we replace the iteration (4.35) with the *inner-outer iteration*

$$\begin{aligned} \left[I - h\tilde{L} \otimes J_0\right] \tilde{\Delta}^{\ell,r+1} &= h\tilde{L}(\tilde{U} - I) \otimes J_0\tilde{\Delta}^{\ell,r} + \boldsymbol{\eta}^\ell, \qquad r = 0, 1, \ldots, \mu - 1, \\ \tilde{\boldsymbol{\gamma}}^{\ell+1} &= \tilde{\boldsymbol{\gamma}}^\ell + \tilde{\Delta}^{\ell,\mu}, \qquad \ell = 0, 1, \ldots. \end{aligned} \tag{4.37}$$

In particular, since $\tilde{\Delta}^{\ell,0} = 0$, the choice $\mu = 1$ corresponds to the approach used by van der Houwen et al. to devise PTIRK methods [52], whereas, by choosing μ large enough to obtain full convergence of the inner-iteration (the one on r), one has that the outer iteration is equivalent to (4.35). Clearly, all the intermediate possibilities can be suitably considered. After the convergence of (4.37), the new approximation is computed as in (4.8), where $\hat{\gamma}_0$ is retrieved from (4.34).

We observe that, since the diagonal entries of the factor $\tilde{L}$ are all equal to a given value, say d_s, then, for performing the inner-outer iteration (4.37) one only needs to factorize the matrix

$$I - hd_sJ_0 \in \mathbb{R}^{m\times m}, \tag{4.38}$$

having the same size as that of the continuous problem (4.28).

Remark 4.5.2. *Actually, in a computational code this matrix can be kept constant until one needs to compute again the Jacobian matrix J_0 and/or to choose a different stepsize h. Here, we deliberately do not take into account this issue, which requires a further analysis (see, e.g., [29] for the code described in [28]). Consequently in the numerical tests we shall use a constant stepsize and evaluate the Jacobian matrix at each integration step.*

Concerning d_s the following result holds true.

Theorem 4.5.1. *Assume that the factorization (4.36) is defined and that the diagonal entries of the factor $\tilde{L}$ are all equal to d_s. Then, with reference to (2.8), one has:*

$$d_s = \begin{cases} \sqrt[s]{\prod_{i=1}^{\frac{s}{2}} \xi_{2i-1}^2}, & \text{if } s \text{ is even,} \\ \sqrt[s]{\frac{1}{2}\prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i}^2}, & \text{if } s \text{ is odd.} \end{cases} \tag{4.39}$$

Proof. Since, by hypothesis (4.36) holds true, we have

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

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

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

and (4.39) follows from Lemma 2.2.1. □

By virtue of the previous result, we have computed the auxiliary abscissae (4.30) by symbolically solving the following set of equations, which is equivalent to requiring that $\tilde{L}$ has constant diagonal entries:

$$\det(\tilde{A}_{\ell+1}) = d_s \det(\tilde{A}_\ell), \qquad \ell = 1, \ldots, s-1, \tag{4.40}$$

where $\tilde{A}_\ell$ denotes the principal leading submatrix of order ℓ of $\tilde{A}$ and d_s is given by (4.39).

We observe that the auxiliary abscissae (4.30) are s whereas the algebraic conditions (4.40) are $s-1$. This means that we can express $s-1$ abscissae as a function of the remaining *free abscissa*. We shall choose such free abscissa in order to optimize the convergence properties of the iteration. To this end, according to a linear analysis of convergence similar to that done in Section 4.3, we apply the splitting procedure (4.37) to the test equation (4.15). Since the problem is linear, the iteration (4.37) consists in solving only the inner iteration (the one with index r), so that we can skip the index ℓ of the outer iteration. By setting, as is usual, $q = h\lambda$ one obtains that the error equation associated with (4.37) is given by

$$\mathbf{e}_{r+1} = q(I - q\tilde{L})^{-1}\tilde{L}(\tilde{U} - I)\mathbf{e}_r \equiv Z(q)\mathbf{e}_r, \qquad r = 0, 1, \ldots, \mu - 1, \tag{4.41}$$

where $\mathbf{e}_r$ is the error vector at step r (see (4.22)), and $Z(q)$ is the iteration matrix induced by the splitting procedure. This latter will converge if and only if the spectral radius of $Z(q)$, $\rho(q)$, is less than 1. According to the definitions given in Section 4.3, in our case, since

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

which is a nilpotent matrix of index s, the iteration is L-convergent if and only if it is A-convergent. Since the iteration is well defined for all $q \in \mathbb{C}^-$ (due to the fact that the diagonal entries of $\tilde{L}$ are all equal to d_s, which is a positive number as shown in (4.39)) and $\rho(0) = 0$, from the maximum-modulus theorem it follows immediately that A-convergence is, in turn, equivalent to require that the *maximum amplification factor* of the iteration,

$$\rho^* = \max_{x \in \mathbb{R}} \rho(ix), \tag{4.42}$$

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

$$\tilde{\rho} = \rho(\tilde{L}(\tilde{U} - I)), \tag{4.43}$$

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

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

Consequently, the smaller ρ^* and $\tilde{\rho}$, the better the convergence properties of the iteration. For this reason, we choose the free auxiliary abscissa in order to (approximately) minimize the maximum amplification factor ρ^* of the iteration (see (4.42)), while fulfilling the conditions (4.40).

In Table 4.4 we list the obtained values for the auxiliary abscissae (4.30) and the diagonal entry d_s of the corresponding factor $\tilde{L}$ (see (4.39)), for a generic HBVM(k,s) with $k \geq s$ and $s = 2, \ldots, 6$. One can see that in all cases the abscissae are distinct and inside the interval $[0,1]$.

We emphasize that, for a given s, the distribution of the auxiliary abscissae $\{\tilde{c}_i\}$ and the factorization (4.36) of the matrix $\tilde{A}$ are both independent of k. Consequently, when one is going to implement this class of methods, it is possible to conjecture a procedure to advance the time that dynamically selects the most appropriate value of k. In so doing, depending on the specific problem at hand and the configuration of the system at the given time, one could easily switch, having fixed s, from a symplectic method (choosing $k = s$ (Gauss method)) to an energy preserving one (choosing $k > s$).

For sake of comparison, in Table 4.5 we list the maximum amplification factors and the nonstiff amplification factors for the following L-convergent iterations applied to the s-stage Gauss-Legendre methods:

(i) the iteration obtained by the original triangular splitting in [52];

(ii) the iteration obtained by the modified triangular splitting in [1];

(iii) the nonlinear iteration obtained by the *blended implementation* of the methods, as defined in Table 4.3;

(iv) the iteration defined by (4.37).

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

Remark 4.5.3. *For sake of accuracy, we stress that, when dealing with the actual implementation of HBVM(k,s) methods, only the blended iteration and the one described in (4.37) can be considered, whereas the triangular splitting defined in [52] and its modified version [1] turn out to be not effective, as was pointed out at the beginning of this section. Consequently, in such a case, one has to consider only the last two groups of columns in Table 4.5.*

4.5.1 Averaged amplification factors

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

$$\rho_\mu^* = \sup_{x\in\mathbb{R}} \sqrt[\mu]{\|Z(ix)^\mu\|}, \quad \tilde{\rho}_\mu = \sqrt[\mu]{\left\|\left[\tilde{L}(\tilde{U}-I)\right]^\mu\right\|}, \quad \rho_\mu^\infty = \sqrt[\mu]{\|(\tilde{U}-I)^\mu\|}, \tag{4.44}$$

Table 4.4: Auxiliary abscissae (4.30) for the HBVM(k,s) and s-stage Gauss method, $s = 2,\dots,6$, and the diagonal entry d_s (see (4.39)) of the corresponding factor $\tilde{L}$.

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

Table 4.5: Amplification factors for the triangular splitting in [52], the modified triangular splitting in [1], the nonlinear iteration in Table 4.3 and the splitting (4.37), for the s-stage Gauss-Legendre formulae. The last two cases coincide with those for the HBVM(k,s) methods, $k \geq s$.

	(i): triangular splitting in [52]		(ii): triangular splitting in [1]		(iii): *blended* iteration in Table 4.3		(iv): triangular splitting (4.37)	
s	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$
2	0.1429	0.0833	0.1340	0.0774	0.1340	0.0774	0.1340	0.0774
3	0.3032	0.1098	0.2537	0.0856	0.2765	0.1088	0.2536	0.0870
4	0.4351	0.1126	0.3492	0.0803	0.3793	0.1119	0.3291	0.0859
5	0.5457	0.1058	0.4223	0.0730	0.4544	0.1066	0.3709	0.0654
6	0.6432	0.0973	0.4861	0.0702	0.5114	0.0993	0.4353	0.0650

Table 4.6: Averaged amplification factors (4.44) for the splitting (4.37), used for the HBVM(k,s) methods, $k \geq s$, when performing $\mu = 1, 2, 3$ iterations.

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

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

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

and

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

In Table 4.6 we list the averaged amplification factors when performing $\mu = 1, 2, 3$ iterations, and considering the infinity norm. As one may see, the resulting iteration turns out to be A-convergent also when using just one inner iteration, unless the case $s = 6$, which requires at least 3 inner iterations.

Remark 4.5.4. *When performing only μ inner-iterations for solving the discrete problem generated by (4.15), we have to consider also the* outer *iteration (i.e., the one on ℓ in (4.37)), even though the problem is linear. In such a case, by setting E_ℓ the error at the ℓ-th outer iteration, it is quite straightforward to see that the error equation is now given by:*

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

Consequently, the convergence analysis made for (4.41) also applies to the present case.

4.6 Computational cost of the triangular splitting implementation

We now analyze the computational complexity of the triangular splitting procedure described in Section 4.5 when the method is applied for approximating the initial value problem (4.28), having

dimension m, by using the stepsize h. In order to reduce the computational cost of the procedure, we first multiply both sides of (4.37) by

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

as done in [17]. Considering that

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

with S strictly lower triangular, system (4.37) then takes the form

$$\left[\frac{1}{hd_s} I - I \otimes J_0\right] \tilde{\Delta}^{\ell,r+1} = \frac{1}{h}(S \otimes I)\tilde{\Delta}^{\ell,r+1} + (C \otimes J_0)\tilde{\Delta}^{\ell,r} + R^{\ell}, \quad r = 0, 1, \ldots, \mu - 1, \tag{4.45}$$

where

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

As a consequence we have now to factor only the matrix

$$\frac{1}{hd_s} I - J_0 \in \mathbb{R}^{m \times m}, \tag{4.46}$$

in place of (4.38). We now show that in the computation of

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

at the right-hand side of (4.45), one can completely eliminate any $O(m^2)$ complexity term (that would be the leading one since, usually, $m \gg s$). This is true at the very first step, since by definition,

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

By setting

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

we have that $\boldsymbol{w}_0 = R^{\ell}$ and, after solving the first step of (4.45), which reads

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

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

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

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

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

and thus $\boldsymbol{w}_1 = (C \otimes J_0)\tilde{\Delta}^{\ell,1} + R^{\ell}$ can be computed with $O(s^2 m)$ *flops*. The same procedure can then be repeated in the subsequent steps, as shown in Table 4.7 (where, as above, e denotes the unit vector in $\mathbb{R}^k$), thus avoiding the $O(sm^2)$ complexity term.

Let us now analyze the computational cost of each step of the procedure in Table 4.7, in terms of *flops*:

- θ: 1 Jacobian evaluation plus $\frac{2}{3}m^3 - \frac{1}{2}m^2 + \frac{11}{6}m$ *flops* ($2m$ operations plus those required to compute a LU factorization);

Table 4.7: Outer-inner iteration for the triangular splitting implementation of HBVMs.

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

$\mu_s = hd_s$

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

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

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

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

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

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

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

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

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

for $r = 0, 1, \ldots$

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

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

if $i > 1$

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

$v_i^{\ell,r+1} = v_i^{\ell,r+1} + H_{i,j}\Delta_j^{\ell,r+1}$

end

end

$\Delta_i^{\ell,r+1} = v_i^{\ell,r+1}\theta^\top$

end

$\tilde{\Delta}^{\ell,r+1} = (\Delta_1^{\ell,r+1}, \ldots, \Delta_s^{\ell,r+1})^\top$

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

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

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

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

end

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

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

4.7 The triangular splitting procedure for separable Hamiltonian problems.

We now describe, according to [10], how the triangular splitting procedure can be further improved, when the method is applied to a separable Hamiltonian problem with Hamiltonian:

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

Consequently, the problem assumes the simplified form

$$q' = p, \qquad p' = -\nabla U(q), \qquad q(0) = q_0, \quad p(0) = p_0 \in \mathbb{R}^m, \tag{4.47}$$

which we plan to solve on the interval $[0, h]$. The HBVM method (3.14) provides the approximations (see (3.12))

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

with stage vectors

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

given by (see (3.11)):

$$Q = \mathbf{e} \otimes q_0 + h\mathcal{I}_s\mathcal{P}_s^\top\Omega \otimes I\, P, \qquad P = \mathbf{e} \otimes p_0 - h\mathcal{I}_s\mathcal{P}_s^\top\Omega \otimes I\, \nabla U(Q), \tag{4.48}$$

where $\nabla U(Q) = (\nabla U(Q_1)^\top, \dots, \nabla U(Q_s)^\top)^\top$ and, as in the previous sections, $\mathbf{e} \in \mathbb{R}^k$ is the unit vector. Substituting the second equation of (4.48) into the first one, also considering that $\mathcal{I}_s\mathcal{P}_s^\top\Omega\mathbf{e} = \mathbf{c}$, and taking into account (2.13) and (4.12), one has

$$Q = \mathbf{e} \otimes q_0 + h\mathbf{c} \otimes p_0 - h^2\mathcal{P}_{s+1}\hat{X}_s X_s\mathcal{P}_s^\top\Omega \otimes I\, \nabla U(Q). \tag{4.49}$$

This problem has (block) dimension k. In order to recover a problem of (block) dimension s, independently of k, similarly to what has been done in Section 4.2, we consider as unknown the (block) vector with the coefficients of the underlying polynomial of degree s:

$$\hat{\gamma} = \mathcal{P}_s^\top\Omega \otimes I\, \nabla U(Q). \tag{4.50}$$

Substituting (4.49) into (4.50), we then obtain the following discrete problem:

$$F(\hat{\boldsymbol{\gamma}}) \equiv \hat{\boldsymbol{\gamma}} - \mathcal{P}_s^\top \Omega \otimes I \, \nabla U \left(\mathbf{e} \otimes q_0 + h\mathbf{c} \otimes p_0 - h^2 \mathcal{P}_{s+1} \hat{X}_s X_s \otimes I \, \hat{\boldsymbol{\gamma}} \right) = \mathbf{0}. \tag{4.51}$$

By taking into account that

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

the application of the simplified-Newton method for solving (4.51) results in the following iteration:

$$\left[I + h^2 X_s^2 \otimes \nabla^2 U(q_0) \right] \Delta^\ell = -F(\hat{\boldsymbol{\gamma}}^\ell), \qquad \hat{\boldsymbol{\gamma}}^{\ell+1} = \hat{\boldsymbol{\gamma}}^\ell + \Delta^\ell, \qquad \ell = 0, 1, \ldots. \tag{4.52}$$

This is the problem which we now attack by means of a triangular splitting procedure. As done in Section 4.5, we introduce a set of auxiliary abscissae

$$\tilde{c}_1 < \cdots < \tilde{c}_s, \tag{4.53}$$

with the corresponding matrix $\tilde{\mathcal{P}}$ defined as in (4.33) and the unknown vector $\tilde{\boldsymbol{\gamma}}$ in form (4.34). Similarly as previously done in (4.35), left-multiplication of (4.52) by $\tilde{\mathcal{P}} \otimes I$ allows to recast the problem in terms of $\tilde{\boldsymbol{\gamma}}$, thus obtaining the following equivalent linear system,

$$\left[I + h^2 \tilde{A} \otimes \nabla^2 U(q_0) \right] \tilde{\Delta}^\ell = \boldsymbol{\eta}^\ell, \tag{4.54}$$

where

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

According to what has been done in Section 4.5, we choose the auxiliary abscissae (4.53) in such a way that $\tilde{A}$ admits the factorization (4.36) with $\tilde{U}$ upper triangular with unit diagonal entries, and $\tilde{L}$ lower triangular with diagonal entries all equal to

$$d_s = \sqrt[s]{\det X_s^2} \tag{4.55}$$

(this can be proved in a similar way as done in Theorem 4.5.1). As observed in Section 4.5, this allows one to express $s-1$ abscissae as a function of a remaining *free abscissa*. The free abscissa will then be chosen in order to (approximately) optimize the convergence properties of the following inner iteration, coupled with the outer iteration (4.54):

$$\left[I + h^2 \tilde{L} \otimes \nabla^2 U(q_0) \right] \tilde{\Delta}^{\ell, r+1} = h^2 \left[\tilde{L} - \tilde{A} \right] \otimes \nabla^2 U(q_0) \tilde{\Delta}^{\ell, r} + \boldsymbol{\eta}^\ell, \qquad r = 0, 1, \ldots. \tag{4.56}$$

Similarly as in (4.37), we have now that the coefficient matrix is lower block triangular, with diagonal block entries all equal to

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

which is a symmetric matrix having the same size as that of the continuous problem (4.47), independently of s. According to [30], a linear convergence analysis of the iteration (4.56) is obtained by considering the scalar problem

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

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

$$Z(x^2) = x^2 (I + x^2 \tilde{L})^{-1} \tilde{L} (I - \tilde{U}). \tag{4.57}$$

Let $\rho(x^2)$ denote the spectral radius of the iteration matrix (4.57). We observe that

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

Therefore, according to [30], iteration (4.56) is L-convergent if the maximum amplification factor of the iteration, defined as

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

is not larger than 1.

Moreover, one has

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

with the non-stiff amplification factor $\tilde{\rho}$ formally still given by (4.43). Clearly, the smaller the parameters ρ^* and $\tilde{\rho}$, the better the convergence properties of the iteration.

In addition to this, by repeating similar arguments as those reported in Section 4.5.1, we also introduce the averaged amplification factors for the iteration (4.56), measuring the "average" convergence when exactly μ iterations are performed. They are defined as (see (4.57))

$$\rho_\mu^* = \sup_{x \in \mathbb{R}} \sqrt[\mu]{\|Z(x^2)^\mu\|}, \quad \tilde{\rho}_\mu = \sqrt[\mu]{\left\| \left[\tilde{L}(\tilde{U} - I) \right]^\mu \right\|}, \quad \rho_\mu^\infty = \sqrt[\mu]{\|(\tilde{U} - I)^\mu\|}, \tag{4.58}$$

where $\|\cdot\|$ is a suitable matrix norm (compare with (4.44)). These parameters are very useful in the actual implementation of the methods, where generally only a small number of iterations is performed. For this reason we choose the free abscissa in order to (approximately) minimize the values of $\rho_\mu^*, \mu = 1, 2, 3, 4$: in particular, it is optimized the first parameter which turns out to be less than 1. This is different from what done in [10], where the free abscissa has been chosen in order to (approximately) minimize the maximum amplification factor ρ^*.

In Table 4.8 we list the computed optimal auxiliary nodes for $s = 2, \ldots, 6$, along with the corresponding diagonal entry d_s, with 36 significant digits: one may see that the auxiliary nodes are all distinct and inside the interval $[0, 1]$.

Table 4.9 shows the convergence factors for the iteration (4.56), where the infinite norm has been used for the computation of (4.58). As one can see, in order to obtain a convergent iteration, one inner iteration is sufficient for the case $s = 2$ and $s = 3$, while two inner iterations are needed in the cases $s = 4$ and $s = 5$. Finally, at least four inner iterations are needed to obtain convergence, in the case $s = 6$.

For sake of completeness we also mention that actually, similarly as done at the beginning of Section 4.6, in order to reduce the computational cost of the procedure, one can solve the following system obtained by multiplying both sides of (4.56) by $h^{-2}\tilde{L}^{-1} \otimes I$:

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

with

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

strictly lower triangular and

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

The remaining details are then similar to those explained in Section 4.6.

Table 4.8: Auxiliary abscissae (4.30) for the HBVM(k,s) and s-stage Gauss method, $s=2,\dots,6$ for separable Hamiltonian problems, and the diagonal entry d_s (see (4.55)) of the corresponding factor $\tilde{L}$.

$s=2$	
$\tilde{c}_1$	0.3
$\tilde{c}_2$	1
d_2	1/12
$s=3$	
$\tilde{c}_1$	0.18838718112360613351895144351002434 2
$\tilde{c}_2$	0.42541922141818347835430054689468788 8
$\tilde{c}_3$	0.87
d_3	0.04110353457217450169152685538590981 74
$s=4$	
$\tilde{c}_1$	0.13839179546033992293368756080079890 5
$\tilde{c}_2$	0.29921388106651576439415717217989267 3
$\tilde{c}_3$	0.53860119088715235705995710475964603 6
$\tilde{c}_4$	0.895
d_4	0.02439750182371332948385961590600250 47
$s=5$	
$\tilde{c}_1$	0.26469193829071739344114929036861174 0
$\tilde{c}_2$	0.34712660870759669498183464008420098 8
$\tilde{c}_3$	0.05364559835125359823531505991964866 1
$\tilde{c}_4$	0.49913966664119541624914013850859470 2
$\tilde{c}_5$	0.771
d_5	0.01613493741827826427253049380882892 56
$s=6$	
$\tilde{c}_1$	0.22598589148959878075904037670795849 6
$\tilde{c}_2$	0.36643189170258729608056886185439036 4
$\tilde{c}_3$	0.43980743420584080268412154191319197 1
$\tilde{c}_4$	0.04059509783777282807206774082004015 12
$\tilde{c}_5$	0.61582504525880070596908268045894827
$\tilde{c}_6$	0.8865
d_6	0.01145509013432089422202647128222134 70

Table 4.9: Amplification factors for the splitting (4.56), for the HBVM(k,s) methods, $k \geq s$, applied to a separable Hamiltonian problem.

s	ρ_1^*	ρ_2^*	ρ_3^*	ρ_4^*	$\tilde{\rho}_1$	$\tilde{\rho}_2$	$\tilde{\rho}_3$	$\tilde{\rho}_4$	ρ_1^∞	ρ_2^∞	ρ_3^∞	ρ_4^∞	ρ^*	$\tilde{\rho}$
2	0.25	0.25	0.25	0.25	0.0833	0.0833	0.0833	0.0833	0.2	0	0	0	0.25	0.0833
3	0.630	0.482	0.447	0.433	0.173	0.113	0.0954	0.0873	0.630	0.170	0	0	0.433	0.0668
4	1.065	0.618	0.602	0.588	0.258	0.130	0.0903	0.0718	1.065	0.452	0.220	0	0.556	0.0328
5	2.310	0.993	0.777	0.714	0.423	0.130	0.0789	0.0588	2.310	0.629	0.372	0.0998	0.582	0.0219
6	5.106	1.579	1.0022	0.830	0.304	0.0797	0.0604	0.0478	3.139	1.328	0.515	0.246	0.543	0.0178

4.8 Numerical Tests

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

(i) the fixed-point iteration;

(ii) the blended iteration described in Table 4.3;

(iii) the triangular splitting iteration described in Table 4.7, by using 2 inner iterations.

We choose 2 inner iterations for the triangular splitting iteration (iii), so that the cost of one outer iteration is comparable to that of one blended iteration (ii). We stress that, for all the three above iterations, the total number of functional evaluations equals the number of iterations times k. Moreover, for the last two iterations, at each step one also needs to evaluate the Jacobian J_0, as well as to factor a matrix having the same size as that of the continuous problem (i.e., (4.29) for (ii) and (4.46) for (iii)).

The first problem is a nonlinear Hamiltonian problem describing the motion of a charged particle, with charge e and mass m, in a magnetic field with Biot-Savart potential [19]. It is defined by the Hamiltonian:

$$H(x,y,z,x',y',z') = \frac{1}{2m}\left[\left(x' - \alpha\frac{x}{\rho^2}\right)^2 + \left(y' - \alpha\frac{y}{\rho^2}\right)^2 + (z' + \alpha\log\rho)^2\right], \tag{4.59}$$

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

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

and the initial values

$$x = 0.5, \qquad y = 10, \qquad x' = -0.1, \qquad y' = -0.3, \qquad z = z' = 0. \tag{4.60}$$

In Table 4.10 we list the results obtained by applying the HBVM$(k,2)$ methods, with $k = 2,4,6,8,10$, for solving this problem over the interval $[0,10^3]$ with stepsize $h = 0.1$. From the results in the table, one infers that:

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

Table 4.10: Results when solving problem (4.59)-(4.60) by using the HBVM$(k,2)$ method with stepsize $h = 0.1$ over the interval $[0, 10^3]$.

k	Hamiltonian error	solution error	fixed-point iterations	blended iterations	splitting iterations
2	$1.6 \cdot 10^{-3}$	$7.18 \cdot 10^{-4}$	79511	66854	48030
4	$8.3 \cdot 10^{-6}$	$2.42 \cdot 10^{-5}$	79846	66884	48252
6	$5.9 \cdot 10^{-9}$	$1.02 \cdot 10^{-5}$	79911	66941	48349
8	$1.7 \cdot 10^{-12}$	$1.02 \cdot 10^{-5}$	79939	66963	48377
10	$4.4 \cdot 10^{-16}$	$1.02 \cdot 10^{-5}$	79962	66976	48402

As a second test problem we consider, on the contrary, a *stiff oscillatory* problem. It is defined as a slight modification of the Fermi-Pasta-Ulam problem described in [49].[3] The Hamiltonian defining this problem is given by:

$$H(q,p) = \frac{1}{2}\sum_{i=1}^{m}(p_{2i-1}^2 + p_{2i}^2) + \frac{1}{4}\sum_{i=1}^{m}\omega_i^2(q_{2i} - q_{2i-1})^2 + \sum_{i=0}^{m}(q_{2i+1} - q_{2i})^4, \tag{4.61}$$

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

$$\omega_i = \omega_{m-i+1} = 10, \qquad i = 1, 2, 3, \qquad \text{and} \qquad \omega_4 = 10^4. \tag{4.62}$$

The starting vector is

$$p_i = 0, \qquad q_i = \frac{i-1}{4m-2}, \qquad i = 1, \dots, 2m. \tag{4.63}$$

Since the Hamiltonian function (4.61) is a polynomial of degree 4, the HBVM$(2s,s)$ method (having order $2s$), is able to exactly preserve the Hamiltonian, for all $s \geq 1$. As an example, we fix $s = 3$ and integrate the problem over the interval $[0, 10]$. In this case, the fixed-point iteration cannot be expected to work, when using stepsizes much larger than $\|\omega\|_\infty^{-1} = 10^{-4}$, as is confirmed by the results in Table 4.11.

Similarly, explicit methods, which exist in this specific case since the problem is separable (see [69, Chapter 8]), suffer from a similar restriction on the stepsize because of stability reasons. In particular we consider a composition method, having order 6, based on the Störmer-Verlet method (see [49, Chapter II.4] for details), requiring 18 function evaluations per step:[4] the results listed in Table 4.12 clearly confirm this fact.

Conversely, the use of Newton-type iterations for solving the discrete problems generated by the HBVM(6,3) method, makes it possible to use much larger stepsizes, thus allowing to approximate the low frequencies without being hindered by the high ones. By using the blended iteration (ii) and the triangular splitting iteration (iii), one obtains the figures in Table 4.13. Even when using very coarse stepsizes, the approximation of the slowly-oscillating components of the solution (24 out of 28) is satisfactory: as an example in Figure 4.5 and Figure 4.6 there is the plot of the slowly-oscillating components q_{14} and p_{14}, respectively, by using a finer step, $h = 10^{-4}$, and a much coarser one, $h = 0.1$.[5] Last but not least, from the figures in Table 4.13, one sees that the triangular splitting procedure (iii) is the most effective one, though using only 2 inners iterations.

[3] The original problem reported in [49] is obtained by setting $m = 3$ and $\omega_i = 50, i = 1, \dots, m$ in (4.61).

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

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

Table 4.11: Fixed-point iterations for solving problem (4.61)-(4.63) on the interval $[0, 10]$, by using the HBVM$(6, 3)$ method with stepsize h (*** means that the iteration does not converge).

h	fixed-point iterations
10^{-4}	2274586
$2 \cdot 10^{-4}$	1901907
$4 \cdot 10^{-4}$	4539930
$5 \cdot 10^{-4}$	***

Table 4.12: Relative error on the Hamiltonian, obtained by using a sixth-order explicit composition method based on the Störmet-Verlet method, for solving problem (4.61)-(4.63) on the interval $[0, 10]$, by using stepsize h (*** means that the iteration does not converge).

h	Hamiltonian error
10^{-5}	$9.2 \cdot 10^{-8}$
$5 \cdot 10^{-5}$	$1.5 \cdot 10^{-3}$
10^{-4}	$8.4 \cdot 10^{-2}$
$2 \cdot 10^{-4}$	***
$4 \cdot 10^{-4}$	***
$5 \cdot 10^{-4}$	***

Table 4.13: Newton-type iterations for solving problem (4.61)-(4.63), on the interval $[0, 10]$ by using the HBVM$(6, 3)$ method with stepsize h.

h	blended iterations	splitting iterations
10^{-4}	1628149	855119
$5 \cdot 10^{-4}$	599728	297919
10^{-3}	240486	140558
$5 \cdot 10^{-3}$	28819	19127
10^{-2}	12616	8839
$5 \cdot 10^{-2}$	2823	1613
10^{-1}	1738	971

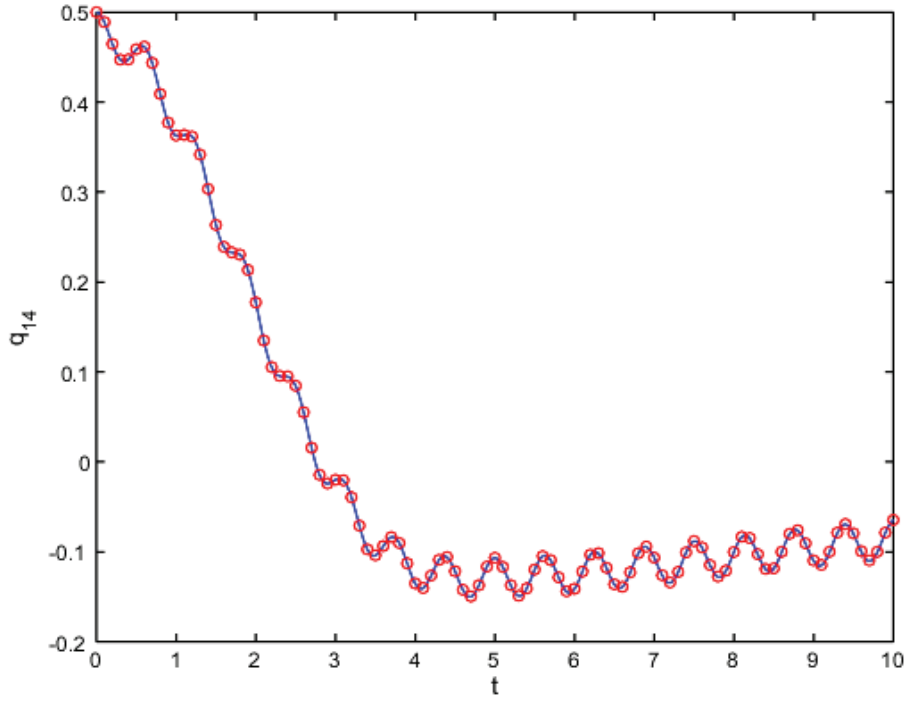

Figure 4.5: Numerical approximation obtained by using the HBVM(6,3) method with stepsizes $h = 10^{-4}$ (continuous line) and $h = 0.1$ (circles).

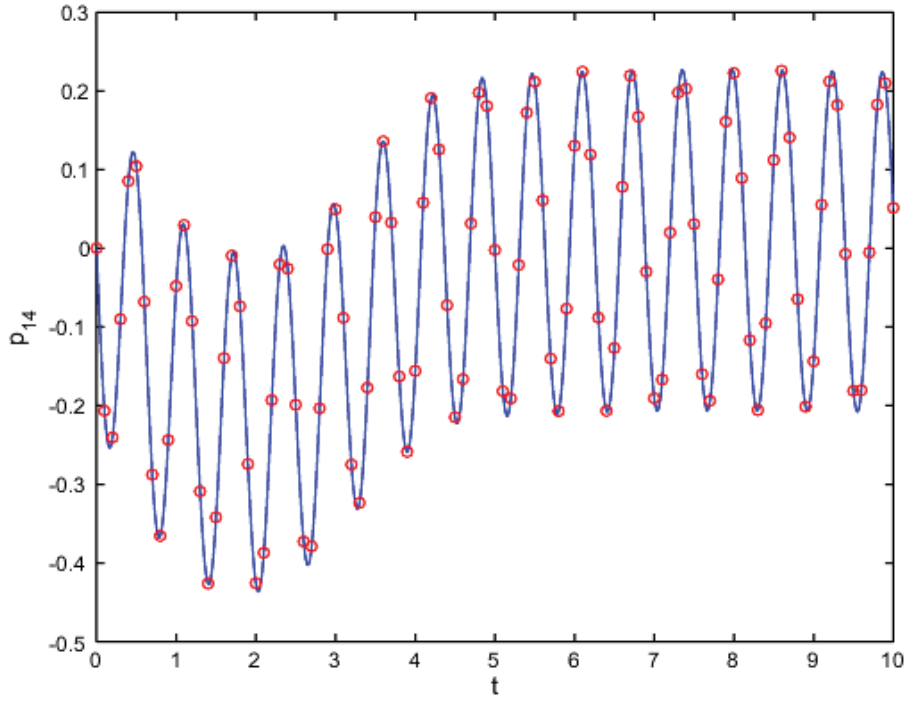

Figure 4.6: Numerical approximation obtained by using the HBVM(6,3) method with stepsizes $h = 10^{-4}$ (continuous line) and $h = 0.1$ (circles).

Chapter 5

Energy conserving methods for the semilinear wave equation

5.1 Introduction to the problem

In the present chapter we discuss energy-conservation issues concerning the well-known semilinear wave equation. For simplicity, though without loss of generality, we shall consider the 1D case,

$$\begin{aligned} u_{tt}(x,t) &= \alpha^2 u_{xx}(x,t) - f'(u(x,t)), \qquad (x,t) \in (0,1)\times(0,\infty), \\ u(x,0) &= \psi_0(x), \\ u_t(x,0) &= \psi_1(x), \qquad x \in (0,1), \end{aligned} \tag{5.1}$$

coupled with suitable boundary conditions. As usual, subscripts denote partial derivatives. We assume that the functions f, ψ_0 and ψ_1 are suitably regular, so that they define a regular solution $u(x,t)$ (f' denotes the derivative of f). The problem is completed by assigning suitable boundary conditions and we shall consider the different cases of periodic boundary conditions,

$$u(0,t) = u(1,t), \qquad u_x(0,t) = u_x(1,t), \qquad t > 0, \tag{5.2}$$

Dirichlet boundary conditions,

$$u(0,t) = \varphi_0(t), \qquad u(1,t) = \varphi_1(t), \qquad t > 0, \tag{5.3}$$

and Neumann boundary conditions,

$$u_x(0,t) = \phi_0(t), \qquad u_x(1,t) = \phi_1(t), \qquad t > 0, \tag{5.4}$$

with $\varphi_0(t)$, $\varphi_1(t)$, $\phi_0(t)$ and $\phi_1(t)$ suitably regular. In all cases, all the functions are assumed to satisfy suitable compatibility conditions, depending on the considered set of boundary conditions.

Remark 5.1.1. *It is worth mentioning that a problem for the semilinear wave equation defined on a generic interval $[a,b]$, could be always transformed to the form (5.1), by means of a linear transformation of the x variable. In such a case, the leading coefficient α in (5.1) changes accordingly (i.e. it becomes $(b-a)^{-1}\alpha$).*

By setting

$$v = u_t, \tag{5.5}$$

and defining the functional[1]

$$\mathcal{H}[u,v](t) = \int_0^1 \left[\frac{1}{2}v^2(x,t) + \frac{1}{2}\alpha^2 u_x^2(x,t) + f(u(x,t))\right] \mathrm{d}x \equiv \int_0^1 E(x,t)\mathrm{d}x, \tag{5.6}$$

we can rewrite (5.1) as the infinite-dimensional Hamiltonian system (for brevity, we neglect the arguments of the functions u and v)

$$\mathbf{z}_t = J\frac{\delta \mathcal{H}}{\delta \mathbf{z}}, \tag{5.7}$$

where

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \qquad \mathbf{z} = \begin{pmatrix} u \\ v \end{pmatrix}, \tag{5.8}$$

and

$$\frac{\delta \mathcal{H}}{\delta \mathbf{z}} = \left(\frac{\delta \mathcal{H}}{\delta u}, \frac{\delta \mathcal{H}}{\delta v}\right)^\top \tag{5.9}$$

is the functional derivative of $\mathcal{H}$. This latter is defined as follows: given a generic functional in the form

$$\mathcal{L}[q] = \int_a^b L(x, q(x), q'(x), \ldots)\mathrm{d}x,$$

its functional derivative is given by:

$$\frac{\delta \mathcal{L}}{\delta q} = \sum_{n=0}^{\infty}(-1)^n \frac{\mathrm{d}^n}{\mathrm{d}x^n}\frac{\partial L}{\partial q^{(n)}}, \qquad \text{with} \qquad q^{(n)} = \frac{\partial^n q}{\partial x^n}. \tag{5.10}$$

In the particular case when $L = L(x, q(x), q'(x))$, that is, when the function L does not depend on $q^{(n)}$ for $n > 1$, as in the case of (5.6), (5.10) becomes:

$$\frac{\delta \mathcal{L}}{\delta q} = \frac{\partial L}{\partial q} - \left(\frac{\mathrm{d}}{\mathrm{d}x}\frac{\partial L}{\partial q'}\right). \tag{5.11}$$

Exploiting (5.11), one can easily verify that (5.7)–(5.9) are equivalent to (5.1). In fact:

$$\mathbf{z}_t = \begin{pmatrix} u_t \\ v_t \end{pmatrix} = J\frac{\delta \mathcal{H}}{\delta \mathbf{z}} = \begin{pmatrix} \frac{\delta \mathcal{H}}{\delta v} \\ -\frac{\delta \mathcal{H}}{\delta u} \end{pmatrix} = \begin{pmatrix} v \\ \alpha^2 u_{xx} - f'(u) \end{pmatrix},$$

or

$$\begin{aligned} u_t(x,t) &= v(x,t), \qquad (x,t) \in (0,1)\times(0,\infty), \\ v_t(x,t) &= \alpha^2 u_{xx}(x,t) - f'(u(x,t)), \end{aligned} \tag{5.12}$$

that is, the first-order formulation of the differential equation in (5.1).

Remark 5.1.2. *Even if we are considering a problem for the semilinear wave equation, the arguments in this chapter can be extended to a generic Hamiltonian PDE in the form (5.7), such as the nonlinear wave equation, the nonlinear Schrödinger equation, the nonlinear beam equation, the Euler equations of hydrodynamics and numerous models that derive from it.*

[1] The domain of integration is the spatial interval on which is defined problem (5.1), in our case $[0,1]$.

As is well known, an important feature of our problem is that, whatever the boundary conditions, the rate of change of the energy density integrated over an interval depends only on the net flux through its endpoints. In fact,

$$\begin{aligned} E_t(x,t) &= v(x,t)v_t(x,t) + \alpha^2 u_x(x,t)u_{xt}(x,t) + f'(u(x,t))u_t(x,t) \\ &= v(x,t)(\alpha^2 u_{xx}(x,t) - f'(u(x,t))) + \alpha^2 u_x(x,t)v_x(x,t) + f'(u(x,t))v(x,t) \\ &= \alpha^2(v(x,t)u_{xx}(x,t) + u_x(x,t)v_x(x,t)) = \alpha^2(u_x(x,t)v(x,t))_x \equiv -F_x(x,t). \end{aligned}$$

Consequently, one has (see (5.6))

$$\dot{\mathcal{H}}[\mathbf{z}](t) = \int_0^1 E_t(x,t)\mathrm{d}x = \alpha^2[u_x(x,t)v(x,t)]_{x=0}^1, \tag{5.13}$$

where, as usual, the dot denotes the time derivative.

We recast the Hamiltonian functional (5.6) in a more convenient form which will be useful in the sequel:

$$\begin{aligned} \mathcal{H}[\mathbf{z}](t) &= \int_0^1 E(x,t)\mathrm{d}x = \int_0^1 \left[\frac{1}{2}v^2(x,t) + \frac{1}{2}\alpha^2 u_x^2(x,t) + f(u(x,t))\right]\mathrm{d}x \\ &= \int_0^1 \left[\frac{1}{2}v^2(x,t) + \frac{1}{2}\alpha^2[(u(x,t)u_x(x,t))_x - u(x,t)u_{xx}(x,t)] + f(u(x,t))\right]\mathrm{d}x \\ &= \int_0^1 \left[\frac{1}{2}v^2(x,t) - \frac{\alpha^2}{2}u(x,t)u_{xx}(x,t) + f(u(x,t))\right]\mathrm{d}x + \frac{\alpha^2}{2}[u(x,t)u_x(x,t)]_{x=0}^1. \end{aligned} \tag{5.14}$$

In the present chapter, we shall focus our attention on numerical techniques based on the method of lines approach, able to provide a full discretization of the original system with the discrete energy behaving consistently with the energy function associated with (5.1) (see (5.13)). In particular, in each of the next three sections of this chapter we shall consider problem (5.1) coupled with the boundary condition (5.2), (5.3), and (5.4), respectively, and we shall consider a finite difference approach to obtain a semi-discrete model whose full-discretization will be accomplished by means of an energy-conserving method in the class of HBVMs. In Section 5.5, the case of periodic boundary conditions is further investigated by considering higher-order finite difference schemes or a Fourier-Galerkin spectral method for the spatial semi-discretization. In the last section we present a few numerical tests in order to show the effective benefit in the use of energy-conserving methods also in the field of Hamiltonian PDEs.

5.2 The case of periodic boundary condition

Let us consider problem (5.1) coupled with the periodic boundary conditions (5.2). In this case the Hamiltonian functional given by (5.14) becomes:

$$\mathcal{H}[\mathbf{z}](t) = \int_0^1 \left[\frac{1}{2}v^2(x,t) - \frac{\alpha^2}{2}u(x,t)u_{xx}(x,t) + f(u(x,t))\right]\mathrm{d}x, \tag{5.15}$$

and from (5.13) one has

$$\dot{\mathcal{H}}[\mathbf{z}](t) = \alpha^2[u_x(x,t)v(x,t)]_{x=0}^1 = 0,$$

because of the periodic boundary conditions (5.2), so that (5.15) is a conserved quantity and at time $t = h$ one has

$$\mathcal{H}[\mathbf{z}](h) - \mathcal{H}[\mathbf{z}](0) = 0.$$

5.2.1 Semi-discretization

For numerically solving problem (5.1)-(5.2), let us introduce the following discretization of the spatial variable,

$$x_i = i\Delta x, \qquad i = 0, \ldots, N, \qquad \Delta x = 1/N,$$

and the vectors:

$$\mathbf{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_{N-1} \end{pmatrix}, \qquad \mathbf{q}(t) = \begin{pmatrix} u_0(t) \\ \vdots \\ u_{N-1}(t) \end{pmatrix}, \qquad \mathbf{p}(t) = \begin{pmatrix} v_0(t) \\ \vdots \\ v_{N-1}(t) \end{pmatrix} \in \mathbb{R}^N,$$

with

$$u_i(t) \approx u(x_i, t), \qquad v_i(t) \approx v(x_i, t) \equiv u_t(x_i, t). \tag{5.16}$$

Because of the periodic boundary condition (5.2), we also set:

$$u_N(t) \equiv u_0(t), \qquad u_{-1}(t) \equiv u_{N-1}(t), \qquad t \geq 0.$$

Approximating the second derivative in (5.12) as

$$u_{xx}(x_i, t) \approx \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{\Delta x^2}, \qquad i = 0, \ldots, N-1, \tag{5.17}$$

yields the following semi-discrete problem

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{p}, \\ \dot{\mathbf{p}} &= -\frac{\alpha^2}{\Delta x^2} T_N \mathbf{q} - f'(\mathbf{q}), \qquad t > 0, \end{aligned} \tag{5.18}$$

and the following approximation of the Hamiltonian (5.15),

$$H \equiv H(\mathbf{q}, \mathbf{p}) = \Delta x \left[\frac{\mathbf{p}^\top \mathbf{p}}{2} + \alpha^2 \frac{\mathbf{q}^\top T_N \mathbf{q}}{2\Delta x^2} + \mathbf{e}^\top f(\mathbf{q}) \right], \tag{5.19}$$

where T_N is a symmetric and circulant matrix,[2]

$$T_N = \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ -1 & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N}, \tag{5.20}$$

and

$$\mathbf{e} = \begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}^\top \in \mathbb{R}^N. \tag{5.21}$$

Problem (5.18) is clearly Hamiltonian. In fact, one has

$$\dot{\mathbf{q}} = \frac{1}{\Delta x} \nabla_{\mathbf{p}} H, \qquad \dot{\mathbf{p}} = -\frac{1}{\Delta x} \nabla_{\mathbf{q}} H,$$

[2] Because of the periodic boundary condition (5.2)

or, by introducing the vector

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

one obtains the more compact form

$$\dot{\mathbf{y}} = J_N \nabla H(\mathbf{y}), \qquad \text{with} \qquad J_N = \frac{1}{\Delta x} \begin{pmatrix} & I_N \\ -I_N & \end{pmatrix}, \tag{5.22}$$

where here and in the sequel we use, when appropriate, the notation $H(\mathbf{y}) = H(\mathbf{q}, \mathbf{p})$. Consequently,

$$\dot{H}(\mathbf{y}) = \nabla H(\mathbf{y})^\top \dot{\mathbf{y}} = \nabla H(\mathbf{y})^\top J_N \nabla H(\mathbf{y}) = 0,$$

since J_N is skew-symmetric. One then concludes that the discrete approximation (5.19) to (5.15) is a conserved quantity for the semi-discrete problem (5.22). By writing (5.19) in componentwise form

$$H(\mathbf{q}, \mathbf{p}) = \Delta x \sum_{i=0}^{N-1} \left(\frac{1}{2} v_i^2 - \alpha^2 u_i \frac{u_{i-1} - 2u_i + u_{i+1}}{2\Delta x^2} + f(u_i) \right),$$

one notices that (5.19) is nothing but the approximation of (5.15) via the composite trapezoidal rule (provided that the second derivative u_{xx} has been previously approximated as indicated in (5.17)).

5.2.2 Full discretization

The Hamiltonian problem (5.22), coupled with the initial condition (see (5.1))

$$\mathbf{y}_0 \equiv \mathbf{y}(0) = \begin{pmatrix} \psi_0(\mathbf{x}) \\ \psi_1(\mathbf{x}) \end{pmatrix},$$

can be discretized in time by using a HBVM(k, s) method. Let us then expand the right-hand side in (5.22) similarly as done in (3.1)–(3.2)

$$\dot{\mathbf{y}}(ch) = \sum_{j \geq 0} \gamma_j(\mathbf{y}) P_j(c), \quad c \in [0, 1], \tag{5.23}$$

with

$$\gamma_j(\mathbf{y}) = \int_0^1 J_N \nabla H(\mathbf{y}(\tau h)) P_j(\tau) \mathrm{d}\tau, \quad j \geq 0, \tag{5.24}$$

and consider the polynomial approximation of degree s given by (see (3.5)):

$$\begin{aligned} \dot{\boldsymbol{\sigma}}(ch) &= \sum_{j=0}^{s-1} P_j(c) \int_0^1 P_j(\tau) J_N \nabla H(\boldsymbol{\sigma}(\tau h)) \mathrm{d}\tau \equiv \sum_{j=0}^{s-1} P_j(c) \gamma_j(\boldsymbol{\sigma}), \quad c \in [0, 1], \\ \boldsymbol{\sigma}(0) &= \mathbf{y}_0. \end{aligned} \tag{5.25}$$

In such a case, one obtains energy conservation since

$$\begin{aligned} H(\boldsymbol{\sigma}(h)) - H(\boldsymbol{\sigma}(0)) &= h \int_0^1 \nabla H(\boldsymbol{\sigma}(\tau h))^\top \dot{\boldsymbol{\sigma}}(\tau h) \mathrm{d}\tau \\ &= h \int_0^1 \nabla H(\boldsymbol{\sigma}(\tau h))^\top \sum_{j=0}^{s-1} P_j(\tau) \gamma_j(\boldsymbol{\sigma}) \mathrm{d}\tau = h\Delta x^2 \sum_{j=0}^{s-1} \gamma_j(\boldsymbol{\sigma})^\top J_N \gamma_j(\boldsymbol{\sigma}) = 0, \end{aligned} \tag{5.26}$$

being $J_N^{-\top} = \Delta x^2 J_N$. Consequently, if one is able to exactly compute the integrals by means of a quadrature rule based at $k \geq s$ points, with k large enough, energy conservation is gained. In the sequel we shall consider the quadrature rule based at $k \geq s$ Gaussian points, so that, according to Remark 3.4.1, we shall obtain energy conservation in the case when H is a polynomial of degree $\nu \geq 2$,[3] that is $f \in \Pi_\nu$, and k is an integer such that (see (3.17))

$$k \geq \frac{1}{2}\nu s \qquad \Leftrightarrow \qquad \nu \leq \frac{2k}{s}. \tag{5.27}$$

If f, and then H, is not a polynomial, the use of a Gaussian quadrature formula of order $2k$ to approximate the integrals in (5.24) would give (see (3.10))

$$\begin{aligned} \gamma_j(\boldsymbol{\sigma}) \;&\equiv\; \int_0^1 J_N \nabla H(\boldsymbol{\sigma}(\tau h)) P_j(\tau) \mathrm{d}\tau \\ &=\; \underbrace{\sum_{\ell=1}^{k} b_\ell P_j(c_\ell) J_N \nabla H(\boldsymbol{\sigma}(c_\ell h))}_{\hat{\gamma}_j(\boldsymbol{\sigma})} \;+\; \Delta_j(h) \equiv \hat{\gamma}_j(\boldsymbol{\sigma}) + \Delta_j(h), \\ \text{with} \quad \Delta_j(h) &= O(h^{2k-j}), \quad j = 0, \ldots, s-1. \end{aligned} \tag{5.28}$$

In such a case, we have a different polynomial $\mathbf{u} \in \Pi_s$ solution of the problem

$$\dot{\mathbf{u}}(ch) = \sum_{j=0}^{s-1} \hat{\gamma}_j(\mathbf{u}) P_j(c), \qquad c \in [0,1], \qquad \mathbf{u}(0) = \mathbf{y}_0, \tag{5.29}$$

in place of $\boldsymbol{\sigma}$ solution of (5.25). As a consequence, by taking into account (5.22), (5.28)–(5.29), and the result of Lemma 2.1.2, the error on the Hamiltonian H, at $t = h$, is:

$$\begin{aligned} H(\mathbf{u}(h)) - H(\mathbf{u}(0)) \;&=\; h \int_0^1 \nabla H(\mathbf{u}(\tau h))^\top \dot{\mathbf{u}}(\tau h) \mathrm{d}\tau \\ &=\; h \int_0^1 \nabla H(\mathbf{u}(\tau h))^\top \sum_{j=0}^{s-1} P_j(\tau) \left(\gamma_j(\mathbf{u}) - \Delta_j(h)\right) \mathrm{d}\tau \\ &=\; h\Delta x^2 \sum_{j=0}^{s-1} \left[\overbrace{\gamma_j(\mathbf{u})^\top J_N \gamma_j(\mathbf{u})}^{=0} - \gamma_j(\mathbf{u})^\top J_N \Delta_j(h) \right] \\ &=\; h \underbrace{\Delta x \cdot N}_{=1} \cdot O(h^{2k}) \;\equiv\; O\left(h^{2k+1}\right). \end{aligned} \tag{5.30}$$

Consequently, choosing k large enough allows us to approximate the Hamiltonian H within full machine accuracy.

Summing up all the previous arguments, and taking into account the results in Chapter 3, we can state the following result.

Theorem 5.2.1. *Assume $k \geq s$, and define $\mathbf{y}_1 = \mathbf{u}(h)$ as the new approximation to $\mathbf{y}(h)$ provided by a HBVM(k, s) method used with stepsize h. One then obtains:*

$$\mathbf{y}_1 - \mathbf{y}(h) = O(h^{2s+1}),$$

[3] Indeed, H contains at least a quadratic term.

that is the method has order $2s$ *(see Corollary 3.1.1). Moreover, assuming that* f *is suitably regular:*

$$H(\mathbf{y}_1) - H(\mathbf{y}_0) = \begin{cases} 0, & \text{if } f \in \Pi_\nu \text{ and } \nu \leq 2k/s, \\ O(h^{2k+1}), & \text{otherwise.} \end{cases}$$

As a consequence of Theorem 5.2.1, we can do similar considerations as those in Remark 3.4.1.

5.3 The case of Dirichlet boundary conditions

Let us now consider the case when the considered problem is given by (5.1) with boundary conditions (5.3). In such a case $\mathcal{H}[\mathbf{z}]$, given in (5.14), is no more conserved. In fact, from (5.13) one obtains:

$$\dot{\mathcal{H}}[\mathbf{z}](t) = \alpha^2[u_x(x,t)v(x,t)]_{x=0}^1 = \alpha^2[u_x(1,t)\varphi_1'(t) - u_x(0,t)\varphi_0'(t)]. \tag{5.31}$$

Equation (5.31) may be interpreted as the instant variation of the energy which is released or gained by the system at time t. Thus, the continuous Hamiltonian (5.6), though no more conserved, has a *prescribed variation in time*. From (5.31), at time $t = h$ one easily obtains:

$$\mathcal{H}[\mathbf{z}](h) - \mathcal{H}[\mathbf{z}](0) = \int_0^h \dot{\mathcal{H}}[\mathbf{z}](t)\mathrm{d}t = \int_0^h \alpha^2[u_x(1,t)\varphi_1'(t) - u_x(0,t)\varphi_0'(t)]\mathrm{d}t. \tag{5.32}$$

5.3.1 Semi-discretization

In order for numerically solving problem (5.1)–(5.3), let us introduce the following discretization of the spatial variable,

$$x_i = i\Delta x, \qquad i = 0, \ldots, N+1, \qquad \Delta x = 1/(N+1), \tag{5.33}$$

and the vectors:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}, \quad \mathbf{q}(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_N(t) \end{pmatrix}, \quad \mathbf{p}(t) = \begin{pmatrix} v_1(t) \\ \vdots \\ v_N(t) \end{pmatrix} \in \mathbb{R}^N, \tag{5.34}$$

with $u_i(t)$ and $v_i(t)$ formally defined as in (5.16). Approximating the second derivatives in (5.12) as follows,

$$u_{xx}(x_i,t) \approx \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{\Delta x^2}, \qquad i = 1, \ldots, N, \tag{5.35}$$

and moreover,

$$u_x(1,t) \approx \frac{u_{N+1}(t) - u_N(t)}{\Delta x}, \qquad u_x(0,t) \approx \frac{u_1(t) - u_0(t)}{\Delta x}, \tag{5.36}$$

we obtain the following semi-discrete approximation to the Hamiltonian (5.14):

$$H = \Delta x \sum_{i=1}^{N} \left(\frac{1}{2}v_i^2 - \alpha^2 u_i \frac{u_{i-1} - 2u_i + u_{i+1}}{2\Delta x^2} + f(u_i) \right) + \alpha^2 \left[u_{N+1} \frac{u_{N+1} - u_N}{2\Delta x} + u_0 \frac{u_0 - u_1}{2\Delta x} \right]. \tag{5.37}$$

Moreover, because of the boundary condition (5.3), we set:

$$u_0(t) = \varphi_0(t), \qquad u_{N+1}(t) = \varphi_1(t), \tag{5.38}$$

so that (5.37) becomes:

$$H = \Delta x \sum_{i=1}^{N} \left(\frac{1}{2} v_i^2 - \alpha^2 u_i \frac{u_{i-1} - 2u_i + u_{i+1}}{2\Delta x^2} + f(u_i) \right) + \alpha^2 \left[\varphi_1 \frac{\varphi_1 - u_N}{2\Delta x} + \varphi_0 \frac{\varphi_0 - u_1}{2\Delta x} \right],$$

which can be rewritten in vector form as

$$H \equiv H(\mathbf{q}, \mathbf{p}, t) = \Delta x \left[\frac{\mathbf{p}^\top \mathbf{p}}{2} + \alpha^2 \frac{\mathbf{q}^\top T_N \mathbf{q}}{2\Delta x^2} + \mathbf{e}^\top f(\mathbf{q}) \right] + \alpha^2 \left[\frac{\boldsymbol{\varphi}(t)^\top \boldsymbol{\varphi}(t)}{2\Delta x} - \frac{\mathbf{q}^\top \boldsymbol{\varphi}(t)}{\Delta x} \right], \tag{5.39}$$

where $\mathbf{e}$ has been defined in (5.21), and moreover:

$$T_N = \begin{pmatrix} 2 & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{N \times N}, \qquad \boldsymbol{\varphi}(t) = \begin{pmatrix} \varphi_0(t) \\ 0 \\ \vdots \\ 0 \\ \varphi_1(t) \end{pmatrix} \in \mathbb{R}^N. \tag{5.40}$$

With reference to (5.39), the corresponding semi-discrete problem is then given by:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{p} \equiv \frac{1}{\Delta x} \nabla_{\mathbf{p}} H, \qquad t > 0, \\ \dot{\mathbf{p}} &= -\frac{\alpha^2}{\Delta x^2} T_N \mathbf{q} + \frac{\alpha^2}{\Delta x^2} \boldsymbol{\varphi} - f'(\mathbf{q}) \equiv -\frac{1}{\Delta x} \nabla_{\mathbf{q}} H, \end{aligned} \tag{5.41}$$

which is clearly Hamiltonian, though the Hamiltonian (5.39) is now non-autonomous, as a consequence of the boundary conditions (5.3). For this reason, the Hamiltonian (5.39) is not conserved, in fact one has (compare with (5.31)),

$$\frac{\mathrm{d}}{\mathrm{d}t} H(\mathbf{q}, \mathbf{p}, t) = \frac{\partial}{\partial t} H(\mathbf{q}, \mathbf{p}, t) = \alpha^2 \left[\frac{\varphi_1(t) - u_N(t)}{\Delta x} \varphi_1'(t) - \frac{u_1(t) - \varphi_0(t)}{\Delta x} \varphi_0'(t) \right]. \tag{5.42}$$

One then obtains the following semi-discrete analogue of (5.32) (see (5.36) and (5.38)):

$$\begin{aligned} &H(\mathbf{q}(h), \mathbf{p}(h), h) - H(\mathbf{q}(0), \mathbf{p}(0), 0) = \\ &= \int_0^h \alpha^2 \left[\frac{\varphi_1(t) - u_N(t)}{\Delta x} \varphi_1'(t) - \frac{u_1(t) - \varphi_0(t)}{\Delta x} \varphi_0'(t) \right] \mathrm{d}t. \end{aligned} \tag{5.43}$$

In order to conveniently handle problem (5.41), we at first transform it into an enlarged autonomous Hamiltonian system, by introducing the following auxiliary conjugate scalar variables,

$$\tilde{q} \equiv t, \qquad \tilde{p}, \tag{5.44}$$

and the augmented Hamiltonian (compare with (5.39)),

$$\begin{aligned} \tilde{H}(\mathbf{q}, \mathbf{p}, \tilde{q}, \tilde{p}) &= \Delta x \left[\frac{\mathbf{p}^\top \mathbf{p}}{2} + \alpha^2 \frac{\mathbf{q}^\top T_N \mathbf{q}}{2\Delta x^2} + \mathbf{e}^\top f(\mathbf{q}) \right] + \alpha^2 \left[\frac{\boldsymbol{\varphi}(\tilde{q})^\top \boldsymbol{\varphi}(\tilde{q})}{2\Delta x} - \frac{\mathbf{q}^\top \boldsymbol{\varphi}(\tilde{q})}{\Delta x} \right] + \tilde{p}, \\ &\equiv H(\mathbf{q}, \mathbf{p}, \tilde{q}) + \tilde{p}. \end{aligned} \tag{5.45}$$

The dynamical system corresponding to this new Hamiltonian function is, for $t > 0$:

$$\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{p} \equiv \frac{1}{\Delta x}\nabla_{\mathbf{p}}\tilde{H}, \\
\dot{\mathbf{p}} &= -\frac{\alpha^2}{\Delta x^2}T_N\mathbf{q} + \frac{\alpha^2}{\Delta x^2}\boldsymbol{\varphi} - f'(\mathbf{q}) \equiv -\frac{1}{\Delta x}\nabla_{\mathbf{q}}\tilde{H}, \\
\frac{\mathrm{d}}{\mathrm{d}t}\tilde{q} &= 1 \equiv \frac{\partial}{\partial\tilde{p}}\tilde{H}, \\
\frac{\mathrm{d}}{\mathrm{d}t}\tilde{p} &= \alpha^2\left[\frac{u_1-\varphi_0(\tilde{q})}{\Delta x}\varphi_0'(\tilde{q}) - \frac{\varphi_1(\tilde{q})-u_N}{\Delta x}\varphi_1'(\tilde{q})\right] \equiv -\frac{\partial}{\partial\tilde{q}}\tilde{H},
\end{aligned} \tag{5.46}$$

with initial conditions given by (see (5.1))

$$\mathbf{q}(0) = \psi_0(\mathbf{x}), \qquad \mathbf{p}(0) = \psi_1(\mathbf{x}), \qquad \tilde{q}(0) = \tilde{p}(0) = 0. \tag{5.47}$$

The first three equation in (5.46) exactly coincide with (5.41) (considering that $\tilde{q} \equiv t$), whereas the last one allows for the conservation of $\tilde{H}$:

$$\tilde{H}(\mathbf{q}(t),\mathbf{p}(t),\tilde{q}(t),\tilde{p}(t)) = \tilde{H}(\mathbf{q}(0),\mathbf{p}(0),0,0) \equiv H(\mathbf{q}(0),\mathbf{p}(0),0), \qquad t \geq 0.$$

Indeed, by virtue of (5.46), one readily sees that

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{H}(\mathbf{q},\mathbf{p},\tilde{q},\tilde{p}) = \overbrace{\nabla_{\mathbf{q}}\tilde{H}^\top\dot{\mathbf{q}} + \nabla_{\mathbf{p}}\tilde{H}^\top\dot{\mathbf{p}}}^{=0} + \underbrace{\frac{\partial}{\partial\tilde{q}}\tilde{H}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{q} + \frac{\partial}{\partial\tilde{p}}\tilde{H}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{p}}_{=0} = 0. \tag{5.48}$$

Remark 5.3.1. *Taking into account that $\tilde{q} \equiv t$, (5.45) and the last equation in (5.46), one has that (5.48) is equivalent to (5.42), and thus one concludes that to keep constant $\tilde{H}(\mathbf{q}(t),\mathbf{p}(t),t,\tilde{p}(t))$ along the solution of (5.46) is equivalent to (5.43). Consequently, by conserving the augmented Hamiltonian $\tilde{H}$, one obtains that H satisfies a prescribed variation in time which, in turn, is consistent with the corresponding continuous one (compare (5.42) with (5.31) and (5.43) with (5.32)).*

In order to simplify the notation, let us set

$$\mathbf{y} = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \\ \tilde{q} \\ \tilde{p}, \end{pmatrix}, \qquad \tilde{J}_N = \left(\begin{array}{cc|c} & \frac{1}{\Delta x}I_N & \\ -\frac{1}{\Delta x}I_N & & \\ \hline & & 1 \\ & & -1 \end{array}\right), \tag{5.49}$$

so that (5.46)–(5.47) can be rewritten as

$$\dot{\mathbf{y}} = \tilde{J}_N\nabla\tilde{H}(\mathbf{y}), \quad t \geq 0, \qquad \mathbf{y}(0) = (\psi_0(\mathbf{x})^\top, \psi_1(\mathbf{x})^\top, 0, 0)^\top \equiv \mathbf{y}_0. \tag{5.50}$$

5.3.2 Full discretization

The full discretization of (5.49)–(5.50) follows similar steps as those seen in Section 5.2.2 for (5.22). By expanding the right-hand side in (5.50) as done in (5.23)–(5.24), and considering $\boldsymbol{\sigma}$, the polynomial approximation of degree s given by (5.25) with H formally replaced with $\tilde{H}$ and J_N with matrix $\tilde{J}_N$ in (5.49), we still obtain energy conservation with similar steps as in (5.26), taking into account that $\tilde{J}_N^{-\top}$ is skew-symmetric, having the same shape as $\tilde{J}_N$ given in (5.49), but with $1/\Delta x$

replaced by Δx. Consequently, similarly as in the case of periodic boundary conditions, if one is able to exactly compute the integrals by means of a Gaussian quadrature rule of order $2k$, energy conservation is gained for the augmented problem (5.49)–(5.50). This is the case, provided that $\tilde{H}$ is a polynomial, that is, $f \in \Pi_\nu$ and $\varphi_0, \varphi_1 \in \Pi_\rho$ and, moreover k satisfies

$$k \geq \frac{1}{2}\max\{\nu s, 2\rho + s - 1, \rho + 2s - 1\} \tag{5.51}$$

(we observe that, in the case $\rho = 0$, such bound reduces to the bound (5.27) obtained in the case of periodic boundary conditions). Differently, one can approximate the integrals by means of a Gaussian quadrature of order $2k$ as in (5.28) but formally replacing H with $\tilde{H}$ and J_N with $\tilde{J}_N$. In such a case we have again a different polynomial $\mathbf{u} \in \Pi_s$, in place of $\boldsymbol{\sigma}$, solution of a problem formally still given by (5.29). As a consequence, with similar steps as in (5.30), the error in the Hamiltonian $\tilde{H}$, at $t = h$, is:

$$\tilde{H}(\mathbf{u}(h)) - \tilde{H}(\mathbf{u}(0)) = O(h^{2k+1}).$$

Consequently, by choosing k large enough, we can approximate the Hamiltonian $\tilde{H}$ within full machine accuracy.

All the above arguments can be summarized by the following theorem, which generalizes Theorem 5.2.1 to the present case.

Theorem 5.3.1. *Assume $k \geq s$, and define $\mathbf{y}_1 = \mathbf{u}(h)$ as the new approximation to $\mathbf{y}(h)$, solution of (5.50), provided by a HBVM(k, s) method used with stepsize h. One then obtains:*

$$\mathbf{y}_1 - \mathbf{y}(h) = O(h^{2s+1}),$$

that is the method has order $2s$. Moreover, assuming that f, φ_0 and φ_1 in (5.1)-(5.3) are suitably regular:

$$\tilde{H}(\mathbf{y}_1) - \tilde{H}(\mathbf{y}_0) = \begin{cases} 0, & \text{if } f \in \Pi_\nu,\ \varphi_0, \varphi_1 \in \Pi_\rho, \text{ and (5.51) holds true,} \\ O(h^{2k+1}), & \text{otherwise.} \end{cases}$$

Clearly, similar considerations to those stated in Remark 3.4.1 can be repeated in the present situation.

5.4 The case of Neumann boundary conditions

Let us now discuss the case when the considered problem is given by (5.1) with the Neumann boundary conditions (5.4). Similarly as in the case of Dirichlet boundary conditions, the Hamiltonian functional $\mathcal{H}[\mathbf{z}]$, given in (5.14), is not conserved. In fact, in the present case, (5.13) reduces to:

$$\dot{\mathcal{H}}[\mathbf{z}](t) = \alpha^2[u_x(x,t)v(x,t)]_{x=0}^1 = \alpha^2[\phi_1(t)v(1,t) - \phi_0(t)v(0,t)]. \tag{5.52}$$

As a consequence, similarly as in (5.32), at time $t = h$ one has

$$\mathcal{H}[\mathbf{z}](h) - \mathcal{H}[\mathbf{z}](0) = \int_0^h \dot{\mathcal{H}}[\mathbf{z}](t)\mathrm{d}t = \int_0^h \alpha^2[\phi_1(t)v(1,t) - \phi_0(t)v(0,t)]\mathrm{d}t. \tag{5.53}$$

In order to numerically solve problem (5.1) with boundary conditions (5.4), we use again the discretization (5.33) of the spatial variable, as well as the vectors defined at (5.34), with $u_i(t)$ and $v_i(t)$ formally defined as in (5.16). Approximating the derivatives as done in (5.35)-(5.36), one obtains

the very same semi-discretization (5.39) of the continuous Hamiltonian functional, with T_N defined as in (5.40), whereas, by considering the approximation (5.36),

$$\boldsymbol{\varphi}(t) = \begin{pmatrix} u_1 - \phi_0(t)\Delta x & 0 & \dots & 0 & u_N + \phi_1(t)\Delta x \end{pmatrix}^\top.$$

Nevertheless, it is preferable to derive another formulation of the semi-discrete Hamiltonian, as described below.

Starting from (5.37), one obtains

$$\begin{aligned}
H &= \Delta x \sum_{i=1}^{N} \left(\frac{1}{2} v_i^2 - \alpha^2 u_i \frac{u_{i-1} - 2u_i + u_{i+1}}{2\Delta x^2} + f(u_i) \right) + \frac{\alpha^2}{2} \left[u_{N+1} \frac{u_{N+1} - u_N}{\Delta x} - u_0 \frac{u_1 - u_0}{\Delta x} \right] \\
&= \Delta x \sum_{i=2}^{N-1} \left(\frac{1}{2} v_i^2 - \alpha^2 u_i \frac{u_{i-1} - 2u_i + u_{i+1}}{2\Delta x^2} + f(u_i) \right) + \frac{\alpha^2}{2} \left[u_{N+1} \frac{u_{N+1} - u_N}{\Delta x} - u_0 \frac{u_1 - u_0}{\Delta x} \right] \\
&\quad + \Delta x \left(\frac{1}{2} v_1^2 - \alpha^2 u_1 \frac{u_0 - 2u_1 + u_2}{2\Delta x^2} + f(u_1) + \frac{1}{2} v_N^2 - \alpha^2 u_N \frac{u_{N-1} - 2u_N + u_{N+1}}{2\Delta x^2} + f(u_N) \right) \\
&= \Delta x \sum_{i=2}^{N-1} \left(\frac{1}{2} v_i^2 - \alpha^2 u_i \frac{u_{i-1} - 2u_i + u_{i+1}}{2\Delta x^2} + f(u_i) \right) + \frac{\alpha^2}{2} \left[\frac{(u_{N+1} - u_N)^2}{\Delta x} + \frac{(u_1 - u_0)^2}{\Delta x} \right] \\
&\quad + \Delta x \left(\frac{1}{2} v_1^2 - \alpha^2 u_1 \frac{-u_1 + u_2}{2\Delta x^2} + f(u_1) + \frac{1}{2} v_N^2 - \alpha^2 u_N \frac{u_{N-1} - u_N}{2\Delta x^2} + f(u_N) \right).
\end{aligned}$$

The previous equation can be cast, more compactly, in vector form as:

$$H \equiv H(\mathbf{q}, \mathbf{p}, t) = \Delta x \left[\frac{\mathbf{p}^\top \mathbf{p}}{2} + \alpha^2 \frac{\mathbf{q}^\top T_N \mathbf{q}}{2\Delta x^2} + \mathbf{e}^\top f(\mathbf{q}) \right] + \alpha^2 \frac{\mathbf{w}(\mathbf{q}, t)^\top \mathbf{w}(\mathbf{q}, t)}{2\Delta x}, \tag{5.54}$$

where $\mathbf{e}$ has been defined in (5.21) and, moreover,

$$T_N = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{N \times N}, \quad \mathbf{w}(\mathbf{q}, t) = \begin{pmatrix} u_1 - u_0(t) \\ 0 \\ \vdots \\ 0 \\ u_{N+1}(t) - u_N \end{pmatrix} \in \mathbb{R}^N. \tag{5.55}$$

The semi-discrete Hamiltonian problem corresponding to (5.54) is then given by:

$$\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{p} \equiv \frac{1}{\Delta x} \nabla_{\mathbf{p}} H, \qquad t > 0, \\
\dot{\mathbf{p}} &= -\frac{\alpha^2}{\Delta x^2} T_N \mathbf{q} + \alpha^2 \frac{\Theta}{\Delta x^2} \mathbf{w}(\mathbf{q}, t) - f'(\mathbf{q}) \equiv -\frac{1}{\Delta x} \nabla_{\mathbf{q}} H,
\end{aligned} \tag{5.56}$$

with

$$\Theta = \begin{pmatrix} -1 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & 1 \end{pmatrix} \in \mathbb{R}^{N \times N}.$$

By considering the Neumann boundary conditions (5.4) and the used approximation (5.36) of the first space derivatives, one has (see (5.55))

$$\frac{\Theta}{\Delta x^2}\mathbf{w}(\mathbf{q},t) = \frac{1}{\Delta x^2}\begin{pmatrix} u_0(t)-u_1 \\ 0 \\ \vdots \\ 0 \\ u_{N+1}(t)-u_N \end{pmatrix} = \frac{1}{\Delta x}\begin{pmatrix} -\phi_0(t) \\ 0 \\ \vdots \\ 0 \\ \phi_1(t) \end{pmatrix} \equiv \frac{1}{\Delta x}\boldsymbol{\phi}(t),$$

so that one can derive the final shape of (5.56):

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{p}, \qquad t>0, \\ \dot{\mathbf{p}} &= -\frac{\alpha^2}{\Delta x^2}T_N\mathbf{q} + \frac{\alpha^2}{\Delta x}\boldsymbol{\phi}(t) - f'(\mathbf{q}), \end{aligned} \tag{5.57}$$

which is clearly Hamiltonian, even though, similarly as for (5.41), the Hamiltonian (5.54) is non-autonomous, as a consequence of the boundary conditions (5.4). Consequently, the Hamiltonian (5.54) is not conserved, in fact one has (compare with (5.52))

$$\frac{\mathrm{d}}{\mathrm{d}t}H(\mathbf{q},\mathbf{p},t) = \frac{\partial}{\partial t}H(\mathbf{q},\mathbf{p},t) = \frac{\alpha^2}{\Delta x}\mathbf{w}(\mathbf{q},t)^\top\frac{\partial}{\partial t}\mathbf{w}(\mathbf{q},t) = \alpha^2(\phi_1(t)v_{N+1}(t) - \phi_0(t)v_0(t)). \tag{5.58}$$

One then obtains the following analogue of (5.53):

$$H(\mathbf{q}(h),\mathbf{p}(h),h) - H(\mathbf{q}(0),\mathbf{p}(0),0) = \int_0^h \alpha^2(\phi_1(t)v_{N+1}(t) - \phi_0(t)v_0(t))\mathrm{d}t, \tag{5.59}$$

which, by taking into account (5.36) and the boundary conditions (5.4), becomes

$$\begin{aligned} &H(\mathbf{q}(h),\mathbf{p}(h),h) - H(\mathbf{q}(0),\mathbf{p}(0),0) = \\ &= \int_0^h \alpha^2\left[\phi_1(t)(v_N(t)+\Delta x\phi_1'(t)) - \phi_0(t)(v_1(t)-\Delta x\phi_0'(t))\right]\mathrm{d}t. \end{aligned} \tag{5.60}$$

As in the case of Dirichlet boundary conditions, we can introduce the couple of auxiliary conjugate variables (5.44) and the augmented Hamiltonian

$$\tilde{H}(\mathbf{q},\mathbf{p},\tilde{q},\tilde{p}) = \Delta x\left[\frac{\mathbf{p}^\top\mathbf{p}}{2} + \alpha^2\frac{\mathbf{q}^\top T_N\mathbf{q}}{2\Delta x^2} + \mathbf{e}^\top f(\mathbf{q})\right] + \alpha^2\frac{\mathbf{w}(\mathbf{q},\tilde{q})^\top\mathbf{w}(\mathbf{q},\tilde{q})}{2\Delta x} + \tilde{p} \equiv H(\mathbf{q},\mathbf{p},\tilde{q}) + \tilde{p}. \tag{5.61}$$

Consequently, the associated Hamiltonian system, coupled with the initial conditions given in (5.47), is:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{p} \equiv \frac{1}{\Delta x}\nabla_{\mathbf{p}}\tilde{H}, \qquad t>0, \\ \dot{\mathbf{p}} &= -\frac{\alpha^2}{\Delta x^2}T_N\mathbf{q} + \frac{\alpha^2}{\Delta x}\boldsymbol{\phi}(t) - f'(\mathbf{q}) \equiv -\frac{1}{\Delta x}\nabla_{\mathbf{q}}\tilde{H}, \\ \frac{\mathrm{d}}{\mathrm{d}t}\tilde{q} &= 1 \equiv \frac{\partial}{\partial\tilde{p}}\tilde{H}, \\ \frac{\mathrm{d}}{\mathrm{d}t}\tilde{p} &= -\alpha^2[\phi_1(\tilde{q})(v_N+\Delta x\phi_1'(\tilde{q})) - \phi_0(\tilde{q})(v_1-\Delta x\phi_0'(\tilde{q}))] \equiv -\frac{\partial}{\partial\tilde{q}}\tilde{H}, \end{aligned} \tag{5.62}$$

where in the last equation (compare with (5.58)) we have considered (5.36). In this way, the first three equations in (5.62) match (5.57) (since $\tilde{q} \equiv t$), whereas, similarly as in (5.48), the last one allows for the conservation of $\tilde{H}$:

$$\tilde{H}(\mathbf{q}(t), \mathbf{p}(t), \tilde{q}(t), \tilde{p}(t)) = \tilde{H}(\mathbf{q}(0), \mathbf{p}(0), 0, 0) \equiv H(\mathbf{q}(0), \mathbf{p}(0), 0), \qquad t \geq 0.$$

Remark 5.4.1. *With similar arguments as in Remark 5.3.1, by conserving the augmented Hamiltonian $\tilde{H}$, one obtains that H satisfies a prescribed variation in time which, in turn, is consistent with the corresponding continuous one (compare (5.58) with (5.52) and (5.59) with (5.53)).*

By introducing the array $\mathbf{y}$ and the matrix $\tilde{J}_N$ as in (5.49), problem (5.62), with the initial conditions (5.47), can be rewritten in the form:

$$\dot{\mathbf{y}} = \tilde{J}_N \nabla \tilde{H}(\mathbf{y}), \qquad t \geq 0, \qquad \mathbf{y}(0) = \left(\ \psi_0(\mathbf{x})^\top, \psi_1(\mathbf{x})^\top, 0, 0 \ \right)^\top \equiv \mathbf{y}_0. \tag{5.63}$$

Concerning the discretization issue, arguments similar to those in Section 5.3.2 apply to the present case. In particular, the following result holds true, the proof being similar to that of Theorems 5.2.1 and 5.3.1.

Theorem 5.4.1. *Assume $k \geq s$, and let define $\mathbf{y}_1 = \mathbf{u}(h)$ the new approximation to $\mathbf{y}(h)$, solution of (5.63), provided by a HBVM(k,s) method used with stepsize h. One then obtains:*

$$\mathbf{y}_1 - \mathbf{y}(h) = O(h^{2s+1}),$$

that is, the method has order $2s$. Moreover, assuming that f, ϕ_0 and ϕ_1 in (5.1)-(5.4) are suitably regular, one has:

$$\tilde{H}(\mathbf{y}_1) - \tilde{H}(\mathbf{y}_0) = \begin{cases} 0, & \text{if } f \in \Pi_\nu, \quad \phi_0, \phi_1 \in \Pi_\rho, \quad \text{with} \\ & 2k \geq \max\{\nu s, 2\rho + s - 1, 2s + \rho\}, \\ O(h^{2k+1}), & \text{otherwise.} \end{cases}$$

As is clear, considerations similar to those stated in Remark 3.4.1 can be repeated also in the present situation.

5.5 The case of periodic boundary conditions revisited

The case of periodic boundary conditions, i.e. (5.1)–(5.2), deserves to be further investigated. In fact, the finite-difference discretization considered above, turns out to provide a second-order spatial accuracy in the used stepsize Δx. When either Dirichlet or Neumann boundary conditions are specified, it is not easy to derive higher-order semi-discrete Hamiltonian formulations of the problem. Conversely, in the case of periodic boundary conditions, the use of higher-order central finite-difference approximations of the derivatives in (5.1), results in a semi-discrete Hamiltonian problem defined by a Hamiltonian function formally still given by (5.19), but with the matrix T_N defined in (5.20) suitably replaced with a different circulant and symmetric band-matrix. As an example, the following matrix provides a fourth-order spatial approximation (see, e.g., [2] for

additional examples):

$$T_N = \begin{pmatrix} \frac{5}{2} & -\frac{4}{3} & \frac{1}{12} & & & \frac{1}{12} & -\frac{4}{3} \\ -\frac{4}{3} & \ddots & \ddots & \ddots & & & \frac{1}{12} \\ \frac{1}{12} & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & \frac{1}{12} \\ \frac{1}{12} & & & \ddots & \ddots & \ddots & -\frac{4}{3} \\ -\frac{4}{3} & \frac{1}{12} & & & \frac{1}{12} & -\frac{4}{3} & \frac{5}{2} \end{pmatrix} \in \mathbb{R}^{N\times N}. \tag{5.64}$$

An alternative approach, which we shall investigate in the sequel, is that of using a Fourier-Galerkin expansion to obtain a spatial approximation. Galerkin methods require to expand the solution of the problem along a basis in which every component satisfies the associated boundary conditions. When the problem at hand is coupled with the periodic boundary conditions (5.2), a trigonometric basis is usually preferred (see for example [35, 42, 75]). For this purpose, let us consider the following complete set of orthonormal functions in $[0,1]$:

$$c_0(x) \equiv 1, \qquad c_k(x) = \sqrt{2}\cos(2k\pi x), \quad s_k(x) = \sqrt{2}\sin(2k\pi x), \qquad k = 1, 2, \ldots, \tag{5.65}$$

so that

$$\int_0^1 c_i(x)c_j(x)\mathrm{d}x = \int_0^1 s_i(x)s_j(x)\mathrm{d}x = \delta_{ij}, \qquad \int_0^1 c_i(x)s_j(x)\mathrm{d}x = 0, \qquad \forall i,j, \tag{5.66}$$

being δ_{ij} the Kronecker symbol. The following expansion of the solution of (5.1)-(5.2) is a slightly different way of writing the usual Fourier expansion in space:

$$\begin{aligned} u(x,t) &= c_0(x)\beta_0(t) + \sum_{n\geq 1}\left[c_n(x)\beta_n(t) + s_n(x)\eta_n(t)\right] \\ &\equiv \beta_0(t) + \sum_{n\geq 1}\left[c_n(x)\beta_n(t) + s_n(x)\eta_n(t)\right], \qquad x\in[0,1], \quad t\geq 0, \end{aligned} \tag{5.67}$$

with

$$\beta_n(t) = \int_0^1 c_n(x)u(x,t)\mathrm{d}x, \qquad \eta_n(t) = \int_0^1 s_n(x)u(x,t)\mathrm{d}x,$$

which is allowed because of the periodic boundary conditions (5.2). Consequently, by taking into account (5.66), the differential equation in (5.1) can be rewritten as:

$$\begin{aligned} \ddot{\beta}_n(t) &= -\alpha^2(2\pi n)^2\beta_n(t) \\ &\quad - \int_0^1 c_n(x) f'\left(\beta_0(t) + \sum_{n\geq 1}\left[c_n(x)\beta_n(t) + s_n(x)\eta_n(t)\right]\right)\mathrm{d}x, \quad n\geq 0, \\ \ddot{\eta}_n(t) &= -\alpha^2(2\pi n)^2\eta_n(t) \\ &\quad - \int_0^1 s_n(x) f'\left(\beta_0(t) + \sum_{n\geq 1}\left[c_n(x)\beta_n(t) + s_n(x)\eta_n(t)\right]\right)\mathrm{d}x, \quad n\geq 1, \end{aligned} \tag{5.68}$$

where the double dot denotes, as usual, the second time derivative. The initial conditions are clearly given by (see (5.1)):

$$\begin{aligned}\beta_n(0) &= \int_0^1 c_n(x)\psi_0(x)\mathrm{d}x, \qquad \eta_n(0) = \int_0^1 s_n(x)\psi_0(x)\mathrm{d}x,\\ \dot\beta_n(0) &= \int_0^1 c_n(x)\psi_1(x)\mathrm{d}x, \qquad \dot\eta_n(0) = \int_0^1 s_n(x)\psi_1(x)\mathrm{d}x.\end{aligned} \tag{5.69}$$

By introducing the infinite vectors

$$\begin{aligned}\boldsymbol{\omega}(x) &= \begin{pmatrix} c_0(x) & c_1(x) & s_1(x) & c_2(x) & s_2(x) & \dots \end{pmatrix}^\top,\\ \mathbf{q}(t) &= \begin{pmatrix} \beta_0(t) & \beta_1(t) & \eta_1(t) & \beta_2(t) & \eta_2(t) & \dots \end{pmatrix}^\top,\end{aligned} \tag{5.70}$$

the infinite matrix

$$D = \mathrm{diag}\begin{pmatrix} 0 & (2\pi)^2 & (2\pi)^2 & (4\pi)^2 & (4\pi)^2 & \dots \end{pmatrix}, \tag{5.71}$$

and considering that (see (5.67))

$$u(x,t) = \boldsymbol{\omega}(x)^\top \mathbf{q}(t), \tag{5.72}$$

problem (5.68) can be cast in vector form as:

$$\begin{aligned}\dot{\mathbf{q}}(t) &= \mathbf{p}(t), \qquad t>0,\\ \dot{\mathbf{p}}(t) &= -\alpha^2 D\mathbf{q}(t) - \int_0^1 \boldsymbol{\omega}(x) f'(\boldsymbol{\omega}(x)^\top \mathbf{q}(t))\mathrm{d}x,\end{aligned} \tag{5.73}$$

with the initial conditions (5.69) becoming, more compactly,

$$\mathbf{q}(0) = \int_0^1 \boldsymbol{\omega}(x)\psi_0(x)\mathrm{d}x, \qquad \mathbf{p}(0) = \int_0^1 \boldsymbol{\omega}(x)\psi_1(x)\mathrm{d}x. \tag{5.74}$$

The following result holds true.

Theorem 5.5.1. *Problem (5.73) is Hamiltonian, with Hamiltonian*

$$H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^\top\mathbf{p} + \frac{\alpha^2}{2}\mathbf{q}^\top D\mathbf{q} + \int_0^1 f(\boldsymbol{\omega}(x)^\top\mathbf{q})\mathrm{d}x. \tag{5.75}$$

This latter is equivalent to the Hamiltonian (5.6), via the expansion (5.67)–(5.72).

Proof. The first statement is straightforward, by considering that

$$\nabla_{\mathbf{q}} f(\boldsymbol{\omega}(x)^\top\mathbf{q})) = f'(\boldsymbol{\omega}(x)^\top\mathbf{q})\boldsymbol{\omega}(x).$$

The second statement then easily follows, by taking into account (5.72), from the fact that, see (5.5), (5.66), (5.67), and (5.70):

$$\begin{aligned}\int_0^1 v(x,t)^2\mathrm{d}x &= \int_0^1 u_t(x,t)^2\mathrm{d}x = \int_0^1 \left(\dot\beta_0(t) + \sum_{n\geq 1}\left[\dot\beta_n(t)c_n(x) + \dot\eta_n(t)s_n(x)\right]\right)^2 \mathrm{d}x\\ &= \dot\beta_0(t)^2 + \sum_{n\geq 1}\left[\dot\beta_n(t)^2 + \dot\eta_n(t)^2\right] = \mathbf{p}(t)^\top\mathbf{p}(t),\end{aligned}$$

and

$$\int_0^1 u_x(x,t)^2 \mathrm{d}x = \int_0^1 \left(\sum_{n\geq 1} 2\pi n \left[\eta_n(t) c_n(x) - \beta_n(t) s_n(x) \right] \right)^2 \mathrm{d}x$$
$$= \sum_{n\geq 1} (2\pi n)^2 \left[\eta_n(t)^2 + \beta_n(t)^2 \right] \;=\; \mathbf{q}(t)^\top D \mathbf{q}(t). \qquad \square$$

5.5.1 Truncated Fourier approximation

In the computational practice, it is mandatory to truncate the infinite expansion (5.67) to a finite sum:

$$u(x,t) \;\approx\; \beta_0(t) + \sum_{n=1}^{N} \left[c_n(x)\beta_n(t) + s_n(x)\eta_n(t) \right] \;\equiv\; u_N(x,t), \tag{5.76}$$

which converges more than exponentially with N to u, if this latter is an analytical function.[4] In other words, we look for an approximation to $u(x,t)$ belonging to the functional subspace (see (5.65))

$$\mathcal{V}_N = \mathrm{span}\{c_0(x), c_1(x), \ldots, c_N(x), s_1(x), \ldots, s_N(x)\}.$$

Clearly, such a truncated expansion will not satisfy problem (5.1)-(5.2). Nevertheless, in the spirit of Galerkin methods [6], by requiring that the residual

$$R(u_N) := (u_N)_{tt} - (u_N)_{xx} + f'(u_N),$$

is orthogonal to $\mathcal{V}_N$, one obtains the *weak formulation* of problem (5.1)-(5.2), consisting in the following $2N+1$ differential equations,

$$\begin{aligned}
\ddot{\beta}_n(t) \;&=\; -\alpha^2 (2\pi n)^2 \beta_n(t) \\
&\quad - \int_0^1 c_n(x) f' \left(\beta_0(t) + \sum_{n=1}^{N} \left[c_n(x)\beta_n(t) + s_n(x)\eta_n(t) \right] \right) \mathrm{d}x, \quad n = 0, \ldots, N, \\
\ddot{\eta}_n(t) \;&=\; -\alpha^2 (2\pi n)^2 \eta_n(t) \\
&\quad - \int_0^1 s_n(x) f' \left(\beta_0(t) + \sum_{n=1}^{N} \left[c_n(x)\beta_n(t) + s_n(x)\eta_n(t) \right] \right) \mathrm{d}x, \quad n = 1, \ldots, N,
\end{aligned} \tag{5.77}$$

approximating the leading ones in (5.68). Correspondingly, one defines the finite vectors (compare with (5.70)) in $\mathbb{R}^{2N+1}$,

$$\begin{aligned}
\boldsymbol{\omega}_N(x) \;&=\; \left(\; c_0(x) \quad c_1(x) \quad s_1(x) \quad c_2(x) \quad s_2(x) \quad \ldots \quad c_N(x) \quad s_N(x) \; \right)^\top, \\
\mathbf{q}_N(t) \;&=\; \left(\; \beta_0(t) \quad \beta_1(t) \quad \eta_1(t) \quad \beta_2(t) \quad \eta_2(t) \quad \ldots \quad \beta_N(t) \quad \eta_N(t) \; \right)^\top,
\end{aligned}$$

and the diagonal matrix (compare with (5.71))

$$D_N = \mathrm{diag} \left(\; 0 \quad (2\pi)^2 \quad (2\pi)^2 \quad (4\pi)^2 \quad (4\pi)^2 \quad \ldots \quad (2N\pi)^2 \quad (2N\pi)^2 \; \right) \in \mathbb{R}^{2N+1 \times 2N+1}. \tag{5.78}$$

Then, considering that (see (5.76))

$$u_N(x,t) = \boldsymbol{\omega}_N(x)^\top \mathbf{q}_N(t), \tag{5.79}$$

[4] We refer, e.g., to [35], for a corresponding comprehensive error analysis.

the equations (5.77), which have to be satisfied by (5.79), can be cast in vector form as:

$$\begin{aligned} \dot{\mathbf{q}}_N(t) &= \mathbf{p}_N(t), \qquad t > 0, \\ \dot{\mathbf{p}}_N(t) &= -\alpha^2 D_N \mathbf{q}_N(t) - \int_0^1 \boldsymbol{\omega}_N(x) f'(\boldsymbol{\omega}_N(x)^\top \mathbf{q}_N(t)) \mathrm{d}x, \end{aligned} \tag{5.80}$$

for a total of $4N+2$ differential equations. Clearly, from (5.69) one obtains that the initial conditions for (5.80) are given by:

$$\mathbf{q}_N(0) = \int_0^1 \boldsymbol{\omega}_N(x)\psi_0(x)\mathrm{d}x, \qquad \mathbf{p}_N(0) = \int_0^1 \boldsymbol{\omega}_N(x)\psi_1(x)\mathrm{d}x. \tag{5.81}$$

The following result then easily follows by means of arguments similar to those used to prove Theorem 5.5.1.

Theorem 5.5.2. *Problem (5.80) is Hamiltonian, with Hamiltonian*

$$H_N(\mathbf{q}_N, \mathbf{p}_N) = \frac{1}{2}\mathbf{p}_N^\top \mathbf{p}_N + \frac{\alpha^2}{2}\mathbf{q}_N^\top D_N \mathbf{q}_N + \int_0^1 f(\boldsymbol{\omega}_N(x)^\top \mathbf{q}_N)\mathrm{d}x. \tag{5.82}$$

We observe that (5.82) is equivalent to a truncated Fourier expansion of the Hamiltonian (5.6) (see also (5.75)). Thus, by introducing the vector $\mathbf{y}_N(t) = \begin{pmatrix} \mathbf{q}_N(t)^\top & \mathbf{p}_N(t)^\top \end{pmatrix}^\top \in \mathbb{R}^{4N+2}$ and the notation $H_N(\mathbf{y}_N) = H_N(\mathbf{q}_N, \mathbf{p}_N)$, we can cast (5.80) in the more compact form

$$\dot{\mathbf{y}}_N = J_N \nabla H_N(\mathbf{y}_N), \qquad \text{with} \qquad J_N = \begin{pmatrix} & I_{2N+1} \\ -I_{2N+1} & \end{pmatrix}, \tag{5.83}$$

with the initial condition (see (5.81))

$$\mathbf{y}_N(0) = \begin{pmatrix} \mathbf{q}_N(0)^\top & \mathbf{p}_N(0)^\top \end{pmatrix}^\top.$$

It is worth mentioning that using the initial conditions (5.81), in place of (5.74), results in an error e_N, in the initial data, given by

$$\begin{aligned} e_N^2 &= \int_0^1 \Big(\psi_0(x) - \boldsymbol{\omega}_N(x)^\top \mathbf{q}_N(0)\Big)^2 \mathrm{d}x + \int_0^1 \Big(\psi_1(x) - \boldsymbol{\omega}_N(x)^\top \mathbf{p}_N(0)\Big)^2 \mathrm{d}x \\ &= \sum_{n>N} \left[\left(\int_0^1 c_n(x)\psi_0(x)\mathrm{d}x \right)^2 + \left(\int_0^1 s_n(x)\psi_0(x)\mathrm{d}x \right)^2 \right] + \\ &\quad \sum_{n>N} \left[\left(\int_0^1 c_n(x)\psi_1(x)\mathrm{d}x \right)^2 + \left(\int_0^1 s_n(x)\psi_1(x)\mathrm{d}x \right)^2 \right]. \end{aligned} \tag{5.84}$$

Unless the finite-difference case, both e_N and the approximation (5.82) to the continuous Hamiltonian, converge more than exponentially in N (e_N to 0 and H_N to H), provided that the involved functions are analytical.

5.5.2 Full discretization

Since (5.83) is, for all $N \geq 0$, a Hamiltonian problem of dimension $4N + 2$ with autonomous Hamiltonian (5.82), this latter is conserved along the solution. Consequently, it is then appropriate

the use of an energy-conserving method for its numerical solution. In particular, Theorem 5.2.1 continues formally to hold for HBVM(k,s) methods. However, the integral appearing in (5.80) needs to be, in turn, approximated by means of a suitable quadrature rule. For this purpose, it could be convenient to use a composite trapezoidal rule, due to the fact that the argument is a periodic function. Consequently, having set

$$\mathbf{g}_N(x,t) = \boldsymbol{\omega}_N(x) f'(\boldsymbol{\omega}_N(x)^\top \mathbf{q}_N(t)), \tag{5.85}$$

the uniform mesh on $[0,1]$,

$$x_i = i\Delta x, \quad i = 0,\dots,m, \quad \Delta x = \frac{1}{m}, \tag{5.86}$$

and considering that

$$\mathbf{g}_N(0,t) = \mathbf{g}_N(1,t),$$

one obtains:

$$\begin{aligned}\int_0^1 \mathbf{g}_N(x,t)\mathrm{d}x &= \Delta x \sum_{i=1}^{m} \frac{\mathbf{g}_N(x_{i-1},t) + \mathbf{g}_N(x_i,t)}{2} + R(m)\\ &= \frac{1}{m}\sum_{i=0}^{m-1} \mathbf{g}_N(x_i,t) + R(m).\end{aligned} \tag{5.87}$$

Let us study the error $R(m)$. For this purpose, we need some preliminary result.

Lemma 5.5.1. *Let us consider the trigonometric polynomial*

$$p(x) = \sum_{k=0}^{K} \left[a_k \cos(2k\pi x) + b_k \sin(2k\pi x)\right], \tag{5.88}$$

and the uniform mesh (5.86). Then, for all $m \geq K+1$, one obtains:

$$\int_0^1 p(x)\mathrm{d}x = \frac{1}{m}\sum_{i=0}^{m-1} p(x_i).$$

Proof. See, e.g., [39, Th. 5.1.4]. □

Lemma 5.5.2. *Let us consider the trigonometric polynomial (5.88) and the uniform mesh (5.86). Then, for all $j = 0,\dots,N$, and provided that $m \geq N+K+1$, one obtains:*

$$\int_0^1 \cos(2j\pi x)p(x)\mathrm{d}x = \frac{1}{m}\sum_{i=0}^{m-1} \cos(2j\pi x_i)p(x_i), \tag{5.89}$$

$$\int_0^1 \sin(2j\pi x)p(x)\mathrm{d}x = \frac{1}{m}\sum_{i=0}^{m-1} \sin(2j\pi x_i)p(x_i). \tag{5.90}$$

Proof. By virtue of the prosthaphaeresis formulae, for all $j = 0,\dots,N$ and $k = 0,\dots,K$, one has:

$$\begin{aligned}\cos(2j\pi x)\cos(2k\pi x) &= \frac{1}{2}\left[\cos(2(k+j)\pi x) + \cos(2(k-j)\pi x)\right],\\ \cos(2j\pi x)\sin(2k\pi x) &= \frac{1}{2}\left[\sin(2(k+j)\pi x) + \sin(2(k-j)\pi x)\right],\\ \sin(2j\pi x)\cos(2k\pi x) &= \frac{1}{2}\left[\sin(2(k+j)\pi x) - \sin(2(k-j)\pi x)\right],\\ \sin(2j\pi x)\sin(2k\pi x) &= \frac{1}{2}\left[\cos(2(k-j)\pi x) - \cos(2(k+j)\pi x)\right].\end{aligned}$$

Consequently, the integrals at the left-hand side in (5.89)-(5.90) are trigonometric polynomials of degree at most $N+K$. By virtue of Lemma 5.5.1, it then follows that they are exactly computed by means of the composite trapezoidal rule at the corresponding right-hand sides, provided that $m \geq N+K+1$. □

By virtue of Lemma 5.5.2, the following result follows at once.

Theorem 5.5.3. *Let the function f appearing in (5.85) be a polynomial of degree ν and let us consider the uniform mesh (5.86). Then, with reference to (5.87), for all $m \geq \nu N + 1$ one obtains:*

$$R(m) = 0 \qquad i.e., \qquad \int_0^1 \mathbf{g}_N(x,t)\mathrm{d}x = \frac{1}{m}\sum_{i=0}^{m-1} \mathbf{g}_N(x_i,t).$$

For a general function f, the following result holds true.

Theorem 5.5.4. *Let the function $\mathbf{g}_N(x,t)$ defined at (5.85), with t a fixed parameter, belong to $W^{r,p}_{per}$, the Banach space of periodic functions on $\mathbb{R}$ whose distribution derivatives up to order r belong to $L^p_{per}(\mathbb{R})$. Then, with reference to (5.86)-(5.87), one has:*

$$R(m) = O(m^{-r}).$$

Proof. See [62, Th. 1.1]. □

We end this section by mentioning that for approximating the integral appearing in (5.80) also different approaches could be used: as an example, we refer to [40], for a comprehensive review on this topic.

5.6 Implementation of the methods

In this section, we sketch the application of a HBVM(k,s) method for solving (5.18) (the application to (5.46), (5.62) and (5.80) is similar). We consider the very first application of the method, so that the index of the time step can be skipped. Since the Hamiltonian of the semi-discrete formulation of the semilinear wave equation is separable[5], similarly as done in Section 4.7, by splitting the stage vector Y of the Runge-Kutta formulation, into Q and P, corresponding to the stages for $\mathbf{q}$ and $\mathbf{p}$ respectively, the discrete problem generated by a HBVM(k,s) method can be recast in terms of the s coefficients of the polynomial (5.29) as (see (4.51)):

$$F(\hat\gamma) \equiv \hat\gamma - \mathcal{P}_s^\top \Omega \otimes I_N \, G\left(\mathbf{e}\otimes \mathbf{q}_0 + h\mathbf{c}\otimes \mathbf{p}_0 - h^2 \mathcal{I}_s X_s \otimes I_N \,\hat\gamma\right) = \mathbf{0}, \tag{5.91}$$

being,

$$G(Q) = \frac{\alpha^2}{\Delta x^2} I_k \otimes T_N Q + f'(Q), \tag{5.92}$$

and (see (4.50))[6]

$$\hat\gamma = \mathcal{P}_s^\top \Omega \otimes I_N \, G(Q) \equiv \begin{pmatrix} \hat\gamma_0 \\ \vdots \\ \hat\gamma_{s-1} \end{pmatrix}.$$

[5] This is not the case when considering different Hamiltonian PDEs, such as, e.g., the nonlinear Schrödinger equation.

[6] Here, as an abuse of notation, $\hat\gamma_j$ is given by the entries of the vector $\hat\gamma_j(\mathbf{u})$ in (5.29) corresponding to the $\mathbf{q}$ components only. Consequently, it has halved dimension w.r.t. this latter vector.

Once (5.91) is solved, the new approximations are then given by (see (2.8)):

$$\mathbf{p}_1 = \mathbf{p}_0 + h\hat{\gamma}_0, \qquad \mathbf{q}_1 = \mathbf{q}_0 + h\mathbf{p}_0 + h^2\left(\frac{1}{2}\hat{\gamma}_0 - \xi_1\hat{\gamma}_1\right).$$

For the solution of (5.91), one could use the following simplified Newton iteration,

$$\left(I + \frac{\alpha^2 h^2}{\Delta x^2} X_s^2 \otimes T_N\right)\Delta^\ell = -F(\hat{\gamma}^\ell) \equiv \boldsymbol{\eta}^\ell, \qquad \hat{\gamma}^{\ell+1} = \hat{\gamma}^\ell + \Delta^\ell \qquad \ell = 0, 1, \ldots, \tag{5.93}$$

which only considers the (main) linear part of the function G (see (5.92)). In order to reduce the computational cost of such iteration, having dimension sN, we use a *blended iteration* formally defined as:

$$\boldsymbol{\eta}_1^\ell = \zeta^2 X_s^{-2} \otimes I_N \boldsymbol{\eta}^\ell, \tag{5.94}$$

$$\Delta^\ell = I_s \otimes M_N^{-1}\left[\boldsymbol{\eta}_1^\ell + I_s \otimes M_N^{-1}\left(\boldsymbol{\eta}^\ell - \boldsymbol{\eta}_1^\ell\right)\right], \qquad \ell = 0, 1, \ldots, \tag{5.95}$$

where

$$\zeta = \min_{\lambda \in \sigma(X_s)} |\lambda|, \qquad M_N = I_N + \left(\frac{\alpha h \zeta}{\Delta x}\right)^2 T_N, \tag{5.96}$$

with $\sigma(X_s)$ denoting the spectrum of matrix X_s. Consequently, the computational cost of each iteration is given by:

- the evaluation of $\boldsymbol{\eta}^\ell$ in (5.93). This requires k evaluations of the right-hand side of the second equation in (5.18) (see (5.91)–(5.93)) plus $(4ks + 3k + s)N$ *flops*;
- the evaluation of $\boldsymbol{\eta}_1^\ell$ in (5.94). Concerning the matrix $\zeta^{-1}X_s$; one can either invert and square it in advance, so that the costs for computing $\boldsymbol{\eta}_1^\ell$ is $2s^2N$ *flops*, or solve 2 tridiagonal linear systems, so that, once the factorization is computed[7], the cost per iteration amounts to $10sN$ *flops*. Consequently, the corresponding computational cost is given by $2\min\{s, 5\}sN$ *flops*;
- the evaluation of Δ^ℓ in (5.95). This requires the solution of $2s$ linear systems with the symmetric matrix M_N plus $2sN$ *flops*. Concerning the matrix M_N, an additional saving of computational effort is gained by retaining only its tridiagonal part (or by considering an approximate inverse).[8] In such a case, after its factorization,[9] one has a cost of less than $10sN$ *flops*. The total cost is then less than $12sN$ *flops*.

In conclusion, the total cost per iteration amounts to k function evaluation plus $(13s + 3k + 2\min\{s, 5\}s + 4ks)N$ *flops*.

It is worth mentioning that the same complexity is obtained in the case of Dirichlet or Neumann boundary conditions, by considering the corresponding tridiagonal matrices (5.40) and (5.55), respectively. Differently, when the Fourier-Galerkin spatial semi-discretization is considered, one obtains formally the same iteration (5.94)-(5.95) but with matrix M_N given by:

$$M_N = I_{2N+1} + (\alpha h \zeta)^2 D_N \in \mathbb{R}^{(2N+1)\times(2N+1)}, \tag{5.97}$$

where the matrix D_N is *diagonal* (see (5.78)). Consequently, also M_N is a diagonal matrix and, therefore, the complexity per iteration, besides the function evaluations of the second equation in

[7] This costs less than $3s$ *flops*

[8] In general, the matrix becomes *banded*, when considering higher-order discretizations, see, e.g., (5.64).

[9] This costs less than $3N$ *flops*.

(5.80) (which are the same as before, i.e. k), decreases. As a matter of fact, the required *flops* per iteration are now $(5s + 3k + 2\min\{s, 5\}s + 4ks)N$.

As a result of the previous arguments, one then expects a complexity per step which is *linear* in the dimension of the problem and, therefore, comparable with that of an explicit method. Moreover, unlike the A-stable HBVM(k, s) methods, explicit methods may suffer from stepsize restrictions due to stability reasons, as we shall see in the numerical tests.

5.7 Numerical tests

Even though the use of energy-conserving methods is quite well understood, proving to be very useful, when speaking about Hamiltonian ordinary differential equations, their use in the framework of Hamiltonian partial differential equation is fairly less obvious. Nevertheless, we report here a few numerical tests which should highlight the usefulness of using energy-conserving methods for solving Hamiltonian PDEs [12, 13, 14].

For this purpose, let us consider the well-known *sine-Gordon* equation, which is in the form (5.1) with $f(u(x,t)) = 1 - \cos(u(x,t))$:

$$u_{tt}(x,t) = u_{xx}(x,t) - \sin(u(x,t)), \qquad x \in [-20, 20], \quad t \geq 0. \tag{5.98}$$

When (5.98) is coupled with the initial conditions,

$$u(x,0) \equiv 0, \qquad u_t(x,0) = \frac{4}{\gamma}\,\mathrm{sech}\left(\frac{x}{\gamma}\right), \qquad \gamma > 0, \tag{5.99}$$

it admits *soliton-like* solutions, as described in [77]. In more details, depending on the value of the positive parameter γ, the solution is known to be given by:

$$u(x,t) = 4\,\mathrm{atan}\left[\vartheta(t;\gamma)\,\mathrm{sech}\left(\frac{x}{\gamma}\right)\right], \tag{5.100}$$

with

$$\vartheta(t;\gamma) = \begin{cases} \frac{1}{\sqrt{\gamma^2-1}}\sin\left(\frac{\sqrt{\gamma^2-1}}{\gamma}t\right), & \text{if} \quad \gamma > 1, \\ t, & \text{if} \quad \gamma = 1, \\ \frac{1}{\sqrt{1-\gamma^2}}\sinh\left(\frac{\sqrt{1-\gamma^2}}{\gamma}t\right), & \text{if} \quad 0 < \gamma < 1. \end{cases} \tag{5.101}$$

The three cases are shown in Figures 5.1–5.3, respectively: the first soliton, shown in Figure 5.1 and obtained for $\gamma > 1$, is named *breather*, whereas the third one, shown in Figure 5.3 and obtained for $0 < \gamma < 1$, is named *kink-antikink*. Clearly, the case $\gamma = 1$, named *double-pole soliton* and shown in Figure 5.2, separates the two different types of dynamics.

Moreover, having fixed the space interval,[10] the Hamiltonian is a decreasing function of γ, as is shown in Figure 5.4. This means that the value of the Hamiltonian (which is a constant of motion) characterizes the dynamics. Consequently, in a neighbourhood of $\gamma = 1$, corresponding to a value $\simeq 16$ for the Hamiltonian, nearby values of the Hamiltonian will provide different types of soliton solutions. Consequently, energy conserving methods are expected to be useful, when numerically solving problem (5.98)-(5.99) with $\gamma = 1$.

Let us then consider problem (5.98)-(5.99), at first with periodic boundary conditions, by using:

[10] I.e., $[-20, 20]$, in our case (see (5.98)).

- a finite-difference approximation with $N = 400$ equispaced mesh points, so that $|\mathcal{H}[\mathbf{z}](0) - H(\mathbf{q}(0), \mathbf{p}(0))| \simeq 1.4 \cdot 10^{-14}$, that is, the value of the Hamiltonian (5.15) is practically matched by the discrete Hamiltonian (5.19);
- a trigonometric polynomial approximation of degree $N = 100$ and $m = 200$ equispaced mesh points.[11] In so doing, the error (5.84) in the initial conditions is $e_N \simeq 1.6 \cdot 10^{-11}$, so that they are quite well matched.

For the time integration, let us consider the following two different second-order methods, used with stepsize $h = 0.5$, for 200 integration steps:

- the symplectic implicit mid-point rule, i.e. HBVM(1,1), for which the Hamiltonian error is $\simeq 4.5 \cdot 10^{-1}$ (though without a drift) both for the finite-difference and the trigonometric polynomial spatial approximations;
- the (practically) energy conserving HBVM(7,1) method, for which the Hamiltonian error is $\simeq 5.7 \cdot 10^{-14}$ when the finite-difference spatial discretization is used and $\simeq 1.9 \cdot 10^{-14}$ when the Fourier approach in space is considered.

As we have sketched in the previous section, in order to efficiently implement these two methods, we here use the *blended implementation* (5.94)-(5.95), by using only the linear part of the equation, so that the approximate Jacobian turns out to be constant and *tridiagonal*, when the finite-difference discretization is used, or *diagonal*, when the Fourier-Galerkin approach is used. In such a case, one only needs to solve linear systems in the form $I_N + (h\zeta/\Delta x)^2 T_N$ or $I_{2N+1} + h^2\zeta^2 D_N$, when a finite-difference or a Fourier spatial discretization, respectively, is used (see (5.96)-(5.97)), where h is the time step and ζ is the parameter given in Table 4.1, depending on the HBVM(k, s) method and independent of the value of k. Since in the former case the matrix is tridiagonal, and diagonal in the latter case (and, moreover, they are constant for all integration steps), the method is computationally inexpensive.

Concerning the finite-difference approach, the error in the numerical Hamiltonian is plotted in Figure 5.5 (top). In Figures 5.6 and 5.7 we plot the numerical approximations to the solution computed by the HBVM(1,1) and HBVM(7,1) methods, respectively. As is clear, the former approximation is wrong, since the method has provided a *breather*-like solution, whereas the latter one has the correct shape (compare with Figure 5.2), thus confirming that energy conservation is an important issue, for such a problem.

Concerning the trigonometric polynomial approximation, the error in the numerical Hamiltonian is plotted in Figure 5.5 (bottom). In Figures 5.8 and 5.9 we plot the numerical approximations to the solution computed by the HBVM(1,1) and HBVM(7,1) methods, respectively. As is clear, also in this case, the dynamics returned by the implicit mid-point rule is wrong, since the method has provided a *breather*-like solution, whereas the approximation obtained by using the (practically) energy-conserving HBVM(7,1) method has the correct shape (compare with Figure 5.2), thus confirming, once more, that energy conservation is an important issue, for such a problem, also when the spatial discretization is done by means of a Fourier spectral method.

Completely similar results are obtained by using the same methods (and with the same spatial grid and temporal stepsize h), when Dirichlet boundary conditions are used:

- in Figure 5.10 there are the plots of the differences $H(\mathbf{q}_n, \mathbf{p}_n, t_n) - H(\mathbf{q}_0, \mathbf{p}_0, 0)$ (see (5.39)) and $\tilde{H}(\mathbf{q}_n, \mathbf{p}_n, \tilde{q}_n, \tilde{p}_n) - \tilde{H}(\mathbf{q}_0, \mathbf{p}_0, 0, 0)$ (see (5.45) and (5.47)), when using the HBVM(1,1) method. It is clear that both of them are large and, as a result, the computed numerical solution, shown in Figure 5.11, is wrong;

[11] In fact, $m = 200$ is an appropriate choice for $N = 100$, in this case.

- in Figure 5.12 are presented the plots of $H(\mathbf{q}_n, \mathbf{p}_n, t_n) - H(\mathbf{q}_0, \mathbf{p}_0, 0)$ (see (5.39)) and of $\tilde{H}(\mathbf{q}_n, \mathbf{p}_n, \tilde{q}_n, \tilde{p}_n) - \tilde{H}(\mathbf{q}_0, \mathbf{p}_0, 0, 0)$ (see (5.45) and (5.47)), when using the HBVM(7,1) method. It is clear that now the augmented Hamiltonian (5.45) is (practically) conserved (maximum error $\simeq 4.4 \cdot 10^{-14}$), whereas the original Hamiltonian (5.39) oscillates around its initial value. The computed solution, shown in Figure 5.13, is now correct.

Analogous results are obtained when Neumann boundary conditions are prescribed for the problem. In fact, by considering the same methods with the same spatial grid and the same temporal stepsize h:

- in Figure 5.14 there are the plots of the differences $H(\mathbf{q}_n, \mathbf{p}_n, t_n) - H(\mathbf{q}_0, \mathbf{p}_0, 0)$ (see (5.54)) and $\tilde{H}(\mathbf{q}_n, \mathbf{p}_n, \tilde{q}_n, \tilde{p}_n) - \tilde{H}(\mathbf{q}_0, \mathbf{p}_0, 0, 0)$ (see (5.61) and (5.47)), when using the HBVM(1,1) method. It is clear that both of them are large and, as a result, the computed numerical solution, shown in Figure 5.15, is wrong;

- in Figure 5.16 are presented the plots of $H(\mathbf{q}_n, \mathbf{p}_n, t_n) - H(\mathbf{q}_0, \mathbf{p}_0, 0)$ (see (5.54)) and of $\tilde{H}(\mathbf{q}_n, \mathbf{p}_n, \tilde{q}_n, \tilde{p}_n) - \tilde{H}(\mathbf{q}_0, \mathbf{p}_0, 0, 0)$ (see (5.61) and (5.47)), when using the HBVM(7,1) method. It is clear that now the augmented Hamiltonian (5.61) is (practically) conserved (maximum error $\simeq 6.4 \cdot 10^{-11}$), whereas the original Hamiltonian (5.54) oscillates around its initial value. The computed solution, shown in Figure 5.17, is now correct.

We now highlight the potentialities of the Fourier-Galerkin spatial approximation, with respect to the finite-difference one, when periodic boundary conditions are specified for the problem: in fact, the Fourier approximation (5.82) to the Hamiltonian converges more than exponentially in the number N of Fourier modes, whereas the finite-difference approximation (5.19) converges only quadratically in Δx. Since also HBVM(7,1) is second order, we then compare the use of such method, with stepsize $h = 40/\ell$ in time and for a total of ℓ time-steps, for solving problem (5.98)-(5.99), with $\gamma = 1$ and periodic boundary conditions, by using:

- the second-order finite-difference spatial discretization with ℓ mesh points (with this choice, one has $\Delta x = h$);

- the Fourier-Galerkin approximation with $N = 100$, and $m = 200$ spatial grid-points, which we maintain fixed independently of the choice of ℓ. This because the obtained spatial approximation yields a far more accurate approximation than the one corresponding to the time discretization.

The following table summarizes the obtained results, from which one obtains that both methods are globally second-order accurate, even though the values of N and m are kept fixed in the second case, thus confirming the exponential convergence of the Fourier approximation. Moreover, by comparing the maximum error on the solution in the finite-difference case (FD-error) and in the Fourier-Galerkin approach (FG-error), one sees that the latter is much more favourable than the former. Consequently, one may conclude that the better the approximation to the continuous Hamiltonian,

ℓ	FD-error	rate	FG-error	rate
400	1.4486e-01	–	1.7883e-03	–
800	3.6900e-02	1.97	4.4985e-04	1.99
1600	9.2702e-03	1.99	1.1262e-04	2.00
3200	2.3204e-03	2.00	2.8171e-05	2.00

the better the approximation to the solution.

Before concluding this section, we perform a further numerical text, where we compare some (practically) energy-conserving HBVMs, with well known explicit methods of the same order for solving problem (5.98)-(5.99), with $\gamma = 1$ and periodic boundary conditions, on the time interval $[0, 100]$. In more details, we compare the following methods:

order 2: the (practically) energy-conserving HBVM(5,1) method and the symplectic Störmer-Verlet method (SV2);

order 4: the (practically) energy-conserving HBVM(6,2) method and the composition method (SV4) based on the symplectic Störmer-Verlet method (each step requiring 3 steps of the basic method), according to [49, page 44];

order 6: the (practically) energy-conserving HBVM(9,3) method and the composition method (SV6) based on the symplectic Störmer-Verlet method (each step requiring 9 steps of the basic method), according to [49, page 44].

To compare the methods, we construct a corresponding *Work-Precision Diagram*, by following the standard used in the *Test Set for IVP Solvers* [78]. In more details, we plot the accuracy, measured in terms of the maximum absolute error, w.r.t. the execution time. All tests have been done by using Matlab v.2014b, running on a dual core i7 at 2.8GHz computer with 8GB of central memory. The curve of each method is obtained by using k (logarithmically) equispaced steps between h_{min} and h_{max}, as specified in Table 5.1.[12] When the stepsize used does not exactly divide the final time $T = 100$, the nearest mesh-point is considered.

Table 5.1: Parameters used for constructing Figures 5.18 and 5.19.

Method	$h_{\max}$	$h_{\min}$	k
HBVM(5,1)	0.5	0.003	10
HBVM(6,2)	0.5	0.1	4
HBVM(9,3)	1	0.25	4
SV2	0.1	0.0006	13
SV4	0.1	0.007	7
SV6	0.1	0.01	5

Figure 5.18 summarizes the obtained results, and one sees that the (practically) energy-conserving HBVMs are competitive, even w.r.t. explicit solvers of the same order. For sake of completeness, in Figure 5.19, we plot the corresponding Hamiltonian error versus the execution time, thus confirming that HBVMs are practically energy conserving also for non polynomial Hamiltonians: in fact, taking aside the coarser time steps, all methods have a Hamiltonian error which is within roundoff errors. On the contrary, for the other methods the decrease of the Hamiltonian error matches their order.

[12] Larger values of $h_{\max}$ for the explicit methods (see Table 5.1) are not allowed because of stability reasons.

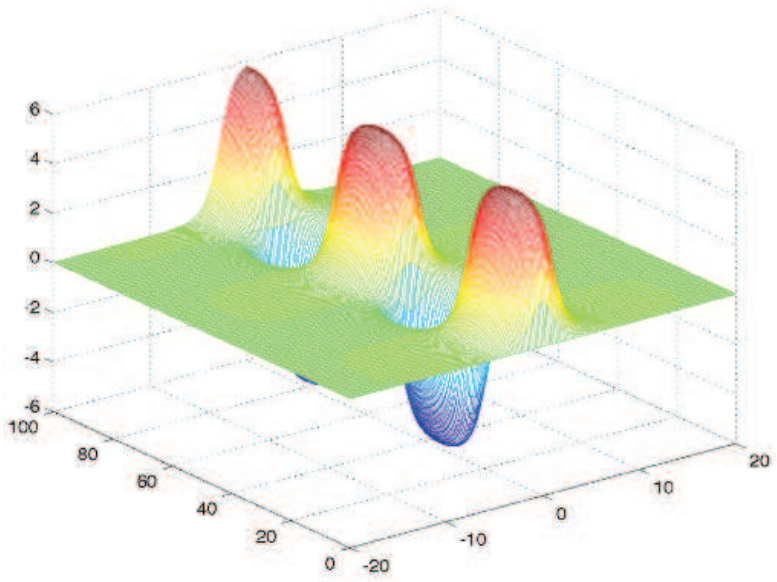

Figure 5.1: *Breather*: soliton-like solution (5.100)-(5.101) of problem (5.98)-(5.99) with $\gamma = 1.01$.

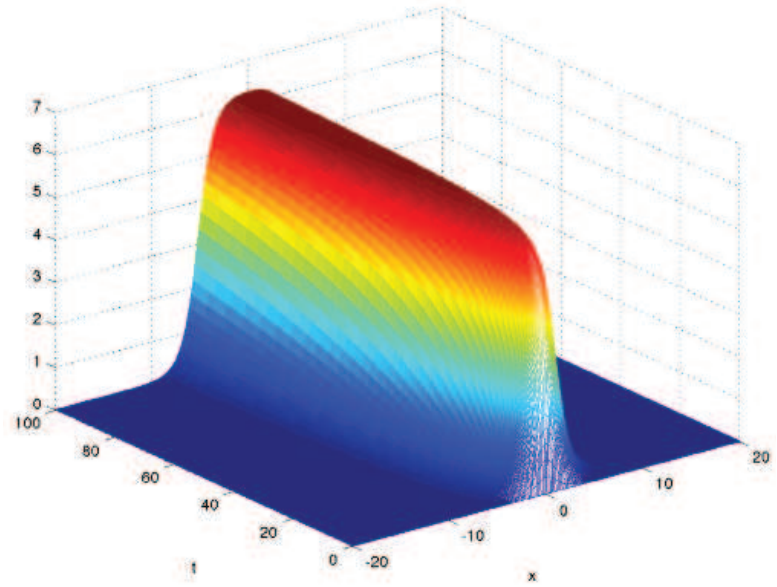

Figure 5.2: *Double-pole*: soliton-like solution (5.100)-(5.101) of problem (5.98)-(5.99) with $\gamma = 1$.

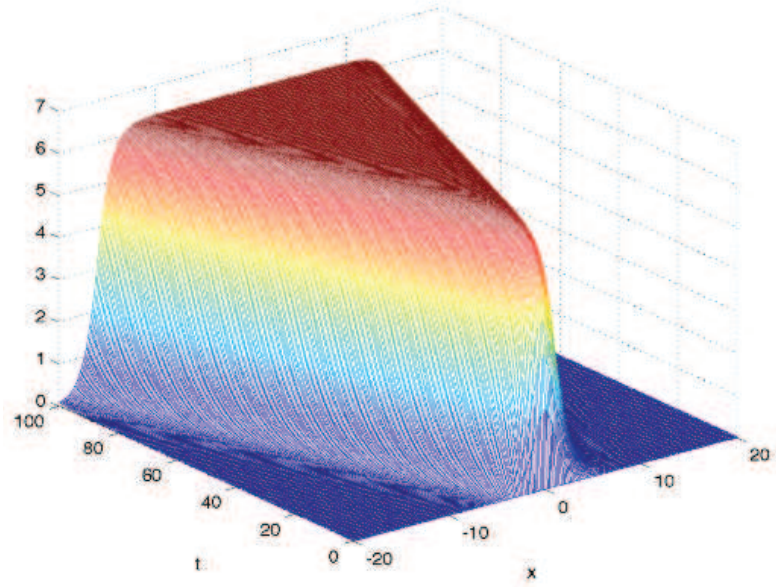

Figure 5.3: *Kink-antikink*: soliton-like solution (5.100)-(5.101) of problem (5.98)-(5.99) with $\gamma = 0.99$.

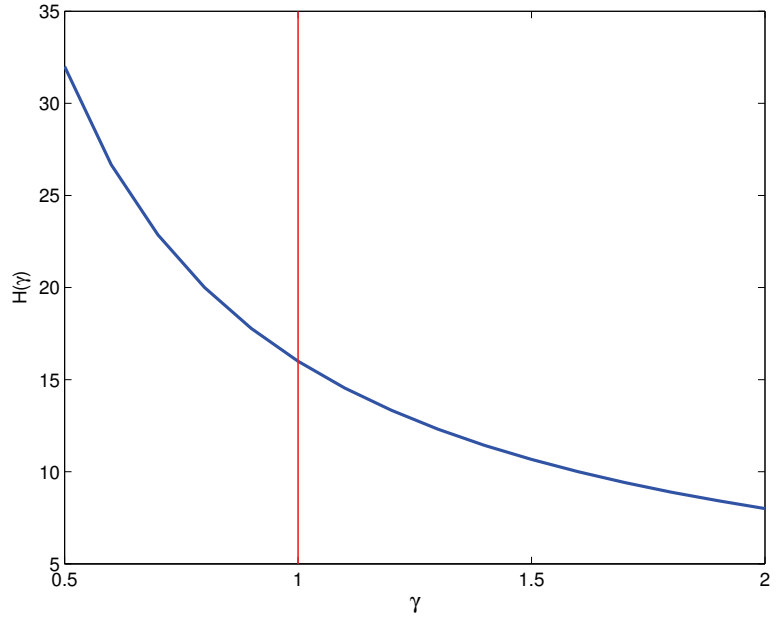

Figure 5.4: Hamiltonian for problem (5.98)-(5.99), as function of γ.

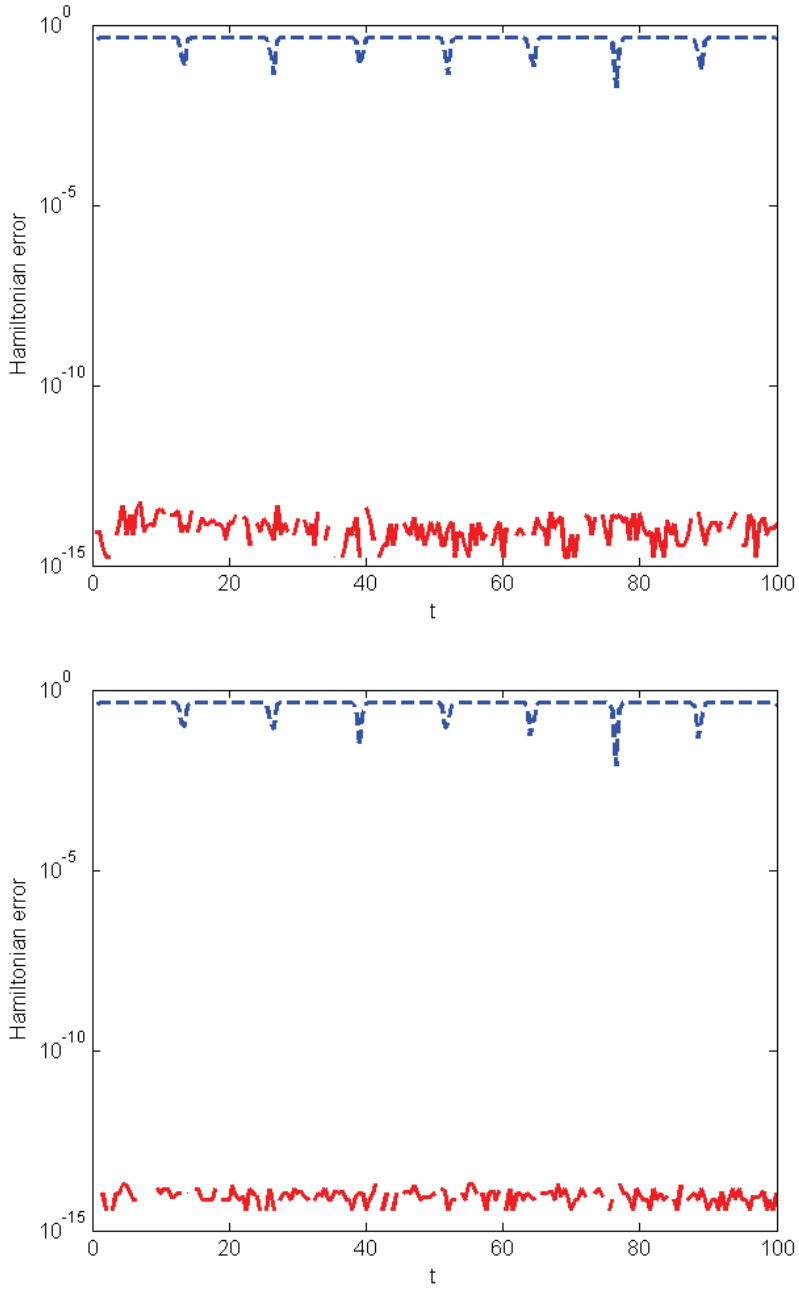

Figure 5.5: Hamiltonian error when solving problem (5.98)-(5.99) with $\gamma = 1$ and periodic boundary conditions, by using a finite-difference ($N = 400$) (top) or a Fourier-Galerkin ($N = 100$) (bottom) spatial discretization and the HBVM(1,1) (dashed blue lines) or the HBVM(7,1) (solid red lines) methods with stepsize $h = 0.5$.

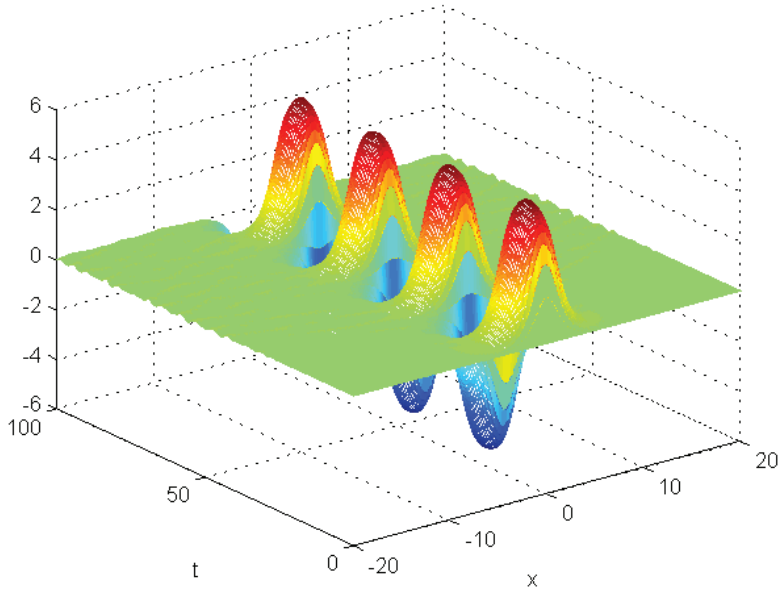

Figure 5.6: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and periodic boundary conditions, computed by the HBVM(1,1) method with stepsize $h = 0.5$ and a finite-difference spatial discretization ($N = 400$).

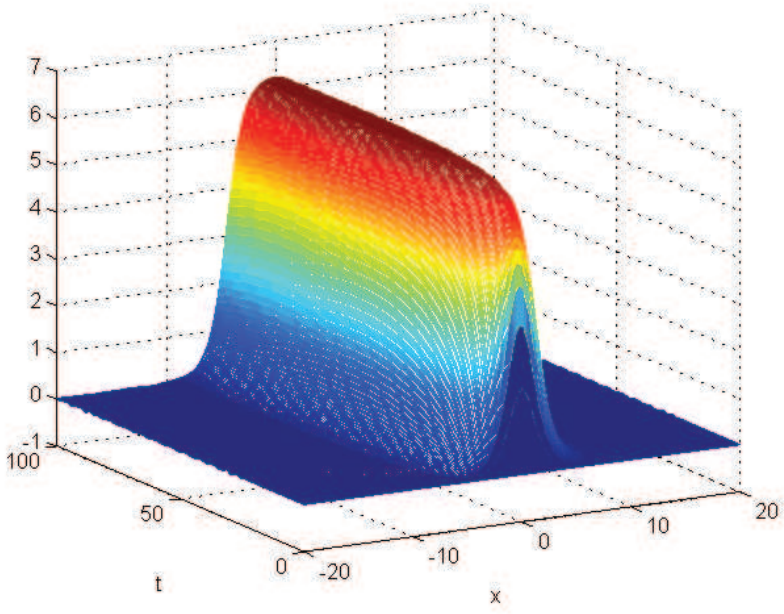

Figure 5.7: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and periodic boundary conditions, computed by the HBVM(7,1) method with stepsize $h = 0.5$ and a finite-difference spatial discretization ($N = 400$).

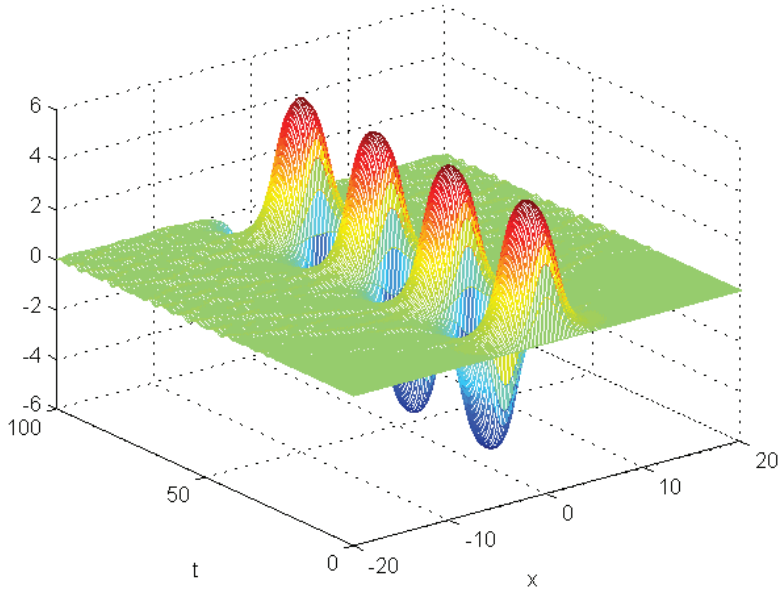

Figure 5.8: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and periodic boundary conditions, computed by the HBVM(1,1) method with stepsize $h = 0.5$ and a Fourier-Galerkin spatial discretization ($N = 100$).

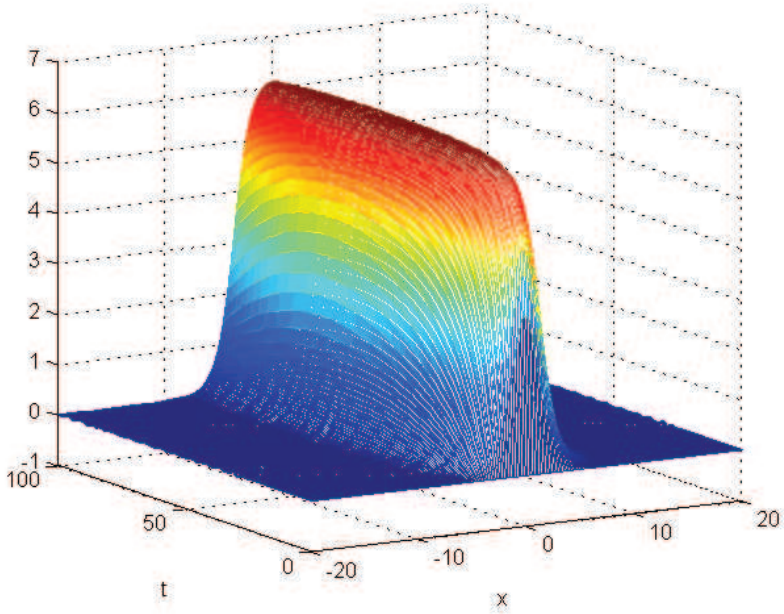

Figure 5.9: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and periodic boundary conditions, computed by the HBVM(7,1) method with stepsize $h = 0.5$ and a Fourier-Galerkin spatial discretization ($N = 100$).

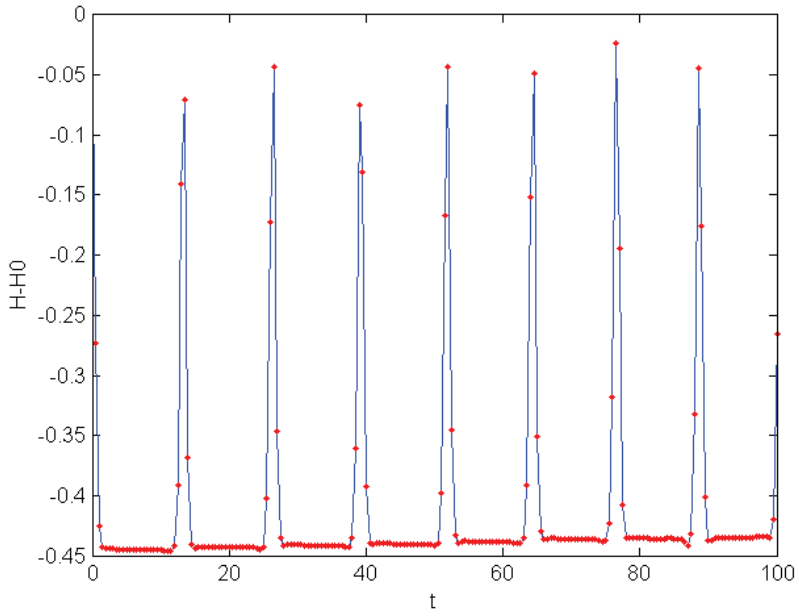

Figure 5.10: Difference with their initial value for the numerical Hamiltonian (solid blue line) and for the numerical augmented Hamiltonian (red dots) when solving problem (5.98)-(5.99) with $\gamma = 1$ and Dirichlet boundary conditions, by using a finite-difference spatial discretization ($N = 400$) and the HBVM(1,1) with stepsize $h = 0.5$.

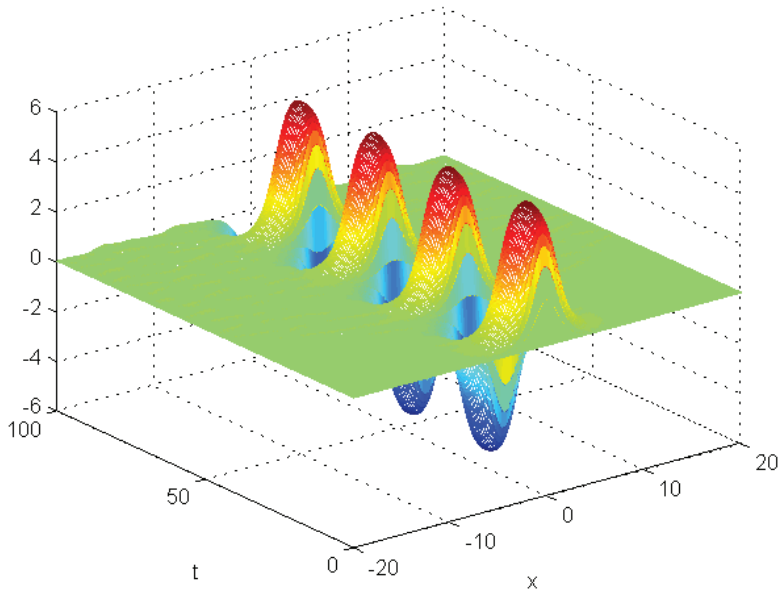

Figure 5.11: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and Dirichlet boundary conditions, computed by the HBVM(1,1) method with stepsize $h = 0.5$ and a finite-difference spatial discretization ($N = 400$).

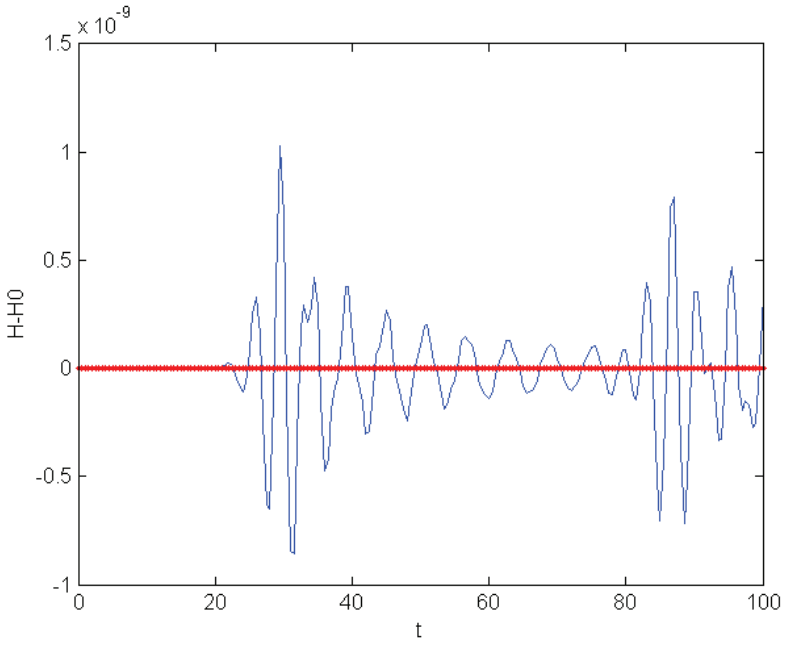

Figure 5.12: Difference with their initial value for the numerical Hamiltonian (solid blue line) and for the numerical augmented Hamiltonian (red dots) when solving problem (5.98)-(5.99) with $\gamma = 1$ and Dirichlet boundary conditions, by using a finite-difference spatial discretization ($N = 400$) and the HBVM(7,1) with stepsize $h = 0.5$.

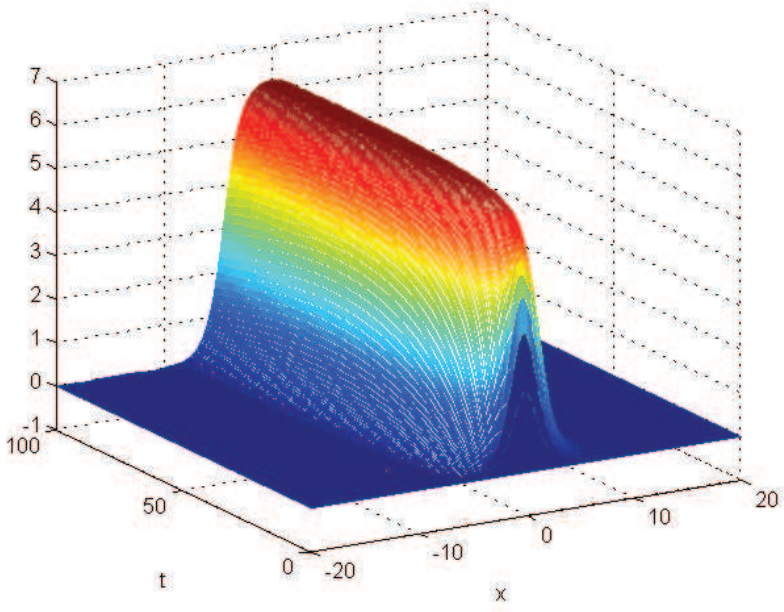

Figure 5.13: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and Dirichlet boundary conditions, computed by the HBVM(7,1) method with stepsize $h = 0.5$ and a finite-difference spatial discretization ($N = 400$).

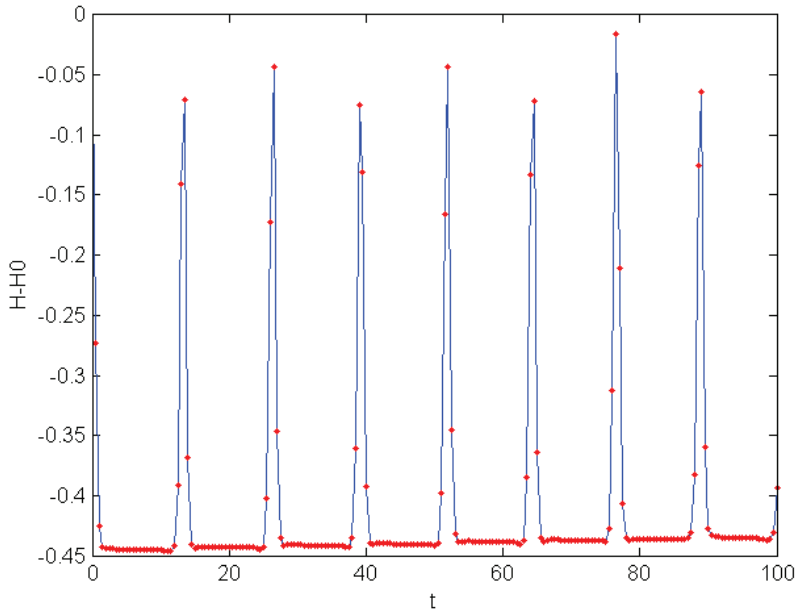

Figure 5.14: Difference with their initial value for the numerical Hamiltonian (solid blue line) and for the numerical augmented Hamiltonian (red dots) when solving problem (5.98)-(5.99) with $\gamma = 1$ and Neumann boundary conditions, by using a finite-difference spatial discretization ($N = 400$) and the HBVM(1,1) with stepsize $h = 0.5$.

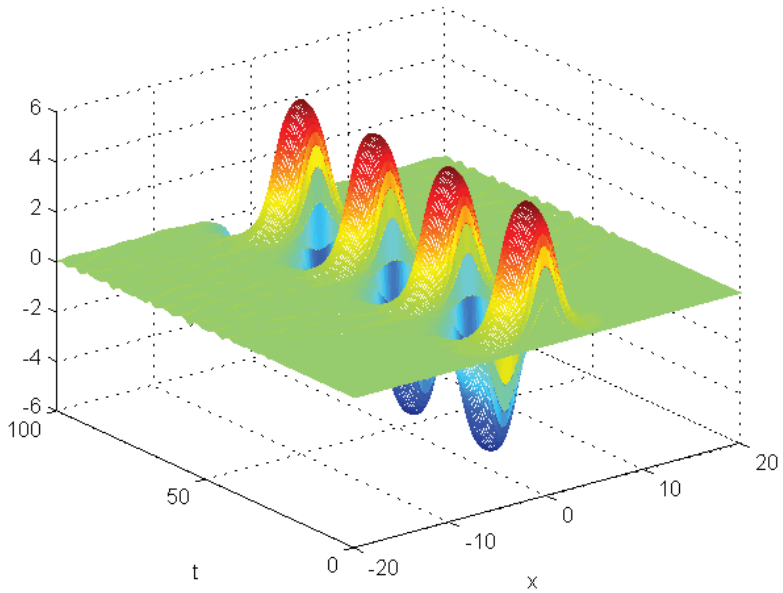

Figure 5.15: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and Neumann boundary conditions, computed by the HBVM(1,1) method with stepsize $h = 0.5$ and a finite-difference spatial discretization ($N = 400$).

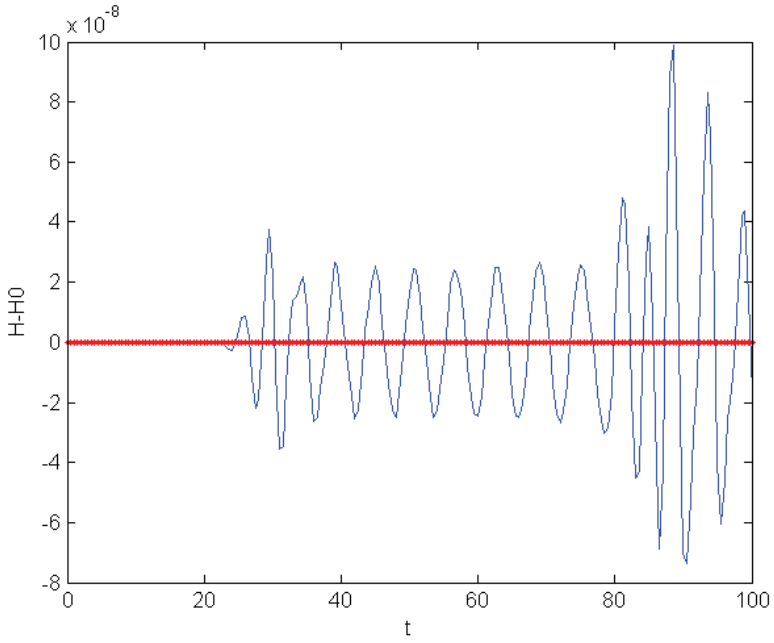

Figure 5.16: Difference with their initial value for the numerical Hamiltonian (solid blue line) and for the numerical augmented Hamiltonian (red dots) when solving problem (5.98)-(5.99) with $\gamma = 1$ and Neumann boundary conditions, by using a finite-difference spatial discretization ($N = 400$) and the HBVM(7,1) with stepsize $h = 0.5$.

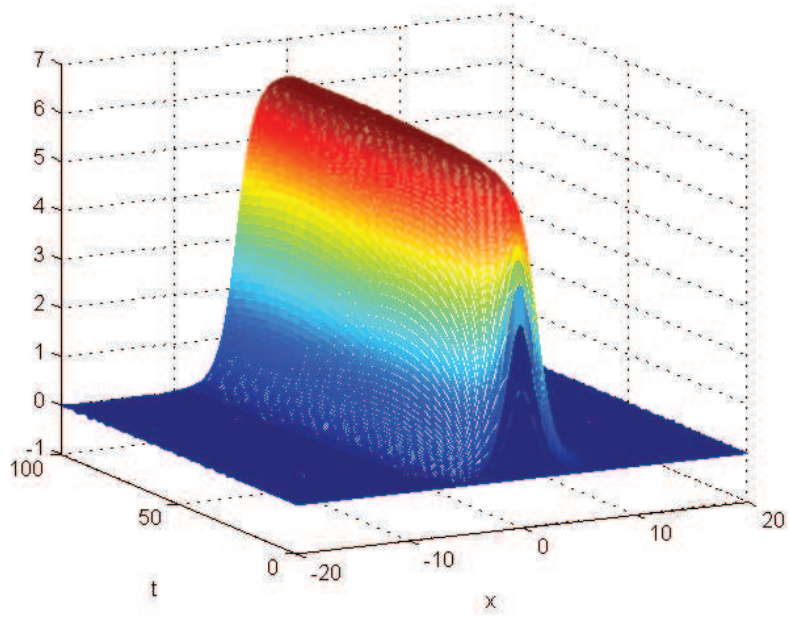

Figure 5.17: Numerical solution of problem (5.98)-(5.99) with $\gamma = 1$ and Neumann boundary conditions, computed by the HBVM(7,1) method with stepsize $h = 0.5$ and a finite-difference spatial discretization ($N = 400$).

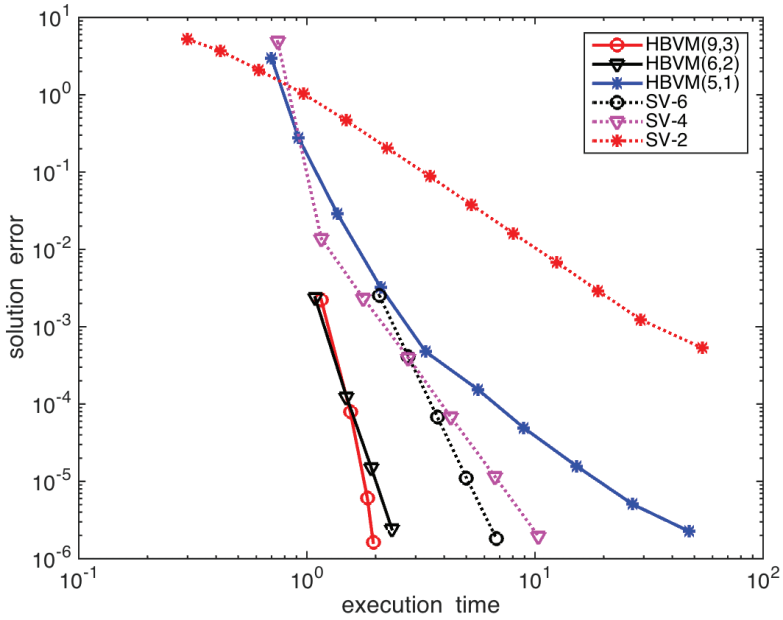

Figure 5.18: *Work-Precision Diagram* for problem (5.98)–(5.99).

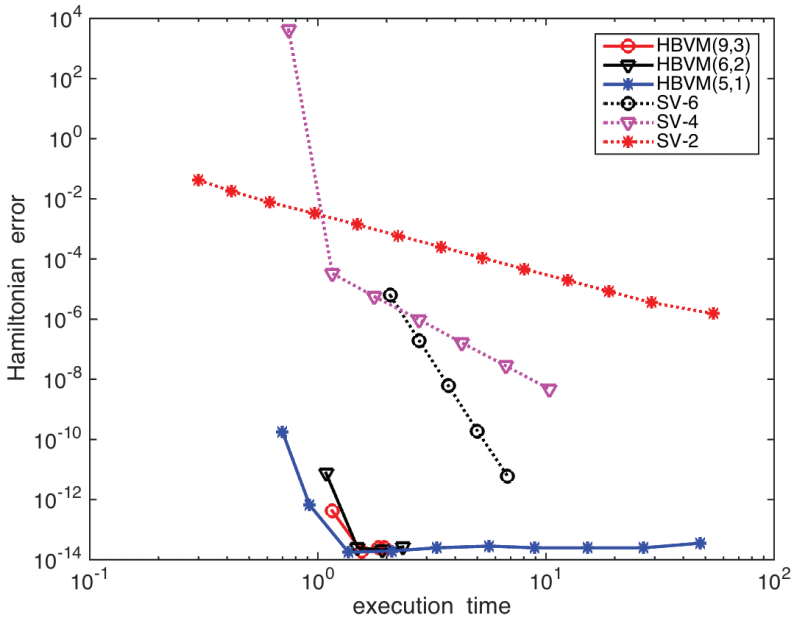

Figure 5.19: Hamiltonian error versus execution time for problem (5.98)–(5.99)

.

Conclusions

In this thesis we have provided a detailed description of the low-rank Runge-Kutta family of Hamiltonian Boundary Value Methods (HBVMs) for the numerical solution of Hamiltonian problems. In particular, we have studied in detail their main property: the conservation of polynomial Hamiltonians, which results into a *practical* conservation for generic suitably regular Hamiltonians. This property turns out to play a fundamental role in some problems where the error on the Hamiltonian, usually obtained even when using a symplectic method, would be not negligible to the point of affecting the dynamics of the numerical solution.

The research developed in this thesis has addressed two main topics. The first one is a new procedure, based on a particular splitting of the matrix defining the method, which turns out to be more effective of the well-known *blended-implementation*, as well as of a classical fixed-point iteration when the problem at hand is *stiff*. This procedure has been applied also to second order problems with separable Hamiltonian function, resulting in a cheaper computational cost.

The second topic addressed is the application of HBVMs for the full discretization of a method of lines approach to numerically solve Hamiltonian PDEs. In particular, we have considered the semilinear wave equation coupled with either periodic, Dirichlet or Neumann boundary conditions, and the application of a (practically) energy conserving HBVM method to the semi-discrete problem obtained by means of a second order finite-difference approximation in space. When the problem is coupled with periodic boundary conditions we have also considered the case of higher-order finite-difference spatial discretizations and the case when a Fourier-Galerkin method is used for the spatial semi-discretization. The proposed methods are able to provide a numerical solution such that the energy (which can be conserved or not, depending on the assigned boundary conditions) *practically* satisfies its prescribed variation in time. A few numerical tests for the *sine-Gordon* equation have given evidence that, for some problems, there is an effective advantage in using an energy-conserving method for the time integration, with respect to the use of a symplectic one. Moreover, even though HBVMs are implicit method, their computational cost for the considered problem turns out to be competitive even with respect to that of explicit solvers of the same order, which, furthermore, may suffer from stepsize restrictions due to stability reasons, whereas HBVMs are A-stable.

Bibliography

[1] P. Amodio, L. Brugnano. A Note on the Efficient Implementation of Implicit Methods for ODEs. *J. Comput. Appl. Math.* **87** (1997) 1–9.

[2] P. Amodio, I. Sgura. High-order finite difference schemes for the solution of second-order BVPs. *Jour. Comput. Appl. Math.* **176** (2005) 59–76.

[3] G. Benettin, A. Giorgilli. On the Hamiltonian interpolation of near to the identity symplectic mappings with application to symplectic integration algorithms. *J. Statist. Phys.* **74** (1994) 1117–1143.

[4] P. Betsch, P. Steinmann. Inherently Energy Conserving Time Finite Elements for Classical Mechanics. *Journal of Computational Physics* **160** (2000) 88–116.

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

[6] J.P. Boyd. *Chebyshev and Fourier spectral methods. Second edition.* Dover Publications, Inc., Mineola, NY, 2001.

[7] T.J. Bridges. Multisymplectic structures and wave propagation. *Math. Proc. Cambridge Philos. Soc.* **121** (1997) 147–190.

[8] T.J. Bridges, S. Reich. Multi-symplectic integrators: numerical schemes for Hamiltonian PDEs that conserve symplecticity. *Physics Letters A* **284** (2001) 184–193.

[9] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Efficient implementation of Gauss collocation and Hamiltonian Boundary Value Methods. *Numer. Algor.* **65**, no. 3 (2014) 633–650

[10] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Efficient implementation of geometric integrators for separable Hamiltonian problems. *AIP Conference Proceedings* **1558**, 734–737 (2013).

[11] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Hamiltonian Boundary Value Methods (HVBMs) and their efficient implementation. *Mathematics in Engineering, Science and Aerospace* **5**,4 (2014) 343–411.

[12] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Energy conservation issues in the numerical solution of the semilinear wave equation. `arXiv:1410.7009`.

[13] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Energy conservation issues in the numerical solution of Hamiltonian PDEs. *AIP Conference Proceedings* **1648**, 020002 (2015).

[14] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Recent advances in the numerical solution of Hamiltonian PDEs. *AIP Conference Proceedings* **1648**, 150008 (2015).

[15] L. Brugnano, F. Iavernaro. Recent Advances in the Numerical Solution of Conservative Problems. *AIP Conference Proc.* **1493** (2012) 175–182.

[16] L. Brugnano, F. Iavernaro. Geometric Integration by Playing with Matrices. *AIP Conference Proceedings* **1479** (2012) 16–19.

[17] L. Brugnano, F. Iavernaro, C. Magherini. Efficient implementation of Radau collocation methods. *Applied Numerical Mathematics* **87** (2015) 100–113

[18] L. Brugnano, F. Iavernaro, T. Susca. Numerical comparisons between Gauss-Legendre methods and Hamiltonian BVMs defined over Gauss points. *Monografias de la Real Acedemia de Ciencias de Zaragoza* **33** (2010) 95–112.

[19] 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 dynamical systems. *Communications in Nonlinear Science and Numerical Simulation* **20**, no.3 (2014) 650–667.

[20] 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.

[21] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian Boundary Value Methods (Energy Preserving Discrete Line Methods). *Journal of Numerical Analysis, Industrial and Applied Mathematics* **5**,1-2 (2010) 17–37.

[22] L. Brugnano, F. Iavernaro, D. Trigiante. Numerical Solution of ODEs and the Columbus' Egg: Three Simple Ideas for Three Difficult Problems. *Mathematics in Engineering, Science and Aerospace* **1**,4 (2010) 407–426.

[23] L. Brugnano, F. Iavernaro, D. Trigiante. Isospectral property of Hamiltonian boundary value methods (HBVMs) and their blended implementation. Preprint, 2010. `arXiv:1002.1387`.

[24] L. Brugnano, F. Iavernaro, D. Trigiante. A note on the efficient implementation of Hamiltonian BVMs. *Journal of Computational and Applied Mathematics* **236** (2011) 375–383.

[25] L. Brugnano, F. Iavernaro, D. Trigiante. The Lack of Continuity and the Role of Infinite and Infinitesimal in Numerical Methods for ODEs: the Case of Symplecticity. *Applied Mathematics and Computation* **218** (2012) 8053–8063.

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

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

[28] L. Brugnano, C. Magherini. The BiM Code for the Numerical Solution of ODEs. *Jour. Comput. Appl. Mathematics* **164-165** (2004) 145–158.

[29] L. Brugnano, C. Magherini. Some Linear Algebra Issues Concerning the Implementation of Blended Implicit Methods. *Numer. Lin. Alg. Appl.* **12** (2005) 305–314.

[30] L. Brugnano, C. Magherini. Recent Advances in Linear Analysis of Convergence for Splittings for Solving ODE problems. *Applied Numerical Mathematics* **59** (2009) 542–557.

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

[32] L. Brugnano, D. Trigiante. Solving Differential Problems by Multistep Initial and Boundary Value Methods. *Gordon and Breach Science Publ.*, Amsterdam, 1998.

[33] J.C. Butcher. Implicit Runge-Kutta processes. *Math. Comp.* **18** (1964) 50–64.

[34] B. Cano. Conserved quantities of some Hamiltonian wave equations after full discretization. *Numer. Math.* **103** (2006) 197–223

[35] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, New York, 1988.

[36] E. Celledoni, R.I. McLachlan, D. Mc Laren, B. Owren, G.R.W. Quispel, W.M. Wright. Energy preserving Runge–Kutta methods. *M2AN* **43** (2009) 645–649.

[37] E. Celledoni, R.I. McLachlan, B. Owren, G.R.W. Quispel. Energy-Preserving Integrators and the Structure of B-series. *Found. Comput. Math.* **10** (2010) 673–693.

[38] G. Dahlquist, Å. Björk. *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

[39] G. Dahlquist, Å. Björk. *Numerical Methods in Scientific Computing, Vol. 1.* SIAM, Philadelphia, 2008.

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

[41] K. Feng. On Difference Schemes and Symplectic Geometry. In *Proceedings of the 1984 Beijing symposium on differential geometry and differential equations*. Science Press, Beijing, 1985, pp. 42–58.

[42] B. Fornberg, G.B. Whitham. A Numerical and Theoretical Study of Certain Nonlinear Wave Phenomena. *Proc. R. Soc. Lond. A* **289** (1978) 373–403.

[43] J. Frank. Conservation of wave action under multisymplectic discretizations. *J. Phys. A: Math. Gen.* **39** (2006) 5479–5493.

[44] J. Frank, B.E. Moore, S. Reich. Linear PDEs and Numerical Methods that Preserve a Multisymplectic Conservation Law. *SIAM J. Sci. Comput.* **28** (2006) 260–277.

[45] H. Goldstein, C.P. Poole, J.L. Safko. *Classical Mechanics*. Addison Wesley, 2001.

[46] O. Gonzales. Time integration and discrete Hamiltonian systems. *J. Nonlinear Sci.* **6** (1996) 449–467.

[47] W. Gröbner. *Gruppi, Anelli e Algebre di Lie*. Collana di Informazione Scientifica "Poliedro", Edizioni Cremonese, Rome, 1975.

[48] E. Hairer. Energy preserving variant of collocation methods. *Journal of Numerical Analysis, Industrial and Applied Mathematics* **5**,1-2 (2010) 73–84.

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

[50] E. Hairer, G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems, 2nd edition.* Springer-Verlag, Berlin, 1996.

[51] E. Hairer, C.J. Zbinden. On conjugate symplecticity of B-series integrators. *IMA J. Numer. Anal.* (2012) 1–23.

[52] P.J. van der Houwen, J.J.B. de Swart. Triangularly implicit iteration methods for ODE-IVP solvers. *SIAM J. Sci. Comput.* **18** (1997) 41–55.

[53] P.J. van der Houwen, J.J.B. de Swart. Parallel linear system solvers for Runge-Kutta methods. *Adv. Comput. Math.* **7**, 1-2 (1997) 157–181.

[54] B.L. Hulme. One-Step Piecewise Polynomial Galerkin Methods for Initial Value Problems. *Mathematics of Computation*, **26**, 118 (1972) 415–426.

[55] F. Iavernaro, B. Pace. *s*-Stage Trapezoidal Methods for the Conservation of Hamiltonian Functions of Polynomial Type. *AIP Conf. Proc.* **936** (2007) 603–606.

[56] F. Iavernaro, B. Pace. Conservative Block-Boundary Value Methods for the Solution of Polynomial Hamiltonian Systems. *AIP Conf. Proc.* **1048** (2008) 888–891.

[57] F. Iavernaro, D. Trigiante. High-order symmetric schemes for the energy conservation of polynomial Hamiltonian problems. *Journal of Numerical Analysis, Industrial and Applied Mathematics* **4**,1-2 (2009) 87–101.

[58] E. Isaacson, H.B. Keller. *Analysis of numerical methods.* J. Wiley & Sons, New York, 1966.

[59] A.L. Islas, C.M. Schober. On the preservation of phase space structure under multisymplectic discretization. *Journal of Computational Physics* bf 197 (no. 2) (2004) 585–609.

[60] A.L. Islas, C.M. Schober. Backward error analysis for multisymplectic discretizations of Hamiltonian PDEs. *Mathematic and Computers in Simulation* **69** (2005) 290–303.

[61] A.L. Islas, C.M. Schober. Conservation properties of multisymplectic integrators. *Future Generation Computer Systems* **22** (2006) 412–422.

[62] A. Kurganov, J. Rauch. The Order of Accuracy of Quadrature Formulae for Periodic Functions. *Advances in Phase Space Analysis of Partial Differential Equations*, A. Bove et al. (eds.), Birkhäuser, Boston, 2009.

[63] J. E. Marsden, G.P. Patrick, S. Shkoller. Multi-symplectic geometry, variational integrators, and nonlinear PDEs. Communications in Mathematical Physics **199** (1999) 351–395.

[64] R.I. Mc Lachlan, G.R.W. Quispel, N. Robidoux. Geometric integration using discrete gradient. *Phil. Trans. R. Soc. Lond. A* **357** (1999) 1021–1045.

[65] Z. Lv, M. Xue, Y. Wang. Legendre polynomials spectral approximation for the infinite-dimensional Hamiltonian systems. Math. Probl. in Engineering (2011) Article ID 824167, 13 pages.

[66] B. Moore, S. Reich. Backward error analysis for multi-symplectic integration methods. *Numer. Math.* **95** (2003) 625–652.

[67] G.R.W. Quispel, D.I. Mc Laren. A new class of energy-preserving numerical integration methods. *J. Phys. A: Math. Theor.* **41** (2008) 045206 (7pp).

[68] J.M. Sanz Serna. Runge-Kutta schemes for Hamiltonian systems. *BIT* **28** (1988) 877–883.

[69] J.M. Sanz Serna, M.P. Calvo. *Numerical Hamiltonian Problems.* Chapman & Hall, London, 1994.

[70] J. Shen. Efficient spectral-Galerkin method I. Direct solvers of second and fourth-order equations using Legendre polynomials. *SIAM Journal on Scientific Computing, 15(6),* 1489-1505

[71] J. Shen. Efficient spectral-Galerkin method II. Direct solvers of second and fourth-order equations using Chebyshev polynomials. *SIAM Journal on Scientific Computing, 16(1),* 74-87.

[72] Y.B. Suris. On the canonicity of mappings that can be generated by methods of Runge–Kutta type for integrating systems $x'' = \partial U/\partial x$. *U.S.S.R. Comput. Math. and Math. Phys.* **29**, 1 (1989) 138–144.

[73] Q. Tang, C.-m. Chen. Continuous finite element methods for Hamiltonian systems. *Applied Mathematics and Mechanics* **28**,8 (2007) 1071–1080.

[74] W. Tang, Y. Sun. Time finite element methods: a unified framework for numerical discretizations of ODEs. *Applied Mathematics and Computation* **219**, 4 (2012) 2158–2179.

[75] S.B. Wineberg, J.F. McGrath, E.F. Gabl, L.R. Scott, C.E. Southwell. Implicit spectral methods for wave propogation problems. *J. Comp. Physics* **97** (1991) 311–336.

[76] E.T. Whittaker, G.N. Watson. *A course in modern analysis, Fourth edition,* Cambridge University Press, 1950.

[77] T.H. Wlodarczyk. *Stability and preservation properties of multisymplectic integrators.* PhD thesis, Department of Mathematics in the College of Sciences at the University of Central Florida, Orlando, Florida, 2007. (`http://etd.fcla.edu/CF/CFE0001817/Wlodarczyk_Tomasz_H_200708_PhD.pdf`)

[78] Test Set for IVP Solvers.
`https://www.dm.uniba.it/~testset/testsetivpsolvers/`