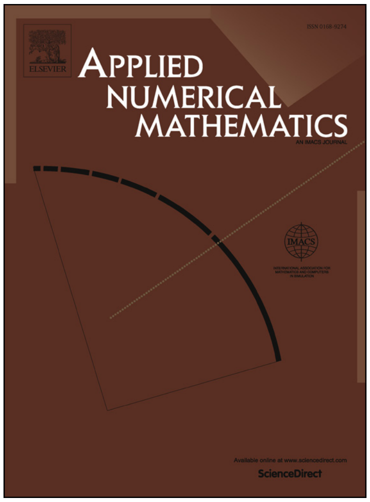
Arbitrary high-order methods for one-sided direct event location in discontinuous differential problems with nonlinear event function



Arbitrary high-order methods for one-sided direct event location in discontinuous differential problems with nonlinear event function

Pierluigi Amodio, Luigi Brugnano and Felice Iavernaro

PII: S0168-9274(22)00106-4
DOI: https://doi.org/10.1016/j.apnum.2022.04.013
Reference: APNUM 4337

To appear in: *Applied Numerical Mathematics*

Received date: 13 March 2022
Revised date: 19 April 2022
Accepted date: 19 April 2022

Arbitrary high-order methods for one-sided direct event location in discontinuous differential problems with nonlinear event function

Pierluigi Amodio[*] Luigi Brugnano[†] Felice Iavernaro[*]

April 21, 2022

Abstract

In this paper we are concerned with numerical methods for the one-sided event location in discontinuous differential problems, whose event function is nonlinear (in particular, of polynomial type). The original problem is transformed into an equivalent Poisson problem, which is effectively solved by suitably adapting a recently devised class of energy-conserving methods for Poisson systems. The actual implementation of the methods is fully discussed, with a particular emphasis to the problem at hand. Some numerical tests are reported, to assess the theoretical findings.

Keywords: discontinuous ODEs, Poisson problems, Line Integral Methods, Hamiltonian Boundary Value Methods, HBVMs, PHBVMs, EPHBVMs.

MSC: 65L05, 65P10.

1 Introduction

In some applications, one faces the problem of solving *discontinuous ODE problems*, namely, problems in the form:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}x = \begin{cases} f(x), & \text{if } g(x) \le 0, \\ \phi(x), & \text{otherwise,} \end{cases} \qquad x(0) = x_0 \in \mathbb{R}^n,$$

with $g : \mathbb{R}^n \to \mathbb{R}$ a suitably regular function, called *event function*, dividing $\mathbb{R}^n$ into two regions where the vector field is defined in different ways. The vector field need not be continuous on the boundary set

$$\Sigma = \{x \in \mathbb{R}^n \,:\, g(x) = 0\}\,. \tag{1}$$

Hereafter, we shall refer to the set Σ as to the *event set* and, moreover, we shall assume, without loss of generality, that $g(x_0) < 0$. This problem has been studied in [18] (see also [17, 20] and references therein), in the case where $g(x)$ is linear or, at most, quadratic: here, the authors define a *direct* method for finding the *event point*, namely, the first point x^* of the trajectory belonging

[*] Dipartimento di Matematica, Università di Bari, Italy. {pierluigi.amodio,felice.iavernaro}@uniba.it

[†] Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Italy. luigi.brugnano@unifi.it

to Σ. We refer to the references in [18] for relevant applications where solving such a problem is needed.

In this paper, we consider the more general case where g is a general polynomial. Consequently, given the ODE-IVP

$$\frac{\mathrm{d}}{\mathrm{d}\tau}x = f(x), \qquad x(0) = x_0 \in \mathbb{R}^n, \tag{2}$$

the problem at hand is that of determining the point x^*, on the solution trajectory, such that

$$g(x^*) = 0 \in \mathbb{R}, \tag{3}$$

where g is a polynomial (however, we shall also sketch the case of a general, sufficiently smooth, function). Hereafter, it is assumed that:

- f is sufficiently smooth;
- at the considered initial point,
$$g(x_0) = -\bar{H} < 0, \tag{4}$$
- along the solution of (2),
$$\frac{\mathrm{d}}{\mathrm{d}\tau}g(x) = \nabla g(x)^\top f(x) \geq \delta > 0, \tag{5}$$
which implies that the set Σ in (1) is attractive for the trajectory starting at x_0. More precisely, any trajectory starting at a point x_0 satisfying (4) and (5), will reach Σ in a *finite time* τ^* (depending on the starting point).[1]

As stated above, we shall refer to the vector x^* satisfying (3) as the *event point*. Its detection is significant in many applications, where it is mandatory that the set Σ is not crossed, but just reached by the trajectory [18]. Therefore, it makes sense to impose the same requirement to a numerical method of approximation.

With these premises, the structure of the paper is as follows: in Section 2 we cast the problem (2)–(4) in Poisson form; in Section 3 we recall the basic facts about the solution procedure for the new formulation, which is duly adapted for the problem at hand; in Section 4 we provide some numerical tests; at last, in Section 5 a few conclusions are given.

2 Poisson formulation

As previously observed, the numerical procedures studied in [18, 20] allow to effectively solve the problem (2)–(4) when g is a *linear* or a *quadratic* function, by using a suitable reparametrization of time. In particular, we shall use a reparametrization akin to that used in [18], i.e.,

$$\omega = g(x(\tau)) + \bar{H}, \tag{6}$$

which allows solving the problem in the interval $\omega \in [0, \bar{H}]$, due to (4) and to the monotonicity of g along the solution of (2) (see (5)). Consequently, by introducing the augmented state vector

$$y = \begin{pmatrix} x \\ \omega \end{pmatrix} \in \mathbb{R}^m, \qquad m := n+1, \tag{7}$$

[1] In fact, by virtue of (5), one obtains that $\tau^* \leq \bar{H}/\delta$.

and using the independent variable

$$t = \omega, \tag{8}$$

we obtain the augmented system, equivalent to (2),

$$\dot{y} = G(y), \qquad t \in [0, \bar{H}], \qquad y(0) = y_0 := \begin{pmatrix} x_0 \\ 0 \end{pmatrix}, \tag{9}$$

where, hereafter, $\dot{y}$ will denote the derivative w.r.t. t, and (see (2) and (7))

$$G(y) := \begin{pmatrix} \frac{f(x)}{\nabla g(x)^\top f(x)} \\ 1 \end{pmatrix}. \tag{10}$$

Clearly, because of (6), the problem (9) has the scalar invariant

$$H(y) := g(x) - \omega + \bar{H}. \tag{11}$$

In fact, one has (see (7) and (10)):

$$\begin{aligned} \dot{H}(y) &= \nabla H(y)^\top \dot{y} = \begin{pmatrix} \nabla g(x)^\top & -1 \end{pmatrix} G(y) \\ &= \begin{pmatrix} \nabla g(x)^\top & -1 \end{pmatrix} \begin{pmatrix} \frac{f(x)}{\nabla g(x)^\top f(x)} \\ 1 \end{pmatrix} = 1 - 1 = 0. \end{aligned} \tag{12}$$

Since (see (4))

$$H(y_0) = g(x_0) - \omega(0) + \bar{H} = -\bar{H} + \bar{H} = 0,$$

at $t = \omega = \bar{H}$ it will be

$$y(\bar{H}) = \begin{pmatrix} x^* \\ \bar{H} \end{pmatrix}, \qquad x^* := x(\bar{H}), \tag{13}$$

such that

$$0 = H(y(\bar{H})) = g(x(\bar{H})) - \omega(\bar{H}) + \bar{H} \equiv g(x^*) - \bar{H} + \bar{H} = g(x^*),$$

thus recovering the event point $x^* \in \Sigma$.

The novelty of the present paper is that of deriving procedures able to reach the event point in a finite number of steps, in the case where $h \in \Pi_\nu$.[2] The basic idea is that of transforming the original system (9) into an equivalent Poisson problem:

$$\dot{y} = B(y)\nabla H(y), \quad t \in [0, \bar{H}], \qquad y(0) = y_0 \in \mathbb{R}^m, \qquad B(y)^\top = -B(y). \tag{14}$$

The following result, based on [23], holds true.

Theorem 1 *Problems (9)–(10) and (14) are equivalent, and possess the invariant $H(y) \equiv 0$, provided that the skew-symmetric matrix $B(y)$ is defined as follows:*

$$B(y) = \frac{G(y)\nabla H(y)^\top - \nabla H(y) G(y)^\top}{\|\nabla H(y)\|_2^2}. \tag{15}$$

[2] As is usual, Π_ν denotes the vector space of polynomials of degree not larger than ν.

Proof First of all, we observe that matrix (15) is well defined, since $\|\nabla H(y)\|_2^2 > 1$ (see (5), (7), and (11)). Next, for the problem (14) one has

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

due to the fact that $B(y)$ is skew-symmetric. Moreover, since $\nabla H(y)^\top G(y) = 0$ (see (10) and (12)), one has:

$$\begin{aligned} B(y)\nabla H(y) &= \frac{G(y)\nabla H(y)^\top - \nabla H(y) G(y)^\top}{\|\nabla H(y)\|_2^2} \nabla H(y) \\ &= \frac{G(y)\overbrace{\nabla H(y)^\top \nabla H(y)}^{=\|\nabla H(y)\|_2^2} - \nabla H(y)\overbrace{G(y)^\top \nabla H(y)}^{=0}}{\|\nabla H(y)\|_2^2} \\ &= G(y). \quad \Box \end{aligned}$$

When matrix $B(y)$ is constant, as in the case of Hamiltonian problems,

$$\dot{y} = J\nabla H(y), \qquad t > 0, \qquad y(0) = y_0, \qquad J^\top = -J, \tag{16}$$

then $H(y)$ is referred to as the *energy*, and its conservation can be effectively and efficiently obtained by solving problem (16) via *Hamiltonian Boundary value Methods (HBVMs)*, a class of energy-conserving Runge-Kutta methods for Hamiltonian problems (see, e.g., [7, 8, 9, 10, 11, 3, 12] and the monograph [5], see also the review paper [6]). Nevertheless, in the case where the problem is not Hamiltonian, HBVMs are no more energy-conserving. When $B(y) = -B(y)^\top$ is not constant, problem (14) is a particular instance of a Poisson problem. This motivates the present paper, where a recently-derived energy-conserving variant of HBVMs for Poisson problems [1] will be suitably adapted for solving problem (14)-(15).

For sake of completeness, we mention that the numerical solution of Poisson problems has been tackled by following many different approaches (see, e.g., [19, Chapter VII] and references therein). More recently, it has been considered in [16], where an extension of the AVF method is proposed, and in [2, 4], where a line integral approach has been used instead. Functionally fitted methods have been proposed in [21, 22, 24].

3 Poisson HBVMs and their enhanced version

Let us sketch the *Poisson HBVMs (PHBVMs)* methods defined in [1], which will be later slightly modified for the problem at hand. Since we deal with one-step methods, we can consider the solution of problem (14) on the interval $[0, h]$, with $h > 0$ the timestep. The basic idea is that of expanding the vector field (14) along the orthonormal Legendre polynomial basis,

$$P_i \in \Pi_i, \qquad \int_0^1 P_i(\xi) P_j(\xi) \mathrm{d}\xi = \delta_{ij}, \qquad i, j = 0, 1, \ldots, \tag{17}$$

with δ_{ij} the Kronecker symbol. In so doing, with similar steps as in [1], by considering the expansions

$$\begin{aligned} &\nabla H(y(ch)) = \sum_{j\ge0} P_j(c)\gamma_j(y), \qquad P_j(c)B(y(ch)) = \sum_{i\ge0} P_i(c)\rho_{ij}(y), \qquad c \in [0,1], \\ &\gamma_j(y) = \int_0^1 P_j(\xi)\nabla H(y(\xi h))\mathrm{d}\xi, \qquad \rho_{ij}(y) = \int_0^1 P_i(\xi)P_j(\xi)B(y(\xi h))\mathrm{d}\xi, \qquad i, j = 0, 1, \ldots, \end{aligned} \tag{18}$$

one obtains:

$$\begin{aligned}\dot{y}(ch) &= B(y(ch))\nabla H(y(ch)) = B(y(ch))\sum_{j\ge0} P_j(c)\gamma_j(y) \qquad (19)\\ &= \sum_{j\ge0} P_j(c)B(y(ch))\gamma_j(y) = \sum_{i,j\ge0} P_i(c)\rho_{ij}(y)\gamma_j(y), \qquad c\in[0,1],\end{aligned}$$

from which one derives that the solution of (14) can be formally written as:

$$y(ch) = y_0 + h\sum_{i,j\ge0}\int_0^c P_i(\xi)\mathrm{d}\xi\rho_{ij}(y)\gamma_j(y), \qquad c\in[0,1]. \qquad (20)$$

In particular, by considering (17) and that $P_0(\xi)\equiv1$, from which $\int_0^1 P_i(\xi)\mathrm{d}\xi = \delta_{i0}$ follows, one has:

$$\begin{aligned}y(h) &= y_0 + h\sum_{j\ge0}\rho_{0j}(y)\gamma_j(y)\\ &\equiv y_0 + h\sum_{j\ge0}\int_0^1 P_j(\xi)B(y(\xi h))\mathrm{d}\xi\int_0^1 P_j(\xi)\nabla H(y(\xi h))\mathrm{d}\xi. \qquad (21)\end{aligned}$$

In order to obtain a polynomial approximation of degree s to y, it suffices to truncate the two infinite series in (19) after s terms:

$$\dot{\sigma}(ch) = \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma)\gamma_j(\sigma), \qquad c\in[0,1], \qquad (22)$$

with $\rho_{ij}(\sigma)$ and $\gamma_j(\sigma)$ defined according to (18) by formally replacing y by σ. Consequently, (20) becomes

$$\sigma(ch) = y_0 + h\sum_{i,j=0}^{s-1}\int_0^c P_i(\xi)\mathrm{d}\xi\rho_{ij}(\sigma)\gamma_j(\sigma), \qquad c\in[0,1], \qquad (23)$$

providing the approximation

$$\begin{aligned}y_1 &:= \sigma(h) = y_0 + h\sum_{j=0}^{s-1}\rho_{0j}(\sigma)\gamma_j(\sigma)\\ &\equiv y_0 + h\sum_{j=0}^{s-1}\int_0^1 P_j(\xi)B(\sigma(\xi h))\mathrm{d}\xi\int_0^1 P_j(\xi)\nabla H(\sigma(\xi h))\mathrm{d}\xi, \qquad (24)\end{aligned}$$

in place of (21). The following results hold true.

Lemma 1 *With reference to (18), for any suitably regular path $\sigma:[0,h]\to\mathbb{R}^m$ one has:*

$$\gamma_j(\sigma) = O(h^j), \qquad \rho_{ij}(\sigma) = \rho_{ji}(\sigma) = -\rho_{ij}(\sigma)^\top = O(h^{|i-j|}), \qquad i,j=0,1,\dots. \qquad (25)$$

Proof See [1, Corollary 1 and Lemma 2]. □

Theorem 2 $H(y_1) = H(y_0), \quad y_1 - y(h) = O(h^{2s+1})$.

Proof See [1, Theorems 1 and 2]. □

3.1 Enforcing (8)

As is clear, when applying (22)–(24), to problem (14)-(15), in order for x^* to be reached at $t = \bar{H}$, it is mandatory that the equality (8) holds at the end of each integration step. Conversely, at $\bar{H}$, one would have $g(x^*) \neq 0$.[3] Consequently, one must have, by setting $e_m \in \mathbb{R}^m$ the last unit vector,

$$e_m^\top y_1 = h\, e_m^\top \int_0^1 \dot{\sigma}(ch)\mathrm{d}c = h,$$

i.e.,

$$e_m^\top \int_0^1 \dot{\sigma}(ch)\mathrm{d}c = 1. \tag{26}$$

For this purpose, we specialize, for the problem at hand, the strategy used in [1] for enforcing the conservation of Casimirs, thus resulting into a specific version of *Enhanced PHBVMs (EPHBVMs)*. Let us then consider, for a generic skew-symmetric matrix

$$\tilde{B}^\top = -\tilde{B} \in \mathbb{R}^{m\times m}, \tag{27}$$

the following modified polynomial in place of (22):[4]

$$\dot{\sigma}_\alpha(ch) = \sum_{i,j=0}^{s-1} P_i(c)\rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha\tilde{B}\gamma_0(\sigma_\alpha), \qquad c \in [0,1], \qquad \sigma_\alpha(0) = y_0, \tag{28}$$

with α a scalar to be determined. The following result holds true.

Theorem 3 *Setting*

$$y_1 \;:=\; \sigma_\alpha(h) \;\equiv\; y_0 + h\sum_{j=0}^{s-1}\rho_{0j}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha h\tilde{B}\gamma_0(\sigma_\alpha), \tag{29}$$

one has $H(y_1) = H(y_0)$, whichever the value of α considered in (28).

Proof In fact, one has:

$$\begin{aligned} H(y_1) - H(y_0) \;&=\; H(\sigma_\alpha(h)) - H(\sigma_\alpha(0)) \;=\; h\int_0^1 \nabla H(\sigma_\alpha(ch))^\top \dot{\sigma}_\alpha(ch)\mathrm{d}c \\ &=\; h\sum_{i,j=0}^{s-1} \underbrace{\int_0^1 \nabla P_i(c) H(\sigma_\alpha(ch))^\top \mathrm{d}c}_{=\gamma_i(\sigma_\alpha)^\top}\, \rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha h \underbrace{\int_0^1 \nabla H(\sigma_\alpha(ch))^\top \mathrm{d}c}_{=\gamma_0(\sigma_\alpha)^\top}\, \tilde{B}\gamma_0(\sigma_\alpha) \\ &=\; h\sum_{i,j=0}^{s-1} \gamma_i(\sigma_\alpha)^\top \rho_{ij}(\sigma_\alpha)\gamma_j(\sigma_\alpha) \;-\; \alpha h \gamma_0(\sigma_\alpha)^\top \tilde{B}\gamma_0(\sigma_\alpha) \;=\; 0, \end{aligned}$$

by virtue of Lemma 1 and (27). □

[3] Actually, x^* is the order $2s$ approximation provided by the method.

[4] Here, we take into account that $P_0(c) \equiv 1$.

At this point, in order to enforce (8), according to (26) we require:

$$1 \;=\; \int_0^1 e_m^\top \dot\sigma_\alpha(ch)\mathrm{d}c \;=\; \sum_{j=0}^{s-1} e_m^\top \rho_{0j}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha e_m^\top \tilde B \gamma_0(\sigma_\alpha),$$

i.e.,

$$\alpha = \frac{\sum_{j=0}^{s-1} e_m^\top \rho_{0j}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - 1}{e_m^\top \tilde B \gamma_0(\sigma_\alpha)}. \tag{30}$$

Since $\sigma_0 \equiv \sigma$, from Theorem 2, we now that the numerator is $O(h^{2s}) + O(\alpha)$. Consequently, from (29) it follows that the order of the method remains $2s$, provided that the denominator in (30) is bounded away from 0. For this purpose, setting (according to (7))

$$\sigma_\alpha(ch) =: \begin{pmatrix} x_\alpha(ch) \\ \omega_\alpha(ch) \end{pmatrix}, \quad x_\alpha(ch) \approx x(ch) \in \mathbb{R}^n, \quad \omega_\alpha(ch) = \omega(ch) \in \mathbb{R}, \qquad c \in [0,1],$$

and recalling that

$$\gamma_0(\sigma_\alpha) = \int_0^1 \nabla H(\sigma_\alpha(ch))\mathrm{d}c = \begin{pmatrix} \int_0^1 \nabla g(x_\alpha(ch))\mathrm{d}c \\ -1 \end{pmatrix} =: \begin{pmatrix} g_0 \\ -1 \end{pmatrix}, \tag{31}$$

the choice

$$\tilde B \;=\; \begin{pmatrix} O & -d_0 \\ d_0^\top & 0 \end{pmatrix}, \qquad d_0 \;:=\; \frac{g_0}{\|g_0\|_2^2}, \tag{32}$$

provides

$$e_m^\top \tilde B \gamma_0 \;=\; d_0^\top g_0 \;=\; 1.$$

In so doing, the approximation (29) becomes, by virtue of (32),

$$y_1 \;=\; y_0 + h\sum_{j=0}^{s-1} \rho_{0j}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - \alpha h \begin{pmatrix} d_0 \\ 1 \end{pmatrix}, \tag{33}$$

with

$$\alpha = \sum_{j=0}^{s-1} e_m^\top \rho_{0j}(\sigma_\alpha)\gamma_j(\sigma_\alpha) - 1. \tag{34}$$

3.2 Discretization

As is clear, the Fourier coefficients

$$\gamma_j(\sigma_\alpha) = \int_0^1 P_j(\xi)\nabla H(\sigma_\alpha(\xi h))\mathrm{d}\xi, \quad \rho_{ij}(\sigma_\alpha) = \int_0^1 P_i(\xi)P_j(\xi)B(\sigma_\alpha(\xi h))\mathrm{d}\xi, \quad i,j = 0,\dots,s-1,$$

need to be numerically computed. For this purpose, we use a Gauss-Legendre formula of order $2k$, with abscissae and weights (c_i, b_i), $i = 1,\dots,k$. In so doing, we obtain a new polynomial approximation, say u_α, in place of σ_α,

$$\dot u_\alpha(ch) = \sum_{i,j=0}^{s-1} P_i(c)\hat\rho_{ij}(u_\alpha)\hat\gamma_j(u_\alpha) - \alpha h \begin{pmatrix} \hat d_0 \\ 1 \end{pmatrix}, \qquad c \in [0,1], \tag{35}$$

where, setting as before,

$$u_\alpha(ch) =: \begin{pmatrix} x_\alpha(ch) \\ \omega_\alpha(ch) \end{pmatrix}, \quad x_\alpha(ch) \approx x(ch) \in \mathbb{R}^n, \quad \omega_\alpha(ch) = \omega(ch) \in \mathbb{R}, \qquad c \in [0,1],$$

$\hat{d}_0$ is defined (compare with (32)) as

$$\hat{d}_0 = \frac{\hat{g}_0}{\|\hat{g}_0\|_2^2}, \qquad \hat{g}_0 = \sum_{\ell=1}^{k} b_\ell \nabla g(u_\alpha(c_\ell h)), \tag{36}$$

and we use the (generally) approximate Fourier coefficients

$$\begin{aligned} \hat{\gamma}_j(u_\alpha) &= \sum_{\ell=1}^{k} b_\ell P_j(c_\ell) \nabla H(u_\alpha(c_\ell h)), \qquad\qquad (37) \\ \hat{\rho}_{ij}(u_\alpha) &= \sum_{\ell=1}^{k} b_\ell P_i(c_\ell) P_j(c_\ell) B(u_\alpha(c_\ell h)), \qquad i,j = 0,\dots,s-1. \end{aligned}$$

At last, α is defined as (compare with (34)):

$$\alpha = \sum_{j=0}^{s-1} e_m^\top \hat{\rho}_{0j}(u_\alpha) \hat{\gamma}_j(u_\alpha) - 1. \tag{38}$$

Setting, as usual (compare with (29)),

$$y_1 \;:=\; y_0 + h \sum_{j=0}^{s-1} \hat{\rho}_{0j}(u_\alpha) \gamma_j(\sigma_\alpha) - \alpha h \begin{pmatrix} \hat{d}_0 \\ 1 \end{pmatrix}, \tag{39}$$

the following results follow.

Theorem 4 $\forall k \ge s \,:\, y_1 - y(h) = O(h^{2s+1})$.

Proof See [1, Theorem 10]. □

Theorem 5 *With reference to (36)–(39), if* $g \in \Pi_\nu$ *and* $\nu \le \frac{2k}{s}$*, then*

$$\hat{g}_0 = g_0 \equiv \int_0^1 \nabla g(x_\alpha(ch))\mathrm{d}c, \quad \hat{\gamma}_j(u_\alpha) = \gamma_j(u_\alpha) \equiv \int_0^1 P_j(\xi) \nabla H(u_\alpha(\xi h))\xi, \quad j = 0,\dots,s-1. \tag{40}$$

Consequently, $H(y_1) = H(y_0)$.

Proof The first statement follows from the fact that the integrands in (40) are polynomials of degree at most $(\nu-1)s + s - 1 = \nu s - 1 \le 2k-1$. Energy conservation is then proved with similar steps as in the proof of Theorem 3, by formally replacing σ_α with u_α. □

Remark 1 *In the case where g is not a polynomial, or is a polynomial but the hypotheses of the previous Theorem 5 are not fulfilled, from [1, Theorem 9] it follows that*

$$H(y_1) = H(y_0) + O(h^{2k+1}).$$

Consequently, a practical *energy-conservation can always be gained, provided that k is chosen large enough so that the energy error falls below the round-off error level.*

Definition 1 *According to [1, Definition 2], we shall refer to the numerical method defined by (36)–(39) as the EPHBVM(k,s) method.*

Remark 2 *We observe that, when $k = s$, the EPHBVM(s,s) method naturally satisfies the constraint (8). Consequently, $\alpha = 0$ and the method coincides with the symplectic s-stage Gauss-Legendre collocation method used for solving (14)-(15).*

For sake of completeness, let us sketch the vector form of the EPHBVM(k,s) method, which can be derived by slightly adapting the arguments in [1, Section 4.1]. For this purpose, let us define the matrices (see (17))

$$\begin{aligned}&\mathcal{P}_s = \left(\ P_{j-1}(c_i)\ \right)_{\substack{i=1,\dots,k\\ j=1,\dots,s}}, \ \mathcal{I}_s = \left(\ \int_0^{c_i} P_{j-1}(\xi)\mathrm{d}\xi\ \right)_{\substack{i=1,\dots,k\\ j=1,\dots,s}} \in \mathbb{R}^{k\times s},\\ &\Omega = \mathrm{diag}(b_1,\dots,b_k),\end{aligned}$$

with (c_i, b_i), $i = 1,\dots,k$, the abscissae and weights of the Gauss-Legendre quadrature, and the vectors (see (37))

$$\boldsymbol{e} = \begin{pmatrix} 1\\ \vdots\\ 1\end{pmatrix} \in \mathbb{R}^k, \qquad \boldsymbol{\phi} = \begin{pmatrix}\phi_0\\ \vdots\\ \phi_{s-1}\end{pmatrix}, \quad \phi_i = \sum_{j=0}^{s-1} \hat\rho_{ij}(u_\alpha)\hat\gamma_j(u_\alpha) \in \mathbb{R}^m, \quad i = 0,\dots,s-1.$$

We also set, being[5]

$$Y \equiv \begin{pmatrix} Y_1\\ \vdots\\ Y_k\end{pmatrix} \in \mathbb{R}^{km}, \qquad Y_i \equiv \begin{pmatrix} x_i\\ \omega_i\end{pmatrix}, \quad i = 1,\dots,k, \tag{41}$$

the stages of the method, and (see (14)-(15)),

$$\mathcal{B}(Y) = \begin{pmatrix} B(Y_1) & & \\ & \ddots & \\ & & B(Y_k)\end{pmatrix} \in \mathbb{R}^{km\times km},$$

one then obtains the discrete problem

$$\mathcal{F}(\boldsymbol{\phi},\alpha) := \begin{pmatrix} \boldsymbol{\phi} - \mathcal{P}_s^\top\Omega \otimes I_m \mathcal{B}(Y)\cdot \mathcal{P}_s\mathcal{P}_s^\top \Omega\otimes I_m \nabla H(Y)\\ \alpha - e_m^\top \phi_0 + 1\end{pmatrix} = \mathbf{0} \in \mathbb{R}^{sm+1}, \tag{42}$$

[5] According to (7), $x_i \in \mathbb{R}^n$ and $\omega_i \in \mathbb{R}$.

where, with reference to (41), and setting $\boldsymbol{c} = (c_1, \dots, c_k)^\top$ the vector of the abscissae:

$$\begin{aligned} Y &= \boldsymbol{e} \otimes y_0 + h\mathcal{I}_s \otimes I_m \boldsymbol{\phi} - \alpha h \boldsymbol{c} \otimes \begin{pmatrix} \hat{d}_0 \\ 1 \end{pmatrix}, \\ \hat{d}_0 &= \frac{\hat{g}_0}{\|\hat{g}_0\|_2^2}, \\ \hat{g}_0 &= \sum_{i=1}^{k} b_i \nabla g(x_i). \end{aligned}$$

Remark 3 *We observe that the above discrete problem (42) has, remarkably, (block) dimension s, independently of the considered value of k [1]. Moreover, it induces a straightforward fixed-point iteration, which converges for all sufficiently small timesteps h, under regularity assumptions on f and g. This iteration will be used for the numerical tests, even though Newton-type procedures, obtained adapting those defined in [3, 9] for HBVMs (see also [13, 14, 15]), could be also considered.*

4 Numerical tests

In this section we present a few numerical tests, concerning the solution of one-sided event location problems, aimed at assessing the theoretical findings. For each problem we prescribe the function $f(x)$ in (2) and the event function $g(x)$ (3), along with the starting point of the trajectory. All the numerical tests have been implemented in Matlab (R2020a) on a 3 GHz Intel Xeon W10 core computer with 64 GB of memory.

Example 1 The first test problem, taken from [18, Example 5.1 (a)], is defined by

$$f(x) = \begin{pmatrix} x_2 \\ \frac{1}{1.2 - x_2} - x_1 \end{pmatrix}, \tag{43}$$

and by the event function

$$g(x) = x_1 + x_2 - 0.4. \tag{44}$$

We choose the initial point

$$x(0) = (-0.2, \, -0.2)^\top, \tag{45}$$

providing the value $\bar{H} \equiv -g(x(0)) = 0.8$ in (4). Since the event function is linear, any EPHBVM(s,s) method (i.e., the s-stage Gauss collocation method) has to provide an order $2s$ approximation to the event point belonging to the event set Σ (1).This is confirmed by the numerical tests listed in Table 1, obtained by using the timesteps

$$h_n := \frac{\bar{H}}{10 \cdot 2^n} \tag{46}$$

for solving the associated Poisson problem (14)-(15). In the table, we have denoted by x_n^* the approximation to the event point obtained with the timestep h_n, and with e_n^* the corresponding error (numerically estimated). As one may see, all approximations belong to the event set Σ (since $g(x_n^*)$ is of the order of the round-off error level) and converge to the event point with the correct order (the last two approximations for $s = 3$ practically coincide).

Table 1: numerical results for problem (43)–(45) solved by the EPHBVM(s,s) method with timestep (46).

	$s=1$			$s=2$			$s=3$		
n	$g(x_n^*)$	e_n^*	rate	$g(x_n^*)$	e_n^*	rate	$g(x_n^*)$	e_n^*	rate
0	-1.11e-16	—	—	-1.67e-16	—	—	-1.05e-15	—	—
1	-5.55e-17	1.99e-04	—	-1.67e-16	1.43e-07	—	-9.44e-16	6.09e-10	—
2	1.11e-16	4.98e-05	2.0	0.00e+00	9.33e-09	3.9	-9.99e-16	1.06e-11	5.8
3	-1.67e-16	1.25e-05	2.0	-2.22e-16	5.90e-10	4.0	-8.88e-16	1.72e-13	6.0
4	-5.55e-17	3.11e-06	2.0	-1.67e-16	3.70e-11	4.0	-1.22e-15	2.60e-15	6.0
5	2.22e-16	7.78e-07	2.0	-1.11e-16	2.31e-12	4.0	-1.11e-15	1.39e-16	***

Table 2: numerical results for problem (43) and (47)-(48) solved by the EPHBVM(k,s) method with timestep $h=\bar{H}/10$.

	s	EPHBVM(s,s)	EPHBVM$(4,s)$
$g(x^*)$	1	1.1148e-05	1.1102e-16
	2	-1.4687e-08	1.1102e-16
	3	-7.8148e-11	-2.2204e-16

Example 2 Next, we consider the problem defined by (43), with a nonlinear (though smooth) event function

$$g(x) = 20x_1 + x_2 - 20\sin x_1 - 0.4, \tag{47}$$

and initial point

$$x(0) = (0,\, -0.2)^\top, \tag{48}$$

providing a value $\bar{H}=0.6$. If we solve the associated Poisson problem (14)-(15) by using the EPHBVM(s,s) and EPHBVM$(4,s)$ methods, $s=1,2,3$, with a timestep $h=\bar{H}/10$, we see that, though $g(x)$ is non-polynomial, the latter methods correctly reaches the event set Σ, as one infers from the results listed in Table 2.

Example 3 At last, let us consider the problem defined by:

$$f(x) = \begin{pmatrix} \frac{1}{1.2+\sin x_2} \\ \frac{1}{1.2-\cos x_1} \\ 1+\cos\|x\|_2^2 \end{pmatrix}, \qquad x(0) = \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix}, \tag{49}$$

with the polynomial event function [6]

$$g(x) = 10^{-3}\left(x_1^3 + 4x_2^7 + x_3^5\right). \tag{50}$$

For the chosen initial point, one obtains $\bar{H}=1.563$. For this problem, any EPHBVM(k,s) method, with

$$k \ge \frac{7s}{2},$$

[6] The scaling factor 10^{-3} in (50) is introduced to have a more compact graphical representation.

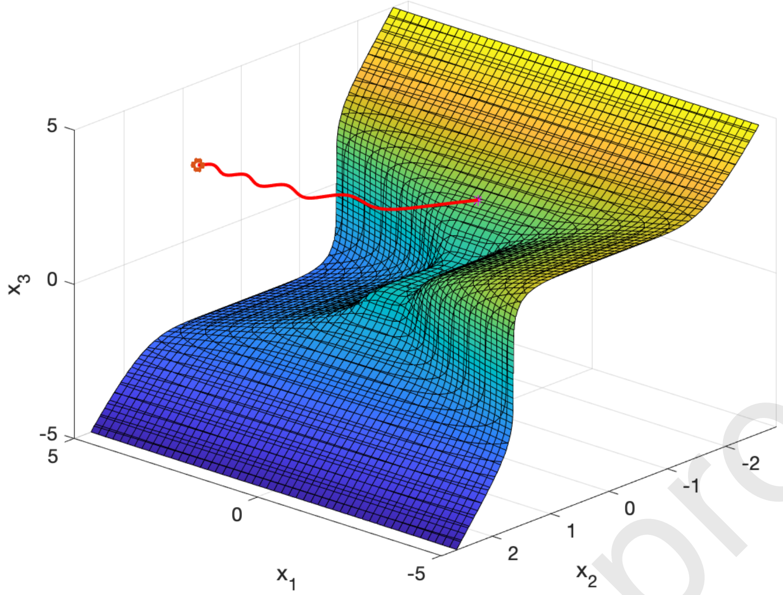


Figure 1: event set (1) for problem (49)-(50), along with the trajectory reaching it computed by using the EPHBVM(11,3) method.

turns out to be energy-conserving for the associated Poisson problem (14)-(15), and therefore, at $t = \bar{H}$ the trajectory exactly reaches the event point lying on the event set Σ. We use the EPHBVM(11,3) method with timestep $h = 10^{-3}\bar{H}$, thus reaching the (approximation of the) event point x^* for which $g(x^*) \approx -5.8 \cdot 10^{-16}$. The event set Σ, along with the computed trajectory, are depicted in Figure 1. For comparison, the EPHBVM(3,3) (i.e., the 3-stage Gauss-Legendre method), using the same timestep, reaches a point $\tilde{x}$ for which $g(\tilde{x}) \approx 4.7 \cdot 10^{-8}$ and, moreover, $\|\tilde{x} - x^*\|_2 \approx 4.8 \cdot 10^{-6}$.

5 Conclusions

In this paper, starting from the methodology introduced in [18], we have introduced a direct method for numerically solving the problem of one-sided event location. The proposed approach is based on a suitable modification of recently derived energy-conserving methods for Poisson problems [1], specifically tailored for the problem at hand. The methods exactly reach the event set, in the case where the event function is a polynomial. Actually, they can be effectively used also in the non-polynomial case, provided that the event function is regular enough. Numerical examples confirm the theoretical findings.

Acknowledgements

The authors wish to thank the *mrSIR project* crowdfunding [25] for the financial support.

Declaration of interest

The authors declare no competing interest.

References

[1] P. Amodio, L. Brugnano, F. Iavernaro. Arbitrarily high-order energy-conserving methods for Poisson problems. *Numer. Algorithms* (2022) `https://doi.org/10.1007/s11075-022-01285-z`

[2] L. Brugnano, M. Calvo, J.I. Montijano, L. Rández. Energy preserving methods for Poisson systems. *J. Comput. Appl. Math.* **236** (2012) 3890–3904. `https://doi.org/10.1016/j.cam.2012.02.033`

[3] L. Brugnano, G. Frasca Caccia, F. Iavernaro. Efficient implementation of Gauss collocation and Hamiltonian Boundary Value Methods. *Numer. Algorithms* **65** (2014) 633–650. `https://doi.org/10.1007/s11075-014-9825-0`

[4] L. Brugnano, G. Gurioli, F. Iavernaro. Analysis of Energy and QUadratic Invariant Preserving (EQUIP) methods. *J. Comput. Appl. Math.* **335** (2018) 51–73. `https://doi.org/10.1016/j.cam.2017.11.043`

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

[6] L. Brugnano, F. Iavernaro. Line Integral Solution of Differential Problems. *Axioms* **7**(2) (2018) article n. 36. `http://dx.doi.org//10.3390/axioms7020036`

[7] 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. `https://doi.org/10.1063/1.3241566`

[8] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian Boundary Value Methods (Energy Preserving Discrete Line Integral Methods). *JNAIAM J. Numer. Anal. Ind. Appl. Math.* **5**, 1-2 (2010) 17–37.

[9] L. Brugnano, F. Iavernaro, D. Trigiante. A note on the efficient implementation of Hamiltonian BVMs. *J. Comput. Appl. Math.* **236** (2011) 375–383. `https://doi.org/10.1016/j.cam.2011.07.022`

[10] L. Brugnano, F. Iavernaro, D. Trigiante. The lack of continuity and the role of infinite and infinitesimal in numerical methods for ODEs: the case of symplecticity. *Appl. Math. Comput.* **218** (2012) 8056–8063. `https://doi.org/10.1016/j.amc.2011.03.022`

[11] L. Brugnano, F. Iavernaro, D. Trigiante. A simple framework for the derivation and analysis of effective one-step methods for ODEs. *Appl. Math. Comput.* **218** (2012) 8475–8485. https://doi.org/10.1016/j.amc.2012.01.074

[12] L. Brugnano, F. Iavernaro, D. Trigiante. Analysis of Hamiltonian Boundary Value Methods (HBVMs): A class of energy-preserving Runge-Kutta methods for the numerical solution of polynomial Hamiltonian systems. *Commun. Nonlinear Sci. Numer. Simul.* **20** (2015) 650–667. https://doi.org/10.1016/j.cnsns.2014.05.030

[13] L. Brugnano, F. Iavernaro, C. Magherini. Efficient implementation of Radau collocation methods. *Appl. Numer. Math.* **87** (2015) 100–113. https:/doi.org/10.1016/j.apnum.2014.09.003

[14] L. Brugnano, C. Magherini. Blended Implementation of Block Implicit Methods for ODEs. *Appl. Numer. Math.* **42** (2002) 29–45. https://doi.org/10.1016/S0168-9274(01)00140-4

[15] L. Brugnano, C. Magherini. Recent Advances in Linear Analysis of Convergence for Splittings for Solving ODE problems. *Appl. Numer. Math.* **59** (2009) 542–557. https://doi.org/10.1016/j.apnum.2008.03.008

[16] D. Cohen, E. Hairer. Linear energy-preserving integrators for Poisson systems. *BIT Numer. Math.* **51** (2011) 91–101. http://doi.org/10.1007/s10543-011-0310-z

[17] L. Dieci, L. Lopez. A survey of numerical methods for IVPs of ODEs with discontinuous right-hand side. *J. Comput. Appl. Math.* **236** (2012) 3967–3991. https://doi.org/10.1016/j.cam.2012.02.011

[18] L. Dieci, L. Lopez. One-sided direct event location techniques in the numerical solution of discontinuous differential systems. *BIT Numer. Math.* **55** (2015) 987–1003. http://doi.org/10.1007/s10543-014-0538-5

[19] E. Hairer, C. Lubich, G. Wanner. *Geometric Numerical Integration, 2nd ed.*. Springer, Berlin, 2006.

[20] L. Lopez, S. Maset. Time-transformations for the event location in discontinuous ODEs. *Math. Comp.* **87** (2017) 2321–2341. https://doi.org/10.1090/mcom/3305

[21] Y. Miyatake. A derivation of energy-preserving exponentially-fitted integrators for Poisson systems. *Comput. Phys. Commun.* **187** (2015) 156–161. http://doi.org/10.1016/j.cpc.2014.11.003

[22] L. Mei, L. Huang, X. Wu. A unified framework for the study of high-order energy-preserving integrators for solving Poisson systems. *J. Comput. Phys.* **450** (2022) 110822. https://doi.org/10.1016/j.jcp.2021.110822

[23] G.R.W. Quispel, H.W. Capel. Solving ODEs numerically while preserving a first integral. *Phys. Letters A* **218** (1996) 223–228. https://doi.org/10.1016/0375-9601(96)00403-3

[24] B. Wang, X. Wu. Functionally-fitted energy-preserving integrators for Poisson systems. *J. Comput. Phys.* **364** (2018) 137–152. https://doi.org/10.1016/j.jcp.2018.03.015

[25] The *mrSIR* project: https://www.mrsir.it/en/about-us/