Applied Numerical Mathematics 200 (2024) 273–285

Contents lists available at ScienceDirect

Applied Numerical Mathematics

journal homepage: www.elsevier.com/locate/apnum

Hermite–Birkhoff spline Quasi-Interpolation with application as dense output for Gauss–Legendre and Gauss–Lobatto Runge–Kutta schemes

Antonella Falini [a], Francesca Mazzia [a,*], Alessandra Sestini [b]

[a] Dipartimento di Informatica, Università degli studi di Bari Aldo Moro, Italy
[b] Dipartimento di Matematica e Informatica "Ulisse Dini", Università degli studi di Firenze, Italy

A R T I C L E   I N F O

Article history:
Received 23 February 2023
Received in revised form 19 July 2023
Accepted 24 July 2023
Available online 28 July 2023

Keywords:
Spline Quasi-Interpolation
B-splines
Hermite-Birkhoff
Continuous extension
Dense output
Runge–Kutta

A B S T R A C T

Spline Quasi-Interpolation (QI) of even degree $2R$ on general partitions is introduced, where derivative information up to order $R \geq 1$ at the spline breakpoints is required and maximal convergence order can be proved. Relying on the B-spline basis with possible multiple inner knots, a family of quasi-interpolating splines with smoothness of order R is associated with each $R \geq 1$, since there is the possibility of using different local sequences of breakpoints to define each QI spline coefficient. By using suitable finite differences approximations of the necessary discrete derivative information, each QI spline in this family can be associated also with a twin approximant belonging to the same spline space, but requiring just function information at the breakpoints.
Among possible different applications of the introduced QI scheme, a smooth continuous extension of the numerical solution of Gauss-Lobatto and Gauss-Legendre Runge-Kutta methods is here considered. When $R > 1$, such extension is based on the use of the variant of the QI scheme with derivative approximation which preserves the approximation power of the original Runge-Kutta scheme.

1. Introduction

In this paper the continuous extension approach presented in [16] in connection with multiderivative Hermite-Obreshkov schemes is generalized, deriving a new family of Hermite-Birkhoff Quasi-Interpolation (QI) schemes requiring multi-derivative information at the spline breakpoints which can have general distribution. The convergence behavior of such class of QI methods is analyzed, together with that of the twin class with suitably approximated derivatives. Furthermore, its combination with some Runge-Kutta (RK) schemes, Gauss-Legendre and Gauss-Lobatto ones, is introduced, with the aim to efficiently obtain a spline approximant preserving the convergence order of the RK numerical solution at the mesh points.

Observe that the application here proposed of the presented family of QI schemes is significant since, when either initial or boundary value ordinary differential equations are considered, one important task to be addressed consists in the computation of a suitable continuous extension or dense output (see Chapter 6 in [8]) of the discrete approximation

* Corresponding author.
E-mail addresses: antonella.falini@uniba.it (A. Falini), francesca.mazzia@uniba.it (F. Mazzia), alessandra.sestini@unifi.it (A. Sestini).

https://doi.org/10.1016/j.apnum.2023.07.023

produced at the mesh points by the numerical scheme. This means that, once a numerical solution $\mathbf{u}_i, i = 1, \ldots, N+1$ is obtained, with $\mathbf{u}_i$ approximating at the mesh point x_i the solution $\mathbf{y}$ of an initial or boundary value (vector) problem (IVP, BVP),

$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)), \qquad a \leq x \leq b \tag{1}$$

(plus initial or boundary conditions), the interest is in the efficient computation of a (vector) function $\mathbf{s}$ such that $\mathbf{s}(x) \approx \mathbf{y}(x), \forall x \in [a\,,\,b]$.

Note that the terminology "dense output" is often adopted in the ODE setting [8,19] as a synonym of continuous extension.

Now the computation of the numerical solution, and consequently of an associated continuous extension is necessarily different for IVPs and BVPs. For IVPs a possibility is the usage of one-step methods which adopt a forward procedure. In such case, referring without loss of generality to the scalar case, it is reasonable to construct the continuous extension with the same kind of procedure and usually it is done by defining in the current interval $[x_n, x_{n+1}]$ a local polynomial p_n that accurately approximates y in such area. Thus, the produced continuous extension is globally a spline with just C^0 or C^1 smoothness which however not necessarily shares the convergence order with the discrete numerical solution. One important use of continuous extensions in this case is to facilitate the variation of the step size, but they are also important for event detection.

Conversely numerical methods for BVPs, e.g. Boundary Value Methods (BVMs), i.e. linear multistep methods with additional boundary conditions, compute all the $u_i, i = 1, \ldots, N+1$ simultaneously. For example the BS-class [17,18] of BVMs is derived collocating the differential equation by using the B-spline basis for splines with simple knots at the mesh points. Note that the peculiarity of the k-step BS method within the BVM family, consists exactly in having naturally associated a continuous spline extension $s \in C^k[a\,,\,b]$ sharing the convergence order with the numerical solution. Considering how s was defined for BS methods, later on in [15] it has been possible to do a generalization analogous to that presented in this paper, deriving an Hermite QI (QIBSH) with maximal smoothness which requires in input only functional and first derivative information at the mesh points which are also the spline breakpoints. The code TOM [4,13] uses the QIBSH spline as continuous extension for all the included classes of BVMs.

Note that the QIBSH spline construction is similar to the procedure here proposed to compute the new QI spline, since, if N is the number of the spline polynomial segments, in both cases $O(N)$ local linear systems have to be solved when a non uniform distribution of the spline breakpoints is adopted. There are however significant differences. First of all, now we are capable of deriving a more flexible QI approach, since an additional input integer parameter is used to define different quasi-interpolating splines for the same set of input information. Furthermore, thanks to the usage of multi-derivative information and to the non maximal smoothness required for the spline at its breakpoints, the main difference descends, which consists in obtaining a more local QI scheme. Note that the less locality of QIBSHs implies for example that a forward implementation procedure could be done for the computation of successive spline segments only accepting a significant delay in the construction with respect to the computation of the occurrences of the discrete solution. Conversely, for the computation of $s|_{[x_n, x_{n+1}]}$ with the approach here introduced, just the computation of u_{n+1} is necessary. As shown in the application to Gauss-Lobatto and Gauss-Legendre schemes, the derivative-free variant of the proposed scheme can preserve the locality of its basic formulation, clearly provided that additional suitable (approximated) functional or derivative values are available at points interior to each sub-interval $[x_n, x_{n+1}]$. This is exactly what happens for the developed application of the new QI operator to collocation methods, selecting in this case the spline breakpoints at the mid-points of successive mesh points. Observe also that the approach here proposed is not only a more local alternative to QIBSHs but it is also an alternative to superconvergent interpolants for Gaussian collocation methods proposed in the literature [1,7].

The paper is structured as follows. In Section 2 some preliminary material is briefly revised. In Section 3 the new class of quasi-interpolants is presented and in the following section its convergence properties are analyzed, giving also numerical evidence of the introduced theoretical results. In Section 5 the application of our QI operator to produce dense-output for some Runge-Kutta collocation schemes is shown together with some results for both, initial and boundary value problems. Finally, Section 6 reports some conclusions.

2. Preliminary related material

In this section the main results introduced in [16] are summarized, since they can be considered the starting point of the research presented in this paper. In such a paper a family of multiderivative methods for initial value problems (IVPs) was studied, the so called one-step symmetric Hermite-Obreshkov (HO) methods firstly introduced in [8]. As a first goal, in [16] their behavior when applied to the numerical solution of Hamiltonian problems was analyzed. Then it was shown that they admit a continuous spline extension with breakpoints at the mesh points, sharing the approximation order with the numerical solution.

Referring for simplicity to the scalar case and denoting with y_0 the given initial condition to be combined for any IVP to the ODE in (1) and with $a = t_0 < \cdots < t_N = b$ a partition of the integration interval, the R-th method of such family can be written as follows,

$$\begin{cases} u_0 &= y_0 \\ u_{n+1} &= u_n + \sum_{j=1}^{R} h_n^j \beta_j^{(R)} (u_n^{(j)} - (-1)^j u_{n+1}^{(j)}),\ n = 0, \dots, N-1, \end{cases} \tag{2}$$

where

$$\beta_j^{(R)} = \frac{1}{j!} \frac{R(R-1)\dots(R-j+1)}{2R(2R-1)\dots(2R-j+1)}, \qquad j = 1, \dots, R, \tag{3}$$

with $h_n := t_{n+1} - t_n$, u_n denoting as usual a discrete approximation of $y(t_n)$ and $u_n^{(j)}$, for $j \geq 1$, the $(j-1)$-th total derivative of f with respect to t computed at (t_n, u_n), which is clearly an approximation of $y^{(j)}(t_n)$.

The spline extension s associated with the numerical solution produced by the R-th HO method has degree $2R$, breakpoints at $t_i, i = 0, \dots, N$, smoothness C^R in the integration interval and it collocates the differential equation at the mesh points with multiplicity R. In [16] an efficient forward approach for its computation suited for IVPs is introduced, expressing each spline coefficient in the B-spline basis as a linear combination of two successive values of the numerical solution and of the corresponding approximations of the derivatives of y up to order R which are involved in the R-th HO method. Furthermore, under the requirement of a smoothly varying mesh and of a sufficiently smooth f, it is proved that such a spline s approximates y with an $O(h^{2R})$ error, where $h := \max_n h_n$, being $2R$ also the order of the associated HO method. For any further detail on what summarized in this section, we refer to [16]. Even if not observed in such a reference, we will show here that the method proposed in [16] to construct such a spline can be interpreted as a QI scheme when the input multi-derivative information it requires at the mesh points are taken from an underlying function to be approximated. Indeed, in the next section we introduce a more general method which allows us to associate with the same set of information, different quasi-interpolating splines sharing the same knot vector, the same degree and smoothness and including the above mentioned one. Besides analyzing their differences, a derivative free variant of such general QI approach will be also introduced.

3. The generalized QI spline scheme

As it is always the case in the QI setting, we are here interested in obtaining with a local approach approximants belonging to a finite dimensional functional space of an underlying function y defined in $[a, b]$. In particular we are interested in deriving spline approximants requiring as input just values of y at the spline breakpoints $a = t_0 < \dots < t_N = b$ together with corresponding derivative values up to a given order (which can be anyway suitably approximated to derive a derivative-free variant of this QI approach). Taken a positive integer $R \geq 1$, chosen according to the smoothness of the function y to be approximated, let us consider the following spline space of dimension $D + 2R + 1$ with $D := (N-1)R$,

$$S^R := \langle B_{-2R}, \cdots, B_D \rangle,$$

where B_i denotes the i-th element of the $C^R[a, b]$ B-spline basis of degree $2R$ with breakpoints belonging to $T := \{t_i, i = 0, \dots, N\}$ and whose associated extended knot vector Θ is derived from T setting the multiplicity of each inner breakpoint to R and adopting coincident auxiliary knots. Regardless from R, each local spline spaces $S_{loc}^{(n)}$, $n = 0, \dots, N-1$, spanned by those B-splines which do not vanish in the interval $I_n := [t_n, t_{n+1}]$, coincides with Π_{2R} and so its dimension is $2R+1$. Then for each j ranging between 0 and R and each n ranging between 0 and $N-1$ we can introduce the local matrix $A_j^{(n)}$ which collocates at t_n and t_{n+1} the j-th derivative of all the B-splines active in I_n. Setting for brevity $\ell_n := R(n-2)$, such local matrix can be written as follows,

$$A_j^{(n)} := \begin{bmatrix} B_{\ell_n}^{(j)}(t_n), & \dots, & B_{\ell_n+2R}^{(j)}(t_n) \\ B_{\ell_n}^{(j)}(t_{n+1}), & \dots, & B_{\ell_n+2R}^{(j)}(t_{n+1}) \end{bmatrix}_{2\times(2R+1)}. \tag{4}$$

After this stage, the definition of a QI spline $s \in S^R$ which approximates y can be written as,

$$s(\cdot) = Q_R^{(HB)}(y) = \sum_{i=-2R}^{D} \mu_i(y) B_i(\cdot). \tag{5}$$

The functionals μ_i appearing in the above formula are linear and local, as usual in the QI setting. In order to explain how they are defined in our approach, let us introduce another local matrix $W^{(n)}$ which is block defined as follows,

$$W^{(n)} := \begin{bmatrix} A_0^{(n)} & \mathbf{0} \\ h_n A_1^{(n)} & -\mathbf{e} \\ h_n^2 A_2^{(n)} & \mathbf{0} \\ \vdots & \vdots \\ h_n^R A_R^{(n)} & \mathbf{0} \end{bmatrix}_{(2R+2)\times(2R+2)}, \tag{6}$$

where $\mathbf{e}$ and $\mathbf{0}$ denote two column vectors both belonging to $\mathbb{R}^2$, with unit and zero entries, respectively. In our approach, in order to compute all the spline coefficients, for each $n = 0, \dots, N-1$, the following local linear system has to be solved,

$$W^{(n)} \begin{pmatrix} \hat{\mathbf{c}}^{(n)} \\ \tau^{(n)} \end{pmatrix} = \begin{pmatrix} y_n^{(0)} \\ y_{n+1}^{(0)} \\ h_n y_n^{(1)} \\ h_n y_{n+1}^{(1)} \\ \vdots \\ h_n^R y_n^{(R)} \\ h_n^R y_{n+1}^{(R)} \end{pmatrix}, \tag{7}$$

where $\hat{\mathbf{c}}^{(n)} \in \mathbb{R}^{2R+1}$, $\tau^{(n)} \in \mathbb{R}$ and with $y_i^{(j)}$ denoting the value at t_i of the j-th derivative of the function y to be approximated. Note that the vector $\hat{\mathbf{c}}^{(n)}$ collects the set of coefficients used to define a local polynomial approximation of y in I_n. Such local approximation interpolates y and its derivatives up to order R at both t_n and t_{n+1}, except for the first derivative. Moreover,

$$\hat{c}_i^{(n)} = \sum_{j=0}^{R} h_n^j (y_n^{(j)} \boldsymbol{\gamma}_{1,n,i}^{(j)} + y_{n+1}^{(j)}\, \boldsymbol{\gamma}_{2,n,i}^{(j)}), \quad 1 \le i \le 2R+1, \tag{8}$$

$$\tau^{(n)} = \sum_{j=0}^{R} h_n^j (y_n^{(j)} \boldsymbol{\gamma}_{1,n,2R+2}^{(j)} + y_{n+1}^{(j)}\, \boldsymbol{\gamma}_{2,n,2R+2}^{(j)}), \tag{9}$$

where the row vectors $\boldsymbol{\gamma}_{n,i}^{(j)} := (\boldsymbol{\gamma}_{1,n,i}^{(j)}, \boldsymbol{\gamma}_{2,n,i}^{(j)})$, for $j = 0, \dots, R$ can be simultaneously obtained by determining the i-th column of the inverse of the transpose of the local matrix $W^{(n)}$. In particular,

$$(\boldsymbol{\gamma}_{n,i})^\top W^{(n)} = (\boldsymbol{\gamma}_{n,i}^{(0)}, \boldsymbol{\gamma}_{n,i}^{(1)}, \dots, \boldsymbol{\gamma}_{n,i}^{(R)}) W^{(n)} = \mathbf{e}_i^\top,$$

where $\mathbf{e}_i$ denotes the i-th column of the identity matrix, from expression (7) it follows that,

$$\mathbf{e}_i^\top \begin{pmatrix} \hat{\mathbf{c}}^{(n)} \\ \tau^{(n)} \end{pmatrix} = (\boldsymbol{\gamma}_{n,i})^\top \begin{pmatrix} y_n^{(0)} \\ y_{n+1}^{(0)} \\ h_n y_n^{(1)} \\ h_n y_{n+1}^{(1)} \\ \vdots \\ h_n^R y_n^{(R)} \\ h_n^R y_{n+1}^{(R)} \end{pmatrix},$$

from which (8) can be easily derived. Observe that in the ODE setting, the expression in (9) is exactly that of the local truncation error of the R-th HO method, see [16]. Indeed, the coefficients are respectively given as $\boldsymbol{\gamma}_{1,n,2R+2}^{(j)} = \beta_j^R$ and $\boldsymbol{\gamma}_{2,n,2R+2}^{(j)} = -(-1)^j \beta_j^R$, where the β_j^R are defined in (3).

The preliminary determination of the vectors $\boldsymbol{\gamma}_{n,i}$ and the definition of the coefficients $\hat{c}_i^{(n)}$ with formula (8) can be of interest for example if more than one function has to be approximated with the proposed QI approach. Furthermore, when the adopted partition of $[a, b]$ is uniform, the preliminary computation of the necessary vectors $\boldsymbol{\gamma}_i = \boldsymbol{\gamma}_{n,i}, 0 \le n \le N-1, 1 \le i \le 2R+1$ can be done, in order to use (8) for defining the spline coefficients, so avoiding the solution of any local linear system during the QI approximation procedure. Different global spline approximants s are obtained by selecting an additional integer number σ, with $0 \le \sigma \le R+1$, which is used to select which set of R successive coefficients of s is defined by using the solution of (7). In fact, using again the index $\ell_n = R(n-2)$, the following setting is adopted,

$$\begin{cases} \mu_{\ell_n+\sigma+i} & := \hat{c}_{1+\sigma+i}^{(n)}, \quad i = 0, \dots, R-1 \qquad n = 1, \dots, N-2, \\ \mu_{\ell_0+i} & := \hat{c}_{1+i}^{(0)}, \quad i = 0, \dots, \sigma + R - 1 \\ \mu_{\ell_{N-1}+\sigma+i} & := \hat{c}_{1+\sigma+i}^{(N-1)}, \quad i = 0, \dots, 2R - \sigma, \end{cases} \tag{10}$$

where clearly each coefficient μ_i depends on the function y to be approximated. We observe that when the maximal value for σ is used, that is $\sigma = R+1$, the restriction of s to $[a, t_{n+1}]$ can be constructed just using information at the breakpoints included in such interval and so the procedure is suited for applications to IVPs. On the other hand, dealing with BVPs, considering that the numerical solution is simultaneously available at all the mesh points $t_n, n = 0, \dots, N$, setting $\sigma = \lfloor \frac{R+1}{2} \rfloor$

is suggested, since the construction is more symmetric. Concerning the computational cost for the spline construction, first of all we can observe that it does not depend on σ because the same set of N local linear systems of the same size has to be solved independently from σ. Furthermore it is linear in N.

4. Convergence properties

In this Section, some theoretical properties regarding the approximation power of the derived QI scheme and of the variant which uses approximated derivatives are described.

Proposition 1. *If $y \in S^R$ then $\tau^{(n)}$ in (9) vanishes. Furthermore, if $y \in C^{(2R+1)}[a\,,\,b]$, then $\tau^{(n)} \in O(h_n^{2R+1})$.*

Proof. If y is a spline in S^R, then $y|_{I_n}$ is a polynomial of degree $2R$. In such case the unique solution of (7) is $(\hat{\mathbf{c}}^{(n)T}, 0)^T$, where $\hat{\mathbf{c}}^{(n)T}$ is the vector collecting the coefficients of such polynomial expressed in the local basis of B-splines in S^R which are active in I_n. As remarked in Section 2, the order of the R-th HO is $O(h^{2R})$, hence the local truncation error has order $O(h_n^{2R+1})$. This completes the proof. □

Corollary 1. *The quasi-interpolation operator $Q_R^{(HB)}$ is a projector on the spline space S^R.*

Proof. The proof follows immediately from the previous proposition, considered the adopted strategy. □

The following Lemma is necessary to prove the main theoretical results regarding the convergence order of the introduced QI spline.

Lemma 1. *The restriction $s|_{I_n}$ of the quasi-interpolant defined in (5) by using (10) depends on the data $y_i^{(j)}$ for $j = 0, \dots, R$ and $i = n_1(\sigma), \dots, n_2(\sigma)$ with $n_1(\sigma) = n_1 = \max\{n - 1 - \bar{\sigma}, 0\}$, $n_2(\sigma) = n_2 = \min\{n + 2 - \bar{\sigma}, N\}$, with*

$$\bar{\sigma} = \begin{cases} 0 & \text{if } \sigma = 0 \\ 1 & \text{if } 0 < \sigma < R+1 \\ 2 & \text{if } \sigma = R+1. \end{cases}$$

Proof. For the proof we need to separately consider the cases $n = 0$, $n = N - 1$ and the intermediate ones. By (10) we can find that,

- For the first interval $I_0 = [t_0, t_1]$, the index $\ell_n = \ell_0 = -2R$, hence, there are $2R + 1$ active splines defined in $\{\theta_{\ell_0}, \dots, \theta_{\ell_0+4R+1}\}$:
  - If $\sigma = 0$,

$$\begin{aligned} s|_{I_0} &= \sum_{i=0}^{R-1} \left(\hat{c}_{1+i}^{(0)} B_{\ell_0+i}(\cdot) + \hat{c}_{1+i}^{(1)} B_{\ell_0+R+i}(\cdot) \right) + \hat{c}_1^{(2)} B_{\ell_0+2R}(\cdot) \\ &= \sum_{i=0}^{R-1} \left(\mu_{\ell_0+i} B_{\ell_0+i}(\cdot) + \mu_{\ell_1+i} B_{\ell_1+i}(\cdot) \right) + \mu_{\ell_2} B_{\ell_2}(\cdot) \end{aligned}$$

  - If $\sigma = R + 1$,

$$\begin{aligned} s|_{I_0} &= \sum_{i=0}^{2R} \hat{c}_{1+i}^{(0)} B_{\ell_0+i}(\cdot) \\ &= \sum_{i=0}^{2R} \mu_{\ell_0+i} B_{\ell_0+i}(\cdot) \end{aligned}$$

  - If $0 < \sigma < R + 1$

$$\begin{aligned} s|_{I_0} &= \sum_{i=0}^{\sigma+R-1} \hat{c}_{1+i}^{(0)} B_{\ell_0+i}(\cdot) + \sum_{i=0}^{R-\sigma} \hat{c}_{1+\sigma+i}^{(1)} B_{\ell_0+R+\sigma+i}(\cdot) \\ &= \sum_{i=0}^{\sigma+R-1} \mu_{\ell_0+i} B_{\ell_0+i}(\cdot) + \sum_{i=0}^{R-\sigma} \mu_{\ell_1+\sigma+i} B_{\ell_1+R+\sigma+i} \end{aligned}$$

- For the last interval $[t_{N-1}, t_N]$, the index $\ell_n = \ell_{N-1} = R(N-3)$, there are exactly $2R+1$ splines active in $\{\theta_{\ell_{(N-1)}}, \dots, \theta_{\ell_{(N-1)+2R+1}}\}$.
  - If $\sigma = 0$

$$
\begin{aligned}
s|_{I_{N-1}} &= \sum_{i=0}^{2R} \hat{c}_{1+i}^{(N-1)} B_{\ell_{N-1}+i}(\cdot) \\
&= \sum_{i=0}^{2R} \mu_{\ell_{N-1}+i} B_{\ell_{N-1}+i}(\cdot)
\end{aligned}
$$

  - If $\sigma = R+1$,

$$
\begin{aligned}
s|_{I_{N-1}} &= \sum_{i=0}^{R-1} \left(\hat{c}_{1+i}^{(N-1)} B_{\ell_{N-1}+i}(\cdot) + \hat{c}_{1+i}^{(N-2)} B_{\ell_{(N-1)}-R+i}(\cdot) \right) + \hat{c}_{1}^{(N-3)} B_{\ell_{(N-1)}-2R}(\cdot) \\
&= \sum_{i=0}^{R-1} (\mu_{\ell_{N-1}+i} B_{\ell_{N-1}+i}(\cdot) + \mu_{\ell_{(N-2)}+i} B_{\ell_{(N-2)}+i}(\cdot)) + \mu_{\ell_{(N-3)}} B_{\ell_{(N-3)}}(\cdot)
\end{aligned}
$$

  - If $0 < \sigma < R+1$

$$
\begin{aligned}
s|_{I_{N-1}} &= \sum_{i=0}^{2R-\sigma} \hat{c}_{1+i+\sigma}^{(N-1)} B_{\ell_{N-1}+i+\sigma}(\cdot) + \sum_{i=0}^{R-\sigma} \hat{c}_{1+\sigma+i}^{(N-2)} B_{\ell_{N-1}-R+\sigma+i}(\cdot) \\
&= \sum_{i=0}^{2R-\sigma} \mu_{\ell_{N-1}+i+\sigma} B_{\ell_{N-1}+i+\sigma}(\cdot) + \sum_{i=0}^{R-\sigma} \mu_{\ell_{N-1}+\sigma+i} B_{\ell_{N-1}-R+\sigma+i}(\cdot)
\end{aligned}
$$

- For the intermediate intervals I_n with $n = 1, \dots, N-2$, and $0 \le \sigma \le R+1$,

$$
\begin{aligned}
s|_{I_n} &= \sum_{i=0}^{R-1} \left(\hat{c}_{1+\sigma+i}^{(n_1)} B_{\ell_n+\sigma+i}(\cdot) + \hat{c}_{1+\sigma+i}^{(n_1+1)} B_{\ell_n+R+\sigma+i}(\cdot) \right) + \hat{c}_{1+\sigma}^{(n_2-1)} B_{\ell_n+2R+\sigma}(\cdot) \\
&= \sum_{i=0}^{R-1} (\mu_{\ell_n+\sigma+i} B_{\ell_n+\sigma+i}(\cdot) + \mu_{\ell_n+R+\sigma+i} B_{\ell_n+R+\sigma+i}(\cdot)) + \mu_{\ell_n+2R+\sigma} B_{\ell_n+2R+\sigma}(\cdot). \quad \square
\end{aligned}
$$

Lemma 2. *If there exist two positive constants η_1, η_2 with $\eta_1 \le 1 \le \eta_2$ such that,*

$$
\eta_1 \le \frac{h_n}{h_{n+1}} \le \eta_2, \quad n = 1, \dots, N-1,
$$

then there exist positive constants $\Gamma^{(j)}$ depending only on η_1, η_2 and on the integer R such that

$$
\|\boldsymbol{\gamma}_{n,i}^{(j)}\|_\infty \le \Gamma^{(j)}, \quad 0 \le n \le N-1, \ 1 \le i \le 2R+1, \quad 0 \le j \le R.
$$

Proof. The proof is analogous to the proof of Lemma 3 in [16]. We reformulate the same steps according to the new notation for multiple inner knots. The entries of each block matrix $A_j^{(n)}$ in (4) depend on the successive ratios h_n/h_{n+1} and, as it is proven in Proposition 1 in [16], the matrix $W^{(n)}$ in (6) is non singular. Therefore, the vectors $\boldsymbol{\gamma}_{n,i}^{(j)}$ continuously depend on the ratios between successive mesh sizes. Hence, since $\|\boldsymbol{\gamma}_{n,i}^{(j)}\|_\infty$ is a continuous function of all the involved ratios that are all bounded, the thesis of the Lemma holds true. $\square$

Corollary 2. *Let $g \in C^R[a, b]$. Then, for $0 \le n \le N-1$ and for $0 \le r \le R$ there holds*

$$
\| \left(Q_R^{(HB)}(g) \right)^{(r)} \|_{\infty, I_n} \le \frac{C_{r,T}}{\hat{h}_n^r} \sum_{j=0}^{R} \hat{h}_n^j \left(\Gamma^{(j)} \|g^{(j)}\|_{\infty, [t_{n_1}, t_{n_2}]} \right),
$$

where $\hat{h}_n := \max_{n_1 \le n \le n_2} h_n$; n_1, n_2 are defined in Lemma 1, and $C_{r,T}$ is a suitable positive constant depending on r and on the ratios between consecutive mesh sizes, with $C_{0,T} = 1$.

Proof. From Lemma 1, the coefficients of the restriction to every subinterval depend on three neighboring intervals, identified by n_1 and n_2. For $r = 0$ the proof can be easily derived from (8). For $r > 0$, it can be derived by using the recursive derivative formulas for the B-spline basis [6]. □

Theorem 1. *Let us assume the hypotheses in Lemma 2 for the distribution of the breakpoints. If $y \in C^{2R+1}[a,b]$, then the r-th derivative of the approximation error of the quasi-interpolant $Q_R^{(HB)}(y)$ satisfies the following inequality,*

$$\| \left(y - Q_R^{(HB)}(y)\right)^{(r)} \|_\infty \leq L\, h^{2R+1-r}\, \|y^{(2R+1)}\|_\infty\,, \quad r = 0,\dots,R, \tag{11}$$

where L is a suitable positive constant depending on R, r and on the positive quantities η_1 and η_2 introduced in Lemma 2.

Proof. The following proof is a generalization of the proof of Theorem 2 in [15] in the case of multiple inner knots.

Let us consider the r-th derivative of the error in the interval $[t_n, t_{n+1}]$, with $0 \leq n \leq N-1$ and for the sake of brevity let us use the notation $\|\cdot\|_n := \|\cdot\|_{\infty,[t_n,t_{n+1}]}$. We have that, if p belongs to the space Π_{2R} of all polynomials with degree less than or equal to $2R$, then the following bounds can be derived:

$$\begin{aligned} \| \left(y - Q_R^{(HB)}(y)\right)^{(r)} \|_n &= \|(y-p)^{(r)} - (Q_R^{(HB)}(y) - p)^{(r)}\|_n \\ &\leq \|(y-p)^{(r)}\|_n + \|(Q_R^{(HB)}(y) - p)^{(r)}\|_n \\ &= \|(y-p)^{(r)}\|_n + \| \left(Q_R^{(HB)}(y-p)\right)^{(r)} \|_n. \end{aligned}$$

Then from Corollary 2 we get the following inequality,

$$\| \left(Q_R^{(HB)}(y-p)\right)^{(r)} \|_n \leq \frac{C_{r,T}}{\hat{h}_n^r} \sum_{j=0}^{R} \hat{h}_n^j \left(\Gamma^{(j)} \|(y-p)^{(j)}\|_{\infty,[t_{n_1},t_{n_2}]}\right).$$

Let p be the Taylor expansion of order $2R$ of $y|_{[t_n,t_{n+1}]}$ about the point t_n, then

$$\|(y-p)^{(r)}\|_n \leq h_n^{2R+1-r} \rho_{2R,r} \|y^{(2R+1)}\|_n,$$

where $\rho_{2R,r}$ is a positive constant depending only on R and r, with $r = 0,\dots,R$.

Using the above bound, the following local error estimate can be computed,

$$\|(y - Q_R^{(HB)}(y))^{(r)}\|_n \leq \hat{h}_n^{2R+1-r} \left[\rho_{2R,r} + C_{r,T} \sum_{j=0}^{R} \rho_{2R,j} \Gamma^{(j)} \right] \|y^{(2R+1)}\|_{\infty,[t_{n_1},t_{n_2}]}.$$

Setting

$$L := \rho_{2R,r} + C_{r,T} \sum_{j=0}^{R} \rho_{2R,j} \Gamma^{(j)},$$

the thesis descends. □

Chosen an integer d ranging between R and $2R$, the required derivative values can be approximated with finite differences formulas as follows

$$\hat{y}_n^{(j)} = y_n^{(j)} + O(h^{d+1-j}), \quad j = 0,\dots,R.$$

The quasi-interpolant $\hat{Q}_R^{(HB)}(y)$ computed using $\hat{y}_n^{(j)}$ instead of $y_n^{(j)}$ satisfies the following Proposition that can be easily derived from Theorem 1.

Proposition 2. *Let $y \in C^{d+1}[a,b]$, with $R \leq d \leq 2R$. Then, under the assumptions of Lemma 2, it holds*

$$\| \left(\hat{Q}_R^{(HB)}(y) - y\right)^{(r)} \| \leq L_r h^{d+1-r} \|y^{(d+1)}\| \quad r = 0,\dots,R.$$

Remark 1. When $\sigma = R+1$ and the required function and derivatives input data are obtained from the R-th HO method the spline s defined via (10) is exactly the collocating spline introduced in [16]. In such case, we observe that its convergence order reduces to $O(h^{2R})$ since approximated input data are used.

In the next Subsection, some numerical experiments are presented to give evidence of the results proved in Theorem 1. While, other tests with approximated derivatives are introduced in Section 5 that give experimental evidence for Proposition 2.

Table 1
Test 1: Approximation error and convergence order for $R = 2$. The values of σ are chosen as $R+1$, 0, $M := \lfloor (R+1)/2 \rfloor$ and $M+1$.

N	$\sigma = 3$	ord	$\sigma = 0$	ord	$\sigma = 1$	ord	$\sigma = 2$	ord
16	4.4×10^{-02}	*	4.7×10^{-02}	*	1.9×10^{-02}	*	1.9×10^{-02}	*
32	9.5×10^{-04}	5.5	9.5×10^{-04}	5.6	5.0×10^{-04}	5.2	5.0×10^{-04}	5.2
64	2.0×10^{-05}	5.6	2.0×10^{-05}	5.6	1.5×10^{-05}	5.0	1.5×10^{-05}	5.0
128	4.8×10^{-07}	5.4	5.3×10^{-07}	5.3	4.7×10^{-07}	5.0	4.7×10^{-07}	5.0
256	1.5×10^{-08}	5.0	1.5×10^{-08}	5.1	1.5×10^{-08}	5.0	1.5×10^{-08}	5.0
512	4.0×10^{-10}	5.2	4.0×10^{-10}	5.3	3.8×10^{-10}	5.3	3.8×10^{-10}	5.3

Table 2
Test 1: Approximation error and convergence order for $R = 3$. The values of σ are chosen as $R+1$, 0, $M := \lfloor (R+1)/2 \rfloor$ and $M+1$.

N	$\sigma = 4$	ord	$\sigma = 0$	ord	$\sigma = 2$	ord	$\sigma = 3$	ord
16	1.2×10^{-03}	*	1.3×10^{-03}	*	4.8×10^{-04}	*	5.7×10^{-04}	*
32	6.8×10^{-06}	7.4	6.5×10^{-06}	7.7	3.6×10^{-06}	7.0	4.0×10^{-06}	7.2
64	3.6×10^{-08}	7.6	3.6×10^{-08}	7.5	2.8×10^{-08}	7.0	2.9×10^{-08}	7.1
128	2.4×10^{-10}	7.3	2.4×10^{-10}	7.2	2.2×10^{-10}	7.0	2.3×10^{-10}	7.0
256	1.8×10^{-12}	7.0	1.9×10^{-12}	7.0	1.8×10^{-12}	6.9	1.8×10^{-12}	7.0
512	1.5×10^{-14}	7.0	1.4×10^{-14}	7.0	1.4×10^{-14}	7.0	1.4×10^{-14}	7.0

Table 3
Test 2: Approximation error and convergence order for $R = 2$. The values of σ are chosen as $R+1$, 0, $M := \lfloor (R+1)/2 \rfloor$ and $M+1$.

N	$\sigma = 3$	ord	$\sigma = 0$	ord	$\sigma = 1$	ord	$\sigma = 2$	ord
16	4.6×10^{-03}	*	4.2×10^{-03}	*	3.3×10^{-03}	*	3.3×10^{-03}	*
32	1.8×10^{-04}	4.7	1.8×10^{-04}	4.5	1.5×10^{-04}	4.5	1.5×10^{-04}	4.5
64	5.8×10^{-06}	4.9	6.1×10^{-06}	4.9	5.3×10^{-06}	4.8	5.3×10^{-06}	4.8
128	1.8×10^{-07}	5.0	1.9×10^{-07}	5.0	1.8×10^{-07}	4.9	1.8×10^{-07}	4.9
256	5.8×10^{-09}	5.0	6.0×10^{-09}	5.0	5.7×10^{-09}	5.0	5.7×10^{-09}	5.0
512	1.2×10^{-10}	5.6	1.2×10^{-10}	5.7	1.1×10^{-10}	5.8	1.1×10^{-10}	5.8

4.1. Numerical experiments with exact data

To show the behavior of the Hermite-Birkhoff quasi-interpolant the following two functions will be approximated by the constructed operator $Q_R^{(HB)}$:

Test 1:

$$y(t) = e^{-t} \sin(5\pi t)\,, \ t \in [-1\,,\, 1]$$

Test 2:

$$y(t) = \frac{e^{-t/\sqrt{\varepsilon}} - e^{(t-2)/\sqrt{\varepsilon}}}{1 - e^{-2/\sqrt{\varepsilon}}}\,, \ t \in [0\,,\, 1]\,, \quad \varepsilon = 10^{-3}$$

The numerical experiments have been performed by running the routine implemented in Matlab R2021b on a MacBook pro, 2,9 GHz Intel Core i7 quad-core. To show the convergence behavior, uniform step-size has been used and the maximum absolute error has been computed. The results for different values of σ and R are shown in Tables 1, 2 for Test 1 and in Tables 3, 4 for Test 2. The compared values of σ are $\sigma = 0, \sigma = R+1$ and fixing $M = \lfloor (R+1)/2 \rfloor, \sigma = M, \sigma = M+1$. The Tables show that the best results are obtained using $\sigma = M$ for both the examples. This is justified by the fact that when the whole interval is considered, then it is better to symmetrically split the additional conditions.

5. Application to dense output for Runge Kutta collocation schemes

Let us consider the solution of problem (1) using one-step schemes. Given $\mathbf{u}_n$, an approximation of $\mathbf{y}$ at x_n, then an approximation $\mathbf{u}_{n+1}$ of $\mathbf{y}$ at x_{n+1} is computed. Referring for simplicity to the scalar case, usually a continuous extension of the discrete solution is a piecewise polynomial $p(x)$ that accurately approximates $y(x)$ in all the sub-intervals $[x_n, x_{n+1}]$.

For the solution of BVPs implicit methods are used and since there is no preferred direction, symmetric formulas are the most suited. The class of two-points BVPs specifies a solution by means of known values of its components at the two ends of the interval. Therefore, the formula must be evaluated at all mesh points simultaneously, i.e., a large system of nonlinear equations needs to be solved. In the nonlinear cases, the solution requires a suitable iterative procedure and if an

Table 4
Test 2: Approximation error and convergence order for $R=3$. The values of σ are chosen as $R+1$, 0, $M := \lfloor (R+1)/2 \rfloor$ and $M+1$.

N	$\sigma=4$	ord	$\sigma=0$	ord	$\sigma=2$	ord	$\sigma=3$	ord
16	1.3×10^{-04}	*	1.2×10^{-04}	*	9.0×10^{-05}	*	1.0×10^{-06}	*
32	1.3×10^{-06}	6.6	1.4×10^{-06}	6.5	1.1×10^{-06}	6.4	1.2×10^{-06}	6.4
64	1.2×10^{-08}	6.9	1.2×10^{-08}	6.8	1.0×10^{-08}	6.7	1.1×10^{-08}	6.7
128	9.6×10^{-11}	6.9	9.7×10^{-11}	7.0	8.9×10^{-11}	6.9	9.2×10^{-11}	6.9
256	7.6×10^{-13}	7.0	7.7×10^{-13}	7.0	7.3×10^{-13}	6.9	7.4×10^{-13}	7.0
512	3.7×10^{-15}	7.7	3.7×10^{-15}	7.7	3.7×10^{-15}	7.7	3.7×10^{-15}	7.7

approximation to $y(x)$ is not satisfactory, the mesh is refined and a larger system of algebraic equations is solved. In this case, a continuous extension is fundamental as it is used to generate starting guesses for the iterative procedure.

For the solutions of BVPs and for the solution of conservative problems, some of the most used collocation Runge-Kutta schemes are the symmetric Gauss-Lobatto or Gauss-Legendre formulas.

Note that a collocation scheme finds the solution in the subinterval $[x_n, x_{n+1}]$ by defining a set of collocation points $x_n + c_i h, i = 1, \ldots, k$, where k denotes the number of stages of the scheme. The collocation polynomial $p_n(x)$, restriction of p to $[x_n, x_{n+1}]$, satisfies $p_n(x_n) = u_n$ and the collocation equations:

$$p_n'(x_n + c_i h) = f(x_n + c_i h, p_n(x_n + c_i h)), \qquad i = 1, \ldots, k. \tag{12}$$

If we are solving an initial value problem the new approximation at x_{n+1} is given by $u_{n+1} = p_n(x_{n+1})$.

Note that, it has been proven that a collocation method is equivalent to a Runge-Kutta scheme [2,8].

For superconvergent methods, like Gauss-Lobatto and Gauss-Legendre, the collocation piecewise polynomial has a smaller order of convergence than the numerical solution at the mesh points $x_n, n = 0, \ldots, N+1$, see [8]. Therefore, for these schemes, it is of interest to look for a different continuous extension sharing with the numerical solution the convergence order.

Observe that superconvergent interpolants for collocation solution of BVPs have been analyzed for example in [1,7]. These interpolants are based on the construction of Continuous RK Formulas or Continuous Parameterized Implicit RK, adding additional stages. Here a quasi-interpolation superconvergence scheme with a totally different approach, based on the approximation of the derivatives, has been considered.

5.1. Gauss-Lobatto

In this subsection the new quasi-interpolant $\hat{Q}_R^{(HO)}$ is applied to construct a continuous extension for the Gauss-Lobatto schemes. The resulting continuous spline has the same order of convergence of the numerical scheme but it quasi-interpolates the values at the mesh points.

The collocation nodes of the Gauss-Lobatto methods are derived by choosing c_i as the zeros of the Lobatto polynomial and include both the endpoints. The Matlab codes `bvp4c` and `bvp5c` use Gauss-Lobatto methods [11,12].

In particular, the order 6 Gauss-Lobatto formula is considered with $k = 4$, and the collocation points are both the end-points of the step and the points $x_n + h(5-\sqrt{5})/10, x_n + h(5+\sqrt{5})/10$. Also this scheme is implemented in the code `bvp5c`.

Although the order of the RK formula is 6, the associated collocation piecewise polynomial p which has degree 4 and is C^1 smooth, is just uniformly of order 5.

In order to obtain a $\hat{Q}_3^{(HO)}$ quasi-interpolation scheme of order 6, it is necessary to choose the spline breakpoint t_n different from x_n, for $n = 0, \ldots, N$, and specifically $t_n = (x_n + x_{n+1})/2$ is set. Then, suitable approximations of $y^{(j)}(t_n)$, for $j = 0, \ldots, 3$ at least of order $6-j$ are needed. Indeed, this is facilitated from having already available in a neighborhood of t_n the polynomial approximation p_n. At each step of the Runge-Kutta scheme, the values of p_n and of its derivatives at the four collocation points in each I_n are available. Since the collocation points are symmetric with respect to t_n, an approximation $\hat{y}_n''$ of order 4 of $y''(t_n)$ and $\hat{y}_n^{iv}$ of order 2 of $y^{iv}(t_n)$ can be computed.

Since the available approximations of y at the mesh points are of order 6, due to superconvergence, by using Taylor expansion, it can be derived that:

$$\hat{y}_n = \frac{(u_n + u_{n+1})}{2} - \frac{h^2}{2^2 \cdot 2!}\hat{y}_n'' - \frac{h^4}{2^4 \cdot 4!}\hat{y}_n^{iv}$$

is an approximation of order 6 of $y(t_n)$ and $\hat{y}_n' = f(t_n, \hat{y}_n)$ is an approximation of order 6 of $y'(t_n)$. Using $\hat{y}_n'$ and the values of f at the four collocation points, an order 4 approximation $\hat{y}_n'''$, the third derivative of $y(t_n)$, can be also computed.

Thereafter, the values $\hat{y}_n = y(t_n) + O(h^6)$, $\hat{y}_n' = y'(t_n) + O(h^6)$, $\hat{y}_n'' = y''(t_n) + O(h^4)$, $\hat{y}_n''' = y'''(t_n) + O(h^4)$ are available. Finally, the $\hat{Q}_R^{(HO)}$ quasi-interpolant spline satisfying Lemma 2, i.e., with order of convergence 6, can be constructed.

Observe that, with respect to the interpolant used in the code `bvp5c`, this quasi-interpolant requires one more function evaluation at t_n. However, this value is necessary in the code for computing the residual, so its usage does not imply any additional cost. The first and last intervals, $[x_0, t_0]$ and $[t_N, x_{N+1}]$, need a different way to compute the continuous extension. One way to work with these two intervals is to compute the Taylor polynomial backward and forward, respectively about t_0 and t_N. To have the correct order of convergence, the polynomial needs the computation of an approximation of the fifth derivative of at least order 1. An order 2 approximation of $y^v(t_n)$ is hence computed by using a five point central difference by using the already available function evaluations.

5.2. Gauss-Legendre

The collocation nodes of the Gauss-Legendre are derived by choosing c_i as the zeros of the Legendre polynomial and do not include the endpoints of the step. Observe that the Gauss-Legendre formula with k stages is of order $2k$, but the collocation polynomial has just uniform order $k+1$.

The codes `colsys/colnew/colmod` use Gauss-Legendre formulas of high order [2,5]. Also the code `bvpsuite` [3] is based on collocation and it is possible to choose as collocation points the zeros of the Legendre polynomial. In all the cited codes the adopted continuous extension is the collocation piecewise polynomial p which in this case is just C^0 smooth.

In this subsection we consider the order 4 and 6 of the Gauss-Legendre formula with $k = 2, 3$ collocation points.

As for the Gauss-Lobatto scheme, the spline breakpoints are chosen as the mid-points of each subintervals t_n.

The order 4 method which has two stages symmetric with respect to the midpoints $x_n + (1/2 - \sqrt{3}/6)h_n$, $x_n + (1/2 + \sqrt{3}/6)h_n$, is combined with our QI scheme with $R = 2$. Then, suitable approximations of $y^{(j)}(t_n)$, for $j = 0, \ldots, 2$ at least of order $4 - j$ are needed.

The value of the collocation polynomial computed at the mid-point is of order 4, therefore, the value of $f(t_n, y(t_n))$ is of order 4 as well. For the second derivative, an approximation of the first derivative of f using central differences is of order 2. So, with one additional function evaluation, a C^1 spline approximation of the continuous solution can be obtained by applying the $\hat{Q}_2^{(HO)}$ operator to breakpoints, approximated as explained.

For the order 6 method, $R = 3$ is adopted. In this case, there are three internal stages and the piecewise collocation polynomial is of degree 3. The internal stages are: $x_n + h_n c_1$; $x_n + h_n c_2$; $x_n + h_n c_3$ with $c_1 = (1/2 - \sqrt{15}/10)$; $c_2 = 1/2$; $c_3 = (1/2 + \sqrt{15}/10)$. The order of the available approximation of the function and of its first derivative at the collocation points is just 4. Thus, better approximations are required at the mid point, which in this case is also a collocation point. The following procedure is adopted: two additional function evaluations at the points $b_{n-} = x_n + h_n(1/2 - \sqrt{5}/10)$; $b_{n+} = x_n + h_n(1/2 + \sqrt{5}/10)$ are needed. These are two special points since it can be proven that the collocation polynomial has superconvergence order 5.

At this step, the information available at the collocation points and at these 2 additional points, allows us to use a five points central finite differences with variable stepsize to compute an approximation $\hat{y}_n''$ of order 4 of the second derivative and analogously an approximation of order 2 of $\hat{y}_n'''$ and $\hat{y}_n^{iv}$.

As for the Gauss-Lobatto scheme we have:

$$\hat{y}_n = \frac{(u_n + u_{n+1})}{2} - \frac{h^2}{2^2 \cdot 2!}\hat{y}_n'' - \frac{h^4}{2^4 \cdot 4!}\hat{y}_n^{iv}.$$

This approximation is now of order 6 and the value of $f(t_n, \hat{y}_n)$ is of order 6 as well. To compute and approximation of order 4 of the third derivative $\tilde{y}_n'''$, a better approximation $\hat{y}_{n1}, \hat{y}_{n3}$ at the internal stages $x_n + hc_1$, $x_n + h_n c_3$ is necessary. This can be done by using a Taylor expansion about t_n, since at these points the values of the needed derivatives to compute an order 4 approximation are known,

$$\hat{y}_{ni} = \hat{y}_n + h_n(c_i - 1/2) f(t_n, \hat{y}_n) + \frac{h_n^2 (c_i - 1/2)^2}{2}\hat{y}_n'' + \\ \frac{h_n^3 (c_i - 1/2)^3}{3!}\tilde{y}_n''' + \frac{h_n^4 (c_i - 1/2)^4}{4!}\hat{y}_n^{iv}$$

The next step is to compute an approximation of order 4 of $y'''(t_n)$ using 5 points finite differences with the values of $f(t_n, \hat{y}_n)$, $f(b_{n-}, p_n(b_{n-}))$, $f(b_{n+}, p_n(b_{n+}))$, $f(x_n + hc_1, \hat{y}_{n1})$, $f(x_n + hc_3, \hat{y}_{n3})$. We call this approximation $\hat{y}_n'''$.

Now we have available all the derivatives of correct order to compute an order 6 $\hat{Q}_3^{(HO)}$ spline quasi-interpolant: the values $\hat{y}_n = y(t_n) + O(h^6)$, $\hat{y}_n' = y'(t_n) + O(h^6)$, $\hat{y}_n'' = y''(t_n) + O(h^4)$, $\hat{y}_n''' = y'''(t_n) + O(h^4)$.

This continuous extension requires 5 more function evaluations on each mesh interval. It is important to note that in the non-linear case, they are computed once and this does not increase the computational cost required to the convergence of the nonlinear iteration procedure. For the extreme intervals $[t_0, x_0]$, $[t_N, x_{N+1}]$ we adopt the same procedure previously described.

5.3. Numerical experiments for boundary value problems

In this subsection we report some numerical experiments related to boundary value problems to show the behavior of the spline QI operator for both the Gauss-Lobatto and the Gauss-Legendre Runge-Kutta schemes. We use as test case the linear boundary value problem `bvpT1`:

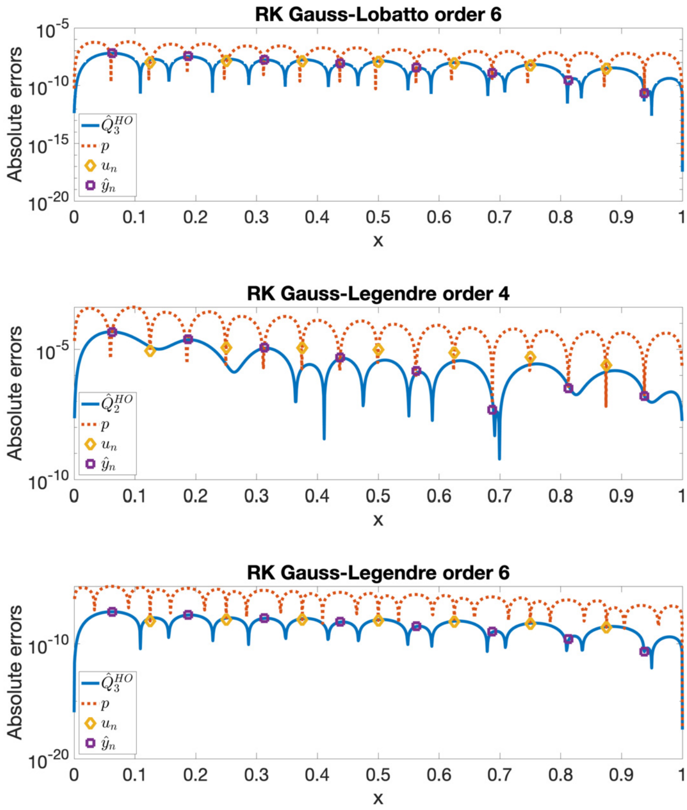

Fig. 1. Numerical Results for bvpT1 with Runge-Kutta collocation schemes: absolute errors of the $\hat{Q}_R^{(HO)}$ (blue solid line) and of piecewise collocation polynomial (red dotted lines) in the interval $[0, 1]$; absolute error values at the RK mesh points (yellow diamonds) and at the mid-points of every subinterval (purple squares). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\varepsilon y'' = y,$$
$$y(0) = 1,\ y(1) = 0,$$

with exact solution:

$$\frac{e^{-x/\sqrt{\varepsilon}} - e^{(x-2)/\sqrt{\varepsilon}}}{1 - e^{-2/\sqrt{\varepsilon}}},$$

where in particular we show the results for $\varepsilon = 0.1$ and $\sigma = M$. The problem is rewritten as a first order system with two components $(y, y')^T$. This is one of the benchmark problems described in the web site "Testset for BVP solvers" [14].

Fig. 1 shows the absolute error of the continuous extension computed using as numerical scheme the Gauss-Lobatto of order 6, the Gauss-Legendre of order 4 and 6, respectively, beside the error at the collocation points and the error at the mesh points.

The picture clearly shows that the absolute error of the quasi-interpolant (blue solid line) is smaller than the absolute error of the collocation piecewise polynomial (red dotted line) and it is comparable to the error at the mesh points (yellow diamonds).

Table 5 shows the error with different stepsizes $h = 1/(N+1)$ and the approximate order of convergence for both the quasi-interpolanting spline and the collocation piecewise polynomial, that clearly respects the theoretical properties. For the Gauss-Lobatto, the collocation piecewise polynomial given by the bvp5c code is used.

5.4. Numerical experiments for initial value problems

The BSHO schemes have been successfully used in [16] for the solution of Initial Value Hamiltonian problems. A class of methods that has nice properties for this kind of differential equation is the one of Gauss-Legendre scheme, that has been used in [9,10,16], to compare the behaviors of the proposed schemes. Here we report some numerical experiments using the classical Kepler problem describing the motion of two bodies subject to Newton's law of gravitation. As it is well-known, the problem is a completely integrable Hamiltonian dynamical system with two degrees of freedom (see, for example, [10]). If the origin of the coordinates system is set on one of the two bodies, the motion of the other body is described by the

Table 5
Example `bvpT1` solved with Gauss-Lobatto and Gauss-Legendre RK. Errors for function and first derivative and estimated order of convergence of the spline quasi-interpolant (err_s, ord) and of the collocation piecewise polynomial (err_p, ord) with different stepsizes.

Gauss-Lobatto order 6				
N+1	err_s, $\sigma=2$, $R=3$	ord	err_p	ord
8	6.3×10^{-08}	*	6.2×10^{-07}	*
16	1.1×10^{-09}	5.8	2.1×10^{-08}	4.9
32	1.9×10^{-11}	5.9	6.8×10^{-10}	4.9
64	3.0×10^{-13}	5.9	2.2×10^{-11}	5.0
128	4.9×10^{-15}	6.0	6.8×10^{-13}	5.0
Gauss-Legendre order 4				
N+1	err_s, $\sigma=1$, $R=2$	ord	err_p	ord
8	4.7×10^{-05}	*	4.3×10^{-04}	*
16	3.4×10^{-06}	3.8	5.8×10^{-05}	2.9
32	2.3×10^{-07}	3.9	7.5×10^{-06}	2.9
64	1.5×10^{-08}	3.9	9.5×10^{-07}	3.0
128	9.5×10^{-10}	4.0	1.2×10^{-07}	3.0
Gauss-Legendre order 6				
N+1	err_s, $\sigma=2$, $R=3$	ord	err_p	ord
8	6.3×10^{-08}	*	1.0×10^{-05}	*
16	1.1×10^{-09}	5.8	7.2×10^{-07}	3.9
32	1.9×10^{-11}	5.9	4.7×10^{-08}	3.9
64	3.0×10^{-13}	5.9	3.0×10^{-09}	4.0
128	4.9×10^{-15}	6.0	1.9×10^{-10}	4.0

Table 6
Absolute errors and estimated order of convergence for the Kepler problem solved with Gauss-Legendre RK and constant stepsize $h=T/(N+1)$.

	$\hat{Q}_2^{(HO)}$ order 4				$\hat{Q}_3^{(HO)}$ order 6			
$N+1$	err_s	ord	$err_{s'}$	ord	err_s	ord	$err_{s'}$	ord
20	2.5×10^{-01}	*	7.1×10^{-01}	*	2.2×10^{-02}	*	7.4×10^{-02}	*
40	2.2×10^{-02}	3.5	7.2×10^{-02}	3.3	3.3×10^{-04}	6.0	1.2×10^{-03}	6.0
80	1.7×10^{-03}	3.7	5.3×10^{-03}	3.8	5.4×10^{-06}	5.9	2.0×10^{-05}	5.8
160	1.2×10^{-04}	3.8	3.6×10^{-04}	3.9	9.8×10^{-08}	5.8	3.6×10^{-07}	5.8
320	7.9×10^{-06}	3.9	3.0×10^{-05}	3.6	1.6×10^{-09}	5.9	7.4×10^{-09}	5.6
640	5.0×10^{-07}	4.0	2.5×10^{-06}	3.6	2.7×10^{-11}	5.9	1.8×10^{-10}	5.4

Hamiltonian function $H(q_1,q_2,p_1,p_2)=(p_1^2+p_2^2)/2-(q_1^2+q_2^2)^{-1/2}$ which is an ellipse in the q_1-q_2 plane with eccentricity e and periodicity $T=2\pi$.

Taking as initial conditions

$$q_1(0)=1-e,\quad q_2(0)=0,\quad p_1(0)=0,\quad p_2(0)=\sqrt{\frac{1+e}{1-e}},$$

using $\mathbf{y}=(q_1,q_2,p_1,p_2)^T$ the following differential equation can be written

$$\mathbf{y}'=\mathbf{f}(\mathbf{y})=\begin{pmatrix} -p_1\\ -p_2\\ \dfrac{-q_1}{(q_1^2+q_2^2)\sqrt{q_1^2+yq_2^2}}\\ \dfrac{-q_2}{(q_1^2+q_2^2)\sqrt{q_1^2+q_2^2}}\end{pmatrix}. \tag{13}$$

The experiments are performed with $e=0.5$ and constant stepsize $h=T/(N+1)$. The problem is integrated in four periods and the errors in the spline quasi-interpolants are computed by evaluating a more accurate solution with a finer mesh obtained with a constant stepsize $h=T/(4(N+1))$. The absolute error is computed by evaluating the quasi-interpolant at 1000 points in the interval $[0,8\pi]$, see Table 6.

6. Conclusions

In this paper a new class of spline quasi-interpolants generalizing the collocation spline of the Hermite-Obreshkov methods for ODEs has been presented. The introduced QI operator is defined by using the function values and its derivatives at the spline breakpoints, which are also the set of multiple knots. In order to have the maximal convergence behavior, the partition of the interval is not required to be strictly uniform, but it just has to be quasi-uniform. The associated class with approximated derivatives is also described which results to be interesting for the construction of dense output for superconvergent collocation methods for ODEs. Some numerical experiments for Gauss-Lobatto and Gauss-Legendre are then presented. Different schemes for the dense output based on interpolation are known in literature, e.g., [1,7] and some comparisons will be object of future work.

Acknowledgements

All the authors are members of the INdAM GNCS national group. The research of Antonella Falini is funded by PON Ricerca e Innovazione 2014-202 FSE REACT-EU, Azione IV.4 “Dottorati e contratti di ricerca su tematiche dell’innovazione” CUP H95F21001230006. Alessandra Sestini and Francesca Mazzia acknowledge the contribution of the National Recovery and Resilience Plan, Mission 4 Component 2 - Investment 1.4 - NATIONAL CENTER FOR HPC, BIG DATA AND QUANTUM COMPUTING - funded by the European Union - NextGenerationEU - (CUP B83C22002830001; CUP H93C22000450007). Finally, Antonella Falini and Francesca Mazzia thank the GNCS for its valuable support under the INDAM-GNCS project CUP_E55F22000270001.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.apnum.2023.07.023.

References

[1] M. Adams, J. Finden, P. Phoncharon, P.H. Muir, Superconvergent interpolants for Gaussian collocation solutions of mixed order BVODE systems, AIMS Math. 7 (4) (2022) 5634–5661, All Open Access, Gold Open Access.

[2] Uri M. Ascher, Robert M.M. Mattheij, Robert D. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Classics in Applied Mathematics, vol. 13, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.

[3] W. Auzinger, K. Burdeos, M. Fallahpour, O. Koch, R. Mendoza, E. Weinmüller, A numerical continuation method for parameter-dependent boundary value problems using bvpsuite 2.0, J. Numer. Anal. Ind. Appl. Math. 16 (1–2) (2022) 1–13.

[4] E. Bertolazzi, A. Falini, F. Mazzia, The object oriented C++ library QIBSH++ for Hermite spline quasi interpolation, arXiv preprint, arXiv:2208.03260, 2022.

[5] J.R. Cash, G. Moore, R. Wright, An automatic continuation strategy for the solution of singularly perturbed nonlinear boundary value problems, ACM Trans. Math. Softw. 27 (2) (2001) 245–266.

[6] C. de Boor, A Practical Guide to Splines, Applied Mathematical Sciences, vol. 27, Springer-Verlag, New York, 2001.

[7] W.H. Enright, P.H. Muir, Superconvergent interpolants for the collocation solution of boundary value ordinary differential equations, SIAM J. Sci. Comput. 21 (1) (1999) 227–254.

[8] G. Hairer, E. Norsett, G. Wanner, Solving Ordinary Differential Equations. I, Springer Series in Computational Mathematics, vol. 8, Springer-Verlag, Berlin, 1993, Nonstiff Problems.

[9] F. Iavernaro, F. Mazzia, A fourth order symplectic and conjugate-symplectic extension of the midpoint and trapezoidal methods, Mathematics 9 (10) (2021) 1103.

[10] F. Iavernaro, F. Mazzia, M.S. Mukhametzhanov, Ya.D. Sergeyev, Conjugate-symplecticity properties of Euler–Maclaurin methods and their implementation on the infinity computer, Appl. Numer. Math. 155 (2020) 58–72.

[11] J. Kierzenka, L.F. Shampine, A BVP solver based on residual control and the Matlab pse, ACM Trans. Math. Softw. 27 (3) (2001) 299–316.

[12] J. Kierzenka, L.F. Shampine, A BVP solver that controls residual and error, J. Numer. Anal. Ind. Appl. Math. 3 (1–2) (2008) 27–41.

[13] F. Mazzia, Mesh selection strategies of the code Tom for boundary value problems, Ann. Univ. Ferrara 68 (2) (2022) 555–574.

[14] F. Mazzia, J.R. Cash, A.M. Nagy, et al., Testset for bvp solvers, https://archimede.dm.uniba.it/~bvpsolvers/testsetbvpsolvers/.

[15] F. Mazzia, A. Sestini, The BS class of Hermite spline quasi-interpolants on nonuniform knot distributions, BIT Numer. Math. 49 (3) (2009) 611–628.

[16] F. Mazzia, A. Sestini, On a class of Hermite-Obreshkov one-step methods with continuous spline extension, Axioms 7 (3) (2018) 58.

[17] F. Mazzia, A. Sestini, D. Trigiante, B-spline linear multistep methods and their continuous extensions, SIAM J. Numer. Anal. 44 (5) (2006) 1954–1973 (electronic).

[18] F. Mazzia, A. Sestini, D. Trigiante, BS linear multistep methods on non-uniform meshes, J. Numer. Anal. Ind. Appl. Math. 1 (1) (2006) 131–144.

[19] L.F. Shampine, L.O. Jay, in: Björn Engquist (Ed.), Dense Output, in: Encyclopedia of Applied and Computational Mathematics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 339–345.