DOI 10.1007/s11182-021-02260-2

Russian Physics Journal, Vol. 63, No. 11, March, 2021 (Russian Original No. 11, November, 2020)

NEW COLLOCATION INTEGRATOR FOR SOLVING DYNAMIC PROBLEMS. I. THEORETICAL BACKGROUND

V. A. Avdyushev

UDC 519.62:531

A new collocation integrator with Lobatto spacings is proposed for numerical solving mixed systems of first- and second-order differential equations for dynamic problems. The general theory of collocation integrators is outlined from which the basic formulas of the new integrator are derived.

Keywords: numerical integrators, collocation methods, ordinary differential equations, dynamical systems.

INTRODUCTION

Collocation integrators [1–4] are simple, graceful, convenient, and at the same time powerful tools for solving differential equations of dynamics. The idea of the collocation integrators is rather transparent, and knowledge of other integrators [3], for example, Runge–Kutta, Gragg–Bulirsh–Shtoer, or Adams integrators are not required to understand it. It is suffice to know only what are differential equations, integration, and interpolation. Therefore, it is well justified to consider the collocation integrators as a whole as an independent class, the most known representatives of which are the implicit Runge–Kutta collocation integrators.

A remarkable feature of the collocation integrators is that their theoretical basis, as well as software implementation, is universal for an arbitrary order [5]. Practically, the order is defined by spacing within a step, namely, by the number and specificity of distribution of nodal values in terms of which all other integrator constants are expressed. In addition, with Gaussian Legendre or Lobatto quadrature spacings, the collocation integrators become geometrical [4]: symmetric and orbitally stable,[1] and with Legendre spacings, also symplectic. It should also be noted that, unlike others, the collocation integrators allow one to design easily within each step an approximate analytical solution, which is convenient to use for obtaining results on a dense time grid.

In the present work, a new collocation integrator Lobbie is proposed with Lobatto spacings. Actually, its prototype is the Everhart integrator widely used in dynamical astronomy [6, 7]. More precisely, it is the result of cardinal revision of the predecessor, although the theory, algorithmization, and program code of the new integrator can only conceptually remind of the author's version of the glorified Everhart integrator.

The general theory of collocation integrators with application to solving the differential equations of the first and second orders is briefly outlined. Particular examples, including the Everhart integrator, are given followed by the derivation of the main formulas for the integrator Lobbie and the special features of their software implementation. The procedure Lobbie is described.

[1] This definition with reference to the Runge–Kutta method was introduced in [5].

National Research Tomsk State University, Tomsk, Russia, e-mail: scharmn@mail.ru. Translated from Izvestiya Vysshikh Uchebnykh Zavedenii, Fizika, No. 11, pp. 131–140, November, 2020. Original article submitted April 28, 2020.

1. COLLOCATION METHODS

1.1. First-order differential equations

Let a dynamic state $\boldsymbol{x}$ be described as a function of time t by the first-order vector differential equation

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}) \tag{1}$$

with the known initial dynamic state at time t_0 :

$$\boldsymbol{x}_0 = \boldsymbol{x}(t_0)\,. \tag{2}$$

Here the prime designates the full time derivative and $\boldsymbol{f}$ is the known vector function of time and dynamic state. It is required to determine the dynamic state of the system at time $t_0 + h$:

$$\boldsymbol{x}(t_0 + h)\,, \tag{3}$$

where h is a small parameter (size of the integration step).

We represent the approximate solution of Eq. (1) in the form of a polynomial

$$\boldsymbol{u}(t) \approx \boldsymbol{x}(t)\,, \tag{4}$$

which must precisely satisfy to the equation at some intermediate moments of time $t_i \in [t_0, t_0 + h]$ $(i = 1, \ldots, s)$ (at collocation points)

$$\boldsymbol{u}'(t_i) = \boldsymbol{f}(t_i, \boldsymbol{u}(t_i)) \quad (i = 1, \ldots, s)\,, \tag{5}$$

and to initial condition (2)

$$\boldsymbol{x}_0 = \boldsymbol{u}(t_0)\,. \tag{6}$$

Then solution (3), with allowance for (4), is approximately determined as

$$\boldsymbol{x}_1 = \boldsymbol{u}(t_0 + h) \approx \boldsymbol{x}(t_0 + h)\,. \tag{7}$$

The geometric sense of (5) is that the tangents to the polynomial at the collocation points must coincide (collocate) with the directions of the vector field generated by the function of the differential equation $\boldsymbol{f}$ (Fig. 1). Though the values of the polynomial can differ considerably from the exact solution.

Collocation conditions (5) can be considered as the Lagrange conditions imposed on the derivative of polynomial (4):

$$\boldsymbol{u}'(t_0 + h\tau) \equiv \boldsymbol{p}(\tau)\,, \tag{8}$$

which in this case plays the role of the polynomial interpolant of the function $\boldsymbol{f}$ with respect to the dimensionless variable τ . According to (5), the Lagrange conditions for the nodal values $c_1, \ldots, c_s$ of the dimensionless variable τ can be represented in the form

$$\boldsymbol{p}(c_i) = \boldsymbol{f}_i \equiv \boldsymbol{f}(t_i, \boldsymbol{u}_i), \quad \boldsymbol{u}_i \equiv \boldsymbol{u}(t_i), \quad t_i = t_0 + hc_i \quad (i = 1, \ldots, s)\,. \tag{9}$$

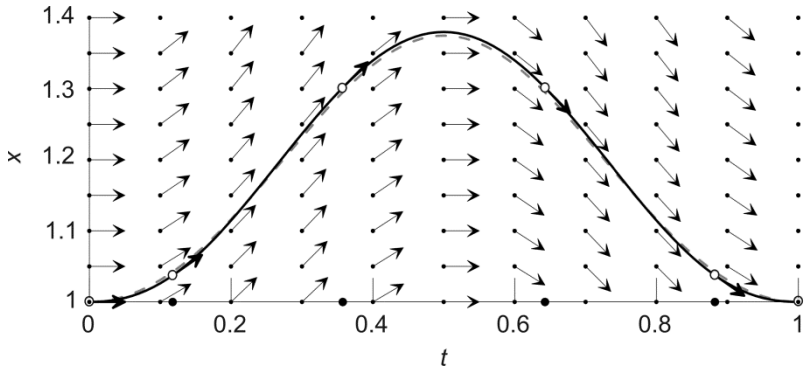

Fig. 1. Collocation conditions for the differential equation $x' = f(t,x) = \sin(2\pi t)x$ with the initial condition $x_0 = x(0) = 1$. The arrows indicate directions of the vector field $(\cos\alpha, \sin\alpha)$ at points (t,x), where $\tan\alpha = f(t,x)$. The black solid curve shows the collocation polynomial u (the approximated solution) with the Lobatto spacings (filled circles) for $s = 6$; the dashed curve shows the exact solution $x = e^{(1-\cos(2\pi t))/2\pi}$; the collocation points for planes (t,x) are indicated by open circles.

Forming the polynomial interpolant $\boldsymbol{p}$ from conditions (9), then integrating relation (8) over τ on the segment $[0,1]$, and taking into account (6) and (7):

$$\int_0^1 \boldsymbol{u}'(t_0 + h\tau)\,\mathrm{d}\tau = \frac{\boldsymbol{u}(t_0 + h\tau)\big|_0^1}{h} = \frac{\boldsymbol{x}_1 - \boldsymbol{x}_0}{h} = \int_0^1 \boldsymbol{p}(\tau)\,\mathrm{d}\tau\,,$$

we obtain the approximate solution

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + h\int_0^1 \boldsymbol{p}(\tau)\,\mathrm{d}\tau\,.$$

The interpolant $\boldsymbol{p}$ (8) is constructed from the intermediate approximate solutions $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s$ (9), that is, $\boldsymbol{p} = \boldsymbol{p}(\tau, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_s)$. Each i th solution is also determined by integrating (8) over τ, but on the segment $[0, c_i]$. Thus, the collocation method of solving differential equation (1) can be represented by the set of formulas

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + h\int_0^1 \boldsymbol{p}(\tau, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_s)\,\mathrm{d}\tau, \quad \boldsymbol{u}_i = \boldsymbol{x}_0 + h\int_0^{c_i} \boldsymbol{p}(\tau, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_s)\,\mathrm{d}\tau \quad (i = 1, \ldots, s)\,. \tag{10}$$

The intermediate solutions in (10) are implicitly expressed through the nonlinear equations. For this reason, all collocation methods are implicit.[2] As a rule, the equations are solved by the fixed-point iteration in the Seidel modifications, that is, refining one by one the intermediate solutions at each iteration step. In fact, the iterative solution of nonlinear equations is the kernel of any collocation integrator, and its efficiency depends in many respects on how successfully organized is the iterative process.

The order p of the method is determined by the number s of collocation points and the specificity of their distribution. The collocation principle allows one to obtain practically any order. So, for any arbitrary distribution

[2] Except the explicit Euler method (of the first order) which is the collocation method with the Radau I spacings for $s = 1$.

$c_1,\ldots,c_s$ (for example, uniform) at least $p = s$ [3]. However, using the nodal values of the Gaussian Legendre, Radau, or Lobatto quadratures, the order can be increased to $p = 2s$, $p = 2s-1$, and $p = 2s-2$, respectively [3, 4, 8, 9]. These nodal values are solutions of the algebraic equations

$$\frac{\mathrm{d}^s}{\mathrm{d}\tau^s}[\tau^s(\tau-1)^s] = 0 \quad \text{(Legendre)},$$

$$\frac{\mathrm{d}^{s-1}}{\mathrm{d}\tau^{s-1}}[\tau^s(\tau-1)^{s-1}] = 0 \quad \text{(Radau I)}; \quad \frac{\mathrm{d}^{s-1}}{\mathrm{d}\tau^{s-1}}[\tau^{s-1}(\tau-1)^s] = 0 \quad \text{(Radau II)}, \tag{11}$$

$$\frac{\mathrm{d}^{s-2}}{\mathrm{d}\tau^{s-2}}[\tau^{s-1}(\tau-1)^{s-1}] = 0 \quad \text{(Lobatto)}.$$

The collocation methods are remarkable in that, in fact, they give the analytical solution within the step in the form of the collocation polynomial

$$\boldsymbol{u}(\tau) = \boldsymbol{x}_0 + h\int_0^\tau \boldsymbol{p}(\tau)\,\mathrm{d}\tau$$

(not only the solution $\boldsymbol{x}_1$ at the end of the step and several intermediate solutions $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s$), which is very convenient for derivation of approximate solutions for a dense time grid. However, it should be borne in mind that the order of accuracy within the step is reduced to $p = s$ [3]. In addition, the presence of the interpolant allows one to obtain a sufficiently good initial approximation of the function $\boldsymbol{f}$ at the next step by extrapolation:

$$\boldsymbol{f}_i = \boldsymbol{p}(1+c_i) \quad (i = 1,\ldots,s)$$

for the subsequent iterative determination of intermediate solutions (10).

It should especially be noted that the collocation methods with the Legendre and Lobatto spacings become geometrical [4]: symmetric and orbitally stable, and the Legendre spacing, also symplectic. It seems that the Legendre spacing is more preferable, because with the same number of collocation points, its order is higher by 2. However, in spite of the fact that the order of the method with the Lobatto spacings is lower, it works a little faster. Indeed, at each step with ni iterations for determining intermediate solutions, the number of calculations of the equation function is equal to $ncf = ni\cdot(s-1)$, whereas with the Legendre spacings, $ncf = ni\cdot s$. In addition, the interpolant of the equation function is constructed for the entire integration segment $[t_0, t_0+h]$ with $c_1 = 0$ and $c_s = 1$, unlike the Legendre spacings for which all nodal points lie inside the segment. Hence, the predictor with the Lobatto spacings is better, which is very important for its software implementation.

As simple examples of the collocation methods, consider the Runge–Kutta methods well-known for a long time and including the explicit and implicit Euler methods (Radau I and II: $s = 1$, $p = 1$), the midpoint method (Legendre: $s = 1$, $p = 2$), the trapezoidal rule (Lobatto: $s = 2$, $p = 2$), and the Simpson method (Lobatto: $s = 3$, $p = 4$). If we take the Lagrange polynomial

$$\boldsymbol{p}(\tau) = \sum_{j=1}^{s} \boldsymbol{f}_j \prod_{k\neq j} \frac{\tau - c_k}{c_j - c_k}$$

as an interpolant of equation function (1), collocation method (10) acquires the classical form of the Runge–Kutta method [1–4]:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + h\sum_{j=1}^{s} b_j \boldsymbol{f}_j, \quad \boldsymbol{u}_i = \boldsymbol{x}_0 + h\sum_{j=1}^{s} a_{ij} \boldsymbol{f}_j \quad (i = 1,\ldots,s), \tag{12}$$

where the constants are expressed through the integrals of the basic Lagrange functions:

$$a_{ij} = \int_0^{c_i} \prod_{k\neq j} \frac{\tau - c_k}{c_j - c_k} \mathrm{d}\tau, \quad b_j = \int_0^1 \prod_{k\neq j} \frac{\tau - c_k}{c_j - c_k} \mathrm{d}\tau .$$

Soon after the publication of the first works on the Runge–Kutta collocation methods [1, 2], E. Everhart [6, 7] proposed to use the polynomial in the canonical form[3]

$$\boldsymbol{p}(\tau) = \sum_{j=1}^{s} \boldsymbol{a}_j \tau^{j-1}$$

as an interpolant. Here $\boldsymbol{a}_j = \boldsymbol{a}_j(\boldsymbol{f}_1,\ldots,\boldsymbol{f}_s)$ $(j = 1,\ldots,s)$. The simple interpolation yields a sufficiently simple form of the approximate solution

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + h\sum_{j=1}^{s} \frac{\boldsymbol{a}_j}{j}, \quad \boldsymbol{u}_i = \boldsymbol{x}_0 + h\sum_{j=1}^{s} \frac{\boldsymbol{a}_j}{j} c_i^j \quad (i = 1,\ldots,s). \tag{13}$$

However, to express the coefficients of the polynomial $\boldsymbol{a}$ through the collocation values of the function $\boldsymbol{f}$, the author expressed the divided differences of the Newton polynomial $\boldsymbol{\alpha}$ directly through the collocation values. Meanwhile, the coefficients of the canonical polynomial are expressed in terms of the divided differences by means of the linear relations

$$\boldsymbol{a}_j = \sum_{i=j}^{s} c_{ij} \boldsymbol{\alpha}_i \quad (j = 1,\ldots,s),$$

where the constants are calculated from the nodal values of spacing $c_1,\ldots,c_s$ using the recurrent formulas

$$c_{ii} = 1 \quad (i > 0), \quad c_{ij} = c_{i-1,j-1} - c_{i-1} c_{i-1,j} \quad (i > j > 0).$$

1.2. Second-order differential equations

Suppose now that the dynamical system is described by the second-order vector differential equation

$$\boldsymbol{x}'' = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{x}') \tag{14}$$

with the known initial dynamic state at time t_0:

$$\boldsymbol{x}_0 = \boldsymbol{x}(t_0), \quad \boldsymbol{x}'_0 = \boldsymbol{x}'(t_0). \tag{15}$$

It is required to determine the dynamic state of the system at time $t_0 + h$:

---

[3] Though the author himself did not present his method as a collocation one.

$$\boldsymbol{x}(t_0+h),\quad \boldsymbol{x}'(t_0+h)\,. \tag{16}$$

Here $\boldsymbol{x}$ and $\boldsymbol{x}'$ are the coordinate and velocity vectors, respectively.

We represent the approximate solutions for the coordinates and velocity in the form of polynomials

$$\boldsymbol{u}(t)\approx\boldsymbol{x}(t),\quad \boldsymbol{v}(t)=\boldsymbol{u}'(t)\approx\boldsymbol{x}'(t)\,, \tag{17}$$

which should satisfy to Eq. (14) at collocation times $t_i\in[t_0,t_0+h]$ $(i=1,\ldots,s)$:

$$\boldsymbol{u}''(t_i)=\boldsymbol{v}'(t_i)=\boldsymbol{f}(t_i,\boldsymbol{u}(t_i),\boldsymbol{v}(t_i))\quad (i=1,\ldots,s)\,, \tag{18}$$

and to the initial condition (15):

$$\boldsymbol{x}_0=\boldsymbol{u}(t_0),\quad \boldsymbol{x}_0'=\boldsymbol{v}(t_0)\,. \tag{19}$$

Then solutions (16) with allowance for (17) are approximately determined as

$$\boldsymbol{x}_1=\boldsymbol{u}(t_0+h)\approx\boldsymbol{x}(t_0+h),\quad \boldsymbol{x}_1'=\boldsymbol{v}(t_0+h)\approx\boldsymbol{x}'(t_0+h)\,. \tag{20}$$

According to (18), the Lagrange conditions for the interpolant $\boldsymbol{p}$ of the function $\boldsymbol{f}$

$$\boldsymbol{u}''(t_0+h\tau)=\boldsymbol{v}'(t_0+h\tau)\equiv\boldsymbol{p}(\tau) \tag{21}$$

can be represented in the form

$$\boldsymbol{p}(c_i)=\boldsymbol{f}_i\equiv\boldsymbol{f}(t_i,\boldsymbol{u}_i,\boldsymbol{v}_i),\quad \boldsymbol{u}_i\equiv\boldsymbol{u}(t_i),\quad \boldsymbol{v}_i\equiv\boldsymbol{v}(t_i),\quad t_i=t_0+hc_i\quad (i=1,\ldots,s)\,. \tag{22}$$

Forming the polynomial interpolant $\boldsymbol{p}$ from conditions (22) and then integrating (21) over τ on the segments $[0,1]$ and $[0,c_i]$ $(i=1,\ldots,s)$, we obtain a set of formulas of the collocation method for differential equation (14):

$$\begin{gathered}\boldsymbol{x}_1=\boldsymbol{x}_0+\boldsymbol{x}_0'h+h^2\textstyle\int_0^1\int_0^\tau\boldsymbol{p}(\tau)\,\mathrm{d}\,\tau^2,\qquad \boldsymbol{x}_1'=\boldsymbol{x}_0'+h\textstyle\int_0^1\boldsymbol{p}(\tau)\,\mathrm{d}\,\tau,\\ \boldsymbol{u}_i=\boldsymbol{x}_0+\boldsymbol{x}_0'hc_i+h^2\textstyle\int_0^{c_i}\int_0^\tau\boldsymbol{p}(\tau)\,\mathrm{d}\,\tau^2,\quad \boldsymbol{v}_i=\boldsymbol{x}_0'+h\textstyle\int_0^{c_i}\boldsymbol{p}(\tau)\,\mathrm{d}\,\tau\quad (i=1,\ldots,s),\end{gathered} \tag{23}$$

where the subintegral polynomial $\boldsymbol{p}$ depends also on the intermediate solutions $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s$ and $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_s$.

1.3. Mixed systems of the first- and second-order differential equations

The problem described by (14) and (15) can be represented in the form of (1) and (2):

$$\boldsymbol{x}'=\dot{\boldsymbol{x}},\quad \dot{\boldsymbol{x}}'=\boldsymbol{f}(t,\boldsymbol{x},\dot{\boldsymbol{x}}),\quad \boldsymbol{x}_0=\boldsymbol{x}(t_0),\quad \dot{\boldsymbol{x}}_0=\dot{\boldsymbol{x}}(t_0)=\boldsymbol{x}_0'\,. \tag{24}$$

In spite of the fact that both problems describe the same dynamical system, according to the collocation principle, their numerical solutions will differ significantly. Indeed, according to (10), for the alternative problem we have the approximate solutions

$$(\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1) = (\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0) + h\int_0^1 (\boldsymbol{q}(\tau), \boldsymbol{p}(\tau))\,\mathrm{d}\tau, \quad (\boldsymbol{u}_i, \boldsymbol{v}_i) = (\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0) + h\int_0^{c_i} (\boldsymbol{q}(\tau), \boldsymbol{p}(\tau))\,\mathrm{d}\tau \quad (i = 1, \ldots, s), \tag{25}$$

where the polynomial $\boldsymbol{q}$ interpolates the right-hand side of the equation for the coordinate vector. The accuracies of solutions (25) (for coordinate and velocity vectors) are of the same order p. However, the solution for the coordinates in (23) has the order $p+1$ owing to double integration; in other words, the coordinates in (23) are determined more precisely than in (25).

Nevertheless, sometimes the user is compelled to use the representation of the dynamical system in the form of Eqs. (24), thereby using the integrator for first-order equations (25) understanding that this inevitably reduces the accuracy of the approximate solution. This is necessary when Eq. (14) describing the dynamical system is supplemented with the first-order equations for some auxiliary dynamic quantities. For example, the mixed systems are used to investigate dynamical chaos [10] as well as for linearization, regularization, and stabilization of the dynamic equations [11–14]. In this case, Eqs. (24) are used instead of Eq. (14) in order that to reduce all equations of the mixed system to the same order. The alternative variant is to differentiate the additional equations so that the system as a whole was described by Eq. (14), though for complex dynamical systems, taking derivatives of functions of the additional equations is often practically impossible.

The natural and effective approach to solving mixed system of equations is the application of the hybrid integrator. If it is required to solve the first-order vector equation for auxiliary dynamic quantities $\boldsymbol{z}$ together with Eq. (14):

$$\boldsymbol{z}' = \boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{z}) \tag{26}$$

with the initial condition $\boldsymbol{z}_0 = \boldsymbol{z}(t_0)$, the set of formulas (23) should be supplemented with the formulas

$$\boldsymbol{z}_1 = \boldsymbol{z}_0 + h\int_0^1 \boldsymbol{r}(\tau)\,\mathrm{d}\tau, \quad \boldsymbol{w}_i = \boldsymbol{z}_0 + h\int_0^{c_i} \boldsymbol{r}(\tau)\,\mathrm{d}\tau \quad (i = 1, \ldots, s). \tag{27}$$

Here the interpolant $\boldsymbol{r}$ of the function $\boldsymbol{g}$ is constructed by analogy with the interpolant $\boldsymbol{p}$ from the intermediate solutions $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s$, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s$, and $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_s$. Thus, the hybrid collocation method for Eqs. (14) and (26) has the form of (23) and (27).

2. INTEGRATOR LOBBIE

For the mixed system of differential equations (14) and (26)

$$\boldsymbol{x}'' = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{z}), \quad \boldsymbol{z}' = \boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{z}), \quad \boldsymbol{x}_0 = \boldsymbol{x}(t_0), \quad \boldsymbol{x}_0' = \boldsymbol{x}'(t_0), \quad \boldsymbol{z}_0 = \boldsymbol{z}(t_0), \tag{28}$$

we take as interpolants the Newton polynomials

$$\boldsymbol{p}(\tau) = \sum_{j=1}^{s} \boldsymbol{\alpha}_j \prod_{k=1}^{j-1} (\tau - c_k) \quad \text{and} \quad \boldsymbol{r}(\tau) = \sum_{j=1}^{s} \boldsymbol{\beta}_j \prod_{k=1}^{j-1} (\tau - c_k) \tag{29}$$

with the Lobatto spacings $(c_1 = 0, c_s = 1)$. Here $\prod_{k=1}^{0} = 1$. The divided differences in (29) are determined in terms of the nodal values of the functions $\boldsymbol{f}$ and $\boldsymbol{g}$ using the recurrent formulas

$$\boldsymbol{\alpha}_j = \boldsymbol{f}_j, \quad \boldsymbol{\beta}_j = \boldsymbol{g}_j, \quad \boldsymbol{\alpha}_j := (\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_k)/(c_j - c_k), \quad \boldsymbol{\beta}_j := (\boldsymbol{\beta}_j - \boldsymbol{\beta}_k)/(c_j - c_k)$$
$$(j = 1, \ldots, s;\ k = 1, \ldots, j-1). \tag{30}$$

Substituting interpolants (29) into (23) and (27) as well as considering that $c_1 = 0$ and $c_s = 1$, we obtain the collocation method for system (28) in the form

$$\boldsymbol{u}_1 = \boldsymbol{x}_0, \quad \boldsymbol{v}_1 = \boldsymbol{x}'_0, \quad \boldsymbol{w}_1 = \boldsymbol{z}_0,$$

$$\boldsymbol{u}_i = \boldsymbol{x}_0 + \boldsymbol{x}'_0 h c_i + h^2 \sum_{j=1}^{s} a_{ij} \boldsymbol{\alpha}_j, \quad \boldsymbol{v}_i = \boldsymbol{x}'_0 + h \sum_{j=1}^{s} b_{ij} \boldsymbol{\alpha}_j, \quad \boldsymbol{w}_i = \boldsymbol{z}_0 + h \sum_{j=1}^{s} b_{ij} \boldsymbol{\beta}_j \quad (i = 2, \ldots, s), \tag{31}$$

$$\boldsymbol{x}_1 = \boldsymbol{u}_s, \quad \boldsymbol{x}'_1 = \boldsymbol{v}_s, \quad \boldsymbol{z}_1 = \boldsymbol{w}_s,$$

where

$$a_{ij} = \int_0^{c_i} \int_0^{\tau} \prod_{k=1}^{j-1} (\tau - c_k) \, \mathrm{d}\tau^2, \quad b_{ij} = \int_0^{c_i} \prod_{k=1}^{j-1} (\tau - c_k) \, \mathrm{d}\tau \quad (i, j = 1, \ldots, s). \tag{32}$$

We designate the k-fold integral of the jth Newton basic function by

$$\gamma_{jk}(\tau) = \int_0^{\tau} \ldots \int_0^{\tau} (\tau - c_1) \ldots (\tau - c_{j-1}) \, \mathrm{d}\tau^k. \tag{33}$$

If one considers $\gamma_{jk}(\tau)$ $(j, k = 1, \ldots, s+1)$ as elements of some matrix Γ of size $(s+1) \times (s+1)$, the constants of the collocation method – integrals of the Newton basic functions (32) – will form its first two columns:

$$a_{ij} = \gamma_{j2}(c_i), \quad b_{ij} = \gamma_{j1}(c_i) \quad (i, j = 1, \ldots, s).$$

Meanwhile, the elements of the matrix Γ for any arbitrary value τ are calculated row-by-row using the recurrent relations:

$$\gamma_{1k} = \tau^k / k! \quad (k = 1, \ldots, s+1); \quad \gamma_{jk} = (\tau - c_{j-1}) \gamma_{j-1,k} - k \gamma_{j-1,k+1} \quad (j = 2, \ldots, s; \, k = 1, \ldots, s - j + 2). \tag{34}$$

Nonlinear equations (31) for $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s$, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s$, and $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_s$ are solved at each step by the fixed-point iteration in the Seidel modification. Before iterations, the solutions $\boldsymbol{u}_1$, $\boldsymbol{v}_1$, and $\boldsymbol{w}_1$ are known together with the divided differences

$$\boldsymbol{\alpha}_1 = \boldsymbol{f}_1 = \boldsymbol{f}(t_0, \boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1) \quad \text{and} \quad \boldsymbol{\beta}_1 = \boldsymbol{g}_1 = \boldsymbol{g}(t_0, \boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{w}_1)$$

at the first collocation point $c_1 = 0$. In the beginning of the iterative process at the second collocation point c_2, the group of solutions $\boldsymbol{u}_2$, $\boldsymbol{v}_2$, and $\boldsymbol{w}_2$ is determined, and from them $\boldsymbol{f}_2$ and $\boldsymbol{g}_2$, by which the divided differences $\boldsymbol{\alpha}_2$ and $\boldsymbol{\beta}_2$ are refined using recurrent formulas (30). Then the divided differences $\boldsymbol{\alpha}_3$ and $\boldsymbol{\beta}_3$ at the third collocation point c_3 are refined in the same way. After successive refinement of all divided differences at the step, the iteration is repeated. The iterative process is continued until the inequality

$$\| \boldsymbol{u}_s - \boldsymbol{u}_s^* \| < \varepsilon \| \boldsymbol{u}_s \| \tag{35}$$

is satisfied. Here $\boldsymbol{u}_s^*$ is the solution $\boldsymbol{u}_s$ at the preceding iteration and ε is a small parameter that determines the accuracy of convergence. By the way, the number of iterations at the step can be preset disregarding condition (35). In

this case, it should be borne in mind that the obtained solution $\boldsymbol{u}_s$ will not correspond to the preset order, and the method will lose the geometrical properties.

As initial values of the divided differences $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, one accepts their estimates obtained from recurrent formulas (30) using extrapolated values of the functions $\boldsymbol{f}$ and $\boldsymbol{g}$:

$$\boldsymbol{f}_i = \boldsymbol{p}(1+c_i), \quad \boldsymbol{g}_i = \boldsymbol{r}(1+c_i) \quad (i=1,\ldots,s). \tag{36}$$

At the first step, in the absence of such estimates, the iterative process begins with zero values of the divided differences, and continues until condition (35) is satisfied.

For multiple derivation of solutions on a dense time grid, it is convenient and expedient to use an appropriate collocation polynomial covering the times of the grid at a certain step rather than to carry out step-by-step integration at each of these times. The collocation polynomials for system (28) at any arbitrary time $t_0 + h\tau$ can be represented in the form

$$\boldsymbol{u}(\tau) = \boldsymbol{x}_0 + \boldsymbol{x}_0' h\tau + h^2 \textstyle\sum_{j=1}^{s} \gamma_{j2}(\tau)\boldsymbol{\alpha}_j, \quad \boldsymbol{v}(\tau) = \boldsymbol{x}_0' + h\textstyle\sum_{j=1}^{s} \gamma_{j1}(\tau)\boldsymbol{\alpha}_j, \quad \boldsymbol{w}(\tau) = \boldsymbol{z}_0 + h\textstyle\sum_{j=1}^{s} \gamma_{j1}(\tau)\boldsymbol{\beta}_j, \tag{37}$$

where $\gamma_{j1}(\tau)$ and $\gamma_{j2}(\tau)$ $(j=1,\ldots,s)$ are calculated by formulas (34).

An alternative method of obtaining the time series of approximate solutions is polynomial interpolation on intermediate solutions, for example, of type (29):

$$(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w})(\tau) = \textstyle\sum_{j=1}^{s} (\boldsymbol{u}_j,\boldsymbol{v}_j,\boldsymbol{w}_j) \prod_{k=1}^{j-1} (\tau - c_k). \tag{38}$$

However, the accuracy of interpolation polynomials (38) between the collocation points appears much lower than the accuracy of collocation polynomials (37) obtained by direct integration of polynomials (29).

The step size h as an integrator parameter is assigned by the user. However, the automatic choice of the step size is possible during step-by-step integration. The h value is chosen so that the approximate estimate of the *s*-order term of the Taylor series for the velocity vector was retained [5]:

$$\| \boldsymbol{e} \|_{\text{cal}} = \frac{h}{s} \| \boldsymbol{\alpha}_s \| \approx \frac{h^s}{s!} \| \boldsymbol{x}'^{(s)} \|, \tag{39}$$

that is, if the term of the series is considered as the principal error term, the step size should be determined as for a method of order $p = s-1$, though the order with the Gaussian quadrature spacings is significantly higher (not less than twice). Unfortunately, it is practically impossible to obtain the estimate of higher order.

Suppose that estimate (39) should be equal to a constant $\| \boldsymbol{e} \|_{\text{tol}}$ assigned by the user. Since

$$\| \boldsymbol{e} \|_{\text{cal}} \approx \frac{h^s}{s!} \| \boldsymbol{x}'^{(s)} \| \quad \text{and} \quad \| \boldsymbol{e} \|_{\text{tol}} \approx \frac{\hbar^s}{s!} \| \boldsymbol{x}'^{(s)} \|,$$

the step size $\hbar$ needed to provide $\| \boldsymbol{e} \|_{\text{tol}}$ can be estimated as

$$\hbar = h \left(\frac{\| \boldsymbol{e} \|_{\text{tol}}}{\| \boldsymbol{e} \|_{\text{cal}}} \right)^{1/s}. \tag{40}$$

After obtaining estimate (40) at the current step, integration is not repeated with the new step size $\hbar$, but at the next step, it is used according to the formula

$$h_{n+1} = rh_n, \quad r = \left(\frac{s}{h_n} \frac{\| \boldsymbol{e} \|_{\text{tol}}}{\| \boldsymbol{\alpha}_s \|} \right)^{1/s}, \tag{41}$$

where n and $n+1$ are the numbers of the current and subsequent steps, respectively. We note that the automatic choice of the step size implies the modification of predictor (36), namely

$$\boldsymbol{f}_i = \boldsymbol{p}(1 + rc_i), \quad \boldsymbol{g}_i = \boldsymbol{r}(1 + rc_i) \quad (i = 1, \ldots, s). \tag{42}$$

Algorithm (41) is effective if h_n and h_{n+1} for any n differ insignificantly. Noticeable changes in the sequence of estimates (41) are observed when the function $\boldsymbol{f}$ behaves irregularly, for example, during integration in the vicinity of its singularities. To avoid large differences between h_n and h_{n+1}, damping of the ratio $r = h_{n+1}/h_n$ is required, that is, the restrictions

$$\sigma^{-1/s} < r < \sigma^{1/s}: \quad r < \sigma^{-1/s} \Rightarrow r = \sigma^{-1/s} \quad \text{or} \quad r > \sigma^{1/s} \Rightarrow r = \sigma^{1/s} \tag{43}$$

should be imposed on it. Here σ defines the range of admissible variations of $\| \boldsymbol{e} \|_{\text{cal}}$. In order that $\| \boldsymbol{e} \|_{\text{cal}}$ changed within one order of magnitude, we should set $\sigma = \sqrt{10} \approx 3.16$. If the left side of inequality (43) is not fulfilled, the step is repeated.

The initial step size h_1 is determined from estimate (41) for $s = 2$ [5]:

$$h_1 = \sqrt{2\eta \frac{\| \boldsymbol{e} \|_{\text{tol}}}{\| \boldsymbol{f}_2 - \boldsymbol{f}_1 \|}}, \quad \boldsymbol{f}_1 = \boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{x}_0', z_0), \quad \boldsymbol{g}_1 = \boldsymbol{g}(t_0, \boldsymbol{x}_0, \boldsymbol{x}_0', z_0), \quad \boldsymbol{f}_2 = \boldsymbol{f}(t_1, \boldsymbol{x}_1, \boldsymbol{x}_1', z_1),$$

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{x}_0'\eta + \tfrac{1}{2}\boldsymbol{f}_1\eta^2, \quad \boldsymbol{x}_1' = \boldsymbol{x}_0' + \boldsymbol{f}_1\eta, \quad z_1 = z_0 + \boldsymbol{g}_1\eta, \quad t_1 = t_0 + \eta. \tag{44}$$

Here η is a small value. If η is so small that in computer arithmetic $\boldsymbol{f}_2 = \boldsymbol{f}_1$, it is increased by an order of magnitude, and estimate (44) is repeated. Estimate (44) of the starting step size is adequate only for the Euler method of the first order. Therefore, for the method of any other order, the starting step size will be much less than that expected. Nevertheless, the step-by-step integration begins with estimate (44), but then with the use of algorithm (41) with allowance for restrictions (43), the step size will gradually reach the proper value of the regular operating mode of the integrator with automatic choice of the step size.

The last step is revealed from the condition

$$(t_0 + \Delta t - t_n)/h_{n+1} < 1, \tag{45}$$

where Δt is the length of the entire interval of integration and t_n is the time of the nth step. Then to obtain the time $t_0 + \Delta t$, the last step size is chosen compulsorily as $h_{n+1} = t_0 + \Delta t - t_n$, and its ratio to the current step size h_n is redefined: $r = h_{n+1}/h_n$.

3. PROCEDURE LOBBIE

The integrator was implemented in Fortran up to order 32 in computer arithmetic with double and quadruple precision. The program procedure of the integrator Lobbie is called by the command

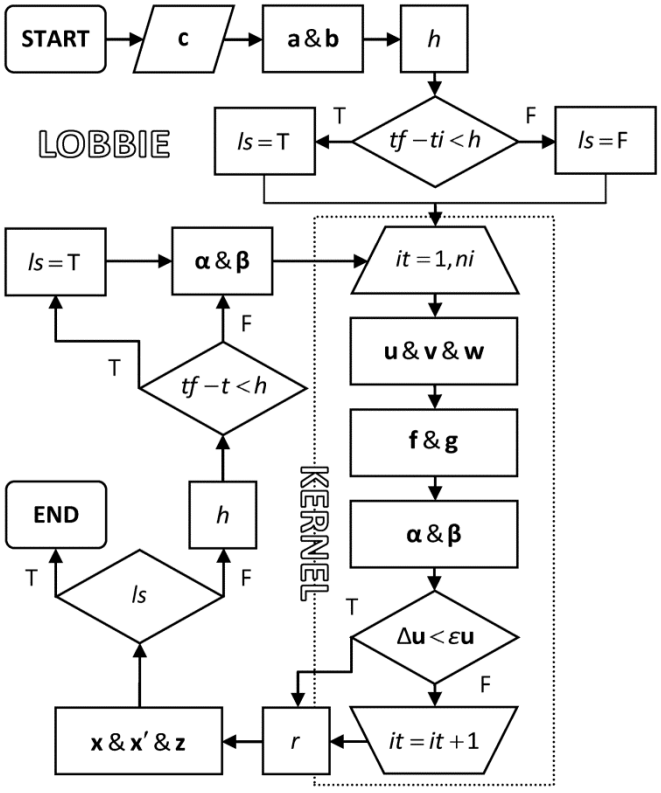

Fig. 2. Flow chart of the procedure Lobbie.

call lobbie (x, y, z, ts, tf, step, etol, nxy, nz, ns, ni, nst, ncf, fun).

Here **x**, **y**, **z** are the arrays of integral variables $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{z}$, respectively: at the input, their values at an initial moment of time t_0 (**ts**); at the output, their values at a final moment of time $t_0 + \Delta t$ (**tf**); **step** is an initial step size h_1: for automatic choice (41) its output value is h_n (the size at the penultimate step), while if **step** is zero, h_1 is assigned by the integrator according to estimate (44); **etol** is $\| \boldsymbol{e} \|_{\text{tol}}$: at zero value, the mode of a constant integration step; **nxy** and **nz** are dimensions of the arrays **x**, **y** ($\dim \boldsymbol{x} = \dim \boldsymbol{x}'$), and **z** ($\dim \boldsymbol{z}$), respectively; **ns** is the number of nodal values s; **ni** is the maximal number of iterations at the step for solving nonlinear equations (31) for $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s$, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s$, and $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_s$; **nst** and **ncf** are the numbers of the steps and calls of the procedure **fun** for evaluation of functions $\boldsymbol{f}$ and $\boldsymbol{g}$ over the entire interval of integration. The procedure **fun** is defined as

subroutine fun(t, x, y, z, f),

where **t** is a current time t; **x**, **y**, **z** are the arrays of integrated variables with values at time t; **f** is the output array of values of functions $\boldsymbol{f}$ and $\boldsymbol{g}$ with dimensions $\dim \boldsymbol{f} + \dim \boldsymbol{g}$.

The computational process in the procedure Lobbie is carried out according to the flow chart presented in Fig. 2. We now briefly describe the main stages of step-by-step integration in the case of a variable step with an automatic choice of its starting value, while time increases $(\Delta t > 0)$.

I. From the data block attached to the procedure, the array of the nodal values $c_1,\ldots,c_s$ is read, and the integrator constants (32) are recurrently calculated by using (34). The starting step size h_1 is estimated by (44). If the condition $h_1 > \Delta t$ is fulfilled, the starting step is considered to be the last, and then the size is set $h_1 = \Delta t$.

II. The intermediate solutions $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s$, $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_s$, and $\boldsymbol{w}_1,\ldots,\boldsymbol{w}_s$ (31) are iteratively determined together with the functions $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_s$ and $\boldsymbol{g}_1,\ldots,\boldsymbol{g}_s$ (28) as well as the divided differences $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_s$ and $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_s$. Iterations ends when condition (35) is satisfied or when the number of iterations reaches its maximum value ni.

III. When the iterative process ends, the solutions $\boldsymbol{x}_1,\boldsymbol{x}'_1,\boldsymbol{z}_1$ (31) are formed taking into account the refined divided differences $\boldsymbol{\alpha}_s$ and $\boldsymbol{\beta}_s$ at the last iteration. The scaling multiplier r (41) is estimated with allowance for damping conditions (43).

IV. If the step is the last one, the procedure ends. Otherwise, the size of the successive step (41) is defined. The fulfillment of condition (45) establishes the last integration step, and then its size is redefined so that to reach the final time $t_0 + \Delta t$.

V. The values of the divided differences (30) are extrapolated using the values of the functions of differential equations (42) and the computing process is repeated from item II, where the obtained solutions $\boldsymbol{x}_1,\boldsymbol{x}'_1,\boldsymbol{z}_1$ are accepted to be the initial ones $\boldsymbol{x}_0,\boldsymbol{x}'_0,\boldsymbol{z}_0$.

CONCLUSIONS

Thus, the theoretical background of the new collocation integrator intended for solving the mixed systems of first- and second-order differential equations of dynamics have been outlined in the paper. The practical implementation of the integrator was considered, and the software procedure Lobbie was described. In future we plan to present results of testing of the new integrator on the example of simple dynamical systems and show its efficiency compared to other integrators widely used in practic.

The research was carried out within the State Assignment of the Ministry of Science and Higher Education of the Russian Federation No. 0721-2020-0049.

REFERENCES

1. A. Guillou and J. L. Soule, Rev. Francaise Informat. Recherche Opérationnelle 3, Ser. R-3, 17–44 (1969).
2. K. Wright, BIT, **10**, 217–227 (1970).
3. E. Hairer, S. P. Nørsett, and G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, Springer (2008).
4. E. Hairer, C. Lubich, and G. Wanner, Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations, Springer (2006).
5. V. A. Avdyushev, Numerical Simulation of Orbits of Celestial Bodies [in Russian], Publishing House of Tomsk State University, Tomsk (2015).
6. E. Everhart, Celest. Mech., **10**, 35–55 (1974).
7. E. Everhart, in: Proc. IAU Colloq. 83 (Rome 1984), ed. by A. Carusi and G. B. Valsecchi, D. Reidel Publishing Co., Dordrecht (1985), pp. 185–202.
8. J. Kuntzmann, Z. Angew. Math. Mech., **41**, 28–31 (1961).
9. J. C. Butcher, Math. Comput., **18**, 50–64 (1964).
10. P. M. Cincotta, C. M. Giordano, and C. Simó, Physica D, **182**, 151–178 (2003).
11. C. A. Burdet, Z. Angew. Math. Phys., **19**, 345–368 (1968).
12. P. Kustaanheimo and E. Stiefel, J. Reine Angew. Math., **218**, 204–219 (1965).
13. V. A. Shefer, Astron. Zh., **68**, 197–205 (1991).
14. J. Baumgarte, Comp. Math. Appl. Mech. Eng., **1**, 1–16 (1972).