A new way of deriving implicit Runge-Kutta methods based on repeated integrals

Hana Mizerová* Katarína Tvrdá **

* Department of Mathematical Analysis and Numerical Mathematics, Comenius University
Mlynská dolina, 842 48 Bratislava, Slovakia

** Faculty of Civil Engineering, Slovak University of Technology in Bratislava
Radlinského 11, 810 05 Bratislava, Slovakia

Abstract

Runge-Kutta methods have an irreplaceable position among numerical methods designed to solve ordinary differential equations. Especially, implicit ones are suitable for approximating solutions of stiff initial value problems. We propose a new way of deriving coefficients of implicit Runge-Kutta methods. This approach based on repeated integrals yields both new and well-known Butcher's tableaux. We discuss the properties of newly derived methods and compare them with standard collocation implicit Runge-Kutta methods in a series of numerical experiments. In particular, we observe higher accuracy and higher experimental order of convergence of some newly derived methods.

Keywords: implicit Runge-Kutta method, Cauchy's repeated integration, modified Newton-Cotes quadratures, stiff initial value problem

1 Introduction

Stiff problems form an important class of ordinary differential equations (ODEs). They appear in many physical and chemical processes, for instance in fluid dynamics, chemical reaction kinetics, medicine, plasticity, neutron kinetics, porous media, gas transmission networks, transient magnetodynamics, in the study of spring and damping system, and others, see [18, 15] and the references

*Corresponding author: hana.mizerova(at)fmph.uniba.sk.
The work of H.M. was supported by the grants VEGA 1/0084/23 and VEGA 1/0709/24. The work of K.T. was supported by the grants KEGA 030STU-4/2023 and VEGA 1/0155/23.

therein. It is difficult to define *stiffness*, but it is even more challenging to solve stiff problems numerically - standard numerical methods are unstable, requiring unexpectedly small discrete step in regions of smooth solution curve. Runge-Kutta methods represent an important class of well-established numerical methods for solving ODEs. The explicit ones with small stability regions turned out to be unsuitable for stiff problems - discrete step size has to meet stability rather than accuracy requirements [2]. However, *implicit Runge-Kutta methods* with large stability regions are frequently used to solve stiff problems despite being more demanding from the computational point of view.

Let us consider an initial value problem (IVP) of the form

$$y'(x) = f(x, y(x)), \ x \in [a, b], \tag{1.1a}$$

$$y(a) = y_a \tag{1.1b}$$

with $a < b$, and $f : [a, b] \times \mathbb{R} \to \mathbb{R}$, $y_a \in \mathbb{R}$ given. The independent variable x not always denotes time, but it is usually referred to as a time variable. Quantity y represents the dependent variable. Knowing the value y_a of a quantity y at time $x = a$ we want to predict what happens to y as time evolves. Initial value problem is an essential tool in scientific computations and mathematical modelling. Therefore it is of a great interest to design efficient and reliable numerical schemes to approximate the exact solutions with a desired precision, especially in the case of stiff systems which are pertinent for many real-world problems described by differential equations.

The famous Euler method published in 1768-1770 is based on a simple idea of a moving particle given by a differential equation: in a short period of time in which the velocity has not changed significantly, the change in position will be approximately equal to the change in time multiplied by the initial velocity. It was at the end of 19^{th} and the beginning of 20^{th} century when Runge, Heun and Kutta generalized the Euler method by allowing several evaluations of the derivative in one computational step. Nowadays, the Runge-Kutta methods (RK) form an important family of numerical schemes used to solve IVPs. See, e.g, [8] and the references therein.

As already mentioned, stiff ODEs exhibit distressing behaviour when solved by classical numerical methods. Since explicit RK applied to stiff equations are usually unstable, and implementation of fully implicit RK is costly, there has been a growing interest in designing implicit methods with reduced computational costs. For instance, the so-called DIRK (diagonal implicit Runge-Kutta) and SDIRK (singly diagonal implicit Runge-Kutta) methods, see, e.g., [18, 13] and the references therein. It is impossible to provide an exhausting literature overview on the design of new implicit Runge-Kutta(-type) methods. Let us mention, for instance, new optimized implicit-explicit Runge-Kutta methods [19], new collocation methods based on weighted integrals [25], a two step fifth order RK for differential-algebraic equations [22], usage of interval analysis tools to compute coefficients of Runge-Kutta methods [21], and implicit seven stage tenth order Runge-Kutta methods based on Gauss-Kronrod-Lobatto quadrature formula [26].

In the present paper we provide a new way of deriving coefficients of implicit Runge-Kutta methods. Our approach yields both new and known schemes depending on the quadrature formula we choose. In particular, we derive a stiffly accurate implicit four stage Runge-Kutta method with explicit first line of order 4. In addition, its experimental order of convergence in the case of linear IVPs is 6. To the best of our knowledge there is no result in the literature for computing coefficients of Runge-Kutta methods based on repeated integration.

The paper is organized as follows: we provide preliminary material in Section 2. Section 3 contains the main result - new strategy of designing implicit Runge-Kutta methods. Several numerical experiments are presented in Section 4. Conclusion is followed by Appendix containing the list of newly derived and some well-known implicit Runge-Kutta methods.

2 Preliminaries

We provide a necessary mathematical apparatus, notation and known results used in the paper.

2.1 A general Runge-Kutta method

Let $y(x)$ be the exact solution of (1.1). Let $N \geq 1$. We seek approximations y_n of exact values $y(x_n)$ at points

$$x_n = a + nh, \ h = \frac{b-a}{N}, \ n = 0, \dots, N.$$

Here h denotes the *discrete step size*. The initial condition (1.1b) yields $y_0 = y(x_0) = y(a) = y_a$.

A general Runge-Kutta method is a single step method employing s stages K_i in one step. To determine value y_{n+1} from known value y_n we compute

$$\begin{aligned} y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i K_i \\ K_i &= f\left(x_n + hc_i, y_n + h \sum_{j=1}^{s} a_{ij} K_j\right), \ i = 1, \dots, s. \end{aligned} \tag{2.1}$$

The choice of coefficients $a_{i,j}$, b_i and c_i defines the method itself. Every Runge-Kutta method can be represented by a practical *Butcher's tableau* [3], see Table 1.

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b^T} \end{array} \quad = \quad \begin{array}{c|ccc} c_1 & a_{1,1} & \dots & a_{1,s} \\ . & . & \dots & . \\ c_s & a_{s,1} & \dots & a_{s,s} \\ \hline & b_1 & \dots & b_s \end{array}$$

Table 1: Butcher's tableau of a general Runge-Kutta method

Obviously, the structure of the matrix $\mathbf{A}$ decides whether the corresponding Runge-Kutta method is explicit (lower triangular matrix with zeros on the diagonal), DIRK (lower triangular matrix), SDIRK (lower triangular matrix with the same diagonal elements) or implicit (full matrix). Typical examples are the explicit and implicit Euler methods, Heun's method - explicit trapezoidal rule, implicit trapezoidal rule, and the prominent explicit fourth-order Runge–Kutta method (RK4). See Table 2.

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

(a) ex Euler

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

(b) im Euler

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(c) Heun

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(d) im trapezoidal

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

(e) RK4

Table 2: Examples of well-known Runge-Kutta methods

2.1.1 A-stability

The concept of A-stability introduced by Dahlquist, originally used for multistep methods, suggests to solve IVP

$$\begin{aligned} y'(x) &= \lambda y(x) \\ y(0) &= 1 \end{aligned} \tag{2.2}$$

with $\lambda \in \mathbb{C}$, $\mathrm{Re}(\lambda) < 0$ numerically. Desirably our numerical solution has the same behaviour as the exact solution $y(x) = \mathrm{e}^{\lambda x}$, namely

$$y_n \to 0 \quad \text{as} \quad n \to \infty. \tag{2.3}$$

Applying a general RK on (2.2) we get, after one computational step, $y_{n+1} = R(h\lambda)y_n$, and by induction,

$$y_{n+1} = R(h\lambda)^n y_0.$$

Thus the stability condition (2.3) is equivalent to $|R(h\lambda)| < 1$.

The function $R(z)$ is called the *stability function* and the set $S = \{z \in \mathbb{C} : |R(z)| \leq 1\}$ is called the *stability domain.* If $S \supset \mathbb{C}^-$ (left half of complex plain), we say the method is *A-stable.* The stability function of a general RK reads

$$R(z) = \frac{\det(\mathbf{I} - z\mathbf{A} + z\mathbf{e}\mathbf{b}^T)}{\det(\mathbf{I} - z\mathbf{A})}, \tag{2.4}$$

where $\mathbf{I}$, $\mathbf{e}$ are the identity matrix and the vector of ones, respectively, see [8, 17]. Figure 1 depicts the stability domains of well-known RK from Table 2.

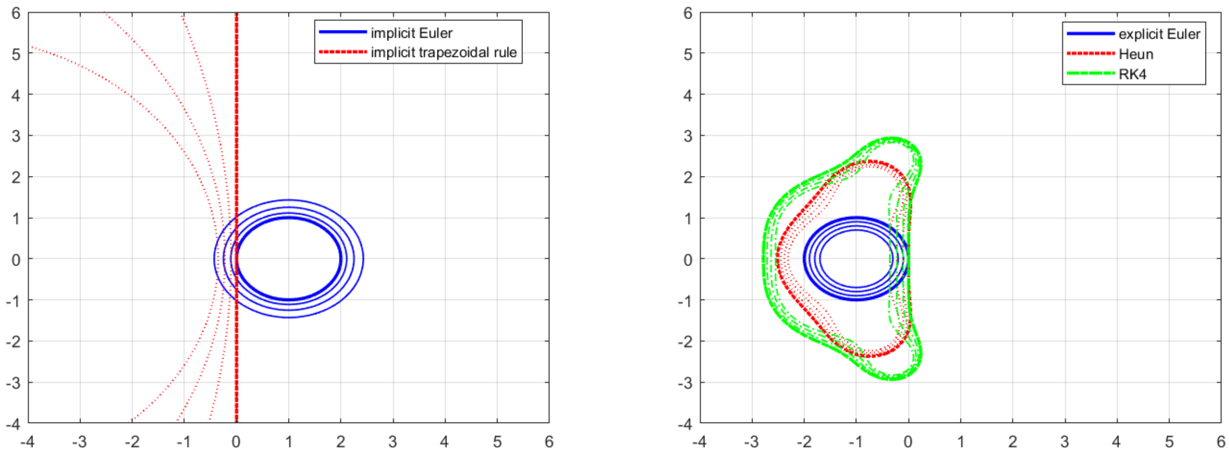

Figure 1: Stability domains of implicit (left) and explicit (right) RK from Table 2

2.1.2 Order conditions

The *order* of a Runge-Kutta method is p if and only if the local truncation error is $\mathcal{O}(h^{p+1})$. There is no explicit Runge-Kutta method of order p with $s = p$ stages for $p > 4$. The highest possible order of an implicit method with s stages is $p = 2s$ and is only attained by Gauss-Legendre methods. See [9] for more details on the so-called *Butcher's barriers*.

One of the great results of Butcher's theory for Runge-Kutta methods are the necessary and sufficient conditions to derive a new method of order p. These *order conditions* or *Butcher's rules* were firstly presented and connected with the rooted tree theory by Butcher, see [4, 7]. These conditions yield a system of equations with the unknowns $a_{i,j}$, b_i and c_i. The number of constraints for each order increases exponentially and is different for one- and high-dimensional problem. Table 3 contains 17 order conditions [12] to be satisfied by a Runge-Kutta method of order $p = 5$. Orders 6, 7 and 8 impose 37, 85 and 200 constraints, respectively. For linear ODEs the number of order conditions can be reduced, see [27]. For instance, to obtain a sixth order method it is enough to satisfy 16 conditions collected in Table 4. A Runge-Kutta method is of order p if and only if all order conditions up to order p are satisfied.

There are further constraints to be imposed in order to obtain a desired structure of the method, for instance:

- explicit: $a_{i,j} = 0, \; \forall \; j \geq 1$
- explicit first line: $a_{1,1} = \ldots = a_{1,s} = 0$
- diagonal implicit: $a_{i,j} = 0, \; \forall j > 1$
- singly diagonal: $a_{1,1} = \ldots = a_{s,s}$
- stiffly accurate: $a_{s,i} = b_i, \; \forall i = 1, \ldots, s$
- fully implicit: $a_{i,j} \neq 0, \; \forall i, j = 1, \ldots, s$.

p	order conditions			
1	$\sum_{i=1}^{s} b_i = 1$			
2	$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$			
3	$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$,	$\sum_{i,j=1}^{s} b_i a_{i,j} c_j = \frac{1}{6}$		
4	$\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4}$,	$\sum_{i,j=1}^{s} b_i c_i a_{i,j} c_j = \frac{1}{8}$,	$\sum_{i,j=1}^{s} b_i a_{i,j} c_j^2 = \frac{1}{12}$,	$\sum_{i,j,k=1}^{s} b_i a_{i,j} a_{j,k} c_k = \frac{1}{24}$
5	$\sum_{i=1}^{s} b_i c_i^4 = \frac{1}{5}$,	$\sum_{i,j=1}^{s} b_i c_i^2 a_{i,j} c_j = \frac{1}{10}$,	$\sum_{i,j=1}^{s} b_i c_i a_{i,j} c_j^2 = \frac{1}{15}$,	$\sum_{i,j=1}^{s} b_i a_{ij} c_j^3 = \frac{1}{20}$
		$\sum_{i,j,k=1}^{s} b_i a_{i,j} c_j a_{i,k} c_k = \frac{1}{20}$,	$\sum_{i,j,k=1}^{s} b_i c_i a_{i,j} a_{j,k} c_k = \frac{1}{30}$,	$\sum_{i,j,k=1}^{s} b_i a_{i,j} c_j a_{j,k} c_k = \frac{1}{40}$,
			$\sum_{i,j,k=1}^{s} b_i a_{ij} a_{j,k} c_k^2 = \frac{1}{60}$,	$\sum_{i,j,k,m=1}^{s} b_i a_{i,j} a_{j,k} a_{k,m} c_m = \frac{1}{120}$

Table 3: Order conditions for Runge-Kutta methods up to order $p = 5$

Note that an additional consistency constraint (row-sum condition), $c_i = \sum_{j=1}^{s} a_{i,j}$, is typically taken into account, but it is not necessary, see, e.g., [28]. We refer the interested reader to [4, 7, 16] for more details.

2.2 Standard implicit Runge-Kutta methods

In what follows we give a brief note on the derivation of standard implicit Runge-Kutta methods (sIRK) which shall be compared to our newly derived methods in terms of order and accuracy.

2.2.1 Collocation methods

Let us integrate the differential equation (1.1a) over the interval $[x_n, x]$ to get

$$y(x) - y(x_n) = \int_{x_n}^{x} f(x, y(x))\,\mathrm{d}x.$$

We approximate the exact value $y(x_n)$ by y_n and replace the integrand $f(x, y(x))$ by its unique Lagrange interpolation polynomial of degree at most $s-1$ corresponding to s points $z_{n,i} = x_n + \tau_i h$,

p	order conditions				
1	$\sum_{i=1}^{s} b_i = 1$				
2	$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$				
3	$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$,	$\sum_{i,j=1}^{s} b_i a_{i,j} c_j = \frac{1}{6}$			
4	$\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4}$,	$\sum_{i,j=1}^{s} b_i a_{i,j} c_j^2 = \frac{1}{12}$,	$\sum_{i,j,k=1}^{s} b_i a_{i,j} a_{j,k} c_k = \frac{1}{24}$		
5	$\sum_{i=1}^{s} b_i c_i^4 = \frac{1}{5}$,	$\sum_{i,j=1}^{s} b_i a_{ij} c_j^3 = \frac{1}{20}$	$\sum_{i,j,k=1}^{s} b_i a_{ij} a_{j,k} c_k^2 = \frac{1}{60}$,	$\sum_{i,j,k,m=1}^{s} b_i a_{i,j} a_{j,k} a_{k,m} c_m = \frac{1}{120}$	
6	$\sum_{i=1}^{s} b_i c_i^5 = \frac{1}{6}$	$\sum_{i,j=1}^{s} b_i a_{ij} c_j^4 = \frac{1}{30}$	$\sum_{i,j,k=1}^{s} b_i a_{ij} a_{j,k} c_k^3 = \frac{1}{120}$	$\sum_{i,j,k,m=1}^{s} b_i a_{i,j} a_{j,k} a_{k,m} c_m^2 = \frac{1}{360}$	
				$\sum_{i,j,k,m,n=1}^{s} b_i a_{i,j} a_{j,k} a_{k,m} a_{m,n} c_n = \frac{1}{720}$	

Table 4: Order conditions for Runge-Kutta methods up to order $p = 6$ for linear ODE's

$0 \le \tau_1 < \ldots \tau_i < \ldots < \tau_s \le 1$, i.e.

$$y(x) \approx y_n + \int_{x_n}^{x} L_{s-1}(x)\,\mathrm{d}x = y_n + \sum_{i=1}^{s} f(z_{n,i}, y(z_{n,i})) \int_{x_n}^{x} \ell_i(x)\,\mathrm{d}x,$$

where

$$L_{s-1}(x) = \sum_{i=1}^{s} f(z_{n,i}, y(z_{n,i}))\ell_i(x), \quad \ell_i(x) = \prod_{i \ne j=1}^{s} \left(\frac{x - z_{n,j}}{z_{n,i} - z_{n,j}} \right).$$

The principle of collocation method requires the above identity to hold at any point $y(z_{n,i})$. Thus, the approximations $y_{n,i} \approx y(z_{n,i})$ can be computed by a nonlinear system of equations

$$y_{n,i} = y_n + \sum_{j=1}^{s} f(z_{n,j}, y(z_{n,j})) \int_{x_n}^{z_{n,i}} \ell_j(x)\,\mathrm{d}x, \quad i = 1, \ldots, s.$$

In case $\tau_s = 1$ we set $y_{n+1} = y_{n,s}$, otherwise we set

$$y_{n+1} = y_n + \sum_{j=1}^{s} f(z_{n,j}, y(z_{n,j})) \int_{x_n}^{x_{n+1}} \ell_j(x)\,\mathrm{d}x.$$

Denoting $K_i = f(z_{n,i}, y(z_{n,i}))$, $c_i = \tau_i$, $i = 1, \dots, s$, and

$$b_i = \int_{x_n}^{z_{n,s}} \ell_i(x)\,\mathrm{d}x, \quad a_{i,j} = \int_{x_n}^{z_{n,i}} \ell_j(x)\,\mathrm{d}x$$

we get a standard implicit Runge-Kutta method with s stages. It shall be referred to as **sIRK**s. For reader's convenience we list the corresponding Butcher's tableaux for $s = 2, 3, 4, 5$ in Appendix, see Tables A3 and A5.

Remark 2.1. Choosing equally spaced points $z_{n,i} = x_n + h\frac{i-1}{s-1}$, $i = 1, \dots, s$, $h = x_{n+1} - x_n$, means that coefficients b_i are computed using a (closed) Newton-Cotes quadrature formula on the interval $[x_n, x_{n+1}]$.

2.2.2 General implicit Runge-Kutta methods

Every collocation method is a Runge-Kutta method but not every Runge-Kutta method is a collocation method. A general IRK can be derived using the so-called *simplifying order conditions* introduced by Butcher [6]. In particular, the derivation of the coefficients b_i and $a_{i,j}$ relies on three of them:

$$\begin{aligned}
B(p) &= \sum_{i=1}^{s} b_i c_i^{k-1} = \frac{1}{k}, \quad \text{for} \quad k = 1, \dots, p \\
C(q) &= \sum_{j=1}^{s} a_{i,j} c_j^{k-1} = \frac{c_i^k}{k}, \quad \text{for} \quad k = 1, \dots, q, \ i = 1 \dots, s \\
D(r) &= \sum_{i=1}^{s} b_i c_i^{k-1} a_{i,j} = \frac{b_j}{k}(1 - c_j^k), \quad \text{for} \quad k = 1, \dots, r, \ j = 1 \dots, s.
\end{aligned} \tag{2.5}$$

The coefficients c_i are chosen according to the quadrature formula as specified below. The following theorem was proved in [6].

Theorem 2.2. *If a Runge-Kutta method satisfies $B(p)$, $C(q)$ and $D(r)$ with $p \le q + r + 1$ and $p \le 2q + 2$, then its order of convergence is p.*

There are three groups of general IRK stemming from the choice of quadrature formula:

i) *Gauss-Legendre methods* which are collocation methods based on the roots of the Legendre polynomial $P_s^*(x)$ shifted to $[0, 1]$. The coefficients b_i and $a_{i,j}$ are obtained from $B(s)$ and $C(s)$, respectively. It can be shown that they are of maximal order $2s$, where s is the number of stages.

ii) *Radau methods* are divided to two subgroups: Radau IA based on the Radau left quadrature formula: $c_1 = 0$ and remaining c_i are the roots of $P_s^*(x) + P_{s-1}^*(x)$, and Radau IIA based on the Radau right quadrature formula: $c_s = 1$ and remaining c_i are the roots of $P_s^*(x) - P_{s-1}^*(x)$. Moreover, Radau IA impose conditions $B(s)$ and $D(s)$, while Radau IIA require conditions $B(s)$ and $C(s)$ to be satisfied. The well-known example of Radau IIA for $s = 1$ is the *implicit Euler method*. The Radau methods are not collocation methods and are all of order $2s - 1$.

iii) *Lobatto methods,* also called *Lobatto III*, are based on the choice $c_1 = 0$, $c_s = 1$ and remaining c_i being the roots of the polynomial $P_s^*(x) - P_{s-2}^*(x)$. Nowadays there are three established subgroups of Lobatto methods based on the choice of the coefficients $a_{i,j}$:
Lobatto IIIA impose condition $C(s)$ and for $s = 2$ yield the *implicit trapezoidal method*, Lobatto IIIB impose $D(s)$, and Lobatto IIIC impose conditions $a_{i,1} = b_i$ for $i = 1, \dots, s$ and $C(s-1)$. They are all of order $2s - 2$.

We refer the reader to [8, 5] for more details. In Appendix we give Butcher's tableaux for selected Gauss-Legendre, Radau and Lobatto methods, see Tables A3 and A4.

2.3 Repeated integrals

We aim to derive coefficients for RK in order to approximate the solution to IVP (1.1). Our new strategy requires the moments of $y'(x)$ and $f(x, y(x))$ to be equal. It means that besides the standard integral identity typical for methods based on numerical integration,

$$\int_{x_n}^{x_{n+1}} y'(x)\,\mathrm{d}x = \int_{x_n}^{x_{n+1}} f(x, y(x))\,\mathrm{d}x \tag{2.6}$$

we want the following identities to be satisfied:

$$\begin{aligned} \int_{x_n}^{x_{n+1}} (x_{n+1} - x) y'(x)\,\mathrm{d}x &= \int_{x_n}^{x_{n+1}} (x_{n+1} - x) f(x, y(x))\,\mathrm{d}x \\ &\cdots\cdots\cdots \\ \int_{x_n}^{x_{n+1}} (x_{n+1} - x)^{\gamma-1} y'(x)\,\mathrm{d}x &= \int_{x_n}^{x_{n+1}} (x_{n+1} - x)^{\gamma-1} f(x, y(x))\,\mathrm{d}x \end{aligned} \tag{2.7}$$

for arbitrary $\gamma \geq 1$ to be chosen later. As we shall see, it is useful to rewrite these moments equations as repeated integrals. This is possible thanks to Cauchy's formula for repeated integration [10] which allows us to compress γ antiderivatives of one function to a single integral. See [14] for the proof.

Theorem 2.3 (Cauchy's repeated integration formula)**.** *Let g be a continuous function on the real line. Then the γ^{th} repeated integral of g, denoted by $g^{(-\gamma)}$, is given by a single integral,*

$$g^{(-\gamma)} := \int_a^b \int_a^{x_1} \dots \int_a^{x_{\gamma-1}} g(x_\gamma)\,\mathrm{d}x_\gamma \dots \mathrm{d}x_1 = \frac{1}{(\gamma-1)!} \int_a^b (b-t)^{\gamma-1} g(t)\,\mathrm{d}t. \tag{2.8}$$

From now on, to avoid confusion with the discretization points x_n, $n = 0, \dots, N$, we use x for each integration variable on the left-hand side of (2.8).

By the Newton-Leibniz formula (the fundamental theorem of calculus) we have

$$y(x) - y(x_n) = \int_{x_n}^{x} y'(x)\,\mathrm{d}x. \tag{2.9}$$

Due to (2.8) identities (2.7) for $\gamma \geq 2$ become

$$\int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} y'(x)\,\mathrm{d}x\,\mathrm{d}x}_{(\gamma-1)-\text{times}}\,\mathrm{d}x = \int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} f(x,y(x))\,\mathrm{d}x\,\mathrm{d}x}_{(\gamma-1)-\text{times}}\,\mathrm{d}x. \tag{2.10}$$

In view of (2.9) the integral on the left-hand side is nothing but

$$\begin{aligned}\int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} y'(x)\,\mathrm{d}x\,\mathrm{d}x}_{(\gamma-1)-\text{times}}\,\mathrm{d}x &= \int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} y(x)\,\mathrm{d}x\,\mathrm{d}x}_{(\gamma-2)-\text{times}}\,\mathrm{d}x - y(x_n)\int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} 1\,\mathrm{d}x\,\mathrm{d}x}_{(\gamma-2)-\text{times}}\,\mathrm{d}x \\ &= \int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} y(x)\,\mathrm{d}x\,\mathrm{d}x}_{(\gamma-2)-\text{times}}\,\mathrm{d}x - y(x_n)\frac{h^{\gamma-1}}{(\gamma-1)!},\;\; h = x_{n+1}-x_n.\end{aligned}$$

Combining the above with (2.10) we get

$$\begin{aligned} hy(x_n) + \int_{x_n}^{x_{n+1}} \int_{x_n}^{x} f(x,y(x))\,\mathrm{d}x\,\mathrm{d}x &= \int_{x_n}^{x_{n+1}} y(x)\,\mathrm{d}x \\ \frac{h^2}{2}y(x_n) + \int_{x_n}^{x_{n+1}} \int_{x_n}^{x}\int_{x_n}^{x} f(x,y(x))\,\mathrm{d}x\,\mathrm{d}x\,\mathrm{d}x &= \int_{x_n}^{x_{n+1}} \int_{x_n}^{x} y(x)\,\mathrm{d}x\,\mathrm{d}x \\ &\dots \\ \frac{h^S}{S!}y(x_n) + \int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} f(x,y(x))\,\mathrm{d}x\,\mathrm{d}x}_{S-\text{times}}\,\mathrm{d}x &= \int_{x_n}^{x_{n+1}} \underbrace{\int_{x_n}^{x} \dots \int_{x_n}^{x} y(x)\,\mathrm{d}x\,\mathrm{d}x}_{(S-1)-\text{times}}\,\mathrm{d}x \end{aligned} \tag{2.11}$$

for $S = \gamma - 1 \geq 1$.

2.3.1 Numerical quadrature for repeated integrals

Another step towards deriving coefficients of new implicit RK is a suitable numerical approximation of repeated integrals in (2.11). We refer to [23, 24], where the detailed derivations of such quadrature formulas were presented. Here we briefly recall the results.

Gauss quadrature for repeated integral. Approximation of a repeated integral $g^{(-S)}$ by means of a general Gauss quadrature formula of order $2m-1$ reads

$$g^{(-S)} \approx \frac{(b-a)^S}{S!}\sum_{j=1}^{m} w_{S,j}\,g(t_j), \tag{2.12}$$

where

$$t_j = \frac{(b-a)(x_j+1)+2a}{2}, \qquad w_{S,j} = \frac{S w_j (1-x_j)^{S-1}}{2^S} \tag{2.13}$$

are the nodal Gaussian points shifted from $[-1, 1]$ to $[a, b]$, and the weights depending on the number of repeated integrals S, respectively. Here $x_j \in [-1, 1]$ are the original nodal points and w_j the corresponding weights of Gauss quadrature formula on $[-1, 1]$. The reader might have noticed that (2.12) with (2.13) stems from Cauchy's formula (2.8). See [23] for more details.

Newton-Cotes quadrature for repeated integral. Another possibility to approximate $g^{(-S)}$ is by the so-called modified Newton-Cotes formulas introduced in [24]. In this case, we firstly approximate function g by its unique Lagrange interpolation polynomial $L_{m-1}(x)$ corresponding to equidistant nodal points and then compute the weights directly, i.e. without applying Cauchy's formula (2.8).

For the case of *closed Newton-Cotes formula* let $H = \frac{b-a}{m-1}$ and $T_j = a + (j-1)H,\ j = 1, \ldots, m$. Then we write

$$g(x) \approx L_{m-1}(x) = \sum_{j=1}^{m} g(T_j)\ell_j(x), \quad \ell_j(x) = \prod_{j \neq i = 1}^{m} \left(\frac{x - T_i}{T_j - T_i} \right),$$

where $\ell_j(x)$ are the basis Lagrange interpolation polynomials. Consequently, for the repeated integral we obtain

$$g^{(-S)} \approx \sum_{j=1}^{m} g(T_j) \int_a^b \underbrace{\int_a^x \ldots \int_a^x \ell_j(x)\,\mathrm{d}x \ldots \mathrm{d}x}_{(S-1)-\text{times}} = H^S \sum_{j=1}^{m} W_{S,j}\, g(T_j), \quad H^S W_{S,j} = \ell_j^{(-S)}. \tag{2.14a}$$

An analogous formula holds for *open Newton-Cotes formulas* which do not include the endpoints a and b. Indeed,

$$g^{(-S)} \approx H^S \sum_{j=1}^{m-2} W_{S,j}\, g(T_j), \quad H^S W_{S,j} = \ell_j^{(-S)}, \quad T_j = a + jH,\ j = 1, \ldots, m-2 \tag{2.14b}$$

with the interval $[a, b]$ being divided into $m - 1$ subintervals of equal length.

3 New families of implicit Runge-Kutta methods

Our motivation is to solve (stiff) initial value problem (1.1) by an implicit Runge-Kutta method (IRK). We propose a new way of deriving the entries of $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{c}$, cf. Table 1. Instead of focusing on simplifying order conditions (2.5) we employ the moment approach and repeated integrals as described in Subsection 2.3 to determine the coefficients $a_{i,j}$, b_i and c_i.

We proceed in four steps:

(Step 1) choose a numerical quadrature and determine c_i accordingly

(Step 2) compute b_i by the chosen quadrature formula

(Step 3) for the number of unknown stages write the corresponding number of identities (2.11) and apply chosen quadrature formula on repeated integrals

(Step 4) solve the resulting system of algebraic equations for unknowns $y_{n,i}$ and get $a_{i,j}$

We shall follow these steps and detail the procedure for three different cases: the cases of closed and open Newton-Cotes quadrature formulas (NC) as well as the case of Gaussian quadratures of Legendre-, Radau- and Lobatto-type.

3.1 New way based on closed Newton-Cotes formulas

(Step 1) For a closed NC with the nodal pints $z_{n,i} = x_n + h\frac{i-1}{s-1}$, $i = 1, \ldots, s$, we have

$$c_i = \frac{i-1}{s-1},\ i = 1, \ldots, s.$$

(Step 2) Integrating (1.1a) over $[x_n, x_{n+1}]$ and applying the closed NC from Step 1 to approximate the integral of $f(x, y(x))$ over $[x_n, x_{n+1}]$ we get

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i K_i, \quad b_i = \int_{x_n}^{x_{n+1}} \ell_i(x)\,\mathrm{d}x,\ i = 1, \ldots, s. \tag{3.1}$$

Recall that $K_i = f(z_{n,i}, y(z_{n,i}))$. The coefficients b_i are simply the integrals of the corresponding Lagrange basis interpolation polynomials, analogously as in the case of standard collocation IRK, cf. Remark 2.1. Indeed, replacing $f(x, y(x))$ by its unique Lagrange interpolation polynomial corresponding to the nodal points from Step 1 we get

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))\,\mathrm{d}x \approx y_{n+1} = y_n + \sum_{i=1}^{s} f(z_{n,i}, y(z_{n,i})) \int_{x_n}^{x_{n+1}} \ell_i(x)\,\mathrm{d}x.$$

The difference compared to collocation methods from Subsubsection 2.2.1 lies in deriving the entries of the matrix $\mathbf{A}$.

(Step 3) Let $s \geq 3$. Since $z_{n,1} = x_n$ and $z_{n,s} = x_{n+1}$, we have $S = (s-2)$ unknown stages. Thus we consider the first S equations of (2.11) to which we apply the modified Newton-Cotes quadrature (2.14a) with $H = \frac{h}{s-1}$. This results in a system of S equations with the unknowns $y(z_{n,2})$, $\ldots$, $y(z_{n,s-1}))$,

$$\begin{aligned} hy_n + H^2\sum_{i=1}^{s} W_{2,i}K_i &= H\sum_{i=1}^{s} W_{1,i}y(z_{n,i}) \\ \frac{h^2}{2}y_n + H^3\sum_{i=1}^{s} W_{3,i}K_i &= H^2\sum_{i=1}^{s} W_{2,i}y(z_{n,i}) \\ &\cdots \\ \frac{h^S}{S!}y_n + H^{S+1}\sum_{i=1}^{s} W_{S+1,i}K_i &= H^S\sum_{i=1}^{s} W_{S,i}y(z_{n,i}), \end{aligned} \tag{3.2}$$

where $K_i = f(z_{n,i}, y(z_{n,i}))$ are parameters. Note that $y(z_{n,1}) = y(x_n) \approx y_n$ is known from the previous computational step and $y(z_{n,s}) = y(x_{n+1}) \approx y_{n+1}$ is given by (3.1) in Step 2.
(Step 4) Each $y(z_{n,i})$, $i = 2, \dots, s-1$ as a solution to (3.2) taking into account (3.1) can be expressed in the form

$$y(z_{n,i}) = y_n + h \sum_{j=1}^{s} a_{i,j} K_j,\ i = 2, \dots, s-1,$$

from which we get the coefficients $a_{i,j}$, $i = 2, \dots, s-1$, $j = 1, \dots, s$. Due to $y(z_{n,1}) = y(x_n) \approx y_n$ we have $a_{1,j} = 0$, $j = 1, \dots, s$, and from (3.1) simply $a_{s,j} = b_j$, $j = 1, \dots, s$.

For $\mathbf{s = 2}$ we obviously get the Lobatto IIIA with 2 stages and order 2, the so-called *implicit trapezoidal method*. Since $K_1 = f(x_n, y(x_n))$ and $K_2 = f(x_{n+1}, y(x_{n+1}))$, there are no unknown stages and thus no need to use repeated integrals (2.11).

For $\mathbf{s = 3}$ there is only one unknown stage $K_2 = f(z_{n,2}, y(z_{n,2}))$ with $z_{n,2} = x_n + \frac{1}{2}h$. Thus, from the first equation of (3.2) we get

$$h y_n + \frac{2H^2}{3} (K_1 + 2K_2) = \frac{H}{3} (y_n + 2y(z_{n,2}) + y_{n+1}),\ H = \frac{h}{2}.$$

In addition, (3.1) results in

$$y_{n+1} = y_n + \frac{h}{6} (K_1 + 4K_2 + K_3).$$

Therefore,

$$\begin{aligned} y(z_{n,2}) &= \left(\frac{3}{2} - \frac{1}{8} - \frac{1}{4} \right) y_n + \frac{h}{4} (K_1 + 2K_2) - \frac{h}{24} (K_1 + 4K_2 + K_3) \\ &= y_n + \frac{h}{24} (5K_1 + 8K_2 - K_3). \end{aligned}$$

We conclude

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ \frac{5}{24} & \frac{8}{42} & -\frac{1}{24} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix}, \quad \mathbf{b}^T = \left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6} \right), \quad \mathbf{c} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ 1 \end{pmatrix}.$$

This method is again known as Lobatto IIIA.

For $\mathbf{s = 4}$ we deal with the well-known 3/8-Simpson rule. It means we divide the interval $[x_n, x_{n+1}]$ into three subintervals of equal length $H = \frac{h}{3}$. Then

$$\int_{x_n}^{x_{n+1}} f(x, y(x))\, \mathrm{d}x \approx \frac{3}{8} H (K_1 + 3K_2 + 3K_3 + K_4),$$

where $z_{n,i} = x_n + (i-1)H$, $i = 1,2,3,4$. Thus $c_i = \frac{i-1}{3}$, $i = 1,2,3,4$, and Step 2 yields

$$y_{n+1} = y_n + \frac{3}{8}H\left(K_1 + 3K_2 + 3K_3 + K_4\right). \tag{3.3}$$

It means $b_1 = \frac{1}{8} = b_4$ and $b_2 = \frac{3}{8} = b_3$. Since the stages K_2 and K_3 including the values $y(z_{n,2})$ and $y(z_{n,3})$ are at this point unknown, we need two additional equations to close the system. Therefore we consider the first two identities of (3.2) to get

$$\begin{aligned} 3Hy_n + \frac{3}{40}H^2\left(13K_1 + 36K_2 + 9K_3 + 2K_4\right) &= \frac{3}{8}H\left(y_n + 3y(z_{n,2}) + 3y(z_{n,3}) + y_{n+1}\right) \\ \frac{9H^2}{2}y_n + \frac{9}{80}H^3\left(12K_1 + 27K_2 + 0K_3 + K_4\right) &= \frac{3}{40}H^2\left(13y_n + 36y(z_{n,2}) + 9y(z_{n,3}) + 2y_{n+1}\right). \end{aligned}$$

Let us denote $\alpha_j \in \{1,3,3,1\}$, $\beta_j \in \{13,36,9,2\}$, $\gamma_j \in \{12,27,0,1\}$. Using (3.3) we further rewrite the latter equations to get

$$y(z_{n,2}) = 2y_n + \frac{2}{3}H\sum_{i=1}^{4}\left(\frac{\beta_i}{10} - \frac{3\alpha_i}{16}\right)K_i - y(z_{n,3}) \tag{3.4a}$$

$$y(z_{n,3}) = 5y_n + \frac{1}{6}H\sum_{i=1}^{4}\left(\gamma_i - \frac{\alpha_i}{2}\right)K_i - y(z_{n,2}). \tag{3.4b}$$

Now we substitute (3.4b) into (3.4a) to obtain

$$y(z_{n,2}) = y_n + \frac{H}{18}\sum_{i=1}^{4}\left(\frac{\alpha_i}{4} - \frac{2\beta_i}{5} + \gamma_i\right)K_i, \tag{3.5}$$

and analogously, (3.4a) into (3.4b) to get

$$y(z_{n,3}) = y_n + \frac{H}{18}\sum_{i=1}^{4}\left(-\frac{5\alpha_i}{2} + \frac{8\beta_i}{5} - \gamma_i\right)K_i. \tag{3.6}$$

Now, let

$$A_i = \frac{\alpha_i}{4} - \frac{2\beta_i}{5} + \gamma_i \in \left\{\frac{141}{20}, \frac{267}{20}, -\frac{57}{20}, \frac{9}{20}\right\}, \quad B_i = -\frac{5\alpha_i}{2} + \frac{8\beta_i}{5} - \gamma_i \in \left\{\frac{63}{10}, \frac{231}{10}, \frac{69}{10}, -\frac{3}{10}\right\}.$$

Hence (3.3), (3.5) and (3.6) become

$$y_{n+1} = y_n + \frac{3H}{8}\sum_{i=1}^{4}\alpha_i K_i, \quad y(z_{n,2}) = y_n + \frac{H}{18}\sum_{i=1}^{4}A_i K_i, \quad y(z_{n,3}) = y_n + \frac{H}{18}\sum_{i=1}^{4}B_i K_i.$$

Recall $H = \frac{h}{3}$ and $K_i = f(z_{n,i}, y(z_{n,i}))$. We have derived four stage IRK given by

$$
\begin{aligned}
y_{n+1} &= y_n + h \sum_{i=1}^{4} \frac{\alpha_i}{8} K_i \\
K_1 &= f(x_n, y_n) \\
K_2 &= f\left(x_n + \frac{h}{3}, y_n + h \sum_{i=1}^{4} \frac{A_i}{54} K_i\right) \\
K_3 &= f\left(x_n + \frac{2}{3}h, y_n + h \sum_{i=1}^{4} \frac{B_i}{54} K_i\right) \\
K_4 &= f\left(x_n + h, y_n + h \sum_{i=1}^{4} \frac{\alpha_i}{8} K_i\right).
\end{aligned}
$$

We shall refer to it as **nIRK4**. Note that Lobatto IIIA with 4 stages differs in the choice of c_i and sIRK4 differs in the coefficients $a_{2,j}$, $a_{3,j}$, $j = 1, 2, 3, 4$.

Theorem 3.1 (properties of nIRK4)**.** *Newly derived four stage implicit Runge-Kutta method given by Butcher's tableau*

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{47}{360} & \frac{89}{360} & -\frac{19}{360} & \frac{3}{360} \\
\frac{2}{3} & \frac{21}{180} & \frac{77}{180} & \frac{23}{180} & -\frac{1}{180} \\
1 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\
\hline
 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8}
\end{array}
\tag{3.7}
$$

is stiffly accurate with explicit first line. It is A-stable with the order of convergence $p = 4$.

Proof. Obviously, the first line is explicit and since $a_{s,j} = b_j$, $j = 1, \ldots, s$ the method is stiffly accurate according to [4]. The method is indeed A-stable. The stability function reads

$$
R(z) = -\frac{z^3 + 12z^2 + 60z + 120}{z^3 - 12z^2 + 60z - 120},
$$

and direct calculation for $z = u + iv \in \mathbb{C}$, yields

$$
|R(z)| \leq 1 \Leftrightarrow u(u^4 + v^4 + 2u^2v^2 + 70u^2 + 30v^2 + 600) \leq 0
$$

which is only true for $u \leq 0$. Thus the stability domain is $S = \{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$. It is easy to see that the method satisfies the simplifying order conditions $B(p)$, $C(q)$ and $D(r)$ with $p = 4$, $q = 3$ and $r = 0$, which by Theorem 2.2 means the order is $p = 4$. Another possibility is to check that all 8 order conditions from Table 3 up to order $p = 4$ are satisfied. □

For $\mathbf{s \geq 5}$ one can analogously as above derive IRK based on closed Newton-Cotes formulas. We give Butcher's tableau and the corresponding order of **nIRK5** in Table A1.

3.2 New way based on open Newton-Cotes formulas

The only difference in deriving the coefficients of IRK by an open Newton-Cotes formula compared to the closed one is that the end points of the interval $[x_n, x_{n+1}]$ are not included. It means the resulting method neither has the explicit first line nor is stiffly accurate according to [4].

(Step 1) In order to derive an s stage Runge-Kutta method, we need to consider $s+1$ subintervals of $[x_n, x_{n+1}]$ with the corresponding nodal points $z_{n,i} = x_n + h\frac{i}{s+1}$, $i = 1, \dots, s$. Obviously,

$$c_i = \frac{i}{s+1},\ i = 1, \dots, s.$$

(Step 2) By the same reason as before, it holds that

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i K_i, \quad b_i = \int_{x_n}^{x_{n+1}} \ell_i(x)\,\mathrm{d}x,\ i = 1, \dots, s. \tag{3.8}$$

Note the difference compared to (3.1) lies in $\ell_i(x)$ being polynomials defined for different nodal points and of different degree.

(Step 3) For $s \geq 2$ we have $S = s$ unknown stages and the corresponding first s equations of (2.11) to be solved. Note that $y(z_{n,1}) \neq y(x_n)$ and $y(z_{n,s}) \neq y(x_{n+1})$. We approximate the integrals in (2.11) by the modified open Newton-Cotes quadrature formula (2.14b) which results in

$$\begin{aligned} hy_n + H^2\sum_{i=1}^{s} W_{2,i}K_i &= H\sum_{i=1}^{s} W_{1,i}y(z_{n,i}) \\ \frac{h^2}{2}y_n + H^3\sum_{i=1}^{s} W_{3,i}K_i &= H^2\sum_{i=1}^{s} W_{2,i}y(z_{n,i}) \\ &\cdots \\ \frac{h^S}{S!}y_n + H^{S+1}\sum_{i=1}^{s} W_{S+1,i}K_i &= H^S\sum_{i=1}^{s} W_{S,i}y(z_{n,i}), \end{aligned} \tag{3.9}$$

where $K_i = f(z_{n,i}, y(z_{n,i}))$ are parameters, and $y(z_{n,1}), \dots, y(z_{n,s})$ are the unknowns. Note that $y(x_n) \approx y_n$ is known from the previous computational step and $H = \frac{h}{s+1}$.

(Step 4) We solve system (3.9) and as a result we obtain the coefficients $a_{i,j}$, $i, j = 1, \dots, s$ from

$$y(z_{n,i}) = y_n + h\sum_{j=1}^{s} a_{i,j}K_j,\ i = 1, \dots, s.$$

For $s = 3, 4$ the newly derived IRK are listed in Table A2 and are referred to as **nIRK3o** and **nIRK4o**. They are both A-stable and of order $p = 4$.

Remark 3.2. Dividing the interval $[x_n, x_{n+1}]$ into s subintervals of equal length $H = \frac{h}{s}$ yields either $(s+1)$ or $(s-1)$ stage method for closed or open NC, respectively.

New way based on Newton-Cotes formulas combined with Cauchy's formula

As we have seen, weights in modified Newton-Cotes quadrature formulas (2.14) are obtained as repeated integrals of the basis Lagrange interpolation polynomials. However, we can also rewrite them using Cauchy's repeated integration formula (2.8) and then compute the weights. When we use this approach in (2.11) in Step 3 we derive new yet different implicit Runge-Kutta methods. Their Butcher's tableaux and corresponding orders for $s = 2, 3, 4, 5, 6$ in the case of closed formulas and for $s = 3, 4$ in the case of open formulas are given in Tables A1,A3 and A2. They are refereed to as **nIRK**s**c** and **nIRK**s**oc**, respectively. Here "**c**" stands for "Cauchy".

3.3 New way based on Gauss quadratures

It turns out that our approach based on repeated integrals combined with Gauss quadratures yields the well-known implicit Runge-Kutta methods. For completeness, let us briefly summarize the results.

Gauss-Legendre quadrature

(Step 1) The choice of c_i is obvious: the roots of the Legendre polynomial $P_s^*(x)$ shifted to $[0, 1]$.
(Step 2) The coefficients b_i are exactly the weights w_i of the corresponding Gauss-Legendre quadrature. Indeed,

$$\int_{x_n}^{x_{n+1}} f(x, y(x))\,\mathrm{d}x \approx h \sum_{i=1}^{s} f(z_{n,i}, y(z_{n,i}))w_i = h \sum_{i=1}^{s} b_i K_i.$$

Recall $K_i = f(z_{n,i}, y(z_{n,i}))$ with $z_{n,i} = x_n + c_i h$, $i = 1, \ldots, s$.
(Step 3) Let $s \geq 2$. We realize that $z_{n,1} \neq x_n$ and $z_{n,s} \neq x_{n+1}$. Thus, we have $S = s$ unknown stages to be determined in order to compute y_{n+1} given y_n. To this end, we consider the first s identities of (2.11). The repeated integrals of $f(x, y(x))$ and $y(x)$ are now approximated by means of (2.12). The system reads

$$\begin{aligned}
hy_n + \frac{h^2}{2!}\sum_{i=1}^{s} w_{2,i}K_i &= h\sum_{i=1}^{s} w_{1,i}y(z_{n,i}) \\
\frac{h^2}{2}y_n + \frac{h^3}{3!}\sum_{i=1}^{s} w_{3,i}K_i &= \frac{h^2}{2!}\sum_{i=1}^{s} w_{2,i}y(z_{n,i}) \\
&\cdots \\
\frac{h^s}{s!}y_n + \frac{h^{s+1}}{(s+1)!}\sum_{i=1}^{s} w_{s+1,i}K_i &= \frac{h^s}{s!}\sum_{i=1}^{s} w_{s,i}y(z_{n,i}).
\end{aligned} \tag{3.10}$$

(Step 4) Solving the underlying system yields $y(z_{n,1})$, $\ldots$, $y(z_{n,s})$, in the form

$$y(z_{n,i}) = y_n + h\sum_{j=1}^{s} a_{i,j}K_j,\ i = 1, \ldots, s,$$

and finally we have the coefficients $a_{i,j}$, $i, j = 1, \dots, s$.

Gauss-Radau quadrature

(Step 1) Set $c_1 = 0$ and c_i stemming from left Radau quadratures or set $c_s = 1$ and c_i stemming from right Radau quadratures as described above in Subsubsection 2.2.2.
(Step 2) Analogously as for the Gauss-Legendre quadrature, the coefficients b_i coincide with the weights of Radau left quadrature formula or Radau right quadrature formula, respectively.
(Step 3) In both cases we have $S = s - 1$ unknown stages. We consider the first $s - 1$ equations of (2.11) where the repeated integrals are approximated by (2.12) using the corresponding nodal points and weights of left or right Radau quadratures, respectively.
(Step 4) Solving the system of $s - 1$ first equations of (3.10) with unknowns $y(z_{n,2}), \dots, y(z_{n,s})$ in the case of left Radau quadrature, or with unknowns $y(z_{n,1}), \dots, y(z_{n,s-1})$ in the case of right Radau quadrature, yields the coefficients $a_{i,j}$, $i, j = 1, \dots, s$. Note that $a_{1,j} = 0$ for left Radau quadrature and $a_{s,j} = b_j$ for right Radau quadrature formulas.

Gauss-Lobatto quadrature

(Step 1) We set $c_1 = 0$, $c_s = 1$ and c_i, $i = 2, \dots, s - 2$, to be the roots of $P_s^*(x) - P_{s-2}^*(x)$.
(Step 2) The coefficients b_i coincide with the weights of chosen Lobatto quadrature formula, and again we have

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i K_i. \tag{3.11}$$

(Step 3) We need to consider $S = s - 2$ equations of (2.11) together with (3.11) on noting that y_n is known from the previous computational step. We approximate the repeated integrals by means of (2.12) using the nodal points and weights corresponding to chosen Gauss-Lobatto quadrature formula.
(Step 4) Coefficients $a_{i,j}$, $i, j = 2, \dots, s - 2$, are obtained from

$$y(z_{n,i}) = y_n + h \sum_{j=1}^{s} a_{i,j} K_j,\ i = 2, \dots, s - 2$$

which solve the system from Step 3 taking into account (3.11). Since $z_{n,1} = x_n$ we have the explicit first line, $a_{1,j} = 0$, $j = 1, \dots, s$. Finally, $a_{s,j} = b_j$, $j = 1, \dots, s$ because $z_{n,s} = x_{n+1}$.

The resulting IRK of Gauss-type (**nIRK-G**s, **nIRK-RI**s, **nIRK-RII**s, **nIRK-L**s according to type of Gauss quadrature) are listed in Tables A4 and A3 and are classified according to existing families of general Gauss-type IRK.

Note that the *advantage of our proposed approach* compared to the derivation of general IRK based on Gauss quadratures as described in Subsubsection 2.2.2 lies in the computation of $a_{i,j}$ and b_i. For instance, in the case of Gauss-Legendre quadrature, our strategy directly yields b_i and requires solving system of s equations of (3.10) to determine $a_{i,j}$. The general approach requires solving $s^2 + s$ equations stemming from the simplifying order conditions $B(s)$ and $C(s)$.

4 Numerical experiments

We present numerical experiments for selected newly derived implicit Runge-Kutta methods and compare *relative errors* and *experimental order of convergence* (EOC) with standard collocation implicit Runge-Kutta methods with the same number of stages.

We compute relative error

$$e := \frac{\max_{n=1,\dots,N} |y(x_n) - y_n|}{\max_{n=1,\dots,N} |y(x_n)|} \tag{4.1}$$

for different values of N, and EOC given by

$$EOC_{N,2N} := \log_2 \left(\frac{e^N}{e^{2N}} \right), \tag{4.2}$$

where e^N is the relative error computed with the discrete step size $h = 1/N$ and e^{2N} is the relative error computed with half the discrete step size.

4.1 Experiment 1: linear stability

Firstly, we consider standard IVP to test linear stability of Runge-Kutta methods,

$$\begin{aligned} y'(x) &= -15y(x), \ x \in [0,1] \\ y(0) &= 1 \end{aligned}$$

with the exact solution $y(x) = \mathrm{e}^{-15x}$.

In Table 5 we present relative errors and EOC for the following four stage stiffly accurate implicit Runge-Kutta methods with the explicit first line:
nIRK4 - new IRK based on closed Newton-Cotes formula,
nIRK4c - new IRK based on closed Newton-Cotes formula using Cauchy's formula,
sIRK4 - standard collocation IRK with $\tau_i = i/4$, $i = 0, \dots, 4$.
We term them "closed" because they include the approximation of function values at the end points, y_n and y_{n+1}.

Table 6 contains relative errors and EOC for the following four stage fully implicit Runge-Kutta methods:
nIRK4o - new IRK based on open Newton-Cotes formula,
nIRK4oc - new IRK based on open Newton-Cotes formula using Cauchy's formula,
sIRK4o - standard collocation IRK with $\tau_i = i/5$, $i = 1, \dots, 4$.
Obviously, we refer to these methods as "open" because, as opposed to the "closed" ones, their stages do not include the end points x_n and x_{n+1}.

The graphs of relative errors in logarithmic scale for the four stage implicit Runge-Kutta methods from Tables 5 and 6 are depicted in Figure 2.

In Table 7 we present the results obtained by selected five stage stiffly accurate implicit Runge-Kutta methods with the explicit first line:

	nIRK4		nIRK4c		sIRK4	
N	e	EOC	e	EOC	e	EOC
2	4.6672e-02	-	1.4050e-01	-	6.4999e-02	-
4	1.7495e-04	8.06	4.7758e-04	8.20	3.4600e-04	7.55
8	2.0422e-06	6.42	5.4487e-05	3.13	1.1297e-05	4.94
16	2.8906e-08	6.14	3.0925e-06	4.14	5.4443e-07	4.38
32	4.4030e-10	6.04	1.8744e-07	4.04	3.1673e-08	4.10
64	6.8360e-12	6.01	1.1621e-08	4.01	1.9436e-09	4.03
128	1.0700e-13	6.00	7.2485e-10	4.00	1.2091e-10	4.01

Table 5: Experiment 1: relative errors and EOC for “closed” four stage IRK

	nIRK4o		nIRK4oc		sIRK4o	
N	e	EOC	e	EOC	e	EOC
2	7.2443e-03	-	2.7660e-01	-	7.8219e-02	-
4	9.5741e-06	9.56	3.7663e-02	2.88	7.1200e-04	6.78
8	2.8238e-08	8.41	1.2664e-04	8.22	1.5671e-05	5.51
16	1.2693e-10	7.80	2.3930e-06	5.76	7.4538e-07	4.39
32	1.8070e-12	6.13	8.8648e-08	4.75	4.3335e-08	4.10
64	8.3000e-14	4.44	4.1962e-09	4.40	2.6583e-09	4.03
128	5.0000e-15	4.08	2.2606e-10	4.21	1.6640e-10	4.00

Table 6: Experiment 1: relative errors and EOC for “open” four stage IRK

nIRK5 - new IRK based on closed Newton-Cotes formula,
nIRK5c - new IRK based on closed Newton-Cotes formula using Cauchy’s formula,
sIRK5 - standard collocation IRK with $\tau_i = i/5$, $i = 0, \ldots, 5$. See the left plot in Figure 3.

Finally, Table 8 contains the relative errors and EOC for selected three stage fully implicit Runge-Kutta methods:
nIRK3o - new IRK based on open Newton-Cotes formula,
nIRK3oc - new IRK based on open Newton-Cotes formula using Cauchy’s formula,
sIRK3o - standard collocation IRK with $\tau_i = i/4$, $i = 1, \ldots, 3$. See the right plot in Figure 3.

Note, in particular, the EOC of our newly derived methods nIRK4, nIRK5 and nIRK3o is 6, 8 and 6, respectively, whereas the theoretical orders of convergence are 4,6 and 4, respectively. The reason in the case of nIRK4 might be that this method satisfies 13 out of 16 order conditions for $p = 6$ corresponding to linear ODEs from Table 4. The first sum for $p = 5$ and the first two for $p = 6$ only yield values $\frac{11}{54}$, $\frac{19}{108}$ and $\frac{1}{540}$ instead of $\frac{1}{5}$, $\frac{1}{6}$ and $\frac{1}{30}$, respectively. Similarly, nIRK3o satisfies 11 out of 16 order conditions from Table 4.

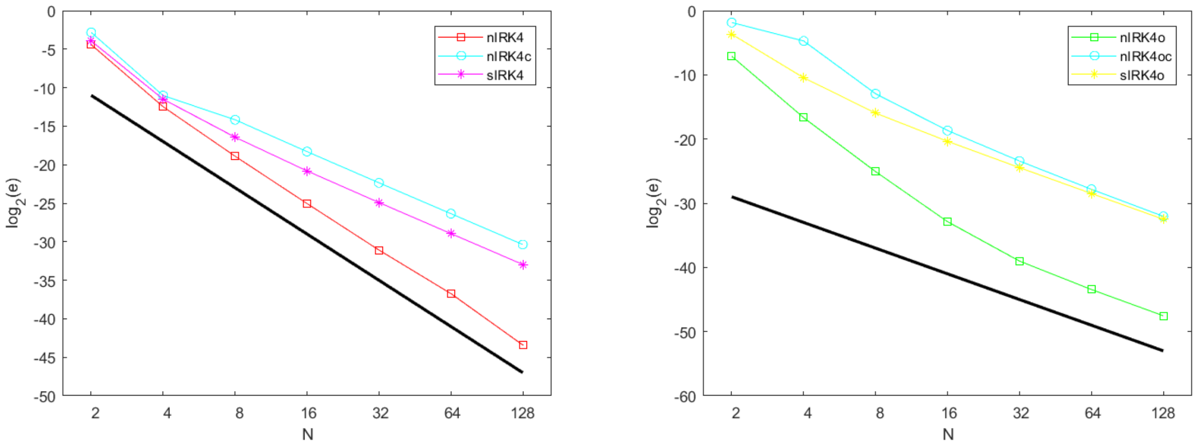

Figure 2: Experiment 1: relative errors for “closed” 4 stage IRK (left) and “open” 4 stage IRK (right); black solid lines represent reference slopes N^{-6} (left) and N^{-4} (right)

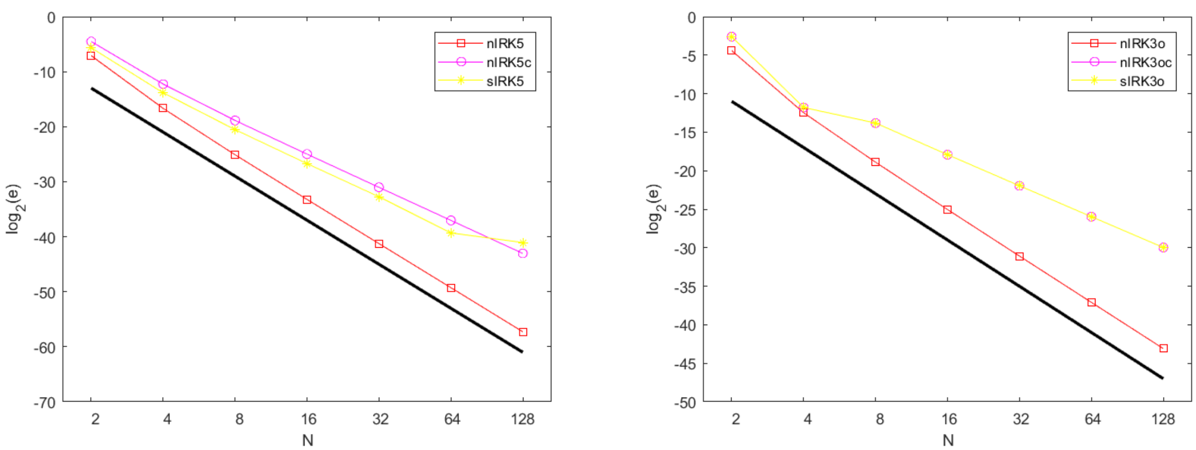

Figure 3: Experiment 1: relative errors for “closed” five stage IRK (left) and “open” three stage IRK (right); black solid lines represent reference slopes N^{-8} (left) and N^{-6} (right)

	nIRK5		nIRK5c		sIRK5	
N	e	EOC	e	EOC	e	EOC
2	7.2419e-03	-	4.3304e-02	-	1.9646e-02	-
4	9.5521e-06	9.56	2.0353e-04	7.73	6.7925e-05	8.18
8	2.7621e-08	8.43	2.0892e-06	6.60	6.5593e-07	6.70
16	9.9975e-11	8.11	2.9938e-08	6.12	9.0914e-09	6.17
32	3.8300e-13	8.03	4.5796e-10	6.03	1.3781e-10	6.04
64	1.0000e-15	8.01	7.1180e-12	6.01	2.1370e-12	6.01
128	0.0000	8.02	1.1100e-13	6.00	3.3000e-14	6.00

Table 7: Experiment 1: relative errors and EOC for "closed" five stage IRK

	nIRK3o		nIRK3oc		sIRK3o	
N	e	EOC	e	EOC	e	EOC
2	4.6672e-02	-	1.6339e-01	-	1.6339e-01	-
4	1.7495e-04	8.06	2.8067e-04	9.18	2.8067e-04	9.19
8	2.0422e-06	6.42	6.9453e-05	2.01	6.9453e-05	2.01
16	2.8906e-08	6.14	4.0354e-06	4.11	4.0355e-06	4.11
32	4.4030e-10	6.04	2.4569e-07	4.04	2.4572e-07	4.04
64	6.8360e-12	6.01	1.5248e-08	4.01	1.5245e-08	4.01
128	1.0700e-13	6.00	9.5129e-10	4.00	9.5024e-10	4.00

Table 8: Experiment 1: relative errors and EOC for "open" three stage IRK

4.2 Experiment 2: stiff equation

Solving stiff equations is difficult for many numerical methods. We test our newly derived implicit Runge-Kutta methods on the stiff initial value problem presented in [1, 20] and compare the results with the standard implicit Runge-Kutta methods with the same number of stages.

Let us consider the IVP

$$y'(x) = -100y(x) + 99\mathrm{e}^{2x}, \;\; x \in [0, 0.5] \tag{4.3}$$

$$y(0) = 0 \tag{4.4}$$

with the exact solution

$$y(x) = \frac{33}{34}(\mathrm{e}^{2x} - \mathrm{e}^{-100x}).$$

We compute the errors given in (4.1) for the values $N = 8, 16, 32, 64, 128$, and the corresponding EOC as defined in (4.2).

Table 9 and Figure 4 indicates better performance of our newly derived implicit Runge-Kutta method with four stages nIRK4 compared to the standard implicit collocation method sIRK4

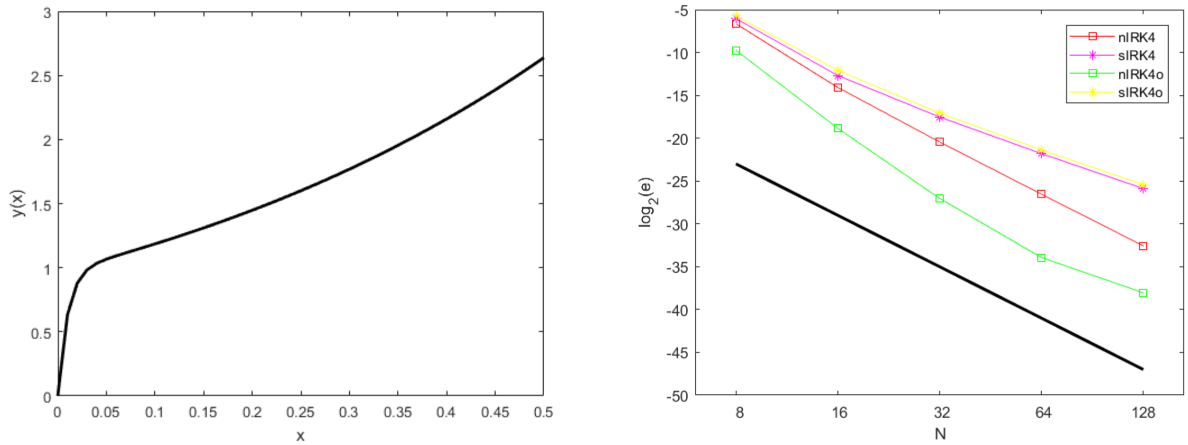

Figure 4: Experiment 2: exact solution (left), relative errors for nIRK4, sIRK4, nIRK4o, sIRK4o; black solid line represents reference slope N^{-6} (right)

when applied to the stiff initial value problem (4.3). EOC of our newly derived method nIRK4 is 6 compared to 4 of sIRK4. Moreover, we can observe that our "open" method nIRK4o has the same EOC=4 as sIRK4o. In particular, both newly derived methods approximate the exact solution with smaller errors for each N.

	nIRK4		sIRK4		nIRK4o		sIRK4o	
N	e	EOC	e	EOC	e	EOC	e	EOC
8	9.9661e-3	-	1.5156e-2	-	1.1745e-3	-	1.8775e-2	-
16	5.7748e-5	7.43	1.5192e-4	6.64	2.1165e-6	9.12	2.3186e-4	6.34
32	7.0353e-7	6.36	5.2735e-6	4.85	7.3150e-9	8.18	7.2459e-6	5.00
64	1.0255e-8	6.10	2.7296e-7	4.27	6.1244e-11	6.90	3.7359e-7	4.28
128	1.5761e-10	6.02	1.6223e-8	4.07	3.5340e-12	4.12	2.2199e-8	4.07

Table 9: Experiment 2: relative errors and EOC for nIRK4, sIRK4, nIRK4o, sIRK4o

4.3 Experiment 3: another stiff equation

In the third experiment we consider the IVP

$$\begin{aligned} y'(x) &= -50(y(x) - \cos x), \;\; x \in [0,1] \\ y(0) &= 0 \end{aligned}$$

with exact solution

$$y(x) = \frac{50(\sin x - 50\cos x)}{2501} - \frac{2500\mathrm{e}^{-50x}}{2501}.$$

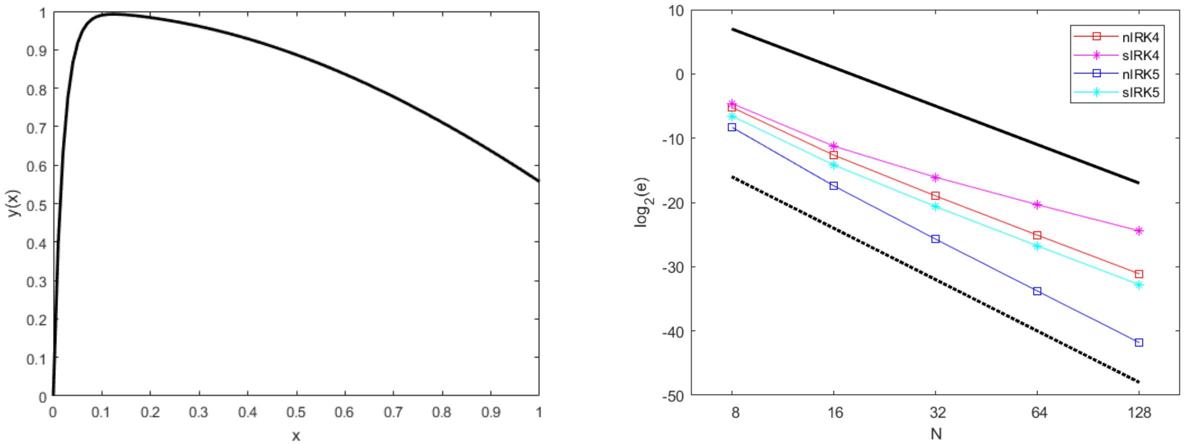

Figure 5: Experiment 3: exact solution (left), relative errors for nIRK4, sIRK4, nIRK5, sIRK5; black solid and dashed lines represent reference slopes N^{-6} and N^{-8} (right)

We perform computations for the values $N = 8, 16, 32, 64, 128$.

Relative errors and EOC presented in Table 10 confirm the same behaviour of methods as observed in Table 9 for the stiff IVP considered in Experiment 2. Namely, newly derived "closed" methods nIRK4 and nIRK5 approximate the exact solution with smaller relative errors and higher EOC (6 and 8) as compared to the standard collocation methods sIRK4 and sIRK5 (with EOC 4 and 6). See also Figure 5.

	nIRK4		sIRK4		nIRK5		sIRK5	
N	e	EOC	e	EOC	e	EOC	e	EOC
8	2.7288e-02	-	4.1498e-02	-	3.2139e-03	-	1.0728e-02	-
16	1.5812e-04	7.43	4.1596e-04	6.64	5.7563e-06	9.13	5.4447e-05	7.62
32	1.9263e-06	6.36	1.4439e-05	4.85	1.8247e-08	8.30	6.1304e-07	6.47
64	2.8079e-08	6.10	7.4739e-07	4.27	6.7591e-11	8.08	8.8112e-09	6.12
128	4.3147e-10	6.02	4.4420e-08	4.07	2.6056e-13	8.02	1.3480e-10	6.03

Table 10: Experiment 3: relative errors and EOC for nIRK4, sIRK4, nIRK5, sIRK5

4.4 Experiment 4: nonlinear stiff equation

In this experiment we consider a simple model of flame propagation,

$$
\begin{aligned}
&y'(x) = y^2(x) - y^3(x),\ x \in \left[0, \frac{2}{\delta}\right] \\
&y(0) = \delta,
\end{aligned}
$$

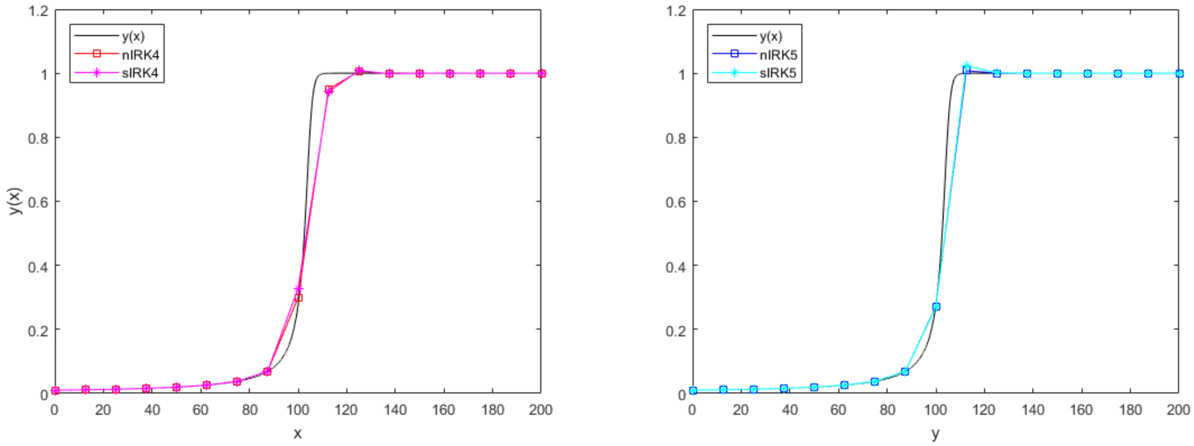

Figure 6: Experiment 4: exact and numerical solutions for $N = 16$

	nIRK4		sIRK4		nIRK5		sIRK5	
N	e	EOC	e	EOC	e	EOC	e	EOC
8	9.8732e-01	-	9.6914e-01	-	8.3152e-01	-	8.2708e-01	-
16	2.3943e-02	5.37	5.1641e-02	4.23	3.6226e-03	7.84	5.1735e-03	7.32
32	9.2683e-04	4.69	1.9117e-03	4.76	8.1630e-05	5.47	1.2471e-04	5.37
64	4.1418e-05	4.48	9.9775e-05	4.26	1.1144e-06	6.19	1.8035e-06	6.11
128	2.1913e-06	4.24	5.7525e-06	4.12	1.3187e-08	6.40	2.3021e-08	6.30

Table 11: Experiment 4: relative errors and EOC for nIRK4, sIRK4, nIRK5, sIRK5

where $\delta << 0$. The exact solution is given by

$$y(x) = \frac{1}{W(A\mathrm{e}^{A-x}) + 1},$$

where $A = 1/\delta - 1$ and $W(t)$ is the so-called Lambert W function [11]. It solves the equation $W(t)\mathrm{e}^{W(t)} = t$. We compute the above IVP by nIRK4, sIRK4 and nIRK5, sIRK5 for $\delta = 0.01$. Figure 6 depicts graphs of exact and numerical solutions computed for $N = 16$. The relative errors and EOC for $N = 2^k$, $k = 3, 4, 5, 6$ are presented in Table 11. See also Figure 7. We can observe smaller relative errors of our newly derived IRK compared to standard collocation IRK. The EOC confirms theoretical order of convergence, $p = 4$ for selected four stage methods and $p = 6$ for selected five stage methods.

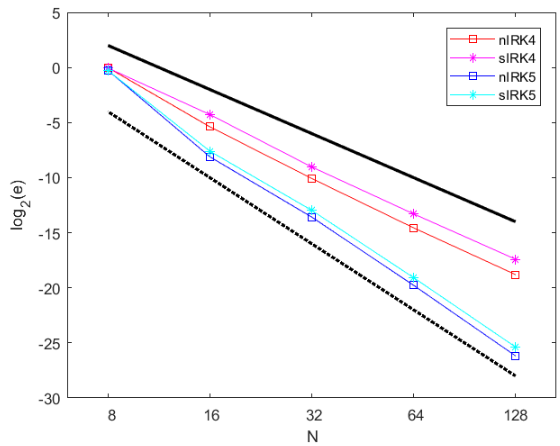

Figure 7: Experiment 4: relative errors for nIRK4, sIRK4, nIRK5, sIRK5; black solid and dashed lines represent reference slopes N^{-4} and N^{-6}, respectively

Conclusion

We have proposed the new way of deriving coefficients of implicit Runge-Kutta methods used for numerical approximation of initial value problems (1.1). Instead of standard collocation approach or the general approach based on simplifying order conditions as described in Subsection 2.1 we have imposed identities for the moments of $y'(x)$ and $f(x, y(x))$. The number of resulting equations containing repeated integrals depends on the number of stages s and the numerical quadrature being used, see Section 3. We have derived several Butcher's tableaux based on closed and open modified Newton-Cotes quadrature formulas with and without employing Cauchy's repeated integration formula, as well as based on Gauss-Legendre, -Radau and -Lobatto quadrature formulas. Various numerical experiments performed for selected newly derived implicit Runge-Kutta methods have confirmed their theoretical order of convergence indicated in Tables A1 and A2. Moreover, the latter methods are A-stable and yield approximation of solutions to considered linear and nonlinear stiff initial value problems with higher accuracy than the standard collocation implicit Runge-Kutta methods with the same number of stages. Especially, recall nIRKs with EOC $2s-2$ for linear IVPs which is typical for Lobatto methods. While nIRK2 and nIRK3 indeed coincide with Lobatto IIIA methods, nIRK4 and nIRK5 *do not* belong to family of Lobatto methods.

References

[1] R. R. Ahmad, N. Yaacob, A.-H. Mohd Murid. Explicit methods in solving stiff ordinary differential equations. *Int. J. Comput. Math.* **81**(11): 1407–1415, 2004.

[2] R. L. Burden, J. D. Faires. Numerical Analysis, 5th edition. PWS-Kent Publishing Company, 1993.

[3] J. C. Butcher. A history of Runge-Kutta methods. *Appl. Numer. Math.* **20**: 247–260, 1996.

[4] J. C. Butcher. Coefficients for the Study of Runge-Kutta Integration Processes. *J. Aust. Math. Soc* **3**: 185–201, 1963.

[5] J. C. Butcher. Integration Processes Based on Radau Quadrature Formulas. *Math. Comput.* **18**: 233–244, 1964.

[6] J. C. Butcher. Implicit Runge-Kutta Processes. *Math. Comput.* **18**: 50–64, 1964.

[7] J. C. Butcher. Numerical Methods for Ordinary Differential Equations. Wiley, 2003.

[8] J. C. Butcher. Numerical Methods for Ordinary Differential Equations. 2nd edition. Wiley, 2008.

[9] J. C. Butcher. The non-existence of ten stage eighth order explicit Runge-Kutta methods. *BIT* **25**: 521–540, 1985.

[10] A. L. Cauchy. Trente-Cinquiéme Leçn. In: Résum é des leçns données á l'Ecole royale polytechnique sur le calcul infinitésimal. Imprimerie Royale, Paris 1823.

[11] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, D. E. Knuth. On the Lambert W Function. *Adv. Comput. Math.* **5**: 329–359, 1996.

[12] J. R. Dormand. Numerical Methods for Differential Equations. Boca Raton CRC Press, 1996.

[13] L. Ferracina, M.N. Spijker. Strong stability of singly-diagonally-implicit Runge-Kutta methods. *Appl. Numer. Math.* **58**(11): 1675–1686, 2008.

[14] G. B. Folland. Advanced Calculus. Prentice Hall, 2002.

[15] G. K. Gupta, R. Sacks-Davis, T. E. Tischer. A review of recent development in solving ODEs. *Comput. Surv.* **17**(1): 5–47, 1985.

[16] E. Hairer, S. P. Nørsett, G. Wanner. Solving ordinary differential equations I.: Nonstiff problems. *Springer Series in Computational Mathematics* **8**, Springer, 1993.

[17] E. Hairer, G. Wanner. Solving ordinary differential equations II.: Stiff and Differential-Algebraic Problems. *Springer Series in Computational Mathematics* **14**, Springer, 1996.

[18] C. Kennedy, M. Carpenter. Diagonally Implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review. NASA/TM-2016-219173. NASA Langley Research Center, 2016.

[19] P. K. Maurya, V. S. Yadav, B. Mahato, N. Ganta, M. K. Rajpoot, Y. G. Bhumkar. New optimized implicit-explicit Runge-Kutta methods with applications to the hyperbolic conservation laws. *Journal of Computational Physics* **446**, 2021.

[20] H. Ramos. A non-standard explicit integration scheme for initial-value problems. *Appl. Math. Comput.* **189**(1): 710–718, 2007.

[21] J. A. dit Sandretto. Runge–Kutta Theory and Constraint Programming. *arXiv preprint* No. 1804.04847, 2018.

[22] L. M. Skvortsov. A fifth order implicit method for the numerical solution of differential algebraic equations. *Comput. Math. and Math. Phys.* **55**: 962–968, 2015.

[23] K. Tvrdá, M. Minárov a. Computation of definite integral over repeated integral. *Tatra Mt. Math. Publ.* **72**: 141–154, 2018.

[24] K. Tvrdá, P. Novotný. Modifications of Newton-Cotes formulas for computation of repeated integrals and derivatives. *Czech. Math. J.* **73**(148): 1175–1188, 2023.

[25] J. Urevc, B. Starman, A. Maček, M. Halilovič. A novel class of collocation methods based on the weighted integral form of ODEs. *Comp. Appl. Math.* **40**(135), 2021.

[26] Y. Ying, N. Yaacob. Implicit 7-stage Tenth Order Runge-Kutta Methods Based on Gauss-Kronrod-Lobatto Quadrature Formula. *MATEMATIKA: MJIAM* **31**(1): 93–109, 2015.

[27] D. W. Zingg, T.T. Chisholm. Runge-Kutta methods for linear ordinary differential equations. *Appl. Numer. Math.* **31**: 227–238, 1999.

[28] Z. Zlatev. Modified Diagonally Implicit Runge–Kutta Methods. *SIAM J. Sci. Stat. Comp.* **2**(3): 321–334, 1981.

Appendix

We provide the list of newly derived as well as of some well-known implicit Runge-Kutta methods. The tables have the following columns: name/type of method, Butcher's tableau, theoretical order of convergence and values of p, q, r corresponding to simplifying order conditions (2.5) satisfied by the underlying methods.

A1 List of new IRK based on closed Newton-Cotes formulas

Table A1 contains new IRK derived by modified closed Newton-Cotes formulas, cf. Subsection 3.1 for derivation of nIRKs and nIRKsc. There is an additional column with the corresponding stability functions $R(z)$ in Table A1.

A2 List of new IRK based on open Newton-Cotes formulas

Table A2 contains new IRK derived by modified open Newton-Cotes formulas, cf. Subsection 3.2 for derivation of nIRKso and nIRK$s$oc.

A3 List of Lobatto IIIA methods

Table A3 contains Lobatto IIIA methods with $s = 2, 3, 4, 5$ stages. The first column indicates which of the proposed methods to derive coefficients result in the well-known Lobatto IIIA for given s. See Subsection 3.3.

A4 List of Gauss-Legendre and Radau methods

Table A4 contains Radau and Gauss-Legendre methods with $s = 2, 3$ stages. Again, the first column contains the well-known name and the acronym of newly proposed approach which results in the same Butcher's tableau. See Subsection 3.3.

A5 List of standard collocation IRK

Table A5, for completeness, contains the list of standard collocation IRK which were used in our numerical experiments. For the derivation of these methods see Subsubsection 2.2.1.

name	s	Butcher's tableau	p	$p\ q\ r$	stability function
nIRK4	4	$\begin{array}{c\|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{47}{360} & \frac{89}{360} & -\frac{19}{360} & \frac{1}{120} \\ \frac{2}{3} & \frac{7}{60} & \frac{77}{180} & \frac{23}{180} & -\frac{1}{180} \\ 1 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$	4	4 3 0	$-\dfrac{\frac{z^3}{120}+\frac{z^2}{10}+\frac{z}{2}+1}{\frac{z^3}{120}-\frac{z^2}{10}+\frac{z}{2}-1}$
nIRK4c	4	$\begin{array}{c\|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{24} & \frac{5}{24} & -\frac{1}{4} & \frac{1}{24} \\ \frac{2}{3} & \frac{1}{12} & \frac{5}{12} & \frac{1}{4} & -\frac{1}{12} \\ 1 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$	4	4 2 2	$-\dfrac{\frac{z^3}{72}+\frac{z^2}{9}+\frac{z}{2}+1}{\frac{z^3}{72}-\frac{z^2}{9}+\frac{z}{2}-1}$
nIRK5	5	$\begin{array}{c\|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{200}{2183} & \frac{429}{2078} & -\frac{109}{1680} & \frac{191}{10080} & -\frac{43}{20160} \\ \frac{1}{2} & \frac{29}{360} & \frac{31}{90} & \frac{1}{15} & \frac{1}{90} & -\frac{1}{360} \\ \frac{3}{4} & \frac{179}{2240} & \frac{377}{1120} & \frac{111}{560} & \frac{167}{1120} & -\frac{31}{2240} \\ 1 & \frac{7}{90} & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90} \\ \hline & \frac{7}{90} & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90} \end{array}$	6	6 4 1	$\dfrac{\frac{z^4}{1680}+\frac{z^3}{84}+\frac{3z^2}{28}+\frac{z}{2}+1}{\frac{z^4}{1680}-\frac{z^3}{84}+\frac{3z^2}{28}-\frac{z}{2}+1}$
nIRK5c	5	$\begin{array}{c\|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{371}{2880} & \frac{79}{720} & \frac{1}{480} & \frac{19}{720} & -\frac{49}{2880} \\ \frac{1}{2} & -\frac{7}{120} & \frac{28}{45} & \frac{1}{15} & -\frac{14}{15} & \frac{49}{360} \\ \frac{3}{4} & \frac{91}{960} & \frac{79}{240} & \frac{21}{160} & \frac{59}{240} & -\frac{49}{960} \\ 1 & \frac{7}{90} & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90} \\ \hline & \frac{7}{90} & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90} \end{array}$	6	6 3 3	$\dfrac{\frac{7z^4}{5760}+\frac{z^3}{64}+\frac{11z^2}{96}+\frac{z}{2}+1}{\frac{7z^4}{5760}-\frac{z^3}{64}+\frac{11z^2}{96}-\frac{z}{2}+1}$

Table A1: nIRKs and nIRKsc based on closed Newton-Cotes formulas for repeated integrals

<table>
<tr><th>name</th><th>s</th><th>Butcher's tableau</th><th>p</th><th>$p\ q\ r$</th></tr>
<tr><td>nIRK3o</td><td>3</td><td>$$\begin{array}{c|ccc} \frac{1}{4} & \frac{101}{240} & -\frac{13}{60} & \frac{11}{240} \\ \frac{1}{2} & \frac{7}{12} & -\frac{1}{6} & \frac{1}{12} \\ \frac{3}{4} & \frac{149}{240} & -\frac{7}{60} & \frac{59}{240} \\ \hline & \frac{2}{3} & -\frac{1}{3} & \frac{2}{3} \end{array}$$</td><td>4</td><td>4 2 1</td></tr>
<tr><td>nIRK3oc</td><td>3</td><td>$$\begin{array}{c|ccc} \frac{1}{4} & \frac{3}{16} & -\frac{1}{24} & \frac{5}{48} \\ \frac{1}{2} & 0 & -\frac{1}{6} & \frac{2}{3} \\ \frac{3}{4} & \frac{9}{16} & -\frac{7}{24} & \frac{23}{48} \\ \hline & \frac{2}{3} & -\frac{1}{3} & \frac{2}{3} \end{array}$$</td><td>4</td><td>4 1 3</td></tr>
<tr><td>nIRK4o</td><td>4</td><td>$$\begin{array}{c|cccc} \frac{1}{5} & \frac{1340}{3131} & -\frac{538}{1343} & \frac{73}{336} & -\frac{6677}{149565} \\ \frac{2}{5} & \frac{1061}{1927} & -\frac{823}{2137} & \frac{107}{336} & -\frac{47}{560} \\ \frac{3}{5} & \frac{571}{1053} & -\frac{31}{112} & \frac{239}{560} & -\frac{31}{336} \\ \frac{4}{5} & \frac{761}{1513} & -\frac{2442}{13907} & \frac{743}{1680} & \frac{902}{29713} \\ \hline & \frac{11}{24} & \frac{1}{24} & \frac{1}{24} & \frac{11}{24} \end{array}$$</td><td>4</td><td>4 3 0</td></tr>
<tr><td>nIRK4oc</td><td>4</td><td>$$\begin{array}{c|cccc} \frac{1}{5} & \frac{719}{1680} & -\frac{673}{168} & \frac{929}{4276} & -\frac{5}{112} \\ \frac{2}{5} & \frac{185}{336} & -\frac{647}{1680} & \frac{485}{1523} & -\frac{47}{560} \\ \frac{3}{5} & \frac{911}{1680} & -\frac{31}{112} & \frac{376}{881} & -\frac{31}{336} \\ \frac{4}{5} & \frac{169}{336} & -\frac{59}{336} & \frac{789}{1784} & \frac{17}{560} \\ \hline & \frac{11}{24} & \frac{1}{24} & \frac{1}{24} & \frac{11}{24} \end{array}$$</td><td>4</td><td>4 0 3</td></tr>
</table>

Table A2: nIRKso and nIRK$s$oc based on open Newton-Cotes formulas for repeated integrals

<table>
<tr><th>name</th><th>s</th><th>Butcher's tableau</th><th>p</th><th>$p\ q\ r$</th></tr>
<tr><td>Lobato IIIA
nIRK2
nIRK2c
sIRK2</td><td>2</td><td>$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$</td><td>2</td><td>2 2 0</td></tr>
<tr><td>Lobato IIIA
nIRK3
nIRK3c
nIRK-L3
sIRK3</td><td>3</td><td>$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$</td><td>4</td><td>4 3 1</td></tr>
<tr><td>Lobato IIIA
nIRK-L4</td><td>4</td><td>$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{5}}{10} & \frac{11+\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & \frac{-1+\sqrt{5}}{120} \\ \frac{1}{2} + \frac{\sqrt{5}}{10} & \frac{11-\sqrt{5}}{120} & \frac{25+13\sqrt{5}}{120} & \frac{25+5\sqrt{5}}{120} & \frac{-1-\sqrt{5}}{120} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$</td><td>6</td><td>6 4 2</td></tr>
<tr><td>Lobato IIIA
nIRK-L5</td><td>5</td><td>$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{21}}{14} & \frac{119+3\sqrt{21}}{1960} & \frac{343-9\sqrt{21}}{2520} & \frac{392-96\sqrt{21}}{2205} & \frac{343-69\sqrt{21}}{2520} & \frac{-21+3\sqrt{21}}{1960} \\ \frac{1}{2} & \frac{13}{320} & \frac{392+105\sqrt{21}}{2880} & \frac{8}{45} & \frac{392-105\sqrt{21}}{2880} & \frac{3}{320} \\ \frac{1}{2} + \frac{\sqrt{21}}{14} & \frac{119-3\sqrt{21}}{1960} & \frac{343+69\sqrt{21}}{2520} & \frac{392+96\sqrt{21}}{2205} & \frac{343+9\sqrt{21}}{2520} & \frac{-21-3\sqrt{21}}{1960} \\ 1 & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \\ \hline & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \end{array}$</td><td>8</td><td>8 5 3</td></tr>
</table>

Table A3: Lobatto IIIA methods

name	s	Butcher's tableau	p	$p\ q\ r$
Radau I nIRK-RI2	2	$\begin{array}{c\|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$	3	3 2 1
Radau IIA nIRK-RII2	2	$\begin{array}{c\|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$	3	3 2 1
Gauss-Legendre nIRK-G2	2	$\begin{array}{c\|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{3+\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$	4	4 2 2
Radau I nIRK-RI3	3	$\begin{array}{c\|ccc} 0 & 0 & 0 & 0 \\ \frac{6-\sqrt{6}}{10} & \frac{9+\sqrt{6}}{75} & \frac{24+\sqrt{6}}{120} & \frac{168-73\sqrt{6}}{600} \\ \frac{6+\sqrt{6}}{10} & \frac{9-\sqrt{6}}{75} & \frac{168+73\sqrt{6}}{600} & \frac{24-\sqrt{6}}{120} \\ \hline & \frac{1}{9} & \frac{16+\sqrt{6}}{36} & \frac{16-\sqrt{6}}{36} \end{array}$	5	5 3 2
Radau IIA nIRK-RII3	3	$\begin{array}{c\|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$	5	5 3 2
Gauss-Legendre nIRK-G3	3	$\begin{array}{c\|ccc} \frac{5-\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9}-\frac{\sqrt{15}}{15} & \frac{5}{36}-\frac{\sqrt{15}}{30} \\ \frac{1}{2} & \frac{5}{36}+\frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36}-\frac{\sqrt{15}}{24} \\ \frac{5+\sqrt{15}}{10} & \frac{5}{36}+\frac{\sqrt{15}}{30} & \frac{2}{9}+\frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{15}{18} \end{array}$	6	6 3 3

Table A4: Radau and Gauss-Legendre methods

name	s	Butcher's tableau	p	$p\ q\ r$
sIRK3o	3	$\begin{array}{c\|ccc} \frac{1}{4} & \frac{23}{48} & -\frac{1}{3} & \frac{5}{48} \\ \frac{1}{2} & \frac{7}{12} & -\frac{1}{6} & \frac{1}{12} \\ \frac{3}{4} & \frac{9}{16} & 0 & \frac{3}{16} \\ \hline & \frac{2}{3} & -\frac{1}{3} & \frac{2}{3} \end{array}$	4	4 3 1
sIRK4o	4	$\begin{array}{c\|cccc} \frac{1}{5} & \frac{11}{24} & -\frac{59}{120} & \frac{37}{120} & -\frac{3}{40} \\ \frac{2}{5} & \frac{8}{15} & -\frac{1}{3} & \frac{4}{15} & -\frac{1}{15} \\ \frac{3}{5} & \frac{21}{40} & -\frac{9}{40} & \frac{3}{8} & -\frac{30}{40} \\ \frac{4}{5} & \frac{8}{15} & -\frac{4}{15} & \frac{8}{15} & 0 \\ \hline & \frac{11}{24} & \frac{1}{24} & \frac{1}{24} & \frac{11}{24} \end{array}$	4	4 4 0
sIRK4	4	$\begin{array}{c\|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{8} & \frac{19}{72} & -\frac{5}{72} & \frac{1}{72} \\ \frac{2}{3} & \frac{1}{9} & \frac{4}{9} & \frac{1}{9} & 0 \\ 1 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ \hline & \frac{11}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$	4	4 4 0
sIRK5	5	$\begin{array}{c\|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{251}{2880} & \frac{323}{1440} & -\frac{11}{120} & \frac{53}{1440} & -\frac{19}{2880} \\ \frac{1}{2} & \frac{29}{360} & \frac{31}{90} & \frac{1}{15} & \frac{1}{90} & -\frac{1}{360} \\ \frac{3}{4} & \frac{27}{320} & \frac{51}{160} & \frac{9}{40} & \frac{21}{160} & -\frac{3}{320} \\ 1 & \frac{7}{90} & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90} \\ \hline & \frac{7}{90} & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90} \end{array}$	6	6 5 1

Table A5: standard collocation methods sIRKso and sIRKs