Runge-Kutta methods of special form

L. Gr. Ixaru

“Horia Hulubei” National Institute of Physics and Nuclear Engineering,
Department of Theoretical Physics,
P.O.Box MG-6, Bucharest, Romania
e-mail: ixaru@theory.nipne.ro

and

Academy of Romanian Scientists,
54 Splaiul Independenței, 050094, Bucharest, Romania
E-mail: ixaru@theory.nipne.ro

Abstract. Many problems in physics and engineering require the solution of stiff systems of
differential equations and only A-stable numerical methods are appropriate for this purpose.
The theory covering the methods in current use, which are methods with constant coefficients,
says that A-stable methods of order greater than 2 are necessarily implicit, but in practice
this asks for time consuming iteration procedures. We present versions of explicit Runge-Kutta
methods of a different form: their coefficients are no more constant but equation dependent.
Due to this peculiarity the new versions are A-stable, such that the iteration process is avoided.
We also give a numerical illustration.

1. Introduction
Quite often the theoretical approach of problems in physics and engineering requires the
numerical solution of differential equations. The literature on such methods is quite vast and
the way of selecting the version which is best suited for the given problem is not always so
simple. The selection is typically restricted on the class of the most popular methods, and the
Runge Kutta methods belong to this. Still, the methods in the Runge Kutta (RK) family are
not equivalent for performances. Some are convenient when the accuracy is an issue while others
act better for stiff problems. In this paper we present some recently formulated versions which
offer a reasonable balance between the two.
To fix the ideas, the one-step problem to solve is

\[ y'(x, y) = f(x, y), \quad x \in [x_n, x_{n+1} = x_n + h], \quad y(x_n) = y_n, \]

and the general algorithm of an s-stage RK method is (see [1], [2])

\[ y_{n+1} = y_n + h \sum_{i=1}^{s} b_i f(x_n + c_i h, Y_i), \]

where

\[ Y_i = y_n + h \sum_{j=1}^{s} a_{ij} f(x_n + c_j h, Y_j), \quad i = 1, 2, ..., s. \]
This allows computing $y_{n+1}$ in terms of the input $y_n$ by the formula written in the first line (so called final stage), where the values of $Y_i$ are as resulting from the set of formulae in the second line (internal stages). The coefficients are usually collected in the Butcher array

$$
\begin{array}{c|cccc}
  c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\
  c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\
\hline
  & b_1 & b_2 & \cdots & b_s
\end{array}
$$

If at least one $a_{ij}$ with $j \geq i$, $i = 1, 2, \ldots, s$, is nonvanishing, then the method is called implicit. Otherwise it is explicit, and if only $a_{i,i-1}$, $i = 2, 3, \ldots, s$, are nonvanishing the method is called diagonally explicit.

**Examples**

1. The wellknown three-stage Kutta method is an explicit method. It has the Butcher array

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

and its order is three.

2. Also of order three is the three-stage diagonally explicit version

$$
\begin{array}{c|cccc}
  0 & 0 & 0 & 0 \\
  c_2 & a_{21} & 0 & 0 \\
  c_3 & 0 & a_{32} & 0 \\
\hline
  & b_1 & b_2 & b_3
\end{array}
$$

if $c_2$ and $c_3$ are correlated, $c_3 = 3c_2(1 - c_2)$, and

$$
a_{21} = c_2, \ a_{32} = c_3, \ b_2 = \frac{3c_2 - 1}{6c_2^2}, \ b_3 = -\frac{1}{18c_2^2(c_2 - 1)}, \ b_1 = 1 - b_2 - b_3,
$$

see [2].

3. The standard four-stage Runge-Kutta method (abbreviated RK4) is diagonally explicit. Its Butcher array is

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

and its order is four. As a matter of fact, if the number of stages is not larger than four, then diagonally explicit versions exist such that their order is equal with the number of stages.

The main element of attractivity with the explicit methods is that they are easy to use: the computation of each new $Y_i$ requires only data available from the previous stages. This is in contrast with the implicit methods where all $Y_i$ are coupled and therefore their computation implies solving a set of nonlinear equations. An iteration procedure is then required, whose success depends on how good are the starting data. Since for methods with constant coefficients only implicit versions can hopefully solve stiff problems, the need for iterations cannot be avoided, and this is a serious difficulty when solving such problems.
2. New versions

What is special with these versions is that their coefficients are no longer constant. They are equation dependent. Versions of this type have been obtained recently in [3] and [4] by means of the exponential fitting formalism. Technical details are not given here, but we list the main results.

2.1. Two-stage explicit method

The reference Butcher array is:

\[
\begin{array}{ccc}
0 & 0 & 0 \\
c_2 & a_{21} & 0 \\
0 & b_1 & b_2 \\
\end{array}
\]

With \( c_2 \in (0, 1] \) as a free parameter the coefficients are:

(i) Standard version:

\[
a_{21} = c_2, \quad b_1 = 1 - 1/2c_2, \quad b_2 = 1 - b_1.
\]

(ii) New version:

\[
a_{21} = c_2, \quad b_1 = 1 - 1/[2c_2(1 - c_2M_2/2)], \quad b_2 = 1 - b_1,
\]

where \( M_2 = h f_y(x_n + c_2h, y)|_{y=Y_2} \).

Remarks

1. Both versions have order two for all \( c_2 \). However, if \( c_2 = 2/3 \) the order of new version becomes three.
2. Due to the presence of the equation dependent \( M_2 \) the coefficients \( b_1 \) and \( b_2 \) are also equation dependent and thus they must be updated at each step.
3. For systems of differential equations \( M_2 \) becomes \( h \) times the Jacobian matrix. Also, \( b_1 \) and \( b_2 \) become matrices.

2.1.1. Stability

A-stability is a necessary condition for a method in order to be successful on stiff problems. The way how a method is checked for stability is based on a simple test equation. This is

\[
y' = \lambda y, \quad y \geq 0, \quad y(0) = y_0
\]

with the exact solution \( y(x) = y_0 \exp(\lambda x) \). If \( \text{Re}(\lambda) < 0 \) the solution tends to zero when \( x \to \infty \) and the same must hold true also for the numerical solution. Since \( f(x, y) = \lambda y \), the first and second internal stages give \( Y_1 = y_n \) and \( Y_2 = (1 + c_2\nu)y_n \), respectively, where \( \nu = \lambda h \), such that the final stage results in \( y_{n+1} = R(\nu)y_n \) where \( R(\nu) = (1 + \nu + b_2c_2\nu^2) \). The solution propagated numerically from 0 up to \( x_N = Nh \) is \( y_N = R^N(\nu)y_0 \) and the condition that \( y_N \to 0 \) if \( N \to \infty \) is satisfied iff \( |R(\nu)| < 1 \) for all \( \nu \) with negative real part. A method with this property for \( R(\nu) \) is called A-stable.

With the parameters of the new method, and by taking into account the fact that \( M_2 = \nu \), the stability function \( R(\nu) \) reads

\[
R_f(\nu) = \frac{2 + (2 - c_2)\nu + (1 - c_2)\nu^2}{2 - c_2\nu}.
\]

When \( c_2 = 1 \) this becomes

\[
R_f(\nu) = \frac{2 + \nu}{2 - \nu},
\]

a form which clearly satisfies the mentioned condition. The new version with \( c_2 = 1 \) is therefore an A-stable method.
2.2. Three-stage diagonally explicit method

The reference Butcher array is now as in Example 2 above. A family of versions with equation dependent coefficients was developed in [4] but two versions are of particular interest for practice. These are:

1. $c_2 = 1/2$, $c_3 = 1$, $b^{den} = 12 - 4M_2 - 2M_3 + M_3M_2$ and
   \[ b_1 = \frac{2 - 3M_2}{b^{den}}, \quad b_2 = \frac{8 - 2M_3 + M_3M_2}{b^{den}}, \quad b_3 = \frac{2 - M_2}{b^{den}}. \quad (5) \]

2. $c_2 = 1$, $c_3 = 1/2$, $b^{den} = 12 - 2M_2 - 4M_3 + M_3M_2$ and
   \[ b_1 = \frac{2 - 2M_2 + M_3 + M_3M_2}{b^{den}}, \quad b_2 = \frac{2 - 5M_3}{b^{den}}, \quad b_3 = \frac{8}{b^{den}}. \quad (6) \]

where $M_i(x) = h f_j(x + c_i h, y)|_{y=Y_j(x)}$. In both versions $a_{21} = c_2$ and $a_{22} = c_3$.

As before, for systems of equations $M_i$ and $b_i$ become matrices. The formulae are the same: the matrix products have to be carried out in the written order and the division by $b^{den}$ must be replaced by multiplication on the right by matrix $b^{den}^{-1}$.

The special importance of these versions is realted to two properties:
(a) they are of order 4, that is higher by one unit than the classical three-stage methods with the same Butcher array;
(b) they are A-stable, with one and the same stability function,
\[ R_f(\nu) = \frac{12 + 6\nu + \nu^2}{12 - 6\nu + \nu^2}. \quad (7) \]

A general remark. Any discussion on the properties of a numerical method tacitly accepts that the method is sharply defined. Mentioning this is redundant for methods with constant coefficients but we have to be careful in the case of methods with equation dependent coefficients. In fact, in the latter case the existence of the method depends on how accurate is the computation of the coefficients. Seen from this perspective, situations may appear when the accurate determination of these parameters becomes a problem. Think for example on eqs. (5) and (6) which, for systems of equations, require inversion of matrix $b^{den}$; if this is singular the inversion is impossible and then the method is not defined.

3. Numerical illustration

The system of two differential equations
\[ y'' = (10\lambda + 9)y' - 10(\lambda + 1)y^2, \quad y'' = -9(\lambda + 1)y' - (9\lambda + 10)y^2, \]
\[ x \in [0, 5], \quad y'(0) = y_0^1, \quad y'(0) = y_0^2, \]
has the exact solution:
\[ y^1(x) = 10(y_0^1 - y_0^2)e^{\lambda x} + (-9y_0^1 + 10y_0^2)e^{-x}, \quad y^2(x) = 9(y_0^1 - y_0^2)e^{-x} + (9y_0^1 + 10y_0^2)e^{-x}. \]

For reasonable sized values of the initial conditions, and when $\lambda$ is negative and big, the solution is largely dominated in magnitude by the component with $e^{-x}$. Yet, the presence of the component with $e^{\lambda x}$ seriously affects the numerical solution if the problem is solved by a method with low stability properties.

To illustrate this phenomenon we take the initial conditions $y_0^1 = y_0^2 = 1$, for which the exact solution is independent of $\lambda$,
\[ y^1(x) = y^2(x) = e^{-x}. \]
Table 1. Absolute error $err = y_1^{\text{exact}}(5) - y_1^{\text{comput.}}(5)$ for the test equation.

| $h$  | RK4     | New method, eq.(5) |
|------|---------|---------------------|
| 1/ 2 | 8.51(+57) | -4.41(-04)          |
| 1/ 4 | -7.82(+83) | -2.72(-05)          |
| 1/ 8 | 4.11(+216) | -1.70(-06)          |
| 1/16 | NaN      | -1.06(-07)          |
| 1/32 | NaN      | -6.62(-09)          |
| 1/64 | NaN      | -4.12(-10)          |
| 1/128| NaN      | -2.55(-11)          |
| 1/256| -9.84(-12) | -1.13(-12)          |

We use $\lambda = -600$.

We compare two methods: standard four-stage method RK4 from example 3, with low stability properties, and the new three-stage method with $c_2 = 1/2$, eq.(5), which is A-stable. The errors in the values at the end point $x = 5$ of the component $y_1$ of the solution are collected on Table 1 for a set of step sizes. The errors in $y_2$ are identical. We see that the first method gives reliable results only for $h = 1/256$, which corresponds to $\nu = \lambda h \approx -2.34$. This is quite normal because this method is known to be stable only if $\nu$ is bigger than approximately $-2.8$. For contrast, no stability problem exists for the new method. Its order four is also confirmed by the data in Table 1.

In conclusion, replacement of the standard methods, based on constant coefficients, by versions with equation dependent coefficients leads to an improvement of the quality of the results in two respects:

(a) the accuracy is increased: the order is typically increased by one unit;

(b) there is a massive improvement for stability: the presented versions are A-stable.

This recommends the new versions for being included in the short list of potential solvers for differential equations.

Acknowledgments

I want to express my gratitude to Prof. A. Sandulescu for gently guiding my first steps in scientific research in the mid-sixties. The investigations presented in this paper benefited partially from financial support by the project PN-II-ID-PCE-2011-3-0092 of the Romanian Ministry of Education and Research.

References

[1] Butcher J C 2008 *Numerical Methods for Ordinary Differential Equations* (Chichester:Wiley)

[2] Lambert J D 1991 *Numerical methods for ordinary differential systems: The initial value problem* (Chichester:Wiley)

[3] D’Ambrosio R, Ixaru L Gr and Paternoster B 2011 *Comput. Phys. Commun.* 182 322

[4] Ixaru L Gr 2012 *Comput. Phys. Commun.* 183 63