APPROXIMATING RUNGE–KUTTA MATRICES BY TRIANGULAR MATRICES

W. HOFFMANN and J. J. B. DE SWART

1Department of Mathematics and Computer Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. email: walter@fwi.uva.nl
2Department of Numerical Mathematics, CWI, P.O. Box 94079
1090 GB Amsterdam, The Netherlands. email: jacques@cwi.nl

Abstract.

The implementation of implicit Runge–Kutta methods requires the solution of large systems of non-linear equations. Normally these equations are solved by a modified Newton process, which can be very expensive for problems of high dimension. The recently proposed triangularly implicit iteration methods for ODE-IVP solvers [5] substitute the Runge–Kutta matrix \( A \) in the Newton process for a triangular matrix \( T \) that approximates \( A \), hereby making the method suitable for parallel implementation. The matrix \( T \) is constructed according to a simple procedure, such that the stiff error components in the numerical solution are strongly damped. In this paper we prove for a large class of Runge–Kutta methods that this procedure can be carried out and that the diagonal entries of \( T \) are positive. This means that the linear systems that are to be solved have a non-singular matrix.


Key words: Numerical analysis, Runge–Kutta methods, Matrix analysis.

1 Introduction and motivation.

For solving the stiff initial value problem

\[
y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad y, f \in \mathbb{R}^d, \quad t_0 \leq t \leq t_e,
\]

one of the most powerful methods is an implicit Runge–Kutta (RK) method. In such a method we have to solve every time step a system of non-linear equations of the form

\[
(1.1) \quad R(Y_n) = 0; \quad R(Y_n) := Y_n - (e \otimes I)y_{n-1} - h_n(A \otimes I)F(Y_n),
\]

where \( A \) denotes the \( s \times s \) matrix containing the parameters of the \( s \)-stage RK method, \( y_{n-1} \) the approximation to \( y(t_{n-1}) \), \( e \) is the \( s \)-dimensional vector with unit entries, \( I \) is the \( d \times d \) identity matrix, \( h_n \) is the step size \( t_n - t_{n-1} \) and \( \otimes \)
APPROXIMATING RUNGE-KUTTA MATRICES

347

denotes the Kronecker product. The $s$ components $Y_{n,i}$ of the $sd$-dimensional solution vector $Y_n$ represent $s$ numerical approximations to the $s$ exact solution vectors $y(t_{n-1} + c_i h_n)$; here, $c$ denotes the abscissa vector and $i$ ranges from 1 to $s$. Furthermore, for any vector $X = (X_i)$, $F(X)$ contains the derivative values $(f(X_i))$. It is assumed that the components of $c$ are distinct and positive.

Once we have solved (1.1), we obtain the step point value $y_n \approx y(t_n)$ by the formula

$$y_n = y_{n-1} + h_n (b^T \otimes I) F(Y_n),$$

where $b$ is a vector of dimension $s$ containing method parameters.

To solve (1.1), in general one uses a Newton-type iteration scheme of the form

$$y^{(j+1)} = y^{(j)} + A y^{(j+1)} = y^{(j)} + \Delta Y^{(j+1)};$$

where $J_n$ is an approximation to the Jacobian of the right hand side function $f$ at $t_{n-1}$, $Y_n^{(0)}$ is the initial iterate to be provided by some predictor formula and $B$ is an $s \times s$ matrix that defines the type of Newton iteration. To get insight in the convergence behaviour of (1.2), we apply the scheme to the scalar test equation $y' = \lambda y$. Defining the iteration error $e_n^{(j)}$ by $Y^{(j)} - Y_n$, we see from (1.1) and (1.2) that these errors are amplified by the matrix $Z$ defined by

$$Z(z) = z(I - zB)^{-1}(A - B); \quad z := \lambda h_n.$$

We introduce the stiff and non-stiff amplification matrices of scheme (1.2), notation $Z_{\infty}(B)$ and $Z_0(B)$, respectively, by

$$Z_{\infty}(B) := \lim_{|z| \to \infty} Z(z) = I - B^{-1}A$$

and

$$Z_0(B) := \lim_{|z| \to 0} \frac{Z(z)}{|z|} = A - B.$$

Choosing $B = A$ would lead to the modified Newton process, for which $Z(z) = 0$ for all $z$. However, the computation of $Y_n^{(j)}$ now requires the solution of a linear system of dimension $sd$. For high-dimensional problems this requires a lot of computational effort. Several attempts have been made to reduce these costs by selecting matrices $B$ different from $A$.

In [1], Cooper and Butcher propose the choice $B = P$, where $P$ is a matrix that has a one-point spectrum. By performing a similarity transformation to (1.2) they arrive at the scheme

$$(I - L \otimes h_n J_n) \Delta X_n^{(j+1)} = -(Q^{-1} \otimes I) R(Y_n^{(j)}),$$

$$Y_n^{(j+1)} = Y_n^{(j)} + (Q \otimes I) \Delta X_n^{(j+1)},$$

where $L$ and $Q$ are lower triangular and orthogonal matrices, respectively, that define the Schur decomposition of $P$. Since the diagonal entries of $L$ are equal, implementing (1.3) requires only one $LU$-decomposition of dimension $d$. 