Received: 25 June 2021 | Accepted: 20 August 2021

DOI: 10.1002/pamm.202100235

On Adaptive Patankar Runge–Kutta methods

Stefan Kopecz[1], **Andreas Meister**[1], and **Helmut Podhaisky**[2,*]

[1] University of Kassel, Department of Mathematics, Heinrich-Plett-Str. 40, 34132 Kassel, Germany
[2] University of Halle, Institute of Mathematics, Theodor-Lieser-Str. 5, 06120 Halle, Germany

We apply Patankar Runge–Kutta methods to $y' = M(y)y$ and focus on the case where $M(y)$ is a graph Laplacian as the resulting scheme will preserve positivity and total mass. The second order Patankar Heun method is tested using four test problems (stiff and non-stiff) cast into this form. The local error is estimated and the step size is chosen adaptively. Concerning accuracy and efficiency, the results are comparable to those obtained with a traditional L-stable, second order Rosenbrock method.

1 Definition of the Scheme

Consider the autonomous initial value problem

$$y'(t) = M(y(t))y(t), \quad y(t_0) = t_0, \quad t \in [t_0, t_{\text{end}}], \tag{1}$$

with $y(t) \in \mathbb{R}^n$ and $M(y(t)) \in \mathbb{R}^{n\times n}$. If $M(y)$ is a graph Laplacian, then positivity and total mass are conserved by the analytical solution $y(t)$, see [1] for details and other schemes.

A Patankar Runge–Kutta method for (1) takes the form

$$\begin{aligned} y^{(k)} &= y^n + h\sum_{\nu=1}^{k-1} a_{k\nu} M(y^{(\nu)})\,\mathrm{diag}(w_\nu)y^{(k)} \\ y^{n+1} &= y^n + h\sum_{k=1}^{s} b_k M(y^{(k)})\,\mathrm{diag}(\widehat{w})y^{n+1} \end{aligned} \tag{2}$$

where w_ν and $\widehat{w}$ are so-called Patankar weight. Autonomous production-destruction systems can be written in the form (1) and the scheme (2) is then equivalent to existing formulations of Patankar Runge-Kutta methods [2, 3].

2 Numerical Experiments

In this note, we focus exclusively on the robust second order Patankar Heun method [2]

$$\begin{aligned} y^{(2)} &= y^n + hM(y^n)y^{(2)}, \\ y^{n+1} &= y^n + \frac{h}{2}\left(M(y^n)\,\mathrm{diag}(\frac{y^n}{y^{(2)}}) + M(y^{(2)})\right)y^{n+1}. \end{aligned}$$

Schemes of order 3 (cf. [3]) and higher (cf. [4]) will be considered in future work.

The following problems are considered:

1. Lotka, A non-stiff problem of Lotka-Volterra type

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} 1-y_2 & 0 \\ y_2 & -1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad y(0) = [1, 2]^\top, \quad t \in [0, 10]. \tag{3}$$

   This problem does not originate from a production-destruction system (although a first integral exists) and hence $M(y)$ is not a graph Laplacian. Consequently, the Patankar method is not guaranteed to preserve quantities.

2. Robertson, cf. [5], with $k_1 = 0.04$, $k_2 = 3 \cdot 10^7$ and $k_3 = 10^4$ in the form

$$\begin{pmatrix} y_1' \\ y_2' \\ y_3' \end{pmatrix} = \begin{pmatrix} -k_1 & k_3y_2 & 0 \\ k_1 & -k_2y_2 - k_3y_3 & 0 \\ 0 & k_2y_2 & 0 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}, \quad y(0) = [1, 0, 0]^\top, \quad t \in [0, 10^5]. \tag{4}$$

   Here, $M(y)$ is a graph Laplacian. The numerical solution will be positive and the total mass is conserving. Usually, one has $t_{\text{end}} = 10^{11}$. This works, too – with smaller errors but we want to see the errors to be able to evaluate the adaptive stepping.

* Corresponding author: e-mail helmut.podhaisky@mathematik.uni-halle.de

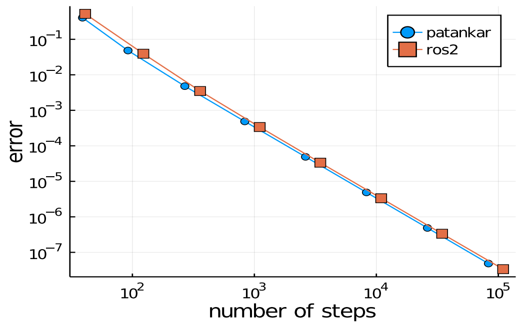

Fig. 1: Results for Lotka-Volterra (3).

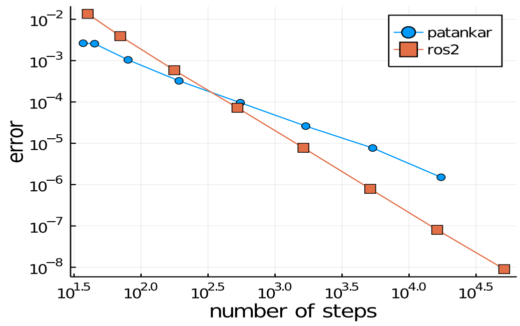

Fig. 2: Results for the Robertson problem (4).

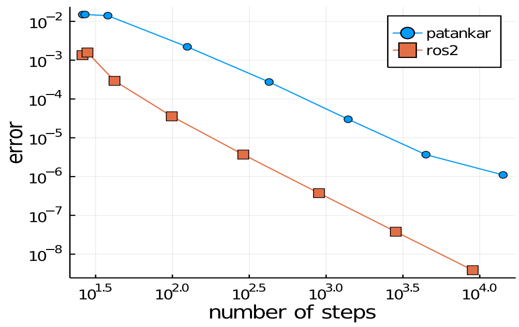

Fig. 3: Results for the pollution problem.

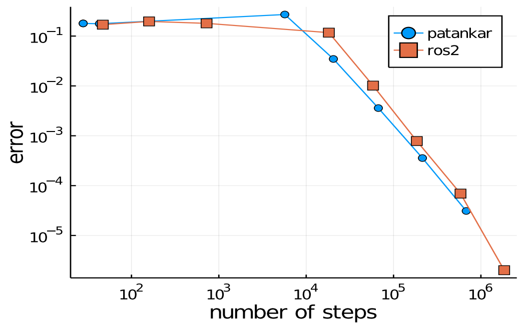

Fig. 4: Results for MAPK.

3. Pollu, a stiff system of 20 components, taken from cf. [5]. This is a production-destruction system which be written (in several different ways) in the form (1) with a graph Laplacian $M(y)$.

4. MAPK, taken from [1], the origin is [6], consisting of six equations. It has a graph Laplacian and it is mildly stiff.

The code has been implemented in Julia and can be downloaded from https://www2.mathematik.uni-halle.de/podhaisky/software/aprk/. Stepsize control is standard and based on the first order error estimate $y^{n+1} - y^{(2)}$. Figures 1-4 shows that the adaptive code works in all cases. Compared with the two-stage second order Rosenbrock method ROS2 [7], the Patankar Runge–Kutta shows a similar accuracy except for problem Pollu.

3 Conclusion

The proposed adaptive second order scheme works robustly. Note, the Patankar Runge-Kutta method outperforms ROS2 for low tolerances in the Robertson problem – in a regime, where positivity of the solution is crucial. A detailed analysis and the use of higher order Patankar Runge–Kutta methods is future work.

Acknowledgements Open access funding enabled and organized by Projekt DEAL.

References

[1] S. Blanes, A. Iserles, and S. Macnamara, Positivity-preserving methods for population models, 2021, arXiv:2102.08242.
[2] H. Burchard, E. Deleersnijder, and A. Meister, Appl. Numer. Math. **47**(1), 1–30 (2003).
[3] S. Kopecz and A. Meister, Appl. Numer. Math. **123**, 159–179 (2018).
[4] P. Öffner and D. Torlo, Appl. Numer. Math. **153**, 15–34 (2020).
[5] F. Mazzia and C. Magherini, Test set for initial value problem solvers, release 2.4, 2008, http://pitagora.dm.uniba.it/ testset.
[6] O. Hadač, F. Muzika, V. Nevoral, M. Přibyl, and I. Schreiber, PLOS ONE **12**(6), e0178457 (2017).
[7] J. G. Verwer, E. J. Spee, J. G. Blom, and W. Hundsdorfer, SIAM J. Sci. Comput. **20**(4), 1456–1480 (1999).

