Bachelor thesis proposal

The Leja method in Python

supervised by
Peter Kandolf, peter.kandolf@uibk.ac.at
Alexander Ostermann, alexander.ostermann@uibk.ac.at

Goal:
The goal of this bachelor thesis is to write a state-of-the-art implementation of the Leja method for computing the action of matrix exponential in Python based, amongst others, on the NumPy and SciPy packages. The developed code is going to be published on the homepage of the numerical analysis group.

Description:
The Leja method, described in Caliari et al. [2004], Bergamaschi et al. [2006], Caliari and Ostermann [2009], Caliari et al. [2014], is a well established method to efficiently approximate the action of the matrix exponential. We denote this action by

$$\exp(A)b, \quad A \in \mathbb{C}^{n\times n}, \quad b \in \mathbb{C}^n.$$

In general, methods that compute $\exp(A)$ and in a second step compute the action $\exp(A)b$ are not feasible for high dimensional matrices, see Higham [2008]. The Leja method is an efficient method to compute directly $\exp(A)b$ for large matrices. A dimension of $n \geq 10^4$ is common. The Leja method was undergoing some changes that focused on a backward error analysis inspired by Al-Mohy and Higham [2011]. Based on the new insights on the method a new implementation was developed. Nevertheless, the project is based on a preprint.

The programming language Python[1] is gaining attention in different sciences and a state-of-the-art implementation in this language can help to promote the numerical method. The aim of this project is to develop such an implementation. In order to do so one needs to study and understand the theoretical background (numerical analysis) of the Leja method as well as the current implementation. The developed implementation should be tested with some numerical experiments based on the discretization of time dependent partial differential equations and various prototypical test cases. In order to employ the newly developed backward error analysis one needs to compute certain numerical coefficients with high precision. The computation of these coefficients can also be part of the thesis.

Requirements:
A good understanding of the topics from the numerical analysis classes as well as an interest in programming is required for this topic.

[1] `https://www.python.org/`, 07.07.15

References

A. H. Al-Mohy and N. J. Higham, Computing the action of the matrix exponential, with an application to exponential integrators, *SIAM J. Sci. Comput.* 2011, vol. 33(2): pp. 488–511.

L. Bergamaschi, M. Caliari, A. Martínez and M. Vianello, Comparing Leja and Krylov approximations of large scale matrix exponentials, In: *Computational Science - ICCS 2006, Pt 4, Proceedings,* 2006, pp. 685–692.

M. Caliari and A. Ostermann, Implementation of exponential Rosenbrock-type integrators, *Appl. Numer. Math.* 2009, vol. 59(3-4): pp. 568–581.

M. Caliari, M. Vianello and L. Bergamaschi, Interpolating discrete advection-diffusion propagators at Leja sequences, *J. Comput. Appl. Math.* 2004, vol. 172(1): pp. 79–99.

M. Caliari, P. Kandolf, A. Ostermann and S. Rainer, Comparison of software for computing the action of the matrix exponential, *BIT Numerical Mathematics* 2014, vol. 54(1): pp. 113–128.

N. J. Higham, Functions of matrices. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA 2008, theory and computation.