▶ Recursive solution of Sylvester equations

Prof. D. Kressner
A. Cortinovis

Consider the matrix Sylvester equation

$$AX - XB = C, \tag{1}$$

with given coefficient matrices $A \in \mathbb{R}^{m \times m}, B \in \mathbb{R}^{n \times n}, C \in \mathbb{R}^{m \times n}$ and an unknown solution matrix $X \in \mathbb{R}^{m \times n}$. Solving this equation for large values of m, n has important applications in dynamical systems, control theory, financial mathematics, and image processing. The main goal of this project is to implement the recursive algorithm described in [2] for solving (1). For the theoretical parts, see, e.g., Section 15.1 in [1].

a) Using the Kronecker product, show that (1) is equivalent to a linear system $\mathcal{A}x = c$ with some matrix $\mathcal{A} \in \mathbb{R}^{mn \times mn}$.

b) Show that (1) admits a unique solution if and only if $\Lambda(A) \cap \Lambda(B) = \emptyset$, where $\Lambda(\cdot)$ denotes the set of eigenvalues of a matrix.

c) A special case of (1), the Lyapunov equation takes the form

$$AX + XA^T = C, \qquad C = C^T. \tag{2}$$

Show that X is symmetric, under the condition that the solution is unique. Give a counter-example, showing that there may be nonsymmetric solutions to (2).

[Optional] Show that if C is negative semi-definite and the eigenvalues of A have negative real part then X is positive semi-definite. Formulate a condition that guarantees positive definiteness of X.

d) Implement the recursive algorithm described in [2] for solving Sylvester algorithms, based on the *real* Schur decompositions of A, B. Measure the execution time wrt m, n of your implementation for sufficiently large, suitably chosen random matrices A, B, C. Determine suitable values of m_b, n_b for stopping the recursion and switching to MATLAB's `sylv` or `lyap`. Provide profiles that offer separate timings for the Schur decomposition, solution of triangular matrix equation, and back transformation. Compare the execution times obtained for your implementation with existing functionality in MATLAB for solving Sylvester equations, such as `sylv` (be careful!), `axxbc`, and `lyap`. You should be able to beat everything that is in MATLAB.

e) Investigate the scalability of your implementation by selecting the number of threads in MATLAB.

[Optional] Run your implementation and investigates the scalability of your implementation on the EPFL cluster using parallel MATLAB.

▶ References

[1] Nicholas J Higham. *Accuracy and stability of numerical algorithms*, volume 80. Siam, 2002.

[2] Isak Jonsson and Bo Kågström. Recursive blocked algorithms for solving triangular systems—part i: One-sided and coupled sylvester-type matrix equations. *ACM Transactions on Mathematical Software (TOMS)*, 28(4):392–415, 2002.