problem sheet 7

discussion: week of Monday, 22.11.2021

7.1. The sequence u_k, $k = 0, 1, \ldots$, given by[1]

$$u_1 := 2, \qquad u_{k+1} = 2^k \sqrt{2\left(1 - \sqrt{1 - (2^{-k}u_k)^2}\right)} \tag{1}$$

converges to the number $\pi = 3.1415\ldots$.

a) Compute (in `matlab`/`python`) the first 30 members of the sequence and the absolute error $|\pi - u_k|$. When is the error minimal?

b) Explain why you should expect that the error grows for $k \geq k_0$ for some k_0. *Extra Problem:* Assume that in exact arithmetic the error is $|u_k - \pi| \approx 2^{-2k}$. Use this to show that the minimal achievable error is reached for $k \approx 17$.

7.2. (Aitken Δ^2 extrapolation)

a) The Aitken Δ^2 method can be used to accelerate the convergence of a sequence $(x_n)_n$ that converges to $x_\infty = \lim_{n\to\infty} x_n$. To that end, assume that the sequence $(x_n)_n$ has the form

$$x_n = x_\infty + Cq^n \tag{2}$$

with (unknown) x_∞, C, $q \in (0, 1)$. Give a formula that "extrapolates" the sought limit x_∞ from 3 successive sequence members x_n, x_{n+1}, x_{n+2} by assuming that all 3 values satisfy (2). Proceeding in this way for every n produces a new sequence $(\widetilde{x}_n)_n$ that (sometimes) converges faster to x_∞ than the original sequence. Apply this method to the sequence $(u_k)_k$ of Problem 7.1 to get a new sequence of improved approximations to π. What is the best possible error?

b) Suppose you don't know the limit π of the sequence $(u_k)_k$. How can you estimate the errors of the approximations u_k? Can you formulate a sensible stopping criterion for the iteration (1)?

7.3. **(upload on TUWEL)** For symmetric, positive definite matrices $\mathbf{A}$, one typically makes a *Cholesky-factorization* instead of an *LU*-factorization. That is, one seeks a lower triangular matrix[2] $\mathbf{C}$ such that

$$\mathbf{C}^\top \mathbf{C} = \mathbf{A}$$

Formulate an algorithm that computes $\mathbf{C}$. Realize your algorithm in Matlab/Python.

Hint: Proceed as in Crout's method.

7.4. ("arrowhead matrix") Let $n = 10$, $\mathbf{e} = (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$, $\mathbf{b} = (1, 0, \ldots, 0)^\top \in \mathbb{R}^n$. Consider the matrix $\mathbf{A} = 10\mathbf{I} + \mathbf{b}\mathbf{e}^\top + \mathbf{e}\mathbf{b}^\top$ and the matrix $\widetilde{\mathbf{A}} := \mathbf{A}(n : -1 : 1, n : -1, 1)$ that is obtained from $\mathbf{A}$ by reversing the numbering of the rows and columns. Use the commands `spy` (`matplotlib.pyplot.spy`) and `lu` (`scipy.linalg.lu`) to visualize the sparsity patterns of $\mathbf{A}$, $\widetilde{\mathbf{A}}$ and the corresponding factors $\mathbf{L}$, $\mathbf{U}$ of the *LU*-factorization. What do you observe? Which variant of the numbering is to be preferred from a cost (i.e., number of floating point operations or storage requirement) point of view?

---

[1] The u_k correspond to the circumference of regular polygons with 2^k edges; this method of approximating π is due to Archimedes

[2] not normalized, i.e., $\mathbf{C}_{ii}$ need not be 1