10. Erstellen Sie eine Bildschirm-Präsentation mit LaTeX-Beamer, in der Sie eine Kurvendiskussion der Funktion
$$f(x) = \frac{e^{-x}}{x^2 - 2x + 2}$$
durchführen. (Es ist die Präsentation auch vorzuführen.)

11. Erstellen Sie eine Bildschirm-Präsentation mit LaTeX-Beamer, in der Sie folgende Aufgabe behandeln (Diskrete Math., SS 2009, Bsp. 11):

    Zeigen Sie: Sei G eine Gruppe, $a \in G$ und $|a| = n$. Dann gilt für alle $m \in \mathbb{Z}$
    $$|a^m| = \frac{n}{\operatorname{ggT}(n,m)}.$$

    Die Präsentation ist auch vorzuführen.

12. Man programmiere eine Maple-Funktion, die zwei Permutationen der Zahlen $\{1,\ldots,n\}$ multipliziert. Es soll also etwa
```
permmult([5,3,4,2,1],[3,2,4,1,5]) = [4,3,2,5,1]
```
gelten. Die Darstellung einer Permutation π in der Form $[a_1, a_2, \ldots]$ soll dabei bedeuten, dass $\pi(1) = a_1$, $\pi(2) = a_2$, etc.

13. Man programmiere eine Maple-Funktion, die die inverse Permutation zu einer Permutation bestimmt. Dabei seien die Permutationen wie in Beispiel 12 dargestellt.

14. Implementieren Sie die QR-Zerlegung einer $m \times n$-Matrix via Householder-Reflektoren in Matlab/Octave.
    (a) Der Algorithmus darf eine Laufzeit von $O(\max(m,n)^4)$ haben.
    (b) Der Algorithmus muss eine Laufzeit von $O(\max(m,n)^3)$ haben. Richtwert: Die QR-Zerlegung einer zufällige 1000×1000-Matrix kann mit einem selbstgeschriebenen Programm unter octave 3.0.1 auf einem Intel® Core™ 2 Duo E8400 mit 3 GHz in 35 Sekunden bestimmt werden.

15. Berechnen Sie in Maple oder Mathematica die LR-Zerlegung folgender Matrizen, so sie existiert.
    (a) $\begin{pmatrix} 2 & 2 & 2 & 4 \\ 4 & 3 & 3 & 1 \\ 0 & 2 & 3 & 5 \\ 0 & 0 & 3 & 2 \end{pmatrix}$
    (b) $\begin{pmatrix} 1 & 8 & 9 & 9 \\ 2 & 1 & 0 & 0 \\ 3 & 9 & 9 & 0 \\ 23 & 10 & 20 & 02 \end{pmatrix}$

    Dabei dürfen keine einschlägigen eingebauten oder in Zusatzpaketen enthaltenen Befehle verwendet werden, sondern es ist Maple bzw. Mathematica als Taschenrechner, der mit Matrizen umgehen kann, zu verwenden. Führen Sie auch die Probe durch.

16. Implementieren Sie die Jacobi-Iteration zur iterativen Lösung linearer Gleichungssysteme in Matlab/Octave und testen Sie sie anhand der Matrix
$$A = (a_{ij})_{\substack{1 \le i \le 100 \\ 1 \le j \le 100}}$$