# 1. Projekt: LR-Zerlegung und Gauß-Elimination

Abgabe: bis 18.11. (23:00) in Moodle; Vorführung der Programme in den Übungen am 19.11./20.11, bzw. zu angekündigten weiteren Terminen.

### Aufgabenstellung:

Implementieren Sie in MATLAB die LR-Zerlegung mit Spaltenpivotisierung sowie die Vorwärts- und Rückwärtssubstitution jeweils als eigene Funktion. Implementieren Sie – unter Benutzung dieser Funktionen – eine Funktion zur Lösung von linearen Gleichungssystemen. Testen Sie Ihre Implementierung an einfachen Beispielen.

### Hinweise zur Implementierung

(i) Überlegen Sie sich die Signaturen für jede der zu implementierenden vier Funktionen. Bei der Vorwärtssubsitution kann dabei anlog zur Rückwärtssubstitution (Programmieraufgabe in Übungsblatt 2) vorgegangen werden. Die Funktion zur LR-Zerlegung soll drei Rückgabewerte haben: die Dreiecksmatrizen `L` und `R` und einen Vektor `P`, der die verwendeten Zeilenvertauschungen kodiert. Bei der Berechnung ist es am einfachsten, den Algorithmus aus der Vorlesung zu benutzen, der die Einträge von `L` und `R` in einer Matrix (nämlich der übergebenen Matrix `A`) ablegt, und erst in einem zweiten Schritt die Matrizen `L` und `R` zu erzeugen.

(ii) Schreiben Sie für jede Funktion ein Skript, das die Richtigkeit der jeweiligen Funktion an einfachen Beispielen automatisch testet. Zum Beispiel können Sie für die Überprüfung der Lösung eines linearen Gleichungssystems eine Matrix $A$ und eine Lösung $x$ vorgeben und mit $b = Ax$ die rechte Seite berechnen. Zum Vergleich zweier Vektoren können Sie z.B. auch die Funktion `norm` verwenden.

(iii) Bei der Implementierung der Funktionen zur Lösung linearer Gleichungssysteme können Sie MATLAB-Funktionen `size` und `length` verwenden, um die Dimension der übergebenen Matrix `A` und des übergebenen Vektors `b` festzustellen. Wenn die Dimensionen nicht zusammenpassen, soll abgebrochen werden (z.B. mit `error(...)`).

(iv) Die `lu(A)`-Zerlegung von MATLAB, die Anweisung `R\b`, etc., dürfen natürlich nur zu Testzwecken benutzt werden.

### Zusatzaufgabe[1]:

Vergleichen Sie die Rechenzeit Ihres Algorithmus mit der Matlab Routine $A\backslash b$ (grafische Darstellung der Rechenzeit in abhängigkeit von der Anzahl $n$ der Unbekannten. Zufallsmatrizen können mit `rand` erzeugt werden und für die Zeitmessung können die Funktionen `tic` und `toc` benutzt werden. Denken Sie an die Möglichkeit einer logarithmischen Darstellung (`loglog`). Verdoppeln Sie am besten in jedem Schritt die Problemgröße ($n = 2, 4, 8, 16, ...$).

[1] Zusatzaufgaben müssen nicht bearbeitet werden, um die Note 1.0 zu erreichen. Eine erfolgreiche Bearbeitung gibt jedoch einen Bonus.

### Projektbericht

Das Ergebnis der Bearbeitung des Projektes soll neben dem lauffähigen Programm-Code ein kurzer Projektbericht sein, den Sie als PDF-Datei abgeben. Dieser soll enthalten:

- Titel des Projektes, Kurs, Name, Datum
- Vorstellung des Projektthemas (zwei Sätze)
- Beschreibung des Verfahrens, eventuell kurze Herleitung.
- Beschreibung wichtiger Aspekte der Implementierung.
- Eventuell (nicht in diesem Projekt): Diskussion der Ergebnisse.
- Eventuell kurze (!) kommentierte Code-Ausschnitte.

Insgesamt sollte der Bericht nicht länger als 2-3 Seiten sein !

### Benotungskriterien

Abzugeben sind ein Projektbericht und der zugehörige MATLAB-Code (4 Funktionen und 4 Test-Skripte), elektronisch in einer .zip-Datei mit dem Dateinamen `<nachname_vorname>.zip`, also zum Beispiel `hausser_frank.zip`, die sie in Moodle hochladen. Die Implementierung kann allein oder zuzweit erfolgen (bitte im Programm-Code kenntlich machen). Der Projektbericht muss einzeln angefertigt werden und das Projekt von jeder(m) abgegeben werden. Auch MATLAB-Programme müssen natürlich eigenständig implementiert werden!

- Basisnote: 1.0
- Leichter Funktionsfehler: +0.3 bzw. +0.4
- Mittlerer Funktionsfehler: +1.0
- Schwerer Funktionsfehler: +2.0
- Unübersichtlicher Quellcode / Schlechte Dokumentation: +0.3 bzw. +0.4
- Unvollständiger Bericht, kleine Lücken/leichte Fehler +0.3 bzw. +0.4
- Unvollständiger Bericht, große Lücken/schwere Fehler +1.0
- Besonders gute Ideen/Ausarbeitung: bis zu −1.0
- Erfolgreiche Bearbeitung der Zusatzaufgabe: −0.3 bzw. −0.4

### Weitere Hinweise

- Kopieren und anschließendes Anpassen von Programm-Code zählt genauso wie das Abschreiben bei einer Klausur als Betrugsversuch. Sollte ich Selbiges (bei der Vorführung oder bei der Korrektur der Berichte bzw. des hochgeladenen Programm-Codes) feststellen, wird das Projekt als nicht bestanden bewertet. Das Erstellen von Plagiaten ohne Angabe von Quellen ist insbesondere im wissenschaftlichen/akademischen Bereich nicht nur unsportlich sondern eben Betrug.

- Mit dem Hochladen Ihres Projektes bestätigen Sie, dass Sie dieses eigenständig angefertigt haben. Alle verwendeten Quellen müssen genannt werden. Wenn Sie im wesentlichen wörtlich zitieren, dann müssen Sie die entsprechenden Stellen als Zitat kennzeichnen. Das gilt auch für Programm-Code.
- Bitte betrachten Sie das Vorführen der Programme in der Übung wie eine Prüfungsituation, d.h. bereiten Sie sich darauf vor.
- Fangen Sie frühzeitig (jetzt :-)) mit der Implementierung an und klären Sie eventuelle Schwierigkeiten in der Übung.
- Nicht vergessen: Es darf auch Spaß machen, ein Lösungsverfahren "zum Fliegen" zu bringen. Lösen Sie mit Ihrem Code auch einmal größere LGSe.