

# Programmieren für Physiker

Interfakultatives Institut für Anwendungen der Informatik  
Institut für Theoretische Teilchenphysik

Prof. Dr. M. Steinhauser, Dr. A. Mildenerger  
<http://comp.physik.kit.edu>

SS 2018 – Blatt 05  
Bearbeitungszeitraum: bis 23. Mai 2018

---

## Aufgabe 13: Gleichungssysteme – Gauß-Algorithmus

Pflichtaufgabe

Es ist ein C++-Programm zu entwickeln, das das lineare Gleichungssystem  $A \cdot x = b$  für gegebenes  $A$  und  $b$  löst.

Lesen Sie hierzu zunächst die ganzzahlige Größe  $n$  des Systems und anschließend die reellen Matrixelemente von  $A$  und die rechte Seite  $b$  aus einer Datei ein. Die Matrixelemente von  $A$  stehen zeilenweise in der Datei. Im Verzeichnis `/home/ck18/daten/` oder auf der WWW-Seite finden Sie die beiden Testdateien `a13-lgs1.dat` und `a13-lgs2.dat`.

Geben Sie nach dem Einlesen bitte zunächst das Gleichungssystem auf dem Bildschirm aus.

Mit Hilfe des Gauß-Algorithmus bringen Sie nun das Gleichungssystem in Dreiecksform und lösen dieses schrittweise. Geben Sie nun den Lösungsvektor aus.

Das Verfahren wird numerisch stabiler, wenn Sie eine sogenannte Spaltenpivotisierung durchführen: Suchen Sie jeweils in der aktuell zu bearbeitenden (Rest-)Spalte das betragsgrößte Element und verwenden Sie dann diese Zeile, um in anderen Zeilen Null-Einträge zu generieren. In Ihrem Programm soll per Eingabe zu entscheiden sein, ob die Spaltenpivotisierung eingesetzt wird oder nicht.

Hinweis: Bei der ersten Datei sollte der Lösungsvektor Sie an eine relativ bekannte irrationale Zahl erinnern. Testen Sie bitte Ihr Programm, bevor Sie es zum Testat vorlegen.

---

## Aufgabe 14: vollkommene Zahlen

freiwillig

Eine natürliche Zahl heißt *vollkommen*, wenn die Summe ihrer echten Teiler genau die Zahl selbst ergibt. (Echte Teiler einer Zahl  $n$  sind alle Teiler der Zahl ohne  $n$  selbst).

Bestimmen Sie mit einem Programm durch Berechnung der Teilersumme alle vollkommenen Zahlen kleiner als 10000.

Bemerkung (nicht für das Programm zu verwenden): Bereits Euklid wusste, dass alle Zahlen der Form  $P_p = 2^{p-1}(2^p - 1)$  vollkommen sind, wenn sowohl  $p$  als auch  $M_p = 2^p - 1$  prim sind. Euler bewies, dass alle geraden vollkommenen Zahlen von dieser Form sind. Ob ungerade vollkommene Zahlen existieren, ist bis heute unklar.

---

---

## ASCII Tabelle

| hex  |     | 0 | 1 | 2 | 3 | 4  | 5 | 6 | 7   | 8  | 9  | A  | B  | C  | D   | E  | F   |
|------|-----|---|---|---|---|----|---|---|-----|----|----|----|----|----|-----|----|-----|
|      | dez | 0 | 1 | 2 | 3 | 4  | 5 | 6 | 7   | 8  | 9  | 10 | 11 | 12 | 13  | 14 | 15  |
| 0x00 | 0   |   |   |   |   |    |   |   | BEL | BS | HT | LF | VT | FF | CR  |    |     |
| 0x10 | 16  |   |   |   |   |    |   |   |     |    |    |    |    |    | ESC |    |     |
| 0x20 | 32  |   | ! | " | # | \$ | % | & | '   | (  | )  | *  | +  | ,  | -   | .  | /   |
| 0x30 | 48  | 0 | 1 | 2 | 3 | 4  | 5 | 6 | 7   | 8  | 9  | :  | ;  | <  | =   | >  | ?   |
| 0x40 | 64  | @ | A | B | C | D  | E | F | G   | H  | I  | J  | K  | L  | M   | N  | O   |
| 0x50 | 80  | P | Q | R | S | T  | U | V | W   | X  | Y  | Z  | [  | \  | ]   | ^  | _   |
| 0x60 | 96  | ` | a | b | c | d  | e | f | g   | h  | i  | j  | k  | l  | m   | n  | o   |
| 0x70 | 112 | p | q | r | s | t  | u | v | w   | x  | y  | z  | {  |    | }   | ~  | DEL |

Zeichen mit Code 0, ..., 31 dienten ursprünglich dazu, Ein-/Ausgabegeräte zu steuern (z.B. Zeilenvorschub). Einige dieser „nicht-druckbaren“ Zeichen spielen heute noch eine Rolle, einige haben eine Zeichenkonstante in C bzw. C++ :

| Zeichenkonstante  | Kürzel           | Erklärung  |
|-------------------|------------------|--|
| <code>\a</code>   | BEL              | Bell: Signalton ausgeben                             |
| <code>\b</code>   | BS               | Backspace: Schritt nach links gehen                  |
| <code>\f</code>   | FF               | Form Feed: neue Seite                                |
| <code>\n</code>   | LF               | Line Feed: neue Zeile                                |
| <code>\r</code>   | CR               | Carriage Return: zurück an Zeilenanfang              |
| <code>\t</code>   | HT               | Horizontal Tab: Tabulator                            |
| <code>\v</code>   | VT               | Vertical Tab: vertikaler Tabulator                   |
|                   | ESC              | Escape: Einleitung von Steuerbefehlen                |
|                   | DEL              | Delete: Zeichen löschen                              |
| <code>\\</code>   | <code>\</code>   | Backslash  |
| <code>\'</code>   | <code>'</code>   | einfaches Anführungszeichen                          |
| <code>\"</code>   | <code>"</code>   | doppeltes Anführungszeichen                          |
| <code>\ooo</code> | <code>ooo</code> | Zeichen mit oktalem ASCII Code <code>ooo</code>      |
| <code>\xhh</code> | <code>hh</code>  | Zeichen mit hexadezimalen ASCII Code <code>hh</code> |

Beispiel: `cout << "\x41\n" ;` druckt ein A und geht in die nächste Zeile.

(Auf Linux-Systemen können Sie diese Informationen auch per `man ascii` erhalten.)

---