

# Lineare Gleichungssysteme, Diagonalmatrix, Probe

Schreiben Sie ein MATLAB-Skript

`diagmatrix`,

das ein gegebenes Gleichungssystem löst und das Ergebnis einer Probe unterzieht.

Die Größe der Matrix und des Inhomogenitätsvektors wird durch die Variable `n` festgelegt, die Sie mit der Routine `input` von der Konsole einlesen. Als Defaultwert setzen Sie den Wert 5 (`isempty`).

$M$  ist eine  $2n \times 2n$  Matrix, deren Hauptdiagonale aus Einsern besteht und deren Nebendiagonalen ausschließlich 0.5 enthalten. Erzeugen Sie diese Matrix mit den Befehlen `eye` und `diag`.

$$M = \begin{bmatrix} 1 & 0.5 & 0 & 0 & \dots \\ 0.5 & 1 & 0.5 & 0 & \dots \\ 0 & 0.5 & 1 & 0.5 & \dots \\ 0 & 0 & 0.5 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Der Vektor  $v$  soll folgende Gestalt haben:

$$v = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 3 \\ \vdots \\ 0 \\ n \end{bmatrix}$$

Erzeugen Sie  $v$  mit `zeros` und der `Doppelpunkt`-Notation.

1. Lösen Sie nun die Gleichung  $My = v$ .
2. Machen Sie die Probe  $check = My - v$ .
  - (a) Überprüfen Sie, ob `check` tatsächlich ausschließlich Nuller enthält. Geben Sie den logischen Vektor dieses Vergleichs in der Form `check: 'Werte des Vergleichs'` formatiert aus. Enthält er nur Einser? Wandeln Sie den Vektor gegebenenfalls in einen Zeilenvektor um.

Bei der Ausgabe von `check` sieht man, dass dieser Test fehlschlägt. Dies liegt daran, dass durch die endliche Genauigkeit der Zahlen stets Rundungsfehler auftreten. Aus diesem Grund ist es nicht sinnvoll auf **Gleichheit** zu prüfen. Es macht lediglich Sinn auf **beinahe Gleichheit** zu prüfen. Dazu kann der *relative* oder der *absolute* Fehler verwendet werden. Hier soll eine absolute Fehlerschranke verwendet werden.

```
error_limit = 1.0e-8;
```

Damit ist nun die Durchführung einer sinnvollen Probe der Form

```
if all ( abs(check) < error_limit )
    disp('Check successful')
else
    disp('Check not successful')
end
```

möglich.

Führen Sie am Ende Ihres Skripts die Probe auf diese Art durch.