

Gaußverteilung

Schreiben Sie eine MATLAB-Funktion `gauss1d`, die mit folgendem Aufruf

```
[g] = gauss1d(x,x0,sigma)
```

die Verteilung [3](#) als Funktion des Vektors x berechnet.

Einführung:

In der Wahrscheinlichkeitsrechnung und auch in der Physik hat die Normalverteilung bzw. Gaußverteilung eine große Bedeutung. Sie ist definiert durch

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x - x_0)^2}{2\sigma^2} \right\}, \quad (1)$$

wobei x_0 und σ die Parameter der Verteilung sind. In x_0 liegt sowohl das Maximum als auch das Symmetriezentrum, und σ ist der Abstand von diesem Zentrum zu den Wendepunkten. Wie für jede Wahrscheinlichkeitsverteilung gilt

$$\int_{-\infty}^{\infty} g(x) dx = 1. \quad (2)$$

In der Physik verwendet man häufig auch eine Summation über mehrere Gaußfunktionen mit jeweils unterschiedlichen Parametern. Dies hat den Sinn, dass man gleichzeitig mehrere Maxima darstellen kann. Allgemein kann man die Funktion dann definieren als

$$g(x) = \sum_{k=1}^n \frac{1}{n\sqrt{2\pi}\sigma_k} \exp \left\{ -\frac{(x - x_{0k})^2}{2\sigma_k^2} \right\}, \quad (3)$$

wobei x_{0k} und σ_k die gleiche Bedeutung wie vorher haben. Die Summation erfolgt über alle n Werte der Parameter x_{0k} und σ_k .

Defaultwerte:

```
x0 = [-1.0, 1.0]
sigma = [ 0.5, 1.0]
```

Zum Setzen der Defaultwerte sind die Funktionen `nargin` und `isempty` zu verwenden. Weitere Informationen finden Sie im Skriptum in [Kapitel 9](#), oder [hier](#).

Fehlermeldungen:

Verwenden sie die Funktion `error` um eine Fehlermitteilung zu generieren, wenn gewisse Inputparameter nicht den gewünschten Forderungen entsprechen. Die unten angegebenen Texte müssen in der jeweiligen Mitteilung enthalten sein.

PROBLEM	FEHLERTEXT MUSS ENTHALTEN
kein x-Vektor bzw. ein leeres Feld als Inputparameter	x values
x0 und sigma haben unterschiedliche Längen	same length

Programmierung:

- Initialisieren Sie g als Feld mit Nullen in der Größe von x .
- Bilden Sie die Summe über alle k -Werte der Parameter ($k = 1 \dots n$) mit Hilfe einer einzigen `for`-Schleife. Vergessen Sie dabei nicht, dass x ein Array sein kann.

Hinweis:

Eine `for`-Schleife hilft bei der Summation über die einzelnen Beiträge zur Summe **3**. Dabei ist es praktisch, wenn man noch vor der Schleife ein Feld erzeugt, das gleich groß ist wie die Inputvariable x , das aber ausschließlich Nullen enthält. Der sinnvollste Befehl dafür ist:

```
g = zeros(size(x))
```

Dann kann man in der Schleife

```
g = g + ...
```

schreiben und bei jedem Durchlauf werden die Werte für einen Peak addiert. Nach dem Ende der Schleife muss man durch die Anzahl der Peaks n , also durch die Länge des Vektors x_0 (oder `sigma`), dividieren. Damit ist dann die Normierung auch bei mehreren Peaks sichergestellt.