

Prof. Dr. Marcin Grzegorzek, Juniorprofessor – Fakultät IV, Elektrotechnik und Informatik  
Hölderlinstr. 3, 57068 Siegen

## Einführung in die Informatik II

### SoSe 2013

### Übung 4

In dieser Übung sollen Sie die Messdaten einer virtuellen Strahlungsquelle untersuchen, um das Zentrum der Strahlung durch Visualisierung der Daten zu finden. Die Daten werden von der Funktion `erzeugeWerte` generiert, und es wird eine 3D-Matrix zurückgegeben. Der Funktion wird ein Parameter übergeben, welcher angibt wie intensiv die Strahlungsquelle ist. Je intensiver die Strahlung, desto einfacher wird die Quelle zu finden sein - oder man schaut in den Quelltext.

#### Aufgabe 1

Rufen Sie die Funktion `erzeugeWerte(m)` mit  $m=1e4$  im Command-Window auf. Damit darf die Strahlungsquelle 10.000 mal strahlen. Etwa so:

```
>> Feld=erzeugeWerte(1e4);  
Elapsed time is 0.324216 seconds.
```

Untersuchen sie mit dem `size`-Befehl die Dimensionen des Feldes und schließen Sie damit auf die Form des untersuchten Gebietes.

Für langsame Rechner wird die Variable `Feld` in der beiliegenden Datei `Daten.mat` bereitgestellt. Die Daten in der Datei entsprechen einem Wert  $m=1e6$ .

#### Aufgabe 2

Nachdem Sie nun herausgefunden haben, dass es sich bei dem Gebiet um einen Würfel der Kantenlänge 20 handelt, können Sie Anfangen die Daten zu visualisieren.

Verwenden sie den `slice`-Befehl, um Schnitte durch das Gebiet zu legen. Schreiben ein *Matlab*-Skript (keine Funktion), um die Befehle in der Kommandozeile nicht immer wiederholen zu müssen und um einfacher auf die Workspace-Variablen zugreifen zu können (dann brauchen Sie die Daten nicht immer wieder zu erzeugen).

**Hilfestellung:** `slice(x,y,z,Feld,sx,sy,sz)`

Die Vektoren  $x,y$  und  $z$  geben die Einteilung der  $x,y$  und  $z$  Achse an. Überlegen Sie anhand der Form des Gebiets, ob diese Variablen überhaupt unterschiedlich sein müssen.

Die Vektoren  $sx,sy$  und  $sz$  geben an, wo die Schnitte durch das Gebiet gelegt werden sollen.

### Aufgabe 3

Erweitern Sie das Skript, indem Sie die Farben interpolieren und die Kantenfarbe löschen.

```
hsurfaces = slice(...);  
set(hsurfaces, 'FaceColor', 'interp', 'EdgeColor', 'none')
```

### Aufgabe 4

Erweitern Sie das Skript, indem Sie Konturlinien um gleiche Intensität zeichnen.

Setzen die Linienfarbe mit der rot, grün und blau Intensität (Wert zwischen 0..1) und die Linienstärke.

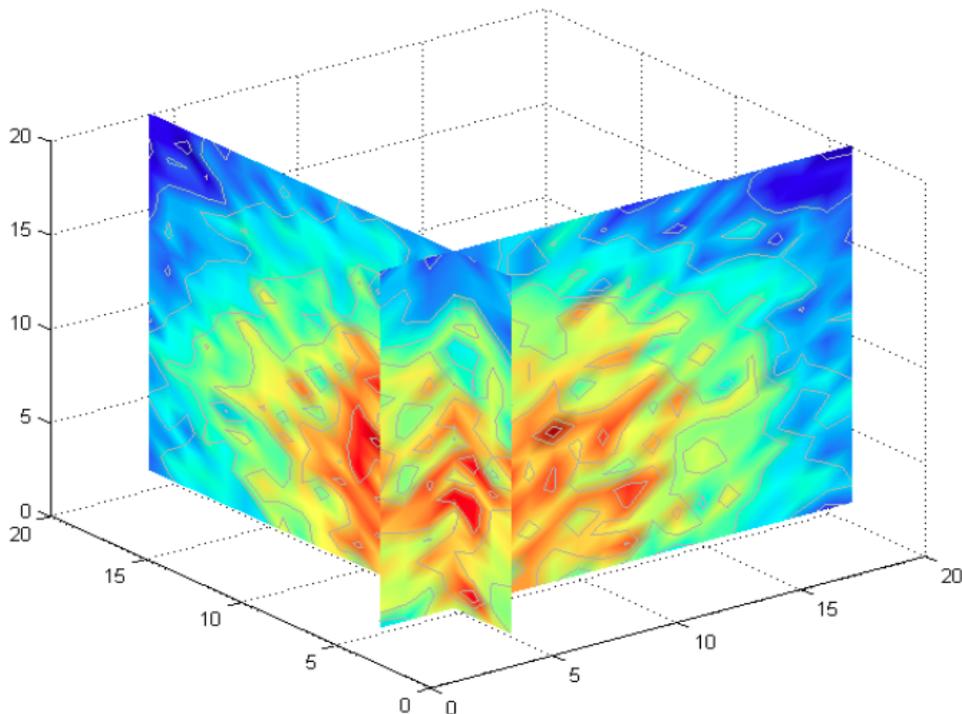
```
hcont = contourslice(...);  
set(hcont, 'EdgeColor' [<rot>, <grün>, <blau>], 'LineWidth', <Linienstärke>)
```

### Aufgabe 5

Programmieren Sie eine Schleife um diese Anzeige, damit  $s_x$ ,  $s_y$  und  $s_z$  so geändert werden, dass die Schnitte durch das Gebiet wandern.

Verwenden Sie innerhalb der Schleife den `drawnow()`-Befehl, damit die Grafik auch tatsächlich aktualisiert wird.

Beispiel für die Visualisierung:



## Aufgabe 6

Schreiben Sie eine Optimierungsfunktion namens `curvefit`, welcher die Messdaten übergeben werden. Die Funktion soll die Parameter `a` und `b` der Fit-Funktion wieder zurückgeben.

Innerhalb der Funktion `curvefit` erstellen Sie eine Funktion `fqsfun`, welche die Fehlerquadratsumme berechnen soll und die Parameter der Fit-Funktion als Vektor übergeben bekommt.

Das Grundgerüst sieht also so aus (mit Programmieranleitung):

```
function [a,b] = curvefit(xdata, ydata)
```

1. Zufällige Startwerte für `fminsearch`.
2. `fminsearch`-Befehl zum minimieren der Fehlerquadratsumme aufrufen
3. Aus den optimierten Parametern `a` und `b` bestimmen.

```
function [fqs] = fqsfun(params)
```

1. `a` und `b` aus `params` bestimmen.
2. Funktionswerte der Fit-Funktion an den Stellen `xdata` mit den Parametern `a` und `b` ausrechnen.
3. Jeweils die Abweichung zwischen den gerade berechneten Funktionswerten und `ydata` bestimmen.
4. Diese Werte quadrieren und summieren. Das ist dann der Rückgabewert.

```
end
```

```
end
```

## Aufgabe 7

Rufen Sie die Funktion `curvefit` auf. Zurückgegeben werden die Parameter `a` und `b`, welche in etwa mit den Werten aus Aufgabe 1 Spiegelpunkt 2 übereinstimmen sollten.

Plotten Sie die gefittete Funktion in das Diagramm mit den Messwerten.

Das Ergebnis sollte in etwa so aussehen:

