# Projekt

Das elektromagnetische Feld in einem Mikrowellenherd erfüllt die Maxwell-Gleichungen:

$$\begin{aligned} \nabla \times \mathbf{E} &= -\mu \mathbf{H}_t \\ \nabla \times \mathbf{H} &= \varepsilon \mathbf{E}_t + \sigma \mathbf{E}, \end{aligned}$$

wobei der Vektor **E** das elektrische Feld und der Vektor **H** das magnetische Feld ist. Die Konstanten $\varepsilon, \sigma$ und $\mu$ entsprechen der Permittivität, der elektrischen Leitfähigkeit und der Permeabilität.

In der Luft gilt $\varepsilon_l = 8.85 \cdot 10^{-12}, \sigma_l = 0$ und $\mu_l = 4\pi \cdot 10^{-7}$ und im Poulet $\varepsilon_p = 4.43 \cdot 10^{-11}, \sigma_p = 3 \cdot 10^{-11}$ und $\mu_p = 4\pi \cdot 10^{-7}$.

(1) Wir nehmen an, dass die Lösung zeit-harmonisch ist:

$$\mathbf{E} = \mathbf{E}(x, y, z)\mathrm{e}^{\mathrm{i}\omega t}, \quad \mathbf{H} = \mathbf{H}(x, y, z)\mathrm{e}^{\mathrm{i}\omega t},$$

mit $\omega = 2\pi f := 2\pi 2.45 \cdot 10^9$. In diesem Fall, zeigen Sie dass, die Maxwell-Gleichungen lauten

$$\begin{aligned} \nabla \times \mathbf{E} &= -\mathrm{i}\omega\mu \mathbf{H} \\ \nabla \times \mathbf{H} &= \mathrm{i}\omega\tilde{\varepsilon} \mathbf{E}, \end{aligned}$$

mit $\tilde{\varepsilon}$ zu bestimmen.

(2) Wir nehmen an, dass $\partial_z \mathbf{E} = \partial_z \mathbf{H} = 0$. Finden Sie eine neue Gleichung für die dritte Komponente des elektrischen Feldes $E_3$. Diese Gleichung ist die Helmholtz-Gleichung in zwei Raumdimensionen:

$$\Delta E_3 + \omega^2 \mu \tilde{\varepsilon} E_3 = 0.$$

Wir wollen jetzt diese Helmholtz-Gleichung mit der Finite-Elemente-Methode lösen !

(3) Zuerst, muss man ein Gitter erzeugen:

```
function [N,T,P]=Gitter(G,w)
%% Gittererzeuger
% G -> Rechengebiet (G=0,G=1,G=2, oder G=3)
% w -> Frequenz von der Helmholtz-Gleichung
% N -> Liste von Knoten (mit x und y Koordinaten):
% T -> Liste von Dreiecke die richtet nach der Liste N
%      die 3 ersten Eingabe sind die Knoten
%      die 3 naechsten enthalten 0 oder 1:
%      1 falls die Dreieckeskante auf dem Rand des
%      Gebiet liegt, und 0 sonst
% P -> Enthaelt die Material-Konstante mu*epsilon fuer jede Dreieck
```

Diese Funktion erzeugt eine Triangulierung (N,T) von einem Einheitesquadrat (falls G=0), einem Dreieck (falls G=1), einer leeren Mikrowelle (falls G=2), und einem Poulet in der Mikrowelle (falls G=3, *finden Sie die Daten auf dem Internet*).

*Hinweis: Für das Einheitsquadrat hat man*
*N=[0 0;1 0;..;..]. T=[1 2 3 1 0 ..;2 4 .. .. .. ..]. P=[1 1].*

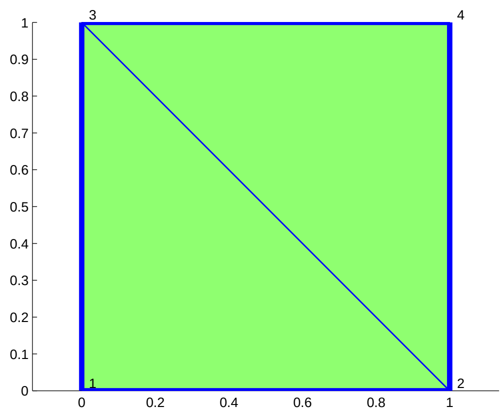
(4) Zur Visualisierung der Triangulierung, benutzt man die folgende Funktion:

```
function PlotGitter(N,T,P)
%% Visualisierung der Triangulierung
% N -> Liste von Knoten
% T -> Liste von Dreiecke
% P -> Enthaelt die Konstante mu*epsilon fuer jede Dreieck
%      mit Farben codiert
%      (in patch, benutzen Sie abs(P(i)) als dritte Komponente)
```

Alle Dreieckeskanten, die auf dem Rand des Rechengebiets liegen, sollen fett gedruckt sein. Für eine kleine Anzahl von Knoten (sagen wir $< 100$) schreiben Sie auch die dazugehörige Nummer auf den Knoten.

*Hinweis: Die Matlab-Funktionen* LINE, PATCH, TEXT, NUM2STR *sind hilfreich.*

Für das Quadrat soll man das folgende Bild bekommen:

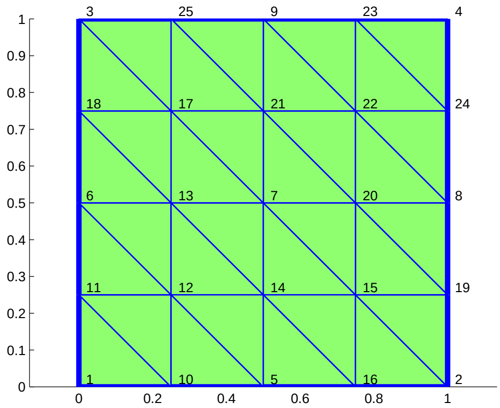



(5) Unser nächstes Ziel ist die Implementierung einer einfachen Gitterverfeinerung. Wir dividieren alle Dreiecken in vier kleinere. Dazu muss man, für jedes Dreieck, drei neuen Knoten einführen (z.B. n4=0.5*(n1+n2)) und damit vier neue Dreiecke definieren.

*Bemerkung*: Für jedes neue Dreieck müssen die Informationen über die Ränder und die Materialkonstanten gespeichert werden.

```
function [Nr,Tr,Pr]=GitVerfeinerung(N,T,P)
%% einfache Gitterverfeinerung
% Nr, Tr, Pr -> Neuen Knoten, Dreiecken und Material-Konstanten
```

Nach zwei Verfeinerungen bekommen wir:




(6) (*Die Finite-Elemente-Methode*) Wir berechnen zuerst die Elementsteifigkeitsmatrix und die Elementmassenmatrix für ein Dreieck mit den Knoten $(x_1, y_1), (x_2, y_2)$ und $(x_3, y_3)$ (die Knoten sind gegen den Uhrzeigersinn nummeriert):

```
function Se=Elementsteifigkeitsmatrix(t)
%% Berechnet die Elementsteifigkeitsmatrix fuer das
%% Dreieck t
% t=[x1 y1;x2 y2;x3 y3] -> Dreieck
```

und

```
function Me=Elementmassenmatrix(t)
%% Berechnet die Elementmassenmatrix fuer das
%% Dreieck t
% t=[x1 y1;x2 y2;x3 y3] -> Dreieck
% hier muss man bei Hand int_Dreieck phi_i phi_j rechnen
```

(7) Löser für die Helmholtz-Gleichung mit Dirichlet-Randbedingungen $g(x, y)$ in einem triangulierten Bereich $N, T, P$:

```
function [u,K,M]=FEHelmholtz2D(g,N,T,w,P)
%% FE Loeser fuer die Helmholtz-Gleichung
% g -> Dirichlet-RB
% w -> Frequenzen fuer die Helmholtz-Gleichung
% Assemblierung + Aufbau des Lastvektor + Loesung
% des Gleichungssytems
```

(8) Visualisierung der Lösung:

```
function PlotLoesung(u,N,T)
%% Plot die Loesung u fuer die gegebene Triangulierung
%% N,T
```

*Hinweis: Man kann* MESH, PATCH *und* COLORBAR *benutzen.*

(9) Testen Sie Ihren Code auf dem Quadrat mit Dirichlet-Randbedingungen $g(x, y) = x + y$ und der Frequenz $\omega = 0$. Wie lautet die exakte Lösung ? Vergleichen Sie die numerische und die exakte Lösung.

(10) Für das Poulet in der Mikrowelle sind die Dirichlet-Randbedingungen gegeben durch

```
g=inline('100*(x==0.5 & 0.1<=y & 0.2>=y)','x','y');
```

Spielen Sie mit dem Code und beschreiben Sie die Lösung.

Bevor Ende Januar 2008, erwarten wir eine kurze Präsentation von Ihrem Projekt.
Herrn Prof. M.J. Gander danken wir für den Vorschlag des Themas.