

Projektbericht zum Projekt der Vorlesung "Numerik der partiellen Differentialgleichungen"

Manuel Baumgartner

13. Januar 2010

1 Zusammenfassung

Die Aufgabe in diesem Projekt war die Implementierung einer vereinfachten Form der Maxwell - Gleichungen, angewandt auf einen Fisch in einer Mikrowelle. Die Vereinfachungen führen letztlich darauf, dass die Helmholtzgleichung zu lösen ist. Die Gleichung soll mit Hilfe der Finite-Elemente-Methode (FEM) mit linearen Dreieckselementen gelöst werden.

2 Mathematische Modellierung

2.1 Herleitung des Modells

Die Maxwell - Gleichungen beschreiben das Verhalten einer Elektromagnetischen Welle und haben laut Aufgabenblatt die Form

$$\begin{aligned}\nabla \times \vec{E} &= -\mu \vec{H}_t \\ \nabla \times \vec{H} &= \epsilon \vec{E}_t + \sigma \vec{E}\end{aligned}$$

auf einem Gebiet $\Omega \subset \mathbb{R}^3$ und für Funktionen $\vec{E}, \vec{H} : \Omega \times \mathbb{R} \rightarrow \mathbb{C}^3$. Die physikalischen (Material-)Konstanten ϵ, σ, μ sind die Permittivität, elektrische Leitfähigkeit und Permeabilität und haben in diesem Modell die Werte

$$\epsilon_L = 8.85 \cdot 10^{-12}, \quad \sigma_L = 0, \quad \mu_L = 4\pi \cdot 10^{-7}$$

für Luft und

$$\epsilon_F = 6.44 \cdot 10^{-10}, \quad \sigma_F = 3 \cdot 10^{-11}, \quad \mu_F = 4\pi \cdot 10^{-7}$$

für den Fisch.

Die erste Annahme ist, dass die Lösung der Maxwell - Gleichungen zeitharmonisch ist, das heisst die Form

$$\vec{E}(x, y, z, t) = E(x, y, z) \exp(i\omega t) \text{ und } \vec{H}(x, y, z, t) = H(x, y, z) \exp(i\omega t)$$

mit $\omega = 2\pi \cdot 2.25 \cdot 10^9$ hat. Bezeichne mit $\vec{E} = (\vec{E}_1, \vec{E}_2, \vec{E}_3)^T$ und $\vec{H} = (\vec{H}_1, \vec{H}_2, \vec{H}_3)^T$ die Komponenten von \vec{E} und \vec{H} und mit $E = (E_1, E_2, E_3)^T$ und $H = (H_1, H_2, H_3)^T$ die Komponenten von H . Dann hat man für die linke Seite der ersten Maxwell - Gleichung:

$$\nabla \times \vec{E} = \begin{pmatrix} \partial_y \vec{E}_3 - \partial_z \vec{E}_2 \\ \partial_z \vec{E}_1 - \partial_x \vec{E}_3 \\ \partial_x \vec{E}_2 - \partial_y \vec{E}_1 \end{pmatrix} = \exp(i\omega t) \begin{pmatrix} \partial_y E_3 - \partial_z E_2 \\ \partial_z E_1 - \partial_x E_3 \\ \partial_x E_2 - \partial_y E_1 \end{pmatrix} = \exp(i\omega t) (\nabla \times E)$$

da für $i = x, y, z$ und $j = 1, 2, 3$ gilt

$$\partial_i \vec{E}_j(x, y, z, t) = \partial_i (E_j(x, y, z) \exp(i\omega t)) = \exp(i\omega t) \partial_i (E_j(x, y, z))$$

Die rechte Seite der ersten Maxwell - Gleichung ergibt:

$$-\mu \vec{H}(x, y, z, t) = -\mu \partial_t (H(x, y, z) \exp(i\omega t)) = -\mu H(x, y, z) \cdot i\omega \cdot \exp(i\omega t)$$

und damit kriegt man für die erste Gleichung:

$$\exp(i\omega t)(\nabla \times E) = -\mu H \cdot i\omega \cdot \exp(i\omega t)$$

Da $\exp(i\omega t) \neq 0$ für alle $t \in \mathbb{R}$ folgt nach Kürzen, dass die erste Maxwell - Gleichung nun die Form

$$\nabla \times E = -i\omega \mu H$$

hat.

Nun analog für die zweite Maxwell - Gleichung:

Die linke Seite gibt wie zuvor $\nabla \times \vec{H} = \exp(i\omega t)(\nabla \times H)$ und die rechte Seite liefert

$$\epsilon \vec{E}_t + \sigma \vec{E} = \epsilon \partial_t(E(x, y, z) \exp(i\omega t)) + \sigma E(x, y, z) \exp(i\omega t) = (\epsilon i\omega + \sigma) E(x, y, z) \exp(i\omega t)$$

Damit lautet die zweite Gleichung nun

$$\exp(i\omega t)(\nabla \times H) = (\epsilon i\omega + \sigma) E(x, y, z) \exp(i\omega t)$$

Kürze $\exp(i\omega t)$ und setze $\tilde{\epsilon} := \epsilon + \frac{\sigma}{i\omega}$:

$$\nabla \times H = \underbrace{(\epsilon i\omega + \sigma)}_{=i\omega \tilde{\epsilon}} E$$

Man kriegt die neuen Maxwell - Gleichungen

$$\begin{aligned}\nabla \times E &= -i\omega \mu H \\ \nabla \times H &= i\omega \tilde{\epsilon} E\end{aligned}$$

auf $\Omega \subset \mathbb{R}^3$.

Eine weitere getroffene Annahme ist, dass $\partial_z E = \partial_z H = 0$. Das heisst, dass sich das elektrische und magnetische Feld in der dritten Koordinatenrichtung nicht ändert. Das bedeutet $\partial_z E_i = \partial_z H_i = 0$ für $i = 1, 2, 3$ und damit folgt für die Maxwell - Gleichungen

$$\begin{pmatrix} \partial_y E_3 \\ -\partial_x E_3 \\ \partial_x E_2 - \partial_y E_1 \end{pmatrix} = \begin{pmatrix} \partial_y E_3 - \partial_z E_2 \\ \partial_z E_1 - \partial_x E_3 \\ \partial_x E_2 - \partial_y E_1 \end{pmatrix} = \nabla \times E = -i\omega \mu \begin{pmatrix} H_1 \\ H_2 \\ H_3 \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} \partial_y H_3 \\ -\partial_x H_3 \\ \partial_x H_2 - \partial_y H_1 \end{pmatrix} = \begin{pmatrix} \partial_y H_3 - \partial_z H_2 \\ \partial_z H_1 - \partial_x H_3 \\ \partial_x H_2 - \partial_y H_1 \end{pmatrix} = \nabla \times H = i\omega \tilde{\epsilon} \begin{pmatrix} E_1 \\ E_2 \\ E_3 \end{pmatrix} \quad (2)$$

Aus den ersten beiden Komponenten von (1) folgt

$$\partial_y E_3 = -i\omega \mu H_1 \quad -\partial_x E_3 = -i\omega \mu H_2 \quad (3)$$

und aus der dritten Komponente von (2):

$$\partial_x H_2 - \partial_y H_1 = i\omega \tilde{\epsilon} E_3 \quad (4)$$

Setze nun (3) in (4) ein:

$$\frac{1}{i\omega \mu} \partial_x^2 E_3 + \frac{1}{i\omega \mu} \partial_y^2 E_3 = i\omega \tilde{\epsilon} E_3 \quad (5)$$

Multipliziere (5) mit $i\omega\mu$ und erhalte

$$\partial_x^2 E_3 + \partial_y^2 E_3 = -\omega^2 \mu \tilde{\epsilon} E_3$$

oder

$$\Delta E_3 + \omega^2 \mu \tilde{\epsilon} E_3 = 0 \quad (6)$$

auf $\Omega \subset \mathbb{R}^3$.

Nun hängt die Funktion $E_3 : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{C}$ nicht von der z-Koordinate von \mathbb{R}^3 ab, da die Annahme $\partial_z E = 0$ impliziert $\partial_z E_3 = 0$ und das kann man als eine gewöhnliche Differentialgleichung auffassen. Nach dem Existenz- und Eindeutigkeitssatz hat diese eine eindeutige konstante Lösung falls x, y als konstant gesehen werden.

Es folgt, dass E_3 eigentlich eine Funktion $E_3 : \Omega \cap \mathbb{R}^2 \rightarrow \mathbb{C}$ ist. Für eine einfache Notation bezeichne $\Omega \cap \mathbb{R}^2$ auch als Ω und damit ist nun die Aufgabe, die Gleichung (6) auf $\Omega \subset \mathbb{R}^2$ zu lösen mit der Dirichlet-Randbedingung $E_3(x, y) = g(x, y)$ auf $\partial\Omega$.

2.2 Testprobleme

2.2.1 Die Poisson-Gleichung

Setzt man in (6) den Parameter ω auf Null, dann bekommt man die Poisson-Gleichung. Das heisst man bekommt das Problem:

$$\begin{cases} \Delta u = 0 & \text{auf } \Omega \subset \mathbb{R}^2 \\ u = g & \text{auf } \partial\Omega \end{cases} \quad (7)$$

Hier soll (7) mit $\Omega = [0, 1]^2$ dem Einheitsquadrat in \mathbb{R}^2 und

$$\begin{aligned} g : \partial[0, 1]^2 &\rightarrow \mathbb{R} \\ (x, y) &\mapsto x + y \end{aligned}$$

betrachtet werden.

Betrachte nun

$$\begin{aligned} u : [0, 1]^2 &\rightarrow \mathbb{R} \\ (x, y) &\mapsto x + y \end{aligned}$$

dann ist sicher $u|_{\partial[0, 1]^2} = g$ und ausserdem löst u die Poisson-Gleichung (7), da

$$\Delta u = \frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = \frac{\partial^2}{\partial x^2}(x + y) + \frac{\partial^2}{\partial y^2}(x + y) = 0$$

Dieses Problem kann man also benutzen um den Code zu testen.

2.2.2 Der Fisch in der Mikrowelle

Für das Modell mit dem Fisch in der Mikrowelle muss man das Problem

$$\begin{cases} \Delta u + \omega^2 \mu \tilde{\epsilon} u = 0 & \text{auf } \Omega \subset \mathbb{R}^2 \\ u = g & \text{auf } \partial\Omega \end{cases} \quad (8)$$

lösen, das heisst die Helmholtzgleichung in zwei Raumdimensionen.

Für das konkrete Modell wähle $\Omega = [0, 0.5] \times [0, 0.3]$ als Mikrowelle, die Randbedingung

$$g : \partial\Omega \rightarrow \mathbb{R}$$

$$(x, y) \mapsto \begin{cases} 100 & (x, y) \in \{0.5\} \times [0.1, 0.2] \\ 0 & \text{sonst} \end{cases}$$

und die Materialkonstanten

$$\epsilon_L = 8.85 \cdot 10^{-12}, \quad \sigma_L = 0, \quad \mu_L = 4\pi \cdot 10^{-7}$$

für Luft und

$$\epsilon_F = 6.44 \cdot 10^{-10}, \quad \sigma_F = 3 \cdot 10^{-11}, \quad \mu_F = 4\pi \cdot 10^{-7}$$

für den Fisch.

3 Einige Implementationsaspekte

3.1 Triangulierung

Um die Triangulierung zu beschreiben wurden drei Listen (=Matrizen) definiert: N, T, P .

N ist eine Matrix mit zwei Spalten und jeder Eckpunkt der Dreiecke ist als Zeile in N mit seinen x und y Koordinaten gespeichert. Die Punkte sind nummeriert durch die Zeilennummer in N .

T ist eine Matrix mit sechs Spalten und speichert die Dreiecke. Jede Zeile speichert ein Dreieck und die ersten drei Einträge enthalten die Nummern der drei Eckpunkte. Die letzten drei Einträge in jeder Zeile bestehen aus 0 oder 1 und geben an, ob eine Dreiecksseite auf dem Rand des Gebietes liegt (=1) oder nicht (=0). Der vierte Eintrag speichert, ob die Kante von Eckpunkt 1 zu Eckpunkt 2 auf dem Rand liegt, der fünfte Eintrag ob die Kante von Eckpunkt 2 zu Eckpunkt 3 auf dem Rand liegt und der dritte Eintrag ob die Kante von Eckpunkt 3 zu Eckpunkt 1 auf dem Rand liegt. Jedes Dreieck bekommt auch eine Nummer und diese Nummer ist die Zeilennummer in der Liste T .

P ist ein Vektor der die Materialkonstante $\mu\tilde{\epsilon}$ für jedes Dreieck speichert. Die Konstante für das Dreieck mit Nummer n ist der n -te Eintrag von P .

3.2 Gitterverfeinerung

Um das Gitter zu verfeinern wird für jedes Dreieck die Mittelpunkte der Kanten berechnet und diese dann verbunden. Das liefert pro vorhandenem Dreieck vier neue.

Zunächst werden die neuen Listen Nr , Tr und Pr mit dem Inhalt der alten initialisiert:

$Nr = N$; $Tr = T$; $Pr = P$;

Die Gitterverfeinerung besteht aus einer grossen for-Schleife über die vorhandenen Dreiecke der Triangulierung, das heisst `for i=1:size(T,1)`. Dann wird mit

`Anz = length(Nr(:,1))`

die vorhandene Anzahl Dreiecke bestimmt und `New_Coord = zeros(3,2)` liefert einen Vektor für die neuen Punkte, das heisst für die Koordinaten der Mittelpunkte der Dreiecksseiten. Die Koordinaten von diesen werden mit

```

New_Coord(1,:) = 0.5.*[N(T(i,1),1) + N(T(i,2),1) ...
N(T(i,1),2) + N(T(i,2),2)];
New_Coord(2,:) = 0.5.*[N(T(i,2),1) + N(T(i,3),1) ...
N(T(i,2),2) + N(T(i,3),2)];
New_Coord(3,:) = 0.5.*[N(T(i,3),1) + N(T(i,1),1) ...
N(T(i,3),2) + N(T(i,1),2)];

```

aus den alten berechnet.

Nun werden zwei neue Vektoren definiert: `num = zeros(1,3)`; und `r = ones(1,3)`; . Hier soll `num` die Nummern der neuen Punkte speichern und `r` wird ein Hilfvektor um zu entscheiden, ob die berechneten Punkte wirklich neu sind oder nicht. Mit

```

for k=1:size(Nr,1)
if New_Coord(1,:)==Nr(k,:)
    r(1) = 0;
    num(1) = k; % Nummer des Punktes in Nr
elseif New_Coord(2,:)==Nr(k,:)
    r(2) = 0;
    num(2) = k;
elseif New_Coord(3,:)==Nr(k,:)
    r(3) = 0;
    num(3) = k;
end
end

```

werden die bereits vorhandenen Punkte in `Nr` untersucht, ob einer der neuen Punkte schon vorhanden ist. Ist das der Fall, wird der Wert vom Vektor `r` an entsprechender Stelle auf 0 gesetzt und in `num` an entsprechender Stelle die bereits vorhandene Nummer der Punktes gespeichert. Um nun zu entscheiden welche Punkte neu sind, muss man bestimmen welche Einträge in `r` ungleich 0 sind und die entsprechenden Indizes ausgeben. Das leistet der Matlab Befehl „find“: `ind = find(r)`; `ind` enthält nun die Indizes der neuen Punkte im Vektor `New_Coord` und diese werden nun via `New_Coord = New_Coord(ind,:)`; aus dem Vektor extrahiert und mit `Nr = [Nr; New_Coord]`; in `Nr` gespeichert.

Für das Nachtragen der Nummern der neuen Punkte in `num` sorgt die Schleife

```

for k=1:length(ind)
    num(ind(k)) = Anz + k;
end

```

Nun können die neuen Dreiecke definiert und in `Tr` abgespeichert werden:

```

Tr = [Tr;    T(i,1) num(1) num(3) T(i,4) 0 T(i,6); ...
      num(1) num(2) num(3) 0 0 0; ...
      num(1) T(i,2) num(2) T(i,4) T(i,5) 0; ...
      num(2) T(i,3) num(3) T(i,5) T(i,6) 0];

```

Die Materialkonstanten sind für die vier neuen Dreiecke gleich wie für das alte, also

```
Pr = [Pr P(i).*ones(1,4)];
```

Nach dem Beenden der grossen for-Schleife über alle Dreiecke kann man die Informationen über die alten Dreiecke eliminieren:

```

Tr = Tr(length(T(:,1))+1:end,:);
Pr = Pr(length(P)+1:end);

```

Die Wirkung der Gitterverfeinerung soll an einem Beispiel gezeigt werden. In Abbildung 1 ist das unverfeinerte Gitter für das Einheitsquadrat $[0,1]^2 \subset \mathbb{R}^2$ zu sehen, sowie das gleiche Gitter wie es immer weiter verfeinert wird.

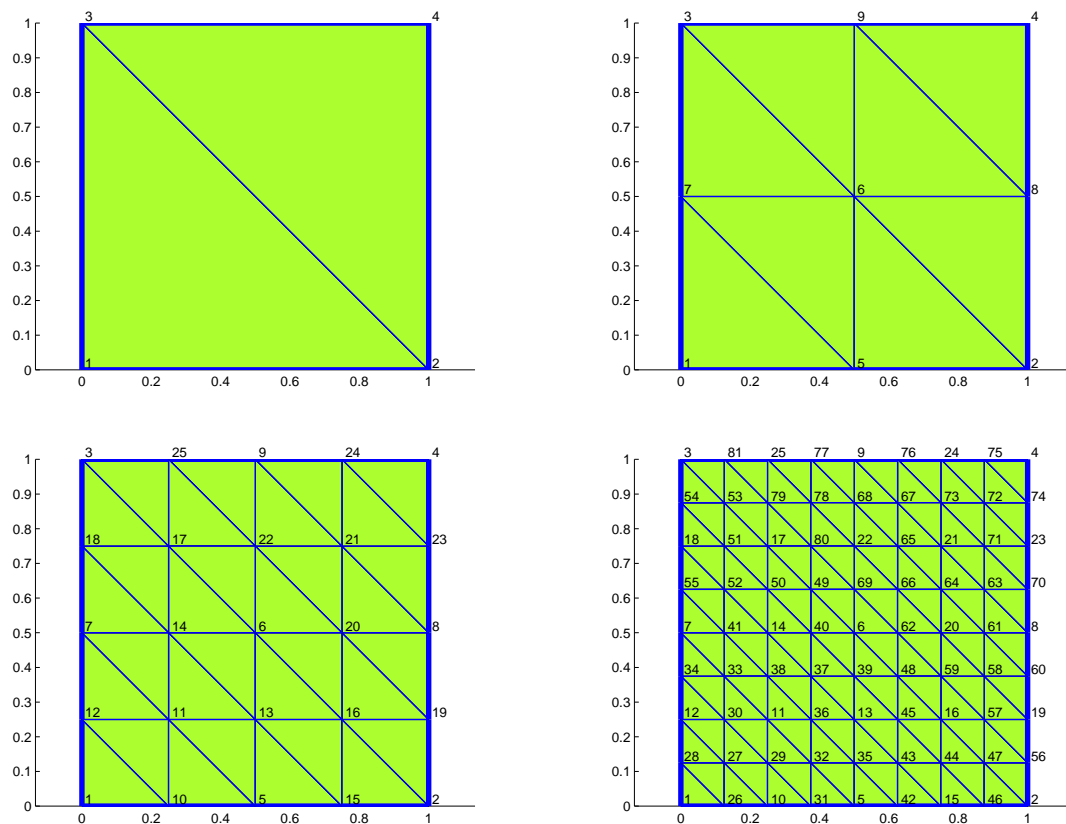


Abbildung 1: Links oben das unverfeinerte Gitter und dann Verfeinerungen.

3.3 Elementmatrizen

Für die FEM betrachtet man statt der ursprünglichen Gleichung (8) die schwache Formulierung dieser Gleichung. Diese soll zunächst hergeleitet werden. Der Galerkin-Ansatz führt dann auf ein sehr grosses Gleichungssystem, aus dem dann die numerische Lösung gewonnen wird. Diese grosse Matrix wird aus zwei kleinen Elementmatrizen A_e und M_e pro Element assembliert. Damit diese beiden Matrizen nicht für jedes Element neu berechnet werden müssen, nutzt man eine Transformation auf ein Referenzdreieck und berechnet alles wesentliche dort.

3.3.1 Schwache Formulierung

Um die schwache Formulierung zu finden multipliziert man die Gleichung (8) mit einer Testfunktion $v \in C_0^1(\Omega)$ und integriert:

$$\int_{\Omega} \Delta u \cdot v + \omega^2 \mu \tilde{\epsilon} \int_{\Omega} u \cdot v = 0$$

Nutze die Green-Formel:

$$-\int_{\Omega} (\nabla u) \cdot (\nabla v) + \int_{\partial\Omega} v \cdot \frac{\partial u}{\partial n} + \omega^2 \mu \tilde{\epsilon} \int_{\Omega} u \cdot v = 0$$

Da $v|_{\partial\Omega} = 0$ verschwindet das Randintegral. Multipliziere noch mit -1 und erhalte

$$\int_{\Omega} (\nabla u) \cdot (\nabla v) - \omega^2 \mu \tilde{\epsilon} \int_{\Omega} u \cdot v = 0$$

Man bekommt eine Bilinearform

$$\begin{aligned} \text{Bil} : H^1(\Omega) \times H^1(\Omega) &\rightarrow \mathbb{R} \\ (u, v) &\mapsto a(u, v) - \omega^2 \mu \tilde{\epsilon} \cdot b(u, v) \end{aligned}$$

wobei

$$\begin{aligned} a : H^1(\Omega) \times H^1(\Omega) &\rightarrow \mathbb{R} \\ (u, v) &\mapsto \int_{\Omega} (\nabla u) \cdot (\nabla v) \end{aligned}$$

und

$$\begin{aligned} b : H^1(\Omega) \times H^1(\Omega) &\rightarrow \mathbb{R} \\ (u, v) &\mapsto \int_{\Omega} u \cdot v \end{aligned}$$

Sei $u_0 \in H^1(\Omega)$ so, dass $u_0|_{\partial\Omega} = g$ im Sinne des Spursatzes. Die schwache Formulierung von (8) lautet damit:

Finde $u \in H^1(\Omega)$ mit $u - u_0 \in H_0^1(\Omega)$ so, dass $\text{Bil}(u, v) = 0$ für alle $v \in H_0^1(\Omega)$.

3.3.2 Berechnung der Elementsteifigkeitsmatrix A_e

Bezeichnen ϕ_1, ϕ_2, ϕ_3 die Basisfunktionen vom Element K , dann hat die Elementsteifigkeitsmatrix A_e die Einträge

$$A_e = (a(\phi_i, \phi_j))_{i,j=1,2,3}$$

Nun betrachtet man das Referenzdreieckselement \hat{K} mit Eckpunkten $(0,0), (1,0), (0,1)$ sowie die Transformationsabbildung

$$\begin{aligned} F_K : \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ (\xi_1, \xi_2) &\mapsto P_1 + (P_2 - P_1 | P_3 - P_1) \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} \end{aligned}$$

wobei P_1, P_2, P_3 die Eckpunkte des Dreiecks K sind und $\widehat{\phi}_1, \widehat{\phi}_2, \widehat{\phi}_3$ die Basisfunktionen von \hat{K} . Diese Funktionen sind gegeben durch

$$\begin{aligned} \widehat{\phi}_1(\xi_1, \xi_2) &= 1 - \xi_1 - \xi_2 \\ \widehat{\phi}_2(\xi_1, \xi_2) &= \xi_1 \\ \widehat{\phi}_3(\xi_1, \xi_2) &= \xi_2 \end{aligned}$$

Damit berechnet man (die genauen Details sind eine Herleitung in der Vorlesung)

$$a(\phi_i, \phi_j) = \frac{1}{2} |\det(J_K)| \cdot (\nabla \widehat{\phi}_i \cdot J_K^{-1}) \cdot (\nabla \widehat{\phi}_j \cdot J_K^{-1})^T$$

mit $J_K := (DF_K)_{(\xi_1, \xi_2)}$ dem Differential von F_K an der Stelle (ξ_1, ξ_2) . Das bedeutet man braucht lediglich die Koordinaten der Knoten des Dreieckselements um die Einträge von A_e zu berechnen, anstatt für jedes Element separat eine aufwändige Integration durchzuführen.

Man bekommt für jedes Element **e1** die Elementsteifigkeitsmatrix **Ae**:

```
Je = [(N(T(e1,2),:) - N(T(e1,1),:))' ...
      (N(T(e1,3),:) - N(T(e1,1),:))'] ; % =J_K
% Determinante von Je
determinante = Je(1,1)*Je(2,2)-Je(1,2)*Je(2,1);

D = (1/determinante).*[-ones(1,2); eye(2)]...
*[Je(2,2) -Je(1,2); -Je(2,1) Je(1,1)];
Ae = 0.5*determinante.*D*D';
```

3.3.3 Berechnung der Elementmassenmatrix M_e

Mit den gleichen Bezeichnungen wie in 3.3.2 hat die Elementmassenmatrix M_e die Einträge

$$M_e = (b(\phi_i, \phi_j))_{i,j=1,2,3}$$

Nutze wieder die gleiche Transformationsabbildung F_K vom Referenzelement \hat{K} auf das Element K wie in 3.3.2. Man berechnet damit die Einträge als

$$b(\phi_i, \phi_j) = |\det(J_K)| \int_{\hat{K}} \hat{\phi}_i \cdot \hat{\phi}_j$$

Nach der Auswertung dieser Integrale kann man die Elementmassenmatrix wie folgt implementieren:

```
Je = [(N(T(e1,2),:) - N(T(e1,1),:))' ...
      (N(T(e1,3),:) - N(T(e1,1),:))'] ;
determinante = Je(1,1)*Je(2,2)-Je(1,2)*Je(2,1);

Me = (abs(determinante)/24).*[2 1 1; 1 2 1; 1 1 2];
```

4 Numerische Ergebnisse

4.1 Poisson-Gleichung

Nun löse die Poisson-Gleichung (7) mit den Modellannahmen wie in 2.2.1 beschrieben. Dort wurde auch bemerkt, dass die gewählten Randbedingungen schon die exakte Lösung liefern. Abbildung 2 zeigt die exakte und die numerische Lösung und man sieht eine sehr gute Übereinstimmung. Das grobe Gitter vom Einheitsquadrat aus Abbildung 1 wurde dazu viermal verfeinert.

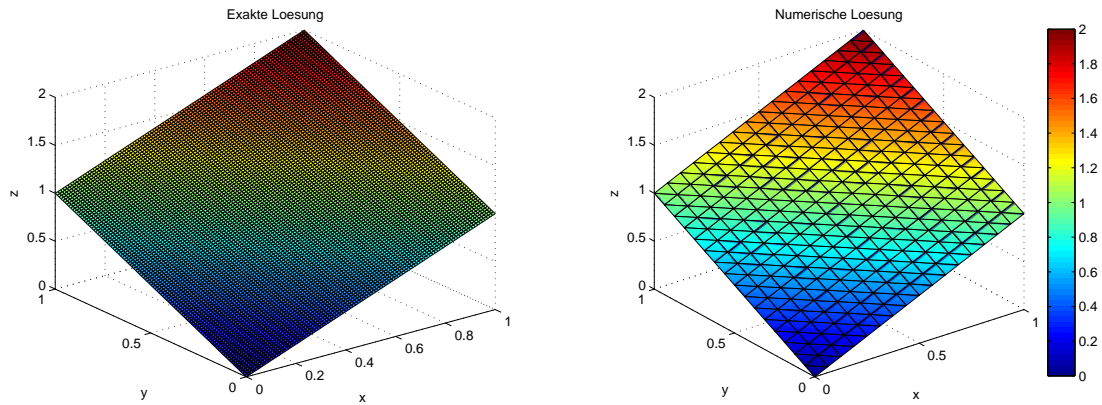


Abbildung 2: Links die exakte Lösung der Poisson-Gleichung und rechts die numerische Lösung.

4.2 Fisch in der Mikrowelle

Wie im Abschnitt 2.2.2 beschrieben soll die Gleichung (8) für einen Fisch in der Mikrowelle gelöst werden. Das unverfeinerte Gitter mit dem Fisch und die zweimalige Verfeinerung ist in Abbildung 3 zu sehen.

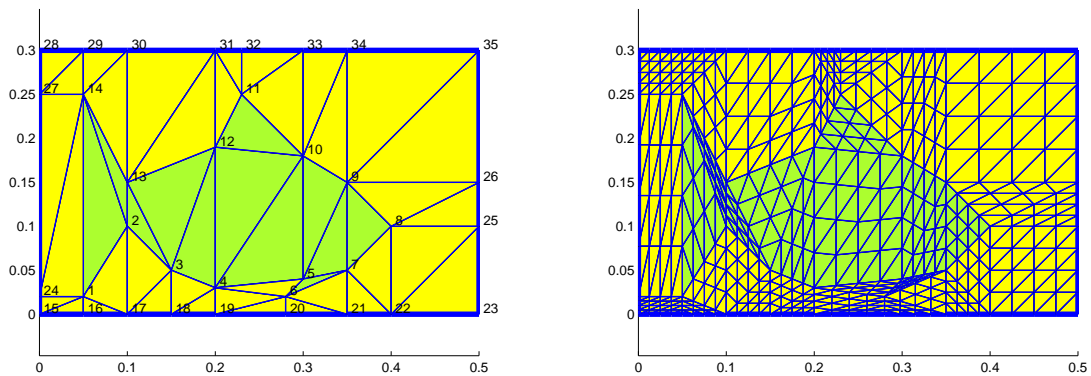
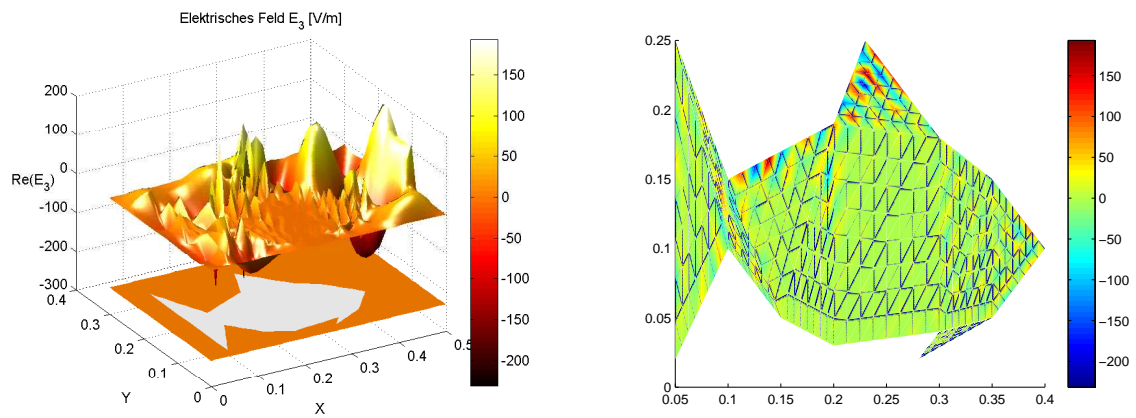
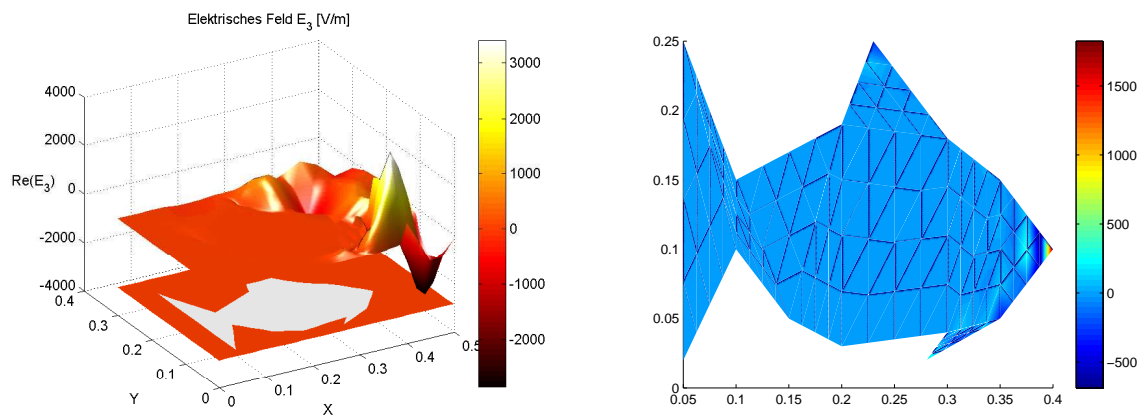


Abbildung 3: Links das unverfeinerte Gitter, rechts nach zweimaliger Verfeinerung.

Zuerst wird die Modellgleichung (8) auf dem zweimal verfeinerten Gitter aus Abbildung 3 gelöst. Abbildung 4 zeigt die numerische Lösung einmal als Plot auf dem ganzen Gebiet und einmal als Plot der Lösung ausschliesslich im Fisch. Die Lösung sieht noch sehr grob aus. Verfeinere nun das Gitter nochmals und löse die Gleichung erneut. Abbildung 5 zeigt wieder die Lösung einmal auf dem ganzen Gebiet und einmal nur im Fisch. Dreht man den Plot der die Lösung auf dem gesamten Rechengebiet zeigt ein wenig (Abbildung 6), kann man die Oszillationen im Fisch schon gut erkennen.

Ein weiteres Verfeinern des Gitter und erneutes Lösen der Gleichung liefert eine numerische Lösung, bei der die Schwingungen in der Nähe von $x = 0$, gut aufgelöst erscheint. Abbildung 7 zeigt



nochmals die Lösung auf dem ganzen Gebiet und auch ausschliesslich im Fisch. Die Schwingungen sind bei einer leichten Drehung des Plots noch besser erkennbar, was Abbildung 8 demonstriert.

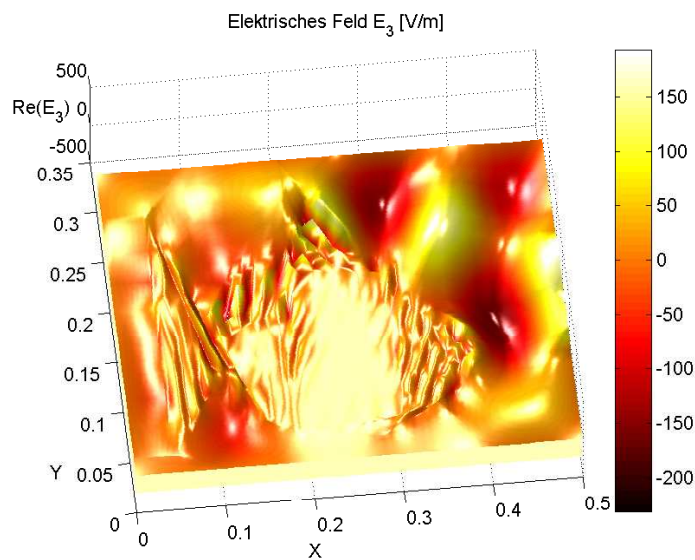


Abbildung 6: Die Lösung von Abbildung 5 links, leicht gedreht.

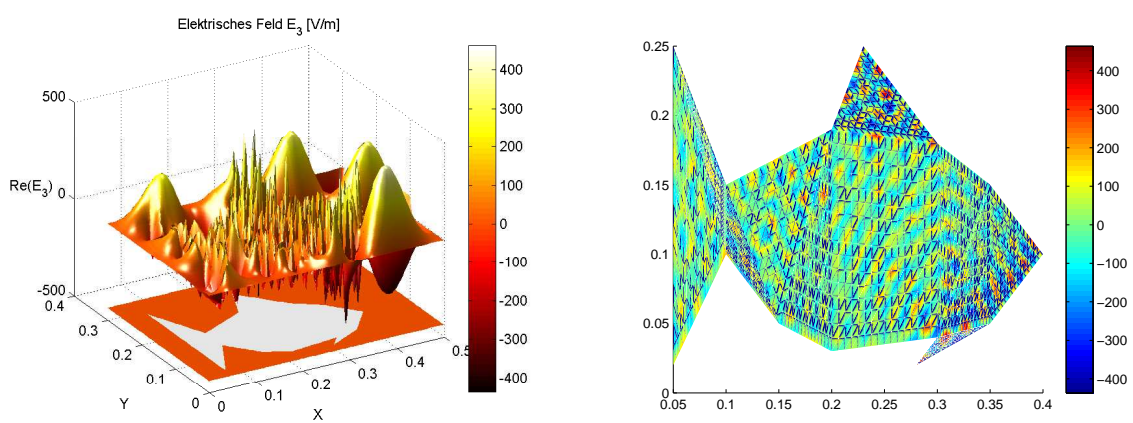


Abbildung 7: Links die Lösung auf dem ganzen Gebiet, rechts nur auf dem Fisch.

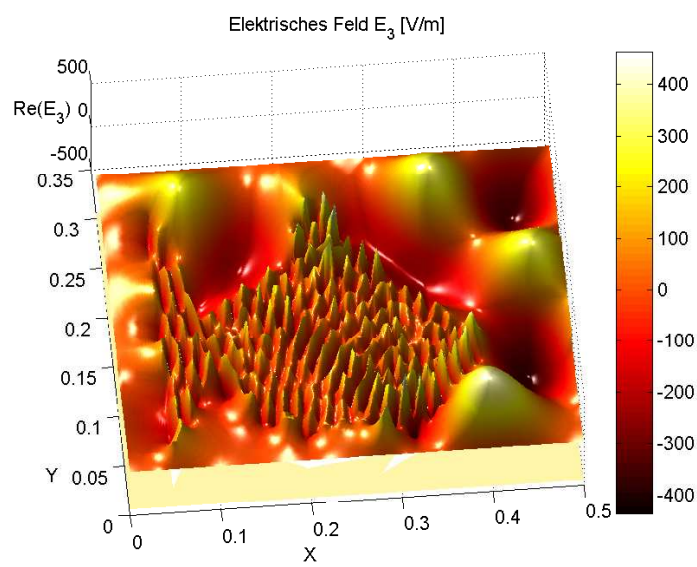


Abbildung 8: Die Lösung von Abbildung 7 links, leicht gedreht.