<text><text>

Johannes-Heidenhain-Gymnasium Traunreut Kollegiatenjahrgang 2008/2010

FACHARBEIT

aus dem Fach

Physik

Thema: Berechnung des Potentialverlaufs bei beliebiger Ladungsverteilung

Verfasser: Martin Gerer Leistungskurs: Physik Kursleiter: StR Reuter, Matthis Abgabetermin: 29. Januar 2010

.....

(Unterschrift des Kursleiters)

Inhaltsverzeichnis

1	Ein		4	
2	Elel	ktrosta	atik	5
	2.1	Elektr	ische Ladungen	5
	2.2	Elektr	ische Felder	6
		2.2.1	Elektrische Feldstärke ${\bf E}$	6
		2.2.2	Elektrischer Fluss Φ_{el}	7
		2.2.3	Das Gaußsche Gesetz	7
	2.3	Potent	tial	8
		2.3.1	Die Maxwell-Gleichungen für elektrostatische Felder	8
		2.3.2	Potentialdifferenzen	9
		2.3.3	Berechnung des Feldstärkevektors ${\bf E}$ mit Hilfe des Potentials $\ .$.	10
	2.4	Poisso	n-Gleichung	10
3	Nui	meriscl	ne Lösung der Poisson-Gleichung	11
	3.1	Lösun	g der Poisson-Gleichung mit Hilfe der Finite-Differenzen-Methode	11
		3.1.1	Differentialquotient und Differenzenquotient	11
		3.1.2	Anwendung der Methode auf die Poisson-Gleichung	12
		3.1.3	Anwendung der Methode zur Berechnung der E-Feld-Vektoren.	13
	3.2	Comp	uter basierte, numerische Berechnung	13
		3.2.1	Vorzug der iterativen Vorgehensweise	13
		3.2.2	Erzeugung der Gitterpunkte mit Hilfe von Arrays	14
		3.2.3	Die Parameter z, $\rho(x, y)$, n, $U_S(x, y)$, s	15
		3.2.4	Die <i>while</i> -Schleife zur Berechnung des Potentialverlaufs	16
		3.2.5	Berechnung der E-Feld-Vektoren	17
		3.2.6	Randbedingungen	17
	3.3	Lösun	gsbeispiele	18
		3.3.1	Feld zweier ungleichnamig gepolter Punktladungen	18
		3.3.2	Feld eines Plattenkondensators	18
		3.3.3	Feld zweier gleichnamig gepolter Ringe	19
		3.3.4	Feld zweier ineinander liegender Ringe	19
		3.3.5	Feld zwischen einer Spitze und einer Platte	19
	3.4	Vergle	ich der Messwerte mit theoretischen Werten	19
	3.5	Fehler	erklärung	20
		3.5.1	Mittelwertbildung mit nur vier Werten	20
		3.5.2	Einfluss der Randbedingungen auf das Ergebnis	20
		3.5.3	Rechendauer (Skalierungsfaktor s)	21

ABBILDUNGSVERZEICHNIS

5	Anhang							
	5.1 Ausführliche Herleitung d. elektr. Flusses in Abh. d. Ladungsverteilung							
	5.2	Bedeu	tung von Gradient, Divergenz und Rotation	23				
	5.3	Grafis	che Darstellung des Arrays <i>neuewerte</i>	24				
	5.4	Ausfül	arliche Anwendung der FDMethode auf die Poisson-Gleichung	25				
	5.5	Interp	retation der Laplace-Gleichung	26				
	5.6	Abbild	lungen	28				
		5.6.1	Schematische Programmstruktur des Algorithmus	28				
		5.6.2	Mittelwertbildung und Randbedingungen	28				
		5.6.3	Feld zweier ungleichnamig gepolter Punktladungen	29				
		5.6.4	Feld eines Plattenkondensators	29				
		5.6.5	Feld zweier gleichnamig gepolter Ringe	30				
		5.6.6	Feld zweier ineinander liegender Ringe	30				
		5.6.7	Feld zwischen einer Spitze und einer Platte	31				
		5.6.8	Randproblem	31				
		5.6.9	d- U -Diagramm	32				
	5.7	5.7Quellcode des Algorithmus						
	5.8							
Li	terat	ur		45				
Ei	desst	tattlich	ne Erklärung	46				

Abbildungsverzeichnis

1	Aufbau der Arrays	14
2	Schematische Programmstruktur	28
3	Mittelwertbildung	28
4	Randbedingungen	28
5	Zwei Punktladungen	29
6	Plattenkondensator	29
7	Zwei gleichnamige Ringe	30
8	Zwei ineinander liegende Ringe	30
9	Spitze und Platte	31
10	Einfluss des Randes	31
11	d- U -Diagramm	32

James Clerk Maxwell, geboren am 13. Juni 1831 in Edinburgh, war mit Gewissheit einer der bedeutendsten Physiker des 19. Jahrhunderts. Mit Fug und Recht kann sein Name in einem Atemzug mit bekannten Persönlichkeiten wie Galileo Galilei, Sir Isaac Newton oder Albert Einstein genannt werden. Letzterer bezeichnete Maxwells Leistungen als "das Tiefste und Fruchtbarste das die Physik seit Newton entdeckt hat"[7].

Maxwell befasste sich mit vielerlei physikalischen Problemen, vor allem aber konzentrierte er sich während seines leider recht kurzen Lebens - er starb am 5. November 1879 im Alter von 48 Jahren an Magenkrebs - auf Phänomene der Elektrizität. Seine erste Arbeit zu diesem Thema, welche sich auf Michael Faradays Kraftlinien bezog, veröffentlichte er bereits 1854, im Jahre seines Studiumabschlusses. Aber schon während seines Studiums an der Universität von Cambridge - er wechselte von Edinburgh dorthin stellte er Gleichungen auf, welche sowohl Faradays Erkenntnisse, als auch André Marie Ampères physikalische Beobachtungen mathematisch in Verbindung setzten. Doch erst im Jahre 1864 in dem Artikel "A Dynamical Theory of the Electromagnetic Field"[4] stellt er seine Differentialgleichungen der Royal Society vor.

Aufgrund der bahnbrechenden Bedeutung dieser Abhandlung habe ich mich nicht nur dazu entschlossen, die ersten Seiten des Artikels als Deckblatt zu wählen, sondern befasse ich mich im Rahmen meiner Arbeit auch mit mehreren Gleichungen Maxwells, welche die Möglichkeit bieten den Potentialverlauf verschiedener Ladungsverteilungen zu berechnen. Dabei ist es notwendig die physikalischen Zusammenhänge der Elektrostatik zu verstehen und eine numerische Lösung der Differentialgleichungen zu finden. Diese Probleme zu lösen machte ich mir zur Aufgabe.

2 Elektrostatik

2.1 Elektrische Ladungen

Anders als in der Elektrodynamik, befasst man sich in der Elektrostatik mit stationären Ladungen. Dabei stellt sich folgende Frage, die die Grundlage der Elektrostatik darstellt:

Welche Kraft \mathbf{F} verursacht eine Ladung q auf eine Probeladung Q? [3, S.59]

Die beiden Ladungen verbindet der Vektor **d**. Die Antwort wurde experimentell festgestellt und erhielt im Coulombschen Gesetz ihre Form:

$$\mathbf{F} = \frac{1}{4\pi\varepsilon_0} \, qQ \, \frac{1}{d^2} \, \hat{\mathbf{d}} \tag{2.1}$$

 ε_0 ist die elektrische Feldkonstante [5, S.1]:

$$\varepsilon_0 = 8,854187817 \cdot 10^{-12} \frac{As}{Vm} \tag{2.2}$$

Außerdem ist das nicht zu vernachlässigende Prinzip der Superposition zu nennen, welches besagt, dass die Kraft, die zwischen zwei Ladungen vorhanden ist, nicht von einer dritten Ladung beeinflusst wird. Daraus lässt sich folgern, dass alle Kräfte, die auf die Probeladung wirken, addiert werden können. [3, S.58]

2.2 Elektrische Felder

2.2.1 Elektrische Feldstärke E

Definitionsgemäß wird der Quotient der Kraft \mathbf{F} (2.1) und der Probeladung q als elektrische Feldstärke \mathbf{E} bezeichnet [2, S.5]:

$$\mathbf{E} = \frac{\mathbf{F}}{Q} \tag{2.3}$$

Wenn man (2.1) in (2.3) einsetzt, erhält man eine von Q unabhängige Gleichung für **E**:

$$\mathbf{E} = \frac{1}{4\pi\varepsilon_0} \ q \ \frac{1}{d^2} \ \mathbf{\hat{d}} \tag{2.4}$$

Dieser Ausdruck beschreibt das Feld, das ausgehend von der einzelnen Ladung q, eine Kraft auf die Probeladung Q an jedem Punkt des Raumes ausübt. Gemäß dem Superpositionsprinzip (siehe Kapitel 2.1) kann bei komplexeren Ladungsverteilungen über alle Einzelkräfte summiert werden.

$$\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^n \frac{q_i}{d_i^2} \hat{\mathbf{d}}_i \quad [3, S.60]$$
(2.5)

Bei einer kontinuierlichen Ladungsverteilung geht diese Summe in ein Integral über:

$$\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0} \int_V \frac{\rho(\mathbf{r}')}{d^2} \hat{\mathbf{d}} \, dV \quad [3, \, S.62]$$
(2.6)

Dabei ist \mathbf{r} die Position des Punkts der Messung zum Ursprung, \mathbf{r}'_i die Entfernung einer Punktladung q_i zum Ursprung und \mathbf{d}_i die Entfernung einer Punktladung q_i zum Punkt der Messung. Die kontinuierliche Ladungsverteilung, welche die Position \mathbf{r}' besitzt, wird mit ρ bezeichnet.

2.2.2 Elektrischer Fluss Φ_{el}

Eine weitere wichtige Größe im Zusammenhang mit dem elektrischen Feld ist der elektrische Fluss Φ_{el} . Im Feldlinienmodell entspricht diese Größe der Anzahl der Feldlinien, welche ein Flächenstück durchsetzen. Man stelle sich eine Kugel mit dem Radius d vor, dessen Mittelpunkt die Ladung q ist. Der elektrische Fluss an einem Punkt $P(\mathbf{r})$, welcher auf der Kugel liegt, entspricht dem elektrischen Fluss $d\Phi_{el}$ des ihn beinhaltenden Flächenstücks. Dieser ist folgendermaßen definiert [2, S.7]:

$$d\Phi_{el} = \mathbf{E} \cdot d\mathbf{A} \tag{2.7}$$

Die Gesamtheit der Feldlinien, die durch die Kugel gehen, kann mit Hilfe eines Integrals berechnet werden [2, S.8]:

$$\Phi_{el} = \oint_A \mathbf{E} \cdot d\mathbf{A} \tag{2.8}$$

Nun wird (2.6) in (2.8) eingesetzt (ausführliche Rechnung im Anhang, Kapitel 5.1). Fertig gekürzt folgt daraus dieses Ergebnis:

$$\Phi_{el} = \frac{1}{\varepsilon_0} \int_V \rho(\mathbf{r}') \, dV \tag{2.9}$$

2.2.3 Das Gaußsche Gesetz

Die soeben hergeleitete Gleichung in Verbindung mit dem Gaußschen Integralsatz führt zum physikalisch sehr wichtigen Gaußschen Gesetz in differentieller Form. Der Gaußsche Satz besagt, dass das Integral von \mathbf{E} über eine geschlossene Fläche dem Integral der Divergenz von \mathbf{E} (Erklärung der Bedeutung im Anhang, Kapitel 5.2) über ein Volumen, welches durch dieselbe Fläche begrenzt ist, entspricht [2, S.8]:

$$\Phi_{el} = \oint_A \mathbf{E} \cdot d\mathbf{A} = \int_{V(A)} \nabla \cdot \mathbf{E} \, dV \tag{2.10}$$

Wenn Gleichung (2.9) mit (2.10) gleichgesetzt wird, folgt [3, S.69]:

$$\int_{V} \nabla \cdot \mathbf{E} \, dV = \frac{1}{\varepsilon_0} \, \int_{V} \rho \, dV \tag{2.11}$$

Gilt das Gleichheitszeichen für zwei gleichartige Integrale, so gilt es auch für die Integranden:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \tag{2.12}$$

2.3 Potential

2.3.1 Die Maxwell-Gleichungen für elektrostatische Felder

Bewegt man eine Probeladung im elektrischen Feld vom Punkt A zum Punkt B, so wird gegen die elektrische Kraft Arbeit verrichtet. Diese hängt nicht vom Weg, sondern ausschließlich von den Endpunkten ab. Felder dieser Art nennt man konservativ. Diese Eigenschaft ist an den Maxwell-Gleichungen für elektrostatische Felder abzuleiten [3, S.232], welche sich wiederum von den allgemein gültigen Gleichungen der Elektrodynamik [3, S.560] ableiten lassen. Da sich Felder in der Elektrostatik nicht bewegen, sind die auftretenden Phänomene zeitunabhängig.

Die erste Maxwell-Gleichung der Elektrostatik (Das Gaußsche Gesetz, Kapitel 2.2.3, Gleichung (2.12)) wurde bereits näher erläutert. Die zweite, für die Konservativität des elektrischen Feldes verantwortliche Gleichung, lautet folgendermaßen:

$$\nabla \times E = 0 \tag{2.13}$$

Diese Formel besagt, dass das elektrische Feld frei von Wirbeln ist (siehe Anhang, Kapitel 5.2). Das heißt, dass Arbeit entlang eines beliebigen, geschlossenen Weges stets 0 ergibt [3, S.76]:

$$\oint \mathbf{F} \cdot d\mathbf{l} = 0 \tag{2.14}$$

2.3.2 Potentialdifferenzen

Aufgrund der Eigenschaft, dass das elektrische Feld konservativ ist, kann nun jedem Punkt im Raum elektrisches Potential U zugewiesen werden. Da Potential allerdings nur zwischen zwei Punkten gemessen werden kann, setzt man den Bezugspunkt ins Unendliche.

Das theoretische Potential am Punkt D, der die Entfernung d zum Ursprung hat und im Feld der Punktladung q liegt, berechnet man folgendermaßen:

(\mathbf{E} wird durch Gleichung (2.4) ersetzt)

$$U(D) = -\int_{\infty}^{d} \mathbf{E} \cdot d\mathbf{l} = -\frac{q}{4\pi\varepsilon_0 d}$$
(2.15)

Die Potentialdifferenz zwischen zwei Punkten A und B kann folgendermaßen berechnet werden:

$$U(B) - U(A) = -\int_{a}^{b} \mathbf{E} \cdot d\mathbf{l}$$
(2.16)

Dem fundamentalen Theorem der Gradienten [3, S.29] zu Folge gilt:

$$U(B) - U(A) = \int_{a}^{b} \nabla \cdot U \cdot d\mathbf{l}$$
(2.17)

Aus (2.16) und (2.17) folgt:

$$-\int_{a}^{b} \mathbf{E} \cdot d\mathbf{l} = \int_{a}^{b} \nabla \cdot U \cdot d\mathbf{l}$$

Da diese Gleichung für beliebige Punkte A, B gilt, müssen auch die Integranden gleich sein [3, S.78]:

$$\mathbf{E} = -\nabla \cdot U \tag{2.18}$$

2.3.3 Berechnung des Feldstärkevektors E mit Hilfe des Potentials

Ausgehend von Formel (2.18) kann die Stärke und die Richtung des elektrischen Feldes an jedem Punkt berechnet werden. Dazu muss der Vektor **E** in seine einzelnen Komponenten aufgeteilt werden, wobei wir uns auf zwei Dimensionen beschränken und die z-Koordinate gleich 0 setzen. Diese Einschränkung wird aus Gründen der Vereinfachung und Anschaulichkeit auch zukünftig beibehalten [6, S.700]:

$$\mathbf{E} = \begin{pmatrix} \mathbf{E}_x \\ \mathbf{E}_y \end{pmatrix} = -\nabla \cdot U = -\begin{pmatrix} \partial_x U \\ \partial_y U \end{pmatrix}$$
(2.19)

Auf diese Weise kann folglich ein Vektor für E berechnet werden, der die Richtung und den Betrag des elektrischen Feldes **E** an jedem beliebigen Punkt hat.

2.4 Poisson-Gleichung

Setzt man Formel (2.18) in Formel (2.12) ein, so ergibt sich daraus die Poisson-Gleichung, welche für die Berechnung des Potentialverlaufs entscheidend ist [3, S.83]:

$$\nabla \cdot \mathbf{E} = \nabla \cdot (-\nabla U) = -\nabla^2 U = \frac{\rho}{\varepsilon_0}$$

$$\boxed{\nabla^2 U = -\frac{\rho}{\varepsilon_0}}$$
bzw.
(2.20)

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = -\frac{\rho}{\varepsilon_0}$$

Die Summe der zweiten Ableitungen des elektrischen Potentials nach allen Koordinaten an einem bestimmten Punkt entspricht der Ladungsdichte an diesem Punkt.

Auf ladungsfreien Raum angewendet ergibt sich die Laplace Gleichung: (Interpretation im Anhang, Kapitel 5.5)

$$\nabla^2 U = 0 \tag{2.21}$$

3 Numerische Lösung der Poisson-Gleichung

3.1 Lösung der Poisson-Gleichung mit Hilfe der Finite-Differenzen-Methode

3.1.1 Differentialquotient und Differenzenquotient

Zur Lösung der Poisson-Gleichung, welche eine partielle Differentialgleichung darstellt, kann die Finite-Differenzen-Methode angewendet werden. Diese numerische Vorgehensweise erhält den Vorzug vor der viel zu umfangreichen analytischen Lösung. Dabei wird zur Berechnung der Ableitung der Differentialquotient (3.1) durch den Differenzenquotient (3.2) ersetzt. Die unendlich kleine Differenz $[x, x_0]$ wird durch h ersetzt. Für $\lim_{h\to 0}$ ist die Umwandlung noch vollkommen korrekt. Gibt man h aber einen endlichen Wert, so wird der bisher kontinuierliche Raum in ein Gitter verwandelt, bei welchem nur noch die Kreuzungspunkte zur Berechnung herangezogen werden. Dieser Schritt verfälscht zwar das Ergebnis, ist jedoch näherungsweise richtig:

$$f'(x) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{h \to 0} \frac{f(x) - f(x - h)}{h}$$
(3.1)

Üblicherweise wird beim Differenzenquotient zwischen Vorwärts-, Rückwärts- und Zentraldifferenzenquotient unterschieden:

1. Ableitung:

$$f'(x)^+ \approx \frac{f(x+h) - f(x)}{h}$$
 bzw. $f'(x)^- \approx \frac{f(x) - f(x-h)}{h}$ (3.2)

bzw.
$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$
 (3.3)

2. Ableitung:

$$f''(x)^+ \approx \frac{f'(x+h) - f'(x)}{h}$$
 etc. (3.4)

Um nun f''(x) ausschließlich in Abhängigkeit von f(x) zu erhalten, wird in den Vorwärtsdifferenzenquotient aus Gleichung (3.4) der Rückwärtsdifferenzenquotient aus Gleichung (3.2) eingesetzt:

$$f''(x) \approx \frac{\frac{f(x+h) - f(x)}{h} - \frac{f(x) - f(x-h)}{h}}{h}$$
$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$
(3.5)

Diese Funktion f''(x) bietet die Möglichkeit, die zweite Ableitung an verschiedenen Punkten im Raum zu berechnen, die voneinander stets den Abstand h besitzen.

3.1.2 Anwendung der Methode auf die Poisson-Gleichung

Nun wenden wir diese Vorgehensweise auf das Beispiel der Poisson-Gleichung (2.20) im Zweidimensionalen an:

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = -\frac{\rho}{\varepsilon_0}$$

Wenn man nun den Differenzenquotient (3.5) für die 2. Ableitung einsetzt, gewisse Vereinfachung mit einbezieht und nach U(x, y) auflöst kommt man zu folgendem Ergebnis: (ausführliche Rechnung im Anhang, Kapitel 5.4)

$$U(x,y) = \overline{U}(x,y) + \rho^*(x,y)$$
(3.6)

Gleichung (3.6) ist die numerische Lösung der Poisson-Gleichung (2.20):

Das Potential an einem Punkt P mit den Koordinaten x, y entspricht der Summe aus $\overline{U}(x, y)$ und $\rho^*(x, y)$, wobei $\overline{U}(x, y)$ den Mittelwert der vier benachbarten Punkte darstellt und $\rho^*(x, y)$ folgendermaßen definiert ist:

$$\rho^*(x,y) = \frac{h^2 \rho(x,y)}{4\varepsilon_0}$$
(3.7)

3.1.3 Anwendung der Methode zur Berechnung der E-Feld-Vektoren

Auch zur Berechnung der E-Feld-Vektoren an verschiedenen Punkten (Kapitel 2.3.3) lässt sich die Finite-Differenzen-Methode anwenden. Allerdings liefert dieses Vorgehen auch hier eine lediglich näherungsweise richtige Lösung. Die partielle Ableitung des Potentials (2.19) wird nämlich mit Hilfe des Zentraldifferenzenquotienten (3.3) berechnet:

$$\mathbf{E} = \begin{pmatrix} \partial_x U \\ \partial_y U \end{pmatrix} = \begin{pmatrix} \frac{U(x+h) - U(x-h)}{2h} \\ \frac{U(y+h) - U(y-h)}{2h} \end{pmatrix} = \begin{pmatrix} U^*(x) \\ U^*(y) \end{pmatrix}$$
(3.8)

3.2 Computer basierte, numerische Berechnung

3.2.1 Vorzug der iterativen Vorgehensweise

Gleichung (3.6) birgt ein Problem. Wenn man nun das Potential für jeden Punkt eines Gitters berechnet, erhält man noch kein realistisches Ergebnis. Das liegt an der Mittelwertbildung. Es tritt nämlich folgendes Phänomen auf:

Der Punkt P_0 wird mit Hilfe seiner benachbarten Punkte P_1 , P_2 , P_3 und P_4 (Punkte mit grauer Füllung in Abbildung 3, Anhang) ermittelt. Nach diesem Rechenschritt wird das Potential von P_1 kalkuliert, dessen Nachbarn der Punkt P_0 und drei weitere Punkte sind (Punkte mit weißer Füllung in Abbildung 3, Anhang). Das heißt, dass zuvor zur Berechnung von P_0 ein falscher Wert von P_1 verwendet wurde. Deshalb muss die Rechnung wiederholt werden. Dadurch konvergiert das Ergebnis schrittweise gegen die theoretische Lösung. Allerdings kann dieses Prinzip sehr aufwendig sein. Aus diesem Grund bietet es sich an, dass man sich gewisse technische Hilfsmittel zu Nutzen macht.

3.2.2 Erzeugung der Gitterpunkte mit Hilfe von Arrays

Da die in Kapitel 3.2.1 beschriebene, iterative Vorgehensweise sehr viele Wiederholungen der Rechnung nach sich ziehen kann, ist diese wie geschaffen für die Möglichkeiten eines Computer.

Der erste Schritt der Programmierung des Algorithmus ist die Erzeugung eines Gitters, wobei die Gitterpunkte durch die Variablen eines zweidimensionalen Arrays festgelegt sind.

Ein Array ist ein Datenfeld, mit dessen Hilfe auf die einzelnen Elemente zugegriffen werden kann. Jedem Element aus der ersten Reihe, in Verbindung mit einem Element aus der zweiten Reihe, wird ein Wert (in diesem Algorithmus beispielsweise das Potential oder die Ladung) zugeordnet.

Die Länge eines solchen zweidimensionalen Arrays ist das Produkt aus der Länge von Reihe 1 n_punkte_x und 2 n_punkte_y . Das entspricht dem Inhalt einer Fläche, auf der die Punkte des Gitters gleichmäßig verteilt sind bzw. der Anzahl n der Gitterpunkte.

Reihe 1	0	0	0	1	1	1	2	2	2	entspricht x-Werten
Reihe 2	0	1	2	0	1	2	0	1	2	entspricht y-Werten
Wert	U_0	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	Potential bei $P(x, y)$

Abbildung 1: Aufbau der Arrays

In diesem Fall stellt Reihe 1 die x-Koordinate, Reihe 2 die y-Koordinate eines Punktes P dar.

Für den Algorithmus sind drei Arrays jener Bauart notwendig. Dadurch erhält jeder Punkt drei Eigenschaften:

- neuewerte: Das Ergebnis, das bei der Lösung der Poisson-Gleichung entsteht, wird in diesem Array gespeichert.
- 2. *altewerte*: Wenn das Potential an jedem Punkt des Gitters berechnet und in *neuewerte* abgelegt wurde, werden die Werte für alle Punkte in *altewerte* kopiert.

Danach kann die Berechnung wiederholt und die neu errechneten Potentiale wieder in *neuewerte* gespeichert werden.

 ladung: In diesem Array kann Ladung auf bestimmte Gitterpunkte gelegt werden. Sie stellt f
ür jede Berechnung des Potentials eine Konstante dar, welche im Voraus gesetzt werden muss.

Die ersten Voraussetzungen wurden nun geschaffen, allerdings müssen noch weitere Überlegungen angestellt werden, um ein realistisches Ergebnis mit probaten Mitteln zu erreichen.

3.2.3 Die Parameter $z, \rho(x, y), n, U_S(x, y), s$

Ausgangspunkt der gesamten Berechnung ist eine Ladungsverteilung. Mit Hilfe von $\rho(x, y)$ (entsprechend im Quellcode: ladung[x][y]) können Ladungen gesetzt werden. Nahe liegender ist es, von festem Potential aus zu gehen, da dieses in der Realität leichter zu erzeugen ist als einzelne, ruhende Ladungen.

Deshalb kann auch ein festes Startpotential $U_S(x, y)$, wenn man es gewissen Punkten des Gitters zuweist, als Initiator dienen. Das gelingt dadurch, dass neuewerte[x][y] in jedem Durchgang der *while*-Schleife (Kapitel 3.2.4) automatisch auf $U_S(x, y)$ zurückgesetzt wird. Mehrere nebeneinander liegende Punkte mit $U_S(x, y)$ entsprechen in der Praxis einer Fläche mit konstantem Potential.

Ein weiterer Parameter, der vor der Berechnung bestimmt werden muss, ist die Anzahl der Gitterpunkte n (entsprechend im Quellcode: n_punkte_x bzw. n_punkte_y). Dieser Parameter nimmt starken Einfluss auf die Genauigkeit der Näherung, da man mit ihm die Entfernung zum Rand bestimmt (siehe dazu Kapitel 3.5.2).

Auch z ist ein wichtiger Parameter. Er ist das Kriterium der *while*-Schleife (Kapitel 3.2.4) und bestimmt somit, wann sie abgebrochen wird.

Der letzte Parameter, der vor dem Start des Algorithmus festgelegt wird, ist der Skalierungsfaktor s (entsprechend im Quellcode: s_faktor), dessen Nutzen in Kapitel 3.5.3 beschrieben wird.

3.2.4 Die while-Schleife zur Berechnung des Potentialverlaufs

Die *while*-Schleife berechnet in jedem Durchgang die neuen Werte des Potentials jedes einzelnen Punktes mit Hilfe des Arrays *altewerte* und speichert diese in das Array *neuewerte*. Dabei wird die in Kapitel 3.1.2 beschriebene Gleichung (3.6) gelöst:

```
349 double p = ladung[i][j]/4/epsilon;
350 neuewerte[i][j] = (altewerte[i + 1][j] + altewerte[i - 1][j] +
altewerte[i][j + 1] + altewerte[i][j - 1]) / 4 + p;
```

Das geschieht solange, bis folgende Bedingung erfüllt ist:

```
while (höchstedifferenz > z || höchstedifferenz = 0)
```

Ist $h\"{o}chstedifferenz$ größer als z übergibt das Array neuewerte seine Daten an altewerte und die Berechnung wird erneut ausgeführt:

```
338 altewerte[i][j] = neuewerte[i][j];
```

Ist höchstedifferenz allerdings kleiner oder gleich z, wird die Schleife abgebrochen und die Daten des Arrays neuewerte an die Plot-Library (Kapitel 5.3, Anhang) übergeben, welche diese grafisch darstellt. Durch höchstedifferenz == 0 wird verhindert, dass die Berechnung schon zu Beginn abgebrochen wird.

Die Variable höchstedifferenz wird folgendermaßen ermittelt:

```
330 höchstedifferenz = Math.abs(neuewerte[i][j] - altewerte[i][j]) /
höchsterwert;
```

Der Betrag der Differenz *neuewerte-altewerte* wird durch den betragsmäßig höchsten Wert geteilt, der in *neuewerte* auftritt. Dadurch wird *höchstedifferenz* standardisiert und Vergleiche verschiedener Ladungsverteilungen sind möglich.

Das Feststellen von *höchsterwert* gelingt mit Hilfe einer *if*-Abfrage, bei welcher stets der Betrag des soeben errechneten Werts mit dem bisher höchsten Wert verglichen wird:

```
314 if (Math.abs(neuewerte[i][j]) > höchsterwert) {
```

```
315 höchsterwert = Math.abs(neuewerte[i][j]);}
```

3.2.5 Berechnung der E-Feld-Vektoren

Um eine Darstellung des **E**-Feld-Vektors an jedem Punkt des Gitters zu erhalten, wird mit Hilfe der Gleichung (3.8) sowohl die x- als auch die y-Koordinante berechnet. Diese Daten werden in zwei zweidimensionale Arrays namens XYZx und XYZy gespeichert:

```
440 XYZx[m][1] = -(neuewerte[j+1][i] - neuewerte[j-1][i])/2;
```

443 XYZy[m][1] = -(neuewerte[j][i+1] - neuewerte[j][i-1])/2;

XYZx und XYZy unterscheiden sich allerdings von denen in Kapitel 3.2.2 beschriebenen Arrays. Die beiden Reihen dieser Arrays entsprechen nicht mehr der x- und der y-Koordinate, sondern jeder Punkt des Gitters erhält einen Index, welcher in Reihe 1 gespeichert wird, und eine Komponente des Feldstärkevektors, gespeichert in Reihe 2. Den Namen der Arrays entsprechend wird in XYZx die x-Komponente und in XYZydie y-Koordinate hinterlegt.

Aus Gründen der Übersichtlichkeit wird die Länge eines jeden Feldstärkevektors durch die Länge des längsten Vektors dividiert. Letztendlich werden auch XYZx und XYZy dem Plotter (Kapitel 5.3, Anhang) übergeben, welcher daraus die Graphen erzeugt.

3.2.6 Randbedingungen

Bei der hier gewählten Vorgehensweise kann nur ein begrenztes Gebiet in Hinsicht auf das Potential untersucht werden. Die Begrenzung findet beim Definieren der Gitterpunkte statt, woraus sich folgendes Problem ergibt:

Wie soll das Potential von einem Punkt berechnet werden, der am äußersten Rand der betrachteten Fläche liegt? Die Problematik liegt darin, dass auf mindestens einer (bei Eckpunkten sogar drei) der vier benachbarten Seiten kein Nachbarpunkt mehr definiert ist. Die Berechnung des Mittelwerts ist somit für die Randpunkte nicht möglich. Dieser Fehler setzt sich wie eine Kettenreaktion für alle anderen Punkte fort. Aus diesem Grund müssen die Ränder explizit behandelt werden.

Eine Möglichkeit das Problem zu lösen ist das Festlegen der äußersten Gitterpunkte des Bereichs A durch einen festen Wert des Potentials. Dadurch entsteht zwar ein

Fehler, allerdings wird die Berechnung nicht unterbrochen und die Darstellung eines näherungsweise richtigen Potentialverlaufs ist möglich. Um das zu bewerkstelligen, ist es nicht notwendig, jeden Randpunkt explizit zu definieren. Viel leichter ist es, die Mittelwertbildung nur für eine kleine Fläche A' durchzuführen. Die Seitenlänge a' dieser Fläche A' ist genau um zwei Punkte kürzer als a. A' liegt folglich in A, besitzt allerdings keinen der Randpunkte von A. Folgende Veränderung muss dafür im Algorithmus vorgenommen werden:

o vorher: for $(i = 0; i < n_punkte_x; i++)$

346 nachher: for $(i = 1; j < n_{ukte_x} - 1; i++)$

Die for-Schleife, welche jedes Element eines Arrays aufruft, erreicht folglich nur noch die Elemente, deren x-Koordinate einen Wert aus dem Intervall [1; $n_punkte_x - 2$] annimmt. (y-Koordinate analog)

Aufgrund der Tatsache, dass die Ränder aus der Berechnung ausgeschlossen sind, wird das Potential dieser auf den Ausgangswert 0 festgesetzt. Die grafische Veranschaulichung ist im Anhang unter Abbildung 4 zu finden.

3.3 Lösungsbeispiele

3.3.1 Feld zweier ungleichnamig gepolter Punktladungen

Durch Platzieren zweier Punktladungen mit verschiedenem Vorzeichen, lässt sich eine gute Abbildung des Potentialverlaufs und der Feldlinien erzeugen. Wie in Abbildung 5 gut zu erkennen ist, verlaufen die Feldlinien von der positiven Ladung zur negativen.

3.3.2 Feld eines Plattenkondensators

Aus der Abbildung 6 lassen sich physikalische Besonderheiten ablesen. Zum einen ändert sich das Potential zwischen den Platten konstant, zum anderen herrscht ein homogenes Feld zwischen den Platten. Anhand der Feldlinien ist aber auch deutlich zu erkennen, dass das Feld an den Rändern der Platten radialsymmetrisch wird.

3.3.3 Feld zweier gleichnamig gepolter Ringe

Am Beispiel von zwei gleichnamig gepolten Ringen (Abbildung 7) erkennt man das Superpositionsprinzip (Kapitel 2.1). Mittig zwischen den beiden Ringen summiert sich das Potential. Des weiteren zeigen die Feldlinien die Gleichnamigkeit der Polung. Keine Feldlinie berührt beide Ringe. Außerdem ist es interessant, dass in den Ringen kein Feld auftritt.

3.3.4 Feld zweier ineinander liegender Ringe

Setzt man zwei unterschiedlich gepolte Ringe ineinander, so entsteht in der Mitte des inneren Ringes feldfreier Raum. Im Raum zwischen den beiden Ringen, sowie außerhalb des äußeren Ringes entsteht ein elektrisches Feld und somit eine Potentialdifferenz. An Abbildung 8 lässt sich deutlich erkennen, dass der innere Ring positiv und der äußere negativ geladen ist.

3.3.5 Feld zwischen einer Spitze und einer Platte

Die Kombination einer Spitze mit einer Platte macht die Eigenschaft der Feldlinien deutlich, stets senkrecht aus einer Oberfläche ein und aus zu treten (Abbildung 9).

3.4 Vergleich der Messwerte mit theoretischen Werten

Um einen Bezug zur Theorie herzustellen, führen wir eine Messung durch. Dabei simulieren wir ein elektrisches Feld mit Hilfe einer einzelnen Punktladung (entspricht dem Parameter ρ), welche sich in der Mitte der betrachteten Fläche befindet. Die Punkte, an welchen das Potential gemessen und berechnet wird besitzen alle die *x*-Koordinate 500 und eine *y*-Koordinaten aus dem Intervall [501 – 799]. Sie liegen somit auf einer Geraden, die von der felderzeugenden Punktladung in Richtung des Randes verläuft.

Die Messung wird mit folgenden Parametern durchgeführt:

 $n=1000^2=1000000;\,\rho(500,500)=1;\,z=0.0001$

Trägt man die Messwerte in ein d-U-Diagramm (Anhang, Abbildung 11) ein, so lässt

sich deutlich erkennen, dass das Potential U mit wachsender Entfernung d zur Punktladung q abnimmt. Wie Gleichung (2.15) bestätigt, gilt: $U \sim d^{-1}$

Der ermittelte Graph ist dem zu erwartenden Graphen der Funktion $f(x) = x^{-1}$ sehr ähnlich, weicht aber entscheidend ab und verläuft nicht asymptotisch. Im Gegensatz zu U(x, y) - die Theorie besagt, dass das Potential erst im Unendlichen 0 wird - nimmt $U_b(x, y)$ den Wert 0 an.

Vergleicht man verschiedene theoretische Werte U(x, y) mit ermittelten Werten des Potentials $U_b(x, y)$, so erkennt man, dass die Genauigkeit unterschiedlich ist. Die besten Ergebnisse werden in größtmöglicher Entfernung zum Rand und zur Ladung erzielt.

Punkt (x,y)	(500,504)	(500, 525)	(500, 568)	(500, 598)	(500,737)
d	4,0	25	68	98	237
$U_b(x,y)$ in GV	38, 1894	7,7211	0,267869	0,011433	$9.31610 \cdot 10^{-14}$
U(x,y) in GV	2,2469	0,35950	0,13217	0,091710	0,037922

3.5 Fehlererklärung

3.5.1 Mittelwertbildung mit nur vier Werten

Dadurch, dass die Finite-Differenzen-Methode (Kapitel 3.1) angewendet wurde, wird das Potential eines Punktes durch den Mittelwert von nur vier Punkten berechnet. Dass das gerade in Bereichen extremer Potentialveränderungen, z.B. in der Nähe der felderzeugenden Ladung, zu Fehlern führt, ist nicht verwunderlich, denn wie im Anhang Kapitel 5.5 (Gleichung (5.10)) erklärt, müsste eigentlich der Mittelwert aller umliegenden Punkte ermittelt werden.

3.5.2 Einfluss der Randbedingungen auf das Ergebnis

Da man zur korrekten Berechnung unendlich viele Punkte einbeziehen müsste, dies aber nicht möglich ist, ergeben sich Ungenauigkeiten. Die Randbedingungen, welche für die Lösung unbedingt notwendig sind, verfälschen die Werte der nahe am Rand liegenden Punkte stark und damit den Wert aller Punkte (Randbedingungen, Kapitel 3.2.6). Das lässt sich auch eindeutig am Verlauf des Graphen (Abbildung 11) festmachen: Das eigentlich asymptotisch verlaufende Potential nimmt den Wert 0 an. Das beeinflusst den Verlauf des gesamten Graphen, der deshalb schneller abnimmt. Zur Verdeutlichung dieser Problematik bietet sich an, einen Punkt in der Nähe des Randes mit der gleichen Ladung zu versehen, wie den Punkt in der Mitte der Fläche. Es wird sich herausstellen, dass der außen liegende Punkt trotz der betragsmäßig gleichen Ladung ein niedrigeres Potential erzeugt. Verdeutlicht wird diese Feststellung in Abbildung 10 im Anhang.

3.5.3 Rechendauer (Skalierungsfaktor s)

Hat man unendlich viel Zeit zur Verfügung, so erhält man mit diesem Programm eine ziemlich genaue Lösung für den Potentialverlauf. Allerdings ist das nicht möglich und so entstehen nicht zu vernachlässigende Fehler. Die Anzahl n der Punkte, die in die Berechnung einbezogen werden, die Komplexität der Ladungsverteilung, sowie das Kriterium z der *while*-Schleife beeinflussen die Rechendauer stark. Dabei gilt:

Je höher n, je komplexer die Ladungsverteilung und je niedriger z desto mehr Zeit benötigt die Berechnung.

Aber nicht nur die Dauer der Berechnung wird durch n vergrößert. Auch wenn der Algorithmus abgeschlossen ist, benötigt der Computer Rechenleistung zur interaktiven graphischen Darstellung der Ergebnisse. Um genaue Werte zu erhalten, aber dennoch den Computer beim Betrachten des Graphen nicht zu überlasten, wurde ein Skalierungsfaktor s eingeführt, der nur einen gewissen Bruchteil der berechneten Punkte an den Plotter übergibt. Im Quellcode wird s als s_faktor bezeichnet. Die theoretischen und die ermittelten Ergebnisse dieser Arbeit weichen zwar an manchen Stellen voneinander ab, allerdings kann festgestellt werden, dass die wichtigsten Phänomene der Elektrostatik realitätsnah darstellbar sind. Für die exakte Berechnung absoluter Werte wären leistungsstarke Computer nötig. Aber auch mit geringerem Aufwand sind beachtliche Ergebnisse möglich. Zumindest qualitativ lassen sich Sinnzusammenhänge der Physik hervorragend verdeutlichen.

Da eben diese Sinnzusammenhänge Inhalte des Leistungskurses Physik darstellen, reizte es mich, ein im Unterricht anwendbares Programm zu entwickeln, das dem Lehrer die Möglichkeit gibt das anfangs rätselhafte Potential zu veranschaulichen und von umständlichen Versuchen, wie die Darstellung der Feldlinien mit Hilfe von Grieskörnern und Rizinusöl, ab zu sehen.

Die Erwartungen, die ich anfangs an meine Arbeit stellte, wurden stetig nach oben korrigiert. Dadurch steigerte ich nicht nur mein Interesse an Physik, Mathematik und Informatik. Das Auseinandersetzen mit umfangreichen physikalischen Abhandlungen gewährte mir darüber hinaus Einblicke in komplexere Themengebiete der Physik. Auch wenn oder gerade weil vieles in den Büchern noch unverständlich wirkt, weckte das Recherchieren das Bedürfnis, sich zukünftig intensiver damit zu beschäftigen.

5 Anhang

5.1 Ausführliche Herleitung des elektrischen Flusses in Abhängigkeit der Ladungsverteilung

$$\Phi_{el} = \oint_{A} \frac{1}{4\pi\varepsilon_{0}} \int_{V} \frac{\rho(\mathbf{r}')}{d^{2}} \hat{\mathbf{d}} \, dV \, d\mathbf{A}$$
$$d\mathbf{A} = \hat{\mathbf{d}} \, dA$$
$$\Rightarrow \Phi_{el} = \oint_{A} \frac{1}{4\pi\varepsilon_{0}} \int_{V} \frac{\rho(\mathbf{r}')}{d^{2}} \hat{\mathbf{d}} \, dV \hat{\mathbf{d}} \, dA$$
$$\hat{\mathbf{d}}^{2} = \hat{d}^{2} \cos \varphi = 1$$
$$\Rightarrow \Phi_{el} = \int_{V} \frac{\rho(\mathbf{r}')}{d^{2}} \, dV \frac{1}{4\pi\varepsilon_{0}} \oint_{A} dA$$
$$\oint_{A} dA = 4\pi d^{2}$$
$$\Rightarrow \Phi_{el} = \frac{1}{\varepsilon_{0}} \int_{V} \rho(\mathbf{r}') \, dV \qquad (5.1)$$

5.2 Bedeutung von Gradient, Divergenz und Rotation

1. Der Gradient: ∇U [3, S.16]

$$\nabla U = \left(\mathbf{\hat{x}} \frac{\partial}{\partial x} + \mathbf{\hat{y}} \frac{\partial}{\partial y} + \mathbf{\hat{z}} \frac{\partial}{\partial z} \right) U$$
(5.2)

Der Gradient ist ein Vektoroperator. Er besteht aus der Summe der partiellen Ableitungen aller Koordinaten.

Bei der partiellen Ableitung wird nach nur einer Variablen abgeleitet und die übrigen konstant gehalten. Das bietet sich bei mehrdimensionalen Gleichungen an. 2. Die Divergenz: $\nabla \cdot \mathbf{E}$ [3, S.17]

$$\nabla \cdot E = \left(\mathbf{\hat{x}} \frac{\partial}{\partial x} + \mathbf{\hat{y}} \frac{\partial}{\partial y} + \mathbf{\hat{z}} \frac{\partial}{\partial z} \right) \cdot \left(E_x \mathbf{\hat{x}} + E_y \mathbf{\hat{y}} + E_z \mathbf{\hat{z}} \right)$$
(5.3)

$$= \left(\frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} + \frac{\partial E_z}{\partial z}\right)$$
(5.4)

Die Divergenz gibt Auskunft darüber, ob Vektoren einer Vektorfunktion geordnet oder ordnunglos verteilt sind. Wenn $\nabla \cdot \mathbf{E} \neq 0$ (gültig für elektrische Felder), so orientieren sich die Vektoren zu einer "Quelle" hin.

3. Die Rotation: $\nabla \times \mathbf{E}$ [3, S.19]

Mit Hilfe der Rotation lassen sich Aussagen über eventuelle Wirbel einer Vektorfunktion treffen. Für das elektrische Feld gilt $\nabla \times \mathbf{E} = 0$. Das bedeutet, das Feld ist wirbelfrei. Diese Eigenschaft ist ausschlaggebend für dessen Konservativität.

5.3 Grafische Darstellung des Arrays neuewerte

Der sogenanten Plotter ist nicht von mir eigenhändig programmiert worden. Ich greife dabei auf eine öffentliche *library* [1] zurück. Diese *library* verarbeitet die Punkte, die ich zuvor errechnet habe, zu einer Grafik. Das Benutzen dieser kostenlos bereitgestellten Software erspart das Programmieren einer sehr aufwändigen 3D-Grafik. Die Tatsache, dass die *library* in der Programmiersprache *java* zur Verfügung stand, bewegte mich dazu, mein Programm ebenfalls in dieser Sprache zu schreiben.

5.4 Ausführliche Anwendung der Finite-Differenzen-Methode auf die Poisson-Gleichung

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = -\frac{\rho}{\varepsilon_0}$$

Einsetzen des Differenzenquotienten:

$$\frac{U(x+h,y) - 2U(x,y) + U(x-h,y)}{h^2} + \frac{U(x,y+h) - 2U(x,y) + U(x,y-h)}{h^2} = \frac{U(x+h,y) + U(x-h,y) + U(x,y+h) + U(x,y-h) - 4U(x,y)}{h^2}$$

Auflösen nach U(x, y) und Verwendung der Abkürzungen [1.], [2.] und [3.]:

$$\begin{aligned} \frac{C(x,y) - 4U(x,y)}{h^2} &= -\frac{\rho(x,y)}{\varepsilon_0} \\ &= \frac{4U(x,y)}{h^2} = \frac{\rho(x,y)}{\varepsilon_0} + \frac{C(x,y)}{h^2} \\ &= U(x,y) = \frac{1}{4} \left[C(x,y) + \frac{h^2 \rho(x,y)}{\varepsilon_0} \right] \frac{[2]}{=} \\ &= U(x,y) = \overline{U}(x,y) + \frac{h^2 \rho(x,y)}{4\varepsilon_0} \frac{[3]}{=} \\ &= \overline{U}(x,y) = \overline{U}(x,y) + \rho^*(x,y) \end{aligned}$$
(5.5)
Abkürzungen:
$$\begin{vmatrix} 1. & C(x,y) = U(x+h,y) + U(x-h,y) + U(x,y+h) + U(x,y-h) \\ 2. & \overline{U}(x,y) = \frac{1}{4}C(x,y) \\ 3. & \rho^*(x,y) = \frac{h^2 \rho(x,y)}{4\varepsilon_0} \end{vmatrix}$$

Anmerkung: Zwei benachbarte Punkte befinden sich im Abstand \boldsymbol{h} zueinander.

Mit Hilfe der Laplace-Gleichung kann das elektrische Potential in ladungsfreiem Raum bei gegebener Ladungsverteilung berechnet werden.

Allerdings ist die Lösung einer partiellen Differentialgleichung nicht eindeutig bestimmbar. Um dieses Problem einzuschränken betrachten wir vorerst das Verhalten der Laplace-Gleichung im Eindimensionalen. Die Laplace-Gleichung sieht, wenn sie nur noch von einer Variablen x abhängt, folgendermaßen aus [3, S.111]:

$$\frac{d^2U}{dx^2} = 0\tag{5.6}$$

Lösungen für diese Differentialgleichung hängen nicht von unendlich vielen Variablen, sondern nurmehr von m und t ab:

$$\int \frac{d^2 U}{dx^2} dx = \int 0 dx$$
$$\frac{dU}{dx} = m \; ; m \in \mathbf{R}$$
$$\int \frac{dU}{dx} dx = \int m \, dx \; ; m \in \mathbf{R}$$
$$U = mx + t \; ; m, t \in \mathbf{R}$$
(5.7)

Offensichtlich handelt es sich dabei um eine Geradengleichung. Das Reduzieren auf eine Dimension lässt sich schwer in die Praxis übertragen. Allerdings hilft uns dieser Schritt Analogien zum komplizierteren Fall der Zweidimensionalität herzustellen. Dafür müssen zwei Aspekte beachtet werden, welche beide durch die Eigenschaften einer Geraden begründet werden:

1. Das Potential eines Punktes P kann man stets durch den Mittelwert des Potentials zweier Punkte ermitteln, welche die selbe Entfernung a zu P haben:

$$U(P) = \frac{1}{2}[U(P+a) - U(P-a)]$$
(5.8)

2. Die Funktion verbindet zwei Punkte stets auf dem direkten und kürzesten Weg. Das heißt, es können keine Maxima oder Minima, abgesehen von Rand/- oder Fixpunkten, entstehen. Eigentlich ist diese Feststellung eine Folge der ersten Eigenschaft, da der Wert eines Maximums nie gleich dem Mittelwert der umliegenden Punkte sein kann.

Die partielle Differentialgleichung, zu der die Laplace-Gleichung in zwei Dimensionen wird, lässt keine derart leichte Lösung mehr zu [3, S.112]:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \tag{5.9}$$

Allerdings gelten auch hier, die für die eindimensionale Laplace-Gleichung aufgestellten Thesen:

 Der Mittelwert aller Punkte, welche die Entfernung r zu einem Punkt P haben, sich folglich auf einem Kreis mit dem Radius r und Mittelpunkt P befinden, entspricht dem Wert von P:

$$U(P) = \frac{1}{2\pi r} \oint_{Kreis} U dl \tag{5.10}$$

 Wiederum kann aus 1. geschlossen werden, dass keine Maxima oder Minima auftreten. Die Funktion erzeugt den Graphen, der die Randpunkte und gewisse gegebene Werte möglichst glatt verbindet.

Diese Eigenschaften erlauben es einem nun, eine annähernd richtige Lösung durch iterative Mittelwertbildung zu erhalten.

5.6 Abbildungen

5.6.1 Schematische Programmstruktur des Algorithmus



Abbildung 2: Schematische Programmstruktur

5.6.2 Mittelwertbildung und Randbedingungen



Abbildung 3: Mittelwertbildung



Abbildung 4: Randbedingungen

5.6.3 Feld zweier ungleichnamig gepolter Punktladungen



Abbildung 5: Zwei Punktladungen



5.6.4 Feld eines Plattenkondensators

Abbildung 6: Plattenkondensator

5.6.5 Feld zweier gleichnamig gepolter Ringe



Abbildung 7: Zwei gleichnamige Ringe

5.6.6 Feld zweier ineinander liegender Ringe



Abbildung 8: Zwei ineinander liegende Ringe

5.6.7 Feld zwischen einer Spitze und einer Platte



Abbildung 9: Spitze und Platte

5.6.8 Randproblem



Abbildung 10: Einfluss des Randes



5.6.9 *d*-*U*-Diagramm

Abbildung 11: *d*-*U*-Diagramm

Der Punkt an dem sich die felderzeugende Ladung befindet hat die Koordinaten (500, 500). Wenn man nun die *x*-Koordinate beibehält, die *y*-Koordinate allerdings von 500 schrittweise erhöht, so steigt der Abstand *d* zum Punkt der Ladung. Aufgrund dieser Tatsache ist das Antragen der *y*-Koordinate äquivalent zu einem *d*-*U*-Diagramm.

5.7 Quellcode des Algorithmus

```
1
 package javaapplication2;
\mathbf{2}
3
4 import java.awt.GridLayout;
<sup>5</sup> import javax.swing.JFrame;
6
7 import javax.swing.SwingWorker;
  import org.math.plot.*;
8
9
  /*
10
   * To change this template, choose Tools | Templates
11
   * and open the template in the editor.
12
   */
13
  /**
14
   *
15
   * @author Martin Gerer
16
   */
17
18
  public class Berechnung extends SwingWorker<Void, Void> {
19
           //Hier kann die Anzahl der Gitterpunkte gesetzt werden!
20
           int n_punkte_x; int n_punkte_y;
^{21}
           //Anzahl der dargestellten Punkte
22
           int s_faktor;
23
           String ladungsverteilung;
24
           int l; int r2; int d; int r3; int r4; int r5; int r6; int x1;
25
           int y1; int x2; int y2; double z;
26
           int progress = 0;
27
28
29
       @Override
30
           public Void doInBackground() {
31
                berechnen();
32
                //Initialize progress property.
33
                return null;
34
           }
35
36
           /*
37
            * Executed in event dispatching thread
38
            */
39
           @Override
40
           public void done() {
41
                System.out.println("done");
42
           }
43
           public void setVariables (
44
```

45	//Hier kann die Anzahl der Gitterpunkte gesetzt werden!
46	<pre>int n_punkte_x, int n_punkte_y,</pre>
47	//Anzahl der dargestellten Punkte
48	int s_faktor, String ladungsverteilung, int 1, int r2, int d,
49	\mathbf{int} r3, \mathbf{int} r4, \mathbf{int} r5, \mathbf{int} r6, \mathbf{int} x1, \mathbf{int} y1, \mathbf{int} x2,
50	int y2, double z
51){
52	this.n_punkte_x = n_punkte_x;
53	this.n_punkte_y = n_punkte_y;
54	this.s_faktor = s_faktor;
55	this.ladungsverteilung = ladungsverteilung;
56	this.l = l;
57	$\mathbf{this} \cdot \mathbf{r}2 = \mathbf{r}2;$
58	$\mathbf{this} \cdot \mathbf{d} = \mathbf{d};$
59	this.r3 = r3;
60	this.r4 = r4;
61	this.r5 = r5;
62	this.r6 = r6;
63	this.x1 = x1;
64	$\mathbf{this} \cdot \mathbf{y1} = \mathbf{y1};$
65	this.x2 = x2;
66	this.y2 = y2;
67	this.z = z;
68	}
69	public void berechnen()
70	{
71	//Start Berechnung—
72	double $h = 1;$
73	
74	//Anzahl der Durchlaufe
75	$int n_durchlauf = 0;$
76	
77	// Epsilon 0
78	double epsilon = $8.854187817E - 12;$
79	
80	//Erzeugen der Arrays
81	double [][] altewerte = new double [n_punkte_x][n_punkte_y];
82	double [][] neuewerte = new double [n_punkte_x][n_punkte_y];
83	double [][] ladung = new double [n_punkte_x][n_punkte_y];
84	
85	$\inf_{i=1}^{n} 1 = 0;$
86	$\operatorname{Int} J = 0;$
87	
87 88	//////Hier kann Ladungsverteilung gesetzt werden!///////
87 88 89	//////Hier kann Ladungsverteilung gesetzt werden!//////

```
// 2 Punktladungen ///////
91
         if (ladungsverteilung == "punktamrand") {
92
         ladung[x1][y1] = 1;
93
         ladung[x2][y2] = -1;
94
95
         }
96
97
         98
99
         if (ladungsverteilung == "einepunktladung") {
100
101
         //ladung[(n_punkte_x)/2][(n_punkte_y)/2] = 1;
102
         ladung[x1][y1] = 1;
103
         ł
104
105
106
         107
108
         double[][] festewerte = new double[n_punkte_x][n_punkte_y];
109
110
111
         112
         if (ladungsverteilung == "kreismitradius") {
113
114
         double w = 0;
115
          for (i = 1; i < n_{m_i} - 1; i++)
116
                for (j = 1; j < n_{y} - 1; j++)
117
                       w = Math.round(Math.sqrt(Math.pow(
118
                               (n_punkte_x/2-i), 2)
119
                              +Math.pow((n_y/2-j),2));
120
                        if(w = r2){
121
                        festewerte [i][j]=1;
122
123
                }
124
         }
125
126
127
         128
         if (ladungsverteilung == "plattenkondensator") {
129
         for ( i=n_punkte_x/4; i <3*n_punkte_x/4; i++){
130
             j=n_vx/2 + d/2;
131
             festewerte [i][j]=1;
132
133
         for (i=n_wkte_x/4; i < 3*n_wkte_x/4; i++)
134
             j=n_vx/2 - d/2;
135
             festewerte [i][j] = -1;
136
```

137138 139 ///// 2 Kreise mit Radius r3 140if (ladungsverteilung == "zweikreisemradius") { 141 double w = 0;142for $(i = 1; i < n_{v_i} - 1; i++)$ 143 for $(j = 1; j < n_v = 1; j++)$ { 144w = Math.round(Math.sqrt(Math.pow(145 $(3*n_punkte_x/4-i), 2)$ 146+Math.pow($(3*n_y/4-j), 2)$); 147if(w = r3){ 148 festewerte [i] [j] = 1;149} 150} 151152for $(i = 1; i < n_{m_i} - 1; i++)$ 153 for $(j = 1; j < n_v - 1; j++)$ { 154w = Math.round(Math.sqrt(Math.pow(155(n punkte x/4-i),2) 156+Math.pow(($n_y/4-j$),2)); 157if(w = r3){ 158festewerte [i] [j] = 1;159160 } 161162 163 164///// 2 Kreise +,- mit Radius r4 165if (ladungsverteilung == "zweikreise+mitradius") { 166 double w = 0;167 for $(i = 1; i < n_{ux} - 1; i++)$ { 168 for $(j = 1; j < n_v = 1; j++)$ { 169 w = Math.round(Math.sqrt(Math.pow(170(3*n punkte x/4-i), 2)171 +Math.pow((3*n punkte y/4-j), 2)); 172if(w = r4){ 173festewerte [i][j]=1;174175ł } 176 177 for $(i = 1; i < n_{ux} - 1; i++)$ { 178 for $(j = 1; j < n_v = 1; j++)$ { 179w = Math.round(Math.sqrt(Math.pow(180



```
for ( i=n_punkte_x / 2; i <=3*n_punkte_x / 4; i++){
227
               j=-i+n_{y}-1;
228
               festewerte[i][j]=1;
229
230
           }
231
232
           233
           if (ladungsverteilung == "viereck") {
234
           for (i = 1; i < n_{m_i} - 1; i++)
235
                   for (j = 1; j < n_v = 1; j++) {
236
                       if (Math.abs(i - n_punkte_x/2))
237
                               <= 1/2 & Math.abs(j - n_punkte_y
238
                                (2) <= 1/2)
239
                           if (Math.abs(i - n_punkte_x/2) = 1/2
240
                                || Math.abs(j - n_punkte_y/2) == 1/2) \{
241
                                festewerte [i][j]=1;
242
                           }
243
                       }
244
                   }
245
               }
246
           }
247
248
           /////// Faradayscher Käfig
249
           if (ladungsverteilung == "faradayscher") {
250
           /////Platten//////
251
           for (i=n_wkte_x/4; i < 3*n_wkte_x/4; i++)
252
               j=n_{x/4};
253
               festewerte [i][j]=1;
254
           }
255
           for (i=n_wkte_x/4; i < 3*n_wkte_x/4; i++)
256
               j=3*n_{x}/4;
257
               festewerte [i][j] = -1;
258
           }
259
           ////Krei//////
260
           double w = 0;
261
            for (i = 1; i < n_{m_i} - 1; i++)
262
                   for (j = 1; j < n_v = 1; j++) {
263
                           w = Math.round(Math.sqrt(Math.pow(
264
                                    (n_punkte_x/2-i), 2)
265
                                   +Math.pow((n_y/2-j),2));
266
                           if(w = n_{w_x/8})
267
                           festewerte [i][j]=1;
268
                           }
269
                   }
270
271
```

```
}
272
273
274
275
276
           double höchsterwert = 0;
277
            int x_wert = 0;
278
            int y_wert = 0;
279
280
            double höchstedifferenz = 1000;
281
            double höchstedifferenzstatus = höchstedifferenz;
282
            int x_differenz = 0;
283
            int y_differenz = 0;
284
285
                this.firePropertyChange("berechnungstatus", "",
286
                          "While-Schleife");
287
288
            //Schleife bis Abstand kleiner als angegeben!
289
            while (höchstedifferenz > z || höchstedifferenz = 0) {
290
291
                this.firePropertyChange("höchstedifferenz",
292
                         höchstedifferenzstatus,
293
                         höchstedifferenz);
294
                h\ddot{o}chstedifferenzstatus = h\ddot{o}chstedifferenz;
295
296
297
                n_durchlauf = n_durchlauf+1;
298
                höchsterwert = 0;
299
                h\ddot{o}chstedifferenz = 0.0;
300
301
             for (i = 1; i < n_punkte_x - 1; i++) {
302
                     for (j = 1; j < n_v = 1; j++) {
303
                          if (festewerte[i][j] != 0){
304
                              neuewerte[i][j] = festewerte[i][j];
305
306
307
                     }
308
309
                //Finden des höchsten Wertes um relativen Bezug
310
                //herzustellen
311
                for (i = 1; i < n_{m_x} - 1; i++)
312
                     for (j = 1; j < n_v = 1; j++) {
313
                         if (Math.abs(neuewerte[i][j]) > höchsterwert) {
314
                              h\ddot{o}chsterwert = Math.abs(neuewerte |i||j|);
315
                              x wert = i;
316
                              y_wert = j;
317
```



//Ausgabe von Durchläufe 362 363 364 //Ende Berechnung 365 366 //Ploten der Punkte 367 //Skalierungsfaktor: 368 369 double [][] werteplot = new double [n_punkte_x / s_faktor] 370 [n_punkte_y / s_faktor]; 371 for $(i = 0; i < n_punkte_x / s_faktor; i++)$ { 372 for $(j = 0; j < n_punkte_y / s_faktor; j++) {$ 373 werteplot [i] [j] = neuewerte [i * s_faktor] [j * 374s_faktor]; } 375 } 376377 // define your data 378 **double**[] x = **new double**[n_punkte_x / s_faktor]; 379 **double**[] y = **new double**[n_punkte_x / s_faktor]; 380 381for $(i = 0; i < n_punkte_x / s_faktor; i++)$ { 382 $x[i] = s_faktor * i * h;$ 383 384 for $(i = 0; i < n_punkte_y / s_faktor; i++)$ { 385 $y[i] = s_faktor * i * h;$ 386 } 387 388 389 create your PlotPanel (you can use it as a JPanel) 390 //with a legend at SOUTH 391Plot3DPanel Potenzial = **new** Plot3DPanel("SOUTH"); 392 Plot2DPanel Feldstärke= **new** Plot2DPanel(); 393 394GridLayout experimentLayout = new GridLayout(1,2);395 JFrame frame = **new** JFrame("Facharbeit"); 396 397 398 frame.setLayout(experimentLayout); 399 put the PlotPanel in a JFrame like a JPanel (Fenster) 400 401 frame.setSize(1000, 600); 402 403 frame.add(Potenzial); 404 frame.add(Feldstärke); 405

```
406
                                 frame.setVisible(true);
407
408
           // add grid plot to the PlotPanel (Legende)
409
        Potenzial.addGridPlot("Potenzialverlauf", x, y, werteplot);
410
411
                                412
                                // add a line plot to the PlotPanel
413
                                int M = (n_{v_1} + (n_{v_1}) + (n_{v_1}) + (n_{v_2}) + (n_{v_1}) + (n_{v_2} + (n_{v_1}) + (n_{v_2}) 
414
                                double [][] XYZx = new double [M][2];
415
                                double [][] XYZy = new double [M][2];
416
                                double [] [] dXYZ = new double [M] [2];
417
                                double [][] Punkte = new double [M] [2];
418
419
420
                                int n=0;
421
422
423
                                    for (i = 1; i < n_{y-1}; i++) {
424
                                             for (j = 1; j < n_{u, 1}; j++) {
425
                                                          if(i%s faktor==0 && j%s faktor==0){
426
                                                                      Punkte[n][0] = i;
427
                                                                      Punkte[n][1] = j;
428
                                                                      n = n+1;
429
                                                         }
430
                                             }
431
                                 }
432
433
                                int m=0;
434
                                    for (i = 1; i < n_punkte_y-1; i++) {
435
                                             for (j = 1; j < n_{ux}-1; j++)
436
                                                          if(i%s_faktor==0 && j%s_faktor==0){
437
438
                                                               XYZx[m][0] = m;
439
                                                               XYZx[m][1] = -(neuewerte[j+1][i]-
440
                                                                                                           -neuewerte[j-1][i])/2;
441
                                                               XYZy[m][0] = m;
442
                                                               XYZy[m][1] = -(neuewerte[j][i+1])
443
                                                                                                          -neuewerte [j][i-1])/2;
444
                                                               m = m+1;
445
                                                          }
446
                                             }
447
448
                                double längste = 0;
449
                                double kürzeste = 0;
450
                                 for (int k = 0; k < M; k++) {
451
```

452	double länge = Math.sqrt(Math.pow(XYZy[k][1], 2)
453	+ Math.pow(XYZx[k])
454	[1], 2));
455	$if (länge > längste) \{$
456	längste = länge;
457	}
458	$if (länge = 0) \{$
459	länge = l ängste/100;
460	}
461	}
462	
463	for $(int k = 0; k < M; k++)$ {
464	$dXYZ[k][0] = XYZy[k][1] / längste*3*s_faktor;$
465	$dXYZ[k][1] = XYZx[k][1] / längste*3*s_faktor;$
466	
467	}
468	
469	Feldstärke.addScatterPlot("Feldstärke", Punkte);
470	Feldstärke.addVectortoPlot(0, dXYZ);
471	
472	
473	
474	this.firePropertyChange("berechnungstatus", "Plotting",
475	"Fertig.");
476	
477	}
478	}

5.8 Programm und Facharbeit in digitaler Form (CD)

Literatur

- [1] BERLIOS: JMathTools. (Stand: 6. Januar 2010). http://jmathtools.berlios.de/doku.php?id=start
- [2] DEMTRÖDER, Professor Dr. W.: Experimentalphysik 2, Elektrizität und Optik. 4.
 Auflage. Springer, 2006. ISBN 3–540–33794–6
- [3] GRIFFITHS, David J.: Introduction to Electrodynamics. 3. Auflage. Prentic Hall, 1999. – ISBN 0–13–805326–X
- [4] MAXWELL, James C.: A Dynamical Theory of the Electromagnetic Field. In: *Philosophical Transactions of the Royal Society of London 155* (1864), S. 459–512
- [5] MESCHEDE, Professor Dr. D.: Gerthsen Physik. 23. Auflage. Springer, 2006. ISBN 3–540–25421–8
- [6] TIPLER, Paul A.: *Physik.* 23. Auflage. Spektrum, 1994. ISBN 3–86025–122–8.
 Übersetzung der 3. Ausgabe der amerikanischen Originalausgabe "Physics for Scientists and Engineers"
- [7] WIKIEPDIA: James Clerk Maxwell. (Stand: 16. Januar 2010). http://de.wikipedia.org/wiki/James_Clerk_Maxwell

Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

....., den

(Ort)

(Datum)

(Unterschrift des Schülers)