Finsterwalder-Gymnasium
Rosenheim

Kollegstufenjahr 1996/1998

Facharbeit
aus dem Fach
PHYSIK

Thema: Simulation der Bewegung von Alphateilchen beziehungsweise Elektronen im elektrischen Feld eines Atomkerns

Verfasser: Clemens Mühlberger

Leistungskurs: Physik
Kursleiter: OStR Udo Graf
Abgabetermin: 2. Februar 1998

Erzielte Note: ..... in Worten: .....
Erzielte Punkte: ..... in Worten: .....
(einfache Wertung)

.....
(Unterschrift des Kursleiters)

Inhaltsverzeichnis

[1] siehe [9]

[2] siehe [9]

[3] siehe [13] und [3]

Kapitel 1

Von der antiken Atomvorstellung bis zum „Rosinenkuchenmodell“

Im Laufe dieser Arbeit möchte ich die Versuche von Philip Lenard und Ernest Rutherford vorstellen, die die Grundlagen für die heutige Vorstellung des Atoms lieferten. Außerdem werde ich die Benutzung meines Computerprogramms erklären, das die bei oben genannten Versuchen auftretenden Bewegungen von geladenen Teilchen im elektrischen Feld eines Atomkerns simulieren soll. Doch bevor es soweit st, reflektiere ich einleitend kurz die Geschichte des Atoms von der griechischen Antike bis zum Beginn des 20. Jahrhunderts.

1.1 Die Anfänge des Atoms

So wurde der Grundstein für die moderne Atomphysik nicht erst im vorigen Jahrhundert, sondern schon in der Antike gelegt, denn etwa seit dem sechsten Jahrhundert vor Christi Geburt beschäftigte man sich mit dem Aufbau, den Bestandteilen und dem (un)endlichen Zerteilen von Materie. Als erster Vertreter des Atomismus ist uns Leukipp von Milet überliefert. Er entwickelte die Theorie, dass jede Substanz aus vielen kleinen, gleichartigen Teilchen samt Hohlräumen bestünde, da sonst ohne leere Räume ein Zerkleinern unmöglich wäre. Leukipp nannte diese Teilchen „Atome[1]“, weil sie selbst nicht mehr teilbar wären. Auch besäßen sie keine sinnlich wahrnehmbaren Eigenschaften, und nur ihre

[1] altgriechisch: $\alpha\tau o\mu o\varsigma$ atomos = unteilbar

Gestalt, Lage und Anordnung zueinander bestimmen die Erscheinungsformen der Dinge, so dass *„nur die Atome und der leere Raum“* [1] existieren. Sein Schüler Demokrit von Abdera ging noch einen Schritt weiter und vermutete, dass die planlose Bewegung der Atome im leeren Raum sowie deren Verbindungen und Trennungen untereinander, die Vorgänge in der Welt bezeichnen würden. Diese Vorgänge, zu denen nicht nur die Dynamik eines Körpers, sondern auch dessen Farbe, Form und Geschmack zählen, sollten nach Leukipp gesetzmäßig stattfinden:

> *„Nichts ereignet sich von selbst, sondern alles geschieht aus einem sinnvollen Grunde und unter dem Druck der Notwendigkeit.“*
>
> [2]

Selbstverständlich hatten Leukipp und Demokrit nicht nur damals sondern sogar bis vor etwa dreihundert Jahren sowohl Befürworter als auch Gegner ihrer Theorien von den unteilbaren Teilchen, den Atomen.

1.2 Thomsons Atommodell

Erst viele Jahrhunderte nach Leukipps und Demokrits Atomhypothesen entdeckte Sir Joseph John Thomson im Jahre 1897 das Elektron. Da sich im Nachhinein durch seine Messungen des Masse-zu-Ladung-Verhältnisses auch herausstellte, dass das Elektron ein Elementarteilchen, also ein Bestandteil des bis dahin unteilbaren Atoms sein mußte, formte er daraus 1907 das „plum pudding[2]“-Modell des Atombaus: Da das Atom im Grundzustand neutral, also weder positiv noch negativ geladen ist, das Elektron aber die einfache negative Elementarladung[3] e besitzt und Bestandteil des Atoms ist, benötigt man positive Ladung zum Ladungsausgleich. Deshalb sollte nach Thomsons Vorstellung das Atom eine Kugel sein, in der die positive Ladung homogen verteilt sei. Im Inneren dieser Kugel seien dann die Elektronen wie die Rosinen in einem Rosinenbrötchen nach einer bestimmten Gesetzmäßigkeit verstreut (vgl. [2] und [3]).

[2] englisch: plum pudding = Rosinenkuchen, Rosinenbrötchen

[3] verwendete Einheiten und Konstanten siehe Anhang C auf Seite 55 bzw. Anhang D auf Seite 56

Kapitel 2

Der Rutherfordsche Streuversuch

2.1 Der Lenardsche Versuch

Um das Atommodell von Thomson zu bewerten bzw. zu widerlegen, startete der in Neuseeland geborene Ernest Rutherford ein Experiment, dessen theoretischern Grundstock jedoch ein Versuch von Philip Lenard bildete: Ihm gelang es nämlich um die Jahrhundertwende, Kathodenstrahlen[1] in einer evakuierten Röhre durch ein sogenanntes „Lenard-Fenster[2]“ zu schicken und dahinter auf einem Leuchtschirm aufleuchten zu lassen (vgl. [1]). Der Umstand, daß Elektronen ab einer bestimmten kinetischen Energie die mehrere tausend Atomschichten dicke Folie durchdringen konnten, was nur mit einem Kern-Hülle-Modell des Atoms erklärbar wäre, ließ Lenard zu dem Ergebnis kommen, daß der größte Teil des Atomvolumens aus elektrischen Feldern bestünde, die durch die positiven und negativen Ladungen im Atom entstehen wurden. Lenard ahnte schon, was nun noch alles zu erwarten sei:

> *„. . . Da müssen diese Strahlen etwas außerordentlich Feines sein, so fein, daß der molekulare Bau der Materie, welcher den immerhin sehr feinen Lichtwellen gegenüber verschwindet, ihnen gegenüber sehr merklich wird. Natürlich wird es dann auch möglich sein können, mit Hilfe dieser Strahlen Auskünfze zu erhalten über die Beschaffenheit der Moleküle und Atome . . . “*
>
> [2]

[1] Elektronenstrahlen, die aus der Kathode von sogenannten Elektronenröhren emittieren.

[2] Eine dünne Metallfolie, die allerdings wegen dem großen Druck durch das Vakuum in der Röhre mit einem Gitter verstärkt wird. Lenard selbst verwendete $3 - 5\,\mu$m dicke Aluminiumfolien.

2.2 Der Versuch von Rutherford

Und Lenard sollte auch Recht behalten, denn nachdem Rutherford im Jahre 1907 von der McGill-Universität in Motreal (Kanada) an die Victoria-Universität Manchester (Großbritannien) gewechselt war, wollten er und seine Assistenten, der Labortechniker William Key und der deutsche Student Hans Geiger, sowie der zwei Jahre später dazustoßende neuseeländische Student Ernest Marsden eine auantitative Aussage über ein eventuelles atomares, positiv geladenes Massenzentrum machen, welches es nach Thomsons Atomvorstellung nciht gegeben hätte. Im Gegensatz zu Lenard verwendeten sie nicht Kathodenstrahlen, sondern die viel energiereicheren Alphastrahlen, die unter anderem durch den natürlichen Zerfall radioaktiver Stoffe entstehen können und aus zweifach positiv geladenen Heliumkernen bestehen (vgl. [3]).

2.2.1 Versuchsaufbau

Die Experimente liefen folgendermaßen ab: Die von dem radioaktiven Präperat **R** ausgehenden Alphateilchen (Abb. 2.1) wurden in einem evakuierten Gefäß **B** bis auf ein enges Strahlenbündel durch ein kleines Loch in einer Bleiblende **D** eliminiert. Dieses Strah-

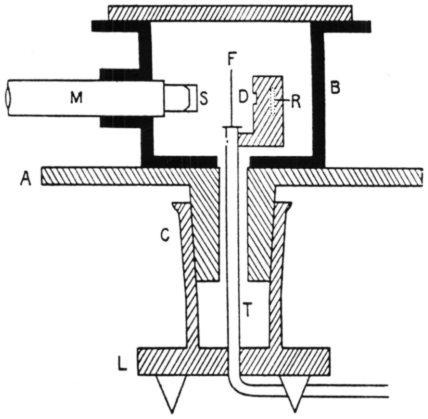


Abbildung 2.1: Versuchsaufbau (aus: [4])

lenbündel traf nun auf eine sehr dünne Metallfolie **F**, welche die Alphateilchen meistens ungehindert durchdrangen. Danach ließ man sie sich noch einige Zentimeter im Vakuum ausbreiten. Möglicherweise entstehende Reflexionen oder sonstige Erscheinungen sollten beobachtet und – falls möglich – gemessen und dann eventuelle Berechnungen hierzu angestrebt werden, um zu erklaren, wie, oder besser gesagt, warum das Experiment gerade

so und nicht anders verlaufen war. Also mußte man die Alphateilchen auf einem fluoreszierenden Schirm **S** auffangen, auf welchem ihre Szintillationen mit einem um das Präparat **R** drehbar angeordneten Mikroskop **M** beobachtet werden konnte. Dadurch, daß das Mikroskop **M** mit dem Schirm **S** verbunden war, konnte man für verschiedene Winkel die Szintillationen pro Zeiteinheit beobachten und so dann für jeden beliebigen Winkel das Verhältnis zwischen den um diesen Winkel abgelenkten Teilchen und der Gesamtzahl der Teilchen, die während dieser Zeiteinheit das Präparat **R** durch die Blende **D** verließen, feststellen (vgl. [5]).

2.2.2 Probleme und Materialwahl

Nach [6] mußten jedoch noch einige technische Schwierigkeiten gelöst, sowie die Wahl der verwendeten Materialien getroffen werden.

So konnte es gut möglich sein, daß einige Teilchen gar nicht abgelenkt werden. Darum muten die, die abgelenkt wurden, erfaßt und für diese eine Formel zur Beschreibung ihres Verhaltens gefunden werden. Zum Beobachten der Auftreffpunkte der Teilchen auf einem Zinksulfidschirm wurde ein Mikroskop in das fluoreszierende Material gesett, wobei man sich aber vor dem Zählvorgang einige Zeit in einem dunklen Raum aufhalten mußte, um seine Augen an die Dunkelheit zu gewöhnen. Erst danach konnte man die Einschläge pro Zeiteinheit an einer bestimmten Stelle unter einem bestimmten Winkel über mehrere Stunden hinweg mit dem Auge zählen.

Die Wahl des Streuteilchens fiel bei RUTHERFORD auf Alphateilchen, die nichts anderes als zweifach positiv geladene Heliumatome sind. Denn erstens tritt Alphastrahlung von Natur aus bei verschiedenen radioaktiven Elementen auf (RUTHERFORD verwendete Radium), und zweitens hatte RUTHERFORD mit Alphateilchen schon etwas Erfahrung gesammelt, da er früher einige Experimente mit ihnen durchgeführt hatte, um zum Beispiel die *„chemische Natur der Alphateilchen“* [3] zu demonstrieren. In diesem Zusammenhang hatte RUTHERFORD auch herausgefunden, daß Alphateilchen die elektrische Ladung $Q_\alpha = +2 \cdot e$ besitzen und in Luft (bei einer Temperatur von 15° C und einem Luftdruck von 1013 hPa) bei einer Energie von 4 bis 8 MeV eine mittlere Reichweite von 26 bis 73 mm haben. Ferner lassen sich Alphateilchen schon durch dünne Aluminiumfolien abschirmen, da sie nur eine mittlere Reichweite von ungefähr 2 bis 48 μm besitzen (vgl. [7]).

Weiter szintilliert Alphastrahlung bei fluoreszierenden Materialien, wie etwa Zinksulfid, blitzt und leuchtet also auf, weswegen Aufschläge von Alphateilchen mit dem menschlichen Auge erfaßbar sind. Erst 1911 konnte Charles Wilson mit seiner Nebelkammer die Bewegung von Alphateilchen direkt sichtbar machen, da sie im Dunst der Nebelkammer eine Spur hinterlassen, ähnlich den Kondensstreifen der Flugzeuge am Himmel. Ihre mittlere Geschwindigkeit beträgt nur 3 bis $7,5\,\%$ der Lichtgeschwindigkeit c oder ungefähr $1,5 \cdot 10^7\,\mathrm{ms}^{-1}$, weswegen nichtrelativistisch gerechnet werden kann. Dies wird aber erst später bei der Berechnung ihrer Flugbahn eine Rolle spielen.

Ein weiteres Problem war die Frage nach dem *„Target“* [3], dem zu „beschießenden“ Gegenstand. Rutherford verwendete sehr dünne Metallfolien, unter anderem Kupfer, Silber, Platin und Gold, mit einer Dicke von einigen hundert Nanometern. Denn erstens muß das Ziel dem ständigen Beschuß mit Alphastrahlen auch für längere Zeit standhalten und zweitens sollte die Wahrscheinlichkeit, daß ein Alphateilchen bei einem Streuversuch zweimal durch hintereinanderliegende Atomkerne gestreut wird, relativ gering und somit zu vernachlässigen sein (vgl. [5]).

2.2.3 Versuchsergebnis

Zunächst sollte Geiger die Szintillationen der Alphateilchen für kleine Streuwinkel beobachten, denn wie erwartet änderten die meisten Alphateilchen bei den Streuversuchen ihre ursprüngliche Flugbahn nur gering oder gar nicht, ähnlich wie die Elektronen bei Lenard. Nachdem Marsden aber im Jahre 1909 Ablenkungen mit Winkeln, die größer als 90° waren, beobachtete und seine Ergebnisse durch mehrere Versuche bestätigt wurden, war Rutherford sehr überrascht:

> *„Geiger kam in großer Aufregung zu mir und sagte: 'Es ist uns gelungen, nach rückwärts gehende α-Teilchen zu beobachten.' Das war wohl das Unglaublichste, was ich je erlebt hatte. Es war fast so unglaublich, als wenn eine Kugel auf einen zurückkäme, die man auf ein Stück Seidenpapier geschossen hat. Einiges Nachdenken brachte mir die Einsicht, daß diese Rückwärtsstreuung aber die Folge eines Zusammenstoßes sein mußte, und als ich Berechnungen machte, sah ich, daß es unmöglich war, irgend etwas in dieser Größenordnung zu bekommen, es sei denn, daß man ein System*

annahm, in dem der größte Teil der Masse des Atoms in einem einzigen Kern konzentriert war ... “

[8]

2.2.4 Rutherfords Atommodell

Somit war zwei Jahre nach Marsdens überraschender Entdeckung ein neues, verbessertes Atommodell entstanden: In der Mitte des Atoms befindet sich der Atomkern, von Rutherford *„Nucleus“* [3] genannt, der die positive Ladung $Q_{Kern} = +Z \cdot e$ (Z ist die Ordnungszahl des Atoms) und fast die gesamte Masse des Atoms innehat, da die Gesamtmasse der Elektronen eines Atoms zu klein ist, um ein einzelnes Alphateilchen derart abzulenken, wie es bei den Streuversuchen manchmal der Fall war. Dieser Kern, dessen Ausdehnung im Vergleich zu derjenigen des Atoms nur $\frac{1}{10000}$ beträgt und somit sehr gering ist, ist von einer homogenen, kugelsymmetrischen Ladungsverteilung $Q_{Umgebung} = -Z \cdot e$ umgeben. Sie besteht aus den Elektronen, die den Kern auf Kreisbahnen umrunden, ähnlich den Planeten, die die Sonne umkreisen. Deshalb wird dieses Atommodell in der Literatur häufig auch als „Planetenmodell“ bezeichnet. Nähert sich nun ein elektrisch geladenes Teilchen mit der Ladung $Q_{Teilchen}$ dem positiv geladenen Atomkern, so wirkt nach dem Coulomb-Gesetz die Kraft

$$F_{Coulomb} = \frac{1}{4 \cdot \pi \cdot \varepsilon_0} \cdot \frac{\mid Q_{Kern} \cdot Q_{Teilchen} \mid}{r^2} \tag{2.1}$$

[9]

auf das Teilchen, was zahlreiche Untersuchungen bestätigten (vgl. [8]). Mit dieser Formel ist es nun auch möglich, den maximal möglichen Radius r_{max} eines Atomkerns zu berechnen:

Da Alphateilchen bei einer Rückwärtsstreuung um den Winkel $\theta = 180°$ das Coulomb-Potential des Kerns

$$\varphi = \frac{1}{4 \cdot \pi \cdot \varepsilon_0} \cdot \frac{Q_{Kern}}{r_{max}} \tag{2.2}$$

[9]

durchlaufen, erhält man durch Anwenden der Definition des elektrischen Potentials φ für die Punktladung $Q_{Teilchen}$ mit der kinetischen Energie $E_{kin\alpha}$ und nach Aufösen von

Gleichung 2.2 nach r_{max}

$$r_{max} = \frac{1}{4 \cdot \pi \cdot \varepsilon_0} \cdot \frac{Q_{Kern} \cdot Q_{Teilchen}}{E_{kin\alpha}}. \tag{2.3}$$

Für Alphateilchen mit einer kinetischen Energie $E_{kin\alpha} = 8,0\,\mathrm{MeV}$ und einem Goldkern mit der Kernladung $Q_{Kern} = 79 \cdot e$ als Streuzentrum folgt zum Beispiel

$$\begin{aligned} r_{max} &= \frac{1}{4 \cdot \pi \cdot \varepsilon_0} \cdot \frac{2 \cdot 79 \cdot e^2}{E_{kin\alpha}} = \\ &= \frac{1\,\mathrm{Vm}}{4 \cdot \pi \cdot 8,8542 \cdot 10^{-12}\,\mathrm{C}} \cdot \frac{2 \cdot 79 \cdot (1,6022 \cdot 10^{-19}\,\mathrm{C})^2}{8,0 \cdot 10^6 \cdot 1,60 \cdot 10^{-19}\,\mathrm{J}} \\ &\approx 2,85 \cdot 10^{-14}\,\mathrm{m} \end{aligned}$$

(vgl. [1])

Der Atomradius für Gold beträgt also höchstens $r_{Au-Kern} \leq r_{max} = 2,85 \cdot 10^{-14}\,\mathrm{m}$.

Kapitel 3

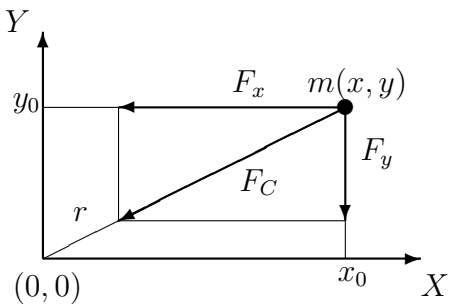
Das Computerprogramm

3.1 Behandlung des physikalischen Kerns

Da die innere Struktur des Programms STREUUNG.PAS und der Unit STREUNIT.PAS durch die eingefügten Kommentarzeilen in Anhang B bzw. in Anhang A ausreichend erklärt sein dürften, erläutere ich in diesem Abschnitt die Benutzung des Computerprogramms. Doch zuvor möchte ich kurz den physikalischen Kern des Programms veranschaulichen, der im Wesentlichen aus der Berechnung der Ortspunkte des Streuteilchens besteht.

3.1.1 Koordinatenberechnung des Streuteilchens im x-y-System

Die Koordinaten (x, y) eines bewegten Streuteilchens mit der Masse m erhält man, wenn man davon ausgeht, daß in einem rechtwinkligen Koordinatensystem (Abb. 3.1.1) von dem Teilchen die vom Abstand r abhängige Kraft $F_{Coulomb}$ (siehe Gleichung 2.1) auf das Streuzentrum wirkt.



(Abb. 3.1.1 nach: [10])

Dieses Zentrum befindet sich im Ursprung und entspricht bei den Simulationen dem Atomkern, dessen Masse größer als die des Streuteilchens ist. Da nach dem berühmten Satz des PYTHAGORAS für r die Gleichung

$$r = \sqrt{x^2 + y^2} \tag{3.1}$$

[11]

gilt, kann man nun $F_{Coulomb}$ in die beiden zu denjeweiligen Koordinatenachsen parallelen Komponenten $F - x$ (in x-Richtung) und F_y (in y-Richtung) auflösen, da ferner nach dem Strahlensatz die Proportionen

$$\frac{F_x}{x} = \frac{F_{Coulomb}}{r} \quad \text{und} \tag{3.2}$$

$$\frac{F_y}{y} = \frac{F_{Coulomb}}{r} \tag{3.3}$$

(vgl. [11])

gelten, aus denen man dann durch Umformen F_x bzw. F_y erhält:

$$F_x = F_{Coulomb} \cdot \frac{x}{r} \,, \tag{3.4}$$

$$F_y = F_{Coulomb} \cdot \frac{y}{r} \tag{3.5}$$

Nach dem Grundgesetz der NEWTONSCHEN Mechanik $\vec{F} = m \cdot \vec{a}$ (siehe [9]) kann man nun die ebenfalls senkrecht aufeinander stehenden Beschleunigungskomponenten a_x und a_y des Teilchens mit der Masse m freistellen und bekommt:

$$a_x = F_{Coulomb} \cdot \frac{x}{m \cdot r} \quad \text{bzw.} \tag{3.6}$$

$$a_y = F_{Coulomb} \cdot \frac{y}{m \cdot r} \tag{3.7}$$

(vgl. [10])

Da nach [9] die Geschwindigkeit $v = v_0 + at$ ist, lassen sich die Geschwindigkeitskomponenten v_x und v_y durch die Gleichungen 3.6 und 3.7 für den jeweiligen Zeitabschnitt Δt berechnen:

$$v_x = v_{x_0} + F_{Coulomb} \cdot \frac{x}{m \cdot r} \cdot \Delta t \,, \tag{3.8}$$

$$v_y = v_{y_0} + F_{Coulomb} \cdot \frac{y}{m \cdot r} \cdot \Delta t \,. \tag{3.9}$$

Hierbei muß man wissen, daß v_{x_0} und v_{y_0} nichts anderes als die Geschwindigkeitskomponenten v_x und v_y zum Zeitpunkt $t = t - \Delta t$ sind. Da nach der Formel für die Geschwindigkeit $v = \Delta x \backslash \Delta t$ die Streckenänderung $\Delta x = v \cdot \Delta t$ [9] ist und der neue Koordinatenpunkt

durch die Addition des alten Punktes mit der jeweiligen Ortsänderung errechnet werden kann, folgen daraus schließlich die beiden Formeln zur Berechnung der neuen Koordinaten des Streuteilchens nach Verstreichen der Zeit Δt:

$$x = x_0 + v_x \cdot \Delta t \quad \text{und} \tag{3.10}$$

$$y = y_0 + v_y \cdot \Delta t \; . \tag{3.11}$$

Dabei ist aber später im Computerprogramm auf ein angemessenes Δt zu achten, da man zwar für einen zu klein gewählten Zeitraum Δt eine sehr genaue Teilchenbahn am Monitor erhält, dafür dauert es aber auch dementsprechend lange, bis diese Bahn vollständig gezeichnet wird. Dies wäre nicht unbedingt von Vorteil für den Anwender. Andererseits kann man für einen zu groß gewählten Zeitabschnitt Δt zu völlig falschen Bahnen gelangen, da die Berechnung der Punkte zu ungenau wird, was man im Grund auch nicht will. Aber durch das Experimentieren mit einigen Zahlenwerten kann man zu einem zufriedenstellenden Resultat für Δt gelangen.

3.1.2 Abbildung der x-y-z-Koordinaten auf dem Bildschirm

Die eigentliche Berechnung der Koordinaten in einem zweidimensionalen System, wie man es beispielsweise vorfindet, wenn man die Flugbahnen der Streuteilchen aus der Vogelperspektive betrachtet, wurde in 3.1.1 schon veranschaulicht. Jedoch sollte der Benutzer auch die Möglichkeit haben, sich ein dreidimensionales, realitätsnäheres Bild von den Teilchenbewegungen anzuschauen. Doch bei der Transformation der 2D-Koordinaten aus 3.1.1 zeigten sich einige Schwierigkeiten, aufgrund derer ich nicht mit absoluter Gewissheit sagen kann, ob die Flugbahnen durch das Compterprogramm räumlich einwandfrei dargestellt werden.

Da die Schrägansicht des Computerprogramms nichts anderes als eine Drehung des gesamten Systems um die z-Achse und dann um die x-Achse bedeutet, gelten hierfür folgende Regeln:

für die Rotation um die z-Achse (vgl. [12]):

$$\begin{aligned} x &= x \cdot \cos\alpha + y \cdot \sin\alpha \\ y &= y \cdot \cos\alpha - x \cdot \sin\alpha \\ z &= z \end{aligned} \tag{3.12}$$

für die Rotation um die x-Achse (vgl. [12]):

$$\begin{aligned} x &= x \\ y &= y \cdot \cos\alpha + z \cdot \sin\alpha \\ z &= z \cdot \cos\alpha - y \cdot \sin\alpha \end{aligned} \tag{3.13}$$

Da ich in der mir zur Verfügung stehenden Literatur keine Angaben zum Verbinden der Gleichungen 3.12 und 3.13 gefunden habe, fügte ich diese schließlich zu

```
posx := x*cos(dreh) - y*sin(dreh)
posy := y*cos(dreh) + x*sin(dreh)
posz := posx*cos(dreh) + z*sin(dreh)
```

(vgl. Quellcode STREUUNG.PAS, Anhang B, S. 27)

zusammen, wobei `dreh` den Winkel α im Bogenmaß darstellt und `posy` als x-Wert und `posz` aly y-Wert der Bildschirmkoordinaten bestimmt wurden.

3.2 Erläuterung des Programms STREUUNG.EXE

Das in der Programmiersprache Turbo Pascal[1] Version 7.0 geschriebene Computerprogramm STREUUNG.EXE führt nach dessen Start eine Fehlerüberprüfung durch, so daß es beim Mangel einer *VGA*[2]-Grafikkarte oder einer Maus mit einer Meldung abbricht. Hat es aber, was der Normalfall sein sollte, fehlerfrei gestartet, so findet man als Benutzer neben einem quadratischen Fenster, in dessen Mitte sich der Atomkern befindet und in welchem die Flugbahnen der Teilchen gezeichnet werden, auch noch acht verschiedene Befehlsfelder – das Hauptmenü – vor. Aus ihm kann man eine Option mit einem Klick auf die linke Maustaste auswählen, sobald man sich mit dem Mauszeiger, einem symbolisierten Radioaktiv-Gefahrenzeichen, darüber befindet. Im Einzelnen möchte ich nun diese acht Optionen des Hauptmenüs erläutern.

[1] Turbo Pascal® ist eingetragenes Warenzeichen von Borland International

[2] *V*ideo *G*raphics *A*rray (hochauflösender Bildschirmmodus)

3.2.1 Die Option `Start`

Hat man als Benutzer das Befehlsfeld `Start` angeklickt, so wird die Flugbahn des eingestellten Streuteilchens aus den aktuellen Parametern, die man am linken Bildschirmrand unter der Überschrift `Ausgangswerte` ablesen kann, berechnet und gezeichnet. Die unter `Ausgangswerte` stehenden Daten beziehen sich immer auf die nächste zu zeichnende Flugbahn des Streuteilchens. Dabei werden die Linien in bis zu zehn verschiedenen Farben gezeichnet, um eine Unterscheidung zu erleichtern. Man kann so viele Linien zeichnen lassen, wie man will, jedoch legt das Programm maximal fünfzig im Speicher ab. Wird diese Höchstzahl überschritten, so wird der Speicher, mit Linie 1 beginnend, neu belegt, so daß die alten Linien verloren gehen.

3.2.2 Die Option `Ansicht`

Mit dem Menüpunkt `Ansicht` kann man zwischen einer Schrägansicht (`schräg`) und einer Draufsicht (`oben`) wählen, wobei die ursprüngliche und die neu ausgewählte Perspektive im `Ansicht`-Fenster abgelesen werden können. Drückt man nun nach der getroffenen Wahl `Okay`, so werden die bis zu maximal fünfzig Linien unter der neu eingestellten Perspektive gezeichnet. Mit `Abbruch` kehrt man ohne Veränderung zurück auf den Hauptbildschirm.

3.2.3 Die Option `Faktoren`

Unter `Faktoren` kann man die Anfangsbedingungen der Simulation für das Streuteilchen festlegen. Hierbei muß man die jeweiligen Grenzwerte, die rechts neben den Befehlsfeldern stehen, sowie die erlaubten Zeichen zur Eingabe der Werte beachten. Drückt man während der Eingabe die ESCAPE-Taste oder wird ein Wert falsch eingegeben, so kehrt man ins `Faktoren`-Fenster zurück und der alte Wert bleibt bestehen. So läßt sich mit dem Knopf `vx` die Geschwindigkeit in Prozent zur maximalen Teilchengeschwindigkeit in x-Richtung eingeben, woraus auch automatisch die kinetische Energie E_{kin} in MeV errechnet wird. Jedoch läßt sich mit dem Befehlsfeld `Ekin` im `Faktoren`-Fenster diese auch direkt eingeben, so daß nun v_x berechnet wird. Weiter kann man mit der Option `delay` die Verzögerung beim Zeichnen der Flugbahn eingeben. Sie erlaubt es dem Benutzer, das Zeichnen der Teilchenbahn in aller Ruhe mitzuverfolgen. Die Einstellung des Stoßparame-

ters p kann man auf zwei Arten erledigen: Klickt man das Feld `p(Zahl)` an, so kann man für den Abstand p des Teilchens zur Achse des Atomkerns eine bis auf drei Dezimalstellen genaue Zahl eingeben. Anders hingegen bei der Option `p(Maus)`: Hier kann der Anwender mit dem Mauszeiger am unteren Bildschirmrand den Startpunkt zwischen dem linken und dem rechten Rahmen des Zeichenfensters frei verschieben. Drückt man nun die linke Maustaste, so wird die aktuelle Position des Mauszeigers als neuer Startpunkt gesetzt. Drückt man hingegen die rechte Maustaste, so bleibt der ursprüngliche Ausgangspunkt bestehen und man kehrt ohne Veränderung ins `Faktoren`-Fenster zurück. Die Option `p(Maus)` funktioniert leider nur in der Draufsicht. Mit `Okay` werden die neu eingestellten Werte als neue Ausgangswerte gesetzt und man landet wieder im Hauptmenü.

3.2.4 Die Option `Partikel`

Die Option `Partikel` ermöglicht es dem Benutzer, das Streuzentrum und das Streuteilchen auszuwählen. Als Streuzentrum stehen vier verschiedene Atomkerne und als Streuteilchen Alphateilchen und Elektronen zur Verfügung. Mit `Okay` werden die neuen Streukörper übernommen, wobei das Zeichenfenster und vorangegangene Linien gelöscht werden. Mit `Abbruch` erreicht man ohne Veränderung wieder den Hauptbildschirm.

3.2.5 Die Option `Info`

Unter dem Befehlsfeld `Info` erhält der Benutzer eine kurze Information zum Computerprogramm. Mit `Okay` kehrt man wieder zurück zum Hauptfenster.

3.2.6 Die Option `Hilfe`

Klickt man das Feld `Hilfe` an, so wird das `Hilfe`-Fenster geöffnet, in welchem die einzelnen Optionen kurz erläutert werden. Dies ist als kleine Hilfestellung für den Benutzer gedacht. Mit `Okay` erreicht man wieder die Hauptebene des Programms.

3.2.7 Die Option `Datei`

Durch diese Option kann der Benutzer seine simulierten Flugbahnen entweder mit `Sichern` in bis zu neun verschiedenen Dateien abspeichern oder mit `Laden` zuvor gespeicherte Li-

nien wieder in das Zeichenfenster zeichnen lassen. Dabei sollte aber beachtet werden, daß mit `Sichern` schon vorhandene Dateien ohne Rücksicht überschrieben werden. Will der Anwender aber eine nicht existierende oder kaputte Datei öffnen, so erhält er eine Fehlermeldung, die er mit `Okay` bestätigen muß. Mit dem Befehlsfeld `Zurück` gelangt man hier übrigens immer zurück zum Hauptbildschirm.

3.2.8 Die Option `Beenden`

Will der Benutzer das Programm beenden, so klickt er im `Beenden`-Fenster auf den `Ja`-Knopf. Klickt er hingegen `Nein` an, so kommt er wieder zum Hauptfenster.

Kapitel 4

Anwendungsmöglichkeiten

Durch die benutzerfreundliche Oberfläche und die fast selbsterklärende Programmstruktur könnte man dieses Computerprogramm schin in der Mittelstufe einsetzen, um zum Beispiel das KEPLERSCHE Gesetz bzw. die KEPLER-Ellipsen anhand der Elektronenbahnen seinen Einsatz finden, wenn man etwa das COULOMB-Gesetz oder den RUTHERFORDSCHEN Streuversuch plastisch am Computer zeigen will. Man kann das Programm aber auch dazu verwenden, um sich zeigen zu lassen, daß schon sehr geringe Änderungen ein völlig anderes Ergebnis, eine völlig andere Flugbahn ergeben können. Außerdem müßte das Programm auf jedem handelsüblichen Personalcomputer mit mathematischem Coprozessor, *VGA*-Grafikkarte und einer Maus lauffähig sein.
Abschließend möchte ich noch darauf hinweisen, daß das Computerprogramm trotz ausgiebiger Überprüfungen und Testläufe keinen Anspruch auf absolute Richtigkeit erheben kann. Ich fand nämlich mit den mir vorhandenen Mitteln keine Hinweise darauf, daß erstens die manchmal sonderbar anmutenden Flugbahnen der Elektronen bestätigt worden wären oder zweitens die Schrägansicht, so wie sie jetzt berechnet wird, völlig korrekt wäre.

Anhang A

STREUNIT.PAS

(Die Unit „StreUnit.PAS[1]“ lag damals als Anlage 1 auf Diskette bei, nun unter *http://www.emerentia.de/Studium/Facharbeit/Facharbeit.html* downloadbar.)

```
Unit StreUnit;                                              {Name der Unit}

INTERFACE

uses crt,dos,graph;                {Bindet die drei Turbo Pascal Units ein}

const                                                 {Konstanten-Vereinbarung}
   virtmaxx=640;
   virtmaxy=480;                       {Grenzen für virtuellen Bildbereich}

   x0:word=0;
   y0:word=0;                          {Koordinaten der linken oberen Ecke}

   xb:word=164;                          {Konstante für das Zeichenfenster}
   CenterX=400;
   CenterY=234;                                  {Koordinaten des Zentrums}

   KMax=18;                                      {Maximale Zahl an Knöpfen}

   Ziffern:set of char=[#43..#46,#48..#57];
                             {erlaubte Zeichen für die Eingabe von Zahlen}

   ScrMask:array [1..16] of word=
        ($F81F,$E007,$C003,$8001,$8001,$0000,$0000,$0000,
         $0000,$0000,$0000,$8001,$8001,$C003,$E007,$F81F);
                                              {Bitfeld 1 für den Mauszeiger}
```

[1] Verwedete Literatur hierfür: [14], [15], [16] und [17].

```
   CrsMask:array [1..16] of word=
        ($0000,$07E0,$07E0,$03C0,$03C0,$0180,$0180,$0240,
         $7E7E,$7C3E,$7C3E,$381C,$381C,$1008,$0000,$0000);
                                          {Bitfeld 2 für den Mauszeiger}
{Bitfeld 1 wird mit dem Bildschirminhalt mit AND, Bitfeld 2 wird mit OR
verknüpft, so daß der Mauszeiger das Aussehen eines Radioaktiv-
Gefahrenzeichens erhält.}

type                                                  {Typen-Vereinbarung}
   s8=string[8];                      {Ein String mit höchstens 8 Zeichen}
   KKoord=record                                 {Variablen eines Knopfes}
   Ktext:s8;                                            {Text des Knopfes}
   Kx1,Ky1,Kx2,Ky2:integer;                     {Koordinaten des Knopfes}
   KStatus:boolean;                    {Knopf gedrückt ⇒ Kstatus = true}
end;

var                                                  {Vaiablendeklaration}
   grDrv,grMode:integer;                  {für Grafik-Treiber und -Modus}
   Path,Pfad:string[32];                    {für Pfad des Grafiktreibers}
   FehlerNummer:integer;                                {für Graphresult}
   maxx,maxy:word;                  {maximale x-, y-Werte im Grafikmodus}
   xx,yy,bs,bb,tt:integer;                              {für die Mausabfrage}
   Ok:boolean;                                           {für Open-Datei}
   MOkay:wordbool;                                            {für Minit}
   Knopf:array [1..KMax] of KKoord;              {Variable für die Knöpfe}

function FExist(Dateiname: String):boolean;                {Ist Datei da?}

{Es folgen Prozeduren zur Maussteuerung, die aus Geschwindigkeitsgründen
in Assembler geschrieben wurden:}
function MInit:wordbool;                          {Ist eine Mouse vorhanden?}
procedure MOn;                                             {Mouse-Zeiger an}
procedure MOff;                                           {Mouse-Zeiger aus}
procedure GetMxyb;                          {Ort und Knopf der Maus abfragen}
procedure SetMxy(y,x:integer);               {Setzt Mauszeiger an x/y-Stelle}
procedure MbPressed(b:integer);                  {Prüft, ob Taste b gedrückt}
procedure MbReleased(b:integer);              {Prüft, ob Taste b losgelassen}
procedure MlimitX(Xlow,Xhigh:integer);               {Setzt Limit horizontal}
procedure MlimitY(Ylow,Yhigh:integer);                 {Setzt Limit vertikal}
procedure SetMGraphC(x,y:integer;var mask);               {Aussehen der Maus}
{Ende der Prozeduren zur Maussteuerung}

function nts(zahl:real):string;              {Eine Real-Zahl in einen String}
function ntsn(zahl0:integer):string;     {Eine Integer-Zahl in einen String}
procedure waitkey;                                  {Wartet auf Tastendruck}
function mitte(x1,x2:integer):integer;         {Berechnet Mitte zweier Werte}
function errorchk:boolean;                  {Prüft auf Graphik- & Mausfehler}
```

```
procedure KInit(z:byte;s:s8;stat:boolean;x1,y1,x2,y2:integer);
                                                {Initialisiert die Knöpfe}

IMPLEMENTATION

function FExist(Dateiname:String):boolean;
var Dummy:SearchRec;                                          {für FindFirst}
begin
   FindFirst(Dateiname,0,Dummy);     {Sucht nach erster vorhandener Datei}
   FExist:=DosError=0;                  {Fehlernummer an FExist übergeben}
end;

function MInit:wordbool;                          {Ist eine Mouse vorhanden?}
begin                              {Wenn MOkay = 0 ⇒ Mouse nicht vorhanden}
   ASM
     XOR ax,ax                                  {ax-Register auf null setzen}
     INT 33h                                     {DOS-Interrupt 33h aufrufen}
     MOV MOkay,ax                  {Rückmeldung des ax-Registers nach MOkay}
     MOV bs,bx                        {Rückmeldung des bx-Registers nach bs}
   end;                            {Ende der Built-In-Assembler-Anweisungen}
   MInit:=MOkay;                 {ax-Register bzw. Variable MOkay übergeben}
end;                                        {Ende der begin-end-Anweisungen}

procedure MOn;Assembler;                                      {Mouse-Zeiger an}
ASM
   MOV ax,01h                                   {ax-Register mit 01h belegen}
   INT 33h                                      {DOS-Interrupt 33h aufrufen}
end;                               {Ende der Built-In-Assembler-Anweisungen}

procedure MOff;Assembler;                                     {Mouse-Zeiger aus}
ASM
   MOV ax,02h                                   {ax-Register mit 02h belegen}
   INT 33h                                      {DOS-Interrupt 33h aufrufen}
end;                               {Ende der Built-In-Assembler-Anweisungen}

procedure GetMxyb;Assembler;               {Ort und Knopf der Maus abfragen}
ASM
   MOV ax,03h                                   {ax-Register mit 03h belegen}
   INT 33h                                      {DOS-Interrupt 33h aufrufen}
   MOV bb,bx       {bb=0: linker Knopf (1=an); bb=1: rechter Knopf (1=an)}
   MOV xx,cx                                        {x-Wert des Mauszeigers}
   MOV yy,dx                                        {y-Wert des Mauszeigers}
end;                               {Ende der Built-In-Assembler-Anweisungen}

procedure SetMxy(y,x:integer);Assembler;
ASM                                          {Setzt den Mousecursor an (x;y)}
   CMP x,virtmaxx                              {Vergleiche x mit virtmaxx}
```

```
   JB @ycomp                              {Ist x<virtmaxx, dann nach @ycomp,}
   MOV x,virtmaxx                             {sonst x = virtmaxx setzen ...}
   JMP @ycomp                                  {... und nach @ycomp springen}
@ycomp:                                              {Die Sprungmarke @ycomp}
   CMP y,virtmaxy                                {Vergleiche y mit virtmaxy}
   JB @Ende                               {Ist y<virtmaxy, dann nach @Ende,}
   MOV y,virtmaxy                             {sonst y = virtmaxy setzen ...}
   JMP @Ende                                   { ... und nach @Ende springen}
@Ende:                                                {Die Sprungmarke @Ende}
   MOV ax,04h                                  {ax-Register mit 04h belegen}
   MOV cx,y                                      {cx-Register mit y belegen}
   MOV dx,x                                      {dx-Register mit x belegen}
   INT 33h                                     {DOS-Interrupt 33h aufrufen}
end;                                 {Ende der Built-In-Assembler-Anweisungen}

procedure MbPressed(b:integer);Assembler;
ASM                                             {Prüft, ob Taste b gedrückt}
   MOV ax,05h                                  {ax-Register mit 05h belegen}
   MOV bx,b                                      {bx-Register mit b belegen}
   INT 33h                                     {DOS-Interrupt 33h aufrufen}
   MOV bs,ax                                 {Variable bs mit ax-Wert belegen}
   MOV tt,bx                                 {Variable tt mit bx-Wert belegen}
   MOV xx,cx                                 {Variable xx mit cx-Wert belegen}
   MOV yy,dx                                 {Variable yy mit dx-Wert belegen}
end;                                 {Ende der Built-In-Assembler-Anweisungen}

procedure MbReleased(b:integer);Assembler;
ASM                                           {Prüft, ob Taste b losgelassen}
   MOV ax,06h                                    {ax-Register mit 06h belegen}
   MOV bx,b                                        {bx-Register mit b belegen}
   INT 33h                                     {DOS-Interrupt 33h aufrufen}
   MOV bs,ax                                 {Variable bs mit ax-Wert belegen}
   MOV tt,bx                                 {Variable tt mit bx-Wert belegen}
   MOV xx,cx                                 {Variable xx mit cx-Wert belegen}
   MOV yy,dx                                 {Variable yy mit dx-Wert belegen}
end;                                 {Ende der Built-In-Assembler-Anweisungen}

procedure MlimitX(xlow,xhigh:integer);Assembler;
ASM                     {Limit (xlow-xhigh) für Maus in x-Richtung setzen}
   CMP xlow,0                                          {Vergleiche xlow mit 0}
   JA @xhighcomp                        {Wenn xlow>0, dann nach @xhighcomp}
   MOV xlow,0                                          {Vergleiche xlow mit 0}
   JMP @xhighcomp                                {Nach @xhighcomp springen}
@xhighcomp:                                        {Die Sprungmarke @xhighcomp}
   CMP xhigh,virtmaxx                          {Vergleiche xhigh mit virtmaxx}
   JB @Ende                                    {Wenn xhigh<0, dann nach @Ende}
   MOV xhigh,virtmaxx                                 {Setze xhigh = virtmaxx}
```

```
   JMP @Ende                                          {Nach @Ende springen}
@Ende:                                               {Die Sprungmarke @Ende}
   MOV ax,07h                                    {ax-Register mit 07h belegen}
   MOV cx,xlow                                  {cx-Register mit xlow belegen}
   MOV dx,xhigh                                {dx-Register mit xhigh belegen}
   INT 33h                                        {DOS-Interrupt 33h aufrufen}
end;                                 {Ende der Built-In-Assembler-Anweisungen}

procedure MlimitY(ylow,yhigh:integer);Assembler;
ASM                        {Limit (ylow-yhigh) für Maus in y-Richtung setzen}
   CMP ylow,0                                         {Vergleiche ylow mit 0}
   JA @yhighcomp                          {Wenn ylow>0, dann nach @yhighcomp}
   MOV ylow,0                                         {Vergleiche ylow mit 0}
   JMP @yhighcomp                                   {Nach @yhighcomp springen}
@yhighcomp:                                        {Die Sprungmarke @yhighcomp}
   CMP yhigh,virtmaxy                            {Vergleiche yhigh mit virtmaxy}
   JB @Ende                                      {Wenn yhigh<0, dann nach @Ende}
   MOV yhigh,virtmaxy                                {Setze yhigh = virtmaxy}
   JMP @Ende                                          {Nach @Ende springen}
@Ende:                                               {Die Sprungmarke @Ende}
   MOV ax,09h                                    {ax-Register mit 08h belegen}
   MOV cx,ylow                                  {cx-Register mit ylow belegen}
   MOV dx,yhigh                                {dx-Register mit yhigh belegen}
   INT 33h                                        {DOS-Interrupt 33h aufrufen}
end;                                 {Ende der Built-In-Assembler-Anweisungen}

procedure SetMGraphC(x,y:integer;var mask);Assembler;
ASM                       {(x;y) für HotSpot; mask für Aussehen (Bitfelder!)}
   MOV ax,09h                                    {ax-Register mit 09h belegen}
   MOV bx,x                                        {bx-Register mit x belegen}
   MOV cx,y                                        {cx-Register mit y belegen}
   LES dx,mask                       {Lädt Adresszeiger von mask nach es:dx}
   INT 33h                                        {DOS-Interrupt 33h aufrufen}
end;                                 {Ende der Built-In-Assembler-Anweisungen}

function nts(zahl:real):string;                          {Real-Zahl in String}
var helpstr:string;
begin
   str(zahl:1:3,helpstr);
   nts:=helpstr;
end;

function ntsn(zahl0:integer):string;                  {Integer-Zahl in String}
var helpstr:string;
begin
   str(zahl0:2,helpstr);
   ntsn:=helpstr;
```

```
end;

procedure waitkey;                                    {Wartet auf Tastendruck}
var ch:char;
begin
   while keypressed do ch:=readkey;
   ch:=readkey;
end;

function Mitte(x1,x2:integer):integer;        {Berechnet die Mitte zweier}
                                     {Zahlen für Textausgabe mit outtextxy}
begin
   Mitte:=x1+(x2-x1) DIV 2;
end;

function errorchk:boolean;              {Prüft auf Graphik- und Mausfehler}
begin
   GetDir(0,path);
   if length(path)>=4 then Pfad:=path+'\*.BGI'
   else Pfad:=path+'*.BGI';
   if (fExist(pfad))=false then             {Ist Grafiktreiber in 'pfad'?}
   begin                                  {Sonst Fehlermeldung und Beenden}
     clrscr;
     writeln('Konnte Grafiktreiber im aktuellen Pfad "',
             +path,'" nicht finden!');
     Halt;
   end;
   grDrv:=detect;
   InitGraph(grDrv,grMode,Path);           {ohne Fehler ⇒ Treiber laden,}
   FehlerNummer:=GraphResult;
   if FehlerNummer<>0 then                          {sonst Meldung und Ende}
     begin
       closegraph;
       clrscr;
       writeln('ACHTUNG! Es ist der Grafikfehler Nr. ',
               +abs(FehlerNummer),' aufgetreten:');
       case abs(FehlerNummer) of
         1:writeln(' - Der Grafiktreiber "*.BGI" wurde '
                   'nicht geladen!');
         2:writeln(' - Es ist kein grafikfähiger Adapter '
                   'vorhanden!');
         3:writeln(' - Die Grafiktreiber-Datei konnte nicht '
                   'gefunden werden!');
         4:writeln(' - Das Grafiktreiber-Programm ist defekt!');
         5:writeln(' - Es ist nicht genug Platz im Hauptspeicher '
                   'für den Grafiktreiber vorhanden!');
         6:writeln(' - Es ist nicht genug Speicher vorhanden!');
```

```
        7:writeln(' - Es ist nicht genug Speicher für die '
                  'Anweisung "FloodFill" vorhanden!');
        8:writeln(' - Die Schrift-Datei "*.CHR" konnte '
                  'nicht gefunden werden!');
        9:writeln(' - Es ist nicht genug Platz im Speicher '
                  'für die Schrift vorhanden!');
        10:writeln(' - Der angegebene Grafikmodus wird '
                  'nicht vom Treiber unterstützt!');
        11:writeln(' - Ein nicht näher klassifizierbarer '
                  'Fehler ist aufgetreten!');
        12:writeln(' - Ein Ein- oder Ausgabefehler ist beim '
                  'Laden von Dateien aufgetreten!');
        13:writeln(' - Die angegebene Schrift-Datei ist '
                  'leider ungültig!');
        14:writeln(' - Die Kennziffer für den Zeichensatz '
                  'ist nicht definiert worden!');
      end;
      Halt(FehlerNummer);
    end
    else
      begin
        maxx:=getmaxx;
        maxy:=getmaxy;
      end;
    if MInit then
      begin                          {Prüfen, ob Maus/-treiber vorhanden,}
        SetMGraphC(8,8,ScrMask);          {Zeiger mit HotSpot bei (8;8)}
        MLimitX(0,maxx-2);                        {Limit in x-Richtung}
        MLimitY(0,maxy-2);                        {Limit in y-Richtung}
        SetMxy(83,60);        {Setzt Mauszeiger auf Koordinaten (85,62)}
        errorchk:=true;
      end
    else                                          {sonst Fehlermeldung}
      begin
        Settextstyle(DefaultFont,HorizDir,1);
        SetTextJustify(CenterText,CenterText);
        outtextxy(Mitte(0,Maxx),30,'Leider ist keine Maus vorhanden.');
        outtextxy(Mitte(0,Maxx),70,'Sie können das Programm nur '
                                   +'mit einer Maus benützen!');
        outtextxy(Mitte(0,Maxx),110,'Zum Beenden beliebige Taste '
                                    +'drücken!');
        waitkey;                                  {Auf Tastendruck warten}
        cleardevice;                                  {Bildschirm löschen}
        errorchk:=false;
      end;
end;
```

```
procedure KInit(z:byte;s:s8;stat:boolean;x1,y1,x2,y2:integer);
begin
   with Knopf[z] do                                    {Initialisiert Knopf z}
     begin
       Ktext:=s;
       Kstatus:=stat;
       Kx1:=x1;
       Ky1:=y1;
       Kx2:=x2;
       Ky2:=y2;
     end;
end;

begin
   if errorchk<>true then Halt;                {Wenn Fehler, dann Abbrechen}
   KInit(1,'Start',false,6,6,80,25);                        {'Start'-Knopf}
   KInit(2,'Ansicht',false,86,6,160,25);                  {'Ansicht'-Knopf}
   KInit(3,'Faktoren',false,6,36,80,55);                 {'Faktoren'-Knopf}
   KInit(4,'Partikel',false,86,36,160,55);               {'Partikel'-Knopf}
   KInit(5,'Info',false,6,66,80,85);                        {'Info'-Knopf}
   KInit(6,'Hilfe',false,86,66,160,85);                    {'Hilfe'-Knopf}
   KInit(7,'Datei',false,6,96,80,115);                     {'Datei'-Knopf}
   KInit(8,'Beenden',false,86,96,160,115);                {'Beenden'-Knopf}
end.
```

Als Anlage 1 lag eine 3,5" Diskette bei, auf der sich neben dem unkommentierten Quellcode der Unit und des Programms noch folgende Dateien befanden:

- `EGAVGA.BGI` (die Grafiktreiber-Datei)
- `STREUUNG.EXE` (das kompilierte, ausführbare Programm)
- `STREUNIT.TPU` (die kompilierte Unit-Datei)
- `STREUUNG.PAS` (der unkommentierte Quellcode des Programms)
- `STREUNIT.PAS` (der unkommentierte Quellcode derUnit)

Anhang B

STREUUNG.PAS

Als Programmiersprache wurde Turbo Pascal® Version 7.0 von Borland International verwendet, da diese Programmiersprache ziemlich weit verbreitet ist, ich diese Sprache in der 9. und 10. Klasse gelernt habe und sie auch an diesem Gymnasium gelehrt wurde/wird.

(Die Quelldatei „Streuung.PAS[1]“ lag damals als Anlage 1 auf Diskette bei, nun unter *http://www.emerentia.de/Studium/Facharbeit/Facharbeit.html* downloadbar.)

```
{$N+}                {Compiler-Befehl: nur für mathematischen Coprozessor}
program Streuung;                                                {Programmname}

uses crt, graph, dos, StreUnit;                  {Einbinden der vier Units}

{Typen-Vereinbarung:}
type                                            {Daten für ein Streuteilchen}
   Teilchen=record;
     Q:shortint;
     Name:string[10];
     M:real;
   end;

   AtomKern=record                               {Daten für einen Atomkern}
     Q:shortint;
     Name:string[6];
     Farbe:word;
     Radius:word;
   end;

   Linie=record                                   {Daten für eine Flugbahn}
     cL:byte;
     vL:real;
```

[1] Verwedete Literatur hierfür: [14], [15], [10], [17] und [12].

```
      pL:real;
      FL:real;
      tL:real;
      KL:shortint;
   end;

   s10=string[10];                          {String mit max. 10 Zeichen}

{Konstanten-Vereinbarung:}
const
   Elektr:Teilchen=(Q:-1;Name:'Elektron';M:0.0015);
   Alpha:Teilchen=(Q:2;Name:'α-Teilchen';M:10);
                                       {Initialisieren der Streuteilchen}
   Cu:AtomKern=(Q:29;Name:'Kupfer';Farbe:6;Radius:6);
   Ag:AtomKern=(Q:47;Name:'Silber';Farbe:8;Radius:9);
   Pt:AtomKern=(Q:78;Name:'Platin';Farbe:12;Radius:15);
   Au:AtomKern=(Q:79;Name:'Gold';Farbe:14;Radius:16);
                                          {Initialisieren der Atomkerne}
   Emina=1.9;                        {minimale Energie des Alphateilchens}
   Emine=0.26;                            {minimale Energie des Elektrons}

   MaxLines=50;                                        {maximal 50 Linien}
   bkcolor=3;                                  {Farbe für den Hintergrund}

{Variablendeklaration:}
var
   i:integer;
   ir:real;                                                {zwei Variablen}
   aktuell,status:byte;                                {für die Knopfabfrage}
   POrig,PNeu:pointer;                {zum Speichern eines Bildausschnittes}
   lx,rx,oy,uy:integer;               {für die Ecken eines Options-Fensters}
   xorig,yorig,xhelp,yhelp:integer;              {Position des Mauszeigers}
   Pers:string[6];                                      {Perspektiventext}
   Winkel:integer;dreh:real;             {der Rotationswinkel; im Bogenmaß}
   x,y,z,posx,posy,posz,vy,vx,p,pAlt,C,Ekin,dt:real;
                                      {für die Berechnung der Ortspunkte}
   col,del,delalt:byte;              {für die Linienfarbe bzw. fürs 'delay'}
   text:string[8];                            {String mit max. 8 Zeichen}
   LineNr,NrAlt:byte;                      {für die Numerierung der Linien}
   SimLinie:array[1..MaxLines] of Linie;
                                        {zum Speichern der Linien im RAM}
   Lfile:file of Linie;              {zum Speichern der Linien auf Diskette}
   Teil:Teilchen;
   Kern:AtomKern;                      {zum initialisieren der Streukörper}

{Ab hier: Unterprogramme von STREUUNG.PAS}
procedure opendatei(var Ok:boolean);
```

```
{Prüft, ob Datei fehlerfrei geöffnet werden kann}
begin
   {$I-}
   Reset(Lfile);
   {$I+}
   OK:=IOResult=0;
end;

procedure SetBox(x1,y1,x2,y2:integer);
{Speichert Bildausschnitt und zeichnet ein Options-Fenster}
var size:word;
begin
   xorig:=xx;
   yorig:=yy;
   mark(POrig);
   size:=ImageSize(x1,y1,x2,y2);
   GetMem(PNeu,Size);                  {Speicher auf dem Heap reservieren}
   MOff;
   GetImage(x1,y1,x2,y2,PNeu^);
   setfillstyle(1,15);
   bar(x1+3,y1+3,x2-6,y2-6);
   setcolor(1);
   line(x1,y1,x1,y2-3);
   line(x1+1,y1+1,x1+1,y2-4);
   line(x1+2,y1+2,x1+2,y2-5);
   line(x2-3,y1,x2-3,y2-3);
   line(x2-4,y1+1,x2-4,y2-4);
   line(x2-5,y1+2,x2-5,y2-5);
   line(x1,y2-3,x2-3,y2-3);
   line(x1+1,y2-4,x2-4,y2-4);
   line(x1+2,y2-5,x2-5,y2-5);
   setfillstyle(1,1);
   bar(x1,y1,x2-3,y1+19);
   setcolor(0);
   line(x1+4,y2,x2,y2);
   line(x1+4,y2-1,x2,y2-1);
   line(x1+4,y2-2,x2,y2-2);
   line(x2,y1+4,x2,y2);
   line(x2-1,y1+4,x2-1,y2);
   line(x2-2,y1+4,x2-2,y2);
   settextjustify(Centertext,Toptext);
   setcolor(15);
   outtextxy(Mitte(x1,x2),y1+6,'\+Knopf[status].Ktext+' \);
   settextjustify(Lefttext,Toptext);
   setMxy(Mitte(x1,x2),Mitte(y1,y2));
   MLimitX(x1,x2);
   MLimitY(y1,y2);
```

```
    setcolor(0);
    settextjustify(Centertext,Toptext);
end;

procedure ReleaseBox(x1,y1:integer);
{Übermalt Options-Fenster mit gespeichertem Bildausschnitt und gibt
den belegten Speicherplatz wieder frei}
begin
    MOff;
    PutImage(x1,y1,PNeu^,NormalPut);
    MLimitX(0,maxx-2);
    MLimitY(0,maxy-2);
    SetMxy(xorig,yorig);
    MOn;
    Release(POrig);                              {Speicher wieder freigeben}
end;

procedure rahmen;
{Zeichnet den Rahmen des Hauptbildschirms}
begin
    setcolor(15);
    line(x0,y0,x0,maxy);
    line(x0,y0,maxx-1,y0);
    line(maxx-4,y0+4,maxx-4,maxy-4);
    line(xb+1,maxy-4,maxx-4,maxy-4);
    line(x0+1,y0+1,x0+1,maxy-1);
    line(x0+1,y0+1,maxx-2,y0+1);
    line(maxx-5,y0+5,maxx-5,maxy-5);
    line(xb+2,maxy-5,maxx-5,maxy-5);
    setcolor(8);
    line(maxx,y0,maxx,maxy);
    line(x0,maxy,maxx-1,maxy);
    line(xb+1,y0+4,maxx-4,y0+4);
    line(xb+1,y0+4,xb+1,maxy-4);
    line(maxx-1,y0+1,maxx-1,maxy-1);
    line(x0+1,maxy-1,maxx-2,maxy-1);
    line(xb+2,y0+5,maxx-5,y0+5);
    line(xb+2,y0+5,xb+2,maxy-5);
    setfillstyle(1,7);
    bar(xb,y0+2,maxx-2,y0+3);
    bar(xb,maxy-3,maxx-2,maxy-2);
    bar(maxx-3,y0+3,maxx-2,maxy-3);
end;

procedure balken;
{Schreibt auf der linken Seite des Monitors alle aktuellen Werte
des Streuteilchens}
```

```
begin
   setfillstyle(1,7);
   bar(x0+2,y0+125,xb,433);
   setcolor(0);
   outtextxy(x0+5,136,'Ausgangswerte:');
   outtextxy(x0+5,137,'_____________');
   outtextxy(x0+7,156,'Teilchen:');
   outtextxy(x0+3+10*8,156,Teil.Name);
   outtextxy(x0+7,172,'Atomkern:');
   outtextxy(x0+3+10*8,172,Kern.Name);
   outtextxy(x0+7,193,'Ansicht: '+Pers);
   outtextxy(x0+7,213,'Geschwindigkeit:');
   outtextxy(x0+13,227,'vx=');
   settextjustify(righttext,toptext);
   outtextxy(x0+10+15*8,227,nts(vx));
   settextjustify(lefttext,toptext);
   outtextxy(x0+10+16*8,227,'%');
   outtextxy(x0+7,247,'kinetische Energie:');
   outtextxy(x0+13,261,'E =');
   settextjustify(righttext,toptext);
   outtextxy(x0+10+15*8,261,nts(Ekin));
   settextjustify(lefttext,toptext);
   outtextxy(x0+10+16*8,261,'MeV');
   outtextxy(x0+7,281,'Verzögerung/Delay:');
   outtextxy(x0+13,295,'d =');
   settextjustify(righttext,toptext);
   outtextxy(x0+10+15*8,295,ntsn(del));
   settextjustify(lefttext,toptext);
   outtextxy(x0+7,315,'Stoßparameter:');
   outtextxy(x0+13,329,'p =');
   settextjustify(righttext,toptext);
   outtextxy(x0+10+15*8,329,nts(p));
   settextjustify(lefttext,toptext);
   outtextxy(x0+10+16*8,329,'fm');
   outtextxy(x0+7,349,'Maximalgeschw.:');
   outtextxy(x0+13,363,'vmax= 3.0E7 m/s');
   outtextxy(x0+7,383,'Linien (max.');
   outtextxy(x0+111,383,ntsn(MaxLines));
   outtextxy(x0+127,383,'):');
   outtextxy(x0+10,397,'nächste Linie:');
   outtextxy(x0+130,397,ntsn(LineNr));
   settextjustify(lefttext,toptext);
end;

procedure Ubalken;
{Zeichnet den unteren Teil des linken Monitorrandes}
begin
```

```
   setfillstyle(1,7);
   bar(x0+2,438,xb,maxy-2);
   setcolor(8);
   line(x0+1,434,xb,434);
   line(x0,435,xb,435);
   setcolor(15);
   line(x0+2,436,xb+1,436);
   line(x0+2,437,xb,437);
   setcolor(0);
   settextjustify(centertext,toptext);
   outtextxy(Mitte(x0+4,xb-4),440,'Facharbeit in Physik');
   outtextxy(Mitte(x0+4,xb-4),454,'von C. Mühlberger');
   outtextxy(Mitte(x0+4,xb-4),468,'Version 1.0');
end;

procedure Tbalken;
{Die Hintergrundfläche der Knöpfe bzw. Tasten}
begin
   setfillstyle(1,7);
   bar(x0+2,y0+2,xb,y0+125);
   setcolor(8);
   line(x0+1,121,xb,121);
   line(x0,122,xb,122);
   setcolor(15);
   line(x0+2,123,xb+1,123);
   line(x0+2,124,xb,124);
end;

procedure Taste(TNum:byte);
{Zeichnung der Knöpfe bzw. Tasten, inklusive Text}
begin
   with Knopf[TNum] do
     begin
       MOff;
       setfillstyle(1,7);
       bar(Kx1+2,Ky1+2,Kx2-2,Ky2-2);
       setcolor(0);
       rectangle(Kx1+1,Ky1+1,Kx2-1,Ky2-1);
       line(Kx1,Ky1+1,Kx1,Ky2-1);
       line(Kx1+1,Ky1,Kx2-1,Ky1);
       line(Kx1+1,Ky2,Kx2-1,Ky2);
       line(Kx2,Ky1+1,Kx2,Ky2-1);
       if KStatus=true then              {wenn Taste gedrückt, dann hier}
         begin
           setcolor(8);
           line(Kx1+2,Ky1+2,Kx2-2,Ky1+2);
           line(Kx1+2,Ky1+2,Kx1+2,Ky2-2);
```

```
            setcolor(7);
            line(Kx1+3,Ky1+3,Kx2-3,Ky1+3);
            line(Kx1+3,Ky1+3,Kx1+3,Ky2-3);
            line(Kx1+3,Ky2-2,Kx2-2,Ky2-2);
            line(Kx2-2,Ky2-2,Kx2-2,Ky1+3);
            line(Kx1+3,Ky2-3,Kx2-3,Ky2-3);
            line(Kx2-3,Ky2-3,Kx2-3,Ky1+3);
            setcolor(0);
            settextjustify(Centertext,Toptext);
            outtextxy(Mitte(Kx1,Kx2),Ky1+7,Ktext);
            settextjustify(Lefttext,Toptext);
          end
        else                                          {..., sonst hier weiter}
          begin setcolor(15);
            line(Kx1+2,Ky1+2,Kx2-2,Ky1+2);
            line(Kx1+2,Ky1+2,Kx1+2,Ky2-2);
            line(Kx1+3,Ky1+3,Kx2-3,Ky1+3);
            line(Kx1+3,Ky1+3,Kx1+3,Ky2-3);
            setcolor(8);
            line(Kx1+2,Ky2-2,Kx2-2,Ky2-2);
            line(Kx2-2,Ky2-2,Kx2-2,Ky1+2);
            line(Kx1+3,Ky2-3,Kx2-3,Ky2-3);
            line(Kx2-3,Ky2-3,Kx2-3,Ky1+3);
            setcolor(0);
            settextjustify(Centertext,Toptext);
            outtextxy(Mitte(Kx1,Kx2),Ky1+6,Ktext);
            settextjustify(Lefttext,Toptext);
          end;
      end;
   MOn;
end;

procedure Gewaehlt(K1,K2:byte);
{Welche Taste, welcher Knopf wurde gedrückt?}
begin
   repeat
     repeat Mbpressed(0);
       if bs=1 then
       for i:=K1 to K2 do
         if (xx>=Knopf[i].Kx1) AND
           (yy>=Knopf[i].Ky1) AND
           (xx<=Knopf[i].Kx2) AND
           (yy<=Knopf[i].Ky2) then
         status:=i;
     until (bs=1) AND (status in [K1..K2]) AND
       (xx>=Knopf[status].Kx1) AND
       (yy>=Knopf[status].Ky1) AND
```

```
        (xx<=Knopf[status].Kx2) AND
        (yy<=Knopf[status].Ky2);
      Knopf[status].KStatus:=true;
      Taste(status);
      repeat GetMxyb;
        if ((xx<Knopf[status].Kx1) OR
          (yy<Knopf[status].Ky1) OR
          (xx>Knopf[status].Kx2) OR
          (yy>Knopf[status].Ky2)) AND
          (Knopf[status].KStatus=true) then
        begin
          Knopf[status].KStatus:=false;
          Taste(status);
        end;
        if ((xx>=Knopf[status].Kx1) AND
          (yy>=Knopf[status].Ky1) AND
          (xx<=Knopf[status].Kx2) AND
          (yy<=Knopf[status].Ky2)) AND
          (Knopf[status].KStatus<>true) then
        begin
          Knopf[status].KStatus:=true;
          Taste(status);
        end;
      until (bb=0);
    until (xx>=Knopf[status].Kx1) AND
      (yy>=Knopf[status].Ky1) AND
      (xx<=Knopf[status].Kx2) AND
      (yy<=Knopf[status].Ky2) AND (bb=0);
    Knopf[status].KStatus:=false;
    Taste(status);
end;

procedure Hintergrund;
{Zeichnet den Hintergrund des Zeichenfensters mit der durch
'bkcolor' festgelegten Farbe}
begin
    setfillstyle(1,bkcolor);
    bar(xb+3,y0+6,maxx-6,maxy-6);
end;

procedure DrawTeil(pold,pnew:real);
{Zeichnet das Streuteilchen an den unteren Rand an die neue
Position 'pnew' und löscht es an der alten Position 'pold'}
var radius:word;
begin
    if Teil.Q=2 then radius:=2 else radius:=1;
    setcolor(bkcolor);
```

```
    setfillstyle(1,bkcolor);
    MOff;
    Pieslice(Round(CenterX+pold),Round(maxy-8),0,360,radius);
    circle(Round(CenterX+pold),Round(maxy-8),radius);
    setcolor(col);
    setfillstyle(1,col);
    Pieslice(Round(CenterX+pnew),Round(maxy-8),0,360,radius);
    circle(Round(CenterX+pnew),Round(maxy-8),radius);
    MOn;
end;

procedure DrawKern;
{Zeichnet den Kern samt Koordinatensystem}
var sx:real;sz,sy:integer;
begin
    DrawTeil(p,p);
    setcolor(Kern.Farbe);
    setfillstyle(1,Kern.Farbe);
    MOff;
    Pieslice(CenterX,CenterY,0,360,Kern.Radius);
    circle(CenterX,CenterY,Kern.Radius);
    Settextjustify(Centertext,Bottomtext);
    outtextxy(CenterX,CenterY-Kern.Radius-4,'Kern');
    setcolor(0);
    {Ab hier Berechnung des Koordinatensystem:}
    sx:=Centerx*cos(dreh)-Centery*sin(dreh);
    sz:=round(sx*cos(dreh)+z*sin(dreh));
    sy:=round(Centery*cos(dreh)+Centerx*sin(dreh));
    line(CenterX,CenterY,Centery+sz-2,Centery+CenterX-sz);
    line(CenterX,CenterY,Centerx+Centerx-sz,sy-Centery+6);
    line(CenterX,CenterY,Centerx,Centery+Centery-sy);
    Settextjustify(Lefttext,Toptext);
    outtextxy(Centery+sz-10,Centery+Centerx-sz,'x');
    outtextxy(Centerx+Centerx-sz+2,6-Centery+sy,'y');
    outtextxy(Centerx-10,Centery+Centery-sy,'z');
    MOn;
end;

procedure changeTeile;
{Ändert Streuteilchen und Streukörper (Option 'Partikel')}
const lx=170;oy=40;rx=420;uy=358;
var
    NameTMom:string[10];NameKMom:string[6];
    sT:string[10];sK:string[6];
begin
    SetBox(lx,oy,rx,uy);
    NameKMom:=Kern.Name;
```

```
NameTMom:=Teil.Name;
outtextxy(Mitte(lx,rx-3),oy+26,'Bitte wählen Sie:');
outtextxy(Mitte(lx,rx-3),oy+40,'(Mit "OKAY" gehen alte Linien');
outtextxy(Mitte(lx,rx-3),oy+52,'und Zeichnungen verloren!)');
KInit(9,'Okay',false,lx+10,uy-27,lx+10+74,uy-27+19);
KInit(10,'Abbruch',false,rx-87,uy-27,rx-87+74,uy-27+19);
KInit(11,'Elektron',false,lx+8,oy+69,lx+82,oy+88);
KInit(12,'a-Teil.',false,lx+8,oy+94,lx+82,oy+113);
KInit(13,'Kupfer',false,lx+8,oy+155,lx+82,oy+174);
KInit(14,'Silber',false,lx+8,oy+180,lx+82,oy+199);
KInit(15,'Platin',false,lx+8,oy+205,lx+82,oy+224);
KInit(16,'Gold',false,lx+8,oy+230,lx+82,oy+249);
for i:=9 to 16 do Taste(i);
settextjustify(Lefttext,Centertext);
str(Elektr.Q:2,sT);
outtextxy(lx+88,Mitte(oy+88,oy+69),'Ladung in e: '+sT);
str(Alpha.Q:2,sT);
outtextxy(lx+88,Mitte(oy+113,oy+94),'Ladung in e: '+sT);
settextjustify(Lefttext,Toptext);
outtextxy(lx+12,oy+119,'urspr. Teilchen: '+Teil.Name);
outtextxy(lx+12,oy+135,'momentanes Teil: '+Teil.Name);
line(lx+8,oy+148,rx-11,oy+148);
settextjustify(Lefttext,Centertext);
str(Cu.Q,sK);
outtextxy(lx+88,Mitte(oy+174,oy+155),'Ordnungszahl:'+sK);
str(Ag.Q,sK);
outtextxy(lx+88,Mitte(oy+199,oy+180),'Ordnungszahl:'+sK);
str(Pt.Q,sK);
outtextxy(lx+88,Mitte(oy+224,oy+205),'Ordnungszahl:'+sK);
str(Au.Q,sK);
outtextxy(lx+88,Mitte(oy+249,oy+230),'Ordnungszahl:'+sK);
settextjustify(Lefttext,Toptext);
outtextxy(lx+12,oy+255,'urspr. Atomkern: '+Kern.Name);
outtextxy(lx+12,oy+271,'momentaner Kern: '+Kern.Name);
line(lx+8,oy+284,rx-11,oy+284);
MOn;
repeat gewaehlt(9,16);
  case status of
  9: begin
       Teil.Name:=NameTMom;
       if Teil.Name='Elektron' then Teil:=Elektr
       else Teil:=Alpha;
       Kern.Name:=NameKMom;
       if Kern.Name='Gold' then Kern:=Au else
         if Kern.Name='Platin' then Kern:=Pt else
           if Kern.Name='Silber' then Kern:=Ag else
             if Kern.Name='Kupfer' then Kern:=Cu;
```

```
    col:=0;
    LineNr:=1;
    ReleaseBox(lx,oy);
    Hintergrund;
    Drawkern;
    DrawTeil(p,p);
    balken;
  end;
10: begin
    MOff;
    Kern:=Kern;
    Teil:=Teil;
    ReleaseBox(lx,oy);
    MOn;
  end;
11: begin
    setcolor(15);
    outtextxy(lx+148,oy+135,NameTMom);
    NameTMom:='Elektron';
    setcolor(0);
    outtextxy(lx+148,oy+135,NameTMom);
  end;
12: begin
    setcolor(15);
    outtextxy(lx+148,oy+135,NameTMom);
    NameTMom:='a-Teilchen';
    setcolor(0);
    outtextxy(lx+148,oy+135,NameTMom);
  end;
13: begin
    setcolor(15);
    outtextxy(lx+148,oy+271,NameKMom);
    NameKMom:='Kupfer';
    setcolor(0);
    outtextxy(lx+148,oy+271,NameKMom);
  end;
14: begin
    setcolor(15);
    outtextxy(lx+148,oy+271,NameKMom);
    NameKMom:='Silber';
    setcolor(0);
    outtextxy(lx+148,oy+271,NameKMom);
  end;
15: begin
    setcolor(15);
    outtextxy(lx+148,oy+271,NameKMom);
    NameKMom:='Platin';
```

```
          setcolor(0);
          outtextxy(lx+148,oy+271,NameKMom);
        end;
      16: begin
          setcolor(15);
          outtextxy(lx+148,oy+271,NameKMom);
          NameKMom:='Gold';
          setcolor(0);
          outtextxy(lx+148,oy+271,NameKMom);
        end;
      end;
   until (status=9) OR (status=10);
end;

procedure Info;
{Gibt eine kurze Info zum Programm aus (Option 'Info')}
const lx=302;oy=105;rx=502;uy=355;
begin
   SetBox(lx,oy,rx,uy);
   outtextxy(Mitte(lx,rx-3),oy+26 ,'Facharbeit im');
   outtextxy(Mitte(lx,rx-3),oy+38 ,'LEISTUNGSKURS PHYSIK');
   outtextxy(Mitte(lx,rx-3),oy+62 ,'Thema: SSimulation der');
   outtextxy(Mitte(lx,rx-3),oy+74 ,'Bewegung von α-Teilchen');
   outtextxy(Mitte(lx,rx-3),oy+86 ,'bzw. Elektronen im');
   outtextxy(Mitte(lx,rx-3),oy+98 ,'elektrischen Feld eines');
   outtextxy(Mitte(lx,rx-3),oy+110,'Atomkerns\);
   outtextxy(Mitte(lx,rx-3),oy+130,'Programmiert von');
   outtextxy(Mitte(lx,rx-3),oy+150,'CLEMENS MÜHLBERGER');
   outtextxy(Mitte(lx,rx-3),oy+170,'FINSTERWALDER-GYMNASIUM');
   outtextxy(Mitte(lx,rx-3),oy+190,'Rosenheim/Obb.');
   outtextxy(Mitte(lx,rx-3),oy+206,'Schuljahr 1997/1998');
   settextjustify(Lefttext,Toptext);
   KInit(9,'Okay',false,Mitte(lx,rx-3)-37,uy-27,Mitte(lx,rx-3)+37,uy-8);
   Taste(9);
   MOn;
   Gewaehlt(9,9);
   ReleaseBox(lx,oy);
end;

procedure Hilfe;
{Gibt einen kurzen Hilfetext aus (Option 'Hilfe')}
const lx=250;oy=30;rx=568;uy=425;
begin
   SetBox(lx,oy,rx,uy);
   outtextxy(Mitte(lx,rx-3),oy+24,
                        'Zur Benutzung des Programms STREUUNG');
   outtextxy(Mitte(lx,rx-3),oy+25,
```

```
                        '_____________________________________________');
   settextjustify(Lefttext,Toptext);
   outtextxy(lx+8,oy+50 ,'START    : Zeichnet die Flugbahn des');
   outtextxy(lx+8,oy+62 ,'         Streuteilchens mit den ein-');
   outtextxy(lx+8,oy+74 ,'         gestellten Parametern.');
   outtextxy(lx+8,oy+98 ,'ANSICHT : Erlaubt die Änderung der');
   outtextxy(lx+8,oy+110,'         Perspektive.');
   outtextxy(lx+8,oy+134,'FAKTOREN: Eingabe der Parameter des');
   outtextxy(lx+8,oy+146,'         Streuteilchens per Tastatur');
   outtextxy(lx+8,oy+158,'         bzw. per Maus.');
   outtextxy(lx+8,oy+182,'PARTIKEL: Sie können das Streuteil-');
   outtextxy(lx+8,oy+194,'         chen und den Kern auswählen.');
   outtextxy(lx+8,oy+218,'INFO     : Eine kurze Information zum');
   outtextxy(lx+8,oy+230,'         Programm wird eingeblendet.');
   outtextxy(lx+8,oy+254,'HILFE    : Diese Hilfe erscheint.');
   outtextxy(lx+8,oy+278,'DATEI    : Sie können ihre simulierten');
   outtextxy(lx+8,oy+290,'         Flugbahnen abspeichern und');
   outtextxy(lx+8,oy+302,'         wieder laden.');
   outtextxy(lx+8,oy+326,'BEENDEN : Hiermit kehren Sie zur DOS-');
   outtextxy(lx+8,oy+338,'         Oberfläche zurück.');
   settextjustify(Centertext,Toptext);
   outtextxy(Mitte(lx,rx-3),oy+354,
                        'Viel Spaß noch beim "Simulieren"!');
   settextjustify(Lefttext,Toptext);
   KInit(9,'Okay',false,Mitte(lx,rx-3)-37,uy-27,Mitte(lx,rx-3)+37,uy-8);
   Taste(9);
   MOn;
   gewaehlt(9,9);
   ReleaseBox(lx,oy);
end;

function R(x,y:real):real;
{Berechnet den Radius R bzw. den Abstand des Streuteilchens zum Atomkern}
begin
   R:=sqrt(sqr(x)+sqr(y));
end;

function F:real;
{Berechnet die Kraft F nach dem Coulomb-Gesetz}
begin
   F:=Kern.Q*C/sqr(R(x,y));
end;

function a(zahl:real):real;
{Berechnet die Teilchenbeschleunigung a in 'zahl'-Richtung}
begin
   a:=(F/Teil.M)*(zahl/R(x,y));
```

```
end;

procedure Start;
{Berechnet und zeichnet die Flugbahnen. Da dieses Programm die Bewegungen
der Streuteilchen in Atomkernnähe nur simulieren soll, habe ich für
(e*e)/(4*π*ε0) den konstanten Wert const=6.25E2 gewählt, was in
Zusammenhang mit einer vereinfachten Masse der Streuteilchen (vgl.
Konstantenvereinbarung: M=10 bzw. M=0.0015) sowie einer vereinfachten
Geschwindigkeit (vgl. procedure Faktoren: vx von 0 bis 100) zu
annehmbaren Ergebnissen führt. Außerdem wurde durch Probieren dt=1
gewählt, was durch Umrechnen (siehe dtVar) zu guten Resultaten
führt (Option 'Start').}
var vxVar,dtVar:real;
begin
   if vx=0 then exit;
   C:=Teil.Q*6.25E2;
   if Teil.Q=-1 then
     begin
       x:=-23.6;
       vxVar:=10*vx;
{Multiplikation mit 10, da vx sonst zu gering wäre, so daß alle
Elektronen sofort in den Kern stürzen würden.}
       dtVar:=dt/100000;
{Division mit 100000, da sonst Linien zu ungenau oder gar falsch
gezeichnet würden!}
     end else
     begin
       x:=-23.4;
       vxVar:=vx;
       dtVar:=dt/1000;
{Für Alphateilchen reicht als Divisor 1000}
     end;
   y:=p/10;
   z:=234;
   vy:=0;                  {Die horizontale Geschwindigkeit ist anfangs 0}
   with SimLinie[LineNr] do                    {Speichern der Liniendaten}
   begin
     cL:=col;
     vL:=vx;
     pL:=p;
     FL:=C;
     tL:=dt;
     KL:=Kern.Q;
   end;
   MOff;
   drawteil(p,p);
   posz:=z;
```

```
   repeat putpixel(round(CenterX+posy*10),
            round(CenterY-posz*10),
            col);
     vxVar:=vxVar+a(x)*dtVar;
     vy:=vy+a(y)*dtVar;
{Berechnen der Koordinaten im x-y-System (vgl. 3.1.1, S. 11)}
     x:=x+vxVar*dtVar;
     y:=y+vy*dtVar;
{Berechnen der Koordinaten im x-y-z-System (vgl. 3.1.2, S. 13)}
     posx:=x*cos(dreh)-y*sin(dreh);
     posy:=y*cos(dreh)+x*sin(dreh);
     posz:=posx*cos(dreh)+posz*sin(dreh);
     delay(del);                                    {mögliche Verzögerung}
   until (posy*10+CenterY<=1) or (posy*10-CenterY>=-1) or
        (posx*10>=CenterY-6) or (posx*10+CenterY<=-25);
   if (LineNr<MaxLines) then LineNr:=LineNr+1
   else LineNr:=1;
   if (col=1) then col:=4                                {Farbe ändern}
     else if (col=6) OR (col=7) OR (col=8) then col:=9
       else if (col=15) then col:=0
         else if (col=Kern.Farbe-1) then col:=Kern.Farbe+1
           else col:=col+1;
   rahmen;
   balken;
   MOn;
end;

procedure Faktoren;
{Beginn der langen Prozedur 'Faktoren' zum Ändern der einzelnen Parameter
für das Streuteilchen (Option 'Faktoren')}
const lx=170;oy=20;rx=498;uy=310;
var
   ch:char;s:array [1..5] of s10;z:byte;
   Klx,Koy,Krx,Kuy,pneu,palt:integer;

procedure Kasten(num:byte);
{Unterprozedur von 'Faktoren'; löscht Zahlen auf dem Balken}
begin
   Klx:=45;
   Koy:=191+34*num;
   Krx:=Klx+90;
   Kuy:=Koy+14;
   setfillstyle(1,7);
   bar(Klx,Koy,Krx,Kuy);
   setcolor(0);
end;
```

```
procedure Krahmen(num:byte;col:boolean);
{Unterprozedur von 'Faktoren'; setzt einen schwarzen Rahmen auf
dem Balken, damit der Benutzer weiß, wo er seine neuen Werte
eingibt bzw. eingeben muß.}
begin
   Klx:=44;
   Koy:=190+34*num-1;
   Krx:=Klx+92;
   Kuy:=Koy+17;
   if col then setcolor(0) else setcolor(7);
   line(Klx,Koy,Klx,Kuy);
   line(Klx,Koy,Krx,Koy);
   line(Krx,Koy,Krx,Kuy);
   line(Klx,Kuy,Krx,Kuy);
end;

function Eingabe(Zeile:byte):s10;
{Unterfunktion von 'Faktoren'; übergibt als Ergebnis die neu
eingegebenen Zahlenwerte}
begin
   MOff;
   s[zeile]:='';
   settextjustify(Lefttext,Toptext);
   i:=48;
   KRahmen(Zeile,true);
   while (ch<>#13) do                                    {ENTER-Taste}
   begin
     ch:=readkey;
     if (ch=#27) then                                   {ESCAPE-Taste}
       begin
         KRahmen(Zeile,false);
         s[Zeile]:='';
         ch:=#0;
         MOn;
         exit;
       end;
     if (ch=#08) then                                      {BackSpace}
       begin
         if i>48 then i:=i-8 else i:=48;
         setcolor(7);
         outtextxy(i,194+34*Zeile,'');
         setcolor(0);
         if length(s[Zeile])>1 then
           delete(s[Zeile],length(s[Zeile]),2)
         else s[Zeile]:='';
       end
     else
```

```
        if (ch=#32) then                                    {SPACE-Taste}
        begin
          i:=i;
        end
        else
          if (ch in Ziffern) then
            begin
              if (ch=#44) then ch:=#46;              {wenn '.', dann ','}
              if (i=48) then
                begin
                  Kasten(Zeile);
                  s[Zeile]:='';
                end;
              outtextxy(i,194+34*Zeile,ch);
              if i<128 then inc(i,8)
              else
                begin
                  i:=128;
                  setcolor(7);
                  outtextxy(i-8,194+34*Zeile,'');
                  setcolor(0);
                  delete(s[Zeile],length(s[Zeile]),1);
                  outtextxy(i-8,194+34*Zeile,ch);
                end;
              s[Zeile]:=s[Zeile]+ch;
            end;
   end;
   KRahmen(Zeile,false);
   if s[Zeile]<>'' then Eingabe:=s[zeile];
   ch:=#0;
   MOn;
end;

procedure Falsch;
{Unterprozedur von 'Faktoren'; gibt eine Fehlermeldung aus, wenn
'p (Maus)' bei Schrägansicht gewählt wurde.}
const lx2=170;oy2=40;rx2=420;uy2=134;
begin
   setbox(lx2,oy2,rx2,uy2);
   outtextxy(Mitte(lx2,rx2-3),oy2+26,'!Achtung!');
   outtextxy(Mitte(lx2,rx2-3),oy2+40,'Diese Option steht jetzt');
   outtextxy(Mitte(lx2,rx2-3),oy2+54,'leider nicht zur Verfügung!');
   settextjustify(Lefttext,Toptext);
   KInit(9,'Okay',false,Mitte(lx2,rx2-3)-37,uy2-27,
            Mitte(lx2,rx2-3)+37,uy2-8);
   Taste(9);
   MOn;
```

```
   Gewaehlt(9,9);
   ReleaseBox(lx2,oy2);
end;

{eigentlicher Teil der Prozedur 'Faktoren'}
begin
   SetBox(lx,oy,rx,uy);
   outtextxy(Mitte(lx,rx-3),oy+26,
            'Bitte auswählen und neuen Wert einge-');
   outtextxy(Mitte(lx,rx-3),oy+43,
            'ben, aber folgende Grenzwerte beachten:');
   outtextxy(Mitte(lx,rx-3),oy+190,
            'Erlaubt sind: "0"-"9"  "-"  "+"  "."  ","');
   outtextxy(Mitte(lx,rx-3),oy+204,
            'Bei [ESC] bzw. rechter Maustaste oder');
   outtextxy(Mitte(lx,rx-3),oy+218,
            'falscher Eingabe bleibt der alte Wert.');
   outtextxy(Mitte(lx,rx-3),oy+232,
            'Die Eingabe mit [ENTER] bestätigen,');
   outtextxy(Mitte(lx,rx-3),oy+246,
            'bei " p (Maus)" mit linker Maustaste.');
   KInit(10,'Okay',false,Mitte(lx,rx)-37,uy-27,
            Mitte(lx,rx)+37,uy-27+19);
   KInit(11,'vx',false,lx+8,oy+60,lx+82,oy+79);
   KInit(12,'Ekin',false,lx+8,oy+85,lx+82,oy+104);
   KInit(13,'delay',false,lx+8,oy+110,lx+82,oy+129);
   KInit(14,'p (Zahl)',false,lx+8,oy+135,lx+82,oy+154);
   KInit(15,'p (Maus)',false,lx+8,oy+160,lx+82,oy+179);
   for i:=10 to 15 do Taste(i);
   settextjustify(Centertext,Centertext);
   outtextxy(Mitte(lx+90,rx-3),oy+70,'0% < vx = 100% ');
   if Teil.Q=2 then outtextxy(Mitte(lx+90,rx-3),oy+95,
            '1.9 MeV < Ekin = 19 MeV')
   else outtextxy(Mitte(lx+90,rx-3),oy+95,
            '0.26 MeV < Ekin = 2.6 MeV');
   outtextxy(Mitte(lx+90,rx-3),oy+120,'0 = delay = 5');
   outtextxy(Mitte(lx+90,rx-3),oy+145,
            '-231.0 fm = p = 231.0 fm');
   if pers='oben' then outtextxy(Mitte(lx+90,rx-3),oy+170,
            'p zw. linkem\rechtem Rand')
   else outtextxy(Mitte(lx+90,rx-3),oy+170,
            'nur unter Ansicht:"OBEN"!');
   settextjustify(Lefttext,Toptext);
   xhelp:=xorig;
   yhelp:=yorig;
   MOn;
   repeat gewaehlt(10,15);
```

```
case status of
10: begin
     MOff;
     xorig:=xhelp;
     yorig:=yhelp;
     ReleaseBox(lx,oy);
     MOn;
   end;
11: begin
     val(Eingabe(1),ir,i);
     if (s[1]<>'') AND (ir<=100) AND (ir>=1) AND (ir<>0) then
       vx:=ir
     else vx:=vx;
     if Teil.Q=2 then Ekin:=vx*Emina/10
     else Ekin:=vx*Emine/10;
     balken;
   end;
12: begin
     val(Eingabe(2),ir,i);
     if (s[2]<>'') AND (ir>=1.9) AND (ir<=19) AND (Teil.Q=2) then
       begin
         Ekin:=ir;
         vx:=Ekin/Emina*10;
       end
     else
       if (s[2]<>'') AND (ir>=0.26) AND
            (ir<=2.6) AND (Teil.Q=-1) then
         begin
           Ekin:=ir;
           vx:=Ekin/Emine*10;
         end
       else Ekin:=Ekin;
     balken;
   end;
13: begin
     val(Eingabe(3),ir,i);
     if (s[3]<>'') AND (ir>=0) AND (ir<=5) then del:=trunc(ir)
     else del:=del;
     balken;
   end;
14: begin
     val(Eingabe(4),ir,i);
     if (s[4]='') then
       begin
         Drawteil(p,p);
         p:=p;
       end
```

```
    else
      if (s[4]='0') OR ((ir<=231.0) AND (ir>=-231.0)) then
        begin
          DrawTeil(p,ir);
          p:=ir;
        end
      else
        begin
          Drawteil(p,p);
          p:=p;
        end;
    balken;
  end;
15: begin
    MOff;
    if (Pers<>'oben') then
      begin
        Releasebox(lx,oy);
        MOn;
        Falsch;
        Faktoren;
      end
    else
      begin
        Releasebox(lx,oy);
        MLimitX(xb+5,maxx-8);
        MLimitY(maxy-10,maxy-10);
        setMxy(CenterX+round(p),maxy-10);
        MOn;
        palt:=round(p);
        repeat
          GetMxyb;
          if palt<>xx-CenterX then
            begin
              MOff;
              Kasten(4);
              pneu:=xx-CenterX;
              outtextxy(48,194+34*4,ntsn(pneu));
              DrawTeil(palt,pneu);
              palt:=pneu;
              MOn;
            end;
        until (bb=1) OR (bb=2);
        case bb of
        1: p:=pneu;
        2: begin
             p:=p;
```

```
                    DrawTeil(pneu,p);
                  end;
                end;
                balken;
                xx:=xhelp;
                yy:=yhelp;
                ch:=' ';
                Faktoren;
                end;
              end;
      end;
   until (status=10);
end;

procedure Ansicht;
{Ändert den Betrachtungsstandpunkt und zeichnet die Linien unter der
neuen Perspektive (Option 'Ansicht')}
const lx=170;oy=40;rx=340;uy=166;
var PersVar:string[6];WinAlt:integer;
begin
   SetBox(lx,oy,rx,uy);
   outtextxy(Mitte(lx,rx-3),oy+26,'Bitte wählen Sie');
   outtextxy(Mitte(lx,rx-3),oy+38,'eine Perspektive:');
   KInit(9,'Okay',false,lx+8,uy-27,lx+8+74,uy-27+19);
   KInit(10,'Abbruch',false,rx-85,uy-27,rx-85+74,uy-27+19);
   KInit(11,'von oben',false,lx+8,oy+50,lx+82,oy+69);
   KInit(12,'schräg',false,rx-85,oy+50,rx-85+74,oy+69);
   for i:=9 to 12 do Taste(i);
   outtextxy(lx+12,oy+74,'vorher: '+Pers);
   outtextxy(lx+12,oy+86,'jetzt : '+Pers);
   PersVar:=Pers;
   WinAlt:=Winkel;
   MOn;
   repeat gewaehlt(9,12);
     case status of
     9: begin
          Pers:=PersVar;
          Winkel:=WinAlt;
          dreh:=Winkel*PI/180;
          ReleaseBox(lx,oy);
          Hintergrund;
          Drawkern;
          NrAlt:=LineNr;
          if NrAlt>=2 then
            for i:=1 to NrAlt do
              begin
                palt:=p;
```

```
                delalt:=del;
                del:=0;
                col:=SimLinie[i].cL;
                vx:=SimLinie[i].vL;
                p:=SimLinie[i].pL;
                dt:=SimLinie[i].tL;
                start;
                MOff;
                drawteil(palt,p);
                LineNr:=i;
                del:=delalt;
                balken;
                MOn;
              end;
          MOff;
          balken;
          MOn;
        end;
      10: ReleaseBox(lx,oy);
      11: begin
          setcolor(15);
          outtextxy(lx+76,oy+86,PersVar);
          PersVar:='oben';
          setcolor(0);
          outtextxy(lx+76,oy+86,PersVar);
          WinAlt:=0;
        end;
      12: begin
          setcolor(15);
          outtextxy(lx+76,oy+86,PersVar);
          PersVar:='schräg';
          setcolor(0);
          outtextxy(lx+76,oy+86,PersVar);
          WinAlt:=15;
        end;
      end;
   until (status=9) OR (status=10);
end;

procedure Datei;
{Beginn der langen Prozedur 'Datei' zum Speichern bzw. Laden
simulierter Teilchenbewegungen (Option 'Datei')}
const lx=170;oy=40;rx=374;uy=178;
var
   Name:string;hStr:string[1];
   speichern:boolean;
```

```
procedure Vorgang(Nummer:byte);
{Unterprozedur von 'Datei'; führt Speichern bzw. Laden durch}

procedure Fehler;
{Unterprozedur der Unterprozedur 'Vorgang' von 'Datei'; gibt bei
einem Datei-lesen-Fehler eine Meldung aus}
const lx3=170;oy3=40;rx3=374;uy3=134;
begin
   xhelp:=xorig;
   yhelp:=yorig;
   setbox(lx3,oy3,rx3,uy3);
   outtextxy(Mitte(lx3,rx3-3),oy3+26,'!Fehler beim Laden!');
   outtextxy(Mitte(lx3,rx3-3),oy3+40,
        'Die '+Knopf[status].Ktext+' konnte');
   outtextxy(Mitte(lx3,rx3-3),oy3+54,
        'nicht gefunden werden!');
   settextjustify(Lefttext,Toptext);
   KInit(9,'Okay',false,Mitte(lx3,rx3-3)-37,uy3-27,
        Mitte(lx3,rx3-3)+37,uy3-8);
   Taste(9);
   MOn;
   Gewaehlt(9,9);
   xorig:=xhelp;
   yorig:=yhelp;
   ReleaseBox(lx3,oy3);
end;

{eigentliche Unterprozedur 'Vorgang' von 'Datei'}
begin
   str(Nummer-9,hStr);
   if length(path)>=4 then Name:=Path+'\Streuung.00'+hStr
   else Name:=path+'Streuung.00'+hStr;
   i:=LineNr;
   if speichern then

     {Speichern von Teilchenbewegungen:}
     begin
       Assign(Lfile,Name);
       Rewrite(Lfile);
       if i>=2 then
         for LineNr:=1 to i-1 do
           Write(Lfile,SimLinie[LineNr])
       else Write(Lfile,SimLinie[1]);
       Close(Lfile);
     end
     else
     {Laden von Teilchenbewegungen:}
```

```
      begin
        Assign(Lfile,Name);
        opendatei(ok);
        if not ok then
          begin
            Fehler;
            exit;
          end
        else
          begin
            i:=0;
            LineNr:=1;
            del:=0;
            Kern.Q:=SimLinie[LineNr].KL;
            case Kern.Q of
              29:Kern:=Cu;
              47:Kern:=Ag;
              78:Kern:=Pt;
              79:Kern:=Au;
            end;
            hintergrund;
            drawKern;
            while not eof(Lfile) do
              begin
                inc(i);
                Read(Lfile,SimLinie[i]);
                palt:=p;
                col:=SimLinie[i].cL;
                vx:=SimLinie[i].VL;
                p:=SimLinie[i].pL;
                C:=SimLinie[i].FL;
                if (C=-6.25E2) then Teil:=Elektr
                else Teil:=Alpha;
                dt:=SimLinie[i].tL;
                start;
                drawteil(palt,p);
              end;
            LineNr:=i+1;
            balken;
            Close(Lfile);
          end;
      end;
end;                                    {Ende der Unterprozedur 'Vorgang'}

procedure Dateien;
{Unterprozedur von 'Datei'; wählt zu bearbeitende Datei aus}
const lx2=170;oy2=40;rx2=420;uy2=340;
```

```
begin
   SetBox(lx2,oy2,rx2,uy2);
   outtextxy(Mitte(lx2,rx2-3),oy2+26,
        'Wählen Sie eine Datei aus...');
   for i:=1 to 9 do
     begin
       str(i,hStr);
       if length(path)>=4 then Name:=Path+'\Streuung.00'+hStr
       else Name:=path+'Streuung.00'+hStr;
       settextjustify(centertext,toptext);
       if FExist(Name) then
         outtextxy(Mitte(lx2+95,rx2-3),
              Mitte(oy2+17+25*i,oy2+36+25*i),
              'Datei belegt')
       else
         outtextxy(Mitte(lx2+95,rx2-3),
              Mitte(oy2+17+25*i,oy2+36+25*i),
              'unbenutzt');
     end;
   KInit(9,'Zurück',false,Mitte(lx2,rx2-3)-37,uy2-27,
        Mitte(lx2,rx2-3)+37,uy2-27+19);
   KInit(10,'Datei 1',false,lx2+8,oy2+44,lx2+82,oy2+63);
   KInit(11,'Datei 2',false,lx2+8,oy2+69,lx2+82,oy2+88);
   KInit(12,'Datei 3',false,lx2+8,oy2+94,lx2+82,oy2+113);
   KInit(13,'Datei 4',false,lx2+8,oy2+119,lx2+82,oy2+138);
   KInit(14,'Datei 5',false,lx2+8,oy2+144,lx2+82,oy2+163);
   KInit(15,'Datei 6',false,lx2+8,oy2+169,lx2+82,oy2+188);
   KInit(16,'Datei 7',false,lx2+8,oy2+194,lx2+82,oy2+213);
   KInit(17,'Datei 8',false,lx2+8,oy2+219,lx2+82,oy2+238);
   KInit(18,'Datei 9',false,lx2+8,oy2+244,lx2+82,oy2+263);
   for i:=1 to 18 do Taste(i);
   xorig:=xhelp;
   yorig:=yhelp;
   MOn;
   repeat gewaehlt(9,18);
     case status of
       9:ReleaseBox(lx2,oy2)
     else
       begin
         ReleaseBox(lx2,oy2);
         Vorgang(status);
         exit;
       end;
     end;
   until status=9;
end;
```

```
{eigentlicher Teil der Prozedur 'Datei':}
begin
   SetBox(lx,oy,rx,uy);
   outtextxy(Mitte(lx,rx-3),oy+26,
        'Wollen Sie eine Datei...');
   outtextxy(Mitte(lx,rx-3),oy+67,
        'oder doch wieder zurück?');
   outtextxy(Mitte(lx,rx-3),oy+81,
        '(SICHERN kann vorhandene');
   outtextxy(Mitte(lx,rx-3),oy+95,
        'Dateien überschreiben!)');
   KInit(10,'Zurück',false,Mitte(lx,rx)-37,uy-27,
        Mitte(lx,rx)+37,uy-27+19);
   KInit(11,'Laden',false,lx+8,oy+40,lx+82,oy+59);
   KInit(12,'Sichern',false,rx-85,oy+40,rx-85+74,oy+59);
   for i:=10 to 12 do Taste(i);
   xhelp:=xorig;
   yhelp:=yorig;
   MOn;
   repeat gewaehlt(10,12);
     case status of
     10: ReleaseBox(lx,oy);
     11: begin
          ReleaseBox(lx,oy);
          speichern:=false;
          Dateien;
          exit;
        end;
     12: begin
          ReleaseBox(lx,oy);
          speichern:=true;
          Dateien;
          exit;
        end;
     end;
   until (status=10);
end;                                          {Ende der Prozedur 'Datei'}

procedure Beenden;
{Beendet das Computerprogramm (Option 'Beenden')}
const lx=302;oy=160;rx=502;uy=250;
begin
   SetBox(lx,oy,rx,uy);
   outtextxy(Mitte(lx,rx-3),oy+30,
        'Wollen Sie das Programm');
   outtextxy(Mitte(lx,rx-3),oy+45,'wirklich beenden?');
   KInit(11,'Ja',false,lx+10,uy-27,lx+84,uy-8);
```

```
   KInit(12,'Nein',false,rx-87,uy-27,rx-13,uy-8);
   for i:=11 to 12 do Taste(i);
   MOn;
   gewaehlt(11,12);
   case status of
   11:begin
        Closegraph;
        Halt(0);
      end;
   12:ReleaseBox(lx,oy);
   end;
end;

{Hauptteil des Programms 'STREUUNG.PAS':}
begin
   {Standardeinstellungen:}
   Pers:='oben';
   Teil:=Alpha;
   Kern:=Au;
   vx:=50;
   dt:=1.0;
   p:=25;
   z:=234.0;
   del:=0;
   col:=0;
   LineNr:=1;
   Ekin:=vx*Emina/10;
   with SimLinie[LineNr] do
     begin
       cL:=col;
       vL:=vx;
       pL:=p;
       FL:=C;
       tL:=dt;
       KL:=Kern.Q;
     end;
   {Den Bildschirm neu zeichnen}
   rahmen;
   Tbalken;
   balken;
   UBalken;
   Hintergrund;
   DrawKern;
   DrawTeil(p,p);
   {Die Tasten einfügen:}
   for i:=1 to 8 do Taste(i);
   MOn;
```

```
   while (aktuell<>8) do
     begin gewaehlt(1,8);
       case status of         1:Start;
       2:Ansicht;
       3:Faktoren;
       4:changeTeile;
       5:Info;
       6:Hilfe;
       7:Datei;
       8:Beenden;
       end;
     end;
   MOff;
   closegraph;
   close(Lfile);
end.                                                {Ende des Programms}
```

Anhang C

Einheiten[1]

m	:	Meter (Längeneinheit)
s	:	Sekunde (Zeiteinheit)
kg	:	Kilogramm (Gewichtseinheit)
A	:	Ampere (Stromeinheit, nach AMPÈRE)
C	:	Coulomb (elektrische Ladung, nach COULOMB; $C = A \cdot s$)
N	:	Newton (Krafteinheit, nach NEWTON; $N = m \cdot kg \cdot s^{-2}$)
J	:	Joule (Energieeinheit, nach JOULE; $J = N \cdot m = kg \cdot m^2 \cdot s^{-2}$)
eV	:	Elektronenvolt ($1\,eV = 1,60 \cdot 10^{-19}\,J; 1\,J = 6,24 \cdot 10^{18}\,eV$)
V	:	Volt (Spannungseinheit, nach VOLTA; $V = J \cdot C^{-1}$)
bar	:	Bar (Luftdruckeinheit; $1\,bar = 103\,mbar = 105\,Pa$ [PASCAL]; $Pa = N \cdot m^{-2}$)
$^\circ$ C	:	Grad Celsius (Temperaturskala nach CELSIUS)

[1] siehe [9]

Anhang D

Konstanten[1]

$$
\begin{aligned}
\pi &= 3,141592654 \\
c &= 2,99792458 \cdot 10^{8}\,\mathrm{m} \cdot \mathrm{s}^{-1} \\
\varepsilon_0 &= 8,8542 \cdot 10^{-12}\,\mathrm{C} \cdot \mathrm{V}^{-1} \cdot \mathrm{m}^{-1} \\
e &= 1,6022 \cdot 10^{-19}\,\mathrm{C} \\
m_\alpha &= 6,6446616 \cdot 10^{-27}\,\mathrm{kg} \\
Q_\alpha &= +2 \cdot e = 3,2044 \cdot 10^{-19}\,\mathrm{C} \\
m_{e^-} &= 9,1094 \cdot 10^{-31}\,\mathrm{kg} \\
Q_{e^-} &= -1 \cdot e = -1,6022 \cdot 10^{-19}\,\mathrm{C}
\end{aligned}
$$

[1] siehe [9]

Anhang E

Personenregister[1]

Ampère, Andrè Marie, 1775-1836; französischer Physiker und Mathematiker

Celsius, Anders, 1701-1744; schwedischer Astronom

Coulomb, Charles Augustin de, 1736-1806; französischer Physiker

Demokrit von Abdera, um 460-370 v. Chr.; griechischer Philosoph, Schüler des Leukipp

Geiger, Hans (Johannes) Wilhelm, 1882-1945; deutscher Physiker, Schüler des Rutherford

Joule, James Prescott, 1818-1889; britischer Physiker

Kay, William; Rutherfords Assistent; Verwalter des Laboratoriums in Manchester, Großbritannien

Kepler, Johannes, 1571-1630; deutscher Astronom

Lenard, Philip, 1862-1947; deutscher Physiker, Hauptgegner der Relativitätstheorie, Nobelpreis 1905

Leukipp von Milet, zweite Hälfte des 5. Jhd. v. Chr.; griechscher Philosoph, Vertreter des Atomismus

Marsden, Ernest, 1889-1970; neuseeländischer Student, Rutherfords Assistent

Newton, Sir (seit 1705) ISAAC, 1643-1727; englischer Mathematiker, Physiker und Astronom, Begründer der Himmelsmechanik und der klassischen theoretischen Physik

Pascal, Blaise, 1623-1662; französischer Philosoph, Mathematiker und Physiker; (nach ihm wurde auch die Programmiersprache Turbo Pascal® benannt)

[1] siehe [13] und [3]

Pythagoras von Samos, 570-497/496 v. Chr.; griechischer Philosoph

Rutherford, Ernest (seit 1930: Lord Rutherford of Nelson), 1871-1937; britischer Physiker, Nobelpreis 1908

Thomson, Sir (seit 1908) Joseph John, 1856-1940; britischer Physiker, Nobelpreis 1906

Volta, Alessandro Graf (seit 1810), 1745-1827; italienischer Physiker

Wilson, Charles Thomas Rees, 1869-1959; britischer Physiker, Nobelpreis 1927

Literaturverzeichnis

[1] A. Müller, E. Leitner, W. Dilg, Physik. Leistungskurs 3. Semester, München, Ehrenwirth Verlag, 1989[7] S. 10, S. 164, S. 174 (Aufgabe 6)

[2] Berhag, Gross, atome kerne quanten, Stuttgart, Ernst Klett Verlag, 1987 S. 4, S. 12-15

[3] E. Segrè, Die großen Physiker und ihre Entdeckungen 2, München, R. Piper & Co. Verlag, 1990[4] S. 114-118

[4] H. Daniel, Physik 1. Mechanik · Wellen · Wärme, Berlin, Walter de Gruyter, 1997 S. 135

[5] G. Thomson, Das Atom, (dt.: H. Krappatsch), Göttingen, Musterschmidt-Verlag, o. J. S. 75

[6] F. Bitter, Currents, fields and particles, Cambridge (MA), Massachusetts Institute of Technology Press, 1956 S. 84ff

[7] H. Lindner, Grundriß der Atom- und Kernphysik, Leipzig, Fachbuchverlag, 1993[17] S. 90f

[8] W. Kuhn, J. Seibert, Ernest Rutherford und seine großen Entdeckungen, in: Praxis der Naturwissenschaften Physik, 15. Juni 1981, Heft 6, S. 161-166

[9] H. Hammer, K. Hammer, Physikalische Formeln und Tabellen, München, J. Lindauer Verlag (Schaefer), 1994[6] S. 9, S. 12, S. 40, S. 74-77

[10] H. Dittmann, H. Jodl, programm ideen physik, München, Bayerischer Schulbuch-Verlag, 1984[1] S. 30, S. 50

[11] F. Barth, P. Mühlbauer, F. Nikol, K. Wörle, Mathematische Formeln und Definitionen, München, Bayerischer Schuluch-Verlag & J. Lindauer Verlag (Schaefer), 1994[6] S. 26f

[12] H. Kopp, Graphische Datenverarbeitung, München/Wien, Carl Hanser Verlag, 1989 S. 9

[13] E. Anger (red. Leitung), Der Brockhaus in drei Bänden, Mannheim, F. A. Brockhaus, 1991

[14] Hilfetexte des Programms Turbo Pascal® 7.0 von Borland International

[15] E. Kaier, Turbo Pascal Wegweiser, Braunschweig/Wiesbaden, Friedr. Vieweg & Sohn Verlagsgesellschaft, 1993

[16] S. Letzel, R. Meyer, MASM: der Makro-Assembler von Microsoft, Bonn, International Thompson Publishing, 1994[1]

[17] C. Edwards, Turbo Pascal Profibuch, Düsseldorf, SYBEX-Verlag, 1987

Erklärung

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

Rosenheim, den 2. Februar 1998 .....

(Unterschrift des Kollegiaten)