

# Deckblatt zum Versuch: \_\_\_\_\_

Angaben zum Experiment	
Name: _____	<div style="border-top: 1px solid black; margin-top: 20px;"> Stempel/ Tutor-Unterschrift / Punkte </div>
Gruppennummer: _____	
Tutor: _____	
Datum der Abgabe: _____	

Die nachstehende Checkliste bildet die Anforderungen an den Bericht im PhysikPraktikum ab.

Allgemeines	Ja	Nein	NA
• Abgabe des Berichts erfolgt pünktlich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Äußere Form des Berichts ist angemessen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Messdaten liegen dem Bericht bei	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Jede gedruckte Seite enthält Namen und Gruppennummer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Eine Nachbesserung ist erforderlich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Strukturierung und Dokumentation	Ja	Nein	NA
• Gliederung des Berichts ist übersichtlich	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Rechenwege zur Ermittlung des Ergebnisses sind nachvollziehbar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Unsicherheiten wurden richtig ermittelt (Fehlerforpflanzung)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Alle quantitativen Ergebnisse enthalten Angaben zur Messunsicherheit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Messunsicherheiten und Ergebnisse werden diskutiert	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Graphische Darstellung	Ja	Nein	NA
• Bildunterschriften sind aussagekräftig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Vollständige Bezeichnung der Achsen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Sinnvolle Skalierung der Achsen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Bei Fit-Analyse: Angabe aller relevanten Parameter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• Darstellung der Messunsicherheiten mittels Fehlerbalken	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Anmerkungen



## Zusammenfassung

Während des Anfängerpraktikumsseminars haben wir uns mit verschiedenen Wahrscheinlichkeitsverteilungen, wie die Binomial-, Normal- oder Poissonverteilung, auseinandergesetzt. In diesem Bericht untersuchen wir ein Zufallsexperiment auf seine Zufallsverteilung.

## Inhaltsverzeichnis

<b>1</b>	<b>Das Zufallsexperiment</b>	<b>2</b>
1.1	Methode der Durchführung . . . . .	2
<b>2</b>	<b>Auswertungen</b>	<b>3</b>
2.1	Diagramme . . . . .	3
2.1.1	Ein Strahlteiler . . . . .	3
2.1.2	Drei Strahlteiler . . . . .	4
2.1.3	Fünf Strahlteiler . . . . .	4
2.2	Analyse . . . . .	5
2.3	Zusammenfassung . . . . .	6
<b>3</b>	<b>Anhang</b>	<b>7</b>
3.1	Code . . . . .	7

# 1 Das Zufallsexperiment

Das Zufallsexperiment, dass wir betrachten, ist das Auftreffen von Photonen auf Strahlteiler verschiedener Anzahl mit verschiedenen Durchlasswahrscheinlichkeiten. Hierbei nehmen wir an dass immer acht Photonen ausgelöst werden und in unser Experiment gelangen. Nun lassen wir die Photonen hintereinander auf die Strahlteiler treffen und untersuchen wie viele Photonen durch alle Strahlteiler gelangt sind. Hierbei ist die Durchlasswahrscheinlichkeit für ein Photon bei allen Strahlteilern gleich. Diesen Versuch führen wir nun für einen, drei und fünf Strahlteiler durch mit Durchlasswahrscheinlichkeiten von ein Viertel, ein Halb und Dreiviertel und das jeweils 500.000.000 Mal und untersuchen dabei die Verteilung der am Ende von allen Strahlteilern durchgelassenen Photonen.

## 1.1 Methode der Durchführung

Wir haben diesen Versuch mithilfe von Java simuliert. Im Anhang ist der Java-Code vorzufinden den wir zur Simulation benutzt haben.

Unserer Programm zur Simulation des Zufallsexperimentes benötigt als Argumente als erstes die Anzahl der Photonen, in unserem Fall acht, dann die Anzahl der Strahlteiler, die Durchlasswahrscheinlichkeit und die Anzahl der Durchführungen des Experimentes, sowie wahlweise den Dateinamen mit Dokumententyp unter dem die Daten abgespeichert werden soll. In diesem Fall werden die Daten nicht ausgegeben, sondern nur als Dokument gespeichert.

Hierbei lassen wir das Programm als erstes testen ob genug Argumente vorhanden sind, sonst gibt es eine Fehlermeldung aus. Als nächstes initialisieren wir die Variablen unseres Programmes als ganze Zahlen beziehungsweise die Wahrscheinlichkeit als Kommazahl. Zusätzlich definieren wir uns die Liste absHaeufigkeit die um eins länger ist als wir Photonen in unser Experiment schicken und die wir benutzen um unsere absoluten Häufigkeiten zu erhalten.

Dann verwenden wir eine For-Schleife die von Null bis zur Anzahl der Durchführungen läuft und nPhotonen gleich unserer Startphotonenzahl setzt. Nun wenden wir in einer inneren For-Schleife, die die Anzahl der Strahlteiler durchläuft, eine Funktion Strahlenteiler auf nPhotonen an und erhalten als Output dieser For-Schleife die Anzahl der durchgelassenen Photonen von dieser Durchführung des Zufallsexperiments. Zum Abschluss erhöhen wir nun in der äußeren For-Schleife an der Stelle unserer Liste absHaeufigkeit, die gleich unserem nach der Durchführung der inneren For-Schleife erhaltenen nPhotonen ist, den Wert um Eins. Hierbei ist es hilfreich, dass Listen in Java mit dem nullten Element beginnen, wodurch wir als das nullte Listenelement das absolute Häufigkeit von Null durchgelassenen Photonen haben können etc. Dies führen wir nun so oft durch wie wir angegeben haben und erhalten so eine Liste mit unseren absoluten Häufigkeiten für unserer Zufallsexperiment.

Als nächstes testen wir mit einem If-Befehl, ob wir das Ergebnis als Dokument speichern sollen oder einfach ausgeben sollen, indem wir die Anzahl unserer Argumente untersuchen.

Als vorletztes definieren wir noch unsere bereits verwendete Funktion Strahlenteiler, wobei wir als erstes unseren Output sout Null setzen, dann durchlaufen wir mit einer For-Schleife alle Photonen die wir in den Strahlteiler schicken und generieren für jedes eine Zufallszahl zwischen Null und Eins und testen ob diese kleiner oder gleich unserer festgelegten Wahrscheinlichkeit ist. Wenn dem so ist erhöhen wir sout um Eins, sonst nicht. Zum Schluss definieren wir noch die Funktion, die unsere Output-Datei generiert. Hierbei durchlaufen wir mit einer For-Schleife unsere Liste absHaeufigkeit und schreiben diese in die entsprechende Zeile des Dokuments in der Form „durchgekommene Photonen“; „absolute Häufigkeit“.

## 2 Auswertungen

### 2.1 Diagramme

#### 2.1.1 Ein Strahlteiler

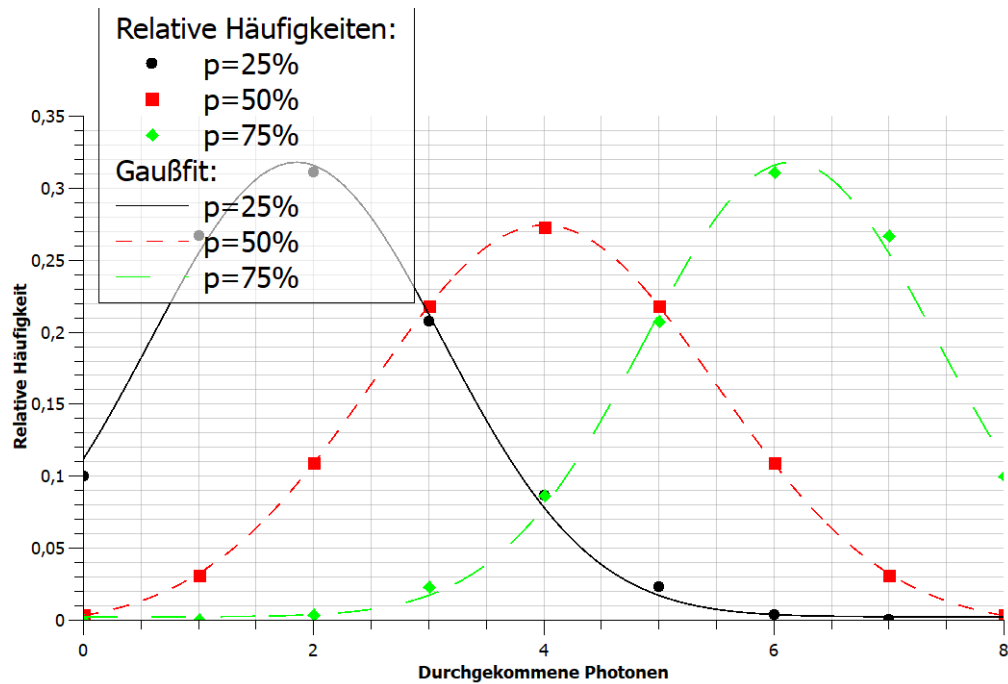


Abbildung 1: Dargestellt sind die relativen Häufigkeiten aufgetragen gegen die durchgekommenen Photonen, bei einem Strahlteiler. Es wurden 500 Millionen Versuche durchgeführt.

### 2.1.2 Drei Strahlteiler

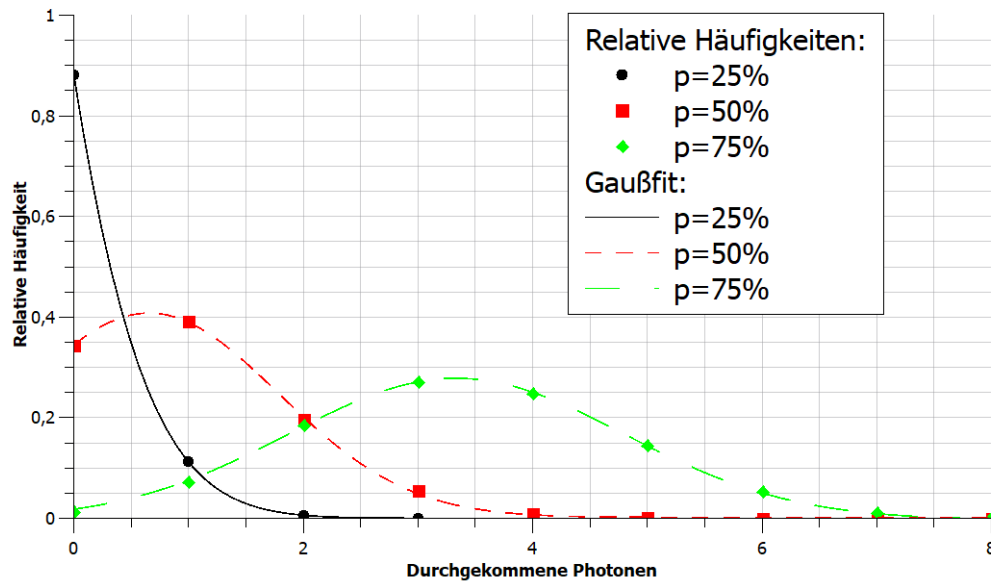


Abbildung 2: Dargestellt sind die relativen Häufigkeiten aufgetragen gegen die durchgekommenen Photonen, bei drei Strahlteilern. Es wurden 500 Millionen Versuche durchgeführt.

### 2.1.3 Fünf Strahlteiler

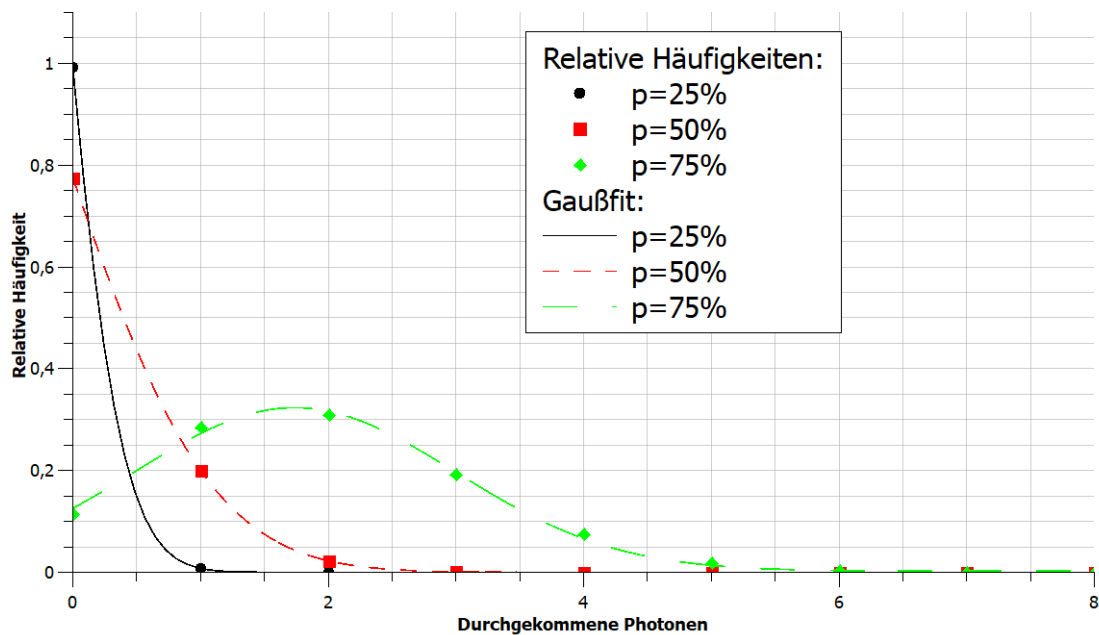


Abbildung 3: Dargestellt sind die relativen Häufigkeiten aufgetragen gegen die durchgekommenen Photonen, bei fünf Strahlteilern. Es wurden 500 Millionen Versuche durchgeführt.

## 2.2 Analyse

Dies sind die verwendeten Formel:

Binomialverteilung:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Normalverteilung:

$$P(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Poissonverteilung:

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

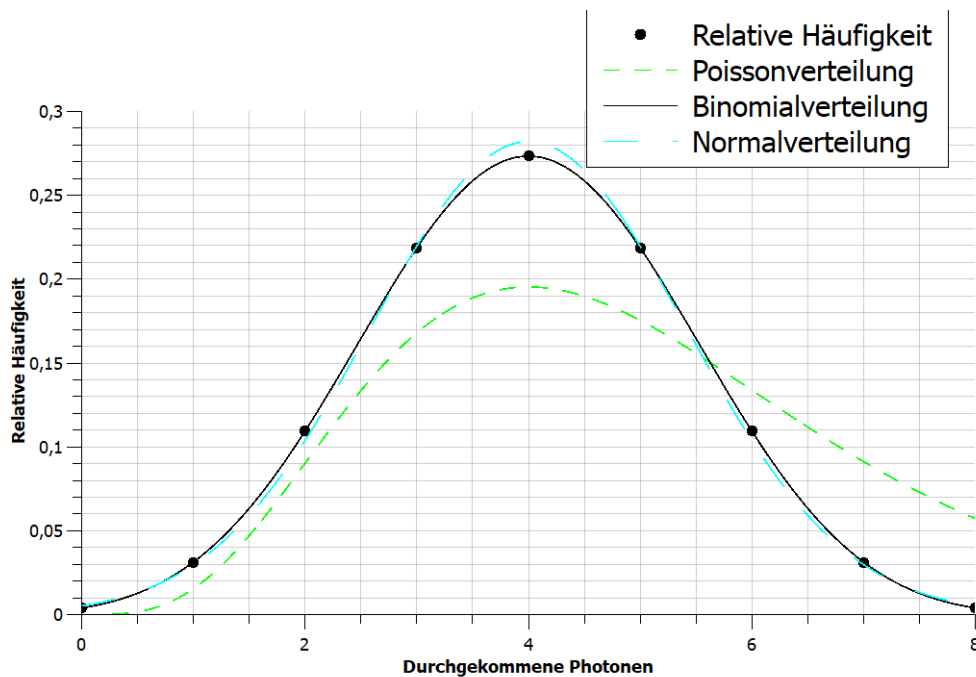


Abbildung 4: Relative Häufigkeit aufgetragen auf durchgekommene Photonen, angenähert durch Poisson-, Binomial- und Normalverteilung. Es wurde 1 Strahlteiler mit einer Durchlasswahrscheinlichkeit von 50% simuliert und 500 Millionen Versuche durchgeführt.

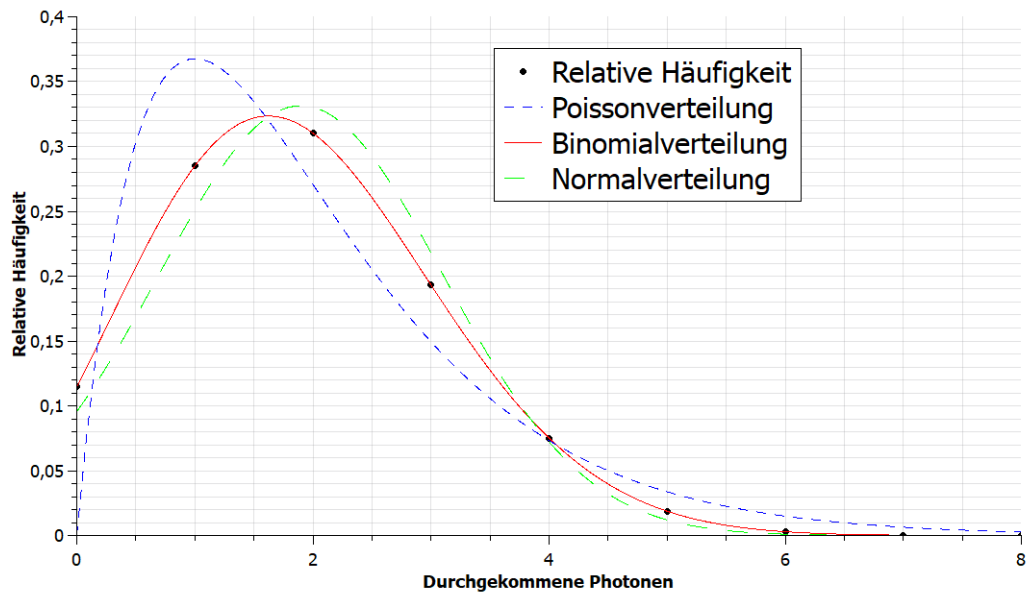


Abbildung 5: Relative Häufigkeit aufgetragen auf durchgekommene Photonen, angenähert durch Poisson-, Binomial- und Normalverteilung. Es wurden 5 Strahlteiler mit einer Durchlasswahrscheinlichkeit von 75% simuliert und 500 Millionen Versuche durchgeführt.

## 2.3 Zusammenfassung

## 3 Anhang

### 3.1 Code

---

```
package kuhn.domenik;

import java.io.FileWriter;
import java.io.IOException;

public class PhotonSim {

    public static void main(String[] args) {

        if(args.length<4){
            System.out.println("Syntax:");
            System.out.println("photonSim <StartPhotonen> <Anzahl Strahlenteiler> <Durchlass  

                Wahrscheinlichkei> <Anzahl Durchlaeufe> [Output File Name]");
            System.exit(0);
        }
        System.out.println();

        int runs = Integer.parseInt(args[3]);
        int startPhotonen = Integer.parseInt(args[0]);
        int nStrahlenTeiler = Integer.parseInt(args[1]);
        double p = Double.parseDouble(args[2]);
        int[] endPhotonen = new int[runs];
        int[] absHaeufigkeit = new int[startPhotonen+1];

        for(int i=0; i<runs; i++){
            int nPhotonen = startPhotonen;
            for(int k=0; k<nStrahlenTeiler; k++){
                nPhotonen = strahlenTeiler(nPhotonen,p);
            }
            endPhotonen[i]=nPhotonen;
            absHaeufigkeit[nPhotonen]++;
        }

        if(args.length==5){
            csvWriter(args[4],absHaeufigkeit);
        }else{
            System.out.println("Durchgekommene Photnen\t\tAbsolute Haeufigkeit");
            for(int i = 0; i<startPhotonen+1;i++){
                System.out.println(i+"\t\t\t"+absHaeufigkeit[i]);
            }
        }
    }

    public static int strahlenTeiler(int photonen, double p){
        int sout=0;
        for(int i=0; i<photonen;i++){
            if(Math.random()<=p)sout++;
        }
        return sout;
    }
}
```



Author1: Domenik Kuhn 10011073

Author2: Finn Feuerstarke 10011078

Gruppe: 220

---

```
public static void csvWriter(String path, int[] data){
    try {
        FileWriter fw = new FileWriter(path);
        for(int i=0; i<data.length; i++){
            fw.write(i+";"+data[i]+"\\n");
        }
        fw.close();
    } catch (IOException e) {
    }
}
}
```

---