

Computerphysik

Hausarbeit 3

Friedrich Hübner 2897111

Fiona Paulus 2909625

25. Juni 2017

Aufgabe 1

Aufgabenteil (a)

Rodrigues-Formel:

$$P_n(x) = \frac{1}{2^n n!} * \frac{d^n}{dx^n} [(x^2 - 1)^n]$$

Daraus ergibt sich für $P_4(x)$:

$$\begin{aligned} P_4(x) &= \frac{1}{2^4 4!} \frac{d^4}{dx^4} ((x^2 - 1)^4) = \frac{1}{2^4 4!} \frac{d^4}{dx^4} (x^8 - 4x^6 + 6x^4 - 4x^2 + 1) \\ &= \frac{1}{384} (1680x^4 - 1440x^2 + 144) = \frac{1}{8} (35x^4 - 30x^2 + 3) \end{aligned}$$

Aufgabenteil (b)

Rekursionsformel:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

Daraus ergibt sich für $P_4(x)$:

$$4P_4(x) = (3+1)P_{3+1} = 7xP_3(x) - 3P_{3-1}(x) = 7x \frac{5x^3}{3x} - 3 \frac{1}{2} (3x^2 - 1) = \frac{1}{4} \left(\frac{1}{8} (35x^4 - 30x^2 + 3) \right)$$

$$\Rightarrow P_4(x) = \frac{1}{8} (35x^4 - 30x^2 + 3)$$

Aufgabenteil (c)

Nullstellenberechnung durch Substitution von $x^2 = z$

$$0 = 35x^4 - 30x^2 + 3$$

$$\Rightarrow 0 = 35z^2 - 30z + 3$$

$$\Leftrightarrow 0 = z^2 - \frac{30}{35}z + \frac{3}{35} \Leftrightarrow z_{1,2} = -\frac{-\frac{6}{7}}{2} \pm \sqrt{\left(\frac{3}{7}\right)^2 - \frac{3}{35}} = \frac{3}{7} \pm \sqrt{\frac{24}{245}}$$

Durch Resubstitution:

$$\begin{aligned}x_0 &= -\sqrt{\frac{3}{7} + \sqrt{\frac{24}{245}}} \\x_1 &= -\sqrt{\frac{3}{7} - \sqrt{\frac{24}{245}}} \\x_2 &= +\sqrt{\frac{3}{7} - \sqrt{\frac{24}{245}}} = -x_1 \\x_3 &= +\sqrt{\frac{3}{7} + \sqrt{\frac{24}{245}}} = -x_0\end{aligned}$$

Berechnung der Wichtungskoeffizienten:

$$\begin{aligned}c_i &= \int_{-1}^1 l_i(x) dx \\c_0 &= \int_{-1}^1 \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} dx \\&\Rightarrow \int_{-1}^1 \frac{(x-x_1)(x+x_1)(x+x_0)}{(x_0-x_1)(x_0+x_1)(2x_0)} dx = \int_{-1}^1 \frac{(x^2-x_1^2)(x+x_0)}{(x_0^2-x_1^2)2x_0} dx = \frac{1}{(x_0^2-x_1^2)2x_0} \left(\frac{2}{3}x_0 - 2x_1^2x_0 \right) \\&= \frac{\frac{1}{3} - x_1^2}{x_0^2 - x_1^2} = \frac{\frac{1}{3} - \frac{3}{7} + \sqrt{\frac{24}{245}}}{\frac{3}{7} + \sqrt{\frac{24}{245}} - \frac{3}{7} + \sqrt{\frac{24}{245}}} = \frac{18 - \sqrt{30}}{36} \\c_1 &= \int_{-1}^1 \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)2x_1(x_1+x_0)} dx = \frac{\frac{1}{3} - x_0^2}{x_1^2 - x_0^2} = \frac{18 + \sqrt{30}}{36} \\c_2 &= \int_{-1}^1 \frac{(x+x_3)(x+x_2)(x+x_0)}{(-x_1+x_3)(x_2-x_1)(-x_1+x_0)} dx\end{aligned}$$

Änderung der Variable $x = -t$

$$\begin{aligned}&= \int_{-1}^1 \frac{(t-x_3)(t-x_2)(t-x_0)}{(x_1-x_3)(x_1-x_2)(x_1-x_0)} dt = c_1 \\c_3 &= \int_{-1}^1 \frac{(x+x_3)(x+x_2)(x+x_1)}{(x_3-x_0)(-x_0+x_2)(-x_0+x_1)} dx\end{aligned}$$

Änderung der Variable $x = -t$

$$= \int_{-1}^1 \frac{(t-x_3)(t-x_2)(x-x_1)}{(x_0-x_3)(x_1-x_2)(x_0-x_1)} dt = c_0$$

0.1 Aufgabenteil (d)

Berechnung mithilfe der Legendre-Polynome:

$$\begin{aligned} \int_{-1}^1 e^{-x} &\approx \sum_{i=0}^3 c_i \cdot e^{-x_i} \\ &= \frac{18-\sqrt{30}}{36} * e^{-\sqrt{\frac{3}{7}+\sqrt{\frac{24}{245}}}} + \frac{18+\sqrt{30}}{36} * e^{-\sqrt{\frac{3}{7}-\sqrt{\frac{24}{245}}}} + \frac{18+\sqrt{30}}{36} * e^{\sqrt{\frac{3}{7}-\sqrt{\frac{24}{245}}}} + \frac{18-\sqrt{30}}{36} * e^{\sqrt{\frac{3}{7}+\sqrt{\frac{24}{245}}}} \\ &= 2.350402092156377 \end{aligned}$$

Berechnung durch direkte Integration:

$$\int_{-1}^1 e^{-x} dx = e - \frac{1}{e} = 2.3504023872876028$$

Berechnung des Fehlers:

$$\Delta = 2.3504023872876028 - 2.350402092156377 = 2.95 \cdot 10^{-7}$$

Berechnung des Fehlers mithilfe der Legendre Polynome

$$\begin{aligned} \Delta &= \int_{-1}^1 f(x) dx - \sum_{i=0}^n f(x_i) = \frac{2^{2n+3}((n+1)!)^4}{((2n+2)!)^3(2n+3)} f^{(2n+2)}(\xi) \\ &= \frac{2^{2*3+3}((3+1)!)^4}{((2*3+2)!)^3(2*3+3)} \frac{d^8}{dx^8} e^{-\xi} \end{aligned}$$

Für $\xi \in [-1, 1] \Rightarrow \xi = -1$ eingesetzt:

$$= \frac{2^9 4!^4}{8!^3 9} e = 7.8271802 \cdot 10^{-7}$$

Aufgabe 2

allgemeine Hinweise

Das Programm zur Aufgabe 2 wurde unter Linux Mint mit "g++ -std=c++11 -g -Wall -Wextra -pedantic-errors pendelschwingdauer.cpp -o pendelschwingdauer.exe" kompiliert.

Lösung

Zur bestimmung der Abweichung der Schwingungsdauer T vom harmonischen Wert als Funktion von ϕ_0 wird mit dem Newton-Cotes-Verfahren numerisch integriert ("pendelschwingdauer.cpp"). Dazu werden die Simpson-Koeffizienten verwendet und in die Newton-Cotes-Formel eingesetzt.

$$\int_a^b f(x)dx \approx \frac{(b-a)}{n} \sum_{i=0}^n \omega_i f(x_i)$$

Wobei ω_i die Simpson Koeffizienten sind und die x_i die zugehörigen Stützstellen. Diese Rechnung wird für ϕ_0 zwischen 0° und 180° wiederholt und in ihr Ergebnis in der Datei "data2.txt" gespeichert.

Das Ergebnis wird mit der Datei "2.plt" geplottet.

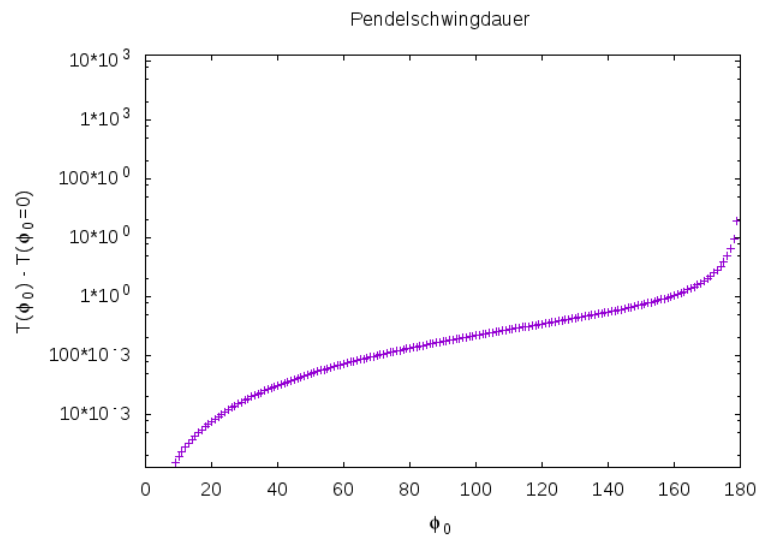


Abbildung 0.1: Abweichung der Pendelschwingdauer T/s vom harmonischen Wert in Abhängigkeit der Auslenkung $\phi_0/^\circ$

Sonstige Abgegebene Dateien

2.plt

Die Plot-Datei für den Plots in der Abgabe.

data2.txt

Ausgabedatei des Programms "pendelschwingdauer.cpp".

Aufgabe 3

allgemeine Hinweise

Das Programm zur Aufgabe 3 wurde unter Linux Mint mit `"g++ -std=c++11 -g -Wall -Wextra -pedantic-errors planckkurve.cpp -o planckkurve.exe"` kompiliert.

Aufgabenteil (a)

Es wurde die Planckkurve $S(\lambda)$ geplottet für die Temperaturen $T = 100K$, $T = 500K$ und $T = 1000K$. ("3a.plt", "3a-.plt").

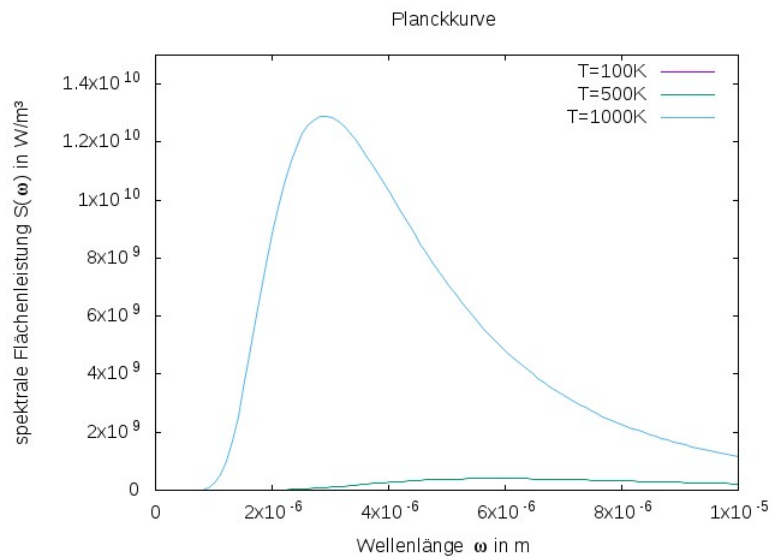


Abbildung 0.2: Planckkurven

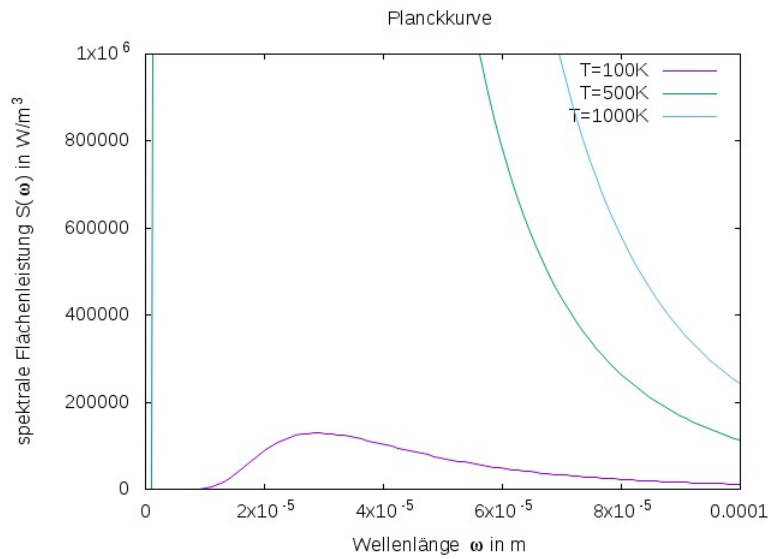


Abbildung 0.3: Planckkurve Ausschnitt

Aufgabenteil (b)

Um den Energieanteil des sichtbaren Lichts als Funktion der Temperatur zu bestimmen, wird über die Energieverteilung $S(\lambda)$ numerisch ("planckkurve.cpp") mit dem Newton-Cotes-Verfahren integriert. Hierzu werden die Milne-Koeffizienten als ω_i in die obige Newton-Cotes-Formel eingesetzt.

$$E(T) = \frac{\int_{380 \cdot 10^{-9}}^{780 \cdot 10^{-9}} S(\lambda) d\lambda}{\int_0^{\infty} S(\lambda) d\lambda} = \frac{1}{\frac{dP}{dA}} \int_{380 \cdot 10^{-9}}^{780 \cdot 10^{-9}} S(\lambda) d\lambda$$

Dies wird für Temperaturen zwischen 100K und 50000K in 200K Schritten wiederholt und die daraus errechneten Werte werden in "data3.txt" gespeichert. Diese Werte werden mit "3b.plt" geplottet.

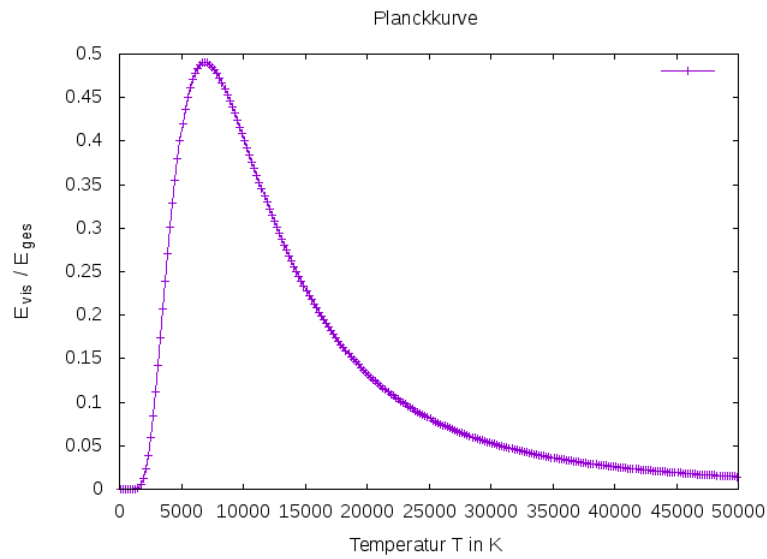


Abbildung 0.4: Energieanteil des sichtbaren Lichts als Funktion der Temperatur

Sonstige Abgegebene Dateien

3a.plt, 3a-.plt, 3b.plt

Die Plotdateien zur Erzeugung der Planckfunktionen und der Funktion zur Abhängigkeit des Energieanteils des sichtbaren Bereichs in Abhängigkeit der Temperatur.

data3.txt

Ausgabedatei des Programms "planckkkurve.cpp".

Aufgabe 4

Allgemeine Hinweise

Das Programm wurde unter Windows 10 mit "g++ -o abgabe3_4.exe -Wall -Wextra -std=c++0x -O2 -static abgabe3_4.cpp" kompiliert.

Das Programm wird gestartet mit 'abgabe3_4 $H_0 \Omega_\Lambda \Omega_m \Omega_r$ ', wobei H_0 in km/s/Mpc ist.

Idee

Da die eine Integralgrenze ∞ ist, kann man die Funktion nicht mit den normalen Methoden integrieren. Deswegen wird zuerst eine Substitution $a = \frac{1}{1+z}$ durchgeführt.

$$t(z) = \frac{1}{H_0} \int_z^\infty \frac{dz'}{(1+z')\sqrt{E(\frac{1}{1+z})}} \quad (0.1)$$

$$= \frac{1}{H_0} \int_{\frac{1}{1+z}}^0 \frac{a'}{\sqrt{E(a')}} - \frac{1}{a'^2} da' \quad (0.2)$$

$$= \frac{1}{H_0} \int_0^a \frac{1}{a' \sqrt{E(a')}} da' \quad (0.3)$$

$$= \frac{1}{H_0} \int_0^a f(a') da' \quad (0.4)$$

Da heute $z = 0$ ist, liegt $z \in [0, \infty)$. Somit liegt $a \in [0, 1)$. Dieses Integral kann nun mit einer der bekannten Methoden berechnet werden.

Da die zu integrierende Funktion bei $a'=0$ nicht definiert ist, muss man noch den Grenzwert für $a' \rightarrow 0$ berechnen:

$$\lim_{a' \rightarrow 0} a'^2 E(a') = \lim_{a' \rightarrow 0} a'^2 \Omega_\Lambda + \Omega_m/a' + \Omega_r/a'^2 + \Omega_k = \infty \text{ für } \Omega_m \neq 0, \Omega_r \neq 0, \text{ sonst } \Omega_k.$$

Damit gilt für $\lim_{a' \rightarrow 0} f(a') = 0$ für $\Omega_m \neq 0, \Omega_r \neq 0$, sonst $1/\sqrt{\Omega_k}$.

Programm

Das Integral wurde mit der (zusammengesetzten) Trapezformel berechnet. Man kann das Ergebnis $t(a)$ wiederverwenden, um $t(a + \Delta a) = t(a) + \frac{1}{H_0} \int_a^{a+\Delta a} f(a') da'$ zu berechnen. Im Programm selber wird zuerst nur die Stammfunktion von f numerisch bestimmt und in einem Array gespeichert. Zur Ausgabe wird linear zwischen den zwei nächsten ermittelten Werten interpoliert und dieser Wert durch H_0 geteilt.

Beispiele

Hier sind die in Aufgabe b) und c) geforderten Diagramme. Beide Kurven wurden in ein Diagramm gezeichnet, das einmal normal und einmal logarithmisch dargestellt wurde.

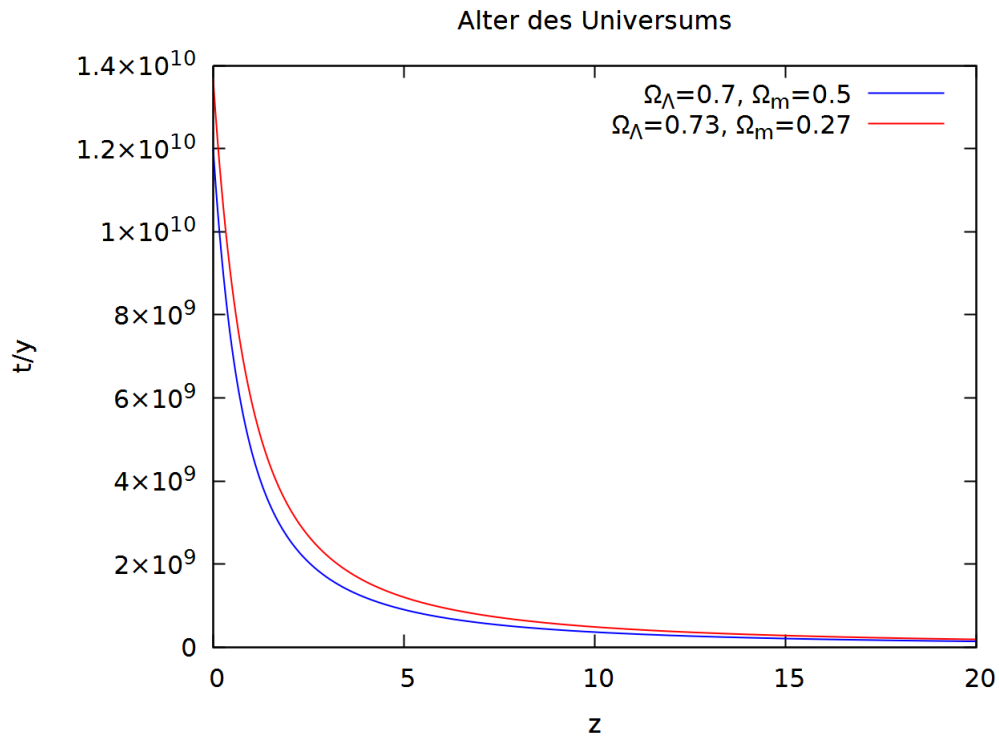


Abbildung 0.5: Normaler Plot

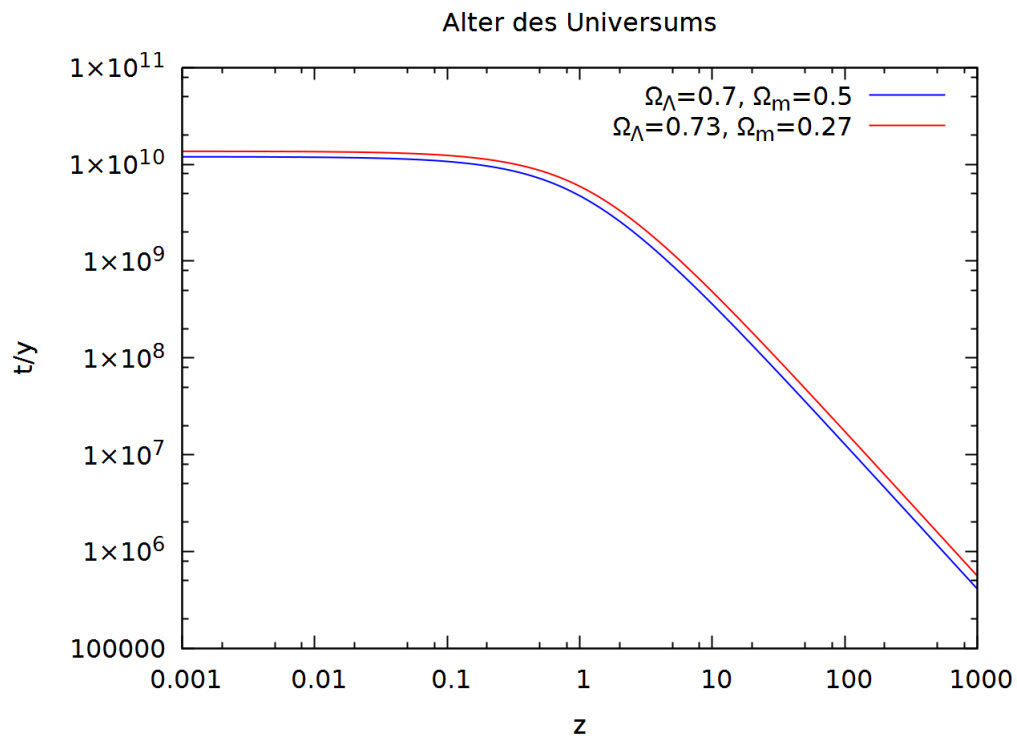


Abbildung 0.6: Logarithmischer Plot

Sonstige abgegebene Dateien

plot_final.plt

Die Plot-Datei für die beiden Plots in der Abgabe.

b.txt

Enthält die Plotdaten für b)

c.txt

Enthält die Plotdaten für c)

Aufgabe 5

Allgemeine Hinweise

Zu a):

Das Programm wurde unter Windows 10 mit "g++ -o abgabe3_5_bessel.exe -Wall -Wextra -std=c++0x -O2 -static abgabe3_5_bessel.cpp" kompiliert.

Das Programm wird gestartet mit 'abgabe3_5_bessel n l r', wobei $n \in \mathbb{N}$ der Parameter der Besselfunktion ist und l, r das Intervall, auf dem die Besselfunktion berechnet werden soll.

Zu b):

Das Programm wurde unter Windows 10 mit "g++ -o abgabe3_5.exe -Wall -Wextra -std=c++0x -O2 -static abgabe3_5.cpp" kompiliert.

Das Programm wird gestartet mit 'abgabe3_5'.

a) (abgabe3_5_bessel.cpp)

Der Funktionswert der Besselfunktion wird mithilfe der zusammengesetzten Trapezformel berechnet. Im Intervall (l, r) wird an 1000 gleichverteilten x -Werten der Funktionswert berechnet und ausgegeben.

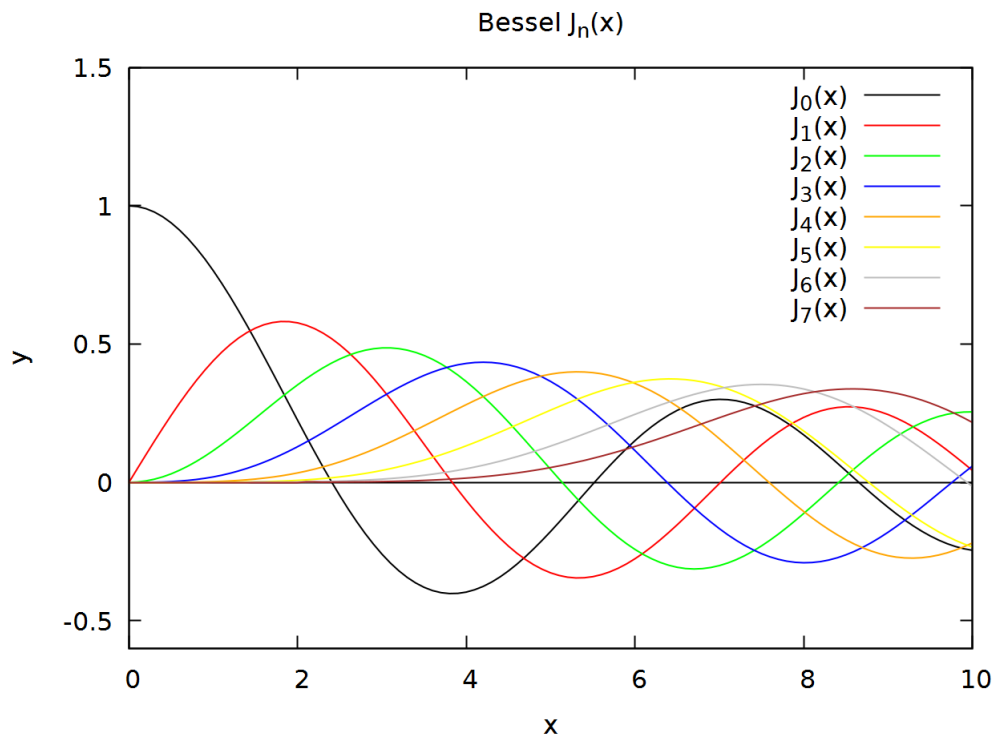


Abbildung 0.7: Die ersten 8 Besselfunktionen

b) (abgabe3_5.cpp)

Zur Berechnung der Besselfunktion wird der Algorithmus aus a) verwendet. Es wird einfach die Formel in der Aufgabenstellung bei mehreren äquidistanten θ ausgewertet. Die Ausgabe erfolgt in slit.txt

Dabei ergibt sich folgendes Bild:

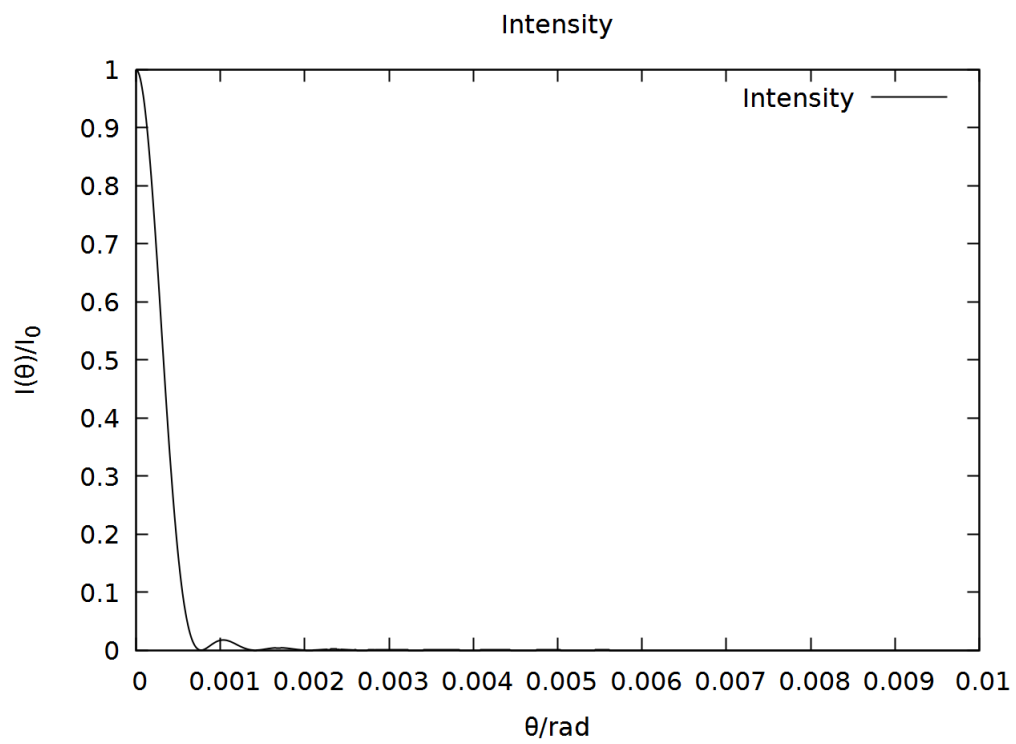


Abbildung 0.8: Beugungsbild an einer Kreisblende

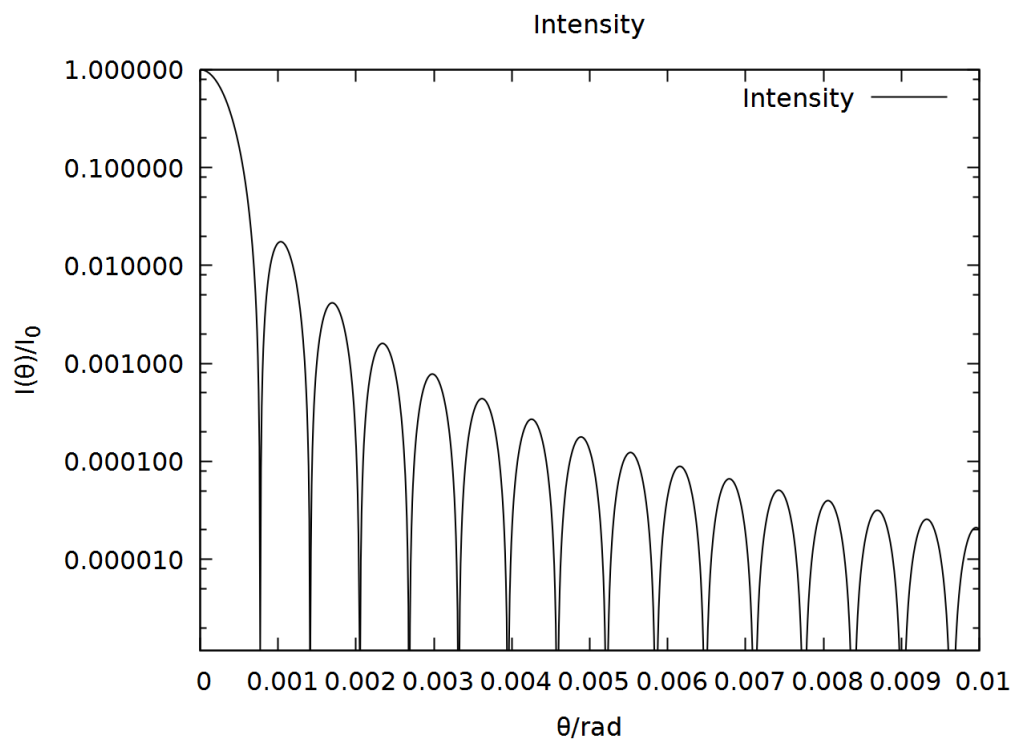


Abbildung 0.9: Beugungsbild an einer Kreisblende (logarithmisch)

c) (abgabe3_5.cpp)

Im selben Programm werden auch noch Maxima und Minima bestimmt. Dazu werden alle Werte durchgegangen. Ist ein Wert sowohl größer als der Vorgänger als auch der Nachfolger, so ist an dieser Stelle ein Maximum. Für Minima ist es analog. Man speichert sich zunächst nur den Index des Punktes, an dem die Extremstelle vorliegt und berechnet später daraus den Winkel.

Dabei ergeben sich folgende Maxima und Minima:

Tabelle 0.1: Maxima

n	θ/rad	$I(\theta)/I_0$
0	0	1
1	0.00103103	0.0174927
2	0.00169169	0.00415657
3	0.00234234	0.0016005
4	0.00298298	0.000779314
5	0.00361361	0.000436851
6	0.00425425	0.000269287
7	0.00488488	0.000177494
8	0.00552553	0.00012319
9	0.00615616	8.89594e-005
10	0.00678679	6.6293e-005

Tabelle 0.2: Minima

n	θ/rad
1	0.000770771
2	0.00141141
3	0.00205205
4	0.00268268
5	0.00331331
6	0.00395395
7	0.00458458
8	0.00521522
9	0.00584585
10	0.00648649

Sonstige abgegebene Dateien

plot_bessel.plt

Die Plot-Datei für die Besselfunktionen

plot_slit.plt

Die Plot-Datei für die Kreisblende

slit.txt

Enthält die Plotdaten für die b)