Auswertung

March 12, 2024

0.0.1 B 3.3 – Reichweite von α-Teilchen

[1]:
```julia
using Plots
using LaTeXStrings
```

Vorbereitung: Abschätzung Anzahl Stöße der α-Teilchen

[2]:
```julia
5.486*10^6/14.53
```

[2]:
```
377563.66139022715
```

Auswertungsteil – 4.2 Reichweiten

1. Zählratenkurven, 2. gegen Impulshöhe 0 extrapolierte Impulshöhenkurven:

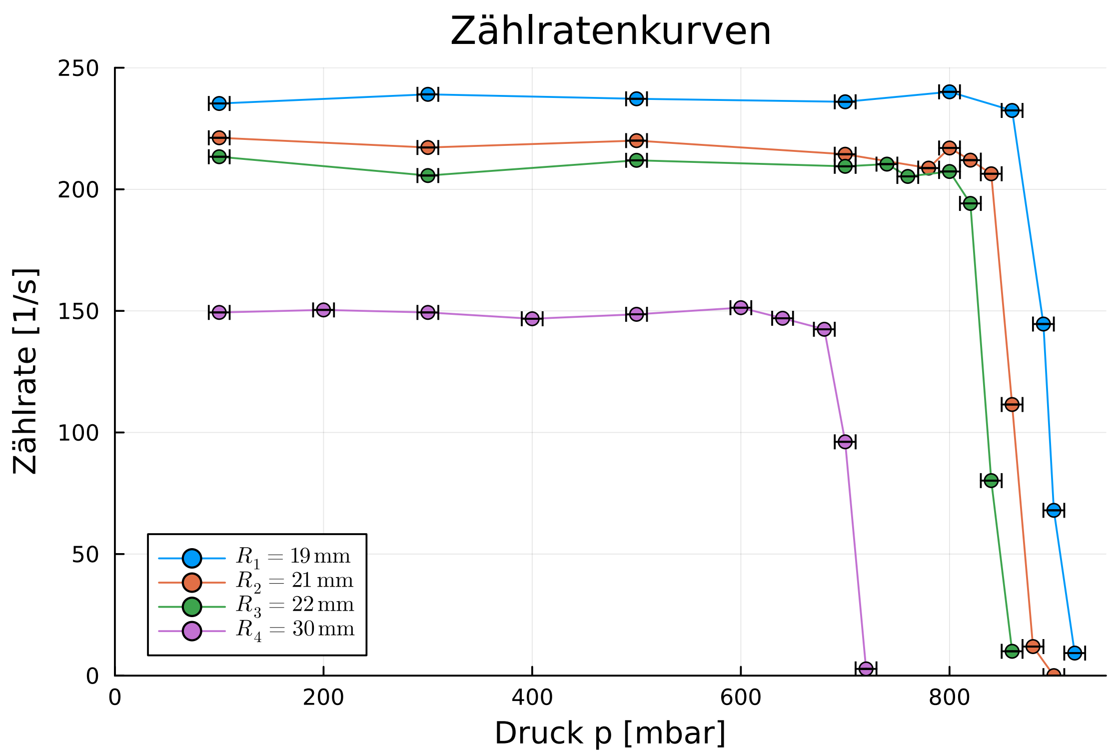
[3]:
```julia
Δp = 10 # mbar
# 1. Abstand R_1 = 19 mm
p1 = [100, 300, 500, 700, 800, 860, 890, 900, 920] # mbar
rate1 = [7059/30, 7172/30, 7117/30, 7080/30, 7203/30, 6973/30, 8676/60, 6118/
 ↪90, 1113/120] # 1/s
Δrate1 = [1/sqrt(7059), 1/sqrt(7172), 1/sqrt(7117), 1/sqrt(7080), 1/sqrt(7203),␣
 ↪1/sqrt(6973), 1/sqrt(8676),
    1/sqrt(6118), 1/sqrt(1113)]
spannung1 = [7.4, 6.2, 4.8, 3.2, 2.3, 1.5, 1.1, 0.9, 0.4] # V
Δspannung1 = [0.3, 0.2, 0.2, 0.3, 0.4, 0.4, 0.4, 0.5, 0.4] # V

# 2. Abstand R_2 = 21 mm
p2 = [100, 300, 500, 700, 780, 800, 820, 840, 860, 880, 900] # mbar
rate2 = [6636/30, 6516/30, 6601/30, 6433/30, 6262/30, 6510/30, 6360/30, 6192/
 ↪30, 6690/60, 1434/120, 0/20] # 1/s
Δrate2 = [1/sqrt(6636), 1/sqrt(6516), 1/sqrt(6601), 1/sqrt(6433), 1/sqrt(6262),␣
 ↪1/sqrt(6510), 1/sqrt(6360),
    1/sqrt(6192), 1/sqrt(6690), 1/sqrt(1434), 0]
spannung2 = [7.3, 6.1, 4.5, 2.7, 1.8, 1.0, 1.2, 0.9, 0.5, 0.4, 0.2] # V
Δspannung2 = [0.3, 0.3, 0.3, 0.4, 0.3, 0.6, 0.4, 0.4, 0.4, 0.3, 0.2] # V

# 2. Abstand R_2 = 22 mm
p3 = [100, 300, 500, 700, 740, 760, 800, 820, 840, 860] # mbar
```

```julia
rate3 = [6402/30, 6170/30, 6358/30, 6284/30, 6311/30, 6159/30, 6220/30, 5826/
 ↪30, 4812/60, 1201/120] # 1/s
Δrate3 = [1/sqrt(6402), 1/sqrt(6170), 1/sqrt(6358), 1/sqrt(6284), 1/sqrt(6311),␣
 ↪1/sqrt(6159), 1/sqrt(6220),
    1/sqrt(5826), 1/sqrt(4812), 1/sqrt(1201)]
spannung3 = [7.2, 6.0, 4.6, 2.6, 2.2, 1.9, 1.3, 1.0, 0.6, 0.4] # V
Δspannung3 = [0.2, 0.2, 0.2, 0.2, 0.2, 0.3, 0.3, 0.3, 0.4, 0.3] # V

# 2. Abstand R_2 = 30 mm
p4 = [100, 200, 300, 400, 500, 600, 640, 680, 700, 720] # mbar
rate4 = [8962/60, 9024/60, 8962/60, 8806/60, 8916/60, 9079/60, 8818/60, 8547/
 ↪60, 5769/60, 332/120] # 1/s
Δrate4 = [1/sqrt(8962), 1/sqrt(9024), 1/sqrt(8962), 1/sqrt(8806), 1/sqrt(8916),␣
 ↪1/sqrt(9079), 1/sqrt(8818),
    1/sqrt(8547), 1/sqrt(5769), 1/sqrt(332)]
spannung4 = [7.2, 6.4, 5.6, 4.7, 3.8, 2.6, 1.9, 1.3, 1.0, 0.4] # V
Δspannung4 = [0.1, 0.2, 0.2, 0.2, 0.3, 0.2, 0.3, 0.3, 0.3, 0.3] # V
;
```

[4]:

```julia
plot(p1, rate1, xerr = Δp, yerr = Δrate1, label = L"R_1 = 19 \mathrm{\, mm}",␣
 ↪legend =:bottomleft, markershape=:circle,
    title = "Zählratenkurven", ylims = (0,250), xlims = (0,950))
plot!(p2, rate2, xerr = Δp, yerr = Δrate2, label = L"R_2 = 21 \mathrm{\, mm}",␣
 ↪markershape=:circle)
plot!(p3, rate3, xerr = Δp, yerr = Δrate3, label = L"R_3 = 22 \mathrm{\, mm}",␣
 ↪markershape=:circle)
plot!(p4, rate4, xerr = Δp, yerr = Δrate4, label = L"R_4 = 30 \mathrm{\, mm}",␣
 ↪markershape=:circle)

xlabel!("Druck p [mbar]")
ylabel!("Zählrate [1/s]")
```

[4]:

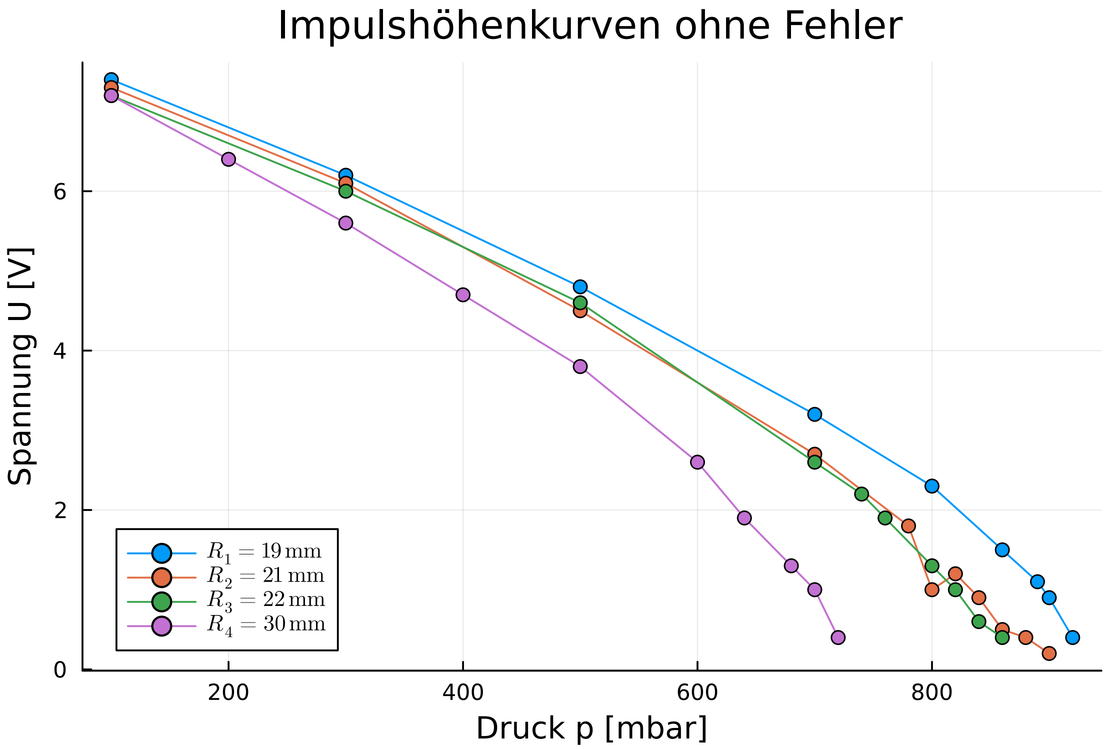


[5]:
```
#Savefig
zählratenkurven = plot(p1, rate1, xerr = Δp, label = L"R_1 = 19 \mathrm{\,␣
↪mm}", legend =:bottomleft,
    markershape=:circle, ylims = (0,250), xlims = (0,950))
plot!(p2, rate2, xerr = Δp, label = L"R_2 = 21 \mathrm{\, mm}", markershape=:
↪circle)
plot!(p3, rate3, xerr = Δp, label = L"R_3 = 22 \mathrm{\, mm}", markershape=:
↪circle)
plot!(p4, rate4, xerr = Δp, label = L"R_4 = 30 \mathrm{\, mm}", markershape=:
↪circle)

xlabel!("Druck p [mbar]")
ylabel!("Zählrate [1/s]")
savefig(zählratenkurven, "../../media/B3.3/zaehlratenkurven.svg");
savefig(zählratenkurven, "../../media/B3.3/zaehlratenkurven.pdf");
```

[6]:
```
plot(p1, spannung1, label = L"R_1 = 19 \mathrm{\, mm}", markershape=:circle,
    title = "Impulshöhenkurven ohne Fehler", legend=:bottomleft)
plot!(p2, spannung2, label = L"R_2 = 21 \mathrm{\, mm}", markershape=:circle)
plot!(p3, spannung3, label = L"R_3 = 22 \mathrm{\, mm}", markershape=:circle)
plot!(p4, spannung4, label = L"R_4 = 30 \mathrm{\, mm}", markershape=:circle)

xlabel!("Druck p [mbar]")
```

```
ylabel!("Spannung U [V]")
```

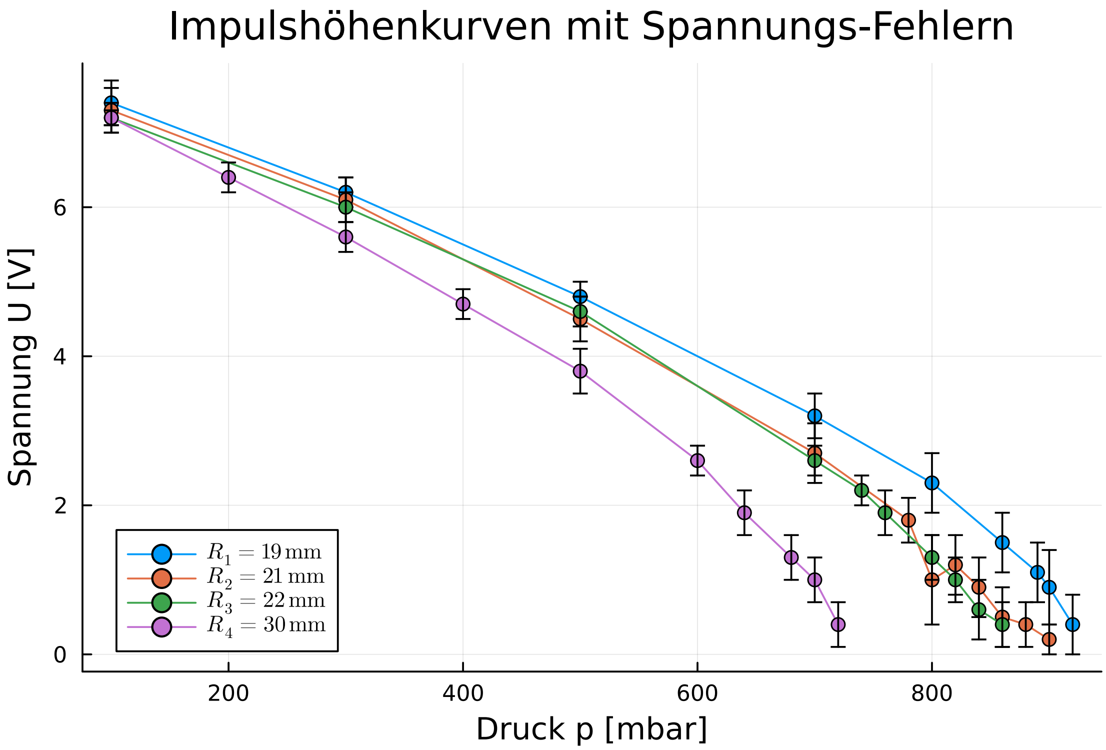
[6]:



[7]:

```
plot(p1, spannung1, yerr = Δspannung1, label = L"R_1 = 19 \mathrm{\, mm}",␣
↪markershape=:circle,
    title = "Impulshöhenkurven mit Spannungs-Fehlern", legend=:bottomleft)
plot!(p2, spannung2, yerr = Δspannung2, label = L"R_2 = 21 \mathrm{\, mm}",␣
↪markershape=:circle)
plot!(p3, spannung3, yerr = Δspannung3, label = L"R_3 = 22 \mathrm{\, mm}",␣
↪markershape=:circle)
plot!(p4, spannung4, yerr = Δspannung4, label = L"R_4 = 30 \mathrm{\, mm}",␣
↪markershape=:circle)

xlabel!("Druck p [mbar]")
ylabel!("Spannung U [V]")
```

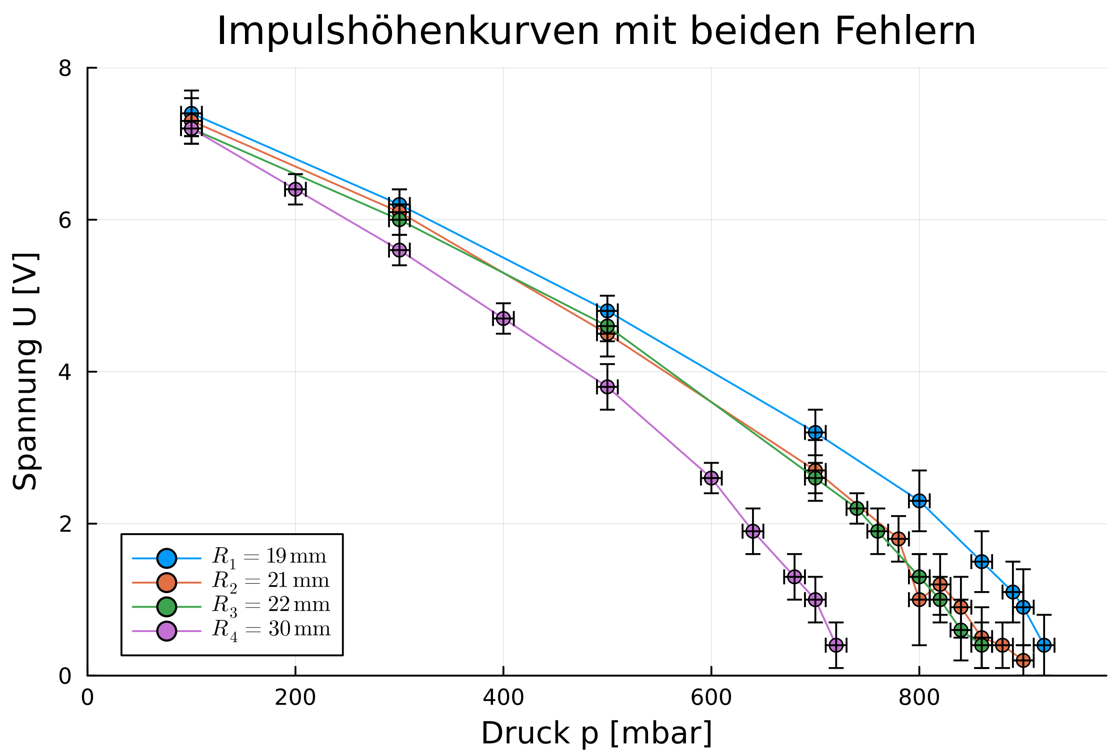
[7]:



[8]:

```
plot(p1, spannung1, xerr = Δp, yerr = Δspannung1, label = L"R_1 = 19 \mathrm{\,␣
↪mm}", markershape=:circle,
    title = "Impulshöhenkurven mit beiden Fehlern", legend=:bottomleft, ylims =␣
↪(0,8), xlims = (0,980))
plot!(p2, spannung2, xerr = Δp, yerr = Δspannung2, label = L"R_2 = 21␣
↪\mathrm{\, mm}", markershape=:circle)
plot!(p3, spannung3, xerr = Δp, yerr = Δspannung3, label = L"R_3 = 22␣
↪\mathrm{\, mm}", markershape=:circle)
plot!(p4, spannung4, xerr = Δp, yerr = Δspannung4, label = L"R_4 = 30␣
↪\mathrm{\, mm}", markershape=:circle)

xlabel!("Druck p [mbar]")
ylabel!("Spannung U [V]")
```

[8]:



[9]:
```
#savefig
spannungskurven = plot(p1, spannung1, xerr = Δp, yerr = Δspannung1, label =␣
↪L"R_1 = 19 \mathrm{\, mm}",
    markershape=:circle, legend=:bottomleft, ylims = (0,8), xlims = (0,980))
plot!(p2, spannung2, xerr = Δp, yerr = Δspannung2, label = L"R_2 = 21␣
↪\mathrm{\, mm}", markershape=:circle)
plot!(p3, spannung3, xerr = Δp, yerr = Δspannung3, label = L"R_3 = 22␣
↪\mathrm{\, mm}", markershape=:circle)
plot!(p4, spannung4, xerr = Δp, yerr = Δspannung4, label = L"R_4 = 30␣
↪\mathrm{\, mm}", markershape=:circle)

xlabel!("Druck p [mbar]")
ylabel!("Spannung U [V]")

savefig(spannungskurven, "../../media/B3.3/spannungskurven.svg");
savefig(spannungskurven, "../../media/B3.3/spannungskurven.pdf");
```

$\Rightarrow$ **$\bar{R}$ zu $1/\bar{p}$ Gerade:**

Nullstellen durch Extrapolation bestimmen => Reichweite zu inversem Druck Gerade

1. Aus Zählratenkurve, 2. Aus Impulshöhenkurve

Rechnerische Geradenanpassung:

[10]:
```julia
function rechnerischeGeradenanpassung(xValues, yValues)
    if length(xValues) != length(yValues)
        return 0
    elseif length(xValues) <= 2
        return 0
    else
        N = length(xValues)
    end

    x = 0 # [x]
    y = 0 # [y]
    xx = 0 # [xx]
    xy = 0 # [xy]

    for i in 1:N
        x += xValues[i]
        y += yValues[i]
        xx += xValues[i]^2
        xy += xValues[i] * yValues[i]
    end
    Δ = N * xx - x * x

    a = (N * xy - x * y) / Δ
    b = (xx * y - x * xy) / Δ

    Δy = 0
    for i in 1:N
        Δy += (a * xValues[i] + b - yValues[i])^2
    end
    Δy_hoch2 = Δy * 1 / (N-2)

    Δa = sqrt(Δy_hoch2 * N / Δ)
    Δb = sqrt(Δy_hoch2 * xx / Δ)

    return a,b, Δa, Δb
end
```

[10]:
```
rechnerischeGeradenanpassung (generic function with 1 method)
```

Nullstellen von 1. und 2. bestimmen durch Extrapolation:

1.

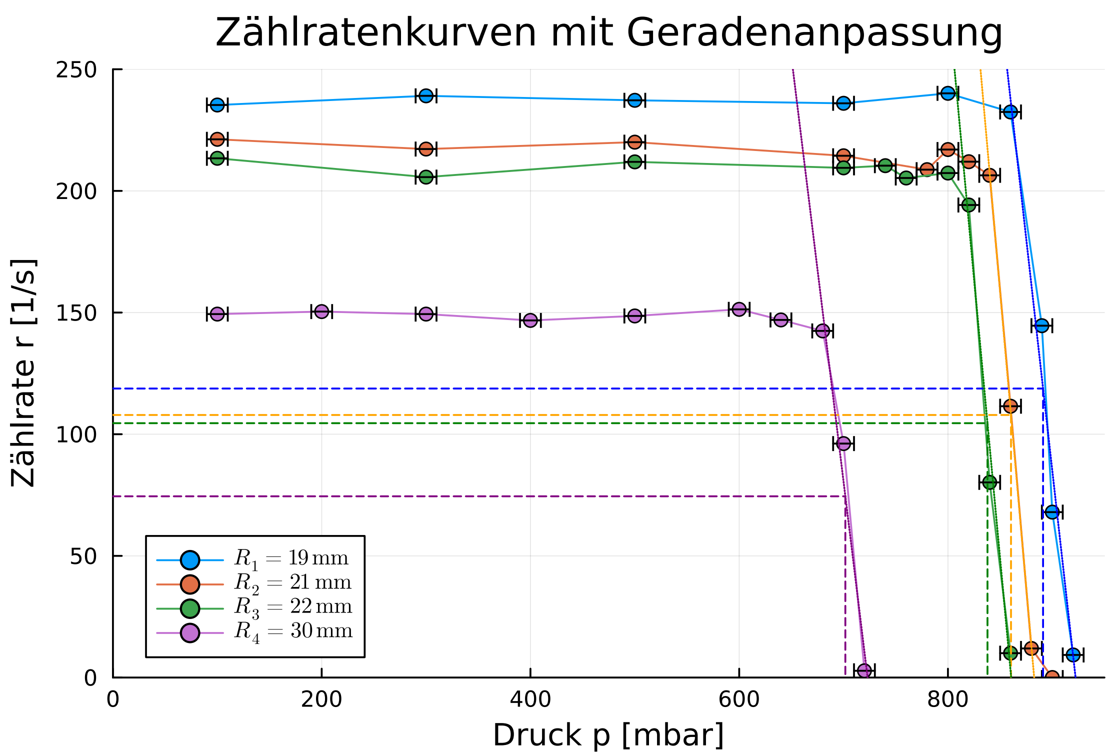
[11]:
```julia
# Gerade über die letzten 3/4 Messwerte:
# R_1: Letzte 4 Messwerte
xWerte1 = [p1[length(p1)-3], p1[length(p1)-2], p1[length(p1)-1], p1[length(p1)]]
yWerte1 = [rate1[length(rate1)-3], rate1[length(rate1)-2],␣
↪rate1[length(rate1)-1], rate1[length(rate1)]]
```

```julia
a1, b1, Δa1, Δb1 = rechnerischeGeradenanpassung(xWerte1, yWerte1)
gerade1(x) = a1 * x + b1
nullstelle1 = -b1/a1

#R_2: Nicht der letzte aber die 3 davor
xWerte2 = [p2[length(p2)-3], p2[length(p2)-2], p2[length(p2)-1]]
yWerte2 = [rate2[length(rate2)-3], rate2[length(rate2)-2], rate2[length(rate2)-1]]
a2, b2, Δa2, Δb2 = rechnerischeGeradenanpassung(xWerte2, yWerte2)
gerade2(x) = a2 * x + b2
nullstelle2 = -b2/a2

# R_3: Letzte 3 Messwerte
xWerte3 = [p3[length(p3)-2], p3[length(p3)-1], p3[length(p3)]]
yWerte3 = [rate3[length(rate3)-2], rate3[length(rate3)-1], rate3[length(rate3)]]
a3, b3, Δa3, Δb3 = rechnerischeGeradenanpassung(xWerte3, yWerte3)
gerade3(x) = a3 * x + b3
nullstelle3 = -b3/a3

# R_4: Letzte 3 Messwerte
xWerte4 = [p4[length(p4)-2], p4[length(p4)-1], p4[length(p4)]]
yWerte4 = [rate4[length(rate4)-2], rate4[length(rate4)-1], rate4[length(rate4)]]
a4, b4, Δa4, Δb4 = rechnerischeGeradenanpassung(xWerte4, yWerte4)
gerade4(x) = a4 * x + b4
nullstelle4 = -b4/a4

plot(p1, rate1, label = L"R_1 = 19 \mathrm{\, mm}", legend =:bottomleft, markershape=:circle,
    title = "Zählratenkurven mit Geradenanpassung", xlim = (0,950), ylim = (0,250), xerr=Δp)
plot!(p2, rate2, xerr=Δp, label = L"R_2 = 21 \mathrm{\, mm}", markershape=:circle)
plot!(p3, rate3, xerr=Δp, label = L"R_3 = 22 \mathrm{\, mm}", markershape=:circle)
plot!(p4, rate4, xerr=Δp, label = L"R_4 = 30 \mathrm{\, mm}", markershape=:circle)

r_mittel1 = (rate1[1] + rate1[2] + rate1[3] + rate1[4] + rate1[5])/(5*2)
Δr_mittel1 = sqrt((1/5*4) * ((0.5*rate1[1]-r_mittel1)^2 + (0.5*rate1[2]-r_mittel1)^2 + (0.5*rate1[3]-r_mittel1)^2 +
    (0.5*rate1[3]-r_mittel1)^2 + (0.5*rate1[4]-r_mittel1)^2 + (0.5*rate1[5]-r_mittel1)^2)) # Mittelwert-Fehler
mittlereRate1(x) = r_mittel1
p_mittel1 = (r_mittel1 - b1)/a1
Δp_mittel1 = sqrt((Δr_mittel1/a1)^2 + (Δb1/a1)^2 + ((r_mittel1-b1)*Δa1/a1^2)^2) # Gaußsche Fehlerfortpflanzung
```

```
plot!(mittlereRate1, [0,p_mittel1], linecolor=:blue, linestyle=:dash, label="")
plot!([p_mittel1, p_mittel1], [r_mittel1, 0], linecolor=:blue, linestyle=:dash,␣
 ↪label="")

r_mittel2 = (rate2[1] + rate2[2] + rate2[3] + rate2[4] + rate2[5] + rate2[6] +␣
 ↪rate2[7])/(7*2)
Δr_mittel2 = sqrt((1/7*6) * ((0.5*rate2[1]-r_mittel2)^2 + (0.
 ↪5*rate2[2]-r_mittel2)^2 + (0.5*rate2[3]-r_mittel2)^2 +
    (0.5*rate2[3]-r_mittel2)^2 + (0.5*rate2[4]-r_mittel2)^2 + (0.
 ↪5*rate2[5]-r_mittel2)^2) +
    (0.5*rate2[6]-r_mittel2)^2 + (0.5*rate2[7]-r_mittel2)^2)
mittlereRate2(x) = r_mittel2
p_mittel2 = (r_mittel2 - b2)/a2
Δp_mittel2 = sqrt((Δr_mittel2/a2)^2 + (Δb2/a2)^2 + ((r_mittel2-b2)*Δa2/a2^2)^2)
plot!(mittlereRate2, [0,p_mittel2], linecolor=:orange, linestyle=:dash,␣
 ↪label="")
plot!([p_mittel2, p_mittel2], [r_mittel2, 0], linecolor=:orange, linestyle=:
 ↪dash, label="")

r_mittel3 = (rate3[1] + rate3[2] + rate3[3] + rate3[4] + rate3[5] + rate3[6] +␣
 ↪rate3[7])/(7*2)
Δr_mittel3 = sqrt((1/7*6) * ((0.5*rate3[1]-r_mittel3)^2 + (0.
 ↪5*rate3[2]-r_mittel3)^2 + (0.5*rate3[3]-r_mittel3)^2 +
    (0.5*rate3[3]-r_mittel3)^2 + (0.5*rate3[4]-r_mittel3)^2 + (0.
 ↪5*rate3[5]-r_mittel3)^2) +
    (0.5*rate3[6]-r_mittel3)^2 + (0.5*rate3[7]-r_mittel3)^2)
mittlereRate3(x) = r_mittel3
p_mittel3 = (r_mittel3 - b3)/a3
Δp_mittel3 = sqrt((Δr_mittel3/a3)^2 + (Δb3/a3)^2 + ((r_mittel3-b3)*Δa3/a3^2)^2)
plot!(mittlereRate3, [0,p_mittel3], linecolor=:green, linestyle=:dash, label="")
plot!([p_mittel3, p_mittel3], [r_mittel3, 0], linecolor=:green, linestyle=:
 ↪dash, label="")

r_mittel4 = (rate4[1] + rate4[2] + rate4[3] + rate4[4] + rate4[5] + rate4[6] +␣
 ↪rate4[7])/(7*2)
Δr_mittel4 = sqrt((1/7*6) * ((0.5*rate4[1]-r_mittel4)^2 + (0.
 ↪5*rate4[2]-r_mittel4)^2 + (0.5*rate4[3]-r_mittel4)^2 +
    (0.5*rate4[3]-r_mittel4)^2 + (0.5*rate4[4]-r_mittel4)^2 + (0.
 ↪5*rate4[5]-r_mittel4)^2) +
    (0.5*rate4[6]-r_mittel4)^2 + (0.5*rate4[7]-r_mittel4)^2)
mittlereRate4(x) = r_mittel4
p_mittel4 = (r_mittel4 - b4)/a4
Δp_mittel4 = sqrt((Δr_mittel4/a4)^2 + (Δb4/a4)^2 + ((r_mittel4-b4)*Δa4/a4^2)^2)
plot!(mittlereRate4, [0,p_mittel4], linecolor=:purple, linestyle=:dash,␣
 ↪label="")
```

```
plot!([p_mittel4, p_mittel4], [r_mittel4, 0], linecolor=:purple, linestyle=:
↪dash, label="")

plot!(gerade1, label ="", linecolor=:blue, linestyle=:dot)
plot!(gerade2, label ="", linecolor=:orange, linestyle=:dot)
plot!(gerade3, label ="", linecolor=:green, linestyle=:dot)
plot!(gerade4, label ="", linecolor=:purple, linestyle=:dot)

xlabel!("Druck p [mbar]")
ylabel!("Zählrate r [1/s]")
```

[11]:



[12]:

```
#savefig
zählrateExtrapoliert = plot(p1, rate1, label = L"R_1 = 19 \mathrm{\, mm}",␣
↪legend =:bottomleft, markershape=:circle,
    xlim = (0,950), ylim = (0,250), xerr=Δp)
plot!(p2, rate2, xerr=Δp, label = L"R_2 = 21 \mathrm{\, mm}", markershape=:
↪circle)
plot!(p3, rate3, xerr=Δp, label = L"R_3 = 22 \mathrm{\, mm}", markershape=:
↪circle)
plot!(p4, rate4, xerr=Δp, label = L"R_4 = 30 \mathrm{\, mm}", markershape=:
↪circle)

c1 = (rate1[1] + rate1[2] + rate1[3] + rate1[4] + rate1[5])/(5*2)
```

```
mittlereRate1(x) = c1
plot!(mittlereRate1, [0,(c1-b1)/a1], linecolor=:blue, linestyle=:dash, label="")
plot!([(c1-b1)/a1, (c1-b1)/a1], [c1, 0], linecolor=:blue, linestyle=:dash,␣
 ↪label="")

c2 = (rate2[1] + rate2[2] + rate2[3] + rate2[4] + rate2[5] + rate2[6] +␣
 ↪rate2[7])/(7*2)
mittlereRate2(x) = c2
plot!(mittlereRate2, [0,(c2-b2)/a2], linecolor=:orange, linestyle=:dash,␣
 ↪label="")
plot!([(c2-b2)/a2, (c2-b2)/a2], [c2, 0], linecolor=:orange, linestyle=:dash,␣
 ↪label="")

c3 = (rate3[1] + rate3[2] + rate3[3] + rate3[4] + rate3[5] + rate3[6] +␣
 ↪rate3[7])/(7*2)
mittlereRate3(x) = c3
plot!(mittlereRate3, [0,(c3-b3)/a3], linecolor=:green, linestyle=:dash,␣
 ↪label="")
plot!([(c3-b3)/a3, (c3-b3)/a3], [c3, 0], linecolor=:green, linestyle=:dash,␣
 ↪label="")

c4 = (rate4[1] + rate4[2] + rate4[3] + rate4[4] + rate4[5] + rate4[6] +␣
 ↪rate4[7])/(7*2)
mittlereRate4(x) = c4
plot!(mittlereRate4, [0,(c4-b4)/a4], linecolor=:purple, linestyle=:dash,␣
 ↪label="")
plot!([(c4-b4)/a4, (c4-b4)/a4], [c4, 0], linecolor=:purple, linestyle=:dash,␣
 ↪label="")

plot!(gerade1, label ="", linecolor=:blue, linestyle=:dot)
plot!(gerade2, label ="", linecolor=:orange, linestyle=:dot)
plot!(gerade3, label ="", linecolor=:green, linestyle=:dot)
plot!(gerade4, label ="", linecolor=:purple, linestyle=:dot)

xlabel!("Druck p [mbar]")
ylabel!("Zählrate r [1/s]")

savefig(zählrateExtrapoliert, "../../media/B3.3/zaehlratenkurven extrapoliert.
 ↪svg");
savefig(zählrateExtrapoliert, "../../media/B3.3/zaehlratenkurven extrapoliert.
 ↪pdf");
```

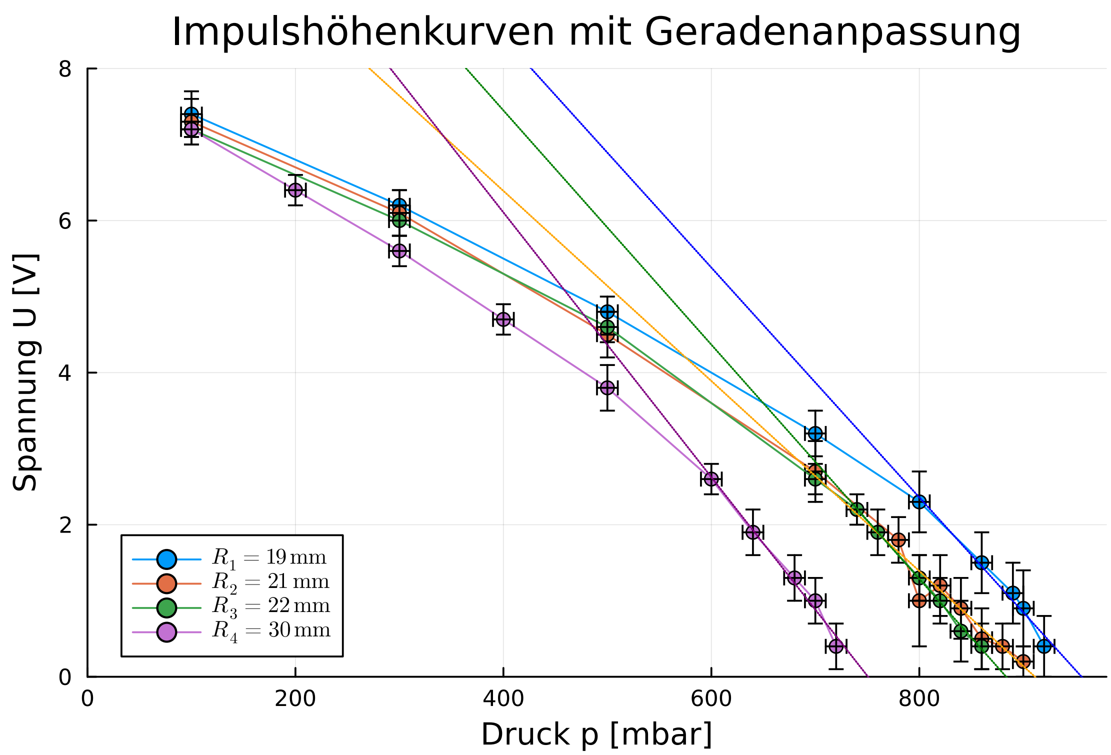
2.

[13]:
```
# Gerade über die letzten 5 Messwerte:
# U_1
```

```
xWerte1 = [p1[length(p1)-4], p1[length(p1)-3], p1[length(p1)-2],␣
 ↪p1[length(p1)-1], p1[length(p1)]]
yWerte1 = [spannung1[length(spannung1)-4], spannung1[length(spannung1)-3],␣
 ↪spannung1[length(spannung1)-2],
    spannung1[length(spannung1)-1], spannung1[length(spannung1)]]
a1, b1, Δa1, Δb1 = rechnerischeGeradenanpassung(xWerte1, yWerte1)
gerade1(x) = a1 * x + b1
nullstelle1b = -b1/a1
Δnullstelle1b = sqrt( (Δb1/a1)^2 + (b1*Δa1/a1^2)^2) # Gaußsche␣
 ↪Fehlerfortpflanzung

# U_2
xWerte2 = [p2[length(p2)-4], p2[length(p2)-3], p2[length(p2)-2],␣
 ↪p2[length(p2)-1], p2[length(p2)]]
yWerte2 = [spannung2[length(spannung2)-4], spannung2[length(spannung2)-3],␣
 ↪spannung2[length(spannung2)-2],
    spannung2[length(spannung2)-1], spannung2[length(spannung2)]]
a2, b2, Δa2, Δb2 = rechnerischeGeradenanpassung(xWerte2, yWerte2)
gerade2(x) = a2 * x + b2
nullstelle2b = -b2/a2
Δnullstelle2b = sqrt( (Δb2/a2)^2 + (b2*Δa2/a2^2)^2)

# U_3
xWerte3 = [p3[length(p3)-4], p3[length(p3)-3], p3[length(p3)-2],␣
 ↪p3[length(p3)-1], p3[length(p3)]]
yWerte3 = [spannung3[length(spannung3)-4], spannung3[length(spannung3)-3],␣
 ↪spannung3[length(rate3)-2],
    spannung3[length(spannung3)-1], spannung3[length(spannung3)]]
a3, b3, Δa3, Δb3 = rechnerischeGeradenanpassung(xWerte3, yWerte3)
gerade3(x) = a3 * x + b3
nullstelle3b = -b3/a3
Δnullstelle3b = sqrt( (Δb3/a3)^2 + (b3*Δa3/a3^2)^2)

# U_4
xWerte4 = [p4[length(p4)-4], p4[length(p4)-3], p4[length(p4)-2],␣
 ↪p4[length(p4)-1], p4[length(p4)]]
yWerte4 = [spannung4[length(spannung4)-4], spannung4[length(spannung4)-3],␣
 ↪spannung4[length(spannung4)-2],
    spannung4[length(spannung4)-1], spannung4[length(spannung4)]]
a4, b4, Δa4, Δb4 = rechnerischeGeradenanpassung(xWerte4, yWerte4)
gerade4(x) = a4 * x + b4
nullstelle4b = -b4/a4
Δnullstelle4b = sqrt( (Δb4/a4)^2 + (b4*Δa4/a4^2)^2)

plot(p1, spannung1, xerr = Δp, yerr = Δspannung1, label = L"R_1 = 19 \mathrm{\,␣
 ↪mm}", markershape=:circle,
```

```
    title = "Impulshöhenkurven mit Geradenanpassung", legend=:bottomleft, ylims = (0,8), xlims = (0,980))
plot!(p2, spannung2, xerr = Δp, yerr = Δspannung2, label = L"R_2 = 21 \mathrm{\, mm}", markershape=:circle)
plot!(p3, spannung3, xerr = Δp, yerr = Δspannung3, label = L"R_3 = 22 \mathrm{\, mm}", markershape=:circle)
plot!(p4, spannung4, xerr = Δp, yerr = Δspannung4, label = L"R_4 = 30 \mathrm{\, mm}", markershape=:circle)

plot!(gerade1, label="", linecolor=:blue, linestyle=:dot)
plot!(gerade2, label="", linecolor=:orange, linestyle=:dot)
plot!(gerade3, label="", linecolor=:green, linestyle=:dot)
plot!(gerade4, label="", linecolor=:purple, linestyle=:dot)

xlabel!("Druck p [mbar]")
ylabel!("Spannung U [V]")
```

[13]:



[14]:

```
#savefig
spannungskurvenExtrapoliert = plot(p1, spannung1, xerr = Δp, yerr = Δspannung1, label = L"R_1 = 19 \mathrm{\, mm}",
    markershape=:circle, legend=:bottomleft, ylims = (0,8), xlims = (0,980))
plot!(p2, spannung2, xerr = Δp, yerr = Δspannung2, label = L"R_2 = 21 \mathrm{\, mm}", markershape=:circle)
```

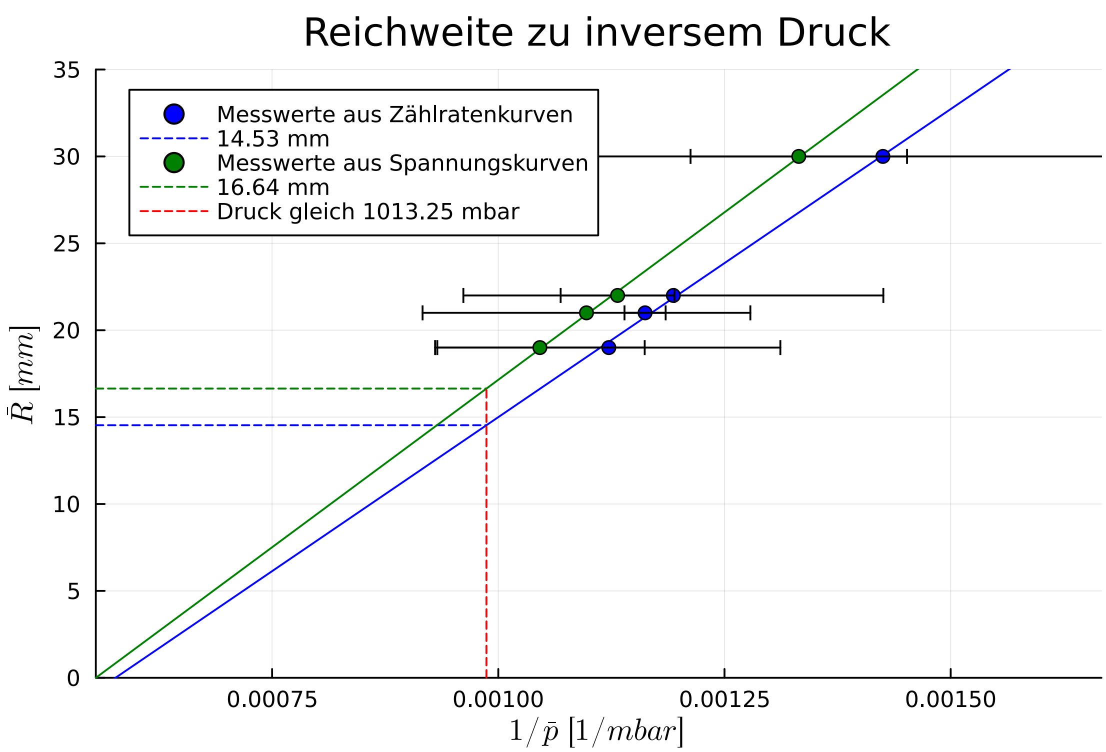
```
plot!(p3, spannung3, xerr = Δp, yerr = Δspannung3, label = L"R_3 = 22␣
↪\mathrm{\, mm}", markershape=:circle)
plot!(p4, spannung4, xerr = Δp, yerr = Δspannung4, label = L"R_4 = 30␣
↪\mathrm{\, mm}", markershape=:circle)

plot!(gerade1, label="", linecolor=:blue, linestyle=:dot)
plot!(gerade2, label="", linecolor=:orange, linestyle=:dot)
plot!(gerade3, label="", linecolor=:green, linestyle=:dot)
plot!(gerade4, label="", linecolor=:purple, linestyle=:dot)

xlabel!("Druck p [mbar]")
ylabel!("Spannung U [V]")

savefig(spannungskurvenExtrapoliert, "../../media/B3.3/spannungskurven␣
↪extrapoliert.svg");
savefig(spannungskurvenExtrapoliert, "../../media/B3.3/spannungskurven␣
↪extrapoliert.pdf");
```

Reichweite zu inverser Druck Geraden durch die gerade bestimmten Nullstellen:

[15]:
```
Δx = 0.005/1013.25^2 # 1/mbar (nur Rundungsfehler + Gaußsche␣
↪Fehlerfortpflanzung)

reichweite1 = [19, 21, 22, 30] # mm
druckInverse1 = [1/p_mittel1, 1/p_mittel2, 1/p_mittel3, 1/p_mittel4] # 1/mbar
ΔpInverse1 = [(1/p_mittel1)^2 * Δp_mittel1, (1/p_mittel2)^2 * Δp_mittel2,
    (1/p_mittel3)^2 * Δp_mittel3, (1/p_mittel4)^2 * Δp_mittel4]
a, b, Δa, Δb = rechnerischeGeradenanpassung(druckInverse1, reichweite1)
gerade1(x) = a*x + b

reichweite2 = [19, 21, 22, 30] # mm
druckInverse2 = [1/nullstelle1b, 1/nullstelle2b, 1/nullstelle3b, 1/
↪nullstelle4b] # 1/mbar
ΔpInverse2 = [(1/nullstelle1b)^2 * Δnullstelle1b, (1/nullstelle2b)^2 *␣
↪Δnullstelle2b,
    (1/nullstelle3b)^2 * Δnullstelle3b, (1/nullstelle4b)^2 * Δnullstelle4b]
a2, b2, Δa2, Δb2 = rechnerischeGeradenanpassung(druckInverse2, reichweite2)
gerade2(x) = a2*x + b2

plot(gerade1, xlims = (-b2/a2, 1/600), ylims = (0,35),label = "", color=:blue)
scatter!(druckInverse1, xerr = ΔpInverse1, reichweite1, markershape=:circle,␣
↪legend=:topleft,
    label="Messwerte aus Zählratenkurven", title = "Reichweite zu inversem␣
↪Druck", color =:blue)
k1 = gerade1(1/1013.25)
Δk1 = sqrt( (Δa*(1/1013.25))^2 + (a*Δx)^2 + (Δb)^2 ) # Gaußsche␣
↪Fehlerfortpflanzung
```

```
f(x) = k1
k1_rounded = round(k1, digits = 2)
plot!(f, [-b2/a2, 1/1013.25], label = "$k1_rounded mm", linestyle=:dash,␣
↪linecolor=:blue)

plot!(gerade2, label="", color=:green)
scatter!(druckInverse2, xerr = ΔpInverse2, reichweite2, markershape=:circle,␣
↪legend=:topleft,
    label="Messwerte aus Spannungskurven", color=:green)
k2 = gerade2(1/1013.25)
Δk2 = sqrt( (Δa2*(1/1013.25))^2 + (a2*Δx)^2 + (Δb2)^2 )
g(x) = k2
k2_rounded = round(k2, digits = 2)
plot!(g, [-b2/a2, 1/1013.25], label = "$k2_rounded mm", linestyle=:dash,␣
↪linecolor=:green)

plot!([1/1013.25, 1/1013.25], [16.55, 0], label = "Druck gleich 1013.25 mbar",␣
↪linestyle=:dash, linecolor=:red)
xlabel!(L"1/\bar{p} \enspace [1/mbar]")
ylabel!(L"\bar{R} \enspace [mm]")
```

[15]:



[16]:

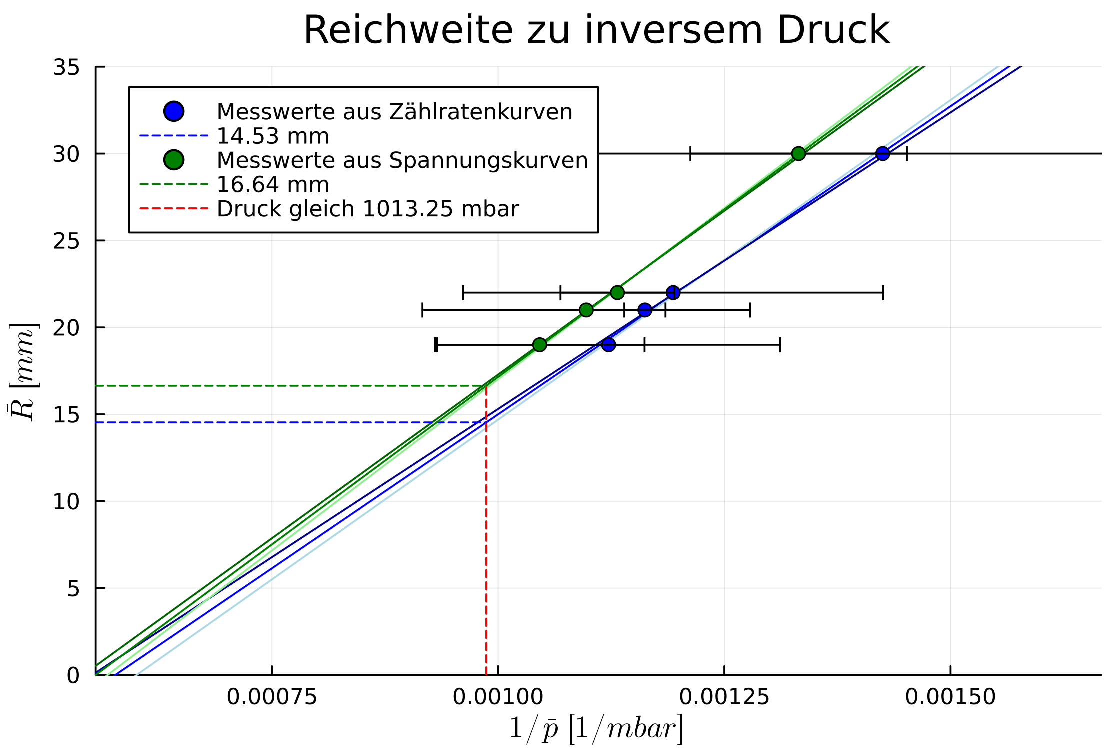
```
#savefig
```

```
reichweiteDruck = plot(gerade1, xlims = (-b2/a2, 1/600), ylims = (0,35),label =␣
 ↪"", color=:blue)
scatter!(druckInverse1, xerr = ΔpInverse1, reichweite1, markershape=:circle,␣
 ↪legend=:topleft,
    label="Messwerte aus Zählratenkurven", color =:blue)
k1 = gerade1(1/1013.25)
f(x) = k1
k1_rounded = round(k1, digits = 2)
plot!(f, [-b2/a2, 1/1013.25], label = "$k1_rounded mm", linestyle=:dash,␣
 ↪linecolor=:blue)

plot!(gerade2, label="", color=:green)
scatter!(druckInverse2, xerr = ΔpInverse2, reichweite2, markershape=:circle,␣
 ↪legend=:topleft,
    label="Messwerte aus Spannungskurven", color=:green)
k2 = gerade2(1/1013.25)
g(x) = k2
k2_rounded = round(k2, digits = 2)
plot!(g, [-b2/a2, 1/1013.25], label = "$k2_rounded mm", linestyle=:dash,␣
 ↪linecolor=:green)

plot!([1/1013.25, 1/1013.25], [16.55, 0], label = "Druck gleich 1013.25 mbar",␣
 ↪linestyle=:dash, linecolor=:red)
xlabel!(L"1/\bar{p} \enspace [1/mbar]")
ylabel!(L"\bar{R} \enspace [mm]")

savefig(reichweiteDruck, "../../media/B3.3/reichweiten inverse druecke.svg");
savefig(reichweiteDruck, "../../media/B3.3/reichweiten inverse druecke.pdf");
```

[17]:
```
fehlergerade1(x) = (a+Δa)*x + (b-Δb)
fehlergerade2(x) = (a-Δa)*x + (b+Δb)
fehlergerade1b(x) = (a2+Δa2)*x + (b2-Δb2)
fehlergerade2b(x) = (a2-Δa2)*x + (b2+Δb2)

plot(gerade1, xlims = (-b2/a2, 1/600), ylims = (0,35),label = "", color=:blue)

plot!(fehlergerade1, label ="", linecolor=:lightblue)
plot!(fehlergerade2, label ="", linecolor=:darkblue)
plot!(fehlergerade1b, label ="", linecolor=:lightgreen)
plot!(fehlergerade2b, label ="", linecolor=:darkgreen)

scatter!(druckInverse1, xerr = ΔpInverse1, reichweite1, markershape=:circle,␣
 ↪legend=:topleft,
    label="Messwerte aus Zählratenkurven", title = "Reichweite zu inversem␣
 ↪Druck", color =:blue)
k1 = gerade1(1/1013.25)
```

```
f(x) = k1
k1_rounded = round(k1, digits = 2)
plot!(f, [-b2/a2, 1/1013.25], label = "$k1_rounded mm", linestyle=:dash,␣
 ↪linecolor=:blue)

plot!(gerade2, label="", color=:green)
scatter!(druckInverse2, xerr = ΔpInverse2, reichweite2, markershape=:circle,␣
 ↪legend=:topleft,
    label="Messwerte aus Spannungskurven", color=:green)
k2 = gerade2(1/1013.25)
g(x) = k2
k2_rounded = round(k2, digits = 2)
plot!(g, [-b2/a2, 1/1013.25], label = "$k2_rounded mm", linestyle=:dash,␣
 ↪linecolor=:green)

plot!([1/1013.25, 1/1013.25], [16.55, 0], label = "Druck gleich 1013.25 mbar",␣
 ↪linestyle=:dash, linecolor=:red)
xlabel!(L"1/\bar{p} \enspace [1/mbar]")
ylabel!(L"\bar{R} \enspace [mm]")
```

[17]:

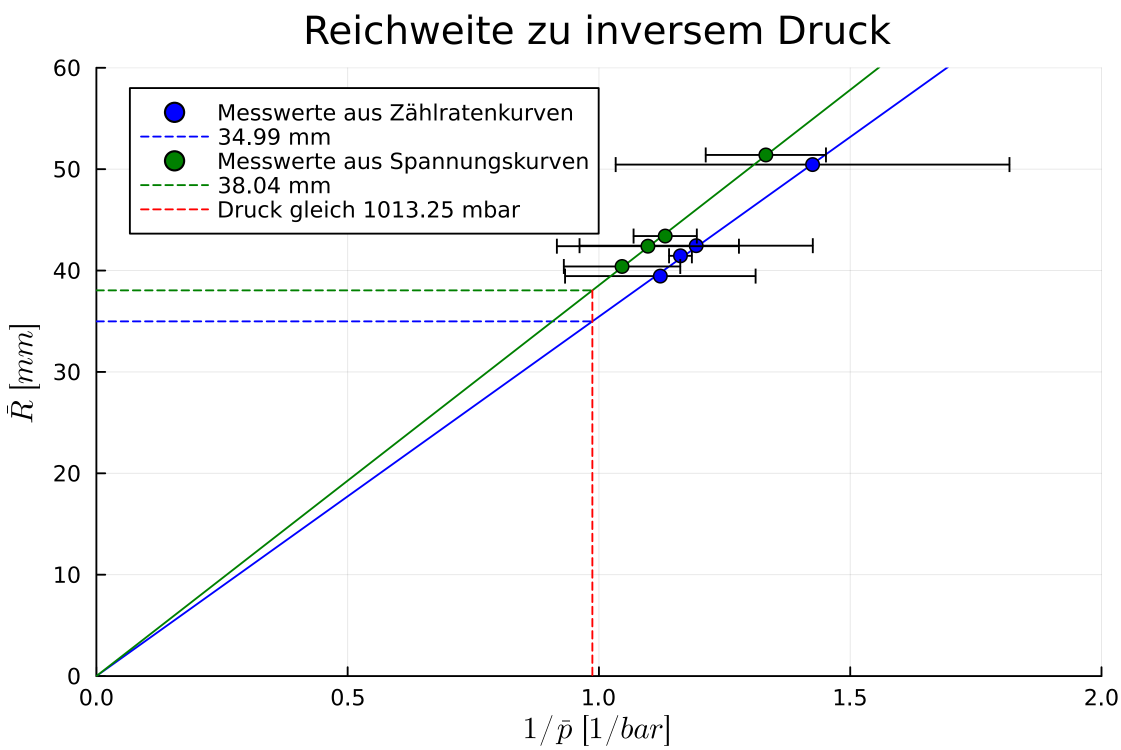


Geraden um b_i nach oben verschoben:

[18]:

```julia
gerade1b(x) = (gerade1(x) - b) * 10^(-3)
gerade2b(x) = (gerade2(x) - b2) * 10^(-3)
reichweite1b = reichweite1 .- b # mm
reichweite2b = reichweite2 .- b2 # mm

druckInverse1scaled = druckInverse1 .* 10^3 # 1/bar
ΔpInverse1scaled = ΔpInverse1 .* 10^3 # Δpscaled = Δp*10^(-3), (1/pscaled)^2 =
↪(1/p)^2*10^6
druckInverse2scaled = druckInverse2 .* 10^3 # 1/bar
ΔpInverse2scaled = ΔpInverse2 .* 10^3

plot(gerade1b, xlims = (0, 2), ylims = (0,60),label = "", color=:blue)
scatter!(druckInverse1scaled, xerr = ΔpInverse1scaled, reichweite1b,
↪markershape=:circle, legend=:topleft,
    label="Messwerte aus Zählratenkurven", title = "Reichweite zu inversem
↪Druck", color =:blue)
k1b = gerade1b(1/1.01325)
Δk1b = sqrt( (Δa*10^(-3)/1.01325)^2 + (a*0.000005/1.01325^2)^2) # Gaußsche
↪Fehlerfortpflanzung
fb(x) = k1b
k1b_rounded = round(k1b, digits = 2)
plot!(fb, [0, 1/1.01325], label = "$k1b_rounded mm", linestyle=:dash,
↪linecolor=:blue)

plot!(gerade2b, label="", color=:green)
scatter!(druckInverse2scaled, xerr = ΔpInverse2scaled, reichweite2b,
↪markershape=:circle,
    label="Messwerte aus Spannungskurven", color=:green)
k2b = gerade2b(1/1.01325)
Δk2b = sqrt( (Δa2*10^(-3)*(1/1.01325))^2 + (a2*0.000005/1.01325^2)^2)
gb(x) = k2b
k2b_rounded = round(k2b, digits = 2)
plot!(gb, [0, 1/1.01325], label = "$k2b_rounded mm", linestyle=:dash,
↪linecolor=:green)

plot!([1/1.01325, 1/1.01325], [k2b, 0], label = "Druck gleich 1013.25 mbar",
↪linestyle=:dash, linecolor=:red)
xlabel!(L"1/\bar{p} \enspace [1/bar]")
ylabel!(L"\bar{R} \enspace [mm]")
```

[18]:



[19]:
```
# savefig
reichweiteDruckVerschobenScaled = plot(gerade1b, xlims = (0, 2), ylims =
    (0,60),label = "", color=:blue)
scatter!(druckInverse1scaled, xerr = ΔpInverse1scaled, reichweite1b,
    markershape=:circle, legend=:topleft,
    label="Messwerte aus Zählratenkurven", color =:blue)
k1b = gerade1b(1/1.01325)
Δk1b = sqrt( (Δa*(1/1.01325))^2 + (a*Δx)^2) # Gaußsche Fehlerfortpflanzung
fb(x) = k1b
k1b_rounded = round(k1b, digits = 2)
plot!(fb, [0, 1/1.01325], label = "$k1b_rounded mm", linestyle=:dash,
    linecolor=:blue)

plot!(gerade2b, label="", color=:green)
scatter!(druckInverse2scaled, xerr = ΔpInverse2scaled, reichweite2b,
    markershape=:circle,
    label="Messwerte aus Spannungskurven", color=:green)
k2b = gerade2b(1/1.01325)
Δk2b = sqrt( (Δa2*(1/1.01325))^2 + (a2*Δx)^2)
gb(x) = k2b
k2b_rounded = round(k2b, digits = 2)
plot!(gb, [0, 1/1.01325], label = "$k2b_rounded mm", linestyle=:dash,
    linecolor=:green)
```

```
plot!([1/1.01325, 1/1.01325], [k2b, 0], label = "Druck gleich 1013.25 mbar",␣
 ↪linestyle=:dash, linecolor=:red)
xlabel!(L"1/\bar{p} \enspace [1/bar]")
ylabel!(L"\bar{R} \enspace [mm]")

savefig(reichweiteDruckVerschobenScaled, "../../media/B3.3/reichweiten inverse␣
 ↪druecke verschoben skaliert.svg");
savefig(reichweiteDruckVerschobenScaled, "../../media/B3.3/reichweiten inverse␣
 ↪druecke verschoben skaliert.pdf");
```

Bragg-Kleemann-Regel zur Bestimmung der Reichweite der α-Teilchen in Aluminium

[20]:
```
R_A(A,  , R_Luft) = 3.2 * 10^(-4) * R_Luft *A^(1/2)/  # g/cm^3
ΔR_A(A,  , R_Luft, Δ , ΔR_Luft) = 3.2 * 10^(-4) * sqrt( ( A^(1/2) * R_Luft * Δ  /
 ↪  ^2 )^2
    + ( A^(1/2) * ΔR_Luft /   )^2 )
```

[20]:
```
ΔR_A (generic function with 1 method)
```

[21]:
```
[R_A(27, 2.7, 34.99), ΔR_A(27, 2.7, 34.99, 0.00005, 1.32)]
```

[21]:
```
2-element Vector{Float64}:
 0.021548251646874673
 0.0008129092769599874
```