Laboratorium z podstaw fizyki Wydziału EliT AGH

Przykłady obliczeń

© Michał Kołodziej 2016, kolodziej.michal@gmail.com

Laboratorium 3 - Rezonans akustyczny

Opis eksperymentu

Generator daje napięcie sinusoidalne

$$U(t) = U_0 sin(2\pi f t) = 10 sin(2\pi \frac{1000}{sek} t)[V]$$

Napięcie to jest podawane na elektromagnes głośnika, powodując drganie membrany z częstotliwością f

Głośnik generuje falę dźwiękową. Fala dźwiękowa jest falą podłużną, czyli drgania poszczególnych części ośrodka składających się w ruch falowy, mają taki kierunek jak kierunek rozchodzenia się fali. Kierunek rozchodzenia się fali płaskiej ($sin(kx - \omega t)$) jest to kierunek przepływu energi.

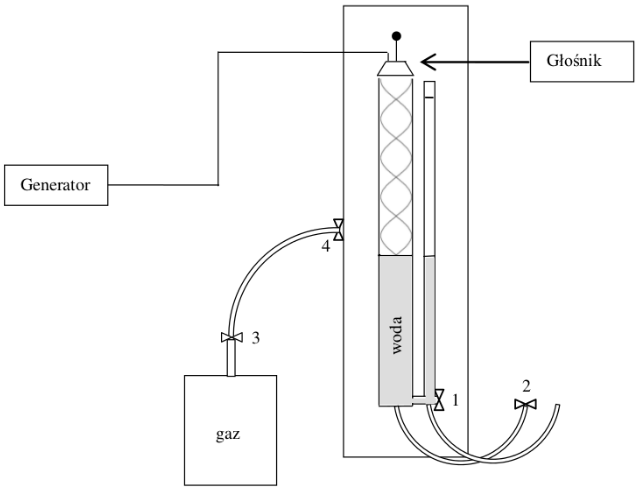
Ważną jednostką jest ciśnienie akustyczne (dźwiękowe). Ciśnienie akustyczne – zmienne w czasie odchylenie od średniej wartości ciśnienia statycznego panującego w ośrodku, występujące podczas rozchodzenia się w nim fali akustycznej. Ciśnienie akustyczne opisuje natężenie dźwięku i wyraża się w paskalach [[https://pl.wikipedia.org/wiki/Ci%C5%9Bnienie_akustyczne (https://pl.wikipedia.org/wiki/Ci%C5%9Bnienie_akustyczne)]].

Fala dźwiękowa w transmitującym medium powoduje odchylenie (ciśnienie akustyczne, dynamiczne ciśnienie) w lokalnym niezależnym ciśnieniu, ciśnieniu statycznym. Ciśnienie dźwiękowe oznaczone jako p (mierzone np. przez mikrofon) można zdefiniować następująco:

$$p_{\mathrm{total}} = p_{\mathrm{stat}} + p$$

. Jednostką SI są paskale Pa

gdzie: p_{total} to całkowite ciśnienie, p_{stat} to statyczne ciśnienie (np. odczytane z barometru).



Równanie fali dźwiękowej

Równanie fali dźwiękowej opisuje sposób propagacji fal dźwiękowych. Równanie fali akustycznej dla ciśnienia dźwiękowego w jednym wymiarze jest dane poprzez:

$$\frac{\partial^2 p}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0$$

gdzie: p to ciśnienie dźwięku w [Pa], x to przemieszczenie cząsteczki (np. tlenu) w [m], c to prędkość dźwięku w [m/s], t to czas w [sek]

Równanie fali akustycznej dla prędkości cząsteczek ma podobną postać:

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0$$

gdzie u jest prędkością cząsteczki w [m/s].

Rozwiązanie równania fali dźwiękowej - fala płaska

Rozwiązaniami tych równań różniczkowych są funkcje postaci:

$$p(x,t) = p_0 cos(\omega t - kx)\ [Pa]$$
$$x(x,t) = x_0 sin(\omega t - kx)\ [m]$$
$$u(x,t) = u_0 cos(\omega t - kx)\ [m/sek]$$

Czyli głośnik emitują falę dźwiękową, czyli zaburzenie ciśnienia $p(t,x)$ albo ruch cząsteczki powietrza z prędkością $u(x,t)$.

Rezonans akustyczny - fala stojąca

Wygenerowana fala przez głośnik to fala dźwiękowa biegnąca, która napotykając lustro wody zostaje odbita. Fale dźwiękowe mogą ze sobą interferować, tworząc fale o nowym kształcie. Tutaj w wynikiu interferencji powstaje fala stojąca: Czyli tuba z wodą jest w tym przypadku pudłem rezonansowym.

Mamy dwie fale, jedną z głośnika poruszającą się w kierunku osi X (stąd $-\omega t$)

$$p_1(x,t) \ = \ p_0 \ \sin(kx - \omega t)$$

i drugą odbitą od powierzchni wody, która porusza się w przeciwnym kierunku do X (stąd $+\omega t$). W naszym przypadku nie wiemy jaką ona będzie miała fazę. Faza fali odbitej jest związana z drogą jaką przebyła fala z głośnika do powierzchni wody, jeżeli ta długość (czyli efektywna długość tuby rezonansowej) będzie równa wielokrotności długości fali λ, to φ = 0, czyli zmieniając efektywną długość tuby rezonansowej zmieniamy φ (robiliśmy to w tym ćwiczeniu).

$$p_2(x,t) \ = \ p_0 \ \sin(kx + \omega t + \varphi)$$

gdzie: ω $= 2\pi f$ to częstotliwość kołowa [rad/sek], $k = \frac{2\pi}{\lambda}$ to liczba falowa [rad/m], λ to długość fali [m].

Fala wynikowa p będzie sumą fal p1 i p2:

$$p(x,t) \ = \ p_0 \ \sin(kx - \omega t) \ + \ p_0 \ \sin(kx + \omega t + \varphi). \ [Pa]$$

Używając zależności trygonometrycznej (suma do iloczynu) dla 'sin(u) + sin(v)' możemy uprościć powyższe równanie do:

$$p(x,t) \ = \ 2\,p_0 \ \cos(\omega t + \frac{\varphi}{2}) \ \sin(kx + \frac{\varphi}{2}). \ [Pa]$$

Równanie to opisuje falę, która oscyluje z upływem czasu t (cos(ωt)), a jej zależność od położenia x jest stacjonarna (nie zależy od czasu - sin(kx)).

W punktach x = 0, λ/2, λ, 3λ/2, ... (φ=0) nazywanych węzłami amplituda p jest zawsze zero, natomiast w punktach x = λ/4, 3λ/4, 5λ/4, ... (φ=0) nazywanych strzałkami, amplituda jest maksymalna.

Odległość pomiędzy dwoma sąsiednimi węzłami albo strzałkami wynosi λ/2.

W naszym przypadku nie wiemy jakie fala z głośnika p1 i fala odbita p2 będą miały fazy, dlatego zmieniamy długość tuby rezonansowej (zmieniając poziom lustra wody), ponieważ zmienia to fazę fali odbitej.

Amplituda drgań osiąga największe wartości (równe $2p_0$) dla położeń x spełniających warunek

$$kx + \frac{\varphi}{2} = n\pi$$

gdzie n=0,1,2,... W tych miejscach ośrodek drga najsilniej (powstają strzałki). Położenie węzłów można znaleźć z równania:

$$kx + \frac{\varphi}{2} = n\pi + \frac{\pi}{2}$$

Wykonanie ćwiczenia

Odczytujemy częstotliwość fali

robimy to poprzez zadanie częstotliwości na generatorze napięcia:

In [1]:
```
czestotliwosc_fali_1 = 800 # Hz
```

Out[1]:
```
800
```

Mierzymy położenie węzłów fali stojącej

Mierzymy wysokości słupa wody dla których była cisza.

In [2]:
```
pomiar_punkty_ciszy_w_gore_1 = [25.9, 47, 69, 87] #cm
```

Out[2]:
```
4-element Array{Float64,1}:
 25.9
 47.0
 69.0
 87.0
```

Wyznaczamy odległości między węzłami

In [3]:
```
function delta(xs) xs[2:end] - xs[1:end-1] end

odleglosc_cisz_1 = delta(pomiar_punkty_ciszy_w_gore_1) #cm
```

Out[3]:
```
3-element Array{Float64,1}:
 21.1
 22.0
 18.0
```

Wyznaczamy średnią odległość pomiędzy węzłami

Średnia (https://en.wikipedia.org/wiki/Standard_deviation#Discrete_random_variable (https://en.wikipedia.org/wiki/Standard_deviation#Discrete_random_variable)):

$$\mu = \frac{1}{N}(x_1 + \cdots + x_N)$$

In [4]:
```
srednia_odleglosc_cisz_1 = mean(odleglosc_cisz_1)
round(srednia_odleglosc_cisz_1, 1) #cm
```

Out[4]: 20.4

Wyznaczamy błąd średniej odległości pomiędzy węzłami

Odchylenie standardowe (https://en.wikipedia.org/wiki/Standard_deviation#Discrete_random_variable (https://en.wikipedia.org/wiki/Standard_deviation#Discrete_random_variable)):

$$\sigma = \sqrt{\frac{1}{N}\left[(x_1 - \mu)^2 + (x_2 - \mu)^2 + \cdots + (x_N - \mu)^2\right]}$$

x1, x2, ..., xN to pomiary.

In [5]:
```
srednia_odleglosc_cisz_1_blad = round(std(odleglosc_cisz_1), 1) #cm
```

Out[5]: 2.1

Wniosek - długość fali

Odległość pomiędzy węzłami fali stojącej to połowa długości fali. Długość fali:

In [6]:
```
dlugosc_fali_1 = round(2 * srednia_odleglosc_cisz_1, 1) #cm
```

Out[6]: 40.7

Wniosek - prędkość fali

Teoretyczna prędkość fali dźwiękowej w powietrzu wynosi 340 [m/s], w CO2 259 [m/s]

Prędkość fali v [m/s] to stosunek długości fali λ [m] do okresu fali T [s]. Wyznaczyliśmy długość fali, natomiast okres fali to odwrotność częstotliwości, stąd prędkość fali:

In [7]:
```
predkosc_fali_1 = round(dlugosc_fali_1/100 * czestotliwosc_fali_1, 2)
```

Out[7]: 325.6

Wizualizacja fal

In [8]:
```
fala_biegnaca(f, λ, x, t) = cos(2*pi*f*t - 2*pi/λ*x) # f [Hz] , λ [m]
fala_stojaca(f, λ, x, t) = 2*sin(2*pi/λ*x)*cos(2*pi*f*t) # f [Hz] , λ [m]
```

Out[8]: fala_stojaca (generic function with 1 method)

In [9]:
```
type ObliczonaFala
    czestotliwosc_fali
    dlugosc_fali
    dlugosc_fali_blad
    predkosc_fali
    fala_biegnaca
    fala_stojaca
end

function wyznacz_dlugosc_fali(pomiar_punkty_ciszy, czestotliwosc_fali)
    odleglosc_cisz = delta(pomiar_punkty_ciszy) #cm
    srednia_odleglosc_cisz = mean(odleglosc_cisz) #cm
    srednia_odleglosc_cisz_blad = round(std(odleglosc_cisz), 2) #cm
    dlugosc_fali = round(2 * srednia_odleglosc_cisz, 2) #cm
    dlugosc_fali_blad = 2 * srednia_odleglosc_cisz_blad
    predkosc_fali = dlugosc_fali / 100 * czestotliwosc_fali
    fala_b = (x, t) -> fala_biegnaca(czestotliwosc_fali, dlugosc_fali/100, x, t)
    fala_s = (x, t) -> fala_stojaca(czestotliwosc_fali, dlugosc_fali/100, x, t)

    ObliczonaFala(czestotliwosc_fali, dlugosc_fali, dlugosc_fali_blad, predkosc_fali, fala
end
```

Out[9]:
```
wyznacz_dlugosc_fali (generic function with 1 method)
```

In [10]:
```
pomiar_punkty_ciszy_1 = [25.9, 47, 69, 87]
czestotliwosc_fali_1 = 800 #Hz
fala1 = wyznacz_dlugosc_fali(pomiar_punkty_ciszy_1, czestotliwosc_fali_1)
```

Out[10]:
```
ObliczonaFala(800,40.73,4.2,325.84,(anonymous function),(anonymous function))
```

In [11]:
```
pomiar_punkty_ciszy_2 = [45, 64, 85]
czestotliwosc_fali_2 = 900 #Hz
fala2 = wyznacz_dlugosc_fali(pomiar_punkty_ciszy_2, czestotliwosc_fali_2)
```

Out[11]:
```
ObliczonaFala(900,40.0,2.82,360.0,(anonymous function),(anonymous function))
```

In [12]:
```
pomiar_punkty_ciszy_3 = [18, 34, 50, 67]
czestotliwosc_fali_3 = 1100 #Hz
fala3 = wyznacz_dlugosc_fali(pomiar_punkty_ciszy_3, czestotliwosc_fali_3)
```

Out[12]:
```
ObliczonaFala(1100,32.67,1.16,359.37,(anonymous function),(anonymous function))
```

In [53]:
```
pomiar_punkty_ciszy_4 = [21, 39, 56, 70, 86]
czestotliwosc_fali_4 = 800 #Hz
fala4 = wyznacz_dlugosc_fali(pomiar_punkty_ciszy_4, czestotliwosc_fali_4)
```

Out[53]:
```
ObliczonaFala(800,32.5,3.42,260.0,(anonymous function),(anonymous function))
```

In [54]:
```
pomiar_punkty_ciszy_5 = [21, 34, 51, 66, 82]
czestotliwosc_fali_5 = 900 #Hz
fala5 = wyznacz_dlugosc_fali(pomiar_punkty_ciszy_5, czestotliwosc_fali_5)
```

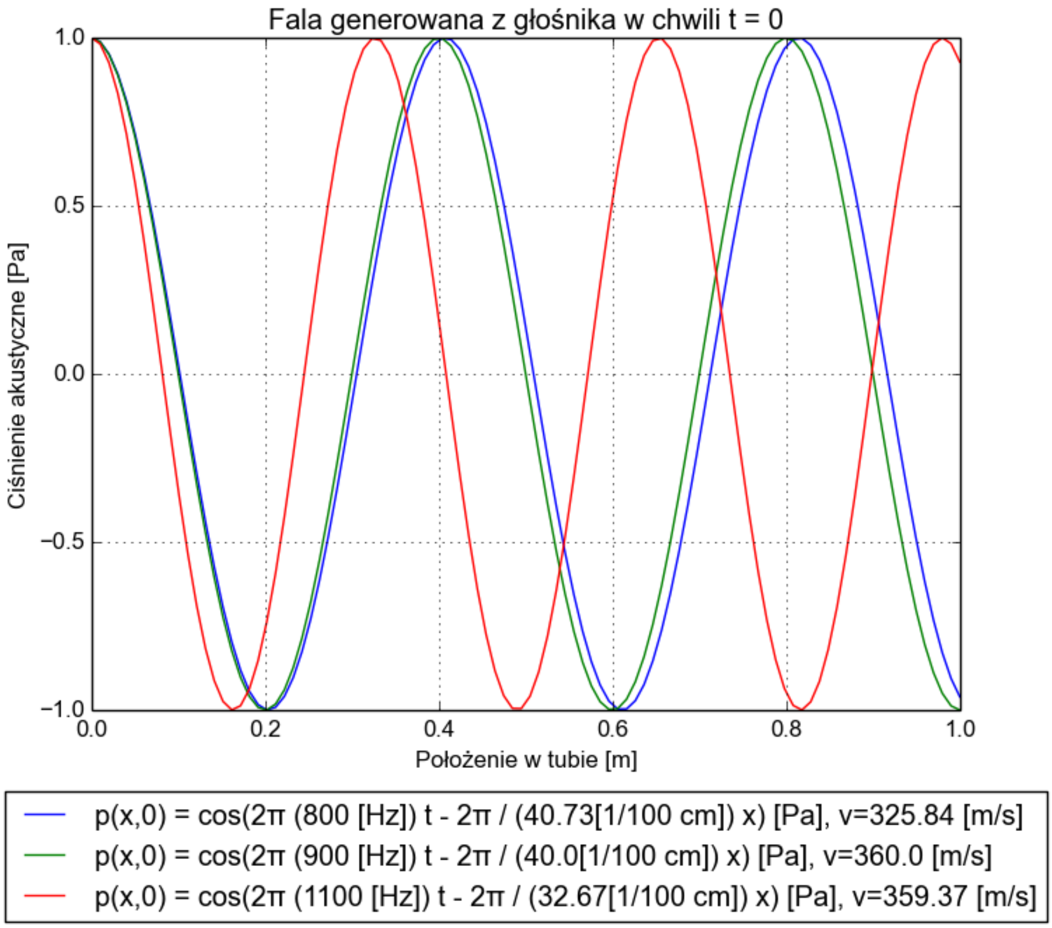
Out[54]:
```
ObliczonaFala(900,30.5,3.42,274.5,(anonymous function),(anonymous function))
```

In [55]:
```
pomiar_punkty_ciszy_6 = [16, 29, 42, 55, 67, 80, 92]
czestotliwosc_fali_6 = 1100 #Hz
fala6 = wyznacz_dlugosc_fali(pomiar_punkty_ciszy_6, czestotliwosc_fali_6)
```

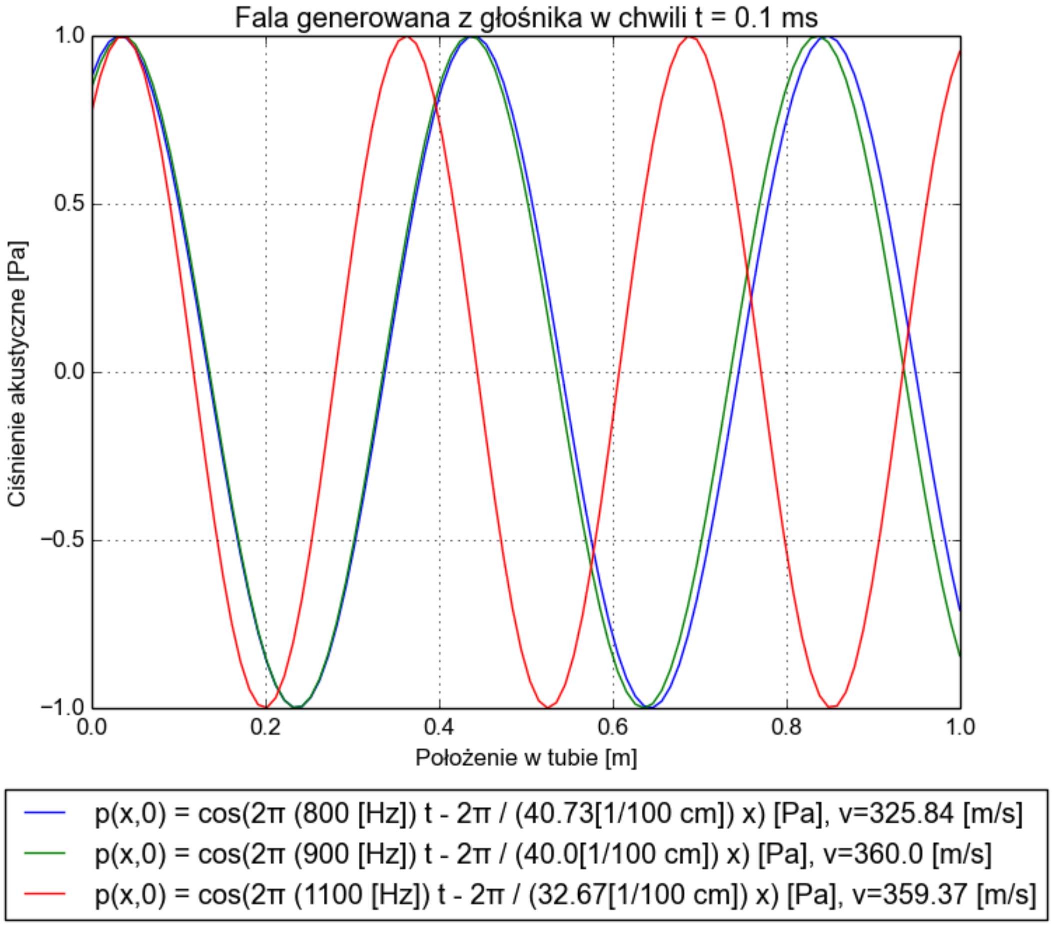
Out[55]:
```
ObliczonaFala(1100,25.33,1.04,278.62999999999994,(anonymous function),(anonymous function
))
```

In [75]:

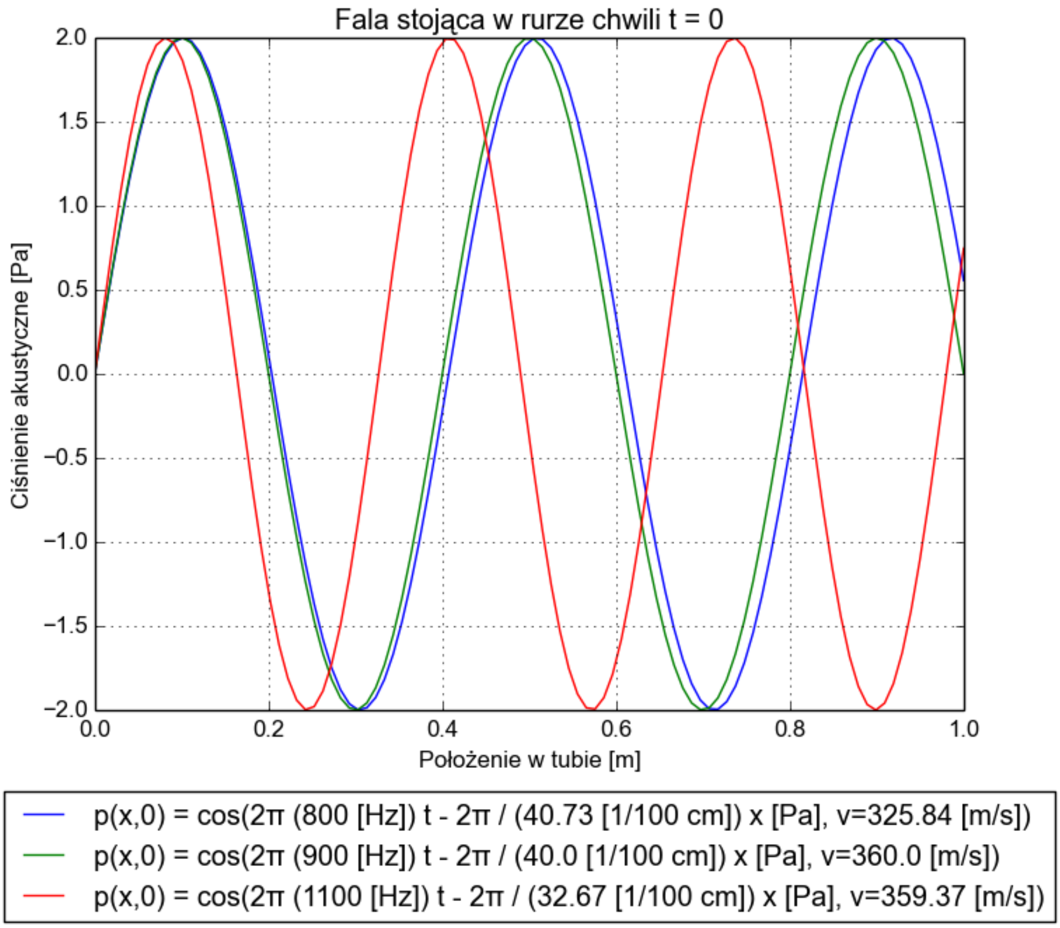
```
using PyPlot

xs = linspace(0, 1, 100)
ys1 = map(x -> fala1.fala_biegnaca(x, 0), xs)
ys2 = map(x -> fala2.fala_biegnaca(x, 0), xs)
ys3 = map(x -> fala3.fala_biegnaca(x, 0), xs)
plot(xs, ys1, xs, ys2, xs, ys3)
legend(
    [
        "p(x,0) = cos(2π ($(fala1.czestotliwosc_fali) [Hz]) t - 2π / ($(fala1.dlugosc_fali
        "p(x,0) = cos(2π ($(fala2.czestotliwosc_fali) [Hz]) t - 2π / ($(fala2.dlugosc_fali
        "p(x,0) = cos(2π ($(fala3.czestotliwosc_fali) [Hz]) t - 2π / ($(fala3.dlugosc_fali
    ],
#      loc="center left", bbox_to_anchor=(1, 0.5)
    loc="upper center", bbox_to_anchor=(0.5, -0.1)
)
ylabel("Ciśnienie akustyczne [Pa]")
xlabel("Położenie w tubie [m]")
# PyPlot.xlim(0.0, 1.2)
# PyPlot.ylim(0.0, 4)
title("Fala generowana z głośnika w chwili t = 0")
grid()
```



In [76]:
```
using PyPlot

xs = linspace(0, 1, 100)
ys1 = map(x -> fala1.fala_biegnaca(x, 0.1/1000), xs)
ys2 = map(x -> fala2.fala_biegnaca(x, 0.1/1000), xs)
ys3 = map(x -> fala3.fala_biegnaca(x, 0.1/1000), xs)
plot(xs, ys1, xs, ys2, xs, ys3)
legend(
    [
        "p(x,0) = cos(2π ($(fala1.czestotliwosc_fali) [Hz]) t - 2π / ($(fala1.dlugosc_fali
        "p(x,0) = cos(2π ($(fala2.czestotliwosc_fali) [Hz]) t - 2π / ($(fala2.dlugosc_fali
        "p(x,0) = cos(2π ($(fala3.czestotliwosc_fali) [Hz]) t - 2π / ($(fala3.dlugosc_fali
    ],
#     loc="center left", bbox_to_anchor=(1, 0.5)
    loc="upper center", bbox_to_anchor=(0.5, -0.1)
)
ylabel("Ciśnienie akustyczne [Pa]")
xlabel("Położenie w tubie [m]")
# PyPlot.xlim(0.0, 1.2)
# PyPlot.ylim(0.0, 4)
title("Fala generowana z głośnika w chwili t = 0.1 ms")
grid()
```

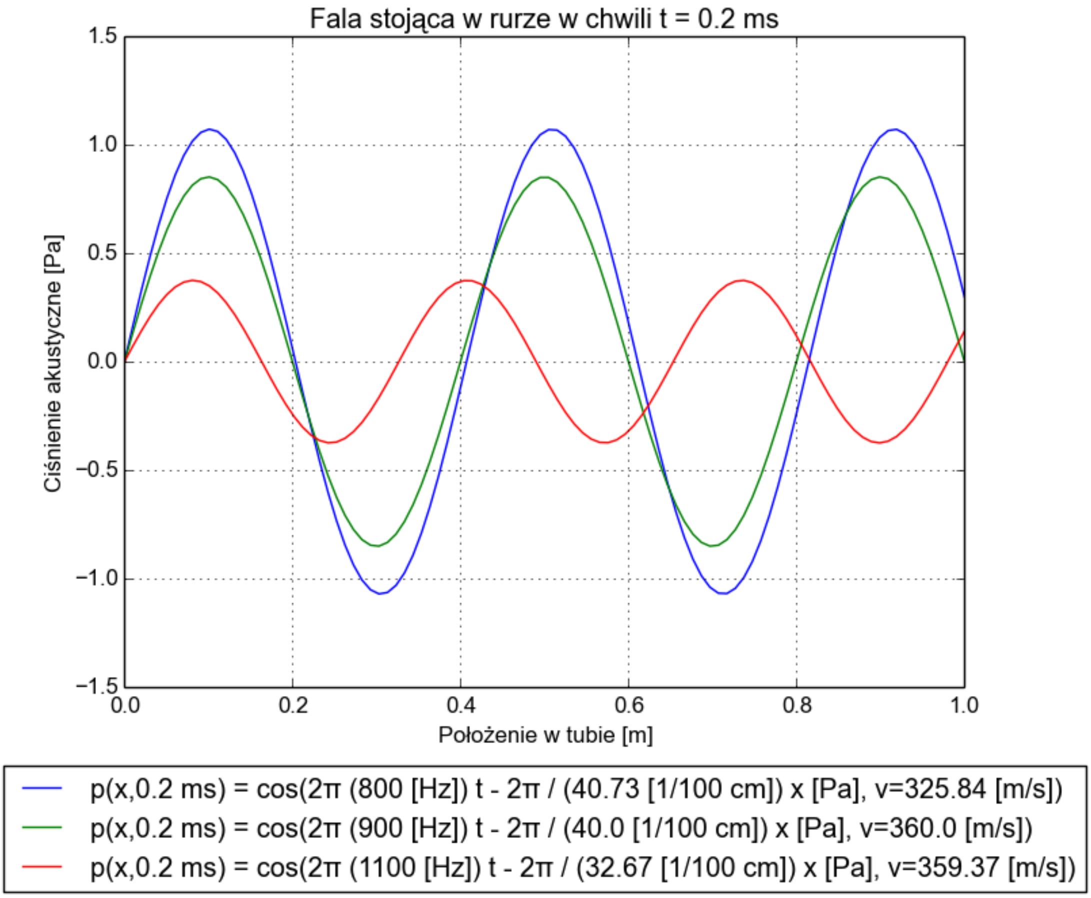


In [79]:
```
using PyPlot

xs = linspace(0, 1, 100)
ys1 = map(x -> fala1.fala_stojaca(x, 0), xs)
ys2 = map(x -> fala2.fala_stojaca(x, 0), xs)
ys3 = map(x -> fala3.fala_stojaca(x, 0), xs)
plot(xs, ys1, xs, ys2, xs, ys3)
legend(
    [
        "p(x,0) = cos(2π ($(fala1.czestotliwosc_fali) [Hz]) t - 2π / ($(fala1.dlugosc_fali
        "p(x,0) = cos(2π ($(fala2.czestotliwosc_fali) [Hz]) t - 2π / ($(fala2.dlugosc_fali
        "p(x,0) = cos(2π ($(fala3.czestotliwosc_fali) [Hz]) t - 2π / ($(fala3.dlugosc_fali
    ],
#     loc="center left", bbox_to_anchor=(1, 0.5)
    loc="upper center", bbox_to_anchor=(0.5, -0.1)
)
ylabel("Ciśnienie akustyczne [Pa]")
xlabel("Położenie w tubie [m]")
# PyPlot.xlim(0.0, 1.2)
# PyPlot.ylim(0.0, 4)
title("Fala stojąca w rurze chwili t = 0")
grid()
```



In [78]:

```
using PyPlot

xs = linspace(0, 1, 100)
ys1 = map(x -> fala1.fala_stojaca(x, 0.2/1000), xs)
ys2 = map(x -> fala2.fala_stojaca(x, 0.2/1000), xs)
ys3 = map(x -> fala3.fala_stojaca(x, 0.2/1000), xs)
plot(xs, ys1, xs, ys2, xs, ys3)
legend(
    [
        "p(x,0.2 ms) = cos(2π ($(fala1.czestotliwosc_fali) [Hz]) t - 2π / ($(fala1.dlugosc
        "p(x,0.2 ms) = cos(2π ($(fala2.czestotliwosc_fali) [Hz]) t - 2π / ($(fala2.dlugosc
        "p(x,0.2 ms) = cos(2π ($(fala3.czestotliwosc_fali) [Hz]) t - 2π / ($(fala3.dlugosc
    ],
#     loc="center left", bbox_to_anchor=(1, 0.5)
    loc="upper center", bbox_to_anchor=(0.5, -0.1)
)
ylabel("Ciśnienie akustyczne [Pa]")
xlabel("Położenie w tubie [m]")
# PyPlot.xlim(0.0, 1.2)
# PyPlot.ylim(0.0, 4)
title("Fala stojąca w rurze w chwili t = 0.2 ms")
grid()
```



Wyznaczenie średniej prędkości fal

Regresja liniowa dwuparametrowa

$$S = \sum_{i=1}^{n} 1 = n,$$

$$S_x = \sum_{i=1}^{n} x_i,$$

$$S_{xx} = \sum_{i=1}^{n} x_i^2,$$

$$S_y = \sum_{i=1}^{n} y_i,$$

$$S_{yy} = \sum_{i=1}^{n} y_i^2,$$

$$S_{xy} = \sum_{i=1}^{n} x_i y_i,$$

$$\Delta = S \cdot S_{xx} - (S_x)^2.$$

Prosta dopasowania:

$$y = ax + b$$

Współczynniki prostej

$$a = \frac{S \cdot S_{xy} - S_x \cdot S_y}{\Delta},$$

$$b = \frac{S_{xx} \cdot S_y - S_x \cdot S_{xy}}{\Delta}.$$

suma odchyleń standardowych wszystkich pomiarów:

$$\sigma_y^2 = S_{yy} - aS_{xy} - bS_y.$$

Błąd kwadratowy a:

$$\sigma_a^2 = \frac{S}{S-2}\frac{\sigma_y^2}{\Delta},$$

Błąd kwadratowy b:

$$\sigma_b^2 = \sigma_a^2 \frac{S_{xx}}{S},$$

In [17]:
```
# http://en.wikipedia.org/wiki/Simple_linear_regression
# http://pl.wikipedia.org/wiki/Metoda_najmniejszych_kwadrat%C3%B3w#Przypadek_klasyczny
function reg_lin_2P(xs,ys)
    n = length(xs)
    Sx = sum(xs)
    Sxx = sum(x->x*x, xs)
    Sy = sum(ys)
    Syy = sum(y->y*y, ys)
    Sxy = sum(zip(xs,ys)) do e
        x, y = e
        x*y
    end

    Δ = (n*Sxx-Sx^2)
    a = (n*Sxy-Sx*Sy)/Δ
    b = 1/n*Sx - a/n*Sx
    σ2ϵ = 1/(n*(n-2)) * (n*Syy-Sy^2-a^2*Δ)
    σ2a = n*σ2ϵ/Δ
    σ2b = σ2a/n*Sxx


    return a,b,σ2a,σ2b
end
```

Out[17]:
```
reg_lin_2P (generic function with 1 method)
```

Regresja liniowa jednoparametrowa

Równanie funkcji dopasowującej:

$$y = ax$$

Sumy:

$$\begin{aligned} S_x &= \sum_{i=1}^{n} x_i, \\ S_{xx} &= \sum_{i=1}^{n} x_i^2, \\ S_{xy} &= \sum_{i=1}^{n} x_i y_i, \end{aligned}$$

Współczynnki a:

$$a = \frac{S_{xy}}{S_{xx}},$$

Suma błędów:

$$\sigma_y^2 = \sum_{i=1}^{n} (y_i - a x_i)^2,$$

Błąd kwadratowy a:

$$\sigma_a^2 = \frac{S}{S-2} \frac{\sigma_y^2}{S_{\ xx}},$$

In [18]:

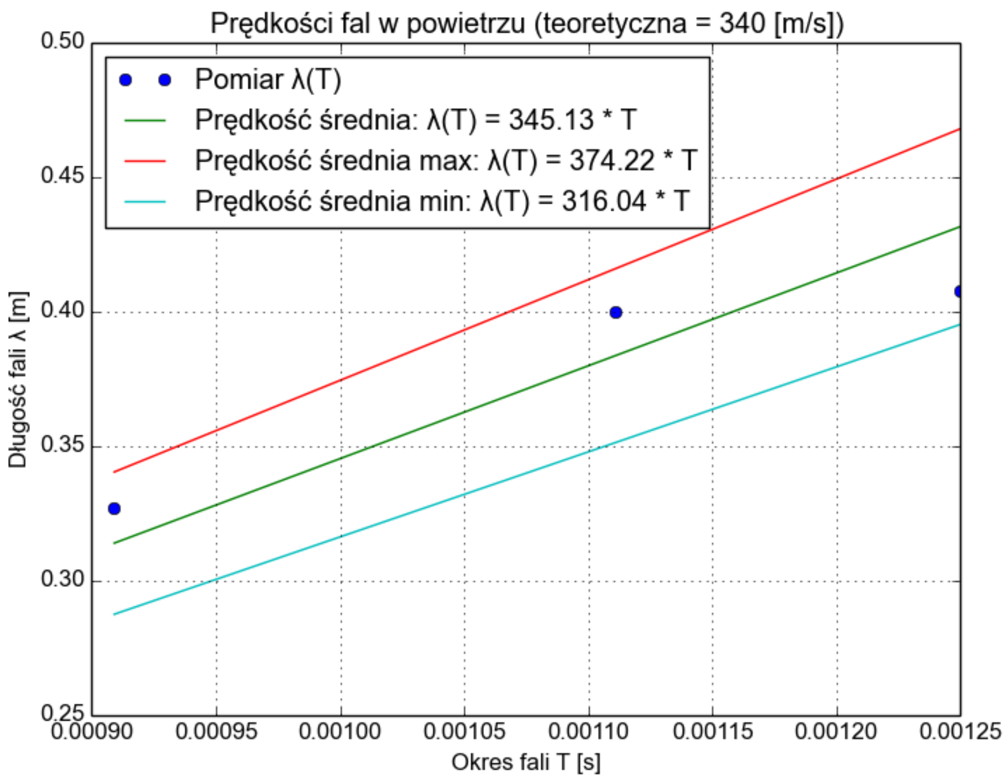
```
# http://en.wikipedia.org/wiki/Simple_linear_regression
# http://pl.wikipedia.org/wiki/Metoda_najmniejszych_kwadrat%C3%B3w#Przypadek_klasyczny
# http://layer.uci.agh.edu.pl/T.Stobiecki/dydaktyka/rachunek_niepewnosci_pomiaru.pdf
function reg_lin_1P(xs,ys)
    n = length(xs)
    Sxx = sum(x->x*x, xs)
    Sxy = sum(zip(xs,ys)) do e
        x, y = e
        x*y
    end

    a = Sxy/Sxx

    S2 = sum(zip(xs,ys)) do e
        x, y = e
        (y - a*x)^2
    end

    Δa = sqrt(n/(n-2)*S2/(Sxx))

    return a,Δa
end
```

Out[18]:

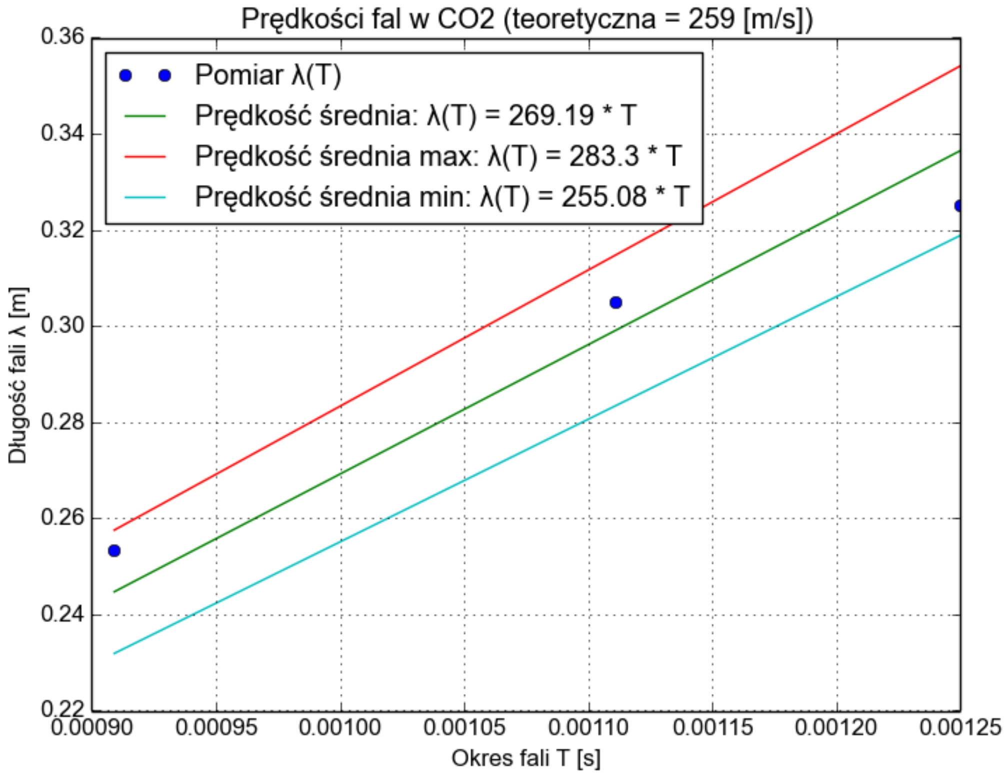
```
reg_lin_1P (generic function with 1 method)
```

Prędkość fali

In [44]:
```
import PyPlot

dlugosci_fal_powietrze = [fala1.dlugosc_fali/100, fala2.dlugosc_fali/100, fala3.dlugosc_fa
okresy_fal_powietrze = [1/fala1.czestotliwosc_fali, 1/fala2.czestotliwosc_fali, 1/fala3.cz
predkosc_srednia_powietrze, predkosc_srednia_powietrze_blad = map(x->round(x,2), reg_lin_1
predkosc_srednia_powietrze_max = round(predkosc_srednia_powietrze + predkosc_srednia_powie
predkosc_srednia_powietrze_min = round(predkosc_srednia_powietrze - predkosc_srednia_powie

PyPlot.plot(
            okresy_fal_powietrze, dlugosci_fal_powietrze,"o",
            okresy_fal_powietrze, map(x-> predkosc_srednia_powietrze * x, okresy_fal_powie
            okresy_fal_powietrze, map(x-> predkosc_srednia_powietrze_max * x, okresy_fal_p
            okresy_fal_powietrze, map(x-> predkosc_srednia_powietrze_min * x, okresy_fal_p
PyPlot.legend(
    [
    "Pomiar λ(T)",
    "Prędkość średnia: λ(T) = $predkosc_srednia_powietrze * T",
    "Prędkość średnia max: λ(T) = $predkosc_srednia_powietrze_max * T",
    "Prędkość średnia min: λ(T) = $predkosc_srednia_powietrze_min * T"

    ],
    "upper left"
)
PyPlot.ylabel("Długość fali λ [m]")
PyPlot.xlabel("Okres fali T [s]")
# PyPlot.xlim(0.0009, 0.0013)
# PyPlot.ylim(0.250, 0.5)
PyPlot.title("Prędkości fal w powietrzu (teoretyczna = 340 [m/s])")
PyPlot.grid()
```



In [57]:
```
import PyPlot

dlugosci_fal_CO2 = [fala4.dlugosc_fali/100, fala5.dlugosc_fali/100, fala6.dlugosc_fali/100
okresy_fal_CO2 = [1/fala4.czestotliwosc_fali, 1/fala5.czestotliwosc_fali, 1/fala6.czestotl
predkosc_srednia_CO2, predkosc_srednia_CO2_blad = map(x->round(x,2), reg_lin_1P(okresy_fal
predkosc_srednia_CO2_max = round(predkosc_srednia_CO2 + predkosc_srednia_CO2_blad, 2)
predkosc_srednia_CO2_min = round(predkosc_srednia_CO2 - predkosc_srednia_CO2_blad, 2)

PyPlot.plot(
            okresy_fal_CO2, dlugosci_fal_CO2,"o",
            okresy_fal_CO2, map(x-> predkosc_srednia_CO2 * x, okresy_fal_CO2),
            okresy_fal_CO2, map(x-> predkosc_srednia_CO2_max * x, okresy_fal_CO2),
            okresy_fal_CO2, map(x-> predkosc_srednia_CO2_min * x, okresy_fal_CO2))
PyPlot.legend(
    [
    "Pomiar λ(T)",
    "Prędkość średnia: λ(T) = $predkosc_srednia_CO2 * T",
    "Prędkość średnia max: λ(T) = $predkosc_srednia_CO2_max * T",
    "Prędkość średnia min: λ(T) = $predkosc_srednia_CO2_min * T"

    ],
    "upper left"
)
PyPlot.ylabel("Długość fali λ [m]")
PyPlot.xlabel("Okres fali T [s]")
# PyPlot.xlim(0.0009, 0.0013)
# PyPlot.ylim(0.250, 0.5)
PyPlot.title("Prędkości fal w CO2 (teoretyczna = 259 [m/s])")
PyPlot.grid()
```



Wyznaczanie adiabaty κ i liczby stopni swobody molekuł powietrza i CO2 na podstawie prędkości dźwięku

Związek łączący prędkość fali z parametrami gazu

$$v^2 = \frac{\kappa RT}{\mu}$$

Gdzie v - prędkość fali

$$\kappa = \frac{C_p}{C_v}$$

adiabata (również oznaczana γ) - stosunki ciepła właściwego przy stałym ciśnieniu do ciepła właściwego przy stałej objętości, gdzie Cp i Cv oznaczają ciepła molowe.

Adiabata jest również związana z liczbą stopni swobody i molekuł gazu w następujący sposób:

$$\kappa = \frac{i+2}{i}$$

Wartości tablicowe dla wybranych gazów: http://en.wikipedia.org/wiki/Heat_capacity_ratio (http://en.wikipedia.org/wiki/Heat_capacity_ratio)

$$T = t + 273.15[K]$$

temperatura bezwzględna, t jest w °C.

$$R = 8.3144621[\frac{J}{mol \cdot K}]$$

$$\mu[kg/mol]$$

masa molowa medium

Można wyznaczyć liczbę stopni swobody molekuł gazu

Liczba stopni swobody:

$$i(v, \mu, T) = \frac{2}{\frac{v^2\mu}{RT} - 1}$$

In [74]:

```
masa_molowa_N2 = 28 # [g/mol]
masa_molowa_O2 = 32 # [g/mol]
masa_molowa_Ar = 40 # [g/mol]

zawartosc_N2_w_powietrzu = 0.78084 # w 100%
zawartosc_O2_w_powietrzu = 0.20946 # w 100%
zawartosc_Ar_w_powietrzu = 0.00934 # w 100%

masa_molowa_powietrza =
    masa_molowa_N2 * zawartosc_N2_w_powietrzu +
    masa_molowa_O2 * zawartosc_O2_w_powietrzu +
    masa_molowa_Ar * zawartosc_Ar_w_powietrzu # [g/mol]

stopnie_swobody_powietrze_teor = 5

masa_molowa_CO2 = 44.0
stopnie_swobody_CO2_teor = 6

R_const = 8.3144621 # [J/mol/K]
```

Out[74]: 8.3144621

Prawo przenoszenia błędów

Mamy z zadanie oszacować błąd wielkości y, która jest funkcją zmierzonych (lub obliczonych) wartości x1, x2, x3 ... i ich błędów Δx1, Δx2, Δx3:

$$y(x_1, x_2, x_3)$$

Błąd wielkości Δy obliczamy z prawa przenoszenia błędów:

$$\Delta y(x_1, x_2, x_3; \Delta x_1, \Delta x_2, \Delta x_3) = \sqrt{\left(\frac{\partial y}{\partial x_1}\Delta x_1\right)^2 + \left(\frac{\partial y}{\partial x_2}\Delta x_2\right)^2 + \left(\frac{\partial y}{\partial x_3}\Delta x_3\right)^2}$$

Błędy obliczania stopni swobody

Przyjmujemy, że mierzymy temperaturę z dokładnością do 5 °C

$$\Delta T = 5[K]$$

Stałą gazową przyjmujemy bez błędu:

$$\Delta R = 0[J/(mol \cdot K)]$$

Bład masy molowej przyjmujemy:

$$\Delta \mu_{Air} = 1[g/mol] = 0.001[kg/mol]$$

Błąd prędkości dźwięku Δv bierzemy z wcześniejszych obliczeń.

$$i(v, \mu, T) = \frac{2}{\kappa - 1}$$

$$\Delta\kappa(v, \mu, T; \Delta v, \Delta\mu, \Delta T) = \sqrt{\left[\frac{\partial\left(\frac{v^2\mu}{RT}\right)}{\partial v}\Delta v\right]^2 + \left[\frac{\partial\left(\frac{v^2\mu}{RT}\right)}{\partial \mu}\Delta \mu\right]^2 + \left[\frac{\partial\left(\frac{v^2\mu}{RT}\right)}{\partial T}\Delta T\right]^2}$$

$$= \sqrt{\left[\frac{2v\mu}{RT}\Delta v\right]^2 + \left[\frac{v^2}{RT}\Delta\mu\right]^2 + \left[-\frac{v^2}{RT^2}\Delta T\right]^2}$$

$$\Delta i(\kappa, \Delta\kappa) = \frac{2}{(\kappa - 1)^2}\Delta\kappa$$

Adiabata powietrza i CO2

In [67]:
```
adiabata(predkosc, masa_molowa, temperatura) = predkosc^2 * masa_molowa / R_const / (tempe

adiabata_powietrza = adiabata(predkosc_srednia_powietrze,
    masa_molowa_powietrza/1000,
    20)

adiabata_CO2 = adiabata(predkosc_srednia_CO2,
    masa_molowa_CO2/1000,
    20)

adiabata_powietrza, adiabata_CO2
```

Out[67]: (1.4150108509404467,1.308786308420574)

Błąd adiabaty powietrza i CO2

In [68]:
```
adiabata_blad(v,μ,T, Δv, Δμ, ΔT) = sqrt(
                             (2*v*μ/R_const/(T + 273)*Δv)^2
                             +(v^2/R_const/(T + 273)*Δμ)^2
                             +(v^2*μ/R_const/((T + 273)^2)*ΔT)^2)
adiabata_powietrza_blad = round(
adiabata_blad(
    predkosc_srednia_powietrze,
    masa_molowa_powietrza/1000,
    20,
    predkosc_srednia_powietrze_blad,
    masa_molowa_powietrza/1000/100,
    5),
2)

adiabata_CO2_blad = round(
adiabata_blad(
    predkosc_srednia_CO2,
    masa_molowa_CO2/1000,
    20,
    predkosc_srednia_CO2_blad,
    masa_molowa_CO2/1000/100,
    5),
2)

adiabata_powietrza_blad, adiabata_CO2_blad
```
Out[68]:
```
(0.24,0.14)
```

Stopnie swobody cząsteczek powietrza i CO2

In [69]:
```
stopnie_swobody(k) = 2/(k-1)
stopnie_swobody_powietrza = stopnie_swobody(adiabata_powietrza)
stopnie_swobody_CO2 = stopnie_swobody(adiabata_CO2)

stopnie_swobody_powietrza, stopnie_swobody_CO2
```
Out[69]:
```
(4.819151102839469,6.4769711138745)
```

Błąd stopni swobody cząsteczek powietrza i CO2

In [72]:
```
stopnie_swobody_blad(κ, Δκ) = 2 / (κ - 1)^2 * Δκ

stopnie_swobody_powietrza_blad = stopnie_swobody_blad(adiabata_powietrza, adiabata_powietr
stopnie_swobody_CO2_blad = stopnie_swobody_blad(adiabata_CO2, adiabata_CO2_blad)

stopnie_swobody_powietrza_blad, stopnie_swobody_CO2_blad
```
Out[72]:
```
(2.7869060822398644,2.9365808366975283)
```