McGill Physical Journal

Measuring the Specific Heat Ratio of Ideal Gases Using Historic Techniques

Kevin Sohn (260782138), Lambert Francis (260861226)

McGill University Department of Physics

June 15, 2022

Tuesday Section

Abstract

In this experiment, we determined the specific heat ratio of four ideal gases (argon, nitrogen, carbon dioxide, air) with two historic methods: method of Clément and Desormes, and method of Rüchardt. Using Clément and Desormes's method, we measured the gas pressures before and after venting, (P_0 and P_1, respectively), and calculated γ with the equation $\gamma = \frac{\ln \frac{P_0}{H}}{\ln \frac{P_0}{P_1}}$, where H is the ambient atmospheric pressure. Using Rüchardt's method, we determined the period of a graphite cylinder (τ) oscillating in a gaseous medium, using the equation, $\gamma = \frac{4\pi^2 mV}{A^2 P \tau^2}$, to calculate γ_{air}. Clement and Desormes's method yielded $\gamma_{Ar} = 1.29 \pm 0.04$, $\gamma_N = 1.071 \pm 0.001$, and $\gamma_{CO_2} = 1.094 \pm 0.003$; Rüchardt's method yielded $\gamma_{air} = 1.338 \pm 0.003$. The previously stated results are all more than 3σ below accepted values. We believe this discrepancy is the result of systematic errors present in the experimental setup, causing under-estimation of our γ values.

1 Introduction

In the 18th-century, a scientist named Joseph Black noticed that equal masses of different substances required different amounts of heat to raise them to the same temperature [1]. This observation led to the definition of specific heat capacity (C), which represents the quantity of heat required per unit mass per unit of temperature increase for a certain substance.

The specific heat ratio is defined as

$$\gamma = \frac{C_p}{C_v} = \frac{n+2}{n}, \tag{1}$$

where C_p is specific heat at constant pressure, C_v is specific heat at constant volume, and n is the number of degrees of freedom for a molecule of gas. If the gas is monatomic, then $n = 3$ and $\gamma = \frac{5}{3}$. If the gas is diatomic, then $n = 5$ and $\gamma = \frac{7}{5}$. If the gas is polyatomic, then $n = 6$ and $\gamma = \frac{4}{3}$.

In 1819, Clément and Desormes designed an experiment that made it possible to experimentally estimate the value of γ for an ideal gas [2]. By cleverly exploiting the properties of adiabatic compression and the resulting cooling of gas inside the flask, they expressed γ as a function of initial pressure (P_0), final pressure (P_1), and the atmospheric pressure (H) from their manometer, where

$$\gamma = \frac{\ln \frac{P_0}{H}}{\ln \frac{P_0}{P_1}} \tag{2}$$

is the new relation.

Almost a hundred years later, in 1929, a scientist named Rüchardt came up with an ingenious new method to estimate the value of γ. It involved inducing a graphite cylinder of mass m to initiate simple harmonic motion within a gas [3]. The pressure difference created by the mass compressing the gas causes it to oscillate at a certain frequency. This simple setup allowed Rüchardt to express γ as a function of the mass's period (τ), container volume (V), cross-sectional area of the tube (A), and pressure (P), where

$$\gamma = \frac{4\pi^2 m V}{A^2 P \tau^2} \tag{3}$$

is the new relation.

The purpose of our experiment is to follow the footsteps of these three scientists and utilize their methods to measure the γ of four different ideal gases: argon, nitrogen, carbon dioxide, and air. Then, we compare our experimentally determined γ with values predicted by theory and values obtained from literature.

2 Materials and Methods

2.1 Clément and Desormes's Method

This experiment used a PASCO PASPORT Dual Pressure Port sensor in order to read the pressure inside a sealed ventable box, capable of storing gas up to 120 kPa. The vent was operated manually via a lever. A schematic of the apparatus setup can be found in Figure 1.

Initially, the box was filled with gas until the pressure hovered around 115 - 120kPa. After thermal equilibrium was achieved, we recorded the pressure as P_0. Then, the box was vented for approximately one second and left to reach thermal equilibrium once more. This new pressure inside the box was recorded as P_1. This process was repeated ten times for each gas. However, unless the pressure inside the box reached below 105kPa, we repeated the venting part of the procedure to save gas and time, setting P_0 equal to the previous P_1 in this case. The specific heat ratios for each gas were determined by Eq. 2.

2.2 Rüchardt's Method

Rüchardt's method was performed using a PASCO PASPORT Motion sensor and a machined graphite cylinder fit into a tube such that it acted as a piston. The cylinder was raised using a handheld squeeze pump. A schematic of the setup can be found in Figure 2.

The cylinder was set into motion by squeezing the pump. Using the data from the motion sensor, the period was determined by dividing the total number of oscillations by the time it took for those oscillations to occur. The mass and diameter of the cylinder and the container volume are known quantities and were used in Eq. 3 to calculate γ_{air}. Finally, from Eq. 1, we determined the number of degrees of freedom of air (n_{air}).

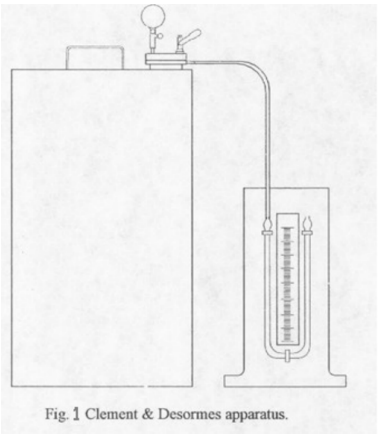


Figure 1: Clément and Desormes setup [4].

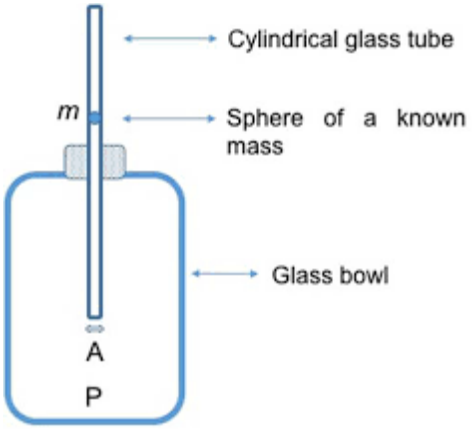


Figure 2: Rüchardt setup [5].

3 Results

3.1 Determination of γ using Clément and Desormes's Method

Data Analysis was performed in Python using Jupyter notebook and the code can be found in Appendix B. Errors were taken to be $\pm$ last digit for digital devices; all pressure uncertainties were 0.01 kPa. All of the errors were propagated using the differential calculus method ($s_f = \sqrt{(\frac{\partial f}{\partial x})^2 s_x^2 + (\frac{\partial f}{\partial y})^2 s_y^2 + ...}$). Linear fits were performed with the scipy.optimize library in Python and uncertainties were adjusted accordingly for the linearized data.

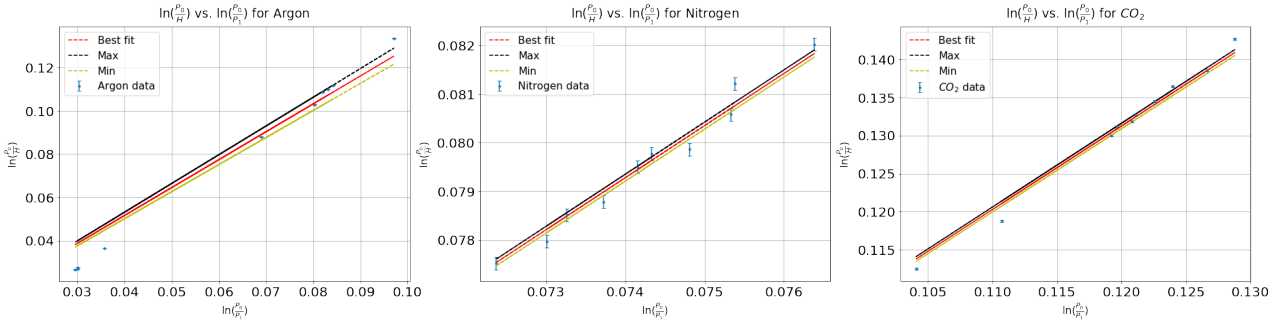


Figure 3: Plot of $\ln \frac{P_0}{H} = \gamma \ln \frac{P_0}{P_1}$ for each gas. The specific heat ratio (γ) was extracted from the slope of the linear fit, where P_0 is the initial venting pressure, P_1 is the pressure at thermal equilibrium after venting, and H is the atmospheric pressure. Two additional linear fits are shown for each gas representing the uncertainty in γ.

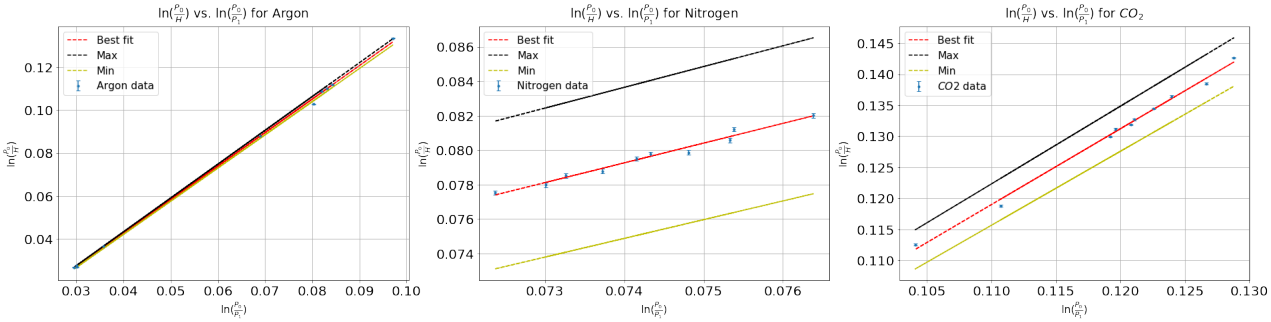


Figure 4: Plot of $\ln \frac{P_0}{H_0} = \gamma \ln \frac{P_0}{P_1} + b$ for each gas. The specific heat ratio (γ) was extracted from the slope of the linear fit, where P_0 is the initial venting pressure, P_1 is the pressure at thermal equilibrium after venting, and H is the atmospheric pressure. Two additional linear fits are shown for each gas representing the uncertainty in γ.

Data	γ_{Ar}	γ_N	γ_{CO_2}	γ_{air}
Fixed y-Intercept Fit	1.29 ± 0.04	1.071 ± 0.001	1.094 ± 0.003	
Free y-Intercept Fit	1.56 ± 0.01	1.15 ± 0.06	1.22 ± 0.03	
Class	1.5 ± 0.4	1.3 ± 0.1	1.2 ± 0.2	1.3 ± 0.1

Table 1: Specific heat ratios extracted from the linear fits in Figures 3 & 4 and from the class averages.

3.2 Determination of γ_{air} using Rüchardt's Method

The mass, diameter, and area of the cylinder, the container volume, and the air pressure were known previous to the experiment. They were $M = (5.612 \pm 0.001) \times 10^{-3}\ kg$, $D = 1.649 \pm 0.002\ cm$, $A = 2.136 \pm 0.005\ cm^2$, $V = (9.612 \pm 0.008) \times 10^{-3}\ m^3$, and $P = 102900 \pm 10\ Pa$, respectively. The mean period of the cylinder's motion, $\tau = 0.582 \pm 0.001\ s$, was calculated by dividing the number of oscillations by the amount of time it took for those oscillations to occur. Using Eq. 3, the specific heat ratio was calculated to be $\gamma_{air} = 1.338 \pm 0.003$.

4 Discussion

The specific heat ratios obtained from literature are $\gamma_{Ar} = 1.667$, $\gamma_N = 1.400$, $\gamma_{CO_2} = 1.289$, and $\gamma_{air} = 1.40$ [6]. From Table 1, we observe that none of our results overlap with accepted values within one standard deviation (σ). In fact, all gases except CO_2 are further than three

σ, demonstrating inconsistency with the accepted values.

Our data is not consistent with having the y-intercept (b) equal zero; allowing fits with a free y-intercept parameter resulted in a statistically significant difference in γ. We note that fits with $b = 0$ result in γ's further from accepted value which is possibly due to inherent systematic underestimation by the equipment.

Comparing the fixed y-intercept values with the class average values from Table 1, we see that all values are within 1σ except nitrogen, which is within 3σ. Even though our data set was consistently lower than the class data, we do not appear to be the outliers since the majority of our measurements are within 1σ.

γ_{air} was determined to be 1.338 ± 0.003, thus, by Eq. 1, $n_{air} = 6$. Also by Eq. 1, $\gamma_N = 1.071 \pm 0.001$ yields $n_N = 28$. n_N being this large indicates under-estimation of γ_N because n_N is inconsistent with the theoretical predictions for diatomic gas ($n = 5$). The accepted value for nitrogen yields $n_N = 5$, which is similar to $n_{air} = 6$. n_{air} being close to n_N is unsurprising because air is primarily composed of two diatomic gases (majority Nitrogen). In fact, our expectation was to have $\gamma_N = \gamma_{air}$ because γ_{air} is identical to γ_N to two decimal places.

5 Conclusions

In this paper, we explored two different methods of determining γ for gases: Clément and Desormes's method for argon, nitrogen, and carbon dioxide, and Rüchardt's method for air. In general, our data did not agree with accepted values; the majority of the specific heat ratios we obtained were more than 3σ away from accepted values. Since the accepted values are from experiments that have reliably determined γ, we attribute our discrepancies to systematic underestimations in our setup. We suspect multiple sources of systematic error: gas leakage, inconsistent venting, and inherent measurement uncertainties present in the PASCO system. The experiment could be improved significantly by minimizing systematic error. We note all our uncertainties should be ≥ 0.01 due to PASCO's limited resolution, but are not, because nitrogen & CO_2 have small data ranges as shown in Figure 3.

Author Contribution Statement: K.S and L.F contributed equally to the report.

References

[1] Rüchardt experiment - Wikipedia. URL https://en.wikipedia.org/wiki/R{\unhbox\voidb@x\bgroup\accent127u\penalty\@M\hskip\z@skip\egroup}chardt{_}experiment. 1

[2] Specific heat capacity - Wikipedia, . URL https://en.wikipedia.org/wiki/Specific{_}heat{_}capacity. 1

[3] Specific heat — physics — Britannica, . URL https://www.britannica.com/science/specific-heat. 1

[4] Practical 2-Clement and Desormes. URL https://dokumen.tips/documents/practical-2-clement-and-desormes.html. 3

[5] M. T. Caccamo, G. Castorina, F. Catalano, and S. Magazù. Rüchardt's experiment treated by Fourier transform. *European Journal of Physics*, 40(2), jan 2019. ISSN 13616404. doi: 10.1088/1361-6404/aaf66c. 3

[6] Specific Heat and Individual Gas Constant of Gases. URL https://www.engineeringtoolbox.com/specific-heat-capacity-gases-d{_}159.html. 4

A Lab Notebook

Lab 1: Ratio of Specific Heats of a Gas

★ Want $\gamma = \frac{c_p}{c_v} = \frac{nR}{n}$.

Connected the air hose to channel 1 on PASCO.
Connected the argon hose to channel 2 on PASCO.

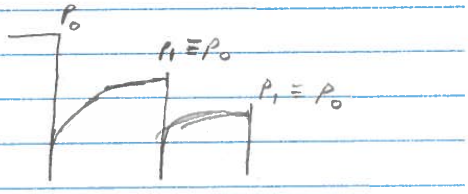
Atmospheric pressure from barometer in class
→ $H = 102.6$ kPa

→ TA told us to have $P_0 \simeq 115$ kPa and set $P_0 = P_1$ after first trial in order to save time and gas.
↳ Apparently shouldn't make a differences as long as the pressure goes down to atmospheric pressure each trial.
↳ Source of error?

Air is made up a collection of elements so What is n=?

Argon:

① $P_0 = 112.03$ kPa, $P_1 = 104.57$ kPa



② $P_0 = 117.26$ kPa, $P_1 = 106.41$ kPa ← uncertainty = 0.01 since that is the last digit displayed by PASCO.

③ $P_0 = 106.41$ kPa, $P_1 = 102.68$ kPa

④ $P_0 = 114.38$ kPa, $P_1 = 105.37$ kPa ★ we used the wrong gas to begin with so we had to restart an hour or so in the lab.

⑤ $P_0 = 105.37$ kPa, $P_1 = 102.30$ kPa

⑥ $P_0 = 114.53$ kPa, $P_1 = 105.39$ kPa

⑦ $\begin{cases} P_0 = 105.39 \text{ kPa} \\ P_1 = 102.26 \text{ kPa} \end{cases}$

⑧ $\begin{cases} P_0 = 114.74 \text{ kPa} \\ P_1 = 105.47 \text{ kPa} \end{cases}$

⑨ $\begin{cases} P_0 = 105.47 \text{ kPa} \\ P_1 = 102.34 \text{ kPa} \end{cases}$

⑩ $\begin{cases} P_0 = 113.71 \text{ kPa} \\ P_1 = 104.93 \text{ kPa} \end{cases}$

error sources: digital reader on PASCO not being precise or systemic

Python → $\gamma_{avg} \simeq 1.163$

$\gamma_{expected} = \frac{5}{3} \simeq 1.66\overline{}$

Nitrogen:

	P_0	P_1
1	110.98	103.14
2	110.87	103.13
3	111.01	103.12
4	111.12	103.16
5	110.92	103.11
6	111.13	103.12
7	111.28	103.20
8	111.09	103.15
9	111.37	103.18
10	111.21	103.14

$\gamma_{avg} \simeq 1.08$

$\gamma_{exp} \simeq \frac{7}{5} = 1.4$

CO_2:

	P_0	P_1
1	117.16	103.80
2	116.98	103.79
3	114.82	103.47
4	118.33	104.03
5	117.37	103.83
6	117.60	103.89
7	116.84	103.71
8	117.06	103.74
9	115.54	103.43
10	117.84	103.82

$\gamma_{avg} \simeq 1.09$

$\gamma_{exp} = \frac{4}{3} \simeq 1.33\overline{}$

Part 2:

1) Cycle = 7 peaks in total.
   ⇒ # of oscillations = 8
   time " " " = 4.670 (s)

2) # of osc. = 8
   time of osc. = 4.660 (s)

3) # = 7
   time = 4.070 (s)

4) # = 8
   time = 4.660 (s)

5) # = 9
   time = 5.240 (s)

6) # = 8
   time = 4.740 (s)

7) # = 7
   time = 4.100 (s)

8) # = 8
   time = 4.640 (s)

9) # = 9
   time = 5.220 (s)

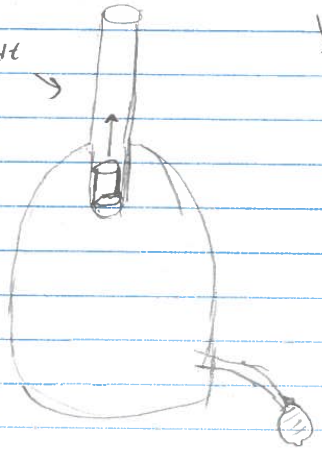
10) # = 8
    time = 4.640 (s)

mass of cylinder = 0.005612 kg
Δm = 1×10^{-6} kg
diameter = 1.649 cm
Δd = 0.002 cm
V = 0.0096119 m^3
ΔV = 8×10^{-6} m^3

$$\ln\left(\frac{P_0}{H}\right) = \ln(P_0) - \ln(H)$$

$$\alpha\left(\frac{P_0}{H}\right) = \sqrt{\left(\frac{1}{P_0}\right)^2(\Delta P_0)^2 + \left(\frac{1}{H}\right)^2(\Delta H)^2}$$

} error propagation (differential)



↳ error and mean propagated and calculated in Jupyter Notebook.

B Python Code

In [109]:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.optimize as opt

plt.rcParams['figure.figsize'] = [19, 25]
plt.rc('font', size = 15)
plt.rc('xtick', labelsize = 15)
plt.rc('ytick', labelsize = 15)
```

Clément and Desormes

In [110]:

```
# Importing data for part 1
argon = np.loadtxt('argon.csv', delimiter = ',');
nitrogen = np.loadtxt('nitrogen.csv', delimiter = ',');
co2 = np.loadtxt('co2.csv', delimiter = ',');

Ar_p1, Ar_p0 = np.transpose(argon);
N_p1, N_p0 = np.transpose(nitrogen);
CO2_p1, CO2_p0 = np.transpose(co2);
H = 102.6;
```

In [111]:

```
# Calculating components for the plots
Ar_ln1 = np.log(Ar_p0/H);
Ar_ln2 = np.log(Ar_p0/Ar_p1);

N_ln1 = np.log(N_p0/H);
N_ln2 = np.log(N_p0/N_p1);

CO2_ln1 = np.log(CO2_p0/H);
CO2_ln2 = np.log(CO2_p0/CO2_p1);
```

In [112]:

```
# Calculating error on measurements using the differential appro
ach
err_p0 = .01; # (kPa)
err_H = .01;  # (kPa) read from barometer in classroom.

def getErr(p0, error_p0, H, error_H):
    err = np.sqrt((1/p0)**2 * error_p0**2 + (1/H)**2 * error_H**
2)
    return err

Ar_ln1_err = getErr(Ar_p0, err_p0, H, err_H);
N_ln1_err = getErr(N_p0, err_p0, H, err_H);
CO2_ln1_err = getErr(CO2_p0, err_p0, H, err_H);
```

In [113]:

```
# Linear model function with b = 0
def f1(x, m):
    return m*x
```

In [114]:

```
# Finding best fit paramters
param1_Ar, cov1_Ar = opt.curve_fit(f1, Ar_ln2, Ar_ln1, p0 = [1])
;
param1_N, cov1_N = opt.curve_fit(f1, N_ln2, N_ln1, p0 = [1]);
param1_CO2, cov1_CO2 = opt.curve_fit(f1, CO2_ln2, CO2_ln1, p0 =
[1]);

print(param1_Ar, param1_N, param1_CO2);
print(np.sqrt(cov1_Ar), np.sqrt(cov1_N), np.sqrt(cov1_CO2));
```

```
[1.29026037] [1.07134467] [1.09358173]
[[0.03810421]] [[0.00096595]] [[0.00299448]]
```

$\gamma_{Ar} = 1.29 \pm 0.04 \gamma_{Ar} = 1.29 \pm 0.04$
$\gamma_{N} = 1.071 \pm 0.001 \gamma_{N} = 1.071 \pm 0.001$
$\gamma_{CO_2} = 1.094 \pm 0.003 \gamma_{CO_2} = 1.094 \pm 0.003$

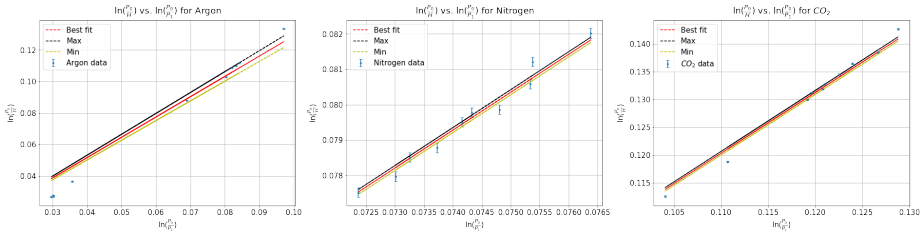
In [115]:

```
# Plotting raw data and linear fit with b = 0
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize = (32,7));

ax1.errorbar(Ar_ln2, Ar_ln1, yerr = Ar_ln1_err, fmt = '.', capsi
ze = 3, label = 'Argon data');
ax1.plot(Ar_ln2, f1(Ar_ln2, param1_Ar), "r--", label = 'Best fit
');
ax1.plot(Ar_ln2, f1(Ar_ln2, param1_Ar+np.sqrt(cov1_Ar.flatten())
), "k--", label = 'Max');
ax1.plot(Ar_ln2, f1(Ar_ln2, param1_Ar-np.sqrt(cov1_Ar.flatten())
), "y--", label = 'Min');
ax1.set_title(r'$\ln(\frac{P_0}{H})$ vs. $\ln(\frac{P_0}{P_1})$
for Argon');
ax1.set_xlabel(r'$\ln(\frac{P_0}{P_1})$');
ax1.set_ylabel(r'$\ln(\frac{P_0}{H})$');
ax1.grid();
ax1.legend();

ax2.errorbar(N_ln2, N_ln1, yerr = N_ln1_err, fmt = '.', capsize
= 3, label = 'Nitrogen data');
ax2.plot(N_ln2, f1(N_ln2, param1_N), "r--", label = 'Best fit');
ax2.plot(N_ln2, f1(N_ln2, param1_N+np.sqrt(cov1_N.flatten())), "
k--", label = 'Max');
ax2.plot(N_ln2, f1(N_ln2, param1_N-np.sqrt(cov1_N.flatten())), "
y--", label = 'Min');
ax2.set_title(r'$\ln(\frac{P_0}{H})$ vs. $\ln(\frac{P_0}{P_1})$
for Nitrogen');
ax2.set_xlabel(r'$\ln(\frac{P_0}{P_1})$');
ax2.set_ylabel(r'$\ln(\frac{P_0}{H})$');
ax2.grid();
ax2.legend();

ax3.errorbar(CO2_ln2, CO2_ln1, yerr = CO2_ln1_err, fmt = '.', ca
psize = 3, label = r'$CO_2$ data');
ax3.plot(CO2_ln2, f1(CO2_ln2, param1_CO2), "r--", label = 'Best
fit');
ax3.plot(CO2_ln2, f1(CO2_ln2, param1_CO2+np.sqrt(cov1_CO2.flatte
n())), "k--", label = 'Max');
ax3.plot(CO2_ln2, f1(CO2_ln2, param1_CO2-np.sqrt(cov1_CO2.flatte
n())), "y--", label = 'Min');
ax3.set_title(r'$\ln(\frac{P_0}{H})$ vs. $\ln(\frac{P_0}{P_1})$
for $CO_2$')
ax3.set_xlabel(r'$\ln(\frac{P_0}{P_1})$')
ax3.set_ylabel(r'$\ln(\frac{P_0}{H})$')
```

```
ax3.grid()

ax3.legend()
```

Out[115]:

```
<matplotlib.legend.Legend at 0x101dd744e0>
```



In [116]:

```
# Linear model function
def f2(x, m, b):
    return m*x + b
```

In [117]:

```
# Finding best fit parameters using scipy
param2_Ar, cov2_Ar = opt.curve_fit(f2, Ar_ln2, Ar_ln1, p0 = [1,1
]);
param2_N, cov2_N = opt.curve_fit(f2, N_ln2, N_ln1, p0 = [1,1]);
param2_CO2, cov2_CO2 = opt.curve_fit(f2, CO2_ln2, CO2_ln1, p0 =
[1,1]);

print(param2_Ar, param2_N, param2_CO2);
print(np.sqrt(cov2_Ar[0][0])); # error on slope obtained from co
variance matrix
print(np.sqrt(cov2_N[0][0]));
print(np.sqrt(cov2_CO2[0][0]));
```

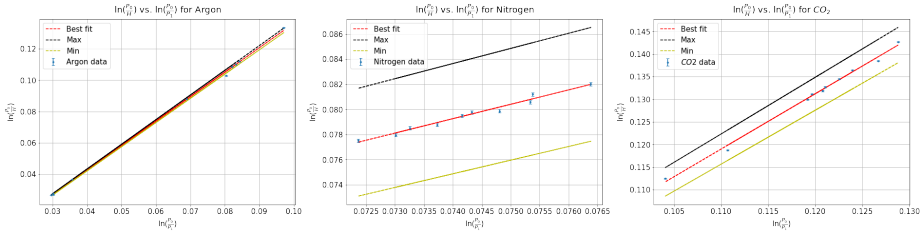
```
[ 1.56112482 -0.01976644] [ 1.14765404 -0.00566936]
[ 1.22209285 -0.01544156]
0.014986128880961437
0.05931668934043129
0.03030303321649553
```

$\gamma_{Ar} = 1.56 \pm 0.02 \gamma_{Ar} = 1.56 \pm 0.02$
$\gamma_{N} = 1.15 \pm 0.06 \gamma_{N} = 1.15 \pm 0.06$
$\gamma_{CO_2} = 1.22 \pm 0.03 \gamma_{CO_2} = 1.22 \pm 0.03$

In [118]:

```
# Plotting raw data and linear fit
fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize = (32,7));

ax1.errorbar(Ar_ln2, Ar_ln1, yerr = Ar_ln1_err, fmt = '.', capsi
ze = 3, label = 'Argon data');
ax1.plot(Ar_ln2, f2(Ar_ln2, param2_Ar[0], param2_Ar[1]), "r--",
label = 'Best fit');
ax1.plot(Ar_ln2, f2(Ar_ln2, param2_Ar[0]+np.sqrt(cov2_Ar[0][0]),
param2_Ar[1]), "k--", label = 'Max');
ax1.plot(Ar_ln2, f2(Ar_ln2, param2_Ar[0]-np.sqrt(cov2_Ar[0][0]),
param2_Ar[1]), "y--", label = 'Min');
ax1.set_title(r'$\ln(\frac{P_0}{H})$ vs. $\ln(\frac{P_0}{P_1})$
for Argon');
ax1.set_xlabel(r'$\ln(\frac{P_0}{P_1})$');
ax1.set_ylabel(r'$\ln(\frac{P_0}{H})$');
ax1.grid();
ax1.legend();

ax2.errorbar(N_ln2, N_ln1, yerr = N_ln1_err, fmt = '.', capsize
= 3, label = 'Nitrogen data');
ax2.plot(N_ln2, f2(N_ln2, param2_N[0], param2_N[1]), "r--", labe
l = 'Best fit');
ax2.plot(N_ln2, f2(N_ln2, param2_N[0]+np.sqrt(cov2_N[0][0]), par
am2_N[1]), "k--", label = 'Max');
ax2.plot(N_ln2, f2(N_ln2, param2_N[0]-np.sqrt(cov2_N[0][0]), par
am2_N[1]), "y--", label = 'Min');
ax2.set_title(r'$\ln(\frac{P_0}{H})$ vs. $\ln(\frac{P_0}{P_1})$
for Nitrogen');
ax2.set_xlabel(r'$\ln(\frac{P_0}{P_1})$');
ax2.set_ylabel(r'$\ln(\frac{P_0}{H})$');
ax2.grid();
ax2.legend();

ax3.errorbar(CO2_ln2, CO2_ln1, yerr = CO2_ln1_err, fmt = '.', ca
psize = 3, label = r'$CO2$ data')
ax3.plot(CO2_ln2, f2(CO2_ln2, param2_CO2[0], param2_CO2[1]), "r-
-", label = 'Best fit');
```

```
ax3.plot(CO2_ln2, f2(CO2_ln2, param2_CO2[0]+np.sqrt(cov2_CO2[0][
0]), param2_CO2[1]), "k--", label = 'Max');
ax3.plot(CO2_ln2, f2(CO2_ln2, param2_CO2[0]-np.sqrt(cov2_CO2[0][
0]), param2_CO2[1]), "y--", label = 'Min');
ax3.set_title(r'$\ln(\frac{P_0}{H})$ vs. $\ln(\frac{P_0}{P_1})$
for $CO_2$');
ax3.set_xlabel(r'$\ln(\frac{P_0}{P_1})$');
ax3.set_ylabel(r'$\ln(\frac{P_0}{H})$');
ax3.grid();
ax3.legend();
```



Class Data

In [119]:

```
# Importing class data
class_mean = np.loadtxt("class_mean.csv", delimiter = ',');
class_err = np.loadtxt("class_err.csv", delimiter = ',');

Ar, N, CO2, air = np.array(np.transpose(class_mean));
Ar_err, N_err, CO2_err, air_err = np.array(np.transpose(class_er
r));
```

In [120]:

```
# Calculating class average standard deviation for all gases
Ar_std = np.std(Ar, ddof = len(Ar)-1);
N_std = np.std(N, ddof = len(N)-1);
CO2_std = np.std(CO2, ddof = len(CO2)-1);
air_std = np.std(air, ddof = len(air)-1);

# Calculating class average error for all gases
Ar_stderr = Ar_std/np.sqrt(len(Ar));
N_stderr = N_std/np.sqrt(len(N));
CO2_stderr = CO2_std/np.sqrt(len(CO2));
air_stderr = air_std/np.sqrt(len(air));

print(Ar_stderr);
print(N_stderr);
print(CO2_stderr);
print(air_stderr);
```

```
0.364405105801518
0.12986683728085016
0.16270339174768417
0.11711921946887328
```

In [121]:

```
# Calculating the class average mean for all gases
Ar_mean = np.mean(Ar);
N_mean = np.mean(N);
CO2_mean = np.mean(CO2);
air_mean = np.mean(air);

print(Ar_mean);
print(N_mean);
print(CO2_mean);
print(air_mean);
```

```
1.5378869565217392
1.2633608695652174
1.2506391304347824
1.287113043478261
```

$\gamma_{Ar,cl} = 1.5 \pm 0.4 \gamma_{Ar,cl} = 1.5 \pm 0.4$
$\gamma_{N,cl} = 1.3 \pm 0.1 \gamma_{N,cl} = 1.3 \pm 0.1$
$\gamma_{CO_2,cl} = 1.2 \pm 0.2 \gamma_{CO_2,cl} = 1.2 \pm 0.2$
$\gamma_{air,cl} = 1.3 \pm 0.1 \gamma_{air,cl} = 1.3 \pm 0.1$

Literature values

$\gamma_{Ar} = 1.667 \gamma_{Ar} = 1.667$
$\gamma_{N} = 1.400 \gamma_{N} = 1.400$
$\gamma_{CO_2} = 1.289 \gamma_{CO_2} = 1.289$
$\gamma_{air} = 1.40 \gamma_{air} = 1.40$

https://www.engineeringtoolbox.com/specific-heat-capacity-gases-d_159.html (https://www.engineeringtoolbox.com/specific-heat-capacity-gases-d_159.html)

Ruchardt's Method Air Specific Heat Ratio Analysis

In [122]:

```
# Importing data for part 2
air = np.loadtxt('air.csv', delimiter = ',');
periods, gammas, times, peaks = np.transpose(air);
M, D, V = 0.005612, 1.649, 0.009612 ## Kg, cm, m^3
error_M, error_D, error_V = 1e-6, 0.002, 8e-6

# Atmospheric pressure
P0, error_P0 = 102.6, 0.1
P0, error_P0 = 102.6*1000, 0.1*1000

g = 9.81;
P = P0 + M*g/A;

periods = times/(peaks+1) # Since number of peaks in between is
1 less than number of oscillations.

# Calculating Area Tube ( A = pi *(D/2)^2)
error_A = np.sqrt((np.pi/2*D**2 * error_D**2))
A = np.pi * (D/2)**2 # cm squared
A = (A /100) / 100    # m squared
error_A = (error_A/100) /100

error_P = error_P0**2 + (9.8/A)**2 * error_M**2 + (M*9.8/A**2)**
2 * error_A**2
error_P = np.sqrt(error_P)

## Period best estimate and uncertainty
period_mean = np.mean(periods)
periods_std = np.std(periods, ddof = len(periods)-1)
period_error = periods_std / np.sqrt(len(periods))
```

Gamma calculation using $\gamma = \frac{4\pi^2 mV}{A^2 P\tau^2}$ $\gamma = \frac{4\pi^2 mV}{A^2 P\tau^2}$ Meaning the error is

$\sigma_\gamma^2 = (4\pi)^2[(\frac{V}{A^2 P\tau^2})^2\sigma_{m^2} + (\frac{m}{A^2 P\tau^2})^2\sigma_{v^2} + (\frac{mV}{A^2 P^2\tau^2})^2\sigma_{P^2} + (\frac{mV}{A^3 P\tau^2})^2\sigma_{A^2} + (\frac{mV}{A^2 P\tau^3})^2$

$\sigma_\gamma^2 = (4\pi)^2[(\frac{V}{A^2 P\tau^2})^2\sigma_{m^2} + (\frac{m}{A^2 P\tau^2})^2\sigma_{v^2} + (\frac{mV}{A^2 P^2\tau^2})^2\sigma_{P^2} + (\frac{mV}{A^3 P\tau^2})^2\sigma_{A^2} + (\frac{mV}{A^2 P\tau^3})^2\sigma_{T^2}]$ by standard error calculation

In [123]:

```
# gamma
gamma = 4*np.pi**2*M*V / (A**2*P*period_mean**2)

# gamma error
gamma_error = (V/(A**2*P*period_mean**2))**2 * error_M**2  # Fir
st term
gamma_error += (M/(A**2*P*period_mean**2))**2 * error_V**2 # ...
gamma_error += (M*V/(A**2*P**2*period_mean**2))**2 * error_P**2
gamma_error += (M*V/(A**3*P*period_mean**2))**2 * error_A**2
gamma_error += (M*V/(A**2*P*period_mean**3))**2 * period_error**
2 # Last term
gamma_error = gamma_error*(4*np.pi)**2 # Common factor
gamma_error = np.sqrt(gamma_error)

print(gamma_error)
print(gamma)
```

```
0.003007324536795l4
1.3379175198583988
```

$\gamma_{air} = 1.338 \pm 0.003 \gamma_{air} = 1.338 \pm 0.003$