There is no logical way to the discovery of elemental laws. There is only the way of intuition, which is helped by a feeling for the order lying behind the appearance.

-A. Einstein

Physical Chemistry Experiments

Wellesley College, Department of Chemistry

Sandor Kadar

(DRAFT)

Table of Contents

Table of Figures

Experiments 1

IR and Raman Studies of Carbon Dioxide Experimentally and Theoretically

Background

You will embark on a journey of discovering the molecular and thermodynamical properties of molecules in gas phase, for most part using the CO_2 molecule as a model. The roadmap for this adventure is shown in Figure 1.1.

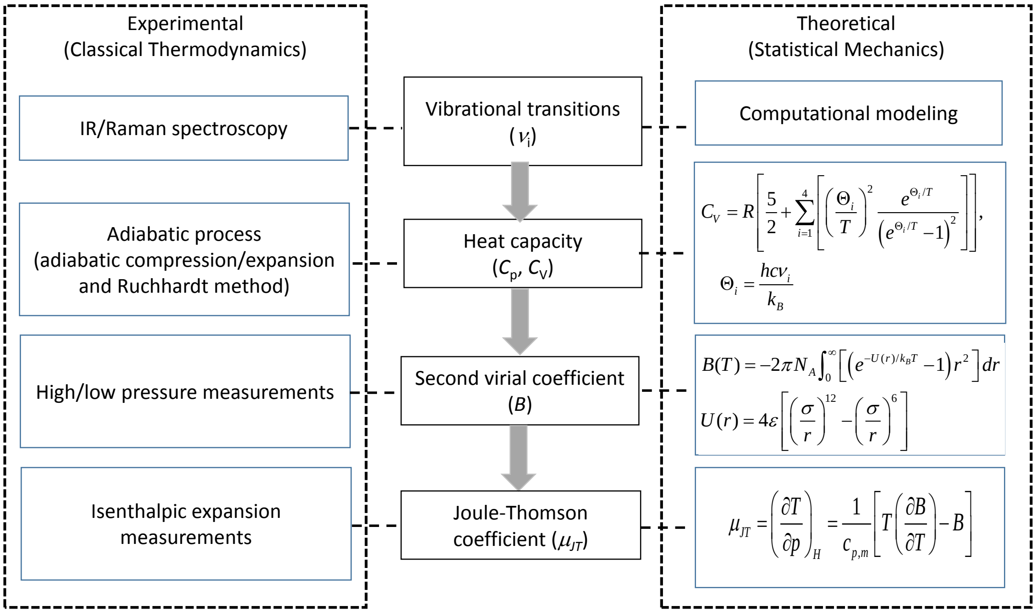

Experimental
(Classical Thermodynamics)
IR/Raman spectroscopy
Adiabatic process
(adiabatic compression/expansion
and Ruchhardt method)
High/low pressure measurements
Isenthalpic expansion
measurements
Vibrational transitions
(ν_i)
Heat capacity
(C_p, C_V)
Second virial coefficient
(B)
Joule-Thomson
coefficient (μ_{JT})
Theoretical
(Statistical Mechanics)
Computational modeling
$C_V = R\left[\frac{5}{2} + \sum_{i=1}^{4}\left[\left(\frac{\Theta_i}{T}\right)^2 \frac{e^{\Theta_i/T}}{\left(e^{\Theta_i/T}-1\right)^2}\right]\right],$
$\Theta_i = \frac{hc\nu_i}{k_B}$
$B(T) = -2\pi N_A \int_0^\infty \left[\left(e^{-U(r)/k_BT}-1\right)r^2\right]dr$
$U(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$
$\mu_{JT} = \left(\frac{\partial T}{\partial p}\right)_H = \frac{1}{c_{p,m}}\left[T\left(\frac{\partial B}{\partial T}\right) - B\right]$

Figure 1.1 Strategy to study molecular and thermodynamical properties of CO_2 experimentally and theoretically

While carbon dioxide is a relatively simple molecule, it has a relatively complex spectroscopic behavior. The main features and some of the details will be examined in this lab to the extent that will help to explore the thermodynamical properties of CO_2.

Considering that the CO_2 molecule is linear, 3*N*-5=4 transitions are expected:

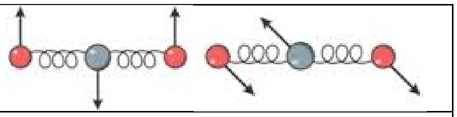
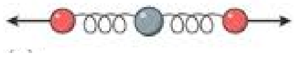
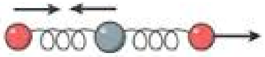
Bending (doubly degenerate) (ν_2)	
Symmetric stretching (ν_1)	
Asymmetric stretching (ν_3)	

Table 1.1 Vibrational modes of CO_2

The numbering of the transitions is as follows (following numbering in the literature):

1: Symmetric stretching
2: Bending
3: Asymmetric stretching

Furthermore, indicating the rotational transitions, the ν_i^J(n) nomenclature will be used, where *n* is the vibrational level (*n* = 0, 1, ….), *J* is the rotational level (*J* = 0, 1, 2,….), and *i* is the identification of the transition as listed above (*i* = 1, 2, 3).

Energetically these transitions all fall into the IR range, however there are some differences. While for a transition to be IR active it has to change the dipole moment of the molecule, a transition that changes the polarizability of a molecule, it will be Raman active. This fact reveals two important points about IR and Raman methods: one, they complement each other, and two, they leverage different properties of molecules. That said, it is important to keep in mind, that regardless of the detecting technique (as to what properties the method is based on), transitions are present regardless. The result of this is quite surprising: transitions that are not IR active can and will interact with IR active transition yielding new IR or Raman active transitions (see Fermi resonance below).

In comparison to the HCl/DCl system that you already studied before, there are some further differences. In the HCl/DCl there was only a "parallel mode" (that is a transition parallel with the axis of the molecule), the $\Delta J = 0$ transition was prohibited, therefore the fundamental frequency of the transition was missing from the spectrum. However, in the CO_2 molecule there are "perpendicular modes" (that is modes perpendicular to the axis of the molecule, namely the two bending modes), the $\Delta J = 0$ selection rule is allowed for other than the $\nu_i^0(0) \rightarrow \nu_i^0(1)$ transition. The transitions with the $\Delta J = 0$ selection rule are referred to as the *Q branch* (and as before the $\Delta J = +1$ transitions constitute the *R branch*, and the $\Delta J = -1$ the *P branch*). Because the transitions associated with the Q branch have only vibrational component, they have the same energy and they all show up at the same frequency, that is at the respective fundamental frequencies.

Another noticeable difference is the missing odd numbered rotational levels for *n* = 0. The explanation for this “odd” phenomenon is beyond the scope of this lab (see Ref. [1]).

The IR Spectrum

A typical IR spectrum of CO_2 is shown in Figure 1.2. The bending (ν_2) and asymmetric stretch (ν_2) transitions are prevailing and some of the combination bands also are identifiable. Most interesting is that they are a result of an IR inactive band (symmetric stretch, ν_1) and an IR active band (ν_2 or ν_3). The fine structure around ν_2 will be addressed later.

Raman Spectrum

A typical Raman spectrum is shown in Figure 1.3. Instead of the expected one transition associated with the symmetric stretch at about ν_1=1336 cm^{-1}, there are two distinct peaks. The reason is that the first harmonic of the bending transition (ν_2 = 667 cm^{-1}) is at almost at the same wave number as the symmetric stretch, and therefore they interact to yield two split peaks due to the phenomenon called *Fermi resonance*. The coupling constant between the two peaks is 110 cm^{-1}. The interesting fact about this phenomenon is that a Raman inactive transition interact with a Raman active one. The conclusion here to be drawn is that transitions do exist regardless of our detecting methods. Just like one of the peaks around 3500 cm^{-1} is a result of an IR active and a Raman active transition.

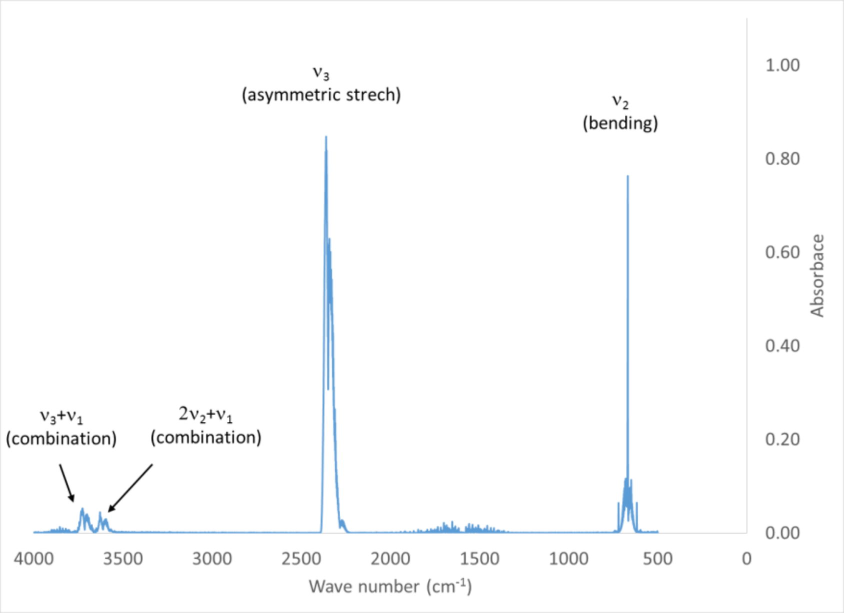


Figure 1.2 IR spectrum of solid carbon dioxide

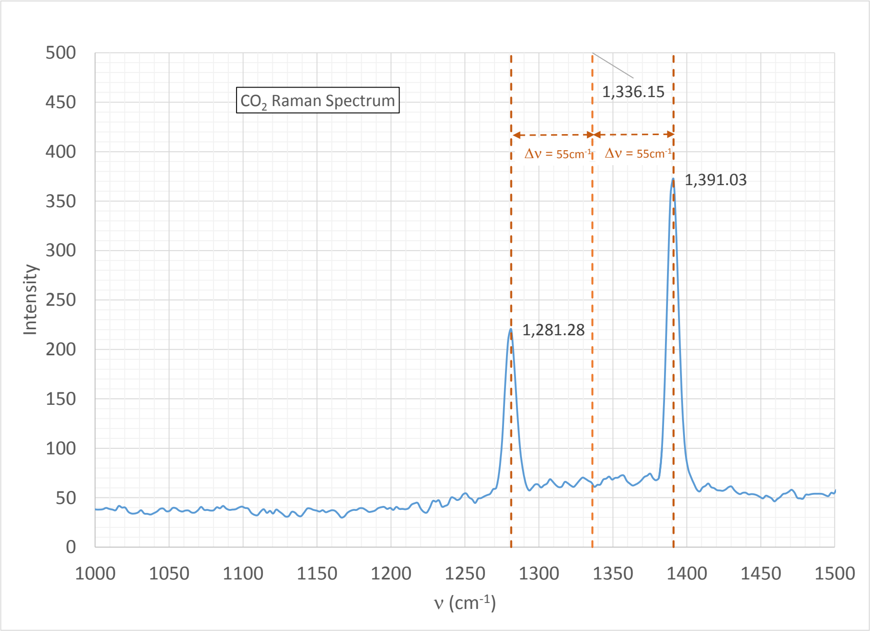


Figure 1.3 Raman spectrum of carbon dioxide

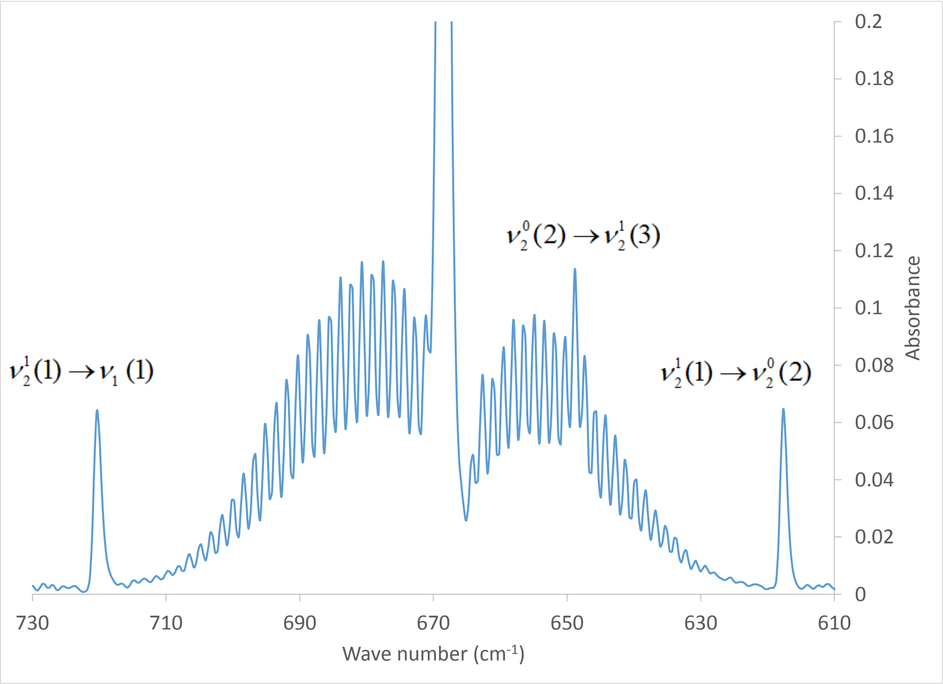


Figure 1.4 Details of the bending transition in the IR spectrum of CO$_2$

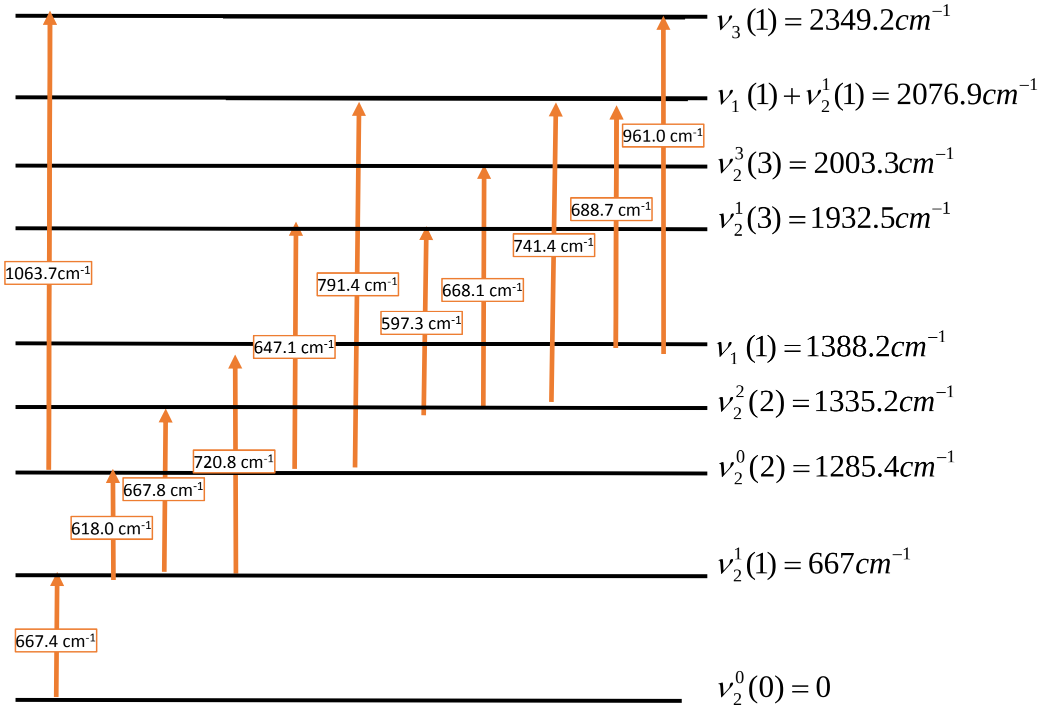

$\nu_3(1) = 2349.2cm^{-1}$
$\nu_1(1) + \nu_2^1(1) = 2076.9cm^{-1}$
$\nu_2^3(3) = 2003.3cm^{-1}$
$\nu_2^1(3) = 1932.5cm^{-1}$
$\nu_1(1) = 1388.2cm^{-1}$
$\nu_2^2(2) = 1335.2cm^{-1}$
$\nu_2^0(2) = 1285.4cm^{-1}$
$\nu_2^1(1) = 667cm^{-1}$
$\nu_2^0(0) = 0$
961.0 cm^{-1}
688.7 cm^{-1}
741.4 cm^{-1}
1063.7cm^{-1}
791.4 cm^{-1}
668.1 cm^{-1}
597.3 cm^{-1}
647.1 cm^{-1}
720.8 cm^{-1}
667.8 cm^{-1}
618.0 cm^{-1}
667.4 cm^{-1}

Figure 1.5 Transitions in the IR/Raman spectrum of CO_2

Details of the bending mode

The details of the IR spectrum of the bending transition is show in Figure 1.4 and the most prevailing transitions are shown in Figure 1.5 and some of the transitions involving this mode are in Figure 1.5 with the respective energy levels, except the $\Delta J = 0$ transitions. Since the $\Delta J = 0$ transitions do not have a rotational component, they all show up at the frequency of the vibrational transition, that is at the same frequency.

Recall that the energy levels are given by the following equation:

$$E = \frac{h}{8\pi^2 Ic} J(J+1) = BJ(J+1), \quad B = \frac{h}{8\pi^2 Ic} \qquad \textbf{(1.1)}$$

where h is the Planck constant, I is the rotational inertia of the molecule, c is the speed of light, B is the collection of all the constants. Based on, the energy level can be characterized in terms of B and the rotational quantum number, J. The table below lists first few transitions in the P and R branches (keep in mind that the odd number of rotational levels are missing for n=0):

P branch					
	Initial level		Final level		**Δ*E***
	J	*E*	*J*	*E*	
$\nu_2^2(0) \rightarrow \nu_2^1(1)$	2	6B	1	2*B*	-4*B*
$\nu_2^4(0) \rightarrow \nu_2^3(1)$	4	20*B*	3	12*B*	-8*B*
$\nu_2^6(0) \rightarrow \nu_2^5(1)$	6	42*B*	5	30*B*	-12*B*
$\nu_2^8(0) \rightarrow \nu_2^7(1)$	8	72*B*	7	56*B*	-16*B*

R branch					
	Initial level		Final level		**Δ*E***
	J	*E*	*J*	*E*	
$\nu_2^0(0) \rightarrow \nu_2^1(1)$	0	0	1	2*B*	2*B*
$\nu_2^2(0) \rightarrow \nu_2^3(1)$	2	6*B*	3	12*B*	6*B*
$\nu_2^4(0) \rightarrow \nu_2^4(1)$	4	20*B*	5	30*B*	10*B*
$\nu_2^6(0) \rightarrow \nu_2^7(1)$	6	42*B*	7	56*B*	14*B*

Table 1.2 Rotational transition energy levels in terms of *B*

From Table 1.2 it is clear, that the rotational levels are staggered 4*B* apart, which means that the rotational peaks are 4*B* apart in both branches. Therefore using the experimental spectrum and eqn. (1.1) the equilibrium bond length of C=O can be estimated:

$$B = \frac{h}{8\pi^2 Ic} = \frac{h}{8\pi^2 2mr^2 c} = \frac{h}{8\pi^2 2\frac{Aw(O)}{N_a} r^2 c} = \frac{hN_a}{8\pi^2 2Aw(O)r^2 c} \rightarrow$$

$$r = \sqrt{\frac{hN_a}{16\pi^2 Aw(O)cB}} = \tag{1.2 }$$

$$= \sqrt{\frac{\left(6.62\times10^{-34}\,\tfrac{kgm}{s^2}\,ms\right)\left(6.022\times10^{23}\,mol^{-1}\right)}{16\left(3.14\right)^2\left(0.016\,\tfrac{kg}{mol}\right)\left(3\times10^8\,\tfrac{m}{s}\right)\left(39.0m^{-1}\right)}} = 1.16\times10^{-10}\,m$$

where $B = 0.39 \pm 0.04\,cm^{-1}$ obtained from average distance between two neighbor peaks in the experimental the spectrum, *r* is the bond length, I=2mr^2 is the rotational inertia of the molecule, *m* is the mass of the oxygen atom.

Computational model of CO_2

The molecular properties of CO_2 can be predicted with a computational modeling quite well. Figure 1.6 shows an overlay of the IR and Raman spectra obtained with a DFT/ B3LYP method and cc-pVTZ basis.

The most noticeable difference is the lack of the rotational details, however the vibrational modes are predicted quite well. Further validation of the model can be accomplished by performing a computational simulation to obtain the potential along the C=O bond which should yield the equilibrium bond length 1.16 Å and bond energy 775 kJ/mol (see Figure 1.7).

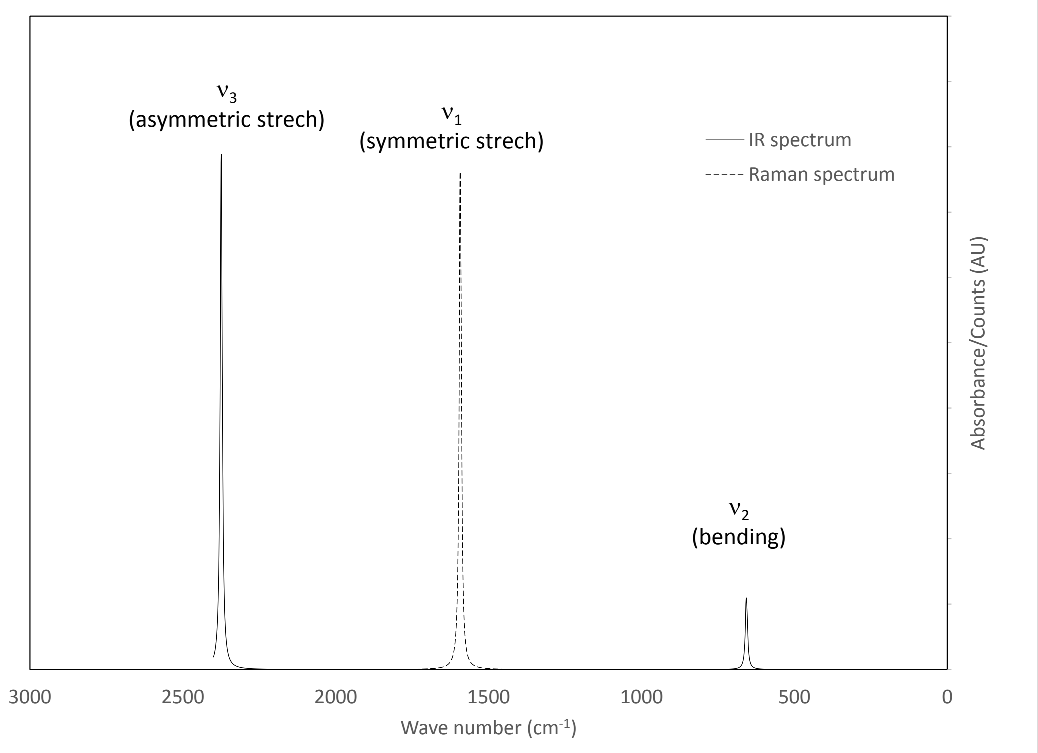


Figure 1.6 Overlaid IR and Raman spectra of CO_2 obtained with computational modeling (DFT/B3LYP method/cc-pVTZ basis)

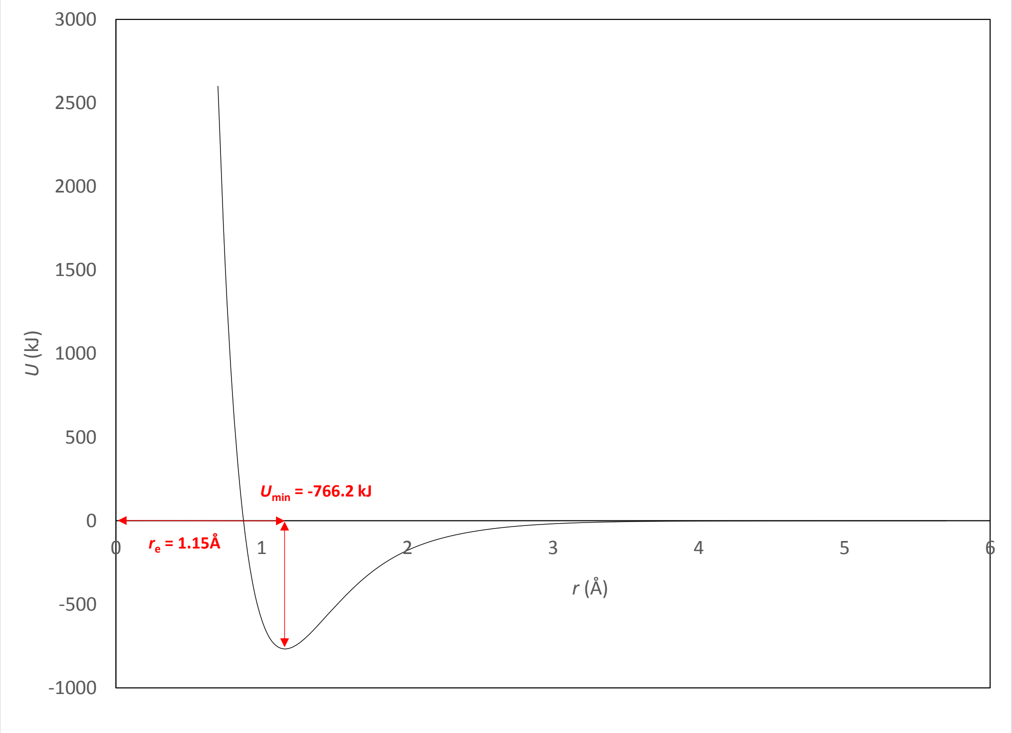


Figure 1.7 Potential scan along the C=O bond (DFT/B3LYP method/cc-pVTZ basis)

One reason for the deviation from the ideal gas behavior is the interaction between molecules which is quantitized with the potential between molecules. The extent of the interaction between two molecules as a function of the distance between them can be accomplished with Gaussian. The Lennard-Jones potential function will be used to describe the potential function, for which the values of σ and ε can be estimated from the graph: the depth of the well is ε and the distance where the potential is zero is σ (see Figure 1.9). You are to perform the potential scan with two method/basis combinations of your choice to estimate the Lennard-Jones constants for CO_2 and you will be provided with the results from Figure 1.9 (this modeling was running for over a week on a server, so be mindful of the time it takes to run these modeling).

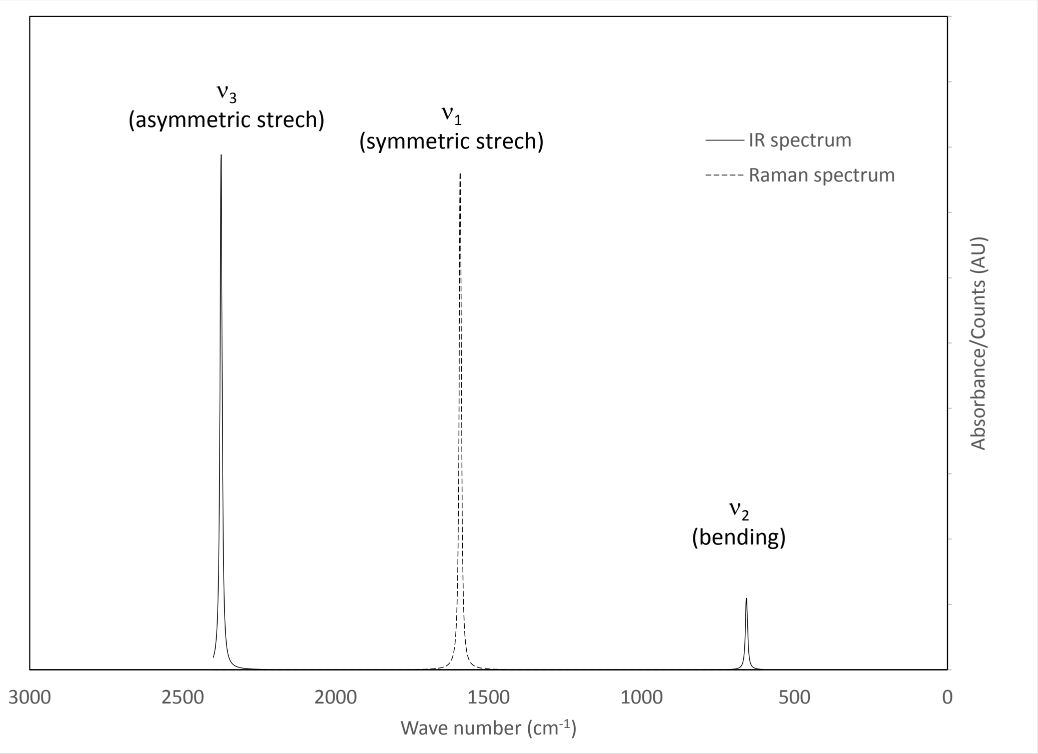


Figure 1.8 Overlaid IR and Raman spectra of CO_2 obtained with computational modeling (DFT/B3LYP method/cc-pVTZ basis)

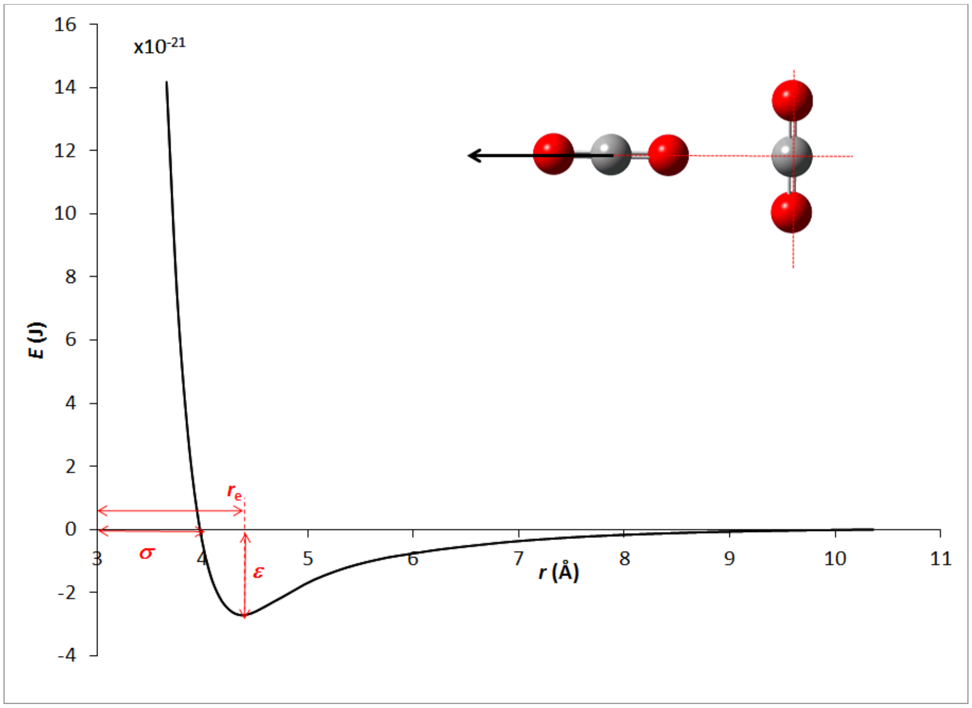


Figure 1.9 Potential scan between two CO_2 molecules as shown in the insert (DFT/B3LYP method/aug-cc-pVTZ basis)

Experimental Procedure

IR Spectroscopy

1. Start the software that operates the Bruker IR Spectrophotometer (the one on the left) and login as **CHEM330L** (password: **pchem**).
2. Remove the gas cell from the desiccator are remove both robber septa.
   Note: the cell has to be stored in the desiccator between experiments as the windows of the cell are made from KBr which is hydroscopic and the windows gets foggy (imparing the optical properties of the windows), and potentially destroyed if exposed to moisture for a long period of time.
3. Flush the cell by allowing He gas to flowing one port and out on the other port for about 30 s.
4. Immediately after stopping the flushing process replace both septa.
5. Take the background scan with the gas cell as prepared (with He in it).
6. Remove the stopper from the needle of the 10-mL gas syringe and fill it with CO_2 by placing the needle into CO_2 gas flow (e.g. from a tubing attached to the gas cylinder) and withdrawing the plunger *slowly.*
   Note: after filling the syringe, put the rubber stopper back on the needle to prevent air and moisture getting in. Before every time you use the syringe to deliver a sample push the plunger forward about 0.2 mL to force the air out that potentially entered the syringe.

7. Find the pin hole on one of the septa and slowly insert and push in the needle of the gas syringe into the pin hole.
   Note: the needle of the gas syringe is dull, the end is closed and there is a pin hole at the end of the needle on the side to allow the gas to flow in and out.
8. Inject about 2 mL gas into the gas cell.
9. Obtain the IR spectrum.
10. Save the spectrum in ASCII, MATLAB, and image format.
11. Replace the cell to the desiccator.

Raman Spectroscopy

The inside view of the Raman Spectrometer (Raman Unit, RU) is shown in Figure 1.10. The sample placed in the sample holder is exposed to a 532nm laser beam. The scattered light at 90^0 is directed to the spectrophotometer detector through a 540nm longpass filter to minimize the stray-beam effect from the exciting beam.

1. Start MATLAB on the attached computer.
2. The function called **OO_Grab** captures a single spectrum from the RU. :

```
function [w s] = OO_Grab(xflag, iTime, iPass, iBox)
    %xflag: flag for type of return scale:
    %   'wl' = wavelength
    %   'wlCO2' = wavelength range relevant to CO2
    %   'wn' = wavenumber
    %   'wnCO2' = wavenumber range relevant to CO2
    %iTime = integration time (in s)
    %iPass = number of spectra to be averaged
    %iBox = number of points to be averaged in one spectrum
```

3. The controls are on the front (see Figure 1.11).Turn on the RU by
   i. Turning on the **Power Switch.**
   ii. Turn the **Key Switch** to the **ON** position.
   iii. Enter the 4 digit code on the keypad. The green Laser On light should come on.
   iv. Pressing “1” on the keypad disables the laser.
4. With the laser enabled, collect the background spectrum.
5. Disable the laser by pressing “1” on the keypad.
   Note: The opening of the sample holder will disengage the laser, however this is a safety feature and not intended for operating the unit. Always disengage the laser by pressing “1” on the keypad.
6. Take several rods of DyIce® from the storage box in the thermostat box.
7. Pick one that fits the sample holder, scrape off the outer layer to remove moisture.
   Note: use thermos-gloves handling DryIce® to avoid freeze burns!

8. Place it in the sample holder so, that the laser can effectively scatter off to the opening of the detector.
9. Close the lid.

10. Enter the four-digit code to enable the laser.
11. Collect the spectrum of the sample.
12. When done, turn off the RU:
    i. Press “1” on the keypad.
    ii. Turn the key yo OFF. Remove and return the key.
    iii. Turn off the power switch

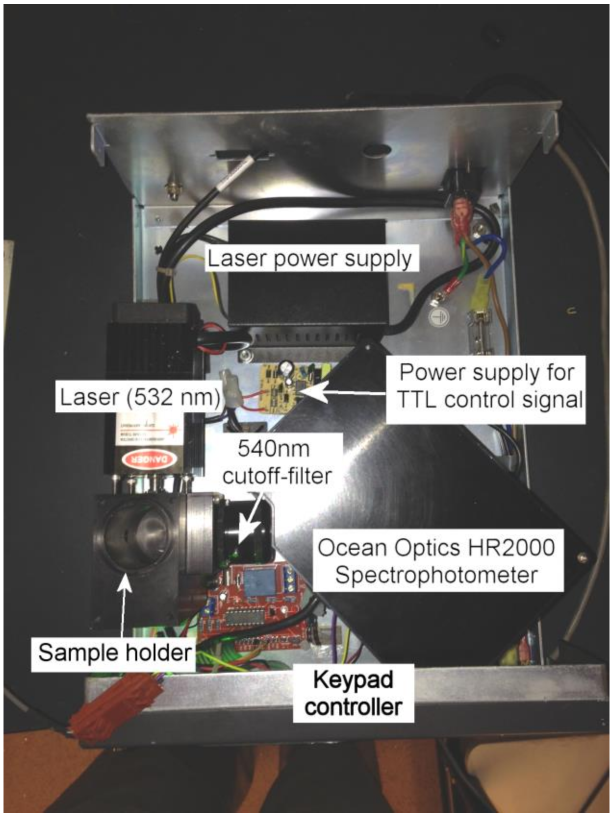


Figure 1.10 Inside view of RU.

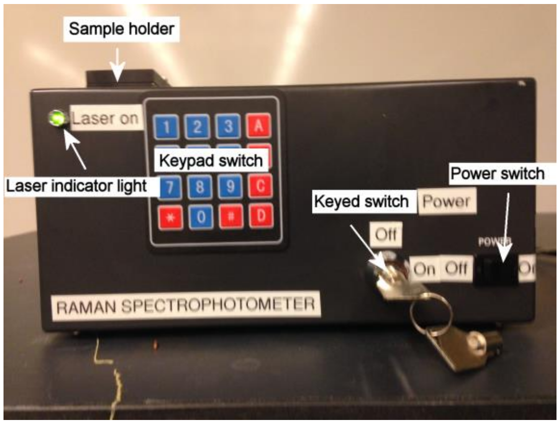


Figure 1.11 Front view of RU.

Computational modeling

1. Perform a modeling of the CO_2molecule using different configurations to obtain the IR and Raman spectra as well as the equilibrium bond length and the bonding energy:
    i. Optimization+Frequency calculation to obtain the IR and Raman spectra.
    ii. Scanning calculation to obtain the equilibrium distance for the C=O bond and the estimate of the bonding energy.
    Note: Try different methods and basis sets based on what you learnt in class. Keep in mind that the CO_2 molecule is relatively small. For the scanning set one of the bonds to 0.7 Å and collect 50 steps with 0.1 Å steps.
2. Export the data from the IR and Raman spectra, as well as from the scanning calculation into a text file.

3. Setup Gaussian to scan the potential between two CO_2 molecules (do the calculations between 0.7 and 10 A).

Calculations

There is no formal lab report due for this lab separately. For the final comprehensive lab report:

1. Obtain the wavenumbers for the IR-active transitions from the IR spectrum.
2. For the Raman spectrum process the background and the spectrum of the sample the most effective way:
    i. Try simply subtract the background from the spectrum of the sample.
    ii. If that is not sufficient, apply the smooth function before the subtraction to both spectra.
    iii. You can also try to fit a 7-11 order polynomial on the IR spectrum, and use it as a background that you subtract from the actual spectrum.
3. Save the predicted IR and Raman spectra from the computational calculation.
4. Save the r, E_{bond}, σ, and ε values from the modeling.
5. Save your Gaussian files.

Experiments 2 -4

Properties of Real Gases

Background

In the complex laws-driven Nature it was a refreshing discovery that the behavior of gases, regardless of their chemical identity can be described with a very simple mathematical relationship, the Ideal Gas Law, eqn. (2.1).

$$\begin{aligned} pV &= nRT \\ pV_m &= RT \end{aligned} \qquad \textbf{(2.1)}$$

The complications arose when gases were cooled down or compressed to high pressure, when they started to show their individual nature. Efforts to describe gases under these conditions resulted in semi-empirical models, such as the van der Waals equation; however, these yielded more questions than answers. A slew of physical constants were measured, such as the constants *a* and *b* in the van der Waals equation, without offering explanation to their values beyond some qualitative considerations. Part of this effort was Joule and Thomson's work, that ultimately laid the path for the industrial processes based on liquefied gases (such as using supercritical carbon dioxide to extract caffeine from coffee beans or to replace highly toxic chemicals in dry-cleaning processes). It wasn't until Statistical Mechanics evolved that quantitative, molecular-level explanation could be offered for the behavior of gases.

In these three labs you will study the behavior of carbon dioxide (and to some extent nitrogen and helium), both experimentally and theoretically, moreover you will compare the results from both approaches. The necessary theoretical concepts will be derived in class (and also please review the provided lecture notes), here only the statistical mechanical results are provided.

From the available variety of experimental approaches, you will use the *virial* version of equation of state and the Joule-Thomson coefficient to characterize the behavior of “real gases.” You will also determine the heat capacity ratio of CO_2 (γ), from which the heat capacity values (C_V and C_p) can be obtained.

The virial equation was developed in an effort to match experimental data with existing mathematical models, just like the van der Waals equation; however, as you will see, Statistical Mechanics can shed some light on the meaning of the empirical results.

The virial equation of state

There are two assumptions made when we use the Ideal Gas Law: the volume of the molecules is negligible, that is the molecules are “point-like,” and that there is no interaction among molecules. The validity of the Ideal Gas Law decreases as the pressure increases and/or the temperature decreases. The validity of the Ideal Gas Law usually is characterized by the *compressibility coefficient, Z*:

$$Z = \frac{pV}{nRT} \qquad Z = \frac{pV_m}{RT} \tag{2.2}$$

The compressibility coefficient is $Z = 1$ for ideal gases. The Ideal Gas Law is a limiting law; the limiting value of the compressibility coefficient is 1 as the pressure approaches zero and the temperature approaches infinity. At moderately low pressures the compressibility coefficient of gases usually is less than 1 (except the noble gases and the hydrogen gas) and at higher pressure it is higher than 1 (see Figure 2.1).

The compressibility coefficient is related to the virial equation of state as shown in (2.3).

$$Z = \left(\frac{pV_m}{RT}\right) = 1 + B_2\left(\frac{1}{V_m}\right) + B_3\left(\frac{1}{V_m}\right)^2 + B_4\left(\frac{1}{V_m}\right)^3 + \ldots \tag{2.3}$$

where B_2, B_3, B_4, ... are the second, third, fourth virial coefficients. The higher the pressure and lower the temperature, the more of the higher terms have to be considered. In this experiment the pressure of the gas will be between 1-35 bar, and in this range only the second virial coefficient has to be considered:

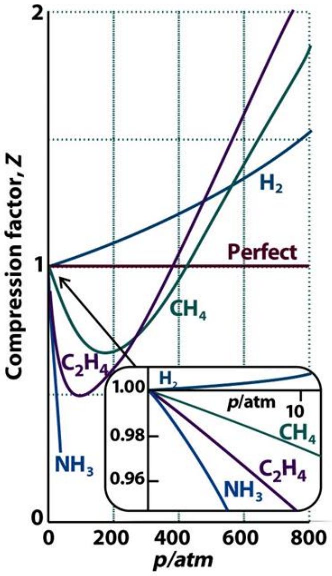


Figure 2.1 Compressibility factor of real gases as a function of pressure

$$Z = 1 + B_2\left(\frac{1}{V_m}\right)$$
$$Z = 1 + A_2 p \qquad \textbf{(2.4)}$$

We will be using the second form in this experiment as we will monitor the pressure of the gas. The relationship between A_2 and B_2 is:

$$B_2 = A_2 RT \qquad \textbf{(2.5)}$$

Another measure of the deviation from the ideal behavior is the *fugacity* coefficient:

$$f = \gamma p \qquad \textbf{(2.6)}$$

Where *f* is the *fugacity*, which is the experienced pressure of the gas, and γ is the fugacity coefficient, which is also dependent on the pressure:

$$\ln \gamma = \frac{B_2 p}{RT} \qquad \textbf{(2.7)}$$

You will also calculate the virial coefficient from two of the available semi-empirical methods shown in eqn. (2.8.)

$$B(T) = 137.6 - 87.7e^{325.7/T}$$

$$B\left(T\right) = -127 - 288\left(\frac{298}{T} - 1\right) - 118\left(\frac{298}{T} - 1\right)^2 \tag{2.8}$$

Both of these equations yield the value of *B*(*T*) in mL/mol (the temperature has to be in K)

The Joule-Thomson coefficient

When a real gas is driven through a porous disk (see Figure 2.2) and enters a lower pressure it will cool down. While the process is adiabatic, distinction has to be made from the classical adiabatic process. In both cases there is no heat exchange between the environment and the system; however, the process through the porous medium happens while the system is in steady state (equilibrium state). In both cases the real gas does expansion work on the environment at the expense of the energy of the system, therefore typically it cools down. The Joule-Thomson coefficient is defined as:

$$\mu_{JT} \equiv \left(\frac{\partial T}{\partial p}\right)_H \tag{2.9}$$

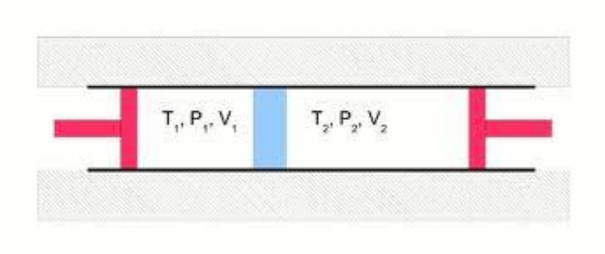


Figure 2.2 Joule-Thomson experiment

It can be shown, that such a process is *isoenthalpic*, that is the enthalpy of the system is constant.

Determination of the heat capacity ratio of gases (γ)-Ruchhardt method

The heat capacity ratio (γ) is typically measured with an adiabatic process. The simplest method is performing adabatic compression or expansion on a gas and measuring *p*, *T*, and *V*. The experimental setup for this approach is a Plexi® glass cylinder with a piston in which these parameters are measured as the gas is quickly compressed or expanded. In the other approach (Ruchhardt method) a sample of the gas is trapped in a reservoir with an exit that has a piston that can move quasi-frictionlessly. When the pressure inside the reservoir and outside is the same, the piston is at rest, and when it is removed from its equilibrium position and let go, it returns to its original position with a damped oscillation. From the period of the oscillation the heat capacity ratio of the gas can be obtained.

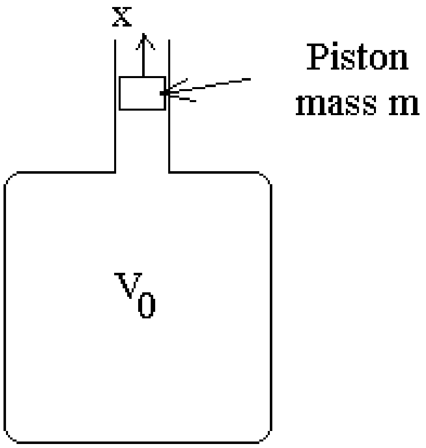


Figure 2.3 Schematics of the Ruchhardt experiment

Determination of the heat capacity ratio of gases (γ)-Adiabatic method

Considering a gas that changes its state from T_1, p_1, V_1 to T_2, p_2, V_2 under adiabatic conditions (dq = 0):

$$\begin{aligned} p_1V_1^{\gamma} &= p_2V_2^{\gamma} \\ T_1V_1^{\gamma-1} &= T_2V_2^{\gamma-1} \end{aligned} \qquad \textbf{(2.10)}$$

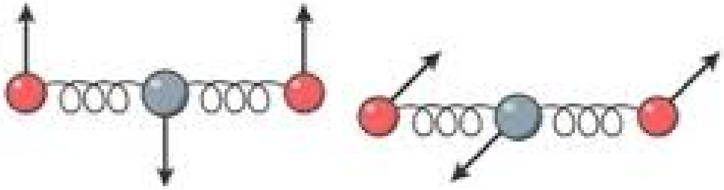
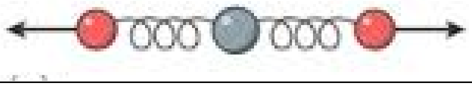
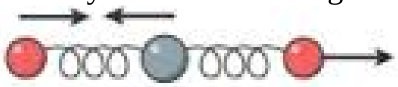
(Please refer to the appendix and the provided lecture notes for the derivation of these relationships). The ratio of specific heats depends on the nature of the gases. The adiabatic process is accomplished by using an Adiabatic Gas Law Apparatus (AGA). While this equipment is not insulated, the compression of expansion is performed so quickly, that the system cannot change heat with its environment in any appreciable extent.

Theoretical calculations

The constant volume heat capacity (C_V) of a molecule has contribution from its translational, rotational, and vibrational modes:

$$C_V = C_{V,trans} + C_{V,rot} + C_{V,vib} \qquad \textbf{(2.11)}$$

Specifically, the CO_2 molecule has 3 translational and 2 rotational freedoms, each contributing 1/2RT to C_V. Since linear molecules have (3N-5) vibrational degrees of freedom, the CO_2 molecule has 4 vibrational modes contributing to C_V. Indeed, there is a symmetric, an asymmetric, and two bending modes (the bending modes are doubly degenerate and they show up at the same frequency):

Fundamental frequency (cm^{-1})	Mode
667.3	Bending (doubly degenerate) 
1340	Symmetric stretching 
2349.3	Asymmetric stretching 

The total vibrational contribution can be calculated from:

$$C_{V,vib} = R\sum_{i=1}^{4}\left[\left(\frac{\Theta_i}{T}\right)^2 \frac{e^{\Theta_i/T}}{\left(e^{\Theta_i/T}-1\right)^2}\right], \qquad \Theta_i = \frac{hc\nu_i}{k_B} \tag{2.12}$$

where k_B = Boltzmann constant (J/K)
ν_i = ith fundamental frequency (m^{-1})
Θ_i = vibrational temperature (of the ith vibrational mode)
c = speed of light (m/s)
T = temperature (K)

(Remember, the bending mode is to be counted twice).

Therefore, the overall C_V value can be estimated from:

$$C_{V,trans} = \frac{3}{2}R$$

$$C_{V,rot} = \frac{2}{2}R$$

$$C_{V,vib} = R\sum_{i=1}^{4}\left[\left(\frac{\Theta_i}{T}\right)^2 \frac{e^{\Theta_i/T}}{\left(e^{\Theta_i/T}-1\right)^2}\right], \qquad \Theta_i = \frac{hc\nu_i}{k_B}$$

$$\begin{aligned} C_V &= C_{V,trans} + C_{V,rot} + C_{V,vib} = \\ &= \frac{3}{2}R + \frac{2}{2}R + R\sum_{i=1}^{4}\left[\left(\frac{\Theta_i}{T}\right)^2 \frac{e^{\Theta_i/T}}{\left(e^{\Theta_i/T}-1\right)^2}\right] = \\ &= R\left[\frac{5}{2} + \sum_{i=1}^{4}\left[\left(\frac{\Theta_i}{T}\right)^2 \frac{e^{\Theta_i/T}}{\left(e^{\Theta_i/T}-1\right)^2}\right]\right] \end{aligned} \qquad \textbf{(2.13)}$$

From Statistical Mechanics we know, that an intermolecular potential, $U(r)$, and the corresponding virial coefficients are related by eqn. (2.14).[2, 3]

$$B(T) = -2\pi N_A\int_0^{\infty}\left[\left(e^{-U(r)/k_BT}-1\right)r^2\right]dr \qquad \textbf{(2.14)}$$

where r = intermolecular separation (m)
$U(r)$ = intermolecular potential (J)
T = temperature (K)
N_A = Avogadro number (mol^{-1})

Commonly used potential function is the Lennard-Jones potential function:

$$U(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \qquad \textbf{(2.15)}$$

where σ = separation where the repulsion turns to attraction between molecules
ε= depth of the potential well (J)

Figure 2.4 shows a typical Lennard-Jones potential and Figure 2.5 shows the calculated temperature dependence of the second virial coefficient of CO_2.

The second virial coefficient is related to the Joule-Thomson coefficient (see eqn. **(2.16**)), which is not surprising since they both describe the behavior of real gases, and they both vanish for ideal gases (for the expression used in **(2.16)** for the Joule-Thomson coefficient please refer to the Appendix).

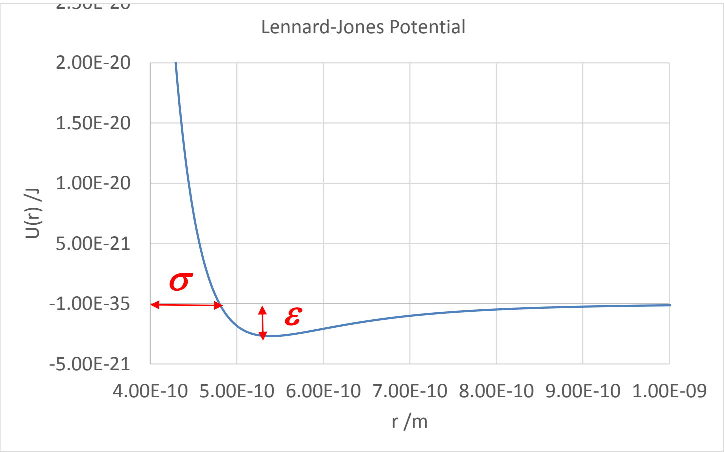


Figure 2.4 The Lennard-Jones potential

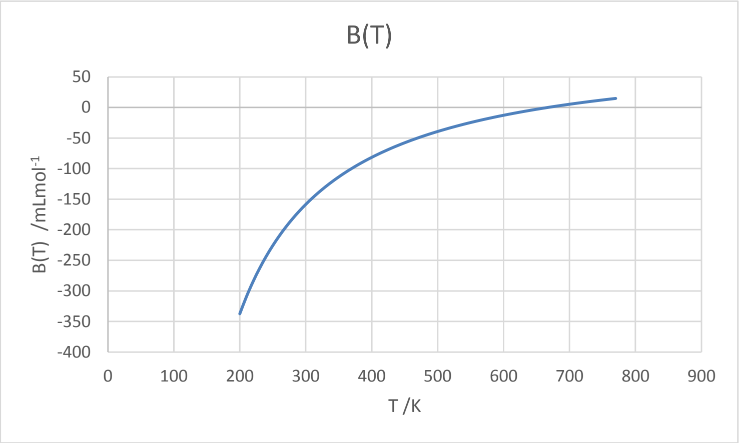


Figure 2.5 *B*(*T*) function

$$pV = nRT + nB(T)p \rightarrow V = \frac{nRT}{p} + nB$$

$$\left(\frac{\partial V}{\partial T}\right)_p = \frac{\partial}{\partial T}\left(\frac{nRT}{p} + nB\right) = \frac{nR}{p} + n\left(\frac{\partial B}{\partial T}\right)$$

$$\mu_{JT} = -\frac{1}{c_p}\left[-T\left(\frac{\partial V}{\partial T}\right)_p + V\right]^{\#} = -\frac{1}{c_p}\left[-T\left(\frac{nR}{p} + n\left(\frac{\partial B}{\partial T}\right)\right) + \frac{nRT}{p} + nB\right] =$$

$$= -\frac{1}{c_p}\left[-\frac{nRT}{p} - nT\left(\frac{\partial B}{\partial T}\right) + \frac{nRT}{p} + nB\right] \tag{2.16}$$

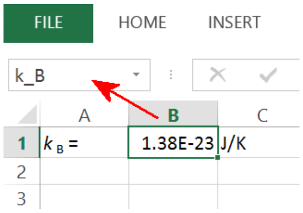
$$= \frac{1}{c_p}\left[nT\left(\frac{\partial B}{\partial T}\right) - nB\right]$$

$$\mu_{JT} = \frac{1}{c_{p,m}}\left[T\left(\frac{\partial B}{\partial T}\right) - B\right]$$

[#]see the Appendix for the derivation of this representation of the JT Coefficient

Pre-Lab work

1. Calculate C_v for CO_2 using eqn. (2.13).
   a. Setup an Excel spreadsheet with the calculations for C_V. Define each constant explicitly (instead of using cell references) for ease of use. E.g. if you put the value of k_B in cell B1, click on the cell on the top left showing the cell reference (B1), and type in a name that is unique and is not used by Excel (e.g. "B10" is not good, as it refers to a cell), in the example below **k_B**.
   b.



	A	B	C
1	k_B =	1.38E-23	J/K
2			
3			

   The constant then be used in all formulae in the workbook.
   c. Show clearly the contribution from translation, rotation, from each mode of vibration, the overall contribution from vibrational modes, as well as the overall estimated value of C_V.
   d. Calculate the value of C_V as a function of the temperature between 100-1000 K at 10 K increments on the same spreadsheet. Some thoughts have to be put into arranging your calculations. A possible header structure for this spreadsheet is shown in the Appendix. While some of the quantities won't change with temperature, it is easier to arrange your data in the fashion shown in the Appendix.
   e. Make a graph of the temperature dependence of C_V and $C_{V,vib}$ (you can arrange those on one graph, just use different vertical axes).
2. Estimate the value of $B(T)$ at the temperature of the experiment (296 K) using eqn. (2.8) semi-empirical equations.
3. Setup an Excel spreadsheet to estimate $B(T)$ with the Statistical Mechanical approach as follows.
   a. Create a list of r values in column A from row 9 down, starting $1x10^{-11}$m all the way to $5x10^{-9}$ m.
   b. In column B from row 9 enter the formula that calculates the Lennard-Jones potential (2.15) at the r value in column A. Use $\sigma = 4.80x10^{-10}$ m and ε/k_B= 195 K (calculate ε).
   c. In column C from row 9 down enter the formula for the function in the bracket in (2.14).
   d. In column C row 7 enter 200, this will be the lowest temperature to evaluate $B(T)$ at.

e. In row 8 enter the formula for (2.14). The sum of the elements in column C approximates the integral part of $B(T)$, preferably in mL. Make sure to perform proper unit conversions!

f. In row 7 horizontally enter temperature values with 10 K increments (the first is 200 K already in place).

g. Fill the cells below row 8. Use absolute references ($) smartly, so you can simply copy column C to the rest of the columns.

h. Copy rows 7 and 8 into a new spreadsheet into columns A and B using **Paste Special/Paste Values and Transposed**. Make a graph of $B(T)$ as a function of T.

i. In column C create the dB/dT values as follows. In row 10 enter the formula to calculate the change of B as a function of temperature between rows 9 and 11, something like: =(B11-B9)/(A11-A9). Copy this formula to the rest of the column C (except the last cell).

j. In column D copy the values of C_p. You already calculated the C_p values as a function of temperature at every 10 K between 100K and 1000K. You can copy those values over or copy over the formula referencing the C_p values. For example if the worksheet that has the C_V-C_p calculations is called “Cv calculations”, and on that sheet the C_p values are in column T from row 12 down, for the first value of C_p enter: **=’Cv calculations’!T12** (the single quotation is necessary around the name of the worksheet if the name has space in it). Copy this cell below in subsequent rows.

k. In column E calculate the Joule-Thomson coefficient based on (2.16). Make a graph of the temperature dependence of the JK coefficient.

Determination of the second virial coefficient of CO_2

Background of the experiment

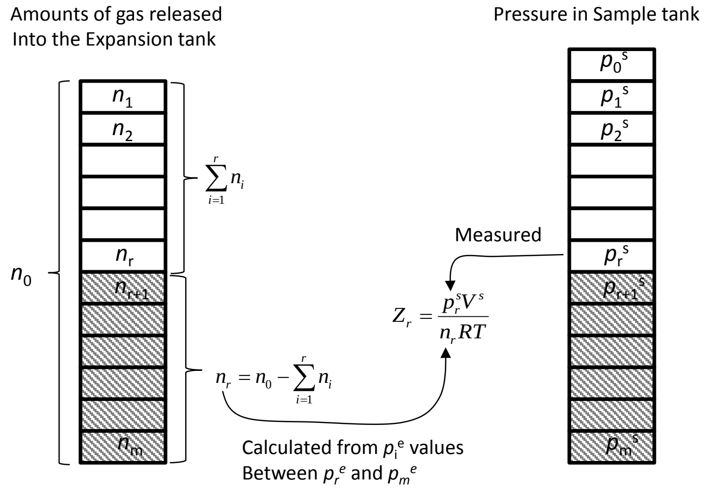


Figure 2.6 Strategy to measure the second virial coefficient

The strategy that we will follow is that we will determine the compressibility factor as a function of pressure (see Figure 2.6). The schematics of the experimental setup are shown in Figure 2.7. We load a “Sample tank” with CO_2 at about 35 bar which is high enough for the gas to show deviation from the ideal behavior. Then we release a small sample of the gas into the “Expansion tank” to a low enough pressure so that using the Ideal Gas Law the amount of released sample can be calculated. After evacuating the “Expansion tank,” we release another small sample from the “Sample tank.” We continue the process until the pressure in the two tanks is the same. The sum of the released samples makes up for the total initial number of moles.

The amount of the first sample released from the Sample tank:

$$n_1 = \frac{p_1^e V^e}{RT} \qquad \textbf{(2.17)}$$

where p_1^e is the pressure in the “Expansion tank” after releasing the first sample, V^e is the volume of the “Expansion tank,” and the pressure in the “Sample tank” drops to p_1^s. The Expansion tank then evacuated, and a second sample is released from the “Sample tank”:

$$n_2 = \frac{p_2^e V^e}{RT} \qquad \textbf{(2.18)}$$

The process is continued until the pressure in the two tanks is the same after the *m*th sample: $p_m^e = p_m^s$. The total amount of gas is the sum of the amounts of the samples released from the "Sample tank":

$$
\begin{aligned}
n_0 &= n_m^s + n_1^e + n_2^e + \ldots + n_m^e = n_m^s + \sum_{i=1}^{m} n_i \\
n_0 &= \frac{1}{RT}\left(p_m^e V^s + p_1^e V^e + p_2^e V^e + \ldots p_m^e V^e \right) \\
n_0 &= \frac{1}{RT}\left(p_m^e V^s + V^e \sum_{i=1}^{m} p_i^e \right)
\end{aligned}
\qquad \textbf{(2.19)}
$$

where n_i is the number of moles in the *i* th release. We need to devise a method to calculate the amount of gas remaining in the sample tank after each release. Consider the *r* th release, after which the remaining gas in the Sample tank is:

$$
n_r = n_0 - \sum_{i=1}^{r} n_i \qquad \textbf{(2.20)}
$$

The combination of Equations **(2.19)** and **(2.20)** yields:

$$
\begin{aligned}
n_r &= \frac{1}{RT}\left(p_m^e V^s + V^e \sum_{i=1}^{m} p_i^e - V^e \sum_{i=1}^{r} p_i^e \right) \\
n_r &= \frac{1}{RT}\left(p_m^e V^s + V^e \sum_{i=r+1}^{m} p_i^e \right)
\end{aligned}
\qquad \textbf{(2.21)}
$$

Substituting Equation (2.21) into (2.2) results in:

$$
\begin{aligned}
Z_r &= \frac{p_r^s V^s}{n_r RT} \\
Z_r &= \frac{p_r^s V^s}{p_m^e V^s + V^e \sum_{i=r+1}^{m} p_i^e} \\
Z_r &= \frac{p_r^s}{p_m^e + \frac{V^e}{V^s} \sum_{i=r+1}^{m} p_i^e}
\end{aligned}
\qquad \textbf{(2.22)}
$$

where Z_r is the value of the compressibility factor after the r th release at the pressure of p_r^s. Equation (2.22) is the basis of the experiment. The pressure has to be measured at every release in both, the “Expansion tank.” and the “Sample tank.” The summation has to be performed after the experiment was completed: for the r the release is the sum of all “Expansion tank” pressure readings from reading r+1 to m.

Notice, that the actual volume of either tank is not relevant, only the ratio of the two volumes, which will also be determined in the experiment. When the experiment is done after the mth reading, once again close the “Sample tank” and evacuate the “Expansion tank” then allow the gas to expand into the “Expansion tank,” when the pressure will be p_f.

$$\begin{aligned}
p_m^s V^s &= p_f\left(V^e + V^s\right) = p_f V^e + p_f V^s \\
p_m^s V^s - p_f V^s &= p_f V^e \\
\left(p_m^s - p_f\right) V^s &= p_f V^e \\
\frac{V^e}{V^s} &= \frac{p_m^s - p_f}{p_f}
\end{aligned} \qquad \textbf{(2.23)}$$

Therefore we can use the Ideal Gas Law to calculate V^e/V^s.

Experimental procedure

1. Close manual valve and start vacuum pump manually (red button) 2. Connect the CO_2 tank and set the pressure to 400PSI on the regulator	

3. Execute the `VC_setup.m` script	`VC_setup.m`
	`%sNI = data acquisition object`

4. Evacuate the whole setup	`PumpDown('All',100,10,sNI);`
	`%100 = fluctuation allowed in Pa`
	`%10 = time in s to wait after pumping down`

5. Load Sample tank with gas to 450 PSI pressure	`LoadGas(450,'PSI',sNI);`

6. Read initial pressure and temperature values
7. Save initial readings
8. Set a flag (`expflag`) to control the experiment (when to stop)
	`psave = [];`
	`ai_data = ReadData(10, 100, sNI);`
	`pe = PresCal30(ai_data(3),'Pa');`
	`ps = PresCal500(ai_data(2),'Pa');`
	`temp = TempTherm(ai_data(1),'K');`
	`psave = [psave; temp ps pe;];`
	`expflag = 0;`

9. Pump down the Expansion tank	`PumpDown('Sample',100,10,sNI);`
10. Load Expansion tank with `psample = 90000` (Pa) gas 11. Check if the pressure is enough (`loaded` = 1) 12. If `loaded` = 0, take a final reading and stop (`expflag = 0)`.	`loaded = LoadSample(psample,'Pa',sNI);`

13. If there is enough gas, allow the system to equilibrate	`equilflag = 0;`
	`while equilflag == 0`
	`    pbefore = PresCal30(ReadData(2, 100, sNI),'Pa');`
	`    pause(60);`
	`    pafter = PresCal30(ReadData(2, 100, sNI),'Pa');`
	`    if abs(pbefore-pafter) <= tol`
	`       equilflag = 1;`
	`    end`
	`end`

14. Read pressure and temperature values in the Sample and in the Expansion tank.	`ai_data = ReadData(10, 100, sNI);`
	`pe = PresCal30(ai_data(3),'Pa');`
	`ps = PresCal500(ai_data(2),'Pa');`
	`temp = TempTherm(ai_data(1),'K');`

15. Repeat steps 9-14 until there is enough gas (set `expflag` = 1).	
16. Close the main valve on the gas cylinder 17. Release the pressure with the butterfly valve on the regulator 18. Disconnect gas hose	

Do NOT disconnect the tubing under pressure! The free end of the tubing may break lose due to the violent gas flow resulting in personal injury! De-pressurize the system by opening all valves BEFORE disconnecting the tubing!

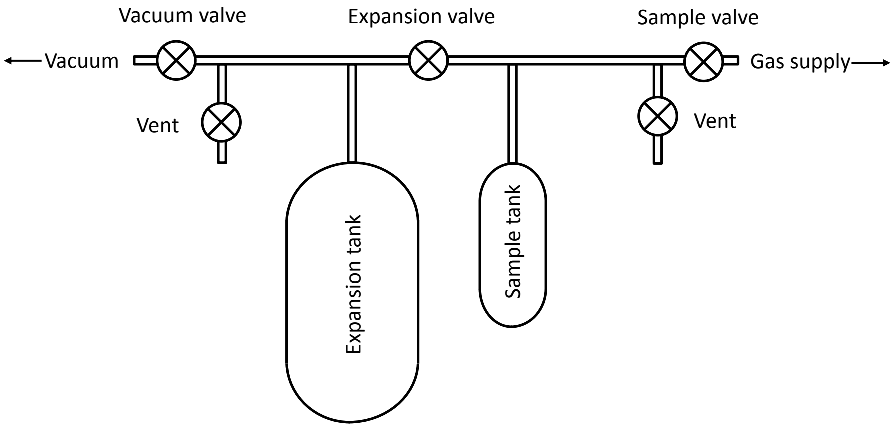


Figure 2.7 Schematics of the experimental setup to measure the second virial coefficient

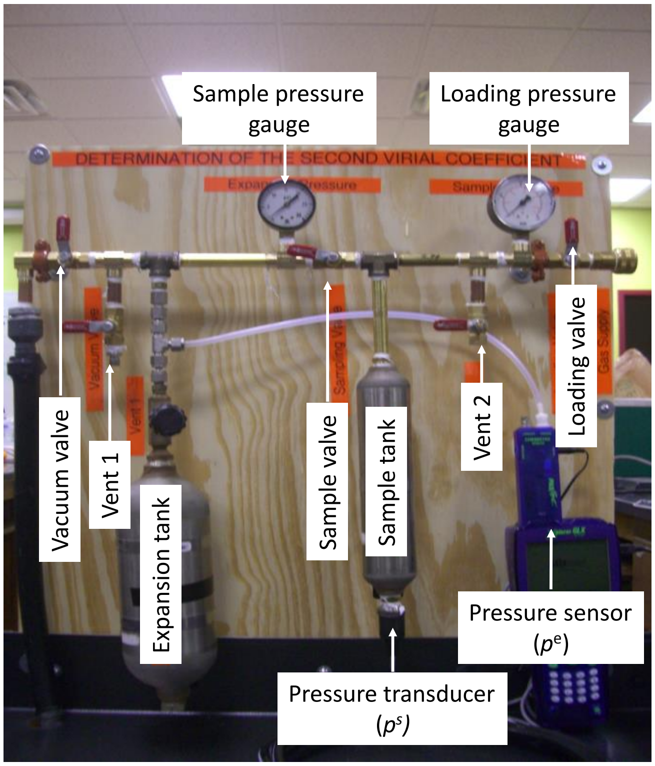


Figure 2.8 The experimental setup to measure the second virial coefficient

Determination of the Joule-Thomson coefficient of CO_2

Background of the experiment

The experiment will be performed with a setup much like the original experiment. The schematics of experimental setup is shown in **Error! Reference source not ound.** and the actual setup is shown in Figure 2.10. The gas is allowed to pass through a porous disk from the high pressure side and leave on the low pressure side (which is open to the atmosphere. The temperature of the gas is measured with fast-response temperature probes on both sides. The pressure on the high pressure side is controlled by an electronic pressure regulator which sets the pressure between 0-10 bar based on a signal between 0-10V. The signal is provided by one of the GLXes in one of two fashion: either the signal is ramped up over the period of 1000s from 0 to 5V which provides pressure between 0 and 7 bar (approximately), or the signal is set to a particular value for a period of time which sets the pressure to a constant value, and after the system settles to an equilibrium, the pressure and temperature difference is measured. Then, a somewhat higher signal (and higher pressure) is set for the next reading. If the signal is ramped up (and therefore the pressure is gradually increased on the high pressure side), the assumption is that the ramping up is slow enough so that the system is at equilibrium all the time, therefore continuous pressure and temperature readings can be made.

Experimental procedure

1. The schematics of experimental setup is shown in **Error! Reference source not ound.** and the actual setup is shown in Figure 2.10.
2. Connect the gas supply to the setup and turn on the heating on the gas regulator.

   Carbon dioxide cools down considerably upon expansion and the regulator has to be heated to prevent freezing!

3. Turn on the fan of the heat exchanger to High.
4. Open the gas supply to the setup and set the supply pressure to about 10 bar.
5. Connect GLX1 to the computer. This device will collect data from the temperature and pressure sensors.
6. Configure Capstone as follows:
   a. Create calculated datasets for the temperature and pressure change across the frit.

   $$\begin{aligned} dT &= T_f - T_i \\ dp &= p_f - p_i \end{aligned} \qquad \textbf{(2.24)}$$

   b. Create a page with digital displays of the temperature, pressure readings and with the two calculated datasets (*dT* and *dp*).

c. Add a graph showing *dT* on the vertical, *dp* on the horizontal axis. Enable linear fitting for this graph. The slope of this graph will be the Joule-Thomson coefficient.
d. Add a graph showing Initial Temperature as a function of the time.

7. On the pressure regulator GLX (GLX2) under **Output** select **Power Amplifier**, and once the automatic calibration is completed, select the Waveform as **DC voltage**, and set the voltage to 5V. Press **F1** which will open the pressure regulator to the system at maximum pressure. Open the flush valve a few times about a second at a time. This will flush the system with CO_2.
8. Change the **DC voltage** to **Ramp**, and set the following parameters:

Amplitude	5 V
Offset	5 V
Period Units	s
Duration	1000 s
Repeat Mode	One Shot
Step (F2 button)	Custom 0.01

This configuration once turned on will ramp up the pressure to about 7atm over the period of 1000s. The process is slow enough so that the system will be likely at

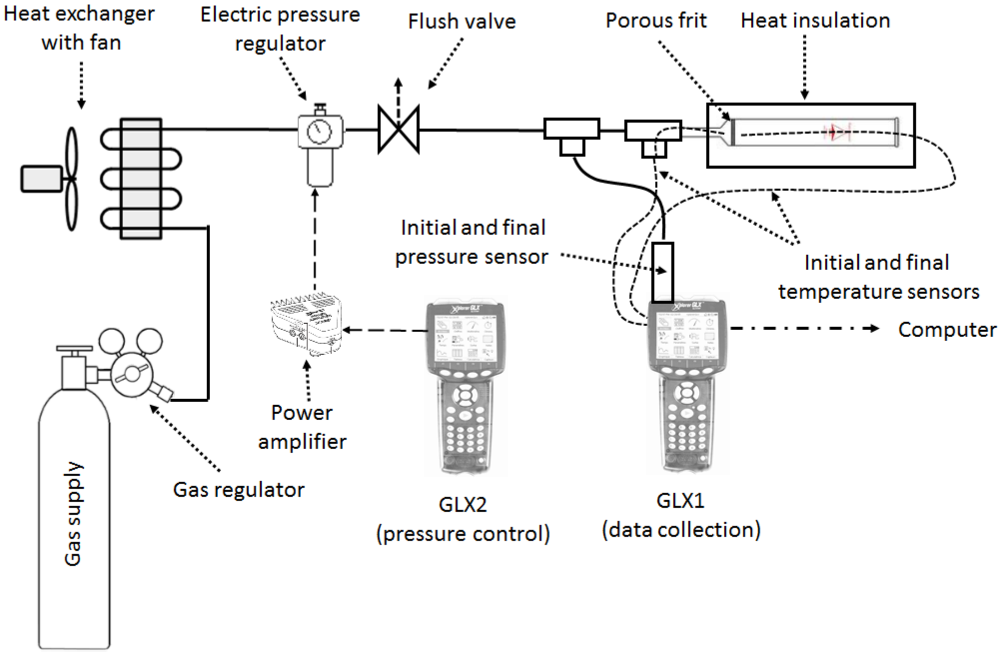


Figure 2.9 Schematics of the experimental setup to measure the Joule-Thomson coefficient

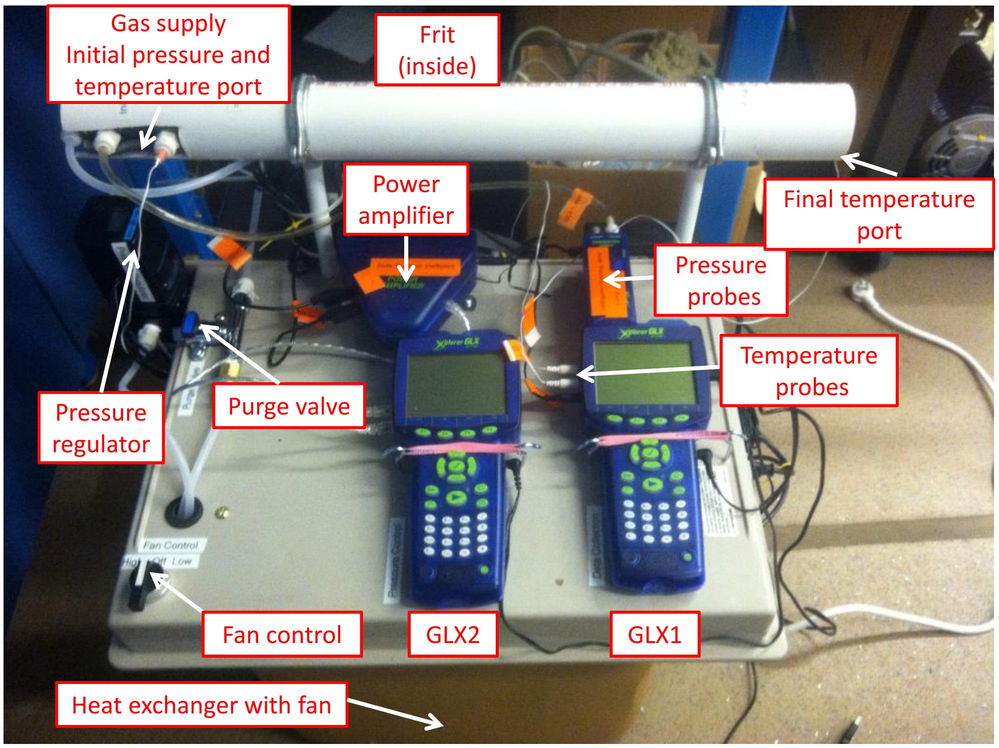


Figure 2.10 The experimental setup to measure the Joule-Thomson coefficient

equilibrium at any point which can be monitored on the Initial Temperature vs. time graph.

9. Press **F1** on GLX2 which will start the ramping of the pressure. Monitor the pressure on the pressure regulator or put Capstone in data monitoring mode and monitor the initial pressure on the computer.
   When the pressure reaches 10 PSI on the regulator (or about 70, kPa), start the data collection.
10. While the data is being collected, monitor the Initial Temperature. It has to be constant within a few tenth of a degree. As the pressure goes up, the gas flow will increase considerably which cools CO_2 down at the entry point of the heat exchanger, and we have to be sure that the gas assumes room temperature (which is the initial temperature).
11. Once the pressure reached about 6.5 bar, stop the data collection and turn off the gas flow by pressing **F1** on GLX2.
12. If the collected data looks acceptable and the Initial Temperature was indeed constant, you are done. In case the heat exchanger couldn't keep the Initial Temperature constant, you need to run the experiment again as follows:
    a. Monitor *dT* instead of the Initial Temperature (change the graph)
    b. On GLX2, change the **Waveform** to **DC Voltage**. Set the voltage to 1V and press **F1** to turn on the gas flow.
    c. Change the data collection mode to **Keep Mode** in Capstone and start the collection (at this point the computer monitors the sensors, but no data is kept).
    d. Monitor the *dT* vs. time graph, an when the *dT* value becomes constant (the system is in equilibrium), take the first reading.
    e. Increase the output voltage on GLX2 to 1.25V.
    f. Wait until *dT* becomes steady and take your next reading.
    g. Continue to increase the output voltage with 0.25 V increments.
    h. When the pressure approaches 700 kPa, stop the experiment.

The pressure probe has the limit of 7 bar. Exposing the probes to higher pressure will damage the probes!

13. Perform the experiment with He, Ar, and N_2 as well.

Determination of the ratio of C_p/C_v of CO_2 –Rucchardt method

Background of the experiment

The setup consists of a 5-gal jug with an apprx. 5ft. vertical Plexi® glass tube attached. There is a short (about 12") Plexi® glass tube that can move frictionlessly inside the long tube. There is a port with a valve for gas supply to the jug and a port for a pressure probe to monitor the pressure. When the short tube is dropped in the long tube, it will bounce up and down with a damped oscillation before coming to a stop.
The pressure inside the reservoir is:

$$p = p_0 + \frac{mg}{A} \qquad \textbf{(2.25)}$$

where p_0 is the atmospheric pressure, *A* is the area of the cross section, and *m* is the mass of the ball. When the ball is removed from its equilibrium, the force returning the ball to its equilibrium is given by:

$$A\,dp = m\frac{d^2x}{dt^2} \qquad \textbf{(2.26)}$$

where, *dp* is the pressure change due to the displacement of the ball. Since the oscillation of the ball has rather high frequency, the compression-expansion of the gas in the reservoir can be considered for practical purposes adiabatic:

$$pV^{\gamma} = const. \qquad \textbf{(2.27)}$$

Taking the derivative of both sides yields:

$$\begin{aligned} &V^{\gamma}dp + p\gamma V^{(\gamma-1)}dV = 0 \\ &V^{\gamma}dp = -p\gamma V^{(\gamma-1)}dV \\ &dp = -\frac{p\gamma V^{(\gamma-1)}}{V^{\gamma}}dV = -\frac{p\gamma}{V}dV \end{aligned} \qquad \textbf{(2.28)}$$

The displacement of the ball (*x*) will change the volume of the gas with *dV*:

$$dV = Ax \qquad \textbf{(2.29)}$$

Combining eqns. (2.26), (2.27)(2.29), and (2.30) yields:

$$\begin{aligned} &-\frac{p\gamma}{V}A^2x = m\frac{d^2x}{dt^2} \\ &-kx = m\frac{d^2x}{dt^2} \end{aligned} \qquad \textbf{(2.30)}$$

This equation is Hook’s Law which characterizes a simple harmonic motion, where *k* is the Hook constant. The Hook constant is related to the angular frequency (ω) of the oscillation. The angular frequency is also related to the period (*T*) of the oscillation:

$$\begin{aligned} &\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{p\gamma A^2}{mV}} \\ &T = \frac{2\pi}{\omega} = 2\pi\sqrt{\frac{mV}{p\gamma A^2}} \end{aligned} \qquad \textbf{(2.31)}$$

From (2.31) one can obtain:

$$\gamma = 4\pi^2 \frac{mV}{pT^2 A^2} \qquad \textbf{(2.32)}$$

According (2.32) by measuring the period of the oscillation, one can calculate the ratio of the heat capacities.

Experimental Procedure

1. Assemble the experimental setup. Connect the pressure/temperature probe to the GLX and the GLX to the computer. Make sure that the pressure is measured in Pascals. Also, put the computer to **Continuous Monitoring** to measure the atmospheric pressure.
2. Configure the experiment for "Delayed start" (when pressure exceeds the atmospheric pressure by about 5,000 Pa) and "Delayed stop" (collect data for 5s). The data collection rate has to be at least 25 Hz.
3. Connect the pressure probe and the gas supply.
4. Open the valve to the gas supply and allow the gas to flow through the system and flush it with the gas (for about 1 min).
5. Close the valve to the gas supply.
6. Start data collection. At this point the computer is waiting for the pressure to pass the trigger limit.
7. Carefully drop the short tube into the long, making sure that the string wouldn't get tangled. The data collection will get triggered and data will be collected for 5s.
8. Once the data collection stopped, pull back the small piston by the string or slowly apply pressure by opening the gas supply and catch the piston as it rises.
9. Slide masses of 25g, 50g, 75g, and 100g inside the piston and repeat the experiment.
10. Fit a "Damped sine oscillation" function to each run in Capstone and record the value of ω.

Determination of the ratio of C_p/C_v of CO_2 –Adiabatic method

Background of the experiment

Recall, that for adiabatic processes the following relationsips hold:

$$\begin{aligned} pV^{\gamma} &= const. \\ p^{1-\gamma}T^{\gamma} &= const. \\ TV^{\gamma-1} &= const. \end{aligned} \qquad \textbf{(2.33)}$$

In this experiment CO_2 will be driven through an adiabatic process during which the V, p, and T data will be collected.
The Adiabatic Gas Law Apparatus (AGLA) setup consists of a Plexi® glass cylinder equipped with a piston which can be moved with a lever very quickly allowing quick change of the volume of the trapped gas inside the cylinder (see Figure 2.11). The cylinder is also equipped with a fast response temperature probe, pressure sensor and a sliding resistor which monitors the position of the piston very accurately (which position in turn is converted to volume with a calibrating equation). The photo and the schematics of the setup are shown in Figure 2.11and Figure 2.12. The piston is operated by pneumatic valves utilizing the same gas as the gas being studied. The valves are controlled by MATLAB via the MODuck interface.

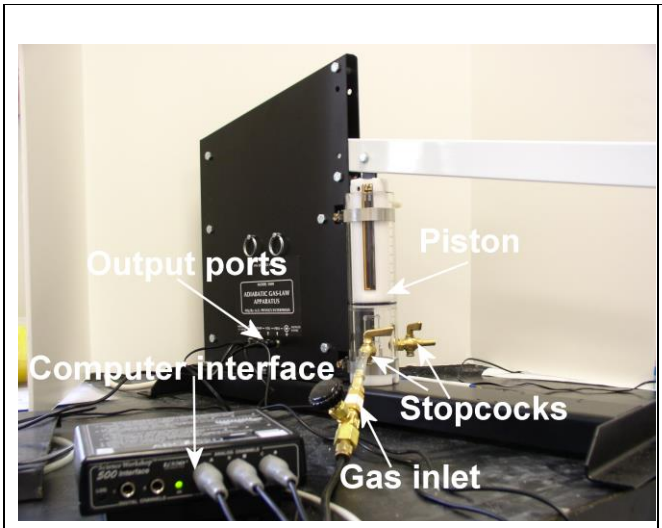


Figure 2.11 Adiabatic Gas Law Apparatus

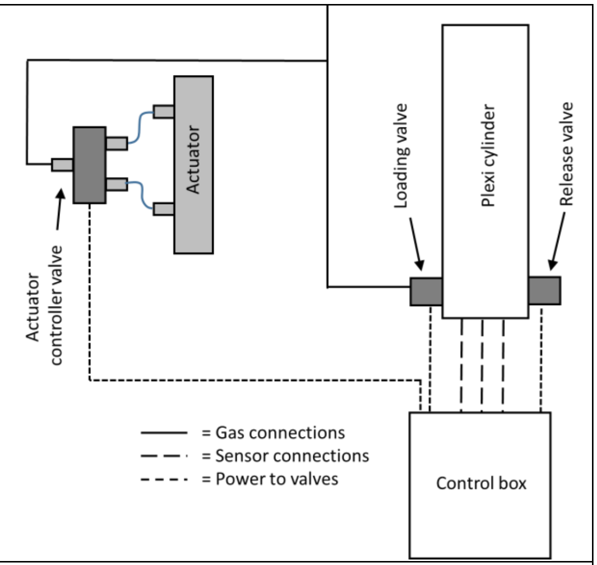


Figure 2.12 The schematics of AGLA

Experimental Procedure

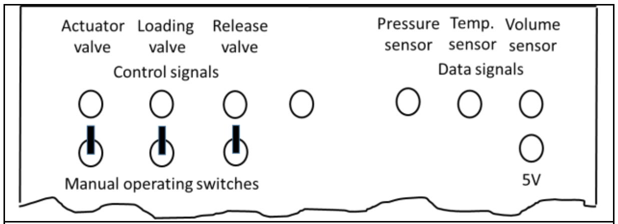


Figure 2.13 Connections to the controller box

1. Connect the gas source to the setup. Set the regulator on the gas source to 60PSI and open the butterfly valve on the regulator.
   <u>Note:</u> Using higher pressure may damage the equipment!
2. The connections to the MODuck interface is shown in Figure 2.13.
3. The following code configures the MODuck interface for control and data collection:

```
devs = daq.getDevices;
dev = devs.ID;
sNI = daq.createSession('ni');
```

```
sNI.addAnalogInputChannel(dev,'ai0','Voltage'); % Data channel for the volume sensor
aiNI1=sNI.Channels(1);
aiNI1.TerminalConfig='SingleEnded';
aiNI1.Range=[-5 5];
sNI.addAnalogInputChannel(dev,'ai1','Voltage'); % Data channel for the temperature
sensor
aiNI2=sNI.Channels(2);
aiNI2.TerminalConfig='SingleEnded';
aiNI2.Range=[-1 1];
sNI.addAnalogInputChannel(dev,'ai6','Voltage'); % Data channel for the pressure sensor
aiNI2=sNI.Channels(2);
aiNI2.TerminalConfig='SingleEnded';
aiNI2.Range=[-1 1];
sNI.addDigitalChannel(dev,'port0/line0','OutputOnly'); %Control signal for the
actuator valve
sNI.addDigitalChannel(dev,'port0/line1','OutputOnly'); %Control signal for the loading
valve
sNI.addDigitalChannel(dev,'port0/line2','OutputOnly'); %Control signal for the release
valve
sNI.addDigitalChannel(dev,'port0/line3','OutputOnly'); %Not used
sNI.outputSingleScan([1 1 1 1]);
```

Save this code in a separate MATLAB file (e.g. AC_setup.m)

4. Make your control script for the experiment. The first line should be invoking the setup script that you just saved (e.g. AC_setup.m)
5. In the string `sNI.outputSingleScan([1 1 1 1])` the elements of the vectors control the valves

[**1** 1 1 1]	Actuator valve	0 = Piston down 1 = Piston up
[1 **1** 1 1]	Loading valve	0 = Valve open 1 = Valve closed
[1 1 **1** 1]	Release valve	0 = Valve open 1 = Valve close

Note: the fourth element is in use.

6. The logic of the experiment is shown in the table below with possible code snippets.

1. Flushing the setup with the gas	• Open release valve
	• Move piston down
		`sNI.outputSingleScan([0 1 0 1]);`
		`pause(1);`

	• Close release valve
	• Open loading valve
	• Move piston up
		`sNI.outputSingleScan([1 0 1 1]);`
		`pause(5);`

	• Close loading valve
	• Open release valve
	• Move piston down
		`sNI.outputSingleScan([1 1 0 1]);`
		`pause(1);`

	• Close all valves	`sNI.outputSingleScan([1 1 1 1]);`
2. Equilibrate gas to room	• Equilibrate cylinder	`pause(20);`

	• Open release valve
	• Close release valve
		`sNI.outputSingleScan([1 1 0 1]);`
		`pause(1)`
		`sNI.outputSingleScan([1 1 1 1]);`

	• Equilibrate cylinder	`pause(20);`

	- Open release valve
	- Close release valve
		`sNI.outputSingleScan([1 1 0 1]);`
		`pause(1)`
		`sNI.outputSingleScan([1 1 1 1]);`

3. Perform experiment	- Initiate piston movement down and
	- Collect data
	- Wait 10 s
	- Move up piston
		`sNI.outputSingleScan([0 1 1 1]);`
		`  for i=1:n`
		`    raw_data = [raw_data;  sNI.inputSingleScan];`
		`  end`
		`pause(10);`
		`sNI.outputSingleScan([1 1 1 1]);`

Note:

- Keep at least 1s pause between opening and closing a valve for proper operation
- The number of points collected during the adiabatic compression (variable n) should be around 100.
- The results are returned in the `raw_data` matrix as voltage signals, which in turn can be converted to physical quantites:

Data column	Physical quantity	Converting equation
Column 1	Volume	`V=smooth(2.8326e-5*raw_data(:,1)+9.191e-5,3);`
Column 2	Temperature	`T = smooth(33.785 .* raw_data(:,2)+295.193,3);`
Column 3	Pressure	`p = smooth(100000 .* (raw_data(:,3)+0.08),3);`

Calculations

1.

Results, Calculations

Determination of the C_v value of CO_2

1. Based on the graph of C_V and $C_{V,vib}$ vs. T, assess how C_V changes with temperature and which mode(s) contribute to that change the most. Provide a reasoning.
2. Calculate the experimental value of γ with the different masses and deferent gases and arrange your data in a spreadsheet.
    a. If you used the Ruchhardt method, fit a damped oscillation curve on your $p(t)$ data, from ω calculate the period (T) of the oscillation and calculate γ from eqn. (2.32).
    b. If you used the adiabatic method, fit a power function over your $p(V)$ data in Capstone (force x_0=0 and B=0 to ensure proper convergence of the fitting). Obtain γ from the exponent of the fitted function.
3. Calculate the respective C_p and C_V values, using the $C_p=C_V+R$ relationship
4. Compare the theoretical and experimental values of C_V for CO_2.
5. Compare the experimental values of γ for the different gases.

Determination of the second virial coefficient of CO_2

6. To evaluate the experimental data for the first part of the experiment, create the following data table in a new spreadsheet:

p^s (Pa)	p^e (Pa)	Sum	Z_r	γ

The Sum column should have the sum of the pressure readings from the row that it is located to the last pressure reading. Use Equation (2.22) to evaluate Z_r.
    a. Create a graph of Z_r as a function of p^s. Determine A_2, and calculate B_2.
    b. Once you obtained B_2, calculate the γ column.
    c. Perform the analysis for all your runs.
7. Compare your experimental result to the calculated value from Statistical Mechanical considerations and from the semi-empirical approach.
8. Perform error analysis knowing that the literature value of B_2 is -125 mL mol^{-1}.[4]
9. Interpret your results for Z_r (trend, interpolated value at $p = 0$, etc.).

Determination of the Joule-Thomson coefficient of CO_2

10. Determine the Joule-Thomson coefficient from the dT/dp graphs for each gas.
11. Interpret magnitude and sign of the Joule-Thomson coefficient for each gas.

12. Compare the experimental and calculated value of the Joule-Thomson coefficient for CO_2.

13. Based on the collected experimental data for the three experiments, assess which gas behaves more “real” under room temperature and which one less. Explain the trend with the molecular structure of the respective gases.

Appendix A: The Derivation of the JT Coefficient

$$S = f(T, p)$$

$$dS = \left(\frac{\partial S}{\partial p}\right)_T dp + \left(\frac{\partial S}{\partial T}\right)_p dT$$

$$\left(\frac{\partial S}{\partial T}\right)_p :$$

$$dp = 0$$

$$dH = TdS + Vdp = T\left[\left(\frac{\partial S}{\partial p}\right)_T dp + \left(\frac{\partial S}{\partial T}\right)_p dT\right] + Vdp = T\left(\frac{\partial S}{\partial p}\right)_T dp + T\left(\frac{\partial S}{\partial T}\right)_p dT + Vdp$$

$$\left(\frac{\partial S}{\partial T}\right)_p = \frac{1}{T}\left(\frac{\partial H}{\partial T}\right)_p$$

$$\left(\frac{\partial S}{\partial p}\right)_T :$$

$$dT = 0$$

$$dH = T\left(\frac{\partial S}{\partial p}\right)_T dp + Vdp$$

$$\left(\frac{\partial H}{\partial p}\right)_T = T\left(\frac{\partial S}{\partial p}\right)_T + V$$

$$\left(\frac{\partial S}{\partial p}\right)_T = \frac{1}{T}\left[\left(\frac{\partial H}{\partial p}\right)_T - V\right]$$

$$\left(\frac{\partial^2 S}{\partial T \partial p}\right) = \frac{\partial}{\partial T}\left(\frac{\partial S}{\partial p}\right)_T = \frac{\partial}{\partial T}\left\{\frac{1}{T}\left[\left(\frac{\partial H}{\partial p}\right)_T - V\right]\right\} = -\frac{1}{T^2}\left[\left(\frac{\partial H}{\partial p}\right)_T - V\right] + \frac{1}{T}\left[\left(\frac{\partial^2 H}{\partial p \partial T}\right) - \left(\frac{\partial V}{\partial T}\right)_p\right]$$

$$\left(\frac{\partial^2 S}{\partial p \partial T}\right) = \frac{\partial}{\partial p}\left(\frac{\partial S}{\partial T}\right)_p = \frac{\partial}{\partial p}\left[\frac{1}{T}\left(\frac{\partial H}{\partial T}\right)_p\right] = \frac{1}{T}\left(\frac{\partial^2 H}{\partial T \partial p}\right)$$

$$\left(\frac{\partial^2 S}{\partial p \partial T}\right) = \left(\frac{\partial^2 S}{\partial T \partial p}\right)$$

$$\frac{1}{T}\left(\frac{\partial^2 H}{\partial T \partial p}\right) = -\frac{1}{T^2}\left[\left(\frac{\partial H}{\partial p}\right)_T - V\right] + \frac{1}{T}\left[\left(\frac{\partial^2 H}{\partial p \partial T}\right) - \left(\frac{\partial V}{\partial T}\right)_p\right]$$

$$\left(\frac{\partial^2 H}{\partial T\partial p}\right)=-\frac{1}{T}\left[\left(\frac{\partial H}{\partial p}\right)_T-V\right]+\left(\frac{\partial^2 H}{\partial T\partial p}\right)-\left(\frac{\partial V}{\partial T}\right)_p$$

$$0=-\frac{1}{T}\left[\left(\frac{\partial H}{\partial p}\right)_T-V\right]-\left(\frac{\partial V}{\partial T}\right)_p$$

$$\left(\frac{\partial H}{\partial p}\right)_T=V-T\left(\frac{\partial V}{\partial T}\right)_p$$

$$\text{Euler relation}:\left(\frac{\partial T}{\partial p}\right)_H\left(\frac{\partial p}{\partial H}\right)_T\left(\frac{\partial H}{\partial T}\right)_p=-1$$

$$\left(\frac{\partial T}{\partial p}\right)_H=-\frac{1}{\left(\partial H/\partial T\right)_p}\left(\frac{\partial H}{\partial p}\right)_T=-\frac{1}{c_p}\left(\frac{\partial H}{\partial p}\right)_T$$

$$\left(\frac{\partial H}{\partial p}\right)_T=-T\left(\frac{\partial V}{\partial T}\right)_p+V$$

$$\left(\frac{\partial T}{\partial p}\right)_H=-\frac{1}{c_p}\left[-T\left(\frac{\partial V}{\partial T}\right)_p+V\right]=\mu_{JT}$$

Appendix B: Derivation of the equations for adiabatic processes

For ideal gases the heat capacity is defined as:

$$\left(\frac{\partial U}{\partial T}\right)_V = C_V \tag{0.1}$$

The First Law of Thermodynamics is:

$$dU = dq + dw \tag{0.2}$$

For an adiabatic process there is no heat exchange between the system and its environment.

$$\begin{aligned} dU &= 0 + pdV = C_V dT \\ -pdV &= C_V dT \end{aligned} \tag{0.3}$$

Assuming that the gas obeys the Ideal Gas Law, the following derivation holds:

$$\begin{aligned} &pV = nRT \\ &pdV + Vdp = nRdT \\ &\frac{1}{nR}\left(pdV + Vdp\right) = dT \end{aligned} \tag{0.4}$$

Substituting of (0.4) into (0.3) yields (using the C_p-$C_V = nR$ relationship):

$$\begin{aligned} &\frac{C_v}{nR}\left(pdV + Vdp\right) = -pdV \\ &\frac{C_v}{C_p - C_v}\left(pdV + Vdp\right) = -pdV \\ &pdV + Vdp = -pdV\,\frac{C_p - C_v}{C_v} \\ &pdV + Vdp = -pdV(\gamma - 1) \\ &Vdp = -\gamma\, pdV \\ &\gamma\frac{dV}{V} + \frac{dp}{p} = 0 \end{aligned} \tag{0.5}$$

where $\gamma = C_p/C_v$, the ratio of the specific heats at constant pressure and constant volume, respectively. Integrating (0.5) yields:

$$\begin{aligned} &\gamma \ln V - \ln p = k \\ &pV^{\gamma} = k \end{aligned} \qquad \textbf{(0.6)}$$

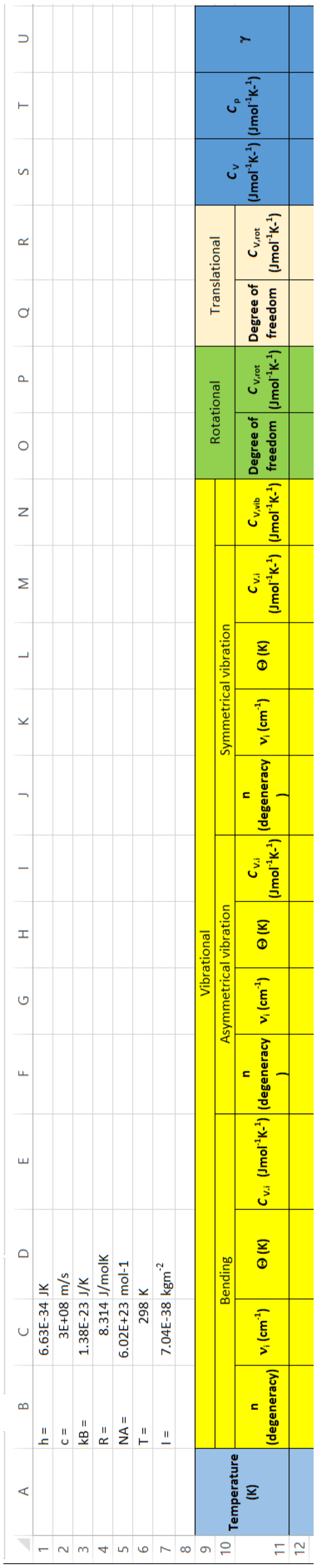
Eliminating pressure from (0.6) would yield a different form of (0.6):

$$TV^{\gamma-1} = k \qquad \textbf{(0.7)}$$

Considering a gas that changes from T_1, p_1, V_1 to $\mathrm{T_2}$, $\mathrm{p_2}$, $\mathrm{V_2}$ these equations would have the form of:

$$\begin{aligned} &p_1 V_1^{\gamma} = p_2 V_2^{\gamma} \\ &T_1 V_1^{\gamma-1} = T_2 V_2^{\gamma-1} \end{aligned} \qquad \textbf{(0.8)}$$

Appendix C: Excel spreadsheet setup for the C_V calculations



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		h =	6.63E-34	JK																	
2		c =	3E+08	m/s																	
3		kB =	1.38E-23	J/K																	
4		R =	8.314	J/molK																	
5		NA =	6.02E+23	mol-1																	
6		T =	298	K																	
7		I =	7.04E-38	kgm^{-2}																	
8																					
9	**Temperature (K)**	Vibrational													Rotational		Translational		**C_V (Jmol^{-1}K-^{1})**	**C_p (Jmol^{-1}K-^{1})**	**γ**
10		Bending				Asymmetrical vibration				Symmetrical vibration											
11		**n (degeneracy)**	**v_i (cm^{-1})**	**Θ (K)**	**$C_{V,i}$ (Jmol^{-1}K-^{1})**	**n (degeneracy)**	**v_i (cm^{-1})**	**Θ (K)**	**$C_{V,i}$ (Jmol^{-1}K-^{1})**	**n (degeneracy)**	**v_i (cm^{-1})**	**Θ (K)**	**$C_{V,i}$ (Jmol^{-1}K-^{1})**	**$C_{V,vib}$ (Jmol^{-1}K-^{1})**	**Degree of freedom**	**$C_{V,rot}$ (Jmol^{-1}K-^{1})**	**Degree of freedom**	**$C_{V,rot}$ (Jmol^{-1}K-^{1})**			
12																					

Experiments 5

Solving an Ordinary Differential Equation (ODE) System

Background

Ordinary Differential Equation Systems in Chemical Kinetics

The rate equations of chemical kinetics are first-order ordinary differential equations. Some examples that you have seen previously are shown below:

A→ Product(s)	$r = -\dfrac{d[\mathrm{A}]}{dt} = k[\mathrm{A}]$	$[\mathrm{A}]_t = [\mathrm{A}]_0 e^{-kt}$
2A → Product(s)	$r = -\dfrac{1}{2}\dfrac{d[\mathrm{A}]}{dt} = k[\mathrm{A}]^2$	$\dfrac{1}{[\mathrm{A}]} = \dfrac{1}{[\mathrm{A}]_0} + kt$

In chemical reactions that involve more species and/or complex reactions (as opposed to elementary reactions as shown above), rate equations become very complex and in those cases typically there is no analytical solution available.

A typical ordinary differential equation of the *i*th species involving *n* species can be written in the form

$$\frac{dc_i}{dt} = f_i(c_1,.....c_n) \quad c(t=0) = c_{i,0} \qquad \textbf{(5.1)}$$

where c_i is the concentration of the *i*th species at time *t*, and $c_{i,0}$ is the concentration of the *i*th species at time *t*=0 (initial conditions). The function $f_i(c_i,\ldots c_n)$ is typically a combination of terms that are made up from concentration(s) of species and some constants (e.g. rate constant). When there is the product of concentrations of more than one species (or the higher power of the concentration of a single species), the equation is *non-linear*. Equations or equation systems involving non-linear kinetic terms have very rich and unique dynamical behavior, discussion of which goes

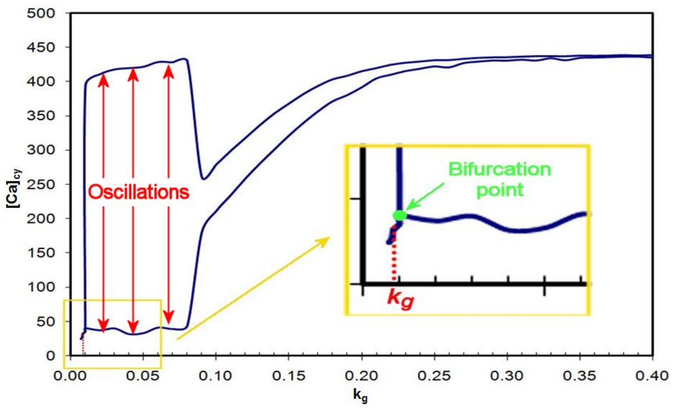


Figure 5.1 Bifurcation diagram. k_g: bifurcation parameter, $[\mathrm{CaCy}]_{\mathrm{cy}}$: cytosolic Ca^{2+}-concentration

beyond the scope of this lab. The few of these properties (related to this lab) are discussed below.

If there is no apparent change over time to the concentration's, and after small perturbation to any of the concentration the system return instantly to the same state, the system is in *stable steady state*, and if it returns to the same state after some "roundtrip" (e.g. some concentrations initially move away from the steady state value, but ultimately the system returns to the steady state then the system is *excitable*. If variation of some parameter results in the loss of the stability of the steady state, the system becomes *oscillatory*, where some or all of the concentrations go through periodic changes over time between two values. Parameter that can invoke this behavior is called a *bifurcation parameter*, and the value of the parameter where that occurs is a *bifurcation point* (see Figure 5.1). Further change of the bifurcation parameter can yield the return of the steady state or further bifurcations, and potentially chaotic behavior. Systems with non-linear dynamics are particularly sensitive to the variation of the bifurcation parameter(s), however they show much less dramatic changes to the variation of other parameters.

When there is no analytical solution available for the differential equation system, the problem has to be solved numerically. Some of such methods will be reviewed in class. While these methods take different approaches, they are similar in the sense that they attempt to evaluate the $f_i(c_i,\ldots c_n)$ functions in discrete time interval in order to obtain the $c_i(t)$ functions, (that is the concentration of the involved species as a function of time) via *numerical integration*. Some of these methods maintain constant time steps (e.g. Euler method), others adjust the time step based on the nature of the $f_i(c_i,\ldots c_n)$ functions (e.g. "predictor/corrector" methods). The latter type of methods are particularly useful integrating systems with concentrations changing several orders of magnitude in a very short time, such as in case of oscillating chemical reactions.

Scientific Modeling

Scientific models are developed to help us to explain experimental results, to gain a better understanding of phenomena in Nature or a subset of Nature which is an experimental configuration or setup. These models are typically some sort of mathematical constructs. There are two, often conflicting expectations towards models: on one hand, they should be able to provide explanations for the observation in an expected depth, on the other hand they should be simple enough that they are tractable, and mathematically feasible. This latter expectation has been hindered by limitations by available computational power. As more computational power became available, more complex, realistic models prevailed.

The objective of this lab will be to implement and verify a mathematical model, which is suitable to provide a plausible explanation for Ca^{2+}-oscillations in *Paramecium* cells and support some relevant experimental data, yet simple enough that it can be managed mathematically.

Deterministic vs. Stochastic Modeling

Deterministic models always generate the same outcome. However, a real cell has many species and many processes, and as you learnt in class, each type of molecule can have many different microstates yielding a variety of macro states. In addition, agonists (such as dopamine, DA) bind to the cell wall and the signal is carried through the cell to its destination plowing through a very much heterogeneous environment. Therefore experimental data is subject to these variations. To address these random variations, there is a number of approaches, one of which is intuitively leveraging the random variations, called *stochastic resonance*. While the experimental data is “noisy”, the underlying periodic signal (oscillation) prevails after the data is processed with a fast Fourier transform (FFT) algorithm (see Figure 5.2).

A model for Ca^{2+}- oscillations in Paramecium cells

Like a good homeowner fixing up his house constantly, implementing new and replacing outdated things, evolution kept adding, removing, changing functionality to cells which resulted in the many different types of cells, each being a fine tuned machinery with many and complex chemical processes. In these complex environments, Ca^{2+} in the cytosol of the cell has been identified as one of the responsible entity to carry signals through the oscillation of its concentration to many cellular processes and between cells.[5] Specifically, the frequency of the oscillation, as rudimentary as it may sound, has been shown to serve as such a signal which is decoded by the target processes.

Each cell implements the Ca-signaling somewhat differently, and many more processes are involved even in one cell than we include in our model, but remember, the objective is to provide a model that is sophisticated enough to explain some experimental results, yet simple enough to manage numerically.

In this lab, we’ll test a combined model assembled from two simpler models, each of which individually captures a subset of the processes. One of the models focuses on the agonist binding process,[6, 7] “CC-model,” and the other one captures the calcium-induced Ca^{2+}-release from the ER,[8-10] “BDG-model” (both abbreviated after the authors). The experimental results that will be used to match against is shown in Figure 5.2

The schematics of the model is shown in Figure 5.3 and Figure 5.4, and the respective differential equations are shown in Figure 5.5. Table 5.4 shows the considered species (dynamical variables) and Table 5.5 shows the explanations for all the other variables, constants, etc. in the model. Please note, constants and variables referring to the schematic model are in the italics type (e.g. k_g), references to the Matlab code are in block type (e.g. **`kg`**).

The differential equations are expressed in terms of *variables* for readability reason and also some of the terms individually can be of an interest. For example, V_3, is Hill-functions describing the binding of multiple Ca^{2+} and IP_3 to the IP_3-receptor (IP_3R), with each step having a different rate (cooperativity effect).

. The objective is ***to test the response of the model to the variation of the bifurcation parameter (by constructing the bifurcation diagram), k_g, which represents the rate determining step of the binding process of the agonist to the receptor, specifically, if the model can reproduce the experimental 0.036Hz frequency identified in the experimental data, and also to predict at what values of the bifurcation parameter the oscillation would cease to exist.***

The Modeling Environment

The Mathematical/Programing Platform

The modeling will be performed in MATLAB with the Systems Biology Toolbox (SBTB) installed. The SBTB allows to create a text file with a specific format which resembles the mathematical model in a much more human-friendly way than the programing environment of MATLAB. While the text file is very convenient, it has to have a specific format/syntax that the SBTB engine can read and process. Once the model is setup in a text file, the SBTB engine can be invoked manually from the MATLAB prompt, or it can be invoked from a MATLAB script. The latter option also allows varying a parameter of the model (which is also present in the text file) and rerun the model with the new value of the parameter automatically. This way the behavior of the model can be tested automatically in response to the variation of a parameter.

The Model File

The structure of the model file is shown in Table 5.1.

`********** MODEL NAME`
`ZQX Model and CC-BDG Model`
	The name of your model (not used in the simulation)

`********** MODEL NOTES`
`Combined model`
	Notes for yourself

********** MODEL STATES d/dt(CaCy) = R1+R2-R3+R4+R5-R6 {isSpecie::concentration} %state d/dt(CaER) = R3-R4-R5 {isSpecie::concentration} %state d/dt(IP3) = R7-R8-R9 {isSpecie::concentration} %state d/dt(GaGTP) = R11-4*R12-R13 {isSpecie::concentration} %state d/dt(DAG) = R7-R14+R15 {isSpecie::concentration} %state d/dt(aPLC) = R12-R16 {isSpecie::concentration} %state	The differential equations for the rates for each species in a new line. Best is using rate variables (R1, R2,…) defined later for visibility . The names in parenthesis represent the concentration of the respective species.
CaCy(0) = 250 CaER(0) = 100 IP3(0) = 0.8 GaGTP(0) = 60 DAG(0) = 40 aPLC(0) = 8	Initial conditions for each species (in mM)

********** MODEL PARAMETERS
kg0 = 0.02 {isParameter} %parameter
V0 = 400 {isParameter} %parameter
k = 70 {isParameter} %parameter
……
	Parameters (e.g. rate constants) of the model. These are not updated automatically during the modeling

********** MODEL VARIABLES
khp = khp0*DAG^2/(KD1^2+DAG^2) {isParameter} %variable
kDAG = kDAG0*DAG^2/(KD1^2+DAG^2) {isParameter} %variable
kPLC = kPLC0*DAG^2/(KD1^2+DAG^2) {isParameter} %variable
…..
	Variables are used to keep the model clean and easy to read. Here, Hill-functions are defined which typically describe reactions involving proteins

********** MODEL REACTIONS
R1 = V0 %reaction
R2 = V1 %reaction
R3 = V2 %reaction
……
	The definition of the reaction variables in terms of parameters

********** MODEL FUNCTIONS
 	
********** MODEL EVENTS
 	
********** MODEL MATLAB FUNCTIONS
function m = modulation(time,kgbase,kgpeak,tbase,tpeak)
per = tbase + tpeak
residue = mod(time,per)
if residue <= tbase
m = kgbase
else
m = kgpeak
end
return
	Custom functions. We will use this option to generate the time-dependent stimulation y the agonist

Table 5.1 Structure of the model file

Running the modeling

There are several ways of running the modeling:

1. A short script can be executed that generates a plot of the $[Ca^{2+}]_{Cy}$ vs. time and FFT vs. frequency plots.

`model = SBmodel('CombCa.txt')`
`iTime = 1024; %327.68;`
`t = (0:1.0:iTime);`
`np=iTime/1.0;`
	`'CombCa.txt':` text file with the model
	model: binary representation of the model
	`iTime:` the duration of the reaction (in s)
	`t:` vector containing the time points where the integration has to be performed (points between 0 and **`iTime`** with 1.0 s step)
	`np`: number of time points

`output = SBsimulate(model,'ode23s',t);`	**`output:`** data structure with results

`fftc=abs(fft(output.statevalues(:,1))).^2;`
`nn = length(fftc);`
`dt=1/iTime;`
`fs=np/iTime;`
`f = (0:dt:fs);`
	`output.statevalues(:,1):` $[Ca^{2+}]_{Cy}$ array with concentration values as a function of time
	`fftc:` array with the FFT values of the [CaCy]
	`nn:` length of the **`fftc`** vector
	`fs:` number of frequency points for FFT
	`f:` array with the frequency points for FFT

`figure(1);`
`plot(t,output.statevalues(:,1));`
`axis([0 iTime 0 1000 ]);`
	`figure(1):` figure for $[Ca^{2+}]_{Cy}$ vs. time
	`plot, axis:` making the plot and set the axes

`figure(2);`
`plot(f,fftc);`
`axis([0 0.2 0 1e12]);`
	`figure(2):` figure for FFT vs. frequency
	`plot, axis:` making the plot and set the axes

Table 5.2 Structure of a basic script to perform the modeling

2. Using the graphics interface to the SBTB. Construct the model and invoke the editor:

```
model = SBmodel('CombCa.txt');
SBedit(model);
```

3. Constructing a more involved scripts that varies the desired parameter or parameters over a range with a given step size (“parameter sweeping”). For a sample and commented script please refer to the Appendix.

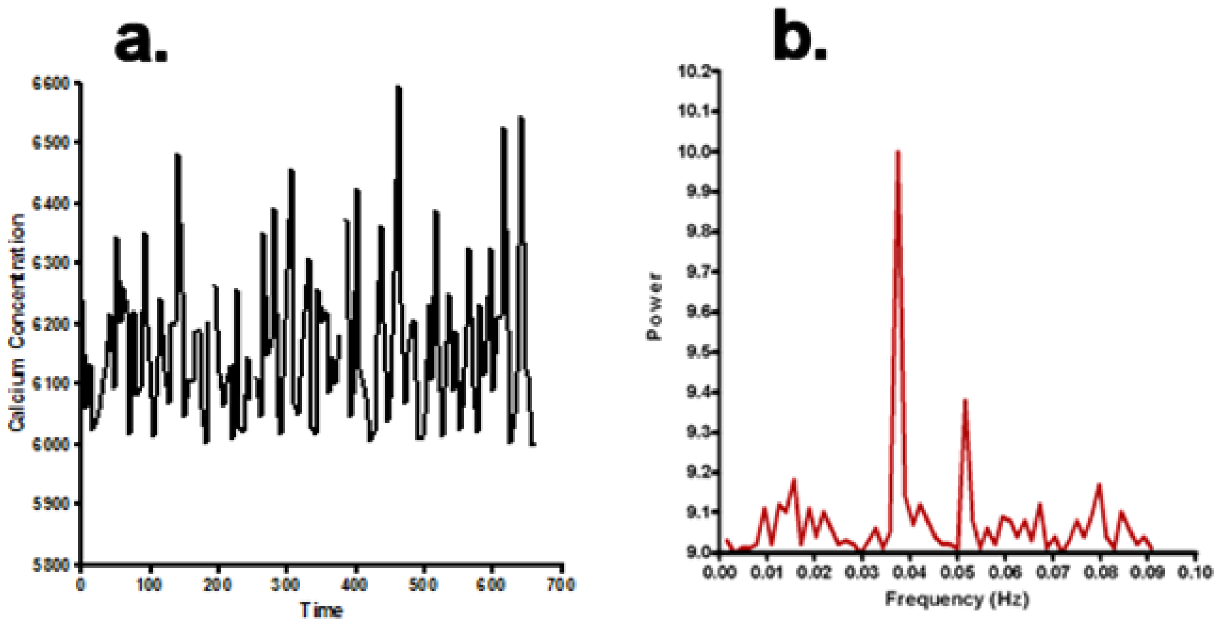
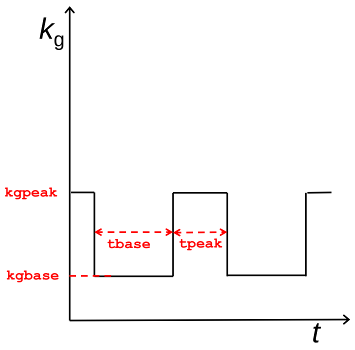
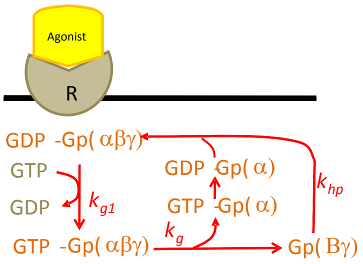


Figure 5.2 Experimental CaCy concentration in a PC12 cell (a) and the respective FFT spectrum (b)

	Step	Representation in model
1.	• Agonist (such as deltamethrin, DA) binds to its receptor on the membrane of the cell (*fast*), typically in a periodic fashion. • G-protein [Gp(αβγ)] gets activated (using the energy from a GTP molecule), which is also *fast*. • G-protein splits into it's Gp(α) and Gp(βγ) components. *This is a slow process, hence it is the rate determining step of the activation process, with a rate constant* k_g (**`kg`**), which varies in time (see on the right) and it is scaled with **`kg0`**. • The binding increases when treatment is added to **`kgadj`** at time **`t1`**. • k_g is set with a function called *modulation*. • Gp(αβγ) is regenerated (k_{hd})	  Agonist R GDP -Gp(αβγ) GTP GDP k_{g1} GTP -Gp(αβγ) GDP Gp(α) GTP -Gp(α) k_g k_{hp} Gp(Bγ) **`kg = kg0*modulation(time,t1,kgadj, kgbase,kgpeak,tbase,tpeak`**

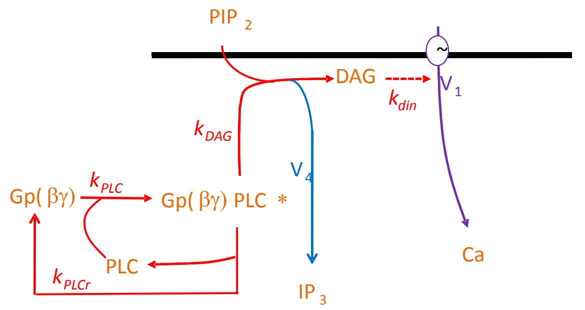
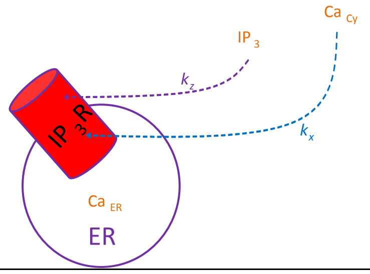
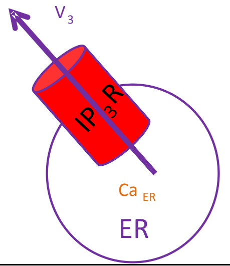
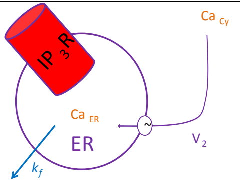
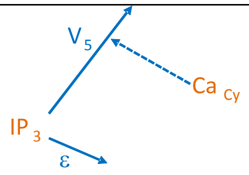
2.	- The Gp(βγ) portion of this protein activates phospholipase C (PLC) which in turn, cleaves (k_{DAG}) phosphatidylinositol 4, 5 bisphospate (PIP_2) to inositol (1,4,5) trisphosphate (IP_3) and diacylglycerol (DAG). - DAG is responsible indirectly for the Ca^{2+}-channels to open on the cell membrane allowing the Ca^{2+} to flood into the cell, which is present at 20,000-fold higher concentration outside the cell (k_{din}). - Activated PLC can get deactivated (k_{PLCr})	 PIP 2 DAG ~ k_{din} V 1 k_{DAG} V 4 Gp(βγ) k_{PLC} Gp(βγ) PLC * Ca PLC k_{PLCr} IP 3
3.	1. IP_3 binds to the IP_3-receptor (IP_3R) of the endoplasmic reticulum (ER) and along with the cytosolic Ca^{2+} (CaCy) activates the receptor.	
4.	2. Upon proper activation of IP_3R, Ca is released from the ER (Ca^{2+}-induced Ca^{2+} release, CICR), V_3	
5.	3. Smooth endoplasmic reticular Ca^{2+} ATPase (SERCA) pumps replete the ER with Ca^{2+} (V_2). Ca^{2+} also leaks out from the ER (k_{f}).	
6.	4. IP_3 is removed via Ca^{2+}-dependent (V_5) and Ca^{2+}-independent (ε) pathways.	

Table 5.3 Details of the model

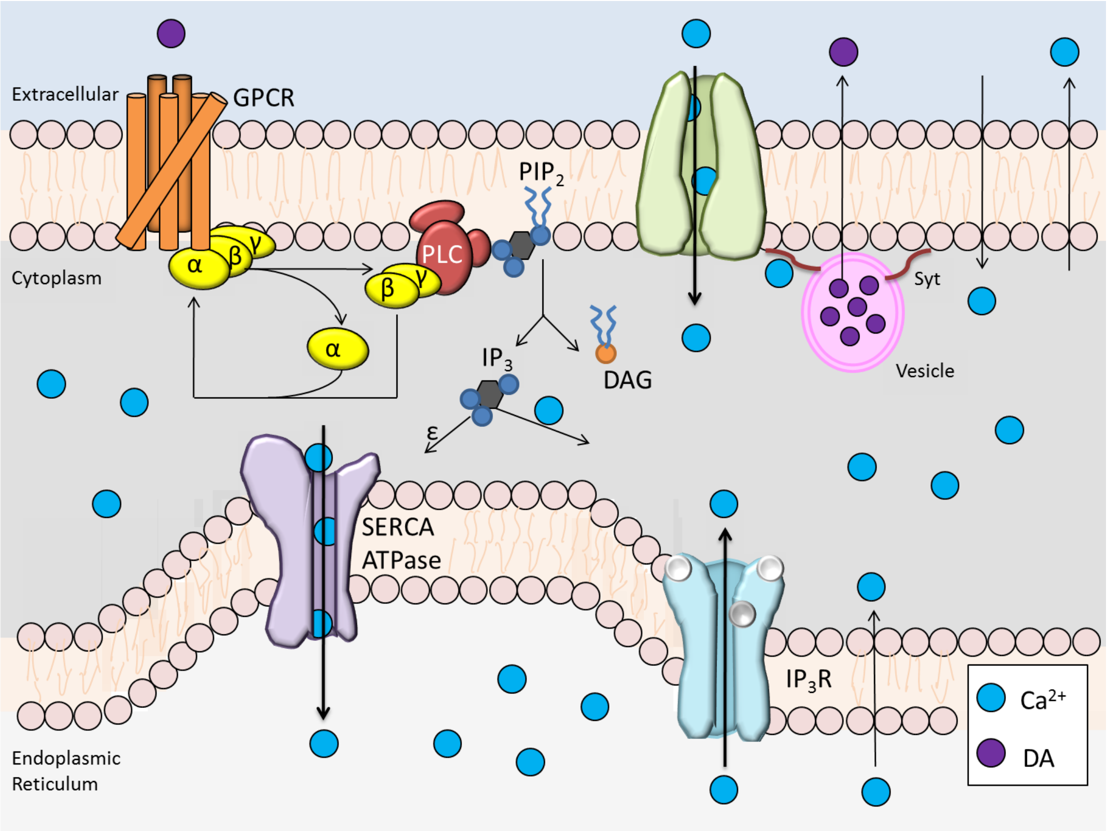


Figure 5.3 The G-protein cascading mechanism and CICR-facilitated Ca^{2+}-dynamics

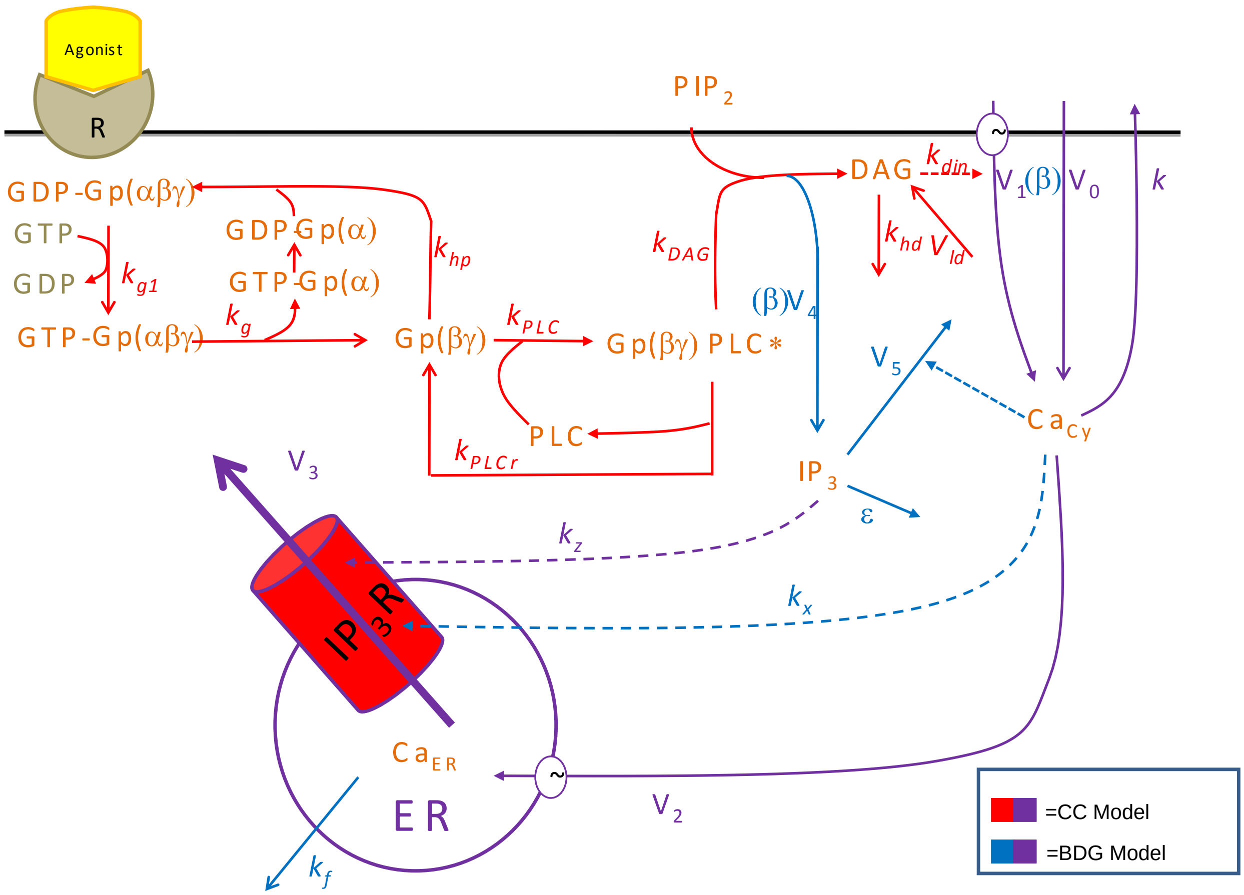


Figure 5.4 Schematics of the model of Ca^{2+}-dynamics in *Paramecium* cells

$$\frac{d[Ca]_{Cy}}{dt} = V_0 + V_1 - V_2 + V_3 + k_f[Ca]_{ER} - k[Ca]_{Cy}$$

$$\frac{d[Ca]_{ER}}{dt} = V_2 - V_3 - k_f[Ca]_{ER}$$

$$\frac{d[IP_3]}{dt} = V_4 - V_5 - \varepsilon[IP_3]$$

$$\frac{d[G_\alpha - GTP]}{dt} = k_g[G_\alpha - GDP] - \\ - 4k_{PLC}[G_\alpha - GTP]^4[PLC] - k_{hg}[G_\alpha - GTP]$$

$$\frac{d[DAG]}{dt} = k_{DAG}[PLC^*] - k_{hd}[DAG] + V_{ld}$$

$$\frac{d[PLC^*]}{dt} = k_{PLC}[G_\alpha - GTP]^4[PLC] - k_{hp}[PLC^*]$$

$$V_1 = k_{din}[DAG]$$

$$V_2 = V_{M2}\frac{[Ca^{2+}]_{Cy}^2}{k_2^2 + [Ca^{2+}]_{Cy}^2}$$

$$V_3 = V_{M3}\frac{[Ca^{2+}]_{Cy}^2}{k_x^2 + [Ca^{2+}]_{Cy}^2}\frac{[Ca^{2+}]_{ER}^2}{k_y^2 + [Ca^{2+}]_{ER}^2}\frac{[IP_3]^4}{k_z^4 + [IP_3]^4}$$

$$V_4 = k_{DAG}[PLC^*]$$

$$V_5 = V_{M5}\frac{[Ca^{2+}]_{Cy}^n}{k_d^n + [Ca^{2+}]_{Cy}^n}\frac{[IP_3]^p}{k_5^p + [IP_3]^p}$$

$$[G_\alpha - GDP] = G_0 - [G_\alpha - GTP] - 4[PLC^*]$$

$$[PLC] = P_0 - [PLC^*]$$

$$k_n = k_n^{'}\frac{[DAG]^2}{K_{D1}^2 + [DAG]^2}, \qquad n = k_{hp}, k_{DAG}, k_{PLC}$$

Figure 5.5 Mathematical model for Ca^{2+}-oscillation

Parameter	Model representation	Maximum value	Description
$[Ca]_{Cy}$	CaCy		Concentration of Ca^{2+} in the cytosol
$[Ca]_{ER}$	CaER		Concentration of Ca^{2+} in the ER
$[IP_3]$	IP3		Concentration of IP$_3$ in the cytosol
$[G_{\mathrm{a}}\text{-}GTP]$	GaGTP		Concentration of activated G-protein on the inner side of the agonist receptor
$[DAG]$	DAG		Concentration of DAG in the cytosol
$[PLC*]$	aPLC		Concentration of aPLC (activated PLC) in the cytosol

Table 5.4 Species (dynamical variables) for the model of Ca^{2+}-oscillations

Parameter	Model representation	Maximum value	Description
k_g	kg0		Stimulation by agonist
k_{PLC}	kPLC	kPLC0	Rate of formation of PLC*
k_{hg}	khg		Rate of hydrolysis of GTP to GDP
k_{DAG}	kDAG	kDAG0	Production of DAG
k_{hd}	khd		Hydrolysis of DAG
V_{ld}	Vld		Passive leak of DAG
k_{hp}	Khp		Hydrolysis of PLC*
G_0	G0		Total concentration of G-proteins
P_0	P0		Total concentration of PLC
k_{din}	kdIN		Rate of Calcium entry from extracellular stores
k	k		Passive efflux of Ca from cytosol
V_0	V0		Passive leak of Ca
V_1	V1		Active influx of ca to cytosol from extracellular stores
V_2	V2	VM2	Pumping rate into ER
V_3	V3	VM3	Pumping rate out of ER
k_y	ky		Threshold for Ca release from ER
k_z	kz		Threshold for activation by IP$_3$
V_4	V4		Agonist dependent production of IP$_3$

V_5	V5	VM5	Ca-dependent IP_3 degradation
k_f	kf		Passive efflux of Ca from ER to cytosol
ε	eps		Ca-independent degradation of IP_3
k_x	kx		Threshold for activation by Ca
(β)			(Production of IP_3 –not explicitly included in this model, replaced by k_g)
n,m,p,q	n,m,p,q		Cooperativity factors (integers)

Table 5.5 Parameters used in the mathematical model for Ca^{2+}-oscillation

Results, Calculations

For your lab report:

1. Perform the modeling with varying k_g over a range that captures the two bifurcation points (or the amplitude of the oscillation falls below 100 nM)
2. Construct the bifurcation diagram.
3. Interpret the meaning of the bifurcation diagram.
4. Identify the value of k_g that matches the experimental data (that is generates oscillating signal with approximately the same frequency.
5. Construct the FFT diagram and identify the value of the oscillation matching the experimental data.
6. Based on the bifurcation diagram, argue if the experimental data was collected near either steady state (either bifurcation point, upper or lower) or in the .middle of the oscillatory range, and predict the effect of small variation to k_g (e.g. some environmental effect).

Appendix A: model file for SBToolbox for modeling Ca^{2+}-oscillations in *Paramecium*

```
********** MODEL NAME
ZQX Model and CC1 Model

********** MODEL NOTES
Combined model

********** MODEL STATES
d/dt(CaCy) = R1+R2-R3+R4+R5-R6 {isSpecie::concentration}[6] %state
d/dt(CaER) = R3-R4-R5 {isSpecie::concentration} %state
d/dt(IP3) = R7-R8-R9 {isSpecie::concentration} %state
d/dt(GaGTP) = R11-4*R12-R13 {isSpecie::concentration} %state
d/dt(DAG) = R7-R14+R15 {isSpecie::concentration} %state
d/dt(aPLC) = R12-R16 {isSpecie::concentration} %state

CaCy(0) = 30
CaER(0) = 100
IP3(0) = 0.80000000000000004
GaGTP(0) = 60
DAG(0) = 40
aPLC(0) = 8

********** MODEL PARAMETERS
kg0 = 0.017000000000000001 {isParameter} %parameter
V0 = 800 {isParameter} %parameter
k = 15000 {isParameter} %parameter
kf = 1 {isParameter} %parameter
eps = 1 {isParameter} %parameter
VM2 = 700 {isParameter} %parameter
VM3 = 150 {isParameter} %parameter
VM5 = 5700 {isParameter} %parameter
k2 = 0.10000000000000001 {isParameter} %parameter
k5 = 1 {isParameter} %parameter
kx = 0.5 {isParameter} %parameter
ky = 0.20000000000000001 {isParameter} %parameter
kz = 0.20000000000000001 {isParameter} %parameter
kd = 0.40000000000000002 {isParameter} %parameter
kl = 1 {isParameter} %parameter
m = 2 {isParameter} %parameter
p = 2 {isParameter} %parameter
n = 4 {isParameter} %parameter
q = 1 {isParameter} %parameter
G0 = 200 {isParameter} %parameter
P0 = 10 {isParameter} %parameter
KD1 = 25 {isParameter} %parameter
kgbase = 0.5 {isparameter} %parameter
kgpeak = 1.1000000000000001 {isparameter} %parameter
kgadj = 12.199999999999999 {isparameter} %parameter
```

```
tbase = 24.699999999999999 {isparameter} %parameter
tpeak = 2 {isparameter} %parameter
khd = 100 {isParameter} %parameter
khg = 0 {isParameter} %parameter
Vld = 1000 {isParameter} %parameter
kdIN0 = 10 {isParameter} %parameter
kdIN0adj = 900 {isParameter} %parameter
khp0 = 0.5 {isParameter} %parameter
kDAG0 = 1000 {isParameter} %parameter
kPLC0 = 1.9999999999999999e-007 {isParameter} %parameter
t1 = 100 {isParameter} %parameter
t2 = 300 {isParameter} %parameter

********** MODEL VARIABLES
khp = khp0*DAG^2/(KD1^2+DAG^2) {isParameter} %variable
kDAG = kDAG0*DAG^2/(KD1^2+DAG^2) {isParameter} %variable
kPLC = kPLC0*DAG^2/(KD1^2+DAG^2) {isParameter} %variable
kdIN = mod_dIN2(time,t2,kdIN0,kdIN0adj) {isParameter} %variable
V1 = kdIN*DAG {isParameter} %variable
V2 = VM2*CaCy^2/(k2^2+CaCy^2) {isParameter} %variable
V3 =
VM3*CaCy^m*CaER^2*IP3^4/((kx^m+CaCy^m)*(ky^2+CaER^2)*(kz^4+IP3^4))
{isParameter} %variable
V4 = kDAG*aPLC {isParameter} %variable
V5 = VM5*IP3^p*CaCy^n/((k5^p+IP3^p)*(kd^n+CaCy^n)) {isParameter}
%variable
PLC = P0-aPLC {isParameter} %variable
GaGDP = G0-GaGTP-4*PLC {isParameter} %variable
kg = kg0*modulation(time,t1,kgadj,kgbase,kgpeak,tbase,tpeak)
{isParameter} %variable

********** MODEL REACTIONS
R1 = V0 %reaction
R2 = V1 %reaction
R3 = V2 %reaction
R4 = V3 %reaction
R5 = kf*CaER %reaction
R6 = k*CaCy %reaction
R7 = V4 %reaction
R8 = V5 %reaction
R9 = eps*IP3 %reaction
R11 = kg*GaGDP %reaction
R12 = kPLC*GaGTP^4*PLC %reaction
R13 = khg*GaGTP %reaction
R14 = khd*DAG %reaction
R15 = Vld %reaction
R16 = khp*aPLC %reaction

********** MODEL FUNCTIONS


********** MODEL EVENTS
```

```
********** MODEL MATLAB FUNCTIONS
function m = modulation(time, t1, kgadj, kgbase,kgpeak,tbase,tpeak)
    per = tbase + tpeak;
    residue = mod(time,per);
        if residue <= tbase
            m = kgbase;
        elseif (time > t1)
            m = kgadj;
        else
            m = kgpeak ;
        end
return

function v = mod_dIN2(time,t2, kdIN0,kdIN0adj)
    if (time >= t2)
          v = kdIN0adj * (time - t2) ;
    else
          v = kdIN0;
    end

return
```

Appendix B: Script file for “Parameter sweep”

```
model_file = 'CombCa.txt';                                                           %Defining the model file
numpar_1 = 100;                                                                      %Number of steps of first parameter for sweeping
(Parameter 1)
numpar_2 = 1;                                                                        %Number of steps of second parameter for sweeping
(Parameter 2)
output_file_root = 'raw/2014_01_';                                                   %Raw Excel data output folder and root name for the
data files
output_image_root = 'images\';                                                       %Image output folder and root name for the data files

par_1 = 0.02;                                                                        %Starting value of Parameter 1
par_2 = 700.0;                                                                       %Starting value of Parameter 2
parrun = 1;
par_1_step = 0.001;                                                                  %Stepsize of Parameter 1
par_2_step = 25.00;                                                                  %Stepsize of Parameter 1
par_2_initial = par_2;                                                               %Saving the initial value of Parameter 2

iTime = 1000;                                                                        %Length of simulation
dt = 0.1;                                                                            %Time interval between integrations
t = (0:dt:iTime);                                                                    %Time vector with time steps dt between 0 and iTime
np=iTime/dt;                                                                         %Number of time points (real)

for kdx = 1:numpar_1                                                                 %Loop to sweep Parameter 1

    par_2 = par_2_initial;                                                           %Reseting the initial value of Parameter 2

    for idx = 1:numpar_2                                                             %Loop to sweep Parameter 2
        Z=[];                                                                        %Array to save [CaCy] vs. t values
        ZEr = [];                                                                    %Array to save [CaER] vs. t values
        ZFFT=[];                                                                     %Array to save FFT of [CaCy] vs. frequency values
        model = SBmodel(model_file);                                                 %Creating the binary version of the model
        modelstructure = SBstruct(model);                                            %Extracting a data structure from the model
        modelstructure.parameters(1).value = par_1;                                  %Setting the new value of Parameter 1
        modelstructure.parameters(2).value = par_2;                                  %Setting the new value of Parameter 2
        model1 = SBmodel(modelstructure);                                            %Creating the binary version of the modified model
        output = SBsimulate(model1,'ode23s',t);                                      %Performing the integration
        fftc=abs(fft(output.statevalues(:,1))).^2;                                   %Calculating the FFT values of [CaCy]
        nn = length(fftc);                                                           %Number of points in FFT (power of 2 typically)
        dt_fft=1/iTime;
        fs=np/iTime;                                                                 %Number of frquency points in the FFT spectrum
        f = (0:dt_fft:fs);                                                           %Vector holding the frequency points
        Z = [Z output.statevalues(:,1)];                                             %Filling Z helper array with the [CaCy] vs. t values
        ZEr = [ZEr output.statevalues(:,2)];                                         %Filling ZEr helper array with the [CaER] vs. t values
        ZFFT = [ZFFT fftc];                                                          %Filling ZFFT helper array with the FFT of [CaCy] vs.
frequency values
        par_1str = num2str(par_1);                                                   %Creating a string with the value of Parameter 1
        par_2str = num2str(par_2);                                                   %Creating a string with the value of Parameter 2
        output_file = [output_file_root '(' par_1str ')(' par_2str ').csv'];         %Creating a name of the output Excel file for [CaCy]
from the root name, and the strings of Parameter 1 and 2
        output_file2 = [output_file_root '(' par_1str ')(' par_2str ')FFT.csv'];     %Creating a name of the output Excel file for FFT data
from the root name, and the strings of Parameter 1 and 2
```

```
        output_file3 = [output_file_root '(' par_1str ')(' par_2str ')ER.csv'];      %Creating a name of the output Excel file for [CaER]
from the root name, and the strings of Parameter 1 and 2
        csvwrite(output_file, Z);                                                    %Writing [CaCy] data into an Excel file
        csvwrite(output_file2, ZFFT);                                                %Writing FFT data into an Excel file
        csvwrite(output_file3, ZEr);                                                 %Writing [CaER] data into an Excel file

        fig1 = figure(1);
        set(fig1,'position', [50 50 950 950]);                                       %Creating the figure with the 3 plots
        subplot(2,2,1);
        plot(t,output.statevalues(:,1));
        axis([0 iTime 0 1000 ]);
        title(['CaCy (par_1=' par_1str ',par_2=' par_2str ')']);
        subplot(2,2,2);
        plot(t,output.statevalues(:,2));
        axis([0 iTime 0 1000 ]);
        title('CaER');
        subplot(2,2,3);
        plot(f,fftc);
        fft_max =  1.05*max(fftc(100:length(fftc)));
        axis([0 0.1 0 fft_max]);
        title('CaCy FFT');


        disp(['par_1=' num2str(par_1) ] )
        saveas(fig1,[output_image_root '(' par_1str ')('  par_2str ')CaCy.jpg']); %Saving the figures in MATLAB and in .jpg format
        saveas(fig1,[output_image_root '(' par_1str ')('  par_2str ')CaCy.fig']); %Saving the figures in MATLAB and in .fig format
        par_2 = par_2 + par_2_step;
    end
    par_1 = par_1 + par_1_step;
end
```

Experiments 6

Solving a Partial Differential Equation (PDE) System

Background

Partial Differential Equation Systems in Chemical Kinetics

We saw that a homogenous chemical system with multiple chemical reactions can be described with an ODE sysyem. However, if the system is not homogenous, the special variation of the concentrations have to be taken into account. Spatially varying concentrations represent concentration gradients which can be taken into account with considering diffusion in the reaction domain.

The general equation for a non-homogenous reaction-diffusion system is

$$\frac{\partial c_k}{\partial t} = D_k \nabla^2 c_k + f_k(c_1,.....c_n) \quad c(t=0) = c_{k,0}$$
$$\nabla^2 c_k = \left(\frac{\partial^2 c_k}{\partial x^2} + \frac{\partial^2 c_k}{\partial y^2} \right) \qquad \textbf{(6.1)}$$

where c_k is the concentration and D_k is the diffusion coefficient of the *k*th species, and $c_{k,0}$ is the concentration of the *k*th species at time *t*=0 (initial conditions), the ∇^2 term has the spatial variables (in the *x* and *y* dimension). The function $f_k(c_i,\ldots c_n)$ contains the reaction rate terms. Mathematically these equations constitute a *partial differential equation system* (PDE).

The solution of such systems can be performed numerically, much like we saw with ODE, however the diffusion term has to be approximated as well. Further complication is that while homogeneous system could be characterized with one set of concentration values, in systems with diffusion each point of the system has a different set of concentrations therefore the PDE has to be solved for each point of the system at every time step. One can imagine how much more tedious such a job is in comparison with the solution of an ODE! Therefore we will consider only a small part of the system with the assumption that it is representative for the entire system.

Numerical Method for Integration

While in the previous lab the ODE system was solved under the hood, now performing the numerical integration is part of the adventure. The Euler-method will be employed, which is perhaps one of the least computationally tasking approach. The method can be summarized with eqns. (6.2).

$$\begin{aligned}
&\frac{\partial c_k}{\partial t} \approx \frac{\Delta c_k}{\Delta t} \approx f_k(c_1,.....c_n) \\
&\Delta c_k = \left(c_{k,t+\Delta t} - c_{k,t}\right) \approx f_k(c_1,.....c_n)\Delta t \\
&c_{k,t+\Delta t} \approx c_{k,t} + f_k(c_1,.....c_n)\Delta t
\end{aligned} \tag{6.2 }$$

where Δt is the time step, $c_{\mathrm{k,t}}$ is the concentration of the kth species at time t, $c_{\mathrm{k,t+\Delta t}}$ is the concentration of the kth species at time t+Δt. The time step is chosen arbitrarily, however care has to be taken. The smaller it is chosen, the more accurate evaluation of the $f(c_{1,}\ldots c_{\mathrm{n}})$ function is possible, however from the perspective of the calculation, the larger it is the faster the calculation goes.

Approximating the Diffusion Terms

For the purpose of this lab, we will consider two spatial dimensions, x and y. We will divide the reaction domain into small “cells”, and assume that each cell communicates with its neighbor cell via diffusion (Figure 6.1). If the cells are small enough, the diffusion term can be approximated with the concentration differences

$$\begin{aligned}
D_k \nabla^2 c_k &\approx D_k \left(\frac{\Delta c_k}{\Delta x^2} + \frac{\Delta c_k}{\Delta y^2} \right) = \\
&= D_k \frac{\left(c_{i+1,j} - c_{i,j}\right) + \left(c_{i-1,j} - c_{i,j}\right) + \left(c_{i,j+1} - c_{i,j}\right) + \left(c_{i-1,j} - c_{i,j}\right)}{\Delta h^2} = \\
&= D_k \frac{c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i-1,j} - 4c_{i,j}}{\Delta h^2}
\end{aligned} \tag{6.3 }$$

where $\Delta h = \Delta x = \Delta y$ is the size of a cell (assuming square cell it is the same in both directions), and D_{k} is the diffusion coefficient of the kth species.

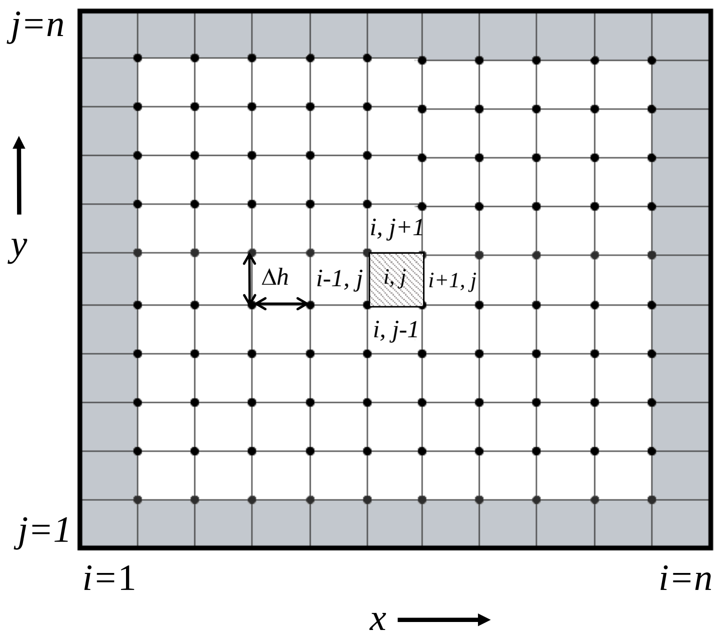


Figure 6.1 Reaction-diffusion system domain

Boundary conditions

Since we included only small part of the actual system, we have to assume that the cells at the boundary will communicate with cells not included in the treatment. The appropriate boundary condition to reflect this assumption can be approximated with setting the concentrations in the first and last rows and columns to the values of the concentrations in the second to last rows and column. For example the concentrations in the first column are set to the concentrations in the third column, concentrations in the last row (*n*th row) are set to the concentrations in the (*n*-2)th row, etc. (see Figure 6.1 grey area). Therefore the cells in the first and last columns and rows communicate with cells outside of the domain the same way as their do with their immediate neighbor cells.

The actual system – Modeling Propagation of Cardiac Potentials with the FitzHugh-Nagumo Model

We saw, that the signal carried by Ca^{2+}-ions inside a cell can be modeled with an ODE system. The same approach can be used to model how the signal is delivered to the cellular mechanism which is responsible for releasing for example dopamine which ultimately carries the signal to the next cell. To model a network of these cells would require the solution of the ODE system for each cell at every time step, and then solving the respective PDE system accounting for the signal propagation (i.e. diffusion of the agonist) among cells. One can imagine the computational task that such an enterprise would entail. Alternatively, we could conceivably bypass what is happening in the cells with the notion that it is triggered by a potential change on the membrane and it ultimately results in a potential change on the membrane, and it is sufficient to follow the potential change across the domain of cells.

To employ this approach we will use a relatively simple model called the *Fitzhugh-Nagumo* model, which models the potential propagation across a cell domain. This simplified model has two variables: the action potential in the domain (*v*) and the so called recovery variable (*r*).

$$\frac{\partial v}{\partial t} = v\left(v-a\right)\left(1-v\right) - r + d\nabla^2 v \qquad (6.4\ )$$
$$\frac{\partial r}{\partial t} = e\left(bv - r\right)$$

where *a,b,* and *e* are constants, and *d* is the “diffusion coefficient”, which in this case the conductance (i.e. coupling strength between cells). The model in this form is dimensionless. Notice that the recovery variable doesn’t propagate between cells, as it is an inherent property of the domain.

Objective of the Lab

The objective of the lab is to show how the action potential propagates through a medium using Fitzhugh-Nagumo model, when the medium is not homogenous.

It has been argued that minor inhomogeneity in the medium, such as a damaged segment in the heart tissue can result in breaking the propagating action potential and ultimately yield a self-sustaining potential wave which in turn can lead to cardiac arrest. If one considers the propagating action potential as a wave of potential change that sweeps through the medium as it carries the signal, a damaged portion of the tissue can break the wave leading to a wave with a free and which starts to rotate locally and will never leave the tissue constantly triggering an action potential which is a known precursor to cardiac arrest (see Figure 6.3 bottom row).

Another source of free end in the action potential is early action potential propagating in the wake of the previous one. The cell needs some time after passing an action potential to recover (this is incorporated into the model as *r*), however an early excitation can initiate an action potential. The generation of a normal action potential (AP) is shown in Figure 6.2. If a stimulation greater than the threshold arrives between *A* and *D*, a new action potential is generated before the resting potential is restored. Spatially, this kind of excitation would occur right behind the previous AP (between *B* and *D* on the time scale), where the medium is not homogeneous, therefore the new action potential will not be uniform, therefore free ends may form (see Figure 6.3 top row).

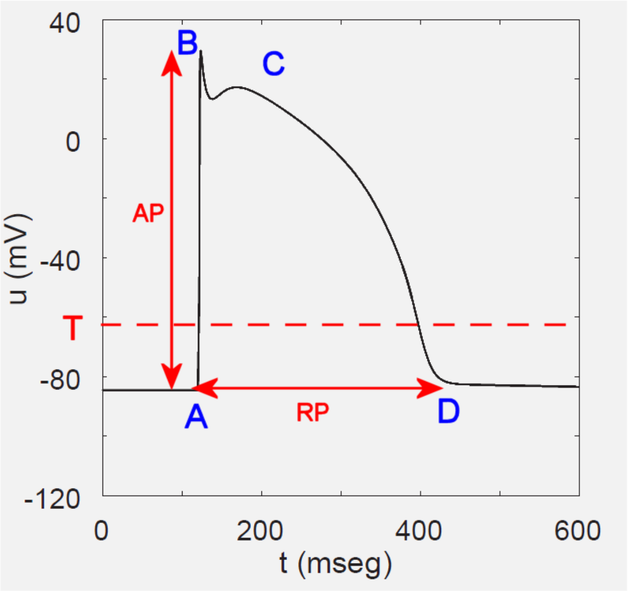


Figure 6.2 Action potential propagation in time in cardiac tissue

AP = action potential; RP = refractory period; T = AP threshold (~-60 mV); A = resting potential, stimulus arrives, Na$^+$ channels open; B = Na$^+$ channels close, K$^+$ channels open; C = Ca^{2+} channels open; D = resting potential restored, Ca^{2+} and K$^+$ channels close

The model therefore should:

- Show that undisturbed action potential propagates through the medium without interruption.
- Demonstrate that action potential which is interrupted in a manner that a free end of the action potential wave is formed will sustain itself indefinitely.
- Create a scenario where a new action potential is formed in the wake of the previous action potential resulting free-ended in rotating potential wave.

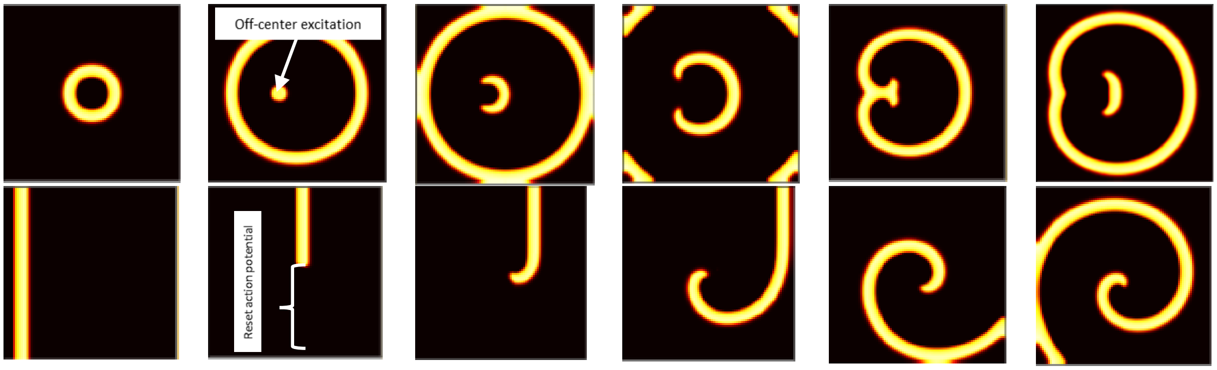


Figure 6.3 Time evolution of action potential with point-like excitation (top row) and with perturbation that resets the action potential in the bottom half of the domain (bottom row)

Computational Setup

1. *Basic parameters*

The snippet below shows basic parameters. The size of the domain is 128x128 (`nrows, ncols`). The time step is `dt`=0.05, and `dur` = 25000 step will provide sufficient information. The size of a cell Δ*h* = 2.0 (`h`). The value of the constants in the model can be changed (`a, b, e, d`), use these as starting values. Arrays for the potential (`v`), recovery (`r`), diffusion terms (`d`), perturbation/excitation (`iex`) have to be initialized.

```
ncols=128;                               % Number of columns in domain
nrows=128;                               % Number of rows in domain
dur=25000;                               % Number of time steps
h=2.0;                                   % Grid size
h2=h^2;
dt=0.05;                                 % Time step
Iex=35;                                  % Amplitude of external current
a=0.1; b=0.8; e=0.006; d=1.0;            % FHN model parameters
n=0;                                     % Counter for time loop
k=0;                                     % Counter for movie frames
done=0;                                  % Flag for while loop

v=zeros(nrows,ncols);                    % Initialize voltage array
r=v;                                     % Initialize refractoriness array
d=v;                                     % Initialize diffusion array
iex=zeros(nrows,ncols);                  % Set initial stim current and pattern
```

2. *Setting the perturbation*

There are two perturbations: one initially generates the action potential, which is applied on for the first `n1e` = 50 time steps, and the second sometime during the course of reaction (between `n2b` and `n2e` steps) to generate a second perturbation which will yield the wave with a free end. If the excitation is point-like (`StimProtocol==1`), the second perturbation should be off-center so that the closer part of the excitation to the previous excitation will die where leaving a crescent-shaped excitation with two free ends. If the first excitation is wave-like (`StimProtocol==2`), that is it is applied to the first 5 columns, the second step should be resetting *v* in some of the cells (e.g. the lower 30 rows) to zero until the excitation dies, leaving the excitation wave in the top 30 rows unchanged, but with a free end:

```
if StimProtocol==1
    iex(62:67,62:67)=Iex;                % If point-perturbation is chosen
else
    iex(:,1:5)=Iex;                      % If strip-perturbation is chosen

end
 n1e=50;                                  % Step at which to end 1st stimulus
switch StimProtocol
    case 1                               % Two-point stimulation
        n2b=3800;                        % Step at which to begin 2nd stimulus
        n2e=3900;                        % Step at which to end 2nd stimulus
    case 2                               % Cross-field stimulation
        n2b=5400;                        % Step at which to begin 2nd stimulus
        n2e=5580;                        % Step at which to end 2nd stimulus
end
```

3. *Setting up the display*

The snippet below will setup a 128x128 pixel size display with the potential showing as a function of time in the spatial domain (*x* and *y* directions):

```
ih=imagesc(v); set(ih,'cdatamapping','direct')
colormap(hot); axis image off; th=title('');
set(gcf,'position',[500 100 512 512],'color',[1 1 1],'menubar','none')

% Create 'Quit' pushbutton in figure window
uicontrol('units','normal','position',[.45 .02 .13 .07], ...
    'callback','set(gcf,''userdata'',1)',...
    'fontsize',10,'string','Quit');
```

4. *Setting up the time loop for integration*

The time loop will include the actual calculation at each time step.

a. *Checking if excitation has to be turned on or off:*

```
if n == n1e      % End 1st stimulus
    iex=zeros(nrows,ncols); % Resetting excitation to zero
end
if n == n2b      % Step to begin 2nd stimulus
    switch StimProtocol
        case 1
            iex(62:67,49:54)=Iex; % Excitation in off-center
        case 2
            v(60:end,:)=0; % Resetting bottom half of the domain to v=0
    end
end
if n == n2e      % End 2nd stimulus
   iex=zeros(nrows,ncols);
end
```

b. *Setting up the boundary conditions*

Setting the potential along the perimeter. For example in first column (`v(1,:)`) to the values in column 3 (`v(1,:)`).

```
v(1,:)=v(3,:);
v(end,:)=v(end-2,:);
v(:,1)=v(:,3);
v(:,end)=v(:,end-2);
```

c. *Calculating the diffusion terms*

Iterating through non-boundary cells to calculate the potential gradient. For each cell the 4 adjacent cells are considered.

```
for i=2:nrows-1
   for j=2:ncols-1
       d(i,j)=(v(i+1,j)+v(i-1,j)+v(i,j+1)+v(i,j-1)-4*v(i,j))/h2;
   end
end
```

d. *Calculating the changes to the potential and recovery values*

This can be done with matrix operations (instead of iterating through each cell) which makes the calculations much faster. The excitation is added (`iex`) which is a set value if excitation is on and zero otherwise. Also, the diffusion terms are added (stored in `d`):

```
dvdt=v.*(v-a).*(1-v)-r+iex+d;
v_new=v + dvdt*dt; % v can NOT be updated before r is calculated, so the
                   % calculated v is stored in v_new temporarily
drdt=e*(b*v-r);
```

e. Updating the potential and recovery values

```
r=r + drdt*dt;
v=v_new; clear v_new
```

f. Updating the display and saving movie frames

Use the snippet as is below to update the display and save every 500th display for the movie file

```
    m=1+round(63*v); m=max(m,1); m=min(m,64);

    % Update image and text
    set(ih,'cdata',m);
    set(th,'string',sprintf('%d  %0.2f   %0.2f',n,v(1,1),r(1,1)))
    drawnow

    % Write every 500th frame to movie
    if rem(n,500)==0
        k=k+1;
        mov(k)=getframe;
    end

    n=n+1;
    done=(n > dur);
    if max(v(:)) < 1.0e-4, done=1; end  %If activation extinguishes, quit early.
    if ~isempty(get(gcf,'userdata')), done=1; end % Quit if user clicks on
'Quit' button.
end
```

5. Saving the movie

Prompts you for a folder and a name for the movie. Use it as is.

```
sep='\';
[fn,pn]=uiputfile([pwd sep 'SpiralWaves.avi'],'Save movie as:');
if ischar(fn)
   % movie2avi(mov,[pn fn],'quality',75)
    movie2avi(mov, [pn fn], 'compression', 'none');
else
    disp('User pressed cancel')
end

close(gcf)
```

Results, Conclusions

1. Assemble the code to perform the modeling
2. Model both types of perturbation to demonstrate the nature of the free end of the potential wave.
3. Generate images from the movie to show your points.
4. Write up a short, “letter-type” lab report with the objective, method used, your findings, and conclusions.

5. Include the code that you used in your submission (in a separate .m file)

References

1. Yung, R.M.G.Y.L., *Atmospheric Radiation*, in *Atmospheric Radiation*. Oxford University Press. p. 88.
2. McQuarrie, D.A., *Second Virial Coefficient*, in *Statistical Mechanics*. p. 233-237.
3. Reif, F., *Fundamentals of statistical and thermal physics*. 2009, Waveland Press. p. 422-424.
4. Lemmon, E.W., M.O. McLinden, and D.G. Friend. *Thermophysical Properties of Fluid Systems, NIST Chemistry WebBook*. [cited 2014 02/23/2014]; Available from: http://webbook.nist.gov.
5. Clapham, D., *Calcium signaling.* Cell, 2007. **131**(6): p. 1047-1058.
6. Cuthbertson, K. and T. Chay, *Modelling receptor-controlled intracellular calcium oscillators.* Cell Calcium, 1991. **12**(2-3): p. 97.
7. Chay, T.R., Y.S. Lee, and Y.S. Fan, *Appearance of phase-locked wenckebach-like rhythms, devil's staircase and universality in intracellular calcium spikes in non-excitable cell models.* Journal of Theoretical Biology, 1995. **174**(1): p. 21-44.
8. Borghans, J.A.M., G. Dupont, and A. Goldbeter, *Complex intracellular calcium oscillations A theoretical exploration of possible mechanisms.* Biophysical Chemistry, 1997. **66**(1): p. 25-41.
9. Dupont, G., *Theoretical insights into the mechanism of spiral Ca2+ wave initiation in Xenopus oocytes*. 1998, Am Physiological Soc. p. 317-322.
10. Goldbeter, A., G. Dupont, and M.J. Berridge, *Minimal model for signal-induced Ca2+ oscillations and for their frequency encoding through protein phosphorylation.* Proceedings of the National Academy of Sciences of the United States of America, 1990. **87**(4): p. 1461.