Magnetic Susceptibility Under Phase Transitions

Tycho Sleator

Goals of the Experiment

In this experiment, a simple and inexpensive technique is used to measure the change of magnetic susceptibility of a nickel slug as it undergoes a phase transition from being ferromagnetic to only being paramagnetic. The temperature of this transition is also measured.

Introduction

The sample is placed in a coil, which is part of an LC oscillator. The effect of the sample is to change the inductance of the coil. The change in inductance produces a corresponding change in frequency of the LC oscillator, which is measured by a computer.

When the sample is placed in the coil the inductance becomes

\[ L = L_0(1 + \alpha \chi), \]  

where \( \chi \) is the magnetic susceptibility [1] of the sample and \( \alpha \) is a constant that depends on geometrical factors, such as the fraction of the coil that is occupied by the sample (filling factor) and the sample's demagnetization coefficient ([1]).

The oscillating frequency of the circuit is

\[ f = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{C/L_0(1 + \alpha \chi)}}. \]  

If \( \chi \) is small compared to unity, the operating frequency can be expanded as

\[ f \approx f_0(1 - \alpha \chi/2), \]  

where \( f_0 = 1/(2\pi\sqrt{L_0C}) \) is the frequency of the circuit with the coil empty.

When the magnetic sample undergoes its phase transition at temperature \( T_C \) the change in frequency is related to the change in magnetic susceptibility by

\[ \frac{(\Delta f)_T}{f_0} = -\frac{1}{2}(\alpha \Delta \chi)_T. \]  

By measuring the frequency change with and without the sample in the coil, one can determine the magnetic susceptibility of the material from Eq. (3). From Eq. (4) one obtains the change of susceptibility when the sample undergoes its transition.

Advance Preparation

1. Find the definitions of \( H \), \( B \), \( M \), and \( \chi \) (Most advanced undergraduate text books will address these ideas). Find the boundary conditions satisfied by \( B \) and \( H \).
2. Define a) paramagnetism, b) diamagnetism, and c) ferromagnetism.
3. What is the “demagnetization factor”. Find the demagnetization factors for a) a long thin rod with magnetization along the axis of the rod, b) a sphere, and c) a thin wide disk with magnetization normal to the plane of the disk.
4. What is the meaning of the “skin depth” \( \delta \) (you can look this up in most condensed matter textbooks, e.g. Ref [1])? Find an expression (e.g. from Ref [1]) for the skin depth for frequency \( \omega \) and electrical conductivity \( \sigma \).
**Apparatus**

**Overview**

In this experiment, the sample is either a nickel cylinder $1'' 	imes 1/4''$ (99.998% pure) or a bundle of nickel wires, each with diameter 0.125 mm (99.98% pure). It is placed in a closely fitted Pyrex tube around which 15 turns of wire is wrapped (the inductance of the coil is about $1\mu$H). The tube and lead wires are mounted on standoffs to an aluminum plate, which is centered in a furnace.

The type K (alumel/chromel) thermocouple is placed near the sample. The signal from the thermocouple is amplified so as to be on the order of several volts (no larger than 5 volts) at 400 °C.

The LC oscillator operates at around 1 MHz. Its output is sent through a comparator that feeds a 5 volt square wave into a counter on a computer interface card.

Both the thermocouple voltage and the oscillation frequency are read by an IBM PC with a Metrobyte "Dash 8" computer interface card containing an analogue-to-digital converter and a counter-timer chip. The frequency is measured by counting the number of cycles that occur in a given gate interval. At equal time intervals (of about 1 second) the gate is triggered and the amplified voltage from the thermocouple is measured so that the computer reads sets of data by itself as the sample is heated or cooled. The measured voltage and frequency are displayed on the computer screen. The configuration of the interface is shown in Figure 1.

**Principles of the Oscillator Circuit**

The oscillator circuit (shown in Fig. 2) is based on the "Colpitts" oscillator configuration [2]. The Colpits oscillator consists of an amplifier (the two transistors in the circuit of Fig. 2) with a feedback circuit made up of two series capacitors in parallel with an inductor. At the resonant frequency of the feedback circuit $\omega_0 = \sqrt{C_1 C_2 / (C_1 + C_2) L}$ the phase of the feedback becomes positive causing the circuit to oscillate at this frequency. The amplifier consists of an FET in a "common source" configuration (see Ref. [2]) followed by a npn transistor in an emitter follower configuration. The emitter follower serves the purpose of lowering the output impedance of the amplifier. The gain of the amplifier is controlled by setting the bias voltage of the FET drain through $R_7$, and has been adjusted to maximize the amplitude of the oscillations.

The output of the oscillator is fed to a comparator (configured with a small amount of positive feedback to prevent oscillations at the switching point), whose output switches between 0 and 5 volts depending on the signal from the thermocouple.

![Oscillator Circuit Diagram](image_url)
on whether the input is positive or negative. The comparator effectively acts as a 1-bit digital-to-analog converter (DAC). The comparator output is fed to the data-acquisition card.

Details of the Data Acquisition

The connections of the experiment to the data acquisition card are shown in Fig. 2. The system clock (at 4.7728 MHz) is used in combination with counter 2 and counter 1 to generate a "gate" signal with a 100 ms period (see figure). This gate signal is fed into the gate input of counter 0 (G0), as well as the digital input line “IP2”. Counter 0 is used to count the number of oscillations of the oscillator during one gate period. Counters (and therefore the gate period) are controlled by the software. The amplified signal from the thermocouple is fed into "in 0" of the analog-to-digital converter.

The software used to make the measurements is written in the C programming language and in assembly code. A listing of the software is given in Appendix A. At fixed time intervals, the computer reads the amplified thermocouple voltage and measures the frequency of the oscillator. These values are saved to a file.

Experimental Procedures

Calibration of the Thermocouple

To determine the temperature from a particular voltage from the thermocouple, the thermocouple first needs to be calibrated against a set of known temperatures. The temperature reading on the oven control is currently used for this. It is not known how accurate this thermometer is. One calibration point can be obtained by noting that before the oven is turned on, the sample is at room temperature. Calibrate the thermocouple during the first run (see next section) by periodically recording the voltage shown on the computer screen and the temperature reading on the oven.

Measurement of the Susceptibility of the Nickel Slug

First measure the frequency difference between the empty coil and the coil with the sample by running the program while the instructor adds and removes the sample from the coil.
Magnetic Susceptibility Under Phase Transitions

Have the instructor place the nickel slug into the coil. Begin recording data by starting the software. Turn on the oven. Continue recording until the oven reaches a temperature of about 450°C. After stopping the program you can copy the data to a different floppy disk so the data can be analyzed during the next run. Start the program again as the oven cools.

Finally, make a run with the empty coil over the same temperature range as the run with the sample.

Measurement of the Susceptibility of the Nickel Wires

Repeat the measurements made in the last section with the nickel wires instead of the slug (it is not necessary to do the run with the empty coil again). Before or after making the run, determine the mass of the Nickel sample.

Questions

The answers to these questions should be integrated into your lab report.

1. For each of the samples, plot the \( f(T)/f_0(T) \) vs. \( T \) for both increasing temperature and decreasing temperature. Give a qualitative description of the data, including any differences between the samples, and between the data for increasing temperature and decreasing temperature. Attempt to explain your observations. For example, explain why the frequency increases when the nickel slug is placed in the coil.

2. Determine the temperature of the phase transition, including the corresponding uncertainty of your measurement.

3. For the sample of Nickel wire, determine the absolute susceptibility of Nickel. To do this, you will need to estimate the filling factor of the sample. The filling factor can be estimated if you know the mass of the sample, the volume of the coil, and the fraction of the inductance attributable to the coil. The volume of the coil is about ..., and the inductance of the coil is about 1\( \mu \)H, whereas the total inductance of the circuit is about 1.6\( \mu \)H. You also need to consider the role of the demagnetization factor. Don’t forget to estimate the uncertainty of your result.

4. Suggest possible ways that the experiment could be improved.

References


Appendix A

!/--------------------------------------------------------------------------!

Experimental Physics, Tycho Sleator, Spring 1998

File "Nickel.c" Last Edit 1/2/98,5/7/98 by Tycho Sleator

This program is used to take data for the nickel magnetic susceptibility experiment. It was adapted from assembly code by Artur Solecki, Spring 1997 and revised by Tycho Sleator.

This program reads the temperature of the sample (thermocouple voltage) and the frequency of the oscillator circuit. Compiled with Turbo C++

Input to DASH 8 card:
1) Thermocouple voltage into analog input 0 ("In 0").
2) Frequency from comparator into "Clk 0".
Magnetic Susceptibility Under Phase Transitions

Output:
1) To screen - 4 columns:
   a) Thermocouple Voltage
   b) Approximate temperature
   c) # of counts in one gate period
   d) Calculated frequency
2) To file: Columns a) and d) of the screen output.

Command line arguments:
d   - delay in milliseconds between data points, default = 1000
name   - name of the output file, default = "out.txt"

Syntax:
nickeld (default values used)
nickeld d name (example: -> "nickel 2000")

#include <stdio.h>  
#include <conio.h>  
#include <dos.h>
#define basadr 300h  /* DASH8 base address */
#define control basadr+7 /* the counter control register address */
#define countr0 basadr+4 /* Counter 0, used as counter */
#define countr1 basadr+5 /* Counter 1, used as the second clock */
#define countr2 basadr+6 /* Counter 2, used as the first clock */
#define status basadr+2 /* status register contains IP2 bit */
#define adlow basadr
#define adhigh basadr+1
#define ecc 10000000b /* end of conversion bit of status register */
#define ip2 00010000b /* ip2 bit of status register. */

/* The following determine the frequency of the square wave used to generate the gate for measuring the oscillator frequency. The frequency of this square wave is 2.386MHz/(count2*count1). */
#define count2 23864 /* number of counts per period of clock 2 */
#define count1 10 /* number of counts per period of clock 1 */

int main(int argc, char *argv[])
{
    unsigned fnc(void);
    double temperature(void);
    void setup(void);

    unsigned count;
    unsigned contin, t_del;
    double freq;
    double temp, volt;
    FILE *ofp;

    if (argc == 3) {
        if ((ofp = fopen(argv[2], "u")) == NULL) {
            printf("Error opening output file.\n");
            exit(1);
        }
    } else
        ofp = fopen("out.txt", "w");

    if (argc < 2)
        t_del = 1000;
    else
        scanf(argv[1], "f", &t_del);

    setup();
contin = 1;
while(contin) {
    volt = temperature();
    temp = (400.0 /3.0) + volt + 24.0;
    count = fnc();
    freq = count * 2.0 * (2.3864/counts)/counts1;
    printf("V = %.4f\n\ntemp = %.1f\n\ntcount = %.6f\n\n\n", volt, temp, count, freq);
    if (fprintf(ofp, "%.4f %.8f\n", volt, freq) < 0) {
        printf("error writing to out.txt\n");
        close(ofp);
        exit(1);
    }
    asm {
        mov ah, 1
        int 16h
        jz nokey
        mov contin, 0
    }
    nokey: delay(t_del);
}
fclose(ofp);
return 0;
}

unsigned fnc() {
    unsigned cnt;
    asm cli;
    skip: asm {
        mov dx, status
            /* Get status register. */
        in al, dx
        test al, ip2
            /* Test IP2 bit of status register. */
        jnz skip
            /* Wait if it is high. */
        mov dx, countr0
            /* If it is low, reset Counter 0 */
        mov al, 0ffh
            /* to ffffh to prepare it for starting */
        out dx, al
            /* on the next hight gate */
        mov al, 0ffh
        out dx, al
    }
    high: asm {
        mov dx, status
            /* Get status register. */
        in al, dx
        test al, ip2
            /* Test IP2 bit of status register. */
        jz high
            /* Wait if it is still low. */
    }
    low: asm {
        /* If it is high */
        mov dx, status
        /^ get status register... */
        in al, dx
        test al, ip2
            /* ...and test IP2 bit. */
        jnz low
            /* Wait if it is still high. */
        mov dx, control
            /* If it is low, */
        mov al, 00000110b
            /* latch Counter 0. */
        out dx, al
        mov al, 00110000b
            /* Set the control register to */
        out dx, al
            /* read low/high byte. */
        mov dx, countr0
            /* Input the low byte from Counter 0. */
        in al, dx
        mov cl, al
            /* Move this byte to cl. */
Magnetic Susceptibility Under Phase Transitions

in al, dx  /* Input the high byte from Counter 0. */
mov ch, al  /* Move it to ch. */
mov ax, Offffh /* Put initial count (fffh) into ax. */
sub ax, cx    /* Compute the difference. */
mov cnt, ax
sti
}
return (cnt);
}

double temperature()
{
    double voltage;
    int ports;

    asm {
        mov al, 0       /* Initiate 12 bit A/D conversion */
        mov dx, adhigh  /* by writing to A/D high byte. */
        out dx, al      /* */
    }

    101: asm {
        mov dx, status  /* Get status register... */
        in al, dx       /* */
        test al, wec     /* ... and test EOC bit (end of conversion). */
        jnz 101          /* If conversion not done (EOC = 1), wait. */
        mov dx, adhigh   /* If done (EOC = 0), read in high byte. */
        in al, dx
        mov ah, al       /* Move this byte to ah. */
        mov dx, alow     /* Read in low byte */
        in al, dx        /* (into al). */
        mov cl, 4        /* logical shift 4 bits to the right. */
        shr ax, cl
        mov ports, ax
    }

    voltage = ((double) ports) * 10 / 4096 - 5.0;

    return (voltage);
}

void setup()
{
    asm {
        mov dx, control /* Set the control register to select */
        mov al, 0b6h     /* counter 2, read/load least/most sig- */
        out dx, al       /* nificent byte and output square wave. */
        mov dx, countr2  /* Load counter 2 with an even number */
        mov bx, counts2  /* 2.3864 MHz/counts1 = frequency of square wave. */
        mov al, bl       /* Load least significant byte. */
        out dx, al
        mov al, bh       /* Load most significant byte. */
        out dx, al
    }

    mov dx, control /* Set the control register to select */
    mov al, 76h      /* counter 1, read/load least/most sig- */
    out dx, al       /* nificent byte and output square wave. */
    mov dx, countri  /* Load counter 1 with an even number */
    mov bx, counts1  /* Fin/counts1 = frequency of square wave. */
    mov al, bl       /* Load least significant byte. */
    out dx, al
    mov al, bh       /* Load most significant byte. */
    out dx, al

    mov dx, control /* Select counter 0, mode 0 (pulse on */
    mov al, 00110000b /* terminal count), load/read low/high */
    out dx, al       /* and binary operation. */
    /* */