

Solution Manual: Steering paramagnetic Leidenfrost drops in an inhomogeneous magnetic field

Muhammad Qasim Khan and Muhamad Sabieh Anwar

Syed Babar Ali School of Science and Engineering

July 16, 2019

1 Velocity in Cartesian and Polar coordinates

$$\mathbf{r} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}}.$$

Taking the derivative with respect to time, we obtain the velocity,

$$\mathbf{v} = \dot{\mathbf{r}} = \dot{x}\hat{\mathbf{i}} + \dot{y}\hat{\mathbf{j}}.$$

The answer is slightly more involved when it comes to polar coordinates. This is because the basis vectors are a function of time. Here is our solution.

Any position vector can be written simply as $\mathbf{r} = r\hat{\mathbf{e}}_r$. The basis vectors can be written in terms of the Cartesian coordinates as,

$$\begin{aligned}\hat{\mathbf{e}}_r &= \cos\theta\hat{\mathbf{i}} + \sin\theta\hat{\mathbf{j}} \\ \hat{\mathbf{e}}_\theta &= -\sin\theta\hat{\mathbf{i}} + \cos\theta\hat{\mathbf{j}}.\end{aligned}$$

Since the cartesian basis is stationary, we can take the derivatives of the basis and see how they change with time. Taking the derivatives with respect to the coordinate theta we find that

$$\begin{aligned}\frac{d\hat{\mathbf{e}}_r}{d\theta} &= -\sin\theta\hat{\mathbf{i}} + \cos\theta\hat{\mathbf{j}} \equiv \hat{\mathbf{e}}_\theta \\ \frac{d\hat{\mathbf{e}}_\theta}{d\theta} &= -\cos\theta\hat{\mathbf{i}} - \sin\theta\hat{\mathbf{j}} \equiv -\hat{\mathbf{e}}_r.\end{aligned}$$

With these derivatives at hand, we can apply the chain rule and find out what the velocity in polar coordinates will be.

$$\mathbf{v} = \dot{\mathbf{r}} = \dot{r}\hat{\mathbf{e}}_r + r\dot{\hat{\mathbf{e}}}_r \tag{1}$$

$$\mathbf{v} = \dot{r}\hat{\mathbf{e}}_r + r\dot{\theta}\hat{\mathbf{e}}_\theta. \tag{2}$$

2 Magnetic energy profile through measurements of the magnetic field strength

The Tables (1) and (2) represent the data taken at certain values of r from the magnet's center with a setup of 4 and 1 magnets respectively using a gaussmeter. The gaussmeter is **LakeShore 410** gaussmeter and it was attached to a transverse probe to measure the component of the field parallel to the plane of the surface.

The setup we will consider in our analysis is the one represented by Table (1). The value of E_{mag} may be written in terms of the properties of liquid oxygen and the magnetic field as $E_{mag} = -\frac{\chi}{2\mu_0}|B|^2$, a fitted plot for this can be seen in Fig (1).

Sr #	$r(\text{mm})$	B(mT)
1	1	-33.7
2	2	-62.4
3	3	-82.5
4	4	-103.4
5	5	-156.5
6	6	-155.9
7	7	-141.1
8	8	-137.2
9	9	-136.2
10	10	-115.5
11	11	-116.3
12	12	-113.4
13	13	-92.5
14	14	-85.4
15	15	-66.7
16	16	-60.6
17	17	-56.9
18	18	-51.8
19	19	-43.5
20	20	-35.3
21	21	-38.2
22	22	-31.5
23	23	-29.1
24	24	-26.1
25	25	-24.1
26	26	-21.6
27	27	-18.8
28	28	-17.6

Table 1. The magnetic field profile for a 4 magnet setup.

Sr #	$r(\text{mm})$	B(mT)
1	1	-53.2
2	2	-68.0
3	3	-85.2
4	4	-89.3
5	5	-95.6
6	6	-91.5
7	7	-89.4
8	8	-87.3
9	9	-74.3
10	10	-69.2
11	11	-52.1
12	12	-42.7
13	13	-37.6
14	14	-30.5
15	15	-27.2
16	16	-21.4
17	17	-18.8

Table 2. The magnetic field profile for a **single** magnet setup.

3 Fitting the magnetic field strength to the magnetic energy profile

Using the MATLAB Curve Fitting app, `cftool`, on a “Custom Equation” setting, the equation can be provided and a fit be found. The parameters of the fit are shown in the corresponding table while the fit equation is,

$$f(r) = -\frac{E_0}{\left(q + \left(\frac{r}{r_0}\right)^6\right)}. \quad (3)$$

	Value	Units
E_0	0.2380	Jm^{-3}
q	0.0076	1
r_0	-0.0268	m

Table 3. Curve fitting for the data in Table (1), for 4 magnets, using the Equation (3).

	Value	Units
E_0	0.0032	Jm^{-3}
q	0.0002	1
r_0	-0.0380	m

Table 4. Curve fitting for the data in Table (1), for a single magnet, using the Equation (3).

The SSE for Table (3) is 66.89, R-square is 0.9782 and RMSE is 1.785.

The SSE for Table (4) is 3.038, R-square is 0.9885 and RMSE is 0.5512.

This model is a good fit for the data obtained as shown in Fig (1) and Fig (2), if the first four values in the table are discarded. This is done because it is not possible to accurately measure the magnetic field near the center of the magnets.

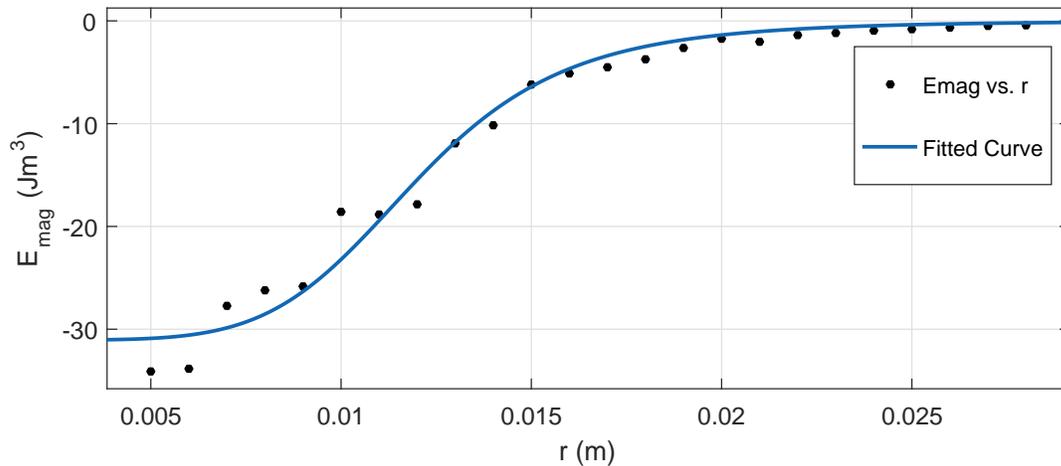


Fig. 1. This graph represents the fitted profile corresponding to Table (1), for 4 magnets. The dots show the values measured through a gaussmeter and the solid line is the best fit to the model used to profile the magnetic field. They are seen to be good fits.

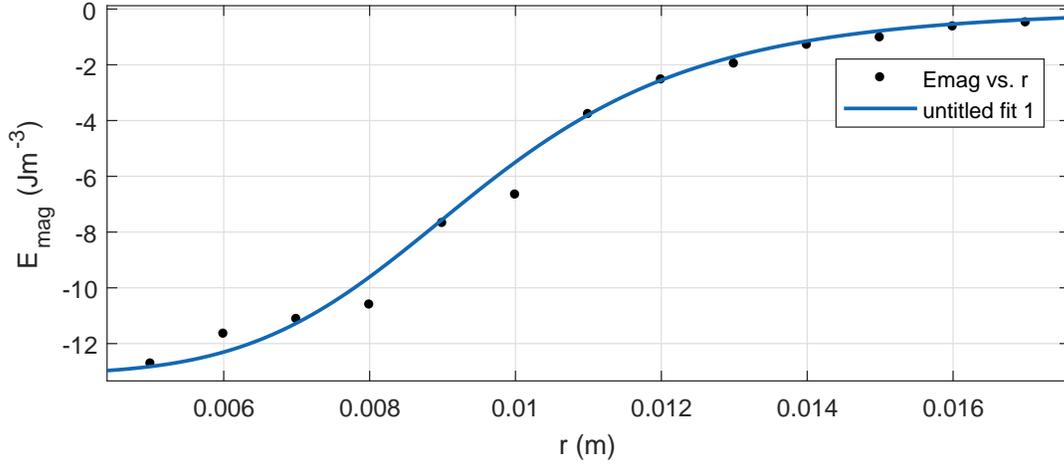


Fig. 2. This graph represents the fitted profile corresponding to Table (2), for a single magnet. The dots show the values measured through a gaussmeter and the solid line is the best fit to the model used to profile the magnetic field. They are seen to be good fits.

4 Tangential and radial components of energy

The total kinetic energy (per unit volume) of the drop is given by,

$$E_{total} = \frac{1}{2}\rho v^2.$$

We can write the velocity in polar coordinates as shown by Equation (2), this results in the expression

$$E_{total} = \frac{1}{2}\rho(\dot{r}^2 + r^2\dot{\theta}^2),$$

It allows us to separate out terms that depend on the angular velocity and the radial velocity. Hence the kinetic energy is seen as a composition of radial and tangential components.

$$E_{total} = E_{radial} + E_{tangential}.$$

Where the components are defined as,

$$E_{radial} = \frac{1}{2}\rho\dot{r}^2, \text{ and}$$

$$E_{tangential} = \frac{1}{2}\rho r^2\dot{\theta}^2.$$

5 Equations of motion for the drop in an inhomogeneous magnetic field

The particle starts out with energy

$$E_{initial} = \frac{1}{2}\rho v^2.$$

When it comes into the influence of the magnetic field, we include the term for the magnetic potential as well. This yields

$$\frac{1}{2}\rho v^2 = \frac{1}{2}\rho\dot{r}^2 + \frac{1}{2}\rho r^2\dot{\theta}^2 - \frac{E_0}{q + (\frac{r}{r_0})^6}. \quad (4)$$

6 Conservation of angular momentum

The conservation of angular momentum is best embodied in the Lagrangian formalism. If we write the Lagrangian for the system as $L = T - V$ with T as the final kinetic energy and V as the magnetic potential energy, the equation of angular motion is given by

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} = \frac{\partial L}{\partial \theta} \quad (5)$$

$$\frac{d}{dt} (r^2 \dot{\theta}) = 0 \quad (6)$$

$$(7)$$

$$r^2 \dot{\theta} = \text{constant} \quad (8)$$

Now it is possible to identify the term on the left as the angular momentum (per unit volume). If the initial angular momentum is bv then at any time, it will remain constant. Hence the constant can be set about a defined axis to bv and we can subsequently rearrange Equation (8) to obtain

$$\dot{\theta} = \frac{bv}{r^2}. \quad (9)$$

6.1 Equation of motion under the conservation of angular momentum.

Substituting Equation (9) into (4) and yields

$$\rho v^2 = \rho \dot{r}^2 + \rho \frac{b^2 v^2}{r^2} - \frac{2E_0}{q + (\frac{r}{r_0})^6}. \quad (10)$$

7 Theoretical prediction of the peripheral radius

Use the trajectories plotted in Tracker to obtain typical values of the initial b and v . Convert these values to meters and provide them to “finding_rp.m” file. This file will numerically evaluate and output all the possible values of r_p . The positive real roots of the equation are the theoretically predicted values of r_p . Below is the actual code used for theoretically predicting r_p .

```
function rpth = finding_rp(v,b)
    rho=1141; % density of liquid oxygen in kgm-3
    E0=0.08362;
    q=0.005678;
    r0=0.02607;
    syms rp;
    rpth = vpasolve(rho*v^2==rho*((b^2)*(v^2)/(rp^2))-2*E0/(q+(rp/r0)^6),rp);
end
```

8 Experimental determination of the value of the peripheral radius

Follow the instructions on in the student manual and store the trajectories of as many particles as possible in .csv format. There is no naming scheme that needs to be followed, the script will load all the .csv files. Then run MasterScript.m in the directory. This script will process the data and calculate the expected theoretical values of the physical constants and store it in a structure called data. It will show the trajectories and the radial velocities on a graph for each trajectory that was stored.

The struct data is an array of size equal to the number of trajectories available and it has 7 fields as follows:

1. **b**: The displacement from the x axis of the particle as it enters the frame.
2. **v**: The initial velocity with which it enters the frame.
3. **rpth**: The peripheral radius predicted theoretically from the equation of motion.
4. **rpexp**: The peripheral radius calculated from the recorded trajectory.
5. **defexp**: The experimentally evaluated angle of deflection, as the particle escapes the magnetic field, with respect to the initial trajectory.
6. **defth**: The theoretically evaluated angle of deflection.
7. **figure**: A handle to the figure corresponding to a particular particle.

The working of `MasterScript.m` is described in the subsections below..

8.1 Reading the files

The script searches the directory that it is in and imports data from all the files with a `.csv` extension. Matlab's default directory reading adds the parent directories as `'.'` and `'..'` hence there is a line of code that eliminates these objects from the structure that is formed.

This `Files` structure has a length equal to the number of trajectories and it has a field `Files.name` which can be used to read data.

8.2 Making an energy profile

Next the script requires that the user input vectors of the horizontal magnetic field and the radius from the center of the magnet at which it was measured.

It uses this data to fit the parameters in Equation (3). The parameters are available as global variables for further use.

8.3 Calculating `bavg`

A trajectory file is read as a table and provided as input to this function. With the coordinates available, `MasterScript.m` can find `bavg`. `bavg` is the displacement from the x-axis at which the particle enters the frame. This is taken to be the average of the first 5 y-coordinates that were recorded.

8.4 Calculating `vavg`

The variable `vavg` refers to the average velocity with which the particle enters the frame. To calculate the average initial velocity `vavg`, the `MasterScript.m` calls a subscript `findvel.m`. This subscript takes the first 12 data points and numerically evaluates the derivatives of position with respect to time to find the velocity. It uses the following equation to evaluate the derivative:

$$f'(x) \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}$$

Finally, it takes the mean of the velocities to output `vavg`.

8.5 Calculating the theoretical value of r_p

The `MasterScript.m` calls the script `finding_rp.m` the same way as in Section 7.

8.6 Calculating the experimental value of r_p

The `MasterScript.m` calls the subscript `tracking_rp.m`. This subscript takes as input the trajectory data. It creates a cubic spline interpolation of the data points to give $r(t)$ and evaluates the derivative. Then it looks for the value of $r(t)$ where the derivative is equal to 0. This is the peripheral radius which the subscript returns.

The subscript also produces plots of the radial coordinate as a function of time and the radial velocity as a function of time.

8.7 Experimental measurement of the angle of deflection

For this purpose, the `MasterScript.m` calls the subscript `finddefexp.m`. This subscript is a function that takes as input the coordinates of the particles. It takes the initial few points and fits a straight line onto them. Then it takes the last few points and fits another straight line to them. It then uses the gradients to find the angle with the x-axis, and takes the difference between the two angles with the x-axis to give the angle between the two lines. Here is the constitution of this script.

```
function angledef = finddefexp(coord)
    line1=fit(coord(1:6,1),coord(1:6,2),'poly1');
    line2=fit(coord(end-5:end,1),coord(end-5:end,2),'poly1');
    coeff1=coeffvalues(line1);
    coeff2=coeffvalues(line2);
    phi1=atand(coeff1(1));
    phi2=atand(coeff2(1));
    angledef=phi2-phi1;
end
```

8.8 Theoretical calculation of the angle of deflection

The angle of deflection is related to the total change in angle over the trajectory of the particle. We can write this simply as,

$$\Delta\theta = \int_{-\infty}^{+\infty} \dot{\theta} dt. \quad (11)$$

This θ is the angle from the polar coordinate system. We know that the angular momentum is conserved and we already derived an expression for $\dot{\theta}$ in Equation (9). Making the substitution results in,

$$\Delta\theta = \int_{-\infty}^{+\infty} \frac{bv}{r^2} dt. \quad (12)$$

Furthermore, the definition of $\dot{r}(t)$ allows us to write,

$$\dot{r} = \frac{dr}{dt}$$

$$dt = \frac{dr}{\dot{r}}.$$

Now we can substitute this into Equation (12) and we have an expression independent of time,

$$\int_{-\infty}^{+\infty} \frac{bv}{r^2} \frac{dr}{\dot{r}} \quad (13)$$

The equation of motion (Equation (10)) can be rearranged to obtain an expression for \dot{r} ,

$$\Delta\theta = \int_{-\infty}^{+\infty} \frac{b}{r^2 \sqrt{1 - \frac{2}{\rho v^2} E_{\text{eff}}}} dr$$

where, $E_{\text{eff}} \equiv \frac{\rho b v^2}{r^2} + E_{\text{mag}}$

The script `finddef.m` numerically evaluates the above integral, and then α is simply given by $\alpha = \Delta\theta - \pi$.

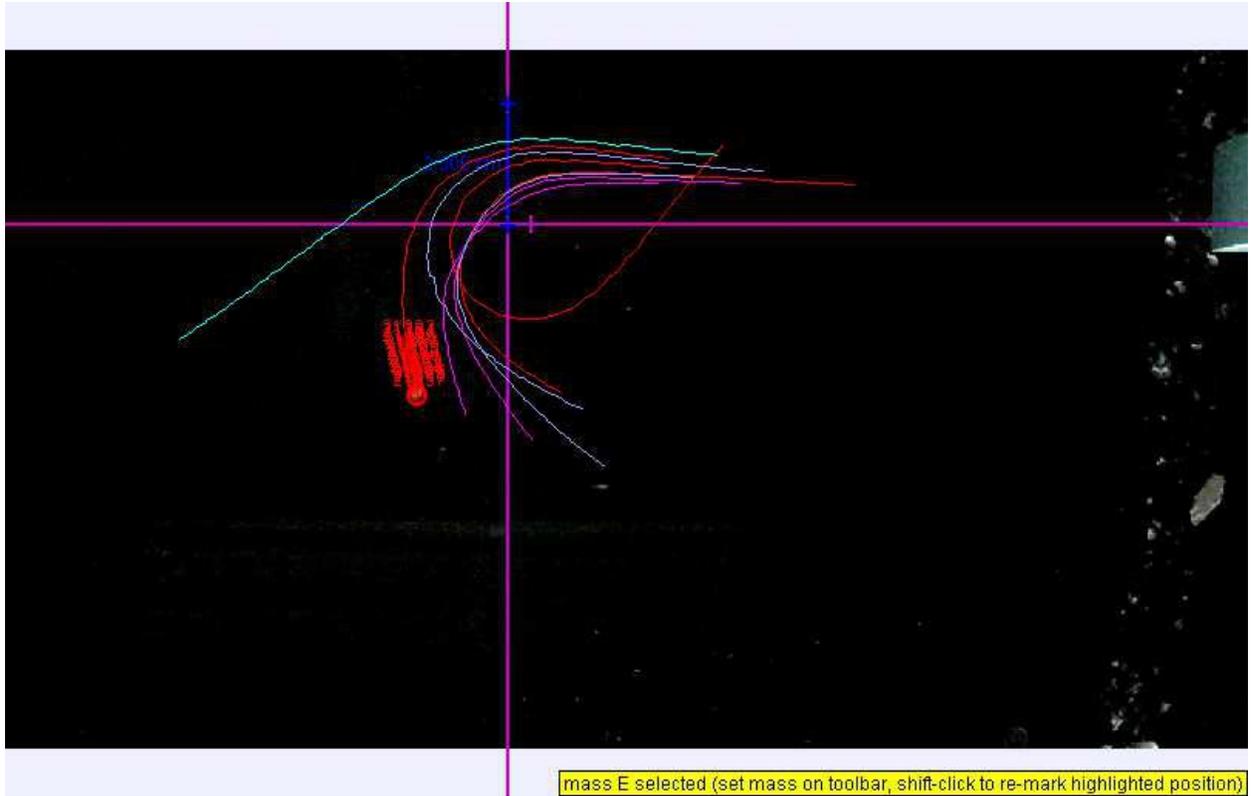


Fig. 3. This is how multiple trajectories will appear on the tracker. the coordinate axes are centered on the magnet and the calibration stick (in blue) provides a reference to the real life setup, allowing us to measure the coordinates accurately.



Fig. 4. The optimal setup looks like this. The light points downward to avoid reflections and glares. The plastic sheets are placed to reflect light from a range of angles. The ruler provides reference for calibration during tracking.

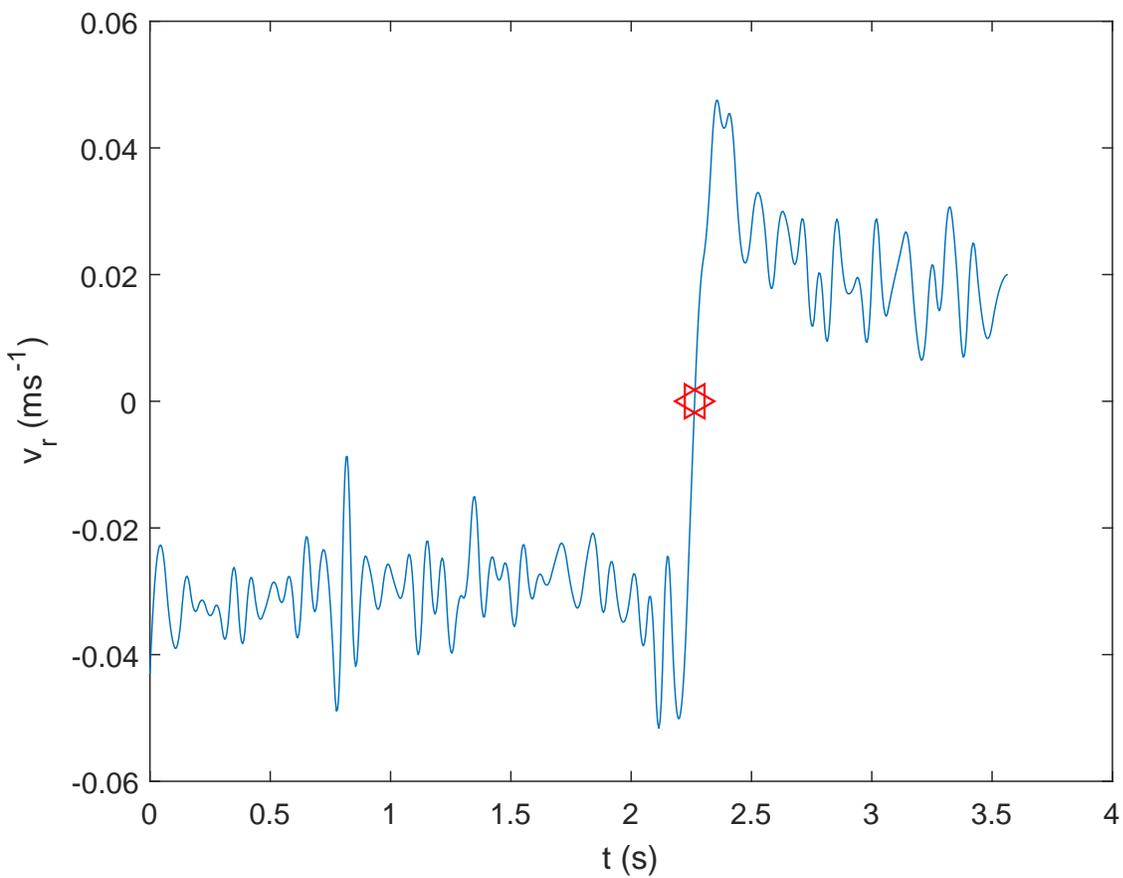


Fig. 5. This is a plot of the radial velocity for a particular drop. The peripheral radius is defined as the point where the radial velocity is zero. The zeros are marked and numerically displayed in MATLAB. The oscillations are due to the limitations of video tracking at a particular framerate. They would significantly reduce at higher framrates.