Practical 2A

Shahab Fatemi

May 2012

Practical Instructions

The practical should be conducted in pairs but discussions in larger groups are highly recommended.
1. Read through the text of the practical.
2. Try to find correct and reasonable answers for all the questions/tasks.
3. Write a report (one report per pair, shows the contribution of each explicitly) of not more than 6 pages (including figures) to present your results. Do not include any part of your code to the report except for the questions that we have asked you to do so.
This practical is supposed to be done in MATLAB, since the code has been written in MATLAB, but you can write your own code in any language. e.g. C/C++, Fortran, Python, ...

Deadline to hand in your report is: **Friday 25 May, before 16:00.**

You should not upload your report on Fronter. Please submit your solutions to my email address (*shahab@irf.se*).

Part I - Particle Motion and Algorithms for Trajectory Computations

The dynamics of a single charged particle was studied in the previous lectures and it was explained that the motion of a charged particle is defined by the equation of motion. Particle motion in the existence of electromagnetic field was defined and its motion, acceleration and the different drift velocities were discussed. We have already seen that solution to the equation of motion can not be obtained easily, especially when time varying fields and external forces are considered. Therefore, numerical methods are used to solve the equations and of course the numerical methods have some errors and the results are not exactly the same as the theoretical solutions. In addition, it takes time for different computational algorithms to be solved which means that not only the method we choose to solve our problem should have the highest accuracy or the lowest calculation errors, but also it should run fast enough. We conclude that finding a method to solve our problem is not the only issue, but choosing the best method which is more suitable for our problem and gives us the more accurate results are essential. The study of different numerical methods which can solve the mathematical equations is called Numerical Analysis. Generally speaking, finding a suitable numerical method to solve a problem always depends on the problem conditions and our expectations of the solutions we get at the end. Hence, before start studying different numerical methods, we should understand the problem we want to solve and also know all the expectations we have by solving that problem. [4]
The equation of motion is an Ordinary Differential Equation (ODE). The equation of motion for a charged particle and the initial position and velocity conditions are:

$$\frac{d\mathbf{v}(t)}{dt} = \frac{q}{m}(\mathbf{E}(t) + \mathbf{v}(t) \times \mathbf{B}(t)) \quad ; \quad \frac{d\mathbf{r}(t)}{dt} = \mathbf{v}(t) \tag{1}$$

$$\textit{Initial conditions:} \quad \mathbf{r}(t=0) = \mathbf{r}_{init} \quad , \quad \mathbf{v}(t=0) = \mathbf{v}_{init}$$

To solve this equation numerically, a numerical method is needed which solves the first derivative equations. The derivative is defined by:

$$\frac{dg(t)}{dt} = \lim_{\Delta t \to 0} \frac{g(t+\Delta t) - g(t)}{\Delta t} \tag{2}$$

where g is a function.
The Taylor series is used to expand g(t+Δt) and if the first derivative function is taken and the higher derivation

terms are neglected, the resulting approximation error which is called truncation error and is specified as the order of Δt shown with $\mathcal{O}(\Delta t^a)$, where 'a' is the error term. Therefore equation 2 is written as:

$$\frac{dg(t)}{dt} = \frac{g(t+\Delta t) - g(t)}{\Delta t} + \mathcal{O}(\Delta t^a) \tag{3}$$

Several numerical methods exist to solve equation 3. Here, two of them are introduced and compared briefly, and finally the one which is more suitable for our application is chosen.

1- Euler Method

In the Euler method, equation 3 is applied to equation 1 and it gives us:

$$\begin{cases} \mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \{\frac{q}{m}(\mathbf{E}(t) + \mathbf{v}(t) \times \mathbf{B}(t))\}\Delta t \\ \\ \mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t \end{cases} \tag{4}$$

and if the solution is calculated for n time steps, the Euler method would be:

$$\begin{cases} \mathbf{v}_{n+1} = \mathbf{v}_n + \{\frac{q}{m}(\mathbf{E}_n + \mathbf{v}_n \times \mathbf{B}_n)\}\Delta t \\ \\ \mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n\Delta t \end{cases} \tag{5}$$

when the different time steps are defined as t_{n+1}=t_n + Δt and Δt is the time step length [2].
The Euler method has the truncation error as the order of 2 (a=2) which is numerically unstable except for the extremely small step lengths. Taking very small step lengths provides another error which occurs due to the computational limitations and is called round-off error. Therefore, this method is not used very often.

2- Leap-Frog Method

The Leap-Frog method is used when only modest accuracy is needed. In this method, the time steps for positions (**r**) are defined for t_k, t_{k+1}, t_{k+2}, ..., t_n where k is an arbitrary integer number between 0 and n, while the velocities are calculated for the time steps halfaway in between like $t_{k+\frac{1}{2}}$, $t_{k+\frac{3}{2}}$, ...,$t_{n+\frac{1}{2}}$ and that is why the method is called the Leap-Frog. If this method is used to solve equation 1, the solution will be:

$$\begin{cases} \mathbf{r}_n = \mathbf{r}_{n-1} + \mathbf{v}_{n-\frac{1}{2}}\Delta t \\ \\ \mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n-\frac{1}{2}} + \{\frac{q}{m}(\mathbf{E}_{n-\frac{1}{2}} + \mathbf{v}_{n-\frac{1}{2}} \times \mathbf{B}_{n-\frac{1}{2}})\}\Delta t \end{cases} \tag{6}$$

The error in the Leap-Frog method is $\mathcal{O}(\Delta t^3)$ which is lower than the Euler method. The problem is that it is difficult to get started [2]. When the initial conditions are defined only for $t = 0$, the conditions for the first half-steps are unknown. This is the main problem of the Leap-Frog method. The advantages of this method are the energy conservation and time reversibility (it can be run both forward and backward in time). It means that a negative or positive value can be chosen for Δt.
This method is applied to the Lorentz force and the velocity is calculated. The final simplified velocity formula is:

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n-\frac{1}{2}}(1 - \frac{1}{2}{\omega_c}^2\Delta t^2) + \frac{q\Delta t}{m}(\mathbf{E} + \mathbf{v}_{n-\frac{1}{2}} \times \mathbf{B}) + \frac{q^2\Delta t^2}{2m^2}(\mathbf{E} \times \mathbf{B}) + \frac{q^2\Delta t^2}{2m^2}(\mathbf{v}_{n-\frac{1}{2}} \cdot \mathbf{B})\mathbf{B} \tag{7}$$

where ω_c is the cyclotron frequency [3].

★ Which method is the best among the explained methods to solve the equation of motion? What are the advantages and disadvantages?

---

1. Download *practical2A.zip* file from Fronter, and extract it in your local computer.
2. Read 'Particle.m' code carefully and try to understand it.
3. Assume the solar wind velocity is u_{sw}=$-400\widehat{x}$ km/s and the interplanetary magnetic field (IMF) is B_{sw}=$5\widehat{y}$ nT. Calculate the convective electric field.
4. Launch a proton in the solar wind condition mentioned earlier with initial velocity $v_{init} = +200\widehat{x}$ km/s from the center of the coordinate system. Choose the time step Δt=0.05 sec and let the proton move for t_{max}=50 seconds. How does the trajectory, velocity and the energy of the particle change during this 50 seconds? (the small circles in trajectory and velocity plots denote the initial values).
5. Estimate the proton's gyro-frequency from the plots.
6. Change the proton's initial velocity to $-200\widehat{x}$ km/s. What are the major differences compared to the previous conditions?
7. Set the proton's initial velocity the same as the solar wind velocity. What do you observe? What are the effects of electric and magnetic fields in the particle's motion?
8. Set the solar wind velocity to zero, proton's initial velocity to $+200\widehat{x}$ km/s and run the code again. Which quantity can we measure from the particle's trajectory? Compare the value you measure with the theoretical formula.
9. Assume the solar wind velocity is u_{sw}=$-400\widehat{x}$ km/s and launch a Helium ion with $m = 2$ and $v_{init} = +200\widehat{x}$ km/s. What do you observe? Discuss your findings and compare them with protons at the same conditions.
10. Let the helium ion moves for 200 seconds. At half of its way, it feels abrupt changes in the solar wind velocity and the magnetic field. Assume the magnetic field changes to $+15\widehat{y}$ nT and the solar wind velocity changes to $-200\widehat{x}$ km/s. How do the trajectory, velocity and energy change? Explain your observations.

Part II - Kinetic Theory

1. Write a function that gets mass (m), density (n), temperature(T), bulk velocity (u) and particle's velocity (v) and calculates the Maxwellian distribution function (f_{sM}). **[0.5p]**

```
function f = maxwellian(m, n, T, u, v)
    kb = 1.38066e-23; % Boltzmann constant

    %%%%%% WRITE YOUR CODE HERE %%%%%%%%%
    f =

end
```

2. Assume that the solar wind protons number density (n) is 3×10^6 m^{-3}, the temperature (T) is 10^6 K and the bulk velocity (u) is 3×10^5 m/s. Bin the velocity space in one-dimension (e.g. along x) equally from 0 to 6×10^5 m/s with velocity step Δv=5×10^3 m/s. Put a proton at each binned velocity and find the proton distribution function. Plot the distribution function and interpret your observations. **[0.5p]**
MATLAB Hints:

```
%-----------------------------------------------
% Bin the velocity space:
v = vmin : dv : vmax;

%-----------------------------------------------
% Calculate distribution function
f = zeros(length(v), 1); % Pre-allocate memory for all f values.

% It would be the best to avoid for-loops (use vectorized technique)
for i=1:length(v)
    f(i) =  maxwellian(m, n, T, u, v(i) )
end
```

```
%--------------------------------------------
% Plot the results
figure;
subplot(2,1,1);
plot(v, f);
% plot in log-scale
subplot(2,1,2);
semilogy(v, f);   % equivalent to plot(v, log10(f) );
```

3. Run your code for different velocity steps (e.g. Δv=2 × 10^4 m/s, Δv=6 × 10^4 m/s and Δv=9 × 10^4 m/s) and discuss the results. **[0.5p]**
4. Now we want to test 3D velocity space.
   Open *vsd.m* file and try to understand the code and run it.
   Assume non-drifting solar wind protons have the density of $8 \times 10^6\ m^{-3}$ and the temperature of $10^5\ ^\circ K$. Bin the velocity space from -200 to 200 km/s with $\Delta v = 10$ km/s and find the 3D Maxwellian velocity space distribution.
5. We already know that:

$$n = \int_{-\infty}^{+\infty} f_{sM}(\mathbf{r}, \mathbf{v}, t) d^3\mathbf{v} \equiv \sum_{v_i, v_j, v_k} f(v_i, v_j, v_k) \Delta v_i \Delta v_j \Delta v_k \tag{8}$$

   How much is the calculated number density (use equation 8)? Compare it with the solar wind number density. **[0.5p]**
6. Change the solar wind number density to n=5 × 10^6 m^{-3}. Run the code and discuss the results. **[0.5p]**
7. Plot the distribution function in the $v_x - v_y$ plane, using MATLAB *surf* command. **[0.5p]**

```
% MATLAB surf command.
% The command below is just an example. Change it appropriately!
surf(v_x, v_y, f(:,:,1) );
```

8. Plot the distribution function in the $v_x - v_y$ plane and compare it with distribution function in the $v_y - v_z$ plane, using MATLAB *contour* command. **[0.5p]**

```
% MATLAB contour command.
% The command below is just an example. Change it appropriately!
contour(v_x, v_y, f(:,:,1) );
```

9. Calculate the solar wind protons number density for T=5 × 10^5 K and T=20 × 10^5 K. What are the densities you get and why are they different? How can we solve this problem? **[0.5p]**
10. **Do it by your own!** Sign up for your group at: *http://www.doodle.com/iqs2v7xrmtv2i63s*
11. Download your group '.mat' file (e.g. group N should download the file 'groupN.mat') from:
    ftp://ftp.irf.se/pub/tmp/outgoing/shahab/

    Each file corresponds the solar wind protons velocity space distributions obtained from a particle simulation code for an observer located at ∼1.0 AU distance from the Sun. The simulated observer collects only the solar wind protons with directional velocity range from 0 to 8 × 10^5 m/s and velocity resolution of 5 × 10^3 m/s in all directions. The magnetic field is constant (+5 $\widehat{y}$ nT), and the solar wind flows along -x axis.

    (a) Load the file into MATLAB. (the velocity space distribution values are stored in 'f').

    ```
    load('groupN.mat');
    ```

    (b) Calculate the solar wind protons number density. **[0.5p]**
    (c) Calculate the solar wind protons fluxes in all directions and find the solar wind bulk flow velocity. (Attach your MATLAB code to the report) **[1p]**
    (d) Plot the distribution function in the $v_x - v_y$ plane and compare it with distribution function in the $v_x - v_z$ plane. The center of the distribution contours should clearly show the bulk velocity in each plane. **[1p]**
    (e) What is the type of the distribution function, Maxwellian or Bi-Maxwellian? Why? **[0.5p]**
    (f) Calculate the pressure tensor and scalar pressure. (Attach your MATLAB code to the report)**[1.5p]**
    (g) Find the solar wind thermal velocity and the solar wind kinetic temperature in all directions. **[1.5p]**

Total: [10p]

Appendix

Maxwellian distribution function:

$$f_{sM}(\mathbf{r},\mathbf{v},t) = n_s(\frac{m_s}{2\pi k_b T_s})^{3/2} \quad exp(-\frac{m_s(\mathbf{v}-\mathbf{u})^2}{2k_b T_s}) \tag{9}$$

Bi-Maxwellian distribution function:

$$f_{sBM}(\mathbf{r},\mathbf{v},t) = n_s(\frac{m_s}{2\pi k_b T_{s||}})^{1/2}(\frac{m_s}{2\pi k_b T_{s\perp}}) \quad exp[-\frac{m_s(\mathbf{v}_{||}-\mathbf{u}_{||})^2}{2k_b T_{s||}} - \frac{m_s(\mathbf{v}_{\perp}-\mathbf{u}_{\perp})^2}{2k_b T_{s\perp}}] \tag{10}$$

where the total kinetic temperature is:

$$T_s = \frac{T_{s||} + 2T_{s\perp}}{3} \tag{11}$$

Zeroth order moment (density):

$$n_s(\mathbf{r},t) = \int_{-\infty}^{+\infty} f_s(\mathbf{r},\mathbf{v},t)d^3\mathbf{v} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_s(\mathbf{r},\mathbf{v},t)dv_x dv_y dv_z \tag{12}$$

First order moment (flux):

$$\Gamma_s(\mathbf{r},t) = \int_{-\infty}^{+\infty} \mathbf{v}\ f_s(\mathbf{r},\mathbf{v},t)d^3\mathbf{v} \tag{13}$$

Bulk flow velocity:

$$\mathbf{u}(\mathbf{r},t) = \Gamma_s(\mathbf{r},t)/n_s(\mathbf{r},t) \tag{14}$$

Second order moment (pressure):

$$\widetilde{P}_s(\mathbf{r},t) = m_s \int_{-\infty}^{+\infty} \mathbf{c}\cdot\mathbf{c}\ f_s(\mathbf{r},\mathbf{v},t)d^3\mathbf{v} \tag{15}$$

where $\mathbf{c} = \mathbf{v}$ - $\mathbf{u}$, $\widetilde{P}_s$ is the pressure tensor, and contains a 3×3 matrix:

$$P_{sij}(\mathbf{r},t) = m_s \int_{-\infty}^{+\infty} c_i c_j\ f_s(\mathbf{r},\mathbf{v},t)d^3\mathbf{v} \tag{16}$$

and the scalar pressure p_s is defined as one third of the trace of P_{ij}.

$$p_s = \frac{1}{3}Tr\widetilde{P}_s(\mathbf{r},t) \tag{17}$$

Third order moment (heat flux):

$$\widetilde{Q}_s(\mathbf{r},t) = \frac{1}{2}m_s \int_{-\infty}^{+\infty} \mathbf{c}\ c^2\ f_s(\mathbf{r},\mathbf{v},t)d^3\mathbf{v} \tag{18}$$

References

[1] Cravens T. E., *Physics of Solar System Plasma*, Cambridge press, 2004

[2] Garcia L. A., *Numerical Methods For Physics*, Prentice Hall, 2nd edition, 2000.

[3] Ledvina S. A., Y. J. Ma, E. Kallio, *Modeling and Simulating Flowing Plasma and Related Phenomena*, Springer Science and Business Media B.V. 2008 - Space Science Rev (2008) 139: 143-189, March 2008.b

[4] Stoer J., R. Bulirsch, *An Introduction To Numerical Analysis*, Springer, 3rd edition, 2002.