# Mechanics,

## the source of it all

## I

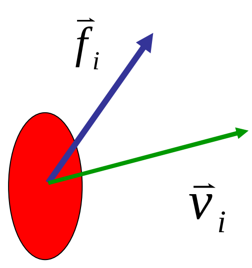
Motion of particles and rigid bodies: introduction to the computer solution of differential equations

Newton: F = m a

$$m \frac{d^2 \vec{r}_i}{dt^2} = \vec{f}_i$$

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i$$

$$\frac{d\vec{v}_i}{dt} = \frac{\vec{f}_i}{m}$$




A system of two coupled vector differential equations. Here the force depends on the position of the particle (and possibly on time). For the time being we ignore effects due to the finite size of the object.

## Example 1:

phase plane for the simple harmonic oscillator

$$m\frac{d^2x}{dt^2} = f(x) \approx -kx$$

Hooke's law

$$k/m = 1$$

$$\frac{d^2x}{dt^2} + x = 0 \Rightarrow \quad \frac{dx}{dt} = y, \quad \frac{dy}{dt} = -x$$

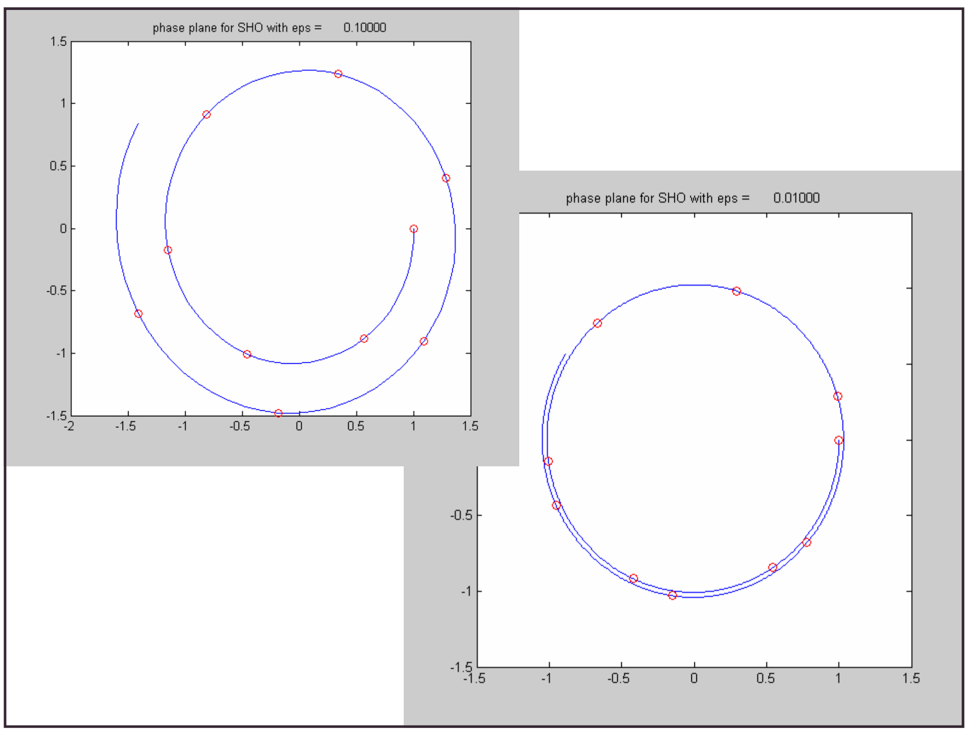
## Numerical scheme 1: 1st order forward Euler's method

$$x(0) = 1$$
$$y(0) = 0$$

$$\varepsilon = 0.1$$

$$x(t+\varepsilon) = x(t) + \varepsilon y(t)$$
$$y(t+\varepsilon) = y(t) - \varepsilon x(t)$$

```
function spring1(eps,N)
% integrates the SHO via 1st order
% forward Euler's method
x(1,1)=1;x(1,2)=0;
for i = 1:N
    x(i+1,1) = x(i,1) + eps*x(i,2);
    x(i+1,2) = x(i,2) - eps*x(i,1);
end
figure; axis([-2, 2 , -2, 2])
plot(x(:,1),x(:,2),x(1:N/10:N,1),x(1:N/10:N,2),'ro')
title(sprintf('phase plane for SHO with eps =…
 %12.5f',eps));
```



## Numerical scheme 2: 1st order backward Euler's method

$$\varepsilon = 0.1$$

$$x(0) = 1$$

$$y(0) = 0$$

$$x(t+\varepsilon) = x(t) + \varepsilon y(t+\varepsilon)$$

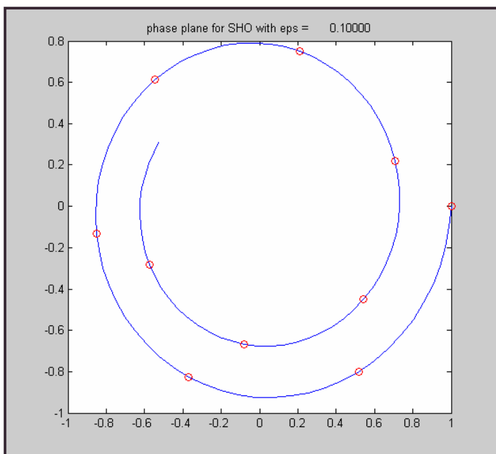
$$y(t+\varepsilon) = y(t) - \varepsilon x(t+\varepsilon)$$

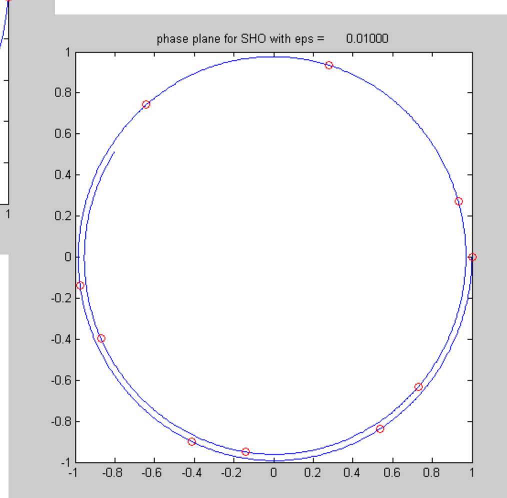
$$\begin{pmatrix} 1 & -\varepsilon \\ \varepsilon & 1 \end{pmatrix} \begin{pmatrix} x(t+\varepsilon) \\ y(t+\varepsilon) \end{pmatrix} = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

$$\begin{pmatrix} x(t+\varepsilon) \\ y(t+\varepsilon) \end{pmatrix} = \frac{1}{1+\varepsilon^2} \begin{pmatrix} 1 & \varepsilon \\ -\varepsilon & 1 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

This is an example of an implicit method: it uses information on the future values of the derivatives to integrate the equations.

```
function spring2(eps,N)
% integrates the SHO via 1st order
% backwards Euler's method
x(1,1)=1;x(1,2)=0; ep2 = 1/(1+eps^2);
for i = 1:N
    x(i+1,1) = ep2*(x(i,1) + eps*x(i,2));
    x(i+1,2) = ep2*(x(i,2) - eps*x(i,1));
end
figure; axis([-2, 2 , -2, 2])
plot(x(:,1),x(:,2),x(1:N/10:N,1),x(1:N/10:N,2),'ro')
title(sprintf('phase plane for SHO with eps =…
%12.5f',eps));
```



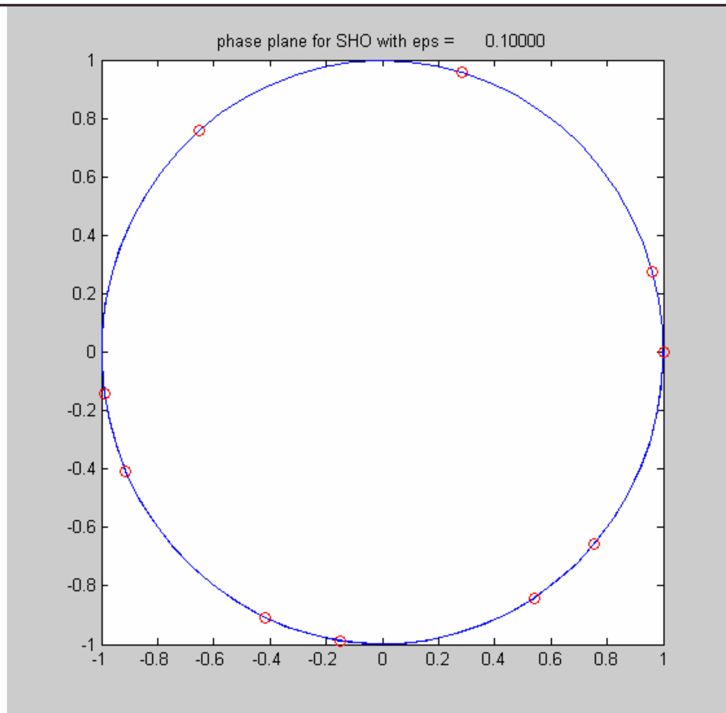
## Numerical scheme 3: 2nd order leap-frog scheme

$$x(0) = 1 \qquad \varepsilon = 0.1$$

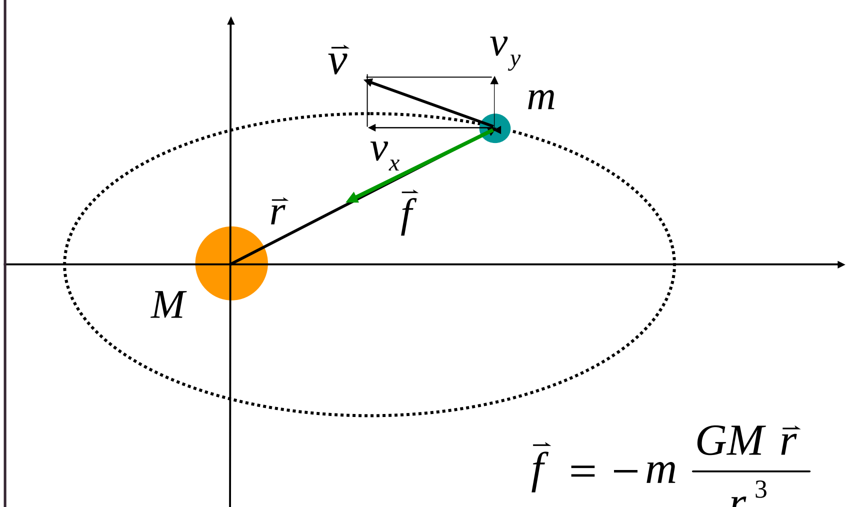
$$y(0) = 0 \qquad y(\varepsilon/2) = y(0) - (\varepsilon/2)x(0)$$

$$x(t+\varepsilon) = x(t) + \varepsilon y(t+\varepsilon/2)$$
$$y(t+\varepsilon/2) = y(t-\varepsilon/2) - \varepsilon x(t)$$

```
function spring3(eps,N)
% integrates the SHO via 2nd order Leap-frog
x(1,1) = 1; x20 = 0;
x(1,2) = x20 - eps*x(1,1)/2;
y(1) = 0;
for i = 1:N
   x(i+1,1) = x(i,1) + eps*x(i,2);
   x(i+1,2) = x(i,2) - eps*x(i+1,1);
   y(i+1)   = (x(i+1,2) + x(i,2))/2;
end
figure; axis([-2, 2 , -2, 2])
plot(x(:,1),y(:),x(1:N/10:N,1),y(1:N/10:N),'ro')
title(sprintf('phase plane for SHO with eps =…
%12.5f',eps));
```




## Example 2: a planet moving around the sun




$$\vec{f} = -m \frac{GM\,\vec{r}}{r^3}$$

Newton's law of gravity

The equations of motion in component form

$$\frac{dx}{dt} = v_x$$

$$\frac{dy}{dt} = v_y$$

$$\frac{dv_x}{dt} = \frac{f_x}{m} = -GM \frac{x}{r^3}$$

$$\frac{dv_y}{dt} = \frac{f_y}{m} = -GM \frac{y}{r^3}$$

Numerical integration of the equations of motion

$$GM = 1$$

$$\varepsilon = 0.1$$

$$\vec{a}(t) := \frac{\vec{f}(t)}{m}$$

$$x(0) = 1 \qquad \Rightarrow r(0) = 1$$

$$y(0) = 0 \qquad 1/r^3(0) = 1$$

$$v_x(0) = 0$$

$$v_y(0) = 1$$

$$\vec{r}(t+\varepsilon) = \vec{r}(t) + \varepsilon\vec{v}(t+\varepsilon/2)$$

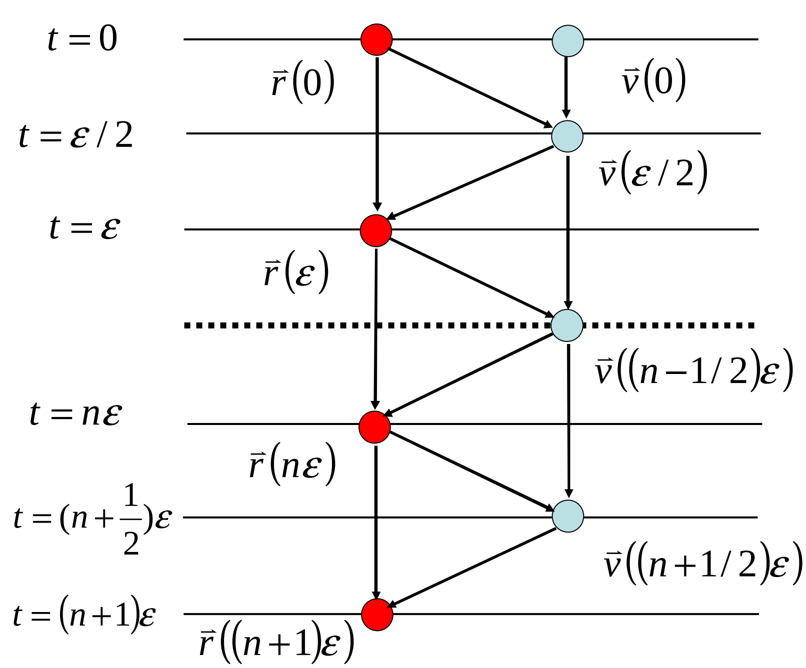
$$\vec{v}(t+\varepsilon/2) = \vec{v}(t-\varepsilon/2) + \varepsilon\vec{a}(t)$$

Where at t=0 we use: $\vec{v}(\varepsilon/2) = \vec{v}(0) + \frac{\varepsilon}{2}\vec{a}(0)$

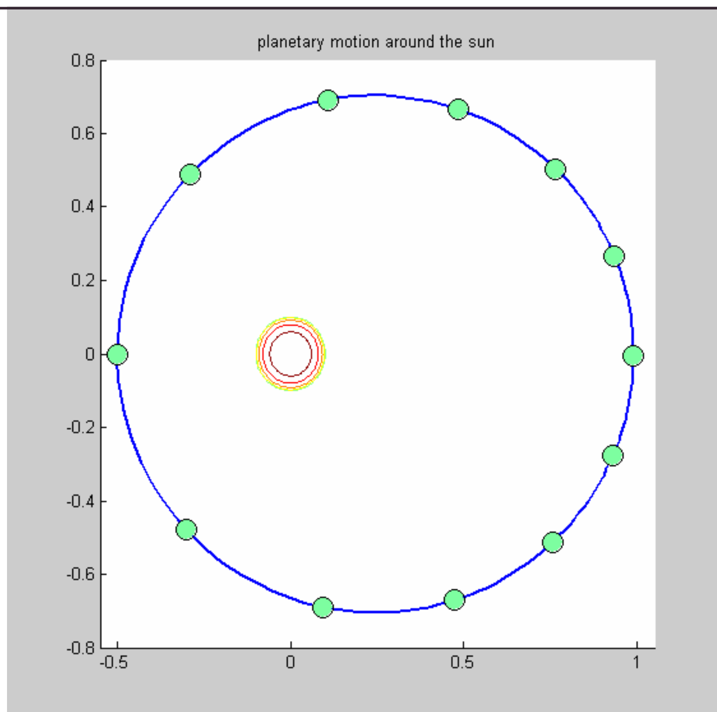
$$x(t+\varepsilon)=x(t)+\varepsilon v_x(t+\varepsilon/2)$$
$$v_x(t+\varepsilon/2)=v_x(t-\varepsilon/2)+\varepsilon a_x(t)$$
$$a_x(t)=-\frac{x(t)}{\left(x(t)^2+y(t)^2\right)^{3/2}}$$

Scheme:
Leap-frog

$$y(t+\varepsilon)=y(t)+\varepsilon v_y(t+\varepsilon/2)$$
$$v_y(t+\varepsilon/2)=v_y(t-\varepsilon/2)+\varepsilon a_y(t)$$
$$a_y(t)=-\frac{y(t)}{\left(x(t)^2+y(t)^2\right)^{3/2}}$$



```
function planet(eps,N)
% integrate equations of planetary motion - Leap-Frog
figure; axis([-.55, 1.05 , -.8, .8]); hold on
[a,b,c]=sphere(100); contour(.05*a,.05*b,.05*c)
r(1,1) = -.5;      r(1,2)=0;        v0=[0,-1.63];
v(1,:)  =  v0 + eps/2 * accel(r(1,1),r(1,2));
for i=1:N
  r(i+1,:)  = r(i,:)  + eps * v(i,:);
  v(i+1,:) = v(i,:)  + eps * accel(r(i+1,1),r(i+1,2));
end
plot(r(:,1),r(:,2),'-b','LineWidth',2)
plot(r(1:14:N,1),r(1:14:N,2),'ro','MarkerEdgeColor','k',...
     'MarkerFaceColor',[.49 1 .63],'MarkerSize',12)
title('planetary motion around the sun';);hold off
function a = accel(x,y)
  r3   =   (x^2+y^2)^(3/2);
  a(1,1)  =  -x/r3;   a(1,2)  =  -y/r3;
```



# The Levitron

A magnetized spinning top can
be made to hover. Similar magnetic poles repel each other, but one cannot simply hang a magnet above another magnet, because the arrangement is unstable. The slightest perturbation will disturb the delicate balance and the suspended magnet will fall.

A spinning magnetic dipole in a magnetic field:

$$\frac{d\vec{\mu}}{dt} = \frac{\mu}{I\omega}\vec{\mu}\times\vec{B}$$

$$m\frac{d^2\vec{r}}{dt^2} = \nabla\left(\vec{\mu}\cdot\vec{B}\right) - mg\vec{z}$$

$$B_z = B_0 + Sz + Kz^2 - \frac{1}{2}Kr^2 + \cdots$$

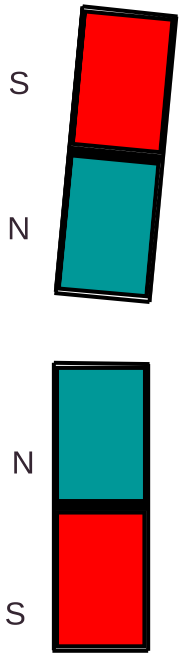
$$B_r = -\frac{1}{2}Sr - Krz + \cdots$$

Balance between Magnetic and Gravitational force:

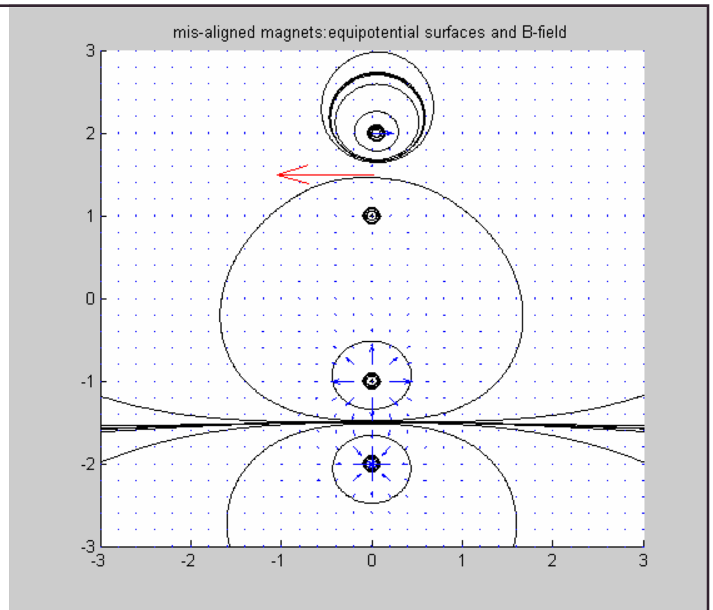
$$S = -\frac{mg}{\mu}$$

Trapping condition:

$$\frac{\left(S/2\right)^2}{B_0K} - 1 > 0$$




Off by 3 degrees




$\mu_1 = 70.911, g = 32 \qquad F_x = .4, F_y = 0, T_z = .004$

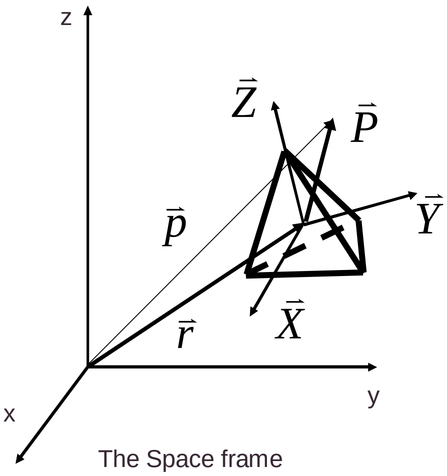
$\mu_2 = 5, m_2 = 5, \theta_2 = 3^o$

```
%dipole magnet in another magnet's B-field
clf;close all; [a,b,c]=sphere(10);
s = 'mis-aligned magnets:equipotential surfaces and B-field';
m1 = 5; m2 = 1; mu1 = 5; mu2 = 5; gee = 32.;
xn(1) =  0; xs(1)=  0; xn(2)= 0+sin(pi/60); xs(2)= 0;
yn(1) = -2; ys(1)= -1; yn(2)= 1+cos(pi/60); ys(2)= 1;
    axis([-3 3 -3 3]);  hold on; xgrid = -3:.02:3; ygrid = -3:.02:3;
    for i=1:2
       contour(.1*a+xn(i),.1*b+yn(i),.1*c)
contour(.1*a+xs(i),.1*b+ys(i),.1*c)
   end
%Forces on magnet 2 (movable) due to magnet 1 (fixed)
%        first compute pairwise distances
  rnn=(((xn(1)-xn(2))^2+(yn(1)-yn(2))^2))^(-3/2);
  rns=(((xn(1)-xs(2))^2+(yn(1)-ys(2))^2))^(-3/2);
  rsn=(((xs(1)-xn(2))^2+(ys(1)-yn(2))^2))^(-3/2);
  rss=(((xs(1)-xs(2))^2+(ys(1)-ys(2))^2))^(-3/2);
```

```
Fxn = (xn(2)-xn(1))*rnn-(xn(2)-xs(1))*rsn;
  Fyn = (yn(2)-yn(1))*rnn-(yn(2)-ys(1))*rsn;
  Fxs =-(xs(2)-xn(1))*rns+(xs(2)-xs(1))*rss;
  Fys =-(ys(2)-yn(1))*rns+(ys(2)-ys(1))*rss;
  Ftoty = mu1*mu2*(Fyn + Fys) - m2*gee
  %find mu1 so magnetic repulsion balances gravity
  if Ftoty ~= 0
      if (mu2*(Fyn + Fys)) ~=0
         mu1 = (m2*gee)/(mu2*(Fyn + Fys));   Ftoty = 0;
      else
         fprintf('impossible to balance');     break
     end
 end
 Ftotx = mu1*mu2*(Fxn + Fxs);
 %plot force through midpoint and compute torque
 Tx = cross([Fxn-Fxs,Fyn-Fys,0],.5*[xn(2)-xs(2),yn(2)-ys(2),0]);
 fprintf('net force and torque on magnet 2 \n')
 fprintf(' force %12.5f %12.5f torque %12.5f \n',Ftotx,Ftoty,Tx(3))
```

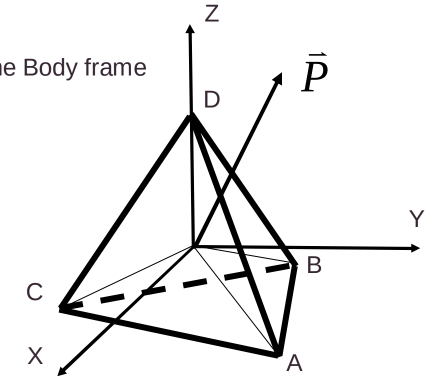
```
quiver((xn(2)+xs(2))/2,(yn(2)+ys(2))/2,3*Ftotx,3*Ftoty,'r')
        % Plot equipotential curves:
  [xx,yy] = meshgrid(xgrid,ygrid);
       p1 = ((xx-xn(1)).^2+(yy-yn(1)).^2).^(-1/2);
       p2 = ((xx-xn(2)).^2+(yy-yn(2)).^2).^(-1/2);
       p3 = ((xx-xs(1)).^2+(yy-ys(1)).^2).^(-1/2);
       p4 = ((xx-xs(2)).^2+(yy-ys(2)).^2).^(-1/2);
       pp = mu1*(p1 - p3) + mu2*(p2 - p4);
    levels = 10.^(-4:4);  levels = [-levels,levels];
    contour(xx,yy,pp,levels), title(s), colormap(1e-6*[1 1 1]);
x1=xx(1:10:301,1:10:301);y1=yy(1:10:301,1:10:301);
p13=p1(1:10:301,1:10:301).^3; p23=p2(1:10:301,1:10:301).^3;
p33=p3(1:10:301,1:10:301).^3; p43=p4(1:10:301,1:10:301).^3;
Bx =  mu1*(-(x1-xn(1)).*p13 + (x1-xs(1)).*p33)...
    + mu2*(-(x1-xn(2)).*p23 + (x1-xs(2)).*p43);
By =  mu1*(-(y1-yn(1)).*p13 +(y1-ys(1)).*p33)...
    + mu2*(-(y1-yn(2)).*p23 +(y1-ys(2)).*p43);
  quiver(x1,y1,Bx,By);    hold off
```

$$\vec{p} = \vec{r} + Q\vec{P}$$




The Space frame

The Body frame




$$Q = \begin{pmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ X_3 & Y_3 & Z_3 \end{pmatrix}$$

X,Y,Z expressed in Space frame provide the three columns of the rotation matrix

# References


- Feynman, Lectures on Physics I:
  Ch.9, Newton’s Laws of Dynamics
- Van Loan, Ch.9, The Initial Value Problem
- MD Simon, LO Heflinger & SL Ridgway, Spin Stabilized Magnetic Levitation, Am.J. Phys. 68(4), April 1997, 286-292.