SOLUTIONS

PROBLEM SET 5 Due Friday October 28

Physics 115B– FALL 2011

Points Assigned

This will be the last problem set before the Midterm Exam on Wednesday November 2.

Analytic:

10 **[1.]** Griffiths Problem 4.18

10 **[2.]** Griffiths Problem 4.19

10 **[3.]** Griffiths Problem 4.22

10 **[4.]** Griffiths Problem 4.24

10 **[5.]** Griffiths Problem 4.27

Comment: For the first part of the course, as you develop skill in programming, the computational problems will not necessarily have anything to do with quantum mechanics.

[6.] Write a C or C++ program to solve the discretized diffusion equation. One approach is to use two one dimensional arrays. First have a loop which gets the new density rhonew,

$$\text{rhonew}[n] = \text{rho}[n] + \frac{Ddt}{dx^2}\big(\,\text{rho}[n+1] - 2\,\text{rho}[n] + \text{rho}[n-1]\,\big)$$

10

Then follow it with a loop which resets the density to these newly computed values.

$$\text{rho}[n] = \text{rhonew}[n]$$

These loops (over n) need themselves to be enclosed by a loop which iterates over some number of time steps: "nested loops". Alternatively you could use a two dimensional array,

$$\text{rhonew}[n][m+1] = \text{rho}[n][m] + \frac{Ddt}{dx^2}\big(\,\text{rho}[n+1][m] - 2\,\text{rho}[n][m] - \text{rho}[n-1][m]\,\big)$$

(Here m is the time index.) What are the advantages/disadvantages of the two approaches?

[7.] So far in class we have not discussed the feasibility of numerical calculations. How can you estimate whether a calculation will be doable? A good crude approach is to think that the clock speed of the chip in your computer (about one GHz) means that the compute can do 10^9 operations (addition, subtraction, $\cdots$) per second. How long should your Kepler problem have taken to execute? Suppose you want to simulate the motion of 10^{11} stars in a galaxy with molecular dynamics. Could your computer do that problem? (How many time steps do you want to do?) If you can't simulate 10^{11} stars, how many could you study? (How long are you willing to run your computer? Will you use a super-computer with 10^5 processors? A petascale computer is described here: http://en.wikipedia.org/wiki/Blue_Waters).

10

70 TOTAL

#1 Griffith Problem 4.18

L_+ is hermitian conjugate of L_- ?

$$\langle \psi | L_+ \psi \rangle = \langle \psi | (L_x + iL_y)\psi \rangle = \langle \psi | L_x | \psi \rangle + i \langle \psi | L_y | \psi \rangle$$

$$= \langle (L_x - iL_y)\psi | \psi \rangle = \langle L_- \psi | \psi \rangle$$

So L_+ and L_- are hermitian conjugates.

(4.112) $\Rightarrow$ $L^2 = L_+ L_- + L_z^2 - \hbar L_z$ $\quad \boxed{L_\pm f_\ell^m = (A_\ell^m) f_\ell^{m\pm 1}}$

For $L_+ L_-$:

$$\langle f_\ell^m | L_+ L_- | f_\ell^m \rangle = (A_\ell^m)^2 \langle f_\ell^{m-1} | f_\ell^{m-1} \rangle = (A_\ell^m)^2$$

$$\| $$

$$\langle f_\ell^m | L^2 - L_z^2 + \hbar L_z | f_\ell^m \rangle$$

$$= (\hbar^2 \ell(\ell+1) - m^2\hbar^2 + m\hbar^2) \langle f_\ell^m | f_\ell^m \rangle$$

So $\boxed{(A_\ell^m) = \hbar \left[\ell(\ell+1) - m(m-1) \right]^{1/2}}$ for $L_+ L_-$

For $L_- L_+$:

$$\langle f_\ell^m | L_- L_+ | f_\ell^m \rangle = \langle f_\ell^m | L^2 - L_z^2 - \hbar L_z | f_\ell^m \rangle$$

$$= \hbar^2 \ell(\ell+1) - m^2\hbar^2 - m\hbar^2 \langle f_\ell^m | f_\ell^m \rangle$$

$\Rightarrow$ $\boxed{(A_\ell^m) = \hbar \left[\ell(\ell+1) - m(m+1) \right]^{1/2}}$ for $L_- L_+$

#2 Griffiths Problem 4.19

$L_x = yp_z - zp_y \quad L_y = zp_x - xp_z \quad L_z = xp_y - yp_x$

(a)

$[L_z, x] = [xp_y - yp_x, x] = [xp_y, x] - [yp_x, x] = -y[p_x, x] = -y(-i\hbar) = \boxed{i\hbar y}$

$[L_z, y] = [xp_y - yp_x, y] = [xp_y, y] - [yp_x, y] = x[p_y, y] = x(-i\hbar) = \boxed{-i\hbar x}$

$[L_z, z] = [xp_y - yp_x, z] = [xp_y, z] - [yp_x, z] = \boxed{0}$

$[L_z, p_x] = [xp_y - yp_x, p_x] = [xp_y, p_x] - [yp_x, p_x] = [x, p_x]p_y = \boxed{i\hbar p_y}$

$[L_z, p_y] = [xp_y - yp_x, p_y] = [xp_y, p_y] - [yp_x, p_y] = -[y, p_y]p_x = \boxed{-i\hbar p_x}$

$[L_z, p_z] = [xp_y - yp_x, p_z] = [xp_y, p_z] - [yp_x, p_z] = \boxed{0}$

(b)

$[L_z, L_x] = [L_z, yp_z - zp_y] = [L_z, yp_z] - [L_z, zp_y]$

$= y[L_z, p_z] + [L_z, y]p_z - z[L_z, p_y] - [L_z, z]p_y$

$= 0 + i\hbar x p_z - z(-i\hbar p_x) + 0 = i\hbar(zp_x - xp_z) = \boxed{i\hbar L_y}$

(c)

$\boxed{[L_z, p^2]} = [L_z, p_x^2] + [L_z, p_y^2] + [L_z, p_z^2]$

$= x[L_z, x] + [L_z, x]x + y[L_z, y] + [L_z, y]y + z[L_z, z] + [L_z, z]z$

$= x(i\hbar y) + (i\hbar y)x + y(-i\hbar x) + (-i\hbar x)y + 0 + 0 = \boxed{0}$

$\boxed{[L_z, p^2]} = [L_z, p_x^2] + [L_z, p_y^2] + [L_z, p_z^2]$

$= p_x[L_z, p_x] + [L_z, p_x]p_x + p_y[L_z, p_y] + [L_z, p_y]p_y + p_z[L_z, p_z] + [L_z, p_z]p_z$

$= p_x(i\hbar p_y) + (i\hbar p_y)p_x + p_y(-i\hbar p_x) + (-i\hbar p_x)p_y + 0 = \boxed{0}$

(d) L_x, L_y, L_z commute with p^2 from (c) above

L_x, L_y, L_z commute with r^2 (from above) and thus commute with $V(r)$ (a function of r^2).

#3 Griffiths Problem 4.22

(a) $L_+ Y_\ell^\ell = 0$ since $m=\ell$ is largest m value possible & L_+ destroys state.

(b) $L_z Y_\ell^\ell = \hbar \ell Y_\ell^\ell \Rightarrow -i\hbar \frac{\partial}{\partial\phi} Y_\ell^\ell = \hbar \ell Y_\ell^\ell$

Let $Y_\ell^\ell = A\,\Theta(\theta)\,\Phi(\phi)$

So $\frac{1}{\Phi}\frac{\partial\Phi}{\partial\phi} = i\ell \Rightarrow \Phi(\phi) = e^{i\ell\phi}$ (Φ part)

Eqn 130: $L_+ Y_\ell^\ell = \hbar e^{i\phi}\left(\frac{\partial}{\partial\theta} + i\cot\theta\frac{\partial}{\partial\phi}\right) Y_\ell^\ell = 0$ (from (a))

$Y_\ell^\ell = A\,\Theta(\theta)e^{i\ell\phi} \Rightarrow \left(\frac{\partial}{\partial\theta} + i\cot\theta\frac{\partial}{\partial\phi}\right) A\,\Theta(\theta)\,e^{i\ell\phi} =$

$$A e^{i\ell\phi}\frac{\partial\Theta}{\partial\theta} + iA\Theta\cot\theta\frac{\partial}{\partial\phi}e^{i\ell\phi} = 0$$

$$\Rightarrow A e^{i\ell\phi}\frac{\partial\Theta}{\partial\theta} - A\Theta e^{i\ell\phi}(\ell)\cot\theta = 0$$

$$\Rightarrow \frac{\partial\Theta}{\partial\theta} - \Theta\ell\cot\theta = 0$$

$$\frac{\partial\Theta}{\Theta} = \ell\cot\theta \Rightarrow \ln(\Theta) = \int \ell\frac{\cos\theta}{\sin\theta}\,d\theta = \ell\ln(\sin\theta)$$

$$\Rightarrow \Theta = (\sin\theta)^\ell$$

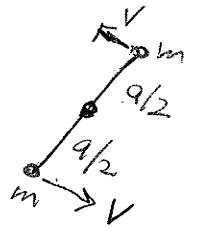
So, $\boxed{Y_\ell^\ell = A\,\Theta(\theta)\,\Phi(\phi) = A(\sin\theta)^\ell e^{i\ell\phi} = A(\sin\theta\, e^{i\phi})^\ell}$

(c) $\langle Y_\ell^\ell | Y_\ell^\ell\rangle = \int_0^{2\pi} e^{i\ell\phi}e^{-i\ell\phi}d\phi \int_0^\pi A^2\sin^\ell\theta\sin\theta\,d\theta = 2\pi A^2\int_0^\pi \sin^{2\ell+1}\theta\,d\theta$

$= 2\pi A^2\frac{(2)(2\ell)!!}{(2\ell+1)!!} = 4\pi A^2\frac{2\cdot4\cdot6\cdots2\ell}{1\cdot3\cdot5\cdots(2\ell+1)} = 4\pi A^2\frac{(2\cdot4\cdot6\cdots2\ell)}{1\cdot3\cdot5\cdots(2\ell+1)}\times\frac{(2\cdot4\cdot6\cdots2\ell)}{(2\cdot4\cdot6\cdots2\ell)}$

$= 4\pi A^2\frac{(2^\ell\ell!)^2}{(2\ell+1)!} \Rightarrow \boxed{A = \frac{1}{2^{\ell+1}\ell!}\sqrt{\frac{(2\ell+1)!}{\pi}}}$ ✓ (3)

#4 Griffiths Problem 4.24



$r \times p$

(a) $L = 2\left(\frac{a}{2} \times mv\right) = amv$

Hamiltonian (Energy):

$$H = KE = \frac{1}{2}mv^2 + \frac{1}{2}mv^2 = mv^2$$

$$\Rightarrow \boxed{H = \frac{L^2}{ma^2}}$$, Eigenvalues of L^2 are $\hbar^2 l(l+1)$

$$H\psi = E\psi \Rightarrow \boxed{E_l = \frac{\hbar^2 l(l+1)}{ma^2}} \quad l = 0, 1, 2, \ldots$$

(b) Eigenfunctions are those of the L^2 operator; spherical harmonics:

$$\boxed{|\psi_{lm}\rangle = Y_l^m(\theta, \phi)}$$ with $\boxed{2l+1}$ degenerate energies

#5 Griffiths Problem 4.27

$$\chi = A\begin{pmatrix} 3i \\ 4 \end{pmatrix}$$

(a) $\langle \chi | \chi \rangle = A^2 (-3i \;\; 4)\begin{pmatrix} 3i \\ 4 \end{pmatrix} = A^2 \cdot 25 = 1$

$$\boxed{A = 1/5}$$

(b) $\langle S_x \rangle = \chi^\dagger \frac{\hbar}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\chi = \frac{1}{25}\left[(-3i \;\; 4)\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 3i \\ 4 \end{pmatrix}\right] = \boxed{0}$

S_x

$\langle S_y \rangle = \frac{1}{25}(-3i \;\; 4)\frac{\hbar}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 3i \\ 4 \end{pmatrix} = \frac{1}{25}(-3i \;\; 4)\begin{pmatrix} -4i \\ -3 \end{pmatrix} = \frac{\hbar}{50}(-24) = \boxed{-\frac{12\hbar}{25}}$

S_y

$\langle S_z \rangle = \frac{1}{25}(-3i \;\; 4)\frac{\hbar}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 3i \\ 4 \end{pmatrix} = \frac{\hbar}{50}(-3i \;\; 4)\begin{pmatrix} 3i \\ -4 \end{pmatrix} = \boxed{-\frac{7\hbar}{50}}$

(c) For spin 1/2 $\langle S_j^2 \rangle = \chi^\dagger \left(\frac{\hbar}{2}\right)^2 \sigma_j^2 \chi = \frac{\hbar^2}{4} I \langle \chi^\dagger | \chi \rangle$ (for all j)

$= \boxed{\hbar^2/4}$

$\sigma_{S_x} = \left(\langle S_x^2 \rangle - \langle S_x \rangle^2\right)^{1/2} = \left(\frac{\hbar^2}{4} - 0\right)^{1/2} = \boxed{\frac{\hbar}{2}}$

$\sigma_{S_y} = \left(\langle S_y^2 \rangle - \langle S_y \rangle^2\right)^{1/2} = \left(\frac{\hbar^2}{4} - \frac{12^2}{25^2}\hbar^2\right)^{1/2} = \hbar\left(\frac{624 - 576}{4 \times 25^2}\right)^{1/2} = \boxed{\frac{7}{50}\hbar}$

$\sigma_{S_z} = \left(\langle S_z^2 \rangle - \langle S_z \rangle^2\right)^{1/2} = \left(\frac{\hbar^2}{4} - \frac{7^2}{4 \times 25^2}\hbar^2\right)^{1/2} = \hbar\left(\frac{25^2 - 49}{2500}\right)^{1/2} = \boxed{\frac{12}{25}\hbar}$

(d) Uncertainty principle check:

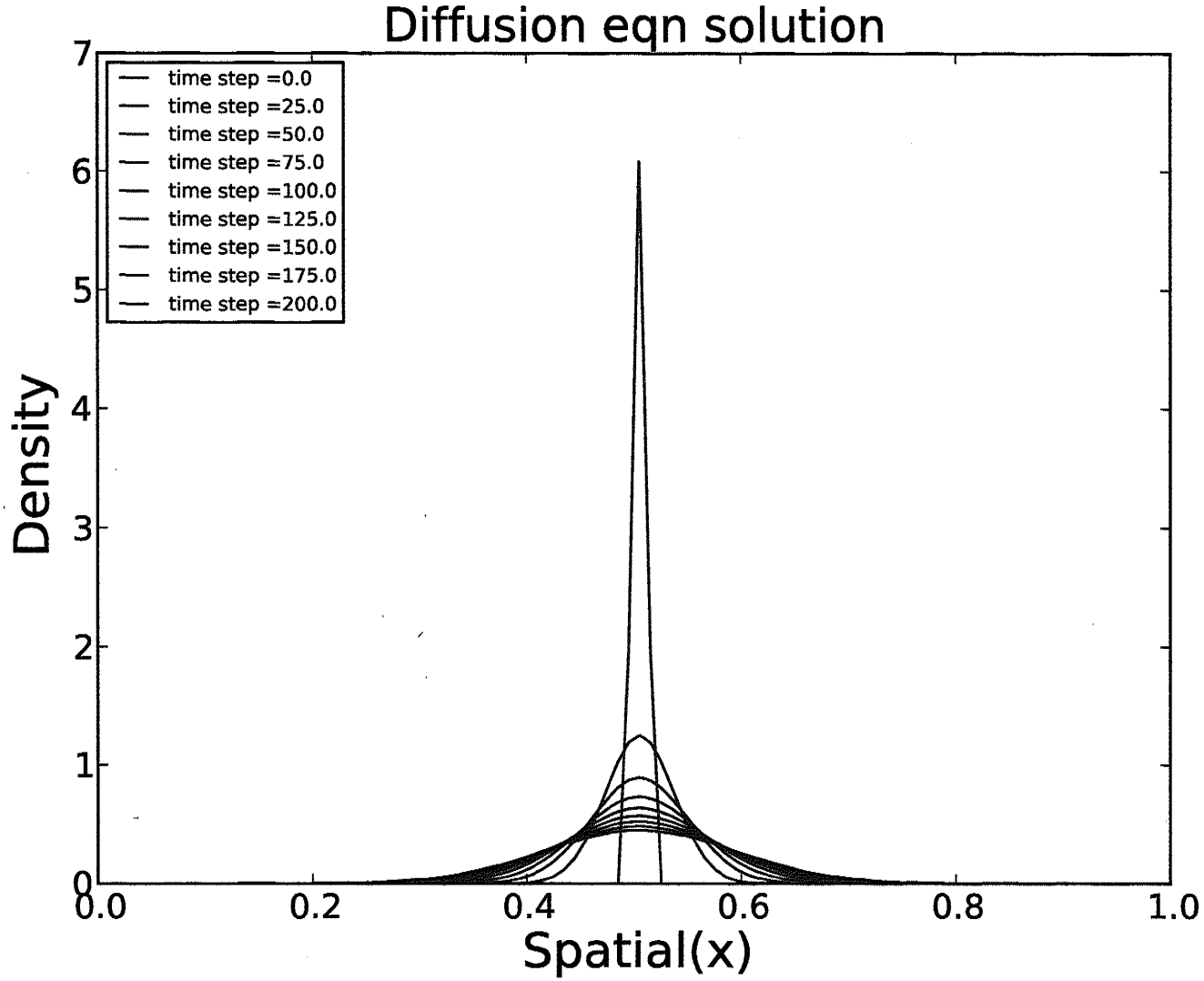
$\sigma_{S_x}\sigma_{S_y} = \frac{7}{100}\hbar^2$ vs $\frac{\hbar}{2}|\langle S_z \rangle| = \frac{\hbar^2}{2}\left(\frac{7\hbar}{50}\right)$ (Equal! but OK)

$\sigma_{S_y}\sigma_{S_z} = \frac{7}{50} \times \frac{12}{25}\hbar^2 \geq \frac{\hbar}{2}|\langle S_x \rangle| = \frac{\hbar^2}{2}(0)$ (yes)

$\sigma_{S_x}\sigma_{S_z} = \left(\frac{\hbar}{2}\right)\left(\frac{12\hbar}{25}\right) = \frac{12\hbar^2}{50} \geq \frac{\hbar}{2}|\langle S_y \rangle| = \frac{\hbar}{2}\frac{7}{50}$ (yes)

```
//
//  P115B  HW5 Prob 6
//   g++ -o diff_eqn.exe diff_eqn.c
//  Program to to solve diffusion equation
//   with forward time algorithm
#include <iostream>
#include <fstream>
#include <math.h>
using namespace std;
int main()
{

//  Assign constants
//
    const int N = 100;         // Number of x-axis grid points
    const int t_steps = 10;  // Number of time steps
    double L = 1.0;            //  X grid runs from 0 to L
    double dx = L/(N-1);       //  Size of x-axis grid
    double dt = 1.0e-6;             // Size of time step
    double D = 2.0;             // D coeff in model
    double D_factor =  (D * dt)/(dx *dx);
//
// Initialize rho arrays
//
    double rho[N],rhonew[N];
    for (int i=0; i< N; i++)
    {
        rho[i]    = 0.0;
        rhonew[i] = 0.0;
    }
    rho[N/2] = 1.0/dx;  // Initial condition delta function at center
// Set up disk files for output
    ofstream fout("diffeqn_out1.dat");
//
// Loop over x grid and time steps
//

    for (int i = 0; i < t_steps; i++)    //  Loop over time steps
    {

      for (int n = 1; n < N-1; n++)      // Calculate new rho values
         {
           rhonew[n] = rho[n] + D_factor * ( rho[n+1] - 2*rho[n] + rho[n-1]);
         }

      for (int n = 0; n < N; n++)        // Reset rho values for next time step & output
         {
           rho[n] = rhonew[n];
           fout << i << "  " << n << "  " << rho[n]<< endl;
         }
     }

    cout << " Finished writing data to file" << endl;
    return 0;
}
```

6


Diffusion eqn solution
time step =0.0
time step =25.0
time step =50.0
time step =75.0
time step =100.0
time step =125.0
time step =150.0
time step =175.0
time step =200.0
Density
Spatial(x)
0
1
2
3
4
5
6
7
0.0
0.2
0.4
0.6
0.8
1.0

#6 Diffusion Equation Solver

$$D\frac{\partial^2 \rho}{\partial x^2} = \frac{\partial \rho}{\partial t} \Rightarrow \rho(x,t)$$

$$\rho(x+dx) = \rho(x) + \rho'(x)\,dx + \frac{1}{2}\rho''(x)\,dx^2$$
$$\rho(x-dx) = \rho(x) - \rho'(x)\,dx + \frac{1}{2}\rho''(x)\,dx^2$$

$$\Rightarrow \rho''(x)\,dx^2 = \rho(x+dx) + \rho(x-dx) - 2\rho(x)$$

$$\Rightarrow \frac{\rho(x,t+dt) - \rho(x,t)}{dt} = D\left[\frac{\rho(x+dx,t) + \rho(x-dx,t) - 2\rho(x,t)}{dx^2}\right]$$

$$\left(\frac{\partial \rho}{\partial t}\right) \qquad D\left(\frac{\partial^2 \rho}{\partial x^2}\right)$$

$$\rho(x,t+dt) = \rho(x,t) + \left(\frac{D*dt}{dx^2}\right)\left[\rho(x+dx,t) + \rho(x-dx,t) - 2\rho(x,t)\right]$$

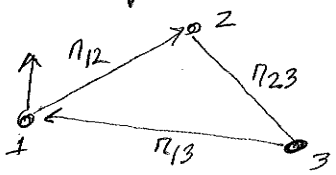
1D arrays are simpler / have lower memory requirement

2D arrays are easier if you want to create graphs/output in the same program (without writing to disk)

#7 Simulating 10^{11} stars in a galaxy

First, we're 3D so we have (3) position & (3) velocity components per star.

For 3 stars system:



If position of star 1 moves from r_1 to r_1', r_{12} & r_{13} must be recomputed.

In general, when looping through one time step for 10^{11} star at star n, all stars from 1 to $n-1$ have moved which requires updates to $(n-1)$ pairwise $\vec{r}$ vectors and re-computes of acceleration & velocity charges for $(n-1)$ new vectors.

Let assume it takes 20 math ops to update 1 pairwise change. So to cycle through 10^x stars would take

$$20 \text{ops} \times \frac{n(n-1)}{2} \approx 10 \times 10^{2x} = 10^{2x+1} \text{ operation/cycle}$$

Now assume we want to update every year for 100 years (10^2 time cycles).

$$\text{Total compute time at } 10^9 \text{ ops/sec (flops)} = \frac{10^{2x+1} \text{ oper/cycle} \times 10^2 \text{ time cycle}}{10^9 \text{ ops/sec}} \approx 10^{2x-6} \text{ sec}$$

For $p=11 \Rightarrow 10^{22-6} = 10^{16}$ seconds $\approx 3 \times 10^8$ years to simulate 10^{11} stars !!

Let say, we only have 1-2 weeks/ computes time is 10^6 sec.

$\Rightarrow 2x-6=6 \Rightarrow x=6$. We could model 10^6 stars (million)

With 10^5 processors (in parallel) $\frac{10^{2x+1} \times 10^2}{10^{14} \text{ ops/sec}} \approx 10^{2x-11}$ $2x \cong 17$ $\approx 10^8$ stars (100 million)