Cornell University

Department of Physics

Phys 480/680, Astro 690 January 27, 2001

Computational Physics, Spring 2001

Problem Set # 1

(Due Friday, February 2 in Computer Lab.)

Agenda and readings for the week of February 1:

Readings marked NR are from *Numerical Recipes: The Art of Scientific Computing*, 2nd edition (in C). Readings marked LN are from the course lecture notes to be found at http://www.ccmr.cornell.edu/~muchomas/P480 .

- Lec1 01/23 [1 1/2 hrs]: Course overview; Introduction to density functional theory.
  Reading: LN "Overview of quantum mechanics . . . ": 1-3.
- Lec2, 01/25 [1 1/13 hrs]: General plan of attack for Scientific Computing; Minimization of energy leanding to Kohn-Sham equations.
  Reading: LN "Overview of quantum mechanics . . . ": 4.1-4.3,5.
- Lec3, 01/26 [1 hr]: Overview of self-consistent solution of Kohn-Sham equations; Overview of numerical integration.
  Reading: LN "Overview of quantum mechanics . . . ": 4.4; LN "Numerical Integration . . . ": 1-2.1; NR 4.0-4.1
- Lecs4-5, 01/29, 01/31 [1 hr @]: Numerical integration; Solution of ODE's; Poisson's equation for atoms.
  Reading: LN "Numerical Integration . . . ": 2.2-4; NR 4.1-4.3, 16.0–16.1
- Lab1, 02/02: Discuss code organization and explore changes of variables for atoms.

Contents

1 Integration Routines **2**
- 1.1 Driver program . . . 2
- 1.2 Trapezoidal Rule . . . 3
- 1.3 Simpson's Rule . . . 3
- 1.4 Debugging . . . 4

2 Application to atoms **4**
- 2.1 Normalization . . . 4
- 2.2 Electron-Nuclear potential energy . . . 4
- 2.3 Electron-electron potential energy . . . 4

1 Integration Routines

1.1 Driver program

Cut and paste the main program below from the HTML version of this problem set. You should place your code for Problems 1.2 and 1.3 at the spot labelled "YOUR ROUTINES GO HERE".

The program below will drive your routines to compute the integral

$$\int_0^1 \frac{1}{\frac{1}{4}+\frac{\pi}{8}} \frac{1}{(1+x^2)^2}\, dx = 1.$$

```
/*
  Main program for PS#1
  Tests Simpson and Trapezoid integration routines
 */


/* Standard C header for using I/O and math libraries */
#include <stdio.h>
#include <math.h>

/* *** YOUR ROUTINES GO HERE *** */
                    .
                    .
                    .


/* Maximum size of N for integrations */
#define Nmx 200000
main()
{
  /* Declare space for arrays, remember that there are N+1 points! */
  /* Double precision for accurate numerical work... */
  double r[Nmx+1],dr[Nmx+1],f[Nmx+1];

  /* Declare working variables */
  int i,N;
  double u;

  /* Trick to get all digits of PI to the full machine precision */
  double pi;
  pi=4.*atan(1.);


  /* Loop over sizes 2, 20, 200, 2000, ..., Nmx */
  for (N=2; N<=Nmx; N*=10) {
    /* For each size, fill in arrays */
    for (i=0; i<=N; i++) {
      /* Let u be equally spaced between 0 and 1, -inclusive- */
      u=((double) i)/((double) N);

      /* Define change of variables */
      r[i]=u;
      dr[i]=1.;
```

```
      /* Specify function */
      f[i]=1/(1+r[i]*r[i])/(1+r[i]*r[i])/(0.25+0.125*pi);
    }

    /* Special cases at end points (reasons for this will become clear...) */
    f[0]=f[0];
    r[0]=r[0];
    dr[0]=dr[0];

    f[N]=f[N];
    r[N]=r[N];
    dr[N]=dr[N];

/* Call integration routines and output results... */
    printf("Trap, Simp: %7d %20.14f,%20.14f\n",N,trapint(f,r,dr,N),simpint(f,r,dr,N));
  }

}
```

1.2 Trapezoidal Rule

Write a function of the following form:

`double trapint(double f[],double r[],double dr[],int N)`

Return value:

- Trapezoid rule approximation to $\int_0^1 f(r(u))\frac{dr}{du}(u)\,du = \int_{r(0)}^{r(1)} f(r)\,dr$.

Input:

- f[]: values of $f(r(u_i))$ for $i = 0 \ldots N$ ($u_i \equiv 1/N$).
- r[]: values of $r(u_i)$.
- dr[]: values of $\frac{dr}{du}(u_i)$.
- N: number of integration intervals.

1.3 Simpson's Rule

Write a function of the following form:

`double simpint(double f[],double r[],double dr[],int N)`

Return value:

- Simpson's rule approximation to $\int_0^1 f(r(u))\frac{dr}{du}(u)\,du = \int_{r(0)}^{r(1)} f(r)\,dr$.

Input:

- f[]: values of $f(r(u_i))$ for $i = 0 \ldots N$ ($u_i \equiv 1/N$).
- r[]: values of $r(u_i)$.
- dr[]: values of $\frac{dr}{du}(u_i)$.
- N: twice the number of integration intervals (i.e., the number the number of sample points *minus* one.)

1.4 Debugging

Run your routines with the driver program given in Problem 1.1 and verify that they exhibit the expected behavior. You will have to demonstrate this behavior for the grader.

2 Application to atoms

Note that, while for now you must rely on "outside sources" for $\psi_{1s}(r)$ and ϕ_{1s} below, soon your your own codes will compute them!

To use your integration routines for the problems below, you will need a change of variables function $r(u)$ for which $r(0) = 0$ and $r(1) = \infty$. One suitable choice is

$$r(u) = \frac{1}{1-u} - 1,$$

but you can get better results with other choices. You may wish to experiment...

Again, you will be asked to demonstrate that your code performs the tasks described below to the grader and in lab. To accomplish this, you may wish to have separate files for each exercise, or to comment in and out appropriate code blocks.

2.1 Normalization

Modify the driver routine given in Problem 1.1 so that it computes the normalization of the textbook ground-state ("1s state") wave function for the hydrogen atom,

$$\psi_{1s}(r) = \frac{1}{\sqrt{\pi}} e^{-r}.$$

Specifically, use your routines to check that the total probability $\mathcal{P}(\vec{x}) = \psi^*(\vec{x})\,\psi(\vec{x})$ of finding the electron adds up to one,

$$\int \mathcal{P}_{1s}(\vec{x})\,dV = \int \psi^*_{1s}(\vec{x})\psi_{1s}(\vec{x})\,dV = \int_0^\infty \psi^2_{1s}(r)\,4\pi r^2\,dr = 1.$$

Hint: Some thought is required to handle properly the endpoints of the integration. This can be done by modifying the code block in the driver routine labelled "Special cases at end points".

2.2 Electron-Nuclear potential energy

Write a main program which uses your subroutines to compute the electron-nuclear potential energy for a single electron in the 1s state,

$$V_{\text{el}-\text{nuc}} = \int n(\vec{x}) V_{\text{nuc}}(\vec{x})\,dV = \int_0^\infty \left(\psi^2_{1s}(r)\right)\left(-\frac{1}{r}\right) 4\pi r^2\,dr.$$

Note: because there is only one electron, $n_{1s}(\vec{x}) = \psi^2_{1s}(\vec{x})$.
Hint: Again, the endpoints of the integration require some care.

2.3 Electron-electron potential energy

Write a main program which uses your subroutines to compute the "Hartree" electron-electron potential energy for the 1s state,

$$V_{\text{el}-\text{el}} = \frac{1}{2}\int \phi(\vec{x}) n(\vec{x})\,dV = \frac{1}{2}\int_0^\infty \phi_{1s}(r)\psi^2_{1s}(r)\,4\pi r^2\,dr.$$

Note: For $n(\vec{x}) = \psi^2_{1s}(\vec{x})$, the potential from Poisson's equation $\nabla^2\phi_{1s}(\vec{x}) = -4\pi n_{1s}(\vec{x})$ is

$$\phi_{1s}(r) = \frac{1 - e^{-2r}}{r} - e^{-2r}.$$

Hint: endpoints ...