

PY 421 - Introduction to Computational Physics

Homework # 1. January 25, 2013.

Due at 3PM on Friday February 1.

The assignment:

For this assignment have to write and return two simple programs, which you should call `hw1a.c` and `hw1b.c`.

The program `hw1a.c` should calculate an approximation to the second derivative of the function $f(x) = \sin(x)$ which uses the five values taken by $f(x)$ at x , $x \pm dx$ and $x \pm 2dx$ to produce a result with an error of order dx^4 . The approximation formula can be seen as a generalization of the central difference approximation.

You are allowed to use the program `derivatives.c` posted in `~rebbi/courseware/code`

to do this part of the assignment. Once you have determined the formula, it should be easy to modify `derivatives.c` to produce the desired result. Also, you may want to change the text output, for example replacing “Second der. c. approx.” with “Higher order approx. to the second deriv.”.

Please note that this program, as well as all programs for future assignments, should be well written: the code should not contain variables or instructions which are not needed, it should only do what it is required to do, the lines should be properly indented, which is very easy to do with the Emacs editor, and the like. In other words, you should pay attention to your style of programming, not only to obtaining the correct results.

The program `hw1b.c` should calculate the integral of $\sin(x)$ between 0.5 and 1.5 using Simpson’s formula. It should ask the user to input the number of subintervals and print out the result, the error, and the error rescaled by dx^4 . For definiteness, the code should implement the formula in Eq. 34 of the lecture notes on integration.

Again, you can take advantage of the program `integration.c` posted in `~rebbi/courseware/code` to do this part of the assignment, but your code should only print out the result of the Simpson’s formula (see my note above about the code doing only what it is required to do.) Please keep the precision `%12.9f` in the `printf` format, as in `integration.c`.

Returning the assignment:

Since you will be returning two separate files, before submitting your work you must put the two files together into what is known as “tar” file (the name comes from “tape archive”).

To generate the tar file I strongly recommend that you create first a temporary directory, which I will call `tmp`, with the command

```
mkdir tmp
```

This would not be strictly necessary, but I suggest that you do it for caution, to avoid erasing the files with your work in case you make a mistake. Then copy the two files `hw1a.c` and `hw1b.c` to the new directory and change directory to `tmp`:

```
cp hw1a.c hw1b.c tmp
```

```
cd tmp
```

Now you can archive together the two files into a single file which I will call `asgn1.tar` (one could use a different name) with the command:

```
tar cvf asgn1.tar hw1a.c hw1b.c
```

“tar” is the Unix command that does the job, the flags “cvf” stand for “compress” (the files will be put together), “verify” (tar will echo the names of the files), and “file name” (the result will be put in the file whose name immediately follows the f). Finally the command line is completed with the names of the files you want to archive together.

You can verify the correctness of what you have done in two ways:

1) if you run the command

```
tar tf asgn1.tar
```

tar will list the files in the archive `asgn1.tar`

2) you can delete the two files

```
rm hw1a.c hw1b.c
```

and run `ls` to check that that the files are no longer there. (This is one of the reasons why I recommend that you work in the `tmp` directory. You will still have the original files in the parent directory in case something went wrong.)

Now you can use tar to recover the files from the archive with the command

```
tar xvf asgn1.tar
```

(Note the flag “x”, which stands for “expand”, instead of the flag “c” you used to put the files together in the archive.)

Now that you have verified that everything is o.k., you can proceed to submit the assignment following the procedure explained below.

Once you have put together the two programs in a single tar file, you should copy this file using one of the CAS 327 workstations onto the file

`~rebbi/courseware/asn/asn1.xyyyyy`,

where xyyyyy stands for a personal identifier which will be communicated to you separately. For example, if you called the tar file with the programs “asn1.tar”, then in the working directory where you have the file you should execute the command

```
cp  asn1.tar  ~rebbi/courseware/asn/asn1.xyyyyy
```

Note that you must type `~rebbi/courseware/asn/asn1.xyyyyy` in full. You cannot use the tab key to complete the name of the file.

The way this works is as follows. The directory `~rebbi/courseware/asn/` has search (i.e. x) permission, but not read (r) or write (w) permission. The files `asn1.xyyyyy` (again, with the students’ identifiers replacing xyyyyy) have been created by me in that directory, but are blank (they contain the words “blank file”.) These files have write (w) permission, but not read permission (r), which means that you can overwrite their contents with the content of the file you are copying over them, but nobody can read those files (except me, of course). Also, since the directory `asn` does not have read permission, nobody can list its content and find out the personal identifiers. The system is not fail-proof, but guarantees some security for the homeworks. So long as you are the only ones to know your identifier, other people cannot mess with your assignment files and in any event nobody can read your work.

For students working on their own personal computers or laptops, within the bu.edu domain it should also be possible to use the `scp` command to copy the assignments onto the `asn` files, e.g.

```
scp asn1.tar username@327lfs:~rebbi/courseware/asn/asn1.xyyyyy
```

Grading:

The two programs will count as two separate problems and each correct solution will be given a score of 50, with points deducted for mistakes or poor code presentation, according to the severity of the error.