PY 421 - Introduction to Computational Physics

First midterm exam.

March 5, 2013.

Do all three problems. Correct solutions will be given 30 points for problem 1 and 35 points each for problem 2 and 3, for a maximum total score of 100. Solutions with errors will be given partial credit, according to the severity of the error.

The solutions to problems 1 and 2 must be returned on an exam book or on sheets of paper. Please **make sure to write your name on the cover of the exam book or on all sheets.** (If you staple the sheets together you may write your name on the first sheet only.)
Please note: The solutions to problems 1 and 2 must include details of the reasoning and mathematical passages followed to arrive at the solution. Simply writing down the requested mathematical answer, even if correct, does not constitute an acceptable solution to the problems.

Problem 3 requires that you write a program. You should copy the completed program, on the CAS cluster, to the file
∼rebbi/courseware/asgn/midterm1.xxyyyy
where xxyyyy stands, as usual, for your personal identifier.

Problem 1

Assume that $f(x)$ is continuous with as many continuous derivatives as needed.
1) Find a numerical approximation to the fourth derivative of f at $x = x_0$, which uses the 5 values taken by f at x_0, $x_0 \pm \delta$ and $x_0 \pm 2\delta$ and has an error of order δ^2. Note that on grounds of symmetry and dimensions the approximation formula will be of the form

$$f^{iv}(x_0) \approx \frac{\alpha[f(x_0+2\delta)+f(x_0-2\delta)]+\beta[f(x_0+\delta)+f(x_0-\delta)]+\gamma f(x_0)}{\delta^4} \tag{1}$$

where α, β, γ are numerical coefficients. For this problem you must find α, β and γ, giving a detailed explanation of how you determined their values, and show that the error is $O(\delta^2)$.

2) Let $g(x) = f''(x)$. Then $f^{iv}(x) = g''(x)$. Show that the formula derived in 1) can also be obtained by using the central difference formula to approximate the value of $g''(x)$, by replacing $g(x_0 + \delta)$, $g(x_0)$, $g(x_0 - \delta)$ in that formula by $f''(x_0 + \delta)$, $f''(x_0)$, and $f''(x_0 - \delta)$, and by using then again the central difference formula to approximate $f''(x_0 + \delta)$, $f''(x_0)$, and $f''(x_0 - \delta)$.

Hint: A simple way (but not the only one) to derive the coefficients is to notice that the formula must give the exact value of the fourth derivative at $x = 0$ for the functions $f(x) = 1, x, x^2, x^3, x^4, x^5$.

Solution

1) The fourth derivatives of $f(x) = 1, x, x^2, x^3, x^4, x^5$ are, in order, $0, 0, 0, 0, 24, 0$. The approximation formula 1 produces a zero result at $x = 0$ for all odd powers of x. Thus we only need to impose the correctness of the formula for $f(x) = 1, x^2, x^4$. Doing the algebra we find the conditions

$$\begin{aligned} \frac{2\alpha + 2\beta + \gamma}{\delta^4} &= 0 \\ \frac{8\alpha\delta^2 + 2\beta\delta^2}{\delta^4} &= 0 \\ \frac{32\alpha\delta^4 + 2\beta\delta^4}{\delta^4} &= 24 \end{aligned} \tag{2}$$

or

$$\begin{aligned} 2\alpha + 2\beta + \gamma &= 0 \\ 8\alpha + 2\beta &= 0 \\ 32\alpha + 2\beta &= 24 \end{aligned} \tag{3}$$

with solution

$$\alpha = 1, \qquad \beta = -4, \qquad \gamma = 6 \tag{4}$$

Since the first term in a Taylor series expansion of $f(x)$ for which the approximation would not work is the term with x^6 and this term gives a contribution $O(\delta^6)/\delta^4$ to the formula the error is $O(\delta^2)$.

2) The central difference approximation to $g''(x_0)$ is

$$\begin{aligned} g''(x_0) &\approx \frac{g(x_0 + \delta) + g(x_0 - \delta) - 2g(x_0)}{\delta^2} \\ &= \frac{f''(x_0 + \delta) + f''(x_0 - \delta) - 2f''(x_0)}{\delta^2} \end{aligned} \tag{5}$$

Using in turn the central difference formula to approximate the second derivatives in Eq. 5 we find

$$f^{iv} \approx \frac{f(x_0+2\delta)+f(x_0)-2f(x_o+\delta)}{\delta^4}$$
$$+\frac{f(x_0)+f(x_0-2\delta)-2f(x_o-\delta)}{\delta^4}-2\,\frac{f(x_0+\delta)+f(x_0-\delta)-2f(x_0)}{\delta^4}$$
$$=\frac{[f(x_0+2\delta)+f(x_0-2\delta)]-4[f(x_0+\delta)+f(x_0-\delta)]+6f(x_0)}{\delta^4} \tag{6}$$

Problem 2

Consider the differential equation

$$\frac{d^2 f(x)}{dx^2} = -p(x) \tag{7}$$

where $f(x)$, $p(x)$ are defined over the interval $0 \leq x \leq L$ with periodic boundary conditions, $p(x)$ is a given function, and $f(x)$ is the function to be determined. Moreover the function $p(x)$ is assumed to satisfy the condition

$$\int_0^L p(x)\,dx = 0 \tag{8}$$

which is required for consistency (as can be shown by taking the integral from 0 to L of both sides of Eq. 7) and the ambiguity coming from the fact that if $f(x)$ is a solution $f(x)+\text{const}$ is also a solution is fixed by demanding that the solution satisfies

$$\int_0^L f(x)\,dx = 0 \tag{9}$$

For a numerical solution the problem is discretized by dividing the interval $0-L$ into N subintervals of length $a = L/N$, the functions f and p are represented by arrays $f_j = f(x=ja)$, $p_j = p(x=ja)$ with $0 \leq j \leq N-1$, and Eq. 7 is replaced by the system of linear equations

$$\frac{f_{j+1}+f_{j-1}-2f_j}{a^2} = -p_j, \qquad 0 \leq j \leq N-1 \tag{10}$$

where periodic boundary conditions are assumed. It is also assumed that the array p_j satisfies the condition

$$\sum_{j=0}^{N-1} p_j = 0 \tag{11}$$

(again required for consistency) and the ambiguity $f_j \to f_j + \text{const}$ is fixed by demanding that the solution satisfies

$$\sum_{j=0}^{N-1} f_j = 0 \tag{12}$$

1) Show how to solve Eq. 10 by using the Fourier transform.

2) Consider the variant of the above problem where one assumes that $f(x)$ and $p(x)$ do not satisfy periodic boundary conditions, but rather Dirichlet boundary conditions

$$f(0) = f(L) = 0, \qquad p(0) = p(x) = 0 \tag{13}$$

In the discretized problem $f(x)$, $p(x)$ are represented by arrays f_j, p_j with j varying now in the range $0 \leq j \leq N$ and f, p satisfying

$$f_0 = f_N = 0, \qquad p_0 = p_N = 0 \tag{14}$$

(no condition needs to be imposed now on the sum of the elements of p.) The variables f_j for $1 \leq j \leq N-1$ are determined by the system of linear equations

$$\frac{f_{j+1} + f_{j-1} - 2f_j}{a^2} = -p_j, \qquad 1 \leq j \leq N-1 \tag{15}$$

(this is the same as Eq. 10 but with a different range for j.)
Show that the solution found in 1) can be applied to this system also by using a suitable extension of the arrays f_j and p_j over the range $0 \leq j \leq 2N-1$. Precisely, consider two arrays g_j, q_j, which are defined over $0 \leq j \leq 2N-1$ with periodic boundary conditions (with period $2N$), whose components for $0 \leq j \leq N$ are the same as those of f and g, i.e.

$$g_j = f_j, \quad q_j = p_j, \qquad \text{for} \quad 0 \leq j \leq N \tag{16}$$

Show that with a suitable definition of the remaining components of g_j, q_j one can find the solution for f by solving the equations

$$\frac{g_{j+1} + g_{j-1} - 2g_j}{a^2} = -q_j, \qquad 0 \leq j \leq 2N-1 \tag{17}$$

by means of the Fourier transform, as done in part 1), and then setting $f_j = g_j$ for $0 \leq j \leq N$.

Hint for part 2: You may find it convenient to think of g and q as being defined over $-N \leq j \leq N$, with $g_{-N} = g_N = 0$ and $q_{-N} = q_N = 0$ in agreement with periodic boundary conditions, and impose then suitable symmetry properties under $j \to -j$.

Solution

1) Let the Fourier transforms of f and p be F and P, respectively. In Fourier space Eq. 10 becomes

$$\lambda_k F_k = P_k, \qquad 0 \leq k \leq N-1 \tag{18}$$

with

$$\lambda_k = \frac{2 - 2\cos(2\pi k/N)}{a^2} \tag{19}$$

In particular we have

$$\lambda_0 = 0 \tag{20}$$

so Eq. 19 with $k = 0$ admits a solution only if $P_0 \propto \sum p_j = 0$, which is satisfied by the assumed form of p. F_0 is then unconstrained, but the condition $\sum_j f_j = 0$ fixes $F_0 = 0$. All the other equations are solved by

$$F_k = \frac{P_k}{\lambda_k}, \qquad 1 \leq k \leq N-1 \tag{21}$$

Thus the solution procedure is the following. Perform the Fourier transform of p_j to find P_k, set $F_0 = 0$ and all other F_k equal to the values given by Eq. 21, and finally perform the inverse Fourier transform of F_k to find f_j.

2) We define

$$q_j = p_j, \quad q_{-j} = -p_j, \qquad j = 0 \text{ to } N \tag{22}$$

and assume periodic boundary conditions with period $2N$. By construction $\sum_j q_j = 0$. We solve

$$\frac{g_{j+1} + g_{j-1} - 2g_j}{a^2} = -q_j, \qquad 0 \leq j \leq 2N-1 \tag{23}$$

by using the Fourier transform as described in 1). The symmetry of q ($q_{-j} = -q_j$) will be respected by the solution and so we will have

$$g_{-j} = -g_j \tag{24}$$

This implies in particular $g_0 = 0$ and $g_N = g_{-N} = 0$ and thus

$$f_j = g_j, \qquad j = 0, N \tag{25}$$

will satisfy the equations and boundary conditions in part 2) of the problem.

Problem 3

Write a program which finds the prime numbers between 2 and N=100 by applying the "sieve of Eratosthenes". In this procedure one lists first all the numbers between 2 and N and one begins by removing from the list all the multiples of 2. One moves then to the next number in the list, i.e. 3, and removes all the multiples of 3. One moves to the next number remaining in the list, 5, and removes all the multiples of 5, and so on. Eventually the only numbers remaining in the list are the primes from 2 on. You may write your program in Fortran, C or C++. You must make sure that it compiles on the workstations in the classroom.

Hint: Declare a logical array `prime` with an index `i` ranging from 2 to N. `prime(i)=.TRUE.` means that the number `i` is prime, `prime(i)=.FALSE.` that it is not. (Or, equivalently, declare an integer array `prime` with the convention that `prime==1` means that `i` is prime, `prime==0` means that `i` is not prime.) Set initially `prime=.TRUE.` (or `prime=1`) and proceed through the elements `prime(i)` starting from `i=2`. When `prime(i)` is `.TRUE.` (or `1`) the code should print out the number `i` and set `prime(i)=.FALSE.` (or `prime(i)=0`) for all the multiples of `i`.

Solution

```
PROGRAM midterm1

  IMPLICIT NONE

  INTEGER, PARAMETER :: N=100
  LOGICAL, DIMENSION(2:N) :: prime
  INTEGER :: i,j

  prime=.TRUE.

  DO i=2,N
```

```
    IF(prime(i)) THEN
       PRINT *,i
       DO j=2*i,N,i
          prime(j)=.FALSE.
       END DO
    END IF
 END DO

END PROGRAM midterm1

END PROGRAM midterm1
```