PY 421 - Introduction to Computational Physics

Second midterm exam.

April 9, 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 3 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 3 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 2 requires that you complete a program. The program to be completed is in the file

~rebbi/courseware/code/midterm2.f90

on the CAS cluster, and you should copy the completed program, always on the CAS cluster, to the file

~rebbi/courseware/asgn/midterm2.xxyyyy

where xxyyyy stands, as usual, for your personal identifier.

Problem 1

`a`, `b`, and `c` are arrays with dimension `1:N`. Consider the following code segments:

A:

```
DO i=1,N-1
   a(i)=a(i+1)+b(i)
END DO
```

B:

```
c(1:N-1)=a(2:N)
a(1:N-1)=b(1:N-1)+c(1:N-1)
```

C:

```
DO i=2,N
   a(i)=a(i-1)+b(i)
END DO
```

D:

```
c(2:N)=a(1:N-1)
a(2:N)=b(2:N)+c(2:N)
```

Are the codes in A: and B: equivalent? (i.e. do they produce the same values in `a` after execution)? Are the codes in C: and D: equivalent?

Hint: Denote by $\tilde{a}$ the values of the a array in memory before execution and implement the first few steps in the codes.

Solution

Proceeding as suggested by the hint, we find for A:

$$\begin{array}{lrcl} i=1: & a(1) & = & \tilde{a}(2)+b(1) \\ i=2: & a(2) & = & \tilde{a}(3)+b(2) \\ & & \dots & \end{array} \tag{1}$$

and for B:

$$c(1)=\tilde{a}(2), \quad c(2)=\tilde{a}(3)\dots \tag{2}$$

followed by

$$a(1)=c(1)+b(1)=\tilde{a}(2)+b(1), \quad a(2)=c(2)+b(2)=\tilde{a}(3)+b(2)\dots \tag{3}$$

and we see that the final results of the two code segments are the same.

For C: we find

$$\begin{array}{lrcl} i=1: & a(2) & = & \tilde{a}(1)+b(2) \\ i=2: & a(3) & = & a(2)+b(2)=\tilde{a}(1)+b(1)+b(2) \\ & & \dots & \end{array} \tag{4}$$

(notice in particular that the final values in the a array only depend on $\tilde{a}(1)$ and the values in b.)

For D: we find

$$c(2) = \tilde{a}(1), \quad c(3) = \tilde{a}(2)\ldots \tag{5}$$

followed by

$$a(2) = c(2) + b(2) = \tilde{a}(1) + b(2), \quad a(3) = c(3) + b(3) = \tilde{a}(2) + b(3)\ldots \tag{6}$$

and we see that after $a(2)$ the final values produced by the two codes for the other elements of a are different.

Problem 2

The program `midterm2.f90` solves the equation

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

where $f(x)$ is the function to be determined, $k(x) = 1 + 0.2\cos(2\pi x/L)$ and $b(x)$ are given functions, and all functions are defined over $0 \leq x \leq L = 256$ with periodic boundary conditions. The equation is solved by the Gauss-Jacobi relaxation method after the x-axis has been discretized by subdividing it into $N = 256$ subintervals of width $a = 1$. The program `midterm2.f90` is posted the the ~rebbi/courseware/code/ directory and is reproduced below. For this problem you should insert into the code the instructions necessary to implement a two-grid method for the solution. The program `midterm2.f90` is already set-up for a two-level implementation, with the numbers of cycles and the numbers of iterations on the fine and coarse lattices already defined in the code to simplify the programming task. The program will compile and execute as posted, but it will only perform iterations on the fine level, whether the user asks for a single level or a two level algorithm. You should complete the program so that it can properly execute a two level algorithm.

Solution

```
PROGRAM midterm2

! This program solves the equation
```

```
! (2*f(i)-f(i+1)-f(i-1))/a**2 + k(i)*f(i) = b(i)

! where periodic boundary conditions are assumed, either by
! straightforward Gauss-Jacobi relaxation, or by a two-level
! implementation of the relaxation method,

  IMPLICIT NONE
  INTEGER, PARAMETER :: REAL8=SELECTED_REAL_KIND(15,300)
  INTEGER, PARAMETER :: N=256
  REAL(REAL8), PARAMETER :: PI=3.1415926535898_REAL8

  REAL(REAL8), DIMENSION(0:N-1) :: f  ! field to be determined
  REAL(REAL8), DIMENSION(0:N-1) :: k  ! position dependent term
                                      ! in the equation
  REAL(REAL8), DIMENSION(0:N-1) :: b  ! given r.h.s. of the equation
  REAL(REAL8), DIMENSION(0:N-1) :: r  ! residue on the fine lattice
  REAL(REAL8), DIMENSION(0:N-1) :: e  ! error, or correction, on the
                                      ! fine lattice
  REAL(REAL8) :: a ! the lattice spacing of the fine lattice
  REAL(REAL8) :: rmax ! maximum residue on the fine lattice


  REAL(REAL8), DIMENSION(0:N/2-1) :: kc  ! position dependent term
                                         ! projected over the
                                         ! coarse lattice
  REAL(REAL8), DIMENSION(0:N/2-1) :: rc  ! residue on the coarse
                                         ! lattice
  REAL(REAL8), DIMENSION(0:N/2-1) :: ec  ! error on the coarse lattice
  REAL(REAL8) :: ac ! the lattice spacing of the fine lattice

  INTEGER :: ncycles,nfineit,ncoarseit ! number of cycles, number
                                       ! pf relaxation iterations
                                       ! on the fine lattice, number
                                       ! of relaxation iterations on
                                       ! the coarse lattice
  INTEGER :: cycle,fineit,coarseit ! loop parameters for the above
  INTEGER :: nlevel ! 1 for a single level relaxation, 2 for two
                    ! level relaxation
```

```
INTEGER :: i

a=1
ac=2
f=0

! some arbitrary r.h.s. for the equation
b=0
b(N/2)=100
b(N/2-1)=50
b(N/2+1)=50

DO i=0,N-1
   k(i)=1+cos((2*PI*i)/N)/5
END DO

! project k onto the coarse lattice, this only needs
! to be done once
kc=(k(0:N-1:2)+k(1:N-1:2))/2

WRITE(*,'("Enter 1 for a single level algorithm, 2 for &
     &a 2 level algorithm, any other integer to exit: ")'&
     ,ADVANCE='NO')
READ *,nlevel
IF(nlevel/=1.AND.nlevel/=2) STOP

ncycles=10
nfineit=3
ncoarseit=10

DO cycle=1,ncycles

   ! perform Gauss-Jacobi relaxations on the fine lattice
   DO fineit=1,nfineit
      f=((CSHIFT(f,1)+CSHIFT(f,-1))/a**2+b)/(2/a**2+k)
   END DO

   IF (nlevel==2) THEN
```

```
! Insert here the code for the two level implementation:

        ! calculate the residue
        r=b-(2*f-CSHIFT(f,1)-CSHIFT(f,-1))/a-k*f

        ! project over the coarse lattice
        rc=(r(0:N-1:2)+r(1:N-1:2))/2

        ! perform Gauss-Jacobi relaxations on the coarse lattice
        ec=0
        DO coarseit=1,ncoarseit
           ec=((CSHIFT(ec,1)+CSHIFT(ec,-1))/ac**2+rc)/(2/ac**2+kc)
        END DO

        ! interpolate to the fine lattice and add to f
        e(0:N-1:2)=ec
        e(1:N-1:2)=ec
        f=f+e

     ENDIF

     ! perform again Gauss-Jacobi relaxations on the fine lattice
     DO fineit=1,nfineit
        f=((CSHIFT(f,1)+CSHIFT(f,-1))/a**2+b)/(2/a**2+k)
     END DO

     ! calculate the residue and print its maximum value
     r=b-(2*f-CSHIFT(f,1)-CSHIFT(f,-1))/a-k*f

     rmax=MAXVAL(ABS(r))
     PRINT '("At cycle ",I3," max. residue=",F14.9)',cycle,rmax

  END DO

END PROGRAM midterm2
```

Problem 3

Let A and B be two non-commuting matrices. For the solution of the time-dependent Schrödinger equation we made use of the identity

$$e^{\epsilon(A+B)} = e^{\epsilon A} e^{\epsilon B} + O(\epsilon^2) \tag{8}$$

Prove that the more symmetric formula

$$e^{\epsilon(A+B)} \approx e^{\epsilon A/2} e^{\epsilon B} e^{\epsilon A/2} \tag{9}$$

is approximate to order ϵ^3, i.e. prove the identity

$$e^{\epsilon(A+B)} = e^{\epsilon A/2} e^{\epsilon B} e^{\epsilon A/2} + O(\epsilon^3) \tag{10}$$

Hint: Expand the exponentials to the appropriate order, being very careful with the order of the factors.

Solution

We expand the exponentials in Eq. 9 keeping terms up to order ϵ^2. For the l.h.s. we find

$$\begin{gathered} e^{\epsilon(A+B)} = I + \epsilon(A+B) + \frac{\epsilon^2}{2}(A+B)^2 + O(\epsilon^3) = \\ I + \epsilon A + \epsilon B + \frac{\epsilon^2}{2}A^2 + \frac{\epsilon^2}{2}AB + \frac{\epsilon^2}{2}BA + \frac{\epsilon^2}{2}B^2 + O(\epsilon^3) \end{gathered} \tag{11}$$

where I stands for the identity matrix. For the r.h.s. we find

$$\begin{aligned} e^{\epsilon A/2} e^{\epsilon B} e^{\epsilon A/2} = \Big(I + \frac{\epsilon A}{2} + \frac{\epsilon^2 A^2}{8} + O(\epsilon^3)\Big)\Big(I + \epsilon B + \frac{\epsilon^2 B^2}{2} + O(\epsilon^3)\Big) \times \\ \Big(I + \frac{\epsilon A}{2} + \frac{\epsilon^2 A^2}{8} + O(\epsilon^3)\Big) = I + \epsilon\Big(\frac{A}{2} + B + \frac{A}{2}\Big) + \\ \epsilon^2\Big(\frac{A}{2} \times \frac{A}{2} + \frac{A}{2}B + B\frac{A}{2} + \frac{A^2}{8} + \frac{B^2}{2} + \frac{A^2}{8}\Big) + O(\epsilon^3) = \\ I + \epsilon A + \epsilon B + \frac{\epsilon^2}{2}A^2 + \frac{\epsilon^2}{2}AB + \frac{\epsilon^2}{2}BA + \frac{\epsilon^2}{2}B^2 + O(\epsilon^3) \end{aligned} \tag{12}$$

in full agreement with the l.h.s.