TFY4235/FY8904 Computational Physics, Spring 2013

Solution Set 4

Problem 1.

Here is a conjugate gradient program. It uses 199 iterations to reach the solution with a presicion of 2×10^{-4}. Rather than using the conjugate gradient version taught in class and listed in *Numerical Recipes*. Rather, I have used a version that is described in Batrouni and Hansen, Journal of Statistical Physics, **52**, 747 (1988) — see equations (32)–(38) there. This version of the conjugate gradient algorithm is the most efficient for this class of problem.

```
      program cg
c Conjugate gradient
      dimension volt(0:1000),cond(0:999)
      dimension p(0:1000),r(999),ap(999)
c Random number initializing
      rinv=0.5/2147483647.
      ibm=1927
      do i=1,1000
      ibm=ibm*16807
      enddo
c Genering the 1000 conductances
      do i=0,999
      ibm=ibm*16807
      cond(i)=ibm*rinv+0.5
      enddo
c Stopping criterion
      pres=2.e-4
c Initializing potentials
      do i=0,1000
      volt(i)=float(i)/1000
      enddo
c Initializing work vectors
c Here is the only place in the routine where b
c enters the calculation
      do i=1,999
      p(i)=-cond(i-1)*(volt(i-1)-volt(i))-cond(i)*(volt(i+1)-volt(i))
      r(i)=p(i)
      enddo
      p( 0)=0.
      p(1000)=0.
c Iteration
      do ite=1,2*1000
```

```
      rps=0.
      do i=1,999
      rps=rps+r(i)*r(i)
      enddo
      if(sqrt(rps).le.pres) goto 200
      do i=1,999
      ap(i)=cond(i-1)*(p(i-1)-p(i))+cond(i)*(p(i+1)-p(i))
      enddo
      am=0.
      do i=1,999
      am=am+p(i)*ap(i)
      enddo
      am=rps/am
      do i=1,999
      volt(i)=volt(i)+am*p(i)
      enddo
      do i=1,999
      r(i)=r(i)-am*ap(i)
      enddo
      rpn=0.
      do i=1,999
      rpn=rpn+r(i)*r(i)
      enddo
      bm=rpn/rps
      do i=1,999
      p(i)=r(i)+bm*p(i)
      enddo
      enddo
c Done
200   continue
      write(*,*) ite
      end
```

It is worth noticing that in the conjugate gradient algorithm the vector b is only used during initialization. See how it is done in the program. (It is not obvious, so look very carefully!)

You should note that the conjugate gradient algorithm has the peculiarity that even if there are bugs in the program, it may still find the solution. However, the convergence will be terrible. Hence, if it seemingly works but it is slow, there are bugs in the program.