NTNU
Institutt for fysikk

TFY4235/FYS8904
Solution problemset 5 Spring 2015

Problem 1.
We present here an "all-in-one" program that writes out (1) a histogram of the largest eigenvalues, a histogram of the smallest eigenvalues and the cumulative density of states (DOS). This last quantity is the number of eigenvalues smaller than a chosen value.

In order to find the largest eigenvalues, we use the simple iterative algorithm given in the lectures: $x'_k = Ax_{k-1}$ og $x_k = x'_k/|x'_k|$. The largest eigenvalue is then given by $\lambda_m = x_{k+1} \cdot x'_k$.

However, there is a problem. The matrix A has its entries symmetrically distributed about zero. Sometimes the smallest eigenvalue — the most negative — is the largest one in absolute value. The algorithm picks out the eigenvalue with the largest absolute value, try e.g., the matrix diag$[-1, 1/2]$. (But, with a caveat: See below.) We solve this problem by simply adding a positive constant to the diagonal of the matrix A, and subtracting it from the eigenvalue when done.

By considering *symmetric* matrices, we avoid complex eigenvalues and the problems of having to work with complex eigenvectors.

In order to find the smallest eigenvalue, we define a new matrix

$$A' = \lambda_m - A \ , \tag{1}$$

and repeat. The smallest eigenvalue in A now becomes the largest eigenvalue in A'.

Lastly, we use the Lambert-Weaire algorithm to map out the distribution of eigenvalues between the smallest and largest eigenvalues of each A. The way I have done it is to choose 100 equally-spaced values for λ and run the Lambert-Weaire algorithm for each. When averaging over many samples, I have lumped together the results for each sample in the corresponding equally-spaced bin. That is, I have lumped the number of eigenvalues less than the 50th λ for sample number one together with the number of eigenvalues less than the 50th λ for sample number two even though they have different largest and smallest eigenvalues. In the end, I have used the *average* smallest and largest eigenvalues to scale the λ-values so that λ number one corresponds to the average smallest eigenvalue and λ number 100 to the average largest eigenvalue.

The Fortran version of the solution is included here. An equivalent implementation in C is included in the *Code* subdirectory.

Listing 1: ranmat.f

```
      program ranmat
c-------------------------------------------------------
c Finding the smallest and largest eigenvalues in random
c matrices based on iteration.
```

```
c Then it uses the Lambert-Weaire algorithm to map the
c distribution of eigenvalues in between.
c-----------------------------------------------------
c       n = size of matrix
c   nsamp = number of samples
c   itemx = max. number of iterations
c      nh = number of bins in histograms of largest
c           and smallest eigenvalues
c  egenmx = value of largest bin in histograms
c  egenmn = value of smallest bin in histograms
c     nlw = number of bins in Lambert-Weaire calc.
c
      parameter(n=100,nsamp=2000,itemx=2000)
      parameter(nh=100,egenmx=20.,egenmn=-20.)
      parameter(nlw=100)
c-----------------------------------------------------
      dimension rmat(n,n),smat(n,n),qmat(n,n),vec(n),vecp(n)
      dimension nhmx(nh),nhmn(nh),nhlw(nlw)
c-----------------------------------------------------
c Opening files to store output data
c
      open(unit=1,file='ranmat_histogram.dat',status='unknown')
      open(unit=2,file='ranmat_dos.dat',status='unknown')
c-----------------------------------------------------
c Initializing random number generator
c
      rinv=1./2147483647.
      ibm=47113321
      do i=1,1000
      ibm=ibm*16807
      enddo
c-----------------------------------------------------
c Initializing histograms
c
c nhmx = histogram for largest eigenvalues.
c nhmn = histogram for smallest eigenvalues.
c
      do ih=1,nh
      nhmn(ih)=0
      nhmx(ih)=0
      enddo
c
c egen = bin size
c
      egen=(egenmx-egenmn)/(nh-1)
c
      avelamsm=0.
      avelambg=0.
c-----------------------------------------------------
c Initializing Lambert-Weaire result vector
c
      do ilw=1,nlw
      nhlw(ilw)=0
```

```
      enddo
c-------------------------------------------------------
c Loop over samples
c
      do isamp=1,nsamp
c-------------------------------------------------------
c Generating the matrix
c
c rmat = random matrix
c
      do i=1,n
      do j=1,i
      ibm=ibm*16807
      rmat(i,j)=ibm*rinv
      rmat(j,i)=rmat(i,j)
      enddo
      enddo
c-------------------------------------------------------
c A necessary trick to stabilize the iteration:
c
      do i=1,n
      rmat(i,i)=rmat(i,i)+1.
      enddo
c-------------------------------------------------------
c Determining the largest eigenvalue
c
c Initializing the eigenvectors vec.
c
      do i=1,n
      vec( i)=1.
      enddo
c-------------------------------------------------------
c The iteration:
c
c biglam = biggest eigenvalue
c
      do ite=1,itemx
c
      do i=1,n
      vecp(i)=0.
      do j=1,n
      vecp(i)=vecp(i)+rmat(i,j)*vec(j)
      enddo
      enddo
c
      sum=0.
      do i=1,n
      sum=sum+vecp(i)*vecp(i)
      enddo
      sum=1./sqrt(sum)
c
      do i=1,n
      vec(i)=vecp(i)*sum
```

```
      enddo
c
      biglam=0.
      do i=1,n
      biglam=biglam+vec(i)*vecp(i)
      enddo
c
      biglam=biglam-1.
c

      enddo
c-----------------------------------------------------------
      do i=1,n
      rmat(i,i)=rmat(i,i)-1.
      enddo
c-----------------------------------------------------------
c Histogram over biggest eigenvalue
c
      ih=int((biglam-egenmn)/egen)
      ih=max(ih,1)
      ih=min(ih,nh)
      nhmx(ih)=nhmx(ih)+1
c
      avelambg=avelambg+biglam
c----------------------------------------------------------
c smalam = smallest eigenvalue
c
c Constructing new iteration matrix
c
      avelam=egenmx
c
      do i=1,n
      do j=1,n
      smat(i,j)=-rmat(i,j)
      enddo
      enddo
c
      do i=1,n
      smat(i,i)=smat(i,i)+biglam
      enddo
c
      do i=1,n
      vec( i)=1.
      enddo
c
      do ite=1,itemx
c
      do i=1,n
      vecp(i)=0.
      do j=1,n
      vecp(i)=vecp(i)+smat(i,j)*vec(j)
      enddo
      enddo
```

```
c
      sum=0.
      do i=1,n
      sum=sum+vecp(i)*vecp(i)
      enddo
      sum=1./sqrt(sum)
c
      do i=1,n
      vec(i)=vecp(i)*sum
      enddo
      smalam=0.
      do i=1,n
      smalam=smalam+vec(i)*vecp(i)
      enddo
c
      smalam=biglam-smalam
c
      enddo
c------------------------------------------------------------
c Histogram over smallest  eigenvalue
c
      ih=int((smalam-egenmn)/egen)
      ih=max(ih,1)
      ih=min(ih,nh)
      nhmn(ih)=nhmn(ih)+1
c
      avelamsm=avelamsm+smalam
c------------------------------------------------------------
c The Lambert-Weaire algorithm
c
      egan=(biglam-smalam)/nlw
c
c Loop over lambda-values
c
      do ilw=1,nlw
c
      alam=smalam+egan*ilw
c
      do i=1,n
      do j=1,n
      smat(i,j)=rmat(i,j)
      enddo
      enddo
c
      nshift=1
c
      do k=n,2,-1
c
      ann=1./(smat(k,k)-alam)
      if(ann.lt.0.) nshift=nshift+1
c
      do i=1,k-1
      do j=1,k-1
```

```
      qmat(i,j)=smat(i,j)-smat(i,k)*smat(k,j)*ann
      enddo
      enddo
c
      do i=1,k-1
      do j=1,k-1
      smat(i,j)=qmat(i,j)
      enddo
      enddo
c
      enddo
c------------------------------------------------------------
c End of Lambert-Weaire iteration
c
      nhlw(ilw)=nhlw(ilw)+nshift
c
      enddo
c-----------------------------------------------------------
c End of loop over samples
c
      enddo
c-----------------------------------------------------------
c Writing the histograms
c
      egenv=egenmn
      do ih=1,nh
      egenv=egenv+egen
      write(1,*) egenv,nhmn(ih),nhmx(ih)
      enddo
c-----------------------------------------------------------
c Writing the DOS
c
      avelambg=avelambg/nsamp
      avelamsm=avelamsm/nsamp
c
      egan=(avelambg-avelamsm)/(nlw-1)
      do ilw=1,nlw
      alam=avelamsm+egan*(ilw-1)
      write(2,*) alam,float(nhlw(ilw))/nsamp
      enddo
c-----------------------------------------------------------
      close(1)
      close(2)
c-----------------------------------------------------------
      end
```

We show the histograms of the smallest and largest eigenvalues in figure 1.
The distribution of eigenvalues found with the Lambert-Weaire algorithm is shown in figure 2.
The data presented here was averaged over 16000 samples. I used 2000 iterations in the iterative algorithms. I checked that this was enough.

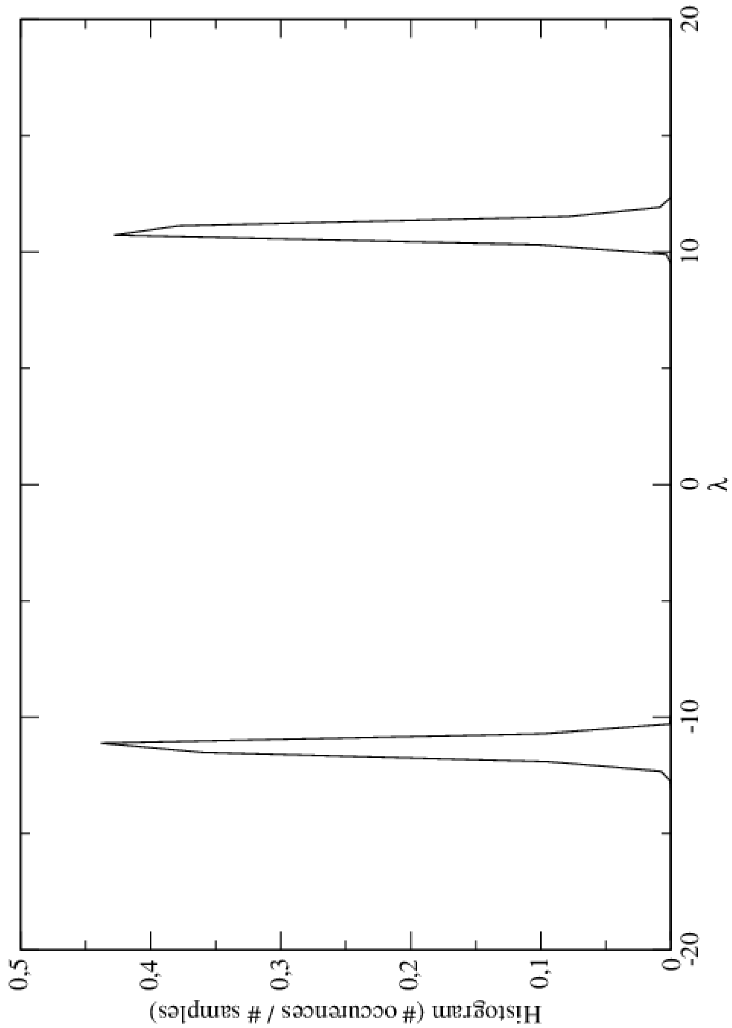

Figure 1: Histograms of the smallest and largest eigenvalues.

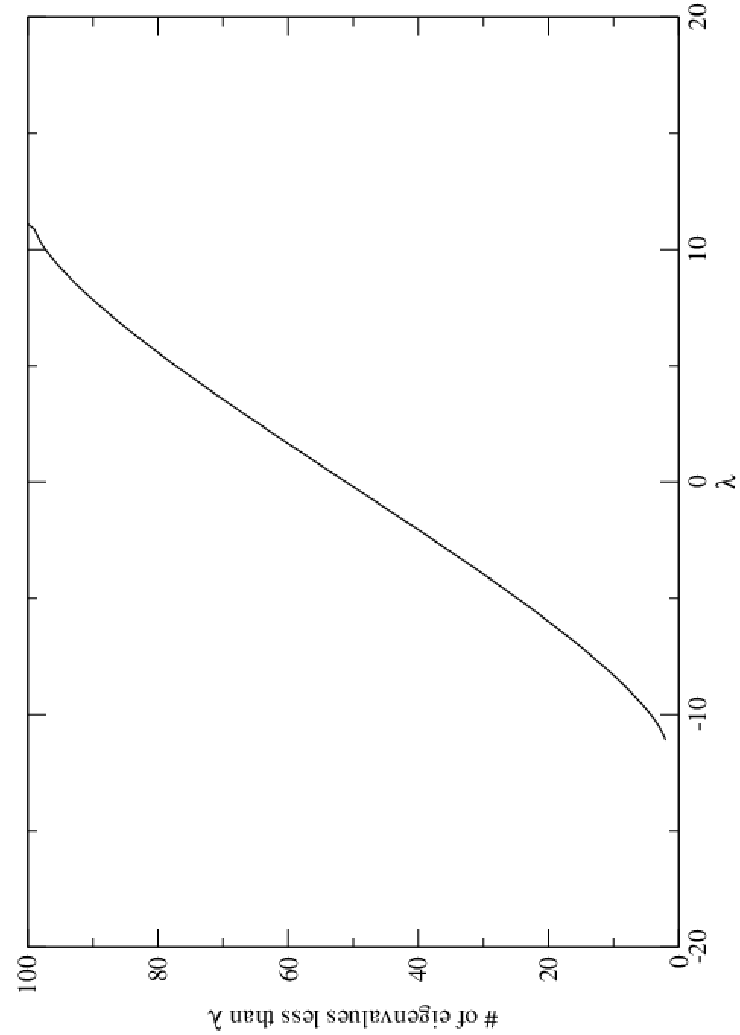

Figure 2: The distribution of eigenvalues found with the Lambert-Weaire algorithm.