TFY4235/FY8904 Computational Physics, Spring 2013

Solution Set 9

Problem 1.

The point of this problem is to set up a simple linear programming problem. We wish to find the length of a string of N spheres of different radii in contact. We define this length as the distance between the centers of sphere 1 and sphere N. We call this distance L and it is equal to $r_1 + 2\sum_{i=2}^{N-1} r_i + r_N$. In order to formulate this as a linear programming problem, we define the position of sphere i to be x_i. The object function is then

$$z = x_1 - x_N \ . \tag{1}$$

We have $N-1$ constraints

$$x_{i+1} - x_i \geq r_{i+1} + r_i \ . \tag{2}$$

We set up a tableau as shown in *Numerical Recipes,* page 430 (setting N=6):

		x_1	x_2	x_3	x_4	x_5	x_6
z	0	1	0	0	0	0	-1
y_1	$-(r_1+r_2)$	-1	1	0	0	0	0
y_2	$-(r_2+r_3)$	0	-1	1	0	0	0
y_3	$-(r_3+r_4)$	0	0	-1	1	0	0
y_4	$-(r_4+r_5)$	0	0	0	-1	1	0
y_5	$-(r_5+r_6)$	0	0	0	0	-1	1

with this tableau it is easy to implement the Kuenzi, Tzschach and Zehnder code of *Numerical Recipies* for this problem. Here is the resulting program:

```
      program simp
c Length of N spheres in a row calculated using the simplex meth.
      parameter(n=20,m=n-1,mp=m+2,np=n+1,m1=0,m2=m,m3=0)
      dimension r(n),a(mp,np),iposv(mp),izrov(np)
      ibm=4711
      do i=1,1000
      ibm=ibm*16807
      enddo
      rinv=0.5/(2.**31-1.)
      do i=1,n
      ibm=ibm*16807
      r(i)=rinv*float(ibm)+0.5
      enddo
```

```
c Correct answer
      dcorr=r(1)+r(n)
      do i=2,n-1
      dcorr=dcorr+2.*r(i)
      enddo
      do j=1,mp
      do i=1,np
      a(j,i)=0.
      enddo
      enddo
      a(1, 2)= 1.
      a(1,n+1)=-1.
      do j=2,m+1
      a(j,1)=r(j-1)+r(j)
      a(j,j )=-1.
      a(j,j+1)= 1.
      enddo
      call simplx(a,m,n,mp,np,m1,m2,m3,icase,izrov,iposv)
      dregn=a(1,1)
      write(*,*) dregn,dcorr
      end
c
c Subroutines from Numerical Recipes
c
      SUBROUTINE simplx(a,m,n,mp,np,m1,m2,m3,icase,izrov,iposv)
      INTEGER icase,m,m1,m2,m3,mp,n,np,iposv(m),izrov(n),MMAX,NMAX
      REAL a(mp,np),EPS
      PARAMETER (MMAX=100,NMAX=100,EPS=1.e-6)
CU USES simp1,simp2,simp3
      INTEGER i,ip,ir,is,k,kh,kp,m12,nl1,nl2,l1(NMAX),l2(MMAX),l3(MMAX)
      REAL bmax,q1
      if(m.ne.m1+m2+m3)pause 'bad input constraint counts in simplx'
      nl1=n
      do 11 k=1,n
      l1(k)=k
      izrov(k)=k
11    continue
      nl2=m
      do 12 i=1,m
      if(a(i+1,1).lt.0.)pause 'bad input tableau in simplx'
      l2(i)=i
      iposv(i)=n+i
12    continue
      do 13 i=1,m2
```

```
      l3(i)=1
13    continue
      ir=0
      if(m2+m3.eq.0)goto 30
      ir=1
      do 15 k=1,n+1
      q1=0.
      do 14 i=m1+1,m
      q1=q1+a(i+1,k)
14    continue
      a(m+2,k)=-q1
15    continue
10    call simp1(a,mp,np,m+1,l1,nl1,0,kp,bmax)
      if(bmax.le.EPS.and.a(m+2,1).lt.-EPS)then
      icase=-1
      return
      else if(bmax.le.EPS.and.a(m+2,1).le.EPS)then
      m12=m1+m2+1
      if(m12.le.m)then
      do 16 ip=m12,m
      if(iposv(ip).eq.ip+n)then
      call simp1(a,mp,np,ip,l1,nl1,1,kp,bmax)
      if(bmax.gt.0.)goto 1
      endif
16    continue
      endif
      ir=0
      m12=m12-1
      if(m1+1.gt.m12)goto 30
      do 18 i=m1+1,m12
      if(l3(i-m1).eq.1)then
      do 17 k=1,n+1
      a(i+1,k)=-a(i+1,k)
17    continue
      endif
18    continue
      goto 30
      endif
      call simp2(a,m,n,mp,np,l2,nl2,ip,kp,q1)
      if(ip.eq.0)then
      icase=-1
      return
      endif
1     call simp3(a,mp,np,m+1,n,ip,kp)
```

```
      if(iposv(ip).ge.n+m1+m2+1)then
      do 19 k=1,nl1
      if(l1(k).eq.kp)goto 2
19    continue
2     nl1=nl1-1
      do 21 is=k,nl1
      l1(is)=l1(is+1)
21    continue
      else
      if(iposv(ip).lt.n+m1+1)goto 20
      kh=iposv(ip)-m1-n
      if(l3(kh).eq.0)goto 20
      l3(kh)=0
      endif
      a(m+2,kp+1)=a(m+2,kp+1)+1.
      do 22 i=1,m+2
      a(i,kp+1)=-a(i,kp+1)
22    continue
20    is=izrov(kp)
      izrov(kp)=iposv(ip)
      iposv(ip)=is
      if(ir.ne.0)goto 10
30    call simp1(a,mp,np,0,l1,nl1,0,kp,bmax)
      if(bmax.le.0.)then
      icase=0
      return
      endif
      call simp2(a,m,n,mp,np,l2,nl2,ip,kp,q1)
      if(ip.eq.0)then
      icase=1
      return
      endif
      call simp3(a,mp,np,m,n,ip,kp)
      goto 20
      END
      SUBROUTINE simp1(a,mp,np,mm,ll,nll,iabf,kp,bmax)
      INTEGER iabf,kp,mm,mp,nll,np,ll(np)
      REAL bmax,a(mp,np)
      INTEGER k
      REAL test
      kp=ll(1)
      bmax=a(mm+1,kp+1)
      do 11 k=2,nll
      if(iabf.eq.0)then
```

```
      test=a(mm+1,ll(k)+1)-bmax
      else
      test=abs(a(mm+1,ll(k)+1))-abs(bmax)
      endif
      if(test.gt.0.)then
      bmax=a(mm+1,ll(k)+1)
      kp=ll(k)
      endif
11    continue
      return
      END
      SUBROUTINE simp2(a,m,n,mp,np,l2,nl2,ip,kp,q1)
      INTEGER ip,kp,m,mp,n,nl2,np,l2(mp)
      REAL q1,a(mp,np),EPS
      PARAMETER (EPS=1.e-6)
      INTEGER i,ii,k
      REAL q,q0,qp
      ip=0
      do 11 i=1,nl2
      if(a(l2(i)+1,kp+1).lt.-EPS)goto 1
11    continue
      return
1     q1=-a(l2(i)+1,1)/a(l2(i)+1,kp+1)
      ip=l2(i)
      do 13 i=i+1,nl2
      ii=l2(i)
      if(a(ii+1,kp+1).lt.-EPS)then
      q=-a(ii+1,1)/a(ii+1,kp+1)
      if(q.lt.q1)then
      ip=ii
      q1=q
      else if (q.eq.q1) then
      do 12 k=1,n
      qp=-a(ip+1,k+1)/a(ip+1,kp+1)
      q0=-a(ii+1,k+1)/a(ii+1,kp+1)
      if(q0.ne.qp)goto 2
12 continue
2 if(q0.lt.qp)ip=ii
      endif
      endif
13    continue
      return
      END
      SUBROUTINE simp3(a,mp,np,i1,k1,ip,kp)
```

```
      INTEGER i1,ip,k1,kp,mp,np
      REAL a(mp,np)
      INTEGER ii,kk
      REAL piv
      piv=1./a(ip+1,kp+1)
      do 12 ii=1,i1+1
      if(ii-1.ne.ip)then
      a(ii,kp+1)=a(ii,kp+1)*piv
      do 11 kk=1,k1+1
      if(kk-1.ne.kp)then
      a(ii,kk)=a(ii,kk)-a(ip+1,kk)*a(ii,kp+1)
      endif
11    continue
      endif
12    continue
      do 13 kk=1,k1+1
13    continue
      a(ip+1,kp+1)=piv
      return
      END
```

Problem 2.

It is not obvious that finding the ground state energy of a system with Hamiltonian

$$H = -a_1\phi_1 - a_2\phi_1\phi_2 - \cdots - a_N\phi_1\cdots\phi_N \tag{3}$$

where $0 \leq \phi_i \leq 1$ is a linear programming problem. However, doing the change of variable

$$\begin{aligned} \psi_1 &= \phi_1 \ , \\ \psi_2 &= \phi_1\phi_2 \ , \\ \psi_3 &= \phi_1\phi_2\phi_3 \ , \\ &\ldots \\ \psi_N &= \phi_1\cdots\phi_N \ , \end{aligned} \tag{4}$$

the Hamiltonian may be written

$$H = -a_1\psi_1 - a_2\psi_2 - \cdots - a_N\psi_N \ . \tag{5}$$

This transformation has introduced a series of constraints between the coordinates,

$$\begin{aligned} &\psi_1 \geq \psi_2 \ , \\ &\psi_2 \geq \psi_3 \ , \\ &\cdots \\ &\psi_{N-1} \geq \psi_N \ , \end{aligned} \tag{6}$$

i addition to

$$\begin{aligned} &\psi_1 \leq 1 \,, \\ &\psi_2 \leq 1 \,, \\ &\cdots \\ &\psi_N \leq 1 \,. \end{aligned} \tag{7}$$

The linear programming problem thus consists in fonding the maximum of the function $z = -H$, where H is defined in (5) given the constraints (6) and (7).
We construct a tableau as done in Problem 1 (assuming the $N = 6$):

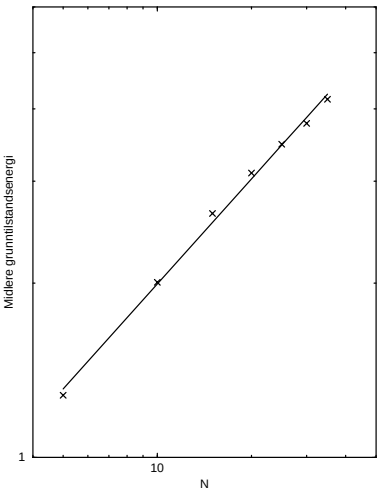
		ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6
z	0	$-a_1$	$-a_2$	$-a_3$	$-a_4$	$-a_5$	$-a_6$
z_1	1	-1	0	0	0	0	1
z_2	1	0	-1	0	0	0	0
z_3	1	0	0	-1	0	0	0
z_4	1	0	0	0	-1	0	0
z_5	1	0	0	0	0	-1	0
z_6	1	0	0	0	0	0	-1
z_7	0	-1	1	0	0	0	1
z_8	0	0	-1	1	0	0	0
z_9	0	0	0	-1	1	0	0
z_{10}	0	0	0	0	-1	1	0
z_{11}	0	0	0	0	0	-1	1

z_1 to z_6 in this tableau constitute constraints (7) and z_7 to z_{11} constitute constraints (6).
Here is the program implementing this:

```
program prot
c Minimum energy of a hierarchic protein model
parameter
c (n=25,np=n+1,m1=n,m2=n-1,m3=0,m=m1+m2+m3,mp=m+2,nsamp=1000)
dimension c(n),a(mp,np),iposv(mp),izrov(np)
ibm=5511
do i=1,1000
ibm=ibm*16807
enddo
rinv=0.5/(2.**31-1.)
emin=0.
do isamp=1,nsamp
do i=1,n
ibm=ibm*16807
c(i)=1.-2.*int(1.+rinv*float(ibm))
enddo
```

```
do j=1,mp
do i=1,np
a(j,i)=0.
enddo
enddo
do i=2,n+1
a(1,i)=-c(i-1)
enddo
do j=2,1+m1
a(j,1)= 1.
a(j,j)=-1.
enddo
do j=2+m1,1+m1+m2
a(j,j-m1 )=-1.
a(j,j-m1+1)= 1.
enddo
call simplx(a,m,n,mp,np,m1,m2,m3,icase,izrov,iposv)
emin=emin-a(1,1)
enddo
write(*,*) n,emin/nsamp
end
```

The subroutines called here are the same as in Problem 1, so I do not repeat them.
I generate 1000 samples for $N = 5$, 10, 15, 20, 25, 30, 35, and 40. Average ground state energy as a function of N is shown in this figure:



The straight line is $E_0 \propto N^{0.6}$. This data series is too small (but took about 10 minutes on my prehistoric SUN Sparc5) to draw any definite conclusions about the dependency of the ground state on N except that it goes to zero per degree of freedom as $N \to \infty$.