F77 Bond-order listing

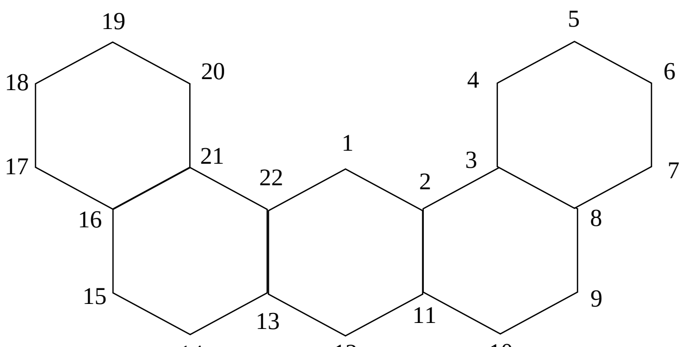
The diagrams on the following page give the structures of two dibenzanthracenes and two dibenzacridines with the atom-numberings used in this calculation. The system studied in R. Mason, *Nature*, **179**, 465–467 (1957) and for which these calculations were done, is in the middle of the page, on the left-hand side.

The numbering here is non-standard but with adjacent atoms carrying consecutive numbers, it is simpler for the programme to find most of the bonds in the system.

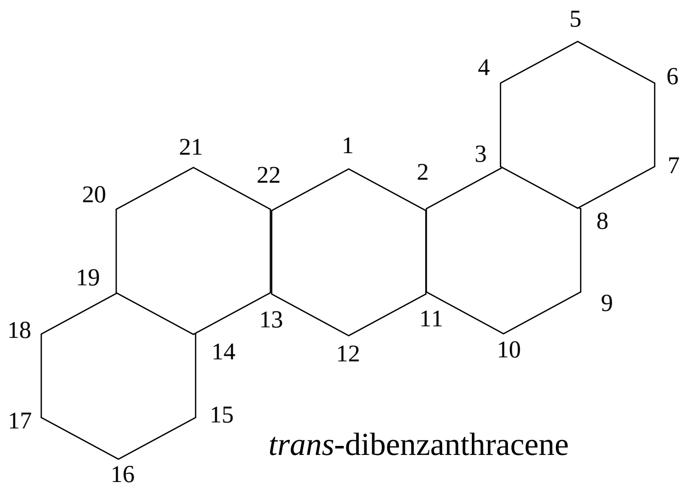
The input file for this calculation for this pair of similar molecules is:

```
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1
1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
```

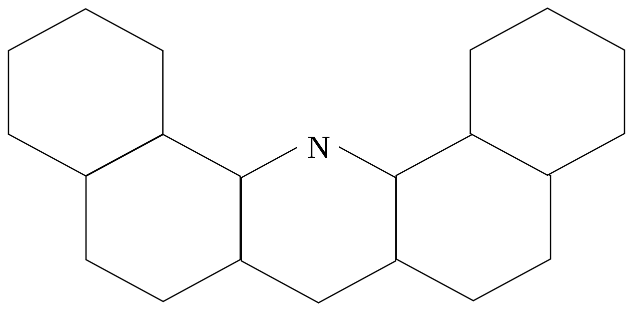
cis-dibenzanthracene and *cis*-dibenzacridene input file



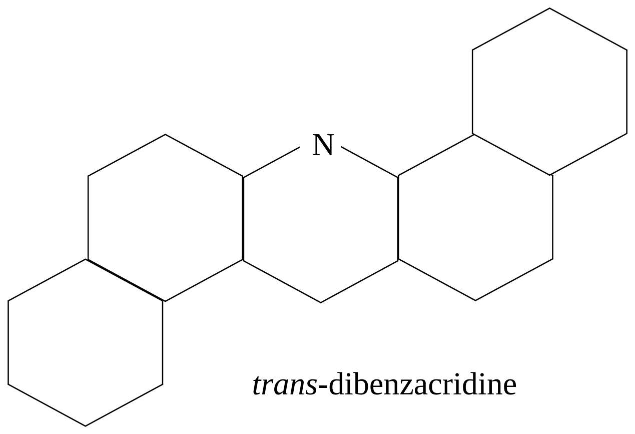
cis-dibenzanthracene



trans-dibenzanthracene



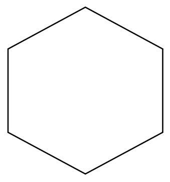
cis-dibenzacridine



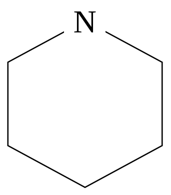
trans-dibenzacridine

Shorthand convention: a single line is a bond between 2 atoms; an intersection of 2 or 3 lines represents a carbon atom (C); if carbon is replaced by another atom, it is printed at the intersection, e.g. N for nitrogen in the acridines. All hydrogen atoms (H) are left unmarked -- there is an H at all binary intersections, but not at the tertiary ones. Thus, benzene (below) is C_6H_6 and pyridine (below) is C_5H_5N; likewise, dibenzanthracene is $C_{22}H_{14}$ and dibenzacridine is $C_{21}H_{13}N$.

The numbering scheme here is what was used in the Mark I programme, not that used by Mason in his 1957 Nature Letter. I use *cis*- and *trans*- as a (non-standard) shorthand to avoid confusion with different numbering systems.



benzene



pyridine

F77 Bond-order listing, continued

```
C       To read 22nd order Huckel matrix for Ron Mason's calculation
C       of bond orders (and polarisabilities).
C       As printed, this programme is for the pure hydrocarbon;
C       for acridines, introduce non-integer matrix element values,
C       as appropriate, just ahead of the jacobi call.

        implicit real*8 (A-H,O-Z)

        real*8   A(22,22),B(22,22),rlamda(22),pij(26)
        integer  Ai(22,22),list(5,2)

        zero=0.0d0
        one=1.0d0
        two=2.0d0

c       input determinant is all integers, i.e. hydrocarbon template:
        read(5,*) Ai
c       IF non-integer elements are required, modify where necessary:
c       print*,Ai
        k=1
        do i=1,22
          do j=1,22
            A(i,j)=Ai(i,j)
C       find where the non-consecutive numbered bonds are:
            if(j.gt.(i+1).and.Ai(i,j).eq.1) then
c       print*,i,j,k,Ai(i,j)
              list(k,1)=i
              list(k,2)=j
              k=k+1
            endif
          enddo
        enddo
```

```
c       print*,A
        print*,' '

c        do i=2,5
C       there are only 4, but 1,22 comes first, so ignore 1st entry
c          print*, (list(i,j),j=1,2)
c        enddo

C       to choose Mason's acridines instead of hydrocarbon template
C       {R.Mason, Nature, vol 179, pp 465-467 (1957)}
C       change all diagonal elements to Mason's values as follows:

        A(1,1)=0.5d0
        do i=2,22
          A(i,i)=0.1d0
        enddo
c       print*,A
c       print*,' '

        call jacobi(A,22,22,rlamda,B,nrot)

C       to sort eigenvalues in numerical order:
        call eigsrt(rlamda,B,22,22)

 11     format(1x,1p6d12.5)

        print*,'nrot = ',nrot
        print*,' '
        do i=1,22
          print*,rlamda(i)
          write(6,11) (B(j,i), j=1,22)
          print*,' '
        enddo

C       to calculate consecutive bond orders:
C       i and j are the atom numbers, l the energy level (eigenvalue)
        do i=1,26
```

```
          pij(i)=one
        enddo
        do i=1,22
          j=i+1
          if (i.eq.22) j=1
            do l=1,22
              if (rlamda(l).gt.zero) then
c       print*,i,j,l,rlamda(l),B(i,l),B(j,l)
                pij(i)=pij(i)+two*B(i,l)*B(j,l)
              endif
            enddo
        enddo

C       to calculate non-consecutively numbered bond orders:
        do k=2,5
          i=list(k,1)
          j=list(k,2)
          m=k+21
c        print*,k,i,j,m
          do l=1,22
            if (rlamda(l).gt.zero) then
c        print*,i,j,l,rlamda(l),B(i,l),B(j,l)
              pij(m)=pij(m)+two*B(i,l)*B(j,l)
            endif
          enddo
c       print*,m,i,j,pij(m)
        enddo

c       print*,pij
        print*,'Bond orders:'
        print*,' '
        do i=1,22
          j=i+1
          if (j.eq.23) j=1
          write(6,12) i,j,pij(i)
        enddo
        print*,' '
        do k=2,5
          i=list(k,1)
          j=list(k,2)
          m=k+21
          write(6,12) i,j,pij(m)
        enddo

 12        format(2I3,f7.3)

        stop
        end

C       the following subroutines are from "Numerical Recipes"
C       W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T.Vetterling,
C       Cambridge University Press, 1985.
C       except for declarations of implicit real*8 (a-h,o-z)

      subroutine jacobi(a,n,np,d,v,nrot)
          implicit real*8 (a-h,o-z)
      parameter (nmax=100)
      dimension a(np,np),d(np),v(np,np),b(nmax),z(nmax)
      do 12 ip=1,n
        do 11 iq=1,n
          v(ip,iq)=0.
11      continue
        v(ip,ip)=1.
12    continue
      do 13 ip=1,n
        b(ip)=a(ip,ip)
        d(ip)=b(ip)
        z(ip)=0.
13    continue
      nrot=0
      do 24 i=1,50
        sm=0.
        do 15 ip=1,n-1
          do 14 iq=ip+1,n
```

```
            sm=sm+abs(a(ip,iq))
14        continue
15      continue
        if(sm.eq.0.)return
        if(i.lt.4)then
          tresh=0.2*sm/n**2
        else
          tresh=0.
        endif
        do 22 ip=1,n-1
          do 21 iq=ip+1,n
            g=100.*abs(a(ip,iq))
            if((i.gt.4).and.(abs(d(ip))+g.eq.abs(d(ip)))
     *         .and.(abs(d(iq))+g.eq.abs(d(iq))))then
              a(ip,iq)=0.
            else if(abs(a(ip,iq)).gt.tresh)then
              h=d(iq)-d(ip)
              if(abs(h)+g.eq.abs(h))then
                t=a(ip,iq)/h
              else
                theta=0.5*h/a(ip,iq)
                t=1./(abs(theta)+sqrt(1.+theta**2))
                if(theta.lt.0.)t=-t
              endif
              c=1./sqrt(1+t**2)
              s=t*c
              tau=s/(1.+c)
              h=t*a(ip,iq)
              z(ip)=z(ip)-h
              z(iq)=z(iq)+h
              d(ip)=d(ip)-h
              d(iq)=d(iq)+h
              a(ip,iq)=0.
              do 16 j=1,ip-1
                g=a(j,ip)
                h=a(j,iq)
                a(j,ip)=g-s*(h+g*tau)
                a(j,iq)=h+s*(g-h*tau)
16          continue
              do 17 j=ip+1,iq-1
                g=a(ip,j)
                h=a(j,iq)
                a(ip,j)=g-s*(h+g*tau)
                a(j,iq)=h+s*(g-h*tau)
17          continue
              do 18 j=iq+1,n
                g=a(ip,j)
                h=a(iq,j)
                a(ip,j)=g-s*(h+g*tau)
                a(iq,j)=h+s*(g-h*tau)
18          continue
              do 19 j=1,n
                g=v(j,ip)
                h=v(j,iq)
                v(j,ip)=g-s*(h+g*tau)
                v(j,iq)=h+s*(g-h*tau)
19          continue
              nrot=nrot+1
            endif
21      continue
22    continue
        do 23 ip=1,n
          b(ip)=b(ip)+z(ip)
          d(ip)=b(ip)
          z(ip)=0.
23    continue
24  continue
      pause '50 iterations should never happen'
      return
      end
```

```
      subroutine eigsrt(d,v,n,np)
         implicit real*8 (a-h,o-z)
      dimension d(np),v(np,np)
      do 13 i=1,n-1
        k=i
        p=d(i)
        do 11 j=i+1,n
          if(d(j).ge.p)then
            k=j
            p=d(j)
          endif
11      continue
        if(k.ne.i)then
          d(k)=d(i)
          d(i)=p
          do 12 j=1,n
            p=v(j,i)
            v(j,i)=v(j,k)
            v(j,k)=p
12        continue
        endif
13    continue
      return
      end
```

The output from this programme comprises a set of 22 eigenvalues, each followed by the corresponding 22-term eigenvector, in **approximately** the correct format for input to the Ferranti Mark I programme.

Then follows a 3-column list of bond orders, with the present non-standard numbering. The 4th column shows the corresponding numbering used by Mason, with the bond order values read from Figure 2 of his paper, *Nature*, **179**, 465-467 (1957); agreement between the two sets of bond orders confirms the equivalence between this F77 programme and the original Ferranti Mark I programme. The non-consecutive bond orders come out in a different order from the originals as they were chosen by hand (and the last one, 16,21 was wrongly chosen as 14,19)!

Table created 2009-01-13 14:11

present numbering			Mason's numbering		
1	2	1.611	10	14	
2	3	1.456	9	14	
3	4	1.593	9	22	
4	5	1.698	21	22	
5	6	1.626	20	21	
6	7	1.703	19	20	
7	8	1.579	8	19	
8	9	1.499	7	8	
9	10	1.779	6	7	
10	11	1.498	6	13	
11	12	1.610	5	13	
12	13	1.610	5	12	1.61
13	14	1.498	4	12	1.50
14	15	1.779	3	4	1.78
15	16	1.499	2	3	1.50
16	17	1.579	2	18	1.58
17	18	1.703	17	18	1.70
18	19	1.626	16	17	1.63
19	20	1.698	15	16	1.70
20	21	1.593	1	15	1.60
21	22	1.456	1	11	1.47
22	1	1.611	10	11	
2	11	1.515	13	14	
3	8	1.543	8	9	
13	22	1.515	11	12	1.52
16	21	1.543	1	2	1.54

Q.E.D.