ISSN 0281-2762

A COMPUTER PROGRAM FOR CALCULATING POSSIBLE TRANSITIONS IN COMPLICATED SPECTRA

Sven-Göran Pettersson

LRAP-14

ABSTRACT

A computer program is described which will, from ordered known levels in a level system, calculate ordered wavelengths or wavenumbers of possible lines. If the program is used with selection rules the output is reduced to more relevant lines.

1. INTRODUCTION

When working with complicated spectra it is always useful to have computation capability to handle large matrices. As an example, for the analysis of the spectrum of O III the computation of all possible differences between all the level values gives rise to more than ten thousand different values. The sorting of these values e.g. after wavelength is very important and normally a special program has to order the values after the computation. This may need big scratch areas.

I have developed a procedure, which will, directly from ordered levels produce ordered calculated values of the differences between the level values. The number of real numbers used internally in the program is only about seven times the number of levels and it is, therefore, possible to execute the program on a small computer.

2. COMPUTATION PROCEDURE

The procedure of the program utilizes the fact that the difference between two level values is an increasing function of the upper level value and a decreasing function of the lower level value. The principle is best explained by an example. Let us take 6 levels with the level values 1, 4, 7, 15, 27 and 31. We can now make a square matrix as shown in Fig. 1. The matrix elements are the differences between the numbers. If we now want to order the differences beginning with the largest number we just have to start at the lower left corner. This is in consequence of a general rule that for any

	1	4	7	15	27	31
1	0	-3	-6	-14	-26	-30
4	3	0	-3	-11	-23	-27
7	6	3	0	-8	-20	-24
15	14	11	8	0	-12	-16
27	26	23	20	12	0	-4
31	30	27	24	16	4	0

Fig. 1 Matrix of differences of numbers.

number in the matrix the number just above and just to the right is smaller.

When 30 is taken out from the matrix the next largest number must be 26 or 27. Now the program has to find the largest of these two which is 27 and this number is then replaced by 24. The next step is to compare 26 and 24 and find the largest which is 26. Then 26 is replaced by both 14 and 23 and so on. In this way the program never has to compare more than 6 different values. As soon as a number is found to be the largest, it can be printed directly in an output file, and no other operation using the memory is needed. For a program with, say, 1000 levels, it has to compare and order at most 1000 differences each time, which is much more favourable than operating on 1000000 different values at the same time.

As an example I have made a computation for about 150 levels, with a computation time of about 25 seconds on a Univac 1108.

3. DECRIPTION OF THE PROGRAM

3.1 Input

Part of a typical input to the program is shown in Table I. The first column gives the energy of a level, here expressed in cm^{-1}. The second group of symbols is a level symbol describing what type of level it is. In the last column there is a combination of numbers also describing the level in a quantum-mechanical way. The first three numbers are the l, L and J quantum numbers. The last number indicates if the level is odd or even.

3.2 Computation

The program is written in the program language ALGOL. It is rather easy to convert it to FORTRAN if necessary. In the beginning of the program there is a test to determine if the levels are ordered after magnitude. This must be done because the procedure is based on this. After the test, the program goes through the procedure previously described, but before the ordered differences between the level values are sent to the output file, they are tested according to selection rules from line 23 to line 27 of the program (Table II). The program begins with the largest wavenumber difference and will stop when it reaches a lower limit given in line 22 of the program. For the present program the listing will stop when the wavenumber is less than 10000 cm^{-1}, which means a wavelength longer than 10000 Å.

3.3 Output

With the "WRITE" command in lines 30 to 32, five wavenumber differences are printed along with their identifications, on each line. In Table III, part of such a listing is shown.

4. APPLICATIONS

When this program was used for O III, initially separate calculations were made for the singlet, triplet and quintet systems. In this way no intercombinations were calculated. To introduce these, all levels in O III were used in a new calculation.

By using selection rules and input data of different types, any relevant coupling condition can be simulated.

This program can be used to solve many other sorting problems. If the matrix elements are certain functions of their positions e.g. if $M_{ij} = x_i x_j$ where x_i and x_j are ordered positive values, the same procedure can be used to order the matrix elements.

Table I. Input data for the program

374575.000	3P11D1	1210
374662.500	3P11D2	1220
374798.598	3P11D3	1230
377375.000	4D00F2	2320
378408.500	3P11P2	1120
378420.898	3P11P1	1110
378438.098	3P11P0	1100
379232.000	4D00D1	2210
379293.000	4D00D2	2220
379356.000	4D00D3	2230
380610.297	4F00F3	3331
380624.500	4F00F4	3321
380668.898	4F00F2	3331
380685.699	4F00F1	3341
380705.000	4D00P2	2120
380716.000	4D00P1	2110
380735.000	4D00P0	2100
381177.898	4F00G4	3431
381211.500	4F00G3	3441
381403.297	4F00G1	3451
381455.699	4F00D1	3231
381472.199	4F00G2	3441
381478.000	4F00D2	3221
381623.500	4F00D4	3211
381643.398	4F00D3	3221
392221.000	5S00P2	120
394090.000	3S12D1	211
394126.000	3S12D2	221
394195.000	3S12D3	231
400354.797	3D11P2	2121
400464.699	3D11P1	2111
400518.398	3D11P0	2101
401379.000	3D11F2	2321
401475.398	3D11F3	2331
401530.000	5D00F2	2320
401609.098	3D11F4	2341
402530.000	5D00D3	2230
403190.000	5F00F3	3331
403207.000	5F00F4	3321
403223.000	5F00F2	3331
403244.000	5F00F1	3341
403503.898	5G00G3	4440
403503.898	5G00G4	4430
403512.398	5G00G1	4450
403512.398	5G00G2	4440

Table II. PROGRAM LISTING

```
BEGIN INTEGER A,B,D,E,G,L,S,R,J,K,M;
READ(M);
BEGIN REAL ARRAY A5(1:2*M),A4(1:5);INTEGER ARRAY B5(1:M);
STRING ARRAY F(6:1:M),F5(6:1:5),F6(6:1:5);
LOCAL LABEL L3;
FORMAT UTF1(D9.3,X2,S6,X1,S6,A1);
FORMAT UTF2(5(D9.4,X2,S6,X1,S6,X2),A1);
FOR J=(1,1,M) DO BEGIN READ (A5(2*J-1),F(J),B5(J));
A5(2*J)=J;END;G=0;
FOR J=(1,1,M-1) DO
BEGIN IF A5(2*J-1) GTR A5(2*J+1) THEN
BEGIN WRITE('UNSORTED'); WRITE(A5(2*J-1));
GO TO L3 END; END;
BEGIN REAL ARRAY C(1:3*M); C(1)=A5(2*M-1)-A5(1);
C(2)=A5(2*M);C(3)=A5(2);R=1;FOR K=(1,1,M*M) DO
BEGIN REAL V1;LOCAL LABEL L1,L2,L5,L6;
INTEGER V2,V3;V1=C(1);V2=C(2);V3=C(3);
L=1;
FOR S=(1,1,R) DO
BEGIN IF V1 LSS C(3*S-2) THEN
BEGIN L=S; V1=C(3*S-2);V2=C(3*S-1);V3=C(3*S) END;END;
IF 100000/V1 GTR 10 THEN BEGIN GO TO L3 END;
A=B5(V2);B=B5(V3); IF ABS(A//1000-B//1000) LSS 2
AND ABS((A-1000*(A//1000))//100-(B-1000*(B//1000))//100)LSS 2
AND ABS((A-100*(A//100))//10-(B-100*(B//100))//10) LSS 2 AND
NOT (((A-100*(A//100))//10)EQL 0 AND ((B-100*(B//100))
//10 EQL 0))
```

```
AND 2*((A-B)//2) NEQ (A-B) THEN
BEGIN G=G+1; A4(G)=V1; F5(G)=F(V3); F6(G)=F(V2);
IF G EQL 5 THEN
BEGIN WRITE(UTF2,A4(1),F5(1),F6(1),A4(2),F5(2),F6(2),
A4(3),F5(3),F6(3),A4(4),F5(4),F6(4),A4(5),F5(5),
F6(5)); G=0 END; END;
IF R NEQ 1 AND L NEQ R THEN
BEGIN C(3*L-2)=C(3*R-2);C(3*L-1)=C(3*R-1);
C(3*L)=C(3*R) END
ELSE BEGIN C(3*L-2)=0; C(3*L-1)=0; C(3*L)=0 END;
R=R-1; D=V3+1; E=V2-1;
IF R NEQ 0 AND D NEQ M+1 THEN
BEGIN FOR J=(1,1,R) DO
BEGIN IF D EQL C(3*J) THEN BEGIN GO TO L1 END;
END; GO TO L5
END ELSE BEGIN IF D NEQ M+1 THEN BEGIN
L5: R=R+1;
C(3*R-2)=(A5(2*V2-1)-A5(2*D-1));C(3*R-1)=A5(2*V2);
C(3*R)=A5(2*D); END; END;
L1: IF R NEQ 0 AND E NEQ 0 THEN
BEGIN FOR J=(1,1,R) DO
BEGIN IF E EQL C(3*J-1) THEN BEGIN GO TO L2 END;
END; GO TO L6
END ELSE BEGIN IF E NEQ 0 THEN BEGIN
L6: R=R+1;
C(3*R-2)=A5(2*E-1)-A5(2*V3-1); C(3*R-1)=A5(2*E);
C(3*R)=A5(2*V3);
L2: END DO;
END; END; END;
L3:
```

```
57  IF G NEQ 0 THEN BEGIN FOR J=(1,1,G) DO BEGIN
58  WRITE(UTF1,A4(J),F5(J),F6(J)) END;
59  BEGIN FORMAT UTF3(D11.3,X2,S6,X2,I4,A1);
60  FOR J=(1,1,M) DO
61  BEGIN WRITE(UTF3,A5(2*J-1),F(J),B5(J)); END; END; END;
62  END;
63  END
```

If V1 in line 28 is changed to 100000000/V1 then the program will give vacuum wavelengths instead of wavenumbers.

Table III. Part of a typical output from the program.

158059.1	2P11P2	3P00P2	158058.0	2P11P1	3P00P2
157845.4	2P11P1	3P00P0	155175.8	2P11P2	3P00S1
153215.5	2P11S1	3S11P2	153038.4	2P21P1	4P11S1
151839.9	2P11P2	3P00D3	151619.8	2P11P2	3P00D2
151468.3	2P11P0	3P00D1	148708.4	3P00D2	4P11D3
142269.4	2P00P1	2P11P1	142269.1	3P00P2	4P11D3
141690.5	2P11P1	2P21P0	141594.8	2P11P2	2P21P1
141376.1	2P11P1	2P21P2	139457.5	3P00S1	4P11S1
131422.1	2P21P2	6D00D3	130406.0	3D00D1	3D12PX
129515.7	3D00F3	3D12DX	129337.5	3D00F4	3D12DX
128018.5	3D00F4	3D12FX	126896.8	3D00D2	3D12DX
126714.3	3S00P1	3S12D1	126562.4	3S00P2	3S12D3
124706.0	3D00P2	3D12DX	121179.4	3P00D2	6D00D3
119939.2	2P00P1	2P11D2	119751.7	2P00P2	2P11D1
114740.1	3P00P2	6D00D3	108528.4	3P00D2	5D00D3
107664.7	3P00D1	5D00F2	107528.4	3P00D2	5D00F2
103158.4	3S00P0	4P00P1	103148.5	3S00P1	4P00P2
102891.6	3S00P2	4P00P2	102783.1	3S00P2	4P00P1
100470.8	2P11S1	3P00S1	100319.6	3S00P2	4P00S1
99111.25	3S00P1	4P00D1	98961.41	3S00P2	4P00D2
97999.35	3P00D3	5S00P2	96957.10	2P21P2	4D00P1
96728.40	2P21P1	4D00P2	96642.70	2P21P0	4D00P1
95316.40	2P21P1	4D00D2	95255.40	2P21P1	4D00D1
94649.60	2P21P2	3P11P2	94461.50	2P21P1	3P11P0
94347.60	2P21P0	3P11P1	92407.70	3S11P2	4P11D3
90903.60	2P21P2	3P11D2	90816.10	2P21P2	3P11D1
86988.90	3S11P0	4P11S1	86986.59	2P11S1	2P21P0
86850.74	3P00D1	4D00P1	86839.74	3P00D1	4D00P2
86672.19	2P11S1	2P21P2	86483.35	3P00D3	4D00P2
85291.40	3P00D2	4D00D2	85230.40	3P00D2	4D00D1
84555.64	3P00D1	3P11P1	84543.24	3P00D1	3P11P2
83509.74	3P00D1	4D00F2	83373.40	3P00D2	4D00F2
83147.50	3P00S1	4D00P2	83059.00	3P11D1	3D12PX
82835.40	3P11D3	3D12PX	82747.25	3S00P1	3S11P1
81420.54	3D00F2	3D11D3	81371.64	3D00F2	3D11D2
81046.59	3D00F4	3D11D3	80880.59	3P00S1	3P11P0
80797.00	3P00D2	3P11D3	80709.74	3P00D1	3P11D1
80487.79	3P00P0	4D00P1	80440.85	3P00D3	3P11D2
80275.15	3P00P2	4D00P1	80264.15	3P00P2	4D00P2
79376.54	3D00F2	5F00D4	79375.40	3P11D3	3D12DX
79225.50	3P11P2	3D12PX	79213.10	3P11P1	3D12PX
79099.54	3D00F2	5F00G4	79068.75	3D00F3	5F00D2
78944.59	3D00F4	5F00G2	78929.75	3D00F3	5F00G3
78883.59	3D00F4	5F00G1	78878.59	3D00F4	5F00D1
78751.59	3D00F4	5F00G3	78744.54	3D00F2	5F00F4