Transputer Implementation for the Shell Model and Sd Shell Calculations

by

Mohammad Riaz

A thesis submitted to the
Faculty of Science
University of Glasgow
for the degree of
Doctor of Philosophy

August, 1990.

To my parents
Choudhary and Mrs. Mohammad Shareef, whose encouragement in studies has always been a great source of inspiration.

Declaration

Except where specific reference is made to the work of others, this thesis has been composed by the author. It has not been accepted in any previous aplication for a degree. I further state that no part of this thesis has already been or currently is being submitted for any degree or qualifiction at any other university.

Mohammad Riaz

Acknowledgements:

There are many people that I would like to thank and convey my gratitude. I would like to thank Prof. R.P. Ferrier, the head of the department for the provision of facilities and over all supervision of research in the department. I would also wish to thank Prof. E.W. Laing for his very useful help during the start of my research in this department.

A great deal of my thanks go to my supervisors, Prof. R.R. Whitehead and Dr. A. Watt, for their constant encouragement and provision of intellectual ideas, over the years. Their kind guidance and genuine approachability have contributed very greatly in the completion of this research. I would also like to thank Dr. I.F. Wright, University of Manchester, for his very useful discussions and comments on my sd shell calculation results.

My thanks also go to my colleagues, Roslan, Djamal, Tengku, Saeed, Khalid, Kaltoum, Allan and Garry for sharing the happy and unforgetful moments as students, over the years.

Last, but not the least, I would like to thank my wife Nasim and children Shamaela, Quratul Eine and Ali for their patience, support and perseverance during my years of studies.

The financial grant for most part of my studies has been provided by the Ministry of Education, Government of Pakistan, Islamabad, which is very gratefully acknowledged. During the last one year financial support was provided by the Charles Wallace Trust of Pakistan through the British Council and by the University of Glasgow, which is highly appreciated and duly acknowledged.

Summary

This thesis consists of two parts. The first part discusses a new Shell model implementation based on communicating sequential processes. The second part contains different shell model calculations, which have been done using an earlier implementation.

Sequential processing computers appear to be fast reaching their upper limits of efficiency. Presently they can perform one machine operation in every clock cycle and the silicon technology also seems to have reached its physical limits of miniaturization. Hence new software/hardware approaches should be investigated in order to meet growing computational requirements. Parallel processing has been demonstrated to be one alternative to achieve this objective. But the major problem with this approach is that many algorithms used for the solution of physical problems are not suitable for distribution over a number of processors.

In part one of this work we have identified this concurrency in the shell model calculations and implemented it on the Meiko Computing Surface. Firstly we have explained the motivation for this project and then give a detailed comparison of different hardware/software that has been available to us and reasons for our preferred choice. Similarly, we also outline the advantages/ disadvantages of the available parallel/sequential languages before choosing parallel C to be our language of implementation. We describe our new serial implementation DASS, the Dynamic And Structured Shell model, which forms basis for the parallel version. We have developed a new algorithm for the phase calculation of Slater Determinants, which is, superior to the previously used occupancy representation method. Both our serial and parallel implementations have adopted this representation.

The PARALLEL GLASNAST, as we call it, PARALLEL GLASgow Nuclear Algorithmic Technique, is our complete implementation of the inherent parallelism in Shell model calculation and has been described

in detail. It is actually based on splitting the whole calculation into three tasks, which can be distributed on the number of processors required by the chosen topology, and executed concurrently. We also give a detailed discussion of the communication/ synchronization protocols which preserve the available concurrency.

We have achieved a complete overlap of the the main tasks, one responsible for arithmetically intensive operations and the other doing searching among, possibly, millions of states. It demonstrates that the implementation of these tasks has got enough built in flexibility that they could be run on any number of processors. Execution times for one and three transputers have been obtained for ^{28}Si, which are fairly good. We have also undertaken a detailed analysis of how the amount of communication (traffic) between processors changes with the increase in the number of states.

Part two describes shell model calculations for mass 21 nuclei. Previous many calculations have not taken into account the Coulomb's interaction, which is responsible for differences between mirror nuclei. They also do not use the valuable information on nucleon occupancies. We have made extensive calculations for the six isobars in mass 21 using CWC, PW and USD interactions. The results obtained in this case include, energy, spin, isospin and electromagnetic transition rates. These result are discussed and conclusions drawn. We concentrate on the comparison of the properties in of each mirror pairs. This comparison is supplemented by tables, energy level diagrams and occupancy diagrams. As we consider mirror pair individually, the mixing of states, which is caused by the short range nuclear force and the Coulomb force, becomes more evident. The other important thing we have noticed is, that some pairs of states swap their places, between a mirror pair, on the occupancy diagram, suggesting that their wave functions might have been swapped.

We have undertaken a detailed study to discover any swapping states. The tests applied to confirm this include comparison of energy, electromagnetic properties and the occupancy information obtained with different interactions. We find that only the 9_1, 9_2 states in Al have swapped over. We also report some real energy gaps which exist on the basis of our calculations for Al.

Contents

PART ONE

Introduction : The Glasgow Shell model implementation[1], was certainly a big single advancement towards making large basis sd-shell calculations. In spite of its innovative features like using m-scheme and the Lanczos method for achieving efficient convergence of states, it had its inherent limitations. The most conspicuous of these limitations is the handling of large basis calculations, corresponding to the shells beyond the sd-shell. It is well known fact that the machine resources required by a calculation of this magnitude are quite enormous. Also it is understandable that the available sequential processors are not going to become hundred/thousand times faster to meet the demands of such calculations, perhaps due to the fundamental physical limits of **computation** and miniaturization[2, 3, 4]. So we believe that a parallel solution to this problem is not only essential but realistic as well.

To achieve this objective, a two pronged effort has been launched by the Glasgow Group. The first alternative considered, was to construct a dedicated 'Shell model processor', along with mapping of the concurrent shell model algorithm onto this hardware. The details of this first project have already been reported in Mackenzie et. al [5].

The second alternative has been to identify the inherent parallelism in the shell model calculations and implement it on any other suitable multiprocessor hardware. The work, which is being presented here, is a complete implementation of such a project. This present technique is based on identifying concurrently executable parts of the shell model calculations and implementing them as independent tasks. These tasks are then distributed over various processors and allowed to run independently as communicating sequential processes[6].

The implementation of the project has been donein two stages. The first one is a serial implementation of the old Glasgow shell model in the language 'C'[7], and it fully exploits the elegance of its dynamic data structures in contrast with the previous Fortran implementation[1]. The second implementation actually descends from this one and is in parallel 'C'. This breaks up the whole calculation into natural and concurrently executable tasks. The required communication protocols are embedded appropriately within these tasks and they can be loaded and run as master and slave processes on separate processors.

We have used a PC with an add-in transputer board and a Meiko Computing Surface for developing and running of these parallel programs. The Meiko machine is based on 32 x T800 transputers, which can be reconfigured to any suitable topology. After [illegible] the actual implementation, we have also made calculations to gauge the optimum flow of traffic on links between different transputers, by using different sizes of data blocks processed on the various transputers. Sample calculations have been made for si^{28} m=0 using different number of processors on Meiko and also on Sun 3/60, 4/110 and sparc station 1, using the serial version. The delightful feature of these calculations has been that the execution (i.e. CPU) time taken, by using multi-tasking on one transputer, has come out to be 3-4 times less than on any of other sequential machines. These results [illegible] obtained by using multi transputer topology have also been reported.

In the next few chapters, we describe the complete development of this project, from conception to completion in the following stages.

- Why we should have parallelism ? Different parallel architectures for achieving it.
- Discovery of the inherent parallelism in the shell model calculations, by using the underlying physical concepts. Once this information is extracted, then such an algorithm can, in principle, be translated into a realistic implementation using more than one forms of parallelism. We explain these forms and conclude that algorithmic parallelism is the most suitable for our present implementation.

- At this next stage we briefly explain different parallel architectures that have been used before and are still available. We also explain our preferred choice, Meiko supercomputer based on T800 transputer architecture and the reasons for doing so.

- After deciding the algorithm and the appropriate hardware, the next most important step is to decide the most suitable implementation language that can translate the semantics of the algorithm in a way that maps very efficiently onto the chosen hardware. We discuss different possibilities and conclude with reasons for our choice of the 'C' language.

- At this stage we fully describe the details of this implementation. This includes the explanations as to what should constitute the master task, the individual slave tasks and the topology used, so that they can preserve concurrency. After this whole explanation, we describe the calculations that have been done by using this distributed version and their comparison with the ones done using the serial version running on different architectures.

The type of parallelism, which we have employed in this implementation is known in literature as 'algorithmic parallelism', explained in detail in section 2.2.2.3. Because of that we are inclined to name this project as PARALLEL GLASgow Nuclear Agorithmic Shell model Techniques, or in short 'PARALLEL GLASNAST'.

CHAPTER 1

PARALLEL COMPUTING

1.1 Why Parallel Computing :

Scientists have always looked for more speed from their computers. As a result of that we have witnessed, in the recent past, an evolution of scientific computers in which more and more concurrent/ parallel arithmetic operations have been allowed. In the last thirty years or so the practical gain in the speed of computers, comes out to be roughly a factor of one million. Out of this a factor of one thousand accounts for the increase in the intrinsic speed of their components, while the other one thousand comes from parallelism. In fact the idea of doing simultaneous calculations with large number of computational units is not a new one. It was recognised even by Babbage[8] in the last century. But the recent developments in the field of vectorization, multiprocessing and multitasking have revolutionalised the whole pattern of scientific calculations. We give below a few reasons why we should have parallel computing.

1.1.1 Limits of Silicon Technology:

The rapid developments in science and engineering need an increase in power by orders of magnitude, while the improvement in the intrinsic properties of the silicon based devices appear to be limited by many factors. The device operation speed, however, may continue to increase but there seems little scope for orders of magnitude. The literature on 'the limits of microelectronics and computing[2, 3, 4] ', is increasing very rapidly. Also, since the vector machines have evolved to a stage where they produce one arithmetic operation in every machine cycle, we cannot expect to see any dramatic advances from them.

New technologies based on entirely different media like gallium arsenide, superconductors and optical computing are still in their stages of development. In any case, these new technologies are also most likely to support the advantages gained by parallel computing.

1.1.2 Parallelism in Physical Systems:

The above discussion explains the fact that if the physical problem were amenable only to serial or vector computations, the outlook for obtaining significant increases in the computational power would be bleak. But fortunately in the real world, most of the computationally demanding scientific problems have got enormous inherent parallelism. This claim is well supported in cases like Monte Carlo Methods for Lattice Gauge Theory, Nuclear Structure, Neural Network, Image Processing and many other real time applications. The mathematical model of such application, normally involves multiplication of high dimensional matrices. We explain in the example given below that matrix multiplication is well suited for parallel evaluation.

Consider two matrices A(NxM) and B(MxL) such that their product is given by Product(NXL). First we expand the two matrices in two dimensions. Then multiply them individually to form the product matrices, which would form the respective elements of the product matrix Product. Symbolically we can write it as :

$$\text{Product}\,(i1, i2) = \Sigma^{M}_{i=1}\, A_{i1,\, i2} \times B_{i2,i3}$$

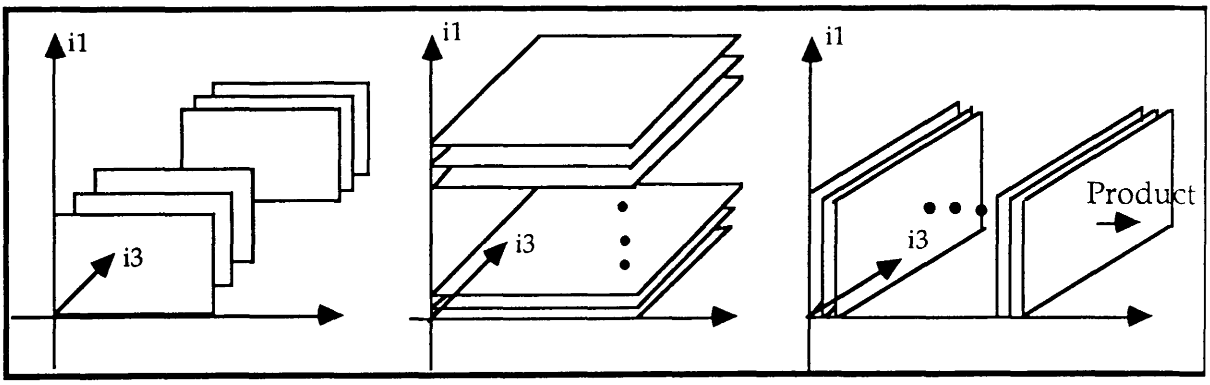


Fig 1.1: Perspective of Matrix Multiplication w.r.t. Parellelisation

Also simple piece of code to achieve the same objective could be as:

```
Function Product(A, B, N, M)
Real A(N, M), B(M, L), Product(N, L)
Product = Sum ( Expand( A, 3, L) *  Expand( B, 1, N), 2)
```

```
Return
End
```

The fig. 1.1 explains, that the evaluation of the components of of the product matrix 'Product', is inherently independent of each other and can be computed in parallel on any number of processors.

1.2 How to achieve parallelism:

A lot of research has been going on to harness the benefits of parallelism by making different innovations in the hardware and software technologies. Both these fields are so complementary that they merit separate brief explanations.

1.2.1 Parallelism in Hardware :

In the last two decades or so many computer systems and architectures have been built which use one form of parallelism or the other. By doing so they have achieved a performance in excess of what is attainable directly from the underlying technology used in the constituent chips. Their architectures have been based on different principles, number and type of processors, memory modules, I/O channels and their over all control structure. Generally these computers can be classified into two categories.:

- Pipelined Computers
- Multiprocessor arrays.

It would be impracticable here, to give a comprehensive description of all designs in these categories; instead we choose to give a brief summary and comparison for a few of them to highlight the comparative advantages and limitations. All these computer architectures are based on design principles generally referred as Flynn Taxonomy[10].

1.2.1.1 Flynn Taxonomy:

(i) SISD - Single Instruction stream, Single Data stream.

(ii) SIMD - Single Instruction stream, Multiple Data Stream.

(iii) MISD - Mutiple Instruction stream, Single Data stream.

(iv) MIMD -Mutiple Instruction stream, Multiple Data stream.

Many machines of today are, however, a hybrid of these design principles. As an example The Cray X-MP has upto four processors (MIMD), but each processor uses pipelining (SIMD) for vectorization. Also wherever there are more than one processors, memory could be local, global or a combination of these. The performance of these computers based on different principles is compared by the millions of floating point operations performed in 1 second or Mflops.

1.2.1.2 Pipelined Vector Computers

In the year 1969, first CDC 7600 machine became operational. This marked the end of the reign of fortran machines in supercomputing. It was fifty times faster than IBM 7090, its senior by ten years. That was about the time when it was realised that higher speeds can be obtained if one gives up the 'von Neuman' architecture of the CPU, which briefly translates as :

(i) Fetch an instruction.

(ii) Decode it (being prepared for anything).

(iii) Fetch the operands (from anywhere).

(iv) Execute the operation and store the results (anywhere).

(v) Start the cycle all over again.

This idea of departure from this most favoured architecture, herald a new era in the development of vector supercomputers. These computers initially suffered from the following problems.

- High peak vector speeds on long vectors contrasted too sharply with scalar speeds achieved.
- High Cost of Software. It appeared to be very expensive task for the manufacturer to develope a complete operating sysytem and vectorizing fortran for a very small number of machines. The users, as well were

required to convert their programs to take advantage of the vector instructions.

- The development cycle was long and costly.

Partly, for some of the above reasons the first two vector supercomputers, CDC STAR-100 and the Texas Instrument's ASC (Advance Scientific Computer) were a commercial failure. CRAY-1[9] was the first computer which made the vector supercomputing acceptable.Its success was achieved by offering solutions like higher scalar speed, simple software approach, batch operating system, excellent hardware reliability and good user services such as time sharing, networking, database systems etc.

The CRAY-XMP supercomputer was an optimization of its predecessor. Most significant of the changes made include:

- Each processor being a stand alone vector processor for large scale computations, rather being a member of the pipeline.
- Each processor had four memory ports; two for vector fetches, one for vector store and one for the independenent I/O.

1.2.1.3 Highly Parallel Computers :

Initially, vector computers were essentially pipelined and based on SIMD design of architecture. The technological improvements later on used multiple pipelines or multiple processors. This resulted in supercomputers like CRAY-XMP and ETA-10 etc., which were an impressive hybrid of SIMD and MIMD architectures. This development had been driven home by the fact that the speed of a single processor/pipeline is getting close to the limit, set by the finite speed of light.

The success of vector computers brought to light a seemingly natural consequence of this type of hybrid architecture. That is to say, 'if we can use a large number of processors in parallel, the computational speed obtained would certainly exceed that of the existing supercomputers; even for relatively low speed per processor.

The Thinking Machines capitalized on this idea and produced a highly parallel supercomputer called Connection Machine CM-2 [11]. This machine consisted of 65,536 processors. .The initial bottleneck for making such machine has been how to prevent different processing units accessing same part of the memory/data at the same time. One alternative used in some vector (SIMD) machines was to use some sort of locking system. But the principle feature of the MIMD architecture is that the operations are done in lock-step mode[9]. This simply means that at a particular tick of the clock, all operations are performed identically on different data. However, some alternatives sought later on include having separate memory attached to each processor and even having a fast on-chip memory along with the processor.

The Connection Machine, basically consists two components, a processor array and a host machine. Each processor has its own memory for data storage. All processors execute a single instruction issued by the host machine, which means that the Connection Machine too has SIMD architecture. However, from hardware point of view, pipeline mechanism and the Connection Machine are very different. But the characterisation of the algorithmic adaptations and their efficiency are largely similar for both types of machines. The reason for this is, that both types of machines belong to SIMD class, and both show serious bottlenecks for arbitrary patterns of memory access. This similarity makes it possible to make a meaningful comparison between the performance of supercomputers and the Connection Machine. The detail description and comparison of performance characteristics of different array processors has been given in Hockney and Jesshope[9] and Zakharove[12].

1.2.1.4 Meiko Concurrent Supercomputer :

The Meiko is a modular computer, having a minimum of one processor(T414 or T800). Extra processors can be added to increase the capacity of the machine indefinitely. It can be connected to a host (Vax, Sun etc.) or can run at its own. A number of utilities like memory handling, computing, I/O etc. are supplied on the same board along with the processor. These boards can easily be added or removed from the machine. The machine itself replicates the function of one procesor which is called transputer. A simple schematic diagram of the Meiko Computing surface is given below. in fig 1.2.

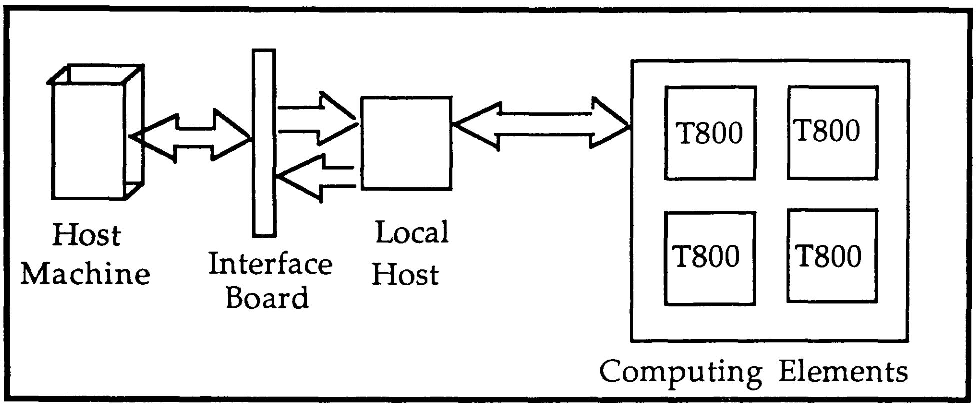


Fig . 1.2: The Meiko Supercomputer

More detailed circuit diagrams can be found in Meiko hardware reference manual[29]. We have used this supercomputer for the implementation of parallel shell model. Because of that we discuss in some details, the architecture and working of its fundamental processing element T800 transputer.

1.2.1.5 The Transputer T800.

The Inmos transputer represents a natural building block for the various multiprocessing systems, because of the provision of the on chip connection links. The Meiko Computing Surface at Glasgow is such a system, which consists of 32 x T800 transputers. They can be electronically reconfigured according to any desired topology most suitable for the application. The transputer T800 is a single chip which provides processing power, memory and communication hardware all on the same 1x1 cm^2 of silicon as shown in fig 1.3.

Important hardware features of the T800 are summerised below.

- It is 32 bit processor with 10 Mips CPU and another floating point co-processor, which is capable of delivering 1.5 Mflops performance.
- It has got a 4 Kbytes of fast on-chip memory in the form of a 50 nsec. RAM.
- The communication hardware consists of four 20 Mbits/sec serial bidirectional links. Both processors on the chip and the four

communication links can operate concurrently.

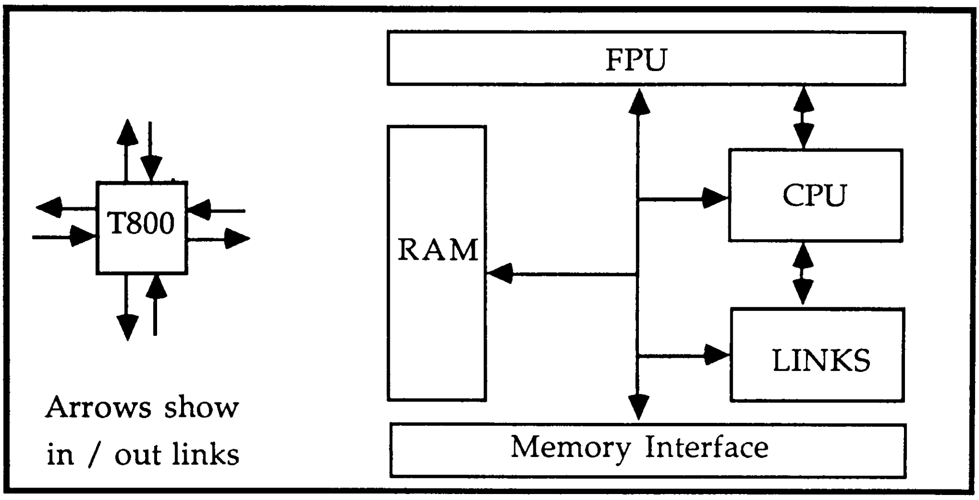


Fig 1.3: T800 Transputer Chip Configuration

It can be seen from the capabilities of the T800 hardware that it is relatively easy to construct a large and powerful MIMD supercomputer using an array of transputers. This would only require two wires per link to provide point-to-point communication between transputers.

Software Support for the Transputer:

The transputer system executes the 'Occam'[13, 14] programming language, in which concurrency can be described between different transputers and indeed within one transputer. All features of this language map precisely onto the transputer hardware. That is why all micro code instruction in the transputer are in Occam. Concisely, T800 can be described as an Occam model of concurrency.

Every process on the transputer is an independent program, with its source code, independent memory and space for local variables. It has got six registers (three for storing a 3-elements operand stack, onefor pointer to the work space for local variable to the process are stored, one register for pointer to the next instruction to be executed and the sixth register is for the operands), as shown in fig 1.4.

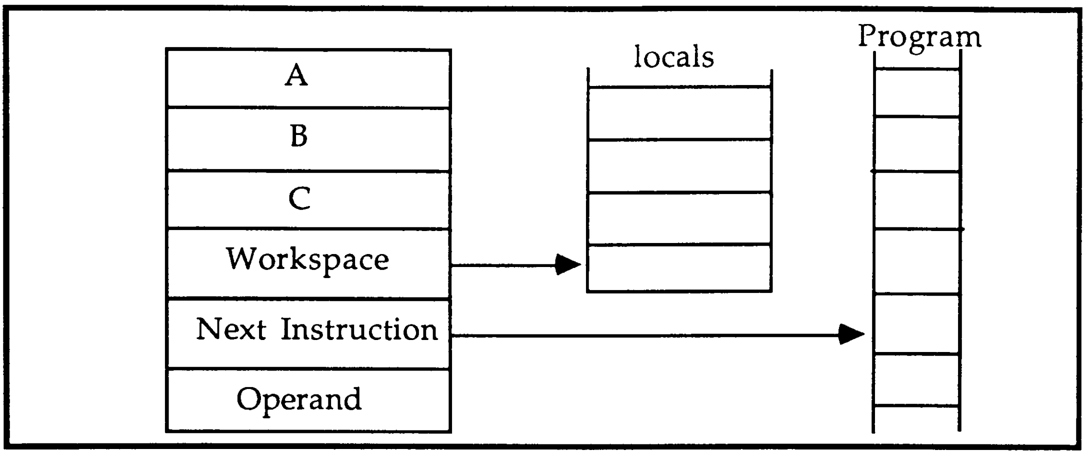


Fig 1.4: Transputer Registers and other work space elements

• In sharp contrast with the conventional processors discussed in the previous section, T800 does not have a complex instruction set. It rather exploits the fast-on-chip memory, a small number of registers and reduced instruction set (RISC). Running of programs using fast memory makes a great difference in the time of execution. This has been reflected in calculations[30] given in table 1 below.

Code/Memory mode	onchip	on chip	off chip	off chip
Data/Memory mode	on chip	off chip	on chip	off chip
Speed in secs.	4.17	10.14	8.55	14.41

table 1.1: Comparisonof execution speed with on/off chip data and code.

1.2.1.5.1 Model and Support for Concurrency :

The transputer provides a high degree of support for concurrency. The number of processes that can run concurrently on a transputer are limited only by the memory constraints. Its micro coded scheduler, in fact allows any number of processes to compete for the processor's time and other resourses. Transputer maintains two queues for the processes, one for active process being executed and the other for those waiting for the I/O or inactive due to some sort of interrupts(e.g. timer). The best form of concurrency on the transputer can be obtained through Occam, because this language embodies Hoar's communicating model of concurrency and has , built into it the communication primitives and concurrency fundamentals.

The processes, which run in parallel on transputer(s), exchange information through the act of simultaneous exchange of information. This is done regardless of the fact, which process is sending or receiving information. Consequently, the synchronisation and transfer of data between two processes takes place through their communication channels and only when both processes are ready.

Another welcome departure, in this case, from the earlier parallel programming technique is that parallelism is not left for the compiler (or vectorizor) to extract from the application. The program-mer is required to make the parallelism explicit. As a result of that, the choice of implementing a multiprocessor code on two or more processors is determined only by issues like load balancing, communication bandwith etc. This improvement has resulted in real term gain, in concurrency and efficiency of execution throughout the network.

1.2.1.5.2 Reconfigurability of Transputer Network :

Early transputer (T_{212}, T_{414}) required to be hand-connected in a desired topology. This had two draw backs.

1- Firstly, it is understandable that good performance gains can be obtained by using networks configured to some optimal topology, which conforms to the inherent parallelism of the problem. On the contrary, designing parallel processes fitting a fixed topology can be difficult as well as inefficient.

2- Secondly, hand-connecting a 32-transputer system, like the Glasgow Meiko Computing Surface, according to a certain topology would be quite tedious. Similarly, hand-connecting a 400 transputer system, like the Edinburgh Concurrent Supercomputer, would be extremely laborious and proned to errors.

On the other hand, it has been observed that the best suitable topology for a transputer network is not always obvious. Some times a grid or a torus (fig 1.5) may be useful, while at the other time a ring or even random graph may prove to be efficient. That is why the T800 based Meiko machines are electronically reconfigurable. In a given array of say 17 transputers, a user can softwire them in any configuration best suitable for the problem. This is achieved through the software switches, which have been built into the operating system.

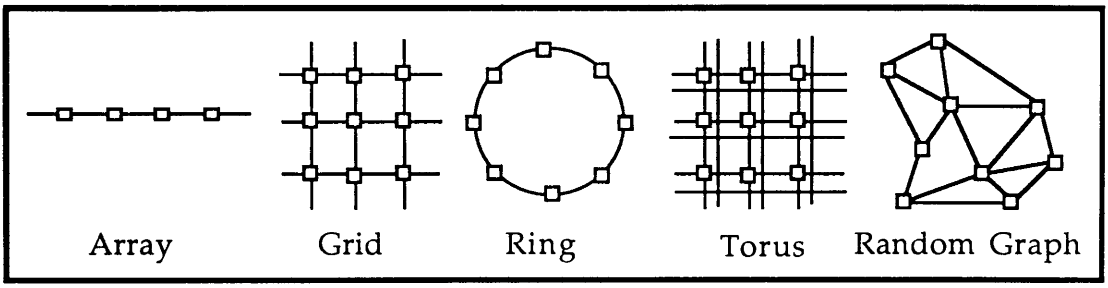


Fig 1.5: Differnt topologies for the transputers configurations

1.2.1.5.3 Versatility

We can count that the T800 transputer has a (possible) total of fifteen activities that can go parallel.

(i) Execute normal instructions (1)

(ii) Execute FPU instructions (1)

(iii) Do input on 4 links (4)

(iv) Do output on 4 links (4)

(v) Wait for event input (1)

(vi) Wait for timer expiration (high/low) (2)

(vii) Wait for time slice to expire (1)

(viii) Do transfers from/to memory (1)

The first four activities mentioned above account for nine separate processes actually doing something (i.e. modifying the contents of memory).

1.2.1.5.4. Portability.

The issue of portability between different machines has never had a completely satisfactory solution. It is generally true that a well designed algorithm developed for a scalar processor will perform respectably on most other scalar machines. But this is not true about supercomputers, which introduce requirements for vectorized/ parallelized algorithm. Also it is not always obvious, how to go about adopting an algorithm to effectively utilize the facilities provided by a target supercomputer. One reason is that optimized algorithms are often tailored to take advantage of the architectural features of the processor on which they were first designed to run.

There is little uniformity of design between the transputer and other processors; and indeed among most of the modern supercomputers. So any software which effectively uses the resources of one kind on one procesor, will fair less well on the others. The T800 transputer is certainly no exception to that. However, interfaces to most high level languages are available. This makes it possible that well designed sequential code in a high level language, can run effectively on the transputer, provided system dependent features like word length etc are not present. We recognise that even this level of portability is quite useful and has been explained in good details in the " Transputer does 5 or more MIPS even when not used in parallel", ref. [16].

A closer examination of the Meiko software reveals that the 'C' and fortran library interfaces IOSUB is relatvely free from machine dependent calls. The calls to functions 'time$elapsed', 'io$ibli' and 'io$iblo' only, belong to the Meiko compiler. The Occam layer, which heavily depends upon the Meiko Occam library, is used only to provide point-to-point synchronous communication between different processes. In principle this can be replaced by any other message passing method.

For instance, an implementation on Intel iPsc/2 has removed the occam layer completely and worked the Intel calls directly into IOSUB library. We realize that this provision of Intel version, not only allows immediate porting of applications between the Intel and Meiko hardware, but also demonstrate a greater scope, for the potential portability that exists in the Meiko software.

1.2.1.5.5 Software Inadequacies :

In spite of the advantages of the transputer arrays that have been discuused, commercially available system software is yet not mature. In developing these programs for the Meiko Computing Surface we experienced grave difficulties, associated mainly with the 'C' library. Only the very simplest disk I/O operations worked as the C language required them to, and we were forced to spend several months writing a number of low level replacements for important C functions like fread(), fwrite(), fscanf() etc. before any real progress could be made. It is to be hoped that as the use of transputer arrays becomes more wide spread such problems will disappear.

1.2.2 Parallelism in software :

In the previous section we discussed the performance and comparison of different architectures in terms of Mflops (million floating point operation) per second, their connectivety, reconfigurability and similar other parameters.

But on the software level there is a host of issues which are of equal significance. The choice of solution to these software issues is so vital that it can make a difference between having a multifold increase in the execution efficiency and the program just running in a sequential format or even not running at all. More benefits of parallel architecture can be brought to fruitation by deciding correctly on the following software issues.

- Level and Type of parallelism suitable to the application. There are a few different types of parallelism which map efficiently to different types of algorithm and the topology of network. They are explained in section 2.2.2.
- Possibilities of code migration, if it already exists.
- Choice of the appropriate programming language.
- Availability of good debugging tools could be essential for writing and maintaining complex codes.

In this section we briefly explain some of these issues, particularly with reference to our present project.

1.2.2.1 Levels of Parallelism :

Parallelism in physical problems can be structured or unstructured. The structured parallelism is the one in which a set of independent identical tasks can operate on different sets of data. While in the unstructured parallelism we have different instructions and or different data streams executing in parallel. the former is easier to to deal with than the later. They can be further classified as explained below.

1.2.2.1.1 Course Grain Parallelism :

In coding a physical problem parts of the computation may be organised in functions/subprograms. During the execution of the programs these functions can execute in parallel, either independently or as part of a concurrent process. This type of parallelism at the function level is called course grain parallelism. Our present implementation falls into this category.

1.2.2.1.2 Medium Grain Parallelism :

It is possible to specify parallelism at the loop level, which can then be discovered by the compiler through the dependence analysis. The fortran programs spend most of their times inside the DO loops [17]. So parallelising loops is one of the most crucial activities during the restructuring of fortran programs that can run on vector supercomputers. Sometimes this does result in dramatic speed ups[17, 18]. This parallelism at the loop level is called medium grain parallelism.

1.2.2.1.3 Fine Grain Parallelism :

The parallelism at the basic block level can be medium or fine grain depending upon the size of the block. These basic blocks can execute concurrently if no inter block dependency exists. Parallelism at the statement or operation level is also important, although the average resulting speed is far less than the loop level, particularly for numerical problems. This type of algorithm is called fine grain parallelism.

Because of the overheads involved, fine grain parallelism is usually exploited inside each processor by using different or multiple functional units or by using pipelining.

Our present implementation of Parallel GLASNAST (details given in the next chapter) is based on using the course grain parallelism. The processes consist of functions and run concurrently with similar other processes and exploit the inherent parallelism which embodies the shell model calculations.

1.2.2.2 Techniques for Exploiting Parallelism :

Once we know which level of parallelism is most beneficial to our application, we can discuss different techniques for exploiting it. Some of the factors which can greatly influence our choice at this stage include the languages available for implementing the model of parallelism along with other implementation parameters. However, an algorithm can be implemented, in many cases, using all the techniques discussed below, but the efficiency may be different in different cases.

1.2.2.2.1 Event Parallelism :

In this type of parallelism, the same code is run independently on each processor, but with different data. This technique can be useful for experimental data analysis, Monte Carlo event generation, low level image processing (on the individual pixels of the image) and similar other application areas. In such cases it seems straight forward to distribute the independent data to each available processor, which would work like a processor farm as shown in the fig 1.6

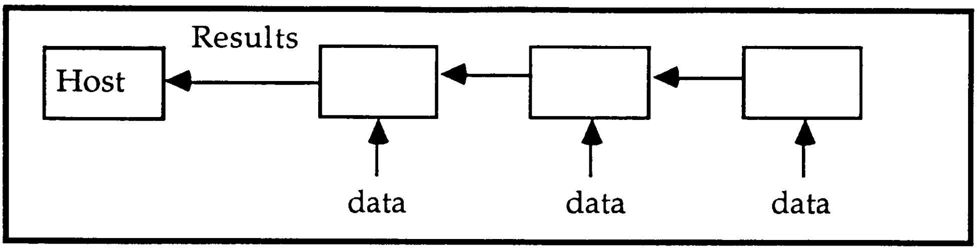


Fig 1.6: Arrangement of processors for event parallelism.

1.2.2.2.2 Geometric Parallelism :

In this type of parallelism, the same code is normally run on each processor, but the data area is split up among the different transputers. The difference with event parallelism, in this case is that inter process communications are necessary during execution for accessing the data held on a different transputer.

The performance is strongly influenced by factors, like number of nearest neighbors and the balance of computation to the communication, which in turn depends on the size of the data area assigned to each processor.

Potentially, deadlock situations are possible. Deadlock[19] is a situation in which two or more processes are waiting indefinitely for conditions which will never hold. More illustrious examples of deadlocks like the Dijkstra's [20] 'Dining Philosophers and Drinking Philosophers problems', reflect that deadlock situations can be fatal, and must be avoided [21]. That is why this technique is trickier than the event parallelism. A typical arrangement of transputers in this case is depicted in fig 1.7.

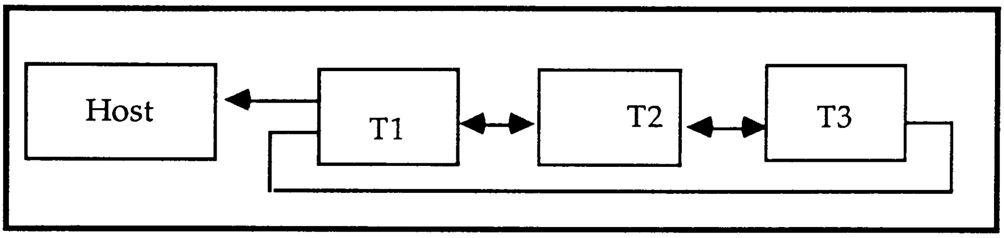


Fig. 1.7 Arrangment of transputers in Geometric Parallelism.

1.2.2.2.3 Algorithmic Parallelism:

This type of parallelism corresponds to using data flow or demand flow style of programming. In this technique a program or a process is partitioned and distributed across a network of processors. Each partition executes in parallel on its own data or the data it receives from other partitions of the program. Inter process communications are essential in this case, and that is one reason that such programs require a more complex control structure. One can expect that processors are required to be arranged in the form of pipes. One likely arrangement of processors is shown in fig 1.8.

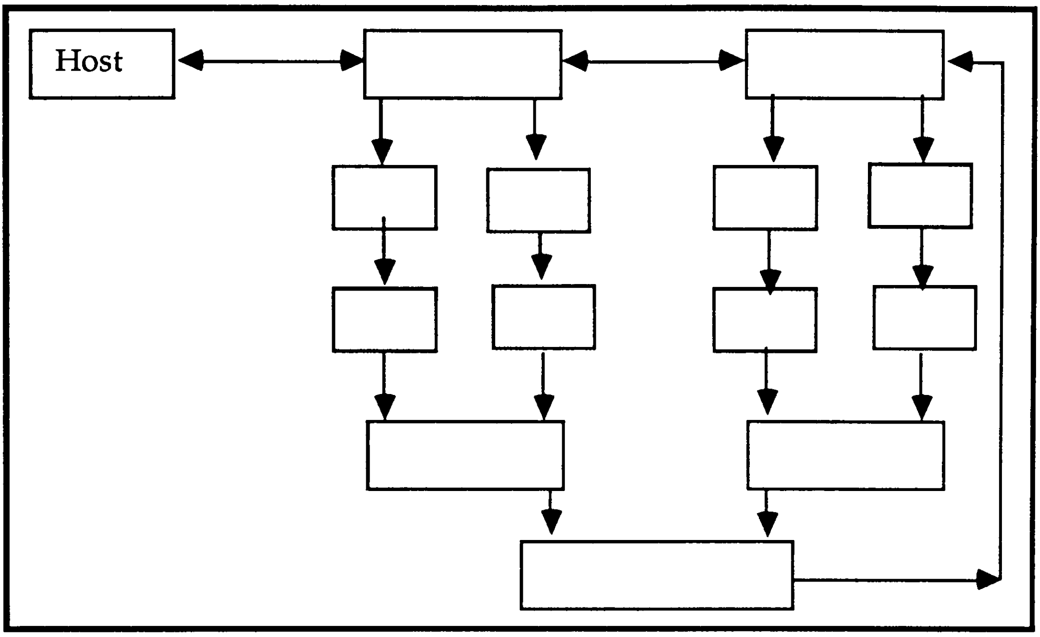


Fig 1.8: Arrangement of transputers for algorithmic parallelism using pipes
Arrows indicate the direction of data flow

Our implementation is based on algorithmic model of parallelism. The whole basis table is partitioned and distributed over the processors connected along the spine. These processors are asked to locate particular basis states by other processors which are engaged concurrently in the multiplication of the Hamiltonian. This interchange of information takes place through the exchange of packets. These processors communicate among themselves to locate the basis state in their respective partitions of the basis table. Once the state has been located then all the necessary information relating to the state like the block pointer where the state exists in the global table, the value of amplitudes etc., is bundled in the packet and transmitted to the calling process after necessary handshaking. Complete detail is given in the next chapter section 3.3.2.

1.2.2.2.4 Combinations :

There can be many different ways in which these three types of parallelism can be combined. Two of these seems to be particularly promising, namely using algorithmic with event and geometric.
The reason for these combinations is that algorithmic parallelism seems to give reasonable efficiencies for small number of processors, but it drops as the number of processors increases. On the other hand geometric parallelism needs a large data area, and direct communication to each of its neighbors, in order to achieve high efficiency. Thus replacing a single transputer in the geometric case by a cluster of transputers in an algorithmic network can work very well. This also increases the amount of data that can be stored at every node.

1.2.2.3 Code Migration :

Over the last three decades people have been developing codes for different applications, bearing in mind the picture of a sequential machines based on the Von Neuman architecture. But in order to benefit from the full potential of the new supercomputers, major modifications to the way in which the codes had been written become necessary. First we describe briefly, why changes to the existing millions of lines of codes are necessary and then propose some ways of achieving it.

1.2.2.3.1 Reasons for Code Migration :

(i) Since the last many years codes have been developed with the perspective about machines that they could only be sequential in their pattern of execution. The result was that any parallelism, which may have been apparent in the mathematical formulation of the problem, got lost in the process.

(ii) The choice of numerical algorithms in solving problems have strongly been influenced by the architecture of the available conventional computers. That is one reason we see implementations of inherently recursive procedures for the problems for which equivalent iterative algorithms have been available.

(iii) The implementation language used almost exclusively by supercomputer users has been fortran, which is a very awkward tool for expressing parallelism (ref. "GOTO considered harmful" Dijkstra[40]). Significant efforts have been made to improve the 'intelligence' of the automatic vectorizing-compilers, in the last few years. But they are still limited in their capabilities. One can say with a fair degree of certainty that , in the foreseeable future, these vectorizers will not be able to substitute a serial (recursion) algorithm by a parallel one.

1.2.2.3.2 Techniques for Code Migration :

(i) The easiest of all is to remove/change any incompatibilities in the software and/or the machine dependent parts (like word length etc.) from the code.

(ii) We can help the compiler in identifying the constructs where vector instructions may replace sequences of scalar instructions. This could be achieved by simplifying the loop structures and/or changing the order in which loops are nested and also trying to get around the non parallel nature of fortran. This modification of the source code can help the compiler to generate more parallel or vectorized code. Such current pre-processors have achieved only a limited success, because the resultant source code is not always readable and hence poses a greater problem of maintainability.

(iii) One practical but time consuming approach can be to spot the most expensive parts of the code and rewrite them. This, however, can be viewed as a local algorithmic change of the code and does not modify the global serial nature of the implementation of the problem.

(iv) The best but expensive in effort would be to throw away the existing code and go back to the orignal mathematical formulation of the problem. Choose the most suitable algorithm and architecture and code it in a way that it takes full advantage of the inherent parallelism of the algorithm.

In our implementation of , Parallel GLASNAST we have used the technique no. (iv) above, and everything has been built in line with the preconceived parallelism in the shell model calculations.

1.3 Choice of the Implementation Language :

Ideally, a language should possess two qualities for the efficient implementation of a project which tends to be distributive in nature.

- **Flexibility -** It must be flexible enough in adapting an algorithm in an optimal fashion to the hardware of the machine. This generally requires a language with special vector/parallel syntax constructions. More importantly it should allow the programmer to express the algorithm in a natural, abstract and modular fashion. The modern languages which are based on data abstraction are 'C', Pascal and Ada.
- **Continuity :** Historical continuity is equally important in the choice of an implementation language and it is of crucial importance when it is intended to run an old implementation on a new machine. Fortran is rich in this quality. But unfortunately it is diametrically opposed to other requirements of flexibility.

The first language which expressed parallelism consisitently was Iverson's APL[23]. In fact it was aimed at the concise expression of problems and not their parallel evaluation. It was based on mathematical concepts and notations, and as such it contrasted very sharply with other available languages in the description and manipulation of data structures. This lead to the development of an idea that a vector processor could be developed with Iverson's APL instrcuctions hardwired into the machine. This idea was brought to fruitation with the introduction of CDC Cyber-205[24] in 1981.

On the other hand, we observe that the fortran has been used almost exclusively for expressing parallelism in the scientific community using supercomputers. One important reason for that is again its historical continuity. We have also observed that when optimum parallel performance of an algorithm is the objective then fortran is less suitable than some other available modern high level languages. This is more so, when the implementation is to start from the algorithm level. In this section we briefly compare the advantages and disadvantages of using the three prospective candidates namely Occam, fortran and 'C'.

1.3.1 Standard Fortran :

Fortran has the advantage of being most widely used plus the availability of large volumes of possibly reusable code. But even many of its most outspoken advocates do not regard it as an ideal tool for scientific and/or parallel programming. Dijkstra's ' GOTO considered harmful' gives an enlightening account behind this proposition. We briefly summerise its widely used constructs which infact are considered by many as its deficiencies, both as a sequential as well as parallel programming language.

(i) COMMON and EQUIVALENCE : These constructs are used for storage allocation. But in fact COMMON makes it impossible to define any scope of variables other than local or global. This implies that it is impossible to restrict the scope of variable to a group of subroutines only.

The EQUIVALENCE statement. is used to set two variables to point to the same location. This practice allows potentially dangerous aliassing of memory locations even between the variables which are of different data types.

(ii) Rigid Source Form : Although we find that the punch cards have been outdated and discontinued on nearly all the machines, but the source form continue to be based on the same outdated model. This is very inappropriate for entering and editing the code on the terminal.

(iii) Lack of Data Structures : Fortran offers only two simple data structures , namely 'array' and 'common block'. Array is also restricted to the elements of same data type, while common block cannot be manipulated as an entity. We feel that the lack of data structures alone is a single big factor which makes it so inflexible.

In our implementation of PARALLEL GLASNAST, packets are exchanged between different processors. A packet is a structure, which tags together many fields, some of them being structures themselves. It seems impossible to achieve this level of abstraction, flexibility and conciseness of expression by the data structures available in fortran.

(iv) Lack of bit data type : Bit is the most fundamental data type and is manipulated by many applications. Many algorithms in physics, including our new shell model implementation requires bit manipulation. In fortran, however this is achieved by using means which are non standard and non portable. This has to be done because of the absence of this useful data type.

(v) Handling of Error Conditions : Error detection and recovery in fortran is fairly inadequate and is restricted to only parity and end of file condition on external files. But other common and important error conditions like overflow, underflow etc. are not specified in the language.

(vi) Precision Control : Fortran seriously lacks the means to control the precision of computer arithmetic. The real and double precision data types do not have guaranteed ranges or degrees of significance and there is no guarantee that the computer arithmetic is stable across a range of architectures.

(vii) Inadequacy of Vector Processors for Supercomputers: The most profitable targets for vectorization are the pieces of code where a single block of instructions is repeatedly being executed. Such looping constructs are usually expressed through DO loops. Vectorization through DO-loops is known as automatic vectorization.

The first drawback in the most versions of vector extended fortran is the lack of compatibility among different versions, available on different machines.

The second drawback of the current vector dialects is their low level character, bearing resemblance only to an assembly language than anything else. This actually emanates from the fact that the explicit vector instructions are required to be inserted inside the code, so that the compiler can detect the vectorizable part of the code. But this renders the program even more unreadable in the presence of nested DO loops and

GOTO's.

Lastly, the limited set of control structures in vector-extended fortran dialects is another serious limitation in allowing it any reasonable degree of freedom and flexibility.

1.3.2 OCCAM :

Occam[14, 15] is the language which is based on the Hoar's CSP[6] (Communicating Sequential Processes) and its precise hardware realization has been built in the form of Inmos transputer. Communication between the occam processes takes place via one way communication channels.

This contrasts sharply with another concurrent language Modula-2[25] and Ada[26], in which inter process communication between concurrent processes takes place via shared memory. They use monitors to guarantee mutual exclusion to the accessing processes.

The Occam implementation of concurrent processes does not require a shared memory, because each transputer has its own fast on-chip RAM. This is how an Occam model of concurrency avoids problem of contention and ensures a secure and side effect free language for a mutiprocessor system. The Modes of achieving concurrency in Modula-2 and Occam are compared in the fig 1.9 below.

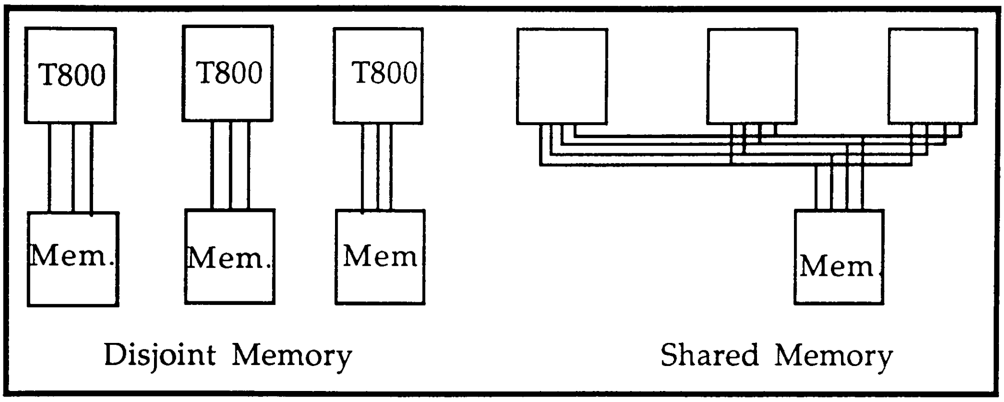


Fig 1.9: Modes of memory access in concurrently executing processors

1.3.2.1 Language Constructs :

In Occam programs are constructed from three primitive processes, namely assignment, output and input. Assignment statement like 'x:= y', sets the value of a variable 'y' to an expression. The statement 'c!y', outputs the value of the variable' y' onto channel 'c'. The statement 'c?x' sets the value of variable 'x' to the value input from channel 'c'. Channel itself is an unbuffered structure which permits the flow of information only in one direction. In other words, a channel can be viewed to be a write-only element for the transmitting process and read-only for the receiving process. Also a transmitting process can only write when the channel is empty and a receiving process, of course can read only when the channel has been written to.

Occam has got three control structures for controlling the order of execution of the processes. They are, SEQ (sequential), PAR (parallel) and ALT (alternate). SEQ and PAR determine whether the processes in the following list should be executed in serial or in parallel mode. ALT causes exactly one from the following list of processes to be executed. Apart from that, conditional constructs like IF and WHILE have also been provided.

1.3.2.2 Disadvantages :

In spite of the great advantages that Occam can offer, we have declined to use it for the implementation of our project for some of its handicaps which have been given below.

(i) It is a very low level language. A choice in favor of Occam would be analogous to the choice between the efficiency (but hazards) of assembly languages and the power (but constraints) of the high level languages.

(ii) Occam does not support any programming constructs which the transputer does not support. For example, recursion is not allowed in Occam.

(iii) The other principal limitation of Occam is the complete absence of dynamic data structures and no support for restrcturing of data types. We realise that the efficiency in the execution of a language is as important as its ability to provide constructs for encapsulating elements of divergent data types, which can then be quickly transmitted over the channels and keep the traffic lines less loaded for most of the times.

In our shell model project we have used such dynamic data structures to encapsulate information of divergent data types into a *packet*. This packet is then transitted to any process-or on the spine by just a single write to an output channel, after the handshake. Contrarily, it could be transmitted by piecemeal and will keep the channels busy for longer time, consequently resulting in longer idle waits for other inter process communications.

1.3.3 Parallel 'C':

The language 'C'[7] has been synonymous with Unix, because this most widely used operating system has been mostly written in this language. Language C matches the capabilities of many computers. It is easier to write portable programs in this language because it is independent of any particular architecture.

Parallel 'C' [28, 29] can be considered to be a superset of 'C', which includes the communication protocols between different channels. We summerise below the outstanding features of 'C' and parallel 'C', as we have chosen this language for our implementation.

(i) **Fundamental Constructs :** The basic data objects provided by 'C' are characters, integers of several size and floating point numbers. It also provides derived types which include enumeration, pointers, unions and structures.

A well structured program requires constructs for the flow of control. In this respect it provides, IF for decision making, WHILE and FOR for looping with a choice for having termination test at the top or at the bottom of the loop. SWITCH is provided for selecting one out of a set of many cases. The parallel 'C' library has a number of functions for reading from and writing to the channels, apart from the tools for achieving process synchronization.

(ii) **User Defined Data Types:** This is a very powerful facility provided by the language. This gives the user freedom of defining such data types which can conform to the requirements of the algorithm and manipulate data more efficiently.

For instance in our implementation we have defined a data type 'basis', which is in fact a structure with seven fields, some of them are structures themselves. A structure of type basis carries with it complete information about a basis state as required by our algorithm. It can be passed around more conveniently and efficiently than passing its individual fields. We have made extensive use of this powerful tool for flexibility and abstraction in defining many data types appropriate to the implementation.

(iii) **Dynamic Data Structures** : The availability of pointers and the ability of the language to do address arithmetic adds a new dimension to its flexibility. In 'C' arguments to the functions are passed by copying the value of the arguments. In Pascal and Algol68 the same effect is called 'call by value'. The 'call by reference' in 'C' is achieved by passing a pointer explicitly. Also, in the case of array names the address of its origin is passed, which means that array arguments are effectively call by reference. Provided the pointers are handled carefully, they introduce a good degree of tidiness in the code apart from delivering the execution efficiency between different parts of the program.

As an example, in our implementation, a pointer to the basis table could be passed between different processors and any part of the data contained in the table could be accessed and manipulated.

(iv) **Pointers to functions** : Unlike many other high level languages, functions can also passed as arguments but through a pointer to the function, because the function itself cannot be a variable. The pointers to functions can be placed in an array and manipulated like any other pointer.

(v) **Dynamic Storage Allocation** : The type structure of 'C' language is similar Algol68 and Pascal. It differs in the strictness with which type mismatches are treated. e.g. some 'C' compiler permit the assignment of a pointer value to an integer variable.

This permissive approach to types adopted in 'C' allows programs such as storage allocator to be written. A statement like

```
        basis_result = (basis  *) malloc( number_of_states * sizeof(basis) )
```

allocates a memory block of size which can hold all the number of states in the basis table. But in a situation where it is not sure about the size of the data being generated during run time, a construct like *realloc ()* can reallocate the space and save the program from crashing for want of memory space.

(vi) The 'C' is fundamentally a stack base language and is best suited to exploit the fast on-chip memory in the transputer for the stack.

(vii) **Operators :** It provides logical operators which many other high level languages also provide. But it also provide many bit level operators which are very helpful in many physical problems. The bitwise operators like AND, OR, XOR etc have been particularly useful in our project when the application of the Hamiltonian is done to the Slater Determinants (ref. section 2.2). Creation and destruction of particle is efficiently achieved by using the bitwise operations to the wave function.

(viii) **User defined Macros :** A user can create macros for certain objects which he thinks are going to be used more frequently than the functions. This is a great flexibility which ensures not only a good measure of tidiness in the program but also ensures its proper readability and hence maintainability. Most 'C' implementations extensively make use of this facility and ours is certainly one such an example.

(ix) **'C' on Supercomputers :** We find, that after its unparalleled success in system programming, 'C' is now being used on supercomputers for some of the very attractive features outlined above.
A superset of 'C', the language C* has already been developed by the Thinking Machine Corporation, to run on their highly parallel computer (65535 processors), the Connection Machine CM-2[11]. Some constructs of C* are very close to C++[27], which is another superset of 'C'.

The main draw back in 'C' is the lack of historical continuity, particularly from the point of view that most of the codes in Physics and indeed in other natural sciences have been written in fortran. However this problem becomes irrelevant in cases like ours, where completely new codes are to be developed for parallel computers.

Looking from another angle, the transforming of code from traditional supercomputers to fine grained parallel computers or indeed writing a new code for it, gives the physicists an ideal opportunity to step out of the software straight jacket of the nineteen fifties.

CHAPTER 2

DASS
Dynamic And Structured Shell model

Introduction :

In this chapter we describe the organisation and implementation of our new shell model program, which is based on exploiting the dynamic data structures. We have used the language 'C'[7] for this purpose. This new version has been developed with a view to make it distribute for running on the Meiko Supercomputer.As it has been developed from the start, we begin with explaining our choice of preferences on which this project has been built. We also explain the project and its variant features in comparison with the previous implementation. For completeness, however we begin with a brief outline of the main features of the old shell model program.

2.1 Choice and Format of Representation :

Two basic operations involved in the shell model calculations are the multiplication of a state by the Hamiltonian and orthognalization of two states. In principle these operations could be defined in any representation of the wave function and the Hamiltonian. But there are certain trade-offs in using either of such representations. We have used Slater Determinants for the representation of the basis states along with the m-scheme and Lanczos method for diagonalization of the Hamiltonian. We would explain here, our chosen representations, their advantages and [illegible] how they are superior to some of those used in the previous implementation[1].

2.1.1 Slater Determinants :

This representation finds its roots from the principle that the many-nucleon wave functions must be antisymmetric under the exchange of any two nucleons.

Suppose $\psi\alpha(1)$, $\psi\beta(1)$ and $\psi\alpha(2)$, $\psi\beta(2)$denote wave functions of particle numbers 1 & 2 and α, β represent some quantum numbers of the state. For such a system of two nucleons, a Slater determinant will be given by

$$\phi_{\alpha\beta} = \frac{1}{\sqrt{2}} \begin{vmatrix} \psi_\alpha (1) & \psi_\beta (1) \\ \psi_\alpha (2) & \psi_\beta (2) \end{vmatrix} = \frac{1}{\sqrt{2}} [\, \psi_\alpha (1)\, \psi_\beta (2) - \psi_\alpha (2)\, \psi_\beta (1) \,] \quad ....(1)$$

The quantum number α could be represented, say in spherical representation by a = (n l j m t_3), where t_3 is the neutron or proton label. The only important information it gives is which states are occupied.

We can represent exactly the same information in terms of creation operators.

$$\phi_{\alpha\beta} = a_\alpha^\dagger\, a_\beta^\dagger \mid 0>$$

where $\mid 0>$ represents vacuum indicating a state with no particles in it. The commutation relation of these creation operators is defined by

$$a_\alpha^\dagger\, a_\beta^\dagger = -\, a_\beta^\dagger\, a_\alpha^\dagger \qquad .... (2)$$

It can be noticed from eq. (1) that $a_\alpha^\dagger\, a_\beta^\dagger = 0$ when $\alpha = \beta$, and same fact is reflected from eq.(2).

We introduce another operator 'a', which operates on the Slater determinants and creates particles in the desired orbits. The following relations hold for the creation and annihilation operators.

$$a_\alpha\, a_\alpha^\dagger \mid 0> = 0, \quad a_\alpha \mid 0> = 0,$$

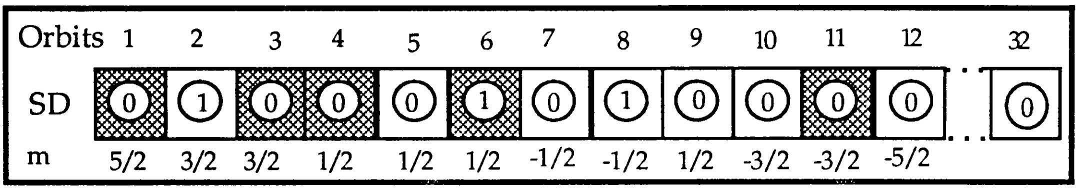
$$\text{and as } a_\alpha\ a_\beta^\dagger = \delta_{\alpha,\,\beta} - a_\beta^\dagger\, a_\alpha^\dagger , \quad a_\alpha^\dagger\, a_\beta^\dagger = \delta_{\alpha,\,\beta} |0>$$

Using relation for the hermitian adjoint of ø i.e. $ø_{\alpha\beta} = <0 \mid a_\beta\, a_\alpha$ and $\psi_{\alpha\beta}^\dagger\, \psi_{\alpha\beta} = 1$, we can write the anticommutation relations as:

$$[\, a_\alpha^\dagger, a_\beta^\dagger\,]_+ = 0 = [\, a_\alpha\, , a_\beta]_+; \quad [\, [\, a_\alpha^\dagger, a_\beta\,]_+ = \delta_{\alpha,\,\beta} \qquad .....(3)$$

The above discussion concludes that the wave function of n nucleons in different orbits can be represented as product of n creation operators applied to vacuum. So in any n particle system, if the states are ordered by some means then we can form the whole basis from the Slater determinant $\mid n_1 \ldots n_i \ldots .>$, where n_i denotes the number of particles in state i. In order to meet the requirements of the Pauli's principle n_i can only be 0 or 1.

This formalism is ideally suited to the computer representation of the wave function, because creation of particle in a single particle state i sets a particular bit while annihilation resets it. A Salter determinant like |ɑ1000101 > will be exactly represented in a 32-bit computer word as shown in fig 10, which could equally be considered as binary number.



Orbits	1	2	3	4	5	6	7	8	9	10	11	12	. . .	32
SD	0	1	0	0	0	1	0	1	0	0	0	0	. . .	0
m	5/2	3/2	3/2	1/2	1/2	1/2	-1/2	-1/2	1/2	-3/2	-3/2	-5/2		

Fig 2.1 : Representation of Occupied orbits by a bit map in a 32 bit word

It is pointed out here that in the diagrams for SD's we have shown the numbering of orbits in the usually familiar pattern i.e. from left to right, while in actual hardware representation is from right to left. This thing has been adequately taken care of at the implementation stage. The SD representation had been used in the Glasgow Shell model[1] earlier and we have chosen to maintain it for some of the following reasons.

(i) We can start the calculations with an explicitly antisymmetric wave function which remains antisymmetric even after the multiplication of the Hamiltonian. Other conventional representation in which the antisymmetry is not explicit gives trouble when used with the Lanczos approach. It is possible that the necessary antisymmetrisation may destroy orthognality and vice versa.

(ii)The elegance of this representation is that it maps beautifully to the computer hardware and provides us with the flexibility of bit manipulation during the multiplication of the Hamiltonian and also at other stages of the calculations.

But there is a price that has to be paid for getting these advantages.

We do not use coupling of angular momentum(A.M.) in this representation. This simply means that our basis functions are not eigen functions of J and T. This results in the loss of these important quantum numbers. Its other implication is that a shell model calculation cannot automatically split into a number of smaller problems for different values of J and T. Consequently, calculation for a much larger number of basis states has to be carried out even when a fewer states are required.

On the other hand, when this SD representation is used along with the m-scheme and Lanczos method (explained in sections 2.12 & 2.4) the benefits achieved far outweigh the drawbacks mentioned above.

2.1.2 The m-Scheme and the Generation of Basis States:

In our representation, an SD represents one basis state and is stored in one computer word, which is 32 bit long in most of the modern processors including transputer T800. As SD shell contains only 12 single particle orbits, so only 24 least significant bits are practically used for the representation of any basis state. For calculations extending beyond the sd shell, however, the remaining 8 bits can be activated.

For the sd shell, the lower 12 bits have been chosen (arbitrarily) to represent orbitals for protons and the upper 12 single particle orbitals are meant for the neutrons. Every bit 'i' has got associated with it a definite value of 'm_i', where mi is the z-component of the total angular momentum in the ith orbit. These m values for respective orbits are shown in fig 2.1. The total M contribution from np protons and nn neutrons is the sum of all the Mn values for neutrons and Mp values for protons, which of course depends upon the orbits being currently occupied.

$$M_J = Mp + Mn = \Sigma_{pi}\ m_{pi}\ + \Sigma_{nj}\ m_{nj}$$

where mpi and mnj represent value of the z component of A.M. for proton and neutron respectively in their ith and jth single particle orbit. The summation extends to number of protons and neutrons. It is clear at this stage that the parameter which m scheme requires include, no. of protons, no. of neutrons, total M value for the state, parity where appropriate and the subshell occupancy if particles are being restricted.

The generation of a complete basis table for a given set of parameters amounts to producing all the 24-bit words with np protons and nn neutrons confined to the upper and lower 12 bits respectively and each basis state having total Σ mi contribution equal to the required M_J. This is achieved by first filling the left most np proton bits and then the left most nn neutron bits by 1 .Now we successively move the rightmost set bits one place to the right, in both cases. Every time we do this, it produces a new 24 bit word. Finally we check, if the total M value of an SD comes out to be equal to the required MJ we store it, otherwise discard it. This also ensures a list of valid SD's in descending order and it rather looks like translating a number into binary notation by subtracting successive powers of 2 in descending order.

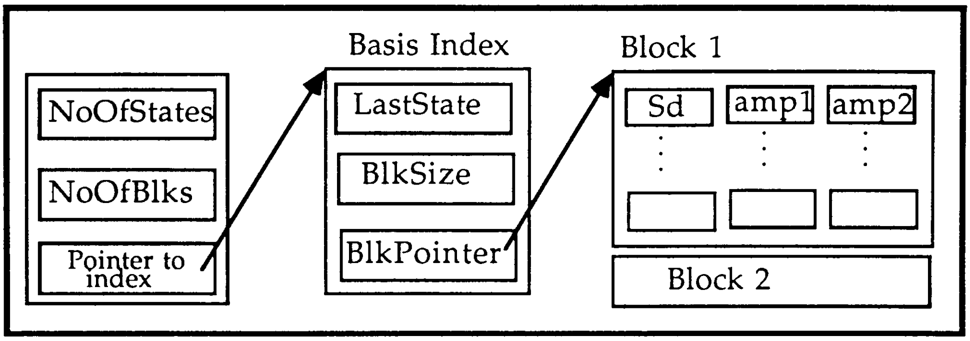


Fig 2.2 : Arrangment of the basis table in the new program

Figure 2.2 shows the general arrangement of the basis table in the new program. New features are the introduction of a block structure in which subsets of the basis states together with their amplitudes are stored dynamically, i.e. the storage blocks containing given subsets have no pre-determined relationships in their memory locations. This allows greater flexibility by allowing the required space to be dynamically allocated and also prepares the way for putting the blocks onto different processors as described in the next chapter. This non contiguous block structure necessitates an accessing table (the block index) which contains for each block the number of basis states in the block, the last SD in the block (to help locate SD's) and a pointer to the beginning of the block. In addition to these there are these global variables which record the total number of basis states, the total number of blocks and a pointer to the block index. The last is necessary because there may be more than one Basis Index each of which refers to its own basis table. This would the case, for example, if transition rates between the states with different total M are to be calculated.

2.1.3 Limitations of m scheme in very large Calculations :

One of the objective of this whole project is to provide an instrument for the shell model calculation beyond the sd shell. So it becomes imperative to discover the limitations (if any) at each stage of its development.

- Firstly, we notice here that, presently 32 bit words are available on most of the processors. If we increase the model space such that the size of the SD word exceeds the size of the CPU word, then there is bound to a loss of efficiency in the primitive manipulations.
- Secondly, potentially there could be problems of memory shortage for large calculations, because the whole basis table is required to reside in the primary memory throughout the whole calculation.

But further investigation reveals that for any sd shell calculation this demand does not exceed 1/2 Mbytes, given that each SD is 32 bits and the two iteration vectors occupy twice this space. This much space seems to be affordable on most of the serial/parallel machines. But for the bigger pf shell calculations, the word length for the SD can be upto 128 bits. In such calculations using a space of dimension 10^6 would require 16 Mbyte for the storage of basis table alone.

By the strength of these arguments, and otherwise, we feel convinced that there is little chance of running a calculation of this size on a serial machine. However, on a parallel processing machine, chances of success can be fairly good provided we use shared data structures. Also the new products in parallel architecture appear to be approaching very fast to a stage where a memory requirement of this size would appear just modest. For instance, the latest addition to our Glasgow Meiko Supercomputer is a domain of two T800 transputers, each with 8Mbytes of memory on board.

It seems fairly convincing that m scheme can be used in quite a beneficial way in the representation of basis states. That is why we have chosen to continue with in DASS and Parallel GLASNAST (ref. chapt. 3).

2.2 Operations on Slater determinants and Phase Calculations :

The shell model Hamiltonian can be expressed in the occupancy number representation, which would include creation and annihilation operators. We demonstrate here the effect of these operators on the SD's in terms of their representation and how the same effect has been achieved by us using equivalent bit level operations. For the sake of simplicity and clarity we confine ourselves to the 12-bit word in these examples.

Consider a state having particles in orbits 1, 2, 4, 9 and 11 which would be represented by an SD

ø = 1 1 0 1 0 0 0 0 1 0 1 0 and also

a_1 ø= 0 1 0 1 0 0 0 0 1 0 1 0

where a is the annihilation operator

In DASS we achieve the same effect by subtracting the logical sum (i.e AND) of ø and the bit mask.

ø = 1 1 0 1 0 0 0 0 1 0 1 0, mask for a_1 =1 0 0 0 0 0 0 0 0 0 0 0,

ø AND mask = 1 0 0 0 0 0 0 0 0 0 0 0

a_1 ø = ø - (ø AND mask)= 0 1 0 1 0 0 0 0 1 0 1 0 .

Similarly, we can apply creation operator $a^\dagger$ to the new SD, say by doing $a6^\dagger$ (a1 ø). To ensure that orbit 6 is empty for creation we simply take logical sum of the SD with mask 0 0 0 0 0 1 0 0 0 0 0 0 and a zero result confirms it.

These creation and annihilation operation are straight forward in a language like 'C', which provides adequate bit level constructs.

2.2.1 Phase Calculation of Slater Determinants :

Consider an operation of creating a particle in, say, 7th orbit. This can be equivalently represented by

$a_7{}^\dagger$ ($a_1{}^\dagger$ $a_2{}^\dagger$ $a_5{}^\dagger$ $a_9{}^\dagger$ | 0> which is equal to

$(-1)^3$ ($a_1{}^\dagger$ $a_2{}^\dagger$ $a_5{}^\dagger a_7{}^\dagger a_9{}^\dagger$ | 0>) because of the commutation relation $a_i{}^\dagger\ a_j{}^\dagger = -\ a_j{}^\dagger\ a_i{}^\dagger$

So every time a creation or annihilation operator is applied, the resultant SD is multiplied with a factor of $(-1)^n$, where n is the number of filled orbits (set bits) starting from the first orbit to and excluding the final orbit where the particle has been created/destroyed.

During each Lanczos iteration this process is repeated hundreds of millions of times, even for quite small calculations. So any inefficiency at this stage could add hours in the total execution time of the program. Earlier Glasgow shell model program[1] used a very efficient algorithm, implemented in assembly language. But this seriously affected the portability of the program. This algorithm has been referred as 'Occupancy Algorithm' in the ensuing discussion.

We have now discovered a more efficient algorithm for the calculation of phase. We call this algorithm 'Parity Representation Algorithm'. It is ideally suited for any language which provides reasonable bit level operations. Our DASS and Parallel GLASNAST implementations use this new algorithm for the phase calculation. We explain both the algorithms in turn.

2.2..2 Phase Calculation Using Occupancy Representation Algorithm:

Consider an SD ø shown in fig 2.3 and operate on it by a_{11} a_1, which also shown in the same figure.

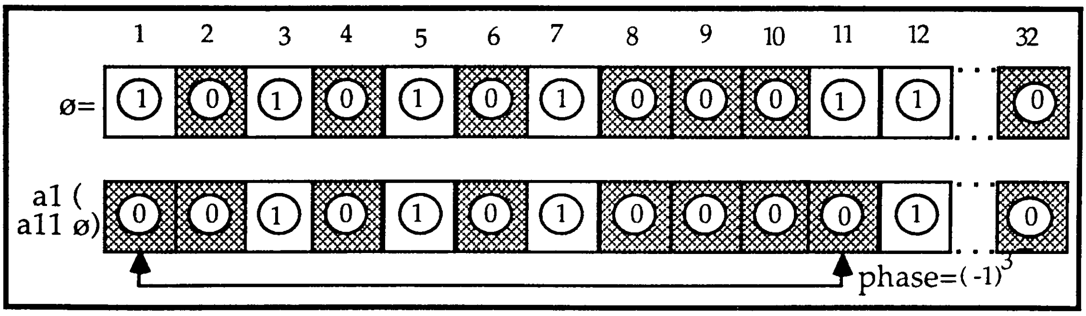


Fig 2.3 : Use of operators on SD's and Calculation of phase

From the implementation point of view :

ø= 1 0 1 0 1 0 1 0 0 0 1 1

Mask for bit 1, Mask1 = 1 0 0 0 0 0 0 0 0 0 0 0, mask11= 0 0 0 0 0 0 0 0 0 0 1 0

We find M =M1 - M11 = 0 1 1 1 1 1 1 1 1 1 0 0

Now the word containing the phase bits of the new SD (denoted by ph) is obtained by finding ø AND M.

ph= 0 0 1 0 1 0 1 0 0 0 0 0

This ph in general, will contain only those 'n' bits in this SD, which contribute towards the phase of the new SD. The only job left now is to count these bits. In the absence of bitwise operations in fortran, this count is achieved by the following algorithm.

```
        Function ReturnPhase(Ph)
        NumOfOnBits =0
        While  $ph \neq 0$  do
                temp = ph -1
                ph = ph AND temp        /* this ph will be same */
      /* as previos one execpt the right most '1' bit replaced by 0.*/
                NumOfOnBits = NumOfOnBits +1
        end  (* while *)
        Return ( $(-1)^{NumOnBits}$)
```

This method takes 5 x (number of 1's) operations, in returning the phase.

2.2.3 Parity Representation Algorithm :

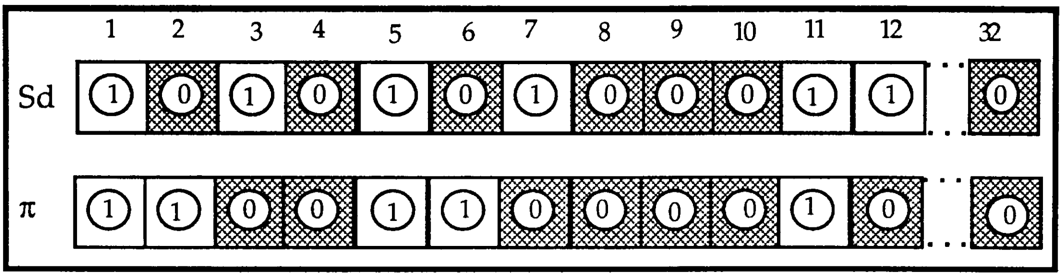
It needs to be pointed out that the term 'parity' refers to its meaning as used in computing science with regards to the transmission of data and not as normally used in physics. By using this method SD's are converted into new representation, in which the phase can be calculated without going through the subtractions and the while loops of the last section. Also conversion from parity, back to occupancy (if needed) can be achieved in just one step. An SD, converted into parity representation, has been denoted by 'π' in further discussions. This conversion algorithm is elegantly simple and is based on setting or resetting the present bit (of SD), depending upon its current status as well the status of the previous bit.

```
        Function Occ_to_parity ( SD)

        Previous_Bit = 0;   Parity_Rep_Of_SD =0
        For ( i = NumberOfOrbits - 1;  i >= 0;  i--)
        { If ($SD_i$ =  1) and  (Previous_Bit=0)  /*i.e if ith bit of sd */
            then    $\pi_i$= 1            /*is set and its prevous bit is 0 */
              Previous_Bit=1                        /* , then ith bit*/
          elseif  ($SD_i$ =1) AND (Previous_BIT = 1)
```

```
                then $\pi_i$ =0 ; Previous_Bit = 0
            elseif  (SD$_i$ = 0) AND (Previous_Bit =1)
                then  $\pi_i$ =1; Previous_bit =1
            else   $\pi_i$ =0; Previous_Bit =0
        }
        Return ( $\pi$)
```

The fig 2.4 shows conversion of an SD to parity representation.



	1	2	3	4	5	6	7	8	9	10	11	12	. . .	32
Sd	1	0	1	0	1	0	1	0	0	0	1	1	. . .	0
π	1	1	0	0	1	1	0	0	0	0	1	0	. . .	0

Fig 2.4 : Conversion of Slater Determinant into Parity representation (π)

The following algorithm converts parity to occupancy representation.

```
        Function parity_to_occ ( $\pi$ )
                SD =  ( $\pi$ ) XOR ( $\pi$ >> 1) /* logical shift one to $\pi$ */
                Return ( SD)              /* take XOR with $\pi$   */
```

Consider an SD 1 0 1 0 0 1 its parity rep. will be π =   1 1 0 0 0 1

(π >> 1) =   0 1 1 0 0 0

SD = π XOR (π >> 1) =   1 0 1 0 0 1

The crucial thing is that in parity representation a set bit means that there is an odd number of occupied orbits to left of and including it in the occupancy representation.

2.2.3.1 Using Operators in Parity Representation :

We can use creation and annihilation operators on π's to yield result in π. Suppose we want to create a particle in the ith orbit. First we create a bit mask with all the bits to the left of 'i' to be zero and all the bits to the right and including i to be 1. In case the particle is being destroyed in the ith orbit then the mask will have only ith bit to be 1 and all the rest will be 0. In next step we find (π XOR mask) which will be the new value of π after creation/ destruction of particle in the ith orbit. If many particles are being created/destroyed then just make as many masks a, b, c ... and operate

$$\pi = ((\pi \text{ XOR } a) \text{ XOR } b) \text{ XOR } c$$

2.2.4 Phase Calculation Using Parity Algorithm :

Once the new value of π is available then the calculation of phase is very trivial in this representation. It requires only one access to read the status of the π_i bits, i being the orbit(s) number(s) where particle(s) have been created/destroyed. If n happens to be the sum of the status values of these bits then phase will be $(-1)^n$.

We try to compare the phase evaluation in occupancy and parity representations in the following example, from the implementation point of view, where we operate by a11 a2.

	Occupancy Representation		parity Resentation
SD =	0 1 1 0 1 1 1 0 0 0 1 1	π	= 0 1 0 0 1 0 1 1 1 1 0 1
		Mask2	= 0 1 1 1 1 1 1 1 1 1 1 1
a2 ∣ SD > =	0 0 1 0 1 1 1 0 0 0 1 1 ↦	XOR	= 0 0 1 1 0 1 0 0 0 0 1 0
		Mask11	= 0 0 0 0 0 0 0 0 0 0 1 1
a11 a2 ∣ SD > =	0 0 1 0 1 1 1 0 0 0 0 1 ↦	XOR	= 0 0 1 1 0 1 0 0 0 0 0 1
	↑ ↑		↑ ↑
	phase = $(-1)^4$		Phase = $(-1)^{0+0}$

table 2.1 Calculation of phase requires reading the status of, only two bits in parity and ten bits in occupancy representation when operated by a11 a2.

The fig 2.5 below shows the machine representation aspect of the above example.

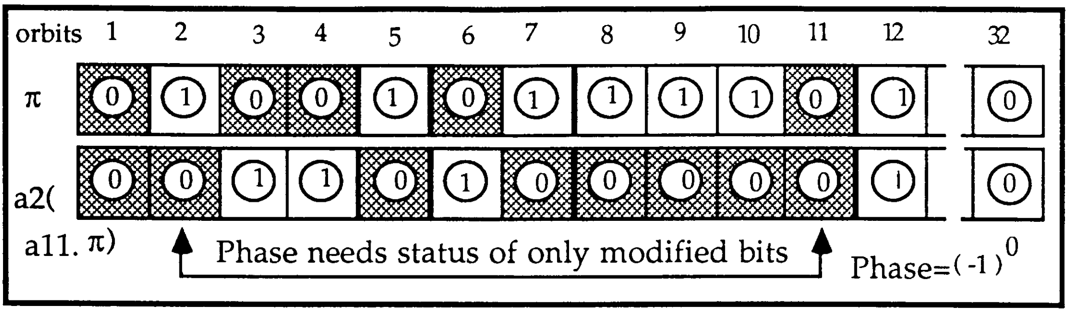

orbits 1 2 3 4 5 6 7 8 9 10 11 12 32
π
0 1 0 0 1 0 1 1 1 1 0 1 0
a2(
a11. π)
0 0 1 1 0 1 0 0 0 0 0 1 0
Phase needs status of only modified bits
Phase=(-1)^0

Fig 2.5 : In Parity algorithm, phase depends on the status of modified bits only

2.2.5 Comparative View of Parity and Occupancy Algorithms :

(i) The algorithm for phase calculation in occupancy representation tab. 2.1 shows that, at least 5 times the number of 1 bits (between the operated orbits) operations must be performed to calculate the correct value of phase. The worst case will occur when all the orbits are occupied and the operator used will be a1 a24 in case of sd shell, and a1 a32 for some nucleus beyond the sd shell. The occupancy algorithm will have to perform 5 x 22 and 5 x 30 operations respectively for the above cases. It may pointed out again here that the phase calculation has to be done during each Lanczos iteration hundreds of millions of times.

On the other hand, the beauty of the parity algorithm is that it has no worst case, as the calculation of phase. requires only one pass through π and always determines the phase in constant time. The reason is that it needs to read the status of only those two bits, which have been modified by the operators, whether they happen to be contiguous or on the two extremes of the computer word. The strength of this argument is such that we can feel confident that the time required for the phase calculation, for the shells beyond the sd shell will be constant for a particular calculation and will be independent of the no. of active orbits.

(ii) The above discussion may tend to give this impression that in the parity representation, SD's and π's are subjected to a lot of bit level operations and this should involve sufficient time overheads to render this algorithm less attractive than it has been presented here.

We have fully investigated any hidden extras before fully adopting it for our DASS and Parallel GLASNAST projects. We analyse this position to clear any hanging shreds of suspicion.

We apply creation/annihilation operators during unpacking of states i.e the application of Hamiltonian explained in the next section. At this stage we store lists of all possible pairs (of particles) that can be created/destroyed. This much has to be done, which ever algorithm one may choose to prefer.

We conducted a lengthy investigation into the possibility of using parity representation throughout the program in place of occupancy representation. It turned out to be awkward to do so because, although it is no more difficult to generate the basis set in parity representation we could not find a satisfactory way of generating it in numerical ordering (which is necessary for the efficient use of Lanczos algorithm). The basis states are still generated and stored in occupancy representation, but the parity representation is generated when required. This is done in fact when a basis state is presented to be operated by the Hamiltonian. The state has to be 'unpacked' to produce a list of occupied orbits and the parity representation can be constructed during unpacking with practically no loss of time. Thereafter the normal process of creating and destroying can be done by bit manipulation and the phase calculation becomes trivial. Generating the occupancy representation of a new state produced by the operation of H is also trivial.

Some further discussion about the parity algorithm has been given in ref [41], which has been submitted for publication/and also attached here as appendix .E E

2.3 The Hamiltonian :

After the appropriate choice of representation and the calculation of basis states, the next important step in the shell model calculations is the formation of the Hamiltonian. It is possible to choose the Hamiltonian in many forms. But, commensurate with our previous choice, we choose the occupancy number representation form of H. The Hamiltonian consists of one-body and two-body matrix elements.

$$H = \sum_{ik} H^{(1)}_{ik} \; a^{\dagger}_{i} \; a_{k} + \frac{1}{4} \sum_{ijkl} H^{(2)}_{ijkl} \; a^{\dagger}_{i} \; a^{\dagger}_{j} \; a_{l} \; a_{k} \qquad \text{........(4)}$$

where 'a' and 'a†' stand for creation and annihilation operators, which obey the commutation relations given in eq (3). The $H_{ij}{}^{(1)}$ and $H_{ijkl}{}^{(2)}$ are one and two-body matrix elements.

Here a potential problem can be encountered due to the fact that computer does not distinguish between a_i† a_j† and a_j† a_i†. We overcome this difficulty by introducing a standard ordering for the orbits and combining the two terms so that the Hamiltonian eq. (4) depends upon the total number of 'n' active particles and can be presented as a purely as a two body operator.

$$H = \sum_{\iota < j, k < l} \left[\frac{1}{(n-1)} H^{(1)}_{ik} \; \delta_{jl} \; + \; H^{(2)}_{ijkl} \right] a^{\dagger}_{i} \; a^{\dagger}_{j} \; a_{l} \; a_{k} \qquad \text{......(5)}$$

2.3.1 Application of the Hamiltonian:

Application of the Hamiltonian on the basis states is effectively the application of creation and annihilation operators on the SD's or π's.Each π from the basis table is taken in turn and the operation a_i† a_j† a_k† a_l† is applied to it . In the example below we explain this whole operation from the implementation point of view of DASS.

Consider an 8-bit SD word which has active particles in 2nd, 3rd, 5th and 7th orbit. The fig 2.6 shows that there are 6 possible pairs that can be destroyed and they are stored in a table. In practice, however we store the masks for each of these pairs (e.g. 00001010 for pair 5,7). Once the particles in orbits 5, 7 have been annihilated (by using π = π XOR Mask), then there could be a number of possible values of k and l where particles can be created, including 5, 7 (i.e. i and j) itself.

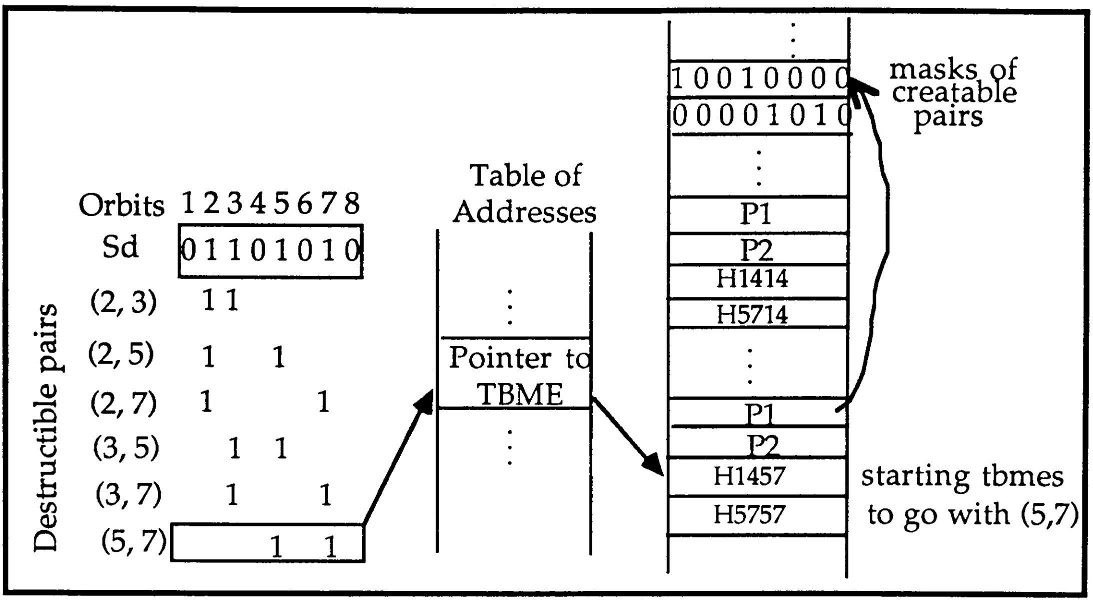


Fig 2.6: Layout of the the Hamiltonian tables and method of use

Each pair of destructible (i, j) particles, points towards to an address in the index table. This in turn points to a location where matrix elements corresponding to (5, 7) and other possible (k, l) pairs have been stored in contiguous locations. Above this address two pointers P1 and P2 have also been stored which point to the beginning and end of the creatable pair's (k, l) list. These pairs have the same value of M and Mt as the annihilated pair (i,j), 5, 7 in this example.

Once this list of particles is available at address P1, we simply run through the list until we find a pair which can be inserted in the original SD minus the pair (5, 7), without violating the Pauli's principle.

2.4 Diagonalization of the Hamiltonian & the Lanczos[30] Method:

The diagonalization of the matrix representation of the Hamiltonian is a very important step, because it yields the eigen values for the respective states. There are many different different techniques for achieving this objective depending upon the suitability of the application. For instance, Pick and Tomasek[31] have preferred Givens[32] and Householder[33] methods over Lanczos method, for their solid state physics calculations. The Glasgow Shell model[1] and many other high energy physics implementations have used Lanczos method or its optimised versions. Before actually adopting the Lanczos algorithm for our purpose, we need to convince ourselves by analysing some of the parameters, which are considered cruisal in computationally intensive physical problems.

(i) **Memory Requirements :** Consider the Hamiltonian to be an N x N symmetric matrix which we intend to tridiagonalize. Also we consider that the storage required by a particular method to tridiagonalize such a matrix would be , only that memory which has been modified by the procedure.

In a reorthognalized Lanczos method, we require space for two current Lanczos vectors vi and Hv_i -> v_{i+1} at any one time, in the fast memory. Also we need space for the Hamiltonian matrix elements, the basis table and space for the program.

The Givens and Householder methods proceed by transforming to zero all the off-diagonal elements in successive rows and columns. More importantly, each transformation alters all the elements of the block matrix H, which has yet to be trdiagonalized. as such all these elements must be stored in RAM. So these methods would require a RAM of the order of N^2 locations for the said matrix.

On the other hand Lanczos method for the same matrix would require only 2N locations in RAM. Also it does not modify H, so H does not have to be stored in RAM. The only memory requirement in this case would be for the current and the preceding vectors.

(ii) **Number of Multiplications Required :**

For the said matrix Givens and Householder methods would require 4/3 N^3 and 2/3 N^3 multiplications[32]. The Lanczos method (with or without reorthognalization) require order N^3 multiplications. But if the matrix has a special form (e.g. sparse matrix), the former methods do not take any advantage of its form and require same number of operations as for the full matrix, while Lanczos method does make use of this symmetry. For instance, if the N x N matrix H has M non-zero elements per row then Lanczos method would require MN^2 multiplications compared to

N^3 by the others two. In most cases if N is 1000 then M is of the order of 10. This gives Lanczos method a saving of factors of 100 or more.

(iii) Convergence of Eigen Values :

As explained in (i), the eigen values in two mentioned methods are not known until H has been diagonalized completely.

But the attractive feature of the Lanczos methods is, that the eigen values of the top left hand kxk block of the main tridiagonal matrix, converges very rapidly toward the eigen values of H, as k is increased. The eigen values which converge in this way, correspond to the lower eigen values of the spectrum. In shell model calculations, we are often interested in the low energy states. The Lanczos method guarantees to find these low energy eigen values in k -1 steps from the kxk submatrix. The resultant effect is that these eigen values are determined, without having to carry the Lanczos process to completion. Whitehead et.al[1] have shown that the first n eigen values $1 \leq n \leq 10$ always converge in less than 100 iterations. In DASS, a state is considered to have converged if its value remains unchanged upto 8 significant figures, from one to the next iteration.

(iv) Accuracy : The problem of accumulating rounding errors in the basic arithmetic operations had long been considered to be a major drawback in Lanczos method, and it rendered this algorithm to be labelled as 'unstable' in its early days. On the other hand, the remedy for instability (i.e full reorthoganalization) made this algorithm expensive, both in terms of arithmetic effort and in storage requirements.

Paige[34, 35] analysed it and found out, how the roundoff error perturbs the output from the Lanczos recurrence. He showed that the orthognality loss among the Lanczos vectors goes hand in hand with the convergence of Ritz values to the eigen value itself, (Ritz values are error bounds of eigen values computed at step say, k). He also showed that the information about eigen values was not corrupted by this orthognality loss and the eigen values could be found to the working precision, if enough steps (iterations) were taken. A number of his insignicant modifications made Lanczos algorithm, an algorithm of excellent stability and accuracy. A relatively more recent investigation by Parlett[36] has also proved that Lanczos method produces very accurate eigen values.

The above arguments lead us to conclude that for very large matrices, which can arise in shell model calculations, Lanczos method (with or without reorthognalization) is certainly superior to other methods in its class. We have adopted this algorithm in both our implementations.

CHAPTER 3

PARALLEL GLASNAST

(PARALLEL Glasgow Nuclear Algorithmic Shell model Technique)

Introduction: In this chapter we describe the complete implementation of a new technique, which we have developed, to carry out the shell model calculation by exploiting its inherent parallelism. We start by formally specifying the concurrently executable tasks of the calculation. Then we explain our implementation of these tasks as communicating sequential processes, firstly on one transputer and then on three and more transputers.

3.1 Parallelism in Shell model Calculations.

The Glasgow Shell model implementation[1] and DASS (ref. chapt. 2) have been based on three important concepts viz. the Slater Determinants, m-scheme and the Lanczos method. As a first step to its parallelization, it is very important that we identify explicitely the fundamental concurrency which exists in using m-scheme and the Lanczos method. We go through the following steps, which are required by a shell model calculation and then identify the concepts which can be executed in parallel.

(i) Generate an ordered set of N Slater Determinants in the occupancy representation ($\phi_1, \phi_2, ..\phi_i, .. \phi_N$).The index i in the basis table serves as an index to the rows and columns of the Hamiltonian and to the rows of the state vectors. This list of Sd's remains available and does not change throughout the calculation.

(ii) Find out all the pairs (ϕ_i, ϕ_j) for which the Hamming distance is <= 4. This is referred as Hamming criterion in further discussions. If a pair fail this condition, its corresponding two body matrix element (tbme) H_{ijkl} is automatically zero.

(iii) Use the evaluation rules and the uncoupled tbmes to compute H_{ij}, the (i, j)th Hamiltonian entry, for all the (i, j) pairs conforming to the above Hamming criterion. (In the following, for brevity, we use H_{ij} to denote any term in H that connects ϕ_i and ϕ_j. Where i and j have a Hamming distance of 4, H_{ij} is a unique uncoupled two body matrix element. When the Hamming distance is 2 or 0, H_{ij} will mean one of a number of possible two body matrix elements.)

(iv) For every $H_{ij} \neq 0$, multiply it by the jth element of the initial vector, and accumulate this product into the ith element of the product (new) vector. Each multiplication adds its contribution to the amplitude of the final vector. Computationally, the most demanding task in this whole process is the multiplication of H into the current vector and the location of the new state. Particularly, when the dimension of the matrix is very large, this would involve a heavy load of arithmetic computations.

(v) The Final step is to extract the eigen values and their corresponding other quantum numbers, like J, T etc., from the eigen vectors.

We identify from the above summary that in any shell model calculation, the following two tasks are computationally most expensive.

- Multiplication of the basis states with the Hamiltonian.
- Location of the new states which are created as a result of the operation of the Hamiltonian.

In fact both these operations are inherently independent of each other and hence can be computed concurrently.

Furthermore, the matrix multiplication, in itself, has been proved to be ideal target for parallelization (ref. section 1.1.2). Two arithmetic processors can proceed independently to compute contributions to a final vector provided they are provided with two distinct entries of the multiplier matrix. In the present case this translates into the possibility of breaking up the initial and final vectors into a number of subvectors each of which may be stored and dealt with on a separate processor. A notional computational arrangement is shown in fig 3.1 (a).

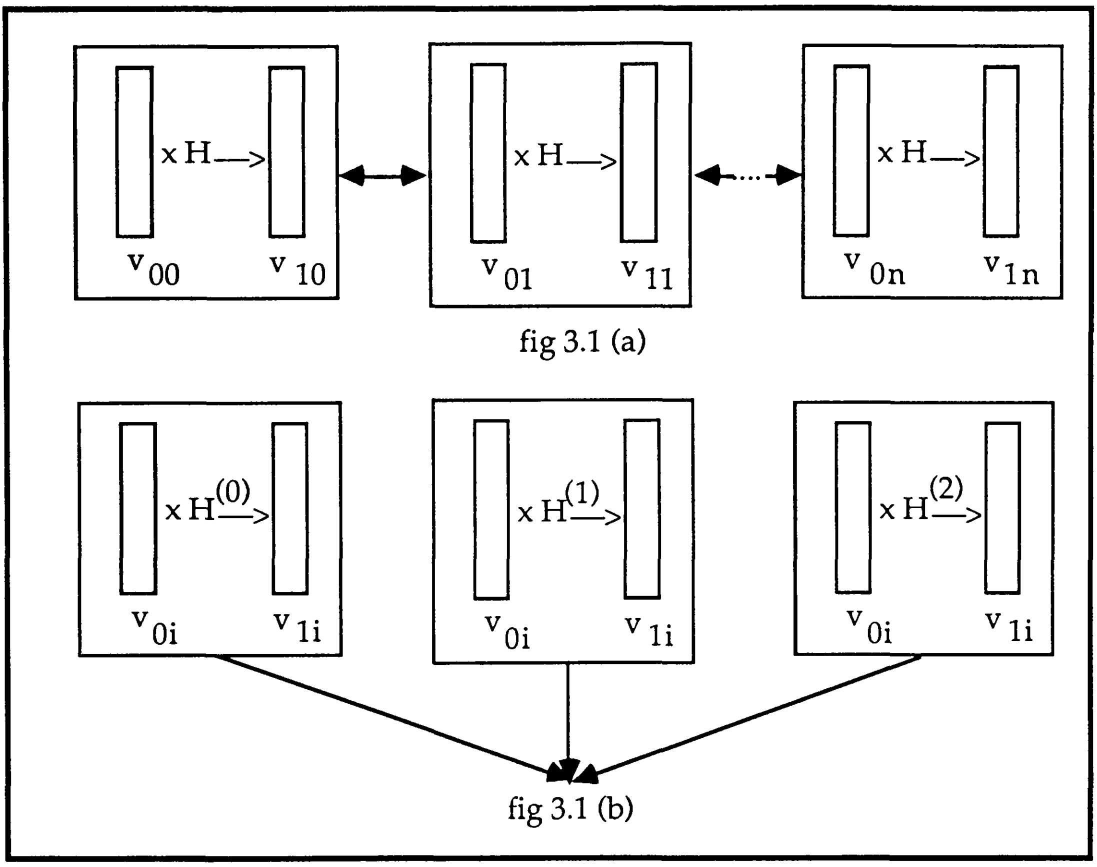


fig 3.1 (a)

fig 3.1 (b)

fig 3.1 Processing stages involved in the multiplication of vectors/subvectors

Each of n processing stages deals with a subvector or block, of the basis table and a current initial and final vectors. The arrows linking the blocks represent the 'off-block' terms, i.e. terms contributing to basis states in block j which arise by the operation of H on basis states in block i. Fig 3.1 (b) show how each processing stage may be further subdivided, with each subprocessor having access to only part of the Hamiltonian.

3.2 Parallelism in shell model: an implementation perspective.

Having identified explicitly the steps which can be executed in parallel for a shell model calculation, the next logical step is to devise an algorithm, which when translated to any realistic implementation, yields the advantages envisaged by the concurrent execution of the most expensive arithmetic modules of the calculation. Although there could be a number of possible ways to achieve this objective, but we have implemented it using the framework of algorithmic parallelism having course granularity (ref. section 1.2.2). Our basic algorithm comprise of the following steps. One way of implementing it could be as shown in fig 3.2.

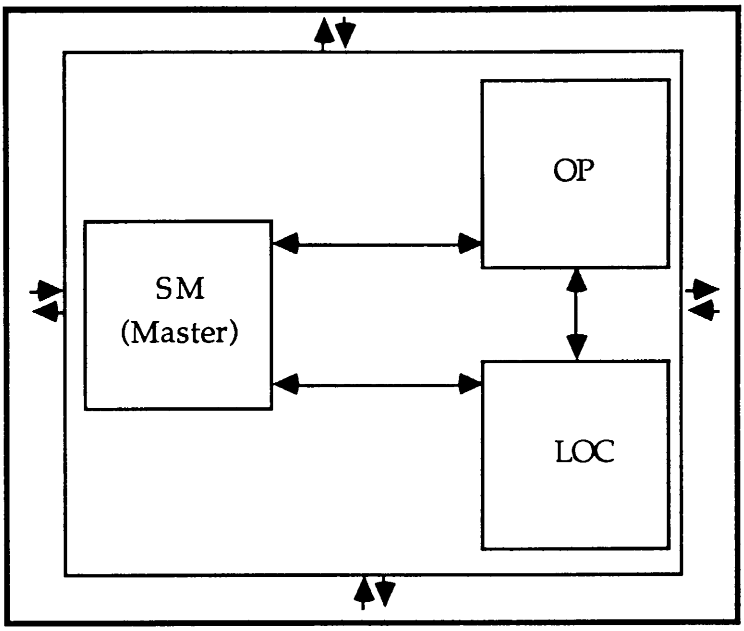


Fig 3.2 An possible topology of the three communicating tasks

- A master task (SM) supervises the entire calculation. It should exercise control over three things. First one , synchronization among the slave tasks. Secondly commanding the other processors to start/stop their respective operation. Thirdly, communication with the outside world (i.e. I/O).
- A second task (OP) is dedicated to the multiplication of states by the Hamiltonian, when ordered by the master.
- The third task (LOC) has the sole responsibility for locating newly produced SD's in the basis table, whenever asked by the OP. task.It should also update the respective amplitude of the located SD by the contribution sent by OP. At the end of a Lanczos iteration it should send the new vectors to SM for writing onto the disk.

We have successfully implemented this algorithm by creating three concurrent tasks, which execute as concurrent sequential processes as shown in fig 3.3. They are executed concurrently on a single transputer as well as on three. The same idea, in principle, can be extended , depending on the availability of more processors and the choice of the topology. In this implementation SM was made to generate the entire basis table and the uncoupled Hamiltonian table together with their respective indexes and to pass them to OP and LOC at the beginng of the calculation.

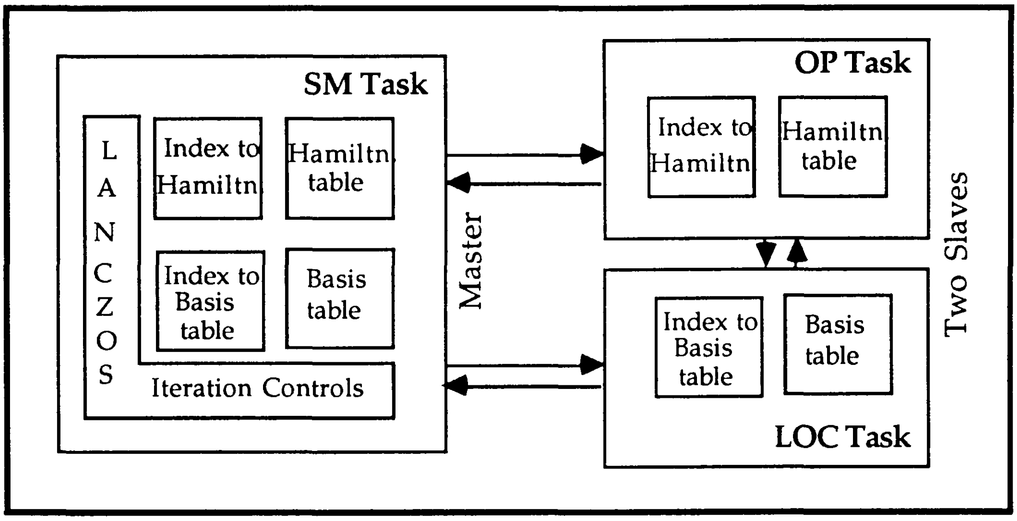


Fig. 3.3 Distribution of data/ operations on concurrent tasks

3.3 Implementation of Parallel Processes on One Transputer.

Before actually entering into the detail discussion of what and how each task accomplishes its job, it is important to explain two things. Firstly, how the operation of the Hamiltonian takes place and what messages are exchanged between concurrent processes. We have already given some explanation to the operation of the Hamiltonian in section 2.3. Here we would supplement it by how it contributes towards respective amplitudes of other state vectors. For doing this update it needs to locate the newly created SD, which is done by the LOC task.

Secondly, we explain the important feature how these processes exchange vital information about state with minimum of message traffic on the network.

3.3.1 The Hamiltonian :

We consider the Hamiltonian as a two body operator although this is not a fundamental restriction. A typical matrix element $< \phi_j | H | \phi_i >$ is zero if the Hamming distance between ϕ_i and ϕ_j is greater than four. This means that if the initial and final states differ by more than two particles (i.e. if the representing digital words have a Hamming distance of more than four), that matrix element will be zero. Because of this Hamming criterion the Hamiltonian matrix becomes irregularly sparse, a feature which can be exploited during its tridiagonalization by Lanczos method.

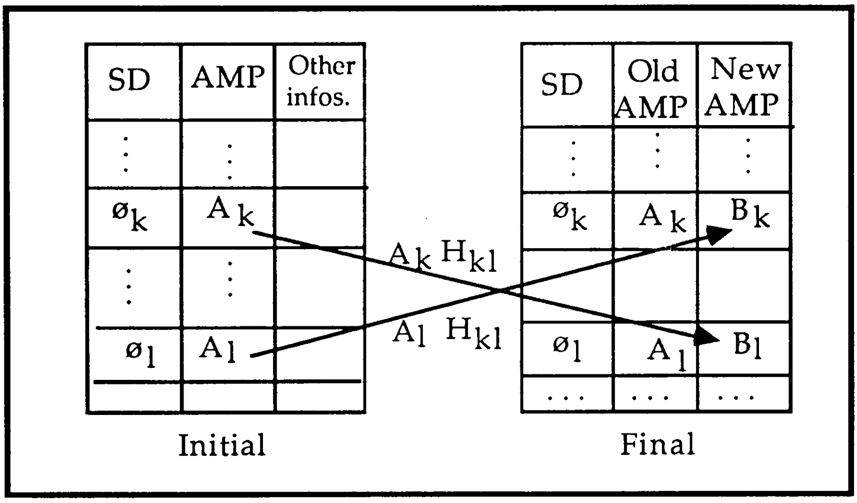


Fig 3.4 Forward & backward contributions from a state k to state l

When we multiply the Hamiltonian with state say ϕ_k, it does two things. Firstly, it creates a new state ϕ_l, which can be located in the ordered basis table. Secondly, it gives us a contribution of $a_k\ H_{kl}$ to the lth amplitude of the new vector, where H_{kl} is part of the two body matrix element connecting the initial and final states. This operation has been depicted in fig 3.4. In addition it is important to make use of the Hermitian symmetry of H. This implies that

$$\text{if } H \phi_k = A_k H_{kl} .\phi l \qquad \text{then} \quad H \phi_l = A_l H_{kl} \cdot \phi_k .$$

We refer to these as backward contributions. The backawrd contributions are treated as follows:

Once ϕ_l has been located and its new amplitude has been updated by $A_k H_{kl}$ its old amplitude A_l is extracted, multiplied by H_{kl} and this is added into the new amplitude for ϕ_k. The proper treatment of these backward contributions is vital for the success of the method. Anticipating a little, it will be apparent that the parts of the basis table containing ϕ_k and ϕ_l may reside on completely different processors and the backward contribution imply a two way flow of information between processors.

3.3.2 Information Exchange via Packets:

During the application of the Hamiltonian, once a new state has been created, it needs to be located in the basis table so that the appropriate contribution to its amplitude can be added. So, it becomes important that the locating task should be provided with enough necessary information which should assist in accomplishing this job efficiently.

We have designed a structure of type *packet*, which encapsulates all the necessary information for locating and updating the tables. The information content of such a packet is shown in fig. 3.5.

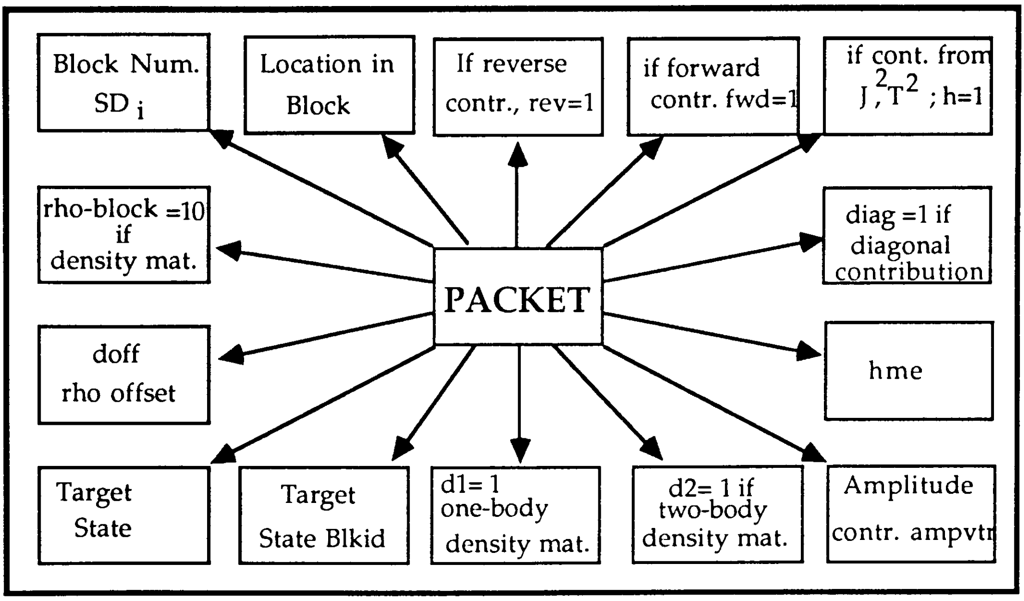


Fig 3.5 Information contents of a packet with some default values of fields.

The most important fields in the packet are the initial state, the target state, the Hamiltonian matrix element H_{kl} associated with this term and an amplitude. Flags in the packet indicate that it is (a) forward, in which case the amplitude field contains A_k H_{kl} and the target state, (b) diagonal, in which case since $\phi_l = \phi_k$ no locating is required, or (c) backward. In this last case the amplitude field contains nothing because ϕ_l has not yet been located in the table. Other fields in the packet are used in other parts of a calculation and need not concern us too much here.

The master task initiates the action in both OP and LOC tasks by starting the first Lanczos iteration. Previously it had been a problem to stop the iteration. But in the present case, task OP sends a packet witch contains a termination flag and is recognized by LOC and later communicated to SM about halting of the iteration.

In addition to exchange of information, between Con-current processes, through packets, similar information can also be generated by creating and executing parallel threads. The transputer hardware supports it and parallel 'C' provides the necessary facilities to create parallel threads. We have made use of this facility by allowing memory access to the concurrent threads through the use of semaphores [19, 31].

3.3.3 Treatment of Backward Contributions :

When OP sends a backward packet to LOC the target state must be located, its old amplitude A_l extracted and multiplied by H_{kl}. When this has been done the contribution $A_k H_{kl}$ has to be added to the new amplitude of the initial SD ϕ_k. But, remembering that ϕ_k and ϕ_l may reside on different processors, this cannot be done immediately. We put $A_l H_{kl}$ into the (previosly empty or meaningless) amplitude field of the packet and set the diagonal flag. In subsequent operations (about which more will be said later) the packet is treated as a diagonal contribution. If the initial state ϕ_k and the final state ϕ_l are on the same processor the backward contribution is used to update B_k, if not the packet is passed on to another LOC processor until it gets to the one dealing with ϕ_k.

3.4 The SM Task :

This is the master process, which has overall responsibility for starting, stopping, synchronizing and collection of results from the slave processes. It performs the following important tasks.

(i) It has the sole responsibility for communication to the outside world (I/O). It reads in data (or reference to it), for the required shell model calculation and outputs the final results.

(ii) It constructs the basis table in the form as described in fig 2.2.

(iii) It sets up the single particle table, pair table, and index tables to the blocks of Hamiltonian an basis table. These steps are sequential and are accomplished in very little time.

(iv) At this stage it distributes the work load to the OP and LOC tasks. They are first probed to ensure the global channels being operational.

(v) SM estimates the memory requirements for the index and blocks of Hamiltonian table and asks OP task to prepare memory blocks of these size. After it receives a 'success' message from OP, it dispatches the index table followed by the Hamiltonian blocks. At this stage, the important protocol of acknowledge/ request is rigorously observed at the transfer of every block of data.

(vi) The transfer of data by SM to LOC task, is carried out in the same way as for OP task except two differences. Firstly, the data transferred consists of index table for basis blocks and the basis table itself. Secondly, transfer of data to LOC task is done simultaneously with the transfer of data to OP task on a different parallel output link from SM.

(vii) So far the communication has been confined between the master and its slaves. Now SM orders OP and LOC simultaneously, to start the first iteration. At this stage the two slave processes enter into communication during the creation and location of states.

In exercising overall control, SM starts the first Lanczos iteration and stops when appropriate to do so. The state vectors are written to and read from the disk by SM and only two vectors are copied to LOC at any one time. SM also orders reorthognalization of vectors to LOC at the completion of iteration. The starting vector, in principle, could be arbitrarily chosen. But our choice had its first amplitude to be one, and the rest of them to be zero.

(viii) The first Lanczos iteration effectively start by ordering both OP and LOC tasks to spring into action. New states are created by OP, bundled into 'packets' and sent to LOC task, which locates it and carries out the necessary update in the respective amplitude of the basis vector. At this stage the traffic on the links between OP and LOC becomes very heavy. These channels could get clogged up resulting in poor performance, if the necessary steps are not taken. This risk can be minimised by choosing an efficient searching algorithm, and by choosing some optimum size for the basis blocks. We have investigated these steps and incorporated some changes which are reflected in our three transputer version of this implementation and the analysis of traffic flow versus blocksize has been reported in section 3.9.

(ix) At the end of each Lanczos iteration SM calls for the LOC task and writes them to the disk. It also obtains estimates of the total idle time each processor had to spent waiting while the other had not yet read the previous message on the channel.

(x) The halting of the slave processes is also done by SM, when the required number of iterations have been complete. Once the number of iterations requested (or the number of states in the calculation, whichever is less) is complete, OP and LOC task stop automatically because they do not receive command to start the next iteration.

(xi) SM calculates the eigen values, their respective angular momenta, iso spin and outputs them.

3.5 The OP Task :

This task starts its operation whenever ordered by SM. During its execution it performs the following functions.

(i) It takes a basis state and unpacks it as explained in section 2.3.1 and fig 2.6. It picks up each pair created in turn and produces new state.

(ii) During this process~~e~~ of unpacking it also creates parity representation of the states with practically very little extra cost. As explained earlier (ref. section 2.2.5), the phase of the new state is also calculated at this stage by using our new parity algorithm (ref. section 2.2.4).

At this stage, two essential ingredient for the determination of phase, SD in parity representation and pair masks are already available. It only remains a trivial step to read the status of the modified bits in the new parity word, which yields the phase. Because it does not involve any overhead like extra disk access, the cost of phase calculation, at this stage, is minimal.

(iii)It calculates the contribution to the amplitude i.e. ± a < j | H | i>, when the Hamiltonian changes a state |i> to state |j>. It wraps up this information about the initial and final state in a packet along with the amplitude contribution and sends it to the LOC task, which receives it and carries out the updates after locating the new state.

(iii) When all states in the basis table have been dealt with, it raises the flag by sending a termination packet, which indicates to LOC that no more states are to be located for this iteration. This is done, by sending an impossibly large value of the block identifier , in our case it is 999 when each block consists of 32768 states. Essentially lay out of this task and the OP task which is explained in the next section, has been shown in fig 3.3.

3.6 The LOC Task.

This task runs concurrently with OP and SM tasks. In fact in some sense it is subservient to both of them. The reason for this is that while SM exercises full control on its start/stop actions, the states are supplied by OP task for it to locate. We find that the effort involved in both the OP and LOC tasks is quite balanced and we have been able to achieve a maximum overlap of both these tasks. The main activity of this task comprises of the following.

(i) It receives a packet from OP task on its input channel with this task.

(ii) It locates the target state contained in the packet, in the basis table.

(iii) It adds the contribution to the respective amplitude of that and discards the packet. In fact it maintains two lists for both the old and the updated amplitudes.

(iv) At the end of each iteration it reorthognalises the new vectors and returns them back to SM for writing onto the disk.

The major activity of this process is searching states in the basis table. The efficiency of this task highly depends on the use of efficient searching algorithm, particularly when the size of the space is ≅ 10^6. We undertook an analysis of different available searching techniques, presented below, before mak~~e~~ing a preferred choice.

3.7 Comparison of different Searching Techniques:

Many searching techniques have been available for use in the LOC task. But some are efficient for smaller set of data while others have some different constraints. One of these techniques is binary search, which we have chosen as our searching algrithm We generate basis table which is already ordered from the start. This makes it more favourable for binary search provided its worst case performance is also comparatively good. We have given below the performance rating of some widely used searching algorithms which has been the basis of our choice.

We give here a brief introduction and performance comparison for a few other searching techniques which were available but we decided not to use them because of their unsuitability, to the present project, which is outlined in the discussion.

(i) Linear Search : In this method we start at one end of the list and compare our key with every member of the list in turn, until success or we exhaust the list. In the worst, when key is the last entry in the list (or absent), the number of comparisons required woud be N, where N is the number of elements in the list. This is simly a very ineficient searching algorithm particularly when the data is sorted, as in our case.

(i) Hash Searching : This is one of the very attractive search techniques. On the average it is considerably faster than even binary search. One needs to compare only one or two elements before hitting the key or declaring its non existence. But its serious drawback is the storing of the huge size of the data table or application of some transformation/ normalization to represent the whole table. This would require memory of the size of the numbers (SD's), which we know would run in millions in the basis table for large calculations. Even when some transformation is used to compress the table, which is usually done, the problem of multiple occupancy or collisions resolution props up. The amount of effort required in using dynamic hashing for collision resolution coupled with other implementation complexities reders it a less favoured a choice for our implementation.

(iii) Binary Search : This is the most widely used searching algorithm for ordered data. One reason for that is it is very efficient even for worst case performance and easy to implement.

It is based on maintaining three indices to the list each pointing to the lower, middle and the uper most entry of the list. We compare the key with the entry pointed to by middle. If it is greater then we keep the first half otherwise second half of the list and find the new middle. This divide and conquer strategy is applied recursively until the entry pointed to by middle matches the key (SD in this case). This algorithm has a guaranteed worst case performance of $\log_2 N$, in a list of N elements. The worst case will occur when the key is the first element in the list. The detailed performance analysis about this algorithm can be found in Knuth[37], Wirth[38] and Sedgewick[39].

3.8 Multitransputer Implementation of PARALLEL GLASNAST :

One underlying idea in this implementation is to demonstrate that OP and LOC tasks can be loaded on separate processor(s) and executed concurrently to achieve a maximum possible overlap. There can be number of possible topologies for this, but the one which we have used is shown in fig 3.6, which is a logical extension of the single transputer version fig 3.2. We have successfully run the program with this configuration, using three transputers and have achieved complete overlap of OP and LOC tasks by introducing certain optimizations, which have been explained in the next sections. The general working of this multitransputer version consists of the following steps.

(i) SM task produces the same data tables as it did in the single transputer version. It checks the number of OP and LOC transputers, say 'n', connected to their respective spines, which must be equal. If 'nb' is the number of basis blocks then it calculates the number of blocks nb/n, to be distributed on each LOC transputer, allowing adjustment for cases when nb and n are even/odd or one.

(i) At this stage SM works out dynamically the starting and ending addresses of each chunk of basis blocks which are to be distributed on each processor. After observing the normal communication protocol (explained earlier) it starts sending these blocks to their destinations.

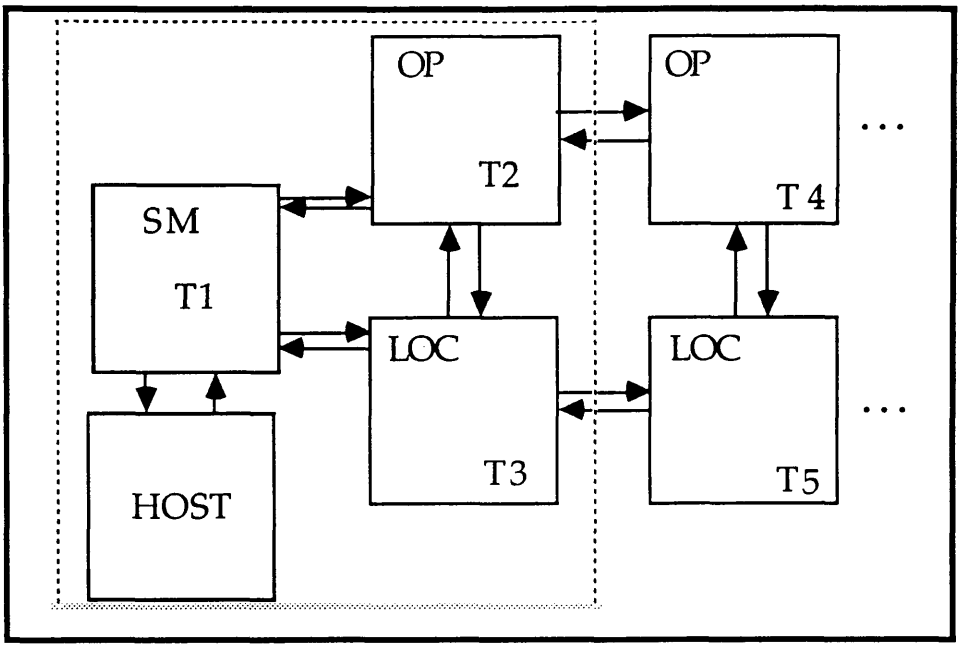


Fig 3.6 Implemented topology for multitransputer PARALLEL GLASNAST

Because of the limited number of communication channels available to each processor, these chunks are propagated from the first to the last LOC processor. The first chunk of states in the original basis table gets loaded on the last processor and the last chunk is seated in the processor, nearest to master, on the link of LOC processors.

(iii) A copy of the complete index to the basis table is however sent to all the processors.

(iv) All the respective OP tasks are propagated with the same set of tables, as in the case of one transputer explained earlier.

(v) SM task orders to start the first Lanczos iteration at this stage and the message reaches all processors on the network. OP starts producing packets by the operation of Hamiltonian, which are sent to the LOC task connected to the OP task. The LOC task looks at the index of blocks and checks if the block_id for the target state is present in the partition it has been alloted. Then if it is present the target state is passed on to the searching algorithm for locating it, otherwise the packet is directed to the neighbouring LOC task, which repeats the same thing. Each state is guaranteed to be located at one processor because the basis blocks on all the LOC processor form a complete set of the basis table.

(vi) At the end of each iteration SM orders the reorthognalization of new vectors and these vectors are collected by it for writing on to the disk.

(vii) When the desired number of iterations are complete, SM finds out the required eigen values and their respective other quantum numbers.

Optimization:

We have tested the multitransputer version using three transputers. As it works, the OP task always delivers a packet to LOC, while it is possible that LOC has not yet picked up even the one left previously on its channel. Converse could happen as well, when LOC has finished earlier and waits for the new packet, but OP has not yet been able to deliver one. This results in idle wait for one of the tasks at each node. As the number of packets exchanged can go in millions, this potential situation can result in poor performance and the loss of synchronization and overlap between the two tasks.

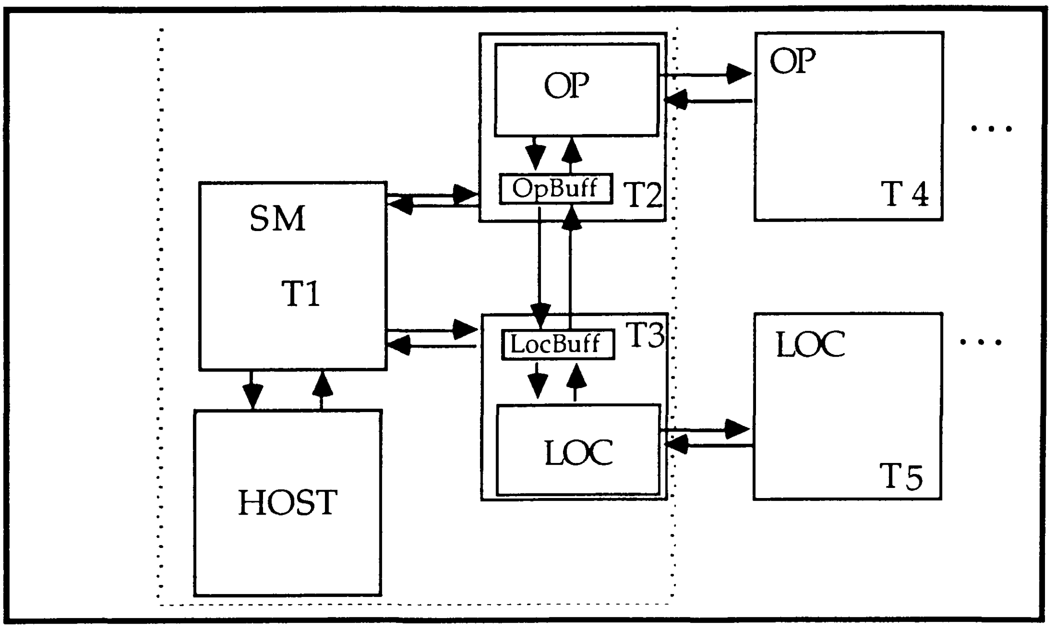


Fig 3.7 Buffered three transputer version of PARALLEL GLASNAST

In order to overcome this problem, we have introduced two buffer processes, one connected to each OP and LOC tasks, fig 3.7. These two processes run at high priority, which means that any packet put in a buffer has to be dealt with a priority higher than whatever the processor is doing at the moment. When OP produces a packet, it puts it in its buffer and gets busy with its operation on next state. The buffer running at high priority delivers the packet to the buffer attached to the LOC task, where it is picked up by the LOC task and the buffer becomes ready for receiving the next packet from the OP buffer.

In our implementation which uses buffers, both OP and LOC tasks keep a record of idle time they spent in waiting for the other task to deliver or receive the packet. With this arrangment we have been able to achieve zero op_wait and loc_wait time. This shows a complete overlap of OP and LOC tasks, which has been one of the main objectives of this implementation.

3.9 Analysis of Traffic variation on Channels with the Block Size.

We have undertaken a detailed experimental study as to how the traffic, i.e the number of packets on different channels varies with the number of basis states in each block (or on each transputer).

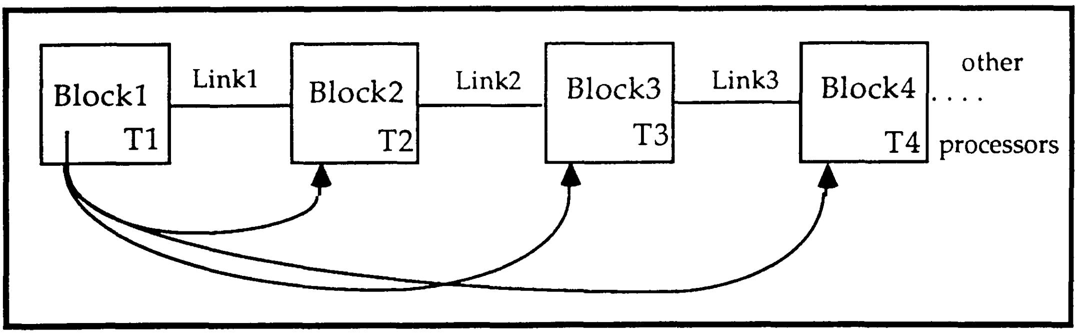


Fig 3.8 Contributions from one block to others being sent on different links

Let n be the number of states resident on each processor. The operation of the Hamiltonian on a state in block1 will create new states. These new states would be located within the same block or in any of the blocks on the subsequent processors. Similarly all states in block1 will send their contributions (in packets) to their parent blocks as well as to other blocks as shown in fig 3.7. Considering only the forward contributions, states in block2 and block3 will send similar packets to other processors down the link. The link2 will carry contributions destined from block2 to block3, .. etc., as well as contributions from block1 (and other processors before block2, if there are any) to these processors.

It is understandable that if block size is chosen very large, most of the target states will be located with the parent blocks. This will result in less traffic on the links. But at the same time, search space within the block will increase accordingly and would result in longer search time and, indeed the operation of H would take longer. This would defeat the whole purpose of distributing blocks on different processors. On the other hand, if the block size is made smaller, it would decrease search time but traffic on the outer channels will increase, which may clog the network.

We present here, in the next few sections, results of our experiments, which can help in choosing the optimum block size for certain number of transputers on the network.

3.9.1 Estimates of Traffic on Links:

The ideal would be to have some means of deciding in advance for each calculation the optimal way to distribute it over the available processors. We have not yet found a completely satisfactory way to do this but we have set up the machinery for studying the problem. We have written a version of the new program specifically for the gathering of information relevant to the issue. The most important consideration is the amount of time that each LOC processor spends passing packets up and down the chain compared to the time spent in actually processing packets referring to its block of the basis table. The basic information requried is the way in which the new SD's produced by H acting on a given basis state are distributed through the basis table. One measure of this is obtained as follows :

The Hamming distance between each pair of states, i and j, in the basis table is found and if it is less than 4 (showing that a two body operator can connect the two states) a count is added to the bin containing | i - j |. The result is a histogram like fig 3.9.

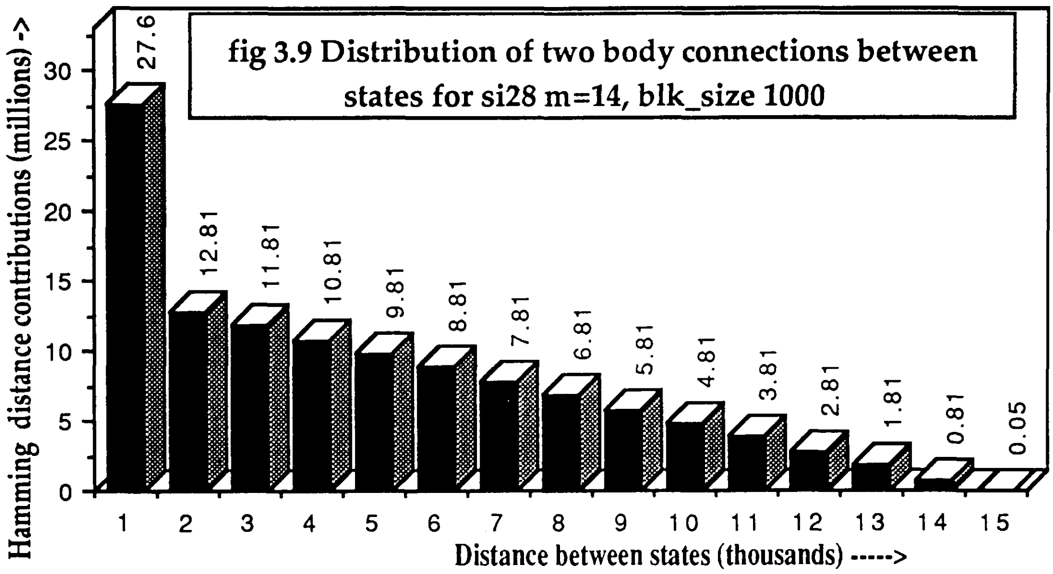


fig 3.9 Distribution of two body connections between states for si28 m=14, blk_size 1000

In this figure the bins are a thousand states wide. The high peak in the final bin shows that a large fraction of the possible two body connections occur over a range of less than a thousand states, but the total number outside this range is much larger. This figure gives, however, a misleading impression. We recall that pairs of states with a Hamming distance of 4 are connected by only one two body matrix element, while those with a Hamming distance of 0 (i.e. initial and final states the same) are connected by n(n-1)/2 two body matrix elements, where n being the number of particles. Since states far apart in the basis table are more likely to have large Hamming distances than small ones (since the table is ordered) the total computational effort spent on nearby connections should be much greater than fig 3.9 indicates. This is confirmed by fig

3.10 which shows counts of terms produced by OP in a Shell model calculation, again displayed as a histogram. Each contribution corresponds to a packet produced by OP and processed by LOC. The distribution is now very sharply peaked in the first bin, which corresponds once more to connections between states within a thousand of each other in the basis table.

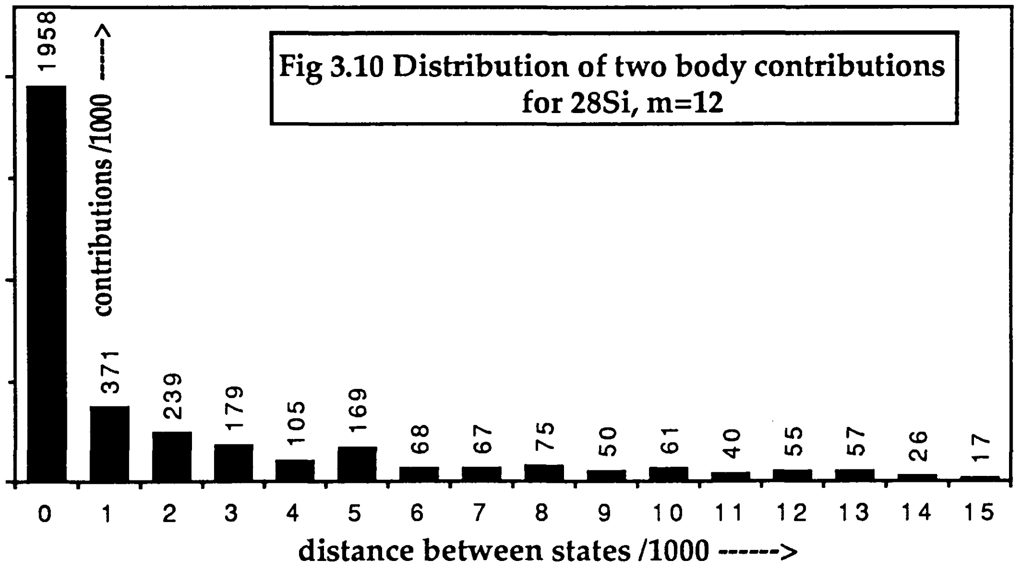


Fig 3.10 Distribution of two body contributions for 28Si, m=12

Consider a long chain of processor pairs. Each OP processor produces (on average) n packets, where

$$n= n_0 + n_1 + n_2 + \ldots..$$

n_i being the number of packets which belong to a processor i jumps away. The total number of packets destined for a given processor, say the middle one, is then

n_0	from its own OP
$+n_1$	from its nearest neighbour
$+n_2$	from its next nearest neighbour
.	
.	
$= n.$	

The total number of packets that have to pass through the middle processor en route to others is

$$
\begin{aligned}
&n1 + n2 + n3 + \ldots. && \text{from its own OP}\\
&\quad +n2 + n3 + n4 + \ldots. && \text{from nearest neighbour}\\
&\qquad +n3 + n4 + \ \ldots && \text{from next neares neighbour}\\
&\qquad \ldots\ldots \\
&= n1 + 2n2 + 3\,n3 + 4n4 + \ldots. = n\,\Delta
\end{aligned}
$$

where $\Delta = \frac{\sum_{i=1} i\,n_i}{n}$ is the average jump per packet (measured in blocks). The total calculation time, assuming that each processor pair actually does the same amount of work (which amounts to neglecting end effects on the chain) and that they are perfectly overlapped is then given by

$$
\begin{aligned}
\text{Total time} &= \text{processing time} + \text{routing time}\\
&= nT_p + n\,\Delta\,T_r
\end{aligned}
$$

where T_p and T_r are the times required to process a packet and to route it

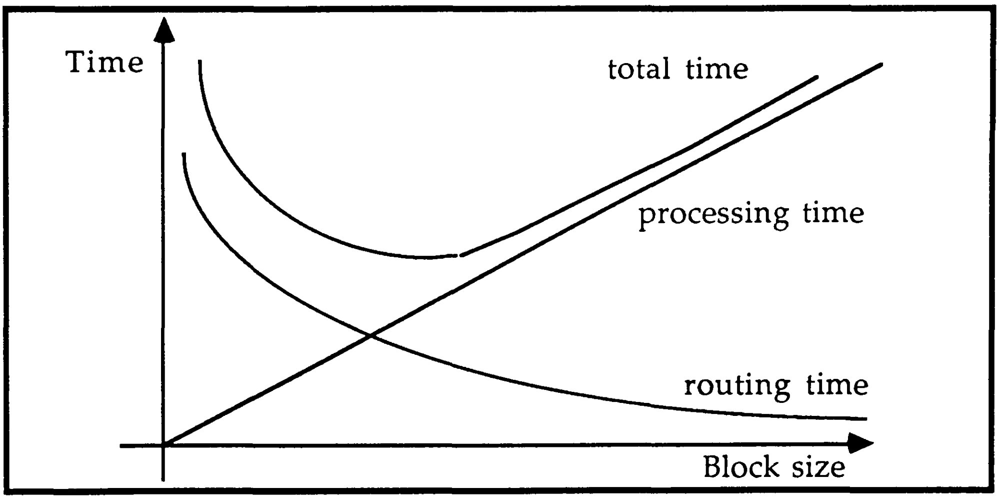


Fig 3.11 Components of the total calculation time as a function of the block size (schemetic)

along the chain respectively. The first term is proportional to the block size (since n is proportional to the blocksize). Obviously the more processors that can work on the calculation the faster it will go. In the second term we might expect the average number of jumps to be proportional to the blocksize but, in fact, it falls off more steeply than 1/n, so this term becomes large for small n (or small blocksize) and vanishes for large n.

Table 3.1 shows that the coefficient of T_r varies with the blocksize using the data from the fig. 3.10.

Blocksize	Δ	Δ x Blocksize
1000	2,39	2390
2000	1.033	2066
4000	0.43	1720
8000	0.14	1120

Table 3.1 Variation of jump size with the block size

This discussion is intended only to illustrate the consideration that enter into the choice of block size and processor topology. In actual calculations the end effects explicitly ignored here are important and will destroy the complete overlap that has been assumed. It will be necessay in further developments to have the possibility of putting different number of states on different processors to restore this overlap. Evaluation of T_p and T_r, the single packet processing and routing times, is also not completely straight forward. Our benchmark for Tp relates to calculations for ^{20}Ne m= 0 with 640 states and with OP and LOC pair on separate transputers with full buffering. For this case T_p is 1.2 ticks per packet. To get an estimate for Tp in other`cases multiply this number by ($\log_2$ (Block size)) / ($\log_2$ (640)). This allows for major source of variability in packet processing times - the logrithmic dependence of the binary search time. On the other hand T_r is much less certain. It depends very much on the way the program is written and on the details of buffering and at present we have no reliable method of measuring it. This is one of the major tasks remaining in this work.

3.9.2 Performance Indicators for the three Transputer implementation :

In this case only one OP-LOC pair is connected to the SM task. These calculations have been done for ^{28} Si using various values of m. Table 3.2 shows the results obtained for the individual cases. For each m value timings have been recorded for the first three iterations. We make the following conclusions from the data presented in this table.

The total time for the first iteration involves no disk access and so may be taken as a measure of the computational work involved. Note that the measured computation times in the third column are substantially constant with each set and differ only slightly from the total time for the first iteration given in column 5. Total times for the 2nd and 3rd iteration increese because of the extra load entailed in the Lanczos re-orthogonalisation and the associated disk accesses. The important thing to note is that the extra overheads represent only a few percent of the computation [illegible] and that this fraction decreases as the calculation gets bigger. As has been mentioned in the present arrangment disk acceses are routed through the master task SM. Transputer compatible hardware exists to allow distributed disk systems to be built and such a system in which each processor pair had access to its own disk would reduce the overheads considerably. For the time being, however, we regard the disk overheads as acceptable.

No. of	M	Real Itrn.	Real Itrn.	Total Itrn	Timing	Number of
		Time in	Time in	Time in	Overhead	Packets
states	value	ticks	in sec.	inticks	D'Acc.etc.	Exchanged
1439	20	327,811	20.98	331,575	1.1 %	231,032
		327,810		340,128	3.75 %	"
		327804		343201	4.69 %	
14,310	14	4,322,420	176.63	4,353,953	0.73 %	2,919,163
		4,322,267	276.62	4,431,222	2.5 %	"
		4,322,856		4,467,386	3.3 %	"
37,086	10	12,888,820	824.88	12,973,968	0.66 %	8,307,065
		12,890,094		13,173,663	2.19 %	"
		12,891,332		13,252,255	2.79 %	"
67,560	6	25,724,541	1646.37	25,878,742	0.59 %	16,018,082
		25,728,591		26,242,020	1.99 %	"
		25,729,756		26,390,981	2.57 %	"
81,122	4	31,800,088	2035.2	31,985,745	0.58 %	19,571,098
		31,805,455		32,428,049	1.95 %	"
		31,806,469		32,591,738	2.46 %	"
93710	0	37,868,728	2423.6	38,084,045	0.56 %	22,913,412
		37,875,423		38,599,073	1.91 %	"
		37,876,202		38,791,957	2.41%	"

Table 3.2 Sample Iteration timings and overhead results for ^{28}Si. Timing are given in transputer low-priority ticks (64 μ secs/tick) & in seconds.

The total time for an iteration on ^{28}Si m=0 is about 40 minutes, only an order of magnitude greater than the old Fortran/Machine code program running on an IBM 360/195. We regard this as very encouraging. Ten processor pairs could bring us in line with the main frame state of the art, even without further development.

3.10 Conclusions and Future Research:

In part one we have demonstrated, through our implementation of PARALLEL GLASNAST, that the shell model calculations can be split into three tasks, which can run concurrently, on one or more transputers. We have achieved this concurrency by running by running the program on one processor as well as on three processors. With the buffering arrangement, we have achieved a complete overlap of the OP and LOC tasks reflecting the proportional distribution of work load and the natural splitting of the project into concurrent components. Execution times have been obtained (for ^{28}Si m=0) for one and three processor versions which are fairly good. A detailed analysis of the traffic flow on different channels has also been undertaken to estimate a reasonable size of the basis block for loading on to respective processors.

We have used one specific topology in this implementation. But the individual tasks have got enough built in flexibility that they can be distributed on any number of processors using any other more efficient topology. Some of theses are suggested below.

(i) The topology given in fig 3.7 and explained in section 3.8 can be used for five or more transputers. But the main problem to be encountered in this case will be the increase in traffic on the links to the master task (SM), because the read/write of vectors (at present) is done by SM. This reflects that the increase in the number of processors along the OP and LOC spines could be beneficial only if this traffic could be cut down. One way of achieving this effect can be to provide each processor with its own enough memory to do its own I/O of vectors. Possibly a PC or a disk along with its memory manager could be attached to each of the processors. This arrangement can reasonably sort out the bottle neck of the disk I/O of vectors.

(ii) The second possible extension, that can be made, is to provide a dedicated communication processor for every OP and LOC pair we use, as shown in fig 3.12 below.

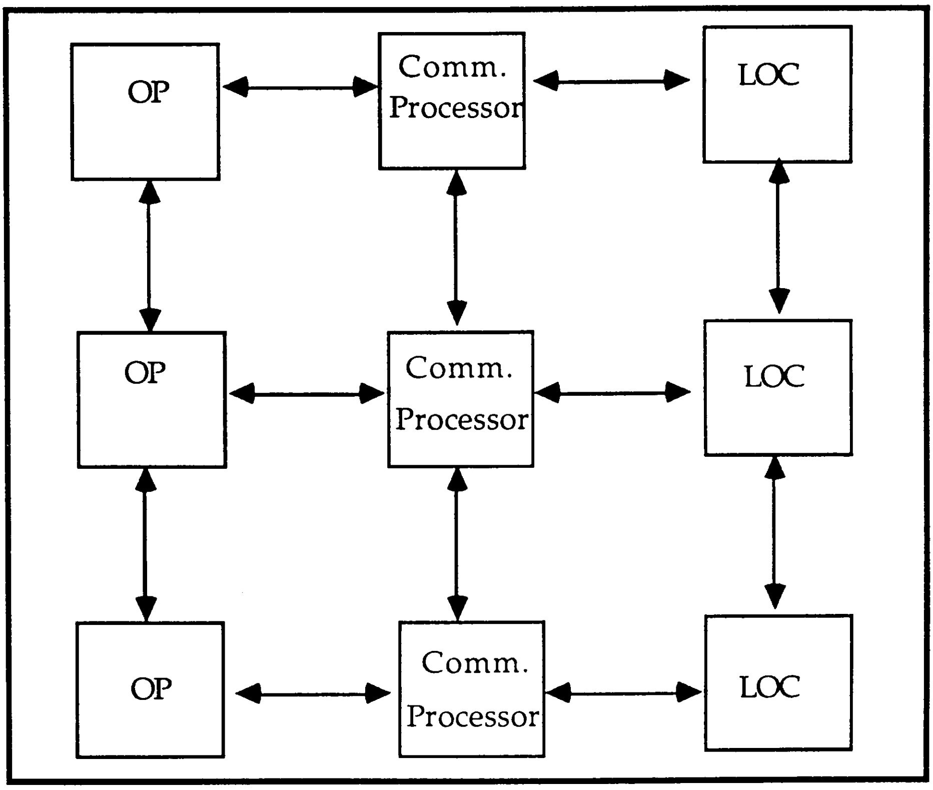


Fig 3..12 : An Alternative Topology using dedicated communication Processor

This communication processor would completely relieve OP and LOC of any communication over heads and buffering will not remain necessary. These main tasks will be left with their main job of multiplication and location only. On the other hand, because the communication processors will not have any other duties to perform except handling the traffic of packets and vectors, they will perform this job very efficiently. If the financial constraints are not very hard then the provision of communication processors coupled with own disk I/O facility for the OP and LOC tasks would certainly be the most efficient arrangement in the present circumstances.

PART TWO

Introduction : Evolution of knowledge has always been a continuous phenomenon, which has been forming a chain of intellectual advances over the years and centuries. Such is the case with the nuclear shell model; which has achieved so much, in understanding the mysteries of nuclear structure. In order to build on this rich heritage, rigorous studies into the nuclear shell model have been initiated in the past, are continuing and shall continue, perhaps for some of the following reasons:

Ø To predict observed energy levels.

Ø To understand the nucleon-nucleon interactions of the nucleus, both within complex nuclei and those of free nucleons.

Ø To help understand and identify the collective nuclear phenomena, for which shell model has provided simple microscopic explanation.

Ø To construct a detailed nuclear wave function for more reliable and realistic calculations.

Ø To complement the above developments by using the latest computational techniques.

Motivation:

Many researchers like Glaudemans, Wildenthal and others have worked to produce a better effective interaction for the two body matrix elements and single particle energies for the Hamiltonian. The sd-shell has been very extensively studied, over the years, both theoretically and experimentally. Many nuclear calculations have also been performed using these interactions. But, generally speaking, they do not consider the following important factors.

1- They do not take into account the effect of Coulomb's force, which is the one big difference between mirror nuclei.

2- No calculations seems to have considered the information on the occupancy of nucleon in different shells and subshells.

We have incorporated this information in our studies and it seems to give us a more detailed and systematic picture of the nucleus. In order to prove the usefulness of these addition we have chosen mass 21 nuclei in our studies for the following reasons.

①- This includes lots of mirrors, which is ideal for the comparison of different nuclear properties, with and without the effect of Coulomb's force.

②- Being an odd nucleus, single particle properties can be easily studied.

③- All the nuclei have got five nucleons in the sd shell. This should illustrate all the complexities of an sd shell nucleus.

④- On the other hand, calculations for five valence particles are not very difficult.

⑤- It is quite well suited for studying the band structure and other collective properties.

⑥- Quite a lot of experimental data is available for the mass 21 system.

⑦- Very timely and recent publication of a detailed study into the structure of ^{21}Ne-^{21}Na mirror nuclei by Hoffmann et. al.[121] has not only been quite useful in the comparison of our results, but also it underlines the growing need for more elaborate studies into mass 21 as a whole.

Present Work : We have made extensive calculations for the six isobars of mass 21 viz. ^{21}O, ^{21}F, ^{21}Ne, ^{21}Na, ^{21}Al, ^{21}Mg. Calculations made in this respect include the following:

1-Spectrum of the excitation energy for all the positive parity states.

2- Occupancies of nucleons in each shell corresponding to different eigen states.

3- Electric transition rates B(E2).

4- Magnetic transition rates B(M1).

All calculations have been made using three different interactions viz. PW, USD and CWC. The first two do not include the effect of Coulomb's force, while the last one does. Nuclear properties of the individual and mirror nuclei have been explored by making comparisons with the experimental data and between different interactions, using appropriate tables and energy level diagrams.

CHAPTER 4

Nuclear Shell Model: A Perspective

The modern nuclear shell model has been the outcome of decades of successive evolution. Like other great theories, in the beginning it had to undergo years of severe criticism before finally becoming a cornerstone in the field of nuclear exploration. In this chapter we describe the early difficult days of this theory followed by its phase of gradual acceptance. Finally we shall discuss the issues that it has raised and the suggested solutions before discussing our present work on the sd shell calculation for the mass 21 .

4.1 Shell Model : Past perspective

Historically speaking, the experimental data on the nuclear properties like binding energies, spin/isospin, parities etc., have usually played a decisive role in determining the the success or failure of nuclear structure models. It has been due to the detailed study of these quantities that led to the successful launch of the nuclear shell model, by Mayer[42, 43] , Jenson and Suess[44].

The history of the shell model begins even earlier in 1932, the year when Heisenberg established the picture of the nucleus as being composed of neutrons and protons, and Barlett[45] introduced proton and neutron shells in light nuclei (analogous to the electronic shells in atoms).

In 1933, Elsasser[47] carried out a more extensive study of the nuclear shell model. On the basis of his experimental results he presented evidence for the magic numbers 50, 82 and 126. He also suggested single nucleon level schemes. This very idea that there are single particle states in nuclei brought to light highly simplifying features of an otherwise very complex many-body problem.

In 1936, an authoritative but highly critical review of the shell model was made by Bethe and Bacher[48]. In this review they included a comparison of the predicted and the experimental results, which were available at that time. They emphasised the need for the two-body interaction in the given shell model configuration and the possible need for configuration mixing. They criticised the level schemes put forward by Elsasser and Barlett, although they "would agree with the experimental results, but they lack theoretical foundation".

Another serious criticism against the shell model came from Neils Bohr in 1936. In his article[49] he based his objection on the short mean free path of nucleons in nuclei. This extract from his paper underlines his manifestation.

"the nuclear models hitherto treated in detail are unsuited to account for the typical properties of nuclei for which, as we have seen, energy exchanges between the individual particles is a decisive factor. . . . . In the atom and in the nucleus we have indeed to do with two extreme cases of mechanical many-body problems for which a procedure of approximation resting on a combination of one-body problems, so effective in the former case, loses any validity in the latter."

Later research has shown that Bohr's criticism was not quite well founded. Use of the Pauli's principle, however, explains that the many body nuclear phenomena can be approximated to a combination of one body problems, as has been true in the atomic case. In an atom electrons revolve under the influence of the nuclear charge, while the nucleon revolve under the influence of an average potential due to the rest of the particles. In a nucleus we can have two igentical particles in the 0s shell, six in the 0p shell and so on. The two particles in the 0s shell exert a strong short range scattering force on each other. But neither of the two particles can jump to the 0p shell, because there is no vacancy Also no particle from an outer shell can jump into the quantum state of either of the particles in 0s shell. This is because, according to the Pauli principle, no two particles can be represented by the same set of quantum numbers. This shows that the interaction between particles in different shells is very well defined. However in the very outer shells where the density of particles is small but the number of available single particle orbits is large, configuration mixing takes place. Not much is yet known about this area of the nucleus. But certainly, for the lower shells, many-body problem can be reduced to a combination of single body problems.

As time passed, the criticism to the nuclear shell model grew from strength to strength. This discouraged many theoretical physicists from using it. Even scientists like Racah preferred to apply the shell model to atomic spectroscopy after being convinced that it was not valid for nuclei. The only few determined ones like, Wigner, Feenberg, Hund, Jahn and some others continued their work on the shell model. However, their work mostly focussed on light nuclei with occasional reference to the nuclei beyond oxygen.

4.2 Renaissance of Nuclear Shell Model.

The year 1948 marked the turning point in the history of shell model. It began with Mayer presenting strong experimental evidence for the reality of magic numbers. This was followed by Nordheim[51] and Feenberg and Hammack[52], who published level schemes consistent with the magic numbers. Although level schemes were based on L-S coupling like Elsasser, but were different from it.

Inspired by a question from Fermi, Mayer realised the decisive role played by spin-orbit interaction. This was quite instrumental in the publication of the same level scheme, based on a strong spin-orbit interaction, by Haxel, Jensen and Suess[44], the same year in 1949.

The paper presented by Mayer[42, 43] was a very detailed and showed how many phenomena, were successfully explained by jj-coupling shell model. She discovered coupling rules for the neutrons and protons in the even-odd nuclei and carried out calculations on simple j^n configurations using a short range attractive potential. Any deviations from the J= j rule were said to be due to the non zero range of the potential.

These and many other publications, that followed, herald a new era of acceptance for the shell model as a reliable means for the study of nuclear properties.

From then on, the mantle of shell model was carried on to greater heights by many researchers like Elliot, Flowers, Glaudmans, French, Halbert, Whitehead, Watt, Brown, Wildenthal, Arima, McGrory, Woods, Warburton, Catford, Wright, Hees, Fifield and many others. Their research brought us a wealth of information, both theoretical and experimental., about the structure of the nucleus. This ranged from the simple shell model calculations about light nuclei to the complex issues like, finding a comprehensive wave function depicting the near exact state of the nucleus. Specialist techniques for diagonalizing Hamiltonian matrices ($\approx 10^6 x 10^6$ or more) well beyond the sd shell were developed. New computational methods, electric and magnetic multipole moments, deformed nuclei and other collective properties were among the new avenues which were explored in nuclear structure.

This all put together made the nuclear shell model a very important paradigm of nuclear physics. We explore the gradual development of these issues in the next few sections

4.3 Shell Model Hamiltonian

The shell model of Jensen and Mayer was based on the following assumptions.

- The single configuration for each nuclear level, i.e. one single orbit for a single nucleon.
- The inner core of nuclei, i.e. ^{16}O or ^{40}Ca (depending on the mass number) formed inert core. This attributed effectively the nuclear interactions to the valance particles.

Based on these assumptions, the shell model achieved an amazing degree of success in explaining the nuclear data. But further work in an attempt to determine a more realistic wave function for the nucleus opened up new areas of research. Some of these issues (e.g. many-body theory) developed into independent new disciplines of nuclear physics. We have given below their brief explanation before attempting to report their present phase of development.

4.3.1 The first wave function of independent nucleons did not take account of the strong short range interaction between the nucleons. The effect of two nucleon interactions had been studied in detail even before the early days of shell model. Several regular exchange interactions were extracted from deutron data and free nucleon scattering experiments. The main objective of these studies was to get more information on the two nucleon system. But the experimental evidence about the strong repulsive force between nucleons at short distances, effectively put an end to these efforts, because the new singular interaction could not be used to calculate matrix elements between states of independent nucleons. Also it indicated that regular interactions, which could be used in the shell model could be obtained after going through a procedure of renormalisation. This approach developed into a new field of nuclear research called many-body theory. Some details on this area can be found in ref. [54, 55].

4.4 Effective Interaction.

Earlier calculations into the energies of the nuclear states were based on a simple shell model wave function. Two body matrix elements (tbme) for the effective interaction were extracted from experimental data. A finite set of matrix elements necessary for the effective interaction can be determined by least square fitting some of the experimental data being studied.In principle, at least, it is possible to construct some procedures to transform from the real interaction to an effective interaction, which when diagonalized in the model space would give exact energies. In a recent work Ji and Wildenthal[57] have used a central Yukawa interaction as an initial interaction in the fitting process. Also another recent attempt has been made by Mother et.al.[56], which has been partially successful.

But the ideas like mass dependence of matrix elements, the role of effective charge on the nucleons and indeed the idea of extending the model space and its implications were not known in the early days of shell model. We try to explain some of the successful attempts made to formulate two body matrix elements for the effective interaction, in line with these ideas.

4.4.1 Amit-Katz Interaction[101].

This could be considered to be one of the first few successful efforts in which they calculated five two body matrix elements for the effective interaction. This interaction reproduced total(binding) energies of all the low lying energy levels for $8 \leq A \leq 16$. These calculations were confined only to 1p shell.and did not make any assumptions about the central field or the two body interaction. Thus the single particle energies (spe), and the interaction in various two particle states were determined only by comparison with experimental results. They used the jj-coupling scheme to construct the n-particle energy matrices. Only P^n configurations were taken into account and all parameters remained fixed throughout the shell.

In many cases the agreement between the calculated and experimental result was an improvement on an earlier work by Kurath[100]. Kurath used LS-coupling for calculating the interaction. They used matrices for 1 to 12 nucleons in the 1p shell as energy matrices, for the 1 to 12 holes in the closed p-shell. The difference in the single particle energy was considered, and so was the principle of Racah[102], that the interaction between two holes in a certain state is the same as the interaction between two particles in the same state.Energy of the holes was obtained by subtracting the binding energy of the ground state of ^{16}O from all the energy levels. For comparison with experiment, Kurath assumed the potential well to be a harmonic oscillator and the interaction between the particles to be central.

4.4.2 Cohen-Kurath Interaction[100].

The 1p shell remained a good testing ground for nuclear models since the beginning of studies in the nuclear structure. Probably, it was due to the simplicity of its configuration and the availability of a huge amount of data for it at that time.

This effective interaction, which consisted of 15 matrix elements for the two body interaction and two single particle energies for the 1p shell, was obtained by fitting energy data about nuclear levels. We briefly compare below the Cohen-Kurath approach to the earlier ones.

1 Earlier attempts used Elliot[103] approach of choosing a particular two body interaction and varying the difference between the single particle energies of 1p3/2 and 1p1/2.

2 Amit and Katz[101] calculated binding energies w.r.t. ^{16}O, while in this case it was done w.r.t. ^4He. This required adding a constant amount 0.13 MeV to all the diagonal matrix elements of their two body interaction and subtracting 1.06 MeV from the single particle energies.

3 They adopted another procedure[104], from spectroscopy and considered two body matrix elements as parameters to the fit. They carried out calculations using 17 parameters (15 tbme and 2 spe) and also with 13 parameters while using potential in LS-coupling.

4.4.3 Kuo Interaction[58].

This was based on the principle that the nucleon-nucleon phase shift is one of the important factors which influences the effective interaction. The Kuo interaction has been used later on by other interactions. The two body matrix elements which could not be fitted against the experimental data were adopted from this interaction.

4.4.4 Preedom-Wildenthal Interaction[59].

This was derived from the Kuo interaction by considering those elements not involving d3/2 particles as free parameters, in a fitting procedure for nuclei $18 \leq A \leq 22$. This only meant that matrix elements involving d3/2 particles were probably not as accurately determined as others. Whitehead, Cole and Watt have made a comparative study of these two interactions. They made calculations for the energy spectra, band shift etc. for many nuclei and their findings have been reported in a series of their publications given in ref.[60]. The two body matrix elements for this interaction are given in appendix A.

4.4.5 Chung Wildenthal Interaction[61].

This was an attempt to improve upon Wildenthal's previous interaction[59]. This consisted of two sets of matrix elements, one for mass region $17 \leq A \leq 28$ and the other for $28 \leq A \leq 39$.

This important improvement in this interaction was that all the matrix elements and single particle energies were considered as parameters in a least square fit to experimental binding energies and excitation energies. However, in this interaction as well, two body matrix elements were assumed to be mass independent.

4.4.6 Kelvin or CWC interaction.

None of the interactions, which we have considered so far, took into account the effect of Coulomb's force between the protons. This implied that the calculations of mirror nuclei, e.g. ^{21}O and ^{21}Al would produce exactly the same energy spectra for both the nuclei if we use e.g. CW interaction explained above. A concerted effort was made by the Glasgow Group to include this effect into the interaction. They fitted the experimental data together with the effect of Coulomb's force to calculate the two body matrix elements for the effective interaction. The matrix elements for this interaction are given in appendix B.

We have used this interaction in our sd shell calculations for mass 21. The net effect of Coulomb's interaction can be clearly seen in the diagrams and spectra presented in chapter 5.

4.4.7 USD Interaction[62].

All effective interactions have been based on the principle that each nucleon moves in its own single particle orbit in a given configuration, under the influence of a potential which is an average of the potential due to rest of the particles. Wildenthal proposed that since the effective potential depends upon the mass number of the nucleus, the tbme should also have this mass dependence. He suggested a scaling factor of $(18/A)^{1/3}$ to the matrix elements. This new set of tbme based upon mass dependence is given in appendix C.

In this interaction the tbme and single particle energies were determined by making a least square fit of the shell model eigen values to the experimental energy levels. In this process tbme for the sd shell were fixed with all parameters of the Hamiltonian, which was certainly a remarkable achievement. This unified interaction for all the sd shell nuclei was obtained in a fit to 447 binding and excitation energies of the sd shell states. The working parameters of this fit were actually 47 best determined linear combinations of tbme. The 19 least well determined linear combinations were fixed to correspond to values of Kuo's matrix elements[58].

Results of further studies by Brown and Wldenthal[63, 64] into ^{27}Al, ^{28}Si, ^{29}Si, ^{23}Mg, reveal that over the mass region $16 \leq A \leq 40$, one should expect a mass dependence of $A^{1/2}$ for the tbme while using delta interaction and harmonic oscillator wave function with $\hbar\omega = 41A^{-1/3}$. A comprehensive examination of the experimental and predicted data using this interaction has been carried out by Brown[53] and Brown and Wildenthal[64]. A very close agreement between both the experimental and the predicted results has been found in the region $21 \leq A \leq 35$.

Studies beyond this range have been carried out by Lickert et. al.[67] and the results so calculated have been compared with the experimental results obtained by Ropke et. al.[68, 69] and Tikkanen et. al.[70]. In these calculations Wildenthal has argued that in this region sd shell levels are fewer and the role of the intruder configurations from the adjacent major shell is more important. In this comparison a detailed agreement with experimental energies and spins for positive parity levels through 6 or 7 MeV for nuclei $23 \leq A \leq 33$ has been obtained when the intruder states are suppressed.

4.4.8 Future Effective Interactions.

In spite of the successes reported so far, the calculation of a comprehensive and unified effective interaction is still far away. Our knowledge from the many-body theory predicts that such an interaction could be a nonlocal and complicated interaction. In addition to two-body terms it will also depend on three-body and higher terms plus the model space adopted. Also, ideally, the specification of the effective interaction could be accomplished directly from the consideration of experimental nucleon-nucleon scattering data coupled with fundamental considerations. If such a free space nucleon-nucleon interaction were to be used for a finite model space, then modifications will have to be made to account for the excluded configuration. The details of some of such attempts for deriving quantitatively successful effective interaction are given in ref. [58], [71], [72] and [73].

4.5 Model Space and the Effective Wave Fuction.

The search for the most comprehensive wave function to describe the state of the nucleus completely, began before the introduction of the shell model, and perhaps shall continue for ever.

One reason for this is that the state of the nucleus depends upon many parameters. Some of them are 'known', while others are partially or completely unknown. Much has been achieved in describing the fundamental properties of the nucleus like energy, spin, isospin, shape and spectroscopic properties etc. But, in complete consistency with the definition of research in such paradigms, it has raised more questions than it has answered. So the research into wave function, which was a small dedicated field of research in the early days, has turned into a multidimensional and multidisciplinary collaborative research.

The second reason is, that including all possible effects which influence the state of the nucleus, increases the dimensions of the model space exponentially. However, the availability of more extensive computational resources like hypercube and other parallel processing techniques (details given in chapter 1), has given rise to a new era in massively parallel computational power. One can still regard this as another stage of development. Although it has advanced the frontiers of computational power far beyond what could have been conceived fifteen years ago, but still it has got a definitely recognizable limit. However, the degree of perfection in the development of nuclear wave function will certainly, as always, depend upon some human intellectual break through towards better understanding of the nucleus, along with the tools required to develope these ideas.

Our new implementation of the Glasgow Shell model (details given in chapter 2 and 3), which is based on the new parallel processing technique is yet another attempt in which we can handle some hundred thousands or perhaps even millions of states (later on), within a very reasonable time. We explain below the gradual evolution of research in both of these very related areas.

4.5.1 Model Wave Function.

The first representation of the nuclear wave function by Jensen and Mayer[42], was based on the single configuration for each nuclear level, i.e single nucleon in one single orbit. The availability of more computational facilities for numerically intensive calculation have now made it possible to implement the principles of configuration mixing with complete internal consistency. This has helped a great deal in understanding the structure of low lying nuclear states.

A modern shell model research is based on selecting the active nucleon orbits plus their allowed configuration and mixing/couplings for a particular model space, determination of the effective single particle potential and effective two body interaction between two nucleons. Isospin is considered to be good quantum number in these calculations. However, the current research in developing a better wave function, is aimed at the theoretical understanding of the complete spectroscopic characterisation of the nuclear levels provided by experiments. This complete characterisation includes prediction of the ground state energy (binding energy), the energies of excited states of all possible spin, parity and isospin values. In addition to this, the other parameters which are calculated include, level densities as function of excitation energy, multipole moments and electromagnetic transitions.

In the light of these developments, model wave functions have been validated against these observables. These wave functions have been used in the study of new nuclear phenomenon like isospin nonconservation, parity nonconservation and double beta decay. Some details of these studies is given in Ormand et. al.[74], Doi et. al.[75] and Brown[76].

4.5.1.1 Model Wave Function and Electromagnetic Properties

Many years of research has shown that beta decay and electromagnetic properties provide the most important, model-independent test for the wave function. It has been mostly beta and gamma decay, which have been most extensively studied and the results have been compared.

Recently, Etchegoyen and Wildenthal[77, 78] have made a detailed comparison of B(E2), B(M1) and Gammow-Teller (GT) matrix elements for

^{20}Ne and ^{21}Ne. Brown and Wildenthal[64] made another extensive comparison of recent experimental and calculated gamma decay results. More importantly this publication claims to have compared the entire body of the available results on M1 and GT for the sd shell states. The predicted results (using USD interaction) have been correct to within 10% or better w.r.t. experimental results for GT matrix elements. The detail discussion of these studies is given in refs. [77], [79-84].

This broad based agreement between the predicted and experimental results, for energies and spectroscopic factors strongly suggests that the physical states are dominated by 1s and 0d degrees of freedom. Also that the model wave function incorporates these degrees of freedom, at least roughly, in a correct way. This success of the model wave function in explaining the structure of the nucleus by using the electromagnetic properties has been well demonstrated in publications referred to earlier in this section. But these properties are strongly dependent on the single particle operators used in these calculations, which consequently affect the model wave function. That is why a lot of work has been done in optimizing the single particle operators and will be discussed in detail in section 1.6 on electromagnetic transitions.

4.5.2 Model Space.

A given shell model calculation is defined by the quantum numbers of the state under consideration, followed by a detailed specification of the basis space for the model and the chosen Hamiltonian. The dimensions of the model space become very large as we increase the number of active particles. In the 1s1/2 - 0d5/2 - 0d3/2 model space, there are only 24 active m-states, but the dimension of the full basis is very large. For ^{28}Si, which has just 12 active particles the size of the configuration in m-scheme M= 0, Tz=0 becomes 93710. While in J-T scheme it is 6706 for

J=3, T=1 configuration. The sheer size of the basis for the sd shell has prevented for quite long time the full basis calculations, until Whitehead and Watt et. al[1] used their then new computational techniques to make the first full basis calculations.

But if we extend the space to f, p and possibly to g and h shells, the size of the matrices becomes prohibitively large for diagonalization, as is reflected in the small example below.

Consider a nucleus 154 Sm 92, whose model space extends to g and h shells. We consider the excitation of the valence particles only.The 12 protons in this case,will occupy 5 orbits 1g7/2, 2d5/2, 3s1/2 and 1h11/2 The 10 neutrons outside the closed shell of N=82 will occupy the six orbits 1h9/2, 2f7/2, 3p3/2, 2f5/2, 3p1/2 and 1i1/2. We shall get the following distribution of positive parity states w.r.t their J values.

J value	Number of States
0+	41,654,193,516,797
2+	346,132,052,934,889
4+	530,897,397,260,575

Firstly, the no. of two body matrix elements required for constructing the Hamiltonian matrix of this size will be very high and they will be even more difficult to determine. Secondly, the diagonalization of such a giant size ($\approx 10^{14}$ for j=0) Hamiltonian matrix becomes impossible for the available techniques and computational methods.

In order to overcome these problems quite a few possible solutions have been proposed in the following and some of them have been actually implemented

Ø To find better models and methods to predict, which of the multiparticle configurations are most important, so that the reach of the active space can be extended without proportional expansion of dimensionalities.

Ø Discovering new and better mathematical and computational techniques for projecting angular momentum, setting up and diagonalizing large matrices; together with the use of faster/parallel computers with larger and faster memories.

Ø Using more discriminating and realistic assumptions about the effective nucleon-nucleon interaction.

Ø More theoretical and experimental study into the effects of the excluded configurations.

The actual implementation of these forceful ideas falls into three broad categories viz. new models supporting shell model, truncation schemes and new computational techniques. We discuss briefly the progress of these categories along the lines.

4.5.2.1 New Models

We can find many nuclear models which are based on the shell model. In most of these cases they were developed to provide a reasonable way of truncation for the otherwise unmanageable large set of basis states. In order to overcome the computational problems of too large model spaces, it has been usually assumed, that part of the nucleons form a closed shell. The excitation of the core (^{16}O, ^{40}Ca, etc.) is usually ignored or to some extent an approximation to it is included.

Some of the models which have received wide recognition include Nilson model[86], Bohr and Mottelston[99] for collective phenomena and the Interacting Boson Model (IBM)[87,88] for transitional and deformed nuclei.These models have been put forward to advance the sphere of shell model calculations into the previously impossible domains. It still remains a challenge, however, to connect together these various models. One promising way to tackle this problem could be to adopt the methods used in Monster-Vampire approach. The details of this approach could be found in ref. [89].

A good recent attempt has attempt has also been made by Iachello[85], in investigating the relationship between the shell model and the IBM. He has considered both the truncated IBM space and the full model basis space and has tried to establish a truncation-mapping procedure leading from the shell model to the IBM.

FDSM- Fermion Dynamical Symmetry Model is another effort in the same direction. This is a truncation scheme based on a class of models introduced by Ginocchio[91]. The prescription of this model arises from the symmetry or group structure of the shell model Hamiltonian or of the state being used. At this stage the degree of success about the approximations made by FDSM is not very clear.

4.5.2.2 Truncation Schemes.

Truncation schemes have been proposed with the common objective, to make approximate shell model calculations in case of impossibly large bases and to provide a better understanding of the conventional nuclear observables with a view to provide a more accurate wave function. We describe here some of the truncation schemes developed so far to achieve these objectives.

4.5.2.2.1 Restricted occupancy.

The underlying principle of this scheme is to omit some configurations, with high single particle energy.

Recently Ji and Wildenthal[16] have used this truncation scheme. They used standard shell model technique of constructing and diagonalizing the Hamiltonian in a finite Hilbert space to treat N=50 isotones. In addition, they used this truncation scheme and restricted the number of excitations across N=40 subshell to a maximum of four. This lead to errors estimated to be less than 20kev. But this kind of truncation seems to be insufficient in more typical situations.

4.5.2.2.2. Pseudo LS-Coupling

LS-coupling cannot be used for truncation beyond the sd shell, because of the strength of the spin-orbit force. The idea of pseudo LS-coupling was put forward long time ago[92,93] to fill this gap. This has not yet been used in a serious calculation. A test however, has been performed on Ni isotopes, whose space extended to f5/2, p3/2 and p1/2 orbits corresponding to a pseudo sd shell. This did show some better convergence than the jj-coupling.

4.5.2.2.3 Hartree-Fock-Bogliubov (HFB) approach.

HFB approach incorporates two most important features of the wave function viz. pairing and tendency to deform. The simple structure of the HFB wave function is achieved at the expense of breaking symmetries and loosing good quantum numbers like angular momentum and particle number. However they may be restored numerically by projection. This very ambitious idea, called Excited Vampire, has been extensively studied by Schmid et. al [89], where more details can be found.

4.5.2.2.4 Neutron-Proton Weak Coupling

Shell model calculations in an untruncated model space are based on forming a complete set of orthonormal basis states. This in turn is provided by the products of all states of the protons with all states of the neutrons. A possible truncation is obtained by first diagonalizing the nn and pp interaction in the separate factors and then retaining only those products, which are built from the lowest few eigen functions in each factor. The complete Hamiltonian, is then diagonalised in this restricted set.

Etchegoyen and Wildenthal[80], have recently employed this truncation scheme to make detailed calculations for ^{21}Ne and ^{20}Ne. In these extensive calculations they have achieved truncation factors from 2.3 to 4.4, while comparing the state dimensions in full and truncated shell model spaces. The B(E2) results have been reflected in a direct examination of the wave function. The overlaps obtained with the shell model wave function, were larger than 0.94 for all the states studied.

Earlier work, along these lines has been done by Mcgrory[93], Wong and Zucker[94], Chiang, Wong and Hsieh[95], Brussard and Glaudemans[107], and can be referred to for further details.

4.6 Electromagnetic Properties of the Nucleus.

It has been known for quite some time the Schrodinger wave equation possesses stationary states and eigen values. This certainly provides a close correspondence between the model eigen values and the observed energy levels, but fails to be complete test for the model. The most stringent test lies with the model wave function. The calculation of the electromagnetic properties is the most straight forward way to achieve this goal. As explained earlier in section 4.4.2, Cohen and Kurath used electromagnetic transition probabilities for the emission of gamma rays between different energy levels, as early as 1965, as a critical test for the validity of wave functions. They performed this test even after going through the normal procedure of fitting the two body matrix elements with the experimental data, both with jj and LS-coupling.

In this section we try to investigate the role played by the electromagnetic properties in explaining issues like the shape of the nucleus, validity of the wave function and ultimately proving why an electromagnetic probe is a preferred way of studying the nucleus. We shall start by looking at the effect of the electromagnetic field on the nucleus followed by the multipole moments, advances in and role played by the single particle operators and finally concluding by how and why these properties help to test the wave function.

4.6.1 Why Electromagnetic Properties are used to validate Nuclear Wave Function.

The electromagnetic properties are used to validate the nuclear wave function for the following reasons.

- It requires the fewest assumptions and gives the most clear cut predictions that are relatively easy to verify experimently.
- The electromagnetic operators are just single particle operators and their form is known for all the multiple orders. This gives considerable insight into a model for deformed nuclei and the limiting cases arising out of it.
- These operators connect the single particle functions to themselves and are less sensitive to admixtures of other configurations.
- In the equivalent experimental methods like nuclear reactions, their procedure is likely to cause some perturbation in the low lying states of the nucleus and also the ultimate fate of the projectile itself is unpredictable. Consequently the amount of uncertainty introduced is difficult to contemplate and compensate.
- Electromagnetic interaction is weak compared to nuclear interactions and is also known to a greater degree of accuracy. Such an electromagnetic probe disturbs the nucleus very little and one can measure the properties of a free and undisturbed nucleus.

These seems to be fairly good reasons for considering the electromagnetic properties to be a good test for the wave function. More over these calculated quantities are also useful indicators for the changes in the wave function.

4.6.2 Interaction with External Electric Field.

Static nuclear electric multipole moments can be studied by considering a nucleus placed in an electric field produced by a potential V(r). The interaction energy of the field with the nucleus can be given by:

$$E = e < \psi^{*}_{J\,m} (1, ..., N) \mid \Sigma \frac{1 + \tau_{i3}}{2} V(r_i) \mid \psi_{J\,m} (1, ..., N) > \quad .... (1.1)$$

where E is the interaction energy, $\psi_{Jm}(1, . . .,N)$ is the nuclear wave function characterised by the angular momentum J and its z-component m and τ_{i3} represents operators for the Pauli's spin matrices. $V(r_i)$ can be conveniently expanded around the centre of mass, because the wave function ψ_{Jm} usually refers to that point as well. Also we can expand the potential in terms of spherical harmonics $Y_{lm}(\theta, \text{ø})$, where θ and ø are spherical angles taken along the z-axis. Supposing that the centre of mass is the centre of the nucleus and r=0 here, we can write:

$$V(r_i) = \sum_{l=0} \sum_{m=-l}^{l} V_{l\,m}(r_i) \,.\, Y_{lm}(\theta_i, \text{ø}_i) \quad ....(1.2)$$

where $V_{lm}(r_i)$ is further given by :

$$Vlm(r) = \int Y^{*}_{lm} (\theta\ \text{ø}) \,.\, V(r) \,.\, d(\cos\theta)\, d\text{ø} \quad ....(1.3)$$

In fact function $V_{lm}(r_i)$ here is like the lth term in the Taylor expansion. Assuming the wavelength $\lambda = \omega/c$ » R, where R is the radius of the nucleus, then we can use $V_{lm}(r) \equiv r^{l}. V_{lm}$, where Vlm depends upon the detailed form of V(r), although it is itself independent of r. Using this approximation in eq 1.1 we get:

$$E = \sum_{l=0} \sum_{m=-l}^{l} V_{l\,m} \cdot Q_{lm} \quad ....(1.4)$$

where

$$Q_{lm} = e < \psi_{J\,m}(1, \;\ldots, N) \mid \sum_i \frac{1 + \tau_{i3}}{2} r_i^l \,.\, Y_{lm}(\Theta_i, \phi_i) \mid \psi_{J\,m}(1, \;\ldots, N >$$

The quantities Qlm in eq. 1.4 are often called as static electric multipole moments of the nucleus, in the state ψ_{Jm}. Eq 1.4 explains that the knowledge of these quantities, uniquely determines the interaction of the nucleus with any outside static electric field.

4.6.3 Interaction with magnetic field.

There are two factors which contribute mostly towards the nuclear magnetic multipole moments.

- The first one comes from the currents of the protons.
- The second one is due to the intrinsic magnetic moments of protons and neutrons.

Both are of the same order of magnitude.

We consider $\underline{l}$ to be the operator for the orbital angular momentum(in the units of $\hbar$)

Consider a charged particle (nucleon) of mass M. Its current will produce an effective magnetic dipole whose operator can be given[70] as:

$$\bar{\mu}_l = \left(\frac{e\,\hbar}{2Mc}\right)\bar{l} \qquad \text{.......(1.5)}$$

where the factor in bracket is called nuclear magneton.

Similarly, for the intrinsic magnetic dipole of the nucleons we can write it in terms of their spins.

$$\bar{\mu}_s = g^{(s)} \,.\, \frac{e\,h}{2Mc} \,.\, \bar{s} \qquad \text{......(1.6)}$$

Here M is the mass of proton and $g^{(s)}$ are called the spin g-factors and their values are given by :

$$g_p^{(s)} = 5.5855,\; g_n^{(s)} = -3.826.$$

The Magnetic dipole moment is the only nucleon property that is needed for the complete specification of the interaction of nuclei with a uniform magnetic field. We can derive a relation for the total magnetic moment of the nucleus in the units of nuclear magneton, by extending the summation over eq 1.5 and 1.6, which gives :

$$\bar{\mu} = \sum_{i=1} (g_i^{(l)} \bar{l} + g_i^{(s)} \bar{s}_i) \qquad ......(1.7)$$

where gi(l) are the orbital g-factors and their values are

$g_p(l) = 1$ and $g_n(l)=0$.

Using the isospin formalism this equation can further be written as :

$$\bar{\mu} = \Sigma \frac{1+\tau_{i3}}{2} \bar{l} + \left(\frac{1+\tau_{i3}}{2} g_p^{(s)} + \frac{1+\tau_{i3}}{2} g_n^{(s)} \right) \bar{s}_i$$

The fact that $\Sigma (\underline{l_i} + \underline{s_i}) = \underline{I}$ (the total angular momentum of the nucleus), we can substitute the value of the respective g-factors which reduces this equation to a more symmetric form.

$$\bar{\mu} = [\frac{1}{2} \bar{J} + 0.38 \sum_{\iota=1}^{A} \bar{S}_\iota] + [\frac{1}{2} . \Sigma \tau_{i3} (\bar{l}_i + 9.41 S_i) \qquad ....(1.8)$$

In this equation for the magnetic dipole moment, the first part is called the isoscalar part of $\underline{\mu}$, because it does not involve any dependence on the isospin operators. While the second term is called the isovector part. We can also notice that contributions of the nucleon spin to the isovector part of the magnetic moment is considerably bigger than the isoscalar part.

4.6.3.1 Quadrupole Moments, Quadrupole Operators and E2 Transition probabilities.

Assuming the charge distribution of the nucleus to be roughly spherical, then the electric quadrupole moment provides a measure of the extent to which this spherical charge distribution of the nucleus deviates. If Qop denotes the operator then the expectation value of the quadrupole moment can be written as:

$$Qop = < J, M = J \mid Qop \mid J, M=J > \qquad (1.9)$$

and the quadrupole operator can be written as :

$$e\,Qop = \int \rho_e(r) \, . \, r^2 . \, (3 \cos^2\theta - 1)\, d\tau \qquad (1.10)$$

where $\rho_e(r)$ is the charge density. Because a neutron carries no charge, therefore its quadrupole moment vanishes.

The diagonal elements of the quadrupole operator Qop give values of the static quadrupole moment while the off-diagonal elements (i.e. transition moments) are involved in the electric quadrupole (E2) processes , like gamma decay, Coulomb's excitation etc.) The transition rates and the amplitudes for such processes depend on the matrix elements of the Electric Quadrupole Tensor 'M_{dt}', given[108] by :

$$M_{dt}(E2, \mu) = \int \rho_e(\underline{r})\, r^2\, Y_{2\mu}(\theta, \o)\, d\tau \qquad (1.11)$$

The transition between the states J_i and J_f is obtained after averaging over the initial quantum numbers M_i and summation over the unobserved final projection quantum number M_f. Since the multipole operators are irreduciible tensors, we can apply the Wigner-Eckarts theorem[66]. The total transition rate will be given by :

$$B(E2, J_i \rightarrow J_f) = \sum_{\mu M2} [< J_f M_f \mid M_{dt}(E2, \mu) \mid J_i M_i]^2$$
$$= \frac{[< J_f \mid\mid M_{dt}(E2) \mid\mid J_i]^2}{(2J_i + 1)} \qquad ....(1.12)$$

4.6.3.2 Magnetic Dipole Moments, Dipole Operator and M1 Transitions.

The nuclear magnetic moment provides a sensitive test for the nuclear scheme, because of the great difference between the g-factors associated with various components of the total nuclear momenta of neutrons and protons. This fact is well illustrated by eq 1.8. We can define the magnitude of the magnetic moment $\underline{\mu}$ in terms of the matrix elements as:

$$\mu = < J, J = M \mid \mu_Z \mid J, M = J > \qquad (1.13)$$

These matrix elements are usually expressed in terms of the magnetic dipole tensor M_{dt}. The nondiagonal matrix elements of the magnetic moment operator determine the amplitudes of M1 transitions. The transition probabilities for the single particle transitions are given[71] by:

$$T(M_\lambda) = \frac{2(\lambda+1)}{\lambda[(2\lambda+1)!!]^2} \cdot \frac{e^2}{\hbar c} \cdot \left(\frac{\omega R}{c}\right)^{2\lambda} \cdot \omega\left(\frac{\hbar/Mc}{R}\right)^2 \cdot$$

$$\left(\frac{1}{2} g_s^\lambda - g_l \frac{\lambda}{\lambda+1}\right)^2 \cdot S(J_i, \lambda, J_f) \cdot R_{\lambda-1}$$

where λ is the multipolarity, ω/c = k, R is the radius of the nucleus, S is a combination of vector addition function, and the radial integral can be approximated to:

$$R_\lambda = \left(\frac{3}{\lambda+3}\right)^2 .$$

The B(M1) value can also be written in terms of the matrix element as:

$$B(M1) = \frac{[< J, M = J \mid \mu_z \mid J, M = J >]^2}{(2J_i + 1)}$$

4.6.4 New Optimizations into the Transition operators.

It has been of considerable interest for quite a long time to use the electromagnetic probes for the nuclear structure studies for the reasons explained at the start of this section. At the same time efforts have been made continuously to determine corrections and renormalizations of the operators used to account for higher order effects of configuration mixing over many major shells. In addition the effect of mesonic exchange currents and isobaric admixtures inside and outside the sd shell have also contributed towards these updates.

One such effort has been made by Brown and Wildenthal[111] to find an empirically optimum operator for the sd shell nuclei. They first deduced the matrix elements from the experimental data on M1 matrix elements and magnetic moments. Then they analysed them with the calculations based on the full basis shell model wave function. Finally, they extracted the parameters for an effective M1 operator. The empirical parameters so deduced, are analogous to the predicted corrections for the free nucleon M1 operator, which arise from the facts of higher order configuration mixing and mesonic exchange curents.

Towner and Khanna[111] also undertook theoretical studies into the optimization aspects of transition operators. They made some predictions, which became a good guide line for many, who worked in this area later on. Etchygoyen et. al.[115] have used these prediction in extracting their empirical values for higher order corrections to the M1 operator. They obtained corrections in terms of the effective single particle matrix elements, which when combined with the model transition densities, yield a minimum deviation of the model predictions of M1 strength, from a set of 250 selected experimental values. a good qualitative and quantitative detail is also given.

Towner and Arima[112] are other good attempts in this process of optimization. They used operators for the M1 transitions, which take into account the effects of mesonic exchange currents and wave function admixtures. Most recently, Brown and Wildenthal[113] have used, their earlier empirically effective operators for making calculations. The calculated Gammow-Teller (GT) matrix elements and M1 transitions were found to be qualitatively in agreement with those of [112], which were based on perturbation theory estimates. These publications [112] and [113] give full detail of the operators used.

A more recent study has been made by Hwang et.al[114]. They used E2 transitions extended operator and a complete second order M1 transition operator. They used them to calculate mixing ratios (E2/M1). On the basis of the results they obtained by using these operators, it has been demonstrated that a satisfactory description of M1 admixtures can be obtained with the extended operators.

4.6.5 E2 and M1 Matrix Elements and the Effective Charges.

The value of the effective charge for proton and neutron has been another parameter in the optimization of single particle operators. Different values have been proposed at different times for different reasons to arrive at the best possible results and we try to summarize these attempts here.

4.6.5.1 E2 Matrix Elements.

Many approaches have been considered to obtain E2 matrix elements, which would yield very close results to the experimental data. One of such approaches has been to use free-nucleon charges i.e ep = 1e and en = 0, in the E2 operator. While some others have used parameterisation of the radial wave function and the least square fit of the shell model densities to the E2 data, in order to get the effective charge.

Brown, Arima and Mcgrory[116], Carchidi and Wildenthal et. al.[117], and Brown et.al.[118] have made calculations using both these approaches and have compared them with experimental data. These studies have been quite useful in building up a consensus with experimental results.

Brown et.al.[118] used mirror nuclei for their calculations and discovered that the isovector effective charge turned out to be quite sensitive to the radial wave function. Perhaps, this is because many of the isovector transitions occur between loosely bound states in the lower part of the shell. They used different radial wave functions and found the values of the effective charges to be between,

$$e_p + \delta e_p - \delta e_n = 1.0e \qquad \text{and} \qquad e_p + \delta e_p - \delta e_n = 0.65e.$$

A very recent attempt towards the calculation of E2-matrix elements and the effective nucleon charge has been reported by Brown and Wildenthal[119]. They used the experimental data from [81] to derive the E2 matrix elements ($[< J_i \mid O(E2) \mid J_f >]^2 / (2j_i + 1)$). They also used various parameters to the radial wave function, including the harmonic oscillator form $\hbar\omega = 45\ A^{-1/3} - 25\ A^{-2/3}$.

The graphs presented reflect a good comparison of the matrix elements obtained using free nucleon charge and the effective charge. Their least-squares fit gave the following values for the effective charges.

$e_p + \delta e_p + \delta e_n = 1.78(3)e$ for isovector effective charge

$e_p + \delta e_p - \delta e_n = 0.8(1)e$ for isoscaler effective charge

Calculations based on the E2 operator using these values of effective charge have produced excellent comparative results.

4.6.5.2 M1 Matrix Elements.

The M1 operator incorporates both spin orbital operators and acts on both the isoscalar and isovector parts. But the coefficient of 's' is much larger than that of 'l'. So the contribution from the isovector part is much greater than from the isoscalar. That is why , Brown et.al.[119] have calculated the isovector and isoscalar parts separately. They have calculated the matrix elements ($B(M1) = [<J_i \mid O(M1) \mid J_f >]^2 / (2J_i + 1)$), and compared with those from the experimental data [120] and [122].

They derived the experimental values for magnetic moments from the data given in [122] and tried to reproduce these values by using free-nucleon prediction and the effective operator. Separate comparisons for the isovector and isoscalar matrix elements for sd shell moments have been made because of the reason explained earlier. The graphs have been presented to reflect the deviations of isovector moments of the M1 operator. However, these deviations in the free nucleon predictions between the isoscalar moments and experiment, still need a significant correction to the free nucleon operator. This deviation has not yet been explained in terms of a single reduction factor.

4.6.6 Units of Electric Quadrupole Moment and Magnetic Dipole Moment.

During the study of mass 21, we have made calculations for the B(E2) and B(M1) values for different nuclei. Our results for these quantities were in the units of e^2xfm^4 and nuclear magneton respectively. While some of experimental results which we compared(e.g. Catford et.al.[134]), were in Weisskopf units. We have calculated this conversion and included in Appendix D.

CHAPTER 5

Calculations and Results

The calculations and results reported in this section relate to the six isobars, ^{21}O, ^{21}F, ^{21}Ne, ^{21}Na, ^{21}Al, ^{21}Mg. The parameters for these calculations (like the effective interaction, s.p.e., etc.), have been so chosen that we have been able to get separate sets of results, including and excluding Coulomb's interaction. In this section we first give a brief explanation, of what studies have been undertaken and the values that have been calculated. Then we shall explain the significance of the results and the predictions arising out of them.

5.1 Study of the nuclei using PW interaction

In this study we have used the effective interaction and the single particle energies which are given in Wildenthal et. al.[59]. This interaction does not take into account the effect of Coulomb's force. As a consequence of that, the energies of protons and neutrons in a shell have the same value. This interaction has been discussed in detail in section 1.4 and values for the two body matrix elements and single particle energies are given in appendix A.

5.1.1 For this interaction we have calculated the energy eigenvalues, their corresponding angular momentum, isospin and positive parity. In the absence of Coulomb's interaction one would expect the eigenvalues and occupancies etc., corresponding to both mirror nuclei, to be exactly equal. Our calculations confirm this fact.These calculations have been made for all the six nuclei and have been compared in tables 5.10-12 and fig 5.7-9.

5.1.2 We have also calculated electric transition rates B(E2) for the gamma transitions between the possible states such that the angular states momentum of the ray conforms to $| J_i - J_f | \leq 2$.
The B(E2) values for all the nuclei are given in tables 5.19-24 & fig 5.16-21.

5.1.3 Magnetic transition rates B(M1), have also been calculated for the nuclear states such that $| J_i - J_f | \leq 1$. These B(M1) values for all the transitions have been given in tables 25-30 & fig 5.22-27.

5.1.4 Occupancies of nucleons in their respective subshells have been calculated. This information, along with the others mentioned above, has been found to be quite useful, particularly in the study of states which have swapped over in a mirror and have been given in table5-31-36 and fig 5.40-45.

5.1.5 The results mentioned above have been compared against the available experimental data. These comparison have been done in the form of tables and energy level diagrams, and have been included appropriately where their discussion has been made.

5.2 Study of the Nuclei Using CWC Interaction.

The detailed discussion about CWC has been already given in section 4.4. The main differences of CWC with the PW interaction however, can be summarised as given below.

- The single particle energies for the protons and neutrons are different in each subshell and also proton values are different from neutron values.
- The Hamiltonian matrix elements include the effect of Coulomb's force and they are different from the ones in PW.

5.2.1 All calculations, which have been noted above for PW interaction have been made for the CWC as well. One of the objectives of these calculations is to demonstrate the difference Coulomb's force makes to the energy eigenvalues and the occupancies of the nucleons. Because of that the tables/diagrams referred to above contain data calculated for CWC interaction as well for quick comparisons.

5.3 Study of Nuclei Using USD interaction.

This interaction was also devised for the sd shell model space, but the parameters defining the interaction are different from PW or CWC. The two body matrix elements in this interaction are mass dependent, which has been explained in detail in section 4.4.7. We have included appropriate amendment in our code to update the tbme for A=21. However, from the point of view of exploring the effect of Coulomb's interaction, this interaction is not much different from the PW. But still we have included the results of its eigenvalues in our comparisons.

5.4 Comparison of Results :

We have made calculations for the six nuclei using three effective interactions namely PW, CWC and USD. The values calculated have been found to be different, because of the fundamental differences for each interaction. One such difference is effect of Coulomb's force. In this section we compare the results obtained with these interactions, in order to explore the effect of Coulomb's force, and also with the experimental results, where they have been available.

5.4.1 Comparison of eigenvalues :

The comparison of energy eigenvalues for the three interactions has been made in the tables 4-9, which are in ascending order of Z from O to Al. Column 'H' in each of these tables gives the energy difference ΔE between PW and CWC interactions, corresponding to each eigen state of that nucleus. The~~se~~ values corresponding to USD interaction have also been included for comparison.

The maximum value for ΔE = 3.81 MeV, is found in table 5.14 for ^{21}Al, where there are maximum number of protons (=6) in these nuclei. In order to elaborate this effect further, we have included six energy level diagrams Fig. 5.10-5.15, which show the variation of levels as we switch from CWC to PW and USD interactions.

These eigenvalues have been compared against the existing experimental energy values. This comparison is reflected in the energy level diagrams Fig. 25.28-31.

It can be seen from the diagrams that many states are in good agreement with the experiment. We concentrate our attention on those states which are relatively in poor agreement with the experiment, as their properties can give new information on the structure of the nucleus. In cases where there is little experimental data, we compare with the results of different interactions from tables 5.13-18. Also it can be seen that in most cases the largest discrepancies occur for high spin states. This might be because configurations outside the sd-shell are more important for these states than for the lower energy low spin states. A summary of all such states for which ΔE is greater than 0.2 is given in the table 5.1 below.

Nucleus	States with 0.2 < ΔE < 1.0	ΔE > 1.0
^{21}O	13_3,11_1, 11_3, 9_3, 9_2, 9_1, 5_2, 5_3, 3_1, 3_2, 3_3,1_1	7_2
^{21}F	17_2, 17_3, 15_2, 13_1, 13_3,9_3, 7_1	
^{21}Ne	19_2, 17_1, 17_2,15_2, 15_3, 13_1, 13_3,11_2,11_3	7_3
^{21}Na	19_2, 17_1, 17_2, 15_2, 15_3, 13_1, 13_3, 11_2, 9_1, 9_2, 1_1	11_3
^{21}Mg	17_1, 17_3, 15_2, 15_3, 13_1, 13_2, 13_3, 11_1, 11_2,	
	11_3, 9_1, 9_2, 9_3, 5_2.	17_2, 15_1
^{21}Al	13_1, 11_1, 11_2, 9_1, 9_2, 9_3, 7_1, 5_2, 3_3	13_1, 11_3, 7_2

table 5.1 : Nuclear states, whose energy changes by 0.2 or more when Coulomb's interaction is switched on or off.

5.4.2 Using Electromagnetic properties to analyse these states.

The states given in the table above have been chosen for further analysis because their energy values seems to be in poor agreement with experimental and/or other transitions. This discrepancy could be due to either or both of the following reasons.

1- The wave function representing the states is not correct.

2- Parameters of the Hamiltonian, like two body matrix elements and single particle energy, corresponding to these states, have not been correctly determined.

In such a situation electromagnetic properties can help to investigate the above reasons. The B(E2) or B(M1) value for a γ_ray depends precisely on the wave functions of the initial and final states involved in the interaction. We have calculated B(E2) and B(M1) values of states for all the six nuclei, which are given in tables 19-24, Fig. 16-21 and the corresponding experimental results for these transitions are given in Fig. 32-34 & 5.35-37. The standard deviation of the calculated results from the experimental results have also been calculated for ^{21}Ne, because the experimental results for it provided the ± deviation. This standard deviation is given in Fig.5.38 for B(E2) and Fig. 5.39 for B(M1).

These calculated transition values between different states agree with experiment in many cases, while it may not be in some other cases. In either case the following logical conclusions can be drawn about the interacting states ψ_i and ψ_f.

Ø If the calculated transitions agree with the experimental ones then:

- ψ_i is correct and ψ_f is also correct.
- ψ_i is incorrect and ψ_f is also incorrect.
- It is not possible that ψ_i is correct and ψ_f is incorrect, or vice versa.

Ø If the calculated values do not agree with the experimental then :

- ψ_i is correct and ψ_f is incorrect.
- ψ_f is correct and ψ_i is incorrect.
- It is not possible that both ψ_i and Ψf are correct.

In order to facilitate comparison we have collected B(E2) and B(M1) values for the states already picked up for investigation in table 5.2 given below. The~~se~~ values corresponding to ^{21}Mg states only have not been entered in this table because of the shear greater number of states which qualify for this. For any reference to their transition rates reference can be made to table nos. 5.19 and 5.25.

Nucleus	States / Values									
^{21}O	13_3	11_3	11_1	9_3	9_2	9_1	5_2	3_1		
B(E2) γ=1	0.04	0.06	1.15	1.24	2.49	.108	1.13	1.97		
γ=2	0.015	0.006	1.96	0.61	.08	.942	4.30			
B(M1)*100	0.13 0	0.011	.0013	0.39	2.78	31.07	29.48	1.55		
^{21}Al	13_3	13_1	11_3	11_2	9_3	9_1	7_2	7_1	5_2	3_3
B(E2) γ=1	.94	15.84	12.29	17.8	10.38	17.53	4.35	65.0	9.16	12.7
γ= 2	.33	13.1	.205	4.14	5.31	.011	8.55	18.24	40.5	-
B(M1)*100	.24	.13	17.42	13.6	.005	16.9	1.00	4.6	38.5	1.56
^{21}F	17_3	17_2	15_2	13_3	13_1	9_3	7_1			
B(E2) γ= 1	0.009	2.57	0.338	0.382	7.55	1.03	2.96			
γ= 2	0.019	0.08	0.172	1.72	12.28	.0174	20.54			
B(M1)*100	.023	30.39	21.52	2.479	78.43	4.98	9.717			
^{21}Ne	19_2	17_2	17_1	15_3	15_2	13_3	13_1	11_2		
B(E2) γ=1	0.017	0.015	8.12	0.33	0.162	0.13	12.36	.27		
γ= 2	0.66	.015	26.71	.937	0.699	0.002	42.03	.26		
B(M1)*100	0.168	10.20	70.78	30.96	5.87	9.25	97.32	10.6		
^{21}Na	19_2	17_2	17_1	15_2	13_3	13_1	11_2	9_2	9_1	
B(E2) γ=1	.036	.919	9.03	1.4	.628	12.48	.003	.565	32.3	
γ= 2	.14	.867	.32.8	.98	.692	37.54	.12	.009	49.51	
B(M1)*100	0.18	15.52	87.24	5.19	10.29	1.8.0	9.77	12.29	58.23	

table 5.2 : Comparison of Electromagnetic γ-decay transition rates (CWC) for states sensitive to the Coulomb's interaction & noted in table 5.1.

5.4.3 Applying the multipole operator test to the wave function of the mirror states.

In order to explain this test we consider ^{21}O-^{21}Al mirror, in which one nucleus has got all valence particles as protons while the other has got all neutrons.

The single particle operators for the electric quadrupole moment for both of these nuclei would be the product of the effective charge on a valence particle (e_p or e_n) and angular momentum term (like $\Sigma(r\ \theta)$, so that

B(E2) for ^{21}O = $(e_n / e_p)^2$ x (B(E2) for ^{21}Al). (A)

In our calculations we have used e_n = 0.5, e_p = 1.5. This relation should hold if the wave functions for both the interacting states are identical in both the nuclei.

Similarly, we know that the operator for the magnetic dipole moment depends upon μ_n and μ_p, the magnetic moments for neutron and proton. In the case of ^{21}O and ^{21}Al, their magnetic moments will be related by the following relation

B(M1) = $(\mu_n/\mu_p)^2$. (B(M1) for ^{21}Al)

provided the condition on the wave functions mentioned earlier holds. Using μ_n = 1.91 and μ_p =2.78 we get :

B(M1) = 0.687 x B(M1) for ^{21}Al (B)

In order to see, how far the wave functions representing different states could be identical to their counterparts in the mirror nuclei, we derive ^{21}O transition values from ^{21}Al values using the above formulae and see how far their wave functions have been found to be identical. We shall concentrate our attention only on those states which have already been picked out in table 5.1 for showing some inconsistent behavior. The first set of ^{21}O values is for B(E2), when angular momentum of the emitted γ ray is equal to one and two respectively, followed by the B(M1) values. The derived values means corresponding values obtained from ^{21}Al data using eq. A and B.

B(E2)	13_3	13_1	11_3	11_2	9_3	7_2	5_2
γ= 1	0.04	1.44	0.06	0.19	1.24	0.04	1.13
Derived using (A).	.094	1.58	1.22	0.178	1.03	0.43	0.91
γ= 2	0.17	0.09	0.006	0.36	0.61	1.14	4.30
Derived using (A).	0.03	1.31	0.020	0.4	0.53	0.85	4.05
B(M1)	0.13	0.05	0.011	7.2	0.39	0.86	29.48
Derived using (B).	0.16	0.09	11.96	9.33	0.003	0.68	26.47

It can be inferred from the above comparison that the wave functions for both the nuclei are not identical.

5.5 Configuration Analysis of States using Occupancy Information.

The information as to how the particles have been distributed among different shells in a particular nucleus, can be quite useful in comparing the structure of two states. Moreover it provides good indications, whether two states can have strong electromagnetic interactions. This ultimately helps in testing the wave function. When this information is organised in the form of occupancy diagram, the presence of rotational bands (if any) can be directly spotted and put to further tests for confirmation. Because of this significance, we explain below how these values have been calculated and used in analysing the structure of the six nuclei in general and their states noted in table 5.1-2, in particular.

5.5.1 Calculation of the Subshell Occupancies.

The sd_shell consists of three subshells, d3/2, s1/2 and d5/2. In our implementation we have represented the basis states by Slater determinants. Our shell model calculations result in a state which is described by thousands of amplitudes, each of which is associated with a Slater determinant. We reduce the information in the shell model wave function, from these thousands of amplitudes to only three occupancy values corresponding to the three subshells in the sd shell. These occupancy values represent the probability for these number of particles to be in these subshells. This information is computed from the eigen vectors.

Let n5/2, n3/2 and n1/2 represent the average number of particles in the d5/2, d3/2 and s1/2 respectively. All the states of the nuclei satisfy the following relation

$$n5/2 + n3/2 + s1/2 = A - 16$$

where A is the mass number of the nucleus and any excitations from the ^{16}O core have been neglected. We have calculated occupancies for positive parity states for all the six nuclei. They conform to the above condition and are given in tables 5.31-36.

5.5.2 Occupancy Diagrams.

It can be seen from the above condition that the three occupancy values are not independent of each other. We have chosen to regard n5/2 and n3/2 as independent variables and n1/2 as dependent variable. So we can plot a two dimensional diagram by taking n5/2 values as abscissae and n3/2 values as ordinates. This will ensure that states with different subshell occupancies will be represented by different points. We have plotted such occupancy diagrams for the six nuclei using CWC interaction.

5.5.3 Significance and Extraction of Physical Meanings from the Diagrams

The next logical step would be to decide how to attribute physical meaning to the points on the occupancy diagrams, in terms of structure of the respective states.

5.5.3.1 We can start by this argument that two states with similar structure will have nearly the same number of nucleons in three subshells. More over such states will appear very close together in the occupancy diagram. However, it may not be true vice versa. This indicates that occupancies reflect some aspects of the structure of the states but not others. We summarise the premises and their logical conclusions, with regards to the structure of the states and the occupancies.

- If two states have the same structure, they will be represented by the same point on the diagram.

- Two different points on a diagram will correspond to two different states having different structures
- Two states inherently different in structure, may not necessarily have different positions on the occupancy diagram.
- If two states are represented by the same point on the diagram, they may not have the same structure.

This can be concluded by saying that, a necessary but not sufficient condition for two states to have same structure is, that they coincide on the occupancy diagram.

5.5.3.2 The situation of points on the diagram can help to assess the probability of an electromagnetic transition between the states corresponding to the points. Large collective transitions can occur only if the initial and final state lie close together on the diagram. This is because transition rates are computed by calculating matrix elements of the single particle operators between an initial and final state. Such an operator cannot connect Slater determinants which differ by more than one particle in any given shell. So the effective sphere of influence of the initial state with regards to an electromagnetic transition is therefore limited within a region of one unit, around the position of initial state. If the final state falls outside the sphere the transition will certainly be very small.

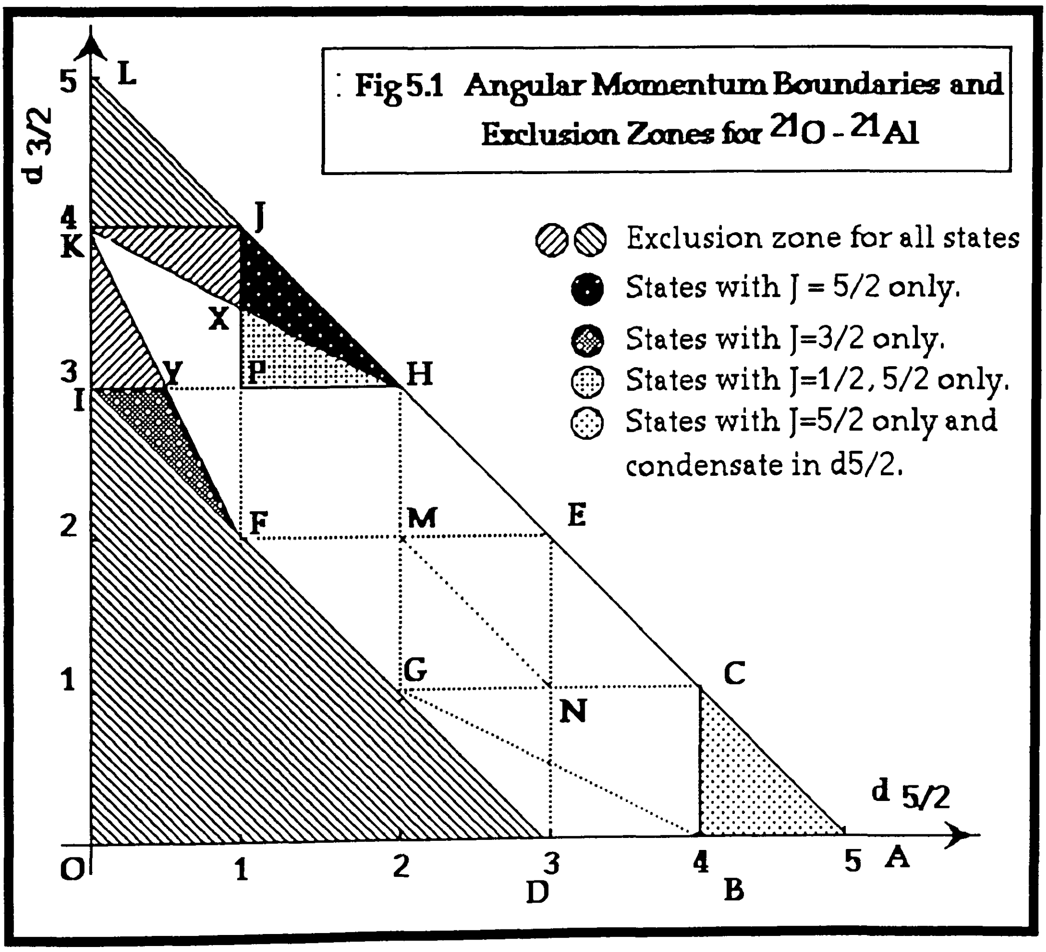
5.5.4 Predicting States to lie within the Angular Momentum Boundaries and Identification of Condensate States.

In mass 21 we have got five valence particles, which includes all possible combinations of protons and neutrons. We can calculate in advance the probable position where an eigen state corresponding to a particular angular momentum could be expected on the occupancy diagram. Also we can mark in advance some exclusion zones, where no state can lie or where highly condensate states could be expected. Different nuclei under investigation have got different numbers of protons, as such they will have different configurations due to the restrictions imposed by the Pauli's principle. However, mirror nuclei will have same angular momenta although the corresponding eigenvalues will be different. So the exclusion zones for one nucleus will apply to its mirror as well, as we discuss them in the next sections.

5.5.4.1 Exclusion Zones and A.M. Boundaries for ^{21}O-^{21}Al Nuclei.

Fig. 39 shows the exclusion zones for both these nuclei, which have got five valence neutrons and five valence protons respectively. We have been able to draw the following conclusions from this occupancy diagram.

- Point A in the diagram represents a highly condensate state in d5/2.
- We have worked out the allowed values of A.M. corresponding to all the points shown on the grid, by coupling angular momenta of particles distributed over various subshells. These values are given below . The points representing these values would help us to mark the allowed/excluded regions for states with different A.M. values.



: Fig 5.1 Angular Momentum Boundaries and Exclusion Zones for ^{21}O - ^{21}Al

▨▧ Exclusion zone for all states
● States with J = 5/2 only.
States with J=3/2 only.
States with J=1/2, 5/2 only.
States with J=5/2 only and condensate in d5/2.
d 3/2
d 5/2
5
4
3
2
1
O
1
2
3
4
5
L
J
K
X
Y
P
H
I
F
M
E
G
C
N
D
B
A

Points	Configuration	Allowed J-Values
A	$(d5/2)^5$	(5/2) only
B	$(d5/2)^4$ $(s1/2)^1$	(1/2,. . .,9/2)
C	$(d5/2)^4$ $(d3/2)^1$	(1/2,. . ., 11/2)
D	$(d5/2)^3$ $(s1/2)^2$	(3/2, 5/2, 9/2)
E	$(d5/2)^3$ $(d3/2)^2$	(1/2,. . ., 13/2)
F	$(d5/2)^1$ $(s1/2)^2$	(1/2,. . .,9/2)
G	$(d5/2)^2$ $(d3/2)^1$ $(s1/2)^2$	(1/2,. . ., 11/2)
H	$(d5/2)^2$ $(d3/2)^3$	(1/2,. . ., 11/2)
I	$(d3/2)^3$ $(s1/2)^2$	(3/2) only
J	$(d5/2)^1$ $(d3/2)^4$	(5/2) only
K	$(d3/2)^4$ $(s1/2)^1$	(1/2) only
M	$(d5/2)^2$ $(d3/2)^3$	(1/2,. . ., 13/2)
N	$(d5/2)^3$ $(d3/2)^1$ $(s1/2)^1$	(1/2,. . ., 13/2)
P	$(d5/2)^1$ $(d3/2)^3$ $(s1/2)^1$	(1/2,. . .,9/2)

- Because d3/2 cannot have more than four particles in case of these nuclei, so triangle LJK forms an exclusion zone for all states. Similarly, because s1/2 cannot have more than two particles, so triangle IDO forms another exclusion zone for all states.
- Using the A.M. values for different points noted above, we have marked the allowed regions for states with different angular momenta, which are given below.

J-Value	Allowed Region bounded by points	J-Value	Allowed Region bounded by points
13/2	ENM	11/2	HCG
9/2	PHCBDF	7/2	PHCBGF
5/2	JHADF	3/2	IHCBDG
1/2	KHCBGF		

- We note that as J decreases from 13/2 to 9/2 the allowed area extends outward. The allowed area for J= 9/2 states includes area for J=11/2, which further includes area allowed for states with J= 13/2.

When we move to J= 7/2, however, the area is actually reduced by triangle BDG, because point D does not include J= 7/2. In going to 5/2, the area extends beyond the area for J= 9/2, and now includes triangles JHP and CAB. The area for J= 3/2 is reduced by triangles JHP and CAB, but increases by IPF. Finally for J= 1/2, areas GBD and IPF are excluded again, but KHP and KPF are included. Thus the allowed areas do not spread out as one would naively expect.

In particular, areas CAB and JHX can be occupied by J=5/2 states only and area IYF can be occupied by J= 3/2 states only.

- Area CAB is of particular interest as it corresponds to a condensation of nearly all the particles into d5/2 shell, which results in very low energy states.

5.5.4.2 Exclusion Zones and A.M. Boundaries for ^{21}F-^{21}Mg Nuclei.

The predicted inclusion and exclusion zones, for eigen states with different A.M., corresponding to these nuclei have been shown in fig 5.2

.

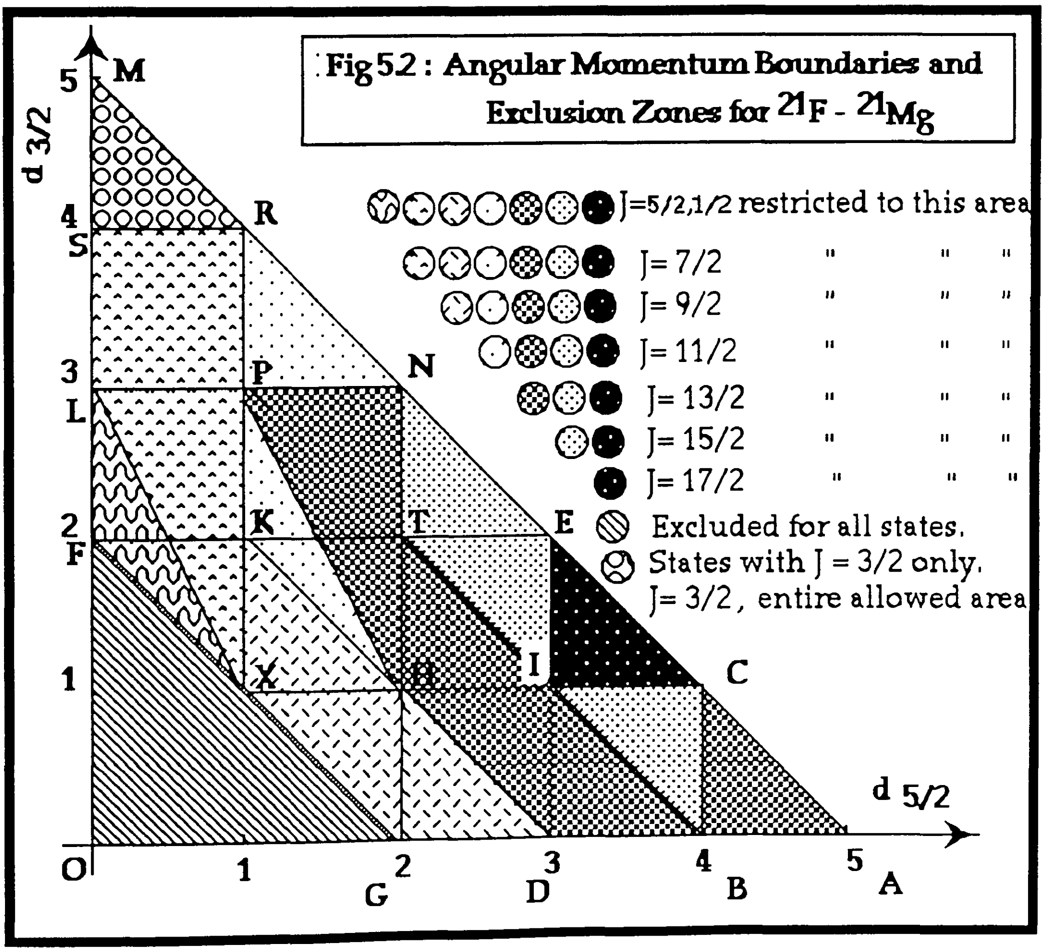


Fig 5.2 : Angular Momentum Boundaries and Exclusion Zones for ^{21}F - ^{21}Mg

We have been able to draw the following important conclusions from this occupancy diagram.

- In case of these mirror nuclei s1/2 shell can accommodate a maximum of three particles in compliance with the Pauli's principle. Because of that, triangle FGO forms the only exclusion zone for all states in this diagram.
- Further marking of exclusion zones for states with certain A.M. has been done by finding the permissible A.M. values for different points on the grid, which are given below.

Points	Configurations	Allowed J-Values
A	$(d5/2)^5$	(1/2,. . ., 13/2)
B	$(d5/2)^4$ $(s1/2)^1$	(1/2,. . ., 15/2)
C	$(d5/2)^4$ $(d3/2)^1$	(3/2,. . ., 17/2)
D	$(d5/2)^3$ $(s1/2)^2$	(1/2,. . ., 13/2)
E	$(d5/2)^3$ $(d3/2)^2$	(1/2,. . ., 17/2)
F	$(d3/2)^2$ $(s1/2)^1$	(1/2,. . .,5/2)
G	$(d5/2)^2$ $(s1/2)^1$	(1/2,. . .,9/2)
H	$(d5/2)^2$ $(d3/2)^1$ $(s1/2)^2$	(1/2,. . ., 13/2)
I	$(d5/2)^3$ $(d3/2)^1$ $(s1/2)^2$	(1/2,. . ., 17/2)
K	$(d5/2)^1$ $(d3/2)^2$ $(s1/2)^2$	(1/2,. . ., 11/2)
L	$(d3/2)^3$ $(s1/2)^2$	(1/2,. . .,7/2)
M	$(d3/2)^5$	(3/2) only
N	$(d5/2)^2$ $(d3/2)^3$	(1/2,. . ., 15/2)
P	$(d5/2)^1$ $(d3/2)^3$ $(s1/2)^1$	(1/2,. . ., 13/2)
R	$(d5/2)^1$ $(d3/2)^4$	(1/2,. . ., 11/2)
S	$(d3/2)^4$ $(s1/2)^1$	(1/2,. . .,7/2)

- Using the information about these points we have marked the A.M. boundaries for populating states with particular A.M. value. Because of the increased number of basis states in this case, some points in a boundary correspond to more than one state for the same value of A.M.
- Unlike the O-Al mirror pair, in this case the permitted A.M.

Boundaries spread outward very symmetrically, as we move from higher to lower spin states. This effect has been depicted in fig. 5.2 and is summarised below by giving the permitted area corresponding to particular A.M value.

J-Value	Allowed Region	J-Value	Allowed Region
	bounded by points		bounded by points
17/2	ECI	15/2	NCBT
13/2	NADHP	11/2	RADK
9/2	RAGX	7/2	RAGXLS
5/2	RAGFS	3/2	entire allowed area

It is clearly reflected from above and fig 40 that each allowed region forms a part of the permitted area for the next lower spin state except J= 3/2.

- It is interesting to note that point M in the diagram is permitted for states with J= 3/2 only, while neighboring points R and S allow the representation of states with A.M. (1/2, . . ., 7/2) and (1/2, . . ., 11/2) respectively. This implies that area MRS is permitted for states J= 3/2 only and all other states are excluded from this area. More over, if there happens to be any state in this area, it will be highly condensate in the d3/2 shell and will be a high energy state.

5.5.4.3 Exclusion Zones and A.M. Boundaries for ^{21}Ne-^{21}Na Nuclei.

We have found that the A.M. boundaries for the states corresponding to these nuclei grow most symmetrically of all the three mirrors we have investigated. These boundaries and exclusion zones have been shown in fig. 5.3. The following details emerge from the diagram and help in understanding the structure of the respective nuclear states.

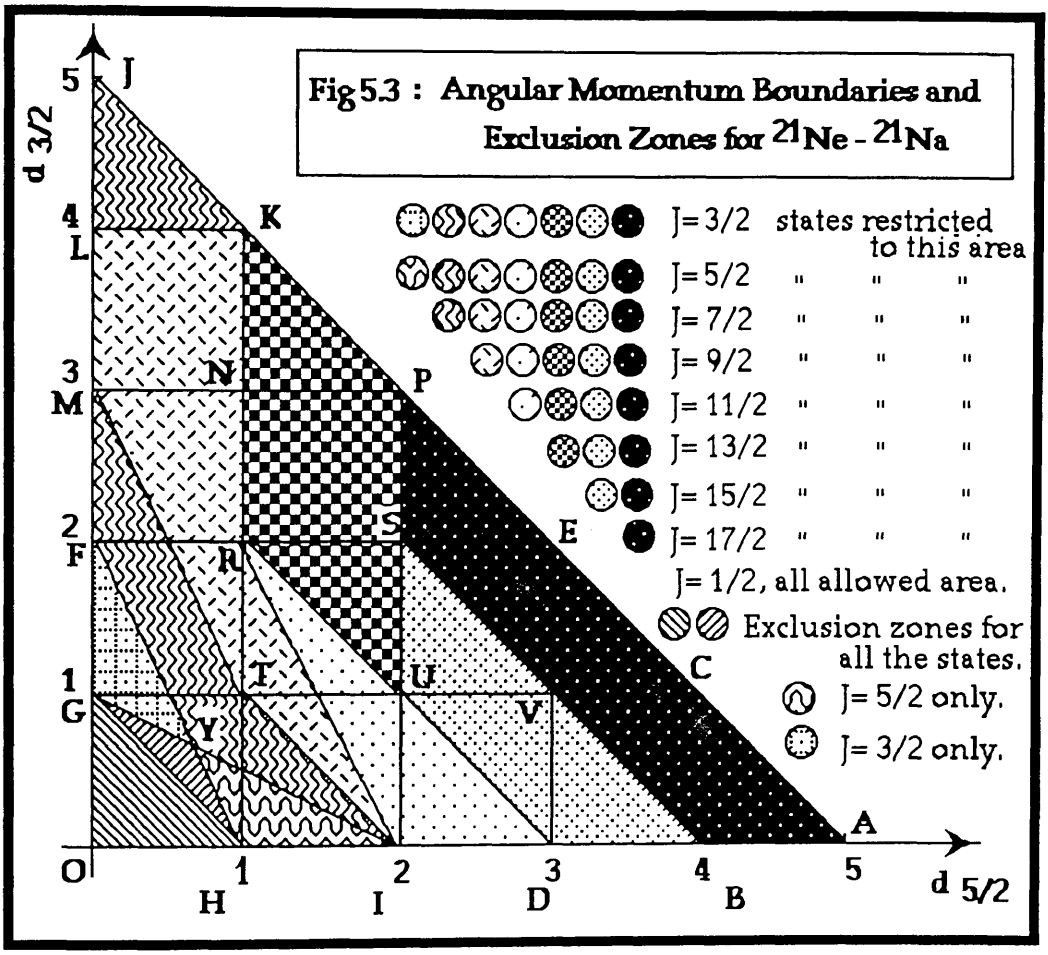


Fig 5.3 : Angular Momentum Boundaries and Exclusion Zones for ^{21}Ne - ^{21}Na

- As an initial framework for predicting the location of states on the diagram, it is helpful to know the configuration of states and the corresponding permitted values of A.M., for different points on the grid. Both these informations have been calculated and are given below.

Points	Configuration	Allowed J-Values
A	(d5/2)^5	(1/2,. . ., 17/2)
B	(d5/2)^4 (s1/2)^1	(1/2,. . ., 17/2)
C	(d5/2)^4 (d3/2)^1	(1/2,. . ., 19/2)
D	(d5/2)^3 (s1/2)^2	(1/2,. . ., 15/2)
E	(d5/2)^3 (d3/2)^2	(1/2,. . ., 19/2)
F	(d3/2)^2 (s1/2)^3	(1/2,. . .,7/2)
G	(d3/2)^1 (s1/2)^4	(3/2) only
H	(d5/2)^1 (s1/2)^4	(5/2)
I	(d5/2)^2 (s1/2)^3	(1/2,. . ., 11/2)
J	(d3/2)^5	(1/2,. . .,7/2)
K	(d5/2)^1 (d3/2)^4	(1/2,. . ., 13/2)
L	(d3/2)^4 (s1/2)^1	(1/2,. . .,9/2)
M	(d3/2)^3 (s1/2)^2	(1/2,. . .,9/2)
N	(d5/2)^1 (d3/2)^3 (s1/2)^1	(1/2,. . ., 13/2)
P	(d5/2)^2 (d3/2)^3	(1/2,. . ., 17/2)
R	(d5/2)^1 (d3/2)^2 (s1/2)^2	(1/2,. . ., 13/2)
S	(d5/2)^2 (d3/2)^2 (s1/2)^1	(1/2,. . ., 17/2)
T	(d5/2)^1 (d3/2)^1 (s1/2)^3	(1/2,. . .,9/2)
V	(d5/2)^3 (d3/2)^1 (s1/2)^1	(1/2,. . ., 17/2)

• With the help of the above information we have marked the areas where eigen states of these nuclei corresponding to certain A.M. can be situated. The special feature of these boundaries is that they grow outward remarkably, symmetrically.

The states with J= 19/2 are permitted to be situated on the line joining the points E and C and all the J= 17/2 states are confined to the area bounded by the points PABS. The area PADU (i.e. PABS + BDUS) represents A.M. boundary for all the J= 15/2 states. An addition of area KPUR to this area i.e. area KADR, gives permitted area for J=13/2 states.Further addition of area DIR to KADR (= KAIR) forms the allowed zone for J= 11/2 states. The points I, M and L have been marked for permitting J= 9/2 states. So an addition of KRITML to KADR (= KAITM) gives permitted boundary for states with J=9/2.

The allowed areas for J= 7/2, 5/2 and 3/2 have found to be more interesting to work out, because of the isolated points G and H, permitted for J= 3/2 only and J=5/2 only, respectively. Points I and F allow states with J values (1/2, . . .,11/2) and (1/2, . . ., 7/2) respectively. This implies that all states ≥ 5/2 must lie to the right of line FH.

Similarly all the J= 3/2 states must lie to above the line GI. This cuts down small areas, FYG, which becomes J= 3/2 states only area and YIH, which becomes J=5/2 states only area. Another small area GYH is cut out which excludes both J= 3/2 as well J= 5/2 states. Area GYH, in fact can be seen to exclusion for all the states apart from GHO, which is the only other exclusion zone for all states because of the constraints on the s1/2 particle numbers (i.e >3).

For J= 7/2, A.M. boundary, the point Y represents states with A.M. values 3/2, 5/2, 7/2 and 11/2. Hence the permitted area for J= 7/2 states will be formed by adding areas MTIYF and JKL to the area for J= 9/2.

Because of the above explanation, we can see that areas FYG for J=3/2 and YIH for J= 5/2 are mutually exclusive. The allowed area for J=5/2 states will be the sum of the area for J= 7/2 and its own exclusive area. Similarly area for J=3/2 will be sum of the area for J= 7/2 and area FYG and YIH. The permitted area for J= 1/2 states will include all allowed areas except exclusive areas FYG and YIH, which are allowed respectively for J= 3/2 and J=5/2 states only.

5.5.5 Comparison of A.M. Boundaries and the calculated state Occupancies

It has always been necessary to test every hypothesis/prediction against the calculated or indeed the experimental results, where ever the later is available. We have already calculated the nucleon occupancies for the nuclei. Separate occupancy diagrams, figs. 5.40-45, have been plotted for these nuclei. It is quite interesting to see how far these occupancy diagrams have conformed to the predicted A.M. boundaries and exclusion zones plotted in figs. 1-3. We undertake this comparison separately for each mirror consisting of two nuclei.

5.5.5.1 Comparison of ^{21}O-^{21}Al Mirror Occupancies.

In case of these nuclei the predicted A.M. boundaries are shown in fig. 5.1 and the diagrams for the actual calculated occupancies are given in figs. 5.40-45. We observe separately for both nuclei,the following features, which show a fairly good agreement between the predicted and calculated location of states in the diagrams.

^{21}O :

- We find by comparison that all states plotted in fig. 5.40 obey the two exclusion zones marked in fig. 5.1.
- It had been predicted in fig.5.1 that only J= 5/2 states can be populated in the triangle CBA. We find that in fig. 5.40 only one J= (5/2)1 state is actual present in this region.

This aspect may, however, be of further investigation, because this state is highly condensate in d5/2, low spin and is low in energy.

- All the 13/2 states in fig.5.40 can be seen to lie in the region MEN predicted in fig.5.1. Similarly 11/2 states lie in the predicted area CGH, 9/2 and 7/2 in the region PHCBDF, as predicted. Also states 3/2 and 5/2 also lie in their expected regions IHCBDG and KHCBGF respectively.

^{21}Al :

In fig. 5.41, we see that states 1_1, 3_1 and 7_1 for ^{21}Al are packed together in a very small area very close to 5_1 state, which is the ground state (ref. table 5.10, figs. 5.7 & 11). Also we find, very close to it on the left, crowding of states 5_2, 3_2 and 9_1. These lie on a line within a maximum distance of 0.25, which is a quite small pro~~m~~porotion of number of particles ~~to~~ in s1/2 shell among these states. This crowding is a strong indication that these states could belong the ground state energy band.

- Another crowding of states is seen near the line representing one particle in the d3/2 shell. These states in their ascending order in the spectrum (ref. table 5.10, figs. 5.7 & 11) are, 7_2, 11_1, 9_3, 11_2, 11_3 and 13_1. Some of these states like 7_2, 9_3 and 11_1 are situated so close to each other that they appear to have a similar structure as well as being member of the same energy band.

- The two highest energy states are located well separated from the rest of the spectrum but not far from each other.
- In the conclusion, we can infer that all the states, which we have calculated, have divided themselves into three different groups, populating three mutually exclusive areas in the diagrams. The three areas lie very close to the lines represented by $(d3/2)^0$, $(d3/2)^1$, $(d5/2)^2$. These three groups of states are strong candidates for, some or all states in a group, belonging to energy bands.

5.5.5.2 Comparison of ^{21}F-^{21}Mg Mirror Occupancies.

The predicted A.M. boundaries for these nuclei has been shown in fig. 5.2, and the actual positioning of states on the grid has been shown in figs. 5.42 and 5.43. The comparison of calculated and predicted location of states in these diagrams has revealed us with the following information.

- All J= 17/2 states do lie in the predicted triangular region ECI, and in fact $(17/2)_3$ is a pure state. A pure state is the one in which the individual particle is not divided between different shells, as in this case there are two particles in the d3/2 and three particles in the d5/2 shells. J=15/2 states lie in area NCBT, J= 13/2 in NADHP, J= 11/2 in RADK, J= 9/2 in RAGX, J= 7/2 states in RAXLS, J= 5/2 states in RAGFS and J= 3/2 states lie in the entire permitted area as predicted in fig. 5.2.
- In the case of ^{21}Mg, we see a cluster of states, 5_1, 9_1, 7_1, 9_2, 13_1 and 13_2. These states are situated very close to the line CB. State 51 is highly condensate in d5/2 shell. Another grouping of states 17_1,15_2,15_3 and 17_2 is located just below the $(d3/2)^1$ line. In the first case the states are so close to each other that their structure appears to very similar. These groups can be further examined for the formation of some energy bands.
- In the case of ^{21}F, the clustering is still manifested but to a lesser degree as compared to ^{21}Mg. The states 15_2 and 17_1. 17_2 and 15_3 have slipped down a bit and 13_3 has been promoted up a bit in d3/2. Similarly 111 and 13_2 states, although still being highly condensate in d5/2 shell, have promoted about 0.1 particle into s1/2 shell. The 17_3 state is a pure state in this case as well.

However, in this case, states 5_1, 1_1, 9_2, 7_1, 13_1 and 11_1 are very close and can be further looked into for being member of some energy band. Also states 5_2, 7_2, 9_3, 11_2 and 15_1 reflect having a band structure.

5.5.5.3 Comparison of ^{21}Ne-^{21}Na Mirror Occupancies.

The predicted A.M. boundaries in this case are shown in fig. 5.3 and the location of calculated states has been depicted in figs. 5.44-45. By comparing the two diagrams we arrive at the following conclusions.

- All calculated states, plotted in figs. 5.44-45, lie precisely within the regions allowed by the A.M. boundaries of fig.5.3.
- Because of the increase in space in this case, most of the states appear to have clustered in one region of figs. 5.44-45. In case of ^{21}Ne, states 3_1, 5_1, 7_1, 11_1, which happens to be the lowest part of the energy spectrum (ref. table 5.12, fig. 5.9, 5.14), lie very close to each other. Also states 11_1, 7_2, 9_2, 13_1, 11_2 and 13_2 lie very close to each other. The three next higher states in the spectrum, 15_1, 17_1 and 11_3 form another cluster in a very small area towards the bottom right of the diagram.

States in these cluster are very close and are strongly suggestive of the presence of special structure.

- In the case of ^{21}Na, we notice some change w.r.t. ^{21}Ne, with regards to the occupancy of some states. This change, however, is towards clustering of states even further. State 3_2 has moved considerably into d5/2 shell and state 1_1 has moved has moved into d5/2 by almost one particle. Representation of high spin states has not changed, perhaps due to the limited number of available configurations. But low spin states have changed in variable proportions.

In spite of the variations, which are due to the change in the wave function of the states, all the states in fig. 5.45 lie entirely within the allowed A.M. boundaries of fig. 5.3.

- States 3_1, 5_1, 7_1, 9_1, 11_1, 9_2, 13_1, 11_2 and 15_1 are situated very close to each other in the diagram and so are likely to have similar structure. These states could also be members of the ground state energy band.

5.6 Single Particle Energy Contours

In the sd shell space, we have seen in the occupancy diagrams that the states tend to be distributed along the lines represented by $(d3/2)^0$, $(d3/2)^1$ and $(d3/2)^2$. One reason for this orientation is the difference in their total single particle energy. This variation between the spe and the occupancy of nucleons is linear and we have plotted these lines in fig. 5.4. These lines are called single particle energy contours. We give below a brief explanation of how we have worked out the coordinates of points, intercepts etc. to plot these contours.

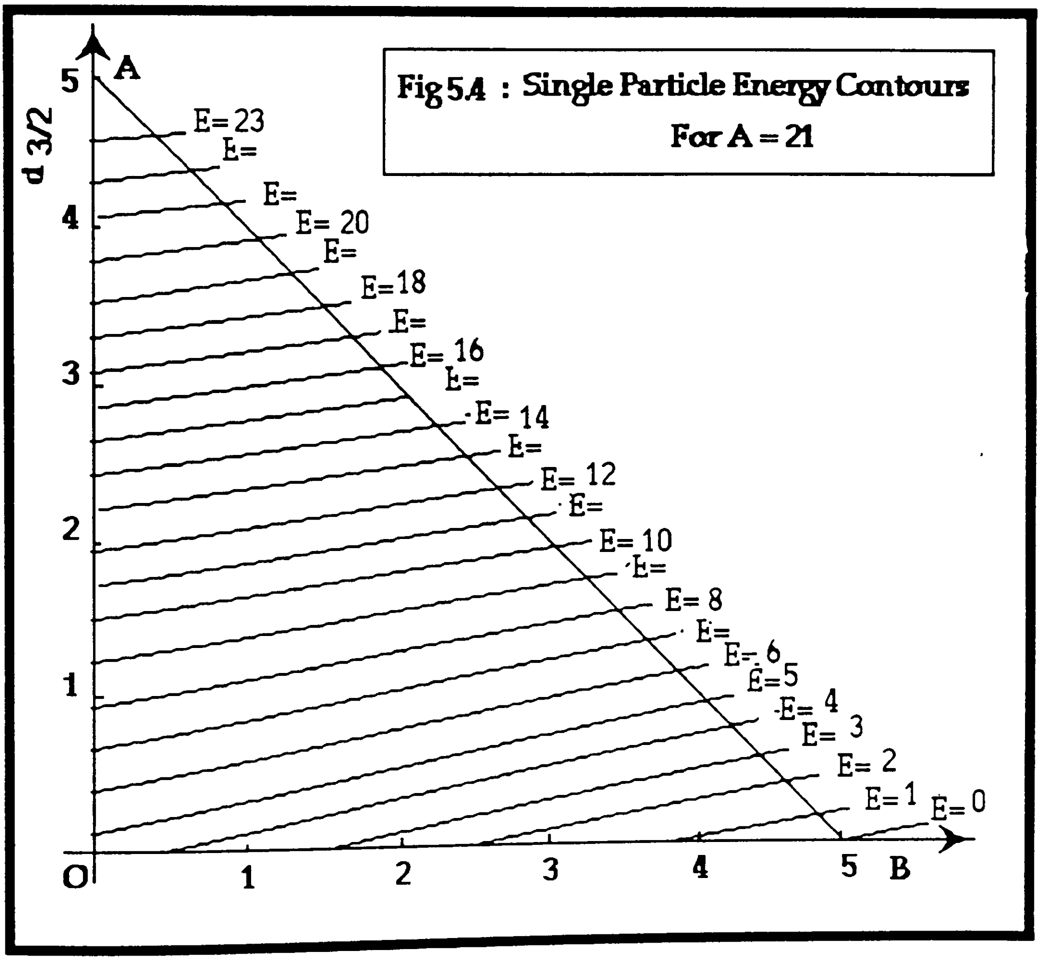


Fig 5.4 : Single Particle Energy Contours For A = 21

Let ε1, ε3, ε5 and n1, n3, n5 denote single particle energies and occupancy values in s1/2, d3/2 and d5/2 shells respectively.

The experimentally determined values for the single particle energies are ε1 = -3.28, ε3= o.93 , ε5= -4.15, which we use in these calculations.

The d5/2 shell being lowes't in energy, is considered to be the ground state, which changes values to $\varepsilon 1 = 0.87$, $\varepsilon 3 = 5.08$, $\varepsilon 5 = 0$.

For mass 21 $n1 + n2 + n3 = 5.$ or $n1 = 5 - (n3 + n5).$

the total single particle energy E, can be given by :

$$E = n1\ \varepsilon 1 + n3\ \varepsilon 3 + n5\ \varepsilon 5$$

$$= n3\ \varepsilon 3 + n5\ \varepsilon 5 + \varepsilon 1\ (5 - n3 - n5)$$

which gives, after substituting actual values :

$$E = 5 \times 0.87 - 0.87 \times n5 + 4.21 \times n3 \qquad \ldots\ldots\ldots\ldots(C)$$

For all the points on the x-axis n3 = 0 and correspondingly

$$n5 = (4.35 - E) / 0.87.$$

As one would otherwise expect, for E= 0, n5 = 5. Similarly we can work out that for E = 1, 2, 3, 4, 5, 6, ..., value for n5 will be 3.85, 2.70, 1.55, 0.40, -0.74, ... respectively.

In the same way we can work out different values of n3 corresponding to respective E values. For doing this we consider the line AB, which is represented by the equation of a line :

$$n5 = 5 - n5\ .$$

Using this equation to eliminate n5 in eq. C, we get :

$$E = 5.08 \times n3 \quad \text{or } n3 = E / 5.08\ .$$

This gives us the value of intercepts made by the single particle energy contours on the d3/2 axis. For E = 0, 1, 2, 3, 4, 5, ..., we we shall have the intercepts equal to 0, 0.2, 0.4, 0.6, 0.8 respectively.

These points have been plotted to give single particle energy contours, which have been shown in fig. 5.4.

5.7 Results and Discussion .

In this section we consider the nuclei individually and try to explain some outstanding features which emerge from the interpretation of these results. It seems appropriate to build upon the conclusions we have already made in section 2.5.5 by making a detailed comparison of occupancy values. Also due to the obvious relationship we shall consider the individually nuclei in the perspective of its mirror pair.

5.7.1 ^{21}O-^{21}Al Mirror Pair.

5.7.1 ^{21}O:

We have noticed in the occupancy diagram fig. 5.1 that all the states have been distributed very close to the lines $(d3/2)^0$, $(d3/2)^1$ and $(d3/2)^2$. The state 5_1 is highly condensate in d5/2 shell, having almost 4.5 particle in this shell. If we refer to fig.5.4, the single particle energy contours, this state is quite close to the point where contribution from spe =0. As the spe increases, we see that corresponding higher energy state continue to spread along these lines to the left until we reach 13_2, which is the highest spin highest energy state in the spectrum.

We explain further, in the following section the distribution of states along these lines, which seems to be quite symmetrical.

- **Explaining the clustering of states along $(d3/2)^0$ line.**

It can be seen from the diagram fig 5.40 that the states 1_1, 7_1 and 9_1 appear to have so much identical structure that they are represented by almost one point.on the occupancy diagram. However, in the energy level diagram fig 5.7, we can see that these states are well separated. Some other states which appear very close to these states in fig 5.40 are 5_3,3_1,5_2 and 3_2. It is now important to have a look at the B(E2) and B(M1) values of these states, to see whether they do exhibit strong electromagnetic interactions. These values have been given in table 5.5.

- **Explaining the clustering of states along (d3/2)1 line.**

We can see in fig5.40 that states 3_3, 7_2, 11_1, and 9_3, lie very close to each other. These are the next higher energy states in the spectrum, fig 5.4 & 10. Occupancy values of these states are nearly equal. This gives a strong indication that these states must have similar structures. The other states which lie along the same line include 11_2, 11_3 and 13_1.

It can be argued that if these states do lie very close to each other physically, then they must exhibit strong electromagnetic interactions. We can check this from their B(E2) and B(M1) values which are given in table 5.5.

Interacting States	B(E2) Values	B(M1) *100 Values	Interacting States	B(E2) Values	B(M1) *100 Values
$5_1 \rightarrow 3_1$	2.73	0.23	$5_1 \rightarrow 1_1$	0.405	
$3_1 \rightarrow 1_1$	1.65	1.53	$3_2 \rightarrow 1_1$		
$7_1 \rightarrow 5_1$	6.75	2.96	$7_1 \rightarrow 3_1$	1.95	
$52 \rightarrow 31$	0.94	30.28			
$5_2 \rightarrow 1_1$	4.28				
$3_3 \rightarrow 1_1$			$9_2 \rightarrow 7_1$	0.11	2.63
$9_2 \rightarrow 5_1$	0.07		$7_2 \rightarrow 5_1$	0.03	0.88
$7_2 \rightarrow 3_1$	1.171		$11_1 \rightarrow 9_1$	1.462	.0004
$11_1 \rightarrow 7_1$	1.95		$9_3 \rightarrow 7_1$	1.24	.07
$9_3 \rightarrow 5_1$	0.61				

table 5.5 B(E2) and B(M1) values for the clustering states of ^{21}O.

- **Explaining the cluster along $(d3/2)^2$ line.**

Only two states 13_2 and 13_1 lie on this line. The first one is a pure state, lower in spe energy to the 13_3, because the later lies on a contour of high spe contour (ref. fig 42). The state 13_3 is the highest energy state in the spectrum. The two states differ almost by 3.55 MeV. Their B(E2) and B(M1) values for these states are given in table 5.19 & 15.

5.7.1.2 ^{21} Al :

This nucleus has got five protons only, outside the core. Its occupancy diagram (fig 5.41) shows that the alignment of states is more pronounced than that of ^{21}O. The states 5_1, 1_1, 3_1, 7_1, 5_2, 3_2, 5_3 and 9_1 one cluster along the line $(d3/2)^0$. The second cluster of states is spread along the line $(d3/2)^1$, which contains 3_3, 7_2, 11_1 and 9_3.The states 11_2, 11_3 and 13_1 are also situated closed by.

Normally one would expect these states to exhibit to exhibit strong electromagnetic interactions. To see this effect we present their

B(E2) and B(M1) values in table 5.6.

Interacting States	B(E2) Values	B(M1) *100 Values	Interacting States	B(E2) Values	B(M1)*100 Values
$5_1 \rightarrow 3_1$	29.40	0.5	$3_1 \rightarrow 1_1$		
$5_1 \rightarrow 1_1$	2.32		$7_2 \rightarrow 5_1$	0.46	1.03
$3_1 \rightarrow 1_1$	17.62	2.40	$72 \rightarrow 31$	8.57	
$7_1 \rightarrow 5_1$	64.48	4.27	$11_1 \rightarrow 9_1$	.02	7.48
$7_1 \rightarrow 3_1$	18.25		$11_1 \rightarrow 7_1$	16.2	
$5_2 \rightarrow 3_1$	9.11	39.88	$9_3 \rightarrow 71$	9.95	.02
$5_2 \rightarrow 1_1$	39.74		$9_3 \rightarrow 5_1$	5.37	
$3_2 \rightarrow 1_1$			$11_2 \rightarrow 9_1$	1.81	12.92
$5_3 \rightarrow 3_1$	7.16	15.44	$11_2 \rightarrow 7_1$	4.01	
$5_3 \rightarrow 1_1$	0.24		$11_3 \rightarrow 9_1$	12.0	17.67
$9_1 \rightarrow 7_1$	17.7	16.40	$11_3 \rightarrow 7_1$	0.21	
$9_1 \rightarrow 5_1$	0.01		$13_1 \rightarrow 1_1$	15.76	0.23
			$13_1 \rightarrow 9_1$	13.19	

table 5.6 B(E2) and B(M1) values for the clustering states of ^{21}Al.

The values shown in the table above signify that electric and magnetic transitions are very strong between the states 5_1, 3_1, 1_1, 7_1 and 5_2.

5.7.1.3 Two Body Interactions :

Before concluding the section on this mirror pair, it is importatnt here to explain why and how the two body interactions affect the occupancy, eigen values and other properties of the nuclear states. In this explaination we can predict if the electromagnetic transition between two states are going to be weak, strong or not possible at all. We consider separartely how these parameters are affected by taking into account the two body interactions.

Without Two Body Interaction:

The fig 5.1 represents the occupancy diagram for this mirror with respect to different A.M. values. If the two body interaction is absent then the points corresponding to particular states(A.M.), will lie exactly on the points as indicated in the discussion that follows the diagram. For instance, J=5/2 state will lie exactly at point A. The corresponding energy spectrum, which is due to the singel paticle energy only, has been shown in fig 5.4.

The B(M1) values between different levels in the same group will be zero, because the exchange of particles in this case must be from s1/2 to d5/2 shell. This implies that ΔJ must be equal to (at least) two, for single particle matrix elements, which is not permitted. So B(E2) values between any two states sitauted along the line OA (fig 5.4) or any line parallel to it, will be zero. Also M1 transitions between the states, in this case, will not be allowed along the vertical line. The reason for this can be explained by the equation

$$\underline{\mu} = \mu_l \, \underline{l} + \mu_s \, \underline{s}$$

where $\underline{\mu}$ is the total magnetic moment operator, μ_l the orbital and μ_s the spin angular momentum operators. A particle in s1/2 shell will have an A.M. value l =0, s =1/2 while particle in d3/2 will have l = 2, s =1/2. we can see that in the case of a transition between these two shells we cannot have $| J_i - J_f | \leq 1$ except when transition is from s1/2 to s1/2. In this case there is no change in occupancy but the state of the nucleus as a whole does change. To summarize, M1 transitions are allowed from d5/2 → d5/2, d3/2 → d5/2, d3/2 → d3/2 and d3/2 → d5/2. This concludes by saying that M1 transitions are allowed allowed between the states parallel to the line AB.

With Two Body Interactions:

Two important phenomenon which give rise to two body (and indeed many body) interactions are the attractive nuclear (short range) force and the repulsive coulomb's force. We explain their collective effect on both these mirror nuclei.

(i) In case of ^{21}Al : Two body interactions will be weak. It is because all the valence particles are protons and the attractive nuclear force between like nucleons is not very strong. Similarly, the repulsive Coulomb's force is also not a very strong force. Consequently, the total force is small and small changes to the single particle picture could be expected. The B(M1) values between different levels in the same group will be small. This should partly explain why the states are beautifully aligned along the lines $(d3/2)^0$, $(d5/2)^1$ and $(d3/2)^2$, in this case.

(ii) in case of ^{21}O : The two body interaction, again, is not very strong. But is stronger than ^{21}Al, because the attractive nuclear force weak and nearly same as before but the repulsive Coulomb's force is absent. This explains why the states in fig 5.40 are slightly scattered than in fig 5.41 for ^{21}Al, indicating slight mixing due to the influence of this force.

5.7.1.4 Conclusions for ^{21}O-^{21}Al Mirror Pair.

- One important observation, which we make here is that the state 9_1 and 9_2 have swapped places in the energy diagrams. fig 5.7 and fig 5.10-11 for ^{21}O and ^{21}Al respectively. This particular behavior has been explained in detail, in section 5.10 on swapping of states.
- Considering the effective charges, we see that B(E2) and B(M1) values for the mirror pair mostly conform to the following conversion. Comparative values have been shown in tables 5.2 and 5.3. The departures from this, however are due to the coulomb's interaction.

B(E2) for ^{21}AL = $(1.5/0.5)^2$ x B(E2) for ^{21}O and

B(M1) for ^{21}AL= $(2.79/1.91)^2$ x B(M1) for ^{21}O

The energy level diagram (Fig.5.7) also reflects a quite symmetric dipping down of the energy levels form ^{21}O to ^{21}AL. The states from 5_1 (ground state) to 9_2 in ^{21}O levels appear compressed and lowered in energy in the mirror nucleus ^{21}Al. Then there is the first energy gap. The energy gap of states, which extends upto 13_1 in ^{21}O also appears symmetrically compressed in ^{21}Al, but the energy difference between the corresponding states is less than that in the case of first group of states. This is followed by another gap where the next state 13_2 in ^{21}O has nearly the same energy as its counterpart in ^{21}Al.

- Real energy gaps : We have further investigated, whether the gaps shown in the energy diagrams are real or they are because of the fact that we have calculated the first few states for every J-value. For this purpose we have calculated all the 119 eigen values which exist in ^{21}Al space.

The ground state being at -5.8571, there are nine states along the (d3/2)0 line (fig 5.41). In the full calculation a real energy gap of 1 Mev has been found between the 9th and 10th states (E= -1.585 & - 0.585), above the ground level. Another real energy gap of ≈ 0.7 Mev has been found between the state 65 (E= 8.308) and state (E= 7.65). No other real gap greater than 0.5 Mev has been found.

5.7.2 ^{21}F-^{21}Mg Mirror Pair :

We discuss both these nuclei, in the light of our calculations, first individually and then as mirror pair.

5.7.2.1 ^{21}F

This nucleus has got 1 proton and 4 neutrons as valence particles. Fig 5.42 shows that the alignment of states along $(d3/2)^0$, $(d3/2)^1$ and $(d3/2)^2$ lines, is still evident but as pronounced as in the case of ^{21}O or ^{21}Al. State 17_3, in this case, is a pure state and is the only one situated on the line $(d3/2)^1$ line. The states 172, 153, 133 and 171 form a group which is situated within a distance of 0.4 particles below the $(d3/2)^1$ line. The rest of states are populated between $(d3/2)^0$ and $(d3/2)^{0.4}$ lines.

This situation also prevails in the energy level diagram figs 5.8, 12. Most of the states are within the first 8.73 Mev, which is the energy level for 15_1. After this there is a gap of ≈ 2 Mev and then the states 17_1, 15_2, 15_3 and 17_2 are present. There is a gap of ≈ 3.5 Mev between 17_2 and 17_3 states, the highest state in the spectrum. The B(E2) and B(M1) values for all these states have been given in table 5.21, 5.27 and figs 5.18, 24..

The two body interactions in case are a bit stronger than the mirror pair discussed earlier in section 5.9.1.3. But still, they are not very strong. As such the contributions to the high spin states are dominated by the single particle energy.

5.7.2.2 ^{21}Mg :

This nucleus has got 4 protons and 1 neutron as valence particles. Because of the increased coulomb's force, we can see in fig 5.43, that the occupancy of some of the states is sharply focussed along the $(d3/2)^1$ line. These states include 17_1, 15_2, 15_3 and 17_2. In the energy spectrum fig 5.8, 13, these states appear grouped together, separated by big energy gaps on either side. State 17_3 is the highest state in the spectrum and is a pure state.

The rest of the states are lie very close to the line (d3/2)0. However among these, states 5_1, 3_1, 9_1 and 7_1 form one cluster and states 9_2, 13_1, 7_2 and 13_2 form another noticeable cluster of states in the occupancy diagram. The B(E2) and B(M1) values for these states have been given in tables 5.22, 28 and also in figs 5.19, 25.

The two body interaction in this case is slightly stronger than in ^{21}F, because of the Coulomb's force. But the attractive force between the nucleons is almost the same. The resultant two body interaction is slightly greater than ^{21}F and is reflected in the comparison of eigen values in fig 5.8 as well as in the occupancy diagrams fig 5.42 and 5.43.

5.7.2.3 Conclusions for ^{21}F-^{21}Mg Mirror Pair :

The state 11_1, promotes 0.25 particles from s1/2 to d3/2 in these two nuclei. In ^{21}Mg, state 13_3 looses 0,25 particles from d3/2 to s1/2 shell, while 15_2 gains same number of particles in the opposite direction. The state 7_2 promotes almost half a particle to d5/2 from s1/2.

The two body interaction is stronger than in case of ^{21}O-^{21}Al pair and causes some mixing of levels. That is why the states are less organised along the pure single particle energy levels.

5.7.3 ^{21}Ne-^{21}Na Mirror Pair :

The results for these nuclei have been discussed below, individually as well as a mirror pair.

5.7.3.1 ^{21}Ne :

This nucleus contains two protons and three neutrons, as the valence particles. Its occupancy diagram fig 5.44 reflects that the regular pattern or alignment or states, which we have witnessed in variable proportions in other two mirror pairs, no longer exists.

This results due to the strong attractive nuclear force (between unlike particles) and the Coulomb's force. Consequently this nucleus is dominated by the strong two body interactions. That is why we do not see energy gaps in the spectrum, fig 5.9. Because of the increased number of protons, the electric transitions are stronger, particularly between the states where $|J_i - J_f| = 1$. This fact has been reflected in the table 5.23 and fig 5.20 which show the B(E2) values for this nucleus, using CWC interaction.

5.7.3.2 ^{21}Na :

This nucleus has 3 protons and two neutrons as valence particles. The occupancy diagram fig. 5.45 for this nucleus is dominated by even stronger two body interactions than ^{21}Ne. The eigen values contain a good contribution from such interactions. The energy spectrum has been compared with ^{21}Ne in fig 5.9. It can be seen that the respective levels do not differ in energy very significantly, because both nuclei are strongly influenced by two body interaction.

5..8 Swapping of States :

We have seen in the previous section that a some points, representing states on the energy level diagram move quite a large distances, when we compare them with the corresponding mirror nucleus. This effect has also been observed in the nuclei when different interactions (CWC & PW) are used for the same mirror pair. We have discovered that some states swap over completely. In this section we test these nuclei to see whether any of their nuclei do exhibit this property. We first give a brief introduction of some parameters, which we have been used during this investigation, then we frame rules for investigation followed by the investigation for each nucleus.

5.8.1 Calculation of Epsilon, Delta, Tau and other Parameters:

In this section we describe some of the parametrs which we have used in our comparisons for finding out swapping states.

5.8.1.1 Epsilon Values:

The epsilon value reflects the percent shift in the energy spectrum, between the same states of two mirror nuclei. We use CWC interactions to fully appreciate the effect the different number of protons in both nuclei. If E1 denotes the energy corresponding to a state of one nucleus and E2 the same value for the same state corresponding to the other nucleus then epsilon is given by :

$$\text{Epsilon} = (E1 - E2) / (E1 + E2) \quad \%$$

We consider the ground state energy of both nuclei to be zero and all E1s and E2s have been updated to this effect. The value of absolute ground energy for each nucleus, is given at the bottom of every table for any likely reference to the actual values of energy for the states.

5.8.1.2 calculation of τe values.

We have studied the changes produced by including/excluding the effect of coulomb's force on the electric and magnetic transition rates. The τe value represents the percent shift, in the spectrum of electric transition rate, between the CWC and PW interaction for the same nucleus and same pair of interacting states. In fact this can be viewed as the percent contribution by the coulomb's interaction in any transition.

The **τe** values for all the nuclei are given in tables 5.16-21.

5.8.1.3 calculation of τm values.

The τm values have exactly the same representation as τe except the difference that τm is for magnetic B(M1) and τe is for electric transition rates.The τm values for all the nuclei are given in tables 22-27.

5.8.1.4 calculation of Delta values.

We have observed that the occupancies of nucleons also changes when we switch between CWC and PW interaction. When any two states appear to have swapped over in any nucleus, which we shall be looking in detail, the occupancies for those states must also have swapped over. This aspect makes it an integral part of our set of parameters required for testing the swapped over states. The delta values are calculated as explained below.

Let n_1, n_3, n_5 and n_1', n_3', n_5' represent the occupancies of nucleons in s1/2, d3/2 and d5/2 shells, for any nucleus using PW and CWC interactions respectively. In the present case $n_1 + n_3 + n_5 = n_1' + n_3' + n_5' = A = 21$.

The vector can be represented by :

$$\zeta = (n_1 - n_1', n_3 - n_3', n_5 - n_5')$$

eliminating n_1 after finding the mod or scaler product of ζ we get :

$$|\delta| = \zeta/A = \mathrm{sqrt}(\,2\,[\ (n_1 - n_1')^2 + (n_3 - n_3')^2 + (n_5 - n_5')^2\,]\,)\ /A$$

$$= \mathrm{sqrt}(\,2\,[\ (n_3 - n_3')^2 + (n_5 - n_5')^2 + (n_3 - n_3')(n_5 - n_5')\ ])\ /A$$

These delta values have been calculated for all the states of each nucleus and presented in tables 5.31-36.

5.8.2 Framework of Investigation:

We shall use the following steps to see if any pair of states have swapped in a certain nucleus.

Epsilon

(i) Pick out states (from tables 5.10-5.12) for which the -values are fairly large. These are the states, for which the energy change is larger between the nuclei of a mirror pair for the same CWC interaction.

(ii) Compare the actual energy values (CWC) for the corresponding mirror nuclei and identify pairs of states, which roughly appear to have swapped their energy values (e.g. ref. fig 5.5)

(iii) Compare B(E2) and B(M1) values for these pairs, for both CWC and PW interactions for the individual mirror nucleus. At this step we actually see if the pairs identified above, show a swap in their electromagnetic properties, and also to which nucleus in the mirror pair the states belong to. Any pair, which filters through at this stage, is considered to be a stronger candidate for the complete swap over of states.

(iv) Compare the occupancy for the states identified in the previous tests. At this stage, we pick out pair(s), which have consisitently shown swapping of values for all the parameters compared so far.

Any pair(s) of state(s), identified at this stage are considered as consisting to two states which swap over completely, with the switching on/off of the Coulomb's interaction.

5.8.3 Swapping of States in ^{21}O-^{21}Al Mirror nuclei :

We refer to table 5.10 and find out that the epsilon values corresponding to states 9_1, 9_2, 7_1, 5_2, 3_1, 3_2 and 1_1 are 8.27, 11.83, 9.63, 10.46, 11.23 and 5.48 respectively. These values are quite a bit higher than the other states, and because of this deviant behavior we make a detailed analysis of these states.

5. 8.3.1 comparison of Epsilon values :

First of all we look into their actual energy values in column D and E of the table 5.10. By doing so we try to locate any of these states which swap their energy values (approximately). A synopsis of the energy values for these pairs of states is given in fig 5.5

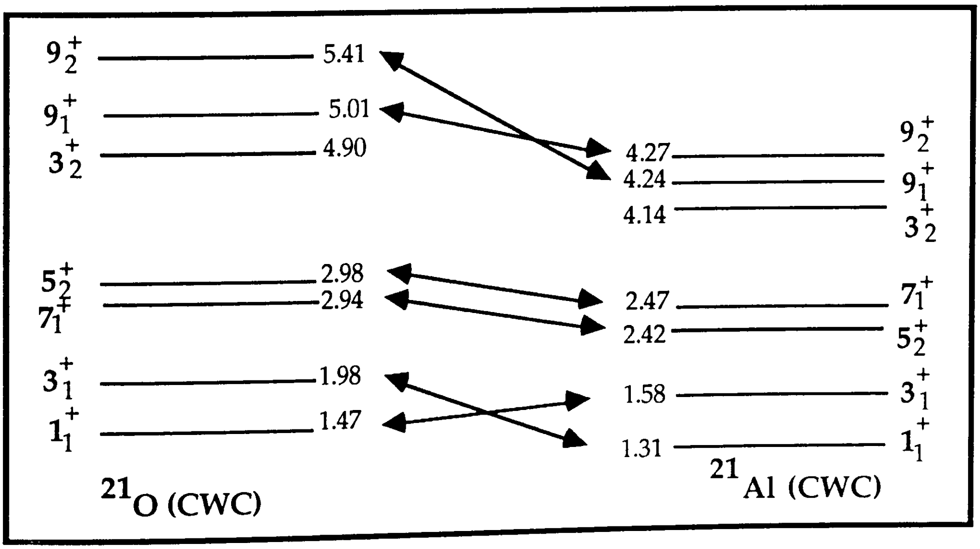


Fig 5. 5 Probable swapping states of ^{21}O and ^{21}Al

. It is reflected in the figure that for some states, energy value of one state in ^{21}O is very close to the energy value of another state in ^{21}Al. This could be considered as first indication that these pairs could be pairs of swapping states. We shall now find out whether these swapping pairs (if they are so), belong to ^{21}O or ^{21}Al. We use comparison of their electromagnetic properties to sort this out.

5.8.3.2 Analysis of Electromagnetic Spectrum for ^{21} O and ^{21}AL :

We confirm our finding of the previous section by comparing the B(E2) and B(M1) values for both nuclei, in turn. The tables 5.7 shows that swapping states do actually belong to ^{21}Al.

States	B(E2)	B(E2)	B(M1)	B(M1)	CONCLUSION
	CWC	PW	CWC	PW	
$9^+_2 > 7^+_1$	6.81	18.43	33.6	12.6	Appear to have swapped over
$9^+_1 > 7^+_1$	17.53	3.30	16.9	37.6	
$7^+_1 > 5^+_1$	65.1	58.0	4.58	3.42	No apparent swap over
$5^+_2 > 3^+_1$	9.17	13.0	38.54	36.52	
$3^+_1 > 1^+_1$	16.6	21.3	2.08	2.71	" "

Table 5.7 Comarison of BE(2) and M1 values for Al ^{21}

The table 5.8 below shows a similar comparison of these parameters for CWC and PW effective interactions with respect to the states identified earlier for ^{21}O.

States	B(E2) CWC	B(E2) PW	B(M1) CWC	B(M1) PW	CONCLUSION
9^{+}_{2}->7^{+}_{1}	2.49	2.05	2.78	8.75	No apparent
9^{+}_{1}->7^{+}_{1}	.11	.37	31.07	16.16	swap over
7^{+}_{1}->5^{+}_{1}	6.83	6.44	3.19	2.38	" "
5^{+}_{2}->3^{+}_{1}	2.91	2.71	29.48	25.41	" "
3^{+}_{1}->1^{+}_{1}	1.97	2.37	1.55	1.89	" "

Table 5.8 Comarison of BE(2) and M1 values for O21

The values in this table show that the B(E2) and B(M1) values for the states 9_1 and 9_2 do not swap over between the two effective interactions for this nucleus.

This gives a strong indication that states 9_1 and 9_2 of ^{21}Al have exchanged their wave functions.

5.8.3.3 Analysis of ^{21}AL states using Nucleon Occupancies :

The final test, for the confirmation of swapping over states is to compare their occupancy values in both the interactions. In this test if the nucleon occupancy of one state in the three shells for PW interaction is found to be equal to the occupancy of other state in the same shells using CWC interaction, we conclude that the wave functions for both states have been swapped. The table 5.9 and fig 5.6 confirm that the states 9_1 and 9_2, have very closely exchanged their occupancies.

The table 5.7 shows that the electromagnetic properties for the states under investigation are markedly closer in case of this nucleus. This indicates that the swapping states belong to this nucleus. To confirm this further we compare the nucleon occupancies for this nucleus. This comparison has been made in table 5.9 and fig 5.6.

States	n3	n3'	n5	n5'
9_1^+	.15	.24	3.01	3.7
9_2^+	.19	.13	3.69	2.99

table 5.9 Comparison of nucleon occupancy for Al21

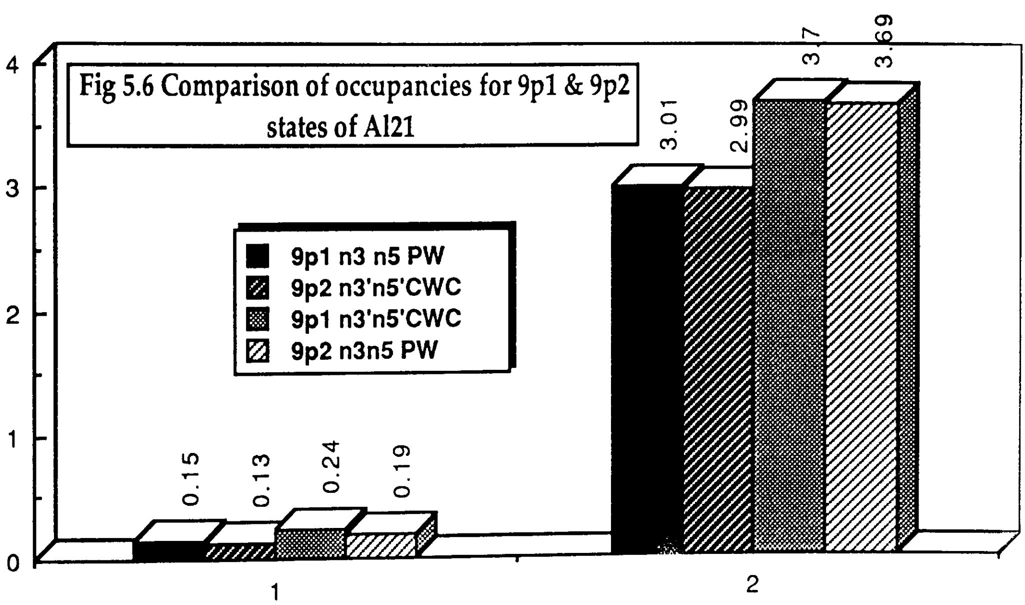


Fig 5.6 Comparison of occupancies for 9p1 & 9p2 states of Al21

It evident that the states under investigation do swap their nucleon occupancies very closely.

So we conclude that 9_1 and 9_2, in ^{21}Al, are the states which strongly indicate that they have exchanged their wave functions.

5.8.4 Swapping States in ^{21}F-^{21}Mg Mirror Pair:

Table 5.11 gives the epsilon values for this mirror pair. Apparently it seems that many states show a strong variation due to the Coulomb's interactions. We have noted these states and followed the steps, mentioned earlier for locating any swapping states. Tables 5.18 and 19 give a comparison for the B(E2) values of these states while tables 5.24 and 25 give their B(M1) values for both the CWC and PW interaction. Our comparison of these parameters for any probable swapping states has concluded that none of the pair of states in this mirror pair all the conditions which we have set out for testing os swapping states.

5.8.5 Swapping States in ^{21}Ne-^{21}Na Mirror Pair:

We have started the investigation into this mirror pair by picking out states with high epsilon values from table 5.12. But it can be seen that there are not many states with this property except 17_2, 11_3, 5_1 and 1_1. One reason for it is that the two nuclei under investigation differ only by one proton and hence Coulomb's component does not contribute substantially. The B(E2) and B(M1) values for these nuclei are compared in tables 5.20, 21 and tables 5.26, 27 respectively. We may find a larger value of τe and τm in case of many states. But this obviously arises from the fact that their actual electromagnetic interaction is very small and the corresponding values of transition rates is very small.

We have applied all the tests laid down in our frame work of investigation for swapping states. But no pair of states, in both these mirror nuclei, has been found to show any sign of exchanging wave functions.

References :

1- R.R. Whitehead, A. Watt, B.J. Cole and I. Morrison, 'Advances in Nuclear Physics', vol. 9, M. Baranger and E. Vogt (eds), publishers : Plenum Publishing (1977).

2- R.W. Keys, Proc. of the IEEE conf. 63(5) 1975 740-767; IBM journal of Res. & Dev., 32(1) (1988) 24-28.

3- O.G.Q. Folberth, Hurden und Grenzen bei der Miniaturisierung digitaler electronik, Elektronische Rechenanlagen, 25(6) (1983) 45-55.

4- C.H. Bennett, R. Landdaur, Scientific American, 7 (1985) 38-46.

5- L.M. Mackenzie, A.M. Macleod, Berry D.J. and Whitehead R.R., Comp. phys. comm. 48 (1988) 229-240.

6- C.A.R. Hoar, 'Communicating sequential processes', Publishers : Prentice-Hall (1985).

7- B.W Kernighan, D.M. Ritchie, 'The C Programming Language', publishers: Prentice-Hall (1988).

8- A. Hayman, 'Charles Babbage, Poineer of the Computer', publishers : Oxford university press (1982) p-242.

9- R.W. Hockney and C.R. Jessope, 'Parallel Computers2': Architecture, programming and Algorithms', publishers: Adam Hilger, Bristol (1988).

10- M.J. Flynn, proc. of IEEE 14, (1966) 1901-1909.

11- W.D. Hillis,'The Connection Machine', Publishers: MIT press, Cambridge (1985).

12- V. Zakharove, IEEE trans. comput. C-33 (1984) 45-78.

13- Inmos Ltd , 'The Occam 2 Reference Manual', publishers: Prentice-Hall International (1988).

14- A.W. Roscoe and C.A.R. Hoare, 'The laws of Occam Programming', Tech. monograph PRG-53, Prog.Res. Group Oxford (1986).

15- K.C. Bowler, R.D. Kenway, G.S. Pawley and D. Roweth, 'An Introduction to Occam 2 Programming', publishers: Chartwell-Bratt, Lund (1987).

16- I.Barron et.al., Electronics Vol.56, no. 23 (1983) 109-115.

17- D.J. Kuck et.al., Proc. of Int. Conf. on Parallel Processing, Aug. (1984).

18- C.D. Polychronopoulos, U. Benerjee, IEEE Transactions on Computers, Vol. C-36, 4 (1987).

19- P.Brench Hansen, 'Operating System Principles', Publishers: Prentice-Hall (1977).

20- E.W. Dijkstra , 'Two Starvation Free Solutions to General Exclusion Principle', EWD 625, Plataan Straat 5, 5671 Al Nuenen, Netherland (1978)

21- A.W. Roscoe and N. Dathi, 'The pursuit of Deadlock freedom', Technical Monograph PRG-57 Prog. Res. Grp., Oxford (1986).

22- K.Hwang and F.A. Briggs, 'Computer Architure and Parallel Processing', McGraw-Hill, New York (1984).

23- K.E. Iverson, ' A Programming Language', John-Wiley, New York (1962).

24- CDC Cyber 200 Model 205 Computer System Hard ware Ref. Manual no. 60256020, Control Data Corporation, Minneapolis, Minnesota, USA.

25- N. Wirth, 'Programmin in MOdula-2, Publishers: Springer Verlag (1980).

26- J.P. Barne 'Programming in Ada', Publishers: ??

27- B.Stroustreep, 'The C++ Programming Language', Addison Wesley Reading (1986).

28- The 3L Limited, Parallel C User Guide (1988).

29- The Meiko Ltd. , The C User and Reference Manual(1989).

30- C. Lanczos, J. Res. Nat. Bur. Stand. (1950) 225.

31- S. Pick and M. Tomsek, Comput. Phys. Commun. 27 (1982) 101.

32- J.H. Wilkinson, 'The algebraic Eigen Value Problem', Publishers: Oxford University Press (1965).

33- A.S Householder, The theory of Matrices in numerical Analysis', Publishers : Blaisdell, New York (1964).

34- C.C. Paige, BIT 10: 183 (1970).

35- C.C. Paige J. Inst. Math. Applic. 10: 373 (1972).

36- B.N. Parlett, The Symmetric Eigen Value Problem', Publishers : Prentice-Hall (1980).

37- D.E. Knuth, 'The Art of Computer Programming', vol 3, ' Sorting and Searching', Publishers Addison and Wesley (1973).

38- N.With, 'Algorithm + Data Structures = Programs', Publishers : Prentice Hall (1976).

39- R. Sedgewick, 'Algorithms', Publishers : Addisonand Wesley (1988).

40- E.W. Dijkstra, 'A Discipline of programming', Publishers : Prentice Hall (1976).

41- M. Riaz and R.R. Whitehead, Comput. Phys. Comm. , preprint.

42- M.G. Mayer, Phys. Rev. 75 (1969).

43- M.G. Mayer, J.D. Jensen, 'Elementary Theory of Nuclear Structure', John Wiley, New York, 1955.

44- O. Haxel, J.D. Jensen and H.E.Sues, Phy. Rev. 75 (1949) 1766.

45- J.H. Barlett, Nature 130 (1932) 165; Phys. Rev. 41 (1932) 370; Phys. Rev. 42 (1932) 145.

46- W.M. Elsasser J. de Phys. et Rad 5 (1933) 549.

47- W.M. Elsasser, J. de Phys. et. Rad 5 (1934) 389; 5 (1934) 635.

48- H.A. Bethe and R.F Bacher, Rev. Mod. Phys. 8 (1936) 82.

49- N. Bohr, Nature 137 (1936) 344.

50- M.G. Mayer, Phys. Rev. 74 (1948) 235.

51- L.W. Nordheim, Phys. Rev. 75 (1949) 1894.

52- E.Feenberg and K.C. Hammack, Phys. Rev. 75 (1949)1877.

53- B.A. Brown, Nucl. Phys. A507 (1990) 25c-42c.

54- B. Zeitnitz (eds), 'Few body problems in Physics', vol1 & vol. 2, Publishers :North-Holland (1983).

55- F.Calogero, A.C. Ciofi, (eds), 'Nuclear Many Body Problem' Proceedings of the International Symposium: Rome; Publishers: Blogna Editric(1973).

56- H.Muther, T. Taigel and T.T.S. Kuo, Nucl. Phys. A482 (1988) 601.

57- Ji Xiangdong, B.H. Wildenthal, Phys. Rev. C37 (1988) 1256.

58- T.T.S. Kuo Nucl. Phys. A103 (1967) 71-96.

59- B.M. Preedom and B.H. Wildenthal, Phys. Rev. C6 (1972) 1633-44.

60- B.J. Cole, A. Watt and R.R. Whitehead, J. Phys. G. Nucl.Phys. vol1: (1975) 303-323; J. Phys. G. Nucl.Phys. vol1:2 (1975) 213-233; Phys. Lett. B 45:5 (1973) ; Phys. Lett. B 49 : 2(1974) 133-136.

61- W. Chung, 'Empirical Renormalization of the shell model Hamiltonian and magnetic dipole moments of shell model nuclei', Ph. D. thesis, Mich. State University, East Lansing (1976).

62- B.H. Wildenthal, Prog. in Part. Nucl. Phys. 11, (1983) 5.

63- B.A. Brown Proc. of the Int. Nucl. Phys. Conf. Ser. no. 86, (1986) 119-137.

64- B.A. Brown and B.H. Wildenthal, Ann. Rev. Nucl. Part. Sci. 38 (1988) 29-66.

65- B.H. Wildenthal, Prog. Part. Nucl. Phys. 11: 5 (1984).

66- B.A. Brown et. al., Ann. Phys., 182 : 191 (1988).

67- Lickert et. al., Z. Phys. A331 (1988) 409.

68- H.Ropke et. al., Z. Phys. A324 :173 (1986).

69- H. Ropke et. al., Z. Phys. A324 :187 (1986).

70- P. Tikkanen et. al. phys. Rev. C36 : 32 (9187).

71- J. Shurpin, T.T. Kuo and D. Strrottman, Nucl. Phys. A408 (1983) 310.

72- E.C. Hallbert, J.B. McGregory, B.H. Wildenthal and S.P. Pandya, Adv. nucl. Phys. 4 (1971) 315.

73- J.B. McGregory and B.H.Wildenthal, Ann. Rev. Nucl. Sci. 30 (1980) 383.

74- W.E. Ormand, 'Isospin - Symmetry Violation in light nuclei', a Ph. D. thesis, Mich. State University, East Lansing (1986).

75- M. Dio, T. Kotani, E. Takasugi, Prog. Theo. Phy. Supp. 83: 1 (1985).

76- B.A. Brown, in 'Nuclear Shell Model', M. Vallieres, B.H. Wildenthal (eds) p. 42, Publishers World Scientific: Singopore (1985).

77- M.C. Etchegoyen, A. Etchegoyen, B.H. Wildenthal, Phys. Rev. C38 : 3 (1988) 1382-1392.

78- M.C. Etchegoyen, A. Etchegoyen, B.H. Wildenthal, Phys. Rev. C39 : (1989) 680-686.

79- B.A. Brown and B.H. Wildenthal, Phys. Rev C28: 2397 (1983).

80- B.H. Wildenthal, proc. of the AIP conf. 135: 89 (1985).

81- B.H. Wildenthal, in 'Weak and Electromagnetic Interaction in Nuclei', H.V. Klapdor (eds), p. 18, Publishers Springer Verlag: Berlin.

82- B.A. Brown and B.H. Wildenthal, At. Data Nucl. Data Tables 33: 347 (1985).

83- B.A. Brown and B.H. Wildenthal,Nucl. Phys. A474: 290 (1987).

84- P.W.M. Glaudmans, in 'Weak and Electromagnetic Interaction in Nuclei', H.V. Klapdor (eds), Publishers Springer Verlag: Berlin.

85- F. Iachello, Nucl. Phys. A507 (1990) 43c-54c.

86- S.G. Nillson, Mat. Fys. Medd. Dan. Vid. Selsk. 29 Nr. 16 (1955).

87- F. Iachello, A. Arima, 'Theory of Interacting Boson Model', Publishers: Cambridge University press, Cambridge (1987); Adv. in Nucl. Phys. 13 (1984) 139;

88- J.P. Elliot Rep. Prog. Phys. 48 (1985) 171.

89- K.W. Schmid, F. Grummer, M.Kyotoku anf Faessler, Nucl. Phys. A452 (1986) 493; Ann. Phys. 190 (1989) 182; Zheng et.al. Nucl. Phys. A494 (1989) 214; E. Hammaren et.al., Nucl. Phys. A454 (1986) 301.

90- S. Cohen and D. Kurath, Nucl. Phys. 73 (9165) 1.

91- J.N. Ginocchio, Ann. Phys. 126 (1980) 234.

92- K.T. Hecht, A. Adler, Nucl.Phys. A137(1969) 129.

93- J.B. McGrory, Symposium of shell model investigations of medium-weight nuclei (University of Tokyo, Tokyo, 1977).

94- S.S.M. Wong and A.P. Zucker, Phys. Lett. B36, 473(1971).

95- H.C. Chiang, M.C. Wang, S.T. Hsieh, Lett. Nuovo Cimento 34(1982) 370; S.T. Hsieh, H.C. Chiang and M.C. Wang, Nucl. Phys. A408 109(1983).

96- A. Chodes, C.B. Thorn, Phys. Rev. D12 (1982) 59; M.V. Barnhill, W.K. Cheng, A. Halprin, Phys. Rev. D20 (1979) 727; G.E. Brown, M.Rho, Phys. Lett. B82 (1979) 177; G.A. Miller, A.W. Thomas, S. The'berge, Phys. Lett. B91 (1980) 192; S. The'berge, G.A. Miller, A.W. Thomas, Can. J. Phys. 60 (1982) 59.

97- M. Bolsterli, J.A. Parmentola, Phys. Rev. D34 (1986) 2112; M. Bolsterli, J.A. Parmentola, in Chiral Solitons edited by K.-F. Liu, publishers: World scienitific, Singapore; M. Bolsterli, J.A. Parmentola, proc. of the topical conf. on nuclear chromodynamics, J. Qiu, D. Sivers (eds), publishers: World Scientific, Singapore 1988 63.

98- R. Goldflan, L.Wilets, Phys. Rev. D25 (1982) 1951; L. Wilets in Chiral Solitons as in ref 57 above.

99- N. Bohr, J.A. Wheeler, Phys. Rev. 56 426 (1939);
A. Bohr, K. Danske Vidensk. Selsk., Mat. _Gys. Meddr. 24 no. 14 (1952). Bohr A. and B.R. Mottleson, K. Danske Vidensk. Selsk., Mat. _Gys. Meddr. 27 no. 16 (1953). K. Adler, A. Bohr, T. Huss, B. Mottleson and A. Winter, Rev. Mod. Phys. 28, 432 (9156).

100- D. Kurath Phys. Rev. 101 (1956) 216.

101- D. Amit A. Katz, Nucl. Phys. 58 (1964) 388.

102- G. Raccah, Bull. Research Council of Isreal 8F (1959) 1.

103- J.P. Elliot and A.M. Lane, Handbuch der Physik. vol. 39, Publishers: Springer Verlag Berlin, 1957.

104- S. Meshkov and C.W. Ufford, Phys. Rev. 101 (1956) 734; S. Goldstein and I. Talmi, Phys. Rev. 102 (1956) 589; S.P. Pandya, Phys. Rev. 103 (1956) 956.

105- E. Rutherdford, Phil. Mag. 21 (1911) 669.

106- A. de-Shalit, I. Talmi, 'Nuclear Structure Theory', Publishers: Academic Press, 1963.

107- P.J. Brussard, P.W.M. Glaudemans, 'Shell model Applications in NUclear Spectroscopy', Publishers: NOrth Holland (1977).

108- A.R. Edmonds, 'Angular Momentum in Quantum Mechanics', Publishers: Princeton University Press (1974).

109- K. Gottfried, 'Quantum Mechanics', Publishers: Benjamin (1969).

110- M.A. Preston, 'Physics of the Nucleus', Publishers: Addison-Wessley (1965).

111- B.A. Brown, B.H. Wildenthal, Nucl. Phys. A474 (1987) 290-306).

112- I.S. Towner, Physics Reports 155 (1987) 263;
A. Arima et. al., in 'Advances in Nucl. Phys.' 18 (1987) 1.

113- B.A. Brown, B.H. Wildenthal, Nucl. Phys. A507 (1990) 25c-42c.

114- I.S. Towner, F.C. Khanna, Nucl. Phys. A399, 334 (1983).

115- Etchegoyen M.C., Etchegoyen A., B.H. Wildenthal, B.A. Brown J. Keinonen, Phys. Rev. C38 3 (1988) 1382-1391.

116- B.A. Brown, A. Arima, J.B. McGrory, Nucl. Phys. A277 (1977) 77.

117- M.Carchidi, B.H. Wildenthal, B.A. Brown, Phys. Rev. C34 (1986) 2280.

118- B.A. Brown et. al. Phys. Rev. C26 (1982) 2247.

119- B.A. Brown, B.H. Wildenthal, Ann. Rev. Nucl. Part. Sci. 38 (1988) 29-66.

120- B.H. Wildenthal and J. Keinonen, unpeblished but quoted in ref 119.

121- A. Hoffman, P. Betz, B.H. Wildenthal, Z. Phys. A. Atomic Nuclei 332 (1989) 289-304.

122- C.M. Lederer, V.S. Shirley, Table of Isotopes, Appendix VII, Publisher: John Wiley (1978).

123- A. Arima, M. Sakkakura, T. Sebe, Nucl. Phys. A170 (1971) 273-308.

124- M.Bini, P.G. Bizzeti_Sona, S. Seuthe, F.B. Waanders and C. Rolf, Proc. of the sixth conf. on Gamma ray Spectroscopy, Belgium 1987.

125- A. B. McDonald, J.R. Patterson and H. Winkler, Nucl. Phys. A137 (1969) 545-560.

126- M. Fernandez, G. Murillo, J. Ramirez et. al. Nucl. Phys. A369 (1981) 425.

127- E. Kuhlmann, A. Hoffmann and W. Albrecht, Z. Physik. A 271 (1974) 49.

128- C. Andritsopoulos, W.N. Catford, E.F. Garman, D.M. Pringle and L.K. Fifield, Nucl. Phys. A372 (1981) 281-300.

129- P.M. Endt and C. Van Der Leun, Nucl. Phys. A310 (1978) 1.

130- M.S. Antony, J. Britz, J. Bueb and A. Pape, Nuo. Cim. 81 : 2 (1984) 414.

131- R.G. Saxtro, R.A. Gough ang J. Cerny, Phys. Rev. C8 1 (1973) 258.

132- E.K. Warburton and D.E. Alburger, Phys. Rev. C23 3 (1981) 1234.

133- D.E. Alburger, C.J. Lister, J.W. Olness and D.J. Millener, Phys. Rev. C23 5 (1981) 2217.

134- W.N. Catford, L.K. Fifield, N.A. Orr and C.L. Woods, Nucl. Phys. A503 (1989) 263-284.

135- P.J. Woods, R.Chapman, J.L. Durell, J.N. Mo et. al., Phys. Lett. B150 1,2,3 (1985) 79.

136- M. Bini, P.G. Bizzeti, A.M. Bizzeti_Sona et. al., Phys. Lett. B231 4 (1989) 346.

```
[illegible] 4 0 0.17850  3.49999 3.49997 3.49997 0.0
3 5 2 6 0 -3.66040  3.49998 3.49997 3.49997 0.0
[illegible] 4 0 1.46740  0.0     0.0     0.0     0.0
    2 6 0 -1.17030  0.0     0.0     0.0     0.0
    2 4 0 2.61180  0.0     0.0     0.0     0.0
        0 0.02720  0.0     0.0     0.0     0.0
    2 2 0 -1.36920  3.49999 3.49996 3.49998 0.0
    2 4 0 -4.09220  3.49998 3.49997 3.49997 0.0
    2 6 0 -0.61270  3.49998 3.49997 3.49997 0.0
      3 0 -4.535 [illegible] 3.49997 3.49997 [illegible]
    2 2 0 -1.77410 [illegible]       0.0     0.0
    2 2 0 1.93780  0.0     0.0     0.0     0.0
      4 0 -1.55690  0.0     0.0     0.0     0.0
2 5 2 2 0 -0.03680  0.0     0.0     0.0     0.0
      3 0 -1.185 [illegible] 0.0     0.0     0.0
    2 2 0 -3.42270  3.50000 3.50000 3.50000 [illegible]
  2 2 0 0.31250  0.0     0.0     0.0     0.0
3 3 2 2 0 -0.21540  0.0     0.0     0.0     0.0
[illegible] 0 -0.78410  3.49998 3.49998 3.49997 0.0
```

Appendix A

```
5 2 5 2 5 2 5 2 2  0 -0.94370  3.49998  3.49998  3.49998  0.0
5 2 5 2 5 2 5 2 6  0 -1.77880  3.49998  3.49997  3.49997  0.0
5 2 5 2 5 2 5 2 10 0 -4.02320  3.50001  3.49997  3.49997  0.0
1 2 5 2 5 2 5 2 6  0 -1.18650  0.0      0.0      0.0      0.0
3 2 5 2 5 2 5 2 2  0-3.20560  0.00001  0.0      0.0      0.0
3 2 5 2 5 2 5 2 6  0-1.89860  0.0      0.0      0.00001  0.0
1 2 1 2 5 2 5 2 2  0-0.42410  0.00001  0.0      0.0      0.0
1 2 3 2 5 2 5 2 2  0 -0.23990  0.0     0.0      0.0      0.0
3 2 3 2 5 2 5 2 2  0 1.64170 -0.00001  0.0      0.0      0.0
3 2 3 2 5 2 5 2 6  0 0.50600  0.0      0.0      0.0      0.0
1 2 5 2 1 2 5 2 4  0  0.17660  3.49998  3.49997  3.49997  0.0
1 2 5 2 1 2 5 2 6  0 -3.66030  3.49998  3.49997  3.49997  0.0
1 2 5 2 3 2 5 2 4  0 1.46740  0.0      0.0      0.0      0.0
1 2 5 2 3 2 5 2 6  0 -1.17090  0.0     0.0      0.0      0.0
1 2 3 2 1 2 5 2 4  0 2.61180  0.0      0.0      0.0      0.0
1 2 5 2 3 2 3 2 6  0 0.02720  0.0      0.0      0.0      0.0
3 2 5 2 3 2 5 2 2  0 -5.36920  3.49999  3.49998  3.49998  0.0
3 2 5 2 3 2 5 2 4  0 -4.05200  3.49998  3.49997  3.49997  0.0
3 2 5 2 3 2 5 2 6  0 -0.61270  3.49998  3.49997  3.49997  0.0
3 2 5 2 3 2 5 2 8  0  -3.83590  3.50001  3.49997  3.49997  0.0
1 2 1 2 3 2 5 2 2  0 -1.73450  0.00001  0.0      0.0      0.0
1 2 3 2 3 2 5 2 2  0 1.93780  0.0      0.0      0.0      0.0
1 2 3 2 3 2 5 2 4  0  -1.56020  0.0     0.0      0.0      0.0
3 2 3 2 3 2 5 2 2  0  -0.03880  0.0     0.0      0.0      0.0
3 2 3 2 3 2 5 2 6  0 -2.18560  0.00001  0.0      0.0      0.0
1 2 1 2 1 2 1 2 2  0 -3.42270  3.50001  3.50000  3.50000  0.0
1 2 1 2 1 2 3 2 2  0  0.31250  0.0      0.0      0.0      0.0
1 2 1 2 3 2 3 2 2  0 -0.21540  0.0      0.0      0.0      0.0
1 2 3 2 1 2 3 2 2  0 -2.78610  3.49998  3.49998  3.49997  0.0
1 2 3 2 1 2 3 2 4  0 -1.09740  3.49997  3.49996  3.49996  0.0
1 2 3 2 3 2 3 2 2   0  0.80970  0.0     0.0      0.0      0.0
3 2 3 2 3 2 3 2 2  0  0.05760  3.49998  3.49998  3.49998  0.0
3 2 3 2 3 2 3 2 6  0  -2.08730  3.50000  3.49997  3.49997  0.0
```

```
5 2 5 2 5 2 5 2 0 2 -2.12430  3.49999  3.49998  3.49998  0.0
5 2 5 2 5 2 5 2 4 2 -1.23120  3.49998  3.49998  3.49998  0.
5 2 5 2 5 2 5 2 8 2  0.16110  3.49998  3.49997  3.49997  0.
1 2 5 2 5 2 5 2 4 2 -0.65940  0.0      0.0      0.0      0.
3 2 5 2 5 2 5 2 4 2  0.40200  0.0      0.0      0.0      0.
3 2 5 2 5 2 5 2 8 2  1.38010 -0.00001  0.0      0.0     -0.
1 2 1 2 5 2 5 2 0 2 -1.40580  0.00001  0.0      0.0      0.
1 2 3 2 5 2 5 2 4 2 -0.84710  0.0      0.0      0.0      0.
3 2 3 2 5 2 5 2 0 2 -3.83670  0.00001  0.0      0.0      0.
3 2 3 2 5 2 5 2 4 2 -0.91490  0.0      0.0      0.0      0.
1 2 5 2 1 2 5 2 4 2 -0.84950  3.49998  3.49997  3.49997  0.0
1 2 5 2 1 2 5 2 6 2  0.78380  3.49997  3.49996  3.49996  0.0
1 2 5 2 3 2 5 2 4 2  0.22090  0.0      0.0      0.0     -0.0
1 2 5 2 3 2 5 2 6 2  0.09030  0.0      0.0      0.0      0.0
1 2 3 2 1 2 5 2 4 2 -1.57100  0.00001  0.00001  0.00001  0.0
1 2 5 2 3 2 3 2 4 2 -0.75310  0.0      0.0      0.0      0.0
3 2 5 2 3 2 5 2 2 2  0.40580  3.49997  3.49998  3.49998  0.0
3 2 5 2 3 2 5 2 4 2  0.32680  3.49997  3.49997  3.49997  0.
3 2 5 2 3 2 5 2 6 2  0.66640  3.49998  3.49996  3.49996  0.0
3 2 5 2 3 2 5 2 8 2 -1.14850  3.50000  3.49997  3.49997  0.
1 2 3 2 3 2 5 2 2 2 -0.09890  0.0      0.0      0.0      0.0
1 2 3 2 3 2 5 2 4 2  0.77960  0.0      0.0      0.0     -0.0
3 2 3 2 3 2 5 2 4 2  1.02300  0.0      0.0      0.0     -0.0
1 2 1 2 1 2 1 2 0 2 -2.26430  3.50001  3.49999  3.49999  0.
1 2 1 2 3 2 3 2 0 2 -0.75430  0.00001  0.0      0.0      0.0
1 2 3 2 1 2 3 2 2 2  0.75250  3.49997  3.49997  3.49997  0.0
1 2 3 2 1 2 3 2 4 2  0.20220  3.49997  3.49996  3.49996  0.
1 2 3 2 3 2 3 2 4 2 -0.20970  0.0      0.0      0.0      0.0
3 2 3 2 3 2 3 2 0 2 -0.28490  3.49998  3.49998  3.49998  0.
3 2 3 2 3 2 3 2 4 2  0.61100  3.49998  3.49997  3.49997  0.
```

5	2	5	2	5	2	5	2	10	0	-4.0307	0.0	0.0	0.0	0.0
5	2	5	2	5	2	5	2	8	2	0.0208	0.0	0.0	0.0	0.2958
5	2	5	2	5	2	5	2	6	0	-1.3434	0.0	0.0	0.0	0.0
5	2	5	2	5	2	5	2	4	2	-1.0399	0.0	0.0	0.0	0.5349
5	2	5	2	5	2	5	2	2	0	-0.8660	0.0	0.0	0.0	0.0
5	2	5	2	5	2	5	2	0	2	-2.0094	0.0	0.0	0.0	0.7769
3	2	5	2	5	2	5	2	8	2	1.3293	0.0	0.0	0.0	-0.0074
3	2	5	2	5	2	5	2	6	0	-1.9409	0.0	0.0	0.0	0.0
3	2	5	2	5	2	5	2	4	2	0.4781	0.0	0.0	0.0	-0.0236
3	2	5	2	5	2	5	2	2	0	-3.3882	0.0	0.0	0.0	0.0
3	2	5	2	3	2	5	2	8	2	-1.1426	0.0	0.0	0.0	0.4066
3	2	5	2	3	2	5	2	8	0	-3.4056	0.0	0.0	0.0	0.0
3	2	5	2	3	2	5	2	6	2	0.5476	0.0	0.0	0.0	0.3553
3	2	5	2	3	2	5	2	6	0	-0.5305	0.0	0.0	0.0	0.0
3	2	5	2	3	2	5	2	4	2	0.6659	0.0	0.0	0.0	0.3899
3	2	5	2	3	2	5	2	4	0	-3.7876	0.0	0.0	0.0	0.0
3	2	5	2	3	2	5	2	2	2	0.5267	0.0	0.0	0.0	0.4102
3	2	5	2	3	2	5	2	2	0	-5.5217	0.0	0.0	0.0	0.0
3	2	3	2	5	2	5	2	6	0	0.8725	0.0	0.0	0.0	0.0
3	2	3	2	5	2	5	2	4	2	-1.2345	0.0	0.0	0.0	0.0065
3	2	3	2	5	2	5	2	2	0	1.7200	0.0	0.0	0.0	0.0
3	2	3	2	5	2	5	2	0	2	-3.8935	0.0	0.0	0.0	0.0716
3	2	3	2	3	2	5	2	6	0	-2.0286	0.0	0.0	0.0	0.0
3	2	3	2	3	2	5	2	4	2	0.8424	0.0	0.0	0.0	-0.0171
3	2	3	2	3	2	5	2	2	0	-0.1337	0.0	0.0	0.0	0.0
3	2	3	2	3	2	3	2	6	0	-2.6098	0.0	0.0	0.0	0.0
3	2	3	2	3	2	3	2	4	2	0.1747	0.0	0.0	0.0	0.5258
3	2	3	2	3	2	3	2	2	0	-0.4708	0.0	0.0	0.0	0.0
3	2	3	2	3	2	3	2	0	2	-0.8119	0.0	0.0	0.0	0.6391
1	2	5	2	5	2	5	2	6	0	-1.3830	0.0	0.0	0.0	0.0
1	2	5	2	5	2	5	2	4	2	-0.6176	0.0	0.0	0.0	0.0443
1	2	5	2	3	2	5	2	6	2	0.3350	0.0	0.0	0.0	0.0193
1	2	5	2	3	2	5	2	6	0	-1.2093	0.0	0.0	0.0	0.0

1	2	5	2	3	2	5	2	4	2	-0.2028	0.0	0.0	0.0	-0.0341
1	2	5	2	3	2	5	2	4	0	1.0366	0.0	0.0	0.0	0.0
1	2	5	2	3	2	3	2	6	0	0.0502	0.0	0.0	0.0	0.0
1	2	5	2	3	2	3	2	4	2	-0.9668	0.0	0.0	0.0	0.0106
1	2	5	2	1	2	5	2	6	2	0.7762	0.0	0.0	0.0	0.2404
1	2	5	2	1	2	5	2	6	0	-3.5513	0.0	0.0	0.0	0.0
1	2	5	2	1	2	5	2	4	2	-0.8184	0.0	0.0	0.0	0.4571
1	2	5	2	1	2	5	2	4	0	0.0660	0.0	0.0	0.0	0.0
1	2	3	2	5	2	5	2	4	2	-0.9602	0.0	0.0	0.0	-0.0022
1	2	3	2	5	2	5	2	2	0	-0.4242	0.0	0.0	0.0	0.0
1	2	3	2	3	2	5	2	4	2	0.2836	0.0	0.0	0.0	-0.0279
1	2	3	2	3	2	5	2	4	0	-1.3218	0.0	0.0	0.0	0.0
1	2	3	2	3	2	5	2	2	2	-0.1106	0.0	0.0	0.0	0.0052
1	2	3	2	3	2	5	2	2	0	1.6277	0.0	0.0	0.0	0.0
1	2	3	2	3	2	3	2	4	2	-0.1856	0.0	0.0	0.0	0.0169
1	2	3	2	3	2	3	2	2	0	0.7599	0.0	0.0	0.0	0.0
1	2	3	2	1	2	5	2	4	0	2.4571	0.0	0.0	0.0	0.0
1	2	3	2	1	2	5	2	4	2	-1.6881	0.0	0.0	0.0	0.0416
1	2	3	2	1	2	3	2	4	2	-0.1653	0.0	0.0	0.0	0.3841
1	2	3	2	1	2	3	2	4	0	-1.3414	0.0	0.0	0.0	0.0
1	2	3	2	1	2	3	2	2	2	0.2732	0.0	0.0	0.0	0.3676
1	2	3	2	1	2	3	2	2	0	-3.0871	0.0	0.0	0.0	0.0
1	2	1	2	5	2	5	2	2	0	-0.6255	0.0	0.0	0.0	0.0
1	2	1	2	5	2	5	2	0	2	-1.3225	0.0	0.0	0.0	-0.0048
1	2	1	2	3	2	5	2	2	0	-1.7223	0.0	0.0	0.0	0.0
1	2	1	2	3	2	3	2	2	0	-0.2569	0.0	0.0	0.0	0.0
1	2	1	2	3	2	3	2	0	2	-0.8385	0.0	0.0	0.0	0.0271
1	2	1	2	1	2	3	2	2	0	0.2719	0.0	0.0	0.0	0.0
1	2	1	2	1	2	1	2	2	0	-3.3275	0.0	0.0	0.0	0.0
1	2	1	2	1	2	1	2	0	2	-2.3068	0.0	0.0	0.0	0.4972

Appendix C

```
5 2 5 2 5 2 5 2 0 2 -2.8197  0.0  0.0  0.0  0.0
5 2 5 2 5 2 5 2 2 0 -1.6321  0.0  0.0  0.0  0.0
5 2 5 2 5 2 5 2 4 2 -1.0020  0.0  0.0  0.0  0.0
5 2 5 2 5 2 5 2 6 0 -1.5012  0.0  0.0  0.0  0.0
5 2 5 2 5 2 5 2 8 2 -0.1641  0.0  0.0  0.0  0.0
5 2 5 2 5 2 5 2 10 0 -4.2256  0.0  0.0  0.0  0.0
1 2 5 2 5 2 5 2 4 2 -0.8616  0.0  0.0  0.0  0.0
1 2 5 2 5 2 5 2 6 0 -1.2420  0.0  0.0  0.0  0.0
3 2 5 2 5 2 5 2 2 0 -2.5435  0.0  0.0  0.0  0.0
3 2 5 2 5 2 5 2 4 2 0.2828  0.0  0.0  0.0  0.0
3 2 5 2 5 2 5 2 6 0 -2.2216  0.0  0.0  0.0  0.0
3 2 5 2 5 2 5 2 8 2 1.2363  0.0  0.0  0.0  0.0
1 2 1 2 5 2 5 2 0 2 -1.3247  0.0  0.0  0.0  0.0
1 2 1 2 5 2 5 2 2 0 -1.1756  0.0  0.0  0.0  0.0
1 2 3 2 5 2 5 2 2 0 -1.1026  0.0  0.0  0.0  0.0
1 2 3 2 5 2 5 2 4 2 -0.6198  0.0  0.0  0.0  0.0
3 2 3 2 5 2 5 2 0 2 -3.1856  0.0  0.0  0.0  0.0
3 2 3 2 5 2 5 2 2 0 0.7221  0.0  0.0  0.0  0.0
3 2 3 2 5 2 5 2 4 2 -1.6221  0.0  0.0  0.0  0.0
3 2 3 2 5 2 5 2 6 0 1.8949  0.0  0.0  0.0  0.0
1 2 5 2 1 2 5 2 4 0 -1.4474  0.0  0.0  0.0  0.0
1 2 5 2 1 2 5 2 4 2 -0.8183  0.0  0.0  0.0  0.0
1 2 5 2 1 2 5 2 6 0 -3.8598  0.0  0.0  0.0  0.0
1 2 5 2 1 2 5 2 6 2  0.7626  0.0  0.0  0.0  0.0
1 2 5 2 3 2 5 2 4 0  0.0968  0.0  0.0  0.0  0.0
1 2 5 2 3 2 5 2 4 2  0.4770  0.0  0.0  0.0  0.0
1 2 5 2 3 2 5 2 6 0 -1.2032  0.0  0.0  0.0  0.0
1 2 5 2 3 2 5 2 6 2 0.6741  0.0  0.0  0.0  0.0
1 2 3 2 1 2 5 2 4 0  2.0664  0.0  0.0  0.0  0.0
1 2 3 2 1 2 5 2 4 2 -1.9410  0.0  0.0  0.0  0.0
1 2 5 2 3 2 3 2 4 2 -0.4041  0.0  0.0  0.0  0.0
1 2 5 2 3 2 3 2 6 0 0.1887  0.0  0.0  0.0  0.0
```

```
3 2 5 2 3 2 5 2 2 0 -6.5058   0.0   0.0   0.0   0.0
3 2 5 2 3 2 5 2 2 2 1.0334    0.0   0.0   0.0   0.0
3 2 5 2 3 2 5 2 4 0 -3.8253   0.0   0.0   0.0   0.0
3 2 5 2 3 2 5 2 4 2 -0.3248   0.0   0.0   0.0   0.0
3 2 5 2 3 2 5 2 6 0 -0.5377   0.0   0.0   0.0   0.0
3 2 5 2 3 2 5 2 6 2 0.5894    0.0   0.0   0.0   0.0
3 2 5 2 3 2 5 2 8 0 -4.5062   0.0   0.0   0.0   0.0
3 2 5 2 3 2 5 2 8 2 -1.4497   0.0   0.0   0.0   0.0
1 2 1 2 3 2 5 2 2 0 -2.1042   0.0   0.0   0.0   0.0
1 2 3 2 3 2 5 2 2 0 1.7080    0.0   0.0   0.0   0.0
1 2 3 2 3 2 5 2 2 2 0.1874    0.0   0.0   0.0   0.0
1 2 3 2 3 2 5 2 4 0  0.2832   0.0   0.0   0.0   0.0
1 2 3 2 3 2 5 2 4 2  0.5247   0.0   0.0   0.0   0.0
3 2 3 2 3 2 5 2 2 0 -0.5647   0.0   0.0   0.0   0.0
3 2 3 2 3 2 5 2 4 2  0.6149   0.0   0.0   0.0   0.0
3 2 3 2 3 2 5 2 6 0 -2.0337   0.0   0.0   0.0   0.0
1 2 1 2 1 2 1 2 0 2 -2.1246   0.0   0.0   0.0   0.0
1 2 1 2 1 2 1 2 2 0 -3.2628   0.0   0.0   0.0   0.0
1 2 1 2 1 2 3 2 2 0 1.2501    0.0   0.0   0.0   0.0
1 2 1 2 3 2 3 2 0 2 -1.0835   0.0   0.0   0.0   0.0
1 2 1 2 3 2 3 2 2 0  0.0275   0.0   0.0   0.0   0.0
1 2 3 2 1 2 3 2 2 0 -4.2930   0.0   0.0   0.0   0.0
1 2 3 2 1 2 3 2 2 2  0.6066   0.0   0.0   0.0   0.0
1 2 3 2 1 2 3 2 4 0 -1.8194   0.0   0.0   0.0   0.0
1 2 3 2 1 2 3 2 4 2 -0.4064   0.0   0.0   0.0   0.0
1 2 3 2 3 2 3 2 2 0  0.3983   0.0   0.0   0.0   0.0
1 2 3 2 3 2 3 2 4 2 -0.5154   0.0   0.0   0.0   0.0
3 2 3 2 3 2 3 2 0 2 -2.1845   0.0   0.0   0.0   0.0
3 2 3 2 3 2 3 2 2 0 -1.4151   0.0   0.0   0.0   0.0
3 2 3 2 3 2 3 2 4 2 -0.0665   0.0   0.0   0.0   0.0
3 2 3 2 3 2 3 2 6 0 -2.8842   0.0   0.0   0.0   0.0
```

Appendix D

The transition probability given by eq (13-16)[1] is :

$$T(E2) = \frac{4\pi}{75} \text{ x } \frac{1}{h} \text{ x } k^5 \text{ x } (B(E2)) \qquad \text{------------------------------(a)}$$

Eqs. (13-2a) and (13-5a)* give :

$$T(E2) = \frac{2(2+1)}{2[(4+1)\,!!]^2} \text{ x } \frac{e^2}{h\,c} \text{ x } (\frac{\omega \text{ x } R}{c})^4 \text{ x } \omega \text{ x } R_\lambda \qquad \text{-------------------------(b)}$$

$$= \frac{4\pi}{75} \text{ x } \frac{1}{h} \text{ x } k^5 \text{ x } \{ \frac{1}{4\pi} \text{ x } e^2 \text{ x } R^4 \text{ x } R_\lambda \qquad e^2 x fm^4$$

Comparing it with (a) we get :

$$B(E2) = 1 \text{ W.U.} = \{ \frac{1}{4\pi} \text{ x } e^2 \text{ x } R^4 \text{ x } R_\lambda \} \qquad e^2 x fm^4$$

Substituting the given value for thr radial integral and $R = 1.2 \text{ x } A^{1/3} = 3.31$ we get :

$$1 \text{ W.U.} = \frac{1}{4\pi} \text{ x } (\, 1.2 x\, 21^{1/3}\ )^4 \text{ x } (\frac{3}{5})^2 \qquad e^2 x fm^4$$

$$= 3.4374$$

While the value suggested for B(E2) in the paper is 1 W.U. = 5.00.
The value calculated above converts my data to conform closely to that reported in the literature [2]. Similarly a short derivation of the W.U. for B(M1) is given below: Referring to eq(13-5c) [1] gives :

$$T_w\ (\mu\,\lambda\,) = 10.(\frac{3}{4})^2.\ [\,\mu_p\,\lambda - \frac{1}{2}]^{-2} \text{ x } \{ \frac{4}{9} \text{ x } \frac{3}{3}^2 \text{ x } \frac{e^2}{h\,c} \text{ x } (\frac{h}{R\,M\,C})^2 \text{ x}$$

$$[\mu_p\,\lambda - \frac{1}{2}]^2 \text{ x}(\frac{\omega R}{c})^2 \text{ x } \omega \,\} \text{ sec}^{-1}$$

$$= \frac{10}{16} \text{ x } 4 \text{ x } 4 \text{ x } (\frac{e\,h}{2MC})^2 \text{ x } \frac{1}{h} \text{ x } (\frac{\omega}{C})^3 = 10 \text{ x } \frac{k^3}{h} \text{ x } (\frac{e\,h}{2MC})^2$$

$$\text{But } T = \frac{8\pi \text{ x } 2}{9} \text{ x } \frac{k^3}{h} \text{ x } B(M1) = \frac{16\pi}{9} \text{ x } \frac{k^3}{h} \text{ x } B(M1)$$

$$\text{Hence } B(M1) = \frac{90}{16\pi} \text{ x } (\frac{e\,h}{2MC})^2 = \frac{45}{8\pi} \text{ x } \mu_0^2 \qquad n \text{ x } m^2$$

$$= 1.7904 \text{ x } \mu_0^2 \qquad n \text{ x } m^2 \quad = 1.7904 \text{ x } 0.01106 = 0.019802 \ \ e^2 \text{ x fm}$$

i.e Numerical value of B(M1) in W.U. = $\frac{1}{0.019802}$ x numerical value in e^2 fm

12

[1]- Preston M.A., Physics of the Nucleus, Addson - Wesley 1962.

[2]- Andritsopoulos G., Catford W.N., et. al. Nucl. Phys. A372 (1981) 281-300.

Appendix E

A New Binary Representation for Slater Determinants

M.Riaz and R.R.Whitehead
Department Of Physics and Astronomy,
The University, Glasgow, G12 8QQ, U.K.

Abstract:

The manipulation of Slater Determinants in the computer (Whitehead et.al., 1977) is generally handled by means of the binary representation, in which each of a set of bits represents the presence or absence of a particular occupied single particle orbital. Alternatively, the more compact, but more complicated, combinatorial representation may be used. Here we describe a new representation which has certain advantages in large scale calculations.

The usual binary representation of many-particle Slater Determinants consists of assigning definite single particle orbits to the bits of a bit field (like a computer word). So, for example, the pattern 011001 indicates, reading from left to right, that the second, third and sixth orbits are occupied. The action of creation and destruction operators is imitated by "atomic" logical (bit manipulation) operators in combination with appropriate algorithms. The creation of a particle in orbit five could be achieved as follows.

```
if ( OSD AND 000010 ) result = NULL
else
        result = OSD OR 000010                    (1)
        sgn = sgn ( OSD, 000010 )
```

where OSD is a Slater Determinant in the occupancy representation. The first line takes care of the possibility that the orbit is already occupied. The logical operations are performed in one machine operation and the time taken is usually independent of the values of the operands. The last line symbolically indicates the necessity of determining the sign, or phase, of the result. This phase follows from the commutation relations obeyed by creation and destruction operators and has no simple logical analogue. The sign , in essence, is given by the number of 1's that stand to the left of the one just created or destroyed. This number normally cannot be determined atomically and in the past great care has had to be exercised in the design of algorithms to compute it (Whitehead et.al., 1977).

The alternative scheme that we introduce here represents a one particle state not as 00001000, for example, but as 00001111, the presence of the occupied orbit being indicated by the transition from 0's to 1's. A two particle state represented in the occupancy scheme as

0001000 OR 0000010 = 00010010

now becomes

0001111 XOR 0000011 = 00011100

The oddness or evenness of the number of occupied orbits to the left of a particular bit position is now given by the presence of a 0 or a 1 in that position and this can be readily determined atomically. It is also very easy to recover the occupancy representation (OSD) from the parity representation (PSD), as we call it, as follows:

OSD = PSD XOR (shr (PSD))

where shr indicates the operation of right shifting the argument one place (without any sign bit propagation).

The formal relationship between the two schemes is as follows. Let A_j be the operator which compliments all the bits to right of bit j-1, and let O_j be the operator which changes the sign of the Slater Determinant if no transition occurs at bit j. Then, if $a_j^\dagger$ and a_j are the ordinary creation and destruction operators for the orbit j, the following correspondences hold :

$$A_j \leftrightarrow a_j{}^\dagger + a_j$$
$$O_j \leftrightarrow 2\, a_j{}^\dagger a_j - 1$$
$$a_j{}^\dagger \leftrightarrow 1/2\,(1 + O_j)\, A_j \qquad = 1/2\ A_j\,(1 - O_j)$$
$$a_j \leftrightarrow 1/2\ (1 - O_j)\, A_j \qquad = 1/2\, A_j\,(1 + O_j)$$

and the commutation algebra of the a's is replaced by the mixed relations

$$A_i\, A_j + A_j\, A_i = 2\delta_{ij}, \quad A_i{}^2 = 1$$
$$O_i\, O_j - O_j\, O_i = 0\ , \quad O_i{}^2 = 1$$
$$A_i\, O_j - O_j\, A_i = -2\, O_j\, A_j\, \delta_{ij}$$

The last of these represents the fact that A_i and O_j commute if $i \neq j$, but anticommute if $i = j$. Using these relations any operator can be expressed in terms of the O and A, though the awkward non-linear features make the calculations tedious.

The algebraic complexity of the new operations is not however reflected in computational complexity. The algorithm (1), for example, would simply be replaced by

```
if ( $O_j$ ( PSD ) = PSD )   result = NULL
else       result = $A_j$ ( PSD )
           Sign = $(-1)^{\text{value of bit j in the PSD}}$
```

The parity representation of Slater Determinants has been implemented in a new shell model program designed to take advantage of the inherent parallelism of the process. In this program we use a hybrid scheme in which bases of Slater Determinants are generated in the occupancy representation and efficiently converted to the parity representation as required. This arrangement gives access to the best features of both representations.

Reference:

R.R. Whitehead, A.Watt, B.J. Cole and I. Morrison, Advences in Nuclear Physics, vol. 9, M. Baranger and E. Vogt (eds.), Plenum Publishing (1977).

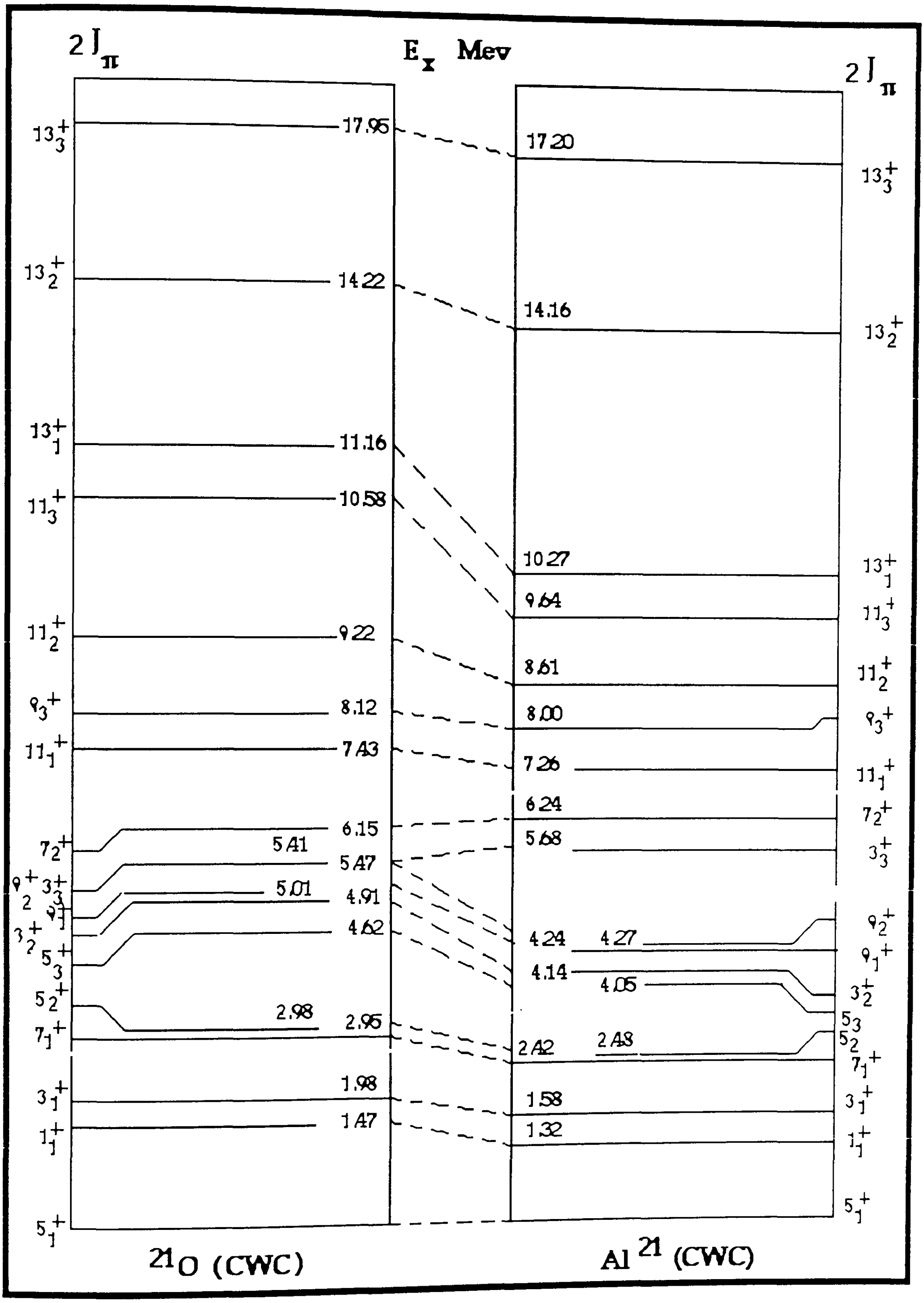


Fig :5.7 Theo. Energy Spectrum of Mirror Nuclei O^{21} and Al^{21}

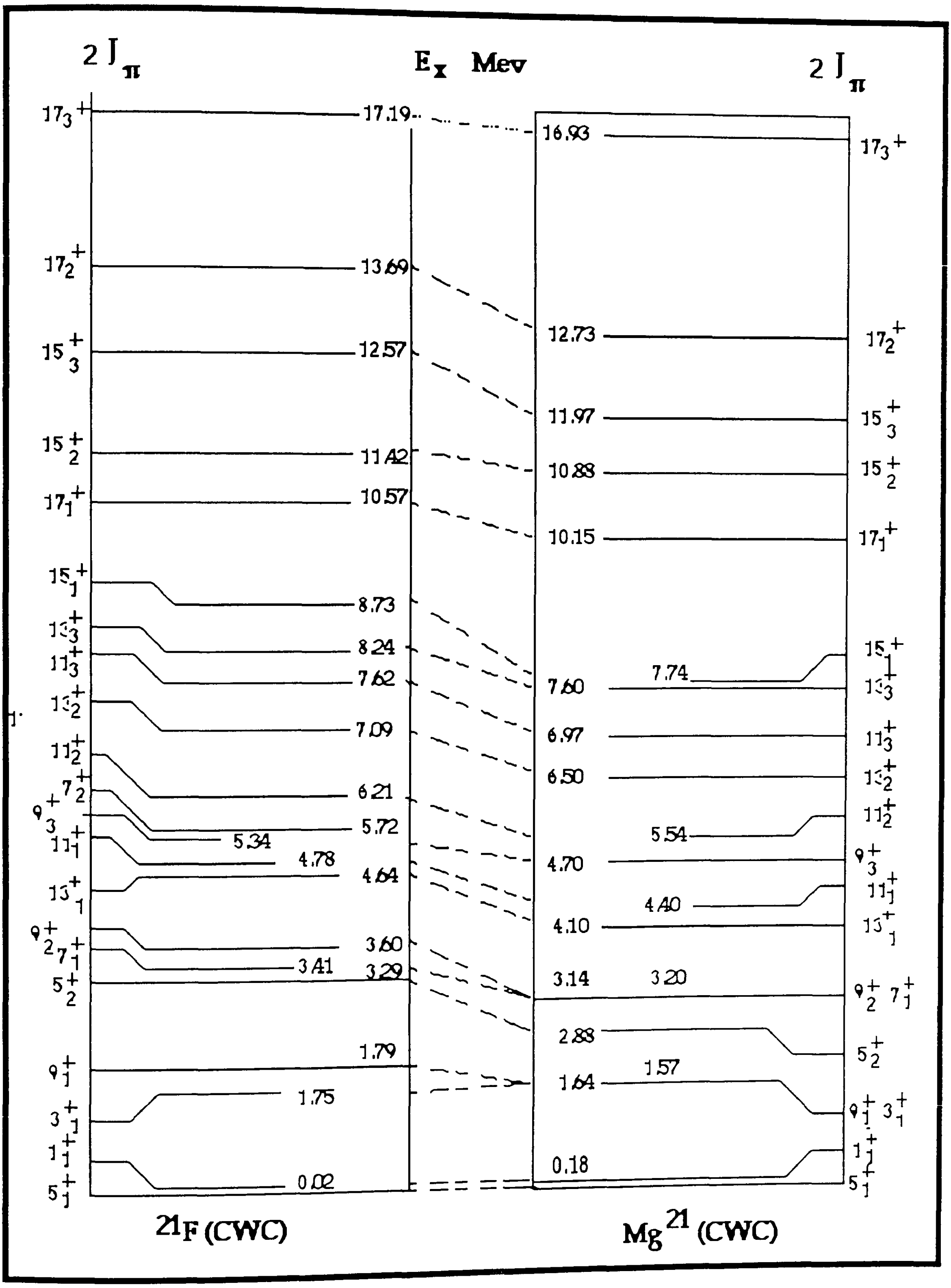


Fig: 5.8 Theo. Spectrum of Mirror Nuclei F^{21} and Mg^{21}

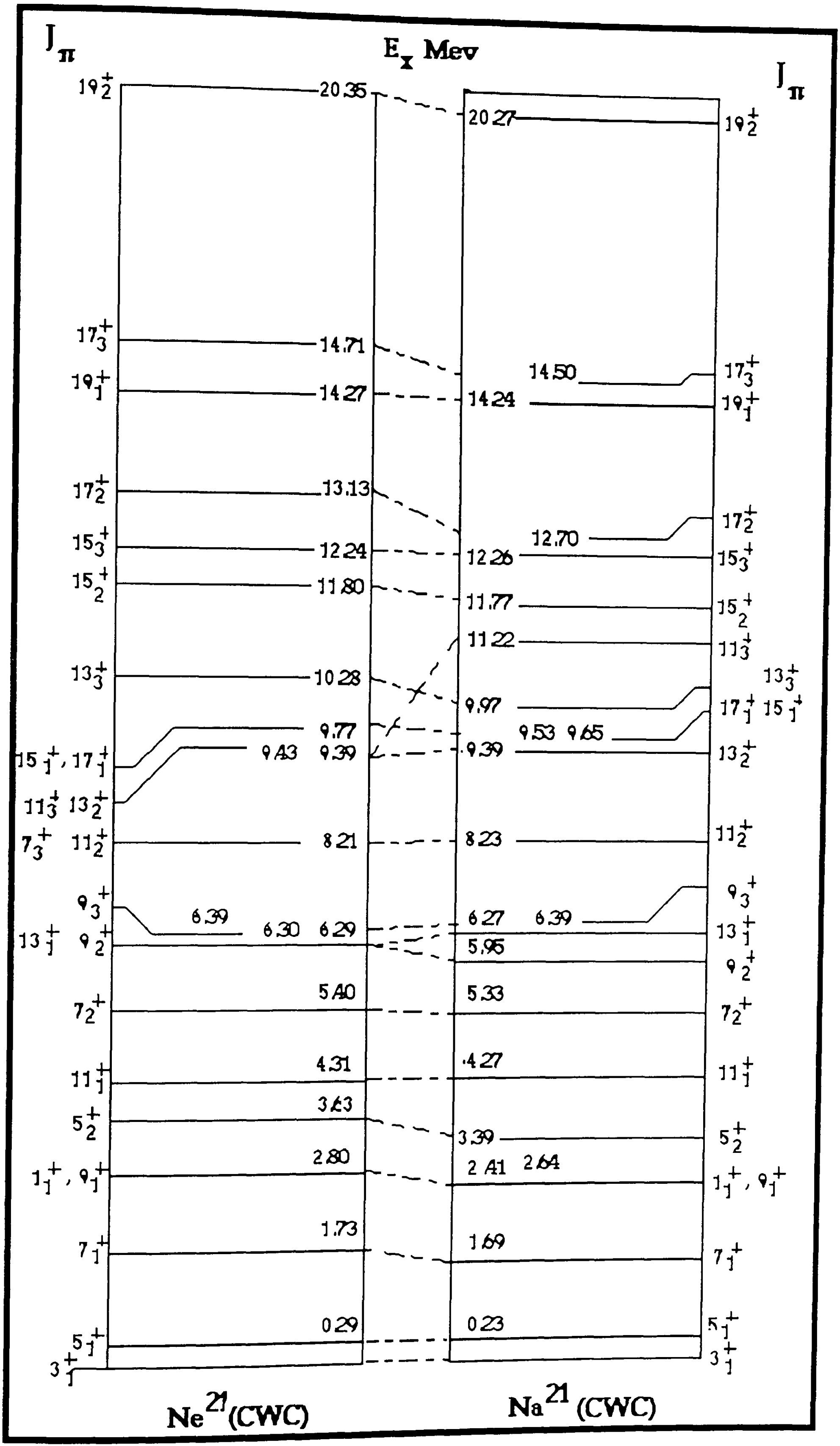


Fig : 5.9 Theo. Spectrum of Mirror Nuclei Ne^{21} and Na^{21}

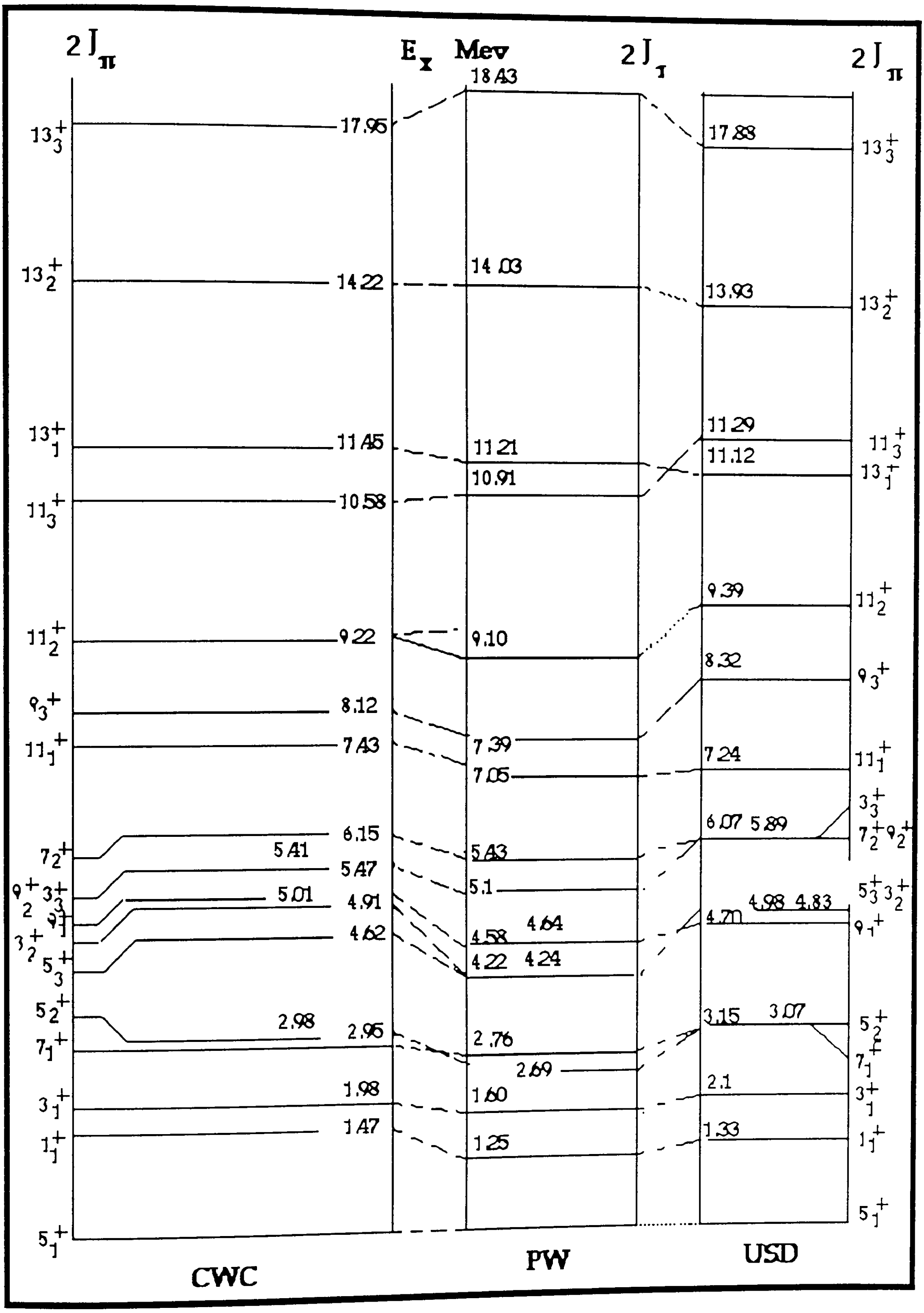


Fig 5.10 Comparison of eigen values for ^{21}O with different effective interactions

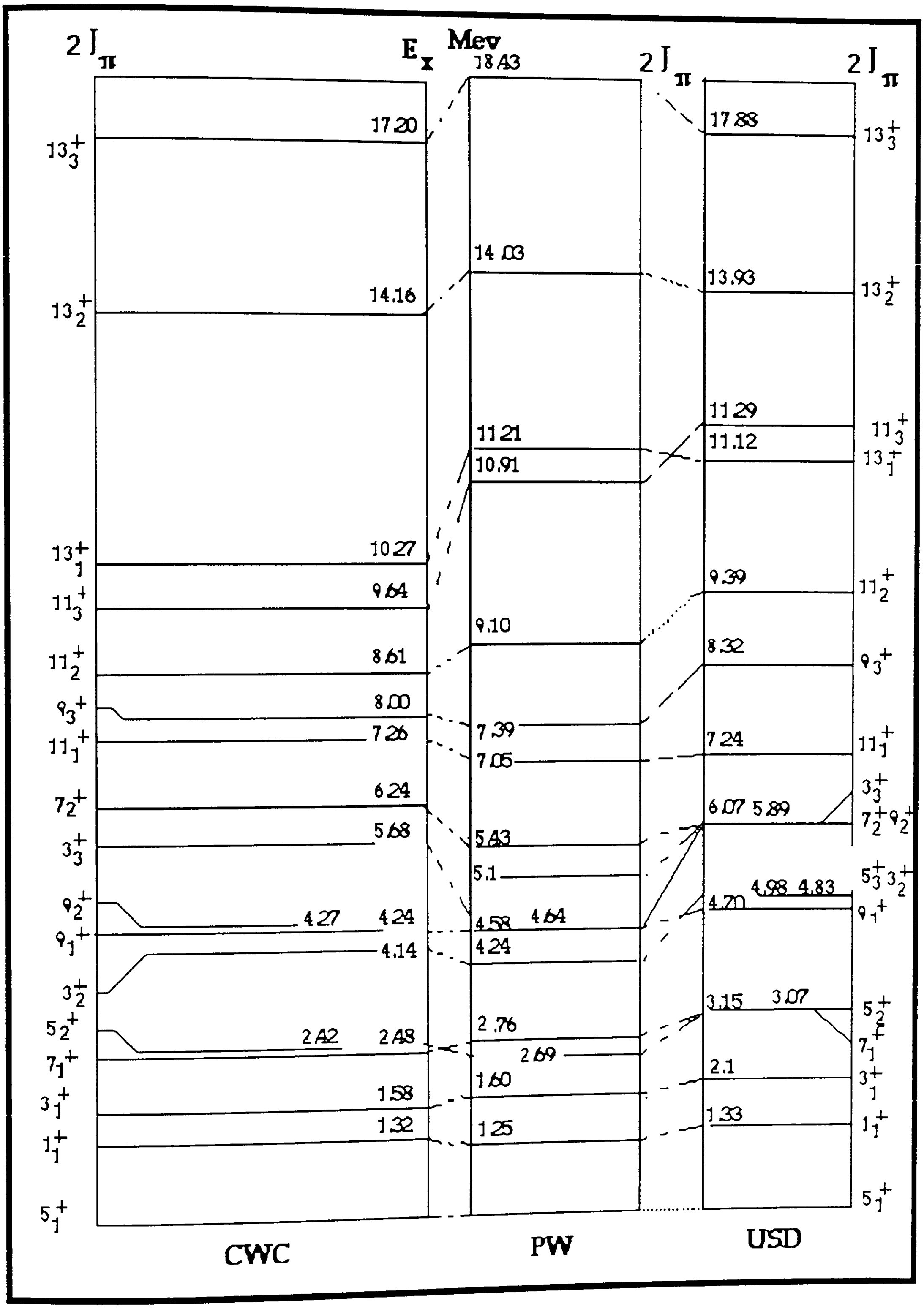


Fig 5.11 Comparison of eigen values for ^{21}Al with different effective transitions

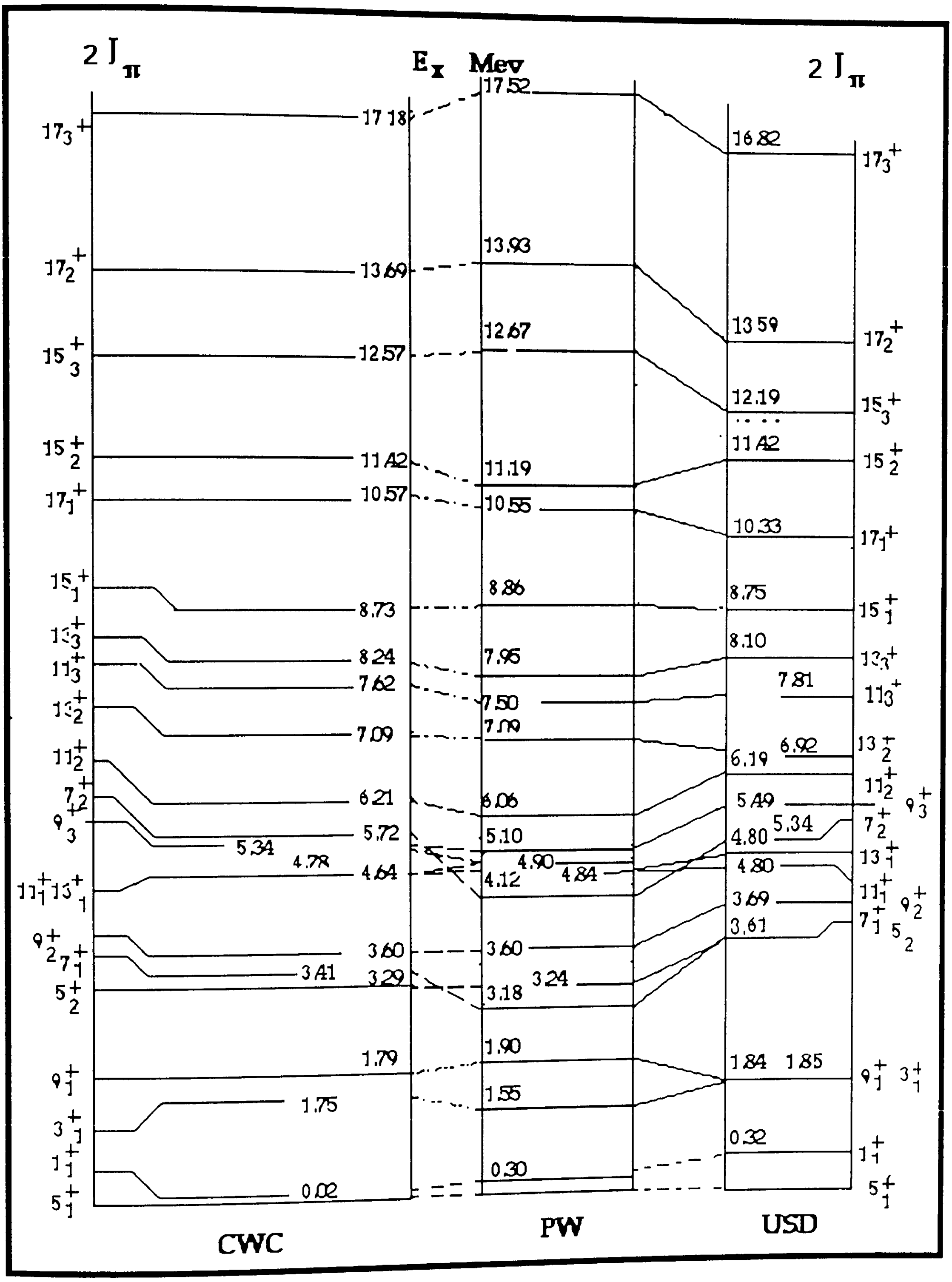


Fig 5.12 Comparison of eigen values for ^{21}F with different effective transitions

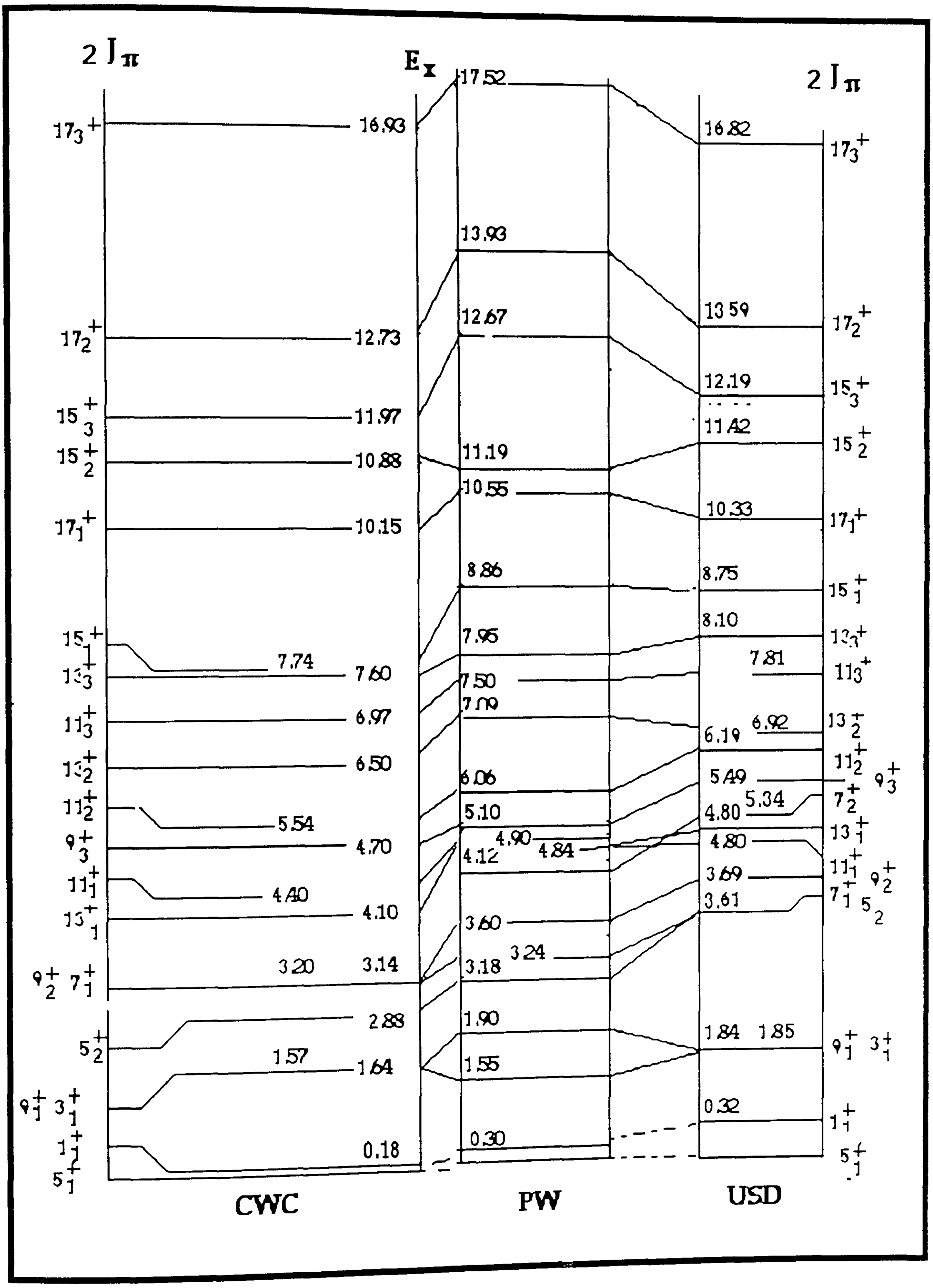


Fig 5.13 Comparison of eigen values for ^{21}Mg using different effective interactions

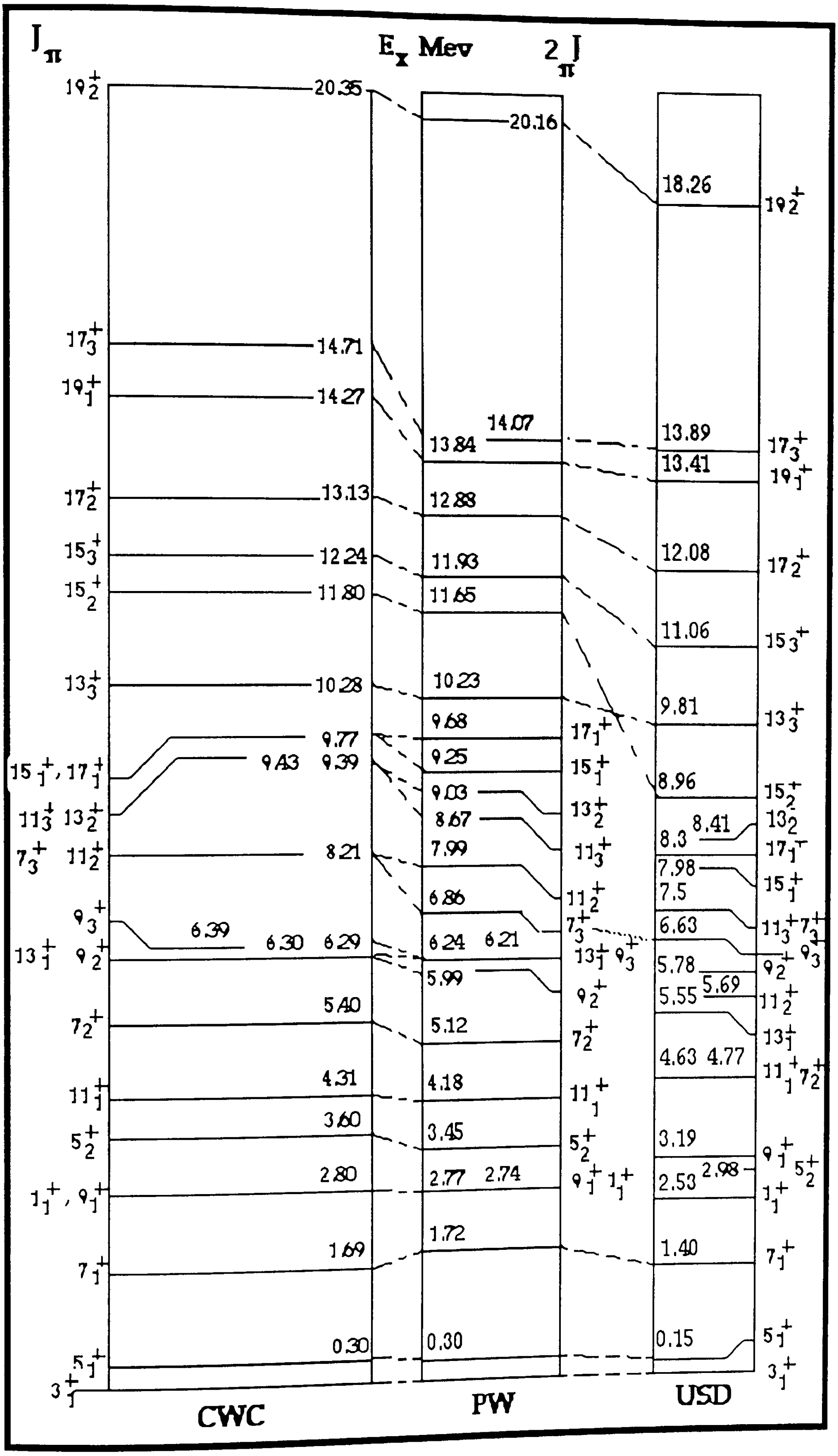


Fig 5.14 Comparison of eigen values for ^{21}Ne with different effective interactions

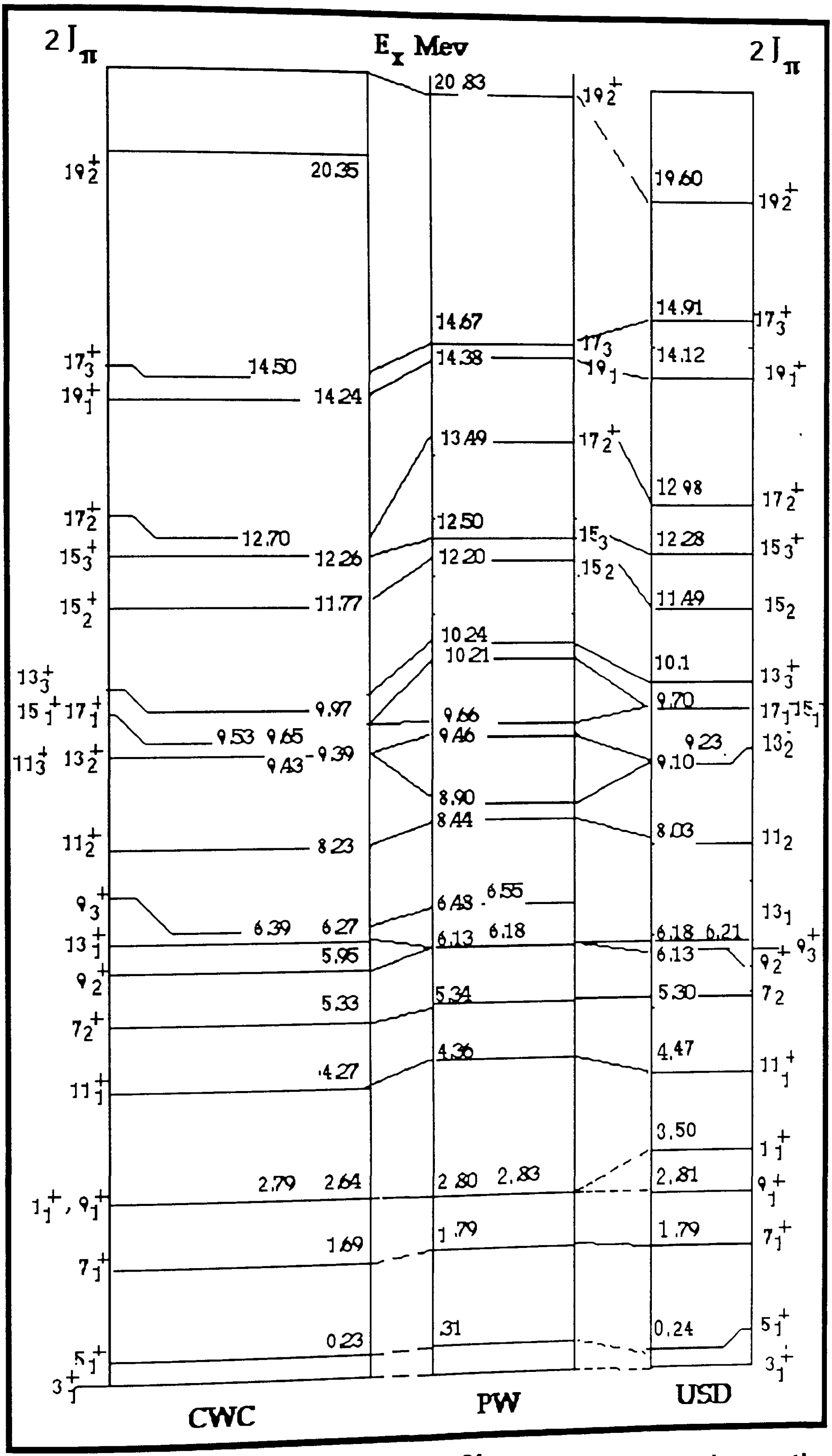


Fig 5.15 Comparison of eigen values for ^{21}Na using different interactions

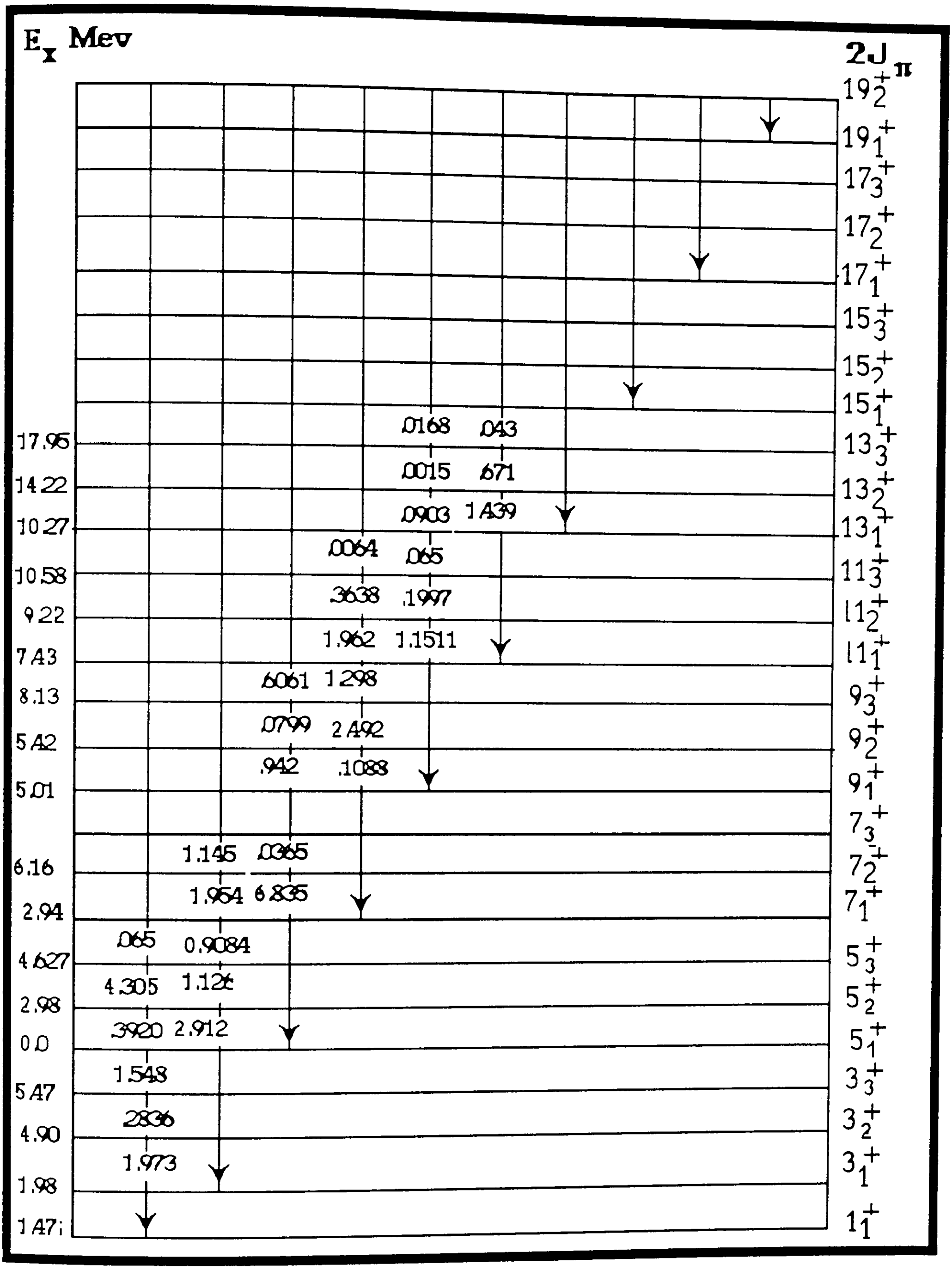


Fig 5.16 Theo. Electric Transition rates B(E2) Values for ^{21}O

Comments : 1- Units $e^2 fm^4$ (1 W.U. = 3.4374 $e^2 fm^4$). 2- States are ordered in increasing J value. States with the same J are ordered according to energy. Transition rates are shown to a state J from states with A.M. not less than J. The B(E2) values of a transition in the opposite direction can be calculated from

$B(E2, J_1 \text{->} J_2) = ((2J_2 + 1)/(2J_1 + 1))\ B(E2, J_2 \text{->} J_1)$.

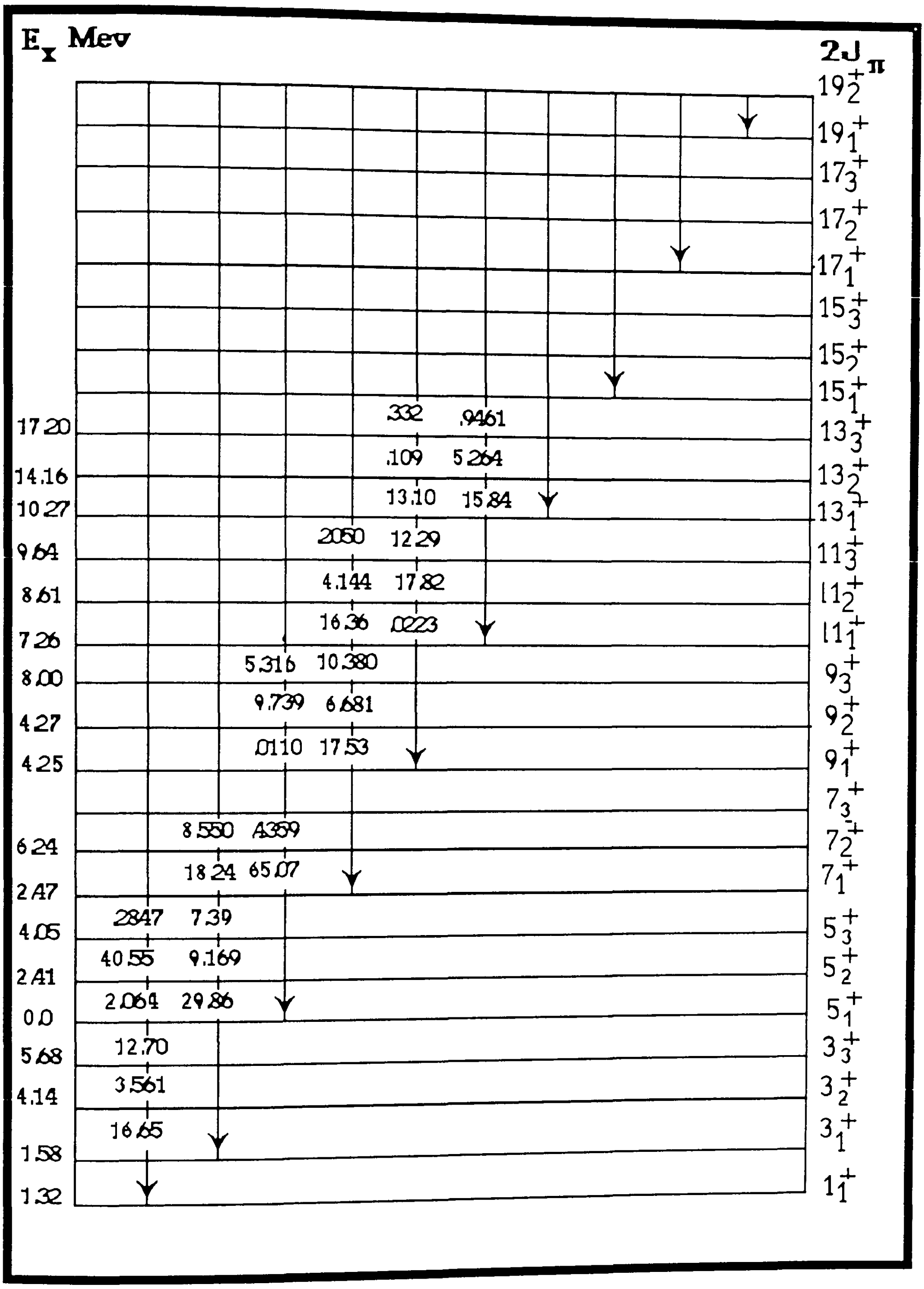


Fig 5.17 Theo. Electric Transition Rates B(E2) for ^{21}Al

Also see caption fig 5.16

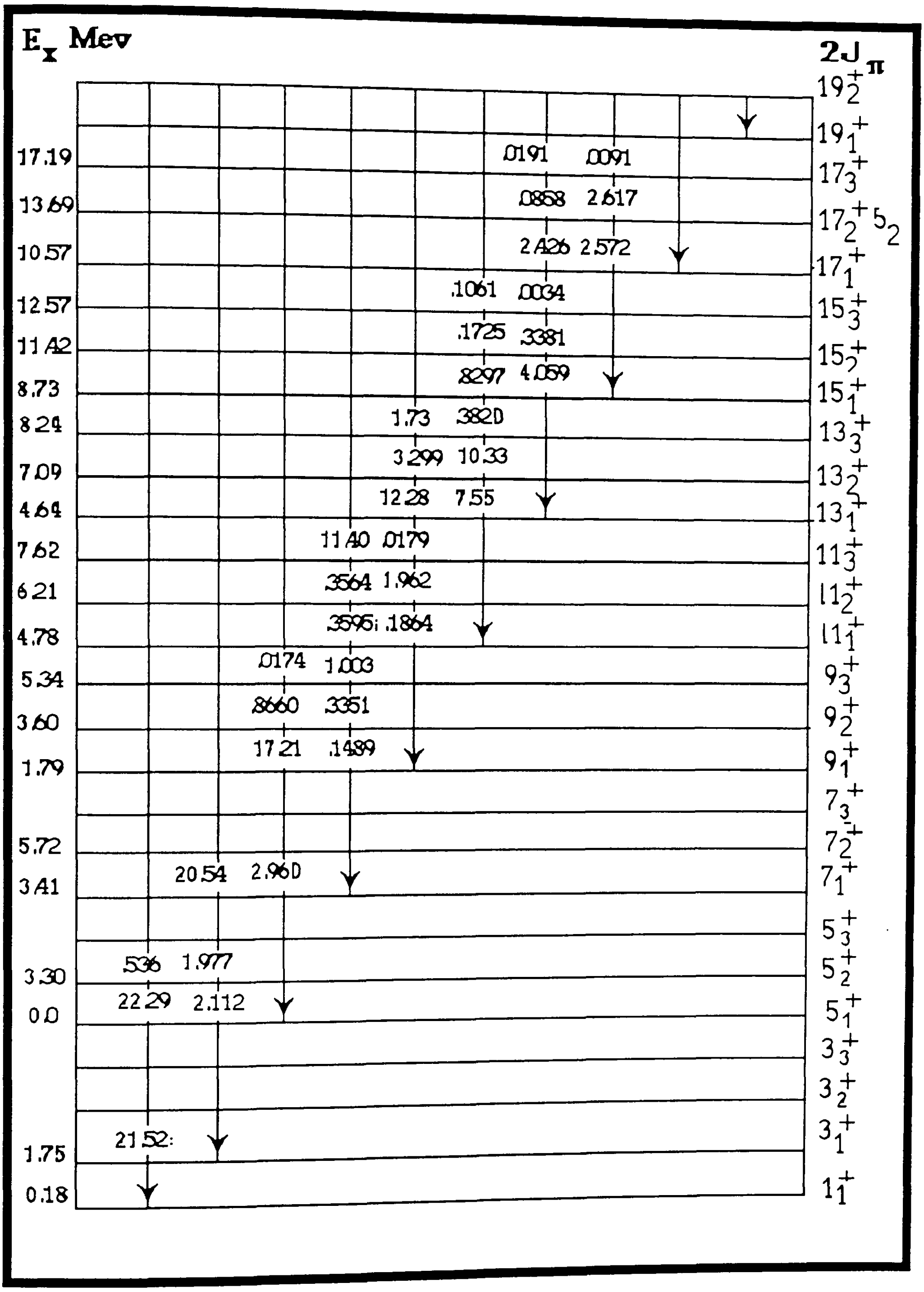


Fig 5.18': Theo. Electric Transition Rates, B(E2) Values for ^{21}F

Also see caption fig 5.16

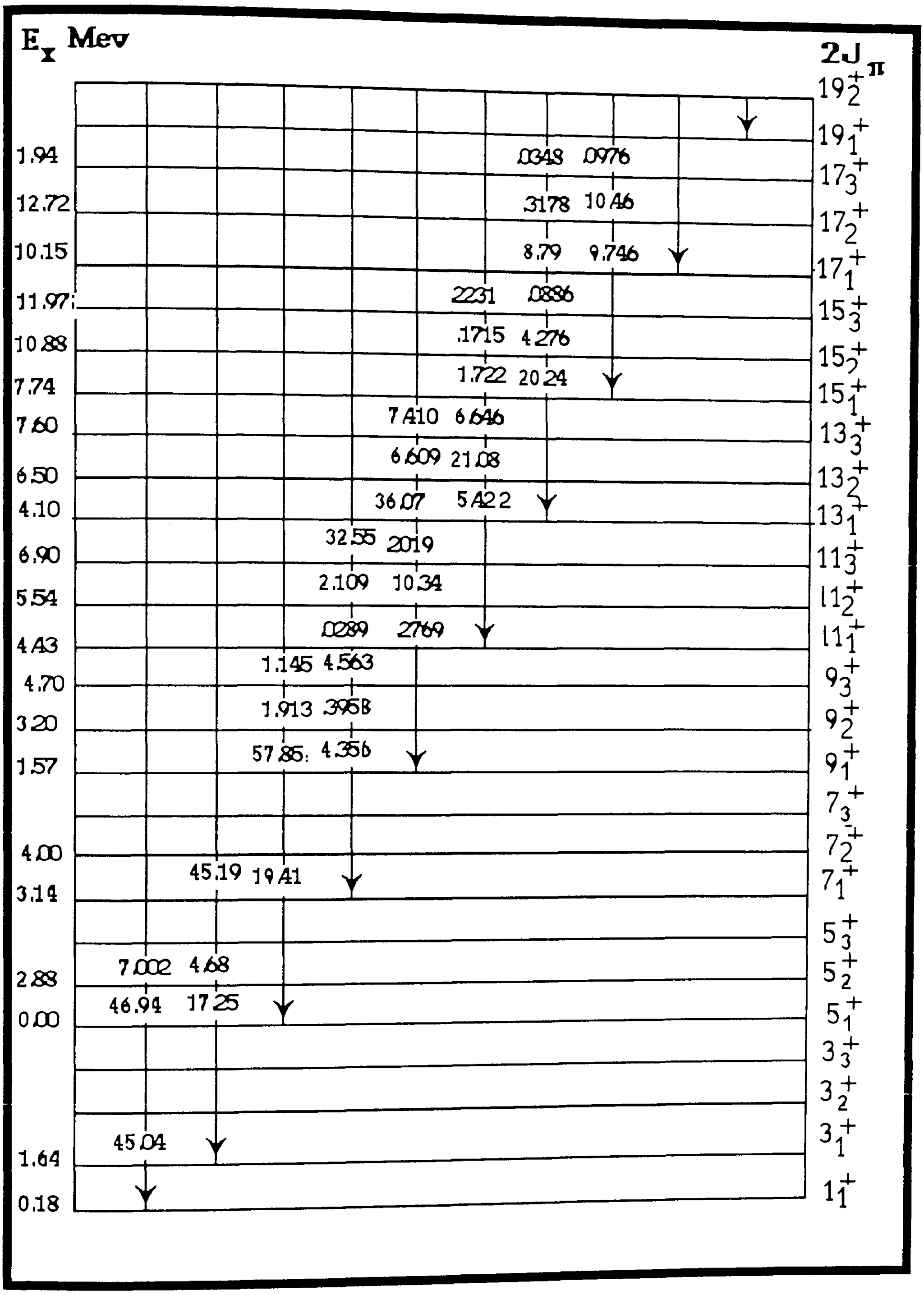


Fig 5.19 Theo. Electric Transition B(E2) Values for ^{21}Mg

Also see caption Fig 5.16

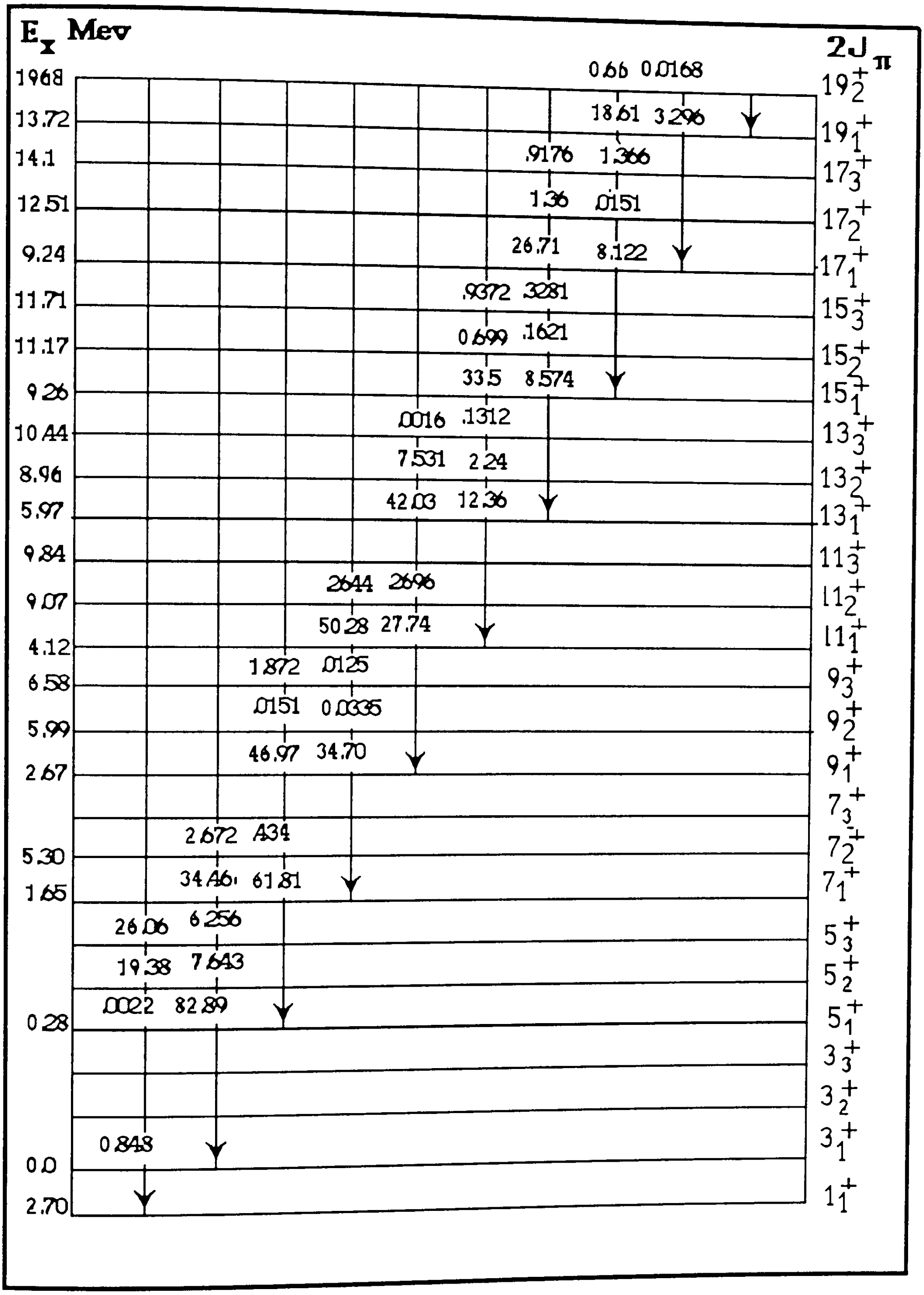


Fig 5.20 Theo. (CWC) Electric Transition Rates, B(e2) values for Ne^{21}

* Also see caption Fig 5.16

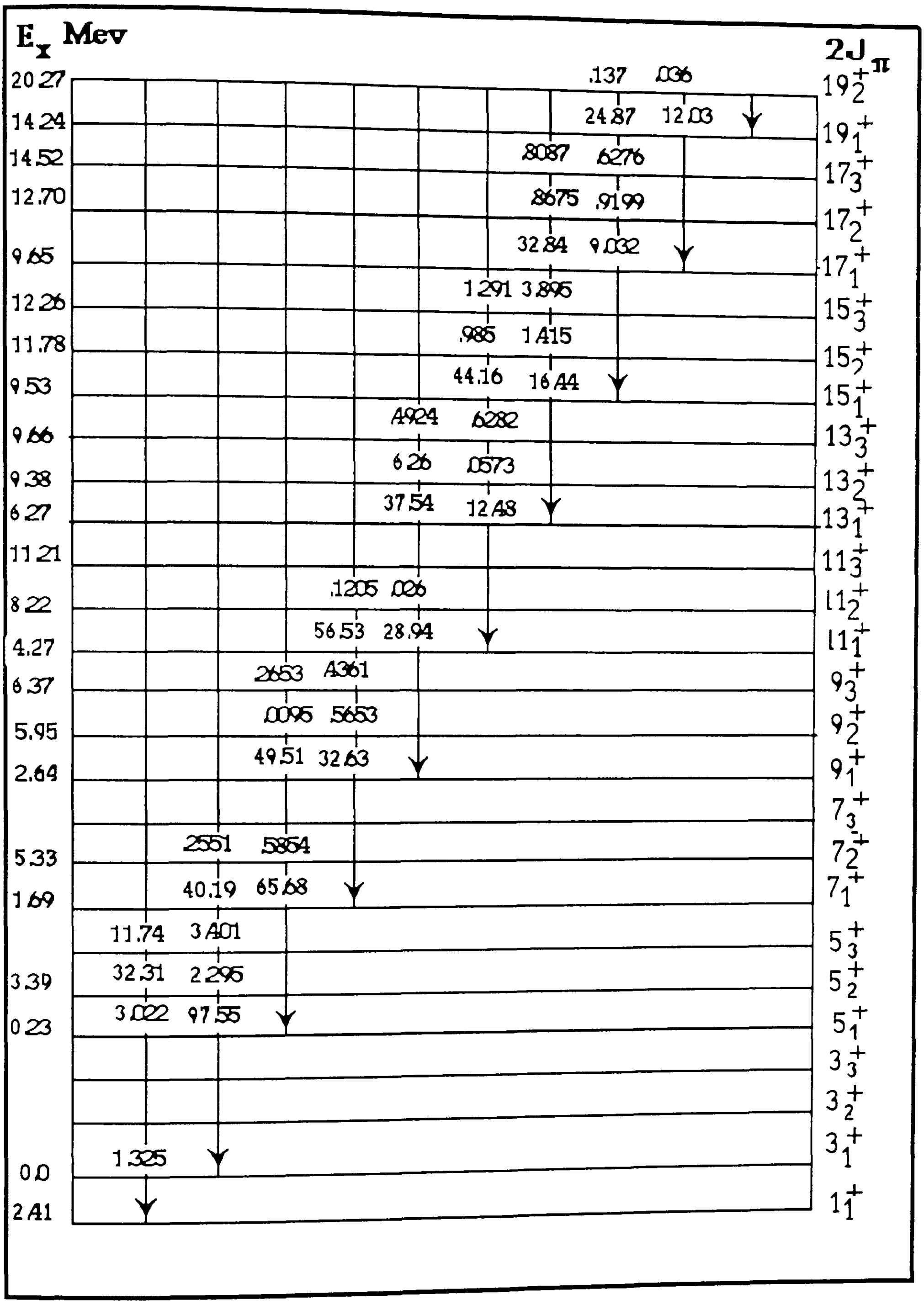


Fig 5.21 Theo. (CWC) Electric Transitions B(E2) Vlaues for Na^{21}

Also see caption Fig 5.16

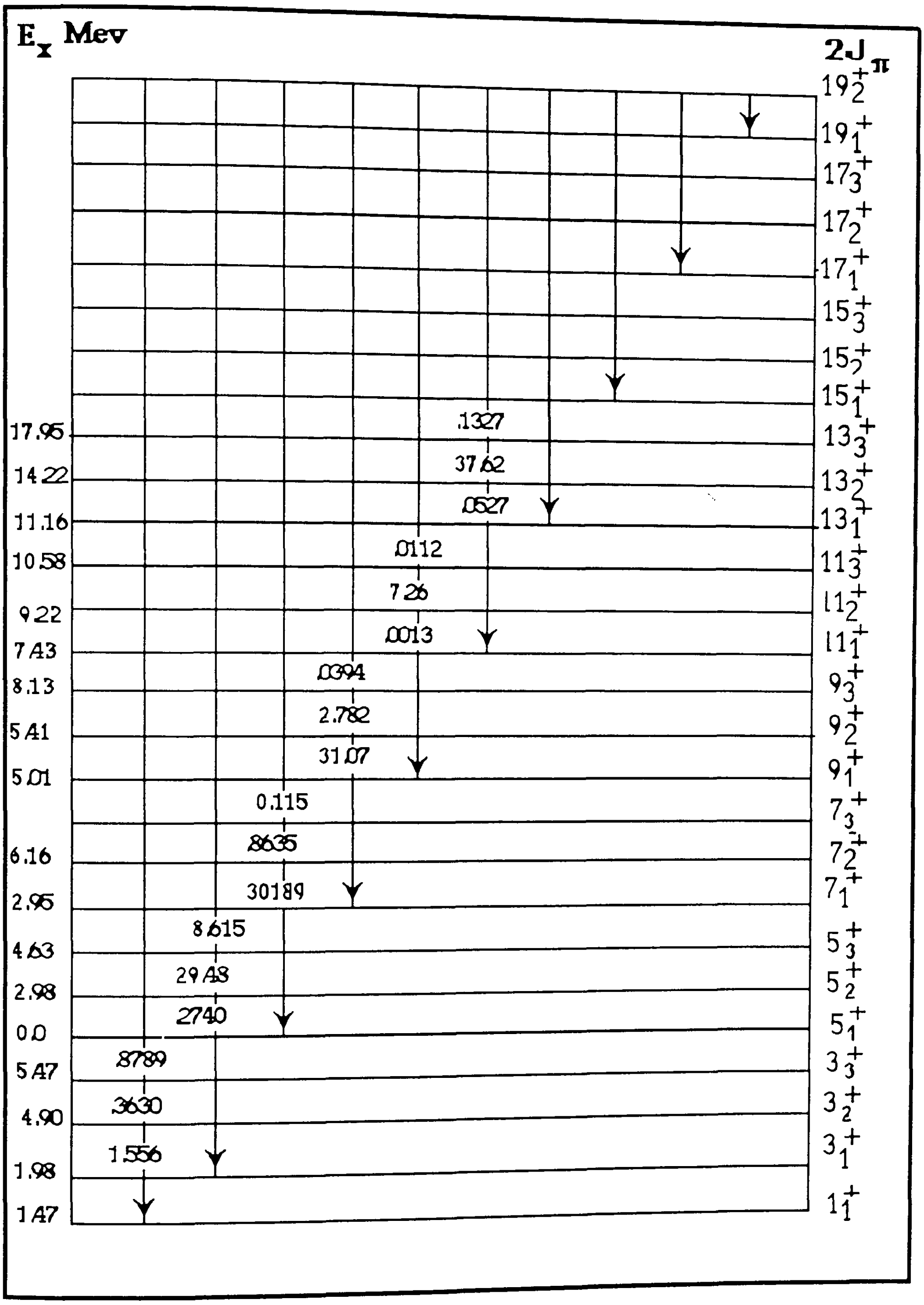


Fig 5.22 Theo. Magnetic Transitions B(M1) Values for ^{21}O

All B(M1) values have been multiplied by 100. Also see caption Fig 5.16

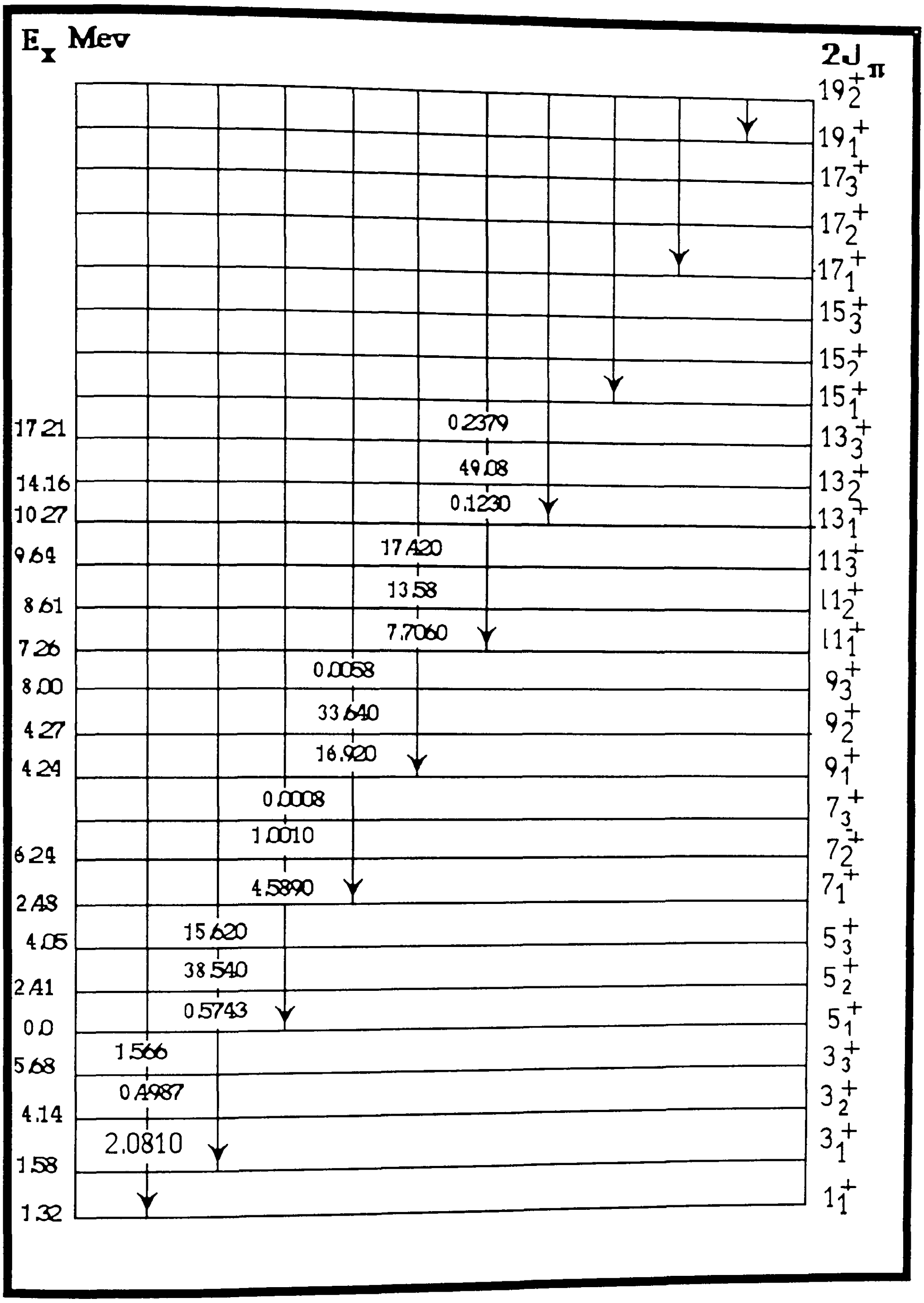


Fig 5.23: Theo. Magnetic Transition Rates B(M1) for ^{21}Al

* All B(M1) values have been multiplied by 100, Also see caption Fig 5.16

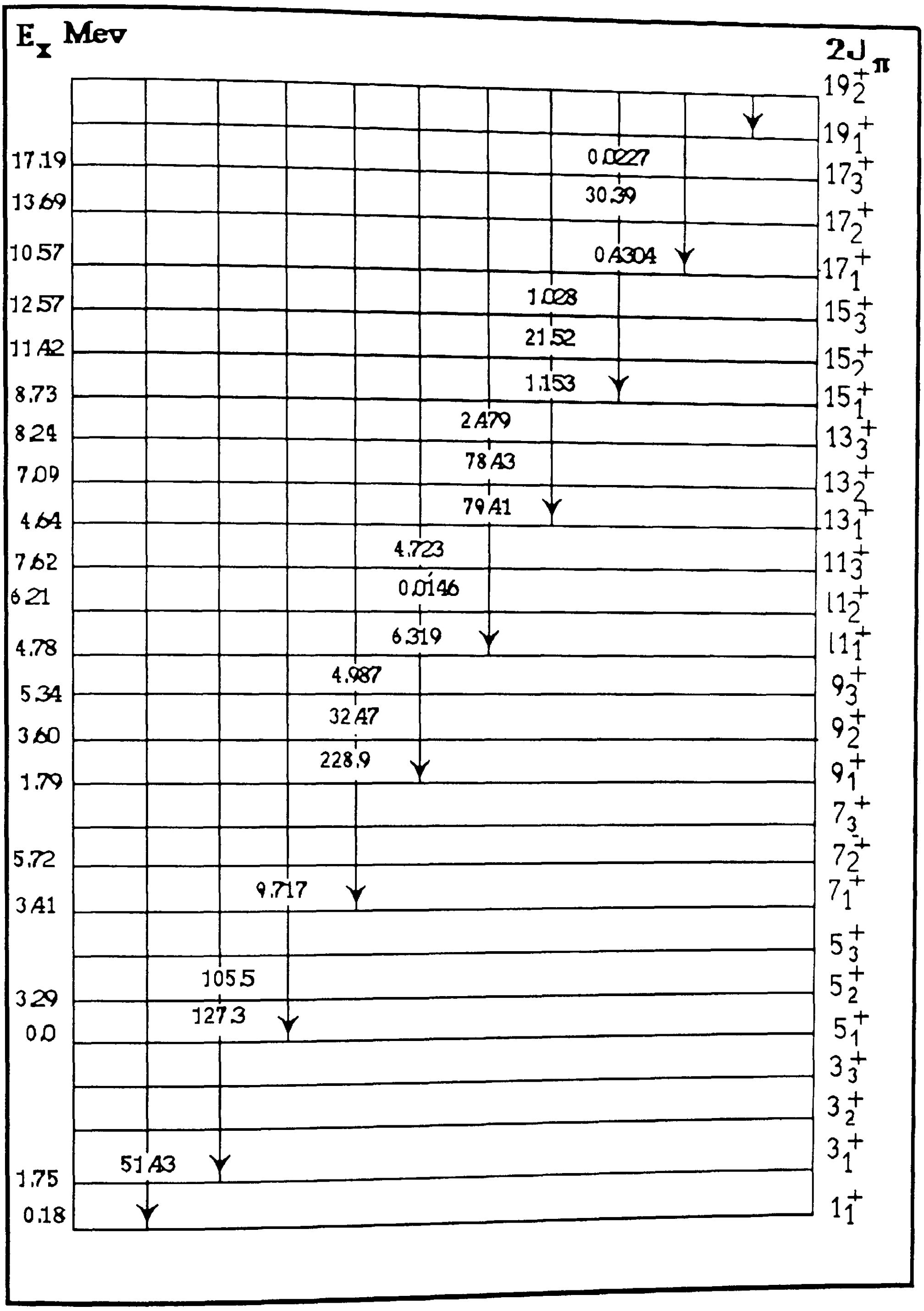


Fig 5.24 Theo. Electric Transitions B(E2) Vlaues for F^{21}

* All B(M1) values have been multiplied by 100, Also see caption Fig 5.16

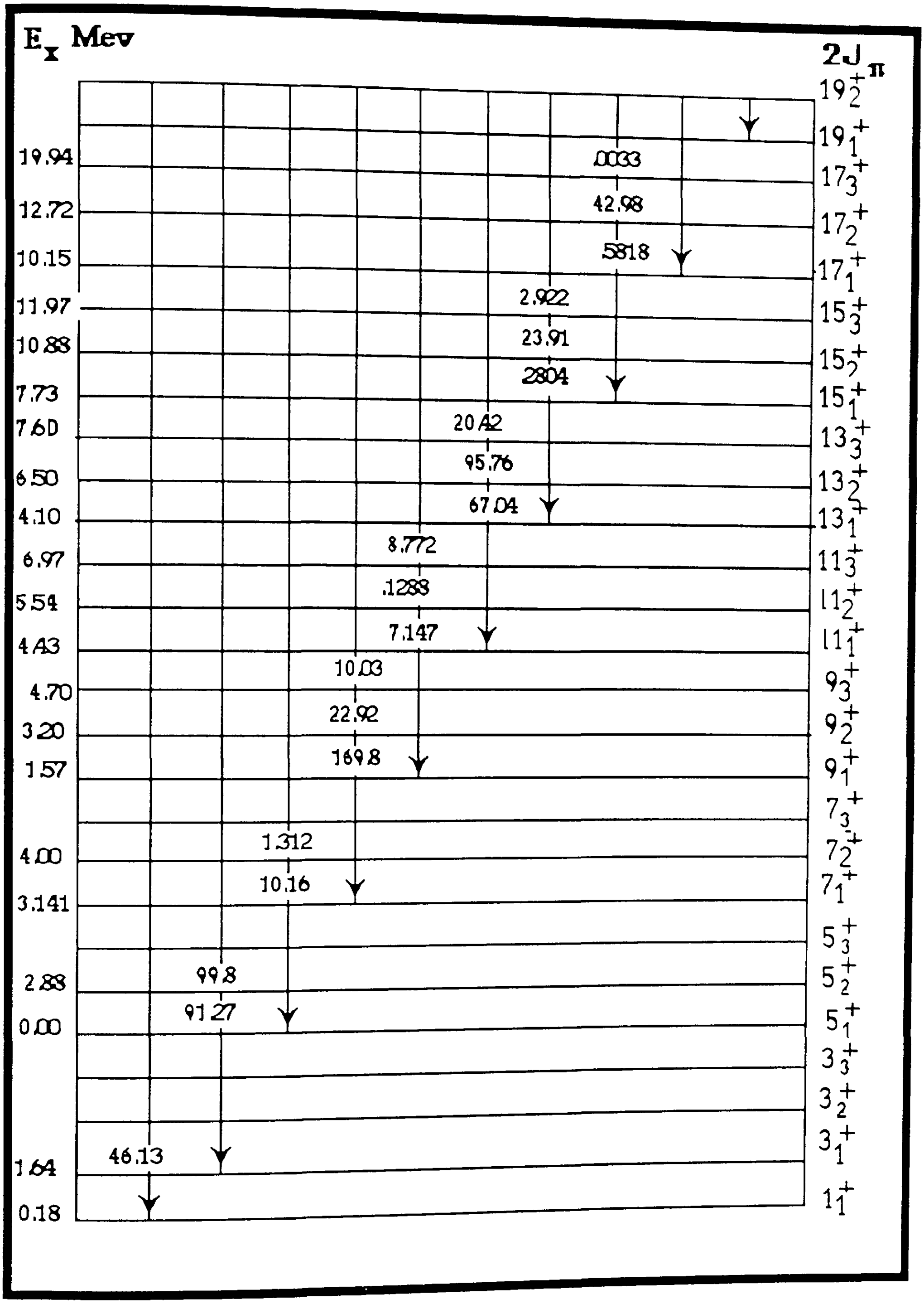


Fig 5.25 Theo. Magnetic Transition Rates, B(M1) Values for ^{21}Mg

All B(M1) Values have been multiplied by 100. Also see caption Fig 5.16

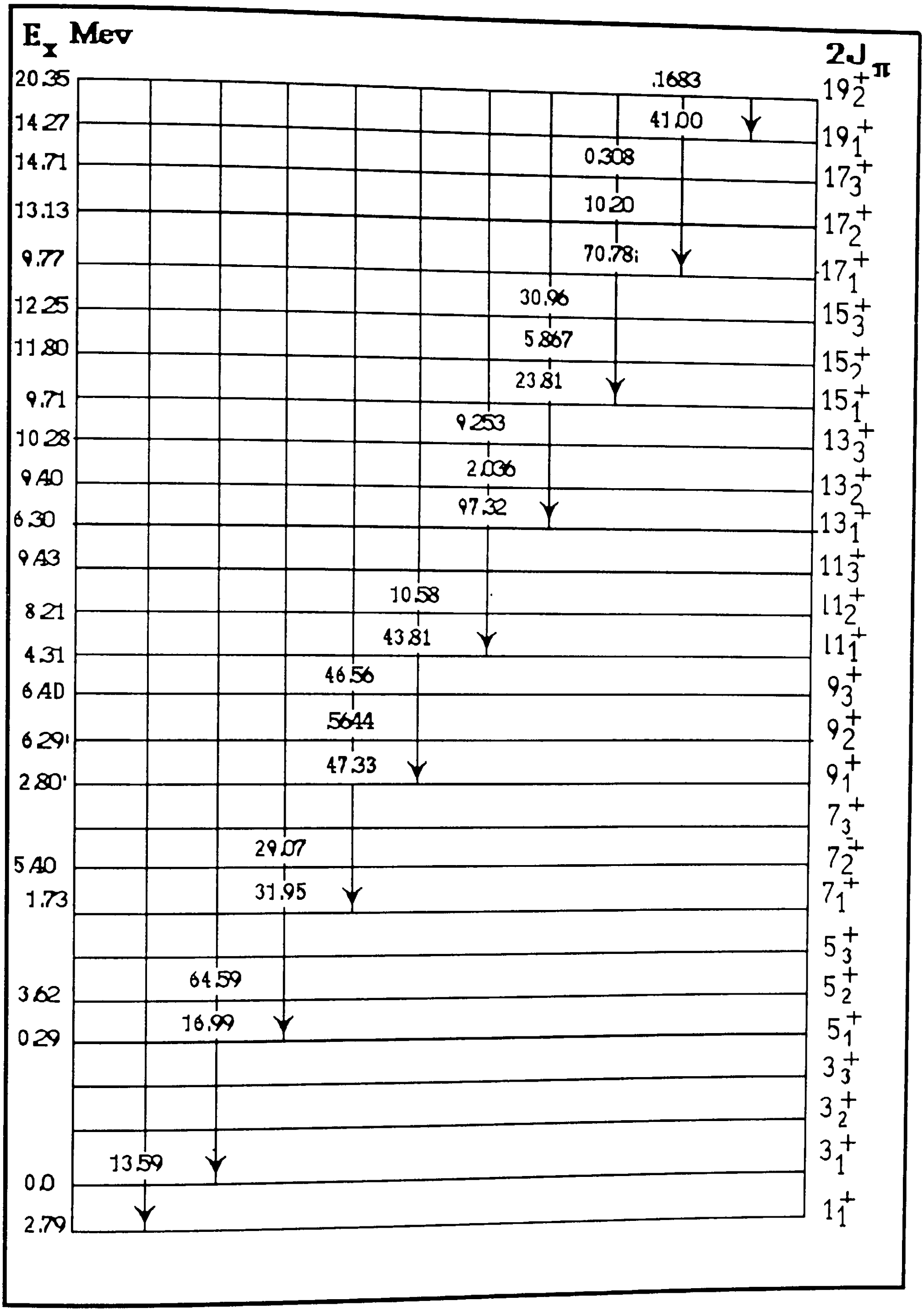


Fig 5.26 Theo. (CWC) magnetic Transitions, B(M1) Values for Ne^{21}

All B(M1) values have been multiplied by 100. Also see caption Fig 5.16

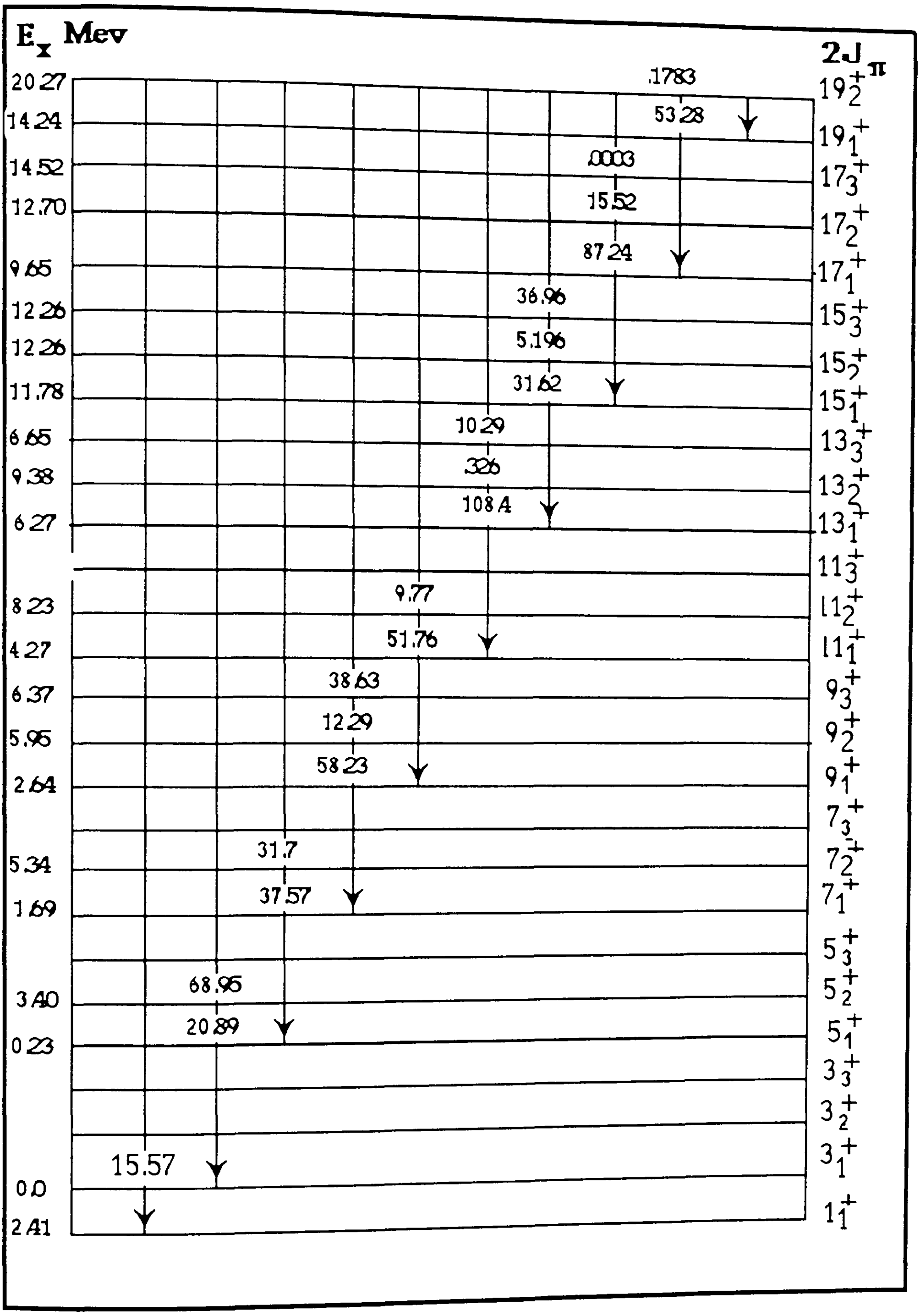


Fig 5.27 Theo. (CWC) Magnetic Transition rates B(M1) for Na^{21}

All B(M1) values have been multiplied by 100. Also see caption Fig 5.16

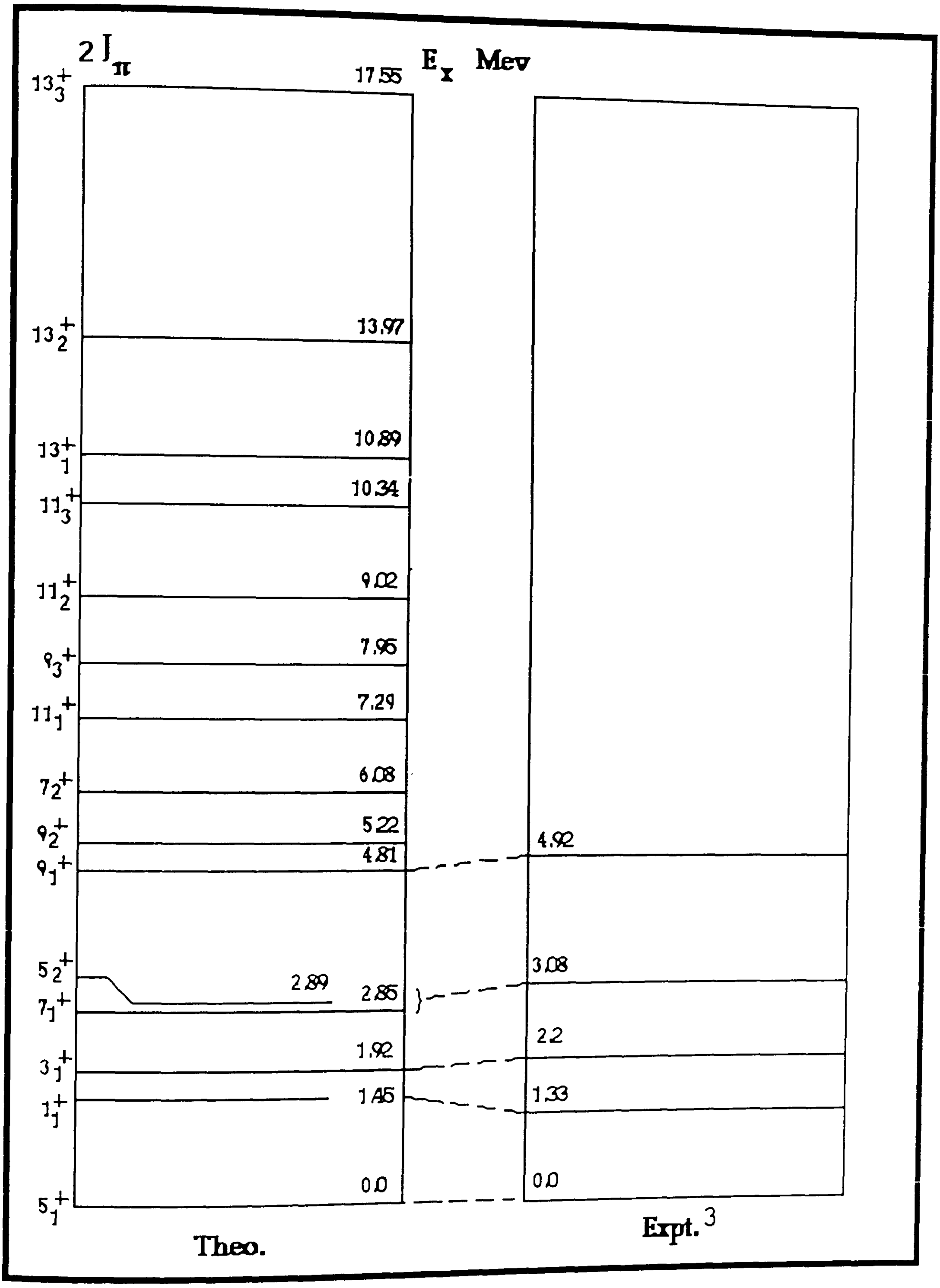


Fig 5.28 Comparison of Energy Spectrum for ^{21}O

3- Catford W.N., Nucl. Phys. A503 263-284 (1989)
W00ds P.J., et. al., Phys. Lett. B 150 n. 1, 2, 3 79-82 (1985)

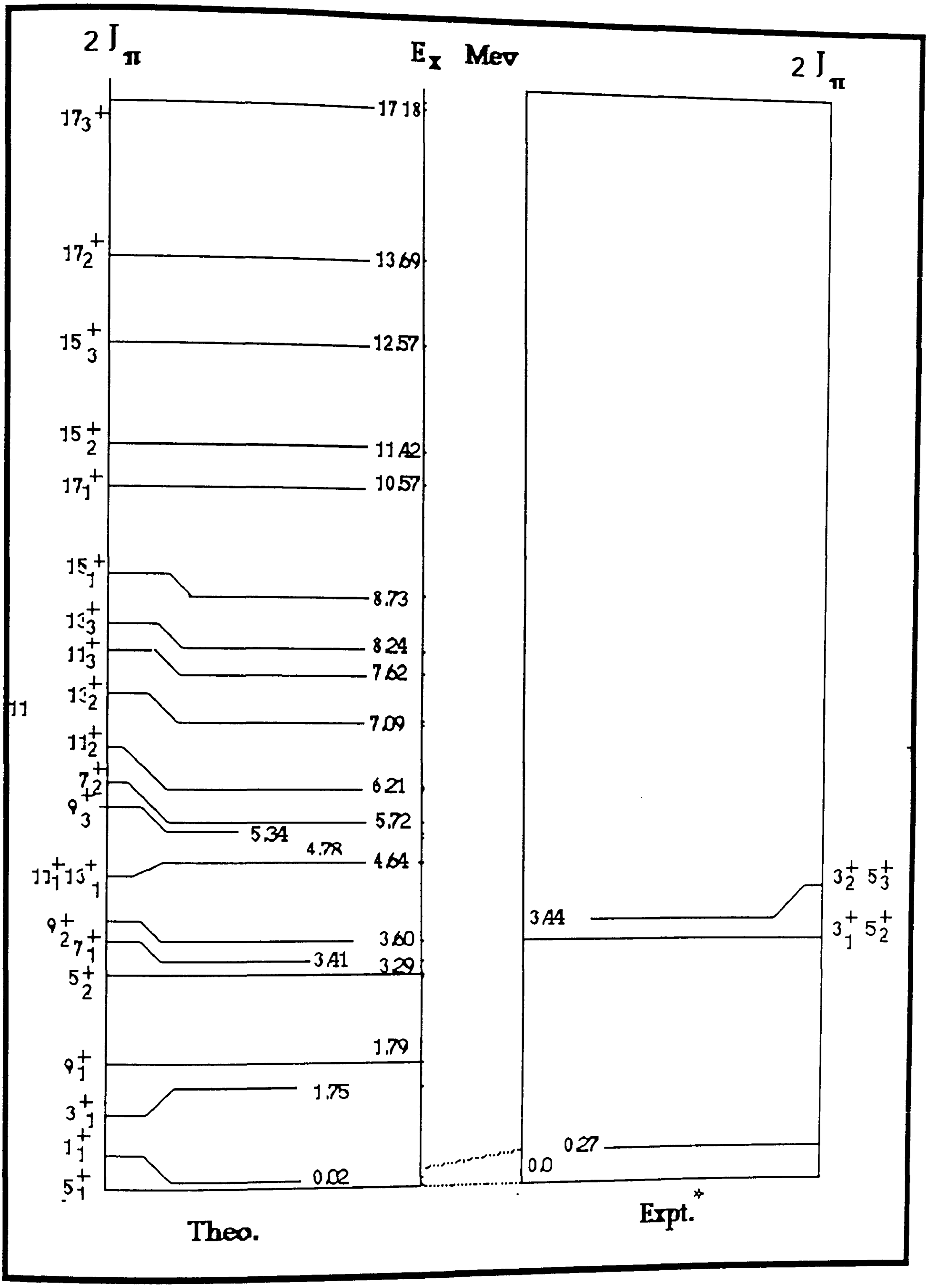


Fig 5.29 : Comparison of The Energy Spectrum for ^{21}F

* Alburger et.al (1981) ph. Rv.23

Antony M.S. et. al Nuv. Cim. 81,2, 414-422 (1984)

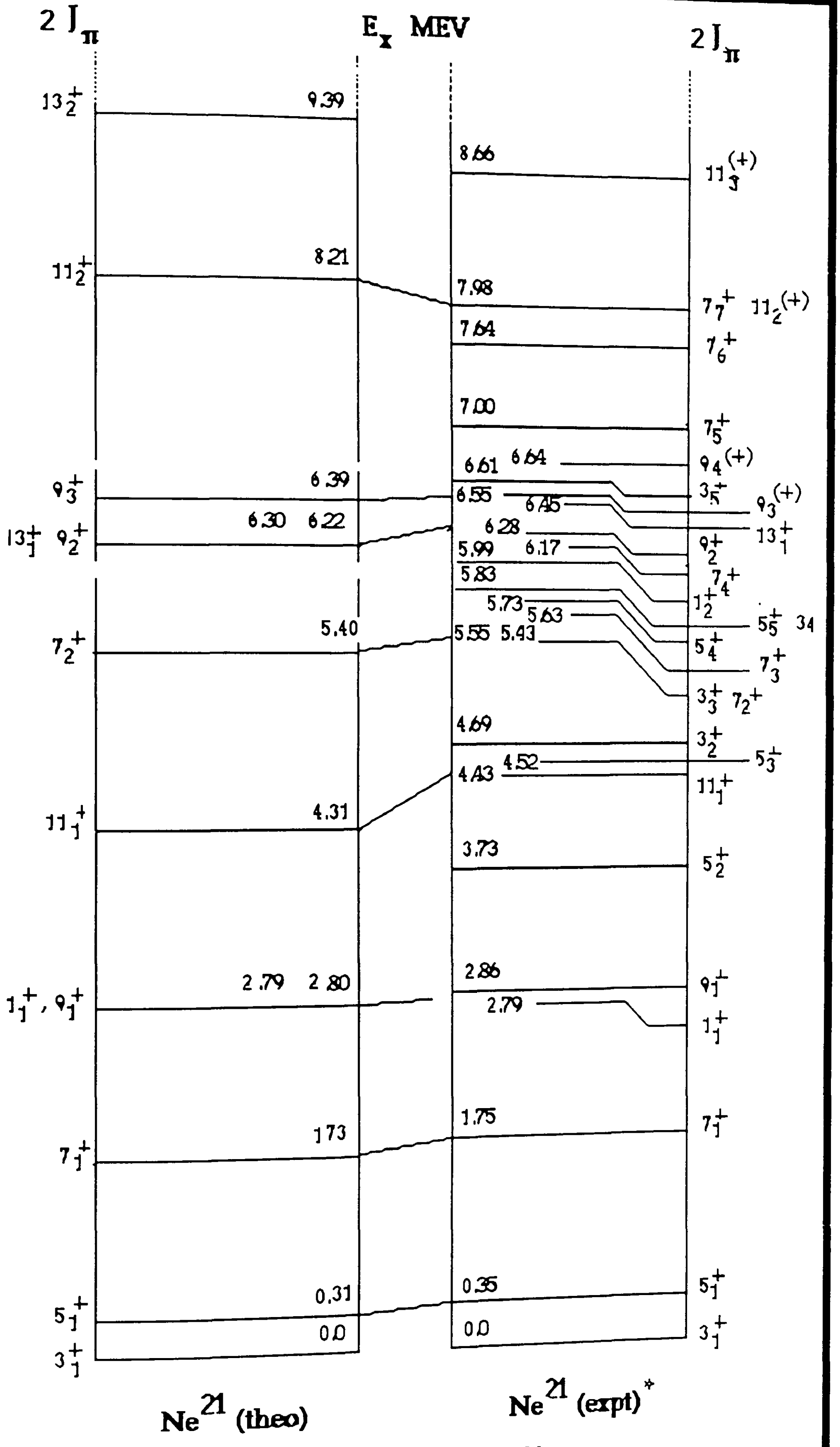
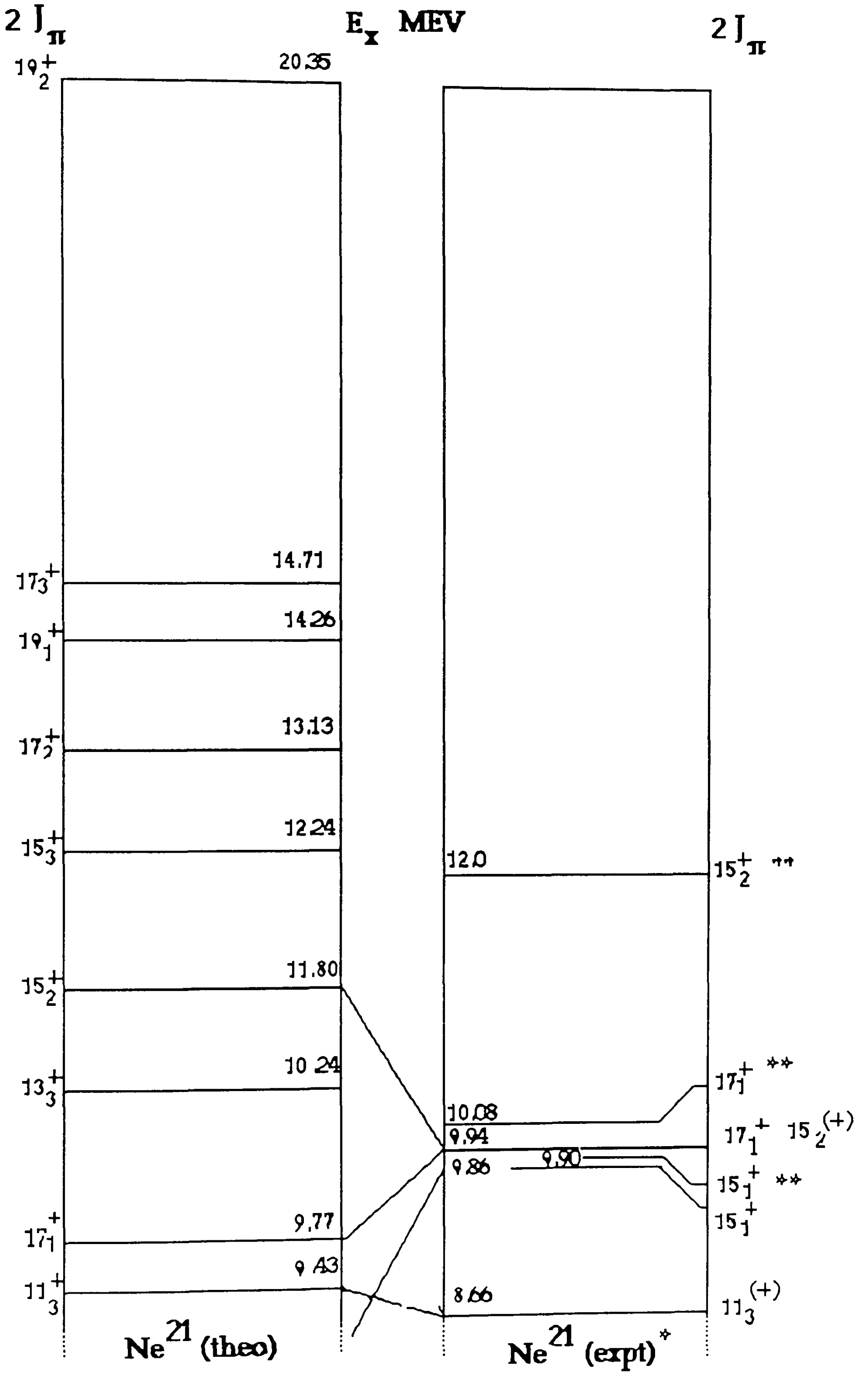


Fig 5.30 Comparison of spectrum for Ne^{21} Spectrum 0 - 9.36 mev



* Hoffman et. al Z. Phys. A. 332, (1989) 289-304.

** I.F. Wright et. al, in symposium in hon. of A. Arima Santa Fe(1989)

Fig 5.30.: Comparison of Spectrum of Ne^{21} (9.43 - 20.35 mev)

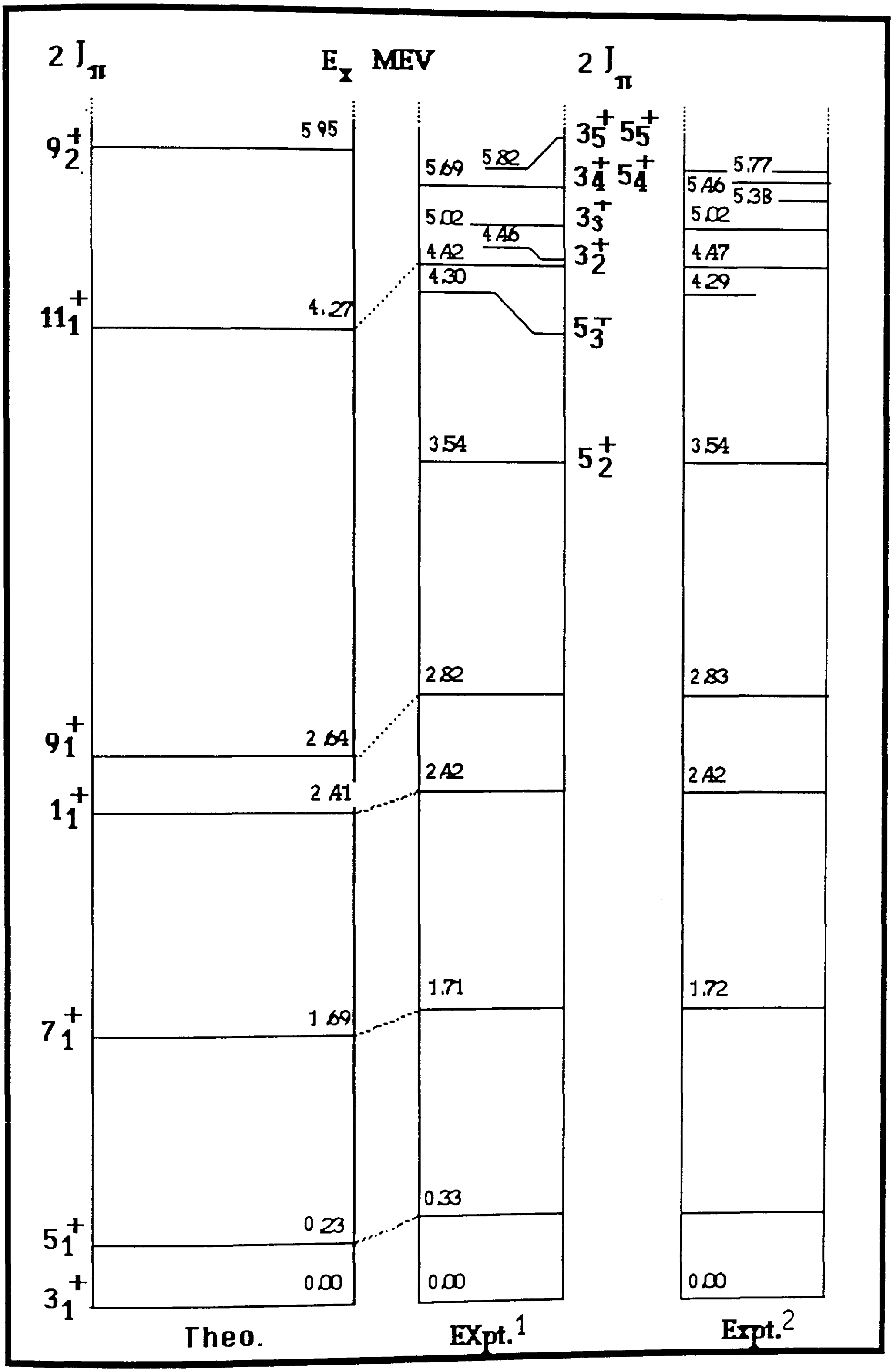


Fig 5.31 Comparison of expt. & Theo results ^{21}Na (0 -> 5.95 Mev)

1-Hofman A., et al. 1989 2- Endt P.M. (1973)

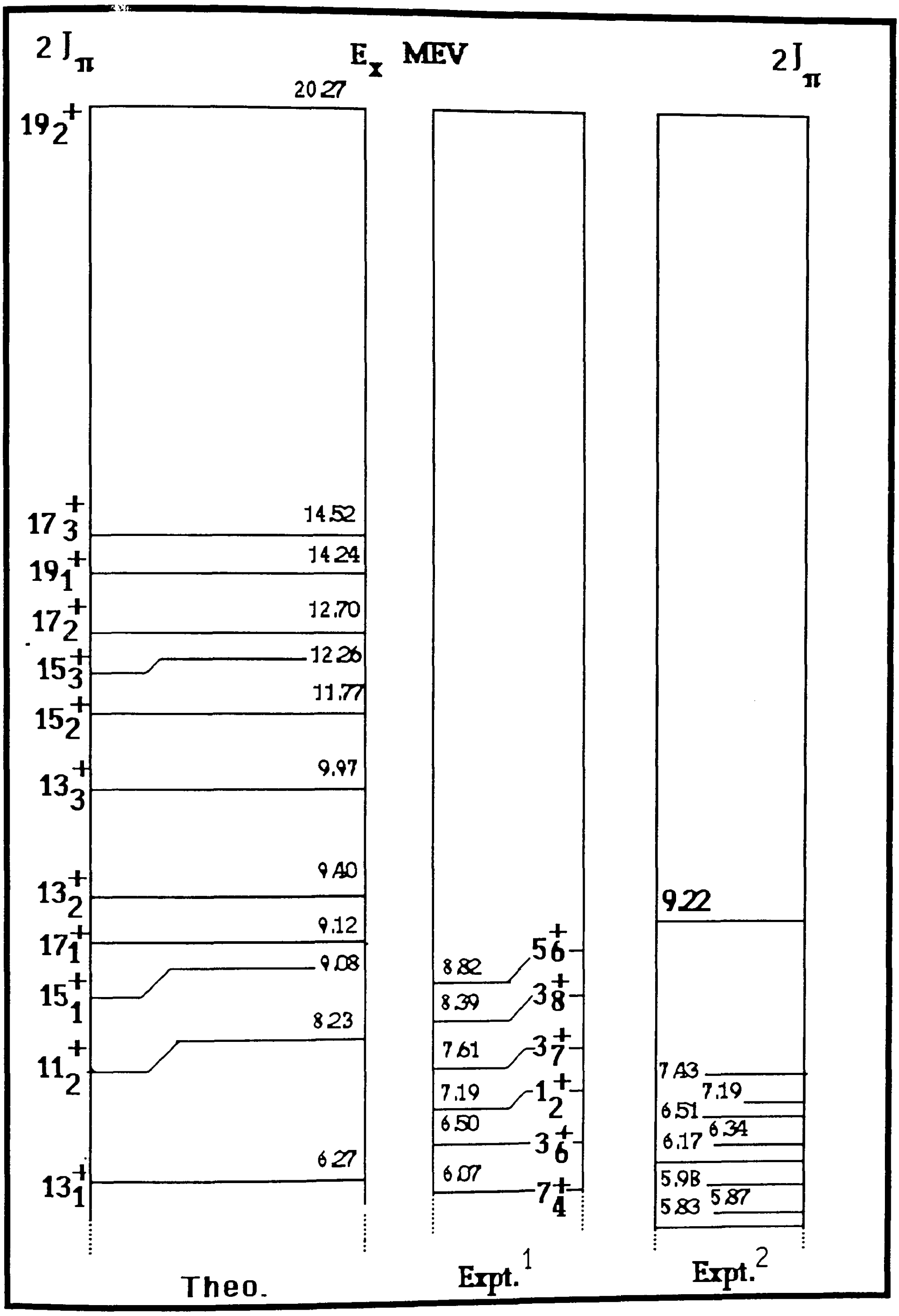
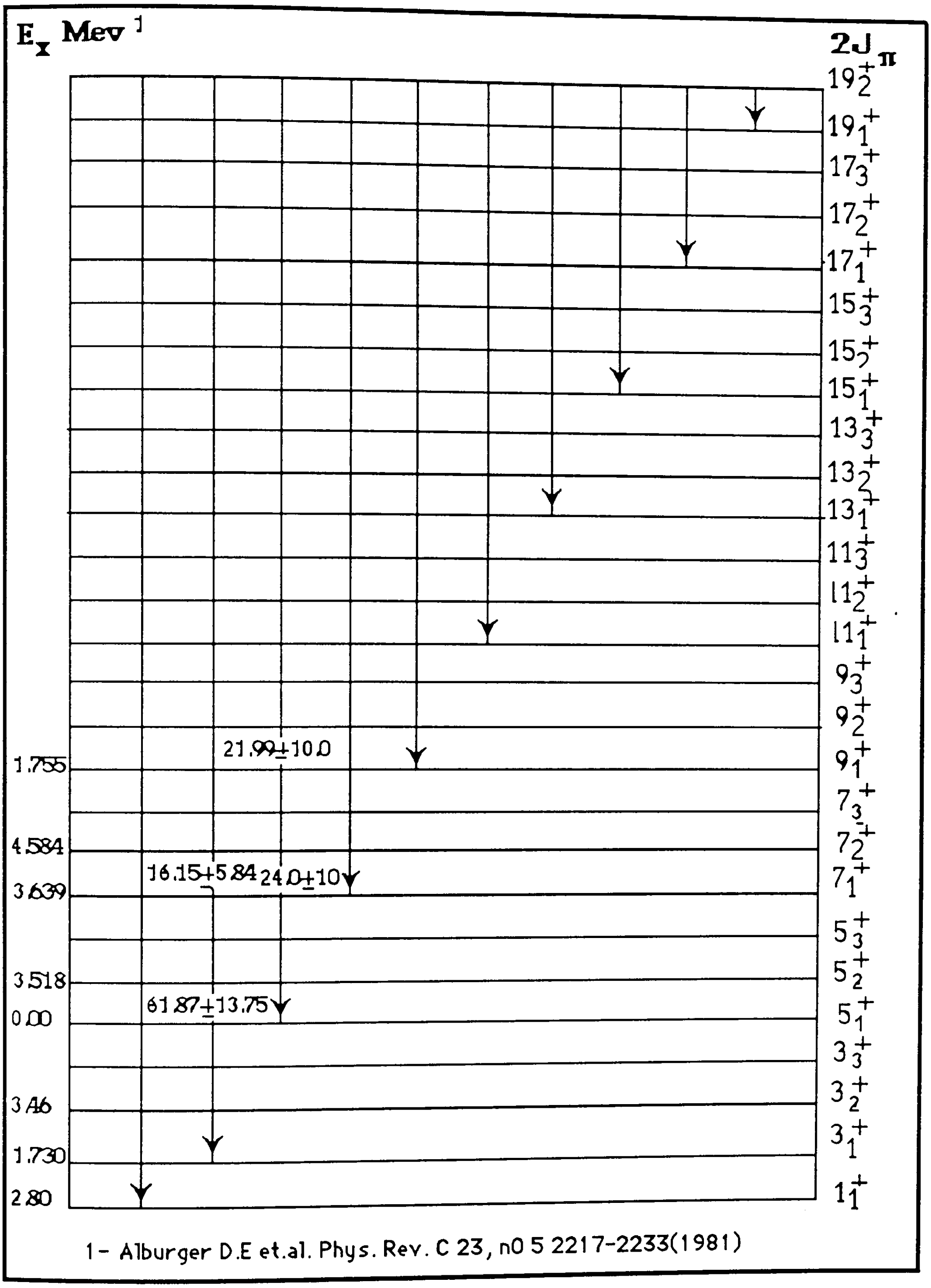


Fig 5.31 Comparison of Expt. & Theo. results ^{21}Na (6.27 ->:20.27Mev)

Hofman A. et.al., Z. Phys. A 332 (1989) 289-304 2- Endt P.M. et. al Nucl Phys. A310 (19



1- Alburger D.E et.al. Phys. Rev. C 23, n0 5 2217-2233(1981)

2- Arima A. et al Nucl. Phys. A170 (1971) for BE2.

Fig 5.32 : B(E2) values expt., for 21F

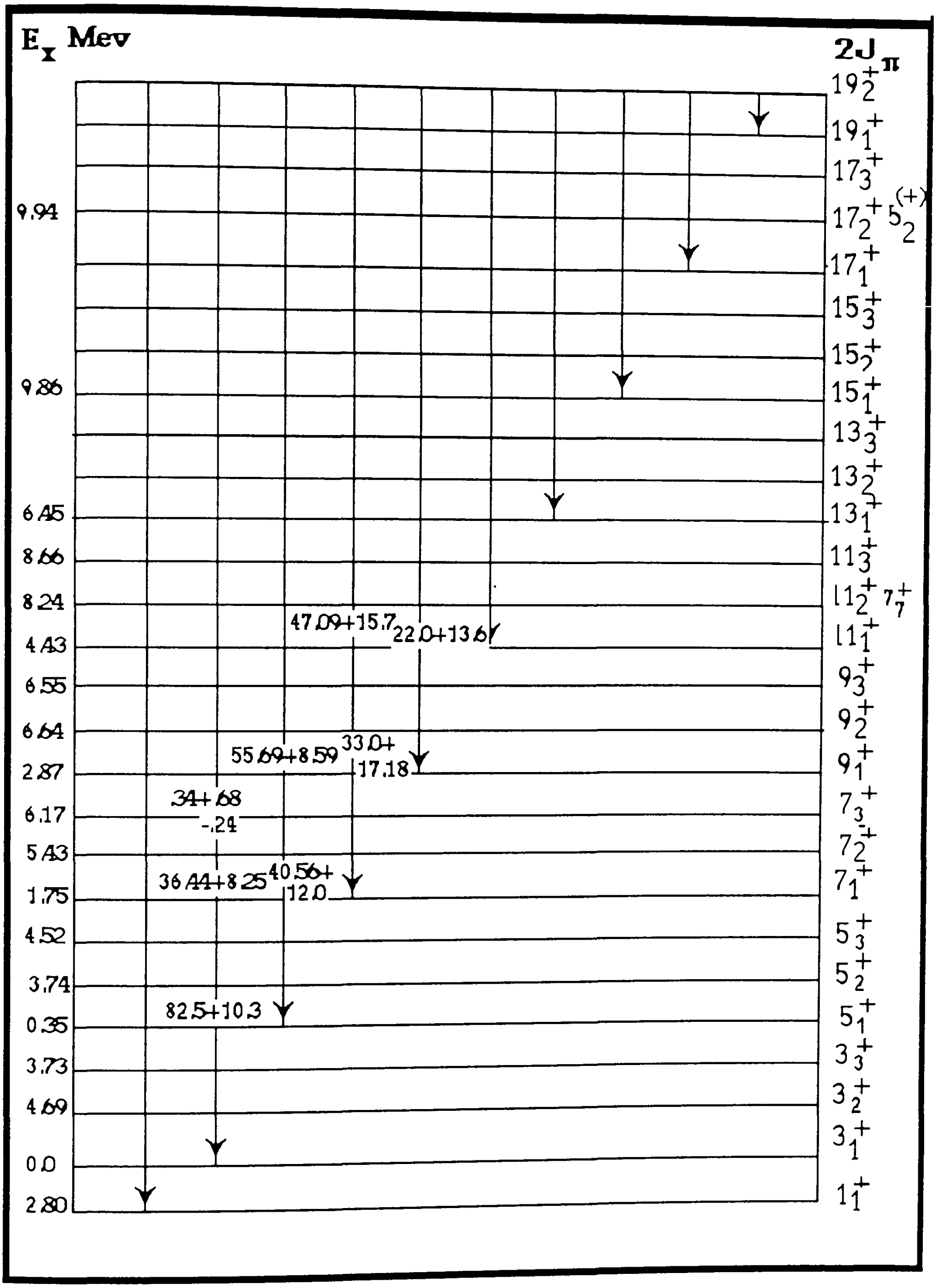


Fig 5.33 Electric Transitions B(E2) Vlaues Eperimental [44] for Ne^{21}

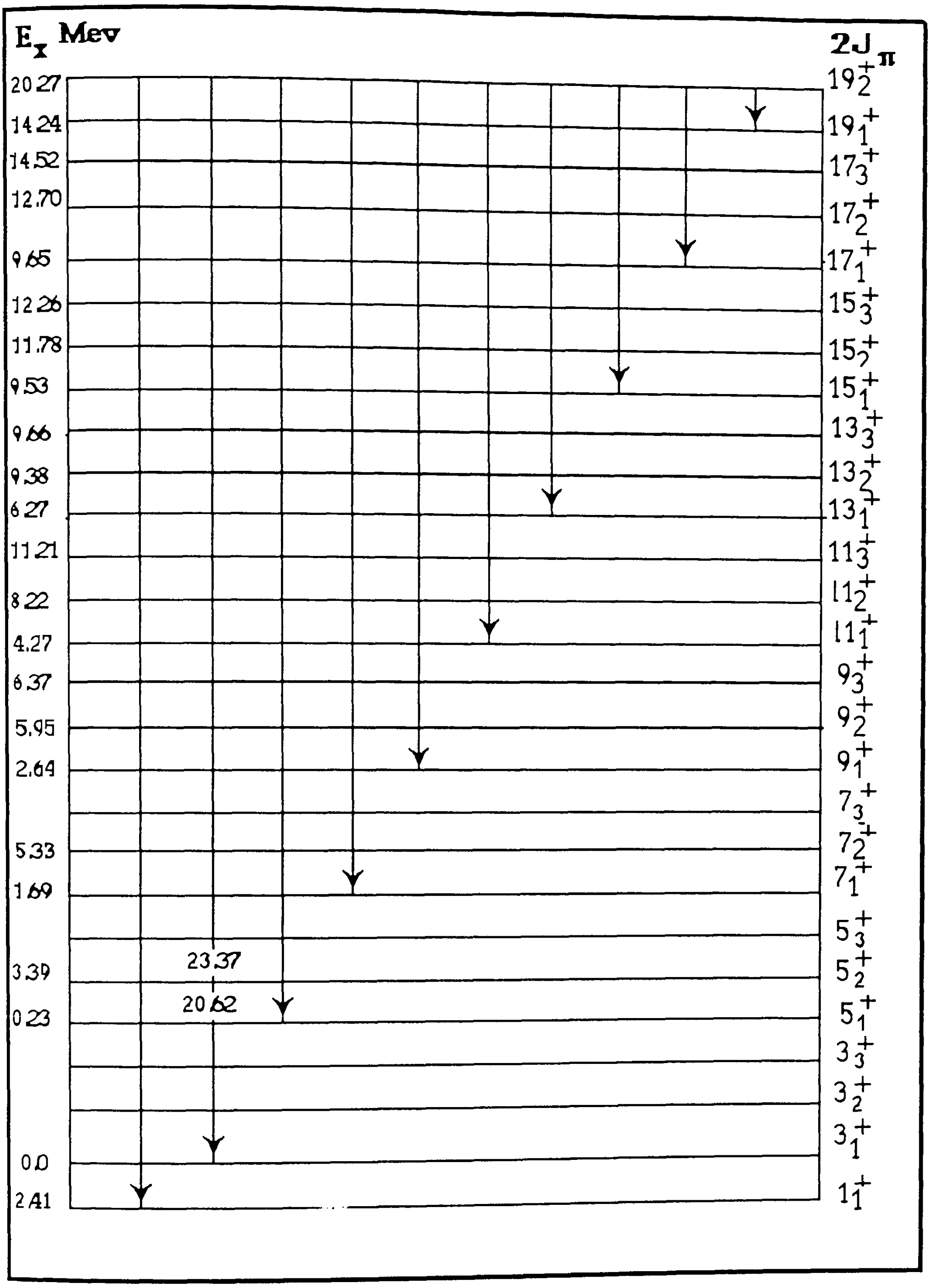


Fig 5.34 Expt.. Electric Transition rates, B(E2) values for ^{21}Na

A.Arima et. al Nucl. Phys. A170 (1971), results in e^2 xfm^4

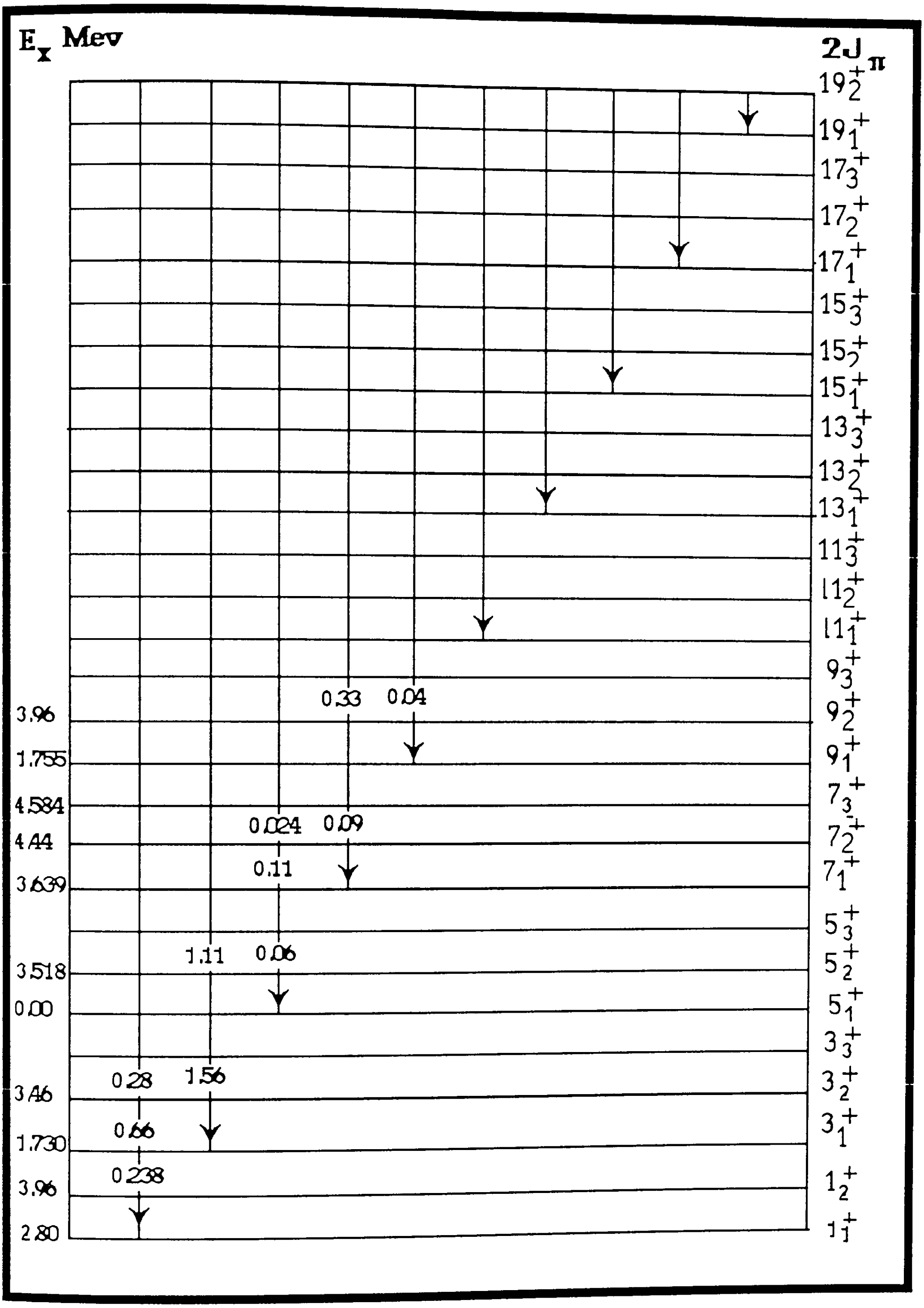


Fig 5.35 Expt. Magnetic Transitions, B(M1) values for ^{21}F

* Energy and B(MI) values taken from Alburger D.E. et.al Phys. rev. C23 1981

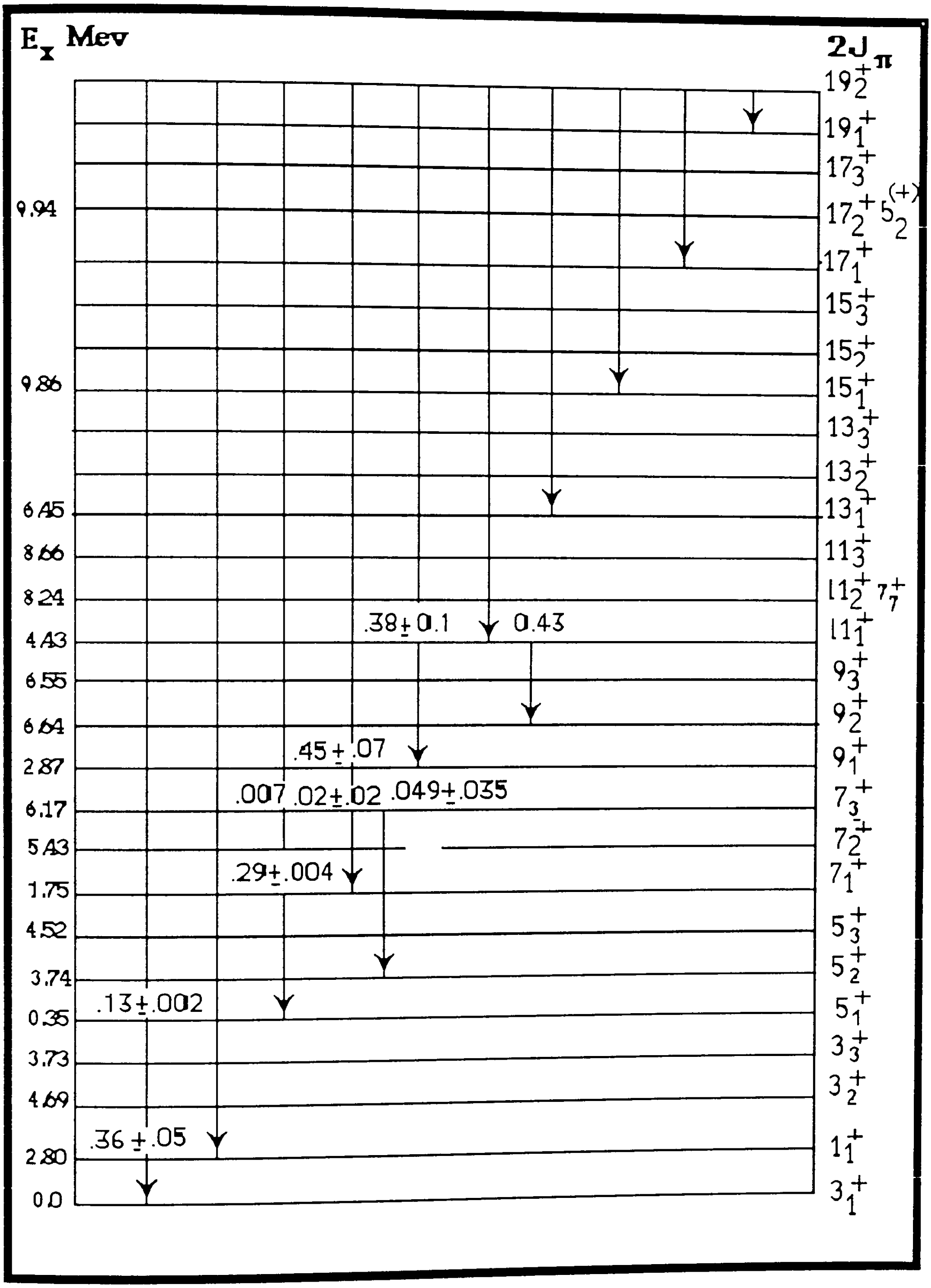


Fig 5.36 Magnet Transitions B(M1) Values Eperimental [**] for Ne^{21}

** Andritsopoulos catford

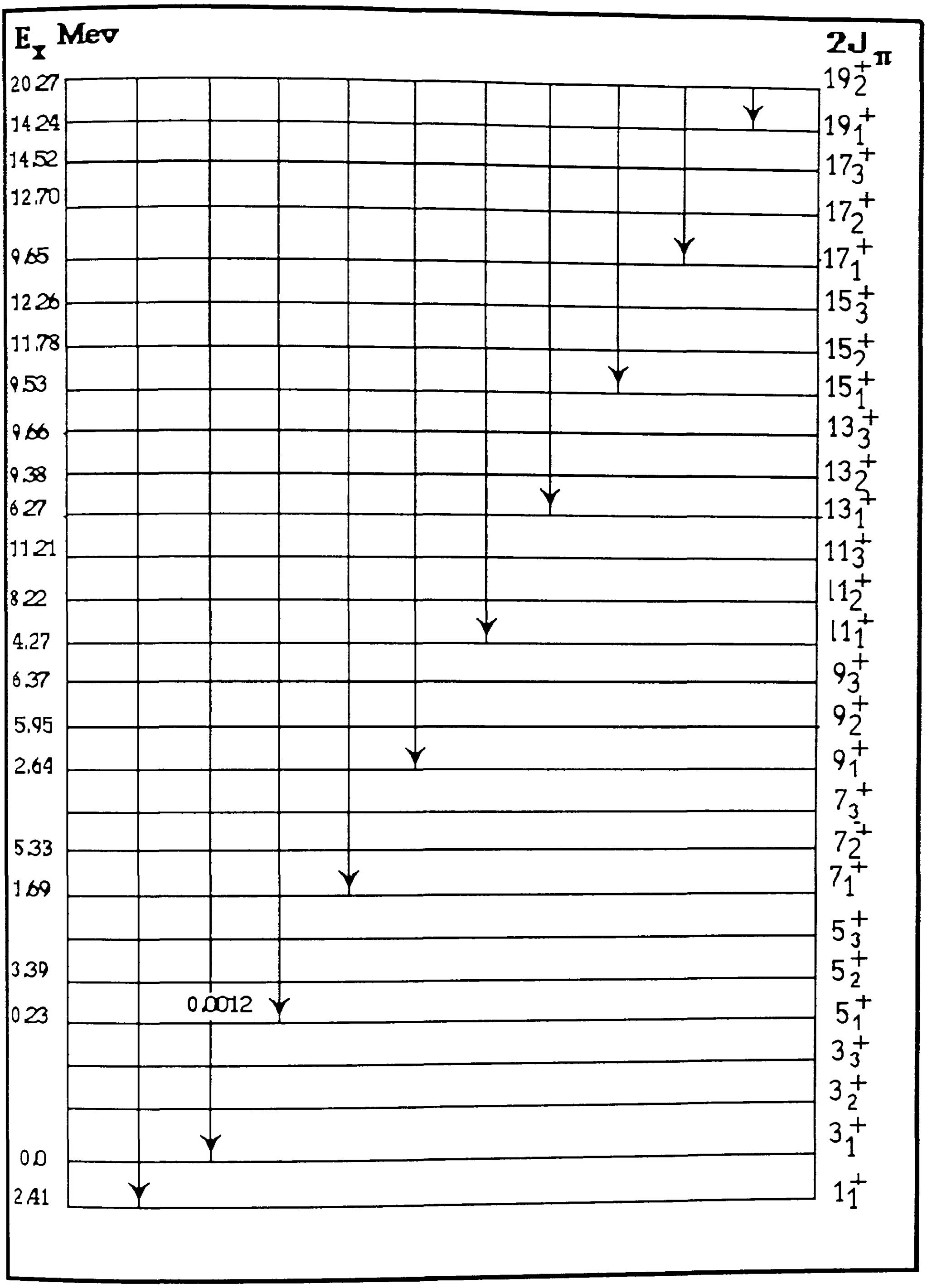


Fig 5.37 Expt. Magnetic Transition rates B(M1) values

Results in $e^2 \times fm^2$, for ref. see fig 5.34

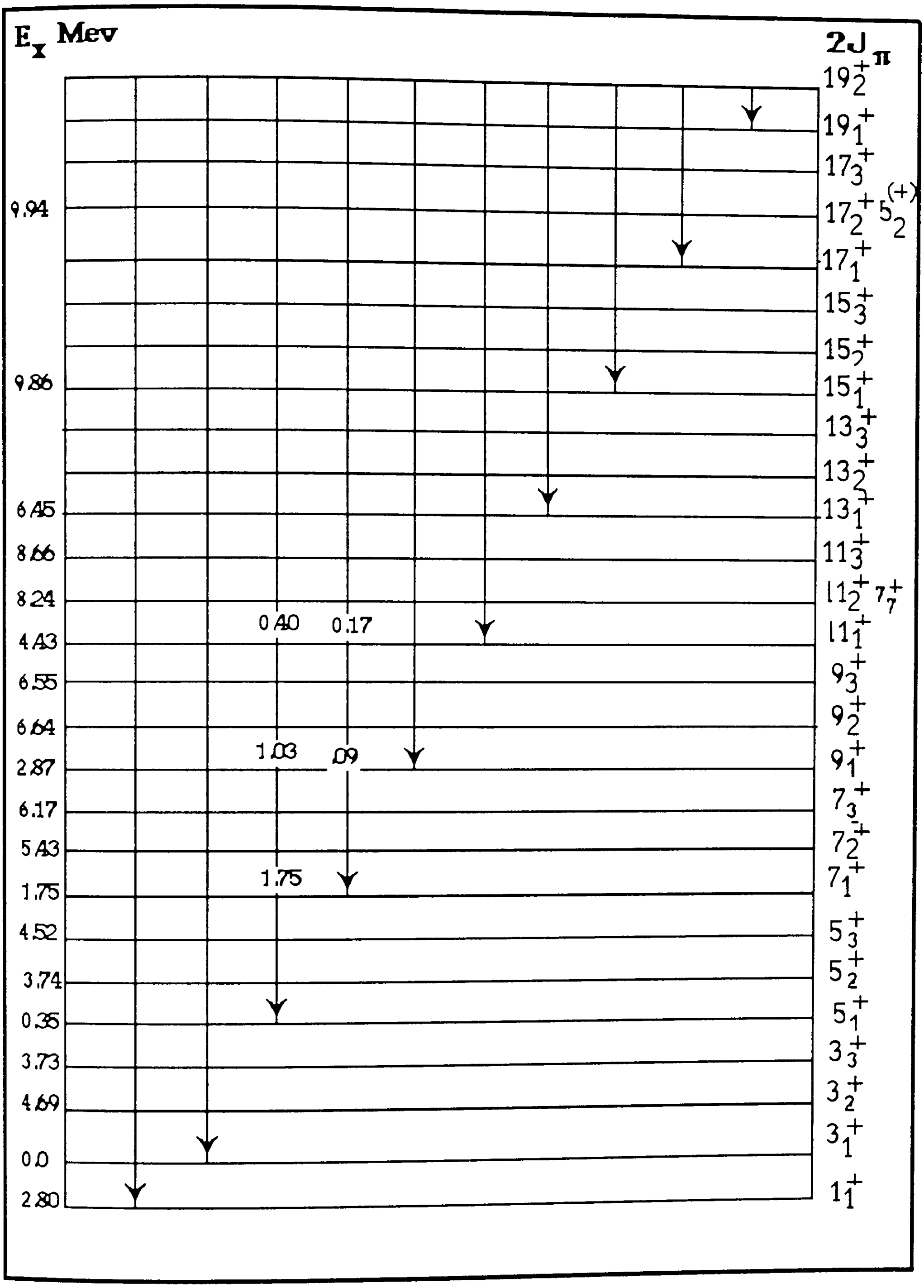


Fig. 5.38 Comaprison of expt. and theo. transition rates by sigma values for Ne^{21} sigma = ($(BE2)_{xpt}$ - $(BE2)_{th}$) / std. deviation

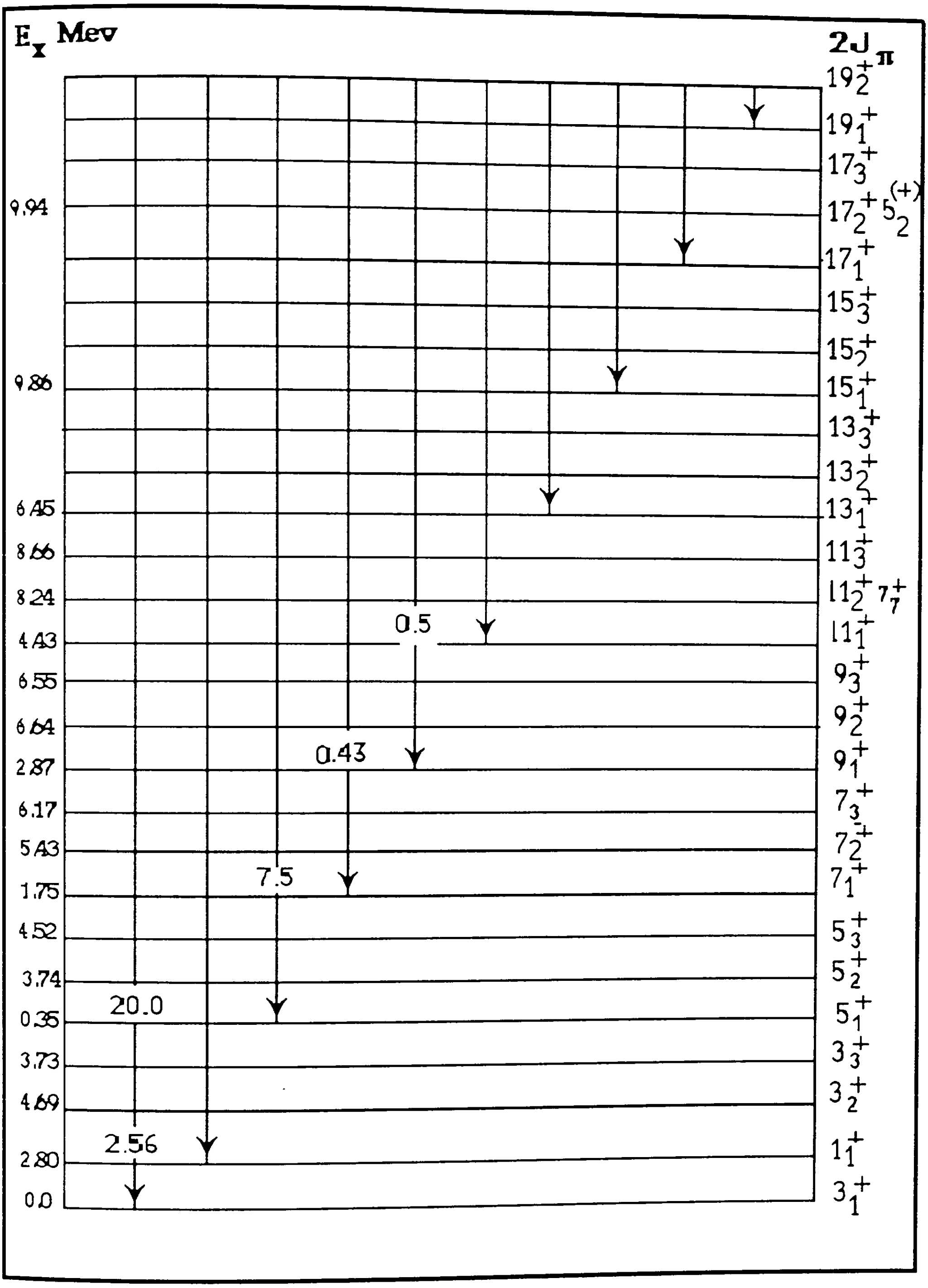


Fig 5.39 Comparison of Expt. and Theo. magnetic transition rates by sigma values for Ne^{21}, sigma = ((BM1)$_{xpt}$ - (B M1)$_{th}$) / std. deviation

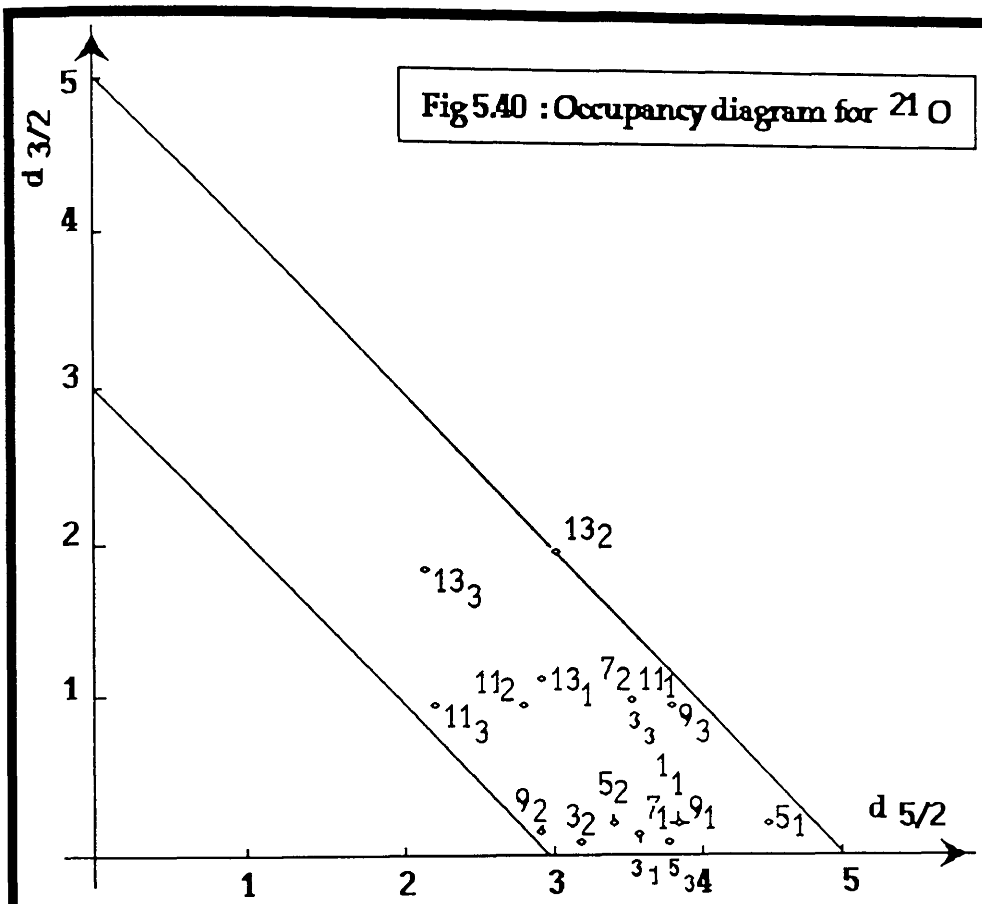


Fig 5.40 : Occupancy diagram for ^{21}O

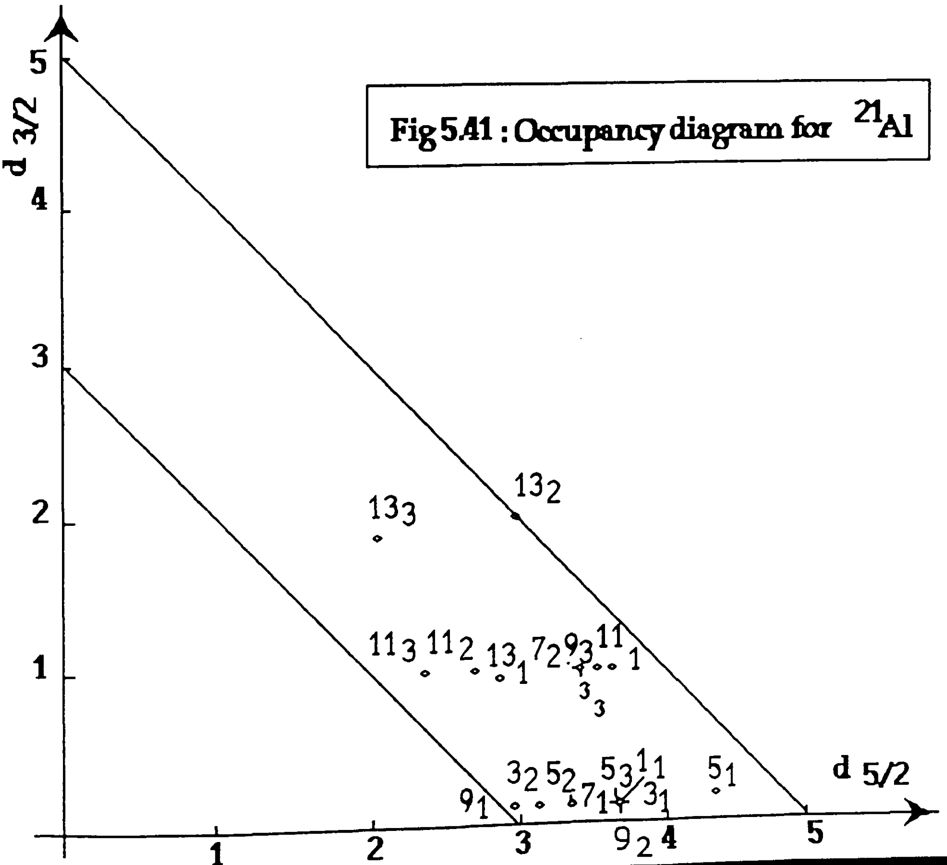


Fig 5.41 : Occupancy diagram for ^{21}Al

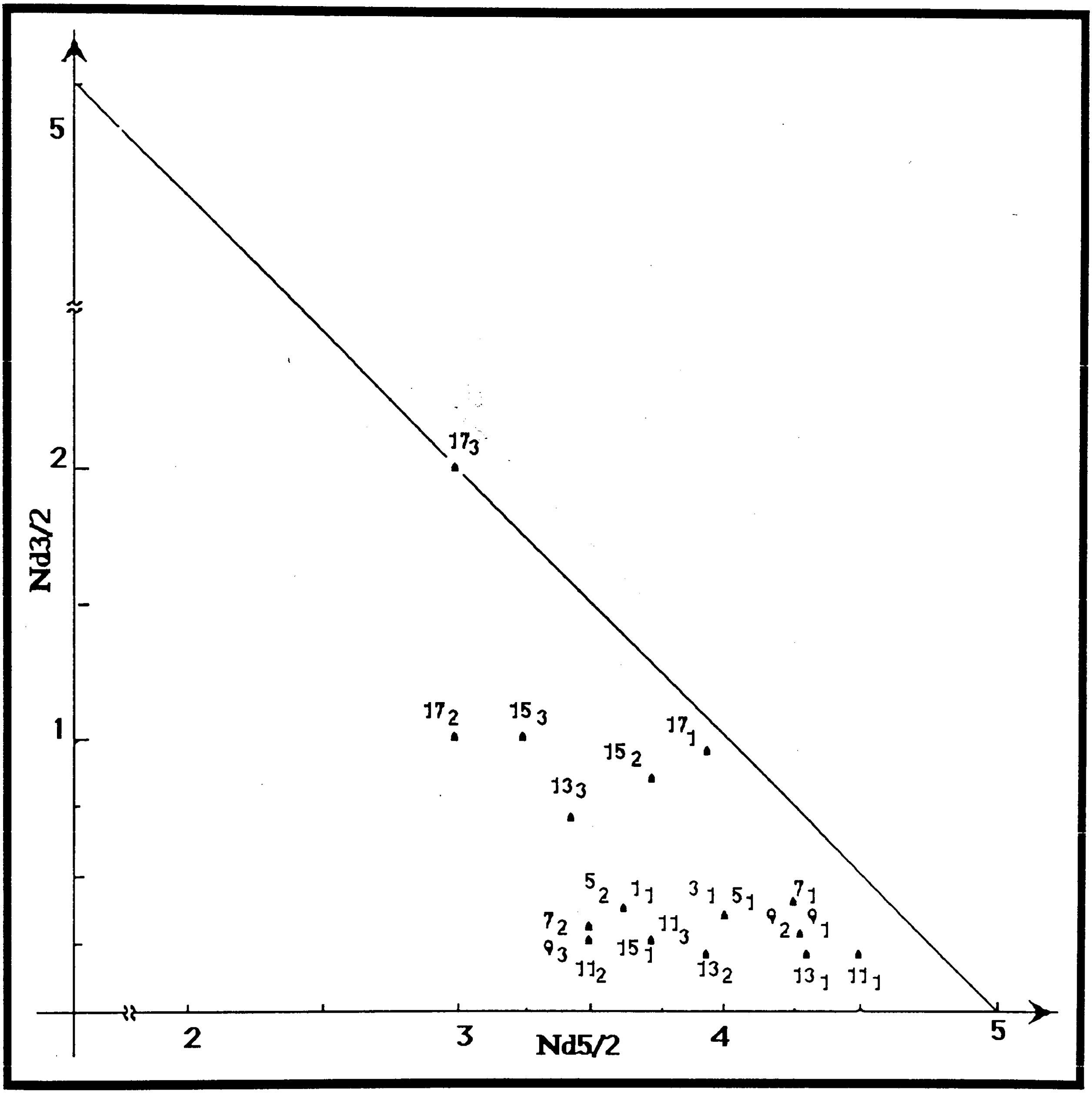


Fig 5.42 : Occupancy diagram for 21F

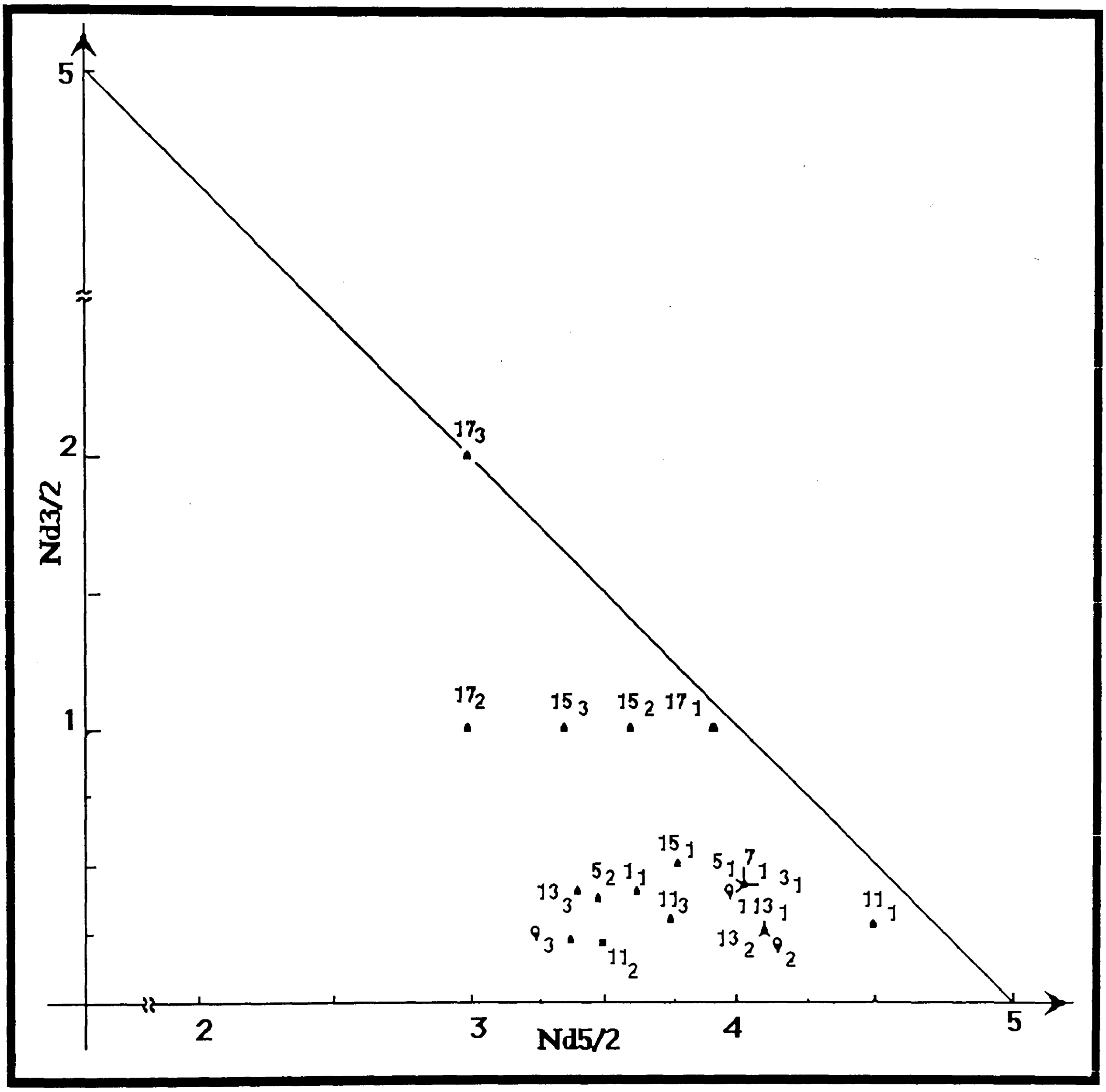


Fig 5.43: Occupancy diagram for 21Mg

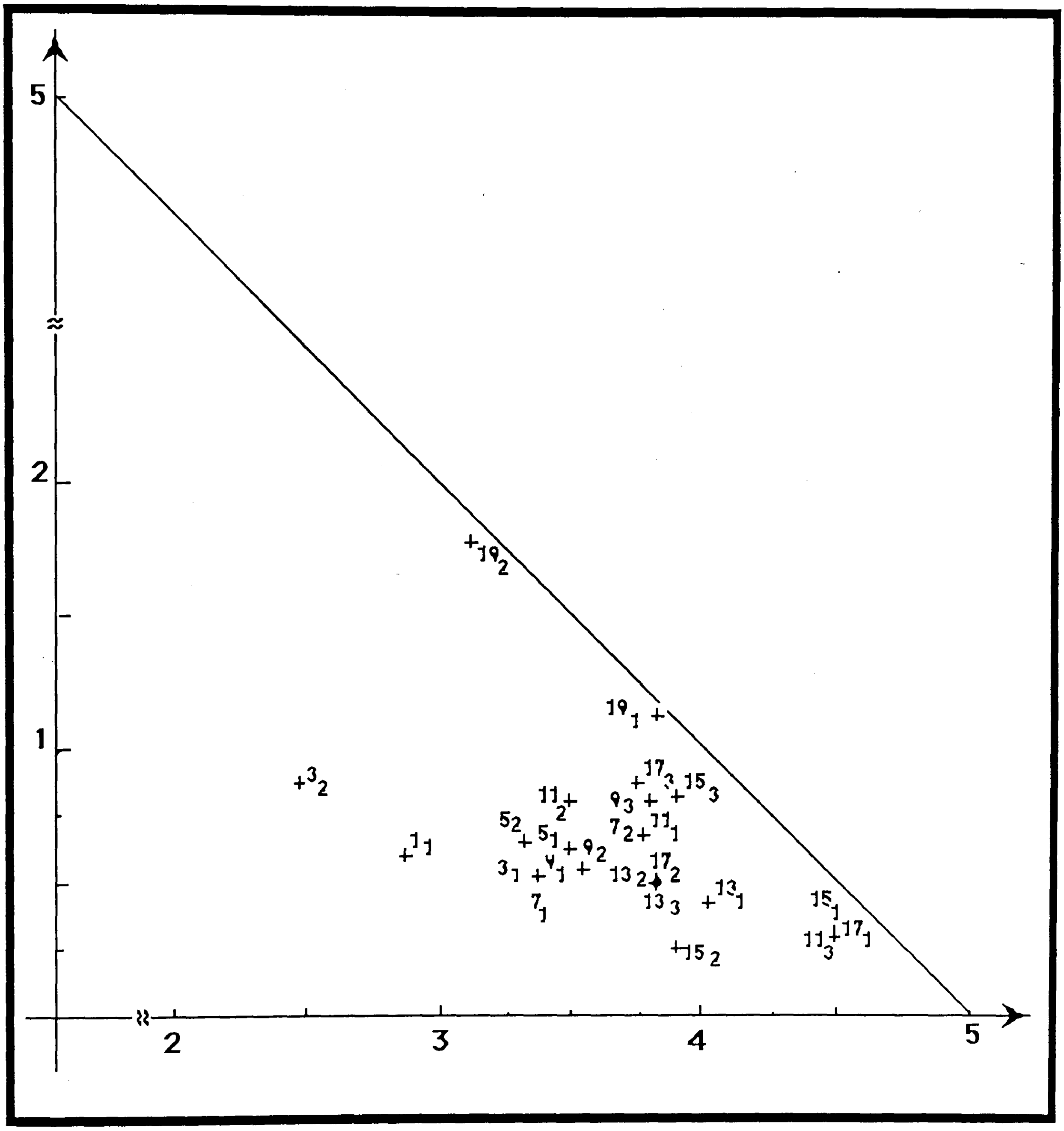


Fig 5.44 : Occupancy diagram for 21Ne

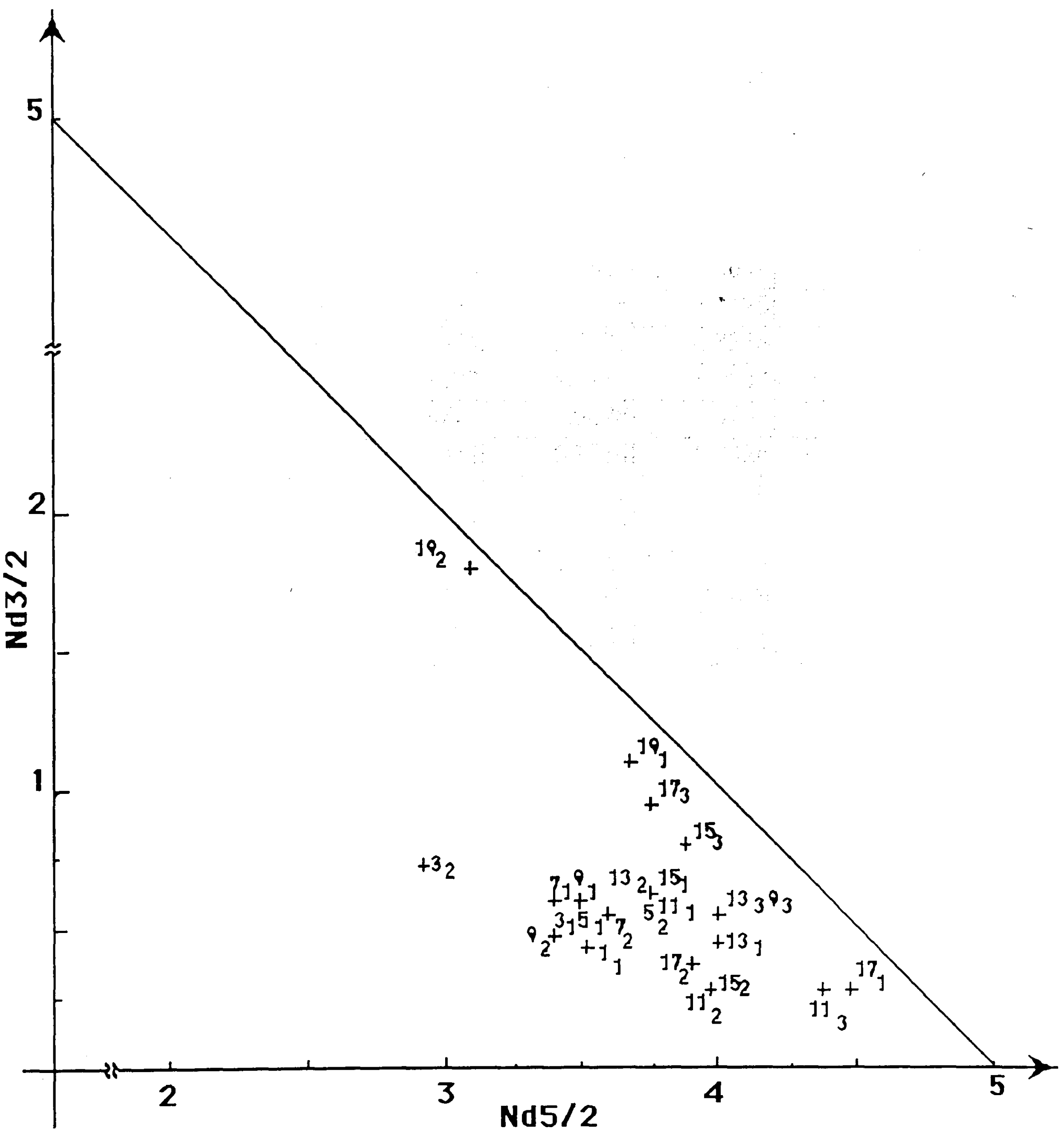


Fig. 5.45 : Occupancy Diagram for 21Na

	A	B	C	D	E	F	G
1	2*SPIN	Energy	Energy	Energy	Energy	Epsilon=	Epsilon
2	CWC	CWC	CWC	w.r.t. grnd	w.r.t. grnd	(E1-E2)/	%
3		E1 : O21	E2 : Al21	O21	Al21	(E1+E2)	
4	13p1	-16.4785	4.4176	11.1595	10.2748	0.0413	4.1275
5	13p2	-13.4191	8.3081	14.2189	14.1653	0.0019	0.1888
6	13p3	-9.6837	11.3496	17.9543	17.2068	0.0213	2.1259
7	11p1	-20.2066	1.4004	7.4314	7.2576	0.0118	1.1832
8	11p2	-18.4161	2.7571	9.2219	8.6143	0.0341	3.4066
9	11p3	-17.0578	3.7868	10.5802	9.644	0.0463	4.6291
10	9p1	-22.6245	-1.6097	5.0135	4.2475	0.0827	8.2712
11	9p2	-22.2198	-1.5854	5.4182	4.2718	0.1183	11.8308
12	9p3	-19.5112	2.1472	8.1268	8.0044	0.0076	0.7588
13	7p1	-24.6902	-3.3779	2.9478	2.4793	0.0863	8.6326
14	7p2	-21.4796	0.3854	6.1584	6.2426	0.0068	0.6790
15	5p1	-27.638	-5.8572	0	0		
16	5p2	-24.654	-3.4384	2.984	2.4188	0.1046	10.4612
17	5p3	-23.0098	-1.8081	4.6282	4.0491	0.0667	6.6737
18	3p1	-25.6559	-4.2755	1.9821	1.5817	0.1124	11.2352
19	3p2	-22.7322	-1.7141	4.9058	4.1431	0.0843	8.4286
20	3p3	-22.1668	-0.1749	5.4712	5.6823	0.0189	1.8927
21	1p1	-26.1624	-4.5349	1.4756	1.3223	0.0548	5.4791
22							
23		*	*********	*********	*********	*	
24		*	Ground ...	Engy O21	= -27.638	*	
25		*	Ground ...	Engy Al21	= -5.8572	*	
26		*	*********	*********	*********	*	

table 5.10: Energy comparison & Epsilon values for mirror nuclei O21_Al21

	A	B	C	D	E	F	G
1	2*SPIN	Energy	Energy	Energy	Energy	Epsilon=	Epsilon
2	CWC	CWC	CWC	w.r.t. grnd	w.r.t. grnd	(E1-E2)/	%
3	CWC	E1 : F21	E2: Mg21	E1 : F21	E2: Mg21	(E1+E2)	
4	17p1	-24.3557	-11.4792	10.5724	10.1542	0.0202	2.0177
5	17p2	-21.236	-8.906	13.6921	12.7274	0.0365	3.6515
6	17p3	-17.7401	-4.6951	17.188	16.9383	0.0073	0.7317
7	15p1	-26.1961	-13.895	8.732	7.7384	0.0603	6.0326
8	15p2	-23.5053	-10.7539	11.4228	10.8795	0.0244	2.4361
9	15p3	-22.3531	-9.6587	12.575	11.9747	0.0245	2.4452
10	13p1	-30.2889	-17.5265	4.6392	4.1069	0.0609	6.0861
11	13p2	-27.8345	-15.1264	7.0936	6.507	0.0431	4.3130
12	13p3	-26.688	-14.0298	8.2401	7.6036	0.0402	4.0174
13	11p1	-30.1451	-17.2003	4.783	4.4331	0.0380	3.7966
14	11p2	-28.7141	-16.0898	6.214	5.5436	0.0570	5.7018
15	11p3	-27.3052	-14.6621	7.6229	6.9713	0.0446	4.4648
16	9p1	-33.1344	-20.0582	1.7937	1.5752	0.0649	6.4858
17	9p2	-31.3278	-18.4287	3.6003	3.2047	0.0581	5.8134
18	9p3	-29.5876	-16.9366	5.3405	4.6968	0.0641	6.4131
19	7p1	-31.5155	-18.4924	3.4126	3.141	0.0414	4.1443
20	7p2	-29.2046	-17.6284	5.7235	4.005	0.1766	17.6646
21	5p1	-34.9281	-21.6334	0	0		
22	5p2	-31.6352	-18.7479	3.2929	2.8855	0.0659	6.5939
23	3p1	-33.1758	-19.9899	1.7523	1.6435	0.0320	3.2040
24	1p1	-34.7418	-21.4515	0.1863	0.1819	0.0120	1.1950
25							
26		*	**********	**********	**********	*	
27		*	Ground St.	Engy F21	-34.9281	*	
28		*	Ground St.	Engy Mg21	-21.6334	*	
29		*	**********	**********	**********	*	

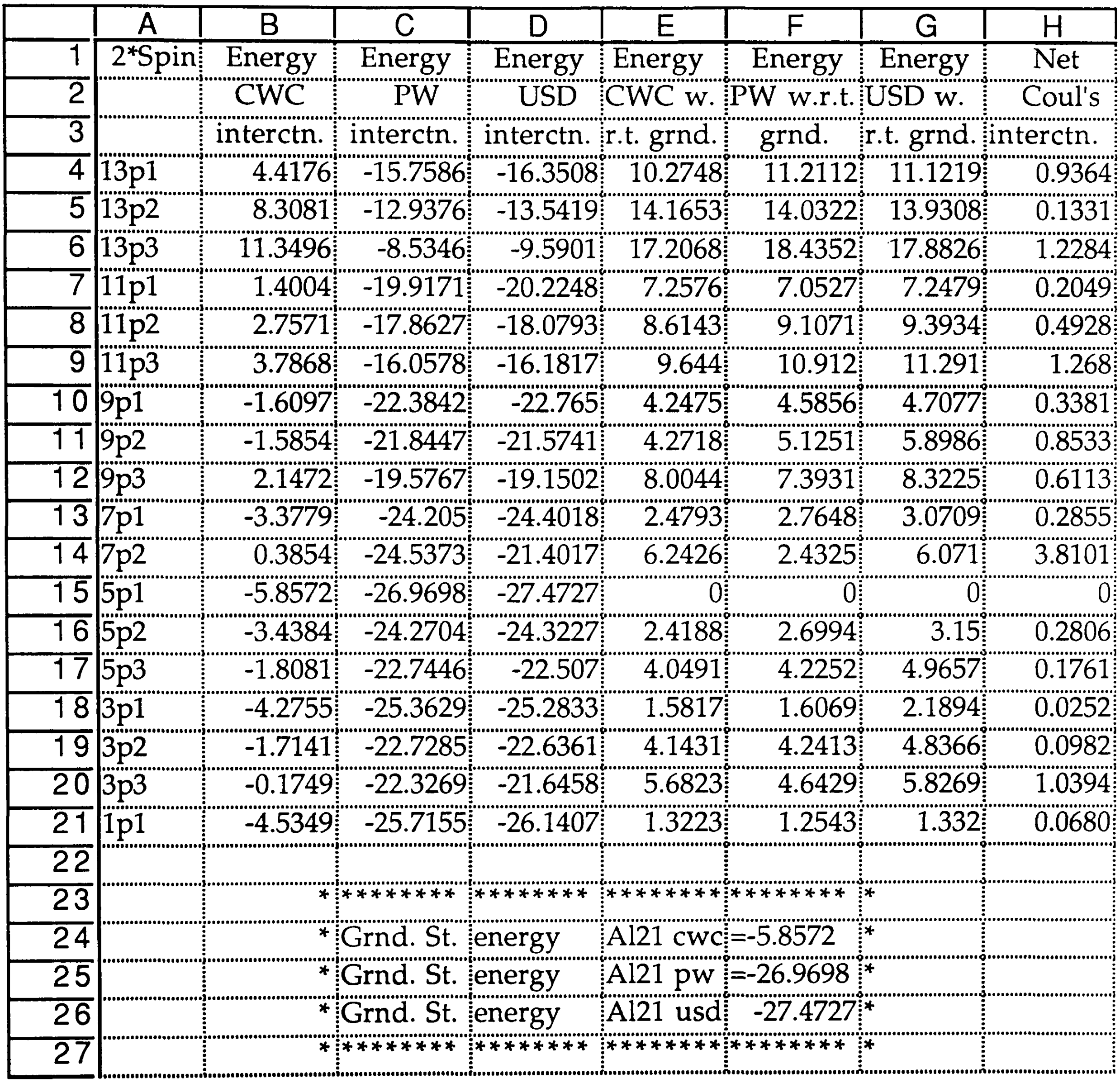
table 5.11: Energy Comparison & Epsilon values for mirror nuclei F21-Mg21

	A	B	C	D	E	F	G
1	2*SPIN	Energy	Energy	Energy	Energy	Epsilon=	Epsilon
2	CWC			w.r.t. grnd	w.r.t. grnd	(E1-E2)/	%
3		E1 : Ne21	E2 : Na21	Ne21	Na21	(E2+E1)	
4	19p1	-25.4331	-21.1054	14.269	14.243	0.00091	0.0912
5	19p2	-19.3473	-15.0729	20.3548	20.2755	0.00195	0.1952
6	17p1	-29.9257	-25.6959	9.7764	9.6525	0.00638	0.6377
7	17p2	-26.5704	-22.6461	13.1317	12.7023	0.01662	1.6622
8	17p3	-24.9884	-20.8239	14.7137	14.5245	0.00647	0.6471
9	15p1	-29.9908	-25.8179	9.7113	9.5305	0.00940	0.9396
10	15p2	-27.9014	-23.5689	11.8007	11.7795	0.00090	0.0899
11	15p3	-27.4565	-23.0836	12.2456	12.2648	0.00078	0.0783
12	13p1	-33.3979	-29.0743	6.3042	6.2741	0.00239	0.2393
13	13p2	-30.3048	-25.9614	9.3973	9.387	0.00055	0.0548
14	13p3	-29.423	-25.3825	10.2791	9.9659	0.01547	1.5470
15	11p1	-35.3876	-31.0778	4.3145	4.2706	0.00511	0.5114
16	11p2	-31.4904	-27.119	8.2117	8.2294	0.00108	0.1077
17	11p3	-30.2699	-24.1385	9.4322	11.2099	0.08612	8.6120
18	9p1	-36.8997	-32.7051	2.8024	2.6433	0.02922	2.9216
19	9p2	-33.4099	-29.3919	6.2922	5.9565	0.02741	2.7407
20	9p3	-33.3025	-28.9759	6.3996	6.3725	0.00212	0.2122
21	7p1	-37.973	-33.6554	1.7291	1.693	0.01055	1.0549
22	7p2	-34.3016	-30.0118	5.4005	5.3366	0.00595	0.5951
23	5p1	-39.4125	-35.1181	0.2896	0.2303	0.11406	11.4060
24	5p2	-36.0769	-31.9487	3.6252	3.3997	0.03210	3.2100
25	3p1	-39.7021	-35.3484	0	0		
26	1p1	-36.9036	-32.9332	2.7985	2.4152	0.07352	7.3518
27							
28		*	*********	**********	**********	********	*
29		*	Ground St.	energy for	ne21 is=	-39.7021	*
30		*	Ground St.	energy for	na21 is=	-35.3484	*
31		*	*********	**********	**********	********	*

table 5.12 : Energy values & Epsilon values for mirror nuclei Ne21-Na21

	A	B	C	D	E	F	G	H
1	2*Spin	Energy	Energy	Energy	Energy	Energy	Energy	Net
2		CWC	PW	USD	CWC w.	PW w.r.t.	USD w.	Coul's
3		interctn.	interctn.	interctn.	r.t. grnd.	grnd.	r.t. grnd.	interctn.
4	13p1	-16.4785	-15.7586	-16.3508	11.1595	11.2112	11.1219	0.0517
5	13p2	-13.4191	-12.9376	-13.5419	14.2189	14.0322	13.9308	0.1867
6	13p3	-9.6837	-8.5346	-9.5901	17.9543	18.4352	17.8826	0.4809
7	11p1	-20.2066	-19.9171	-20.2248	7.4314	7.0527	7.2479	0.3787
8	11p2	-18.4161	-17.8627	-18.0793	9.2219	9.1071	9.3934	0.1148
9	11p3	-17.0578	-16.0578	-16.1817	10.5802	10.912	11.291	0.3318
10	9p1	-22.6245	-22.3842	-22.765	5.0135	4.5856	4.7077	0.4279
11	9p2	-22.2198	-21.8447	-21.5741	5.4182	5.1251	5.8986	0.2931
12	9p3	-19.5112	-19.5767	-19.1502	8.1268	7.3931	8.3225	0.7337
13	7p1	-24.6902	-24.205	-24.4018	2.9478	2.7648	3.0709	0.1830
14	7p2	-21.4796	-24.5373	-21.4017	6.1584	2.4325	6.071	3.7259
15	5p1	-27.638	-26.9698	-27.4727	0	0	0	0.0000
16	5p2	-24.654	-24.2704	-24.3227	2.984	2.6994	3.15	0.2846
17	5p3	-23.0098	-22.7446	-22.507	4.6282	4.2252	4.9657	0.4030
18	3p1	-25.6559	-25.3629	-25.2833	1.9821	1.6069	2.1894	0.3752
19	3p2	-22.7322	-22.7285	-22.6361	4.9058	4.2413	4.8366	0.6645
20	3p3	-22.1668	-22.3269	-21.6458	5.4712	4.6429	5.8269	0.8283
21	1p1	-26.1624	-25.7155	-26.1407	1.4756	1.2543	1.332	0.2213
22								
23		*	********	********	*******	********	*	
24		*	Grnd. St.	energy	O21 cwc	-27.638	*	
25		*	Grnd. St.	energy	O21 pw	-26.9698	*	
26		*	Grnd. St.	energy	O21 usd	-27.4727	*	
27		*	********	********	*******	********	*	

table 5.13 : Comparison of CWC-PW-USD energy values for O21



	A	B	C	D	E	F	G	H
1	2*Spin	Energy	Energy	Energy	Energy	Energy	Energy	Net
2		CWC	PW	USD	CWC w.	PW w.r.t.	USD w.	Coul's
3		interctn.	interctn.	interctn.	r.t. grnd.	grnd.	r.t. grnd.	interctn.
4	13p1	4.4176	-15.7586	-16.3508	10.2748	11.2112	11.1219	0.9364
5	13p2	8.3081	-12.9376	-13.5419	14.1653	14.0322	13.9308	0.1331
6	13p3	11.3496	-8.5346	-9.5901	17.2068	18.4352	17.8826	1.2284
7	11p1	1.4004	-19.9171	-20.2248	7.2576	7.0527	7.2479	0.2049
8	11p2	2.7571	-17.8627	-18.0793	8.6143	9.1071	9.3934	0.4928
9	11p3	3.7868	-16.0578	-16.1817	9.644	10.912	11.291	1.268
10	9p1	-1.6097	-22.3842	-22.765	4.2475	4.5856	4.7077	0.3381
11	9p2	-1.5854	-21.8447	-21.5741	4.2718	5.1251	5.8986	0.8533
12	9p3	2.1472	-19.5767	-19.1502	8.0044	7.3931	8.3225	0.6113
13	7p1	-3.3779	-24.205	-24.4018	2.4793	2.7648	3.0709	0.2855
14	7p2	0.3854	-24.5373	-21.4017	6.2426	2.4325	6.071	3.8101
15	5p1	-5.8572	-26.9698	-27.4727	0	0	0	0
16	5p2	-3.4384	-24.2704	-24.3227	2.4188	2.6994	3.15	0.2806
17	5p3	-1.8081	-22.7446	-22.507	4.0491	4.2252	4.9657	0.1761
18	3p1	-4.2755	-25.3629	-25.2833	1.5817	1.6069	2.1894	0.0252
19	3p2	-1.7141	-22.7285	-22.6361	4.1431	4.2413	4.8366	0.0982
20	3p3	-0.1749	-22.3269	-21.6458	5.6823	4.6429	5.8269	1.0394
21	1p1	-4.5349	-25.7155	-26.1407	1.3223	1.2543	1.332	0.0680
22								
23		*	********	********	********	********	*	
24		*	Grnd. St.	energy	Al21 cwc	=-5.8572	*	
25		*	Grnd. St.	energy	Al21 pw	=-26.9698	*	
26		*	Grnd. St.	energy	Al21 usd	-27.4727	*	
27		*	********	********	********	********	*	

table 5.14 Comaprison of CWC-PW-USD energy values for Al21

	A	B	C	D	E	F	G	H
1	2*Spin	Energy	Energy	Energy	Energy	Energy	Energy	Net
2		CWC	PW	USD	CWC w.	PW w.r.t.	USD w.	Coul's
3		interctn.	interctn.	interctn.	r.t. grnd.	grnd.	r.t. grnd.	interctn.
4	17p1	-24.3557	-27.907	-28.1754	10.5724	10.5517	10.3358	0.0207
5	17p2	-21.236	-24.5293	-24.9162	13.6921	13.9294	13.595	0.2373
6	17p3	-17.7401	-20.9365	-21.6835	17.188	17.5222	16.8277	0.3342
7	15p1	-26.1961	-29.5909	-29.7615	8.732	8.8678	8.7497	0.1358
8	15p2	-23.5053	-27.2659	-27.0823	11.4228	11.1928	11.4289	0.2300
9	15p3	-22.3531	-25.7707	-26.3243	12.575	12.688	12.1869	0.1130
10	13p1	-30.2889	-33.6125	-33.6287	4.6392	4.8462	4.8825	0.2070
11	13p2	-27.8345	-31.3683	-31.5822	7.0936	7.0904	6.929	0.0032
12	13p3	-26.688	-30.507	-30.4128	8.2401	7.9517	8.0984	0.2884
13	11p1	-30.1451	-33.5584	-33.6953	4.783	4.9003	4.8159	0.1173
14	11p2	-28.7141	-32.3971	-32.3153	6.214	6.0616	6.1959	0.1524
15	11p3	-27.3052	-30.938	-30.6949	7.6229	7.5207	7.8163	0.1022
16	9p1	-33.1344	-36.5495	-36.6671	1.7937	1.9092	1.8441	0.1155
17	9p2	-31.3278	-34.7837	-34.8192	3.6003	3.675	3.692	0.0747
18	9p3	-29.5876	-33.3497	-33.0202	5.3405	5.109	5.491	0.2315
19	7p1	-31.5155	-35.2698	-34.8993	3.4126	3.1889	3.6119	0.2237
20	7p2	-29.2046	-34.3345	-34.0841	5.7235	4.1242	4.4271	1.5993
21	5p1	-34.9281	-38.4587	-38.5112	0	0	0	0.0000
22	5p2	-31.6352	-35.2173	-34.8298	3.2929	3.2414	3.6814	0.0515
23	3p1	-33.1758	-36.9035	-36.6576	1.7523	1.5552	1.8536	0.1971
24	1p1	-34.7418	-38.151	-38.1911	0.1863	0.3077	0.3201	0.1214
25								
26		*	*********	*********	*********	*********	*	
27		*	Grnd. St.	energy	F21 cwc	-34.9281	*	
28		*	Grnd. St.	energy	F21 PW	-38.4587	*	
29		*	Grnd. St.	energy	F21 USD	-38.5112	*	
30		*	*********	*********	*********	*********	*	

table 5.15: Comparison of CWC-PW-USD energy values for F21

	A	B	C	D	E	F	G	H
1	2*Spin	Energy	Energy	Energy	Energy	Energy	Energy	Net
2		CWC	PW	USD	CWC w.	PW w.r.t.	USD w.	Coul's
3		interctn.	interctn.	interctn.	r.t. grnd.	grnd.	r.t. grnd.	interctn.
4	17p1	-11.4792	-27.907	-28.1754	10.1542	10.5517	10.3358	0.3975
5	17p2	-8.906	-24.5293	-24.9162	12.7274	13.9294	13.595	1.2020
6	17p3	-4.6951	-20.9365	-21.6835	16.9383	17.5222	16.8277	0.5839
7	15p1	-13.895	-29.5909	-29.7615	7.7384	8.8678	8.7497	1.1294
8	15p2	-10.7539	-27.2659	-27.0823	10.8795	11.1928	11.4289	0.3133
9	15p3	-9.6587	-25.7707	-26.3243	11.9747	12.688	12.1869	0.7133
10	13p1	-17.5265	-33.6125	-33.6287	4.1069	4.8462	4.8825	0.7393
11	13p2	-15.1264	-31.3683	-31.5822	6.507	7.0904	6.929	0.5834
12	13p3	-14.0298	-30.507	-30.4128	7.6036	7.9517	8.0984	0.3481
13	11p1	-17.2003	-33.5584	-33.6953	4.4331	4.9003	4.8159	0.4672
14	11p2	-16.0898	-32.3971	-32.3153	5.5436	6.0616	6.1959	0.5180
15	11p3	-14.6621	-30.938	-30.6949	6.9713	7.5207	7.8163	0.5494
16	9p1	-20.0582	-36.5495	-36.6671	1.5752	1.9092	1.8441	0.3340
17	9p2	-18.4287	-34.7837	-34.8192	3.2047	3.675	3.692	0.4703
18	9p3	-16.9366	-33.3497	-33.0202	4.6968	5.109	5.491	0.4122
19	7p1	-18.4924	-35.2698	-34.8993	3.141	3.1889	3.6119	0.0479
20	5p1	-21.6334	-38.4587	-38.5112	0	0	0	0.0000
21	5p2	-18.7479	-35.2173	-34.8298	2.8855	3.2414	3.6814	0.3559
22	3p1	-19.9899	-36.9035	-36.6576	1.6435	1.5552	1.8536	0.0883
23	1p1	-21.4515	-38.151	-38.1911	0.1819	0.3077	0.3201	0.1258
24								
25		*	********	********	********	********	*	
26		*	Grnd. St.	energy	Mg21 cwc	-21.6334	*	
27		*	Grnd. St.	energy	Mg21 pw	-38.4587	*	
28		*	Grnd. St.	energy	Mg21 usd	-38.5112	*	
29		*	********	********	********	********	*	

table 5.16: Comparison of CWC-PW-USD energy vales for mg21

	A	B	C	D	E	F	G	H
1	2*Spin	Energy	Energy	Energy	Energy	Energy	Energy	Net
2		CWC	PW	USD	CWC w.	PW w.r.t.	USD w.	Coul's
3		interctn.	interctn.	interctn.	r.t. grnd.	grnd.	r.t. grnd.	interctn.
4	19p1	-25.4331	-33.0652	-33.0786	14.269	14.387	14.1225	0.1180
5	19p2	-19.3473	-26.6171	-27.6011	20.3548	20.8351	19.6	0.4803
6	17p1	-29.9257	-37.2353	-37.4665	9.7764	10.2169	9.7346	0.4405
7	17p2	-26.5704	-33.9547	-34.221	13.1317	13.4975	12.9801	0.3658
8	17p3	-24.9884	-32.7747	-32.2455	14.7137	14.6775	14.9556	0.0362
9	15p1	-29.9908	-37.7921	-37.4361	9.7113	9.6601	9.765	0.0512
10	15p2	-27.9014	-35.2454	-35.7111	11.8007	12.2068	11.49	0.4061
11	15p3	-27.4565	-34.9505	-34.9192	12.2456	12.5017	12.2819	0.2561
12	13p1	-33.3979	-40.8967	-41.0208	6.3042	6.5555	6.1803	0.2513
13	13p2	-30.3048	-37.9862	-37.9698	9.3973	9.466	9.2313	0.0687
14	13p3	-29.423	-37.2079	-37.102	10.2791	10.2443	10.0991	0.0348
15	11p1	-35.3876	-43.0873	-42.7257	4.3145	4.3649	4.4754	0.0504
16	11p2	-31.4904	-39.009	-39.1618	8.2117	8.4432	8.0393	0.2315
17	11p3	-30.2699	-38.5291	-38.0986	9.4322	8.9231	9.1025	0.5091
18	9p1	-36.8997	-44.562	-44.3883	2.8024	2.8902	2.8128	0.0878
19	9p2	-33.4099	-41.2532	-41.0669	6.2922	6.199	6.1342	0.0932
20	9p3	-33.3025	-40.9657	-40.9858	6.3996	6.4865	6.2153	0.0869
21	7p1	-37.973	-45.6525	-45.4058	1.7291	1.7997	1.7953	0.0706
22	7p2	-34.3016	-42.1074	-41.8239	5.4005	5.3448	5.3772	0.0557
23	7p3	-31.4733	-40.3665	-40.9141	8.2288	7.0857	6.287	1.1431
24	5p1	-39.4125	-47.1385	-46.9533	0.2896	0.3137	0.2478	0.0241
25	5p2	-36.0769	-43.8606	-43.4698	3.6252	3.5916	3.7313	0.0336
26	3p1	-39.7021	-47.4522	-47.2011	0	0	0	0.0000
27	1p1	-36.9036	-44.6196	-43.6504	2.7985	2.8326	3.5507	0.0341
28								
29		*	********	*********	********	********	*	
30		*	Grnd. St.	energy	ne21 cwc	-39.7021	*	
31		*	Grnd. St.	energy	ne21 pw	-47.4522	*	
32		*	Grnd. St.	energy	ne21 usd	-47.2011	*	
33		*	********	*********	********	********	*	

table 5.17: Comparison of CWC-PW-USD energy values for Ne21

	A	B	C	D	E	F	G	H
1	2*Spin	Energy	Energy	Energy	Energy	Energy	Energy	Net
2		CWC	PW	USD	CWC w.	PW w.r.t.	USD w.	Coul's
3		interctn.	interctn.	interctn.	r.t. grnd.	grnd.	r.t. grnd.	interctn.
4	19p1	-21.1054	-33.0652	-33.0786	14.243	14.387	14.1225	0.1440
5	19p2	-15.0729	-26.6171	-27.6011	20.2755	20.8351	19.6	0.5596
6	17p1	-25.6959	-37.2353	-37.4665	9.6525	10.2169	9.7346	0.5644
7	17p2	-22.6461	-33.9547	-34.221	12.7023	13.4975	12.9801	0.7952
8	17p3	-20.8239	-32.7747	-32.2455	14.5245	14.6775	14.9556	0.1530
9	15p1	-25.8179	-37.7921	-37.4361	9.5305	9.6601	9.765	0.1296
10	15p2	-23.5689	-35.2453	-35.7111	11.7795	12.2069	11.49	0.4274
11	15p3	-23.0836	-34.9506	-34.9192	12.2648	12.5016	12.2819	0.2368
12	13p1	-29.0743	-40.8968	-41.0208	6.2741	6.5554	6.1803	0.2813
13	13p2	-25.9614	-37.9862	-37.9698	9.387	9.466	9.2313	0.0790
14	13p3	-25.3825	-37.2079	-37.102	9.9659	10.2443	10.0991	0.2784
15	11p1	-31.0778	-43.0873	-42.7257	4.2706	4.3649	4.4754	0.0943
16	11p2	-27.119	-39.009	-39.1618	8.2294	8.4432	8.0393	0.2138
17	11p3	-24.1385	-38.5291	-38.0986	11.2099	8.9231	9.1025	2.2868
18	9p1	-32.7051	-44.5621	-44.3883	2.6433	2.8901	2.8128	0.2468
19	9p2	-29.3919	-41.2533	-41.0669	5.9565	6.1989	6.1342	0.2424
20	9p3	-28.9759	-40.9657	-40.9858	6.3725	6.4865	6.2153	0.1140
21	7p1	-33.6554	-45.6525	-45.4058	1.693	1.7997	1.7953	0.1067
22	7p2	-30.0118	-42.1074	-41.8239	5.3366	5.3448	5.3772	0.0082
23	5p1	-35.1181	-47.1385	-40.9141	0.2303	0.3137	6.287	0.0834
24	5p2	-31.9487	-43.8608	-46.9533	3.3997	3.5914	0.2478	0.1917
25	5p3	-30.9873	-42.9983	-42.6256	4.3611	4.4539	4.5755	0.0928
26	3p1	-35.3484	-47.4522	-47.2011	0	0	0	0.0000
27	1p1	-32.9332	-44.6196	-43.6504	2.4152	2.8326	3.5507	0.4174
28								
29		*	********	********	********	********	*	
30		*	Grnd. St.	energy	na21 cwc	-35.3484	*	
31		*	Grnd. St.	energy	na21 pw	-47.4522	*	
32		*	Grnd. St.	energy	na21 usd	-47.2011	*	
33		*	********	********	********	********	*	

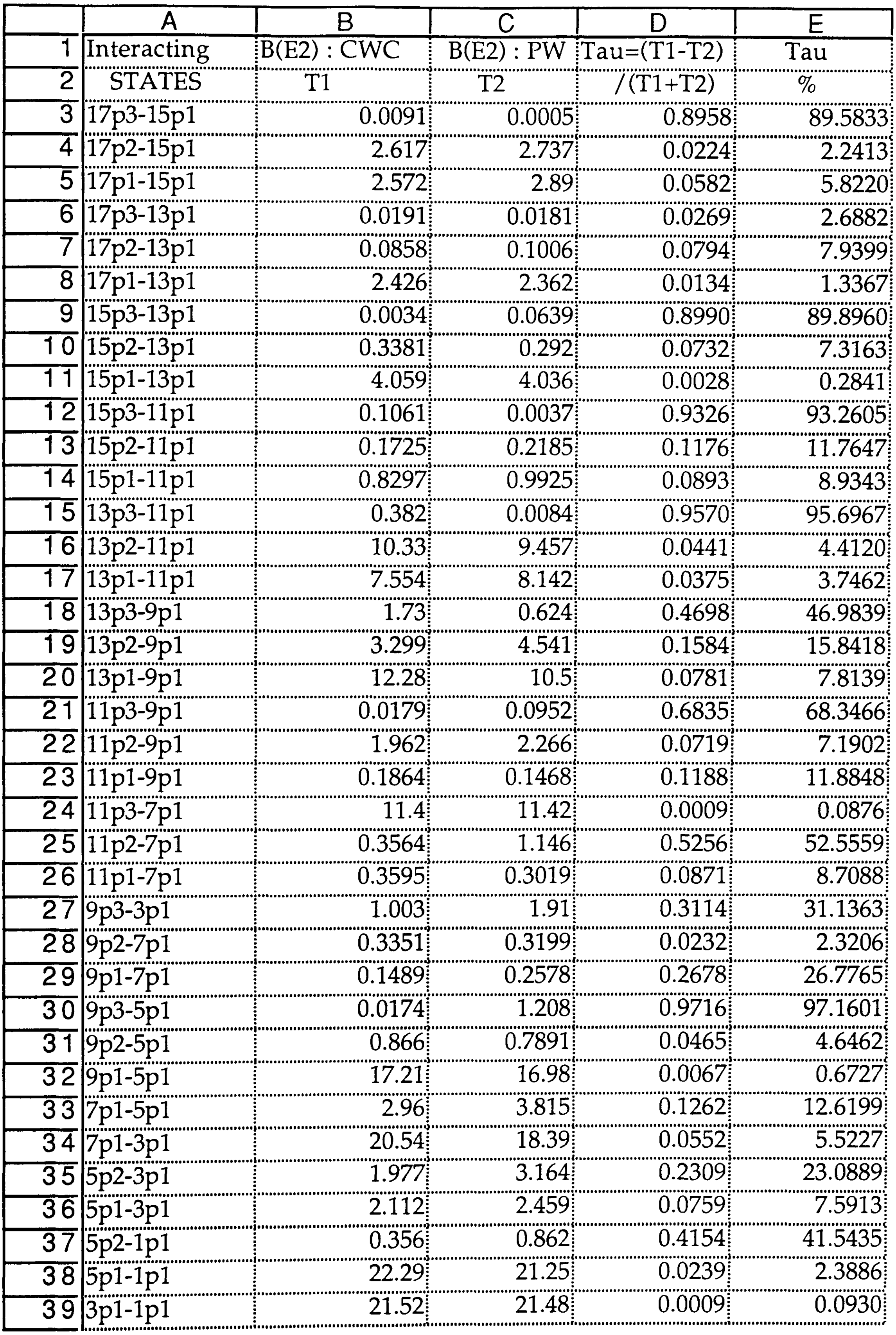
table 5.18: Comparison of CWC-PW-Usd energy Values for na21

	A	B	C	D	E
1	Interacting	B(E2) : CWC	B(E2) : PW	Tau=(T1-T2)/	Tau
2	STATES	T1	T2	(T1+T2)	%
3	13p3-11p1	0.0432	0.0121	0.562387	56.2387
4	13p2-11p1	0.671	0.3311	0.339188	33.9188
5	13p1-11p1	1.439	1.423	0.005590	0.5590
6	13p3-9p1	0.0168	0.0156	0.037037	3.7037
7	13p2-9p1	0.0015	0.0087	0.705882	70.5882
8	13p1-9p1	0.0903	0.0022	0.952432	95.2432
9	11p3-9p1	0.0648	0.0081	0.777778	77.7778
10	11p2-9p1	0.19971	0.7884	0.595774	59.5774
11	11p1-9p1	1.511	1.56	0.015956	1.5956
12	11p3-7p1	0.0064	0.0006	0.828571	82.8571
13	11p2-7p1	0.3638	0.4429	0.098054	9.8054
14	11p1-7p1	1.962	1.984	0.005575	0.5575
15	9p3-7p1	1.298	0.9646	0.147353	14.7353
16	9p2-7p1	2.492	2.047	0.098039	9.8039
17	9p1-7p1	0.1088	0.366	0.541702	54.1702
18	9p3-5p1	0.6061	0.4611	0.135870	13.5870
19	9p2-5p1	0.0799	0.0016	0.960736	96.0736
20	9p1-5p1	0.942	1.206	0.122905	12.2905
21	7p2-5p1	0.0365	0.1628	0.633718	63.3718
22	7p1-5p1	6.835	6.441	0.029678	2.9678
23	7p2-3p1	1.145	1.327	0.073625	7.3625
24	7p1-3p1	1.954	2.296	0.334496	33.4496
25	5p3-3p1	0.9084	1.254	0.218204	21.8204
26	5p2-3p1	1.126	1.441	0.122711	12.2711
27	5p1-3p1	2.912	2.731	0.032075	3.2075
28	5p3-1p1	0.0648	0.0638	0.007776	0.7776
29	5p2-1p1	4.305	4.595	0.032584	3.2584
30	5p1-1p1	0.392	0.4018	0.012346	1.2346
31	3p3-1p1	1.548	0.0444	0.944235	94.4235
32	3p2-1p1	0.2836	1.726	0.717755	71.7755
33	3p1-1p1	1.973	2.367	0.090783	9.0783

table 5.19: Comparison of B(E2) values for O21 using CWC and PW Interactions

	A	B	C	D	E
1	Interacting	B(E2) : CWC	B(E2) : PW	Tau=(T1-T2)/	Tau
2	STATES	T1	T2	(T1+T2)	%
3	13p3-11p1	0.9461	0.1094	0.79270	79.2705
4	13p2-11p1	5.264	2.98	0.27705	27.7050
5	13p1-11p1	15.84	12.81	0.10576	10.5759
6	13p3-9p1	0.332	0.1404	0.40559	40.5588
7	13p2-9p1	0.1090	0.0785	0.16267	16.2667
8	13p1-9p1	13.1	0.0202	0.99692	99.6921
9	11p3-9p1	12.29	0.073	0.98819	98.8191
10	11p2-9p1	17.82	7.095	0.43046	43.0464
11	11p1-9p1	0.0223	14.04	0.99683	99.6828
12	11p3-7p1	0.205	0.0054	0.94867	94.8669
13	11p2-7p1	4.144	3.986	0.01943	1.9434
14	11p1-7p1	16.36	17.85	0.04355	4.3555
15	9p3-7p1	10.38	8.682	0.08908	8.9078
16	9p2-7p1	6.861	18.43	0.45744	45.7435
17	9p1-7p1	17.53	3.294	0.68363	68.3634
18	9p3-5p1	5.316	4.149	0.12330	12.3296
19	9p2-5p1	9.739	0.0148	0.99697	99.6965
20	9p1-5p1	0.011	10.86	0.99798	99.7976
21	7p2-5p1	0.4359	1.465	0.54138	54.1375
22	7p1-5p1	65.07	57.96	0.98507	98.5071
23	7p2-3p1	8.55	1.94	0.94210	94.2098
24	7p1-3p1	18.24	20.66	0.41458	41.4584
25	5p3-3p1	7.39	11.29	0.13810	13.8105
26	5p2-3p1	9.169	12.97	0.16886	16.8856
27	5p1-3p1	29.86	24.58	0.53769	53.7692
28	5p3-1p1	0.2847	57.37	0.72440	72.4402
29	5p2-1p1	40.55	41.36	0.00989	0.9889
30	5p1-1p1	2.064	3.616	0.27324	27.3239
31	3p3-1p1	12.7	0.3993	0.93903	93.9035
32	3p2-1p1	3.561	15.54	0.62714	62.7140
33	3p1-1p1	16.65	21.3	0.12253	12.2530

table 5.20: Comparison of B(E2) values for Al21 using CWC and PW Interactions



	A	B	C	D	E
1	Interacting	B(E2) : CWC	B(E2) : PW	Tau=(T1-T2)	Tau
2	STATES	T1	T2	/(T1+T2)	%
3	17p3-15p1	0.0091	0.0005	0.8958	89.5833
4	17p2-15p1	2.617	2.737	0.0224	2.2413
5	17p1-15p1	2.572	2.89	0.0582	5.8220
6	17p3-13p1	0.0191	0.0181	0.0269	2.6882
7	17p2-13p1	0.0858	0.1006	0.0794	7.9399
8	17p1-13p1	2.426	2.362	0.0134	1.3367
9	15p3-13p1	0.0034	0.0639	0.8990	89.8960
10	15p2-13p1	0.3381	0.292	0.0732	7.3163
11	15p1-13p1	4.059	4.036	0.0028	0.2841
12	15p3-11p1	0.1061	0.0037	0.9326	93.2605
13	15p2-11p1	0.1725	0.2185	0.1176	11.7647
14	15p1-11p1	0.8297	0.9925	0.0893	8.9343
15	13p3-11p1	0.382	0.0084	0.9570	95.6967
16	13p2-11p1	10.33	9.457	0.0441	4.4120
17	13p1-11p1	7.554	8.142	0.0375	3.7462
18	13p3-9p1	1.73	0.624	0.4698	46.9839
19	13p2-9p1	3.299	4.541	0.1584	15.8418
20	13p1-9p1	12.28	10.5	0.0781	7.8139
21	11p3-9p1	0.0179	0.0952	0.6835	68.3466
22	11p2-9p1	1.962	2.266	0.0719	7.1902
23	11p1-9p1	0.1864	0.1468	0.1188	11.8848
24	11p3-7p1	11.4	11.42	0.0009	0.0876
25	11p2-7p1	0.3564	1.146	0.5256	52.5559
26	11p1-7p1	0.3595	0.3019	0.0871	8.7088
27	9p3-3p1	1.003	1.91	0.3114	31.1363
28	9p2-7p1	0.3351	0.3199	0.0232	2.3206
29	9p1-7p1	0.1489	0.2578	0.2678	26.7765
30	9p3-5p1	0.0174	1.208	0.9716	97.1601
31	9p2-5p1	0.866	0.7891	0.0465	4.6462
32	9p1-5p1	17.21	16.98	0.0067	0.6727
33	7p1-5p1	2.96	3.815	0.1262	12.6199
34	7p1-3p1	20.54	18.39	0.0552	5.5227
35	5p2-3p1	1.977	3.164	0.2309	23.0889
36	5p1-3p1	2.112	2.459	0.0759	7.5913
37	5p2-1p1	0.356	0.862	0.4154	41.5435
38	5p1-1p1	22.29	21.25	0.0239	2.3886
39	3p1-1p1	21.52	21.48	0.0009	0.0930

table 5.21: comparison of B(E2) values for F21 using CWC and PW interactions

	A	B	C	D	E
1	Interacting	B(E2) : CWC	B(E2) : PW	Tau=(T1-T2)/	Tau
2	STATES	T1	T2	(T1+T2)	%
3	17p3-15p1	0.0976	0.0001	0.9980	99.7953
4	17p2-15p1	10.46	10.04	0.0205	2.0488
5	17p1-15p1	9.746	12.11	0.1082	10.8163
6	17p3-13p1	0.0348	0.0433	0.1088	10.8835
7	17p2-13p1	0.3178	0.2907	0.0445	4.4536
8	17p1-13p1	8.79	7.778	0.0611	6.1082
9	15p3-13p1	0.0886	0.6401	0.7568	75.6827
10	15p2-13p1	4.276	1.988	0.3653	36.5262
11	15p1-13p1	20.24	18.74	0.0385	3.8481
12	15p3-11p1	0.2231	0.1878	0.0859	8.5909
13	15p2-11p1	0.1715	0.2084	0.0971	9.7131
14	15p1-11p1	1.722	2.356	0.1555	15.5468
15	13p3-11p1	6.646	0.0924	0.9726	97.2575
16	13p2-11p1	21.08	17.63	0.0891	8.9124
17	13p1-11p1	5.42	7.623	0.1689	16.8903
18	13p3-9p1	7.41	2.15	0.5502	55.0209
19	13p2-9p1	6.609	17.05	0.4413	44.1312
20	13p1-9p1	36.07	333.49	0.8048	80.4795
21	11p3-9p1	0.2019	1.849	0.8031	80.3111
22	11p2-9p1	10.34	12.39	0.0902	9.0189
23	11p1-9p1	0.2769	0.3708	0.1450	14.4975
24	11p3-7p1	32.55	36.73	0.0603	6.0335
25	11p2-7p1	2.109	4.292	0.3410	34.1040
26	11p1-7p1	0.0289	0.1642	0.7007	70.0673
27	9p3-7p1	4.563	3.841	0.0859	8.5911
28	9p2-7p1	0.3958	2.079	0.6801	68.0136
29	9p1-7p1	4.356	2.726	0.2302	23.0161
30	9p3-5p1	1.145	3.937	0.5494	54.9390
31	9p2-5p1	1.913	2.124	0.0523	5.2267
32	9p1-5p1	57.85	58.15	0.0026	0.2586
33	7p1-5p1	19.41	19.36	0.0013	0.1290
34	7p1-3p1	45.19	40.61	0.0534	5.3380
35	5p2-3p1	4.683	9.109	0.3209	32.0911
36	5p1-3p1	17.25	18.42	0.0328	3.2801
37	5p2-1p1	7.005	8.009	0.0669	6.6871
38	5p1-1p1	46.94	44.92	0.0220	2.1990
39	3p1-1p1	45.04	40.7	0.0506	

table 5.22: Comparison of B(E2) values for Mg21 using CWC and PW interactions

	A	B	C	D	E
1	STATES	B(E2):CWC T1	B(E2):PW T2	Tau_e	Tau_e %
2	19p2-17p1	0.0168	0.0257	0.209412	20.9412
3	19p1-17p1	3.296	3.255	0.006259	0.6259
4	19p2-15p1	0.6642	0.7006	0.026671	2.6671
5	19p1-15p1	18.61	21.86	0.080306	8.0306
6	17p3-15p1	1.366	1.232	0.051578	5.1578
7	17p2-15p1	0.0151	0.3201	0.909905	90.9905
8	17p1-15p1	8.122	6.547	0.107369	10.7369
9	17p3-13p1	0.9176	0.626	0.188909	18.8909
10	17p2-13p1	1.36	2.008	0.192399	19.2399
11	17p1-13p1	26.71	25.83	0.016749	1.6749
12	15p3-13p1	0.3281	0.0687	0.653730	65.3730
13	15p2-13p1	0.1621	0.2586	0.229380	22.9380
14	15p1-13p1	8.574	8.43	0.008469	0.8469
15	15p3-11p1	0.9372	0.483	0.319814	31.9814
16	15p2-11p1	0.6994	0.0291	0.920110	92.0110
17	15p1-11p1	33.5	35.31	0.026304	2.6304
18	13p3-11p1	0.1312	0.0242	0.688546	68.8546
19	13p2-11p1	2.24	2.38	0.030303	3.0303
20	13p1-11p1	12.36	12.1	0.010630	1.0630
21	13p3-9p1	0.0016	0.0029	0.288889	28.8889
22	13p2-9p1	7.531	5.8	0.129848	12.9848
23	13p1-9p1	42.03	42.75	0.008493	0.8493
24	11p2-9p1	0.2696	0.4837	0.284216	28.4216
25	11p1-9p1	27.74	27.09	0.011855	1.1855
26	11p2-7p1	0.2644	0.4655	0.275517	27.5517
27	11p1-7p1	50.28	50.33	0.000497	0.0497
28	9p3-7p1	0.0125	0.0032	0.592357	59.2357
29	9p2-7p1	0.0335	0.0783	0.400716	40.0716
30	9p1-7p1	34.7	31.16	0.053750	5.3750
31	9p3-5p1	1.872	1.091	0.263584	26.3584
32	9p2-5p1	0.0151	0.8156	0.963645	96.3645
33	9p1-5p1	46.97	46.33	0.006860	0.6860
34	7p2-5p1	0.434	1.066	0.421333	42.1333
35	7p1-5p1	61.81	59.7	0.017365	1.7365
36	7p2-3p1	2.672	1.617	0.245978	24.5978
37	7p1-3p1	34.46	34.37	0.001308	0.1308
38	5p3-3p1	5.256	4.481	0.079593	7.9593
39	5p2-3p1	7.643	7.697	0.003520	0.3520
40	5p1-3p1	82.89	80.08	0.017242	1.7242
41	5p3-1p1	26.06	22.61	0.070886	7.0886
42	5p2-1p1	19.38	17.92	0.039142	3.9142
43	5p1-1p1	0.0022	0.0093	0.617391	61.7391
44	3p1-1p1	0.848	1.019	0.091591	9.1591
45					
46					
47					

table 5.23: Comparison of B(E2) values for Ne21 using CWC and PW interactions

	A	B	C	D	E
1	STATES	B(E2):CWC T1	B(E2):PW T2	Tau_e	Tau_e %
2	19p2-17p1	0.0359	0.0185	0.3199	31.9853
3	19p1-17p1	12.03	10.99	0.0452	4.5178
4	19p2-15p1	0.1369	0.2419	0.2772	27.7191
5	19p1-15p1	24.87	30.01	0.0937	9.3659
6	17p3-15p1	0.6276	0.4988	0.1143	11.4347
7	17p2-15p1	0.9199	0.0119	0.9745	97.4458
8	17p1-15p1	9.032	6.516	0.1618	16.1821
9	17p3-13p1	0.8077	0.3638	0.3789	37.8916
10	17p2-13p1	0.8675	1.989	0.3926	39.2613
11	17p1-13p1	32.84	33.27	0.0065	0.6504
12	15p3-13p1	3.895	0.489	0.7769	77.6916
13	15p2-13p1	1.415	4.776	0.5429	54.2885
14	15p1-13p1	16.44	15.27	0.0369	3.6897
15	15p3-11p1	1.291	0.4751	0.4620	46.1978
16	15p2-11p1	0.985	0.0015	0.9970	99.6959
17	15p1-11p1	44.16	46.91	0.0302	3.0197
18	13p3-11p1	0.6282	0.0009	0.9971	99.7139
19	13p2-11p1	0.0573	0.4555	0.7765	77.6521
20	13p1-11p1	12.48	11.8	0.0280	2.8007
21	13p3-9p1	0.4924	0.1785	0.4679	46.7879
22	13p2-9p1	6.26	7.064	0.0603	6.0342
23	13p1-9p1	37.54	38.54	0.0131	1.3144
24	11p2-9p1	0.0259	0.0148	0.2727	27.2727
25	11p1-9p1	28.94	26.58	0.0425	4.2507
26	11p2-7p1	0.1205	0.0119	0.8202	82.0242
27	11p1-7p1	56.53	58.67	0.0186	1.8576
28	9p3-7p1	0.4361	0.2609	0.2514	25.1363
29	9p2-7p1	0.5653	0.0547	0.8235	82.3548
30	9p1-7p1	32.63	29.5	0.0504	5.0378
31	9p3-5p1	0.2653	0.0242	0.8328	83.2815
32	9p2-5p1	0.0095	0.054	0.7008	70.0787
33	9p1-5p1	49.51	50.53	0.0102	1.0196
34	7p2-5p1	0.5854	0.576	0.0081	0.8094
35	7p1-5p1	65.68	62.72	0.0231	2.3053
36	7p2-3p1	0.2551	0.0007	0.9945	99.4527
37	7p1-3p1	40.19	39.96	0.0029	0.2870
38	5p3-3p1	3.401	2.15	0.2254	22.5365
39	5p2-3p1	2.295	1.394	0.2442	24.4240
40	5p1-3p1	97.55	94.47	0.0160	1.6040
41	5p3-1p1	11.74	22.17	0.3076	30.7579
42	5p2-1p1	32.31	26.21	0.1042	10.4238
43	5p1-1p1	3.022	2.275	0.1410	14.1023
44	3p1-1p1	1.325	3.53	0.4542	45.4171

Table 5.24: Comparison of B(E2) values for Na21 using CWC and PW interactions

	A	B	C	D	E
1	Interacting	100*CWC	100*PW	Tau= (T1-T2)/	Tau_m%
2	STATES	T1	T2	(T1+T2)	
3	13p3->11p1	0.1327	0.1043	0.119831	11.9831
4	13p2->11p1	37.62	40.82	0.040796	4.0796
5	13p1->11p1	0.0527	0.5439	0.823332	82.3332
6	11p3->9p1	0.0112	1.025	0.978383	97.8383
7	11p2->9p1	7.26	5.77	0.114351	11.4351
8	11p1->9p1	0.0013	0.7278	0.996434	99.6434
9	9p3->7p1	0.0394	3.199	0.975667	97.5667
10	9p2->7p1	2.782	8.751	0.517558	51.7558
11	9p1->7p1	31.07	26.16	0.085794	8.5794
12	7p3->5p1	0.1154	18.95	0.987894	98.7894
13	7p2->5p1	0.8635	1.625	0.306008	30.6008
14	7p1->5p1	3.189	2.384	0.144446	14.4446
15	5p3->3p1	8.615	7.446	0.072785	7.2785
16	5p2->3p1	29.48	25.41	0.074148	7.4148
17	5p1->3p1	0.274	0.0186	0.872864	87.2864
18	3p3->1p1	0.8789	1.193	0.151600	15.1600
19	3p2->1p1	0.363	0.6825	0.305595	30.5595
20	3p1->1p1	1.556	1.889	0.096662	9.6662

table 5.25 : Magnetic transitions, B(M1) values for 21O

	A	B	C	D	E
1	Interaccting	CWC *100	PW *100	Tau=(T1-T2)/	
2	STATES	T1	T2	(T1+ T2)	Tau_m %
3	13p3->11p1	0.2379	0.1499	0.226921	22.6921
4	13p2->11p1	49.08	58.68	0.089087	8.9087
5	13p1->11p1	0.128	0.7818	0.718619	71.8619
6	11p3->9p1	17.42	1.474	0.843972	84.3972
7	11p2->9p1	13.58	8.294	0.241657	24.1657
8	11p1->9p1	7.706	1.046	0.760969	76.0969
9	9p3->7p1	0.0058	4.599	0.997481	99.7481
10	9p2->7p1	33.64	12.58	0.455647	45.5647
11	9p1->7p1	16.92	37.61	0.379424	37.9424
12	7p3->5p1	0.0008	27.24	0.999941	99.9941
13	7p2->5p1	1.001	2.335	0.399880	39.9880
14	7p1->5p1	4.589	3.426	0.145103	14.5103
15	5p3->3p1	15.62	10.7	0.186930	18.6930
16	5p2->3p1	38.54	36.52	0.026912	2.6912
17	5p1->3p1	0.5743	0.0267	0.911148	91.1148
18	3p3->1p1	1.566	1.714	0.045122	4.5122
19	3p2->1p1	0.4987	0.9812	0.326036	32.6036
20	3p1->1p1	2.081	2.715	0.132193	13.2193
21					

table 5.26 : Magnetic transitions, B(M1) values for 21Al.

	A	B	C	D	E
1	STATES	100*CWC	100* PW	Tau = (T1 -T2)/	Tau_m%
2		T1	T2	(T1 +T2)	
3	17p3-15p1	0.0227	1.497	0.9701	97.0126
4	17p2-15p1	30.39	36.39	0.0898	8.9847
5	17p1-15p1	0.4304	2.853	0.7378	73.7833
6	15p3-13p1	1.028	8.666	0.7879	78.7910
7	15p2-13p1	21.52	34.26	0.2284	22.8397
8	15p1-13p1	1.153	3.64	0.5189	51.8882
9	13p3-11p1	2.479	0.2733	0.8014	80.1402
10	13p2-11p1	78.43	78.54	0.0007	0.0701
11	13p1-11p1	79.41	81.18	0.0110	1.1022
12	11p3-9p1	4.723	18.27	0.5892	58.9179
13	11p2-9p1	0.0146	1.056	0.9727	97.2726
14	11p1-9p1	6.319	4.485	0.1698	16.9752
15	9p3-7p1	4.987	7.626	0.2092	20.9229
16	9p2-7p1	32.47	26.29	0.1052	10.5174
17	9p1-7p1	228.9	209.8	0.0435	4.3538
18	7p1-5p1	9.717	11.97	0.1039	10.3887
19	5p3-3p1	14.9	21.77	0.1873	18.7347
20	5p2-3p1	105.5	85.45	0.1050	10.5001
21	5p1-3p1	127.3	129.6	0.0090	0.8953
22	3p1-1p1	51.43	57.29	0.0539	5.3900

table 5.27 : Magnetic transitions, B(M1) values for 21F.

	A	B	C	D	E
1	STATES	100*CWC	100*PW	Tau= (T1-T2)/	Tau_m %
2		T1	T2	(T1 +T2)	
3	17p3-15p1	0.0033	2.04	0.9968	99.6770
4	17p2-15p1	42.98	44.47	0.0170	1.7038
5	17p1-15p1	0.5818	1.866	0.5246	52.4634
6	15p3-13p1	2.922	13.28	0.6393	63.9304
7	15p2-13p1	23.91	42.84	0.2836	28.3596
8	15p1-11p1	0.2804	2.998	0.8289	82.8941
9	13p3-11p1	20.42	0.267	0.9742	97.4187
10	13p2-11p1	95.76	82.44	0.0747	7.4747
11	13p1-11p1	67.04	85.7	0.1222	12.2168
12	11p3-9p1	8.772	19.83	0.3866	38.6616
13	11p2-9p1	0.1288	0.7184	0.6959	69.5940
14	11p1-9p1	7.147	5.983	0.0887	8.8652
15	9p3-7p1	10.03	7.087	0.1719	17.1934
16	9p2-7p1	22.92	23.9	0.0209	2.0931
17	9p1-7p1	169.8	172.4	0.0076	0.7598
18	7p2-5p1	1.312	2.683	0.3432	0.7732
19	7p1-5p1	10.16	11.36	0.0558	0.9902
20	5p2-3p1	99.8	74.91	0.0984	0.9974
21	5p1-3p1	91.27	101.7	0.2378	0.9937
22	3p1-1p1	46.13	50.67	0.3759	0.9926

table 5.28 : Magnetic transitions, B(M1) values for 21Mg.

	A	B	C	D	E
1	Interacting	100*CWC	100*PW	Tau=(T1-T2)/	Tau_m %
2	STATES	T1	T2	(T1+T2)	
3	19p2-17p1	0.1683	0.0714	0.404255	40.4255
4	19p1-17p1	41	49.78	0.096717	9.6717
5	17p3-15p1	0.308	1.664	0.687627	68.7627
6	17p2-15p1	10.2	9.992	0.010301	1.0301
7	17p1-15p1	70.78	57.65	0.102235	10.2235
8	15p3-13p1	30.96	0.8673	0.945500	94.5500
9	15p2-13p1	5.867	36.72	0.724470	72.4470
10	15p1-13p1	23.81	31.33	0.136380	13.6380
11	13p3-11p1	9.253	9.373	0.006443	0.6443
12	13p2-11p1	2.036	4.793	0.403719	40.3719
13	13p1-11p1	97.32	86.32	0.059900	5.9900
14	11p2-9p1	10.58	11.37	0.035991	3.5991
15	11p1-9p1	43.81	40.85	0.034963	3.4963
16	9p3-7p1	46.56	15.17	0.508505	50.8505
17	9p2-7p1	0.5644	36.52	0.969561	96.9561
18	9p1-7p1	47.33	33.41	0.172405	17.2405
19	7p2-5p1	29.07	26	0.055747	5.5747
20	7p1-5p1	31.95	35.17	0.047974	4.7974
21	5p3-3p1	7.422	7.524	0.006825	0.6825
22	5p2-3p1	64.59	72.87	0.060236	6.0236
23	5p1-3p1	16.99	11.19	0.205820	20.5820
24	3p1-1p1	13.59	11.34	0.090253	9.0253

table 5.29: Magnetic transitions, B(M1) values for 21Ne.

	A	B	C	D	E
1	STATES	100*CWC	100*PW	Tau= (T1- T2)/	Tau_m%
2		T1	T2	(T1 +T2)	
3	19p2-17p1	0.1783	0.0822	0.368906	36.8906
4	19p1-17p1	53.28	66.22	0.108285	10.8285
5	17p3-15p1	0.0003	1.669	0.999641	99.9641
6	17p2-15p1	15.52	16.13	0.019273	1.9273
7	17p1-15p1	87.24	65.72	0.140690	14.0690
8	15p3-13p1	36.96	0.1335	0.992802	99.2802
9	15p2-13p1	5.196	43.26	0.785537	78.5537
10	15p1-13p1	31.62	39.32	0.108542	10.8542
11	13p3-11p1	10.29	12.19	0.084520	8.4520
12	13p2-11p1	0.326	2.397	0.760558	76.0558
13	13p1-11p1	108.4	97.91	0.050846	5.0846
14	11p2-9p1	9.777	10.26	0.024105	2.4105
15	11p1-9p1	51.76	50.77	0.009656	0.9656
16	9p3-7p1	38.63	14.57	0.452256	45.2256
17	9p2-7p1	12.29	44.04	0.563643	56.3643
18	9p1-7p1	58.23	40.7	0.177196	17.7196
19	7p2-5p1	31.7	37.9	0.089080	8.9080
20	7p1-5p1	37.57	36.08	0.020231	2.0231
21	5p3-3p1	15.72	6.04	0.444853	44.4853
22	5p2-3p1	68.95	84.03	0.098575	9.8575
23	5p1-3p1	20.89	16.56	0.115621	11.5621
24	3p1-1p1	15.57	39.71	0.436686	43.6686

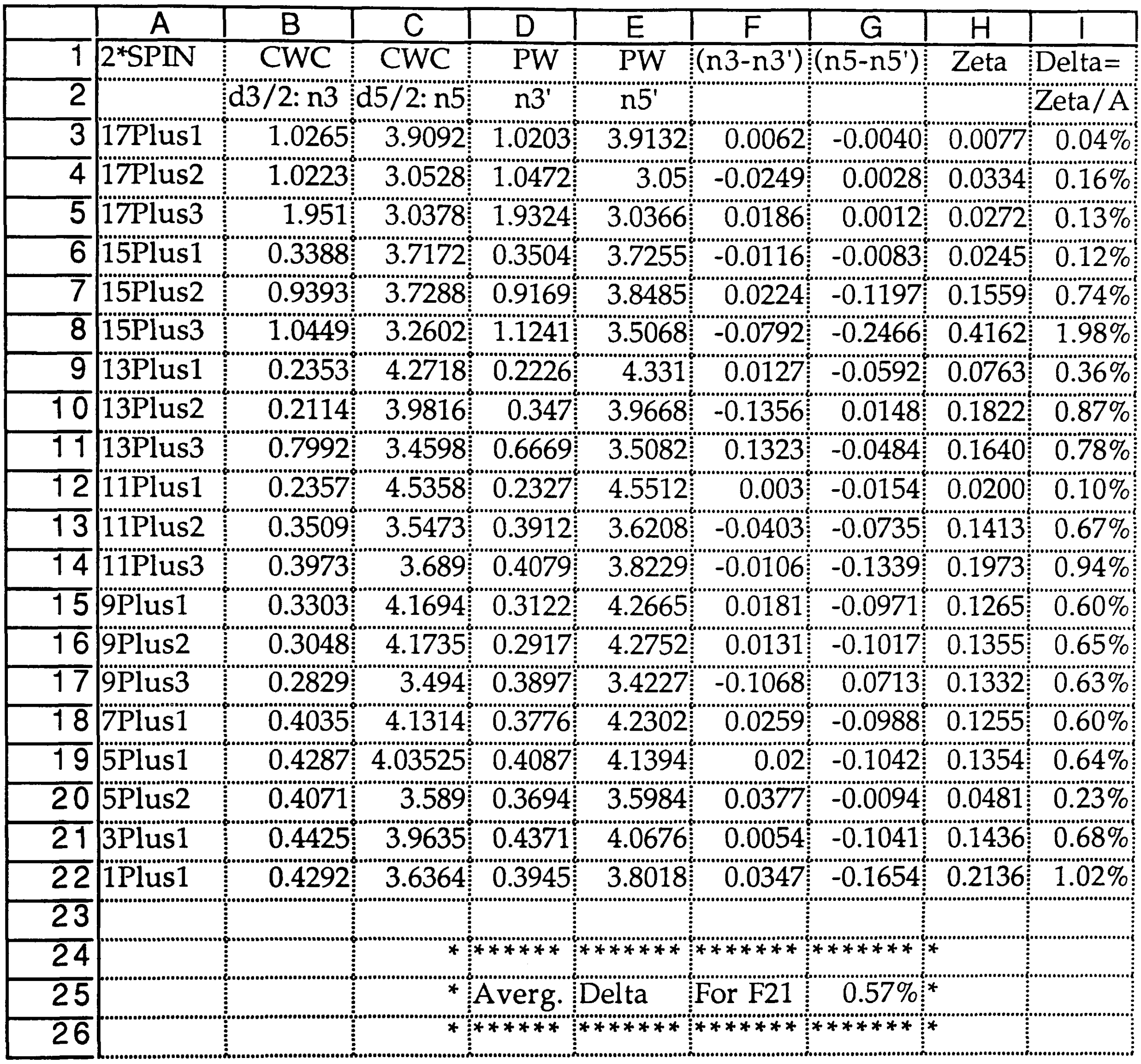
table 5.30 : Magnetic transitions, B(M1) values for 21Na.

	A	B	C	D	E	F	G	H	I
1	2*SPIN	CWC	CWC	PW	PW	(n3-n3')	(n5-n5')	Zeta	Delta=
2		d3/2: n3	d5/2: n5	n3'	n5'				Zeta/A
3	13Plus1	1.1147	2.8855	1.0886	2.9116	0.0261	-0.0261	0.0369	0.18%
4	13Plus2	1.9989	2.9981	1.9993	2.9991	-0.0004	-0.0010	0.0018	0.01%
5	13Plus3	1.8863	2.1162	1.91206	2.0891	-0.0258	0.0271	0.0374	0.18%
6	11Plus1	1.0724	3.7579	1.0482	3.8144	0.0242	-0.0565	0.0694	0.33%
7	11Plus2	1.02832	2.8933	1.0245	2.904	0.0038	-0.0107	0.0133	0.06%
8	11Plus3	1.0106	2.2446	1.0115	2.2159	-0.0009	0.0287	0.0400	0.19%
9	9Plus1	0.2297	3.7872	0.246	3.7462	-0.0163	0.0410	0.0506	0.24%
10	9Plus2	0.16	2.9009	0.1305	2.9855	0.0295	-0.0846	0.1052	0.50%
11	9Plus3	1.0548	3.7972	0.9844	3.8418	0.0704	-0.0446	0.0872	0.42%
12	7Plus1	0.2298	3.7642	0.1742	3.8177	0.0556	-0.0535	0.0772	0.37%
13	7Plus2	1.0859	3.6487	1.0586	3.7142	0.0273	-0.0655	0.0806	0.38%
14	5Plus1	0.2113	4.5067	0.1828	4.5541	0.0285	-0.0474	0.0584	0.28%
15	5Plus2	0.2353	3.4116	0.1935	3.4697	0.0418	-0.0581	0.0734	0.35%
16	5Plus3	0.1526	3.4824	0.1359	3.4564	0.0167	0.0260	0.0527	0.25%
17	3Plus1	0.1641	3.5905	0.1828	4.5541	-0.0187	-0.9636	1.3761	6.55%
18	3Plus2	0.0934	3.1928	0.129	3.6075	-0.0356	-0.4147	0.6132	2.92%
19	3Plus3	1.1523	3.483	0.6243	3.4029	0.5280	0.0801	0.8093	3.85%
20	1Plus1	0.2815	3.7157	0.2496	3.7511	0.0319	-0.0354	0.0478	0.23%
21									
22									
23									
24			*	******	******	*******	*******	*	
25			*	Averg.	Delta	For O21	1.1526%	*	
26			*	******	******	*******	*******	*	

table : 5.31 Occupancy and Delta values for O21

	A	B	C	D	E	F	G	H	I
1	2*SPIN	cwc	cwc	PW	PW	(n3-n3')	(n5-n5')	Zeta	Delta=
2		d3/2: n3	d5/2: n5	n3'	n5'				Zeta/A
3	13Plus1	1.1054	2.8946	1.0886	2.9116	0.0168	-0.0170	0.0239	0.11%
4	13Plus2	1.9992	2.9969	1.9993	2.9991	-0.0001	-0.0022	0.0032	0.02%
5	13Plus3	1.8952	2.1084	1.91206	2.0891	-0.0169	0.0193	0.0257	0.12%
6	11Plus1	1.0657	3.6009	1.0482	3.8144	0.0175	-0.2135	0.2904	1.38%
7	11Plus2	1.0277	2.854	1.0245	2.904	0.0032	-0.0500	0.0686	0.33%
8	11Plus3	1.012	2.4421	1.0115	2.2159	0.0005	0.2262	0.3202	1.52%
9	9Plus1	0.1596	3.0177	0.246	3.7462	-0.0864	-0.7285	1.0965	5.22%
10	9Plus2	0.1935	3.6852	0.1305	2.9855	0.0630	0.6997	1.0369	4.94%
11	9Plus3	1.0694	3.7262	0.9844	3.8418	0.0850	-0.1156	0.1467	0.70%
12	7Plus1	0.2219	3.7648	0.1742	3.8177	0.0477	-0.0529	0.0714	0.34%
13	7Plus2	1.09	3.5676	1.0586	3.7142	0.0314	-0.1466	0.1891	0.90%
14	5Plus1	0.2096	4.3721	0.1828	4.5541	0.0268	-0.1820	0.2407	1.15%
15	5Plus2	0.2236	3.4229	0.1935	3.4697	0.0301	-0.0468	0.0581	0.28%
16	5Plus3	0.1452	3.6133	0.1359	3.4564	0.0093	0.1569	0.2287	1.09%
17	3Plus1	0.1562	3.5262	0.1828	4.5541	-0.0266	-1.0279	1.4728	7.01%
18	3Plus2	0.0882	3.2599	0.129	3.6075	-0.0408	-0.3476	0.5228	2.49%
19	3Plus3	1.152	3.3313	0.6243	3.4029	0.5277	-0.0716	0.7012	3.34%
20	1Plus1	0.274	3.7213	0.2496	3.7511	0.0244	-0.0298	0.0389	0.19%
21									
22			*	*******	*******	*******	*******	*	
23			*	Averg.	Delta F	or Al21	1.73%	*	
24			*	*******	*******	*******	*******	*	

table 5.32 : Occupancy and Delta values for Al21



	A	B	C	D	E	F	G	H	I
1	2*SPIN	CWC	CWC	PW	PW	(n3-n3')	(n5-n5')	Zeta	Delta=
2		d3/2: n3	d5/2: n5	n3'	n5'				Zeta/A
3	17Plus1	1.0265	3.9092	1.0203	3.9132	0.0062	-0.0040	0.0077	0.04%
4	17Plus2	1.0223	3.0528	1.0472	3.05	-0.0249	0.0028	0.0334	0.16%
5	17Plus3	1.951	3.0378	1.9324	3.0366	0.0186	0.0012	0.0272	0.13%
6	15Plus1	0.3388	3.7172	0.3504	3.7255	-0.0116	-0.0083	0.0245	0.12%
7	15Plus2	0.9393	3.7288	0.9169	3.8485	0.0224	-0.1197	0.1559	0.74%
8	15Plus3	1.0449	3.2602	1.1241	3.5068	-0.0792	-0.2466	0.4162	1.98%
9	13Plus1	0.2353	4.2718	0.2226	4.331	0.0127	-0.0592	0.0763	0.36%
10	13Plus2	0.2114	3.9816	0.347	3.9668	-0.1356	0.0148	0.1822	0.87%
11	13Plus3	0.7992	3.4598	0.6669	3.5082	0.1323	-0.0484	0.1640	0.78%
12	11Plus1	0.2357	4.5358	0.2327	4.5512	0.003	-0.0154	0.0200	0.10%
13	11Plus2	0.3509	3.5473	0.3912	3.6208	-0.0403	-0.0735	0.1413	0.67%
14	11Plus3	0.3973	3.689	0.4079	3.8229	-0.0106	-0.1339	0.1973	0.94%
15	9Plus1	0.3303	4.1694	0.3122	4.2665	0.0181	-0.0971	0.1265	0.60%
16	9Plus2	0.3048	4.1735	0.2917	4.2752	0.0131	-0.1017	0.1355	0.65%
17	9Plus3	0.2829	3.494	0.3897	3.4227	-0.1068	0.0713	0.1332	0.63%
18	7Plus1	0.4035	4.1314	0.3776	4.2302	0.0259	-0.0988	0.1255	0.60%
19	5Plus1	0.4287	4.03525	0.4087	4.1394	0.02	-0.1042	0.1354	0.64%
20	5Plus2	0.4071	3.589	0.3694	3.5984	0.0377	-0.0094	0.0481	0.23%
21	3Plus1	0.4425	3.9635	0.4371	4.0676	0.0054	-0.1041	0.1436	0.68%
22	1Plus1	0.4292	3.6364	0.3945	3.8018	0.0347	-0.1654	0.2136	1.02%
23									
24			*	******	*******	*******	*******	*	
25			*	Averg.	Delta	For F21	0.57%	*	
26			*	******	*******	*******	*******	*	

table 5.33: Occupancy and Delta values for F21

	A	B	C	D	E	F	G	H	I
1	2*SPIN	CWC	CWC	PW	PW	(n3-n3')	(n5-n5')	Zeta	Delta=
2		d3/2: n3	d5/2: n5	n3'	n5'				Zeta/A
3	17Plus1	1.0235	3.883	1.0203	3.9132	0.0032	-0.0302	0.0406	0.19%
4	17Plus2	1.0168	3.0825	1.0472	3.05	-0.0304	0.0325	0.0446	0.21%
5	17Plus3	1.9595	3.0343	1.9324	3.0366	0.0271	-0.0023	0.0368	0.18%
6	15Plus1	0.2967	3.7413	0.3504	3.7255	-0.0537	0.0158	0.0676	0.32%
7	15Plus2	0.9585	3.5917	0.9169	3.8485	0.0416	-0.2568	0.3376	1.61%
8	15Plus3	1.0495	3.3839	1.1241	3.5068	-0.0746	-0.1229	0.2443	1.16%
9	13Plus1	0.186	4.0922	0.2226	4.331	-0.0366	-0.2388	0.3663	1.74%
10	13Plus2	0.4368	3.4439	0.347	3.9668	0.0898	-0.5229	0.6849	3.26%
11	13Plus3	0.8975	3.5008	0.6669	3.5082	0.2306	-0.00740	0.3210	1.53%
12	11Plus1	0.2333	4.4774	0.2327	4.5512	0.0006	-0.0738	0.1039	0.49%
13	11Plus2	0.3044	3.5485	0.3912	3.6208	-0.0868	-0.0723	0.1951	0.93%
14	11Plus3	0.345	3.6927	0.4079	3.8229	-0.0629	-0.1302	0.2412	1.15%
15	9Plus1	0.333	4.0844	0.3122	4.2665	0.0208	-0.1821	0.2442	1.16%
16	9Plus2	0.31	4.112	0.2917	4.2752	0.0183	-0.1632	0.2190	1.04%
17	9Plus3	0.2624	3.4111	0.3897	3.4227	-0.1273	-0.0116	0.1888	0.90%
18	7Plus1	0.4046	3.9451	0.3776	4.2302	0.0270	-0.2851	0.3855	1.84%
19	5Plus1	0.4242	3.9555	0.4087	4.1394	0.0155	-0.1839	0.2498	1.19%
20	5Plus2	0.3999	3.4877	0.3694	3.5984	0.0305	-0.1107	0.1401	0.67%
21	3Plus1	0.4248	3.9364	0.4371	4.0676	-0.0123	-0.1312	0.1948	0.93%
22	1Plus1	0.4256	3.6035	0.3945	3.8018	0.0311	-0.1983	0.2612	1.24%
23									
24			*	*******	*******	********	********	*	
25			*	Averg.	Delta F	or Mg21	1.0875%	*	
26			*	*******	*******	********	********	*	

table 5.34 : Occupancy and delta values for Mg21

	A	B	C	D	E	F	G	H	I
1	2*SPIN	CWC	CWC	PW	PW	(n3-n3')	(n5-n5')	Zeta	Delta=
2		d3/2: n3	d5/2: n5	n3'	n5'				Zeta/A
3	19Plus1	1.1735	3.8264	1.1783	3.8216	-0.0048	0.0048	0.0068	0.03%
4	19Plus2	1.8264	3.1735	1.8216	3.1783	0.0048	-0.0048	0.0068	0.03%
5	17Plus1	0.3283	4.5159	0.365	4.4898	-0.0367	0.0261	0.0463	0.22%
6	17Plus2	0.5339	3.8112	0.5833	3.8323	-0.0494	-0.0211	0.0886	0.42%
7	17Plus3	0.914	3.7441	0.8433	3.7729	0.0707	-0.0288	0.0871	0.41%
8	15Plus1	0.6459	3.7497	0.7386	3.7289	-0.0927	0.0208	0.1191	0.57%
9	15Plus2	0.2646	3.8732	0.7633	4.108	-0.4987	-0.2348	0.9175	4.37%
10	15Plus3	0.8398	3.8796	0.4241	3.6855	0.4157	0.1941	0.7631	3.63%
11	13Plus1	0.483	4.0192	0.5267	4.0032	-0.0437	0.0160	0.0542	0.26%
12	13Plus2	0.5942	3.8525	0.5792	4.0147	0.0150	-0.1622	0.2195	1.05%
13	13Plus3	0.3385	3.8019	0.3564	4.0751	-0.0179	-0.2732	0.3996	1.90%
14	11Plus1	0.671	3.8349	0.6849	3.8488	-0.0139	-0.0139	0.0340	0.16%
15	11Plus2	0.3248	4.02	0.2792	4.2357	0.0456	-0.2157	0.2785	1.33%
16	11Plus3	0.5209	3.6214	1.1005	3.3521	-0.5796	0.2693	0.7105	3.38%
17	9Plus1	0.5864	3.5587	0.5854	3.6449	0.0010	-0.0862	0.1212	0.58%
18	9Plus2	0.5185	3.4543	0.5888	3.5394	-0.0703	-0.0851	0.1906	0.91%
19	9Plus3	0.5706	3.9601	0.4563	4.0571	0.1143	-0.0970	0.1509	0.72%
20	7Plus1	0.5843	3.4154	0.5844	3.5302	-0.0001	-0.1148	0.1624	0.77%
21	7Plus2	0.6625	3.7148	0.6827	3.7188	-0.0202	-0.0040	0.0318	0.15%
22	7Plus3	0.6459	3.2188	0.6894	3.4009	-0.0435	-0.1821	0.2932	1.40%
23	5Plus1	0.6049	3.4832	0.5921	3.6348	0.0128	-0.1516	0.2059	0.98%
24	5Plus2	0.6291	3.365	0.6061	3.4738	0.0230	-0.1088	0.1405	0.67%
25	5Plus3	0.5494	2.9556	0.5703	3.0618	-0.0209	-0.1062	0.1669	0.79%
26	3Plus1	0.571	3.4055	0.5379	3.568	0.0331	-0.1625	0.2103	1.00%
27	1Plus1	0.6401	2.8745	0.5752	3.0004	0.0649	-0.1259	0.1542	0.73%
28									
29			*	******	******	*******	*		
30			*	Avg. fo	r Ne21	1.06%	*		
31			*	******	******	*******	*		

table 5.35 : Occupancy and delta values for Ne21

	A	B	C	D	E	F	G	H	I
1	2*SPIN	CWC	CWC	PW	PW	(n3-n3')	(n5-n5')	Zeta	Delta=
2		d3/2: n3	d5/2: n5	n3'	n5'				Zeta/A
3	19Plus1	1.1789	3.8211	1.1783	3.8216	0.0006	-2.6433	3.7378	17.80%
4	19Plus2	1.8211	3.1789	1.8216	3.1783	-0.0005	-1.3567	1.9190	9.14%
5	17Plus1	0.353	4.4477	0.365	4.4898	-0.0120	-4.1248	5.8419	27.82%
6	17Plus2	0.4469	3.8813	0.5833	3.8323	-0.1364	-3.2490	4.6942	22.35%
7	17Plus3	0.9444	3.7477	0.8433	3.7729	0.1011	-2.9296	4.0735	19.40%
8	15Plus1	0.6322	3.7482	0.7386	3.7289	-0.1064	-2.9903	4.3061	20.51%
9	15Plus2	0.2666	3.8907	0.7633	4.108	-0.4967	-3.3447	5.1176	24.37%
10	15Plus3	0.8526	3.8513	0.4241	3.6855	0.4285	-3.2614	4.3412	20.67%
11	13Plus1	0.4999	3.9197	0.5267	4.0032	-0.0268	-3.4765	4.9356	23.50%
12	13Plus2	0.56	3.8115	0.5792	4.0147	-0.0192	-3.4355	4.8722	23.20%
13	13Plus3	0.3267	3.8391	0.3564	4.0751	-0.0297	-3.7187	5.2802	25.14%
14	11Plus1	0.6764	3.7975	0.6849	3.8488	-0.0085	-3.1639	4.4805	21.34%
15	11Plus2	0.3328	3.9895	0.2792	4.2357	0.0536	-3.9565	5.5578	26.47%
16	11Plus3	0.547	3.5829	1.1005	3.3521	-0.5535	-2.2516	3.6393	17.33%
17	9Plus1	0.5904	3.5234	0.5854	3.6449	0.0050	-3.0595	4.3233	20.59%
18	9Plus2	0.5505	3.36	0.5888	3.5394	-0.0383	-2.9506	4.2001	20.00%
19	9Plus3	0.547	4.0281	0.4563	4.0571	0.0907	-3.6008	5.0294	23.95%
20	7Plus1	0.58	3.3944	0.5844	3.5302	-0.0044	-2.9458	4.1691	19.85%
21	7Plus2	0.6855	3.6914	0.6827	3.7188	0.0028	-3.0361	4.2917	20.44%
22	7Plus3	0.5288	3.7662	0.6894	3.4009	-0.1606	-2.7115	3.9531	18.82%
23	5Plus1	0.6022	3.4742	0.5921	3.6348	0.0101	-3.0427	4.2959	20.46%
24	5Plus2	0.6107	3.251	0.6061	3.4738	0.0046	-2.8677	4.0523	19.30%
25	5Plus3	0.5808	3.0845	0.5703	3.0618	0.0105	-2.4915	3.5161	16.74%
26	3Plus1	0.5717	3.4134	0.5379	3.568	0.0338	-3.0301	4.2615	20.29%
27	1Plus1	0.6574	2.8942	0.5752	3.0004	0.0822	-2.4252	3.3731	16.06%
28									
29			*	*******	******	*******	*******	*	
30			*	Averg.	Delta F	or Na21	20.62%	*	
31			*	*******	******	*******	*******	*	

table 5.36 : Occupancy and delta values for Na21