Rev. Mexicana Astron. Astrof., **23**, 107-118 (1992)

THE OPACITY PROJECT — THE TOPBASE ATOMIC DATABASE

W. Cunto and C. Mendoza*

IBM Venezuela Scientific Center

RESUMEN. TOPBASE es el primer prototipo de un DBMS específico para la manipulación eficiente de la gran cantidad de datos atómicos que resultaron del "Proyecto de la Opacidad". Se describen los detalles de su enfoque conceptual, el diseño físico y el lenguaje de consulta. La capacidad presente de búsqueda, manipulación y depliegue gráfico de los datos de TOPBASE se ilustra con algunos ejemplos.

ABSTRACT. TOPBASE is the first prototype in the development of a tailored DBMS for the efficient manipulation of the large volume of atomic data that resulted from the "Opacity Project". We describe the details of the adopted conceptual approach, its physical design and query language. A few examples are included to illustrate its present search, data manipulation and graphic capabilities.

Key words: ATOMIC PROCESSES

I. INTRODUCTION

The name "Opacity Project" (OP) refers to an international collaboration that was formed in 1984 to re-estimate stellar envelope opacities and to calculate the extensive atomic data required (Seaton 1987). The project has involved research groups from France, Germany, the United Kingdom, the United States and Venezuela. The approach adopted by the OP to calculate opacities is based on a new formalism of the equation of state (Hummer and Mihalas 1988) and on the mass production of accurate atomic properties such as energy levels, f-values and photoionization cross sections (Seaton *et al.* 1992). In this respect, the OP data accuracy has been shown to be comparable with the state of the art in atomic computations and within the level of agreement with experiment attained by such theoretical methods (see, for instance, Allard *et al.* 1990 and Butler *et al.* 1990, 1991). Moreover, the volume of data generated, ~ 0.5 Gb, and their completeness surpass those of currently used atomic datasets by as much as orders of magnitude (Mendoza 1992).

Since such prime atomic data are likely to be used in several research fields, we have been motivated to make a definite effort to enhance their accessibility in terms of the requirements of prospective users. We are concerned then with the definition, development, distribution and maintenance of a specific, portable and low-cost database management system (DBMS) to facilitate the intensive use—both on-line and batch modes—of the OP atomic dataset. TOPBASE is our first attempt in this undertaking. It addresses all the requirements previously mentioned, emphasizing, amongst them, portability and time/space performance. The latter is managed by exploiting the specific properties of the OP dataset and by adopting efficient computational methods.

Due to needs for atomic data in astrophysics and fusion research, international meetings are periodically held to compile, assess and recommend atomic data and bibliographies (see reports edited by Merts *et al.* 1980, Eissner 1986, Eissner and Kingston 1988, and Zeippen and Le Dourneuf 1991). Current projects in the field of atomic databases (Berrington 1986, Burgess *et al.* 1989, 1991, Martin 1992, and references therein) are closely related to such activities rather than with a wide distribution of atomic data products. In spite of meetings between data producers and users (see, for instance, Hansen 1990), little attention has been devoted to

[*] Visiting Fellow, IVIC, Caracas, Venezuela

the problem of atomic database standards. We think that a set of standards is necessary in order to optimize and normalize contributions to the field, no matter how small, which will no doubt involve an increasing participation of outsiders.

The present stage of development of TOPBASE is that of a robust prototype running on the VM/CMS* and AIX* operating systems, and it is to undergo detailed auditing by a group of test users. It handles ~ 0.5 Gb of compact data with special attention to fast searches along isoelectronic and isonuclear sequences, spectroscopic series and to the sorting of energies and wavelengths. Its user-oriented query language has been devised to provide concise, yet powerful, query capabilities. Graphic commands have also been embedded into the code to illustrate the versatility of the tool. Although its current version is for interactive mode only, the next development stage will incorporate suggestions and recommendations by test users, and the operation of batch mode will be addressed.

This report describes the main features of TOPBASE, the adopted approach (Section II), its structure (Section III) and query language (Section IV). A collection of examples in Section V gives a closer look at the characteristics of the implementation. More detailed information can be found in the TOPBASE user's manual (Cunto and Mendoza 1991).

II. TOPBASE APPROACH

The TOPBASE approach is closely related to the physics of the OP atomic data and to the most frequent search patterns likely to be repeated by users and applications. It thus ensures a natural interface and efficient time/space performance.

TOPBASE manipulates data related to two main atomic properties: (i) bound states belonging to the non-relativistic term structure and (ii) dipole allowed transitions between given pairs of such bound states. The OP dataset volume and these two atomic properties condition TOPBASE to be structured into three entities: term energies (e), f-values (f) and photoionization cross sections (p). Although photoionization is a bound-state property, and it is therefore functionally dependent on the energy entity, the p entity is handled separately because of its large volume.

The energy entity e contains identified term energies for ionic bound states. Each bound state (level), i say, is assigned a record that is uniquely addressed by a key defined in terms of the following attributes:

nz - Chemical element nuclear charge;
ne - Ion electron number;
islp - The total quantum numbers of the spectroscopic series containing the i level, defined as

$$\mathbf{islp} = 100(2S+1) + 10L + P$$

where S is the total spin quantum number, L is the total orbital angular momentum quantum number and P the parity (0 for even and 1 for odd);
ilv - The i level energy position within its spectroscopic series.

Indexing in the e entity is organized such that a hierarchy of access is enforced. This means that a single element with nuclear charge **nz** gives rise to **nz** ions with **ne=1,nz** electrons; each ion has several spectroscopic series defined by their total quantum numbers **islp**; and each series contains several levels labeled by **ilv** in ascending energy order.

The f entity contains weighted oscillator strengths (gf-values) and wavelengths for allowed transitions between pairs of bound states, $i \rightarrow j$ say, contained in the e entity. Each transition is uniquely addressed by a key defined by the attributes:

nz - Chemical element nuclear charge;
ne - Ion electron number;
islp - The total quantum numbers of the spectroscopic series containing the i level;
jslp - The total quantum numbers of the spectroscopic series containing the j level;
ilv - The i-level energy position within its spectroscopic series;
jlv - The j-level energy position within its spectroscopic series.

[*] CMS and AIX are trade-marks of the IBM Corp.

Indexing in the *f* entity imposes a similar access hierarchy to that described for the *e* entity.

The photoionization entity *p* may be regarded as an addendum of the *e* entity. Both entities share the same accessing scheme, that is, each state key identifies a unique photoionization cross section. Since the cross section is a complicated function of photoelectron energy, the number of points varies among states, and therefore the *p* entity is voluminous and embodies $\sim$ 90% of the OP dataset. Thus, for both convenience and performance, the *e* and *p* entities are kept physically separated although logically they are closely related.

The design framework takes into consideration the types of queries most likely to be performed by users. Let us consider, for example, queries addressing the *e* entity (the three entities are accessed separately). If each key attribute is assigned a value, or a range of values specified in terms of its lower and upper bounds, the query is referred to as *isonuclear* if the nuclear charge is kept constant at `nz=nz1` and the ion electron number is given a range `ne=ne1,ne2`. Similarly, if the number of ion electrons is assigned a fixed value `ne=ne1` and the nuclear charge is given a range `nz=nz1,nz2`, the query is said to be of the *isoelectronic* type. If `nz=nz1`, `ne=ne1` and `islp=islp1,islp2` the query addresses multiple data belonging to a single ion, and it may be called an *ionic* query. In the case of a query with `nz=nz1`, `ne=ne1`, `islp=islp1` and `ilv=ilv1,ilv2`, data within a single spectroscopic series are accessed, i.e., a *spectroscopic* query. Finally, the user may be interested in states within an energy range `e=e1,e2`; this query is then referred to as an *energy* query. When searches are performed in the *f* entity, the query types are basically the same as those mentioned except that the energy query becomes a *wavelength* query. Queries may indeed get more complicated but the majority can be reduced to combinations of the above types.

Since data access is on a read-only basis, specific compact and efficient data structures have been devised to implement indexes and tables. Access-time efficiency is ensured through a physical design tailored to economize in the number of data transfers to main memory from the bulk of the database that resides in secondary storage. Data loading from secondary storage is really the main bottle-neck in system performance due to the volume of data that may be involved. This problem becomes eminent in smaller platforms, e.g., stand-alone workstations with large memories and fast CPU but relatively poor disk access times. For these reasons, the TOPBASE data manipulation scheme has been implemented at two levels: (i) searches in and expensive block data retrieval from secondary storage and (ii) cheap and versatile functions (sorting, column/row selections and exclusions and graphic displays) performed iteratively on the data in main storage in order to satisfy the user's ultimate requirements.

An important question concerns the extent of data manipulation capabilities that the package should offer beyond that of data retrieval and display. To find a proper answer one must analyse issues such as portability, the user's own personal processing of the retrieved data and the local availability of powerful tools external to the environment. We support the view that the user should be aware of a powerful environment where interaction with the data is encouraged. The present TOPBASE version allows the user to calculate a fairly large and expandable set of built-in functions of the stored atomic parameters. For instance, radiative lifetimes can be quickly computed from the stored f-values and displayed by the DBMS at user's request. The program can display and print data and also export/import them to disk files. Graphic displaying and plotting is implemented through interfaces with standard graphic packages. The VM/CMS version interfaces with the graphic package GDDM* and an interface with Exponent Graphics** is planned for the AIX version.

Points discussed above furnish support to the decision to develop a specific DBMS rather than choosing one available in the market. General purpose DBMSs are usually designed for transaction processing and are not suited for handling large volumes of data in numerically intensive modes. Because of their wide scope in data processing, generic systems include powerful facilities for data updating, recovery and security that are out of the context of read-only databases and therefore add unnecessary overheads. Finally, the performance of such systems is usually linked to hardware/software investments, which is not the case in TOPBASE.

III. GENERAL STRUCTURE AND DATA MANIPULATION SCHEME

The data are organized in three entities, the *e*, *f* and *p* entities, and a set of indexes all resident in secondary storage (see Figure 1). When TOPBASE is invoked, the indexes are loaded into main memory such that the

* GDDM is a trademark of the IBM Corp.

** Exponent Graphics is a trademark of IMSL Inc.

query processor is ready to use them for searching the required data in the bulk of the database. Indexes have been carefully structured so as to (i) reduce a single search to one disk access when performed in the e and f entities and to two disk accesses when it is carried out in the p entity (see Figure 1), (ii) optimize multiple searches assuming the query types described in section II and (iii) display tables of contents of the database.

Data compactness and fast access are two main features of TOPBASE. In order to meet such requirements, physical media such as main and secondary storage are managed jointly with the logical handling of the data. Two data structures in main storage have been implemented for this purpose: the *view* and the *table*. They are shown in Figure 2 and described briefly below.

A search in the database is performed according to user selected criteria that generate a subset of highly cohesive data sharing a common meaning. This subset is loaded into special buffers located in main memory to allow further manipulation. The data structure implemented in main memory to store the loaded subset is referred to as the *view*. Each view has an associated *descriptor* that registers the selection criteria and view bounds. Note that the basic idea of the view is to perform only one expensive search operation in secondary storage, and from that point on, to allow for inexpensive related queries on the highly cohesive data subset in main storage. Binary images of views can be archived in and restored from secondary storage. The package offers display facilities on different output devices, namely, the monitor, printer and disk files.

Logical reorganizations of the data stored in a view are possible through the concept of *tables*. A table is a vector array that enables or disables data items within the view, according to selection criteria, inclusion/exclusion facilities and sorting. Tables can be output on several devices (monitor, printer, disk files), and graphic displays of table columns and photoionization cross sections can also be rendered.

IV. QUERY LANGUAGE

Since TOPBASE is destined to be used on different platforms and from a large variety of terminals, its on-line user interface has been implemented as a command interpreter that recognizes a simple although powerful query language.

A command consists of a **<command_name>** usually followed by one or more **<arguments>** and, in most cases, by an <option> list. The command format takes the form

<command> ⟶ **<command_name> <arguments> <options>**

The **<command_name>** is an alphanumeric symbol based on a verb that describes the function the user wants the DBMS to perform. The present version offers five types of commands.

- *View* commands: perform functions on the view data structure;
- *Table* commands: perform functions on the table data structure;
- *Graphic* commands: display graphic output of tables and photoionization cross sections;
- *Index* commands: provide tables of contents of the database;
- *Constant* commands: list values of frequently used atomic constants.

The **<arguments>** are positional operands that specify the information on which the system operates when it performs a command. They can be an entity (e, f or p), a pair of row limits, or a list of entity attributes and entity functions. The <options> are sequences of attribute items used to specify the selection criteria required in an operation, the execution control of the command or for input/output routing. Each attribute item is specified by an attribute identifier and an attribute range. The latter may be given a single value or a range of values; when an attribute is omitted, or its range is left undefined, the whole attribute range is assumed. In the present version of TOPBASE there are four types of option lists.

- **<descriptor_options>**: they specify the atomic data subset required when a view is created and take the form of a list of key attribute ranges. The view descriptor is updated with the option ranges at every view creation. This option list is preceded by a left parenthesis, "(".
- **<selector_options>**: they select the view subset when tables are created. They share exactly the same format as **<descriptor_options>** but do not lead to updates of the view descriptor.
- **<graphic_options>**: they provide specifications required in graphic commands, e.g., number of plots/screen, scale ranges and types, headers, etc. This option list is also preceded by a left

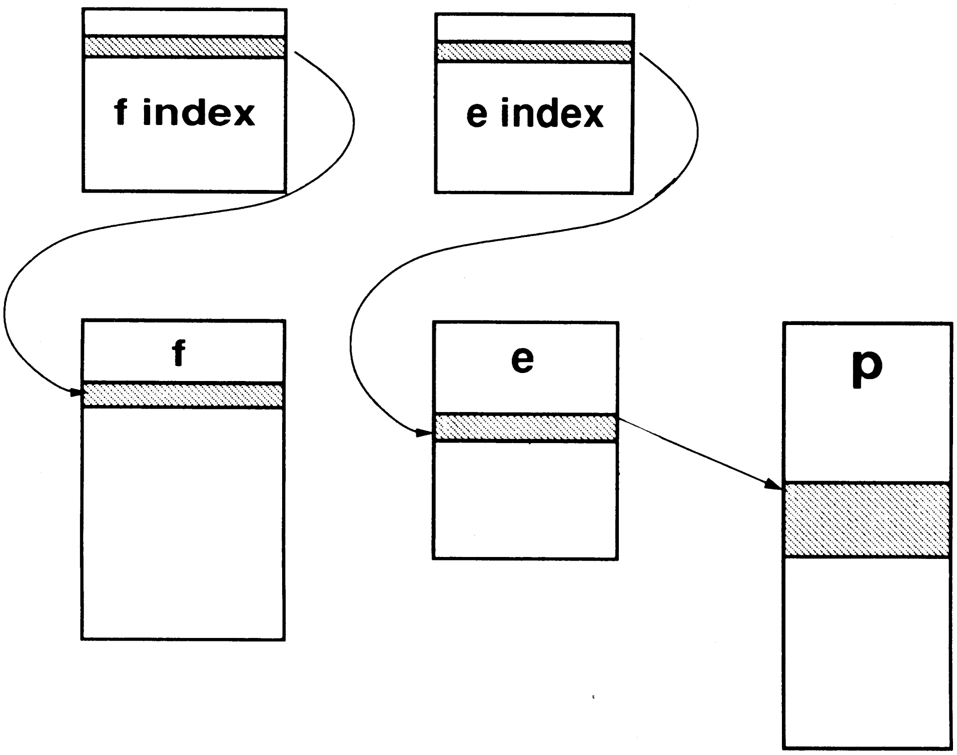


Fig. 1. TOPBASE general structure showing the *e*, *f* and *p* entities and the index tables. Disk accesses for single queries are also depicted.

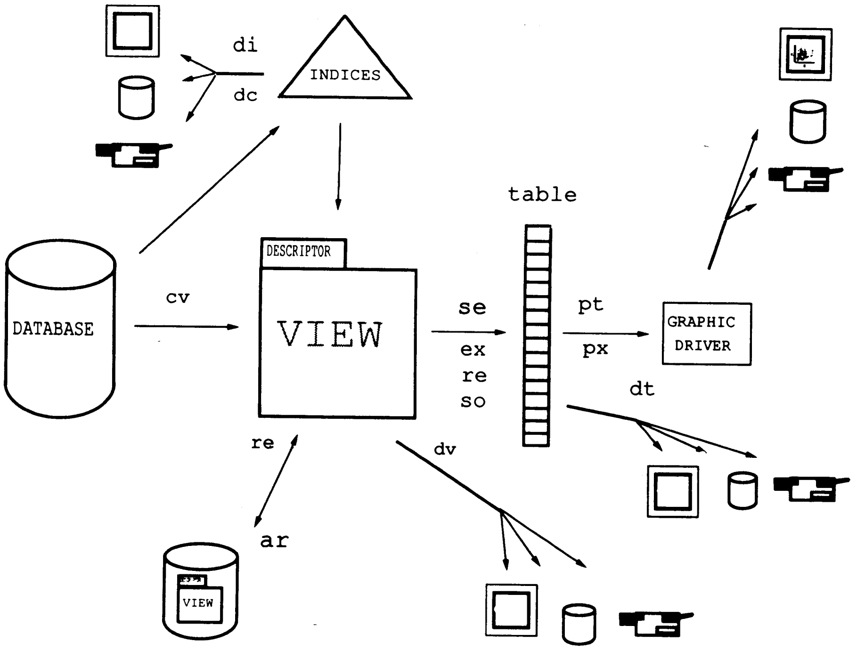


Fig. 2. TOPBASE data manipulation scheme showing the two main data structures, the *view* and the *table*, the display and graphic capabilities and query commands.

parenthesis, "(".

- `<input/output_options>`: they determine the I/O device required by the command. The input and output options are preceded by the "<" and ">" indicators respectively.

The following subsections provide a general description of the main commands. More detailed information is contained in the TOPBASE user's manual (Cunto and Mendoza 1991).

IV.1. View commands

A view is a user selected subset of one of the TOPBASE entities loaded into main memory for further manipulation (see section III). A structure called the *view buffer* has been devised to store the view in main memory. This view buffer groups vector arrays, one for each attribute, and it is headed by a descriptor that registers the attribute subdomains. There are five commands associated with views.

cv - The *create view* command resets the view buffer, computes the entry points in the specified entity according to the `<descriptor_options>`, loads the matching data into the view buffer and updates the descriptor associated with the view;

dv - The *display view* command lists a selected part of the current view on different output devices, acting on a range of rows and/or a view subset defined by a list of `<selector_options>`;

av - The *archive view* command unloads a binary image of the current view in secondary storage by supplying a file specification;

rv - The *restore view* command loads a previously archived view into main memory;

dd - The *display descriptor* command outputs the current view descriptor.

IV.2. Table commands

A table is a logical data structure defined from a view. Although only one table can be defined at a time, the aims of this data structure are (i) to allow fast and iterative modifications of the data currently stored in the view buffer, (ii) to avoid delays caused by recurrent secondary storage access and (iii) to structure data output according to user requirements. When a view is created or restored, the initial table comprises the complete view. Table rows can be selected, excluded, reinstated, sorted—all of them generating new tables—and displayed. There are five table commands.

dt - The *display table* command lists the attributes contained in, or the built-in functions computed from the attributes of, the active rows of the current view (inactive rows are not considered). The built-in functions for the different entities are as follows.

(i) For all the entities: record number (**i**); nuclear charge (**nz**); inverse of the nuclear charge (**zinv**); ion electron number (**ne**); i-level total quantum numbers (**islp**); i-level position (**ilv**); i-level configuration (**iconf**) and i-level statistical weight (**gi**).

(ii) For the e entity only: the level identification type T or C (**t**); its identification index (**it**); principal quantum number of the active electron (**ln**); orbital angular momentum of the active electron (**ll**); its identification ambiguity (**ac**); the energy relative to the ionization limit (**e**); the energy relative to the ground state (**te**); quantum defect (**qd**); effective quantum number (**eqn**) and its radiative lifetime (**rl**).

(iii) For the p entity only: the number of points in photoionization cross section (**np**).

(iv) For the f entity only: the j-level total quantum numbers (**jslp**); j-level position (**jlv**); j-level configuration (**jconf**); j-level statistical weight (**gj**); wavelength (**wl**); weighted f-value (**gf**) and weighted transition probability (**ga**).

se - The *select* command picks out a logical table from the current view. Initially, the table conforms with the complete view that has been loaded into memory, resulting from commands such as **cv** or **rv**. In subsequent selections, a new table is generated each time from the previous active table.

ex - The *exclude* command inactivates rows from a table.

re - The *reinstate* command reactivates a selection of excluded rows into the current table.

so - The *sort* command allows the user to sort the rows of the active table in ascending order of one of the floating point attributes of the active table, thus creating a new table. If no sort argument is given, the table is reset to its original order.

IV.3. Graphic commands

The current TOPBASE prototype offers two basic graphic commands acting upon table contents:

pt - The *plot table* command plots the column content of a pair of attributes or attribute functions of the active table;

px - The *plot cross section* plots the photoionization cross section of each state contained in the active table.

These two commands are differentiated since `pt` treats each active record as a point in the resulting graph while `px` plots a graph for each active record. One, two or four graphs/screen (graphs/page) may be plotted in the latter case. In table plotting (`pt`) the abscissa and the ordinate must be specified, and they can take most table attributes or attribute functions listed in section IV.2. When plotting cross sections (`px`) the abscissa can be the photoelectron energy e (abscissa default) or the photon wavelength `wl`; the ordinate is the cross section `xsec`, the evident default.

IV.4. Index commands

The *display index* command (`di`) allows the user to display tables of contents of the data included in TOPBASE for each ion system. This information is valuable before considering creating views of the specific entities. They provide summaries of target representations, number of SLP spectroscopic series and states in the e entity, and number of $SLP - SLP'$ transition series and transitions in the f entity.

IV.5. Constant command

TOPBASE can list values of frequently used atomic constants. At present, only atomic weights, Rydberg constants and ionization potentials can be specified with the *display constant* command `dc`.

V. GUIDED EXAMPLES

Example 1

A frequent task in atomic physics is to plot quantum defects for states within a spectroscopic series, especially when comparing with experiment and to estimate numerical accuracy and the effect of series perturbers. In order to illustrate the TOPBASE features, we have chosen for this example the *np* $^2P^0$ series of the astrophysically important C II ion.

The first operation is to access the energy entity e and to load into memory (i.e., create a view) the $^2P^0$ states of C II . This is accomplished by issuing the command

```
cv e (nz=6 ne=5 islp=211
```

Note that the `islp=211` is determined by the relation

$$\texttt{islp} = 100(2S+1) + 10L + P \ ,$$

where $S = 1/2$ is the total spin quantum number, $L = 1$ is the total orbital angular momentum quantum number, and $P = 1$ denotes odd parity. As a result, 11 states are loaded into the view buffer from disk. Their attributes may be displayed on the screen with the commands **dv** (display view), or **dt** (display table) since the view has not been logically modified, giving

```
 I  NZ  NE  ISLP  ILV   T  IT  LN  LL  AC            E(RYD)

 1   6   5   211    1   T   1   2   1           -1.78769E+00
 2   6   5   211    2   T   1   3   1           -5.90082E-01
 3   6   5   211    3   T   1   4   1           -3.10093E-01
 4   6   5   211    4   C   2   0   0           -2.42561E-01
 5   6   5   211    5   T   1   5   1           -1.89475E-01
 6   6   5   211    6   T   2   3   0           -1.63384E-01
 7   6   5   211    7   T   1   6   1           -1.24104E-01
 8   6   5   211    8   T   1   7   1           -9.01943E-02
 9   6   5   211    9   T   1   8   1           -6.82796E-02
10   6   5   211   10   T   1   9   1           -5.34454E-02
11   6   5   211   11   T   1  10   1           -4.29586E-02
```

The table lists for each state its term energy and configuration assignment. For a T-type state (see Mendoza 1992), IT denotes the corresponding target-state index, and LN and LL represent the active-electron n and l quantum numbers. For a C state, IT is an index in a list of configurations for the $(N+1)$-electron system. We note that there two states that do not belong to the *np* series, namely ILV=4 and ILV=6. The two records can be logically excluded from the view, thus creating a table, by typing the commands

```
ex (ilv=4
ex (ilv=6
```

The command

```
dt nz ne islp ilv e qd iconf > prt
```

is used to print a custom-made table containing the actual quantum defects and the electron configurations. It prints the table

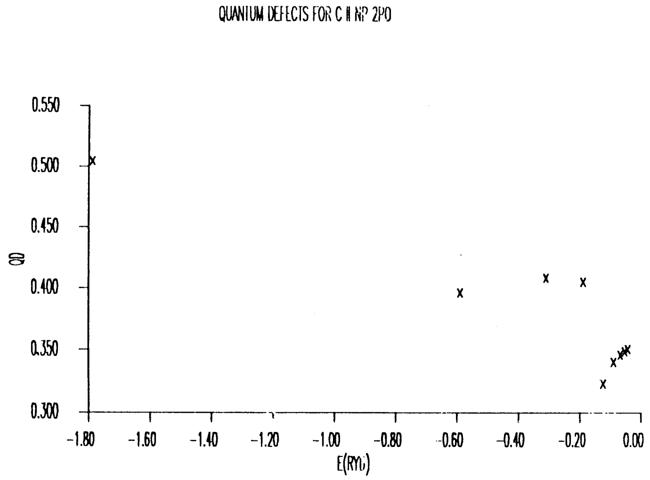
```
NZ  NE  ISLP  ILV        E(RYD)          QD  ICONF

 6   5   211    1  -1.78769E+00   5.042E-01  2S2 2P
 6   5   211    2  -5.90082E-01   3.964E-01  2S2 3P
 6   5   211    3  -3.10093E-01   4.084E-01  2S2 4P
 6   5   211    5  -1.89475E-01   4.053E-01  2S2 5P
 6   5   211    7  -1.24104E-01   3.228E-01  2S2 6P
 6   5   211    8  -9.01943E-02   3.405E-01  2S2 7P
 6   5   211    9  -6.82796E-02   3.461E-01  2S2 8P
 6   5   211   10  -5.34454E-02   3.488E-01  2S2 9P
 6   5   211   11  -4.29586E-02   3.505E-01  2S2 10P
```

Finally, a plot of quantum defects (**qd**) vs. energy (**e**) for states contained in the table is displayed with the graphic command

```
pt e qd (fm=m,nl hd='QUANTUM DEFECTS FOR C II NP 2PO'
```

which displays the following headed graph with marked points but no joining lines



Example 2

We want to find the largest absorption gf-values in C, N and O for transitions with wavelengths in the range 3000 Å to 3200 Å. The required data reside in the *f* entity, and are loaded into memory with the command

```
cv f (nz=6,8 e=3000.0,3200.0
```

which creates a view with 441 transitions. The initial table, which overlaps the created view, is sorted in ascending order of gf-values with the command

```
so gf
```

and displayed with the command

```
dt
```

giving the following screen display

```
   I  NZ  NE  ISLP  JSLP   ILV  JLV         GF      WL(A)
 817   8   7   421   410     1    4  -2.69E+00  3.161E+03
 894   8   6   311   310     2    3  -2.48E+00  3.055E+03
 979   8   6   521   520     1    1  -2.28E+00  3.079E+03
1453   8   6   321   320     3    7  -1.47E+00  3.186E+03
1481   8   4   320   311     4    8  -1.43E+00  3.128E+03
1896   6   2   311   300     4    5  -1.03E+00  3.059E+03
1931   8   4   321   310     3    5  -1.00E+00  3.095E+03
2033   6   2   320   311     3    5  -9.50E-01  3.043E+03
2083   7   3   211   200     4    5  -9.22E-01  3.163E+03
2256   8   4   331   320     3    6  -8.04E-01  3.084E+03
2364   8   6   320   311     5   16  -7.50E-01  3.185E+03
2546   7   3   220   211     3    5  -6.55E-01  3.052E+03
2974   8   4   120   111     5    8  -5.04E-01  3.198E+03
3102   7   4   311   300     5    4  -4.70E-01  3.074E+03
3418   8   4   320   311     6   11  -3.91E-01  3.000E+03
3472   8   4   111   100     6    7  -3.79E-01  3.086E+03

DO YOU WANT TO SEE MORE? ==>
```

Positive answers to the question will lead to a complete scroll of the table. Interruptions of this process will return the interface to the command line.

Example 3

In this example, we want to plot the photoionization cross sections of the ground states of the first four Mg-like ions. This requirement may be fulfilled by issuing the command

```
cv p (ne=12 ilv=1
```

which creates a view with the cross sections of the ground states of the seven Mg-like ions contained in the *p* entity. The initial table is displayed with command

```
dt
```

which leads to the following display on the screen

```
I  NZ  NE  ISLP  ILV       E(RYD)    NP

1  12  12   100    1  -5.64727E-01  669
2  13  12   100    1  -1.38689E+00  335
3  14  12   100    1  -2.46589E+00  257
4  16  12   100    1  -5.34104E+00  293
5  18  12   100    1  -9.07903E+00  275
6  20  12   100    1  -1.37687E+01  272
7  26  12   100    1  -3.35246E+01  234
```

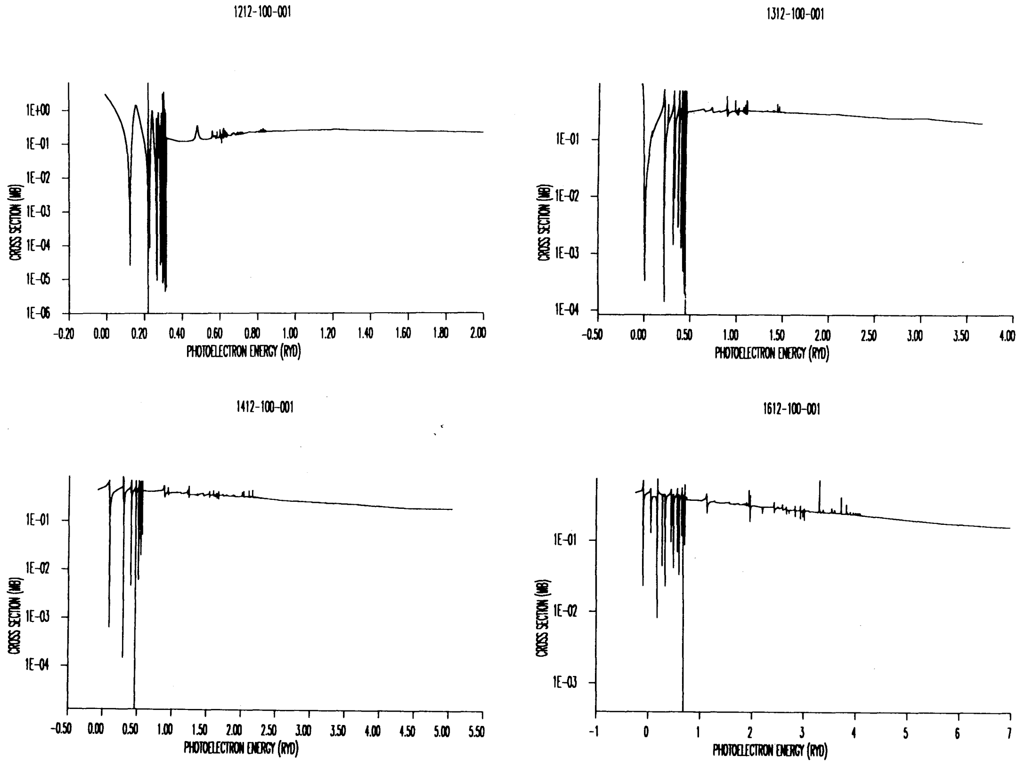
As we are only interested in the the first four ions, they may be selected with the command

```
se 1 4
```

Finally, for comparative purposes, the four cross sections can be plotted on a single graph by typing the command

```
px (ng=4
```

obtaining the following graphic screen display



Example 4

The command

```
di f (nz=6 ne=4,5
```

provides a summary of the data contained in the *f* entity for C III and C II (three-column format)

```
NZ = 6    NE = 4    NSLP-SLP = 18    NTRAN = 1376

 I  ISLP-JSLP  NTRAN   I  ISLP-JSLP    NTRAN   I  ISLP-JSLP  NTRAN

 1   111 100    121    2   111 110       88    3   120 111    110
 4   121 110     64    5   121 120       80    6   130 121     56
 7   131 120     80    8   131 130       56    9   140 131     48
10   311 300     99   11   311 310       99   12   320 311     99
13   321 310     72   14   321 320       72   15   330 321     56
16   331 320     72   17   331 330       56   18   340 331     48
```

```
NZ = 6    NE = 5    NSLP-SLP = 24    NTRAN = 2222

 I  ISLP-JSLP  NTRAN   I  ISLP-JSLP    NTRAN   I  ISLP-JSLP  NTRAN

 1   210 201     80    2   211 200       99    3   211 210    110
 4   220 211    110    5   221 210       90    6   221 220     90
 7   230 221     63    8   231 220       70    9   231 230     49
10   240 231     42   11   241 230       42   12   241 240     36
13   410 401     81   14   411 400      128   15   411 410    144
16   420 411    224   17   421 410       72   18   421 420    112
19   430 421     56   20   431 420      182   21   431 430     91
22   440 431    143   23   441 430       42   24   441 440     66
```

Example 5

The command

```
dc eip (nz=7 ne=3,6
```

lists the ionization potential for N II to N V

```
NZ = 7 NE = 3 EIP = 7.18774E+00 RYD
NZ = 7 NE = 4 EIP = 5.68369E+00 RYD
NZ = 7 NE = 5 EIP = 3.47720E+00 RYD
NZ = 7 NE = 6 EIP = 2.17775E+00 RYD
```

VI. FINAL REMARKS

As soon as TOPBASE was in working order, it began to be used in the lengthy process of the OP data checking. Program debugging evolved jointly with data correction, which may be considered a sound approach. Such auditing activities will continue as new data are calculated and incorporated in the OP atomic database. TOPBASE is therefore set to keep an active role in future data maintenance activities.

Batch access facilities are planned for the second version of the program. Perhaps, this access mode will account for more than 80% of its total use. In addition, facilities to create specific sub-databases will be incorporated to allow for more specialized and efficient use of the data, and the user will be given the possibility to define and execute mathematical expressions of rows and columns.

An important issue in the near future involves the distribution of the whole package, that is, of the OP atomic database and the TOPBASE DBMS. We recommend that a data center should assume the responsibility of both data and software maintenance. This proposal would also enforce standardization criteria that would enable the user community to count on a framework suitable for intensive sharing of applications, data and results.

ACKNOWLEDGEMENTS

This work has been supported by IBM Venezuela. We are grateful to Professor M.J. Seaton, Dr. K.A. Berrington and Dr. C.J. Zeippen for useful comments regarding many aspects of TOPBASE.

REFERENCES

Allard, N., Artru, M.C., Lanz, T., and Le Dourneuf, M. 1990, *Astron. Astrophys. Suppl. Ser.*, **84**, 563.

Berrington, K.A. 1986, in *Atomic Data Workshop: Low Energy Collision Theory Techniques for Atomic Excitation and Radiative Data*, W.B. Eissner (ed), DL/SCI/R24 (Daresbury Laboratory, UK), 38

Burgess, A., Mason, H.E., and Tully, J.A. 1989, *Astron. Astrophys.*, **217**, 319.

Burgess, A., Chidichimo, M.C., Mason, H.E., and Tully, J.A. 1991, *J. Physique IV*, **C1**, C1-303.

Butler, K., Mendoza, C., and Zeippen, C.J. 1990, in *Atomic Spectra and Oscillator Strengths for Astrophysics and Fusion Research*, J.E. Hansen (ed.), (North- Holland, Amsterdam), 124.

Butler, K., Mendoza, C., and Zeippen, C.J. 1991, *J. Physique IV*, **C1**, C1-135.

Cunto, W., and Mendoza, C. 1991, *TOPBASE User's Manual*, Report CSC-02-91 (IBM Caracas Scientific Center, Venezuela).

Eissner, W.B. (ed) 1986, *Atomic Data Workshop: Low Energy Collision Theory Techniques for Atomic Excitation and Radiative Data*, DL/SCI/R24 (Daresbury Laboratory, UK).

Eissner, W., and Kingston, A.E. (eds.) 1988, *Atomic Data Workshop: Assessment of data for photo-ionization and photo-excitation and for electron impact excitation of atomic ions*, (Daresbury Laboratory, UK).

Hansen, J.E. (ed) 1990, *Atomic Spectra and Oscillator Strengths for Astrophysics and Fusion Research*, (North-Holland, Amsterdam)

Hummer, D.G. and Mihalas, D. 1988, *Ap. J.*, **331**, 794.

Martin, W.C. 1992, *Lecture Notes in Physics*, in press.

Mendoza, C. 1992, *Lecture Notes in Physics*, in press.

Merts, A.L., Mann, J.B., Robb, W.D., and Magee, N.H. 1980, *Los Alamos Scientific Report* LA-8267-MS.

Seaton, M.J. 1987, *J. Phys. B*, **20**, 6363.

Seaton, M.J., Zeippen, C.J., Tully, J.A., Pradhan, A.K., Mendoza, C., Hibbert, A., and Berrington, K.A. 1992, *Rev. Mex. Astron. Astrophys.*, this volume.

Zeippen, C.J., and Le Dourneuf, M. (eds.) 1991, *Third Atomic Data Workshop*, *J. Physique IV*, **C1**.

W. Cunto and C. Mendoza: IBM Venezuela Scientific Center, P.O. Box 64778, Caracas 1060A, Venezuela.