706 J. Chem. Inf. Comput. Sci. 1996, 36, 706−710

Spreadsheet and Computing Language for a Chemical Application†

V. Viossat and A. Dereigne

Maitrise de Chimie, Université Pierre et Marie Curie BC65 4 Place Jussieu, Paris, France

Received September 14, 1995⊗

Recently a new teaching in computer science has been created for chemistry students. Our goal is to show that programming or using a spreadsheet can be a tool in understanding chemistry. The behavior of atoms studied in the course of general chemistry (atomic properties, periodic table, wave functions) is the selected subject. The students must construct and use a database of elements which permits them to solve relevant problems (research of the alkali metals, displaying of 2D and 3D graphics to represent the periodic properties, electronic density, and radial wave and angular functions) with a spreadsheet and a computing language.

INTRODUCTION

Recently a new teaching in computer science[1−3] has been created for chemistry students. Using software (text processor, database, and spreadsheet) is an important skill for a chemistry student, but it is also essential to possess computer literacy in our increasingly technical world.

Commercial spreadsheet programs have achieved immense popularity in the business world. In the future students will have access to spreadsheets in employment situations. Using a spreadsheet in chemistry is important; indeed, numerous papers were published between 1987 and 1995. Spreadsheet applications have different topics: quantum mechanics,[4,5] kinetics,[6−11] electrochemistry,[12−17] thermodynamics,[18] and laboratory operations.[19−21]

One of our goals is to show that programming or using a spreadsheet can be a tool in understanding chemistry. In this exercice, we considered the behavior of atoms studied in the course of general chemistry (atomic properties, periodic table, and wave functions).

A specific grammar must rule the spreadsheet and high level language (Pascal). Therefore, it seems very important to us that the students understand well that a spreadsheet, or a programming language are not disjointed sets. In fact, many softwares and programming languages have many common points.

PASCAL AND SPREADSHEET LANGUAGES

This programming course introduces to the students the tools to solve a problem with a computing language and a spreadsheet.

The different selective topics are useful to

−recall concepts and programming methods,
−study the Pascal language thoroughly (data structures and instructions),
−study the spreadsheet (Quattro Pro) with an innovative method (syntax of the language),
−establish similarities and differences between Pascal and Quattro languages.

Table 1. Pascal and Spreadsheet Comparison: (a) Program, (b) Data Types and (c) Instructions

a)

Pascal .... -------> sequence of instructions

-----------> grammar

Spreadsheet -------> succession of actions

b)

types	PASCAL	SPREADSHEET
standard	integer,**real,character**, boolean	**real, character**
structured	**array,string,record,file**	**array,string,record , file**

c)

instructions	PASCAL	SPREADSHEET
assignment	direct	direct {LET location,value:type}
	read	{GETLABEL prompt, location} {GETNUMBERPrompt,location}.
	expression	formula
	function	@function
repetition	while.....do... repeat ....until.. for ....to....do...	{FOR Counterloc, start#, stop#, step#,startloc}
decision	if...then...else goto	{IF condition } {BRANCH location}

Let us recall that the resolution of a problem requires

−defining the problems,
−selecting and representing algorithms,
−coding algorithms.

Only after unambiguously defining the problem, organizing a solution, and sketching out the step-by-step details of the algorithm can we begin to code. This translation requires a language; we will employ Pascal and the spreadsheet languages.

The comparison between Pascal and spreadsheet is presented in Table 1 regarding the particular program, data types, and instructions.

Program. The sequence of instructions or succession of actions are similar (Table 1a).

† Paper presented at the XIth International Conference on Computers in Chemical Research and Education, Paris, France, July 17−21, 1995.
⊗ Abstract published in *Advance ACS Abstracts*, April 1, 1996.

S0095-2338(95)00119-3 CCC: $12.00

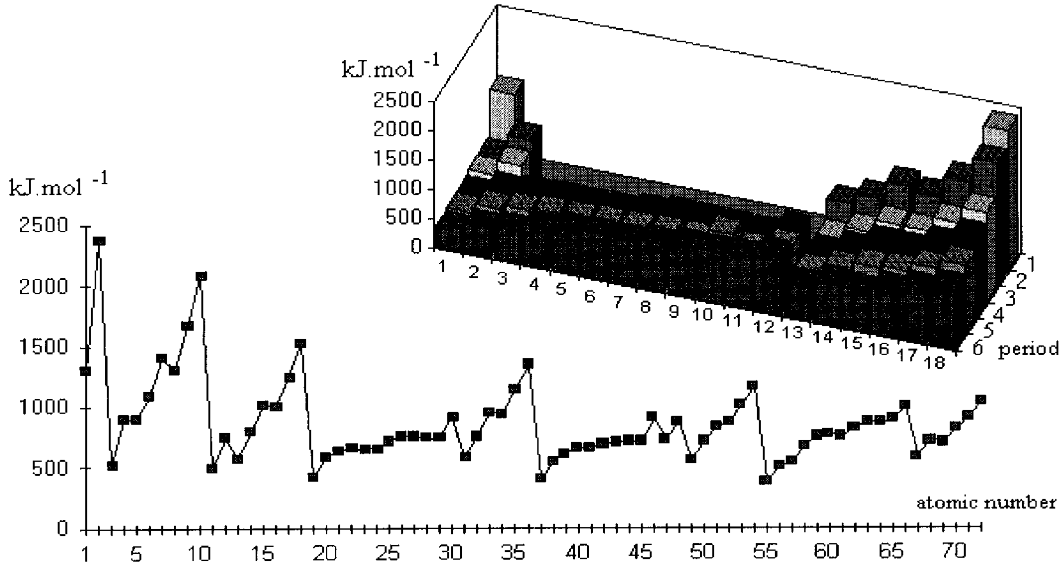


Figure 1. First ionization energy versus atomic number.

Data Types. A data type is described by a set of rules that give the specific format for elements of that type. Pascal types are well-known, whereas the concept of data spreadsheet types only virtually exists. As shown in Table 1b, the integer and Boolean types are not explicit for the spreadsheet; indeed the integer type is included in the real type, and the Boolean type is similar to an integer. The false value is zero, and the true one is any integer. The structured types are subjacent.

Instructions. Similarly, Pascal instructions (assignment, repetition, and decision) are very well-known. The same types of instructions are present in the spreadsheet (Table 1c).

The assignments in the worksheet can be realized by

- entering values (labels or numbers) directly or by using the read procedure which is equivalent to macro command (i.e., GETLABEL)
- entering formulas (arithmetic, logical, and text),
- using functions.

In the spreadsheet, the only intruction for repetition is "{FOR counterloc, start#, stop#, step#, startloc}", whereas two instructions are similar for the decision.

CHEMICAL APPLICATION

For this application, the subject is the creation of a chemical database based on the periodic table using Pascal and spreadsheet languages. This topic has been developed by several authors with interactive databases.[22−26] Hodson[27] presents a pedagogical experience in which the students themselves trained with precompiled databases, then updated, modified, and built new chemical databases for the elements. The students were encouraged to explore their own ideas, manipulate information, and test their hypotheses.

Our approach is different and more complete since the students constructed a worksheet with the different functions and used the worksheet as a database by writing a program or using spreadsheet.

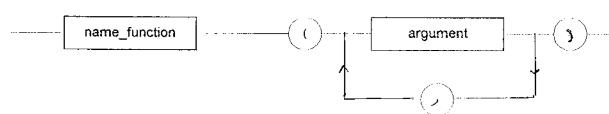
For the construction of a chemical database, the symbol, atomic number, number of isotopes, isotopic mass, isotopic abundance, and ionization energy have been introduced and then the period, atomic mass, and repartition of electrons in the two outermost shells are calculated (Table 2). For example, the period is determined from the atomic number (Z → 86) by Pascal and spreadsheet functions:

Table 2. Spreadsheet Display

atomic number	symbol	period	s	p	d	s	p	atomic weight	ionization energy (kJ·mol^{-1})	alkali
1	H	1	1	0	0	0	0	1.00798	1312	
2	He	1	2	0	0	0	0	4.0026	2372.3	Li
3	Li	2	2	0	0	1	0	6.94093	513.3	Na
4	Be	2	2	0	0	2	0	9.01218	899.4	K
5	B	2	2	0	0	2	1	10.81	800.6	Rb
6	C	2	2	0	0	2	2	12.011	1086.2	Cs
7	N	2	2	0	0	2	3	14.0068	1402.3	

* *Pascal language :*

syntax function:




```
function fperiod(num:integer):integer;
      begin if num<3 then fperiode:=1 else if num<11 then
                  fperiod:=2 else if num<19 then fperiod:=3.....end;
   begin
      write( 'give the atomic number'); readln(num); period:= fperiod(num)
....end.
```

* *Spreadsheet language:*

syntax: ***@IF function***


```
period<-
@IF(num<3;1;@IF(num<11;2;@IF(num<19;3;@IF(num<37;4;@IF(num<55;5;
6)))))
```

In order to use the data, different topics are useful for instance to

- find the element presenting the largest ionization energy,
- find elements belonging to the same chemical family,
- search an element corresponding to a specific electronic configuration,

−display graphs in 2D and 3D to represent the periodic properties and wave functions.

This last point is developped only with the spreadsheet.

Two examples (search the alkali family and graphs 2D and 3D) have been developed.

Search the alkali family with testing the filling of the (n−1)d and ns subshells from the database to avoid the particular cases (Cr, Cu, Nb) among the 86 elements.

Pascal Language. The block_d and block_s are the sets of electrons in the (n−1)d and ns subshells; alkali are the set of alkali elements.

```
 procedure alkaline (block_s, block_d, symbol:tab;var alkali:tab) ;
var i, j: integer;
begin
j:=1;
for i:=1 to 86 do if (block_s[i]=1 and block_d[i]=0)
                       then begin  alkali[j]:=symbol[i];
                                   j:=j+1;
                            end;
```

Spreadsheet Language. The different steps are

−define line_res, cell to contain the value of the line to fill alkali element,

−test the contents of the block_s and block_d; (label inst_ suiv),

−fill the block_alkali if the test is verified. (label display),

```
          {LET line_res;0} {FOR line;4;89;1;inst_suiv}
Inst_Suiv {IF @cellindex("contents"; block_s;0;line-4)=1} {IF @cellindex("contents";
            block_d;0;line-4)=0} {BRANCH display}
Display    {LET line_res;line_res+1}
           {PUT block_alkali;0;line_res-1;
                 @cellindex("contents";block_symbol;0;line-4)=0}
```

The result of the execution is presented in Table 2.

Display Graphs in 2D and 3D. This point is developed only with the spreadsheet. Two applications are presented

−the first ionization energy as a function of the atomic number Z (Figure 1),

−the different probabilities of finding electron in the quantum-mechanical description of the atom (Figures 2 and 3).

For this last point, data are relative to the atomic orbital for hydrogenic atoms (orbitals s, p, and d).

Different states of the electron correspond to different wave functions which can be represented by the radial factor $R(r)$ and the angular factor (Θ*Φ).

The probability of finding an electron at each point is represented by the squares of the radial functions $R(r)$ (Figure 2). These plots give the electronic density as a function of distance from nucleus measured along a given direction (at fixed θ and φ). It is seen that the s orbitals have a large density at the nucleus, while the p and d orbitals have zero probability.

The probability of finding an electron at a distance r is determined by the square of the function (r*$R(r)$), the radial distribution functions. They give the probability per unit length that the electron will be found at distance r from the nucleus, regardless of direction. The number of maxima in these functions is (n−l) (Figure 2).

The angular distribution of the electron can be shown by plots of the function (Θ*Φ) in one or more planes through the origin. The function (Θ*Φ), for fixed r, represents the

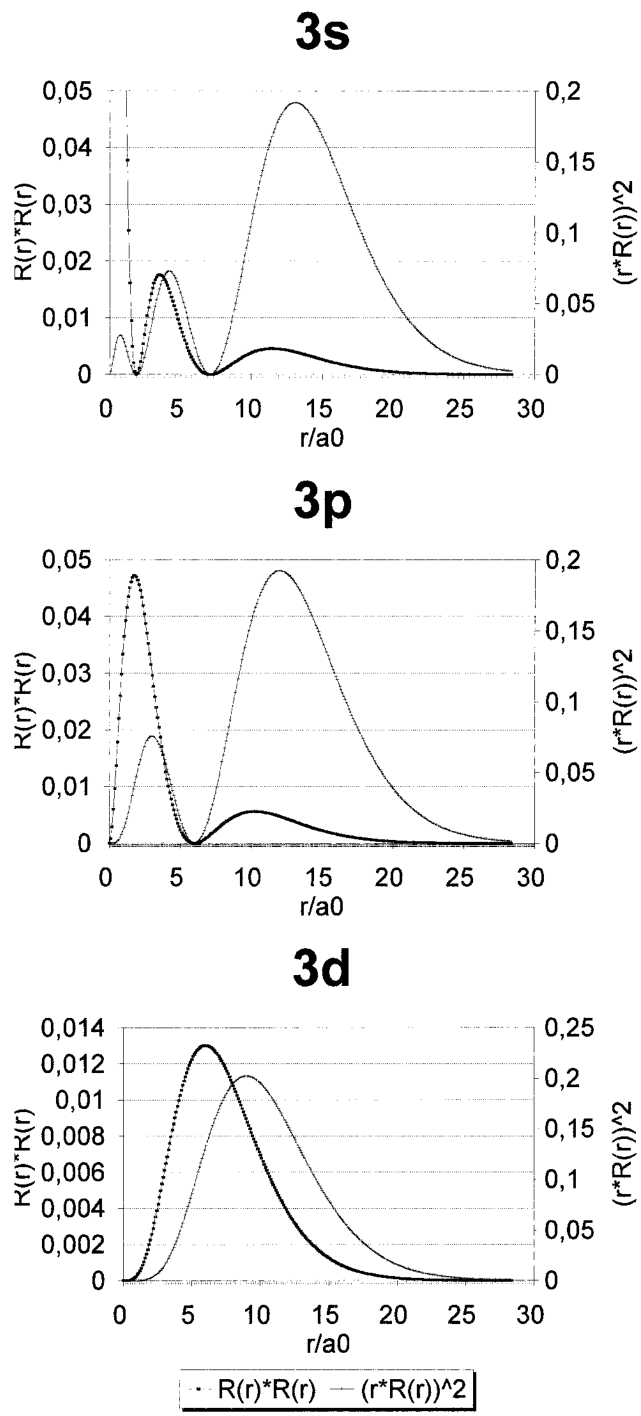


Figure 2. Electronic density $R^2(r)$ (Å^{-3}) and radial distribution function $(r*R(r))^2$ (Å^{-1}).

relative value of the orbital in the particular θ and φ directions (Figure 3).

The s orbitals have spherical symmetry.

For the p orbitals, the result is two circular lobes tangential to the horizontal axis, the upper lobe corresponding to the positive values of cos(θ) and the lower lobe to the negative values.

For the d orbitals, the d(z^2) function is particular and the d(x^2−y^2), d(xy), d(xz), and d(yz) functions have the same shape but with lobes oriented along different and precise directions. The d(z^2) function displays large positive lobes along the z axis and small negative lobes in the xy plane.

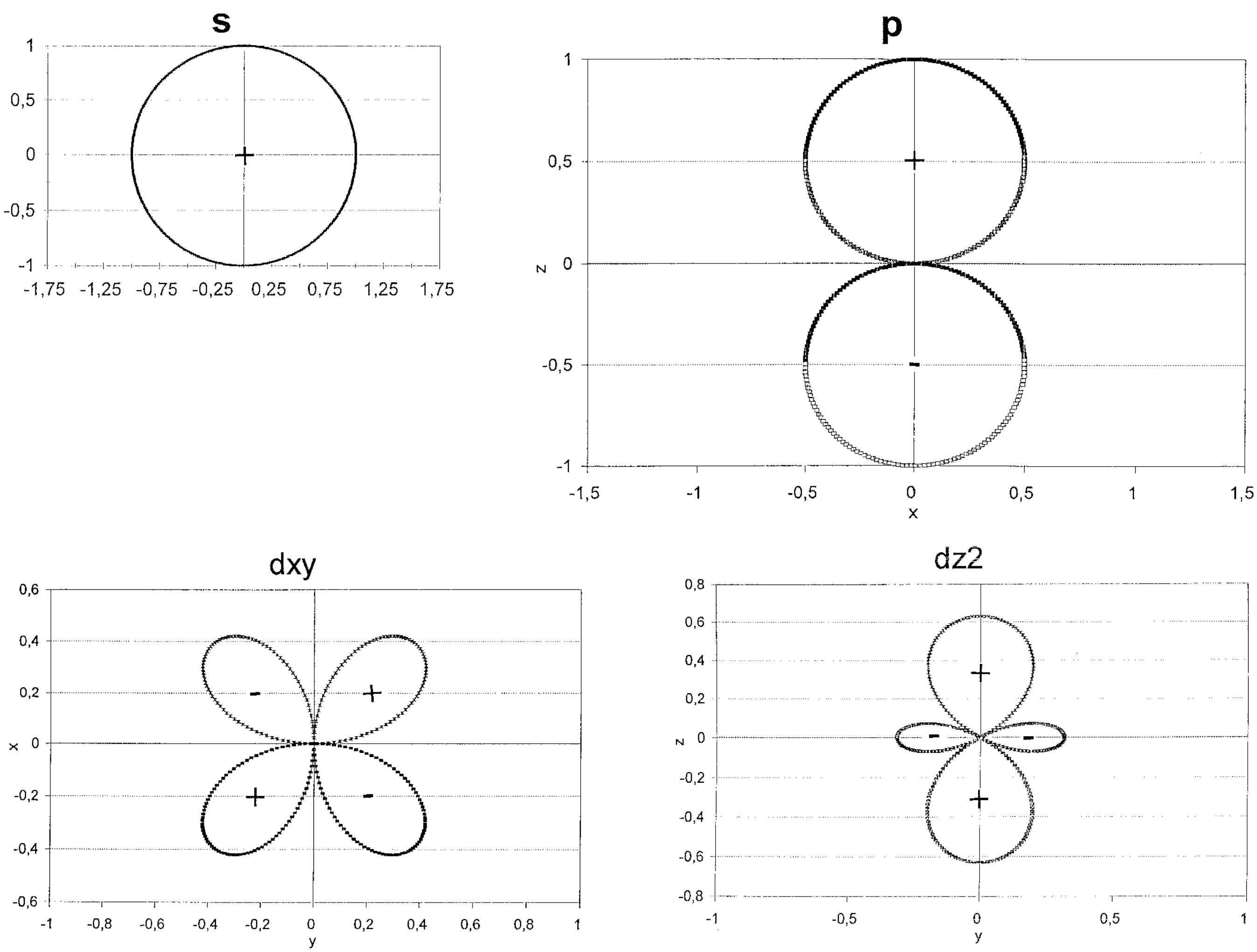


Figure 3. Electronic angular distribution.

CONCLUSION

The students built a database about the elements with the spreadsheet Quattro Pro or a programming language. They showed great enthusiasm for the course in which they discovered the similarities and the differences of these tools.

After the data base construction, the students solved a few problems:

- −finding the element presenting the largest ionization energy,
- −search of the family for a given group,
- −displaying different graphs in 2D or 3D (periodic properties, electronic density, radial wave and angular functions).

It is worth mentioning that the spreadsheet is easier and more attractive to use than the programming language; meanwhile macros literacy requiers a serious initiation and apprenticeship.

These innovative pedagogical experiences provide

- −acquisition of knowledge (Pascal and spreadsheet languages with introduction of a corresponding grammar),
- −complete clarification of knowledge in general and quantum chemistry.

REFERENCES AND NOTES

(1) Viossat, V.; Dereigne, A.; Brosset, F. Nouvelles approches du tableau de Mendeleiev avec l'informatique. *Actes des 6[ièmes]Journées sur les méthodes informatiques dans l'enseignement de la Chimie·Paris* **1993**, 34.

(2) Viossat, V.; Dereigne, A.; Brosset, F. Apprentissage de la programmation. Application en Chimie. *Revue de l'EPI* **1994**, *73*,171.

(3) Viossat, V.; Dereigne, A.; Brosset, F. Grammaire d'un tableur et application chimique. *7ième MIEC, Université de Bourgogne (imprim service)* **1995**.

(4) Rioux, F. Numerical solutions for Schrödinger's equation. *J. Chem. Educ. Software* **1990**, 3B(2), 13−15, 21−8, 33−58.

(5) Knudson, Stephen, K. Direct solution of Schrödinger equation for vibration. *J. Chem. Educ.* **1991**, *68*, 2, A39.

(6) El Shayal, I. Solving differential equations on a spreadsheet: the Euler Method. *Chem. Eng*. **1990**, *97*(8), 149−50.

(7) El Shayal, I. Solving differential equations on a spreadsheet: The Runge Kutta method. *Chem. Eng*. **1990**, *97*(8), 153−4.

(8) Coe, D. A. Numerical solutions of kinetics equations on a spreadsheet. *J. Chem. Educ.* **1987**, *64*,*6*, 496−7.

(9) Shumaker, R. C.; Boxenbaum, H.; Thompson, G. A. Absplots: a Lotus 1 2 3 spreadsheet for calculating and flotting drug absorption rates. *Pharm. Res.* **1988**, *5*(4), 247−8.

(10) Elvin, H. A cumulative count method for determining the half-life of Barium 137 and Gallium 68 radioactive isotopes. *J. Chem. Educ.* **1991**, *68*, 2, A41.

(11) Marzzacco, C, J. Spreadsheet simulation of a simple kinetic system. *J. Chem. Educ.* **1993**, *70*,12, 993.

(12) Salit, M. L. Spreadsheet programs as tools for analytical chemistry. *An. Chem*. **1988**, *60*(11), 731A−732A, 735A−738A.

(13) James, O.; Currie, R. V., Jr.; Whiteley, Jr. Advanced titration plots using spreadsheet. *J. Chem. Educ.* **1991**, *68*,11, 923.

(14) Chesick, J. P. Spreadsheet in general chemistry. Their introduction and application to complex titration. *J. Chem. Educ.* **1994**, *71*,11, 934.

(15) Carter, D. R.; Frye, M. S.; Mattson, W. A. A direct approach for solving simultaneous equations combining a basic understanding of

equilibria with the convenience and power of a spreadsheet. *J. Chem. Educ.* **1993**, *70*,2, 67.

(16) Breneman, G. L.; Parker, O. J. Spreadsheet titration of diprotic acids and bases. *J. Chem. Educ.* **1992**, *69*,1, 46.

(17) Schwartz, L. L. Spreadsheet to predict values and detect systematics errors, uncertainty of a titration equivalence point. A graphical method using.spreadsheet. *J. Chem. Educ.* **1992**, *69*, 11, 879.

(18) Coe, D. Using a spreadsheet to calculate the fugacity of a Van der Waals gas. *J. Chem. Educ*. **1987**, *64*,2, 137−9.

(19) McGee, W. W.; Mattson, G. Using an electronic spreadsheet in the design of exercices for a polymer laboratory course. *J. Chem. Educ.* **1993**, *70*,9, 756−8.

(20) Sunheim, B. R. Spreadsheet model, column operations. *J. Chem. Educ.* **1992**, *69*,12, 1003.

(21) Sunheim, B. R. Modelling a thermostated water bath with a spreadsheet. *J. Chem. Educ.* **1992**, *69*,8, 650.

(22) Kotz, J. C. KC? Discoverer. A computer program for descriptive inorganic chemistry. *J. Chem. Educ*. **1989**, *66*,9, 750.

(23) Banks, A. J. The periodic table stack. *J. Chem. Educ*.: Software *1989*, 1C(1), 19.

(24) Rose, M. Video disc display program. *J. Chem. Educ*.: Software **1989**, 2A(2),13, 29, 35.

(25) Farris, M. The periodic table stack. *J. Chem. Educ.* **1989**, 66, 9, 756.

(26) Cabrol-Bass, D. Software. Propriétés des corps simples. Société Chrysis:France, 1993.

(27) Hodson, D. Using and constructing chemical databases. *Educ. Chem*. **1991**, 78.

CI9501195