

# Exploring Do-It-Yourself Approaches in Computational Quantum Chemistry: The Pedagogical Benefits of the Classical Boys Algorithm

Gabriele Orsini\*

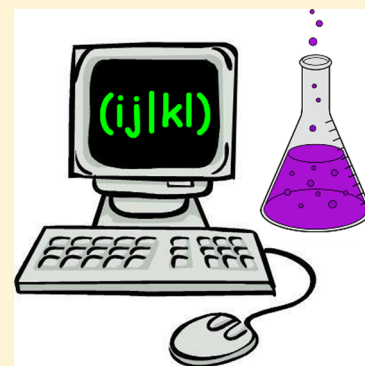
<sup>†</sup>Department of Chemical Engineering and Materials Science, University of Pisa, largo L. Lazzarino 1, 56122 Pisa, Italy

<sup>‡</sup>Industrial Technical School “G. Galilei”, via Campo d’Appio 90, 54033 Carrara, Italy

## Supporting Information

**ABSTRACT:** The ever-increasing impact of molecular quantum calculations over chemical sciences implies a strong and urgent need for the elaboration of proper teaching strategies in university curricula. In such perspective, this paper proposes an extensive project for a student-driven, cooperative, from-scratch implementation of a general Hartree–Fock self-consistent-field code. The project is primarily (but not exclusively) intended for a one-semester laboratory course in graduate physical-chemistry curricula: students are divided into working groups devoted to the implementation of specific subroutines, which are then gradually assembled in the final code. The resulting program is not limited to the overused, trivial case of two-electron diatomic systems, but can treat arbitrary closed-shell molecules with first- and second-row atoms at the STO-3G basis-set level and includes very useful features such as geometry optimization, population analysis, and calculation of dipole moments. The heart of the program is the by now historical algorithm devised by S. F. Boys for evaluating generic molecular integrals through repeated differentiation of fundamental integrals involving only s-type Gaussian orbitals. Though highly inefficient and completely outdated from a computational point of view, this method is reasonably fast for small- and even medium-size molecules (for instance, fixed-geometry calculations on methane and benzene take about 1 s and 6 min, respectively, on a modest home computer). More importantly, unlike newer advanced algorithms, it can be easily understood and implemented by students: thus, it becomes particularly appealing for a didactics based on reflexivity and active knowledge building, rather than on a comfortable but often premature and uncritical use of ready-made professional software packages.

**KEYWORDS:** Graduate Education/Research, Physical Chemistry, Computer-Based Learning, MO Theory, Quantum Chemistry, Theoretical Chemistry



## INTRODUCTION

As a matter of fact, despite its growing tremendous implications on both a theoretical and a practical level, computational quantum chemistry still occupies a rather marginal position in university curricula. Consequently, related teaching strategies have been developed to a very limited extent as compared to other branches of chemistry. Current didactic models for this subject are mostly focused on a direct approach of students to professional software packages such as GAUSSIAN,<sup>1–3</sup> SPARTAN,<sup>4–6</sup> GAMESS,<sup>7–9</sup> and many others. These powerful tools rely on decades of research and experience and are provided with friendly graphical interfaces which enable even the most inexperienced user to perform complex molecular calculations in very short times: with no doubt, they are an invaluable resource for an up-to-date education of chemists and material scientists, as well as of specialized engineers and biologists. However, a teaching strategy based exclusively on software training is quite a risky one. There is significant danger, in fact, that students assume a passive role in front of the computer and restrict themselves to mechanically providing inputs and receiving outputs in a black-box fashion, with little if any awareness of what the software really does.

Therefore, the introduction of an alternative, more pedagogically active dimension would be a very useful support for the ordinary software-oriented didactics. Particularly intriguing in this light is the exploration of “do-it-yourself” (DIY) approaches: that is, guiding students to build by themselves a from-scratch code in order to perform their own molecular calculations at the most general possible level. This may appear, at first glance, a rather hard and difficult teaching aim, but the educational advantages involved are certainly worthwhile:

- *Reflexivity.* Code writing is a deeply reflexive activity. Students are stimulated to think in a clear and orderly manner, deciding what and how the computer should exactly do in order to perform each given task: attention to every detail is crucial, as every single character of the code may be a potential source of error. Such work will make them far more aware of the procedures underlying molecular calculations and, ultimately, more critical in using professional softwares.

- *Interdisciplinarity.* Implementing computational quantum techniques is a very good opportunity for students to consolidate and increase not only chemical, but also physical, mathematical, and computer skills, with a significant expansion of their cultural horizon.
- *Gratification.* When, after a demanding work of writing and debugging, the code finally does what it is intended to do, a pleasant sense of “mission accomplished” pervades students, increasing their research spirit and confidence in their own intellectual potentialities. As a result, they will be stimulated to face any scientific problem with autonomous, original approaches, rather than to systematically rely on “prepackaged” solutions.

In recent and less recent years, other authors felt the need for enhancing student participation in computational-quantum-chemistry courses. Their didactic proposals, however, were generally limited to the trivial case of two-electron atoms<sup>10,11</sup> or diatomic molecules.<sup>12</sup> These simple systems are useful in order to illustrate the very principles of computational procedures, but are hardly expected to arouse genuine student interest.

In this paper, an extensive project is described for a student-driven, cooperative, from-scratch implementation of a Hartree–Fock (HF) self-consistent-field code capable of treating arbitrary closed-shell molecules with first- and second-row atoms at the STO-3G basis-set level. At the end of the work, students will be able to use their own “creature” on a virtually infinite variety of small- and even medium-size molecules, obtaining in reasonable times important quantities such as bond lengths and angles, ionization potentials, partial atomic charges from population analysis, dipole moments and so on. The project is primarily conceived for a one-semester course in the fourth or fifth year of physical-chemistry curricula (usually corresponding to a graduate level, with some differences from country to country), but extensions to other profiles are possible as well.

The heart of every HF code lies in the technique for evaluating molecular integrals over basis-set functions. Nowadays, professional softwares exploit advanced algorithms<sup>13</sup> which minimize calculation times, but are by no means suitable for didactic purposes due to their high mathematical and programming complexity. On the contrary, the repeated-differentiation algorithm proposed by the British theoretical chemist S. F. Boys (1911–1972) in his milestone 1950 paper<sup>14</sup> (and used here) is computationally inefficient, but simple enough to be understood and implemented directly by students: thus, it can be an extremely precious tool for a successful development of the DIY perspective.

This paper is organized as follows. An overview of the Boys algorithm is first given. The structure of the didactic project is then illustrated, including hour distribution and task partition among student groups for a typical one-semester course. Finally discussed are performances and didactic potentialities of the resulting HF code. Importantly, the online electronic [Supporting Information](#) is intended to be an essential part of the paper and provides a fully tested and commented [Fortran code \(DIYRHF\)](#), which can serve as a reference model in supervising students, as well as [sample worksheets](#), [test data](#), and a [“Mathematical Guidelines”](#) (MG) document with technical details and programming hints for the crucial parts of the code.

## MOLECULAR INTEGRALS AND THE BOYS ALGORITHM

In the HF scheme, one looks for the ground-state wave function of a given  $N$ -electron atom or molecule in the approximate form of a single determinant of  $N$  one-electron wave functions (spin orbitals). In particular, this paper considers only even- $N$  molecules with a closed-shell structure (Restricted Hartree–Fock, RHF): that is, electrons are arranged two by two with opposite spin in  $N/2$  molecular orbitals. These, in turn, can be expressed as linear combinations of the atomic orbitals of a predetermined basis set; the issue is then to find the combination coefficients which minimize the total energy of the molecule for a given nuclear geometry. What results, in the end, is a matrix eigenvalue problem<sup>15</sup> in which matrix elements are determined by molecular integrals<sup>13,15</sup> over basis orbitals.

Four types of integrals must be evaluated: overlap integrals, kinetic-energy integrals, nuclear-attraction integrals (two-center integrals), and electron-repulsion integrals (four-center integrals). The latter ones are the most complicated and have the form:

$$(\mu\nu\lambda\sigma) = \int \int \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \phi_\lambda(\mathbf{r}_2)\phi_\sigma(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \quad (1)$$

Here,  $\phi_\mu$ ,  $\phi_\nu$ ,  $\phi_\lambda$ , and  $\phi_\sigma$  are, respectively, the  $\mu$ -th,  $\nu$ -th,  $\lambda$ -th, and  $\sigma$ -th basis orbitals,  $\mathbf{r}_1$  and  $\mathbf{r}_2$  indicate position vectors ( $x_1, y_1, z_1$ ) and ( $x_2, y_2, z_2$ ), and the integration is extended to the whole six-dimensional space ( $\mathbf{r}_1, \mathbf{r}_2$ ). The definitions of the other integral types are reported in chapter III of the [MG](#). The total number of unique integrals of the form (1) increases very rapidly with the number  $B$  of basis orbitals, approximately as  $B^4/8$  (the division by 8 is due to the index symmetry of [eq 1](#):  $(\mu\nu\lambda\sigma) = (\nu\mu\lambda\sigma) = (\mu\nu\sigma\lambda) = (\lambda\sigma\mu\nu)$ , etc.); as typical examples, minimal calculations on a molecule of water ( $B = 7$ ), benzene ( $B = 36$ ), and glucose ( $B = 72$ ) will require, respectively, 406,222,111 and 3,454,506 integrals. Therefore, the amount of evaluation work needed per integral plays a computationally crucial role.

This quantity is of course strictly dependent on the choice of the basis set. One would be naturally inclined to use orbitals which closely resemble solutions of the one-electron Schrödinger equation around each nucleus of the molecule. In this sense, the most popular ones are certainly the so-called Slater orbitals,<sup>13,15</sup> which are given by (up to a multiplicative normalization constant):

$$\phi_S[\mathbf{A}, P, Q, R, \zeta](\mathbf{r}) = (x - A_x)^P (y - A_y)^Q (z - A_z)^R \exp(-\zeta|\mathbf{r} - \mathbf{A}|) \quad (2)$$

Here,  $\mathbf{A} \equiv (A_x, A_y, A_z)$  is the position vector of the orbital center;  $P, Q$ , and  $R$  are non-negative integers which determine the orbital angular momentum (for instance,  $P = Q = R = 0$  for  $s$ -type orbitals and  $H = 1, K = L = 0$  for  $p_x$ -type orbitals); and  $\zeta$  is a properly chosen positive exponent. Slater orbitals are in fact good basis functions for HF calculations and were extensively investigated until the 1960s, but unfortunately, evaluating integrals of the form (1) with these orbitals proves to be extremely difficult and time-consuming. With a genial intuition, Boys proposed in his famous 1950 paper<sup>14</sup> to approximate Slater orbitals by linear combinations of Gaussian orbitals (contractions):

$$\phi_S[\mathbf{A}, P, Q, R, \zeta](\mathbf{r}) \approx \sum_{m=1}^M d_m \phi_G[\mathbf{A}, P, Q, R, \alpha_m](\mathbf{r}) \quad (3)$$

with

$$\begin{aligned} \phi_G[\mathbf{A}, P, Q, R, \alpha_m](\mathbf{r}) \\ = (x - A_x)^P (y - A_y)^Q (z - A_z)^R \exp(-\alpha_m |\mathbf{r} - \mathbf{A}|^2) \end{aligned} \quad (4)$$

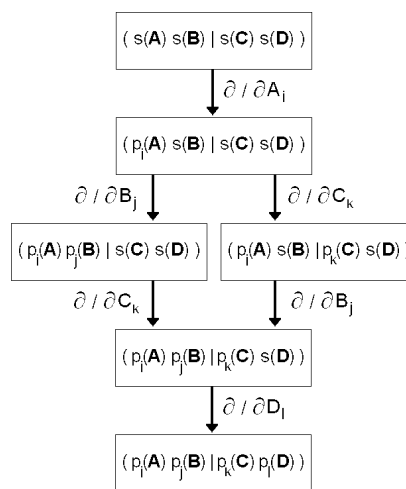
In eq 3,  $M$  is the predetermined number of Gaussian orbitals to be used to reproduce a single Slater orbital: the higher  $M$ , the better the accuracy (in practice,  $M = 3$  suffices in most cases; the famous STO-3G is indeed a minimal basis set of three-term contractions). Contraction coefficients  $d_m$  and exponents  $\alpha_m$  are to be determined according to a proper least-squares fitting procedure.<sup>15</sup> The key point is that molecular integrals over Gaussian orbitals are far simpler to evaluate with respect to Slater orbitals. The introduction of Gaussian contractions was therefore a decisive factor in elevating quantum chemistry to a quantitative computational dimension, and nowadays, their use has become a universal standard in molecular calculations.

How can Gaussian molecular integrals be evaluated? Again in his 1950 work, Boys proposed a conceptually very simple algorithm, which is the core of this paper and is based on the two following points:

- Simple analytical expressions are available (using Fourier-representation techniques) for fundamental integrals involving only s-type Gaussian orbitals.<sup>13</sup>
- As it can be easily verified from eq 4, differentiating a Gaussian orbital with respect to a coordinate of its center yields (up to a multiplicative constant) a new Gaussian orbital of higher angular momentum with the same center and exponent. As an example, differentiating the s-type orbital  $\phi_G[\mathbf{A}, 0, 0, 0, \alpha]$  (more compactly,  $s[\mathbf{A}, \alpha]$ ) with respect to  $A_x$  yields the orbital  $\phi_G[\mathbf{A}, 1, 0, 0, \alpha]$  ( $p_x[\mathbf{A}, \alpha]$ ).

On the basis of these two points above, as illustrated in Figure 1, one can obtain molecular integrals over Gaussian orbitals of arbitrarily high angular momentum by successive analytical differentiation of the correspondent fundamental-integral expressions with respect to the appropriate center coordinates. The resulting formulas can then be used for the separate evaluation of every single integral occurring in the HF scheme (chapter III of the MG reports all necessary formulas for all integral types; however, students should exercise to autonomously derive at least some part of them).

This procedure is a straightforward one, but has very limited efficiency. In fact, integrals involving orbitals with same centers and/or exponents can be shown to share many intermediate quantities, which do not need to be recomputed for every integral; taking advantage of this and other facts (negligibility of many integrals due to their very small absolute values, symmetry of the molecular geometry, etc.) can bring very significant computational savings, especially for large molecules with large basis sets.<sup>13,16</sup> The Boys algorithm was used quite frequently in the 1950s and 1960s, but was then quickly abandoned in favor of more complex and powerful block-evaluation methods. With this regard, it is worth noting that, already in 1970, the first edition of the famous GAUSSIAN software did not employ the raw Boys algorithm but rather the so-called Pople-Hehre algorithm (achieving high performances



**Figure 1.** Illustration of the Boys algorithm for the evaluation of electron-repulsion integrals involving s- and p-type Gaussian orbitals. Starting from the expression of a fundamental integral involving only s-type orbitals with given centers and exponents, any other integral involving same centers and exponents can be obtained (up to a multiplicative constant) by successive analytical differentiation with respect to the appropriate center coordinates. Six irreducible integral classes can be identified. Overlap, kinetic and nuclear-attraction integrals can be evaluated in an analogous fashion.

through a special system of coordinate rotations within every group of integrals involving orbitals with same centers and exponents).<sup>13,17</sup>

Despite that, the Boys algorithm has a unique virtue of tremendous didactic relevance: it is the only molecular-integral method which can be easily understood and implemented by university students into a general program for quantum calculations. On the contrary, all newer algorithms require an extremely specialized mathematical and programming background, and are totally out of reach for students. Therefore, retrieving the “old and dusty” Boys algorithm from the “attic” of quantum chemistry is indispensable in order to achieve an authentic, fully effective DIY dimension in computational courses. This concept is the principal motivation of the present work.

## THE DIDACTIC PROJECT

How to actualize the DIY dimension in the concrete teaching practice? The indications of this section are primarily intended for the setup of a one-semester laboratory course in graduate physical-chemistry curricula (fourth or fifth year of studies), but can certainly be adjusted to other degrees (material science and engineering, theoretical biology, physics, mathematics) and levels (undergraduate, Ph.D.).

### General Considerations

The fundamental aim of the proposed course is to give students the opportunity of a pedagogically active approach to computational quantum chemistry and, in particular, to lead them to an autonomous (as from scratch as possible) implementation of a general RHF program based on the Boys algorithm. The course should consist of two distinct parts:

- A minor, introductory period of frontal lessons about RHF theory and related technical issues (basis sets, molecular integrals, matrix manipulation, etc.).

Table 1. Distribution of Time and Tasks for a Typical 60-h Course with Three Working Groups

Working Group	Period 1 (10 h) Introductory Lessons	Period 2 (50 h) Implementation of the RHF Code <sup>a,b,c</sup>				
		2a (6 h)	2b (16 h)	2c (10 h)	2d (14 h)	2e (4 h)
1	Fundamentals of RHF theory (done by all groups)	Overlap integrals [III] 56	Electron-repulsion integrals [III] 314 (done by all groups)	Building of the basis set [II] 114	Geometry optimization 76	Analysis of some molecules using the final code (done by all groups)
2		Kinetic-energy integrals [III] 59		Assembling of matrix F [IV] 69	Molecular properties 130	
3		Nuclear-attraction integrals [III] 75		Matrices P, C, E and total energy [I, V] 82	Main iteration scheme [I, V] 100	

<sup>a</sup>Details for the crucial implementation tasks are provided in the [Supporting Information](#) (MG). <sup>b</sup>Roman numerals in square brackets indicate chapter numbers in the MG. <sup>c</sup>Arabic numerals correspond to Fortran instruction lines in the code DIYRHF.

- A second, more extended period of laboratory activity. Students are divided into working groups devoted to the implementation of specific subroutines, which are then gradually assembled in the final RHF code.

Necessary prerequisites for a fruitful attendance include, in addition to a basic knowledge of atomic and molecular orbital theory, a good background in mathematics (vectors, matrices, differential and integral calculus) and elementary programming skills. These are all standard topics of undergraduate curricula; however, it is likely that some students will encounter more or less pronounced difficulties while writing their own code portions. Other possible limitations to the implementation project may derive from a too low number of attending students and/or a limited time availability. In such cases, coding workloads can be reduced by directly supplying students with some parts of the final code; if properly measured, such easing will not compromise the DIY “flavour” of the course.

### Partitioning the Implementation Tasks

As better detailed in dedicated textbooks<sup>15</sup> and in chapter I of the MG, the RHF iterative scheme for a specified closed-shell molecule consists of the following steps:

- (1) Choose a proper basis set of atomic orbitals (due to its implementation simplicity, STO-3G is certainly the best choice in a pedagogic context; also, its computational lightness compensates the inefficiency of the Boys algorithm and allows reasonable calculation times).
- (2) Evaluate all necessary molecular integrals over basis orbitals.
- (3) Generate an initial guess for the density matrix **P**.
- (4) Use molecular integrals and **P** to build the Fock matrix **F**.
- (5) Through a proper diagonalization procedure on **F**, obtain an approximation to molecular orbitals (expressed as linear combinations of basis orbitals) in terms of an eigenvector matrix **C**, and to related orbital energies in terms of an eigenvalue matrix **E**.
- (6) Use **C** to build a new matrix **P** and, subsequently, a new matrix **F**.
- (7) Check whether the density matrix and the associated total energy have converged according to suitable predetermined criteria. If not, go back to step (5).
- (8) Once the scheme has converged, use the final forms of **P**, **F**, **C**, and **E** to evaluate all desired molecular properties (such as ionization potentials, partial atomic charges from population analysis, dipole moments, etc.). Also, the RHF calculation can be repeated for different nuclear positions so as to determine the optimal molecular geometry (in terms of bond lengths and angles) which minimizes the total energy.

Matrices **P**, **F**, **C**, and **E** are defined in chapters IV and V of the MG. Every programming language is equally valid in principle for implementing the RHF scheme; however, the use of “good old” Fortran is especially recommendable due to both its extreme syntactic simplicity and the abundance of freely Internet-available, highly reliable libraries of general mathematical utility (in particular, diagonalization and minimization subroutines are, respectively, needed at steps (5) and (8) of the scheme). On the basis of the above steps, a possible partition of implementation tasks for a typical course of 60 h with three working groups is proposed in Table 1.

It is convenient to divide the overall laboratory period into a number of logically unitary subperiods (five subperiods are proposed in Table 1 as a summary guideline, but a different division can of course be chosen depending on time availability and the number of groups). Importantly, the evaluation of electron-repulsion integrals is a crucial part of the RHF program which every student should deal with, therefore the related implementation task should be simultaneously assigned to all groups. Another common subperiod should be envisaged after the completion of the program, in order to highlight its potentialities through a teacher-guided application to some illustrative molecules. On the whole, in the case of Table 1, each group will have to produce about 570 Fortran instruction lines, meaning an average of less than 13 lines per laboratory hour without considering homework time (the number of instruction lines in the code DIYRHF for every single implementation task is indicated in Table 1).

### Organizing and Supervising the Group Work

At the beginning of each subperiod, each group will be provided with proper working documentation: an unambiguous description for all input and output parameters of each subroutine to be implemented, an account of involved mathematical details, programming hints, and finally, test data (that is, expected outputs for given inputs) to check the correct functioning of the finished code. All such material can be found in the electronic [Supporting Information](#) of the paper. Ideally, each group should be followed by a dedicated tutor, which will help students in solving possible programming problems and ultimately evaluate the overall quality of their work. At the end of every subperiod, each group will share their own code with the other ones and hold a brief presentation about its development (logical structure of the code, major difficulties encountered, adopted solutions, etc.).

A final note is for the cooperative dimension of the didactic project. This pedagogically crucial element should be constantly emphasized and stimulated throughout the course. Implementing a general quantum-chemistry program is a demanding aim, which requires students to establish a good collaboration at

both intra- and intergroup levels. Accordingly, the final course mark for each student should be determined by a properly weighted sum of a personal component (deriving from a written and/or oral examination) and a collective component expressing the working quality of his/her group (timeliness in code delivery, accuracy of the code, effectiveness in intermediate presentations, etc.).

Up to now, the full project has not yet been tested in a “real-world” classroom and no comprehensive data are available on related pedagogic outcomes. However, as a preliminary indication, four graduate physical-chemistry students were personally guided by the author to the implementation of the only overlap and kinetic-energy integrals during a minor laboratory course in spring 2014. Students were divided into two pairs (one devoted to overlap integrals and one to kinetic-energy integrals). Both pairs were provided with detailed step-by-step worksheets (as available in the electronic [Supporting Information](#)) and assisted with fully tailored support throughout the whole implementation process. Despite this, only one pair was able to complete its own task within 6 laboratory hours as envisaged in [Table 1](#) (8 h were taken by the other pair; both pairs declared an additional homework time of about 6 h). These results suggest that [Table 1](#) should be regarded as an ideal efficiency standard rather than a strict, rigorous schedule. In any case, the surprising enthusiasm of those few students at the end of the activity (when their code was incorporated into a full RHF program and successfully tested on several molecules) was a powerful incentive to the development of the present work.

## THE FINAL CODE

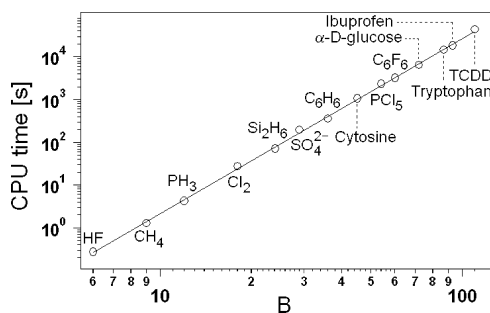
Once the implementation project is successfully accomplished, students will have available a flexible tool for performing quantum calculations in a critical, “open-box” fashion.

### Calculation Times

Though unavoidably far slower with respect to professional softwares, a DIY code based on the raw Boys algorithm can be reasonably quick with not only small- but also medium-size molecules. [Figure 2](#) shows the STO-3G calculation time taken by DIYRHF on a typical present-day home computer vs the number  $B$  of basis orbitals for a selection of inorganic and organic molecules with fixed standard geometries. This time is primarily associated with the evaluation of electron-repulsion integrals (see above the “[Molecular Integrals](#)” section) and, as expected, scales with  $B$  as  $B^4$  (for comparison, the scaling exponent can be reduced up to only 1 (linear scaling) using state-of-art HF algorithms).<sup>16</sup> A time of about 1 s and 6 min is, respectively, needed for methane and benzene, but with some patience one can also face an intriguing variety of larger molecules of biological relevance, such as ([Figure 2](#)) DNA bases (cytosine,  $B = 45$ ), carbohydrates ( $\alpha$ -D-glucose,  $B = 72$ ), amino acids (tryptophan,  $B = 87$ ), pharmaceuticals (ibuprofen,  $B = 93$ , a typical analgesic),<sup>18</sup> and toxic agents (2,3,7,8-tetrachlorodibenzo-*p*-dioxin or TCDD,  $B = 110$ , the contaminant involved in the 1976 Seveso accident).<sup>18</sup> From a pedagogical point of view, taking several hours to perform a calculation of this sort with one’s own DIY program can be far more gratifying than a comfortable few-minutes run on a ready-made professional software.

### Geometry Optimization

An extremely useful feature, which can be included in the implementation project with only a small additional effort, is



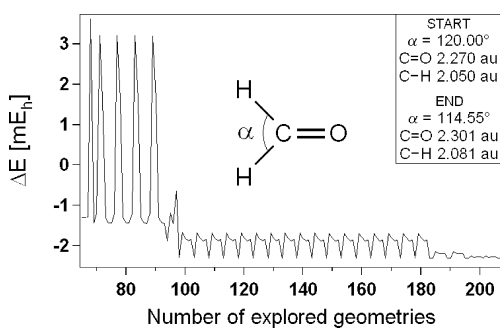
**Figure 2.** CPU time taken by the code DIYRHF for fixed-geometry STO-3G calculations of various molecules, as a function of the number of basis orbitals (bilogarithmic plot). Nuclear coordinates were preliminarily generated using standard values for bond lengths and angles or directly downloaded from the PubChem Internet database<sup>18</sup> in the case of cytosine,  $\alpha$ -D-glucose, tryptophan, ibuprofen, and TCDD. All calculations were carried out on a typical home computer (Asus X58LE, 2.16 GHz, 667 MHz front-side bus, 3 GB RAM, Windows XP Professional). The solid line is obtained by linear fitting and has a slope of  $4.09 \pm 0.03$ . Further details on calculation settings and results are provided in the electronic [Supporting Information](#) (archive DIYRHF\_TEST).

geometry optimization. For students, in fact, investigating the shape of molecules and comparing their own calculations with experimental data is a very exciting and stimulating activity. With this regard, it is worth noting that, despite its minimal character, the STO-3G is often surprisingly accurate in predicting bond lengths and angles (in some cases, more accurate than larger basis sets).

Such good behavior is actually nothing but the lucky result of a cancellation of errors;<sup>19</sup> however, this is of no relevance in a DIY pedagogical context. To perform geometry optimization, a suitable search algorithm is needed to explore the space of nuclear coordinates (starting from an initial guess) and to locate the absolute minimum of the total energy. Professional softwares generally rely on quasi-Newton algorithms,<sup>20</sup> which are very powerful but require the calculation of first and second energy derivatives with respect to nuclear coordinates. Furthermore, implementing the evaluation of these quantities is an exceptionally laborious task and is not recommendable for a didactic project. It is then convenient to turn to direct-search algorithms,<sup>20,21</sup> which are less powerful but have the advantage not to need any energy derivatives. Students should just be asked to take an already available direct-search code of general utility and to properly incorporate it into their RHF program as an optional optimization tool. DIYRHF uses a compass-search<sup>21</sup> code by J. Burkardt,<sup>22</sup> but other algorithms can be chosen as well. As an illustrative example, [Figure 3](#) shows the DIYRHF optimization path in the case of a formaldehyde molecule (three degrees of freedom). Typically, starting from a reasonable initial guess, direct-search algorithms need to explore some hundreds of geometries (and thus to perform as many RHF calculations) before locating the energy minimum to a good accuracy (in the case of [Figure 3](#), a total of 207 geometries were explored before convergence). Therefore, direct-search optimization is in practice applicable only to small-size molecules.

### Molecular Properties

A last point of interest is the determination of molecular properties from the final wave function, which is, in fact, the ultimate aim of any quantum calculation. An immediately



**Figure 3.** Full STO-3G geometry optimization of a formaldehyde molecule using DIYRHF. Initial guess geometry and final optimized geometry are specified in the right inset. The total energy, expressed as variation with respect to the initial value ( $-112.352042 E_h$ ), is plotted vs the number of geometries explored by the program. The first 64 points involve high energy peaks and are not shown for better clarity. The optimization algorithm converged after 207 explorations to a minimum value of  $-112.354343 E_h$ , requiring a total CPU time of 1080 s on a typical home computer (see the caption of Figure 2 for computer specifications). Further details on calculation settings and results are provided in the [Supporting Information](#) (archive DIYRHF\_TEST).

available quantity is the first-ionization potential (coinciding, according to Koopmans theorem,<sup>15</sup> with the absolute value of the energy of the highest occupied molecular orbital). Other significant examples are the Mulliken/Löwdin population analysis<sup>15</sup> for obtaining partial atomic charges and the evaluation of electric dipole moments.<sup>15</sup> The first one is just a straightforward manipulation of matrices **P** and **S**, whereas the second one involves a more demanding task, in that one needs to preliminarily implement the calculation of basis-set matrix elements for the dipole-moment operator. Both features are included in the code DIYRHF (formulas for the dipole-moment integrals are provided in chapter III of the [MG](#)).

The present overview gives only a broad indication on performances and potentialities of the expected DIY program. Depending on students' needs and teachers' preferences, many details can of course be changed, simplified, or even eliminated. The final code can also be employed as a starting material for more advanced DIY courses; didactically feasible extensions include unrestricted (UHF) calculations,<sup>15</sup> the application of external electric fields (so as to investigate geometry deformation and dielectric constants) and the use of larger basis sets (as an unavoidable consequence, calculation times will considerably grow in the latter case: for instance, using even a light basis set such as the 4-31G<sup>15</sup> will increase the calculation time for benzene ( $B = 66$ ) by a factor of about 11 with respect to the STO-3G case ( $B = 36$ ), from 6 min to more than 1 h on a typical home computer).

## CONCLUSION

Among the large variety of techniques for evaluating molecular integrals, the 1950 Boys algorithm is probably the only one which can be easily understood and implemented by university students. Therefore, it can play a crucial role in building a DIY pedagogical dimension for computational quantum chemistry (as opposed to a passive, black-box use of professional software packages).

In this light, the present paper has discussed the very intriguing possibility of leading students to a cooperative, from-scratch implementation of a general-utility RHF/STO-3G code

(capable of treating arbitrary closed-shell molecules with first- and second-row atoms, and optionally enriched with tools such as geometry optimization, population analysis and evaluation of dipole moments).

In 1950s and 1960s, the limited available computer resources made RHF calculations with the Boys algorithm almost prohibitive. As a consequence, this technique was quickly abandoned in favor of more complex and powerful ones. Nowadays, however, technology has made a huge progress. Tests with the code DIYRHF (expressly written as a teacher support, along with a "Mathematical Guidelines" document providing technical details for the crucial parts of the code) show that even a modest home computer can run the Boys algorithm in reasonable times, at least in the case of small- and medium-size molecules. A technically obsolete tool can be thus fully rediscovered in a concrete didactic perspective.

Supervising students in the implementation of their own "DIY GAUSSIAN" may appear a very demanding and even audacious aim, but is certainly an invaluable pedagogical opportunity in terms of reflexivity, interdisciplinarity and gratification. In any case, the structure of the didactic project is highly flexible and can be easily adjusted and redimensioned to properly meet students' needs and teachers' preferences.

## ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available on the [ACS Publications website](#) at DOI: [10.1021/acs.jchemed.5b00317](https://doi.org/10.1021/acs.jchemed.5b00317).

Sample worksheets for implementing the STO-3G basis set and two-center integrals (archive DIYRHF\_WORKSHEETS) ([ZIP](#))

Fully commented Fortran code of the program (archive DIYRHF\_CODE) ([ZIP](#))

Test data for the crucial parts of the code and for the overall program (archive DIYRHF\_TEST) ([ZIP](#))

List of Supporting Information materials included (README\_FIRST) ([PDF](#))

Mathematical guidelines to the implementation of a general RHF program using the Boys algorithm ([PDF](#))

## AUTHOR INFORMATION

### Corresponding Author

\*E-mail: [orsiniga@libero.it](mailto:orsiniga@libero.it).

### Notes

The authors declare no competing financial interest.

## REFERENCES

- (1) Rogers, D. W. *Computational Chemistry Using the PC*; Wiley: Hoboken, 2003; Chapters 8–10.
- (2) Pearson, J. K. Introducing the Practical Aspects of Computational Chemistry to Undergraduate Chemistry Students. *J. Chem. Educ.* **2007**, *84*, 1323–1325.
- (3) Withnall, R.; Chowdhry, B. Z.; Bell, S.; Dines, T. J. Computational Chemistry Using Modern Electronic Structure Methods. *J. Chem. Educ.* **2007**, *84*, 1364–1370.
- (4) Feller, S. E.; Dallinger, R. F.; McKinney, P. C. A Program of Computational Chemistry Exercises for the First-Semester General Chemistry Course. *J. Chem. Educ.* **2004**, *81*, 283–287.
- (5) Clauss, A. D.; Nelsen, S. F. Integrating Computational Molecular Modeling into the Undergraduate Organic Chemistry Curriculum. *J. Chem. Educ.* **2009**, *86*, 955–958.

- (6) Johnson, L. E.; Engel, T. Integrating Computational Chemistry into the Physical Chemistry Curriculum. *J. Chem. Educ.* **2011**, *88*, 569–573.
- (7) Simeon, T.; Aikens, C. M.; Tejerina, B.; Schatz, G. C. Northwestern University Initiative for Teaching NanoSciences (NUITNS): An Approach for Teaching Computational Chemistry to Engineering Undergraduate Students. *J. Chem. Educ.* **2011**, *88*, 1079–1084.
- (8) Ruddick, K. R.; Parrill, A. L.; Petersen, R. L. Introductory Molecular Orbital Theory: An Honors General Chemistry Computational Lab As Implemented Using Three-Dimensional Modeling Software. *J. Chem. Educ.* **2012**, *89*, 1358–1363.
- (9) Jensen, J. H.; Kromann, J. C. The Molecule Calculator: A Web Application for Fast Quantum Mechanics-Based Estimation of Molecular Properties. *J. Chem. Educ.* **2013**, *90*, 1093–1095.
- (10) Frankel, E. C.; Davis, D. D. A Computer Assisted Experiment in Quantum Mechanics. *J. Chem. Educ.* **1973**, *50*, 80–81.
- (11) Hoffman, G. G. Self-Consistent Field Calculations on Atoms Using Excel. *J. Chem. Educ.* **2005**, *82*, 1418–1422.
- (12) Page, T. R.; Boots, C. A.; Freitag, M. A. Restricted Hartree-Fock SCF Calculations Using Microsoft Excel. *J. Chem. Educ.* **2008**, *85*, 159.
- (13) Gill, P. M. W. Molecular Integrals over Gaussian Basis Functions. *Adv. Quantum Chem.* **1994**, *25*, 141–205.
- (14) Boys, S. F. Electronic Wave Functions. I. A General Method of Calculation for the Stationary States of Any Molecular System. *Proc. R. Soc. London, Ser. A* **1950**, *200*, 542–554.
- (15) Szabo, A.; Ostlund, N. S. *Modern Quantum Chemistry*; Dover: New York, 1996; Chapter 3.
- (16) Jensen, F. *Introduction to Computational Chemistry*; Wiley: Hoboken, NJ, 2007; Parts 3.8.5 and 3.8.6.
- (17) Pople, J. A.; Hehre, W. J. Computation of Electron Repulsion Integrals Involving Contracted Gaussian Basis Functions. *J. Comput. Phys.* **1978**, *27*, 161–168.
- (18) PubChem Compound database. <https://pubchem.ncbi.nlm.nih.gov> (accessed Apr 2015).
- (19) Davidson, E. R.; Feller, D. Basis Set Selection for Molecular Calculations. *Chem. Rev.* **1986**, *86*, 681–696.
- (20) Zerner, M. C. Analytic Derivative Methods and Geometry Optimization. In *Modern Quantum Chemistry*; Dover: New York, 1996; Appendix C.
- (21) Kolda, T. G.; Lewis, R. M.; Torczon, V. Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Rev.* **2003**, *45*, 385–482.
- (22) Burkardt, J. COMPASS\_SEARCH: The Compass Search Optimization Algorithm. [http://people.sc.fsu.edu/~jburkardt/f77\\_src/compass\\_search/compass\\_search.html](http://people.sc.fsu.edu/~jburkardt/f77_src/compass_search/compass_search.html) (accessed Aug 2015).